

Z-Stack

User's Guide For

Chipcon CC2430ZDK

ZigBee 1.0 Release

Version	Description	Date
1.0	Initial release.	12/22/2005

Table of Contents

1. INTRODUCTION.....	1
1.1. SCOPE	1
2. PRODUCT PACKAGE DESCRIPTION.....	1
2.1. INSTALLATION PACKAGE CONTENTS	1
2.2. DEVELOPMENT BOARDS	1
2.3. CABLES.....	2
3. INSTALLATION REQUIREMENTS.....	2
3.1. HOST COMPUTER REQUIREMENTS	2
3.2. TARGET DEVELOPMENT SYSTEM REQUIREMENTS	2
4. PRODUCT INSTALLATION PROCEDURES.....	3
4.1. INSTALL Z-STACK PACKAGE	3
4.2. INSTALL IAR EW8051 PACKAGE	3
4.3. DEVICE IEEE ADDRESSES	3
5. CONFIGURING AND USING Z-STACK.....	3
5.1. CONFIGURING Z-STACK.....	3
5.2. SELECTING THE LOGICAL DEVICE TYPE	3
5.3. SELECTING THE PROFILE.....	4
5.4. BUILDING THE SLC AND SRC SAMPLE DEVICES	4
5.5. BUILDING AN SLC COORDINATOR DEVICE.....	5
5.6. BUILDING AN SRC ROUTER DEVICE.....	8
6. Z-STACK DEMONSTRATION.....	11
6.1. SWITCHES AND LEDs	11
6.2. INITIAL LOADING OF 64-BIT IEEE ADDRESS	12
6.3. NON-VOLATILE MEMORY	12
6.4. RUNNING THE SAMPLE APPLICATION (SLC AND SRC).....	12
6.4.1. SAMPLE USE OF DEVICE BINDING.....	13
6.4.2. SAMPLE USE OF SERVICE DISCOVERY	13
6.4.3. SENDING APPLICATION DATA.....	14
7. DEVELOPING APPLICATIONS WITH Z-STACK.....	14
8. NON-BEACON VS. BEACONS.....	14
9. CHANNEL SELECTION	15
9.1. ENERGY LEVEL.....	16
APPLICABLE DOCUMENTS.....	16

Table of Figures

FIGURE 1: CHIPCON SMARTRF04EB EVALUATION BOARD WITH CC2430EM	1
FIGURE 2: CHIPCON CC2430DB DEVELOPMENT BOARD	2
FIGURE 3: CC2430DB JOYSTICK	11
FIGURE 4: CC2430DB LEDs	11
FIGURE 5: SMARTRF04EB LEDs	12

Table of Tables

TABLE 1: DEFAULT CHANNEL SELECT BIT MAP	15
---	----

1. Introduction

1.1. Scope

This document accompanies the Figure 8 Wireless Z-Stack™ solution. Z-Stack™ is a complete protocol stack and solution conforming to ZigBee Alliance standards (www.zigbee.org).

2. Product Package Description

2.1. Installation Package Contents

The downloaded Z-Stack installation package contains all of the documentation and software required to install, configure, and develop applications using Z-Stack. The package employs a Microsoft Windows-based installation application which guides the installation process.

2.2. Development Boards

Two Chipcon SmartRF04EB evaluation boards, each to be fitted with a CC2430EM evaluation module, are contained in the development kit that is used with the Z-Stack package:



Figure 1: Chipcon SmartRF04EB Evaluation Board with CC2430EM

Five compact Chipcon CC2430DB evaluation boards are also contained in the development kit:

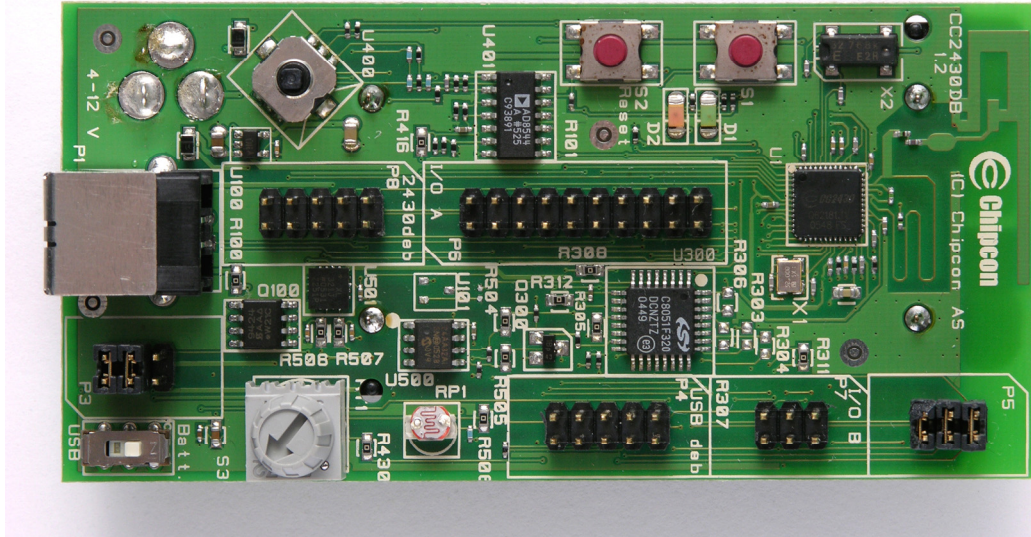


Figure 2: Chipcon CC2430DB Development Board

2.3. Cables

All necessary cabling has been included with the development kit. To support program download and debugging of SmartRF04EB and CC2430DB boards, a USB cable should be connected from each target board to the host PC. RS232 cables may be connected between the serial ports on SmartRF04EB boards (9-pin connectors) and the host PC to utilize Z-Trace and other programs included with the Z-Stack package.

