UG113

EM34x Development Kits User's Guide

Silicon Labs offers both an EM34x Low Cost Remote Development Kit and an EM34x Full-Feature Remote Development Kit. This document describes the kits and explains how to set up their hardware and use the Debug Adapter (ISA3). For information about developing custom applications based on Silicon Labs reference designs, see QSG105, *Getting Started with Silicon Labs Wireless Network Software* and AN906, *RF4CE ZigBee Remote Control Firmware Guide*, as well as the help and documentation included with your development environment.

New in This Revision

Updated to add the EM34x Full-Feature Remote Development Kit.

Contents

1 About This Guide 4
1.1 Purpose 4
1.2 Audience 4
1.3 Documentation Conventions 4
2 Introducing the EM34x Development Kits 4
2.1 Overview 4
2.2 Support 5
2.3 Documentation 5
2.4 Before You Begin 5
2.5 Hardware Requirements 5
2.6 Software Requirements 5
3 Development Kit Hardware Components 6
3.1 Overview 6
3.2 EM341 Remote Controls 7
3.2.1 EM341 Low Cost Remote Control 7
3.2.2 EM341 Full-Feature Remote Control 9
3.3 EM35x Breakout Board 11
3.4 EM34x Modules 12
3.5 Ember Debug Adapter (ISA3) 13
3.6 Telegesis USB Stick (Full Feature Remote Kit Only) 13
3.7 Other Kit Components 13
4 Acceptable Power Sources for Normal Operation 14
5 Software 16

 Copyright © 2015 by Silicon Laboratories

5.1 Network Stack 16
5.2 Hardware Abstraction Layer 16
5.3 Sample Applications 16
5.3.1 Ember Desktop 17
6 Setting up Hardware Components 18
6.1 Overview 18
6.2 EM34x Radio Communications Module (RCM) 18
6.3 Hardware Connections 18
6.4 Connecting the Breakout Board and RCM 19
6.5 Breakout Board Power Configurations 19
6.6 Breakout Board Serial Configurations 21
6.7 Common Hardware Configurations 22
6.8 Connecting the RF Cable to the RCM 26
7 Using the Ember Debug Adapter (ISA3) 27
7.1 Overview 27
7.2 Connectivity and Power 27
7.3 Ethernet Settings 28
7.4 Using the Admin Interface over USB 28
7.5 Using the Admin Interface over Ethernet 28
7.6 Using the Serial Ports 28
7.7 Updating the Firmware 29
8 Ember Debug Adapter (ISA3) Commands 30
8.1 Syntax Conventions 30
8.2 Entering Commands 30
8.3 Command List 30
8.3.1 button 30
8.3.2 debug 31
8.3.3 config 31
8.3.4 connections 31
8.3.5 help 31
8.3.6 hostname 32
8.3.7 ip 32
8.3.8 port 32
8.3.9 power 32
8.3.10 reset 33

8.3.11 usbaddr 33
8.3.12 version 33
9 Software Tools Overview 34
9.1 Programming your Application on the EM34x 34
9.2 Programming Manufacturing Tokens 34
9.3 Programming Certificates & Installation Codes 34
9.4 Using the Debugger in IAR Embedded Workbench with an Ember Debug Adapter (ISA3) 35

1 About This Guide

1.1 Purpose

Silicon Labs offers an EM34x Low Cost Remote Development Kit and an EM34x Full-Feature Remote Development Kit. This document describes the kits and explains how to set up their hardware and use the Debug Adapter (ISA3).

This guide assumes that you have already:

- Opened and set up your development kit as described in the Quick Start Guide (QSG) included with the kit.
- Registered your kit and downloaded the software stack as described in the QSG.
- Downloaded the remaining software components as described in the release notes installed with the software stack.

1.2 Audience

This document is intended for embedded software and hardware engineers who are responsible for developing wireless applications with the one of the EM34x development kits. 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
Italics	Identifies on-screen software menu options.	*Connect*
UPPERCASE	Identifies a keyboard key.	ENTER
Right-angle bracket	Delimits a series of software program menu options to be clicked.	*Open* > *Save*
`Courier`	Identifies software code and, in body text, variables.	`void Main(String[] argv)` `the buffer variable`
`Courier Italics`	Identifies a variable name.	*`ipAddress`*

2 Introducing the EM34x Development Kits

2.1 Overview

The EM34x Development Kits are used for evaluation and application design and development. The kits contains a variety of hardware for quickly creating your own prototype, EmberZNet, Silicon Labs' implementation of the ZigBee ZRC stack software specification, and various development tools that enable the development of the application software.

EmberZNet runs on top of an IEEE 802.15.4-compliant radio and link layer software. Combined with the powerful IEEE 802.15.4 compliant EM34x radios, the EmberZNet network stack provides complete networking services from the physical layer up to a reliable application profile support layer. The stack produces ZRC networks that are reliable, flexible, secure, and easy to use.

EmberZNet provides a common API (Application Programming Interface) that supports the following layers of the OSI model (Open System Interconnection Reference Model):

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

Each EM34x development kit includes:

- Ember Desktop, which facilitates application development and debugging.
- The 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 EM35x Breakout Board, see document TS6, *EM35xx Breakout Board Technical Specification*.
- Finally, the remote controls provide a low cost, form factor reference design.

A compiler, such as IAR-EWARM, is required for development. A 30 day evaluation of IAR is available directly from IAR. See the QSG for more info on how to obtain an evaluation of the IAR-EWARM.

2.2 Support

Once you receive your development kit, register at https://siliconlabs.force.com/KitRegistration to download the development software from the Silicon Labs' technical support portal, https://www.silabs.com/support. Use the support portal to contact Customer Support for any questions you might have during the development process.

2.3 Documentation

A quick start guide is shipped with the development kit. The stack installer provides a documentation index (in documentation/index.htm and also linked from a Start Menu entry) that contains links to documentation locations and brief descriptions of each document's purpose. Documents are also available through the support portal as Adobe PDF files.

2.4 Before You Begin

The EM34x Development Kit is designed only for the operating conditions and interfaces specified in documents TS7, *Debug Adapter (ISA3) Technical Specification*, TS6, *EM35x Breakout Board Technical Specification*, and TS8, *EM3xx Module Technical Specification*. Any modifications or alterations to the hardware are liable to cause irreparable damage to the EM34x Development Kit, and shall render its warranty null and void.

If you wish to use the EM34x 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.

2.5 Hardware Requirements

- PC with Ethernet connectivity
- Network hub with DHCP server

2.6 Software Requirements

- Microsoft Windows XP or Windows 7
- Java, version 1.6 or later
- Adobe Acrobat Reader (available free from http://get.adobe.com/reader/)

3 Development Kit Hardware Components

3.1 Overview

Figure 1 shows the key components in the EM34x Low Cost Remote Development Kit (EM34X-DEV).

