

EmberZNet Stack Release Notes

Product: Release Notes for EmberZNet 5.6.0 GA

Release Date: June 24, 2015

1 Overview

This release contains the GA release of EmberZNet PRO 5.6.0.

It contains the following:

- Hardware Abstraction Layer (HAL) software provided as source code
- ZigBee PRO Networking stack
- Ember ZCL Application Framework (AFV2) including Sample Applications
- ISA3 Hardware and Programming utilities and hardware
- Ember Desktop software for network debugging
- InSight AppBuilder Version 2 software for rapid ZigBee Application Development
- Hardware Reference Designs and Datasheet
- Application, Serial-uart-only and Serial-uart-ota Bootloaders
- Full set of Documentation including API guides and Application Notes

1.1 Software Version

Stack Software: Version 5.6 build 231. The version number is defined as SOFTWARE_VERSION in the source code in the "stack/config/config.h".

This stack software release is intended to be used with the following versions of the other components:

- Ember Desktop 3.3.1883
- ISA3 Utilities Version 4.5.0.0 build 81
- IAR Embedded Workbench for ARM 7.30.1
- ECC Library 1.0.1.20 (Available upon request via Ember support)

1.2 What's New

1.2.1 What was changed in EmberZNet 5.6.0:

- Certifiable ZigBee Smart Energy 1.2a application layer support.
- Serial boot loading via USB port on EM3582, EM3586, EM3588, EM3592, EM3596 and EM3598 variants.
- Full-featured SIMEEv2 (Simulated EEPROM) with increased token storage, including the ability to upgrade a device from SIMEEv1 to SIMEEv2.
- Enhanced network scalability features including improved MTORR and Broadcast handling in large networks.

- Network Coprocessor support on EM342.
- Support for changing node type between Sleepy and non-sleepy End Device during a rejoin. See `emberFindAndRejoinNetworkWithNodeType()` API documentation in `stack/include/network-formation.h`
- Buffered TX serial mode in `serial.c` driver is being deprecated and is planned for removal in a future release. Please migrate any Buffered Serial applications to use FIFO serial in anticipation of this change.

2 Using This Release

2.1 Development Kit Installation

Refer to the Quick Start Guide included in your Development Kit for detailed step-by-step instructions about initially setting up your kit. See section 3 - Installation of these release notes for detailed instructions on installation.

2.2 Support

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

2.3 Accessing Documentation

- To view the full set of documentation, click on:
Start → All Programs → Ember → Ember Stack 5.6.0 → Documentation
Or open the "documentation/index.htm" file from the root directory of the installed Ember stack. The documentation includes HTML pages describing the Ember Stack API and the Ember HAL API, along with a Development Kit Users Guide, an Application Developers Guide, the chip data sheet, and various Application Notes.
- **To view information on the sample applications, click on:**
Start → All Programs → Ember → Ember Stack 5.6.0 → Samples
Or open the "app/sampleApps.html" file from the root directory of the installed Ember stack. This includes explanations of how the sample applications work, what commands are supported, notes, and limitations. It also includes shortcut links to open the project files. IAR Embedded Workbench for ARM must be installed before these project links will work properly.
- **To view information on the ISA3 and associated utilities, click on:**
Start → All Programs → Ember → ISA3 Utilities → Documentation
Or open the "docs/index.html" file from the root directory of the installed ISA3 Utilities. This includes explanations of how to use the ISA3 Command line utilities, how to program certificates and install codes, as well as general documentation about the operation and use of the ISA3.
- **To view documentation for IAR Embedded Workbench, click on:**
Start → All Programs → IAR Systems → IAR Embedded Workbench for ARM → Release Notes

2.4 The File Layout

When the Ember stack software is installed, the following directories are present:

- **app**: contains sample application utilities and source code.
 - **app/framework**: contains source files for Application Framework Version 2. This code is configured using the AppBuilder GUI Tool to build applications that implement ZigBee application profiles such as ZigBee Smart Energy (SE) and Home Automation (HA) profiles. This directory also includes a sample-apps folder containing InSight Configurator (isc) files to build
 - **app/nodetest**: contains pre-built hex images of the nodetest image packaged with a bootloader. Additional NodeTest binaries can be found in the build/nodetest-... subdirectories for specific chip variants.
 - **app/util**: contains useful utility libraries for application development.
- **build**: contains the pre-built stack libraries. Note that when based non-application-framework based sample applications are built (such sink, sensor, sleepy-sensor, mobile-sensor), their resulting binaries are in a subdirectory under this directory.
- **em35x-ezsp-images**: contains network-coprocessor binaries and host image samples for EM35x-EZSP applications
- **documentation**: contains documentation files. Open "documentation/index.htm" to see a list of all documentation files.
- **hal**: contains the HAL source code
- **stack**: contains header files necessary to build EmberZNet applications.
- **tool**: contains the AppBuilder configuration settings, bootloader images and project files and bootloader Java application.
 - **tool/appbuilder**: contains project templates, xml files and properties files for Application Builder tool.
 - **tool/image-builder**: contains image-builder upgrade file used by ZigBee OTA Bootload cluster.
 - **tool/bootloader-em{Chip Designation}**: contains the pre-built bootloader images and project files for building bootloaders with custom modifications.
 - **tool/usb-host-drivers**: contains drivers to allow customers to use USB to communicate with an EM358x product.

2.5 Selecting a Stack for AppBuilder

When using AppBuilder to generate projects it is important to select an installed stack to build those projects against. AppBuilder uses information contained in the installed stack (from tool/appbuilder) to expose features that are supported by that particular stack. To add a stack for AppBuilder to use:

1. Select → File Preferences
2. Select AppBuilder from the left side of the dialog box that appears
3. Click on the "Add" button
4. Navigate to the directory that your stack is installed in (default is C:/Users/*Username*/ember/EmberZNet5.6.0-GA/) and click OK.

When creating a new AppBuilder project, a dialog box appears asking the user to select a stack to use for this project. To change the stack version use the drop down box just above the "Add Stack Version" button on the AppBuilder Preferences window. When creating a new configuration (using File → New → Application Framework Configuration) the stack that is being used for this configuration is listed in the upper left corner. The stack version must be changed prior to starting a configuration, open configurations are not changed to a new stack version if the stack version is changed.

3 Installation

3.1 Install Ember Desktop

Ember Desktop can be installed by running the installer. The Quick Start Guide included in your Development Kit provides details about installing Ember Desktop for the first time and running the First-Time Setup Wizard. Note: Only the latest Ember Desktop (version mentioned in section 1.4) or later version can be used with this stack version. **Note:** Ember Desktop must be used with a 32-bit JRE. If you are using a 64-bit system, you must install a 32-bit JRE.

3.2 Install the Ember Stack

The Ember Stack installer should be installed first, and performs the following actions:

- Installs the files needed to develop applications on the Ember Stack into a directory specified by the user (The default location is C:/Users/*Username*/ember/EmberZNet5.6.0-GA/ on Windows 7 and C:/Documents and Settings/*Username*/ember/EmberZNet5.6.0-GA/ on Windows XP).
- Adds a shortcut to the Start Menu into the folder specified by the user (default directory is ember/EmberZNet5.6.0-GA/).
- Adds an entry to “add/remove programs” (accessed from Windows Control Panel) so the stack can be uninstalled
- This does not affect any environment variables

This release should be installed in a different directory from prior installations. It is possible for multiple installations to exist together

3.3 Install IAR Embedded Workbench for ARM

- A CD including IAR Embedded Workbench for ARM is included with your Development Kit. Depending on the type of Development Kit purchased, this may be a 30-day evaluation CD, or a fully licensed copy of EWARM.
- Insert the CD into your CD Drive and follow the instructions on the installer that should automatically appear. If the Installer does not automatically start, browse to your CD Drive, and click on autorun.exe.
- Refer to the “QuickStart Installation Information” section of the IAR installer for additional information about the installation process and how to configure your license.
- If you are receiving this release as a product upgrade, the installer for IAR EWARM 7.30.1 is available at <ftp://files.iar.com/pub/silabs/EWARM-7301.exe>

3.4 Install the ISA3 Utilities

The installer will perform the following actions:

- Installs the Ember InSight Adapter drivers so that the InSight Adapter can be used both over USB and Ethernet
- Installs the following command line utilities:
 - em3xx_buildimage.exe
 - em3xx_load.exe
 - em3xx_isa.exe
 - em3xx_convert.exe

– stm32_load.exe

- Adds an entry to “add/remove programs” (accessed from Windows Control Panel) so that the utilities can be uninstalled.
- Modifies your PATH environment variable so that the command line utilities can be easily executed from a Windows Command Prompt.