3. Installation Requirements

3.1. Host Computer Requirements

Z-Stack and Z-Trace are designed for installation on a personal computer running Microsoft Windows XP Professional or Windows 2000. The following are the minimum requirements for the platform hosting Z-Stack and Z-Trace:

- .NET 1.1 Framework
- Windows XP Service Pack 1 (if using Windows XP)
- 1 serial port for Z-Trace communication with the SmartRF04EB board
- 1 USB port for download/debug of SmartRF04EB and CC2430DB boards

3.2. Target Development System Requirements

Z-Stack provides a complementary offering to the IAR Embedded Workbench (EW8051) suite of software development tools. These tools support project management, compiling, assembling, linking, downloading, and debugging for various 8051-based processors, including the Chipcon CC2430 family. The following is required support for the Z-Stack target development system:

- IAR EW8051 (<http://www.iar.com/>)

4. Product Installation Procedures

4.1. Install Z-Stack Package

Install the Figure 8 Wireless Z-Stack and Z-Trace files and programs from the downloaded package. Run the windows-based installation program, *CC-2430-IAR-1.0-1.3.0.exe*, to create the required directory structure and to load all software and documentation files. Review the README file for a synopsis of new features and changes with this Z-Stack release.

4.2. Install IAR EW8051 Package

Install the Embedded Workbench for 8051 from IAR Systems: <http://www.iar.com/>. The project and library files included in this release of Z-Stack require the use of EW8051 version 7.20A or newer. When considering an upgrade to a newer version of EW8051, it is necessary to verify that installed project and library files are compatible with the newer development tools. Check the Chipcon website (<http://www.chipcon.com/>) for additional information relating to use of the IAR Systems toolset with the CC2430.

4.3. Device IEEE Addresses

Each CC2430DB and CC2430EM board in the development kit has been pre-programmed with a unique 64-bit IEEE address. These addresses, assigned by Chipcon, are stored in *Little-Endian* format, in the upper 8 bytes of FLASH memory on the CC2430 processor. The IEEE address is displayed on a sticker affixed to the bottom of each CC2430DB and CC2430EM board

This FLASH memory location would also be used for factory commissioning of IEEE addresses on devices that use Z-Stack. For CC2430-F128 devices, used on all boards in the development kit, the least significant byte of the IEEE address is located at “linear” FLASH memory address 0x1FFF8, corresponding to “banked” address 0x3FFF8.

Z-Stack treats the IEEE address area of FLASH as “write once” memory. When an attempt is made to write an IEEE address to that location (via Z-Trace, etc.), it will succeed only if the current contents are empty (0xFFFFFFFFFFFFFFFF). In other words, any 8-byte pattern other than all 0xFF values, is considered to be a valid IEEE address and won’t be modified.

5. Configuring and Using Z-Stack

5.1. Configuring Z-Stack

For the purposes of this release, the Logical Device Type and Profile are pre-configured. Details on configuring and programming sample applications for Z-Stack are provided in the sections beginning with “Building the SLC and SRC Sample Devices”.

5.2. Selecting the Logical Device Type

Z-Stack can be configured in one of three ways:

- ZigBee Coordinator – This device is configured to start the IEEE 802.15.4 network and will serve as the PAN Coordinator in that network.
- ZigBee Router – This device is configured to associate to a ZigBee Coordinator, then allow other devices to associate to it. It will route data packets in the network.
- ZigBee End Device – This device is configured to join a pre-existing network and will associate with a ZigBee Coordinator or ZigBee Router.

The above Logical Device Type selections map to the Full Function Devices (FFD) and Reduced Function Devices (RFD) in the following manner:

- An FFD can be a ZigBee Coordinator, ZigBee Router or ZigBee End Device, depending on the Logical Device Type configuration.
- An RFD must be a ZigBee End Device and cannot serve as a ZigBee Coordinator or ZigBee Router.

5.3. Selecting the Profile

Once the Logical Device Type has been selected, the device can now be configured for an application. Supported applications, in the Home Control - Lighting Profile, for this release of Z-Stack include:

- Switch Remote Control (SRC03391)
- Switch Load Controller (SLC03394)
- Dimmer Remote Control (DRC03392)
- Dimmer Load Controller (DLC03395)
- Light Sensor Monochromatic (LSM02080)
- Occupancy Sensor (OS03393)
- Generic Application (used to build custom applications)
- Serial Application (used for serial cable replacement)
- MAC-OSAL (used for testing with the MAC Z-Trace commands)
- NWK Layer (used for testing with the NWK Z-Trace commands)

