Z-Stack
User's Guide For
Chipcon CC2420DB

ZigBee 2006 Release
Version 1.4.2

Document Number: F8W-2005-0019

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

Version	Description	Date
1.0	Initial release.	09/25/2005
1.1	Changed logo on title page, changed copyright on page footer.	03/21/2006
1.2	Changed sample applications to SampleSwitch and SampleLight.	12/13/2006
1.3	Changed sample application to SampleApp.	04/10/2007
1.4	Modified description of compiler version in Section 4.3.	05/17/2007

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1. Scope

This document accompanies the Texas Instruments Z-Stack™ solution for use with CC2420DB evaluation boards. Z-Stack is a complete protocol stack and solution targeting 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 or more Chipcon CC2420DB evaluation boards, using CC2420s (IEEE 802.15.4 radios) are contained in the development kit accompanying Z-Stack. These boards may be used to demonstrate or develop ZigBee applications based on the Z-Stack software package.

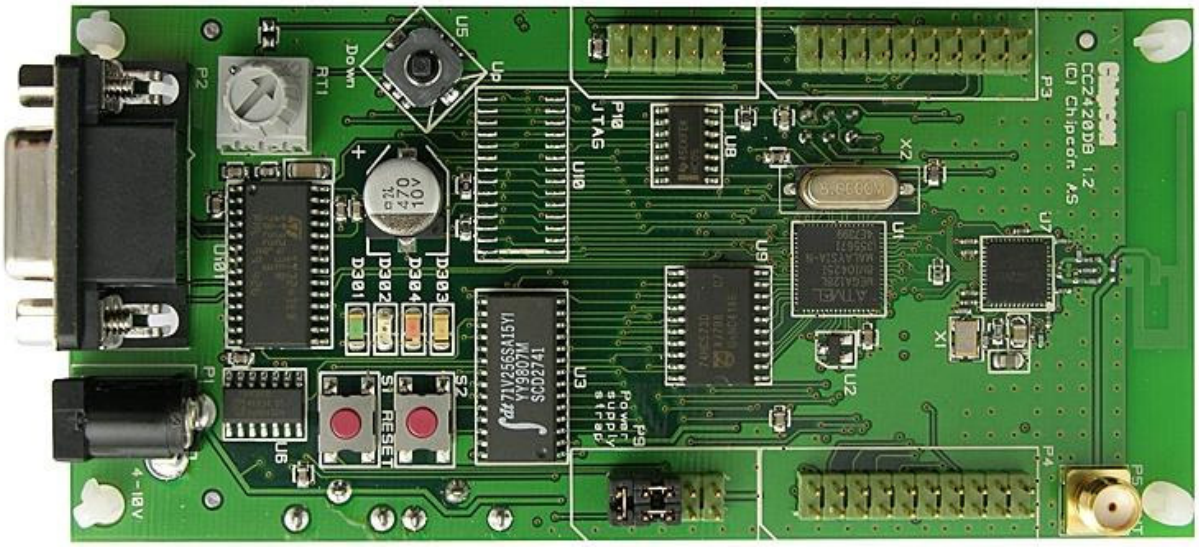


Figure 1: Chipcon CC2420DB Development Board

2.3. Cables

All necessary cabling has been included with the development kit. To support download and debugging of the development board, a USB cable should be connected from an AVR JTAGICE mkII unit (Atmel Corporation) to the PC. RS232 cables may be connected between the serial port on CC2420DB 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 CC2420DB board
- 1 USB or serial port for JTAGICE mkII communication to the CC2420DB board

3.2. Target Development System Requirements

Z-Stack provides a complementary offering to the IAR Embedded Workbench (EWAVR) suite of software development tools. These tools support project management, compiling, assembling, linking, downloading, and debugging for various Atmel AVR processors. Additional processor-level support is provided by the AVR Studio 4 program from Atmel. The following are required support for the Z-Stack target development system:

- IAR EWAVR (http://www.iar.com/)
- AVR Studio 4 (http://www.atmel.com/)
- AVR JTAGICE mkII (http://www.atmel.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 installation program, ***ZStack-AVR-1.4.2.exe***, which will create the required directory structure and 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 AVR Studio 4

Download a copy of the AVR Studio 4 installation program from the Atmel Corporation website: http://www.atmel.com/products/avr/. Locate the package used for this release by following the links: Tools & Software→AVR Studio 4→AVR Studio 4.11 (build 401). Note that this software evolves, so it's a good idea to periodically check for updates or newer versions. Refer to Appendix A for initially setting up to use the Atmel JTAGICE mkII device under the control of the AVR Studio 4 program.

4.3. Install IAR EWAVR

Install the Embedded Workbench for Atmel AVR from IAR Systems: http://www.iar.com/. The project and library files included in this release of Z-Stack were built and tested with EWAVR version 4.21A. When considering an upgrade to a newer version of EWAVR, it is necessary to verify that installed project and library files are compatible with the newer development tools. Refer to Appendix B for initially setting up to use the Atmel JTAGICE mkII device under the control of the EWAVR downloader and C-Spy debugger.

4.4. Save EEPROM Contents for Each CC2420DB Board

Each of the CC2420DB boards 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 lowest 8 bytes of EEPROM on the AVR processor. This places the least significant byte (LSB) at location 0x000 and the most significant byte (MSB) at location 0x007. The IEEE address is displayed on a sticker affixed to the bottom of each CC2420DB.

Before any programming or debugging is performed on CC2420DB boards, the contents of their EEPROM memories should be preserved on disk on the development computer. This allows the IEEE address (and other EEPROM contents) to be restored if the EEPROM gets erased or corrupted. **Important:** follow the instructions in Appendix D to save a copy of the EEPROM contents for each of the CC2420DB boards.

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 the Z-Stack sample application are provided in the sections beginning with “Building SampleApp Devices”.

5.2. Logical Device Types

Z-Stack devices 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.

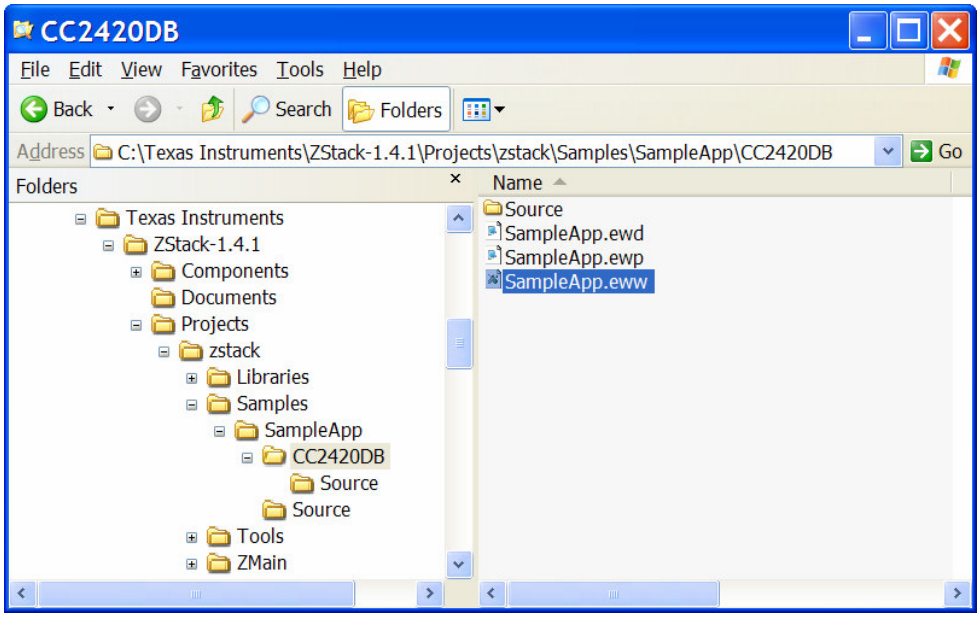
5.3. Building SampleApp Devices

The remainder of this section describes programming CC2420DB 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, a well as, to build a generalized “Demo” device which permits selection of Coordinator and Router operation by setting one jumper on the CC2420DB board. The following examples provide details on building and running the “Demo” sample application on two or more devices.