Note: The ISA3 Utilities installer should always be run after all previous installers, including after IAR Embedded Workbench for ARM.

3.5 Install FTDI USB Drivers

In order to use the USB interface of the breakout board for UART connectivity, drivers for the FTDI USB ↔ Serial converter must be installed. The driver can be acquired from <http://www.ftdichip.com/Drivers/VCP.htm>

4 Known/Fixed Issues

4.1 Fixed Issues

4.1.1 Fixed Issues in EmberZNet 5.6.0

- N/A

4.2 Known Issues

- Issue 135649: Multi-networking can cause APS frame counter confusion between networks. Workaround: Use emberAfSecurityInitCallback to add EMBER_NO_FRAME_COUNTER_RESET to EmberInitialSecurityBitmask.
- Issue 90870: Sleepy End Device sends Network Leave (Announce) but not Leave Response.
- Issue 122276: Sleepy-sensor NCP devices draw too much current after performing a rejoin.
- Issue 121984: Turning off NCP concentrator support does not disable all stack concentrator logic.
- Issue 119939: ZDO IEEE Request’s APS ACK proxied by parent incorrectly includes long source address.
- Issue 119037: Packet-buffer.c Assert at line 352 occurs during rapid packet transmission and Partner-link-key-exchange.
- Issue 113976: EZSP command functions does not always properly parse arrays if sendCommand() aborts
- Issue 113243: Parent router isn’t checking against child IDs when assigning IDs to new nodes
- Issue 105181: Sleepy End Device with no serial ports configured does not decrease current to sleep level.
- Issue 103833: Second energy scan request caught in first energy scan request knocks node out of network.
- Issue 92638: Assert when sending reports to non-responding device.
- Issue 82600: Setting a non-zero MAC Filter Table Size but no MAC Filter Table Entries causes NCP resets when joining a network

- Issue 61008: Scanning state machine (stack level or form-and-join util) can get stuck in "scanning" state indefinitely
- Issue 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.
- Issue 60970: TC link key should be used for Transport Key to rejoining devices even if decision is Send Key In Clear
- Issue 60944: EZSP-SPI NCP may become unresponsive if callbacks are received during ECC operations.
- Issue 60868: Extra retries seen on ZDO requests; ZDO response going out before APS ACK.
- Issue 60858: Sleepy broadcast payload is sometimes corrupted when relaying to child
- Issue 60774: MTORR reception by concentrator neighbor sometimes causes assert in route-discovery.c (emHandleRouteCommand).
- Issue 60757: Indirect Transaction Expiry route error not being sent when TX failure count is reached on parent router.
- Issue 90325: If you program an image onto an EM35x device using JTAG, the device must be power cycled after programming in order to for the device to enter deep sleep. If Serial Wire is used as the programming interface than a power cycle is not required after programming.
- Issue 62126: Packet stream and non-Packet debug stream should be better synchronized by ISA3 in Ember Desktop event capture.
- Issue 106307: Nodetest calChannel command does not wake the radio to work properly.
- Issue 87700: Sometimes the MAC ACKs associated with a sleepy device's transmissions get a false CRC failure due to timing issues of PTI (packet trace interface) and the software when the radio is shutting down as the MAC ACK arrives, resulting in wrong interpretation of CRC while in fact it is correct on the chip.
- Issue 82812: Timer Registers naming inconsistent in regs.h
- Issue 71020: emberSetRadioPower no longer defined in bootloader PHY context, use emSetPhyRadioPower() as a workaround in the bootloader.
- Issue 70851: System-timer implementation precludes runtime adjustment to GPIO wake handling
- Issue 70799: Overlay of RESETINFO on CSTACK causes IAR stack overflow warning

5 Deprecated APIs and Functionality in this Release

- Mobile End Device Support
In the next release, we will no longer support Mobile End Devices. Mobile End Devices are an Ember proprietary feature that is not ZigBee Pro compliant.
- Non-Appbuilder based sample applications
In the next release, we will no longer provide non-Appbuilder based sample applications, such as Sensor and Sink. Instead, there will be Appbuilder-based sample applications to cover this functionality.
- EM351 NCP Support
In the EmberZnet 5.6 release, we will no longer support the EM351 as an NCP platform. All users running NCP applications on an EM351 are recommended to upgrade to an NCP supported platform.

