



Z-Stack User's Guide For CC2430ZDK

ZigBee 2006 Release
Version 1.4.3

Document Number: F8W-2005-0036

Texas Instruments, Inc.
San Diego, California USA
(619) 497-3845

Revision	Description	Date
1.0	Initial release.	12/22/2005
1.1	Changed logo on title page, changed copyright on page footer.	02/27/2006
1.2	Added Figure 3 showing SOC_BB Battery Board.	06/25/2006
1.3	Changed sample applications to SampleSwitch and SampleLight.	12/13/2006
1.4	Changed sample applications to SampleApp.	04/10/2007
1.5	Removed reference to Chipcon web site from Section 4.2.	05/31/2007
1.6	Changed IAR compiler version to 7.30B in Section 4.2.	12/21/2007

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.....	3
3. INSTALLATION REQUIREMENTS.....	3
3.1. HOST COMPUTER REQUIREMENTS	3
3.2. TARGET DEVELOPMENT SYSTEM REQUIREMENTS	3
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.....	4
5.1. CONFIGURING Z-STACK.....	4
5.2. LOGICAL DEVICE TYPES	4
5.3. BUILDING SAMPLEAPP DEVICES.....	4
5.4. BUILDING A SAMPLELIGHT COORDINATOR DEVICE	4
6. Z-STACK DEMONSTRATION.....	7
6.1. SWITCHES AND LEDs	7
6.2. INITIAL LOADING OF 64-BIT IEEE ADDRESS	9
6.3. RUNNING THE SAMPLE APPLICATION.....	9
7. PANID AND CHANNEL SELECTION.....	10
7.1. ENERGY LEVEL.....	11
APPLICABLE DOCUMENTS.....	12

Table of Figures

FIGURE 1: CHIPCON SMARTRF04EB EVALUATION BOARD WITH CC2430EM	1
FIGURE 2: CHIPCON CC2430DB DEVELOPMENT BOARD.....	2
FIGURE 3: CHIPCON SOC_BB BATTERY BOARD	2
FIGURE 4: CC2430DB JOYSTICK	8
FIGURE 5: CC2430DB LEDs.....	8
FIGURE 6: SMARTRF04EB LEDs	8
FIGURE 7: SMARTRF04EB COORDINATOR JUMPER	9

Table of Tables

TABLE 1: DEFAULT CHANNEL SELECT BIT MAP	10
---	----

1. Introduction

1.1. Scope

This document accompanies the Texas Instruments Z-Stack™ solution for use with the CC2430 ZigBee Development Kit. 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, may be used to demonstrate or develop ZigBee applications based on the Z-Stack software package:



Figure 1: Chipcon SmartRF04EB Evaluation Board with CC2430EM

Five smaller Chipcon CC2430DB evaluation boards are also contained in CC2430ZDK and CC2431ZDK ZigBee development kits:

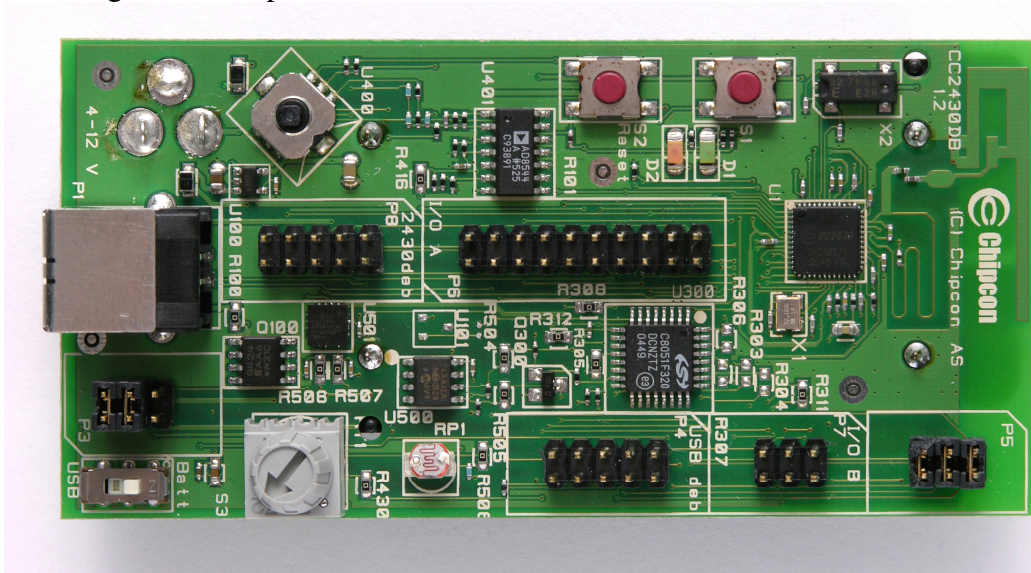


Figure 2: Chipcon CC2430DB Development Board

Ten compact Chipcon SOC_BB evaluation boards, each to be fitted with a CC2430EM or a CC2431EM evaluation module, are contained in CC2431ZDK ZigBee development kits:

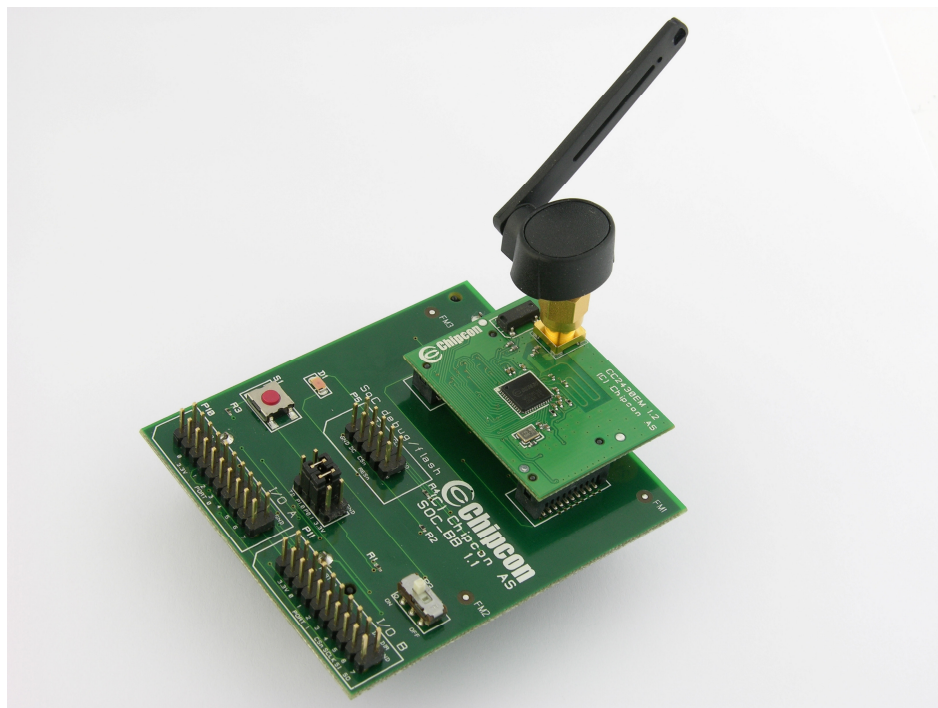


Figure 3: Chipcon SOC_BB Battery Board

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 port on SmartRF04EB boards (9-pin connector) and the host PC to utilize Z-Tool™ and other programs included with the Z-Stack package.

3. Installation Requirements

3.1. Host Computer Requirements

Z-Stack and Z-Tool 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-Tool:

- .NET 1.1 Framework
- Windows XP Service Pack 1 (if using Windows XP)
- 1 serial port for Z-Tool 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 CC243x 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 Texas Instruments Z-Stack files and programs from the downloaded package. Run the windows-based installation program, **ZStack-CC2430-1.4.3.exe**, which will create the required directory structure and load all software and documentation files. Review the README.txt 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 were built and tested with EW8051 version 7.30B. 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.

