
EMBER[®] EM35X DEVELOPMENT KIT USER GUIDE

This document describes the Ember EM35x Development Kit and explains how to set up its hardware and use the Ember Debug Adapter (ISA3).

New in This Revision

EM35x module manufacturer and part number changes.

Contents

1	About This Guide	4
1.1	Purpose	4
1.2	Audience	4
1.3	Documentation Conventions	4
2	Introducing Ember Development Kits	4
2.1	Overview	4
2.2	Development Kit Contents	5
2.3	Hardware Requirements	6
2.4	Software Requirements	6
2.5	Hardware	6
2.6	Software	6
2.6.1	Ember Desktop	6
2.6.2	IAR Embedded Workbench	7
2.7	Documentation	7
2.8	Before You Begin	7
3	Development Kit Components	8
3.1	Overview	8
3.2	Hardware	8
3.2.1	EM35x Breakout Board	9
3.2.2	EM35x Module	10
3.2.3	Debug Adapter (ISA3)	11
3.3	Acceptable Power Sources for Normal Operation	11
3.3.1	Power-Over-Ethernet (PoE) Switch	11
3.3.2	USB Power Supply with Adapters	13
3.4	Software	13

UG110

3.4.1	Network Stack	13
3.4.2	Hardware Abstraction Layer.....	14
3.4.3	Sample Applications.....	15
3.4.4	Ember Desktop.....	15
4	Setting Up Hardware Components.....	16
4.1	Overview.....	16
4.2	EM35x Radio Communications Module (RCM)	17
4.3	Hardware Connections	17
4.4	Connecting the Breakout Board and RCM	17
4.5	Breakout Board Power Configurations	19
4.6	Breakout Board Serial Configurations.....	21
4.7	Common Hardware Configurations	22
4.8	Connecting the RF Cable to the RCM.....	25
5	Installing the Software	26
5.1	Overview.....	26
5.2	Installing the Ember Stack.....	26
5.3	Installing Ember Desktop.....	26
5.4	Installing IAR Embedded Workbench for ARM	26
5.5	Installing the Debug Adapter (ISA3) Utilities	26
5.6	Installing FTDI USB Drivers	27
6	Using the Debug Adapter (ISA3)	27
6.1	Overview.....	27
6.2	Connectivity and Power.....	27
6.3	Ethernet Settings	28
6.4	Using the Admin Interface over USB.....	28
6.5	Using the Admin Interface over Ethernet	28
6.6	Using the Serial Ports.....	28
6.7	Updating the Firmware	29
7	Debug Adapter (ISA3) Commands.....	30
7.1	Syntax Conventions.....	30
7.2	Entering Commands.....	30
7.3	Command List	30
7.3.1	button	30
7.3.2	debug	31
7.3.3	config.....	31

7.3.4	connections	31
7.3.5	help.....	31
7.3.6	hostname.....	32
7.3.7	ip.....	32
7.3.8	port	32
7.3.9	power.....	32
7.3.10	reset	33
7.3.11	usbaddr	33
7.3.12	version.....	33
8	Software Tools Overview.....	34
8.1	Using the EM35x Nodetest Application	34
8.2	Running a Packet Error Test	35
8.3	Testing Deep Sleep Current.....	36
8.4	Example of Opening, Building, and Programming a Sample Application	38
8.5	Programming your Application on the EM35x	39
8.6	Programming Manufacturing Tokens	39
8.6.1	Programming Certificates & Installation Codes	39
8.7	Using the Debugger in IAR Embedded Workbench with an Ember Debug Adapter (ISA3)	39

1 About This Guide

1.1 Purpose

This document describes the Ember EM35x Development Kit and explains how to set up its hardware and use the Ember Debug Adapter (ISA3). For information on developing and debugging applications with Ember Desktop, see its on-line HTML help.

For information on designing and debugging an application based on Silicon Labs' EmberZNet PRO software stack, refer to documents UG103.3, *Ember Application Development Fundamentals: Design Choices*, UG104, *Testing and Debugging Applications for the Ember EM2xx and EM35x Platforms*, and 120-035X-000, the *EM35x Data Sheet*. See section 2.7, Documentation, for a list of other EM35x documents of interest.

1.2 Audience

This document is intended for embedded software and hardware engineers who are responsible for developing wireless applications with the Ember Development Kit. This document assumes that the reader has a solid understanding of embedded systems design and programming in the C language. Experience with networking and radio frequency systems is useful but not expected.

1.3 Documentation Conventions

Notation	Meaning	Example
<i>Italics</i>	Identifies on-screen software menu options.	<i>Connect</i>
UPPERCASE	Identifies a keyboard key.	ENTER
Right-angle bracket	Delimits a series of software program menu options to be clicked.	<i>Open > Save</i>
Courier	Identifies software code and, in body text, variables.	<code>void Main(String[] argv) the buffer variable</code>
<i>Courier Italics</i>	Identifies a variable name.	<i>ipAddress</i>

2 Introducing Ember Development Kits

2.1 Overview

The EM35x Development Kit is used for application design and development. The Development Kit contains a variety of hardware for quickly creating a prototype of the customer's hardware, EmberZNet PRO, Silicon Labs' implementation of the ZigBee Pro stack software specification, and various development tools which enable the development of the customer's application software.

EmberZNet PRO runs on top of an IEEE 802.15.4-compliant radio and link layer software. EmberZNet PRO is a self-organizing, self-healing mesh networking protocol stack. Combined with the powerful IEEE 802.15.4 compliant Ember radios, the EmberZNet network stack provides complete networking services from the physical layer up to a reliable application profile support layer. The stack produces networks that are reliable, flexible, secure, and easy to use.

EmberZNet PRO¹ provides a common Application Programming Interface (API) that utilizes the underlying layers. This API provides support for the following layers of the Open System Interconnection Reference Model (OSI Model):

- **PHY:** radio control
- **MAC:** medium access
- **LINK:** route discovery
- **NETWORK:** routing, association
- **APPLICATION:** Application Profile Support for reliable message delivery

The EmberZNet PRO stack can be used to build ZigBee-based applications. Applications that are built on this stack benefit from the following features:

- PHY and MAC layers comply with the IEEE 802.15.4-2003 standard
- LINK and NETWORK layers comply with the latest ZigBee specification
- The APS layer provides the interface to the application for data transfer and management, including reliable transport and binding.

The Development Kit includes Ember Desktop, which facilitates application development and debugging. Debug Adapters (ISA3s) incorporate Power-over-Ethernet (PoE) capability, which can simplify node deployment in the test environment. Breakout Boards provide a direct connection for debugging customer-designed hardware. For detailed information about the Ember customer hardware debug interface, see document TS6, the *EM35x Breakout Board Technical Specification*.

2.2 Development Kit Contents

Table 1 summarizes the components and the specified count for each component in the EM35x Development Kit.

Table 1. EM35x Development Kit Components

Component	EM35x Development Kit
Debug Adapter (ISA3)	3
EM357 Sample Chip	10
EM35x Breakout Board	3
EM35x Modules	9 ²
Data Emulation Interface (DEI) Cable	3
Happy Fun Ball	1
Packet Trace Port Cable	3
Radio Frequency Cable	1
Power-over-Ethernet Switch with Power Supply	1
USB Power Supply with Adapters	3
<i>EM35x Quick Start Guide</i>	1
Ember Desktop CD	1
IAR Embedded Workbench CD	1
US-to-multinational AC Plug Power Adapter Kit	1
AAA Battery Pack	3
Ethernet Cable	3
USB Cable	3

¹ The release version of EmberZNet PRO at the time of this writing is 4.7.

² Three EM35x modules are installed on EM35x Breakout Boards; six EM35x modules are included in the EM35x Module Box.

2.3 Hardware Requirements

- PC with Ethernet connectivity
- Network hub with DHCP server

2.4 Software Requirements

- Microsoft Windows XP or Windows 7.0
- Java, version 1.5V6 or later
- Adobe Acrobat Reader (available free from <http://get.adobe.com/reader/>)

2.5 Hardware

- **Breakout Boards** serve as test beds for network hardware and applications. Each Breakout Board has an associated EM35x Module, which can be deployed as a node within a network application. Each Breakout Board connects with a Debug Adapter (ISA3) through two communication links: the Packet Trace Port and the DEI Port. This allows the Breakout Board's EM35x Module to interface with Ember Desktop through the Debug Adapter (ISA3)'s Packet Trace Port and its wired Ethernet connection. These interconnections are described later in this chapter. The Breakout Boards also have a prototype breadboard area for application-specific hardware development.
- **EM35x modules** contain the Ember EM35x System-on-Chip (SoC) solution providing a complete radio and microcontroller solution. All modules are pin-compatible, and they can be directly attached to a Breakout Board.
- **Debug Adapters (ISA3s)** each connect a Breakout Board to the Ethernet. Each adapter transmits network data collected through its Packet Trace Port and conveys it over its Ethernet connection to Ember Desktop. It also picks up any messages or new software that is addressed to this Breakout Board.

The Packet Trace Port is a cabled communication link between the Debug Adapter (ISA3) and the EM35x Module mounted on the Breakout Board.

- **Packet Trace Port Cables** connect each Debug Adapter (ISA3) to a Breakout Board.
- **AAA Battery Packs** that attach to a Breakout Board enable deployment of battery-operated nodes.
- **USB Power Supply with Adapters** power Breakout Boards and / or Debug Adapter (ISA3)s from wall sockets.
- **Ethernet Cables** connect Debug Adapter (ISA3)s to an Ethernet subnet.
- **Data Emulation Interface (DEI) cables** provide a peripheral interface connection between the Debug Adapter (ISA3) and Breakout Board. The DEI cable supports UART functionality.

2.6 Software

The Ember Development Kit contains two CDs for the following:

- Ember Desktop
- IAR Embedded Workbench

2.6.1 Ember Desktop

Installation components and associated documentation:

- EmberZNet PRO network stack library and APIs: debug and release builds (see Table 3).
- HAL library
- Ember Desktop
- Sample programs (see Table 4)

2.6.2 IAR Embedded Workbench

Licenses and support are provided separately and can be obtained from IAR Systems at <http://www.iar.com>.