- Switching to standard variable types `<stdint.h>` and `<stdbool.h>`
In the next release, we will switch the code base over to the standard variable types defined by `<stdint.h>` and `<stdbool.h>`. The existing variable types will remain defined to assist with backwards compatibility.

```
int8u → uint8_t
int8s → int8_t
int16u → uint16_t
int16s → int16_t
int32u → uint32_t
int32s → int32_t
boolean → bool
TRUE → true
FALSE → false
```

The `<stdint.h>` types are compatible with the existing types.

The `<stdbool.h>` types are **not** compatible with the existing boolean type. The underlying definition of `boolean` is an 8 bit value (`int8u/uint8_t`). A compiler's definition of `bool` as defined by `<stdbool.h>` is not an 8 bit value. This means the compiler will probably generate an error if code tries to intermix `bool` and `boolean` uses without casting.

6 Ember Desktop/Application Framework

Please refer to the Application Framework release notes for information.

7 Intended Behavior

Here are some commonly reported issues that can be misleading behavior but function as intended:

- Broadcasts with empty payload are not heard through the loopback by the sending node
 - **Problem:** When sending a broadcast with no payload, the sending node does not get an `emberIncomingMessageHandler()` call for the broadcast loopback. Other devices in the network will receive the broadcast as expected.
 - **Workaround:** add a 1 byte payload to a broadcast that the sender expects to hear through the broadcast loopback.
 - **Issue Reference:** 59946
- When a node hears one of its own messages, the `lastHopLqi` and `lastHopRssi` parameters should have invalid values.
 - **Problem:** the last read values for LQI and RSSI are used when a device hears a message from itself
 - **Workaround:** a device should not use LQI and RSSI values when the source address is itself.
 - **Issue Reference:** 59620
- Parent could allow a device rejoining with an in-use ID to cause a node on the network to change its ID
 - **Problem:** If a child attempts a rejoin using a `shortID` that is already assigned to another device, the parent does not allow the rejoin and sends an ID conflict message causing the joined device using the same ID to change its ID.
 - **Workaround:** none, intended behavior.
 - **Issue Reference:** 62472

- A route discovery is not initiated if a source routed unicast is sent with the EMBER_APS_OPTION_FORCE_ROUTE_DISCOVERY_APS option
 - **Problem:** the source route option takes precedence over the route discovery option. This could be setup to return an error, but the designed behavior is to allow the source route option (the route being used is a many-to-one route) to take precedence over the route discovery option.
 - **Workaround:** none, intended behavior
 - **Issue Reference:** 62551

- Route Records are not sent to Trust Center from a router after it reboots
 - **Problem:** Route table data (such as the concentrator flag associated with route destinations that are concentrators) are stored in RAM, which is cleared upon reboot. When the router reboots, it has no Many-to-one Route [MTOR] to the Trust Center [TC], so it has no reason to send Route Records when communicating to the TC. Furthermore, routing to the TC becomes almost impossible because no known route exists; the exception to this is when the router in question is a neighbor of the TC, in which case a one-hop route is inferred by the router once it finds the destination in its neighbor table. However, even in the inferred, one-hop route case, there is no actual route entry and thus no concentrator status to be preserved yet, so no Route Record arrives for the TC to process.
 - **Workaround:** While it is possible for the TC (assuming its running our EmberZNet PRO stack) to infer a similar one-hop route back to the router in the neighboring case and thus route its replies, this wont work for routers that are non-neighbors of the TC. In order to properly correct the situation for all routers, the TC needs to be doing Many-to-one Route Requests [MTORR] periodically to advertise its status as a concentrator. Once the MTORR is received by the router, a route table entry for the TC will be created, and Route Records will resume when unicasting to the TC.
 - **Issue Reference:** 60750