4.3. Device IEEE Addresses

Each CC2430DB, CC2430EM, and CC2431EM 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, located in the upper 8 bytes of FLASH memory on the CC2430 or CC2431 processor. The IEEE address is displayed on a sticker affixed to the bottom of each CC2430DB, CC2430EM, or CC2431EM 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-Tool, 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 ZigBee 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 SampleApp Devices”.

5.2. Logical Device Types

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

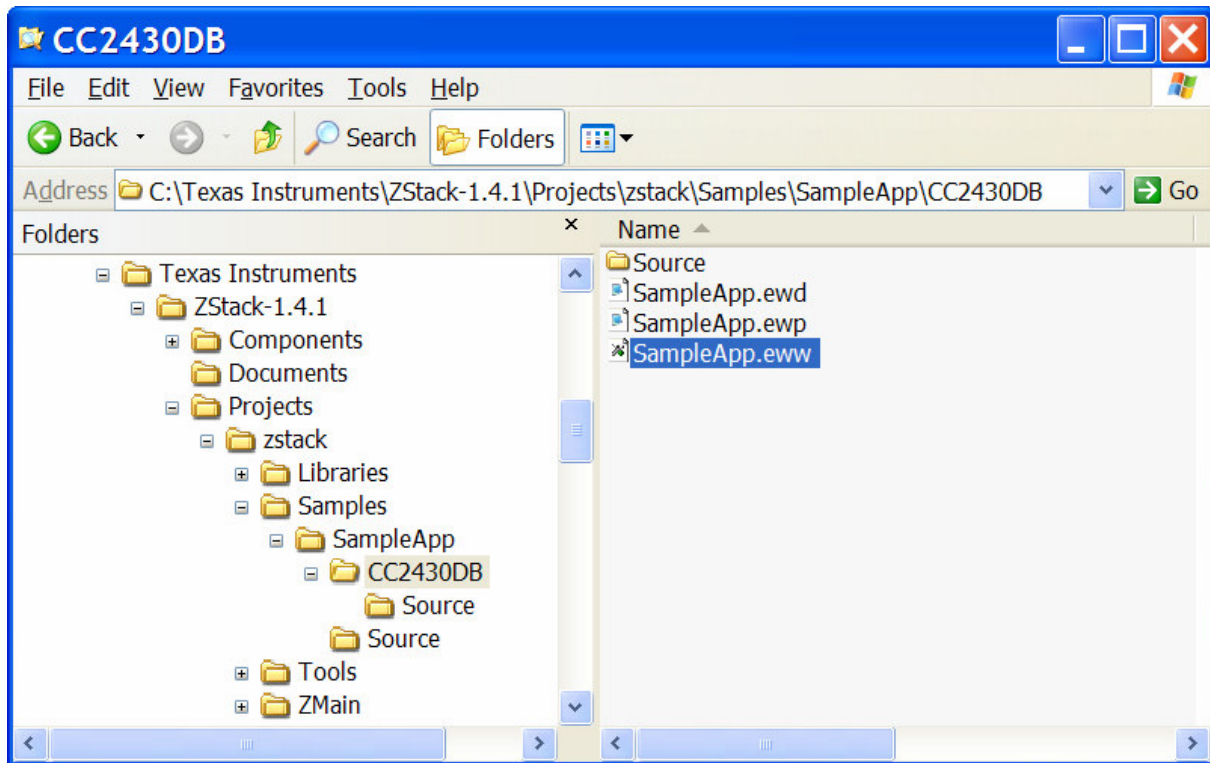
- ZigBee Coordinator – This device is configured to start the IEEE 802.15.4 network and will serve as the PAN Coordinator in that network.
- ZigBee Router – This device is configured to join an existing network, associate to a Coordinator or Router, and then allow other devices to associate to it. It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