Note: The appropriate IAR Embedded Workbench version varies with the revision of EmberZNet PRO. Check the release notes to verify which IAR compiler version you should use.

2.7 Documentation

Document 120-4031-000, *EM35x Quick Start Guide*, is shipped with the EM35x Development Kit. All other documents listed below are available from Silicon Labs' support page (www.silabs.com/zigbee-support) as Adobe PDF files or as on-line HTML help:

- EM35x Development Kit User's Guide (this document)
- The *Ember Application Development Fundamentals* series (UG103.n)
- *Testing and Debugging Applications for the Ember EM2xx and EM35x Platforms* (UG104)
- *Advanced Application Programming with the Stack and HAL APIs* (UG105)
- *EmberZNet API Reference: For the EM350 SoC Platform* (120-3022-000)
- *EM35x Breakout Board Technical Specification* (TS6)
- *Ember Debug Adapter (ISA3) Technical Specification* (TS7)
- *EM35x Data Sheet* (120-035X-000)
- *Ember Desktop User's Guide* (UG111 and on-line HTML help)
- Application notes (various)

2.8 Before You Begin

The EM35x Development Kit is designed only for the operating conditions and interfaces specified in documents TS7, the *Debug Adapter (ISA3) Technical Specification*, and TS6, the *EM35x Breakout Board Technical Specification*. Any modifications or alterations to the hardware are liable to cause irreparable damage to the EM35x Development Kit, and shall render its warranty null and void.

If you wish to use the EM35x Development Kit in a non-standard fashion, consult with Silicon Labs Customer Support or Sales before starting custom development. This is especially true if you are embarking on an aggressive development cycle, plan to use custom hardware, or wish to use alternate software configuration files, images, or hardware configurations. Contact us and describe your plan, so we can help you minimize the risk of delays in your development cycle.

3 Development Kit Components

3.1 Overview

Figure 1 shows the components in the EM35x Series Development Kit.



Figure 1. EM35x Series Development Kit

This chapter describes the hardware and software components in the EM35x Series Development Kit.

3.2 Hardware

The EM35x Series Development Kit development board set has three major hardware components:

- EM35x Breakout Board
- EM35x Module
- Debug Adapter (ISA3)

The EM35x Series Development Kit also includes the following components:

- US-to-Multinational AC Plug Power Adapter Kit
- Power-Over-Ethernet (PoE) Switch
- USB Power Adapters
- Miscellaneous Cables (USB, DEI, Packet Trace Port, and RF Coax)

For detailed information about the Breakout Board, see document TS6, the *EM35x Breakout Board Technical Specification*.

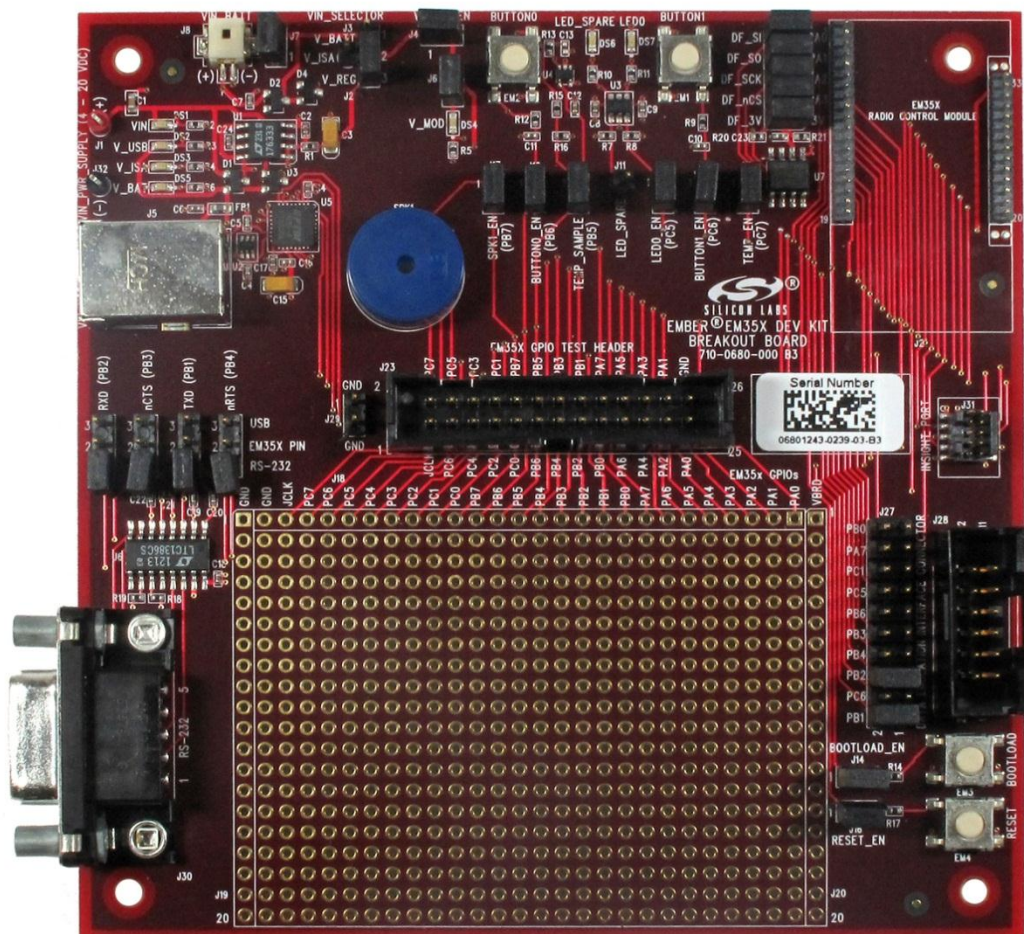


Figure 2. EM35x Breakout Board

3.2.2 EM35x Module

The EM35x Series Development Kit Module pictured in Figure 3 is used together with the Development Kit Breakout Board to prototype customer hardware, and to develop and debug application software. Its low-power design represents a good starting point for your own product design.

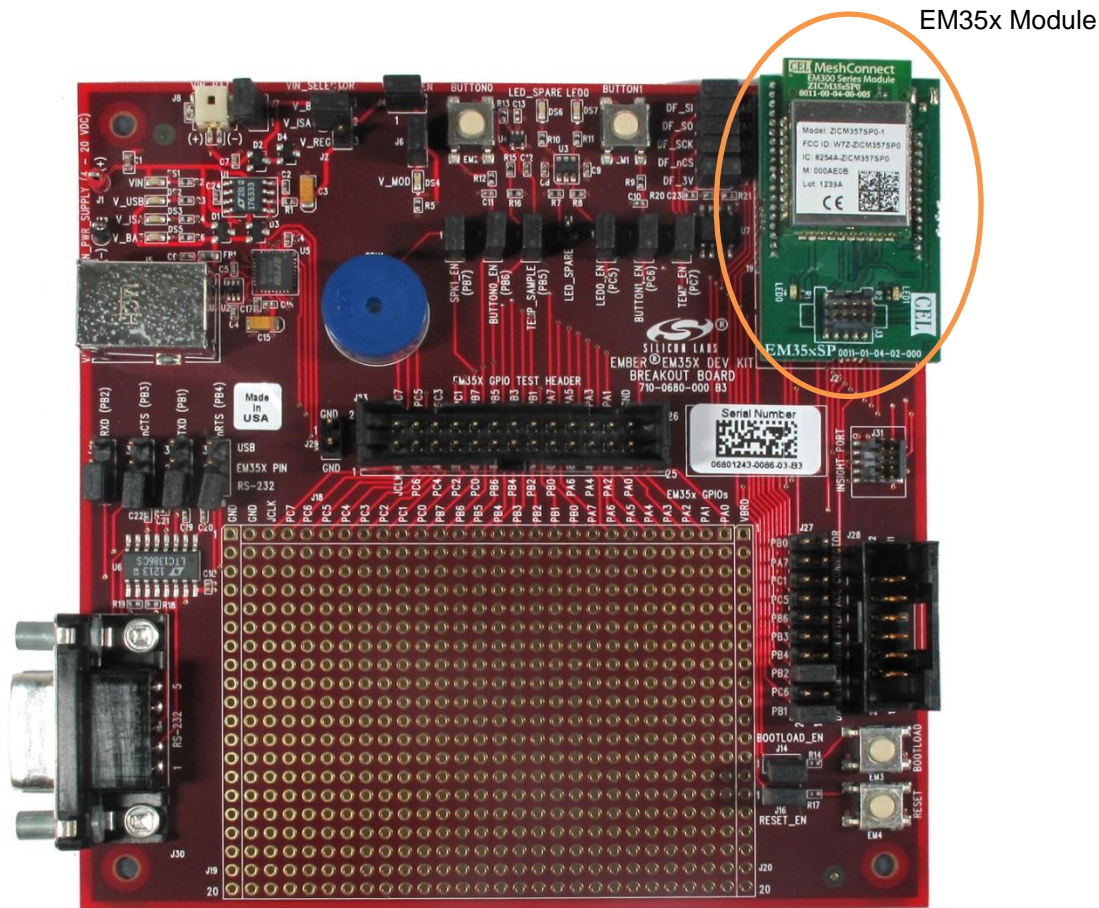


Figure 3. EM35x Breakout Board with EM35x Module installed

The EM35x Module includes these components:

- The Ember EM35x SoC (IEEE 802.15.4-compliant, 2.4 GHz radio transceiver with ARM[®] Cortex[™]-M3 microcontroller).
- All off-chip components required for optimum SoC performance
- Two LEDs
- Packet Trace Port connector

Once you move away from the EM35x Module and toward application-specific hardware, refer to the extensive Ember reference design library at www.silabs.com/zigbee-support. Each reference design provides a schematic, Bill of Materials (BOM), characterization data, and layout files. They are designed and delivered in a manner that reduces both time to market and hardware implementation risk. For assistance in locating the best design for your needs, contact support at www.silabs.com/zigbee-support.