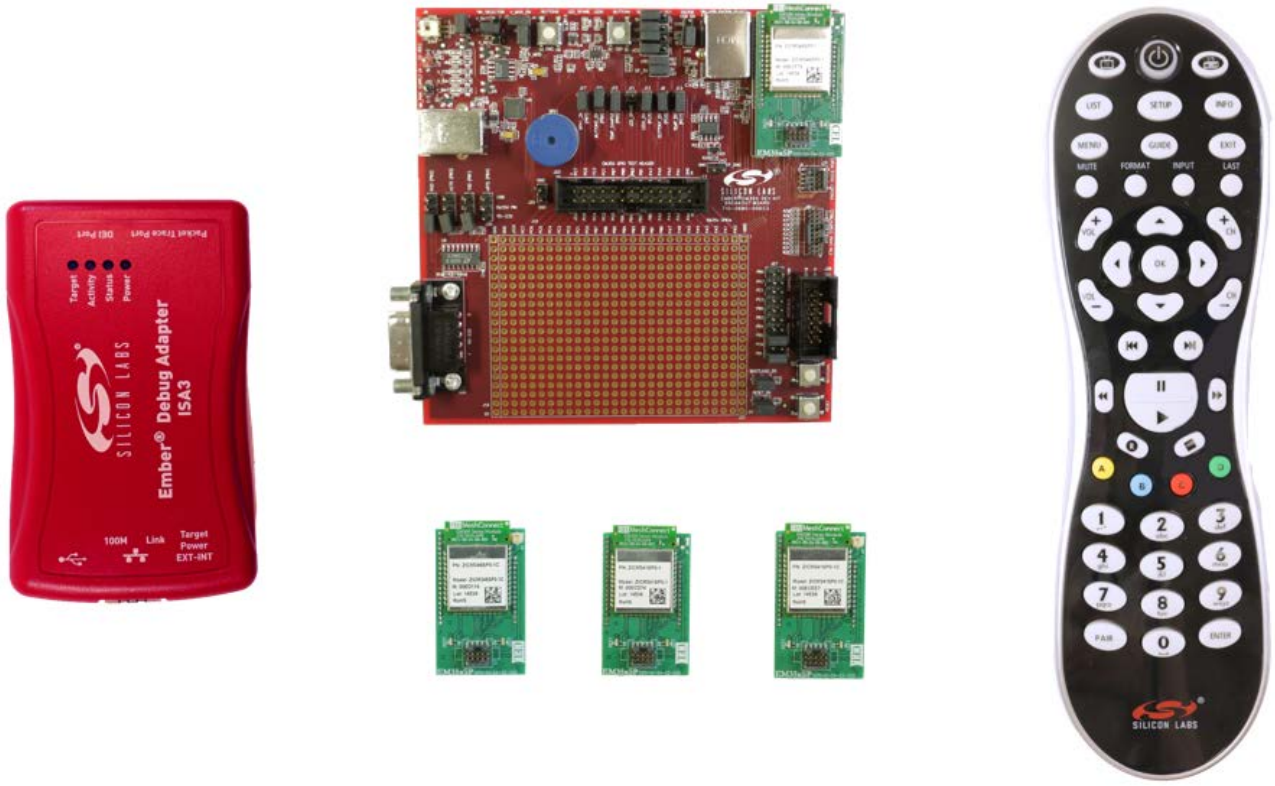


Figure 1. EM34x Low Cost Remote Development Kit

The inventory of development kit components is included in the development kit's Quick Start Guide. This chapter describes the hardware components in the EM34x Series Development Kit.

The EM34x development kits have four major hardware components:

- Either an EM341 Full-Feature or a Low Cost Remote Control
- EM35x Breakout Board
- EM34x Modules
- Ember Debug Adapter (ISA3)

The Full-Feature Remote kit also contains a Telegesis USB stick.

As well as these major components, each EM34x development kit also contains:

- USB Power Adapter
- Miscellaneous cables (Ethernet, USB, DEI, Packet Trace Port, and RF Coax)

3.2 EM341 Remote Controls

The **Remote Controls** provide a form factor reference design. Bothe remotes are programmable via the Packet Trace connector and the Ember Debug Adapter (ISA3). They provide an ideal solution to quickly prototype and develop a fully functioning ZRC remote control application. For information about both Remote Controls, see the reference design files at http://www.silabs.com/products/wireless/zigbee/Pages/zigbee-reference-designs.aspx

3.2.1 EM341 Low Cost Remote Control

The Low Cost Remote Control has the following features:

- 47 buttons
- Bi-color backlit Power button
- IR support
- Packet Trace connector (under battery cover)
- Powered by 2x AAA batteries

The case is secured by 8 locking tabs (see Figure 2 for approximate location). Care should be taken if opening the case as the tabs may break. The remote must be in the case for the batteries and keyboard to function properly.

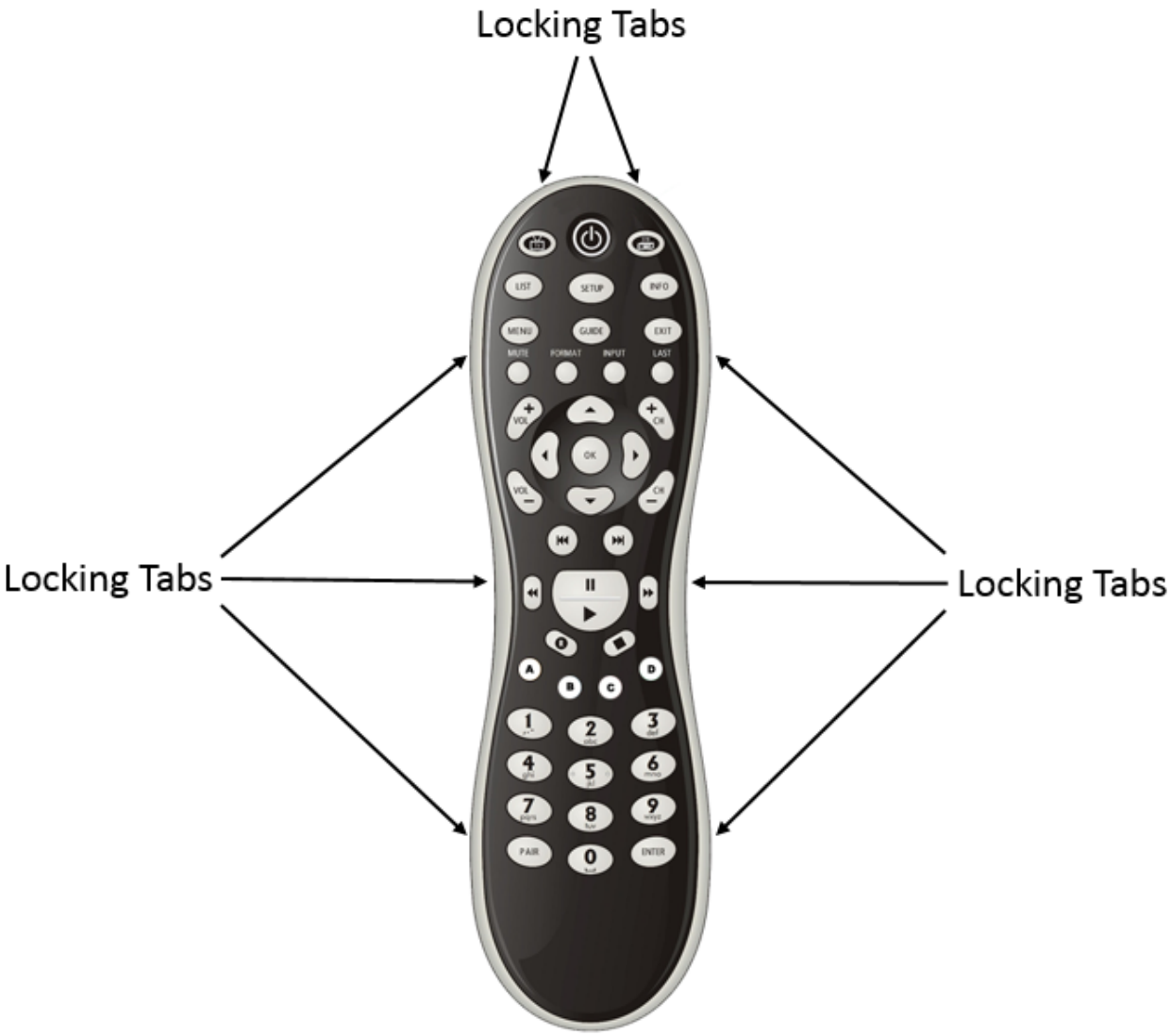


Figure 2.