5.3. Building SampleApp Devices

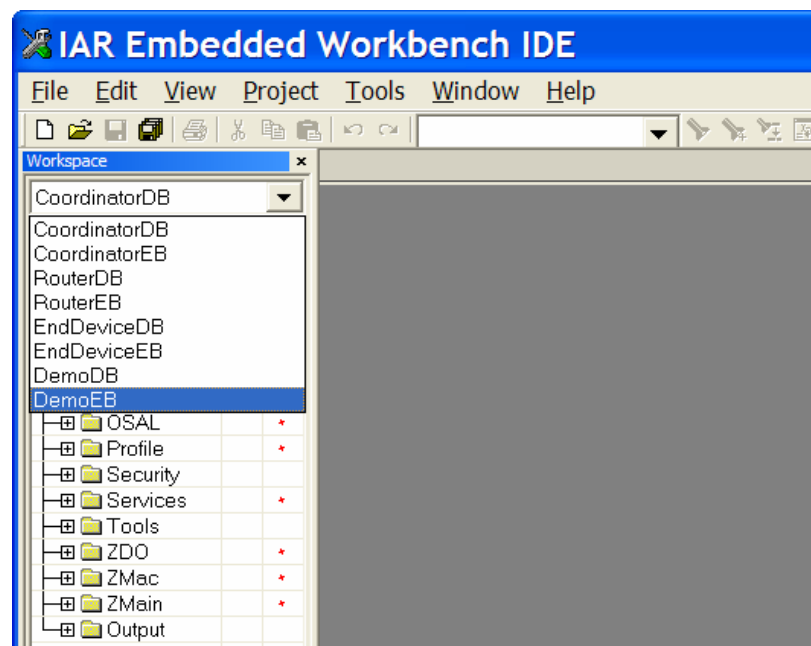
The remainder of this section describes programming SmartRF04EB+CC2430EM boards to set up a simple ZigBee network with 2 or more nodes, a SampleApp Coordinator and one or more SampleApp Routers. The SampleApp project file provides configurations to uniquely build either of these devices, as well as, to build a generalized “Demo” device which permits selection of Coordinator and Router operation by setting one jumper on the SmartRF04EB board. The following examples provide details on building and running the “Demo” sample application on two or more devices.

5.4. Building a SampleLight Coordinator Device

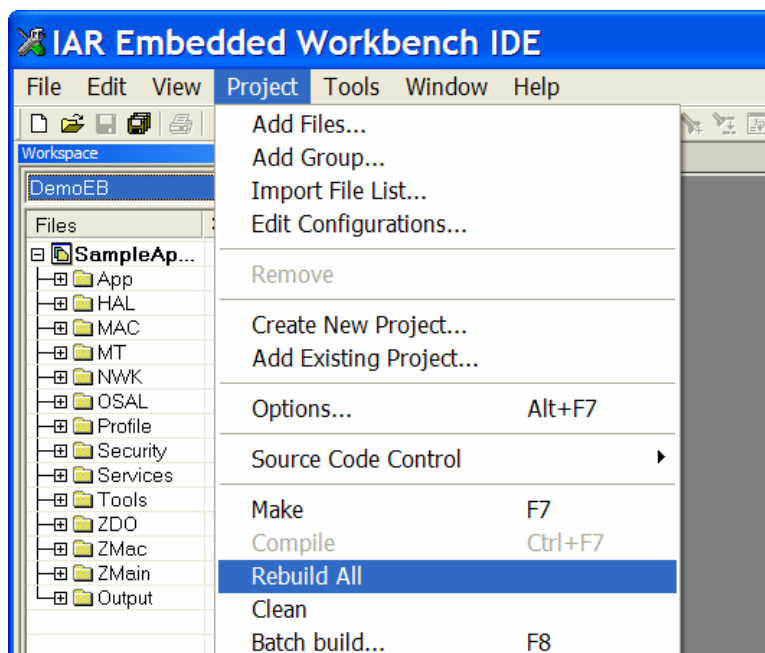
- Make sure all tools have been installed (Sections 4.1 – 4.2)
- Connect a SmartRF04EB board to the development PC with a USB cable. Apply power by moving switch S3 toward the CC2430EM board. If Windows needs to install a driver, browse to *C:\Program Files\IAR Systems\Embedded Workbench 4.0\8051\drivers\Chipcon* to locate the necessary files.
- Navigate to the SampleApp project directory and launch the IAR Embedded Workshop by double clicking on the *SampleApp.eww* file:



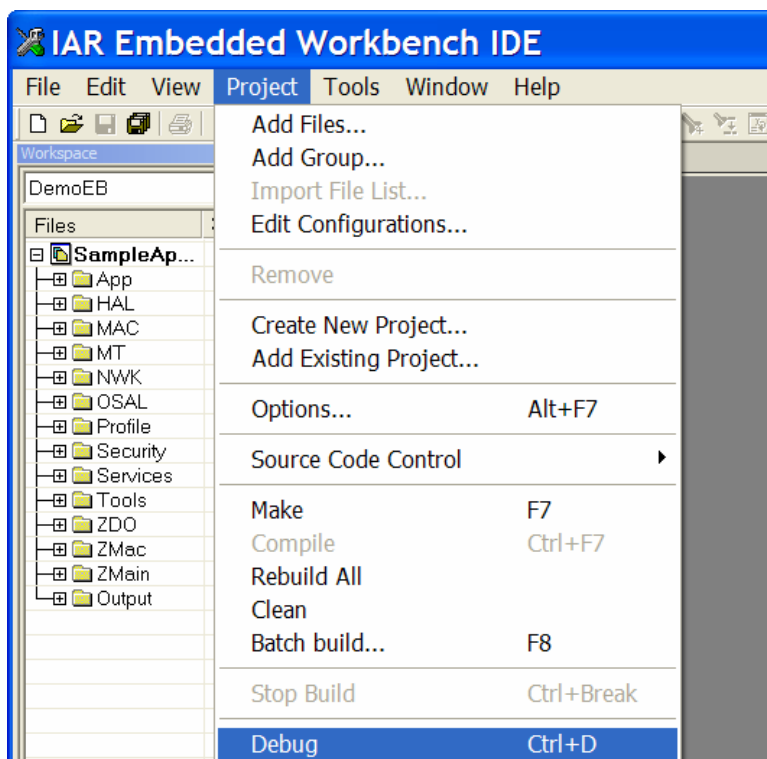
- Select the DemoEB project from the *Workspace* pull-down menu:



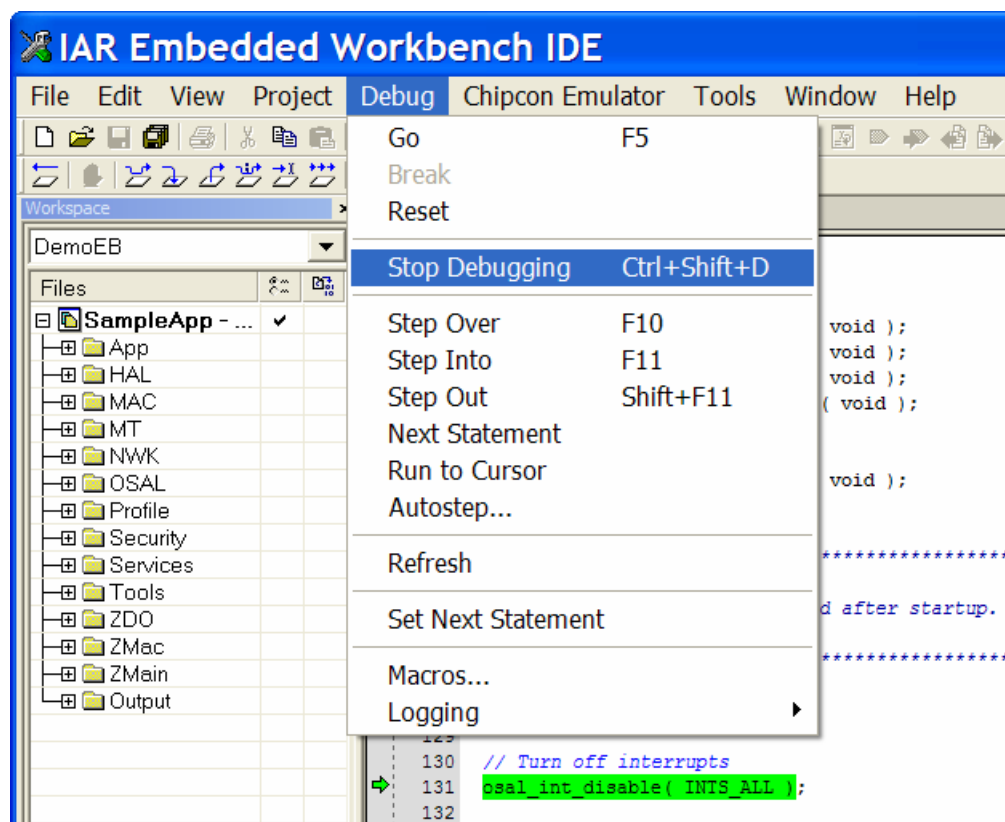
- Build the application by pulling down the *Project* menu and clicking on **Rebuild All**:



- Download the application by pulling down the **Project** menu and clicking on **Debug**:



- After downloading to the CC2430EM is complete, exit the debugger by pulling down the **Debug** menu and clicking on **Stop Debugging**:



- Remove power from the SmartRF04EB by moving switch S3 away from the CC2430EM board. Disconnect the SmartRF04EB from the USB cable and set it aside.
- Repeat the previous steps to program more CC2430EM boards. At least two boards must be programmed to run the “Demo” sample application.
- When all devices have been programmed, exit the Embedded Workbench IDE.

6. Z-Stack Demonstration

6.1. Switches and LEDs

In this, and other Z-Stack sample application documents, references are made to switches and LEDs located on 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.

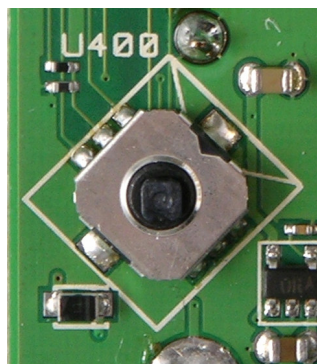


Figure 4: CC2430DB Joystick

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 provides 2 colored LEDs, designated D1 and D2, which provide the four Z-Stack LED indications as shown in the table below:

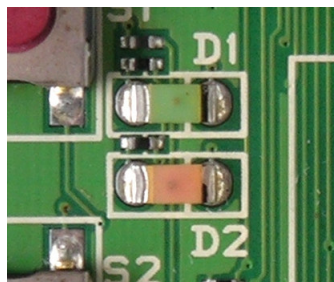


Figure 5: CC2430DB LEDs

LED	LABEL	COLOR
LED1	<i>D1</i>	Green
LED2	<i>D2</i>	Red
LED3	<i>D2</i>	Red
LED4	<i>D1</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 four LED indications from Z-Stack applications as shown in the table below:

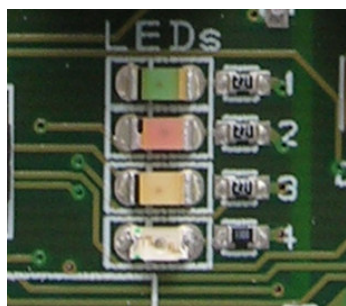


Figure 6: SmartRF04EB LEDs

LED	LABEL	COLOR
LED1	<i>1</i>	Green
LED2	<i>3</i>	Yellow
LED3	<i>3</i>	Yellow
LED4	<i>1</i>	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 during startup, blinking LED1 (green). On SmartRF04EB boards, the LCD shows the ► character in all display positions. This prompts the user to establish a “temporary” address by pressing SW5 (joystick center). On SmartRF04EB boards, the LCD now shows the 64-bit address that was created. This temporary address allows Z-Stack to start normally, so that the developer can later use Z-Tool to restore the proper 64-bit extended address, located on a sticker on the bottom surface of the circuit board.

6.3. Running the Sample Application

Initially, place all the devices on the same table or work area. You will 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.