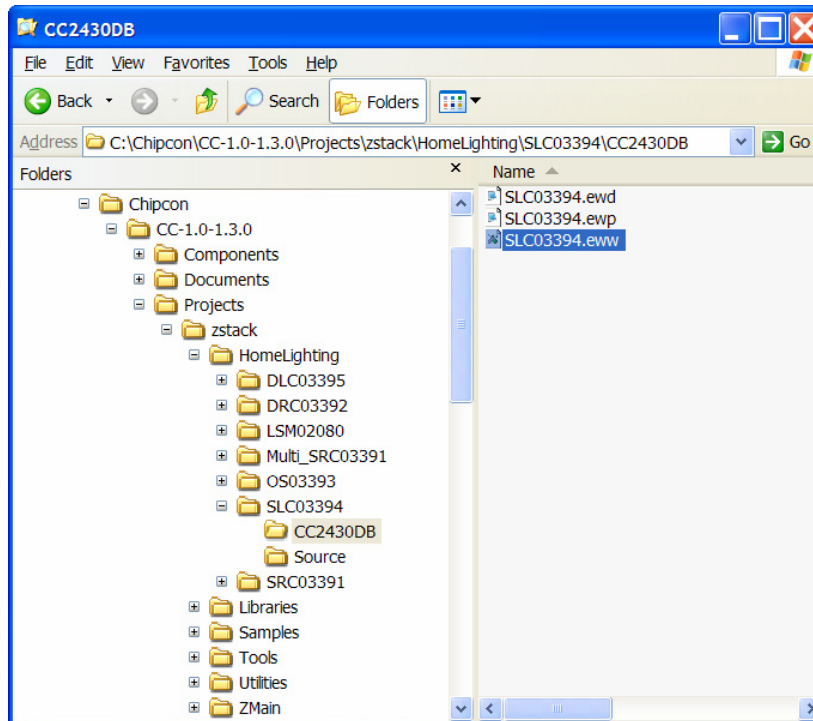
Multiple application profiles can be configured on a ZigBee device. Profiles are configured to operate ovetop *Endpoints* which provide unique addressing for the application. By convention, *Endpoint 0* is reserved for the ZigBee Device Object, which performs device specific processing and implements functions required to support the selected *Logical Device Type*.

5.4. Building the SLC and SRC Sample Devices

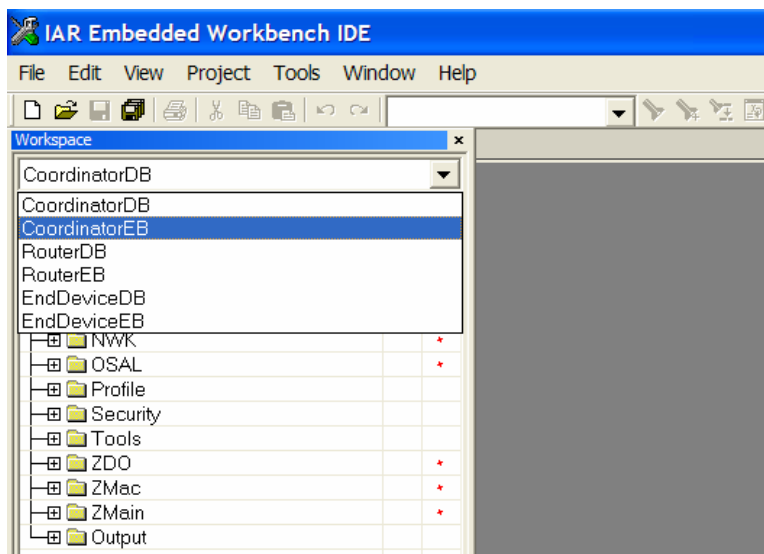
The remainder of this document describes setting up a network with 3 or more nodes, an SLC coordinator and 2 or more SRC routers. Notice that the procedures in Sections 5.5 and 5.6 are identical, only folder names and filenames are different. The other 4 Home Control-Lighting sample devices (DLC, DRC, LSM, OS), Generic Application, Serial Application, MAC-OSAL, and NWK Layer, can be built and loaded using the same general procedures.

5.5. Building an SLC Coordinator Device

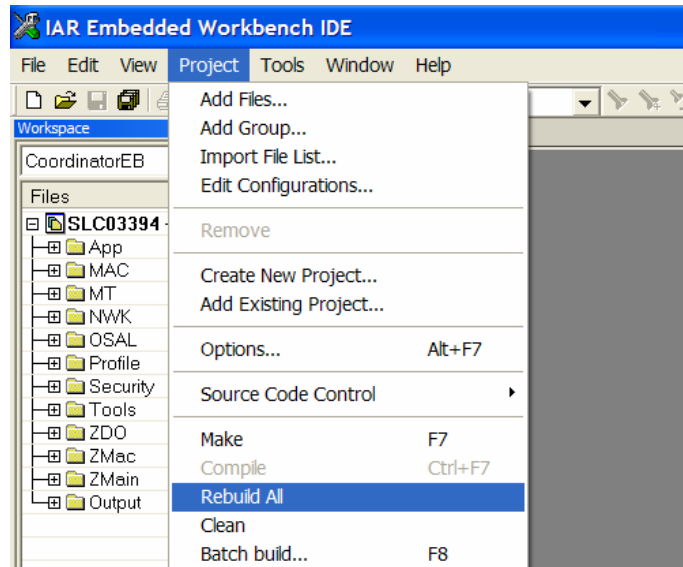
- Make sure all tools have been installed (Sections 4.1 – 4.2)
- Navigate to the SLC03394 project directory and launch the IAR Embedded Workshop by double clicking on the **SLC03394.eww** file:



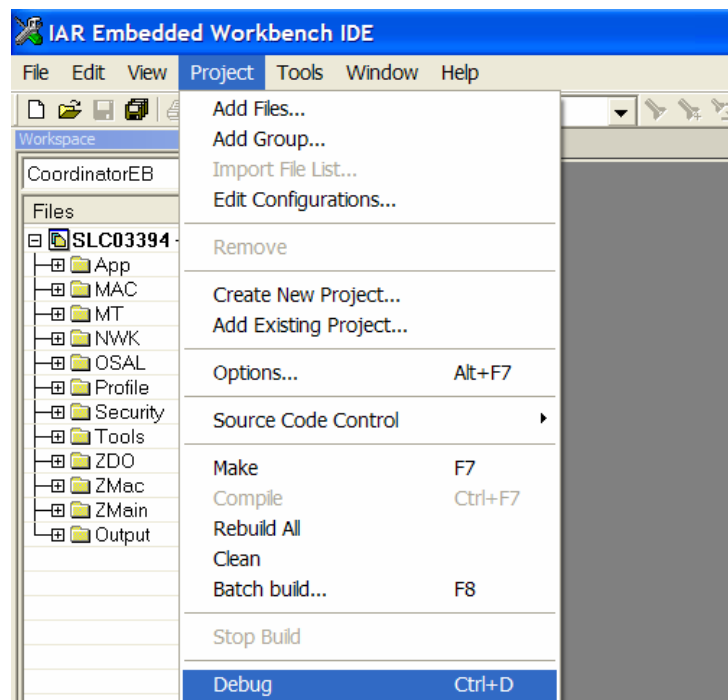
- Select the CoordinatorEB project from the **Workspace** pull-down menu:



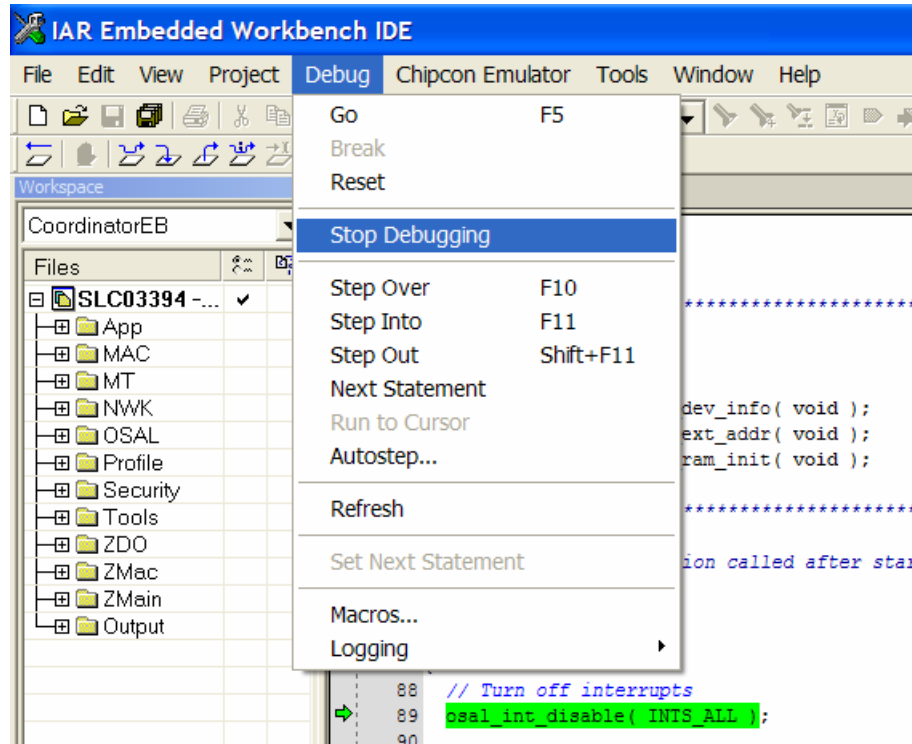
- Build the application by pulling down the **Project** menu and clicking on **Rebuild All**:



- Connect a SmartRF04EB board to the development PC with a USB cable. Apply power by moving switch S3 toward the CC2430EM board. If Windows needs to install a driver, browse to *C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\drivers\chipcon* to locate the necessary files.
- Download the application by pulling down the **Project** menu and clicking on **Debug**:



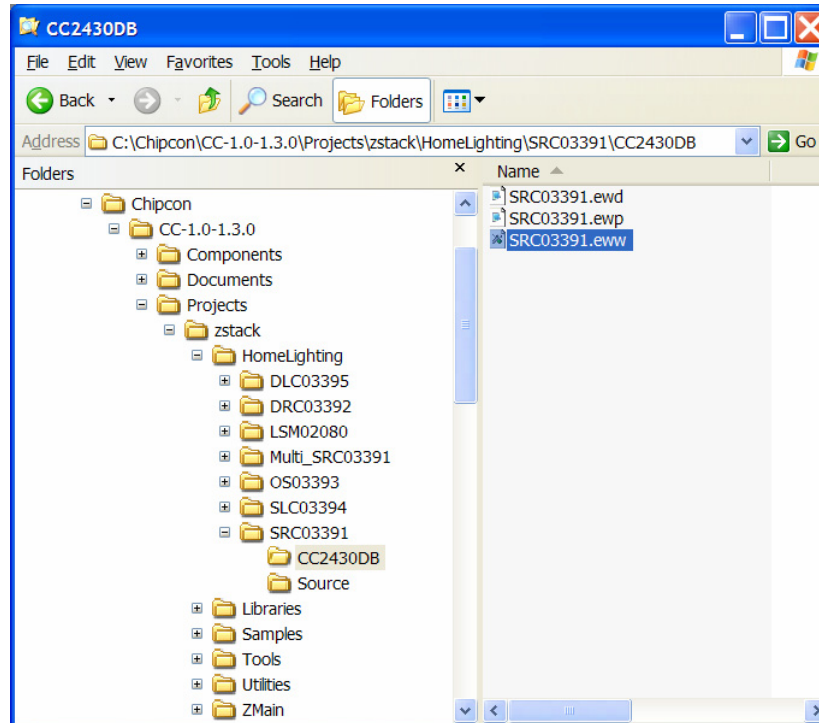
- After downloading to the SmartRF04EB is complete, exit the debugger by pulling down the **Project** menu and clicking on **Stop Debugging**. Exit the Embedded Workbench IDE.



