Getting Started with SimpliciTI™ and the eZ430

Mike Claassen

m-claassen@ti.com

Abstract

The first half of this module gives an overview of the components of the SimpliciTI network stack and their interaction in a SimpliciTI network.

The second half of this module is hands on and covers setting up a SimpliciTI network and the utilization of the Smart RF Studio Tool to set up the radio component of the stack.

The full IAR Studio Tool and an eZ430-RF2500 board are required for this module.

Agenda: Introducing SimpliciTI™

- SimpliciTI™ Overview
- Topology Examples
- Stack Architecture and API Details
- eZ430-RF2500 Data Hub Example
- Device Objects Overview
- Build Time Configuration Options

Benefits of this Training

- Gain a working knowledge of low bandwidth RF communication topologies and capabilities
- Evaluate the multiple capabilities of the SimpliciTI™ protocol stack
- Be able to design and demonstrate a functional RF network

SimpliciTI™ Overview

The basics behind the stack

What is SimpliciTI™?

- Low Power: **Supports sleeping devices for low power consumption**
- Low Cost: **Uses < 8K FLASH and < 1K RAM depending on platform**
- Flexible: **Simple star w/ extender and/or p2p communication**
- Simple: **Utilizes a very basic core 6 instruction API**
- Versatile: **Currently ported to the MSP430+CC1100/2500, CC111x/251x, CC2520 and CC2430/31**

SimpliciTI targets quick time-to-market wireless solutions for low power, low cost, and low data rate networks without the need to know the details of the network support.

Basic Network Topology

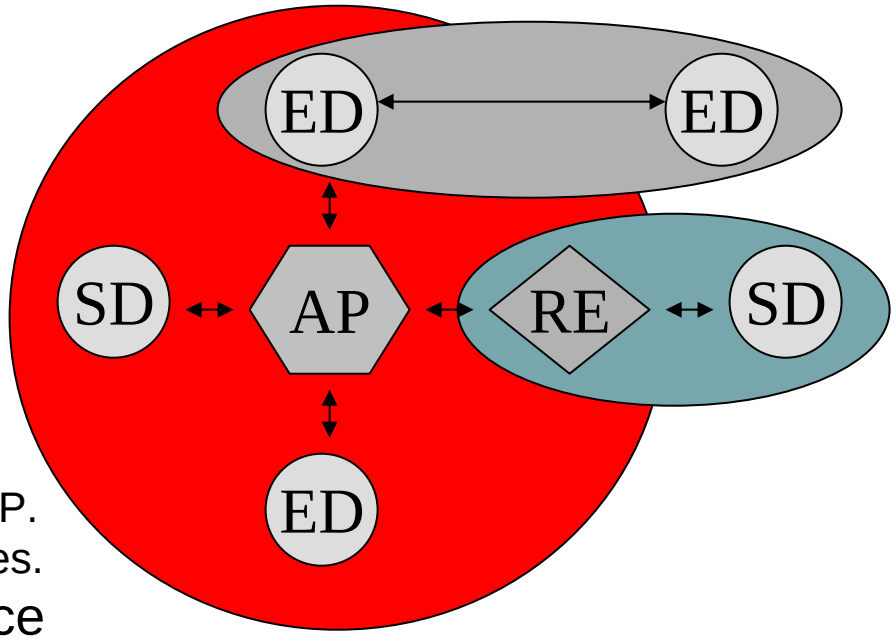
Device Configurations:

AP – Access Point
- •Allows Access to the Network
- •Stores and Forward Messages
- •Serves as a Range Extender

RE – Range Extender
- •Repeats message Traffic
- •Like the AP, Device is always on

ED – End Device
- •No Store & Forward Services from AP.
- •Can sleep, but can't poll for messages.

SD – Sleeping (polling) End Device
- •Requires Store and Forward Services from the AP



Topologies:

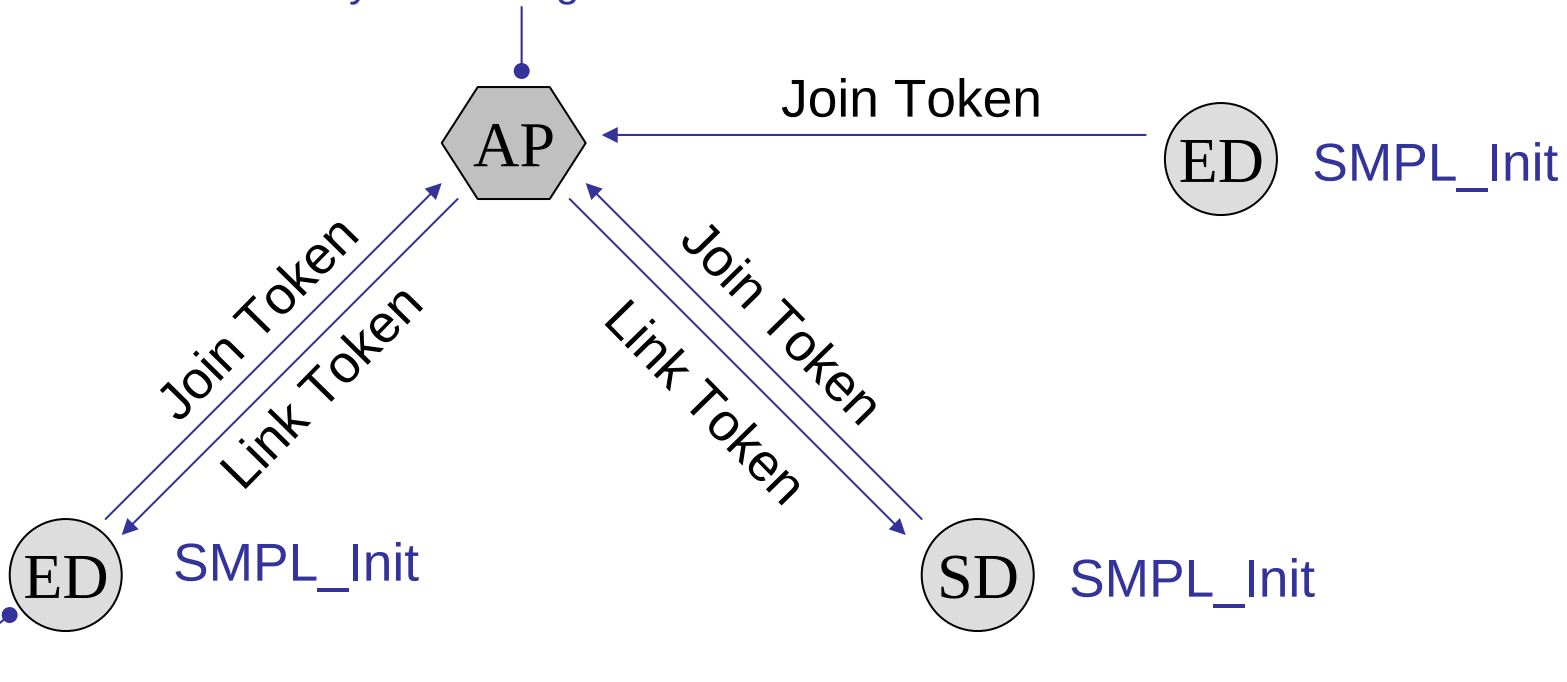
- Access Point Star
- Access Point Star w/ Range Extender
- Peer to Peer

SimpliciTI™ Overview

Topology Examples

AP Managed: Peer to Peer Topology

1. Access Point must be powered on and initialized before the rest of the network.
3. If the Join Token matches, the AP will return a Link Token
6. The Access Point will establish Store and Forward data storage for the polling End Devices to ensure delivery of messages intended for them.

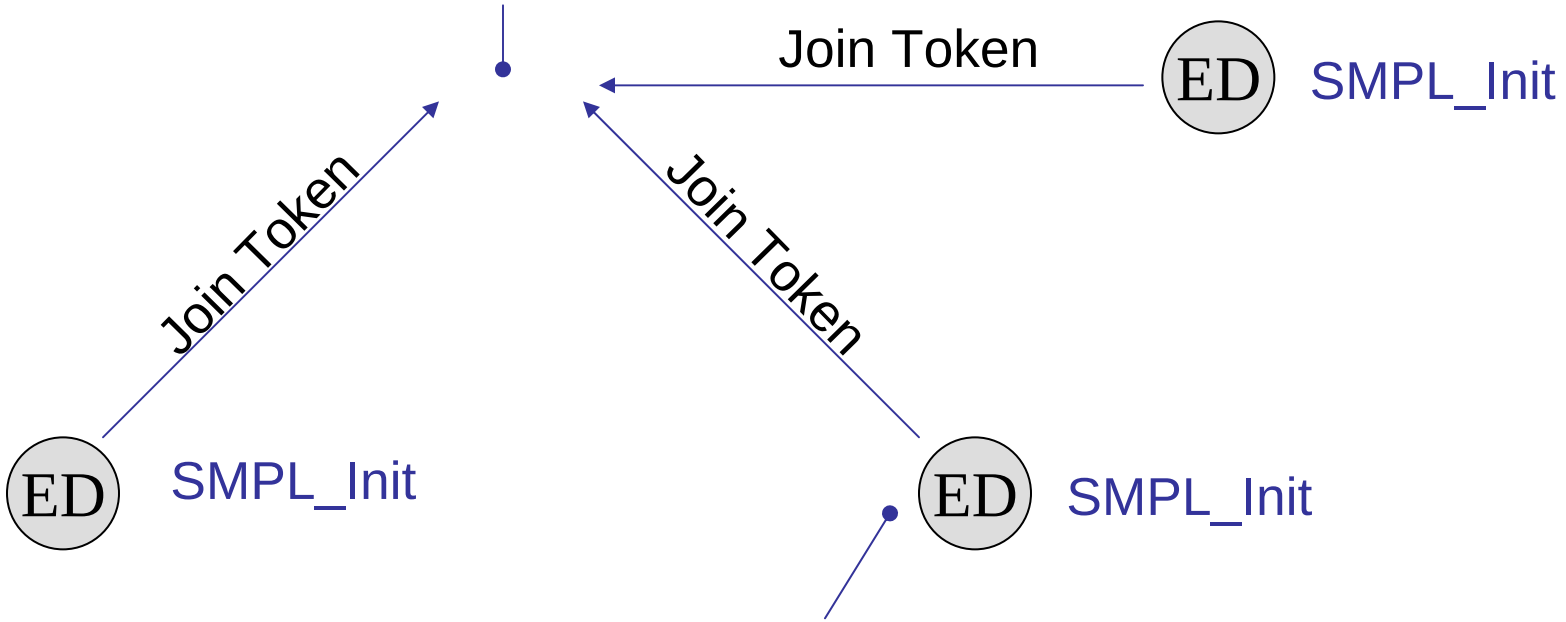


2. End Devices upon initialization, send a Join Token request to the network
4. The End Device overwrites its default Link Token w/ the new Link Token received from the AP

5. Sleeping End Devices follow the same process.
6. The AP is using information in the Join frame to determine this is a polling device.

No AP: Peer to Peer Topology

No Access Point exists, so the Join Token request falls on deaf ears. The SMPL_Init returns nothing, the ED's time out and use their default link token. A smpl_Status_t code is returned to the ED that the join request failed.

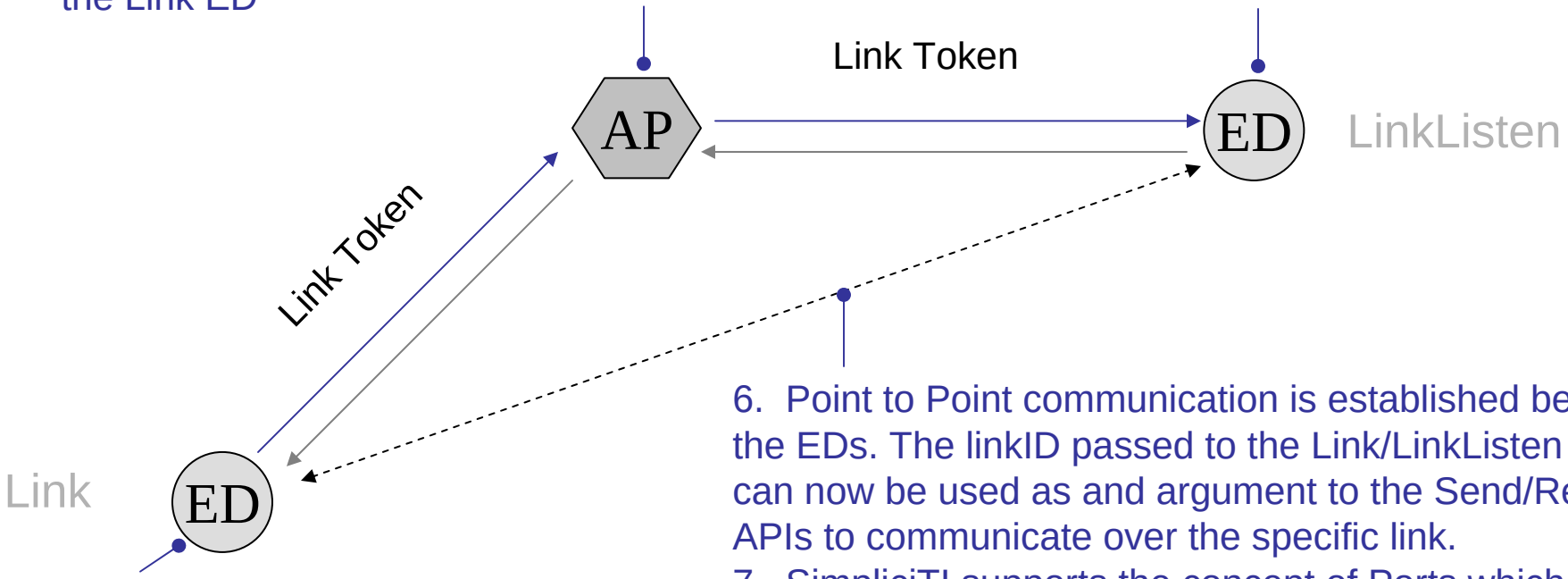


Since no Store and Forward Service exists, the polling End Devices cannot sleep as they may miss any asynchronous messages sent to them. An AP to required enable polling End Device functionality.