Figure 2. Low Cost Remote Control

3.2.2 EM341 Full-Feature Remote Control

The Full Feature Remote Control has the following features:

- 47 buttons
- Backlit Power button
- Backlit keyboard
- Acceleration sensor (for backlight control)
- Codec and Microphone for voice support
- IR Support
- Packet Trace connector (under battery cover)
- Powered by 2x AA batteries

The remote comes standard with the hardware codec. Modification is required in order to support the software codec. Below is the minimum procedure to rework the PCB to support the software codec. The reference design files reflect both BOM options.

1. Open the case and remove the PCB.

   The case is secured by 14 locking tabs (see Figure 3 for approximate location). Care should be taken if opening the case as the tabs may break. The remote must be in the case for the batteries and keyboard to function properly.
2. Remove Q6 (PFET) and U6 (codec) from the PCB.
3. Install R8 (10-ohm/0402/±5%), R40 (330-ohm/0402/±5%), and R41 (330-ohm/0402/±5%).
4. Clean any flux and residue from the rework from the PCB.
5. Reassemble the PCB into the case.
6. Reprogram the remote with the software codec firmware.

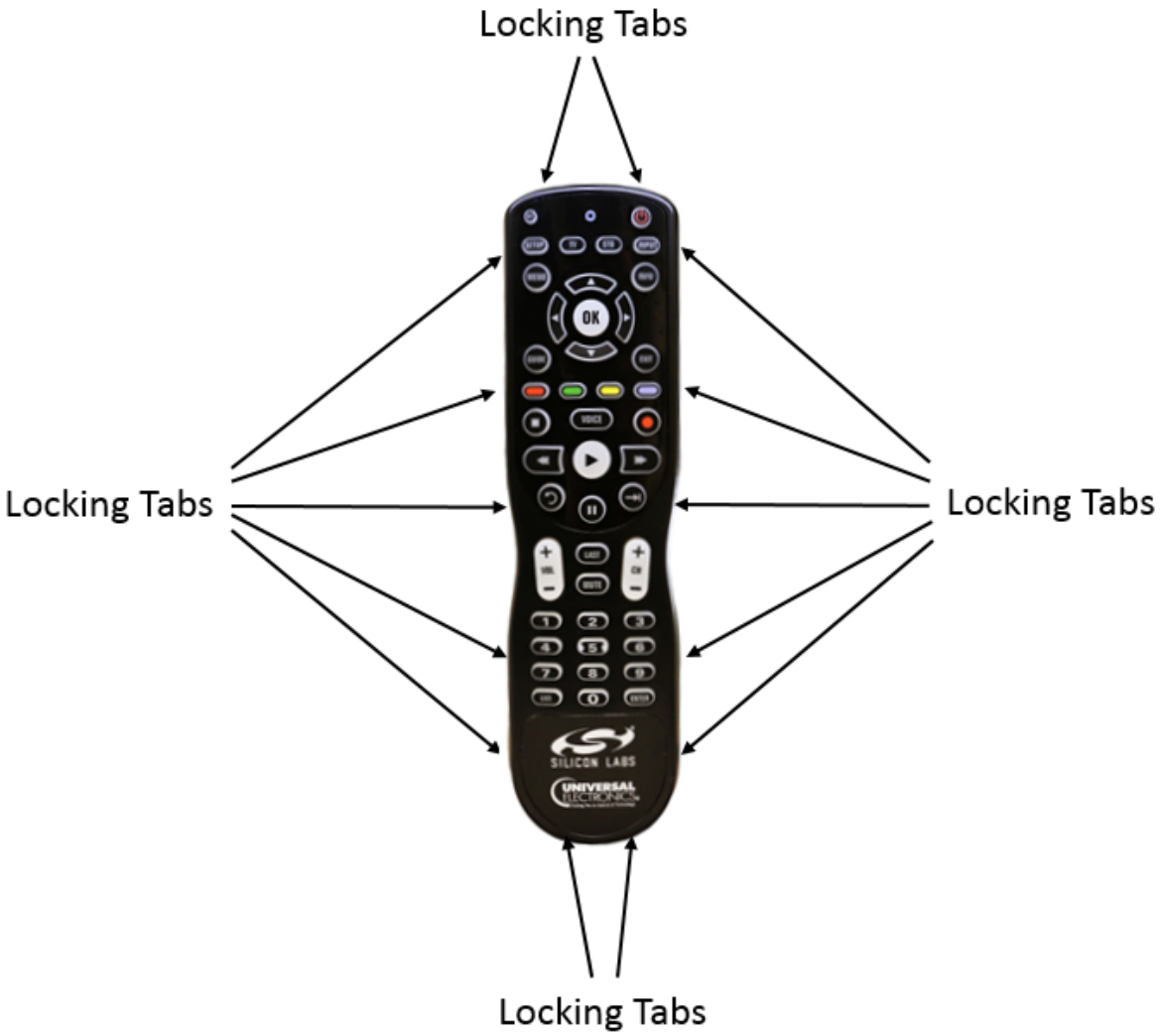


Figure 3. Full-Feature Remote Control

3.3 EM35x Breakout Board

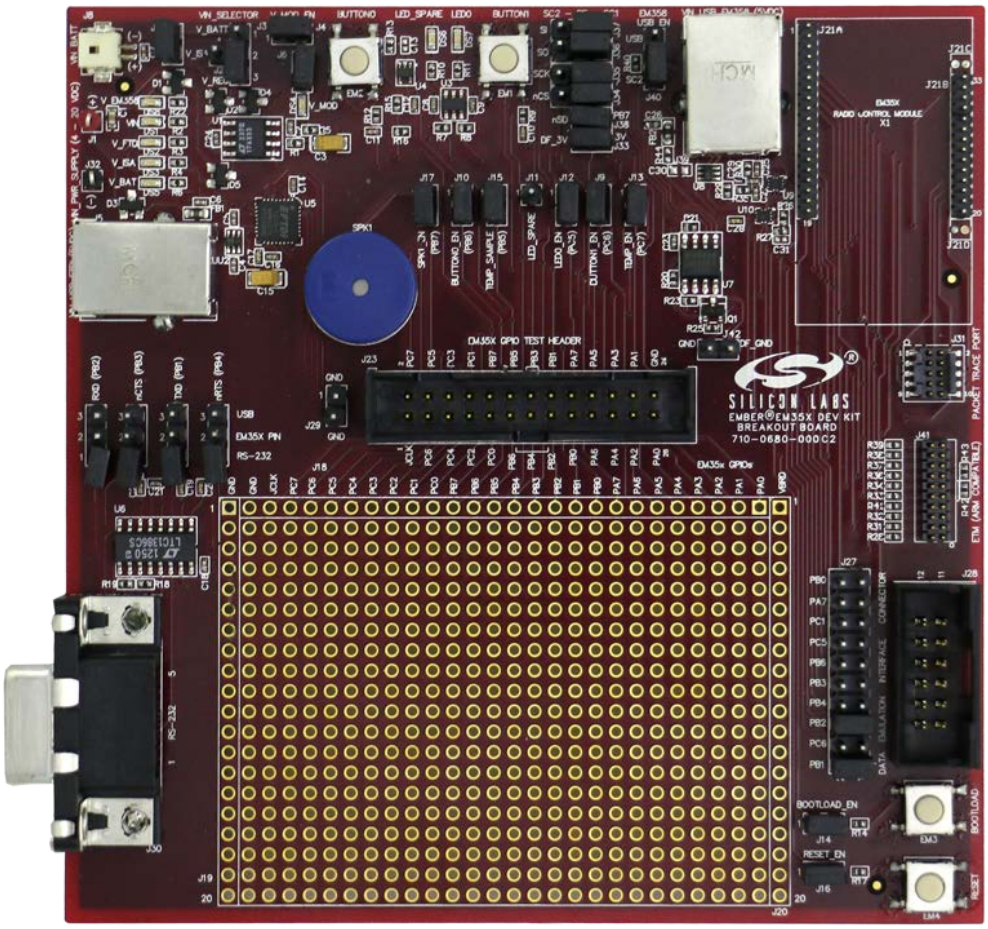


Figure 4. EM35x Breakout Board

The **Breakout Board** (Figure 4) serves as a test bed for network hardware and applications. The Breakout Board will have an associated EM34x Module, which can be deployed as a node within a network application. The 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 EM34x 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 Board also has a prototype breadboard area for application-specific hardware development. The Breakout Board has the following components:

- External power supply connectors
- One DB-9 serial port connector for RS-232 serial communication
- One USB connector for EM35x SC1 virtual COM port serial communication
- One header with TTL-compliant UART signals for EM35x SC1 SPI serial communication
- A buzzer, two buttons, two LEDs, and a temperature sensor for application development purposes
- Hardware reset buttons for the EM34x module
- Prototype or bread-boarding area (28 x 20, 0.1" pitch array) for application-specific hardware development
- One data emulation interface connector

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

3.4 EM34x Modules

EM34x modules contain the EM34x System-on-Chip (SoC) IC providing a complete radio and microcontroller solution. The kit includes both EM341 (ZRC Controller) and EM346 (ZRC Target) module variants. The EM34x module pictured in Figure 5 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.