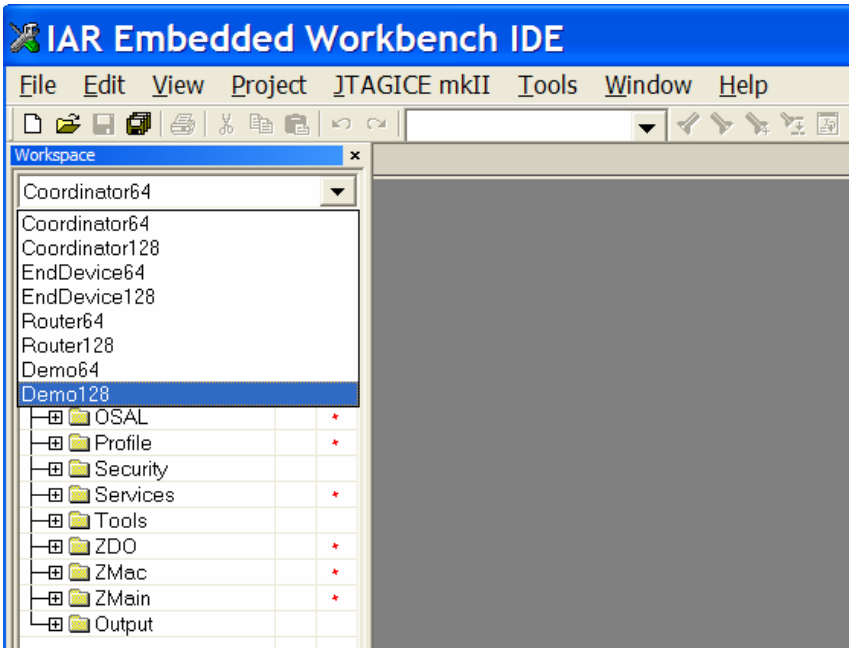
5.4. Building a SampleApp “Demo” Device

- Make sure all tools have been installed (Sections 4.1 – 4.3)
- With power removed from the CC2420DB board, connect the JTAGICE mkII device to the 10-pin JTAG connector on the CC2420DB. Make sure the JTAG cable is oriented properly (pin 1 “points to” the RS232 connector) and apply power to the CC2420DB. LEDs should now be lit on top of the JTAGICE unit.

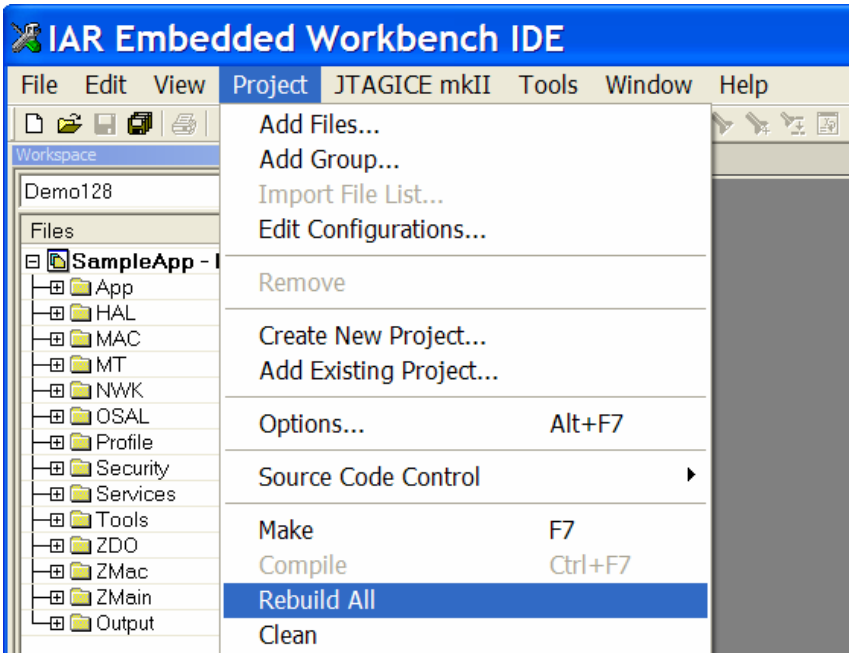
- Make sure the AVR Fuses and LockBits are correct (Appendix C)
- Make sure initial EEPROM contents have been saved (Section 4.4)
- Navigate to the SampleApp project directory and launch the IAR Embedded Workshop by double clicking on the ***SampleApp.eww*** file:



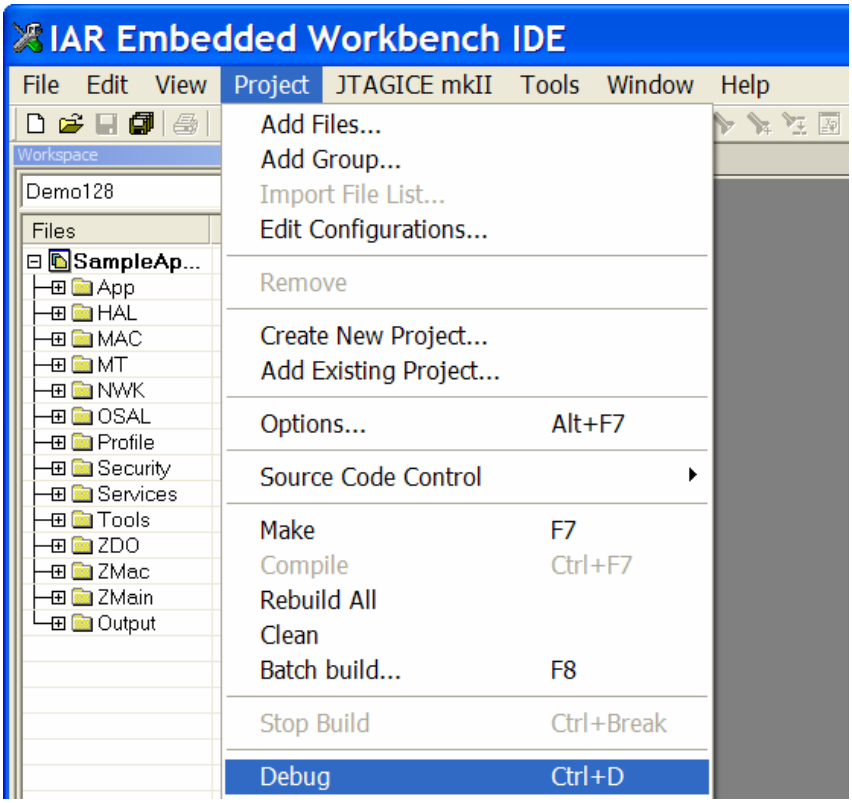
- Select the Demo128 project from the ***Workspace*** pull-down menu:



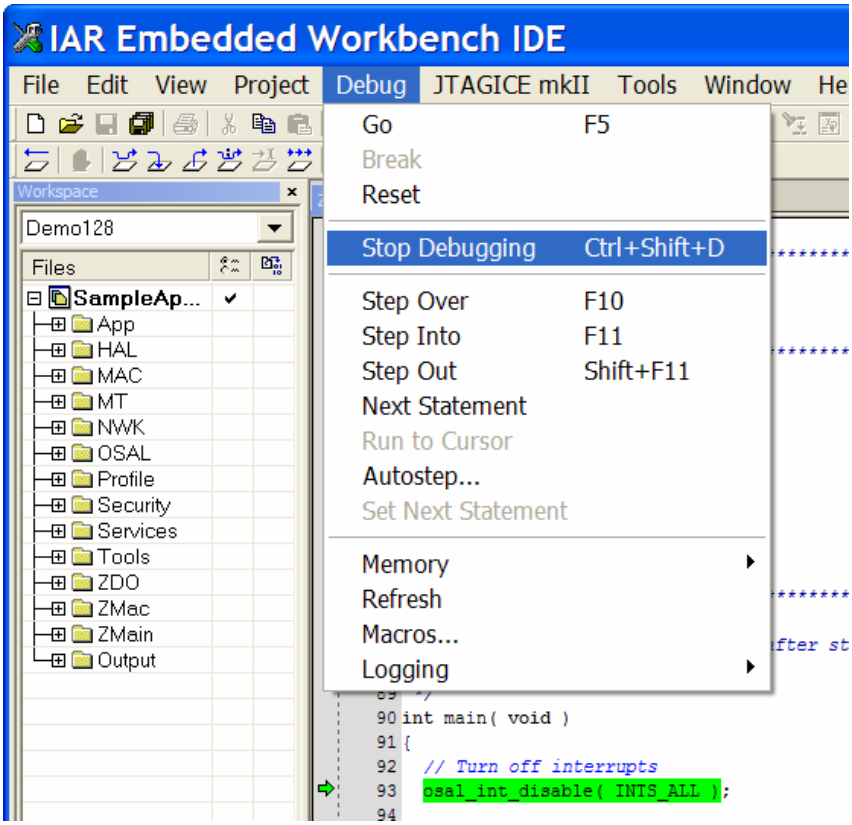
- Build the application by pulling down the ***Project*** menu and clicking on **Rebuild All**:



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



- After downloading to the CC2420DB is complete, exit the debugger by pulling down the ***Debug*** menu and clicking on **Stop Debugging**:



- Repeat the previous steps to program more CC2420DB 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 CC2420DB has a 5-position joystick, designated U5, which provides these switch inputs as shown in the table below.

Application and operational status are displayed via four LEDs, commonly referred to as LED1 through LED4. The CC2420DB has 4 colored LEDs, designated D301–D304, which provide these LED indications as shown in the table below.

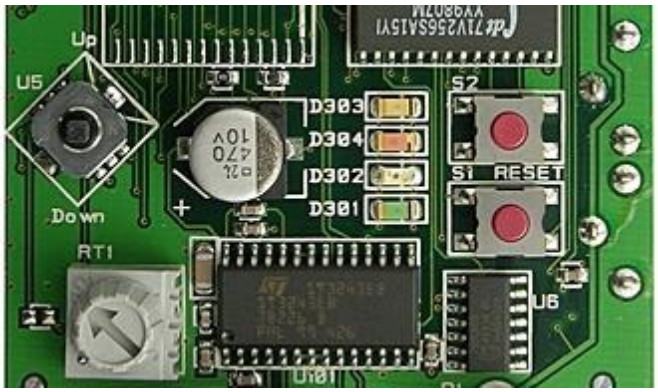


Figure 2: CC2420DB Switches and LEDs

SWITCH	JOYSTICK
SW1	***Up*** position
SW2	right position
SW3	***Down*** position
SW4	left position
SW5	press down

LED	LABEL	COLOR
LED1	***D301***	Green
LED2	***D302***	Orange
LED3	***D304***	Red
LED4	***D303***	Yellow

6.2. Initial Loading of 64-Bit IEEE Address

Normally, Z-Stack loads the 64-bit IEEE address from EEPROM upon power-up or reset. When the address has been reset (0xFFFFFFFFFFFFFFFF) by erasing the EEPROM, the program waits in a loop during start-up, 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 so that the developer can later use Z-Tool to enter the proper 64-bit extended address. If necessary, the user can reload previously saved EEPROM contents by following the procedure in Appendix E.

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 CC2420DB 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 *P3-5* and *P3-6*, as shown in Figure 3 below. Make sure that only one of the CC2420 boards has a jumper in this position.

Figure 3: CC2420 Coordinator Jumper

Initially, begin execution of the programmed SampleApp by applying power to the device that is configured as the ZigBee Coodinator. If LED1 (green) flashes, press SW5 (joystick center) to create a temporary IEEE address for the board (see Section 6.2). The device now performs a scan of the programmed ZigBee channel (see Section 7), temporarily flashing LED4 (yellow). Once the device successfully starts up a network, LED3 (red) will turn on and LED4 will stop flashing.

Next, power up a ZigBee Router device (no jumper on *P3*). If LED1 (green) flashes, press SW5 (joystick center) to create a temporary IEEE address for the board (see Section 6.2). This device now scans the programmed ZigBee channel for a network, temporarily flashing LED4 (yellow). Once it joins the network started by the Coordinator, LED3 (red) will be turned on and LED4 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 LED4 (yellow) 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 LED4. Pressing button SW2 on a device toggles that device's membership in Group 1, allowing the user to enable/disable LED4 flashing on that device.

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

7. PanID and Channel Selection

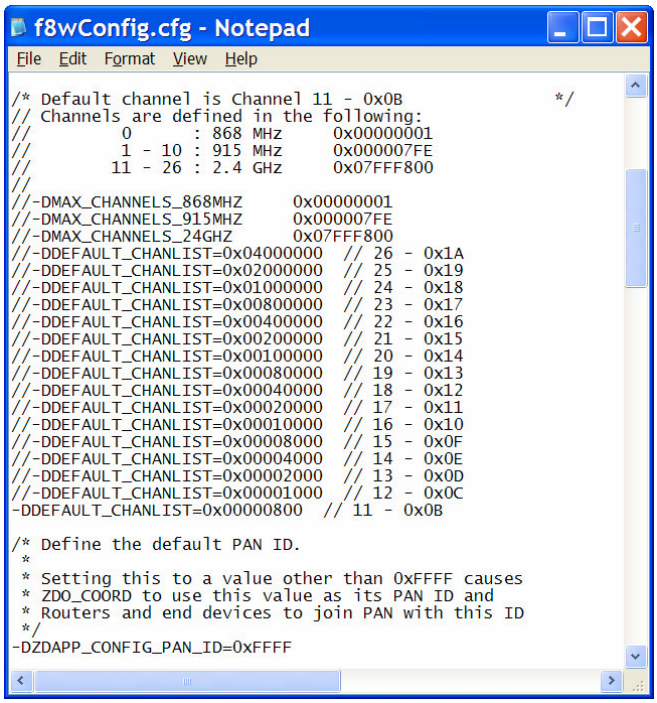
The ZigBee specification defines the use of a 14-bit Personal Area Network Identifier (PanID) to uniquely identify a network. Z-Stack provides the user with two methods of selecting a PanID when starting or joining a network by setting the value of *ZDAPP_CONFIG_PAN_ID*. Setting this parameter to 0xFFFF 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 (11^{th} 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\CC2420DB 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.



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.

8. External RAM

The CC2420DB has capability to provide up to 64 Kbytes of external (to the MCU) RAM via 32Kx8 chips labeled U3 and U10. Normally, CC2420DB boards have one external RAM chip, at U3, which can be accessed at memory locations 0x8000 through 0xFFFF. The Z-Stack projects utilize this memory space when the *EXTERNAL_RAM* compile option is placed in the linker command file. Z-Stack only places the OSAL heap in the external memory; the processor stack memory and compiler allocated RAM variables are always located in the MCU internal RAM.