After each of the CC2430EM boards has been programmed with the “Demo” configuration of the Z-Stack SampleApp, one of the boards needs to be designated as a ZigBee Coordinator. This is done by placing a jumper across pins *P10-9* and *P10-11* on the SmartRF04EB, as shown in Figure 7 below. Make sure that only one SmartRF04EB board has a jumper in this position.



Figure 7: SmartRF04EB Coordinator Jumper

Initially, begin execution of the programmed SampleApp by applying power to the device that is configured as the ZigBee Coordinator. If LED1 (green) flashes, see Section 6.2 to determine whether you need to create a temporary IEEE address for the board. The device now performs a scan of the programmed ZigBee channel (see Section 7), temporarily flashing LED1. Once the device successfully starts up a network, LED3 (yellow) will turn on and LED1 will stop flashing.

Next, power up a ZigBee Router device (no jumper on *P10*). If LED1 (green) flashes, see Section 6.2 to determine whether you need to create a temporary IEEE address for the board. This device now scans the programmed ZigBee channel for a network, temporarily flashing LED1. Once it joins the network started by the Coordinator, LED3 (yellow) will be turned on and LED1 will stop flashing. If desired, turn on more Router devices and each of them will turn on their LED3 after joining the network.

Once the network has been formed, the SampleApp will provide a very simple demonstration of ZigBee wireless communication. The sample application performs the following three functions:

- Periodic (about 5 seconds) broadcast of a message to all network devices
- When button SW1 is pressed, broadcast of a message to devices subscribed to Group 1
- When button SW2 is pressed, toggles a devices' membership in Group 1

When each SampleApp device starts up, it is subscribed to Group 1 and will receive and process messages sent to Group 1 from any other device. In this demonstration, a device will flash its LED1 (green) when a Group 1 message is received. So, when the network is initially started up, pressing button SW1 on any device will broadcast a message, causing all of the other devices to flash their LED1. Pressing button SW2 on a device toggles that device's membership in Group 1, allowing the user to enable/disable LED1 flashing on that device.

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.

7. PanID and Channel Selection

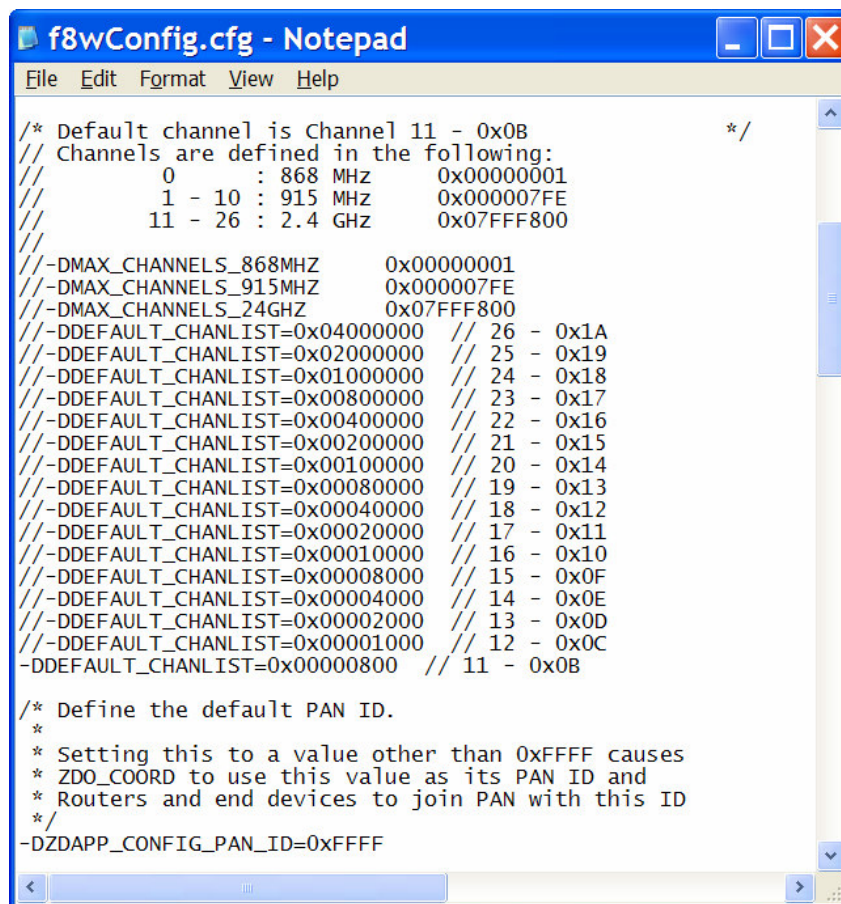
The ZigBee specification defines the use of a 14-bit Personal Area Network Identifier (PanID) to uniquely identify a network. Z-Stack provides the user with two methods of selecting a PanID when starting or joining a network by setting the value of `ZDAPP_CONFIG_PAN_ID`. Setting this parameter to `0xFFFF` causes a device to join the “best” network it can discover, any other value causes it to use the exact value specified.