End Device Point to Point links

3. Serving the function of a range extender, the AP will repeat the message to extend the radio range of the Link ED
5. The Message is relayed back by the AP to the Link ED

1. An ED will accept invitations to Link with the LinkListen command
4. If the Link Token matches, the LinkListen ED will reply back with it's corresponding Link information



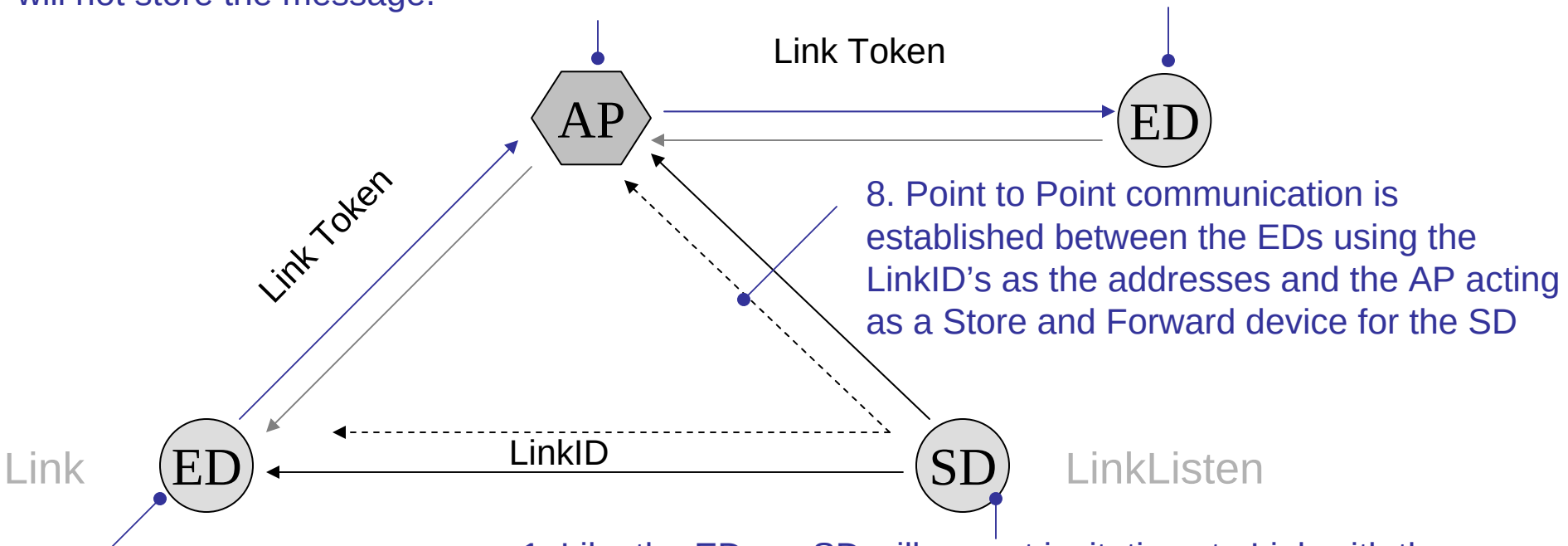
2. To connect to an ED in LinkListen mode, an ED will issue a Link command that will contain its Link Token

6. Point to Point communication is established between the EDs. The linkID passed to the Link/LinkListen APIs can now be used as and argument to the Send/Receive APIs to communicate over the specific link.
7. SimpliciTI supports the concept of Ports which will allow multiple connections between the same EDs

Sleeping Device Point to Point links

3. The AP now serves as a Store and Forward device for application level messages as well as a repeater. It will repeat the Link request but will not store the message.

4. Even though this ED receives the Link message from the AP, it is not in LinkListen mode, so it will ignore the message.



2. As before, to connect to an SD in LinkListen mode, the ED will issue a Link command that will send its Link Token to the network.

1. Like the ED, an SD will accept invitations to Link with the LinkListen command

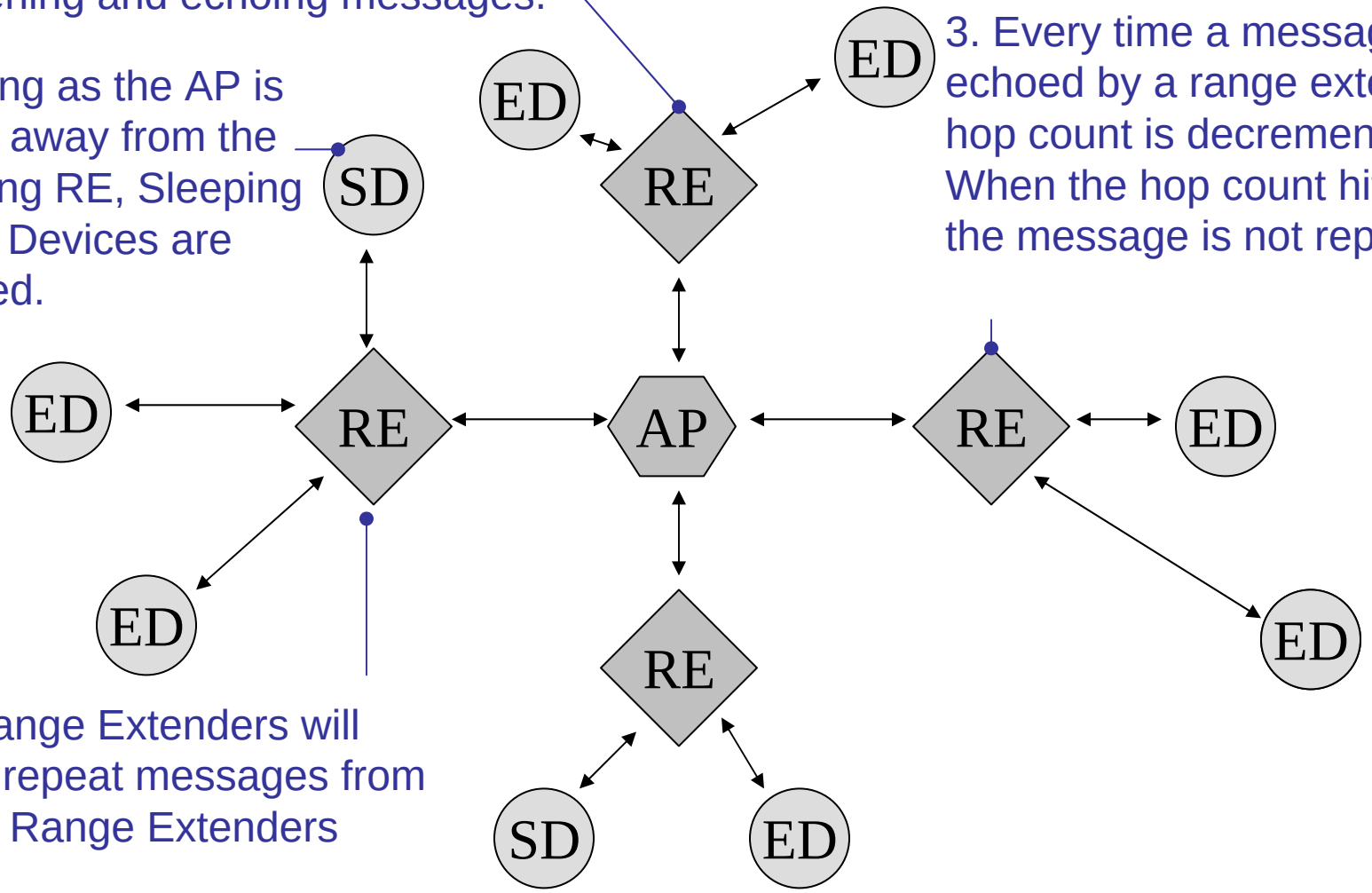
5. Upon receiving the Link message repeated by the AP (assuming Link ED is out of range) & accepting it, the SD will return a link reply message and go to sleep. It will then wake up and poll the AP for application level messages sent to its links.

Adding a Range Extender

1. The Range Extender extends the network by listening and echoing messages.

3. Every time a message is echoed by a range extender a hop count is decremented. When the hop count hits 0, the message is not repeated

5. As long as the AP is one hop away from the supporting RE, Sleeping (polling) Devices are supported.

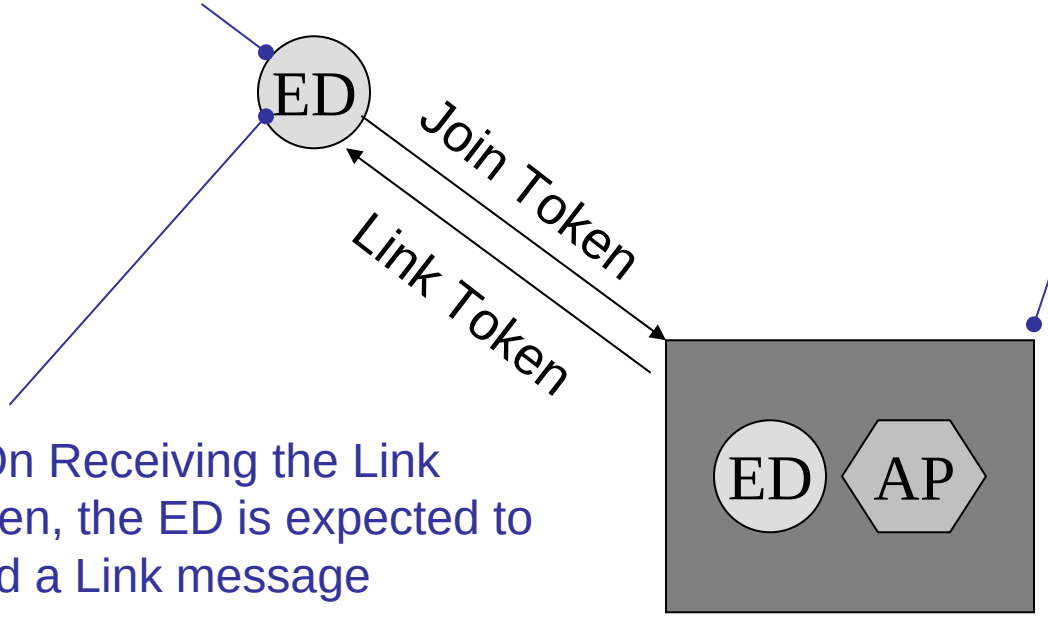


2. Range Extenders will NOT repeat messages from other Range Extenders

Star Topology using a Data Hub

2. On initialization, the ED sends a Join Token to the network

8. Sleeping End Devices can also be supported as the AP still performs Store and Forward.



1.The Data Hub is an AP with ED functionality

3. The AP part of the Data Hub returns the Link Token

4. The Data Hub ED goes into a LinkListen Mode expecting a LinkID to be immediately sent

5. On Receiving the Link Token, the ED is expected to send a Link message

6. Number of connections supported is defined by the config files.

7. Point to Point communication between the ED's can still be done.

SimpliciTI™ Overview

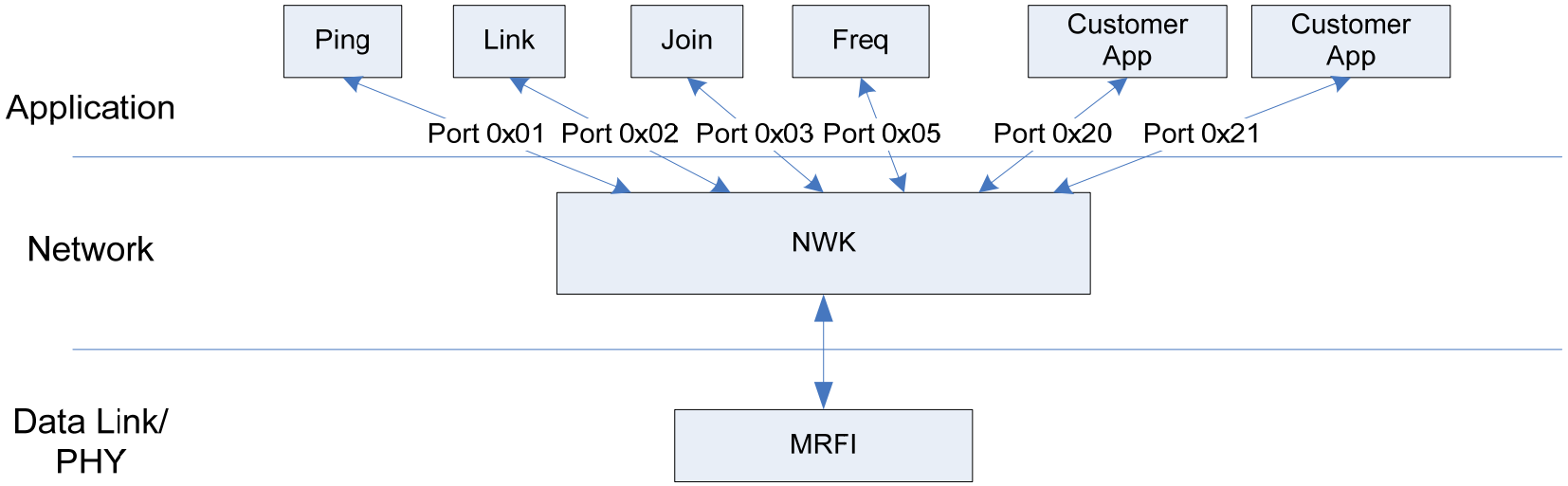
Stack Architecture and API Details

SimpliciTI™ API

- **Initialization**
  - smplStatus_t SMPL_Init(uint8_t (*callback)(linkID_t));
- **Linking (bi-directional by default)**
  - smplStatus_t SMPL_Link(linkID_t *linkID);
  - smplStatus_t SMPL_LinkListen(linkID_t *linkID);
- **Peer-to-peer messaging**
  - smplStatus_t SMPL_Send(lid, *msg, len);
  - smplStatus_t SMPL_Receive(lid, *msg, *len);
- **Configuration**
  - smplStatus_t SMPL_Ioctl(object, action, *val);