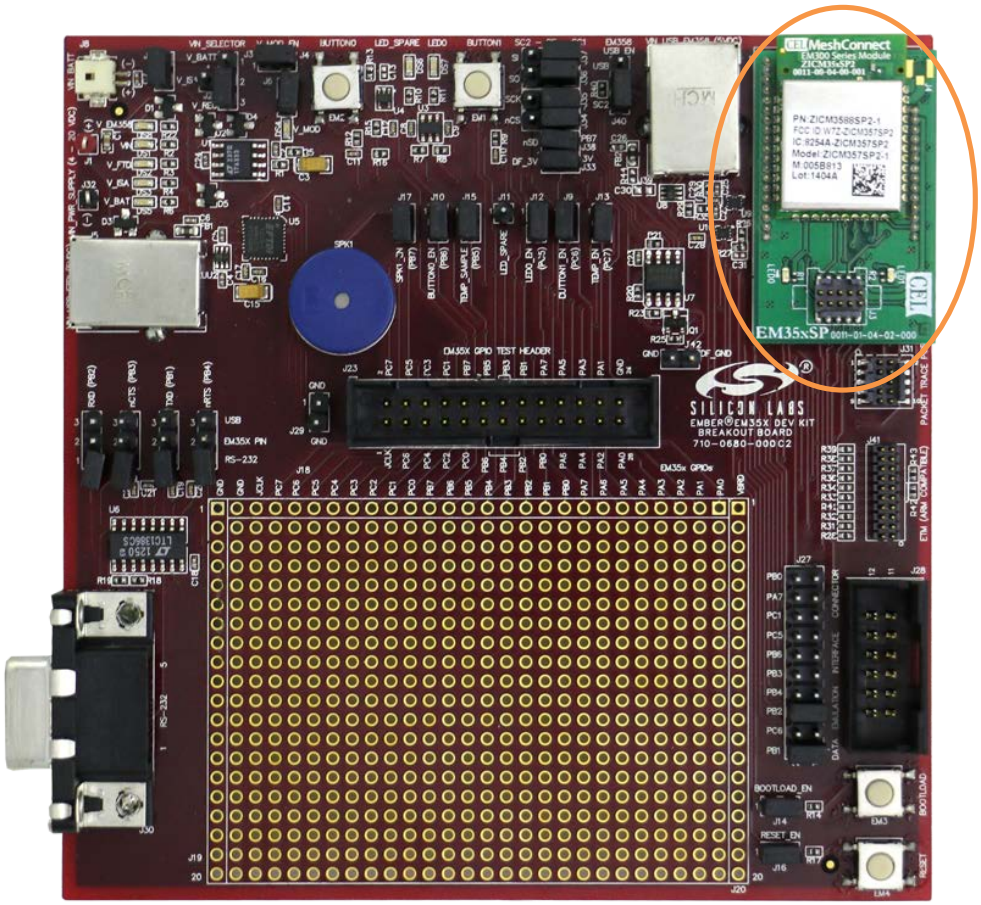


Figure 5. EM35x Breakout Board with EM34x Module installed

The EM34x Module includes these components:

- The Ember EM34x 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

For detailed information about the modules, see document TS8, the *EM3xx Module Technical Specification*.

Once you move away from the EM3xxx Module and toward application-specific hardware, refer to the extensive Ember reference design library available through the Silicon Labs support portal. 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 Customer Support.

3.5 Ember Debug Adapter (ISA3)

The **Ember Debug Adapter (ISA3)** connects a Breakout Board to the Ethernet. The debug 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 the Breakout Board. The debug adapter 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 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

3.6 Telegesis USB Stick (Full Feature Remote Kit Only)

The **Telegesis USB Stick** (Figure 6) contains an EM3588 chip preprogrammed with an example target application designed to show some of the remote's functionality.

Figure 6. Telegesis USB Stick

3.7 Other Kit Components

- The **Packet Trace Port Cables** connect each Debug Adapter (ISA3) to a Breakout Board, Telegesis USB stick, or a Remote Control. When connecting the Packet Trace cable to the USB Telegesis stick, you must use the included male header connector. The Packet Trace Port is a cabled communication link between the Debug Adapter (ISA3) and the EM34x SoC on the module, remote control and USB stick. It can also be used to program the EM34x SoC.
- **USB Power Supply** powers Breakout Boards and / or Debug Adapters (ISA3s) from wall sockets.
- **Ethernet Cables** connect Debug Adapters (ISA3s) to an Ethernet subnet,
- **Data Emulation Interface (DEI) cables** provide a peripheral interface connection between the Ember Debug Adapter (ISA3) and Breakout Board. The DEI cable supports UART functionality.

4 Acceptable Power Sources for Normal Operation

Table 1 lists all of the components and the acceptable combinations of power sources that yield normal operation for the EM35x Breakout Board. Potential power sources include the following:

- 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 1. Safe Combination of Operating Power

	PoE	External DC	Packet Trace Port	USB	Breakout Board
Debug Adapter (ISA3)	X	N/A	N/A	X	N/A
EM34x Module	N/A	N/A	X[1]	N/A	X
EM35x Breakout Board	N/A	X[2]	X[3]	X	N/A

Notes:

1. 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).
2. External dc power between 4 V and 20 V can be connected to clip leads J1 (+) and J32 (-) on the EM35x Breakout Board.
3. 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).

