

Release Notes

Product: EM2420 Developer / Jump Start Kit, EmberZNet PRO 3.3.3 GA

Release date: April 14th, 2009

1 Overview

This release contains the GA release of EmberZNet 3.3.3 for the EM2420/AVR which includes:

- Pre-built ZigBee Networking stack.
- Hardware Abstraction Layer (HAL) software to support the EM2420
- Bugfixes from EmberZNet 3.3.x
- InSight Desktop Software and Ember Backchannel for network debugging.
- Sample Applications and Utilities for the EM2420 provided as source code.
- Standalone-bootloader and associated sample application for the AVR ATmega128 that allows bootloading of devices running either V1 or V2 bootloader.
- Documentation and Application Notes for the EM2420.

1.1 Software Versions

The stack software is version 3.3 build 43. The version number is defined as `SOFTWARE_VERSION` in the source code and can be found in `"stack/config/config.h"`. InSight Desktop version 2.0 is included along with the stack release. See the Quick Start Guide for information on installing this software.

1.2 Accessing Documentation

To view the full set of documentation, open the "**documentation\index.htm**" file from the directory of the unpacked Ember stack. The documentation includes HTML pages describing the Ember Stack API and the Ember HAL API, along with a Developer Kit User's Guide, an Application Developer's Guide, the EM2420 and EM250 data sheet, and various Application Notes.

To view information on the sample applications, open the "**app\sampleApps.html**" file from the the unpacked Ember stack. This includes explanations of how the sample applications work, what commands are supported, notes, and limitations.

1.3 The stack layout

When the ember stack software is unpacked, there are five directories:

- **hal**: contains the HAL source code.
- **stack**: contains the included files needed to link with the pre-compiled stack libraries.
- **app**: contains sample application utilities and source code.
- **tool**: contains pre-built bootloaders (application & standalone), backchannel server firmware image, etrim tool used during the build process, and java sample code.
- **documentation**: contains all documentation files.

1.4 The IAR compiler

EmberZNet 3.3.x software release is designed to use IAR compiler EWAVR 4.10B or EWAVR 4.20D. Note that IAR project files are setup for use with 4.10, since the 4.20 project files are incompatible. When using 4.20 you will see an error asking to upgrade the project file to the new version. This is safe to do.

1.5 A note on security and routing

Security requires using many-to-one routing to the trust center ensure that key exchanges can happen correctly. Many to one routing ensures that the parent of any joining device has a route to the trust center and that parents send route records on behalf of their joining children which means the trust center will have a source route back to the joining device. If a network does not use many to one routing and has devices that are more than 2 hops from the trust center, joining will be unreliable. Looking at an ISD trace will show that in LOW_RAM_CONCENTRATOR mode, route records are sent before each message that is sent to the trust center. This is expected.

1.6 Using EZSP

Before using the EM2420 as a network co-processor (using EZSP) it is important to check the value of the token `TOKEN_MFG_ASH_CONFIG`. The default value for the 40 bytes of this token is “FF”. Some EM2420s are set with the value of “00”, which will prevent EZSP from running. If the values are incorrect, use Rangetest to set the value of the token.

2 Known Issues

- IAR Compiler Issue in 4.10B:
 - Problem: The IAR Compiler will sometimes pass incorrect arguments to the linker, which will cause the following error during the link step:
 - `Error[e12]: Unable to open file '\'`
 - This error occurs most frequently with the simple-lighting-light and simple-lighting-switch Ember sample applications.
 - Workaround: use IARWB 4.20D
- Case 7571: `emberSendReply()` requires a payload
 - Problem: If `emberSendReply()` is used with an empty payload, the resulting message will not trigger an `emberIncomingMessageHandler()` on the device that is meant to receive the message.
 - Workaround: add a minimum of a 1 byte payload when using `emberSendReply()`
- Case 7426: em2420 UART using EZSP `mfglib` asserts when receiving a packet >127 bytes
 - Problem: when an EM2420 EZSP-UART device is in manufacturing mode (has called `mfglibStart` with a value of 1), and receives a packet of 127 or 128 bytes (including 802.15.4 length byte), the device will assert. Sending packets up to the 802.15.4 maximum works (128 – including length byte).
 - Workaround: use `mfglib` in a controlled environment and do not send packets larger than 126 bytes.
- Case 7090: 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.
- Case 6099: it takes at least 2 seconds for a node to leave the network.
 - Problem: ZigBee requires devices to send a Leave message when they are leaving a network. In order to ensure that this message is sent out, devices wait 2 seconds.

- Workaround: none
- Case 6207: When a node "hears" one of its own messages, the lastHopLqi and lastHopRssi parameters should have invalid values.
 - Workaround: a device should not use LQI and RSSI values when the source address is itself.
- Case 6790: parent could allow a device rejoining with an in-use ID to succeed
 - 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 it's ID.
 - Workaround: none, intended behavior.
- Case 7932: 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.

3 Bugs fixed in 3.3.3

3.1 Bugs Fixed in The Stack / NCP image

- Bug 10350: Remove cancelMessage EZSP frame from the NCP
- Bug 10238: Delete child if an incoming message reveals that it has moved
- Bug 10308: buffer leak in End Device Bind on the Coordinator where the contents of the second request were stored but never released.
- Bug 10117: Parent should delete any route entries to a device that has joined or rejoined as a child. Device announce messages should cause all devices to delete routes to that new (or rejoined) device.

4 Bugs fixed in 3.3.2

- Bug 9383: [NCP: ASH only] should not disconnect when it receives certain erroneous frames: do not disconnect on receiving a frame with an invalid ACK number or an illegal control byte. Instead reject the frame and send a NAK. ACK timeouts are the only cause for disconnecting.

- Bug 9920: [Sensor/Sink] added help menu back into Sensor-debug since we now have the space
- Bug 9974: [Stack] emberStartScan() should not return success if the scan started but failed and stopped before returning from emberStartScan(). This was confusing emberFindAndRejoinNetwork(), which assumed that a return of EMBER_SUCCESS meant that it could expect the scan complete handler to be called later.
- Bug 10045: [Doxygen] Correct the doxygen for emberSetReplyFragmentData().
- Bug 10102: [NCP: ASH only] Wrong usage of write() corrupts data on serial error
- Bug 10135: Make emberScanForJoinableNetwork() check for an all-zero extended PAN ID.
- Bug 10160: ZDO IEEE request to your child from a parent leaks two packet buffers
- Bug 10206: If a child polls twice for an indirect message (because it doesn't hear the first ack), the second ack does not have the frame pending bit set.