3.2.3 Debug Adapter (ISA3)

The Debug Adapter (ISA3) provides an efficient and configurable debug interface to the Development Kit Breakout Board for processing emulation and debug commands. It also provides an interface to Ember Desktop for monitoring and managing network data.

The Debug Adapter (ISA3) has the following components:

- Debug Port interface to the Breakout Board's Extended Debug Interface Connector
- Packet Trace Port interface to the radio communication module, providing programming and debugging services
- TCP/IP 10/100 Ethernet interface with Power-over-Ethernet functionality
- USB Interface

3.3 Acceptable Power Sources for Normal Operation

Table 2 lists all of the components and the acceptable combinations of power sources that yield normal operation. Potential power sources include the following:

- Power-Over-Ethernet (PoE) Switch
- AAA battery pack (3 VDC)
- USB cable
 - Connected to PC
 - Connected to USB Power Supply with Adapters
- Power over Packet Trace Port on Debug Adapter (ISA3) (the target power select switch on the front panel must be set to *Int* (Internal)).
- From RCM dedicated power
- External dc adapter (not supplied with the Development Kit)

Table 2. Safe Combination of Operating Power

	PoE	External DC	AAA Battery Pack	Packet Trace Port	USB	Breakout Board
Debug Adapter (ISA3)	X	N/A	N/A	N/A	X	N/A
EM35x Module	N/A	N/A	N/A	X ³	N/A	X
EM35x Breakout Board	N/A	X ⁴	X	X ⁵	X	N/A

3.3.1 Power-Over-Ethernet (PoE) Switch

The EM35x Series Development Kit contains a NETGEAR 8-port Power-over-Ethernet (PoE) switch that supplies power to a Debug Adapter (ISA3) over a standard TIA/EIA-568 category 5 cable. The Debug Adapter (ISA3) can, in turn, supply power to an EM35x Breakout Board and the Module that is mounted on it. Thus, you can place Breakout Boards wherever an Ethernet cable connection is available.

The dc operating power for data terminal units is fed through unused pairs 7/8 and 4/5. For more information about the NETGEAR switch, see its online technical specifications.

The NETGEAR PoE Switch that is supplied in the EM35x Series Development Kit is certified by Silicon Labs. If you wish to use a different PoE switch, consult Customer Support at www.silabs.com/zigbee-support.

³ Power is sourced from the Debug Adapter (ISA3) to the RCM module using the Packet Trace Port cable connected to the Debug Adapter (ISA3) with its target power select switch set to *Int* (Internal).

⁴ External dc power between 4 V and 20 V can be connected to clip leads J1 (+) and J32 (-) on the EM35x Breakout Board.

⁵ Power is sourced from the Debug Adapter (ISA3) to the EM35x Breakout Board using the Packet Trace Port cable connected to the Debug Adapter (ISA3) with its target power select switch set to *Int* (Internal).

3.3.1.1 *US-to-Multinational AC Plug Power Adapter Kit*

The US-to-Multinational AC Plug Power Adapter Kit (Figure 4) is a set of 3 power adapters for use with the NETGEAR PoE Switch. These three interchangeable plugs are provided with various port configurations suitable for various locales: North America, Europe, United Kingdom, and Australia.



Figure 4. US-to-Multinational AC Plug Power Adapter Kit

3.3.2 USB Power Supply with Adapters

The USB power adapter (Figure 5) is a linear regulated wall plug power supply with 5 V dc at 1A regulated output. It has a USB type-B connector for connecting power via a USB cable to the USB port of the Debug Adapter (ISA3) or Breakout Board. Four interchangeable snap-in input plugs are provided with various port configurations suitable for various locales: North America, Europe, United Kingdom, and Australia.



Figure 5. USB Power Supply with Interchangeable AC Adapters

3.4 Software

The EM35x Series Development Kit contains the following software components:

- Network Stack
- Hardware Abstraction Layer
- Sample Applications
- Ember Desktop

3.4.1 Network Stack

The EmberZNet PRO network stack is an advanced implementation of the ZigBee Pro wireless protocol stack which provides mesh routing algorithms for creating reliable, flexible, and secure networks. The stack's APS layer provides interfaces for delivering messages reliably. The network stack API is documented in documents UG105, *Advanced Application Programming with the Stack and HAL APIs*, and UG100, the *EZSP Reference Guide*, as well as 120-3022-000, *EmberZNet API Reference: For the EM35x SoC Platform*, also available in HTML.

The network stack is delivered as a collection of libraries that you can link to your applications. The libraries are available in both release and debug build versions:

- **Debug build** implements features that facilitate application debugging, such as error checking and an interface to Ember Desktop, which provides information about traffic flow, API traces, error conditions, and more.
- **Release build** omits debugging functionality, so it is smaller and runs more efficiently. Use this build in your application after it is fully debugged and ready for product release.
- **Stub library** is a library used for resolving references but which has no functionality. For those libraries which are not needed by the application, the stub library must be linked in order for the stack to build properly.

Table 3 lists the EmberZNet PRO stack and associated libraries. It shows the base library name, but it does not duplicate entries for the debug and stub libraries because they are self-explanatory.

Table 3. EmberZNet PRO Stack and Associated Libraries

Library Name	Description
alarm-library	Alarm library
binding-table-library	Binding table library
end-device-bind-library	End device bind library
security-library-core	Core security library
security-library-core-end-device	Core security library for end devices. This library is smaller than the full core security library.
security-library-link-keys	Library to handle optional link keys
zigbee-pro-stack	Core ZigBee Pro stack functionality

3.4.2 Hardware Abstraction Layer

The hardware abstraction layer (HAL) acts as a conduit between the network stack and the node processor and radio. Separating network stack functionality from the specific hardware implementation enables easy portability. HAL code is provided as a library that you can alter in order to customize, extend, or reduce device functionality across various hardware platforms. The HAL API is documented in document 120-3022-000, the *EmberZNet API Reference: For the EM35x SoC Platform*.

3.4.3 Sample Applications

The EM35x Series Development Kit contains sample applications. These demonstrate aspects of the network stack that show the breadth of stack functionality. Silicon Labs provides an IAR project and workspace file for each application. Table 4 lists the sample applications that are in the current distribution. All applications are in the `app` directory.

Table 4. Development Kit Sample Applications

Application name	Directory	Description
Ember application framework, version 2 (AFV2) Smart Energy / Home Automation	<code>framework</code>	An application framework that can be set up to implement any ZigBee application based on the ZigBee Cluster Library (ZCL), optionally including users' extensions.
Nodetest	<code>nodetest</code>	A low-level test program meant for functional testing of RF modules.
Sensor & Sink	<code>sensor</code>	An example of a complete, non-ZCL- and non-AFV2-based application that implements a distributed sensor network with one or more data collection points (called "sensors") and one or more data storage points (called "sinks").
ZigBee Device Object (ZDO) Sample	<code>zdo</code>	An example of how to send ZDO messages using the provided ZDO utility functions.
Manufacturing Library Sample	<code>mfglib</code>	An example of how to use the manufacturing library.
Standalone Bootloader Demo	<code>standalone-bootloader-demo</code>	An example of how to use the Ember Standalone Bootloader in the context of an application.

3.4.4 Ember Desktop

Ember Desktop is a graphical tool incorporating a Network Analyzer utility that manages the Development Kit hardware and displays network and node activity in real time (see Figure 6). It provides a rich and flexible interface to Ember-embedded networks, which helps you develop and debug new network applications.

Ember Desktop includes these features:

- Ember AppBuilder, a tool for generating ZigBee-compliant applications, made up of the Ember application framework and a graphical tool for configuring the included source code. Ember AppBuilder gives you an interface for turning on or off embedded clusters and features in the code compiled into a finished application.
- An Integrated Development Environment (IDE) with multiple editor panes that provide tiered views of network activity, letting you drill down from a high-level map of node interactions to the details of each packet.
- Customizable filters that let you specify exactly which network activities to display.
- Log files that save captured data, so you can step through transactions and events for detailed analysis.
- A file browser that lets you easily upload new applications to any connected node.

An Adapters view shown in Figure 6 allows you to interact with the Debug Adapters (ISA3s) and RCM to download new firmware, reset, and manage each of the devices.

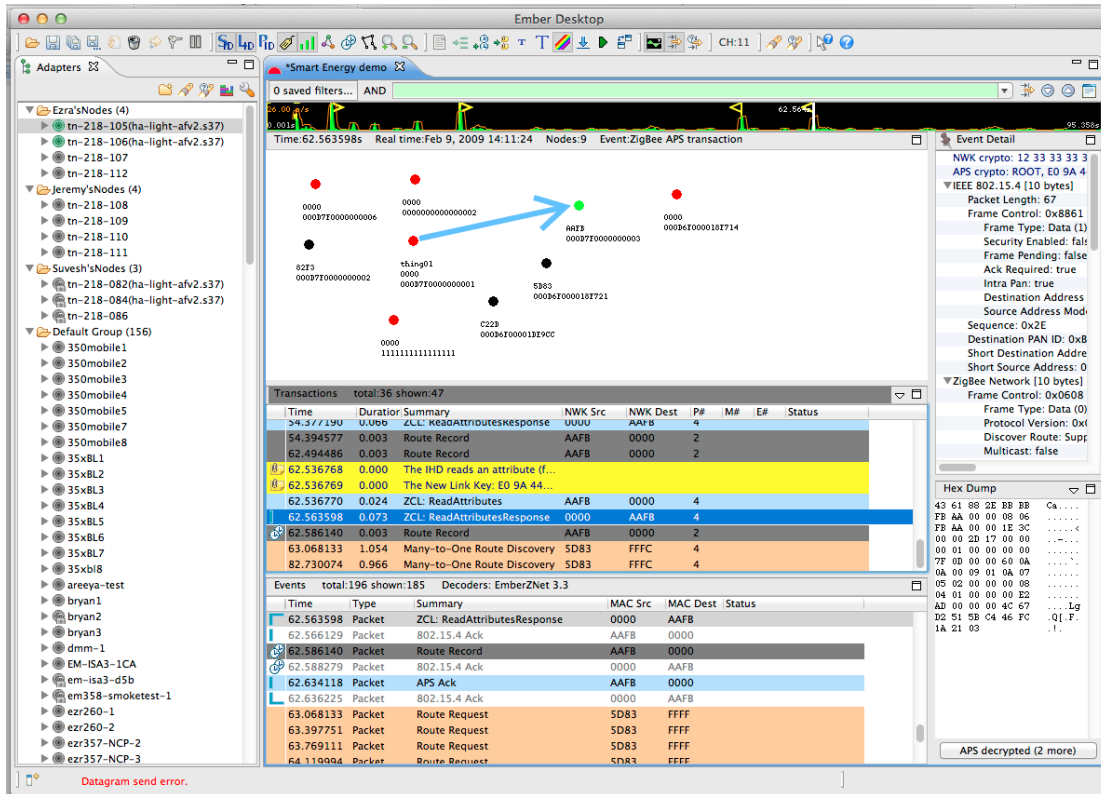


Figure 6. Ember Desktop

For detailed information about Ember Desktop, refer to document UG111, the *Ember Desktop User's Guide*, and Ember Desktop's on-line HTML help.