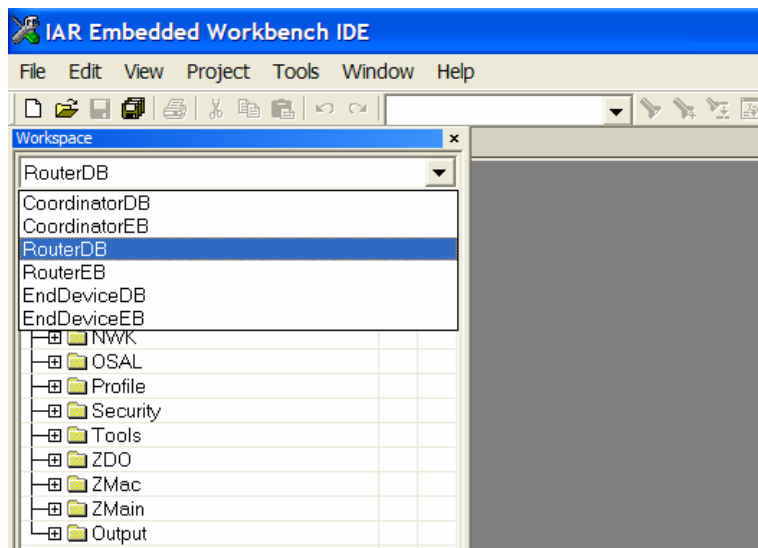
- Remove power from the CC2430EB by moving switch S3 away from the CC2430EM board. Disconnect the board from the USB cable and set it aside.

5.6. Building an SRC Router Device

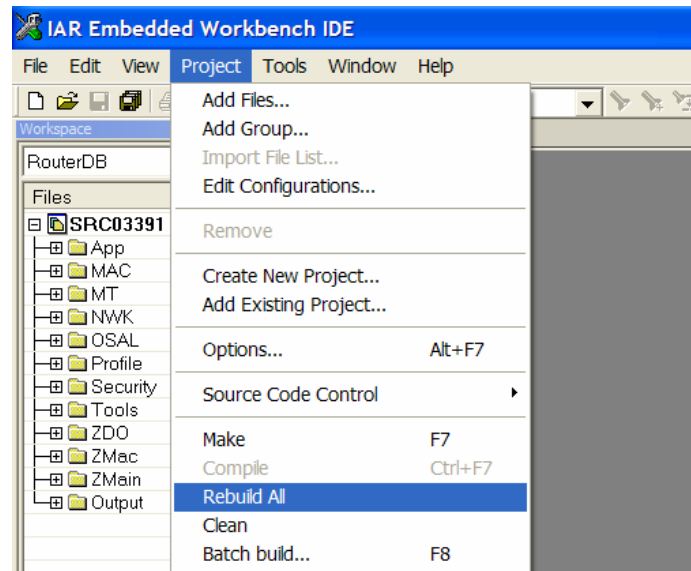
- Make sure all tools have been installed (Sections 4.1 – 4.2)
- Navigate to the SRC03391 project directory and launch the IAR Embedded Workshop by double clicking on the **SRC03391.eww** file:



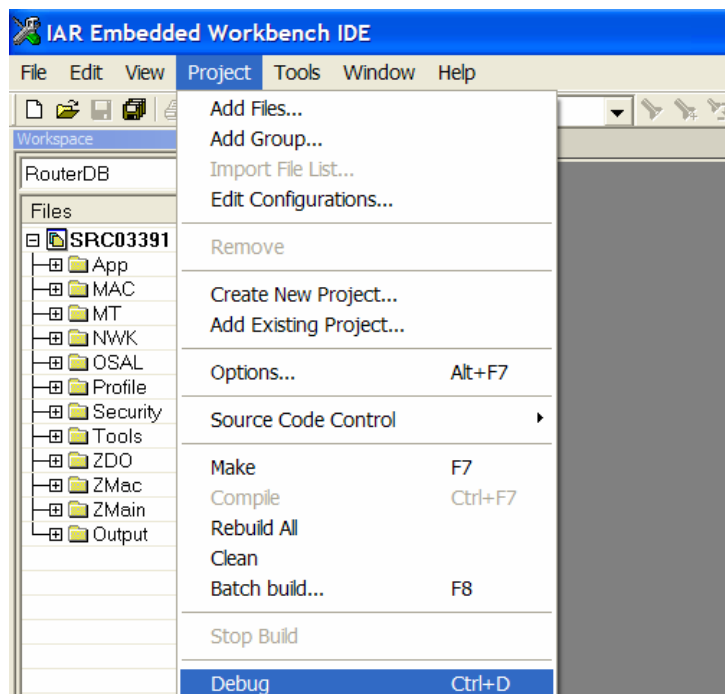
- Select the RouterDB project from the **Workspace** pull-down menu:



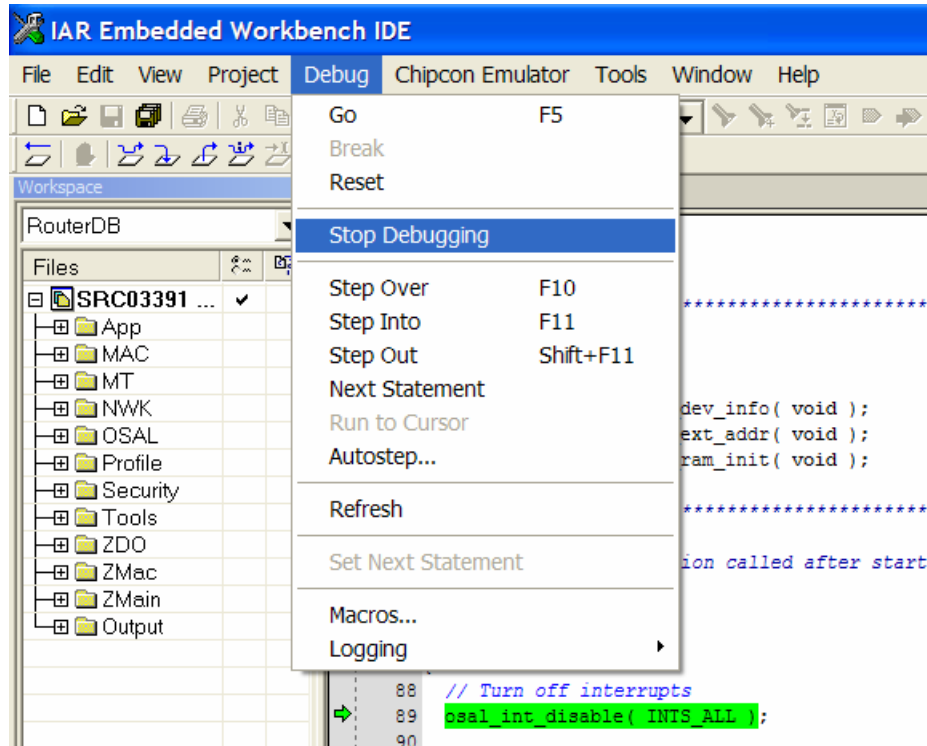
- Build the application by pulling down the **Project** menu and clicking on **Rebuild All**:



- Connect a CC2430DB board to the development PC with a USB cable. Apply power by moving switch S3 to the USB position. If Windows needs to install a driver, browse to **C:\Program Files\IAR Systems\Embedded Workbench 4.05\8051\drivers\chipcon** to locate the necessary files.
- Download the application by pulling down the **Project** menu and clicking on **Debug**:



- After downloading to the CC2430DB is complete, exit the debugger by pulling down the **Project** menu and clicking on **Stop Debugging**:



- If one or more additional SRC03391 Router devices are to be programmed, repeat the previous three steps, starting with connection of a CC2430DB to the development PC.
- When all devices have been programmed, exit the Embedded Workbench IDE.

6. Z-Stack Demonstration

6.1. Switches and LEDs

In this, and other Figure 8 Wireless documents, references are made to switches and LEDs on the evaluation boards. These devices are used to control certain Z-Stack features and display status.

Certain procedures require user input via switches, commonly referred to as SW1 through SW5. The SmartRF04EB and CC2430DB boards have a 5-position joystick, designated U400, which provides these switch inputs as shown in the table below. Pressing the joystick toward the U400 label (up position) activates the SW1 input. Switch inputs SW2 – SW4 result from pressing the joystick to the right, down (away from U400), and left positions, respectively. SW5 occurs when the joystick is pressed straight down when in the center position.

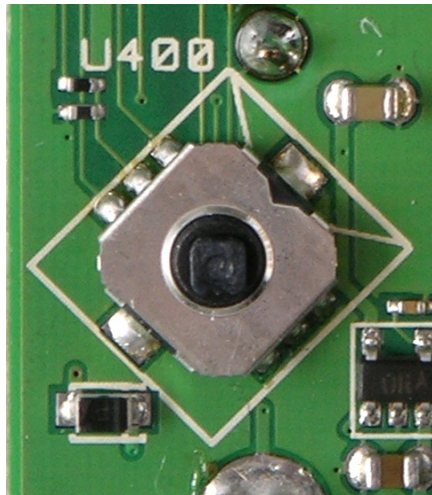


Figure 3: CC2430DB Joystick

SWITCH	JOYSTICK
SW1	<i>U400</i> position
SW2	right position
SW3	down position
SW4	left position
SW5	press down

Z-Stack sample applications display various operational status LEDs, commonly referred to as LED1 through LED4. The CC2430DB has 2 colored LEDs, designated D1 and D2, which provide these LED indications as shown in the table below:

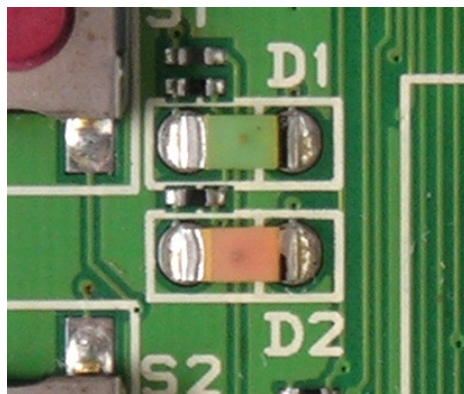


Figure 4: CC2430DB LEDs

LED	LABEL	COLOR
LED1	<i>D1</i>	Green
LED2	<i>D2</i>	Red
LED3	<i>D3</i>	Red
LED4	<i>D4</i>	Green

