CC2530ZDK Sensor Demo

User’s Guide

Table of contents

1 Introduction

This document is a user's guide describing the details of how to run the Sensor Demo application that is pre-programmed on the CC2530EM's in the CC2530ZDK.

The Sensor Demo application is built on top of TI's ZigBee® software implementation Z-Stack™. Z-Stack is TI's ZigBee compliant protocol stack for a growing portfolio of IEEE 802.15.4 products and platforms. Z-Stack supports the CC2530 and is compliant with both the ZigBee 2007 (ZigBee and ZigBee PRO) and ZigBee 2006 specification. More information and links to download Z-Stack for CC2530 can be found on the product folder www.ti.com/z-stack [1].

The Sensor Demo application consists of collector devices and sensor devices. The collector devices help to establish the network topology (starting the network, acting as router, etc.), and one collector node is used as gateway to the PC. The sensor devices will periodically send temperature reports towards this central gateway that is connected via a serial interface to the PC. The sensor data is visualized on the PC using a tool called ZSensorMonitor. The ZSensorMonitor is described in the ZSensorMonitor User's Guide [2].

Chapter 3 describes the necessary hardware and software for running the application, while Chapter 4 describes installation and setup of the required software. Chapter 5 describes the necessary steps to run the Sensor Demo application, and the software for this demo. This document also describes how to set up the hardware and software development environment in order to develop your own ZigBee applications.

Refer also to the CC2530ZDK Quick Start Guide [3] for a description of the steps necessary to run the out-of-box demo on the CC2530ZDK. The CC2530ZDK User's Guide document [4] gives more information about the hardware included in the CC2530ZDK kit.

For more details about ZigBee and Z-Stack see also the ZigBee Developer's Guide document [5] that is included in the Z-Stack documentation [1].

2 Abbreviations

API	-	Application Programming Interface
BB	-	Battery Board
EB	-	Evaluation Board
EM	-	Evaluation Module
GUI	-	Graphical User Interface
LED	-	Light Emitting Diode
PAN	-	Personal Area Network
ZDK	-	ZigBee Development Kit
Z-Stack	-	TI's ZigBee software implementation

3 Prerequisites

3.1 Hardware

To successfully download and run the software described in this document a CC2530ZDK is needed or alternatively the following material:

- 2 x SmartRF05EB or SmartRF05BB (At least one of the boards must be a SmartRF05EB)
- 2 x CC2530EM with antennas
- Serial cable
- USB cable
- 4 x AA batteries

This will enable two CC2530 based ZigBee nodes which is the minimum number of nodes to run this application. More nodes are recommended for visualization; however, two is the minimum for this application.

3.2 Software

The ZSensorMonitor software (see Figure 1 below) is required in order to run the Sensor Demo application with the preprogrammed firmware on the CC2530EM's part of CC2530ZDK. ZSensorMonitor can be downloaded from the CC2530ZDK website [6].

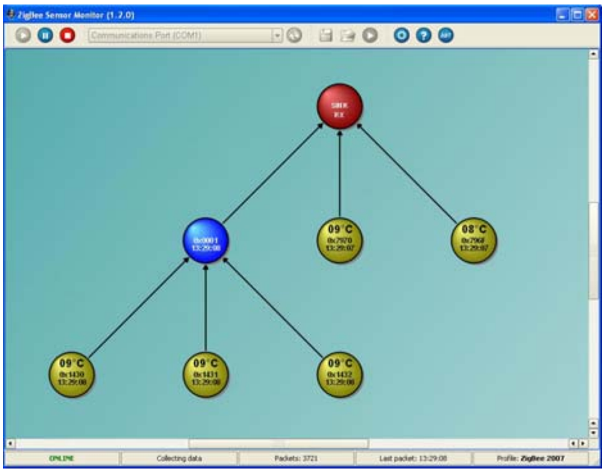


Figure 1 ZSensorMonitor

For further code development and modification of the firmware the following software is required in addition:

- Z-Stack 2.3.0 for CC2530. This software can be downloaded from the Z-Stack product folder on the web [1].

- IAR Embedded Workbench for 8051 version 7.51. A free 30 days evaluation is available on www.iar.com/ew8051. The CC2530ZDK also includes a CD with this software. Refer to the CC2530ZDK User's Guide [4] for further information.

- Sensor Demo application software package. This package includes source code for the demo application and can be downloaded from the CC2530ZDK website [6].

4 Installation & Setup

In order to run the Sensor Demo with the preprogrammed firmware on the CC2530EM's part of CC2530ZDK, follow the steps described in the CC2530ZDK Quick Start Guide document [3].

For software development and modification of the preprogrammed firmware the following steps are needed in addition:

1. Install Z-Stack™ 2.3.0 for CC2530.
   It is recommended to download and install the swrc126b.zip (56.9MB) as it contains the ZStack-CC2530-2.3.0-1.3.0.exe (the "full" CC2530 Z-stack v.2.3.0 & Application Examples v.1.3.0), which gives you additional application examples to work with.
2. Install IAR Embedded Workbench for 8051. Please consult Z-Stack's release notes to see which version of IAR to use. (More info about IAR EW 8051 is found on IAR's websites www.iar.com/ew8051).
3. Download the CC2530 Sensor Demo zip file (SWRC147A.zip) from the kit website [6] and unzip this file into the following directory of the Z-Stack installation:
   ..\Projects\zstack\Samples\Projects\zstack\Samples

   If Z-Stack was installed on the default path, the folder to copy the files into will be:
   C:\Texas Instruments\ZStack-CC2530-2.3.0-1.4.0\Projects\zstack\Samples

4. The Getting Started Guide and Z-Stack Developer's Guide found in the Z-Stack folders give further information on how to start ZigBee software development with Z-Stack.