The USB power adapter (Figure 7) is a linear regulated wall plug power supply with 5 V dc at 1 A regulated output. It has a USB type-B connector for connecting power via a USB cable to the USB port of the Ember 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 7. USB Power Supply with Interchangeable AC Adapters

5 Software

The EM34x Development Kit provides customers with access to the following software components:

- Network Stack
- Hardware Abstraction Layer
- Ember Application Framework with sample scenarios
- Ember Desktop

5.1 Network Stack

The EmberZNet provides an advanced implementation of the ZigBee Remote Control 2.0 specification. ZigBee Remote Control provides a global standard for advanced RF remotes that removes line-of-sight restrictions while also delivering two-way communication and longer range

5.2 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 combination of pre-built libraries for complex, stack-critical functionality and C source code that you can alter in order to customize, extend, or reduce device functionality across various hardware platforms.

5.3 Sample Applications

The Ember ZNET ZRC software contains sample applications. These demonstrate aspects of the ZRC stack, based on the ZRC 2.0 specification. Silicon Labs provides an IAR project and workspace file for each application as well as a S.37 binary file. Table 2 lists the sample applications that are in the current distribution. All applications are in the app directory.

Table 2. Development Kit Sample Applications

Application name		Description
ZRC 2.0 Controller		The ZRC2.0 controller application demonstrates basic ZRC network functionality and ZRC2.0 application functionality. This application acts as a generic remote control. It utilizes the ZRC Profile plugin to provide core application-level ZRC functionality and the ZRC GDP and ZRC2.0 plugins to provide ZRC2.0-specific behavior.
ZRC 2.0 Target		The ZRC2.0 target application demonstrates basic ZRC network functionality and ZRC2.0 application functionality. This application acts as a generic target device. It utilizes the ZRC Profile plugin to provide core application-level ZRC functionality and the ZRC GDP and ZRC2.0 plugins to provide ZRC2.0-specific behavior.
Full Features Controller		ZRC 2.0 controller application for use on the Full Featured Remote Control.
Full Featured Target		ZRC 2.0 target application for use on the breakout board with the EM346 module, or on the EM3588 Telegesis USB stick.
Low Cost Controller		ZRC 2.0 controller application for use on the Low Cost Remote Control

The target will automatically start the network. The controller is responsible for initiating pairing to an existing network. The controller application will disable its receiver automatically to conserve power. This is representative of a sleepy device that is generally a transmit-only device that acts when directed by the user (i.e., a button press).

The controller application is allowed to pair to two targets, i.e., a set-top box (STB) and a TV target. The target

application acts as either a STB or a TV target and can accept up to maximum 5 pairings.

Pressing and holding the **[PAIR]** button (low cost remote) or **[SET]** button (full-feature remote) and then pressing the **[STB]** button on the remote control will cause the device to initiate the binding process to discover, pair, and validate with a target. .

Note: When remote control is connected to the debug adapter, the controller firmware will auto-remap some key-matrix GPIO pins for debugging and some buttons will not be available. The key combination for pairing when debugging are changed as shown below:

- Press SETUP + TV keys to bind to the TV target
- Press SETUP + STB keys to bind to the STB target

The ZRC GDP and ZRC 2.0 plugins manage discovery, pairing, and validation and will notify the application when binding completes. If the binding completes successfully, the controller can begin sending actions to the target.

Pressing the STB or the TV button switches the controller to STB or TV mode accordingly. Buttons pressed in STB mode will be sent to the STB target and buttons pressed in TV mode will be sent to the TV. Depending on the mode, either an RF or IR signal will be generated. Some keys (such as power) may send both an RF command and an IR command.

When actions are sent, the ZRC ZRC2.0 plugin will keep track of timing in order to automatically repeat actions for keys that are held down.

The target provides feedback by playing tunes. A rising two-tone tune indicates a successful operation while a falling two-tone tune indicates a failure. A brief tone indicates that the device has performed a long-running action and is waiting for a result.

All application code is contained in the ZRC-callbacks.c files within each application directory.

5.3.1 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. 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:

Application Builder, sometimes referred to as AppBuilder, is an interactive GUI tool that allows you to configure a body of Silicon Labs-supplied code to implement devices, such as a ZRC 2.0 controller or a ZRC 2.0 target, within an application profile.

Network Analyzer manages the Development Kit hardware and displays network and node activity in real time. It includes:

- Provides 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.

For detailed information about these tools and the development environment itself, refer to its online help.

6 Setting up Hardware Components

6.1 Overview

This chapter contains the following sections:

- EM34x 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

6.2 EM34x Radio Communications Module (RCM)

The EM34x Series Development Kit contains various types of EM34x Radio Communications Modules designed by Silicon Labs' partner, California Eastern Laboratories (CEL). These modules contain either the EM341 or EM346 IC and are configured to use either the PCB antenna or the on-board RF connector (U.FL). These configuration options are noted in the Module Product Label, as shown the example label in Figure 8. This label can be found on the bottom of the module. The top line contains the Silicon Labs Part Number, while the bottom line contains the Module Serial Number.

Figure 8. Module Product Serial Number Label

The module shipped already installed on the Breakout Board is module variant EM346-MOD-ANT-C. All other module variants are contained in the EM34x Development Kit. Refer to Table 3 for configuration comparisons of the modules included in your development kit.

Table 3. Module Product Variant Matrix

Silicon Labs Part Number	CEL Part Number	Silicon Labs Product Number	IC	Antenna or Connector	PA or non-PA	Qty Included in Kit
EM341-MOD-ANT-C	ZICM341SP0-1-SL	0740	EM341	Antenna	Non-PA	1
EM341-MOD-RF-C	ZICM341SP0-1C-SL	0741	EM341	Connector	Non-PA	1
EM346-MOD-ANT-C	ZICM346SP0-1-SL	0742	EM346	Antenna	Non-PA	1
EM346-MOD-RF-C	ZICM346SP0-1C-SL	0743	EM346	Connector	Non-PA	1

6.3 Hardware Connections

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

6.4 Connecting the Breakout Board and RCM

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

To remove the radio communication module:

Carefully pull the RCM away from the Breakout Board.

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 attach the Debug Adapter (ISA3):

7. Plug the Packet Trace Port cable into the Debug Adapter (ISA3).
8. 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.
9. 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).

6.5 Breakout Board Power Configurations

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

- Breakout Board Power Supplied by Debug Adapter (ISA3) (Figure 9): 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 (DS3) will illuminate when power is applied.

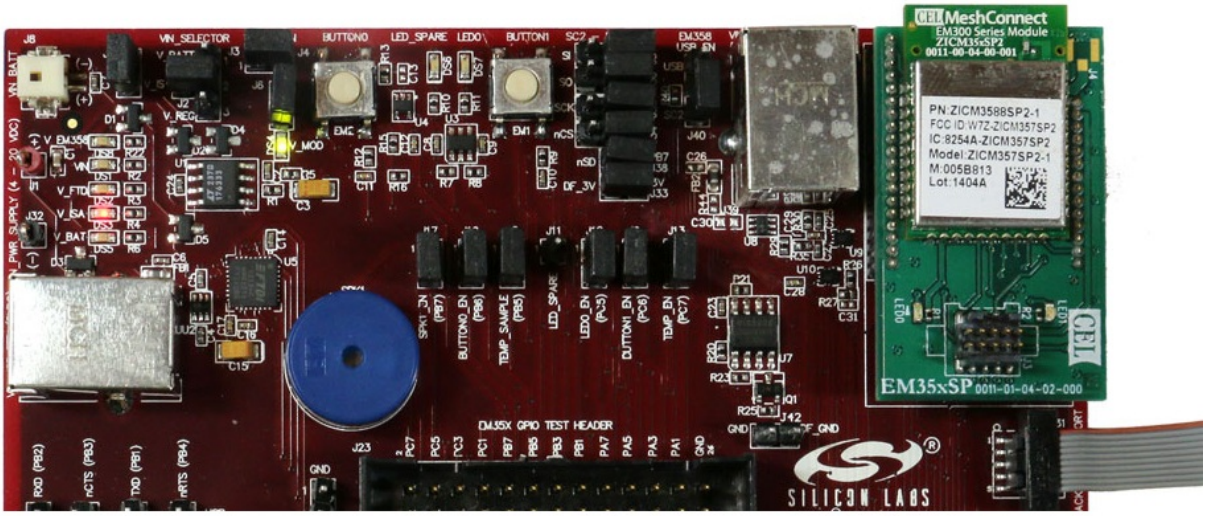


Figure 9. Breakout Board Power Supplied by Debug Adapter (ISA3)

- Breakout Board Power Supplied by USB (Figure 10): The Breakout Board may also be powered through the USB connector (J5), 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_FTDI LED (DS2) will illuminate when 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.