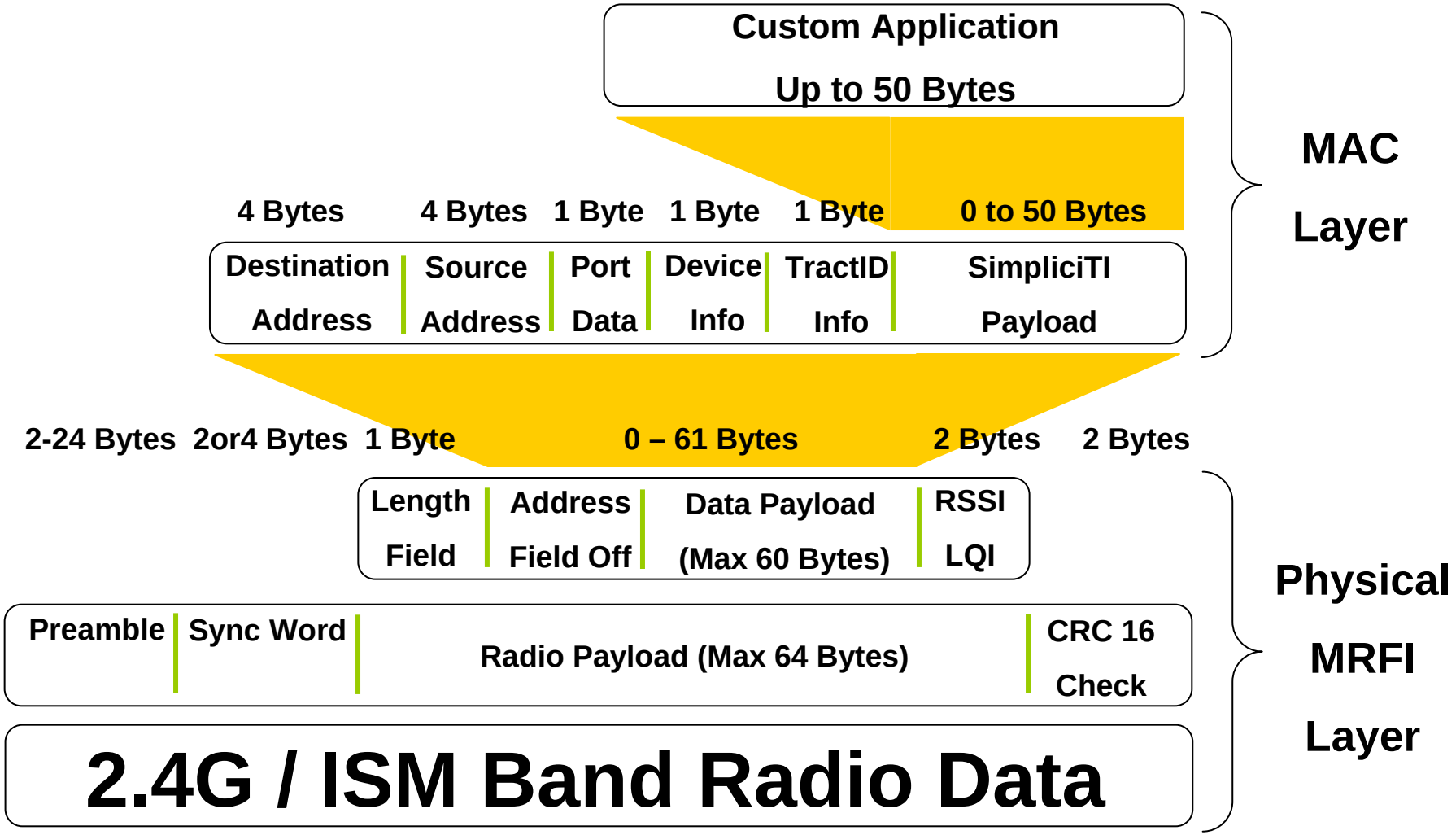
Architectural Overview

- Network Support
  - Init (Join Token/Link Token exchange)
  - Ping (Debug Only)
  - link / linklisten (Establish peer-to-peer connections)
  - nwk mgmt (General nwk mgmt, poll port)
  - send / receive
  - I/O
- Layers
  - MRFI
  - NWK
  - nwk applications (Ports < 0x20)
  - customer applications (Ports ≥ 0x20)



- SimpliciTI Address
  - HW addr (4 byte) + Port
  - Statically assigned HW addr
  - Ports allow for more than one link between two devices

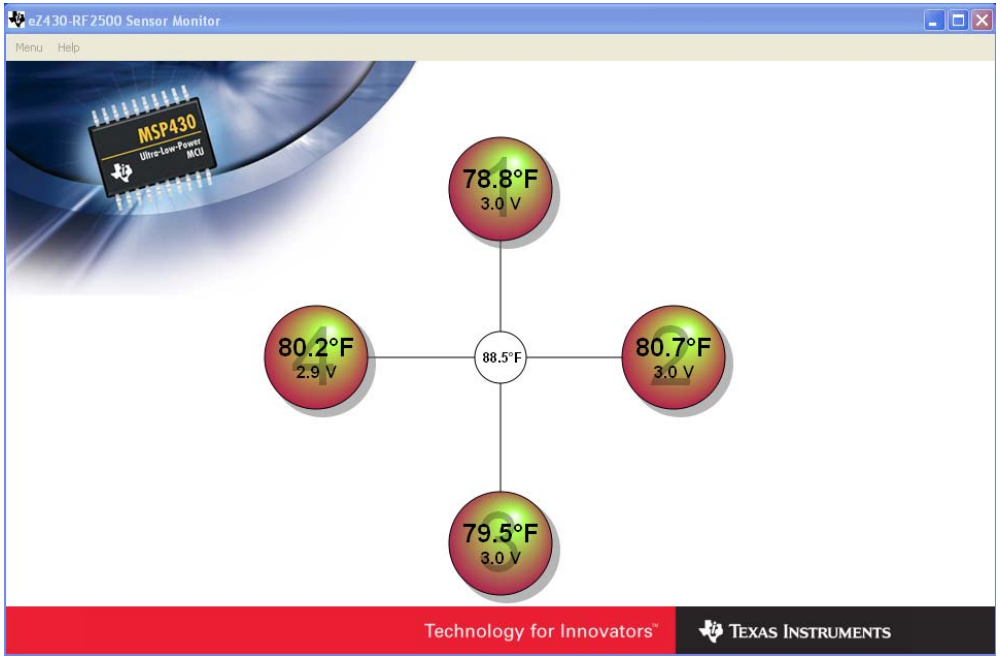
SimpliciTI™ – CC2500/CC1100

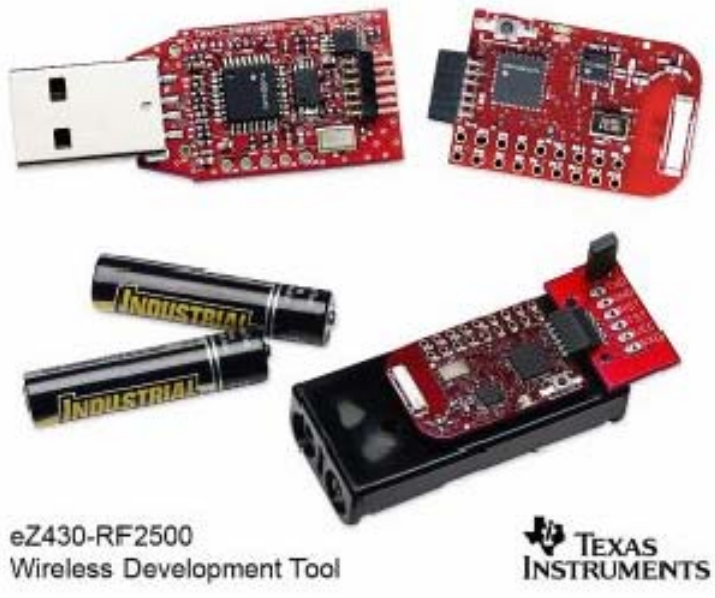


SimpliciTI™ Device Objects Overview

eZ430-RF2500 Data Hub Example

Temperature Monitor Demo





P.nbr.	Time (us)	Length	Dest. Address	Source Address	Port	Device Info	Transaction ID	Applicaton payload	Unknown Port	RSSI (dBm)	LQI	FCS
					Encryption Number	Rec.Type Send.Type HCount						
1141	+332509 =538690411	16	0x865B93D4	0x9E5CA29F	NO 0x20	ALWAYS_LISTEN END_DEVICE 03	0x55	04 01 1E	0x20	-53	11	OK
1142	+224373 =538914784	16	0x865B93D4	0x5CE2BCD6	NO 0x20	ALWAYS_LISTEN END_DEVICE 03	0x77	17 01 1E	0x20	-61	11	OK
1143	+123656 =539038440	16	0x865B93D4	0x138B4378	NO 0x20	ALWAYS_LISTEN END_DEVICE 03	0xF9	08 01 1D	0x20	-67	13	OK
1144	+136375 =539174815	16	0x865B93D4	0xC71F648A	NO 0x20	ALWAYS_LISTEN END_DEVICE 03	0xCC	04 01 1E	0x20	-69	17	OK

Getting Started

1. Download IAR Kickstart Tool

A. http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html
B. Loads the Driver for the eZ430 USB UART

2. Download eZ430-RF2500 Sensor Monitor Demo

A. http://www.ti.com/litv/zip/slac139b
B. Loads the Sensor Monitor Application

3. Download SmartRF Studio

A. http://focus.ti.com/docs/toolsw/folders/print/smartrftm-studio.html
B. Can be used to modify the Radio settings

4. Install IAR Kickstart Tool

5. Plug in eZ430-RF2400 Toolstick in the USB port

A. USB Human Interface Device Installed
B. USB Communications Port Installed
C. C:\Program Files\IAR Systems\Embedded Workbench 5.0\430\drivers\TIUSBFET\WinXP