5 ZigBee Sensor Demo

5.1 Introduction

The CC2530ZDK Sensor Demo application is a sensor network demonstration where sensor nodes periodically report their data to a central gateway device. The gateway communicates with a PC for visualizing the sensor data received. The ZSensorMonitor PC application is used to visualize the data from the sensors on the network. More information about ZSensorMonitor is found in the ZSensorMonitor User's Guide [2].

The Sensor Demo application consists of two different device types:

Collectors
After power-up the collector nodes are configured as ZigBee routers in the network, and will try to discover and join a ZigBee network. When the user presses **joystick up** the device will reconfigure itself to act as a ZigBee coordinator and start up an own network with a unique PAN id. This device will also be the gateway device in the network i.e. the device that should be connected to the PC for visualization of the sensor data. When the user presses **joystick right** on this device it will accept binding requests from other nodes. Make sure only one node (i.e. the node connected to the pc) in the system is in this accept mode.

While the gateway node is the collector node that is connected to the PC and functions as the ZigBee Coordinator, the other collector nodes function as ZigBee routers in the network. The routers help to extend the covered range and number of sensors that can be supported by providing hops for message routing. After a collector node (that should act as router) is successfully started the user can start periodic reporting by pressing **joystick down**. The collector node will then send 'dummy' reports towards the gateway node. These reports do not contain sensor data but are used to visualize the collector node on the ZSensorMonitor PC tool.

Sensors
The sensor devices are configured as ZigBee end devices. After power-up these nodes will automatically join the network and bind to the gateway node if the user remembered to allow the

gateway to accept binding; see Collectors description above. Hence, the sensor nodes should be powered up after the gateway node in the system. When the user presses **joystick down** the sensor node will start periodic reporting towards the gateway, and the temperature reported will be displayed in the GUI on ZSensorMonitor. As the sensor nodes are configured to be ZigBee end devices they can use their sleep mode to save power (more details in Section 5.2.2).

The following section describes how to setup and run the Sensor Demo application.

5.2 Start up

This section describes the necessary steps to start up the devices on the network.

NB: The CC2530EM's in the CC2530ZDK are preprogrammed with the Sensor Demo application where 2 of the nodes are programmed as collectors and 5 of the nodes are programmed as sensors. If you are not using a CC2530ZDK or wish to reprogram the EM's, consult Appendix A: Programming the boards.

NB: Before powering up the boards make sure that the EM Selection switch (P19) is placed in position *SOC/TRX*. Also make sure that the RS232 Enable switch (P14) is placed in position *Enable* on the board to be connected with serial cable.

5.2.1 Setting up the collectors

1. Connect one of the devices that are programmed as collector to the PC with a serial cable. This device will be the gateway device that receives reports from the sensors and sends the reports to the PC application for visualization.

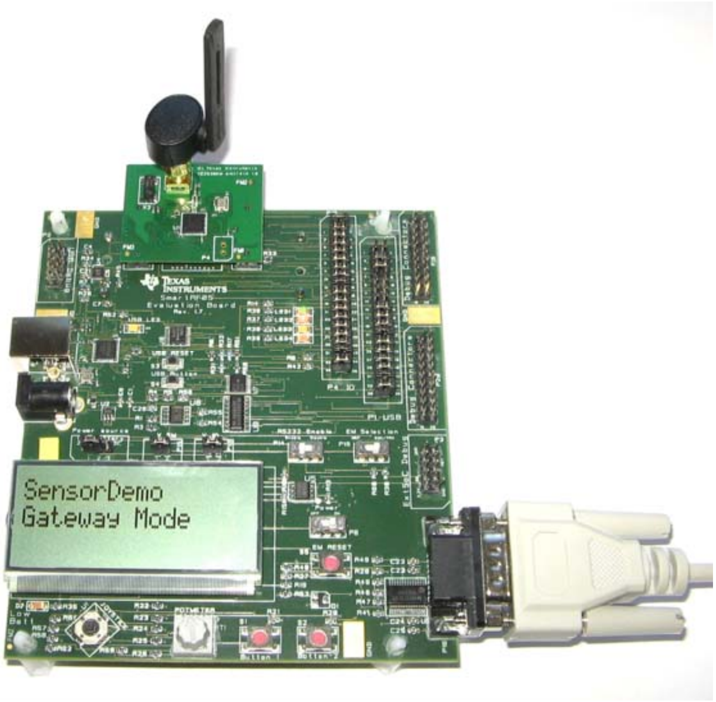


Figure 2 Gateway Node with serial cable

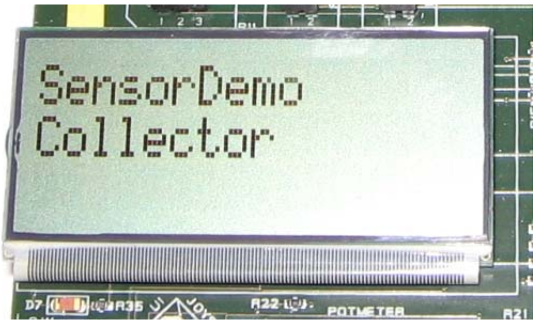
2. Power up the collector node that is connected to the PC (the gateway node). Press **joystick center** and hold it pressed during power up to bypass NV information i.e. start up with fresh network information[1]. In this state the node is a ZigBee router searching for a network; hence, LED 1 and LED 2 will be blinking indicating that the device is searching for a PAN to join.