4 Setting Up Hardware Components

4.1 Overview

This chapter contains the following sections:

- EM35x Radio Communications Module (RCM)
- Hardware Connections
- Connecting the Breakout Board and RCM
- Breakout Board Power Configurations
- Breakout Board Serial Configurations
- Common Hardware Configurations
- Connecting the RF Cable to the RCM

4.2 EM35x Radio Communications Module (RCM)

The EM35x Series Development Kit contains various types of EM35x Radio Communications Modules designed by Silicon Labs' partner, California Eastern Laboratories (CEL). These modules contain either the EM357 or EM351 IC and are configured to use either the PCB antenna or the on-board RF connector (U.FL). Some module variants also include a power amplifier. These configuration options are noted in the Module Product Label, as shown the example label in Figure 7. The top line contains the Silicon Labs Part Number, while the bottom line contains the Module Serial Number. The Module Serial Number is also listed in the Hardware Identification List attached to your *Quick Start Guide*. You can determine the EUI-64 of each module based on this list.



Figure 7. Module Product Serial Number Label

The modules shipped already installed on Breakout Boards are module variant EM357-MOD-ANT-C. All other module variants are contained in the Module Variety Pack. Refer to Table 5 for configuration comparisons of the modules included in your development kit.

Table 5. Module Product Variant Matrix

Silicon Labs Part Number	CELPART Number	Silicon Labs Product Number	IC	Antenna or Connector	PA or non-PA	Qty Included in Kit
EM357-MOD-ANT-C	ZICM357SP0-1-SL	0730	EM357	Antenna	Non-PA	5
EM357-MOD-RF-C	ZICM357SP0-1C-SL	0731	EM357	Connector	Non-PA	1
EM357-MOD-LR-RF-C	ZICM357SP2-1C-SL	0732	EM357	Connector	PA	1
EM357-MOD-LR-AN-C	ZICM357SP2-1-SL	0733	EM357	Antenna	PA	2

4.3 Hardware Connections

To avoid communication problems, verify that each radio communication module is firmly seated in its Breakout Board connector, and its Packet Trace Port cable is properly seated.

4.4 Connecting the Breakout Board and RCM

Refer to document 120-4031-000, *EM35x Quick Start Guide*, included in your development kit for detailed step-by-step instructions for setting up your hardware.

Two hardware modules attach to the EM35x Breakout Board:

- Radio Communication Module (RCM)
- Debug Adapter (ISA3)

CAUTION: Observe electrostatic discharge (ESD) precautions when removing, handling, or replacing RF communication modules.

UG110

To attach the radio communication module:

Align the radio communication module to its footprint outline on the Breakout Board, and then press it into place. Make sure that the RCM sits firmly in its mating connector.

To remove the radio communication module:

Carefully pull the RCM away from the Breakout Board.

To attach the Debug Adapter (ISA3):

1. Plug the Packet Trace Port cable into the Debug Adapter (ISA3).
2. Plug one end of the Packet Trace Port cable into the 10-pin Packet Trace Port (J31) on the Breakout Board, and the other end into the Debug Adapter (ISA3). If desired, the Packet Trace Port cable may be connected directly to the RCM instead of the Breakout Board, but the Breakout Board connector allows for straight cable paths and more convenient swapping of modules.
3. Verify that the Debug Adapter (ISA3)'s Target Power Select switch is set to Int (Internal).

To remove the Debug Adapter (ISA3):

1. Unplug the Packet Trace Port cable from the Breakout Board (or radio communication module).
2. Unplug the Packet Trace Port cable from the Debug Adapter (ISA3).

4.5 Breakout Board Power Configurations

The Breakout Board may be powered in one of four ways:

- Breakout Board Power Supplied by Debug Adapter (ISA3) (*Figure 8*): The default configuration is to supply power via the Debug Adapter (ISA3) and Packet Trace Port cable. Ensure the power switch on the Debug Adapter (ISA3) is in the INT (Internal) position. The power supply of the Debug Adapter (ISA3) is also controlled via the admin interface. You may need to connect to the admin port of the Debug Adapter (ISA3) and issue the “power on” command. No external power supply should be connected to the Breakout Board when supplying power from the Debug Adapter (ISA3). The power supply jumper on the Breakout Board should be in the horizontal position, connecting V_ISA to the center pin. The V_ISA LED will illuminate when power is applied.

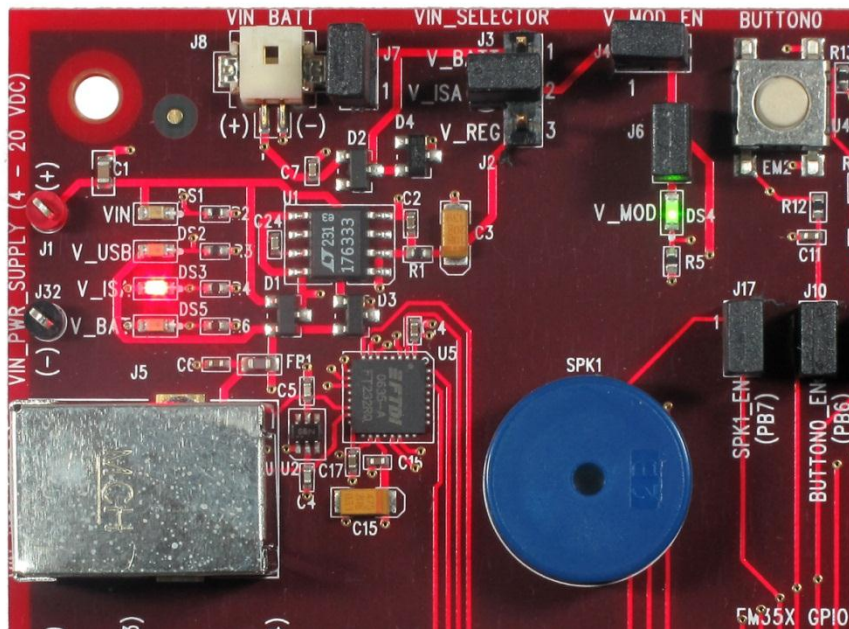


Figure 8. Breakout Board Power Supplied by Debug Adapter (ISA3)

- **Breakout Board Power Supplied by USB (Figure 9):** The Breakout Board may also be powered through the USB connector, either using a USB cable plugged in to the USB power adapter included in the development kit, or by connecting it to your PC. The power supply jumper on the Breakout Board should be in the lower vertical position, connecting V_REG to the center pin. The V_USB LED will illuminate when power is applied. When powering the Breakout Board in this way, ensure the power switch on the Debug Adapter (ISA3) is in the EXT (External) position.

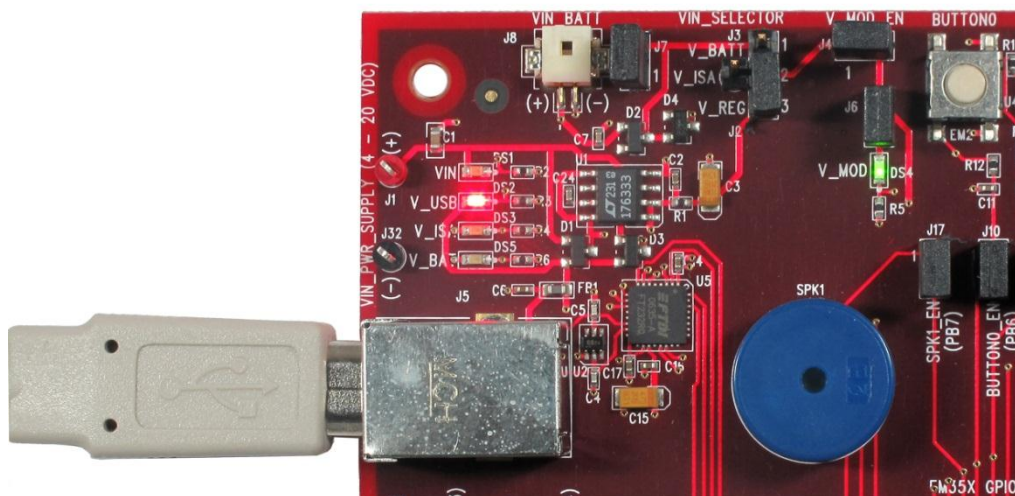


Figure 9. Breakout Board Power Supplied by USB

- Breakout Board Power Supplied by DC Jumpers (Figure 10): The Breakout Board may also be powered by an external power source that provides 4-20 V dc. The power supply should be connected to jumpers J1 (+ supply) and J32 (ground). The power supply jumper on the Breakout Board should be in the lower vertical position, connecting V_REG to the center pin. The VIN LED will illuminate when power is applied. When powering the Breakout Board in this way, ensure the power switch on the Debug Adapter (ISA3) is in the EXT (External) position.