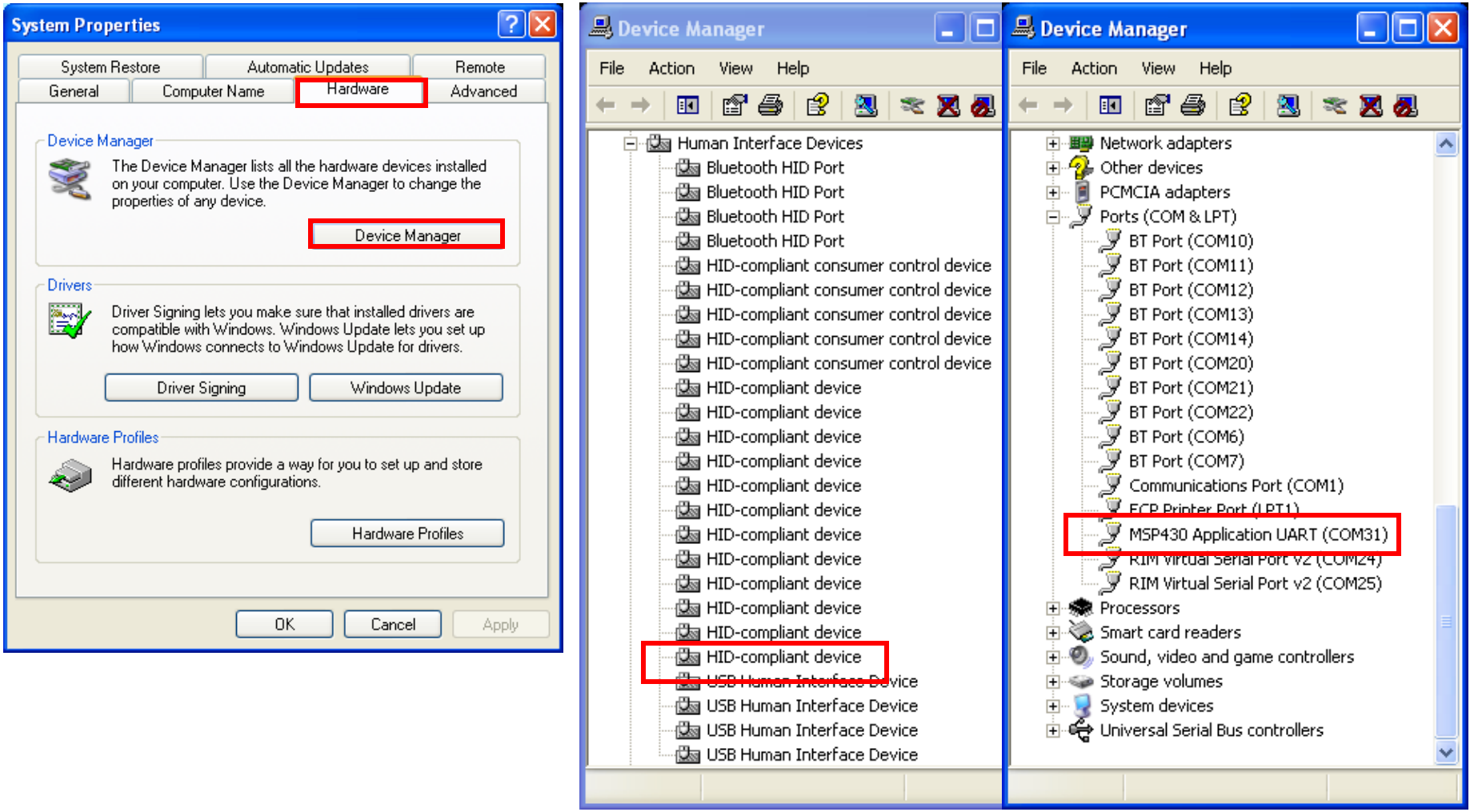
6. Install the eZ430-RF2500 Sensor Monitor Demo