[1] See also section 5.3.7 Bypass Non-Volatile memory

3. On the gateway node press the **joystick up**. The node will then reconfigure itself as a ZigBee coordinator and start up a network. LED 1 will be set ON to indicate the device successfully established a network.

4. Press **joystick right** on the gateway node. This will let the node accept application binding requests from other nodes. LED 2 will be set ON to indicate that the node is accepting bind requests. The LCD on the EB board will display “Gateway Mode” as seen in Figure 2.

5. Power up the other nodes that are programmed as collectors. Press **joystick center** during power up to start with fresh network information[1]. LED 1 and LED 2 will start blinking for a short period indicating that the node is searching for a PAN to join. After a short period LED 1 will be set ON to indicate that the node has joined the network. LED 2 will be set ON to indicate that the device is successfully bound to the gateway node. The LCD on the collector node will display the text as seen in Figure 3..



SensorDemo
Collector

Figure 3 Collector

5.2.2 Setting up the sensors

6. Power up all the nodes that are programmed as sensors (for the CC2530ZDK you should have mounted the corresponding 5 CC2530EM's on the SmartRF05BB boards as shown in Figure 4 Sensor device using SmartRF05BBFigure 4). Also press **joystick center** during start up on these nodes[1]. These nodes will automatically join the network and bind to the gateway. The LED's will be blinking slowly to indicate that the device is in the process of joining/starting a network. When the sensor is successfully joined to the network and bound to the gateway the LED's on the sensor will start blinking rapidly.

   NB: The sensors are ZigBee end devices and are using the power saving feature of Z-Stack. When using power saving the LED's will blink rapidly instead of being solid on since the LED's are turned off when the device goes to sleep and on when it wakes up from sleep.

   NB: Since the temperature sensors on the CC2530EM's are not calibrated, this sensor application is self calibrating and assumes a temperature of 22º Celcius when the sensor is powered up. If the sensor is powered up at other temperatures the code for the sensor can be modified (DemoSensor.c) to calibrate for another starting temperature.

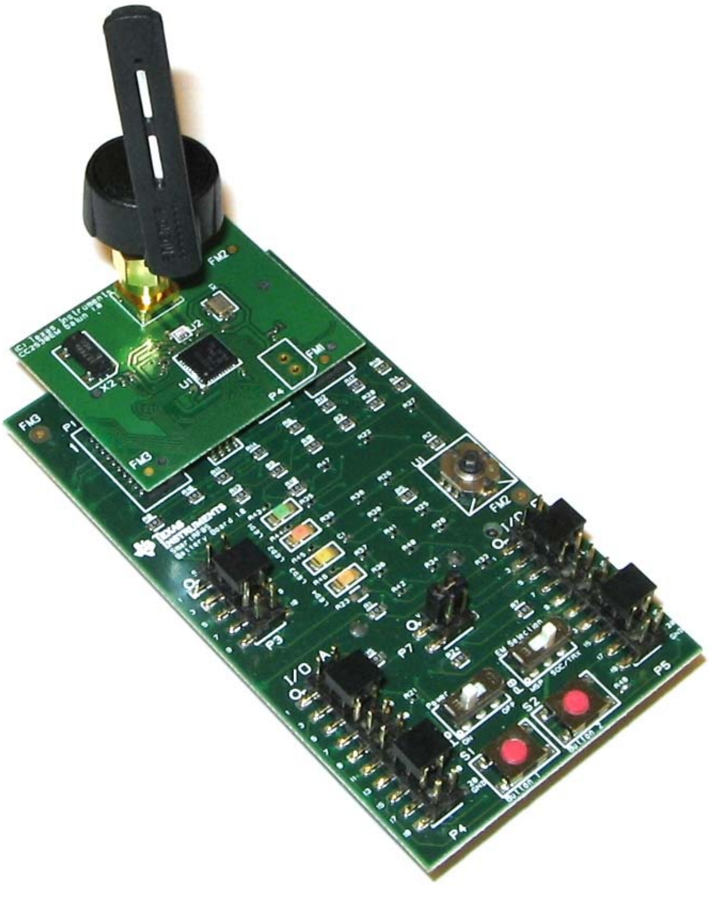
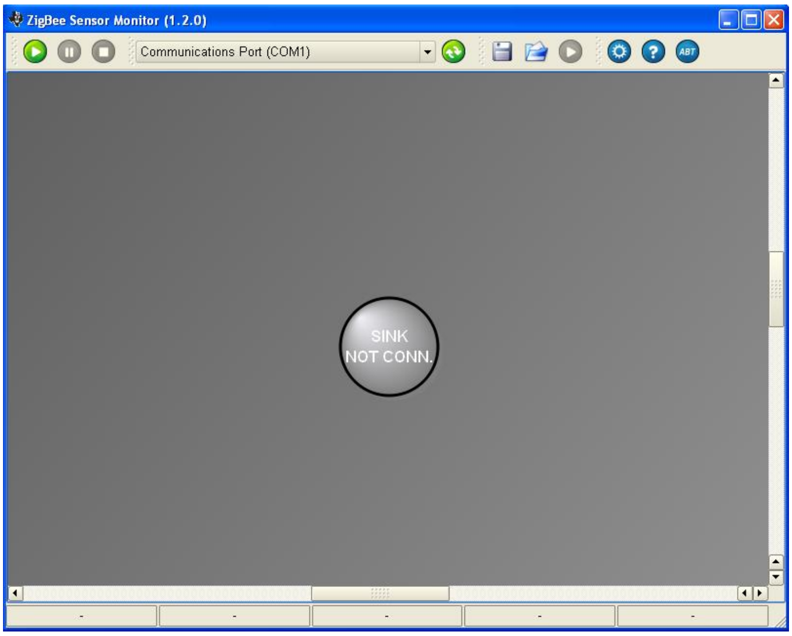