The SmartRF04EB has 4 colored LEDs, designated 1 - 4. The CC2430EM module does not have connections to the red LED (2) or the blue LED (4). Therefore, LEDs 1 and 3 are used to provide all LED indications from Z-Stack applications as shown in the table below:

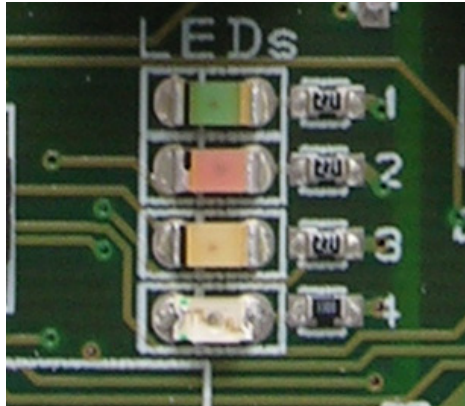


Figure 5: SmartRF04EB LEDs

LED	LABEL	COLOR
LED1	1	Green
LED2	3	Yellow
LED3	3	Yellow
LED4	1	Green

6.2. Initial Loading of 64-Bit IEEE Address

Normally, Z-Stack loads the device's 64-bit IEEE address from FLASH upon power-up or reset. When the address has been reset (0xFFFFFFFFFFFFFFFF) by erasing the FLASH, the program waits in a loop, blinking LED1 (green). This prompts the user to establish a "temporary" address by pressing SW5 (joystick center). This temporary address allows Z-Stack to start normally and then Z-Trace can be used to enter the proper 64-bit extended address, located on a sticker on the bottom surface of the circuit board.

6.3. Non-Volatile Memory

Once a device has been setup, the 64-bit address has been set, and started operation as a coordinator or has associated to another device, it will save its operating state in non-volatile (NV) memory. When the device is restarted (reset or power on), it will normally use this saved operating state to resume operation at its previous state. Inclusion of the NV restore function in a program is controlled by the *NV_RESTORE* compile flag in the project files. During program development and debugging, it is often convenient to bypass the restoration of NV parameters by disabling the *NV_RESTORE* option. If you would like to bypass the NV restore function, press and hold the SW5 button while the device is rebooting. This will load the 64-bit address but skip restoration of the operating state parameters.

6.4. Running the Sample Application (SLC and SRC)

Initially, place all the devices on the same table or work area. You will initially establish the network while the devices are all in view of each other. Later, you can experiment with various distances and different power-up sequences.

To begin execution of the programmed sample application, apply power to each programmed SmartRF04EB and CC2430DB board or press the RESET (S2) button on each board, in the following sequence:

Start with the device which will run the **Switch Load Controller** (coordinator) application. If this board's IEEE address has not yet been set-up, follow the steps in Section 6.2. Once this device has been initialized and begun operation, LED3 will be turned on. If the LED3 is not turned on, after a period of time (based on the number of channels selected), it could mean that it didn't find a channel with a suitable energy level (see section 9.1) and could not start a network.

Next, power up or RESET the first **Switch Remote Control** (router) device. If this board has not had its unique extended address set-up, follow the steps in Section 6.2. Once this device has begun operation and successfully joined (associated), the LED3 will be turned on.

Finally, power up or RESET the remaining **Switch Remote Control** (router) devices. Once again, if these boards have not had unique 64-bit extended addresses set-up, follow the steps in Section 6.2. Once these devices have begun operation and successfully joined the network, their LED3s will be turned on.

The discussion above assumes each device has been programmed and disconnected from the development PC. When necessary, a target device can be controlled from the IAR IDE, providing for standard debugging features such as breakpoints, single-stepping, viewing of memory and register contents, etc.

6.4.1. Sample Use of Device Binding

After association, the devices need to be either bound together or they need a destination address to be established. Perform an *End Device Bind* with the following steps:

- On the first **Switch Remote Control (SRC)** device, press the joystick to the right (toward center of the board) and release to send an *End Device Bind Request*. This will initiate an “*End Device Bind Period*” of about 5 seconds.
- Before the “*End Device Bind Period*” expires, on the **Switch Load Controller** device, press the joystick to the right and release. The SRC device should turn on its LED1 to indicate success. LED1 flashes if the binding process is not completed before the “*End Device Bind Period*” expires.
- Repeat the previous two steps with the remaining **Switch Remote Control** device(s) and the **Switch Load Controller** device.

6.4.2. Sample Use of Service Discovery

After association, the devices need to be either bound or they need a destination address to be established. As an alternative to binding, an SRC device can initiate an *Auto Match* function.

This function uses the Zigbee Match Description Request to get the address of a device that accepts the output messages of the SRC device. For an example, take a look at the function `ZDApp_AutoFindDestination()` in `ZDApp.c`.

- On the first **Switch Remote Control (SRC)** device, press the joystick to the left (away from the center of the board) to initiate a *Service Discovery Match*. The SRC device should turn on its LED1 to indicate success.
- Repeat the previous step with the other **Switch Remote Control** device(s).

6.4.3. Sending Application Data

After the SRC devices have successfully associated, and have either been bound or auto matched to the SLC, pressing the joystick to the UP position on an SRC device will turn LED1 on the SLC device ON or OFF. This demonstrates a simple light-switch application. Note that the SRC devices send ON or OFF commands, not TOGGLE commands, to the SLC device. This allows the possibility that multiple SRC devices can get “out of phase” with each other, requiring two joystick presses on an SRC to toggle the SLC device’s LED1.