http://www.ti.com/litv/zip/slac139b

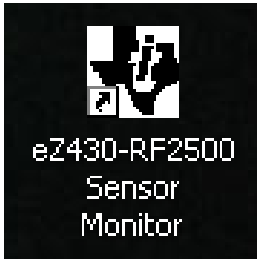
7. Install the Smart RF Studio Tool

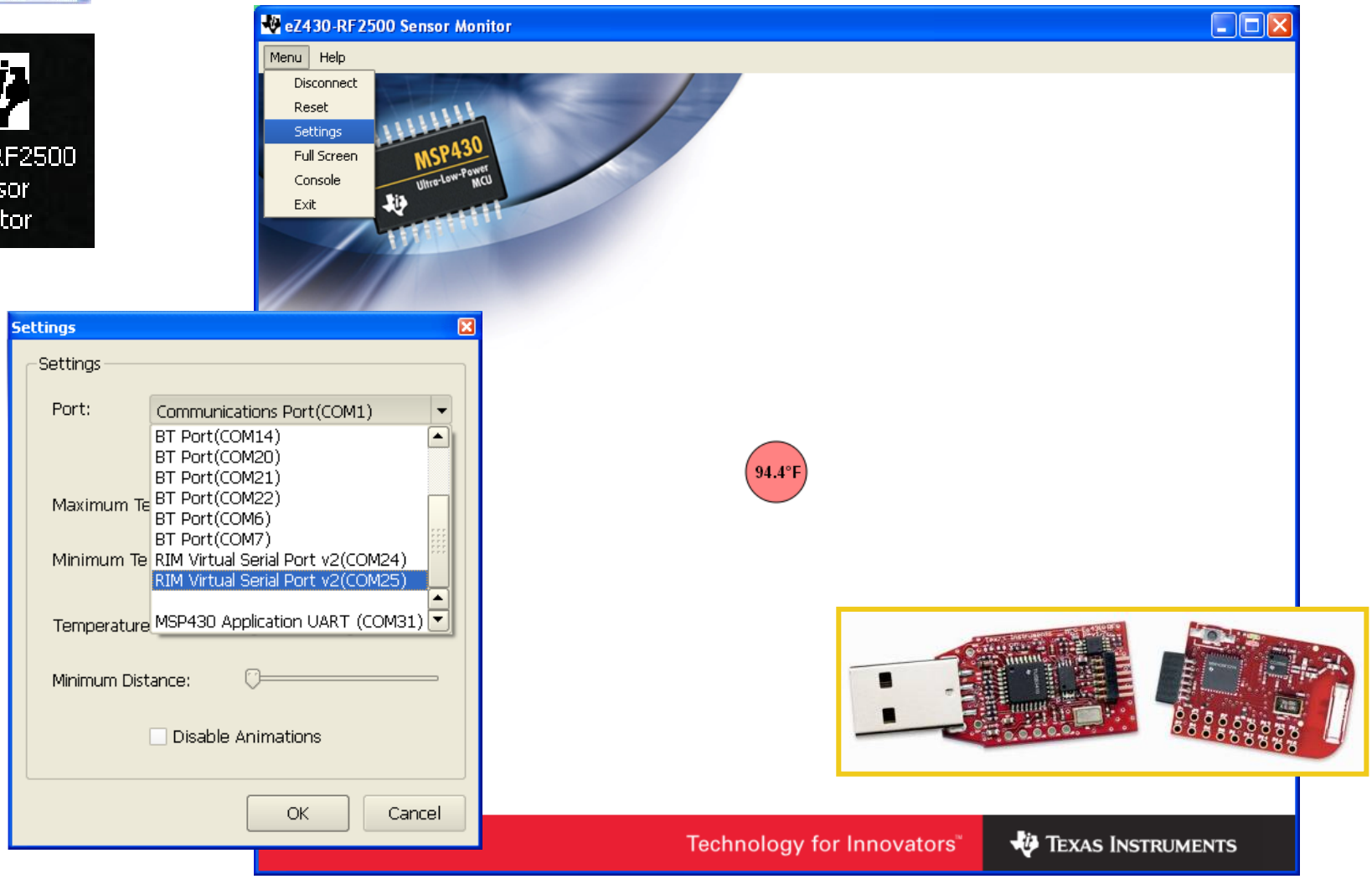
http://www.ti.com/litv/zip/swrc046k

USB Driver Locations

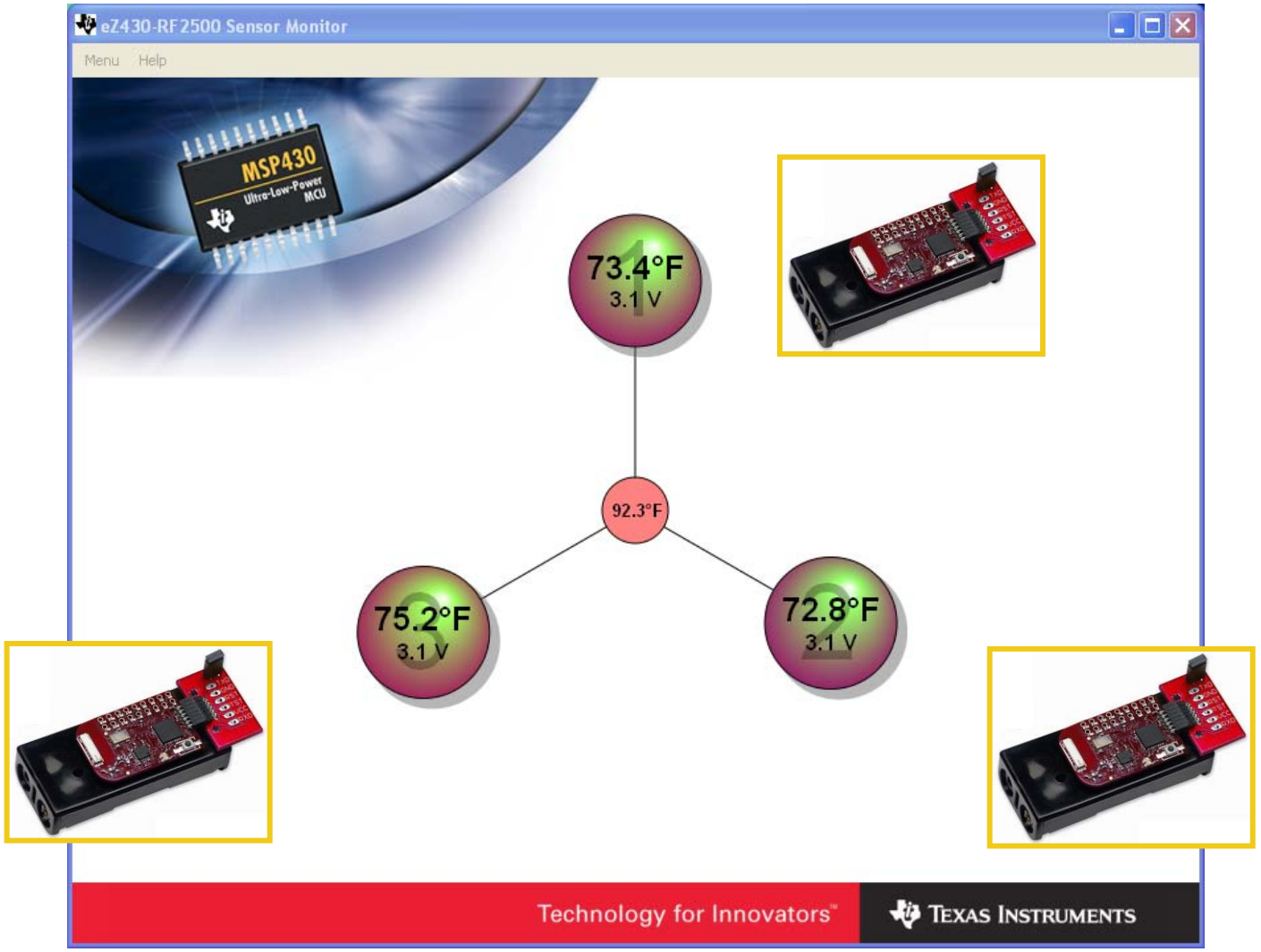


eZ430-RF2500 Sensor Monitor

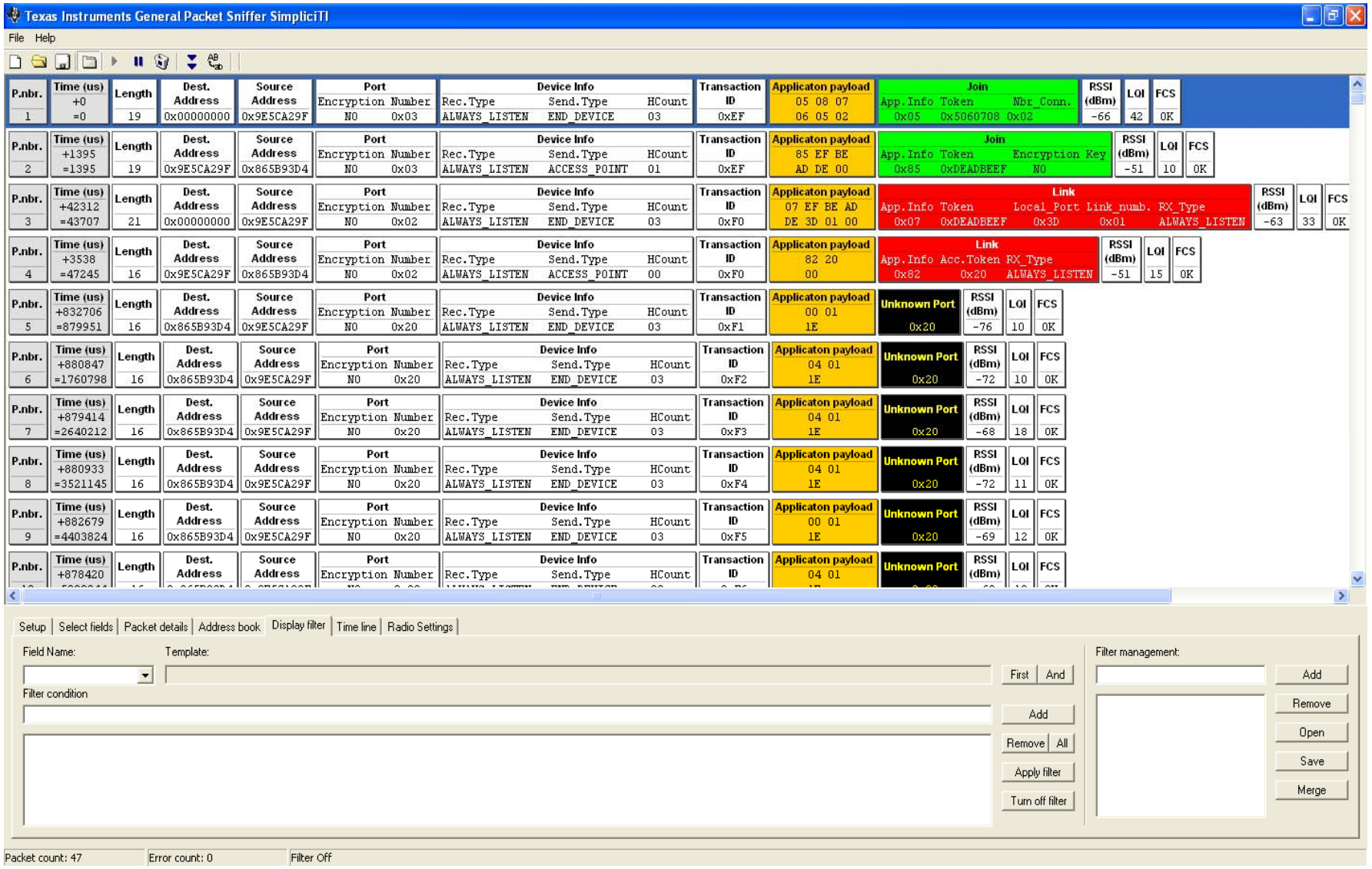




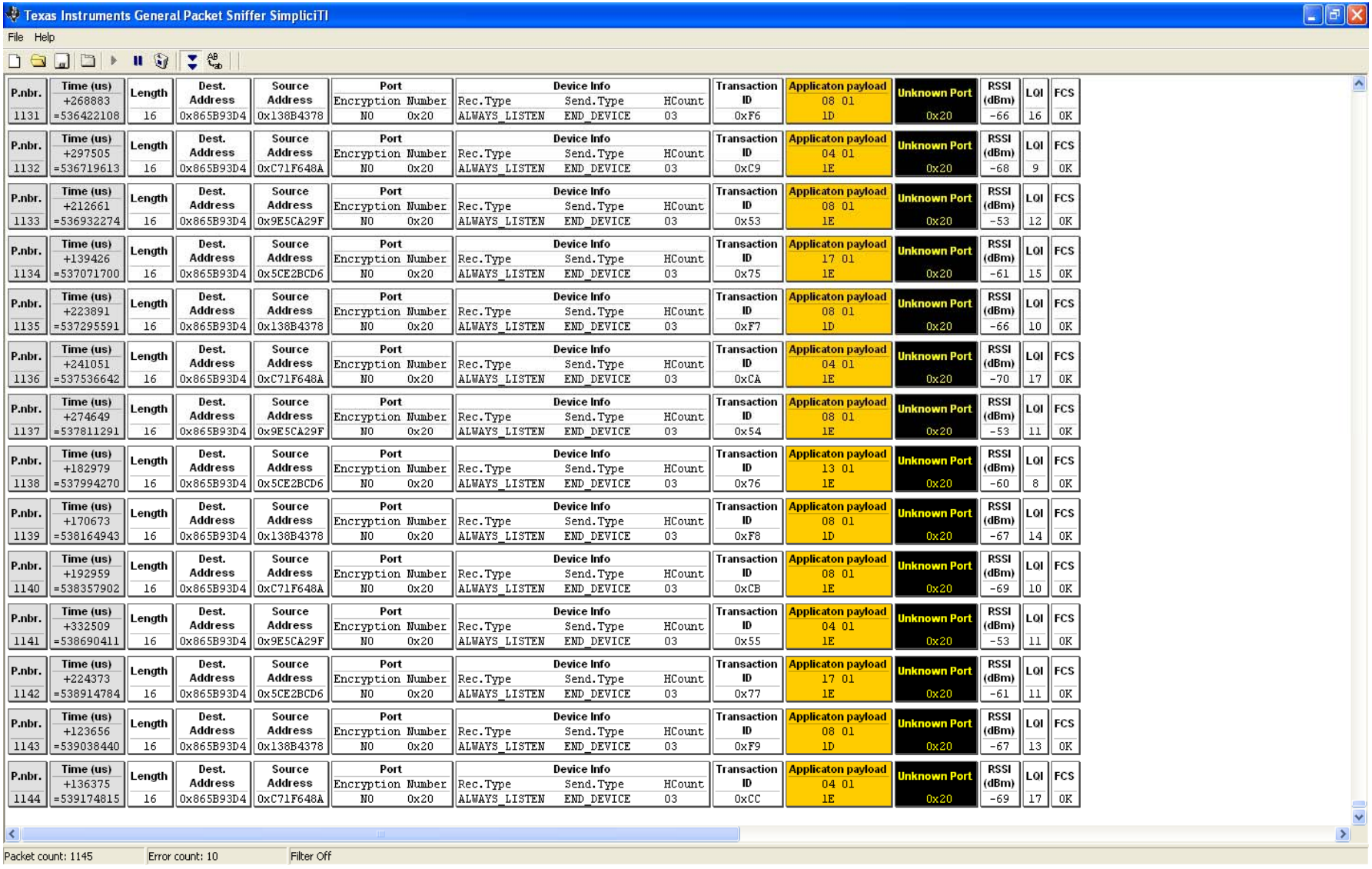
Power On Battery RF2500



View of Join Token Transaction



View of End Device Interaction



P.nbr.	Time (us)	Length	Dest. Address	Source Address	Port		Device Info			Transaction ID	Applicaton payload	Unknown Port	RSSI (dBm)	LQI	FCS
					Encryption	Number	Rec.Type	Send.Type	HCount						
1131	+268883 =536422108	16	0x865B93D4	0x138B4378	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF6	08 01 1D	0x20	-66	16	OK
1132	+297505 =536719613	16	0x865B93D4	0xC71F648A	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xC9	04 01 1E	0x20	-68	9	OK
1133	+212661 =536932274	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x53	08 01 1E	0x20	-53	12	OK
1134	+139426 =537071700	16	0x865B93D4	0x5CE2BCD6	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x75	17 01 1E	0x20	-61	15	OK
1135	+223891 =537295591	16	0x865B93D4	0x138B4378	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF7	08 01 1D	0x20	-66	10	OK
1136	+241051 =537536642	16	0x865B93D4	0xC71F648A	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCA	04 01 1E	0x20	-70	17	OK
1137	+274649 =537811291	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x54	08 01 1E	0x20	-53	11	OK
1138	+182979 =537994270	16	0x865B93D4	0x5CE2BCD6	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x76	13 01 1E	0x20	-60	8	OK
1139	+170673 =538164943	16	0x865B93D4	0x138B4378	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF8	08 01 1D	0x20	-67	14	OK
1140	+192959 =538357902	16	0x865B93D4	0xC71F648A	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCB	08 01 1E	0x20	-69	10	OK
1141	+332509 =538690411	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x55	04 01 1E	0x20	-53	11	OK
1142	+224373 =538914784	16	0x865B93D4	0x5CE2BCD6	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x77	17 01 1E	0x20	-61	11	OK
1143	+123656 =539038440	16	0x865B93D4	0x138B4378	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF9	08 01 1D	0x20	-67	13	OK
1144	+136375 =539174815	16	0x865B93D4	0xC71F648A	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCC	04 01 1E	0x20	-69	17	OK

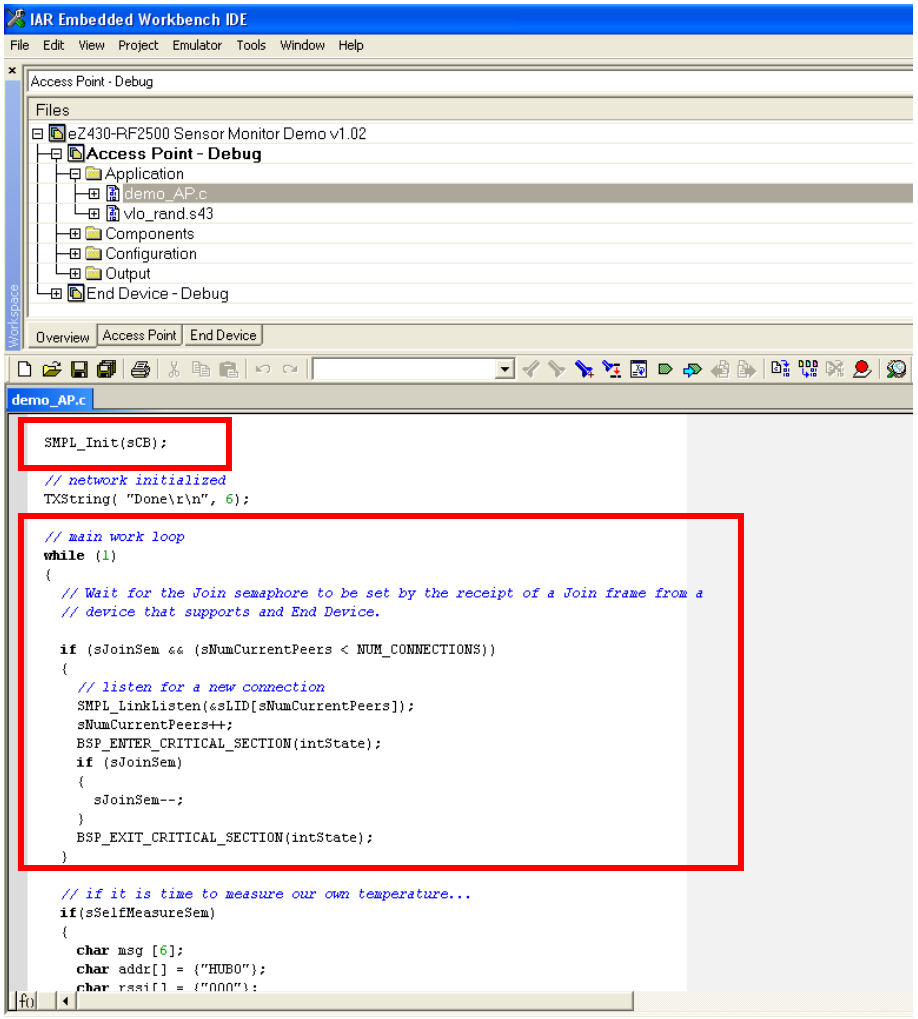
Packet count: 1145 | Error count: 10 | Filter Off

Device Objects - Access Point

- MSP430 Initialized
- Board initialized
- SMPL_Init
  - Set up as an Access Point
  - Callback utilized – Sets Semaphore

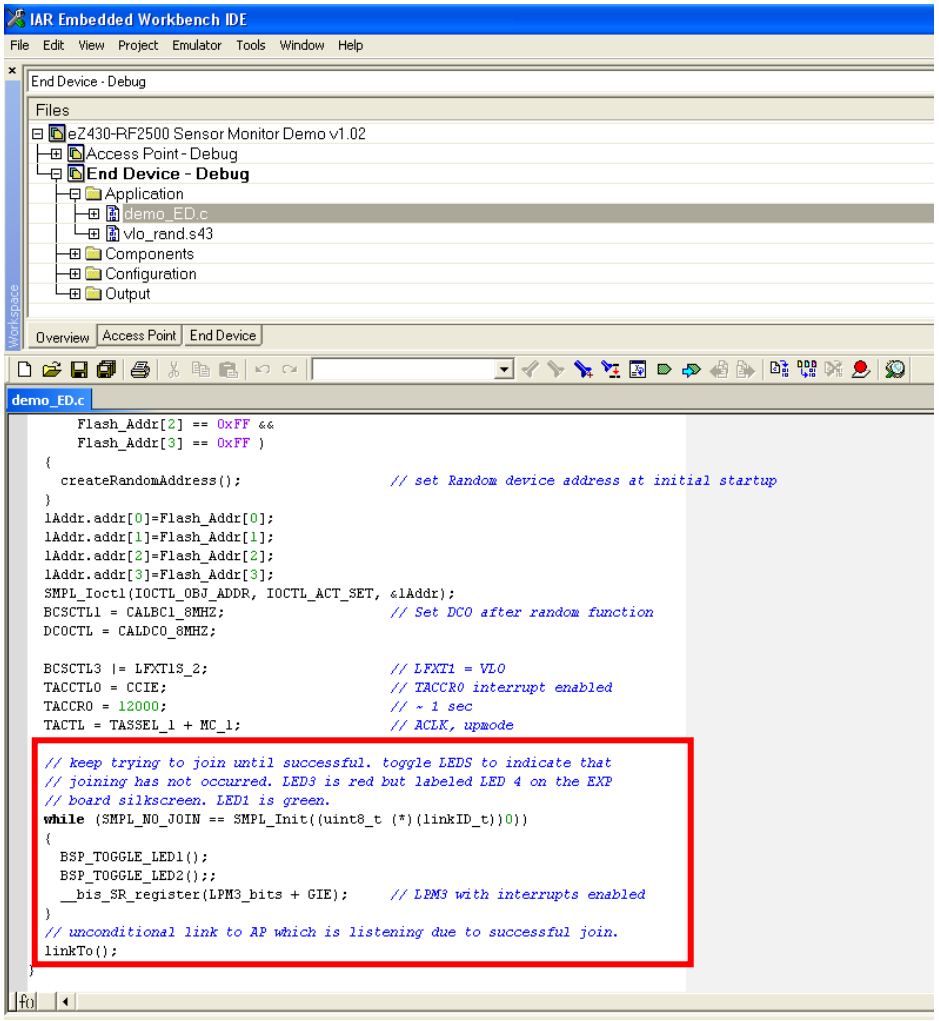
```
/*-------------------------------------------------------------------------------
* Runs in ISR context. Reading the frame should be done in the
* application thread not in the ISR thread.
-------------------------------------------------------------------------------*/
static uint8_t sCB(linkID_t lid)
{
  if (lid)
  {
    sPeerFrameSem++;
  }
  else
  {
    sJoinSem++;
  }
  // leave frame to be read by application.
  return 0;
}
```