Figure 4 Sensor device using SmartRF05BB

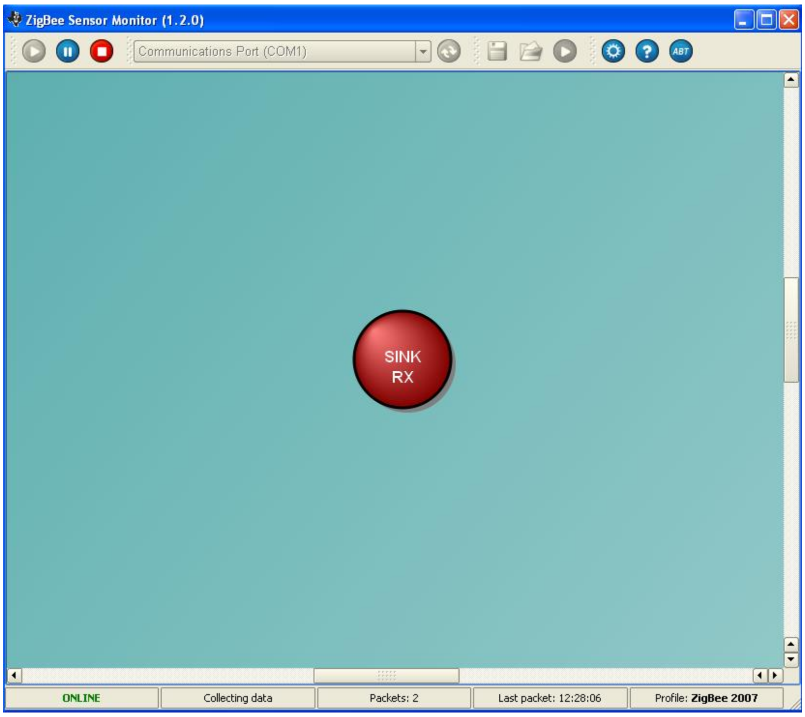
5.2.3 Starting up the ZSensorMonitor and the sensor reporting

1. Start the ZSensorMonitor PC application and choose the correct COM port on your system from the dropdown list in the top (Check Windows Device Manager if in doubt of which COM port to use on your system):



Communications Port (COM1)

2. Press the **play** button in the top left corner. The Sink node (gateway node) will turn red to indicate that the gateway node is successfully connected.



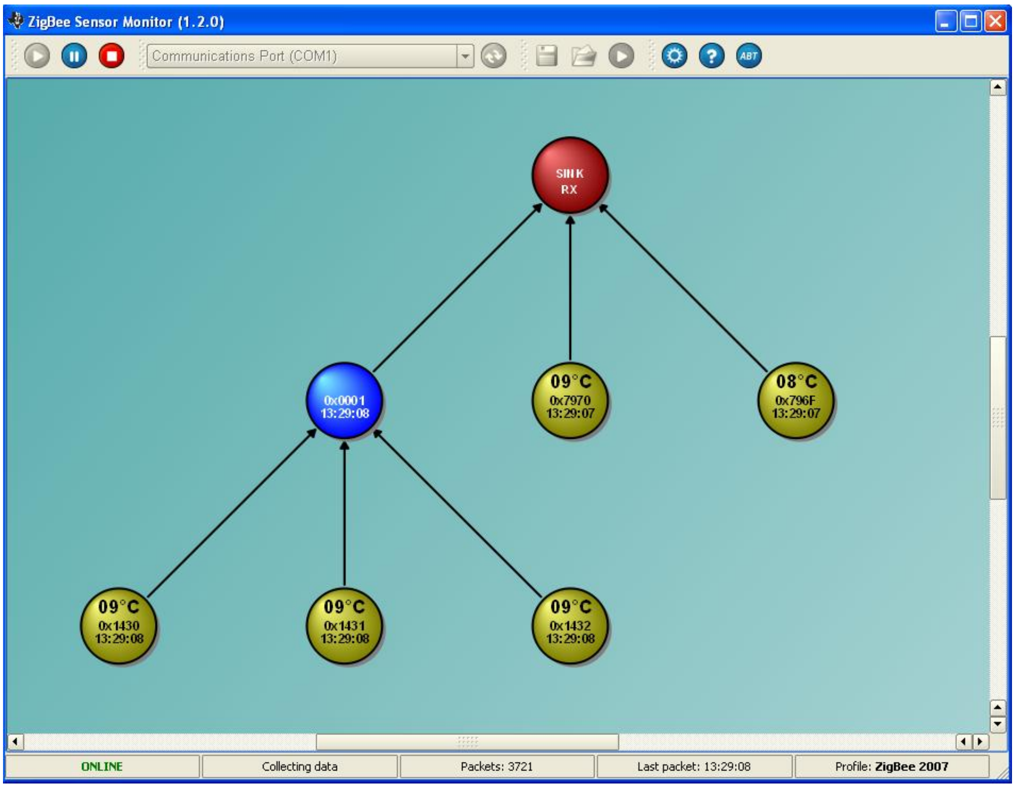
Communications Port (COM1)

ONLINE | Collecting data | Packets: 2 | Last packet: 12:28:06 | Profile: **ZigBee 2007**

3. After **joystick down** is pressed on the devices they will start reporting and be visualized in the ZSensorMonitor GUI as shown below, showing temperature from each sensor and the time of

the last report. As mentioned in the introduction it is important to also press **joystick down** on the collector nodes in order to get them and the network topology displayed.