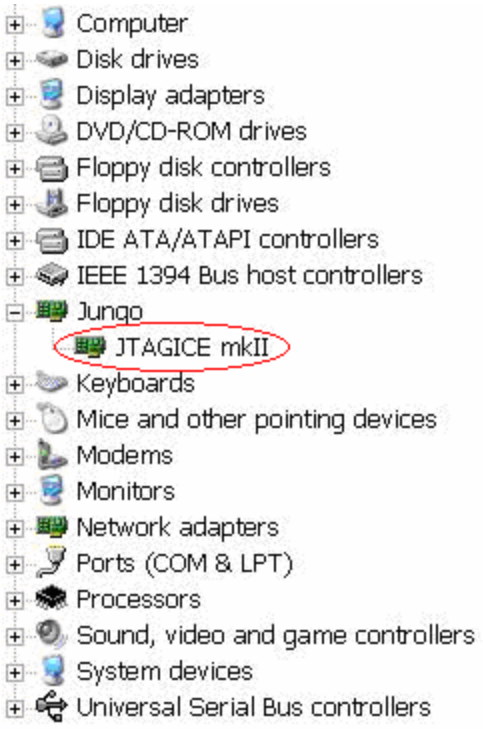
9. Low Voltage Detection

The ATmega128 processor is capable of detecting a “brown-out” situation and forcing a reset to occur. Referring to Appendix C, fuses are set to enable brownout detection at a level of 2.7 volts. When the supply voltage drops below that level and then recovers, the ensuing reset will indicate a low-voltage situation. The Z-Stack program responds to a low-voltage reset by hanging in a loop, sequencing through the 4 LEDs. This is intended to eliminate erratic operation when the supply voltage is marginal (batteries too low). Pressing the reset button on the CC2420DB board clears the low-voltage reset condition and allows the processor to attempt normal operation.

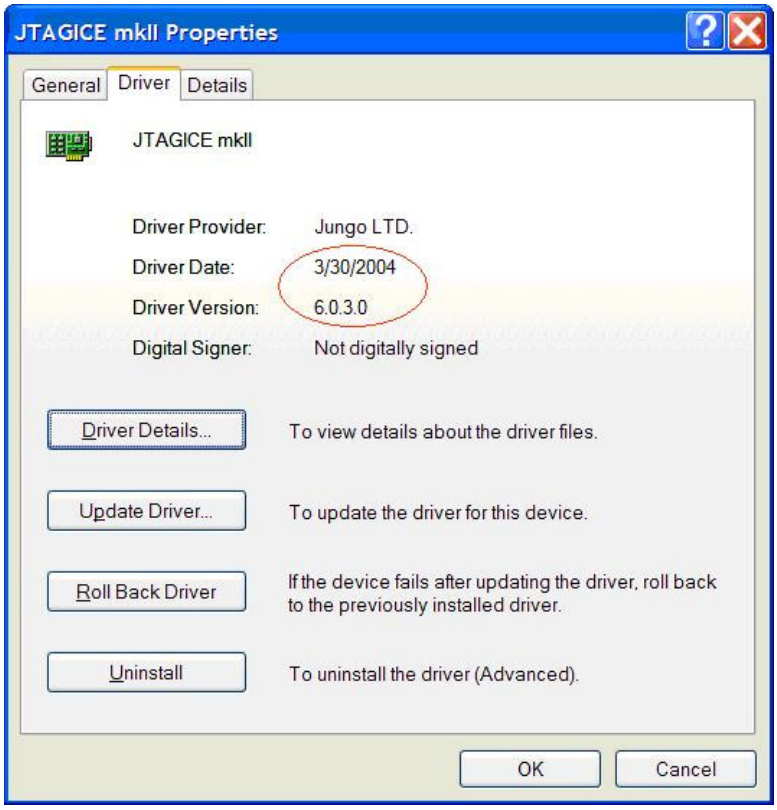
A. Setting Up The JTAGICE mkII For AVR Studio

The JTAGICE mkII device will be used by the Atmel AVR Studio IDE to configure processor Fuses and LockBits. The JTAGICE mkII connects to the host computer via USB or RS-232B. This document assumes use of the USB connection to obtain higher download and debugging speeds, as well as, no external power supply being required. The AVR Studio program supplies a USB driver to work with the JTAGICE mkII unit. Follow this general sequence to install the driver and set up the JTAGICE mkII and AVR Studio for initial usage:

- With power removed from the CC2420DB board, connect the JTAGICE mkII device to the 10-pin JTAG connector on the CC2420DB. Make sure the JTAG cable is oriented properly (pin 1 “points to” the RS232 connector) and apply power to the CC2420DB.
- Plug the USB cable into the JTAGICE mkII and into an available USB slot in the host computer. Apply power to the JTAGICE mkII by moving the small switch on the rear toward the center of the unit. The LEDs on top of the unit should turn on.
- Follow the prompts from Windows concerning detection of new hardware and installing driver software. Don’t be concerned with warnings about an unsigned driver.
- Navigate to the Windows Device Manager by following the links:
  Start→Settings→Control Panel→System→Hardware→Device Manager
- Find the device associated with the JTAGICE mkII and expand as shown:

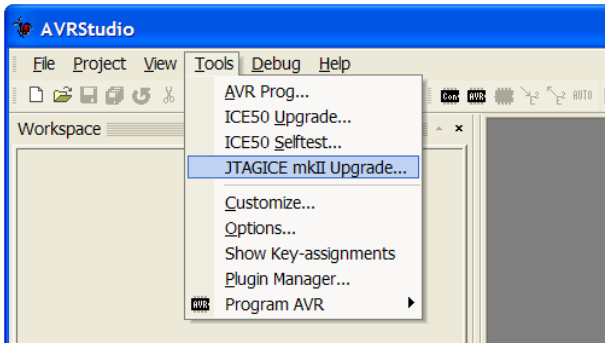


- Right click on the JTAGICE mkII icon to display a list of options. Select the **Properties** item and click on the ***Driver*** tab of the JTAGICE mkII Properties box. Verify that the proper driver has been installed:

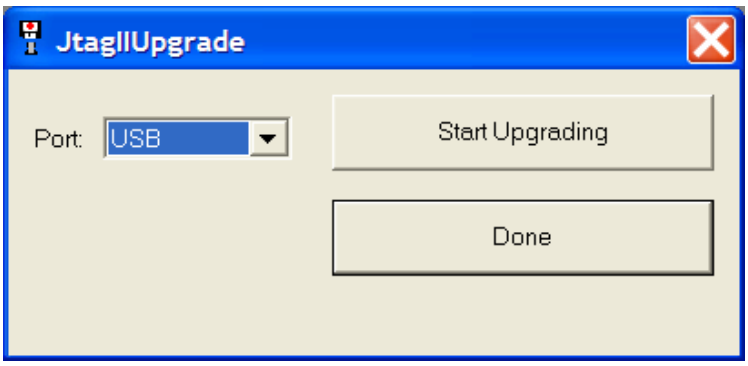


- Close all windows that were opened to get to this point

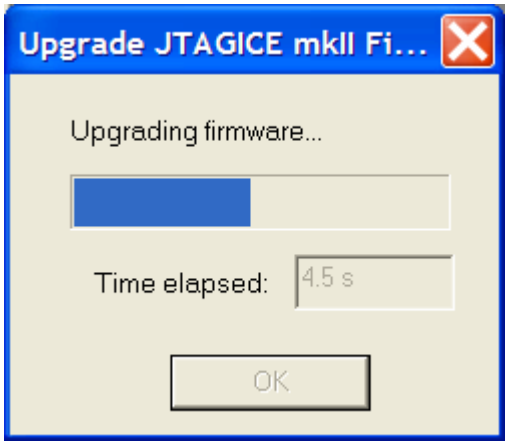
- On initial installation, it is likely that the firmware on the JTAGICE mkII device will be out of date. Launch the AVR Studio program and select ***JTAGICE mkII Upgrade…*** from the ***Tools*** pull-down menu:



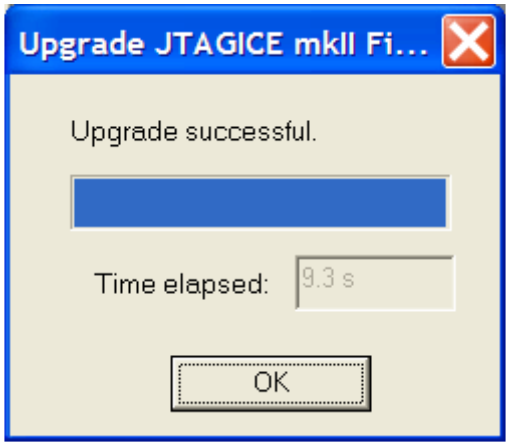
- Select the ***USB*** item in the ***Port:*** box and click on ***Start Upgrading***:



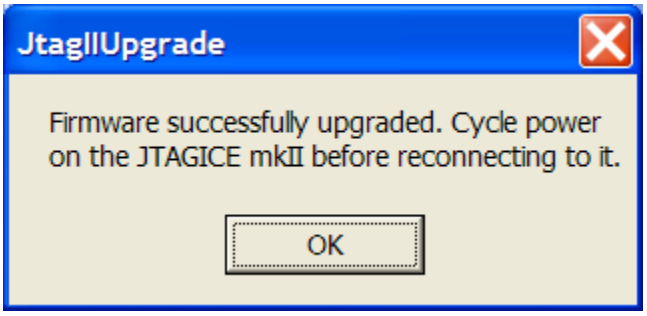
- Click the ***OK*** button to start the upgrade. The upgrade progress will be displayed:



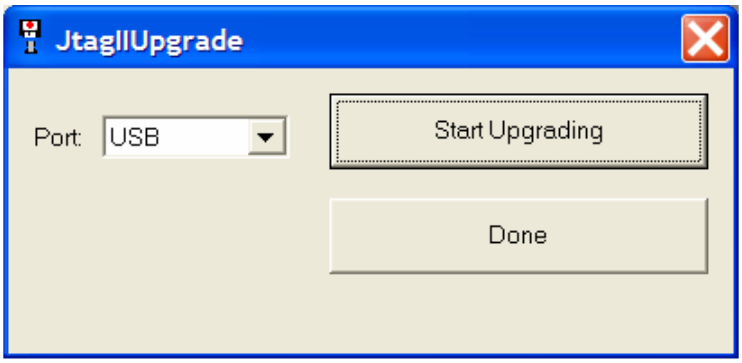
- When the firmware upgrade is complete, click the ***OK*** button to continue***:***



- Click the ***OK*** button to complete the firmware upgrade:



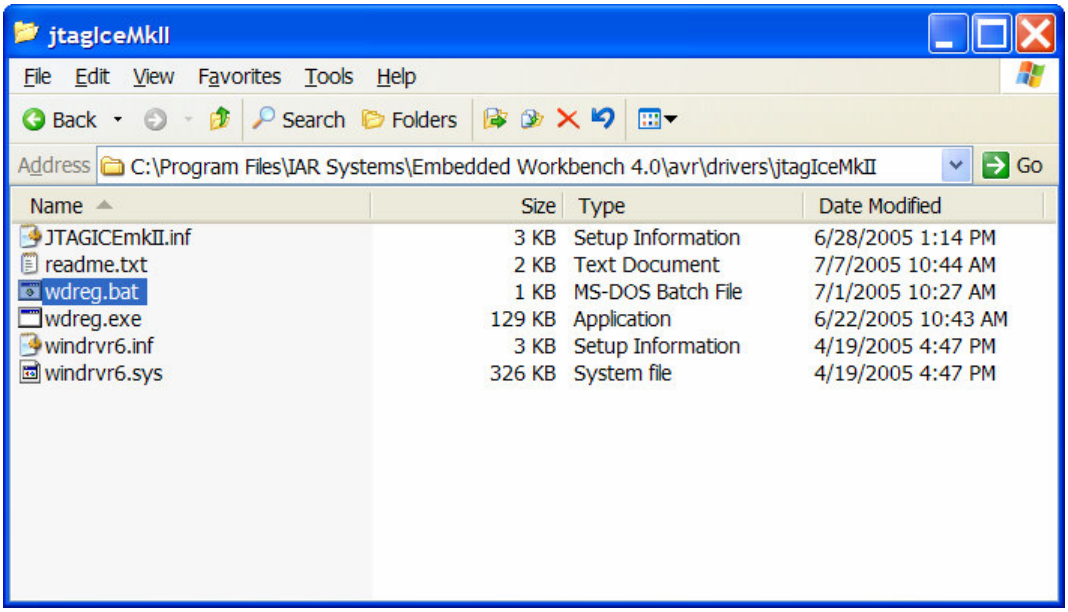
- Cycle power on the JTAGICE mkII unit and click the ***Done*** button:



B. Setting Up The JTAGICE mkII For EWAVR

The JTAGICE mkII device will be used by the IAR Embedded Workbench IDE for downloading and debugging programs on the CC2420DB. The JTAGICE mkII connects to the host computer via USB or RS-232. This document assumes use of the USB connection to obtain higher download and debugging speeds, as well as, no external power supply being required. The IAR EWAVR program supplies drivers to work with the JTAGICE mkII unit. Follow this sequence to install or update the drivers:

- Remove power from the JTAGICE mkII by moving the small switch on the rear away from the center of the unit. The LEDs on top of the unit should turn off.
- Using Windows, navigate to ***…\avr\drivers\jtagIceMkII*** folder shown below:



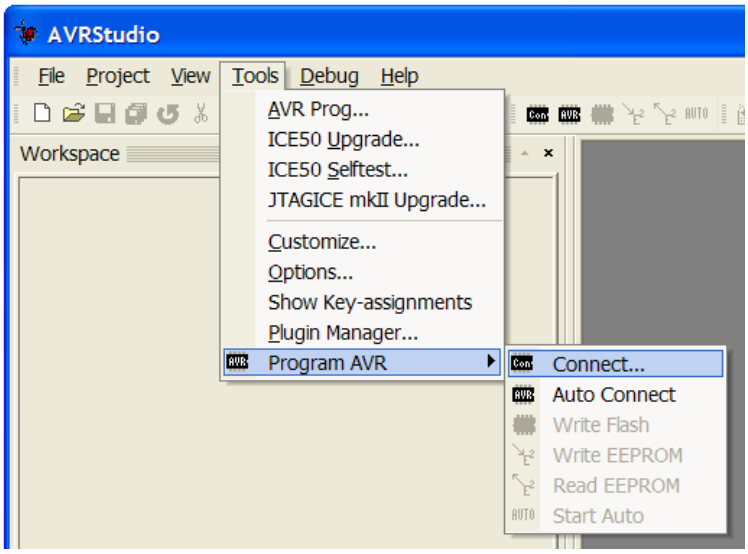
- Double click on the ***wdreg.bat*** file to update the drivers required by EWAVR.