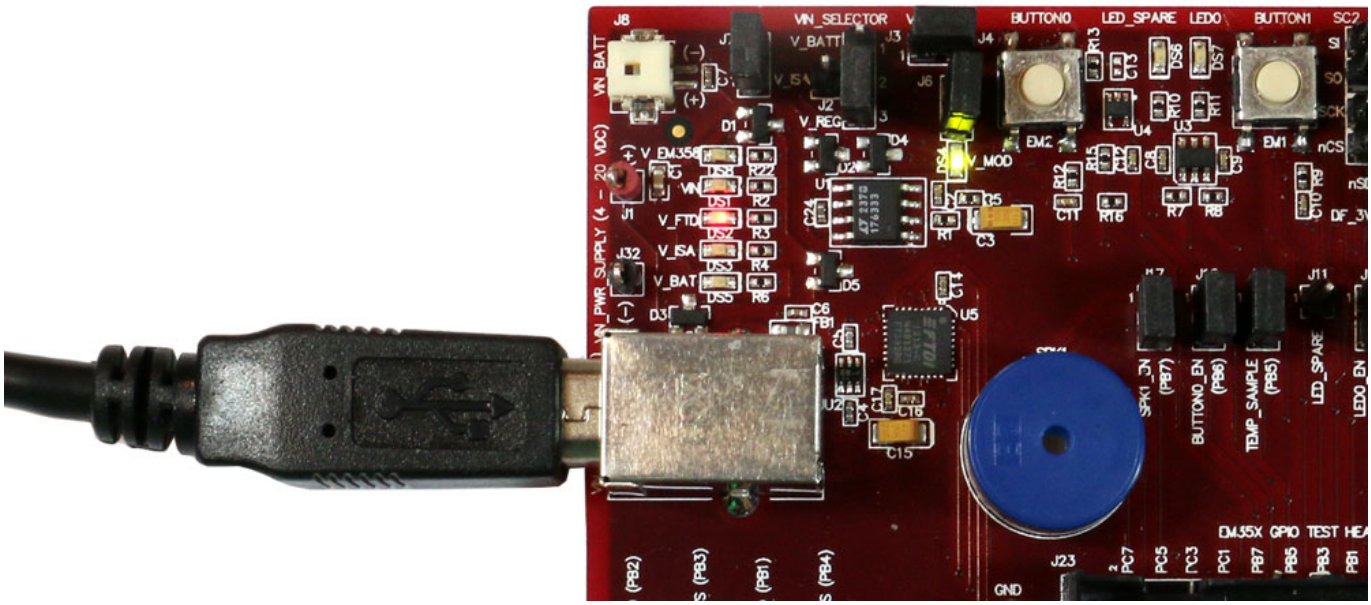


Figure 10. Breakout Board Power Supplied by USB (J5)

- Breakout Board Power Supplied by DC Jumpers (Figure 11): 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 (DS1) 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.

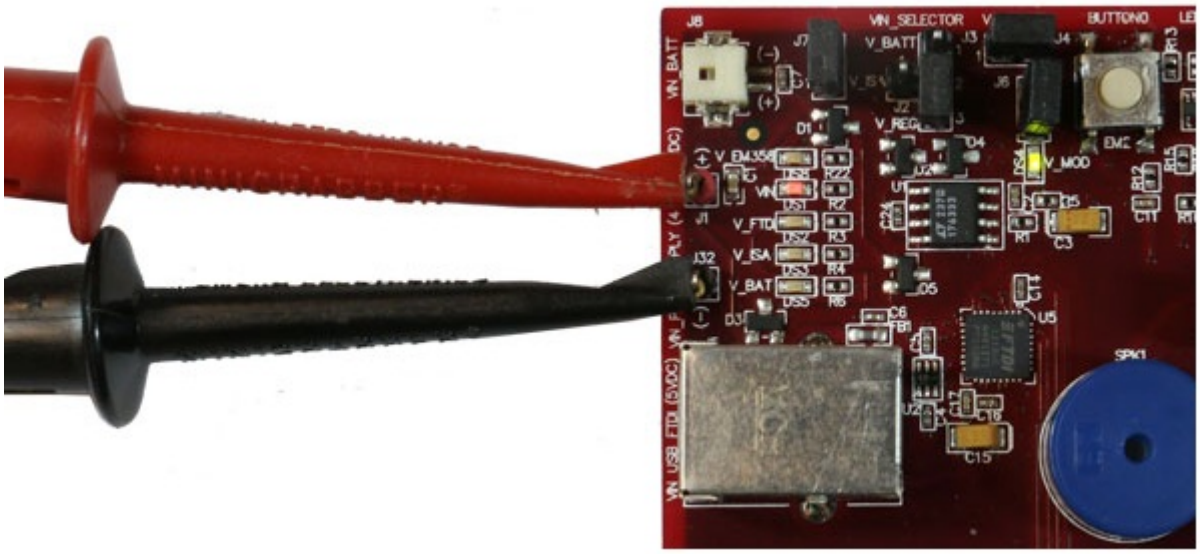


Figure 11. Breakout Board Power Supplied by DC Jumpers

6.6 Breakout Board Serial Configurations

Serial connectivity to the Breakout Board can be made in a number of ways. The default configuration of your development kit will match your expected usage scenario.

- **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 as described in the stack release notes, and connect a USB cable to J5 of the Breakout Board and to 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.

6.7 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.