The topology below can be obtained by starting two of the sensor nodes and the 2nd collector node first. Then press **joystick left** on the gateway device. This will disallow further nodes to associate directly with the gateway. When the 3 last sensor nodes are powered up they will instead associate to the collector node with address 0x0001 as shown below.



ONLINE | Collecting data | Packets: 3721 | Last packet: 13:29:08 | Profile: **ZigBee 2007**

4. Details about the ZSensorMonitor user interface are found in ZSensorMonitor User's Guide [2].

5.3 *Software*

This section describes the software that is used to build the hex files for the Sensor Demo application. The source code for this software is found in the CC2530ZDK Sensor Demo zip file found on the CC2530ZDK website [6]. See also Section 3.2 to make sure that you have set up the correct environment to work with this code.

5.3.1 Network Settings

The software that is programmed on the CC2530ZDK uses IEEE 802.15.4 channel 11 (2405 MHz). The PAN ID used is dependent on the last 2 bytes of the Coordinator's IEEE address (i.e. ZDAPP_CONFIG_PAN_ID=0xFFFF in the file f8wConfig.cfg in Z-Stack).

It is possible to choose another channel and PAN ID by using other settings in the file f8wConfig.cfg in Z-Stack. See also Z-Stack Developer's Guide [5] for further information.

5.3.2 Software structure

The IAR project (SensorDemo.eww) is organized into 4 different configurations:

- CollectorEB: This configuration is used for the collectors on either SmartRF05EB or SmartRF05BB target boards. This configuration is using the ZigBee 2007 stack profile.

- SensorEB: This configuration is used to build software for the sensor devices on either SmartRF05EB or SmartRF05BB targets boards. As the CollectorEB configuration it uses the ZigBee 2007 stack profile.

- CollectorEB-PRO: Same as CollectorEB but using the ZigBee 2007 PRO stack profile i.e ZIGBEEPRO compile option is enabled.

- SensorEB-PRO: Same as SensorEB but using the ZigBee 2007 PRO stack profile i.e ZIGBEEPRO compile option is enabled.

The IAR project files are found in the directory "*..\Projects\zstack\Samples\SensorDemo\CC2530DB"* after the software package is unzipped (see Section 4).

If Z-Stack was installed on the default path the IAR project files will be found on the following directory:
"C:\Texas Instruments\ZStack-2.2.0-1.3.0\ZStack-CC2530-2.2.0-1.3.0\Projects\zstack\Samples\SensorDemo\CC2530DB"

The source code for the Sensor Demo application is structured into the following files:

- DemoApp.h: Contains common constants and definitions, and interfaces to common functions for the application.
- DemoAppCommon.c: Contains implementation of common functionality between collector and sensor.
- DemoCollector.c: Contains implementation of the collector application.
- DemoSensor.c: Contains implementation of the sensor application.

The source files are found in the directory "..*\Projects\zstack\Samples\SensorDemo\Source".*

5.3.3 Collectors

This section describes how the software of the collector devices operates.

At start up the collector nodes are configured as ZigBee routers. When the user presses **joystick up** on the collector device, it will set its logical type to ZigBee coordinator with the function *zb_WriteConfiguration()* and then do system reset. This is implemented in the function *zb_HandleKeys()* in DemoCollector.c in the handler for HAL_KEY_SW_1.

After reset of the device it reads the configuration and starts up as a ZigBee coordinator. This is done to minimize the user interaction on the other collectors that will automatically start as routers and request to join the network.

When the user presses **joystick right** (HAL_KEY_SW_1) on a collector device it will accept incoming binding requests by calling *zb_AllowBind()*, and the device will go into gateway mode. When in the gateway mode the collector device can communicate with the ZSensorMonitor using the serial interface.

The rest of the collectors will after joining the network send out binding requests automatically, to bind to a gateway. After the collector is successfully bound to a gateway the user can press **joystick down** to start sending reports from the collector. These reports do not contain data, but are merely used to visualize these nodes on the PC user interface (ZSensorMonitor). The rest of the collector devices (i.e. the collectors that are not in gateway mode) will not collect data but function as ZigBee routers for the Sensor Demo i.e. they allow sensors to associate to them and thereby extend the range by routing Zigbee packets in the network.

Pressing **joystick left** on a collector device will make it disallow further association requests. This feature is used in the steps of the CC2530ZDK Quick Start Guide document to extend the network over several hops (see also step 3 in Section 5.2.3).

5.3.4 Sensors

This section describes the operation of the sensor device software, implemented in the file DemoSensor.c.

The sensor devices are configured as ZigBee end devices. When the sensor device starts it will try to join any ZigBee network. This is controlled by the setting ZDAPP_CONFIG_PAN_ID in the file f8wConfig.cfg. If the device shall be configured to join only a specific PAN ID, the define ZDAPP_CONFIG_PAN_ID shall be set to a different value than 0xFFFF; instead use the PAN ID to join.

After successful joining to a network it will start sending binding requests to find a gateway device. This is initiated in the callback function *zb_StartConfirm(),* which is called when the device has joined the network. The function call *osal_set_event()* with the event parameter MY_FIND_COLLECTOR_EVT initiates this binding process.

After successful binding the variable appState is set to APP_REPORT and periodic reporting will start when the user presses **joystick down** (HAL_KEY_SW_3). In this state the sensor will send a report every 2 seconds (using the OSAL timer). The report period is defined by the variable myReportPeriod in DemoSensor.c.

The sensor node will measure its temperature and supply voltage using the onboard ADC and include the measurement in every report. This is done using the functions *readTemp()* and *readVoltage()* (in the file DemoSensor.c).