C. Setting AVR Processor Fuses and LockBits

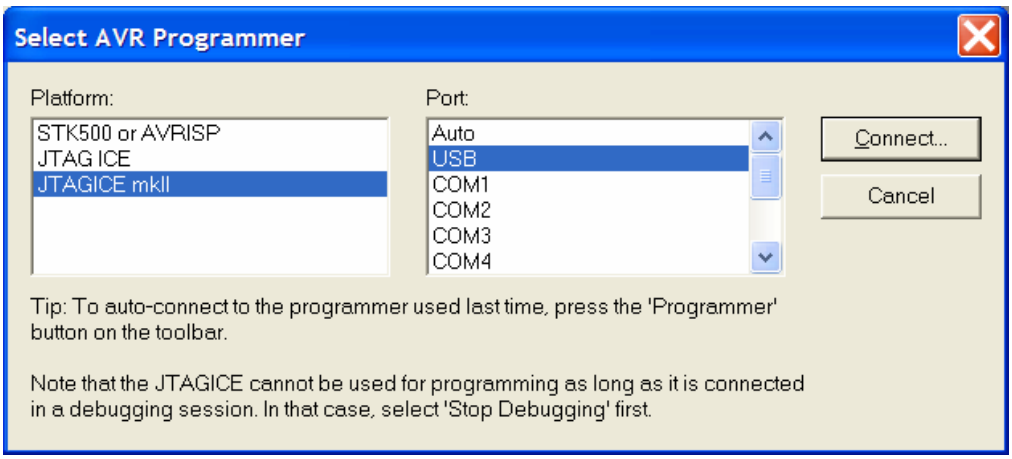
Before working with a Chipcon CC2420DB board, the Fuses and LockBits on its AVR processor should be initialized to default settings. Launch the AVR Studio program and perform the following sequence for each of the CC2420DB boards:

- With power removed from the CC2420DB board, connect the JTAGICE mkII device to the 10-pin JTAG connector on the CC2420DB. Make sure the JTAG cable is oriented properly (pin 1 "points to" the RS232 connector) and apply power to the CC2420DB. Red and green LEDs should now be lit on top of the JTAGICE unit.

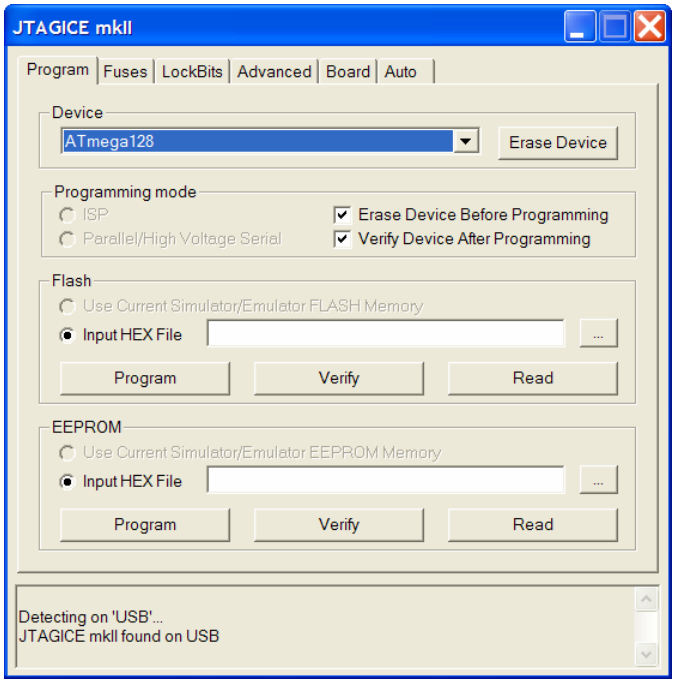
- Select ***Program AVR→Connect…*** from the ***Tools*** pull-down menu:



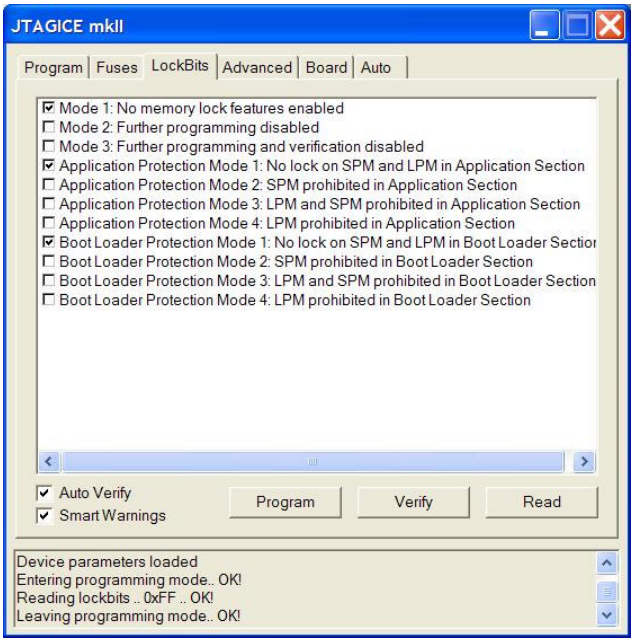
- Select ***JTAGICE mkII*** and ***USB,*** then click on the ***Connect…*** button:



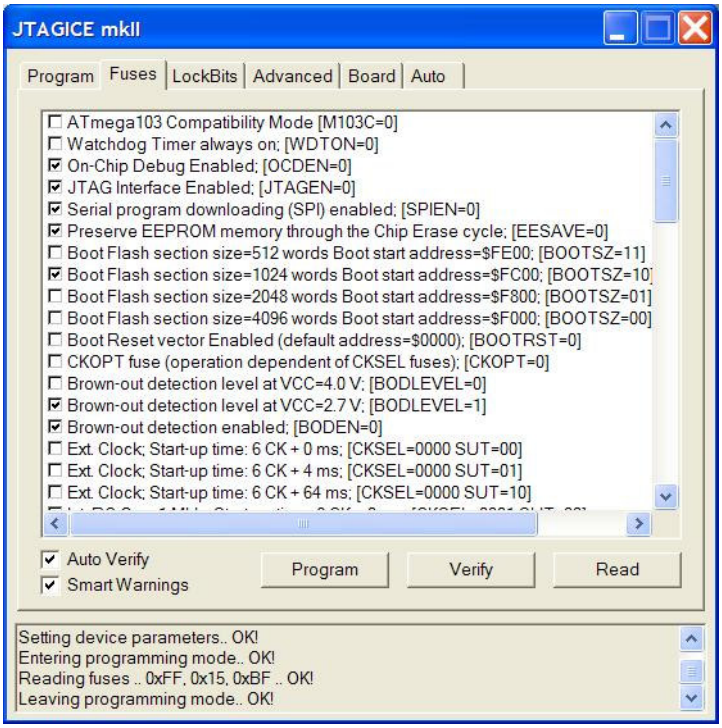
- Select the ***Program*** tab and check for the settings shown below. If different, make the required changes. Make sure the selected ***Device*** is the ATmega128, processor used on the CC2420DB. Check the dialog box below to verify successful connection via USB:

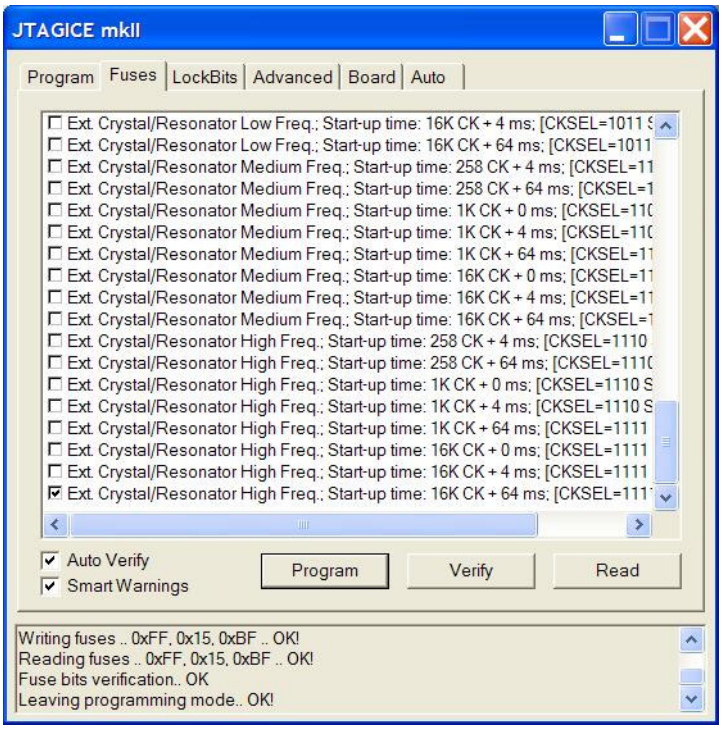


- Select the ***LockBits*** tab and check for the settings shown below. If different, make the required changes and click on the ***Program*** button. Check the dialog box below to verify that the lock bits were programmed properly:



- Select the ***Fuses*** tab and check for the settings shown below. Make sure that the ATmega103 Compatibility Mode fuse is disabled. Scroll down to check all fuse settings. If different, make the required changes and click on ***Program*** button. Check the dialog box below to verify that the fuses were programmed properly:



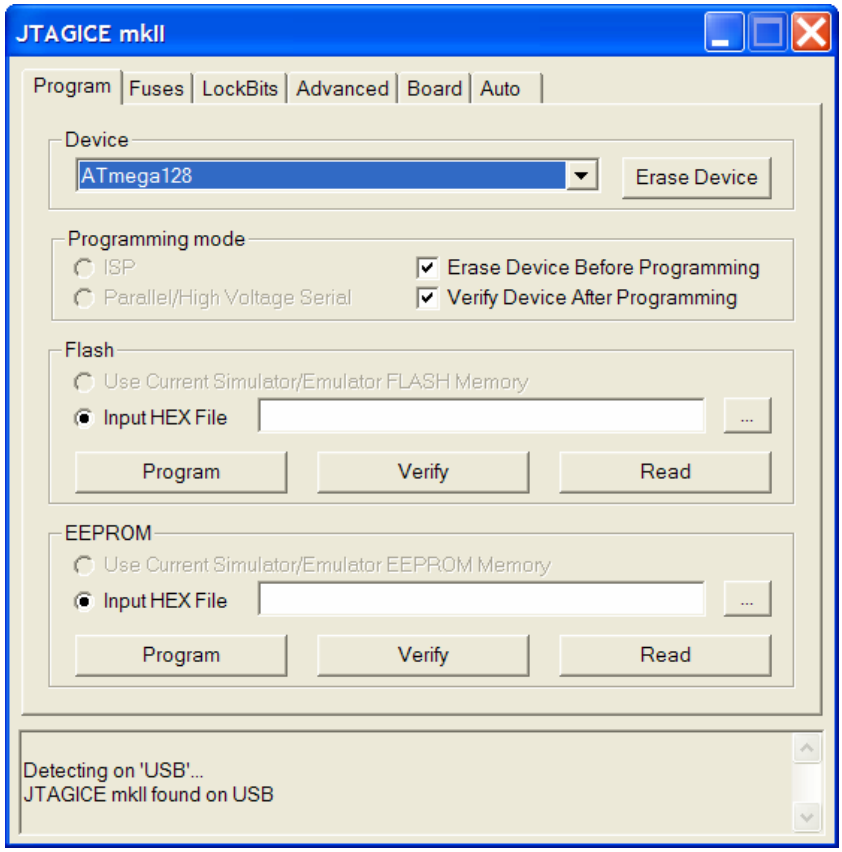


D. Saving EEPROM Contents To Disk

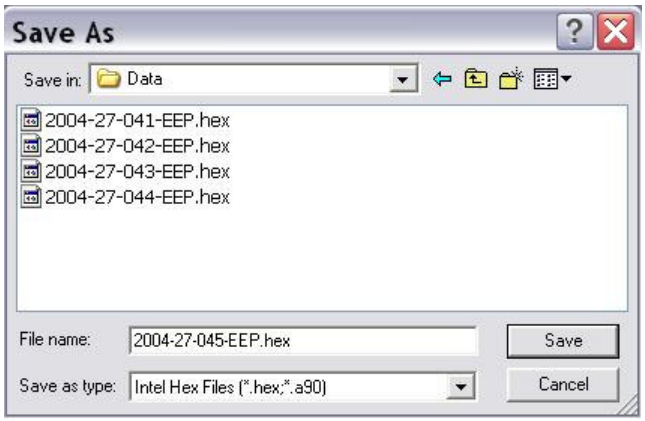
The EEPROM memory on the CC2420DB board is used to store its unique 64-bit IEEE address and possibly other non-volatile information. Occasionally, the contents of the EEPROM need to be saved to a disk file so that they may be restored at a later time. The AVR Studio program, in conjunction with the JTAGICE mkII device, will be used to upload the EEPROM data from the AVR processor and record it to disk files specified by the user.

When a program is loaded into the AVR processor, via the JTAG interface, the programming device can be configured to erase or preserve the contents of the EEPROM memory. For this reason, it is important to save the EEPROM contents to a disk file before creating and loading programs for the first time. The following procedure can be used to save EEPROM data initially or at any later time:

- With power removed from the CC2420DB board, connect the JTAGICE mkII device to the 10-pin JTAG connector on the CC2420DB. Make sure the JTAG cable is oriented properly (pin 1 "points to" the RS232 connector) and apply power to the CC2420DB. Red and green LEDs should now be lit on top of the JTAGICE unit.
- Launch the AVR Studio IDE and click on the ***AVR*** icon. When AVR Studio detects the JTAGICE mkII unit and connects to the processor on the CC2420DB, the **JTAGICE mkII** programming settings box will appear. Select the ***Program*** tab:



- Click on the ***Read*** button in the EEPROM area to display the **Save As** dialog box:



- Browse for the appropriate folder to save the EEPROM data file. When Z-Stack was installed, the following folder was created to store the initial EEPROM files:

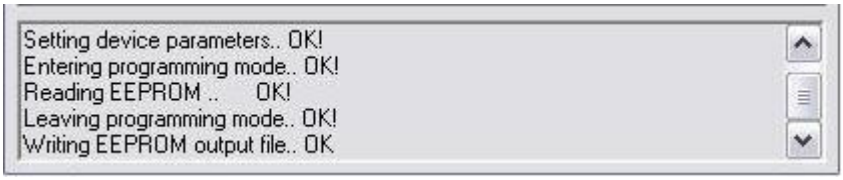
  C:\Texas Instruments\ZStack-1.4.2\Projects\zstack\Tools\Data

  This is the recommended location to save the initial EEPROM data files. Note that the above example shows that 4 files have previously been saved in the selected folder.

  The application developer may decide to store additional items in the EEPROM memory. It may then be useful to save copies of EEPROM data at later times. It is recommended that these files be stored in a different location than the initial files. As with any valuable data, it's a good idea to make back-up copies of all saved EEPROM data files.

- Type a new filename into the **File name:** box and click on the ***Save*** button to start the upload and file saving process. Any convenient file naming convention can be used, as long as the resulting EEPROM data file can later be correlated with the CC2420DB board that it belongs to. The example above shows filenames that were created using the serial numbers found on the bottom surface of the CC2420DB boards.