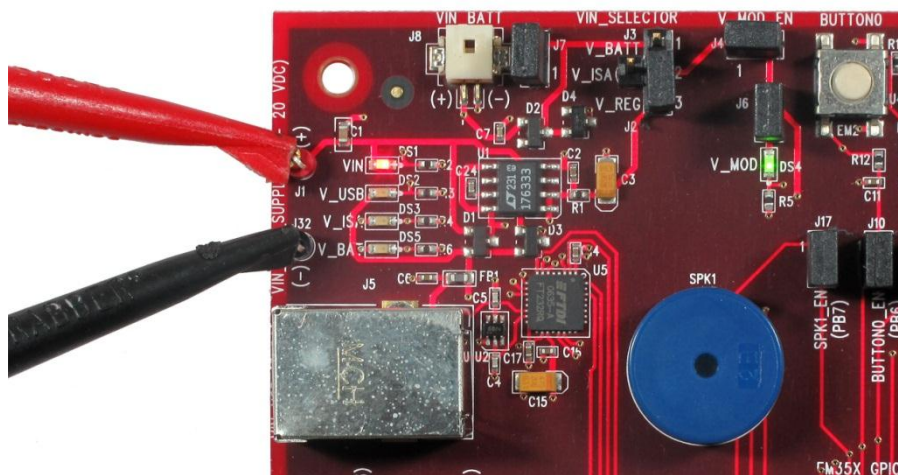


Figure 10. Breakout Board Power Supplied by DC Jumpers

- **Breakout Board Power Supplied by Battery Connector (Figure 11):** Finally, the Breakout Board may also be powered via the battery connector, using the supplied battery packs and 2x AAA batteries. The power

Rev. 0.7

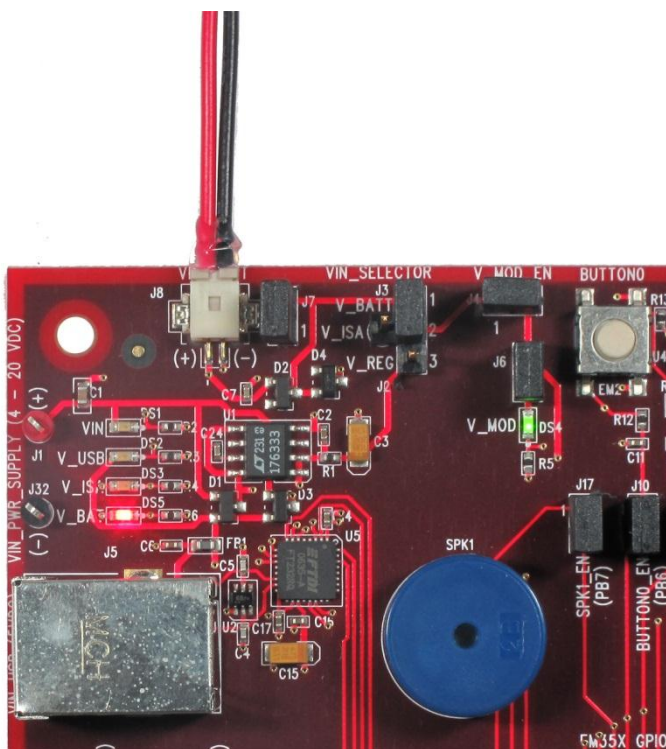


Figure 11. Breakout Board Power Supplied by Battery Connector

21

- **To use the pass-through UART feature of the Debug Adapter (ISA3):** Plug the larger 12-pin DEI cable into the DEI header on the Breakout Board and the connector on the Debug Adapter (ISA3). Ensure that jumpers are installed across the 8th (PB2) and 10th (PB1) rows of headers next to the DEI port on the Breakout Board to pass the UART RX and TX signals through. The serial port selection jumpers between the USB and DB-9 connectors should be removed, or placed on their center pins. Connection to the UART pass-through port can then be made using either of the two options below:
 - a. Connect to the Debug Adapter (ISA3) using Ember Desktop. Launch the console. Choose the Serial 1 tab.
 - b. Run a terminal application (for example, Microsoft Telnet Client, HyperTerminal, PuTTY). Connect to port 4901 on the Debug Adapter (ISA3).
- **To use the RS-232 interface:** Connect a serial cable to the DB-9 connector on the Breakout Board. Ensure the four jumpers above the DB-9 connector are positioned towards the DB-9 connector to send the UART signals to the RS-232 transceiver, and remove the jumpers from the row of headers next to the DEI port or make sure the DEI cable is disconnected.

- **To use the USB interface:** Install the FTDI USB drivers as described in Section 5.6, Installing FTDI USB Drivers, and connect a USB cable to the Breakout Board and your PC. Ensure the four jumpers above the DB-9 connector are positioned towards the USB connector to send the UART signals to the FTDI chip, and remove the jumpers from the row of headers next to the DEI port or make sure the DEI cable is disconnected.

4.7 Common Hardware Configurations

Common configurations of the Breakout Board and Debug Adapter (ISA3) can be seen in Figure 12, Figure 13, Figure 14, and Figure 15.

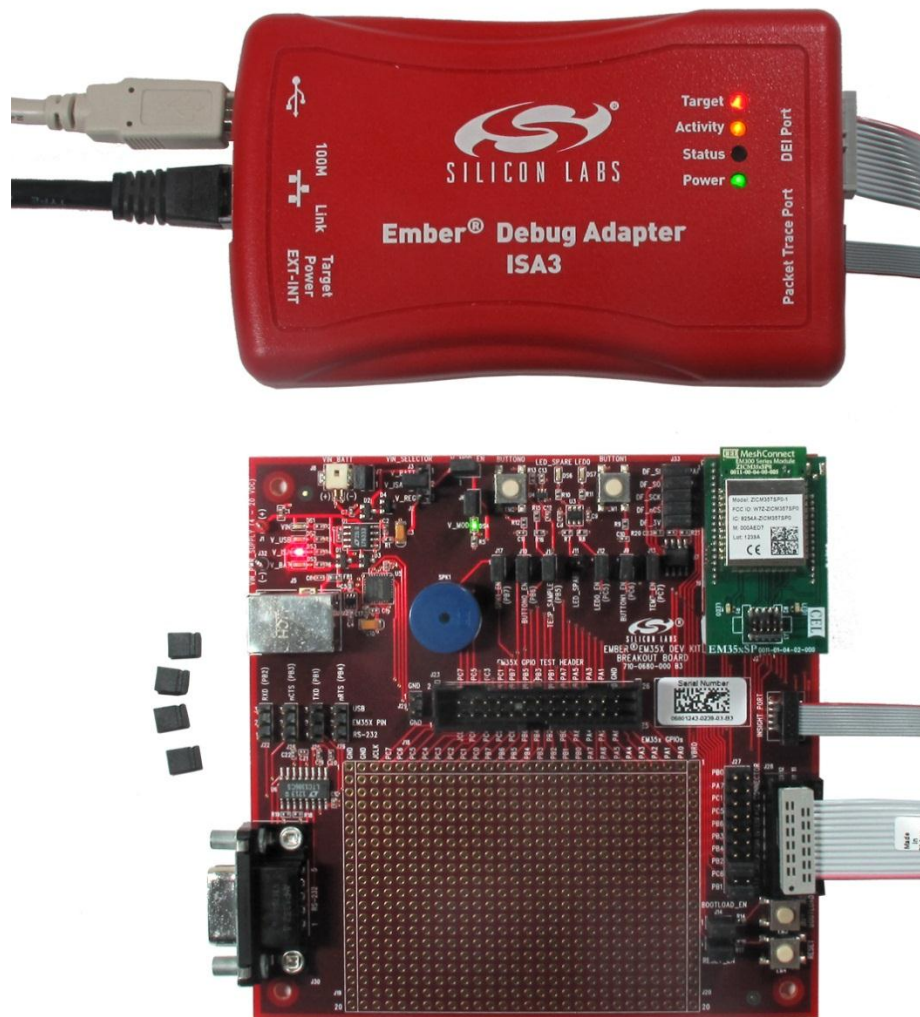


Figure 12. Debug Adapter (ISA3) Supplying Power; Pass-through UART Being Used for Serial Communication