NB: Since the temperature sensors on the CC2530EM's are not calibrated this sensor application is self calibrating and assumes a temperature of 22º Celcius when the sensor is powered up. If the sensor is powered up at other temperatures the code for the sensor can be modified (DemoSensor.c) to calibrate for another starting temperature. This can be modified in the function *readTemp() in the line:*
temp = 22 + ((adcValue - voltageAtTemp22) / TEMP_COEFFICIENT); by changing the value 22 to a different number, and starting the device at the correct temperature.

The sensor report messages are built and sent in the function *sendReport().* Each 5th report has the acknowledgement request flag set. This is done in order to minimize traffic on the network and to be able to detect when the reports can not reach the gateway successfully. The number of reports before an acknowledgement is requested is set by the define ACK_REQ_INTERVAL.

When a report fails up to the number of times defined by REPORT_FAILURE_LIMIT the device will stop sending reports to the current gateway. It will then go into a state where it deletes the binding to the current gateway and tries to find a new gateway to bind to. This is initiated in the function *zb_SendDataConfirm()*.

5.3.5 Sensor message format

This section describes the format of the messages from the gateway to the PC on the serial interface, and the messages over the air between the devices.

The payload of the over the air sensor report messages have the following format:

1 byte	**1 Byte**	**2 Bytes**
Temperature	Voltage	Parent short address

- Temperature is in signed int8 format defined as temperature in ºC.
- Voltage is the chip supply voltage (in Volts x 10)
- The parent short address field is included to inform the gateway which device is the parent of this sensor. This way the ZSensorMonitor PC tool is able to correctly visualize the node topology in the GUI.

This message is built and sent in the function *sendReport()* in DemoSensor.c.

5.3.6 Serial interface Gateway message formats

The following messages are sent on the serial interface between the gateway[2] and the ZSensorMonitor PC tool:

ZB_RECEIVE_DATA_INDICATION
The ZB_RECEIVE_DATA_INDICATION message is used by the gateway to forward received sensor reports to the PC tool on the serial interface. This message follows the format described in section 3.7.2.5 of the document Z-Stack Monitor and Test Interface [7]. The data field is 4 Bytes long and contains exactly the same fields as the sensor message format of section 5.3.5.

SYS_PING_REQUEST
This message is sent by ZSensorMonitor to the gateway on the serial interface to check proper connection. Refer to the document Z-Stack Monitor and Test Interface [7] for further description of the different fields.

This message has the following format:

SOF **(1)**	**Len** **(1)**	**CMD0** **(1)**	**CMD1** **(1)**	**FCS** **(1)**
0xFE	0	0x00	0x21	--

[2] The code for the gateway functionality is implemented in DemoCollector.c.

SYS_PING_RESPONSE
This message is sent by the gateway to the PC tool as a response to the SYS_PING_REQUEST.

The format of this message is shown below:

SOF (1)	Len (1)	CMD0 (1)	CMD1 (1)	Profile (2)	FCS (1)
0xFE	2	0x61	0x01	0x004x	--

The **Profile** field will have one of the following values depending on which stack profile ZigBee or ZigBee PRO is used in the network:

- ZigBee 2007: 0x0040
- ZigBee PRO 2007: 0x0041

5.3.7 Project configuration options

ZigBee stack profile
The IAR project file for the Sensor Demo application provides configurations for the ZigBee 2007 and the ZigBee 2007 PRO stack profile. Choose the CollectorEB and SensorEB to start the network as ZigBee 2007, and choose CollectorEB-PRO and SensorEB-PRO to start the network as a ZigBee 2007 PRO network.

Bypass Non-Volatile memory
The project files for the Sensor Demo have the NV_INIT and NV_RESTORE compile options enabled. This means that basic NV items and network information (short address, PAN id etc.) is restored from Non-Volatile memory when the device cycles power. Please consult the Z-Stack compile options document for a description of the different compile options [8].

In order to bypass Non-Volatile memory items push **joystick center** (press down like a button) and keep it pressed while powering up the board. **Do not release the joystick until a few seconds after power up**.

Security
Security can be enabled by defining the compile option SECURE=1 in the IAR projects. Please consult the Z-Stack compile options document for a description of the different compile options [8]. Further info on the use of security can be found in the Z-Stack Developer’s Guide [5].

Network bandwidth (traffic rate)
When using the application in a larger network the default values for report interval, end device poll interval etc., may cause the application to use a high amount of bandwidth in the network. There are several parameters in Z-Stack and in the Sensor Demo application that can be adjusted to control the amount of traffic in the network.

By default the sensor and collector nodes will send a report every 2 seconds. In order to decrease traffic rate this interval can be increased. The report interval is controlled with the variable *myReportPeriod* in the files DemoSensor.c and DemoCollector for the sensor and collector application respectively.

The end device poll rate is by default in Z-Stack set to 1 second. This parameter is controlled in the file f8wConfig.cfg by setting the -DPOLL_RATE to another value. Increasing this interval will cause less traffic in the network, at the expense of longer response times for packets going to an end device.

6 Appendix A: Programming the boards

This section describes two alternatives on how to program the boards. The boards can either be programmed directly from IAR EW. Alternatively if the hex files already exist, the TI Flash Programmer tool can be used. In either case the CC2530EM will have to be mounted on top of a SmartRF05EB which is connected to the PC with a USB connection.

Please ensure that you have installed the necessary software as described in Chapter 4.

6.1 Program the boards with IAR

The following section describes the steps needed to build the projects and download hex files to the hardware using IAR EW 8051.