- LinkListen invoked
  - Until # of Connections are met
  - Allows additional EDs to Join
  - BSP_ENTER_CRITICAL_SECTION
    - Context Saves SimpliciTI Int State
    - Disables Interrupts
  - BSP_ENTER_CRITICAL_SECTION
    - Context Restores SimpliciTI Int State
    - Enables Interrupts



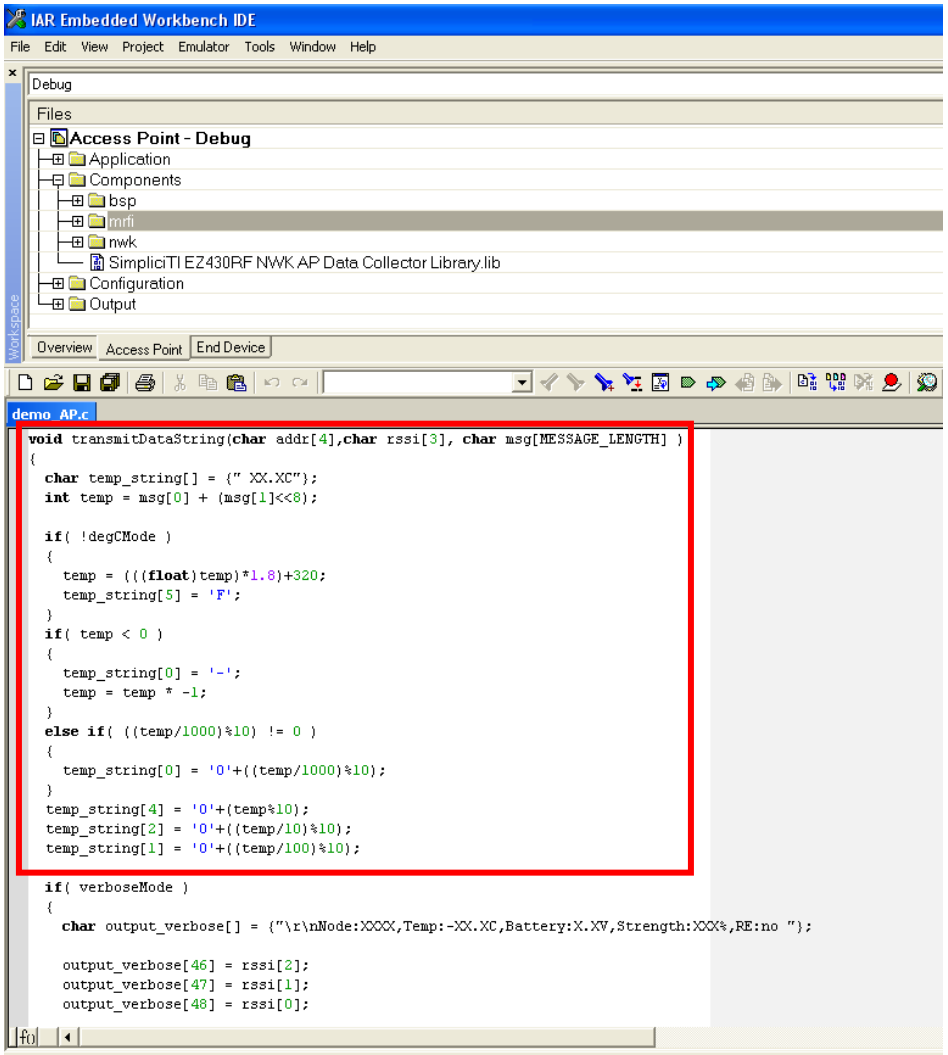
Device Objects - End Device

- MSP430 Initialized
- Board initialized
- SMPL_Init
  - Set up as End Device
  - Option to Sleep
  - No Call Back
  - Join Token Sent, & repeat if no reply
- Link process Initiated
  - Assumes LinkListen ED from AP
- ADC Sampled
  - Twice per Second (Temp & Vcc)
- SMPL_Send
  - Two Byte Temperature
  - One Byte Vcc
- MSP430 Functionality
  - Controls Radio On/Off
  - ADC Interrupt on Conversion

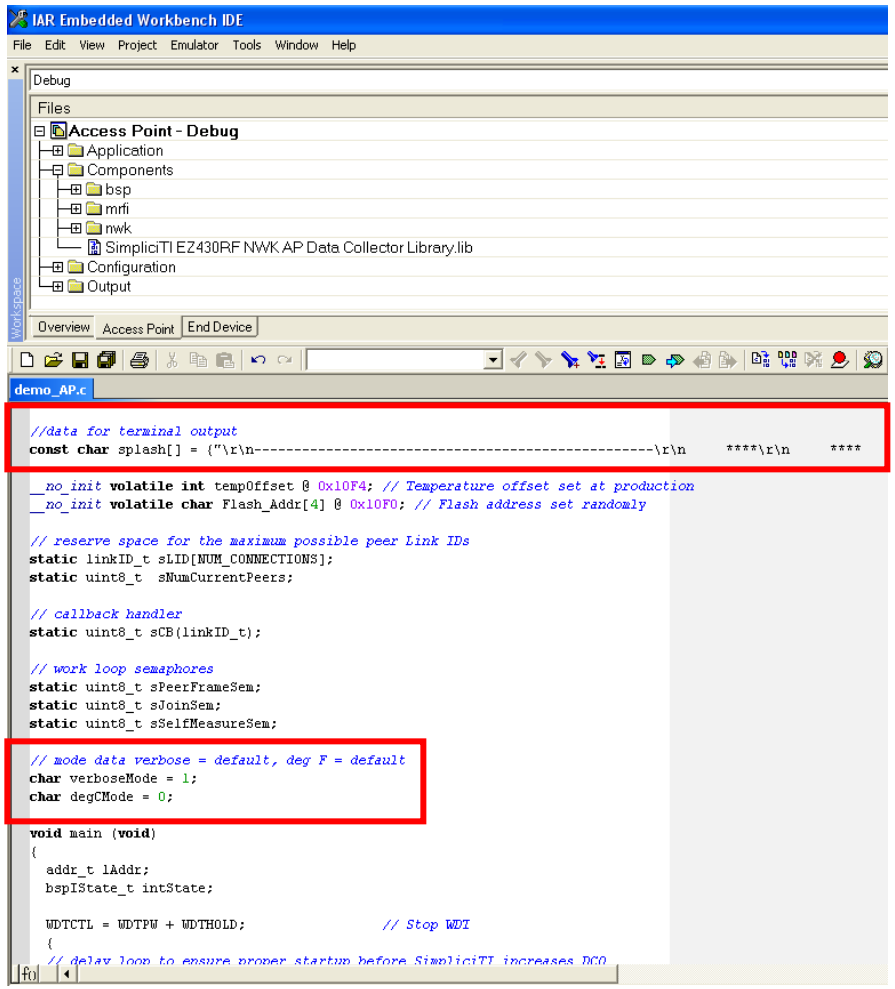


Device Objects - Access Point

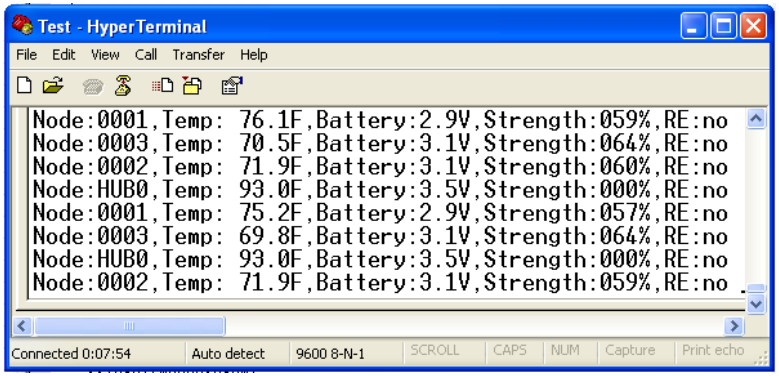
- MSP430 Functionality
  - ADC Interrupt on Conversion
    - Local Temperature
    - Vcc Conversion
  - USB Backchannel UART
    - Local Temperature
      - Address Hub
    - Remote Temp and Vcc
      - LinkID Address
    - Two Modes
      - eZ430-RF2500 Sensor Monitor
      - UART Terminal Program
- SimpliciTI Configuration
  - Star hub in the network
    - 1 AP per net (Join Token)
  - Always-on
  - Configured as a Data Hub
    - AP + ED



UART Terminal Output Types

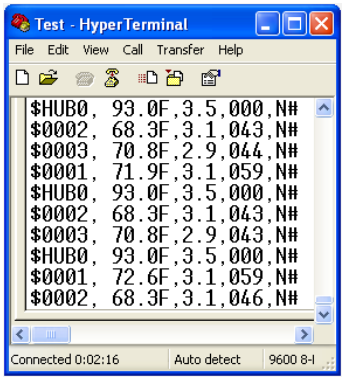



```
----------------------------------------------------
     ****
     ****              eZ430-RF2500
     ******o****       Temperature Sensor Network
********_///_****      Copyright 2007
 ******/_//_/*****     Texas Instruments Incorporated
  ** ***(__/*****      All rights reserved.
      **********       Version 1.02
       ******
        ***
----------------------------------------------------
```