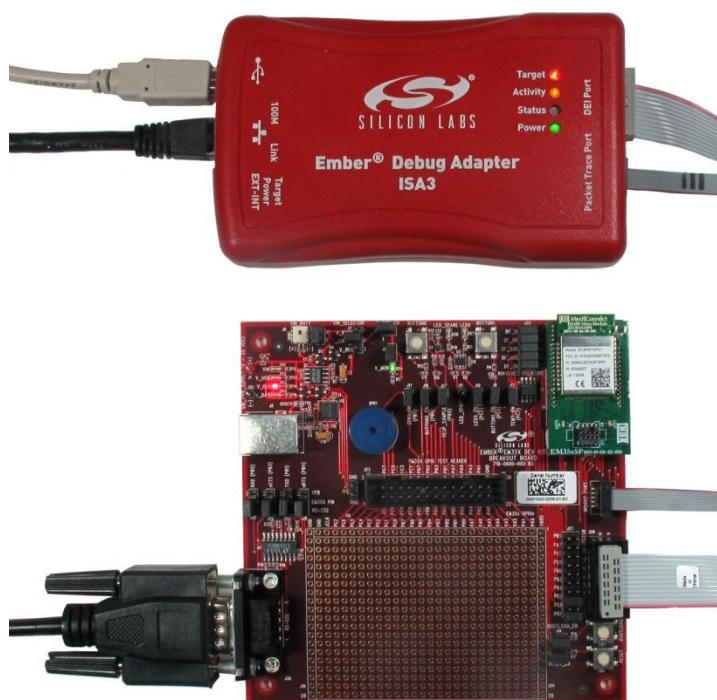


Figure 13. Debug Adapter (ISA3) Supplying Power; RS-232 Serial Communication

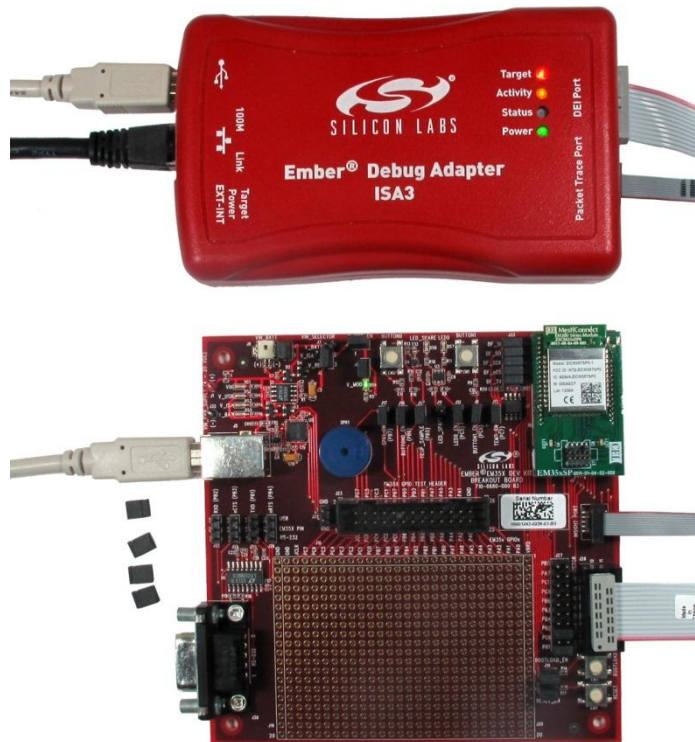


Figure 14. Power Supplied Via USB Connector; Pass-through UART Being Used for Communication



Figure 15. Debug Adapter (ISA3) Supplying Power; FTDI Serial <-> USB Converter Used for Communication

4.8 Connecting the RF Cable to the RCM

As detailed in Table 5, depending on the configuration of the module, some RCMs include a PCB antenna while others include an RF connector (type U.FL) where an external antenna can be connected via an RF adapter cable. See Figure 16 for an example of a module with a PCB antenna. The RF connector may also be used as a connection to test equipment. The development kit contains an RF adapter cable (type U.FL) that plugs into the RF Connector on these modules. See Figure 17 for an example of a module with this RF connector.



Figure 16. Module with PCB Antenna



Figure 17. Module with U.FL RF Connector

5 Installing the Software

5.1 Overview

Ember software for the EM35x platform is distributed through Silicon Labs' support portal at www.silabs.com/zigbee-support. You must contact Customer Support for a portal username.

5.2 Installing the Ember Stack

The Ember Stack installer, `ember-stack-<version>-em35x-dev.exe`, should be installed first. It performs the following actions:

1. Installs the files needed to develop applications on the Ember Stack into a directory specified by the user (The default location is `C:\Program Files\Ember\{release version}\em35x`.)
2. Adds a shortcut to the Start Menu into the folder specified by the user (default directory is `Ember\{release version}\em35x`).
3. Adds an entry to "Add/Remove Programs" (accessed from Windows Control Panel) so the stack can be uninstalled.

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

5.3 Installing Ember Desktop

Ember Desktop can be installed by running the `Insight_Desktop_Installer_<version>.exe` self extracting executable found on the portal. Document 120-4031-000, *EM35x Quick Start Guide*, included in your EM35x Series Development Kit, provides details about installing Ember Desktop for the first time and running the First-Time Setup Wizard.

5.4 Installing IAR Embedded Workbench for ARM

A CD including IAR Embedded Workbench for ARM is included with your development kit. Depending on which development kit you ordered (IAR or not), you will receive either a 30-day evaluation version of EWARM or a licensed version.

Note: The appropriate IAR Embedded Workbench version varies with the revision of EmberZNet PRO. Check the release notes to verify which IAR compiler version you should use.

To install the IAR Embedded Workbench for ARM:

1. 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`.
2. Refer to the "Quickstart Installation Information" section of the IAR installer for additional information about the installation process and how to configure your license.

5.5 Installing the Debug Adapter (ISA3) Utilities

The installer `ISA3_Utilites_<version>.exe` performs the following actions:

1. Installs the USB driver for the Debug Adapter (ISA3).
2. Installs the following command line utilities:
 - `em3xx_load.exe`
 - `em3xx_isa.exe`
 - `em3xx_convert.exe`
3. Adds an entry to "add/remove programs" (accessed from Windows Control Panel) so that the utilities can be uninstalled.

4. Modifies your PATH environment variable so that the command line utilities can be easily executed from a Windows Command Prompt.

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

5.6 Installing FTDI USB Drivers

To use the USB interface of the Breakout Board for UART connectivity, you must install the drivers for the FTDI USB<->Serial converter. You can obtain this driver from FTDI at <http://www.ftdichip.com/Drivers/VCP.htm>.

These drivers are not required if you connect to the UART of the Breakout Board using the RS-232 port or via the pass-through UART functionality of the Debug Adapter (ISA3).

6 Using the Debug Adapter (ISA3)

6.1 Overview

The Debug Adapter (ISA3) provides an Ethernet connection to EM35x devices. You can use the Debug Adapter (ISA3) to perform the following tasks:

- Program and debug using Serial Wire or JTAG.
- Monitor packets sent and received by the radio.
- Send and receive serial port data.
- Send and receive debugging data.
- Turn power on and off.
- Control reset, bootloader and GPIO signals.

6.2 Connectivity and Power

All the features of the Debug Adapter (ISA3) are available via the Ethernet connection. The Debug Adapter (ISA3) can also be connected to a PC via USB to change the Ethernet settings.

When connected via USB, the Debug Adapter (ISA3) will also be powered via USB. It should be connected to a powered USB hub, or port on your PC that is capable of supplying full power to connected devices.

When connected via Ethernet, the Debug Adapter (ISA3) may be powered in one of two ways:

1. Using an Ethernet switch that supplies Power Over Ethernet (PoE).
2. Using a USB power supply.

The Debug Adapter (ISA3) will also supply power to the connected EM35x device when the power switch is in the INT (Internal) position. If the connected EM35x device has its own power supply, ensure the power switch is kept in the EXT (External) position.

You can connect via Ethernet to the ports listed in Table 6 using either Ember Desktop or a terminal application (for example, Microsoft Telnet Client, HyperTerminal, or PuTTY).

Table 6. Debug Adapter (ISA3) Ports

Port	Use
4900	Virtual serial port on EM35x via Packet Trace Port cable.
4901	Physical serial port on EM35x via Data Emulation Interface cable.
4902 or 23	Admin interface.

6.3 Ethernet Settings

By default, Debug Adapters (ISA3)s are configured to get an IP address via DHCP. They are assigned hostnames of EM-ISA3-00, EM-ISA3-01, etc. You can use Ember Desktop to discover the Debug Adapters (ISA3)s on the network and see their IP addresses.

If you are unable to use DHCP, the Ethernet configuration may be changed using the admin interface over USB as described in “Using the Admin Interface over USB.” The following example changes the settings to a static IP address of 192.168.1.100 with a netmask of 255.255.255.0 and a gateway of 192.168.1.1.

```
>em3xx_isa.exe --admin "ip static 192.168.1.100 255.255.255.0 192.168.1.1"
em3xx_isa version 1.0.9
DLL version 1.0.13, compiled Aug 19 2009 19:42:00
```

```
Success: ip: (after reboot) Static 192.168.1.100 255.255.255.0 192.168.1.1
```

```
>em3xx_isa.exe --admin "ip dhcp off"
em3xx_isa version 1.0.9
DLL version 1.0.13, compiled Aug 19 2009 19:42:00
```

```
Success: ip: (after reboot) DHCP off
```

You can change the hostname to make it more descriptive or to avoid conflicts if Debug Adapters (ISA3)s from multiple development kits are connected to the same network. To change the hostname, use the admin command `hostname set <hostname>`.

6.4 Using the Admin Interface over USB

1. Connect the Debug Adapter (ISA3) to the PC using USB. Only connect one Debug Adapter (ISA3) at a time by USB.
2. Run `em3xx_isa.exe` with the `--admin` option.
3. Configure the Debug Adapter (ISA3) using the commands described in Chapter 7, Debug Adapter (ISA3) Commands.

6.5 Using the Admin Interface over Ethernet

1. Connect the Debug Adapter (ISA3) to the network using Ethernet.
2. Connect to the admin port on the Debug Adapter (ISA3) using either Ember Desktop or a terminal application:
 - Connect to the Debug Adapter (ISA3) using Ember Desktop. Launch the console. Choose the Admin tabOR
 - Run a terminal application (e.g., Microsoft Telnet Client, HyperTerminal, PuTTY). Connect to port 4902 or 23 on the Debug Adapter (ISA3).
3. Configure the Debug Adapter (ISA3) using the commands described in Chapter 7, Debug Adapter (ISA3) Commands.

6.6 Using the Serial Ports

Connect to port 4900 or 4901 on the Debug Adapter (ISA3) using either Ember Desktop or a terminal application. Port 4900 connects to the virtual serial port on the EM35x via the Packet Trace Port cable. The virtual serial port has no baud rate or other settings.

Port 4901 connects to the physical serial port on the EM35x via the Data Emulation Interface (DEI) cable. The default settings are 115200 baud, 8-n-1 (8 data bits, no parity bit, 1 stop bit). To change the default settings, use the port admin command. For example, for 19200 baud, 8-n-1, use the command port 1 19200 8-n-1.

6.7 Updating the Firmware

Periodically, Silicon Labs provides updated firmware for the Debug Adapter (ISA3). You can perform the update using em3xx_isa.exe or em3xx_load.exe. Both utilities automatically update the firmware to the latest version each time they run.