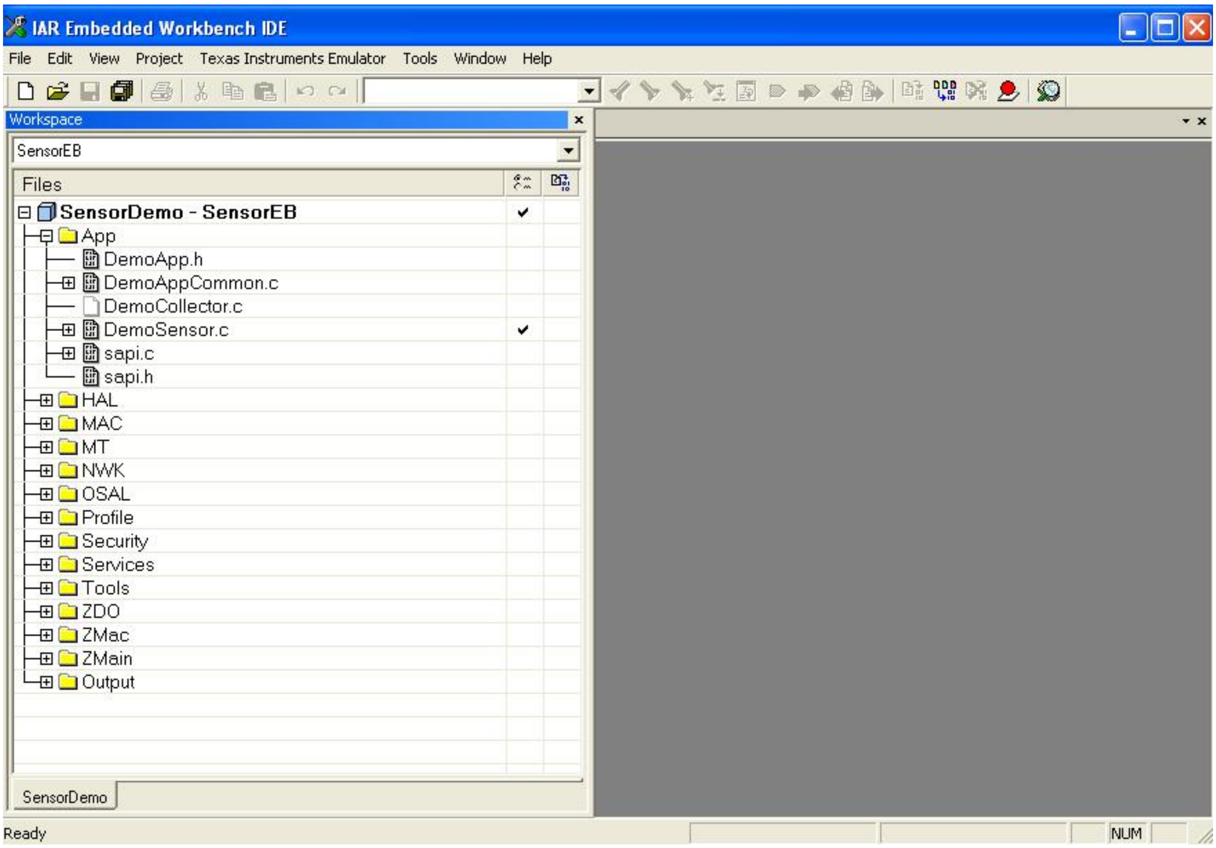


Figure 5 SensorDemo workspace in IAR

1. Open IAR Embedded Workbench
2. Open the workspace file **SensorDemo.eww** with IAR. This file is found in the subfolder *\Projects\zstack\Samples\SensorDemo\CC2530DB* in the Z-Stack installation folders.
3. There are four different configurations in the IAR project; CollectorEB, SensorEB, CollectorEB-PRO and SensorEB-PRO.

   CollectorEB and SensorEB configurations are using the ZigBee 2007 stack profile, while CollectorEB-PRO and SensorEB-PRO are using ZigBee 2007 PRO.

   Select the CollectorEB or CollectorEB-PRO configuration to program at least one node as Collector.

   (The software built with these configurations can also run on the SmartRF05BB)

4. Select **Project->Rebuild All**. This will perform a full rebuild on the selected project.

5. Attach the CC2530EM board to program to a SmartRF05EB.
6. Connect the SmartRF05EB to the PC with a USB cable.
7. Select **Project->Debug** to program the CC2530.
8. Stop the debugger by selecting **Debug -> Stop Debugging**.
9. Repeat steps 3 to 5 to program additional nodes. Make sure at least one node is programmed as collector and at least one as sensor. Also make sure all nodes are using the same stack profile either ZigBee 2007 or ZigBee 2007 PRO.