7. Developing Applications with Z-Stack

To build a better system understanding of Zigbee, please read the Zigbee Implementer’s Guide:

- [ZigBee Implementers Guide_F8W-2004-0007_.pdf](#)

To understand the available configuration and startup options, please read the Zigbee Device Object’s Programmer’s Guide:

- [Z-Stack Device Object Programmer's Guide_F8W-2004-0008_.pdf](#)

To develop or change the existing sample applications, read the following programmer’s guides in the ZStack documentation section:

- [Z-Stack SLC03394 Programmer's Guide_F8W-2003-0028_.pdf](#)
- [Z-Stack SRC03391 Programmer's Guide_F8W-2003-0027_.pdf](#)

8. Non-Beacon vs. Beacons

By default (as installed), each project is setup to be a non-beaconing network. This means that coordinators don’t transmit a periodic beacon and that end devices have to poll their parent to check for waiting data. When beacons are enabled, the coordinators will transmit periodic beacons. Instead of polling their parent, end devices will look for the coordinator’s beacon, which will contain an indication when data is waiting for the end device.

To enable beacons and establish the beaconing rate, change the `DEFAULT_BEACON_ORDER` and `DEFAULT_SUPERFRAME_ORDER` definitions (in `ZDApp.c`) to the desired beacon rate. The available beacon rates are defined in `NLMEDE.h`. You must also change the stack profile (`STACK_PROFILE_ID` in `nwk_globals.c`) to either `GENERIC_STAR` or `GENERIC_TREE` (Tree mode is NOT recommended). The beacons will not work with `HOME_CONTROLS` or `BUILDING_AUTOMATION`. You only need to change and download the Coordinator, since the routers and end devices will adapt to their coordinator.

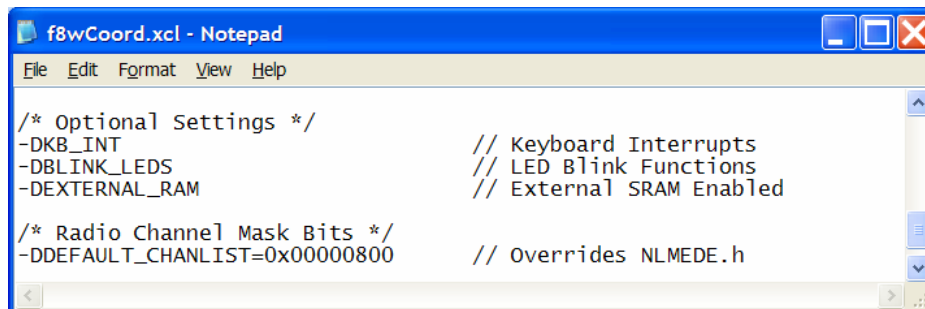
9. Channel Selection

The 802.15.4 specification defines 16 channels in the 2.4 GHz frequency range. These channels are assigned numbers 11 through 26. Z-Stack initially defaults to channel 11, but the user can select a different channel by changing *DEFAULT_CHANLIST*. This parameter is a bit map field, with each bit representing a single channel. As shown below, the initial default channel 11 (0xB) is represented by 0x00000800 (11th bit in the field, starting from bit 0).

Channel Number	Bit Map Field
11	0x00000800
12	0x00001000
13	0x00002000
14	0x00004000
15	0x00008000
16	0x00010000
17	0x00020000
18	0x00040000
19	0x00080000
20	0x00100000
21	0x00200000
22	0x00400000
23	0x00800000
24	0x01000000
25	0x02000000
26	0x04000000

Table 1: Default Channel Select Bit Map

DEFAULT_CHANLIST is defined in the NLMEDE.h file. In addition, it can also be specified as a compile option in a project's linker command file – this overrides the value in the NLMEDE.h file. The linker command files are found in the ...\\Projects\\Tools\\CC2430DB folder. As shown below, the last line in the *f8wCoord.xcl* file specifies the channel(s) that will be used when the Coordinator device starts up. This is the recommended location for developers to establish specific channel settings for their projects. Similar settings are also provided in the *f8wRouter.xcl* and *f8wEndev.xcl* files. This feature allows developers set up a “personal” channel to avoid conflict with others. Multiple channels can be specified by including the appropriate bits.



```

f8wCoord.xcl - Notepad
File Edit Format View Help

/* Optional Settings */
-DKB_INT // Keyboard Interrupts
-DBLINK_LEDS // LED Blink Functions
-DEXTTERNAL_RAM // External SRAM Enabled

/* Radio Channel Mask Bits */
-DDEFAULT_CHANLIST=0x00000800 // Overrides NLMEDE.h
  
```

9.1. Energy Level

The coordinator will start a network on a selected channel only if the energy level on that channel is below a threshold value. The threshold value is set to -45dBm and can be modified by changing the MAX_SCAN_ENERGY definition in the ZMac_CB_CC2430.c file. The value of this parameter minus 83 gives the maximum tolerated energy level in dBm. To ensure that the coordinator will always find a suitable channel to start the network on, it is recommended that more than one channel is selected.

Applicable Documents

Internal Documents

1. Serial Port Interface, Figure 8 Wireless Document F8W-2003-0001
2. OS Abstraction Layer API, F8W Document F8W-2003-0002
3. Z-Stack NWK API, F8W Document F8W-2003-0008
4. Z-Stack APS API, F8W Document F8W-2003-0025
5. Z-Stack AF API, F8W Document F8W-2003-0010

External Documents

6. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Standard 802.15.4, 05/12/2003.