To update a Debug Adapter (ISA3) over USB, run em3xx_isa.exe with no options. For example:

```
>em3xx_isa.exe
em3xx_isa version 1.0.9
DLL version 1.0.13, compiled Aug 19 2009 19:42:00
Updating firmware: Ember InSight Adapter V1 compiled May 26 2009 13:47:00 V0.2b4
Replacing firmware: Ember InSight Adapter V1 compiled Feb 26 2009 21:54:00
... Firmware update successful. CRC=8DA7
Waiting for new firmware to boot
New firmware booted successfully
```

To update a Debug Adapter (ISA3) over Ethernet, run em3xx_isa.exe with the --ip option. For example:

```
>em3xx_isa.exe --ip 192.168.1.100
em3xx_isa version 1.0.9
DLL version 1.0.13, compiled Aug 19 2009 19:42:00
Connecting to IP address 192.168.1.100
Updating firmware: Ember InSight Adapter V1 compiled May 26 2009 13:47:00 V0.2b4
Replacing firmware: Ember InSight Adapter V1 compiled Feb 26 2009 21:54:00
... Firmware update successful. CRC=8DA7
Waiting for new firmware to boot
New firmware booted successfully
```

7 Debug Adapter (ISA3) Commands

7.1 Syntax Conventions

Element	Meaning	Example
<i>Literal</i>	A command or argument that is entered exactly as shown.	hostname read
<i>Variable</i>	An argument that is entered as the desired value for that variable.	hostname set <i>hostname</i>
[]	An argument chosen from a list.	[on off]

7.2 Entering Commands

- Commands can be entered over USB (see Section 6.4, “Using the Admin Interface over USB”) or Ethernet (see Section 6.5, “Using the Admin Interface over Ethernet”).
- Commands are not case sensitive.
- Arguments are separated with a single space.
- Backspace can be used to correct mistakes.

7.3 Command List

- button
- config
- connections
- debug
- help
- hostname
- ip
- port
- power
- reset
- usbaddr
- version

7.3.1 button

```
button [0|1] [press|release]
```

Description

Simulates button 0 or 1 on the Breakout Board being pressed or released.

The Data Emulation Interface cable must be connected.

Example

```
button 1 press
```

7.3.2 debug

```
debug [on|off|disable]
```

Description

Turns BackChannel Debug on (the default), off (for sleepy devices), or disable (for read-protected devices). With no arguments specified, it prints the current debug status.

Examples

```
debug
debug off
debug disable
```

7.3.3 config

```
config
```

Description

Displays the configuration of the Debug Adapter (ISA3):

- hostname and static IP settings
- IP mode (static or DHCP) and current IP settings
- serial port settings
- firmware version
- serial number and Ethernet MAC address
- type, EUI and board name of the attached EM35x device

You can also view the configuration using Ember Desktop.

7.3.4 connections

```
connections
```

Description

Displays active and inactive TCP connections across the various Ethernet ports of the ISA3.

Example

```
connections
```

7.3.5 help

```
help
```

Description

Displays a list of all Debug Adapter (ISA3) commands.

7.3.6 hostname

```
hostname set hostname
hostname read
```

Description

Sets or displays the hostname of the Debug Adapter (ISA3).

The Debug Adapter (ISA3) must be reset for a change to take effect.

You can also view the hostname using Ember Desktop.

Example

```
hostname set mydevice3
```

7.3.7 ip

```
ip static ipaddress netmask gateway
ip dhcp [on|off]
```

Description

Sets the static IP address, netmask and gateway. These are only used when DHCP is turned off.

Turns DHCP on or off.

The Debug Adapter (ISA3) must be reset for a change to take effect.

Examples

```
ip static 192.168.1.100 255.255.255.0 192.168.1.1
ip dhcp off
```

7.3.8 port

```
port 1
port 1 baud
port 1 baud [5|6|7|8]-[n|o|e]-[1|2]
```

Description

Displays the configuration of the serial port.

Sets the baud rate, data bits (5, 6, 7 or 8), parity (none, odd or even) and stop bits (1 or 2).

Examples

```
port 1
port 1 57600
port 1 115200 8-n-1
```

7.3.9 power

```
power [on|off|read]
```

Description

Turns the power to the attached EM35x device on or off. Displays the current power setting.

You can also turn the power on and off using Ember Desktop.

Example

```
power on
```


7.3.10 reset

```
reset  
reset adapter  
reset host [hold]
```

Description

Resets the attached EM35x device if no arguments are specified

Resets the Debug Adapter (ISA3) when “adapter” argument is specified.

Resets the host MCU (if using the EM35x NCP Host Breakout Board and DEI cable is connected to ISA3), optionally holding the host MCU in reset state if “hold” is specified.

You can also reset the device using Ember Desktop.

7.3.11 usbaddr

```
usbaddr read  
usbaddr set [0|1|2|3|FF]
```

Description

Displays the USB address configuration of a specific Debug Adapter (ISA3).

Sets the USB address configuration, range is 0-3 or FF (default).

The Debug Adapter (ISA3) must be reset for a change to take effect. See document AN717, *Programming Options for the EM35x Platform*, for step by step instructions on setting the address of a Debug Adapter (ISA3).

7.3.12 version

```
version
```

Description

Displays the firmware version of the Debug Adapter (ISA3).

You can also view the version using Ember Desktop.

8 Software Tools Overview

8.1 Using the EM35x Nodetest Application

The nodetest application replaces the rangetest application provided with earlier releases. This new application supports a more advanced command parser and provides results in a more consistent, easy to parse format.

The Nodetest application is pre-programmed on to the Radio Communication Modules (RCMs) included with your EM35x Development Kit. The pre-built application image is also located at {stack installation directory}\app\nodetest\nodetest.s37.

You may communicate with the serial interface of the Nodetest application using the methods described in Section 6, "Using the Debug Adapter (ISA3)." A command line interface is supported on both the Virtual UART ("Serial 0" tab in the Ember Desktop Console view) and the Physical UART ("Serial 1" tab in the Ember Desktop Console view). If using the Physical UART, ensure that the serial port is configured for 115200 baud, 8 data bits, No parity, and 1 stop bit. By default the Debug Adapter (ISA3) is configured in this way, but it can be adjusted in this manner:

- Connect to the admin interface of the Debug Adapter (ISA3).
- Issue the baud rate configuration comment for the pass-through UART (port 1): port 1 115200

The Nodetest application provides on-line help about the available commands that can be viewed by executing the command "help". All input parameters to Nodetest are specified in hexadecimal without a "0x" prefix.

8.2 Running a Packet Error Test

A typical usage scenario for the Nodetest application is measurement of Packet Error Rate (PER). PER is defined in 802.15.4 as the percentage of transmitted packets that are not detected correctly. The Nodetest application provides two commands "rx" and "tx" that are designed to interoperate with each other across two nodes running this application or with the "receive" and "transmit" commands from the legacy Rangetest application used in earlier releases of the EM250 and EM260 platforms.

PER measurement procedure:

1. Load the Nodetest application on two EM35x nodes.
2. Set the radio to the desired channel on both nodes with the "setchannel n" command where "n" is the desired channel number in hex.
3. Issue the "rx" command on the first node.
4. Issue the "tx a" command on the second node. (This command will transmit 10 (hex a) packets of the form required for PER analysis. For actual PER analysis, send 1000 or more packets.)
5. Observe the PER result in the third column of the "rx" command output, for example6:

```
> rx
{{(rx)} test start ('e'nd)}
#{{(rx)}
  {num}    {seq}    {per} {lqi}  {rssi}{gain} {status} {time}          {length}}
{ { 1} { 1} { 0} {0x00} {-22} {0x31} {0x4000} {0x000F3716} {0x12} }
{ { 2} { 2} { 0} {0x00} {-22} {0x31} {0x4000} {0x000F9FE6} {0x12} }
{ { 3} { 3} { 0} {0x00} {-22} {0x31} {0x4000} {0x000008B6} {0x12} }
{ { 4} { 4} { 0} {0x00} {-22} {0x31} {0x4000} {0x00007186} {0x12} }
{ { 5} { 5} { 0} {0x00} {-22} {0x31} {0x4000} {0x0000DA57} {0x12} }
{ { 6} { 6} { 0} {0x00} {-22} {0x31} {0x4000} {0x00014327} {0x12} }
{ { 7} { 7} { 0} {0x00} {-22} {0x31} {0x4000} {0x0001ABF6} {0x12} }
{ { 8} { 8} { 0} {0x00} {-22} {0x31} {0x4000} {0x000214C6} {0x12} }
{ { 9} { 9} { 0} {0x00} {-22} {0x31} {0x4000} {0x00027D96} {0x12} }
{ {10} {10} { 0} {0x00} {-22} {0x31} {0x4000} {0x0002E666} {0x12} }
{{(rx)} test end}

>
```

To end the rx test, press the 'e' key. To reset the rx test statistics, end and restart the rx test.

⁶ This example may show different length packets than you see.

8.3 Testing Deep Sleep Current

- Deep sleep current can be easily measured using the Nodetest application running on a Breakout Board. Deep sleep current can only be measured when power is supplied to the board from a source other than the Debug Adapter (ISA3). The buffer logic on the Debug Adapter (ISA3) consumes extra current which will not allow for accurate measurements.
- If a Debug Adapter (ISA3) is connected to the Breakout Board, then the Debug Adapter (ISA3)'s Target Power Select Switch should be switched to EXT (External) so that the Debug Adapter (ISA3) is not supplying power.
- Supply power through the USB jack, the dc jack, or the AAA Battery Pack. (The USB jack is recommended and reviewed in these test steps because it is the simplest.)
- Rotate the VIN_SELECTOR jumper (J2/J3) on the Breakout Board to the V_REG position (pins 2 and 3). If power is supplied properly and all four power/LED jumpers (J2/J3, J4, J6, and J7) in the upper left corner of the board are installed, then only the red V_USB LED and the green V_MOD LED will be illuminated.
- To properly measure current, remove the V_MOD jumper (J6) from the Breakout Board. This ensures that the green V_MOD LED is not being driven. If supplying voltage through the J8 battery connector, also remove J7 so the V_BATT LED DS5 is not driven.
- Remove the V_MOD_EN jumper (J4) from the Breakout Board to take current measurements. Then, connect an ammeter in place of the jumper.
- If using EM35x Breakout Board revision B0 or later, remove the DataFlash jumpers at J33 to isolate the DataFlash IC from the rest of the circuit.