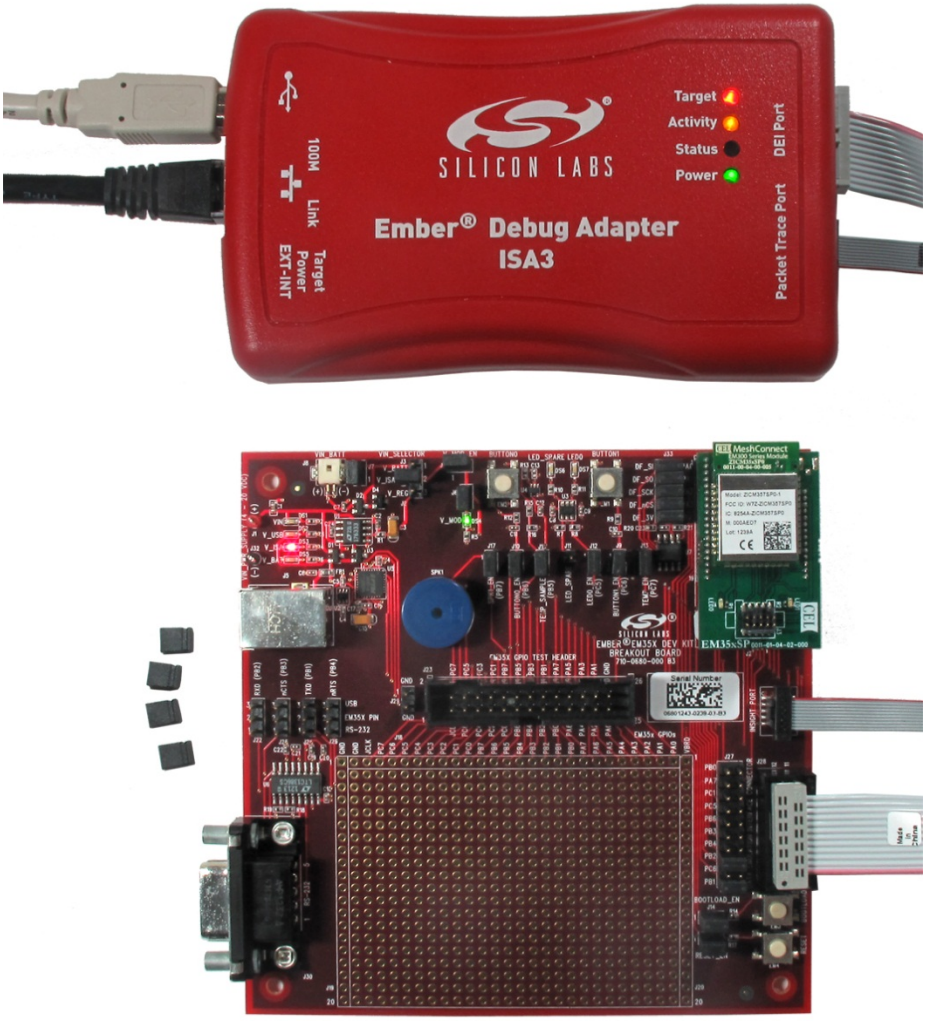


Figure 12. Ember Debug Adapter (ISA3) Supplying Power; Pass-through UART Being Used for Serial Communication

Figure 13. Ember Debug Adapter (ISA3) Connecting Packet Trace to Remote Control for programming

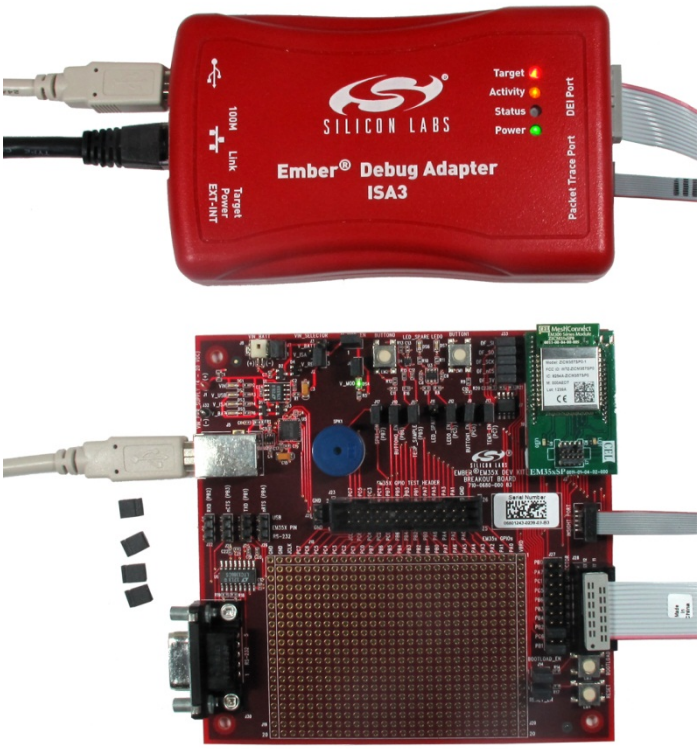


Figure 14. Power Supplied Via USB Connector; Pass-through UART Being Used for Communication

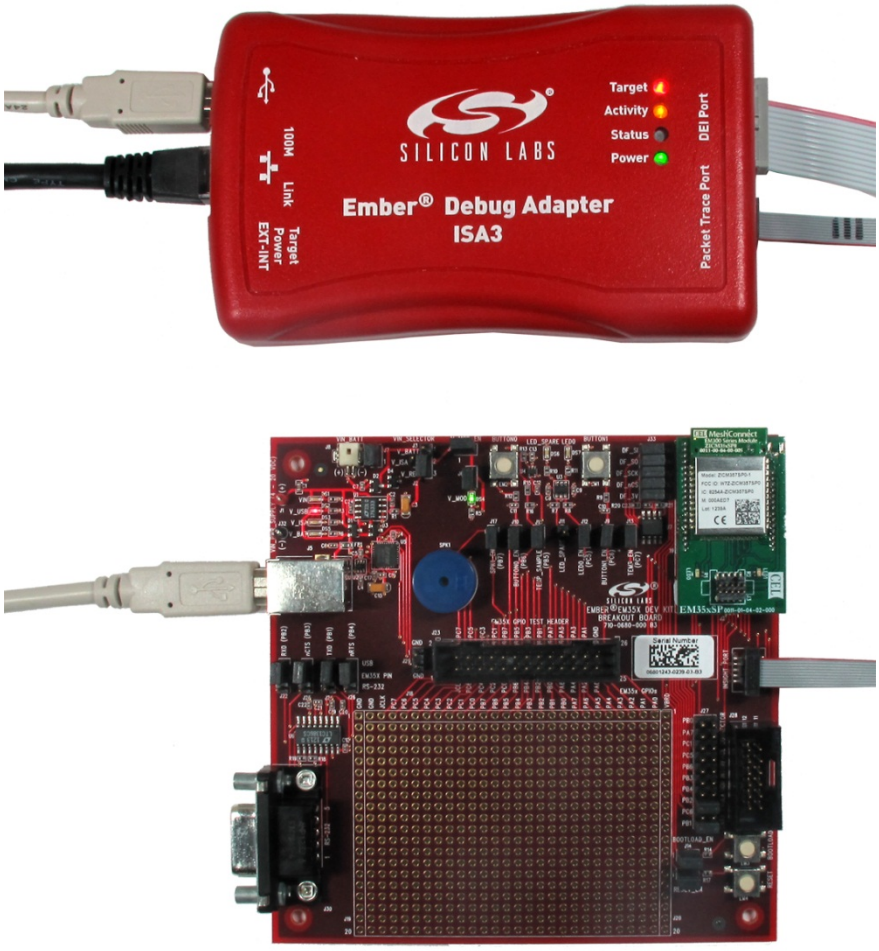


Figure 15. Ember Debug Adapter (ISA3) Supplying Power; FTDI Serial <-> USB Converter Used for Communication

6.8 Connecting the RF Cable to the RCM

As detailed in Table 3, 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

7 Using the Ember Debug Adapter (ISA3)

7.1 Overview

The Ember Debug Adapter (ISA3) provides an Ethernet connection to EM35x devices. You can use the debug adapter 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.

7.2 Connectivity and Power

All the features of the debug adapter are available via the Ethernet connection. The debug adapter can also be connected to a PC via USB to change the Ethernet settings.

When connected via USB, the debug adapter 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 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 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 4 using either Ember Desktop or a terminal application (for example, Microsoft Telnet Client, HyperTerminal, or PuTTY).

Table 4. Ember 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.