verboseMode = 1

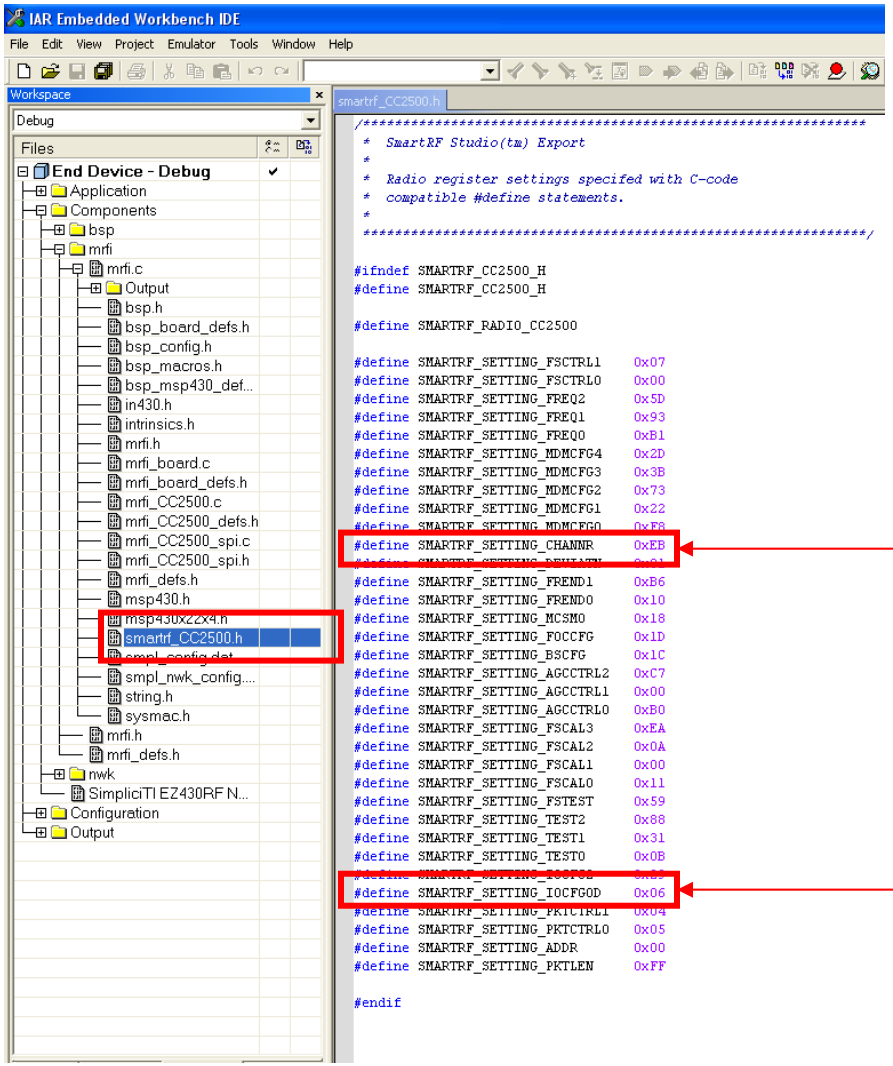
verboseMode = 0 →



SimpliciTI™ Overview

Radio Settings and RF Studio

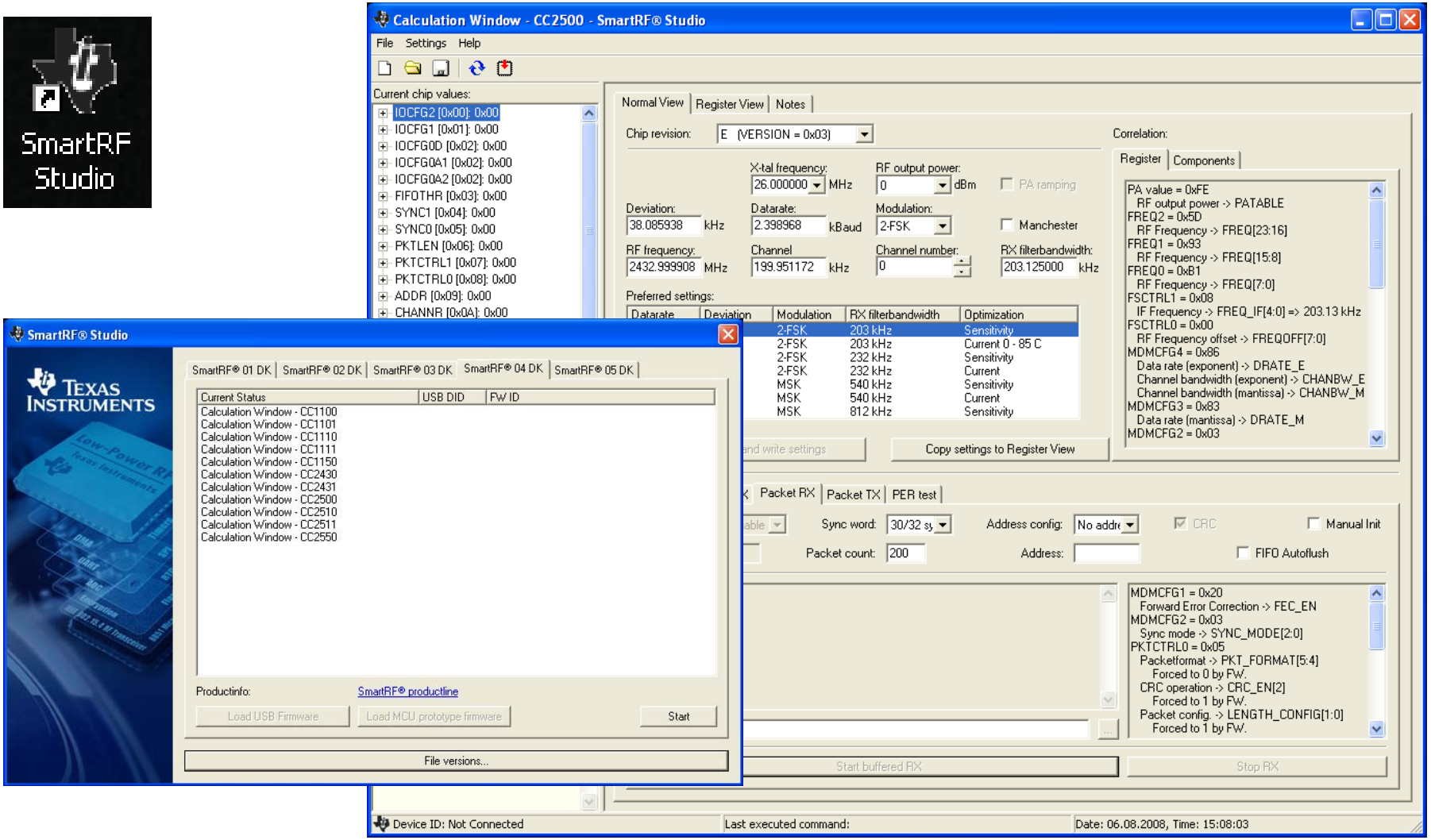
SimpliciTI™ Radio Settings



CHANNR – Sets the Channel

IOCFG0D – Sets the ISR Pin

SmartRF Studio

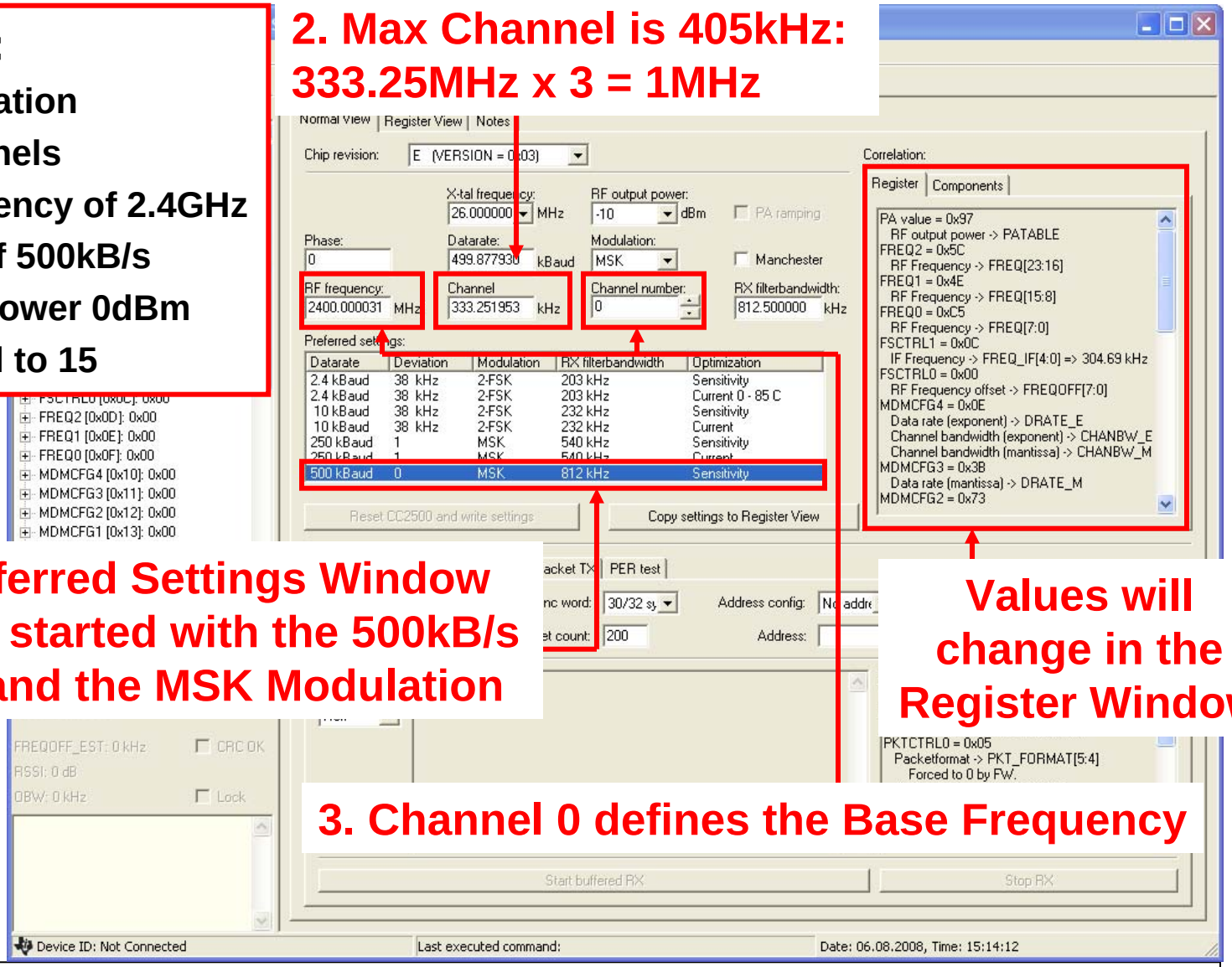


Example

Set Radio To:

1. MSK Modulation
2. 1MHz Channels
3. Base Frequency of 2.4GHz
4. Data Rate of 500kB/s
5. RF output power 0dBm
6. Set Channel to 15

2. Max Channel is 405kHz: 333.25MHz x 3 = 1MHz



1. The Preferred Settings Window can get you started with the 500kB/s Data Rate and the MSK Modulation

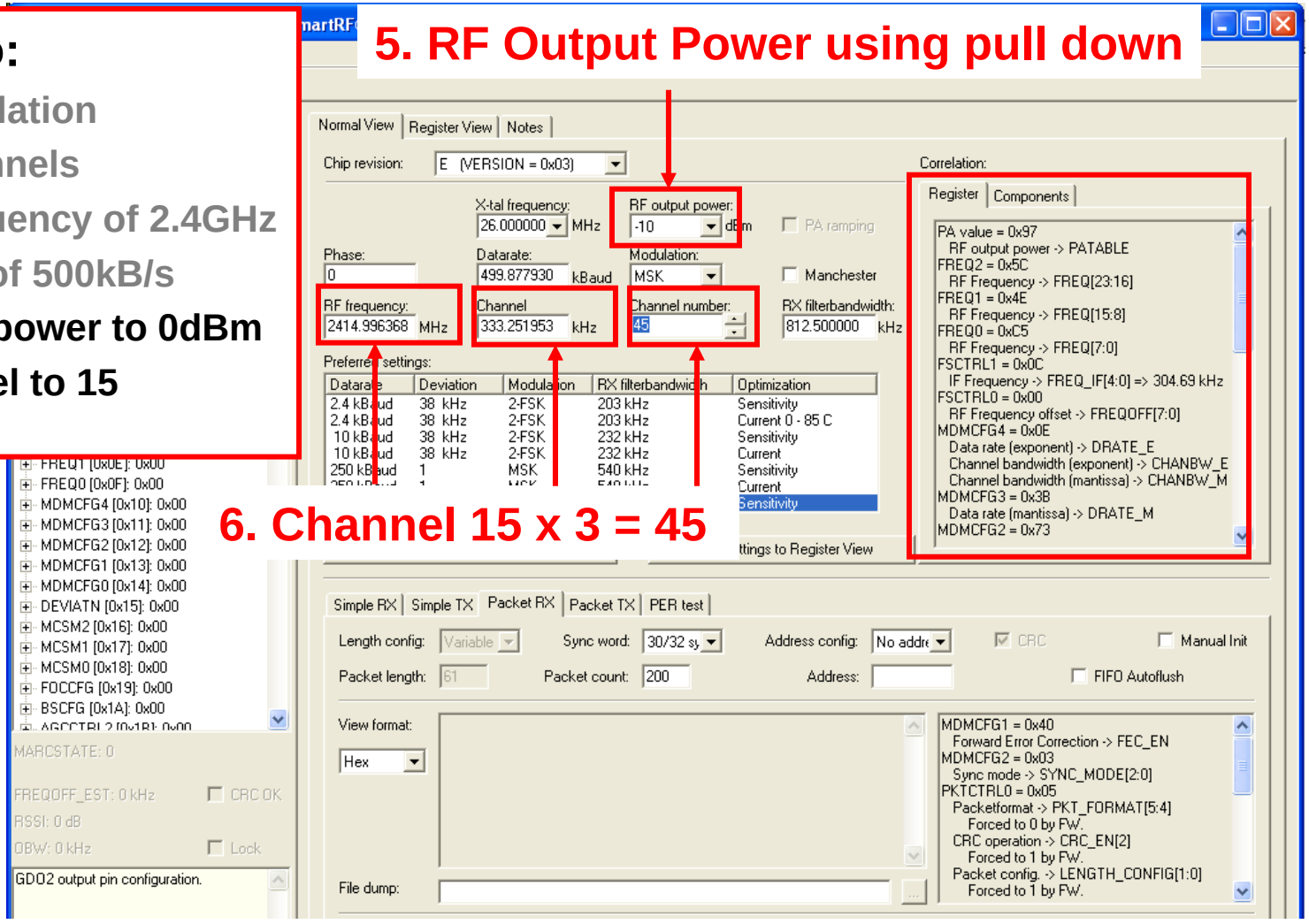
Values will change in the Register Window

3. Channel 0 defines the Base Frequency

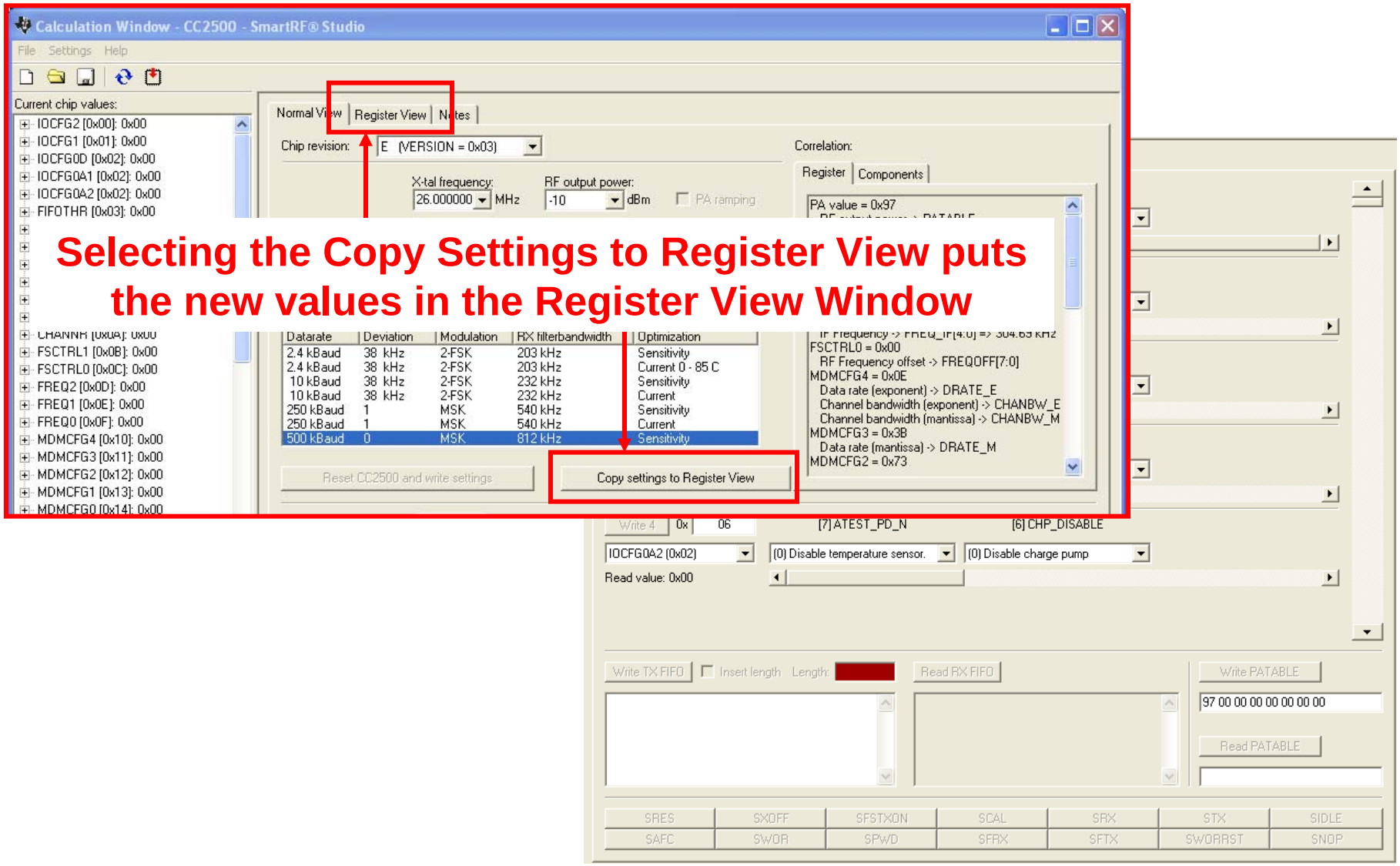
Example

Set Radio To:

1. MSK Modulation
2. 1MHz Channels
3. Base Frequency of 2.4GHz
4. Data Rate of 500kB/s
5. **RF output power to 0dBm**
6. **Set Channel to 15**

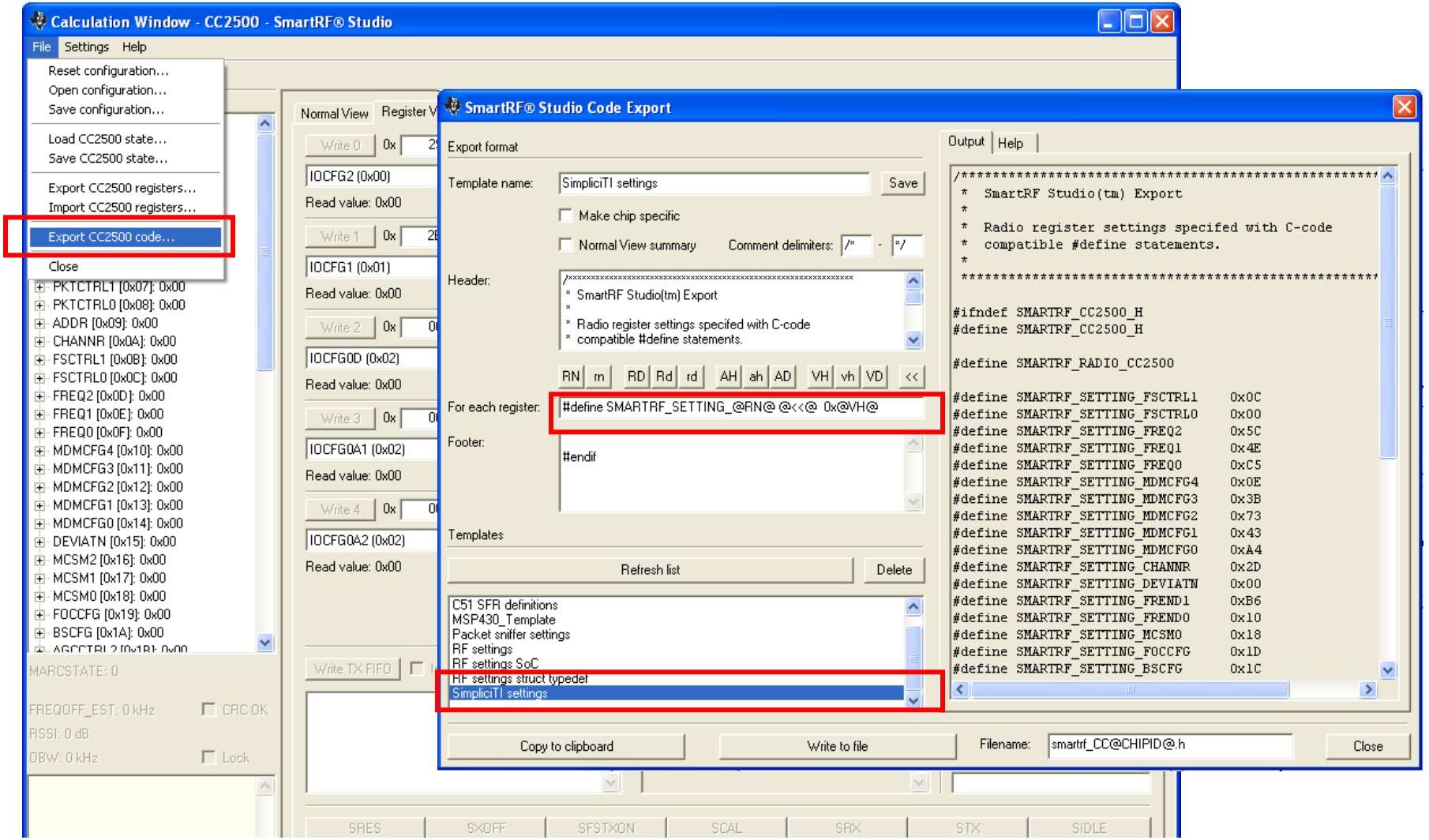


To Review Actual Register Settings

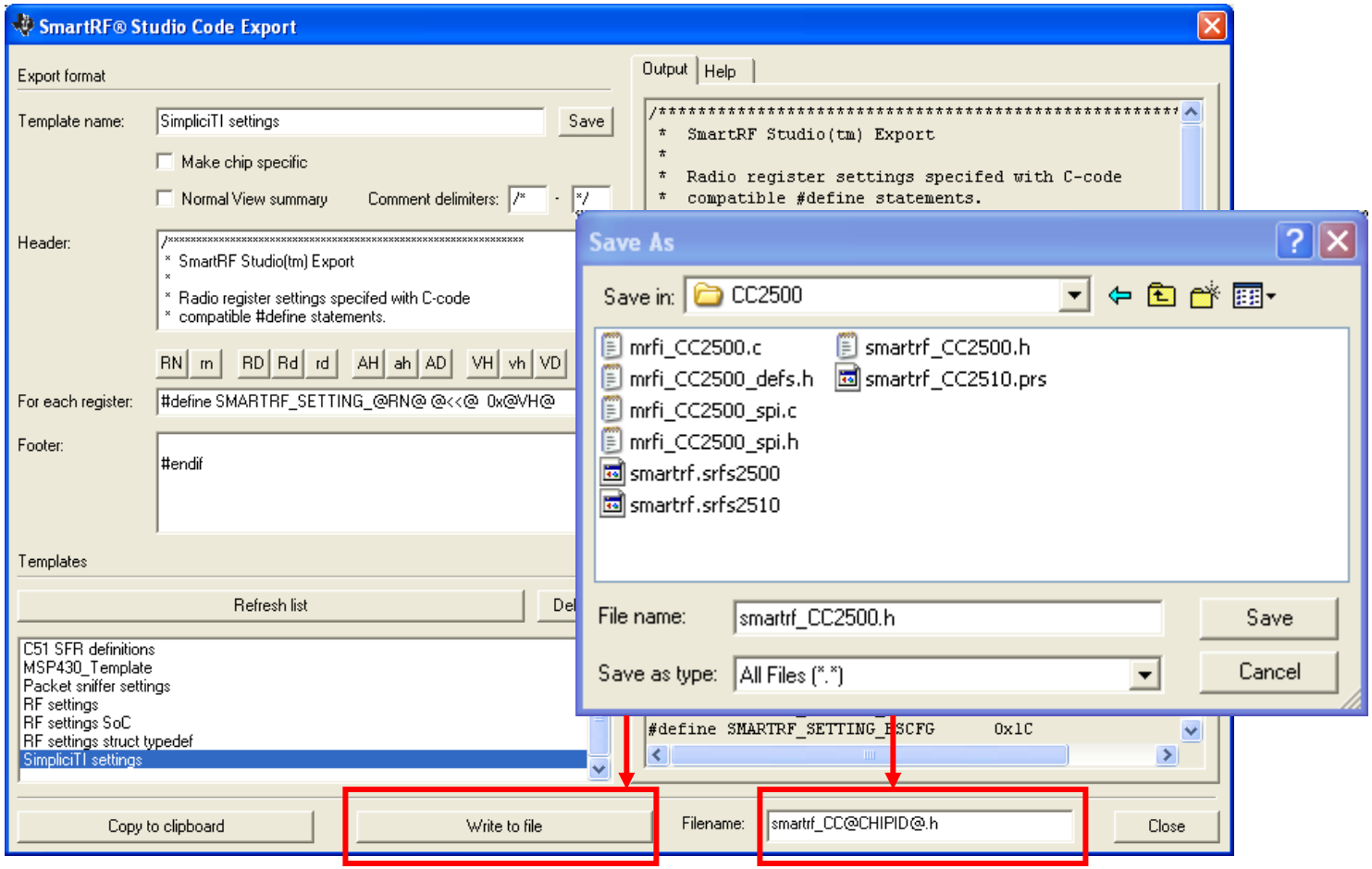


Selecting the Copy Settings to Register View puts the new values in the Register View Window

Exporting the settings to Code

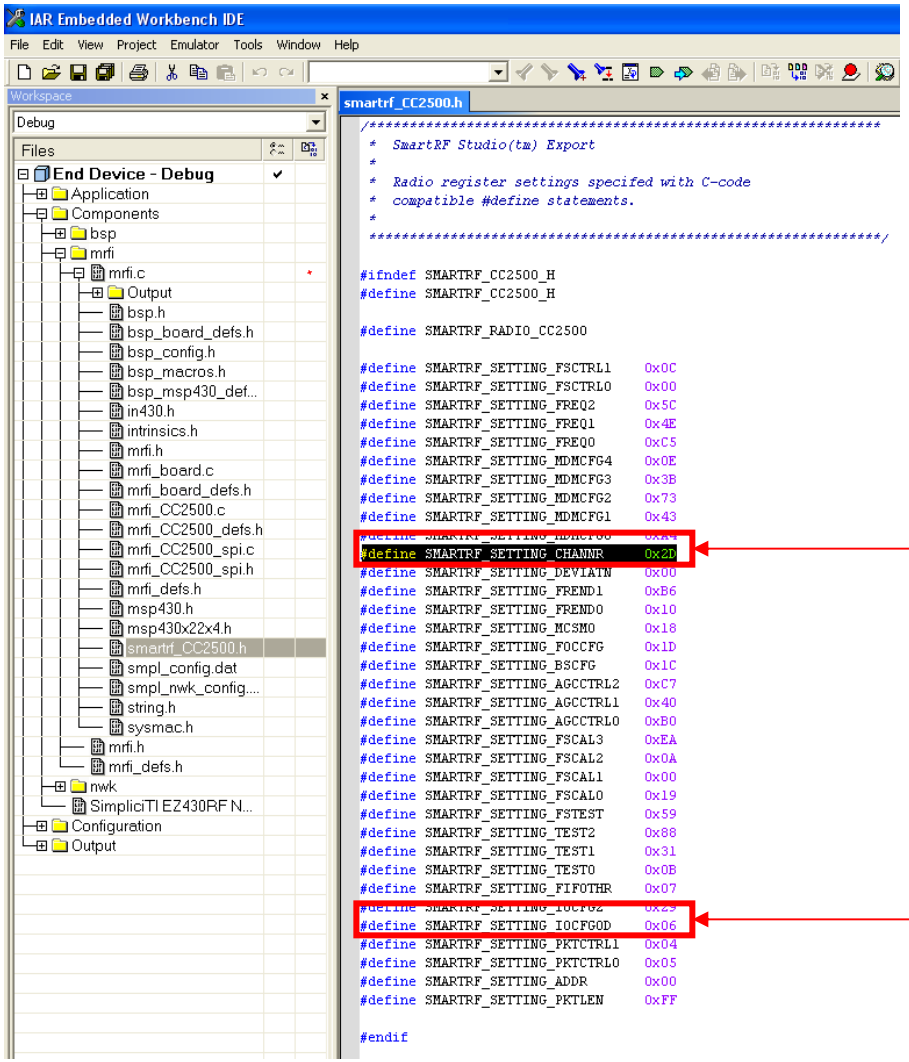


Over-Writing the SimpliciTI™ Radio File



C:\eZ430-RF2500 Wireless Sensor Monitor IAR Source v1.02\Components\mrfi\radios\CC2500

SimpliciTI™ Radio File Updated



CHANNR – 2D = 45
(2*16 + 13)

IOCFG0D – Still needs to be 0x06 for SimpliciTI to work

Getting Started with SimpliciTI™ and the eZ430

Questions?

m-claassen@ti.com