The IEEE 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 and *ZDAPP_CONFIG_PAN_ID* may be defined as a compile options the IAR IDE, as well as, in a project's configuration command file. Configuration command files are located in the applicable ...\\Projects\\Tools\\CC2430DB folder. As shown below, lines in the *f8wConfig.cfg* file specify the channel(s) and PanID that will be used when the Z-Stack devices start up. This is the recommended location for developers to establish specific settings for their projects. This feature allows developers set up a "personal" channel and PanID to avoid conflict with others. Multiple channels can be specified by including the appropriate bits in the *DEFAULT_CHANLIST* definition.



```

/* Default channel is Channel 11 - 0x0B */
// Channels are defined in the following:
//      0      : 868 MHz      0x00000001
//      1 - 10 : 915 MHz      0x000007FE
//      11 - 26 : 2.4 GHz      0x07FFF800
//
//--DMAX_CHANNELS_868MHZ      0x00000001
//--DMAX_CHANNELS_915MHZ      0x000007FE
//--DMAX_CHANNELS_24GHZ      0x07FFF800
//--DDEFAULT_CHANLIST=0x04000000 // 26 - 0x1A
//--DDEFAULT_CHANLIST=0x02000000 // 25 - 0x19
//--DDEFAULT_CHANLIST=0x01000000 // 24 - 0x18
//--DDEFAULT_CHANLIST=0x00800000 // 23 - 0x17
//--DDEFAULT_CHANLIST=0x00400000 // 22 - 0x16
//--DDEFAULT_CHANLIST=0x00200000 // 21 - 0x15
//--DDEFAULT_CHANLIST=0x00100000 // 20 - 0x14
//--DDEFAULT_CHANLIST=0x00080000 // 19 - 0x13
//--DDEFAULT_CHANLIST=0x00040000 // 18 - 0x12
//--DDEFAULT_CHANLIST=0x00020000 // 17 - 0x11
//--DDEFAULT_CHANLIST=0x00010000 // 16 - 0x10
//--DDEFAULT_CHANLIST=0x00008000 // 15 - 0x0F
//--DDEFAULT_CHANLIST=0x00004000 // 14 - 0x0E
//--DDEFAULT_CHANLIST=0x00002000 // 13 - 0x0D
//--DDEFAULT_CHANLIST=0x00001000 // 12 - 0x0C
-DDEFAULT_CHANLIST=0x00000800 // 11 - 0x0B

/* Define the default PAN ID.
 *
 * Setting this to a value other than 0xFFFF causes
 * ZDO_COORD to use this value as its PAN ID and
 * Routers and end devices to join PAN with this ID
 */
-DZDAPP_CONFIG_PAN_ID=0xFFFF

```

7.1. 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 *mac_scan.c* file (available only with TIMAC source file distribution). 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, F8W Document F8W-2003-0001
2. OSAL API, F8W Document F8W-2003-0002
3. Z-Stack API, F8W Document F8W-2006-0021

External Documents

4. 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.