These jumper settings can be seen in Figure 18.

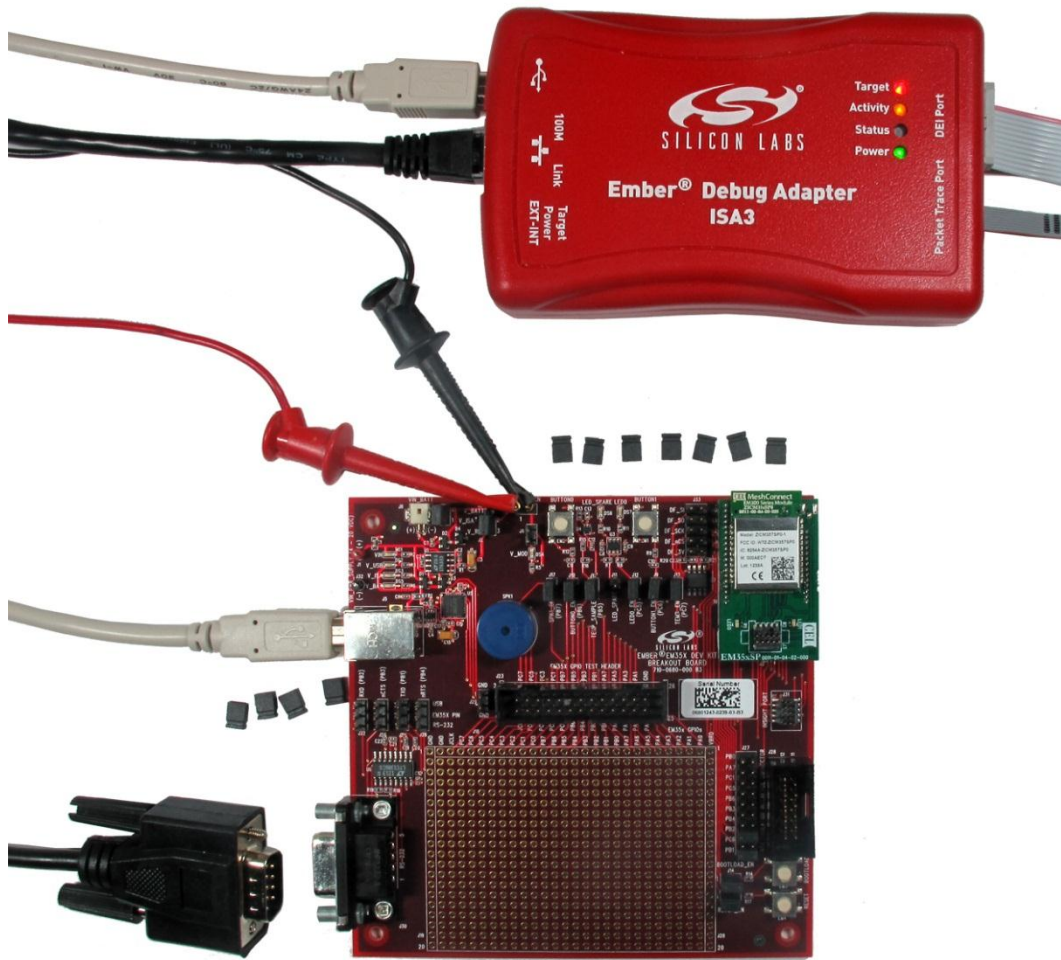


Figure 18. Jumper setting for Deep Sleep Current Testing

- Given an RCM programmed with Nodetest, execute the "shutdown" command:

```
> shutdown
```

Will wake from GPIO activity on the following port pins:

Port B, pin 6.

Port C, pin 6.

Entering SLEEPMODE_POWERDOWN via halSleep...

- Once the node is asleep, disconnect the Packet Trace Port cable, the DEI cable, the RS232 cable, and remove all four UART jumpers (J22, J24, J25, and J26) from the Breakout Board between the USB and RS232 connectors to obtain the lowest current settings. All other jumpers may remain installed.
- Waking up is accomplished via either activity on the SerialWire debugger or pressing BUTTON0 or BUTTON1.

To measure deep sleep while the Packet Trace Port cable is still attached, issue the "swjdisable 1" command in Nodetest to disable the SerialWire/JTAG port, and place the SWJ pins in a low current state. While the SWJ is disabled, the Packet Trace Port/debugger will not be functional until either the command "swjdisable 0" is issued or the chip resets.

8.4 Example of Opening, Building, and Programming a Sample Application

Follow these steps to open, build, and program the sink example application from the standard Ember installer directory.

1. Open the sink.eww file (sink workspace) in IAR Embedded Workbench for ARM (EWARM). This can be done by either double clicking the sink.eww file (located in the "app\sensor\ewb-em357" subdirectory within your EmberZNet installation) or from inside of EWARM using File > Open > Workspace and navigating to the sink.eww file.
2. Select Project > Make (F7) from inside of EWARM.
3. After making the project, the firmware file used for programming is found at "build\sink-cortexm3-iar-em357-em3xx-dev0680\sink.ebl (an application image in Ember bootloader file format)
4. To program this file to the node, connect your Debug Adapter (ISA3) to the chip and your computer (this example assumes the Debug Adapter (ISA3) is connected via USB. See the next section for how to program if the Debug Adapter (ISA3) is connected via Ethernet.
5. Open a command line of your choice (such as DOS or cygwin).
6. Navigate to the directory where the sink.ebl file was built.
7. Invoke the command "em3xx_load.exe" without any build options. This will ensure that em3xx_load.exe can be found in your path and will cause em3xx_load.exe to print its on-line help menu.
8. To download the ebl file to the EM357 chip, invoke the command "em3xx_load.exe sink.ebl". (Note that this application also requires that you load bootloader firmware to the chip. This must be done before the application firmware will run. Bootloader firmware can be found as *.s37 files within the tool/bootloader-em357 or bootloader-em351 directory within your EmberZNet PRO installation and is loaded in a manner similar to that of the *.ebl file above. If bootloader firmware is not already present on the chip when the application is loaded, em3xx_load will display a warning about a "bootloading and application mismatch".)
9. Download will complete when em3xx_load prints the string "Run (by toggling nRESET)".

8.5 Programming your Application on the EM35x

Programming EM35x flash is accomplished with the `em3xx_load.exe` utility. The `em3xx_load` utility is included as part of the Debug Adapter (ISA3) Utilities Installer, in the "bin" subdirectory. Note that `em3xx_load` is not directly compatible with the options used by `em2xx_load` for the EM250 and EM260 platforms.

A complete summary of `em3xx_load` functionality with brief descriptions of each option can be found in `em3xx_load`'s own help menu. Invoking "`em3xx_load.exe`" without options or invoking "`em3xx_load.exe --help`" will print this help menu.

The most common usage of `em3xx_load` is to simply program flash. `em3xx_load` takes the S37 specific version of Motorola S-record file formats, and these files use the `.s37` extension. If you are using a bootloader, `em3xx_load` can also load a `.ebi` file, which is a proprietary, binary file format designed for applications that use an Ember Bootloader.

Given an S37 file called "file.s37", the most basic command for programming flash is simply:

```
em3xx_load.exe file.s37
```

`Em3xx_load` will then print out the versions it is using, the devices it is connected to, the steps it is performing, and programming progress. By default, the last step `em3xx_load` performs is to run the chip by toggling the nRESET pin.

By default, `em3xx_load.exe` assumes the Debug Adapter (ISA3) is connected via USB when no other options are given. If the Debug Adapter (ISA3) is connected via Ethernet, use the `--ip` option to specify the ip address or hostname of the Debug Adapter (ISA3) to be used. Given a Debug Adapter (ISA3) with an IP address of 123.123.123.123, programming can be accomplished with the command:

```
em3XX_load.exe --ip 123.123.123.123 file.s37
```

8.6 Programming Manufacturing Tokens

The `em3xx_load` utility has the ability to program manufacturing tokens much like the `em2xx_patch` tool did. In the case of the EM357, these tokens are contained in the Customer Information Block (CIB). For more information on using the `em3xx_load` utility, refer to document UG107, *EM35x Utilities Guide*. For more information about CIB tokens and how to program them, refer to document AN710, *Bringing Up Custom Devices for the EM35x Platform*.

8.6.1 Programming Certificates & Installation Codes

Certificates and installation codes can be programmed using the same `em3xx_load` tool that is used to program other tokens. There is a separate application note which describes the process in detail. For more information, see document AN708, *Setting Manufacturing Certificates and Installation Codes*.

8.7 Using the Debugger in IAR Embedded Workbench with an Ember Debug Adapter (ISA3)

1. Start IAR Embedded Workbench. Open the workspace (*.eww).
2. Select Project > Options. Select the Debugger > J-Link/J-Trace category. Select the Connection tab.
3. Select TCP/IP Communication and enter the IP address of the Debug Adapter (ISA3). Click OK.
4. Embedded Workbench is not able to program the flash on the EM35x. Every time the project is re-built, you must manually load it onto the EM35x (using `em3xx_load.exe`) before starting a debug session.
5. Once the correct image has been loaded onto the EM35x, select Project > Debug without Downloading to start the debug session.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page for ZigBee products:
www.silabs.com/zigbee-support and register to submit a technical support request

Patent Notice

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories, Silicon Labs, and Ember are registered trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.