- Returning to the Program Dialog box, AVR Studio uploads the contents of EEPROM, graphically showing a progress bar at the lower left edge of the screen. After a few seconds, the specified file is written to disk, and the following information should be displayed in the status box:

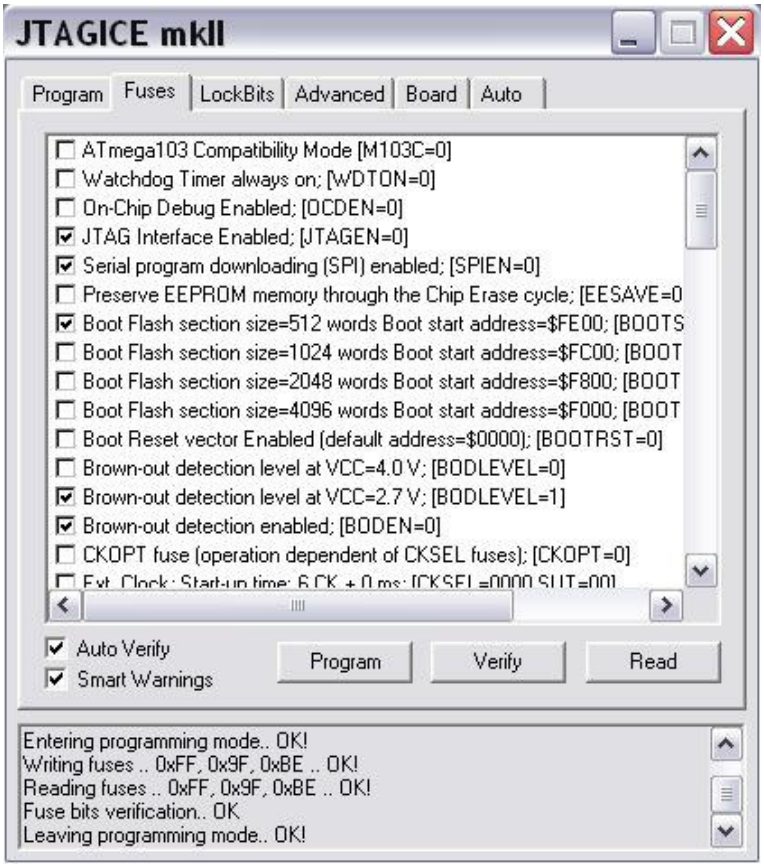


- Close the **JTAGICE mkII** programming settings box.

E. Restoring EEPROM Contents From Disk

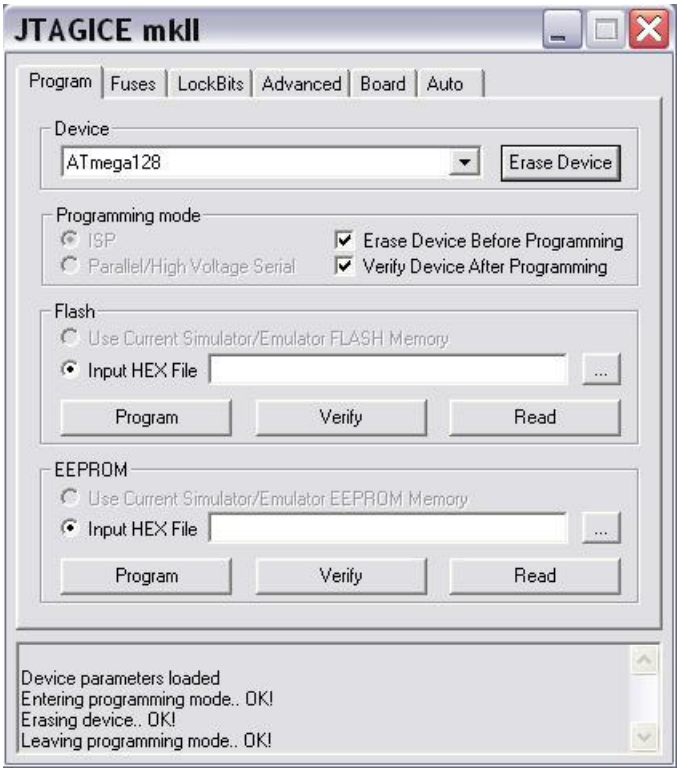
The EEPROM memory on the CC2420DB board is used to store its unique 64-bit IEEE address and possibly other non-volatile information. Occasionally, the contents of the EEPROM need to be restored from a disk file. The AVR Studio program, in conjunction with the JTAGICE mkII device, will be used to download EEPROM data from a disk file specified by the user to the AVR processor EEPROM memory. The following procedure can be used to restore EEPROM data to the AVR processor:

- With power removed from the CC2420DB board, connect the JTAGICE mkII device to the 10-pin JTAG connector on the CC2420DB. Make sure the JTAG cable is oriented properly (pin 1 "points to" the RS232 connector) and apply power to the CC2420DB. LEDs should now be lit on top of the JTAGICE unit.
- Open the AVR Studio IDE and click on the ***AVR*** button icon. When AVR Studio detects the JTAG ICE mkII unit and connects to the processor on the CC2420DB, the **JTAGICE mkII** programming settings box will appear. Select the ***Fuses*** tab:



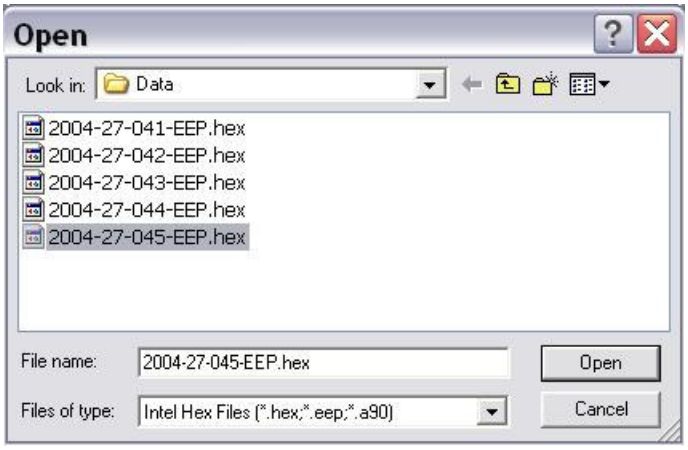
- Disable the *Preserve EEPROM memory through the Chip Erase Cycle* fuse and click on the ***Program*** button. Examine the status box for proper fuse programming.

- Select the ***Program*** tab and click on the ***Erase Device*** button. Examine the status box to verify correct operation, as shown below:



- Return to the ***Fuses*** tab, enable the *Preserve EEPROM memory through the Chip Erase Cycle* fuse and click on the ***Program*** button. Examine the status box for proper fuse programming. Now return to the ***Program*** tab.

- Click on the ***Browse*** button in the EEPROM area to display the **Open** dialog box:

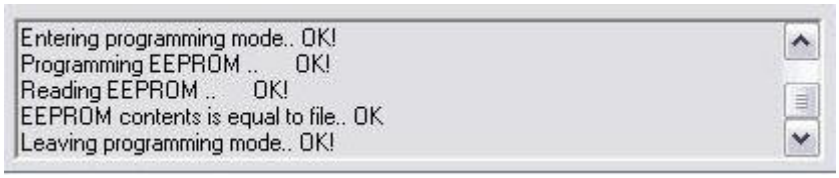


- Browse for the appropriate folder to find the needed EEPROM data file. When ZStack was installed, the following folder was created to store the initial EEPROM files:

  C:\Texas Instruments\ZStack-1.4.2\Projects\zstack\Tools\Data

- Select an EEPROM data file and click the ***Open*** button to return to the JTAGICE dialog..

- Back at the JTAGICE Dialog box ,click the ***Program*** button in the EEPROM area to start the download process. AVR Studio downloads EEPROM, graphically showing a progress bar at the lower left edge of the screen. After a few seconds, EEPROM programming is complete, and the following information should be displayed in the status box:



- Close the **JTAGICE mkII** programming settings box.

F. 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.