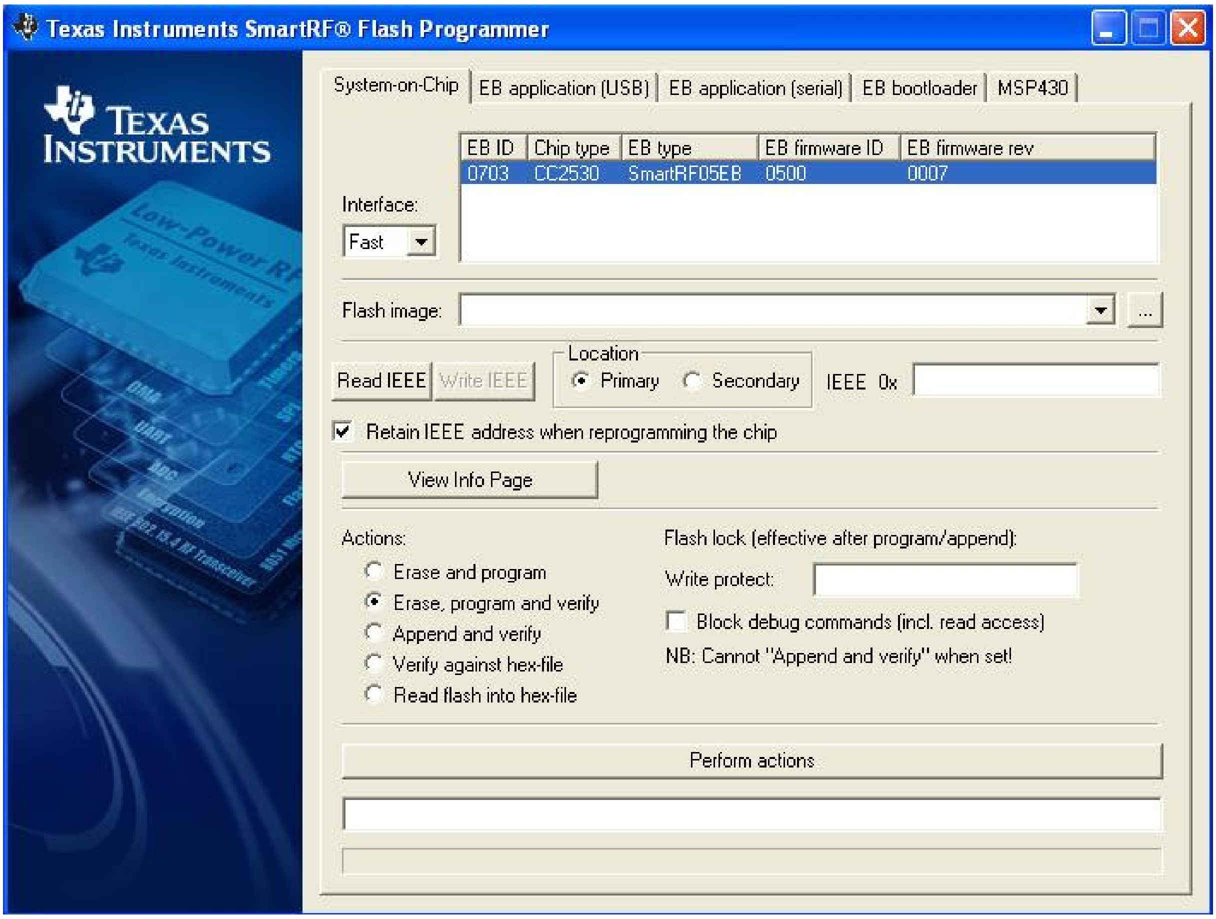
6.2 Alternative: Download hex files with the Flash Programmer

It is also possible to program the boards with the Texas Instruments Flash Programmer if you already have built hex files (either built yourself using the IAR EW as described above or as part of the software package; copied in Step 3 in Chapter 4).
Simply follow these steps:

1. Download and install the Texas Instruments Flash Programmer. This program can be downloaded from the CC2530DK website [6] or directly from http://focus.ti.com/docs/toolsw/folders/print/flash-programmer.html.
2. Attach the CC2530EM board to program to the SmartRF05EB.
3. Connect the SmartRF05EB to the PC with a USB cable.

Open the Texas Instruments Flash Programmer, and choose the System-on-Chip tab. The connected device is shown in the list as shown in

4. Figure 6.
5. In the Flash image field browse to the correct hex file. The hex files for the Sensor Demo application is found in the *Exe* folders for each of the configurations e.g. for the SensorEB-PRO configuration it is found in the folder *SensorDemo\CC2530DB\SensorEB-PRO\Exe* in the Sensor Demo source package.
6. Make sure the ‘Erase, program and verify’ action is checked. Also make sure that ‘Retain IEEE address when programming the chip’ is checked to keep its stored IEEE address.
7. Push the ‘Perform actions’ button to program the device.
8. Make sure at least one node is programmed as collector and at least one as sensor. Also make sure all nodes are using the same stack profile either ZigBee 2007 or ZigBee 2007 PRO.



EB ID	Chip type	EB type	EB firmware ID	EB firmware rev
0703	CC2530	SmartRF05EB	0500	0007

Interface: Fast

Flash image:

Location: ● Primary ○ Secondary

IEEE 0x

☑ Retain IEEE address when reprogramming the chip

Actions:

- ○ Erase and program
- ● Erase, program and verify
- ○ Append and verify
- ○ Verify against hex-file
- ○ Read flash into hex-file

Flash lock (effective after program/append):

Write protect:

☐ Block debug commands (incl. read access)

NB: Cannot "Append and verify" when set!

Figure 6 Flash programmer

References

[1] **Z-Stack product folder:**

http://www.ti.com/z-stack

[2] **ZSensorMonitor User's Guide:**

This document is found in the installation folder after installing ZSensorMonitor

[3] **CC2530ZDK Quick Start Guide:**

http://www.ti.com/lit/swra274

[4] **CC2530ZDK User's Guide:**

http://www.ti.com/lit/swru208a

[5] **Z-Stack Developer's Guide**

This document is found in the *Documents* folder of the Z-Stack installation.

[6] **CC2530ZDK website:**

http://focus.ti.com/docs/toolsw/folders/print/cc2530zdk.html

[7] **Z-Stack Monitor and Test Interface.pdf**

This document is found in the *Documents* folder of the Z-Stack installation.

[8] **Z-Stack Compile Options.pdf**

This document is found in the *Documents* folder of the Z-Stack installation.

Document History

Revision	Date	Description/Changes
A	2010.05.14	Updated to support Z-stack 2.3.0
-	2009.05.25	Initial version