7.3 Ethernet Settings

By default, the Ember Debug Adapter (ISA3) is configured to get an IP address via DHCP. It is assigned a hostname of EM-ISA3-00. You can use Ember Desktop to discover the debug adapter on the network and see its IP address.

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>`.

7.4 Using the Admin Interface over USB

1. Connect the debug adapter to the PC using USB.
2. Run em3xx_isa.exe with the --admin option.
3. Configure the debug adapter using the commands described in Section 8, Ember Debug Adapter (ISA3) Commands.

7.5 Using the Admin Interface over Ethernet

1. Connect the debug adapter to the network using Ethernet.
2. Connect to the admin port on the debug adapter) using either Ember Desktop or a terminal application:
   - Connect to the debug adapter using Ember Desktop. Right click on the EM-ISA3 device and select Launch Console. Choose the Admin tab

   OR

   - 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 using the commands described in Section 8, Ember Debug Adapter (ISA3) Commands.

7.6 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.

7.7 Updating the Firmware

Periodically, Silicon Labs provides updated firmware for the Ember 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 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
```

8 Ember Debug Adapter (ISA3) Commands

8.1 Syntax Conventions

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

8.2 Entering Commands

- Commands can be entered over USB (see Section 7.4, “Using the Admin Interface over USB”) or Ethernet (see Section 7.5, “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.

8.3 Command List

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

8.3.1 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
```

8.3.2 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
```

8.3.3 config

```
config
```

Description

Displays the debug adapter's configuration:

- 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.

8.3.4 connections

```
connections
```

Description

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

Example

```
connections
```

8.3.5 help

```
help
```

Description

Displays a list of all debug adapter commands.

8.3.6 hostname

```
hostname set hostname
hostname read
```

Description

Sets or displays the hostname).

The debug adapter must be reset for a change to take effect.

You can also view the hostname using Ember Desktop.

Example

```
hostname set mydevice3
```

8.3.7 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 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
```

8.3.8 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
```

8.3.9 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
```

8.3.10 reset

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

Description

Resets the attached EM35x device if no arguments are specified

Resets the debug adapter 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.

8.3.11 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 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).

8.3.12 version

```
version
```

Description

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

You can also view the version using Ember Desktop.

9 Software Tools Overview

See the development kit Quick Start Guides for instructions on installing and running the remote control precompiled target and sample applications. See QSG105, *Getting Started with Silicon Labs Wireless Networking Software*, for information on how to configure and compile your own applications based on the reference designs described in AN906. *RF4CE ZigBee Remote Control Firmware Guide*,

9.1 Programming your Application on the EM34x

Programming EM34x flash is accomplished with the em3xx_load.exe utility, either via the Windows Command Prompt or via the Upload Application action in Ember Desktop's Adapters view. The em3xx_load utility is included as part of the Debug Adapter (ISA3) Utilities Installer, in the "bin" subdirectory.

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. More information about using Ember Desktop to upload applications can be found in the *Ember Desktop User's Guide* and in the online help for Ember Desktop

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 .ebl 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
```

9.2 Programming Manufacturing Tokens

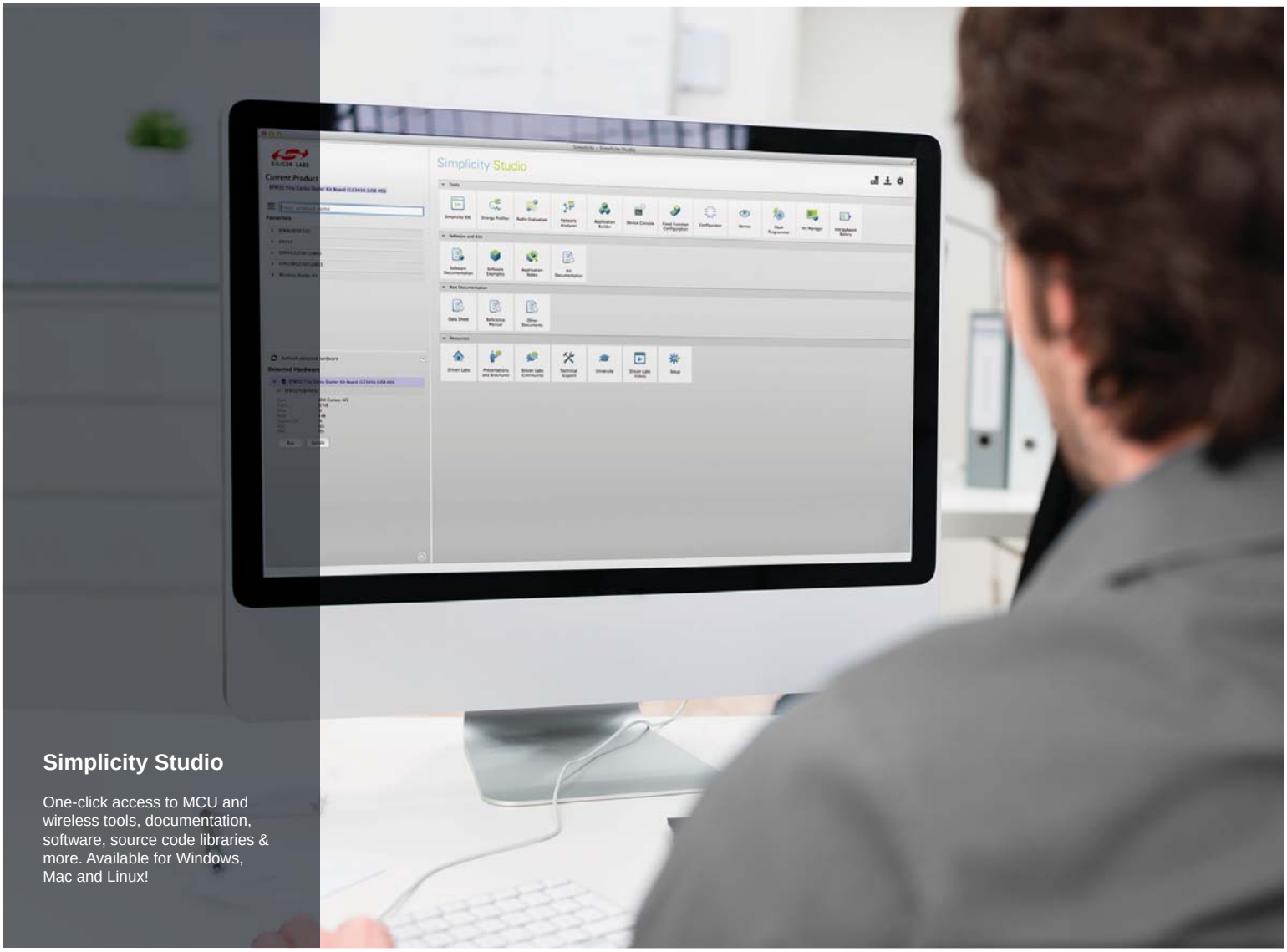
The em3xx_load utility has the ability to program manufacturing tokens. In the case of the EM34x, these tokens are contained in the Customer Information Block (CIB). For more information on using the em3xx_load utility, refer to document UG107, *EM35xx Utilities Guide*. For more information about CIB tokens and how to program them, refer to document AN710, *Bringing Up Custom Devices for the EM35xx Platform.*

9.3 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*.

9.4 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. Click OK.
4. Embedded Workbench is not able to program the flash on the EM34x. Every time the project is re-built, you must manually load it onto the EM34x (using em3xx_load.exe) before starting a debug session.
5. Once the correct image has been loaded onto the EM34x, select Project > Debug without Downloading to start the debug session



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem ®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com