Getting Started with SimpliciTI™ and the eZ430

Texas Instruments

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 to show our ez430-RF2500 demo.

Agenda: Introducing SimpliciTI™

- SimpliciTI™ Overview
- Topology Examples
- Stack Architecture and API Details
- 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

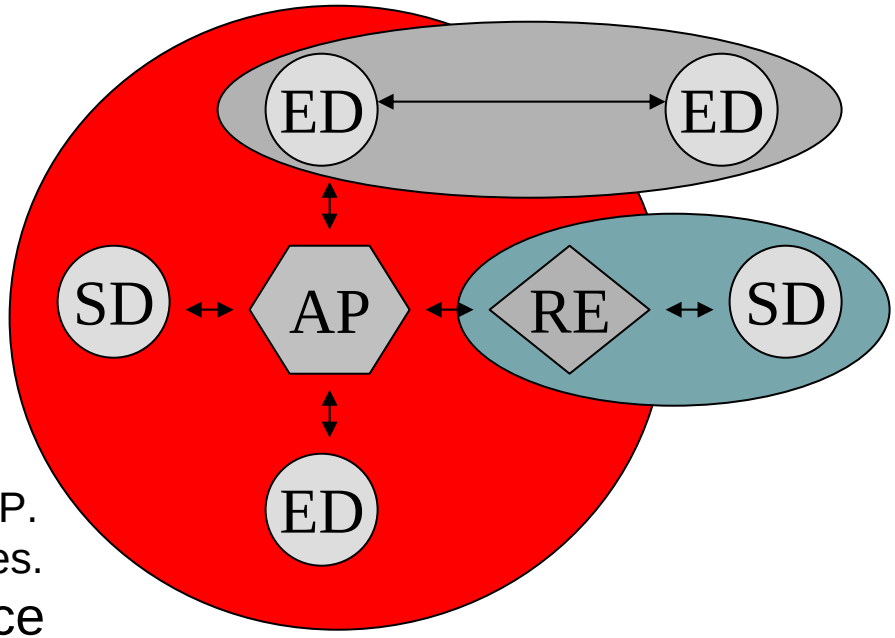
Device Configurations:

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



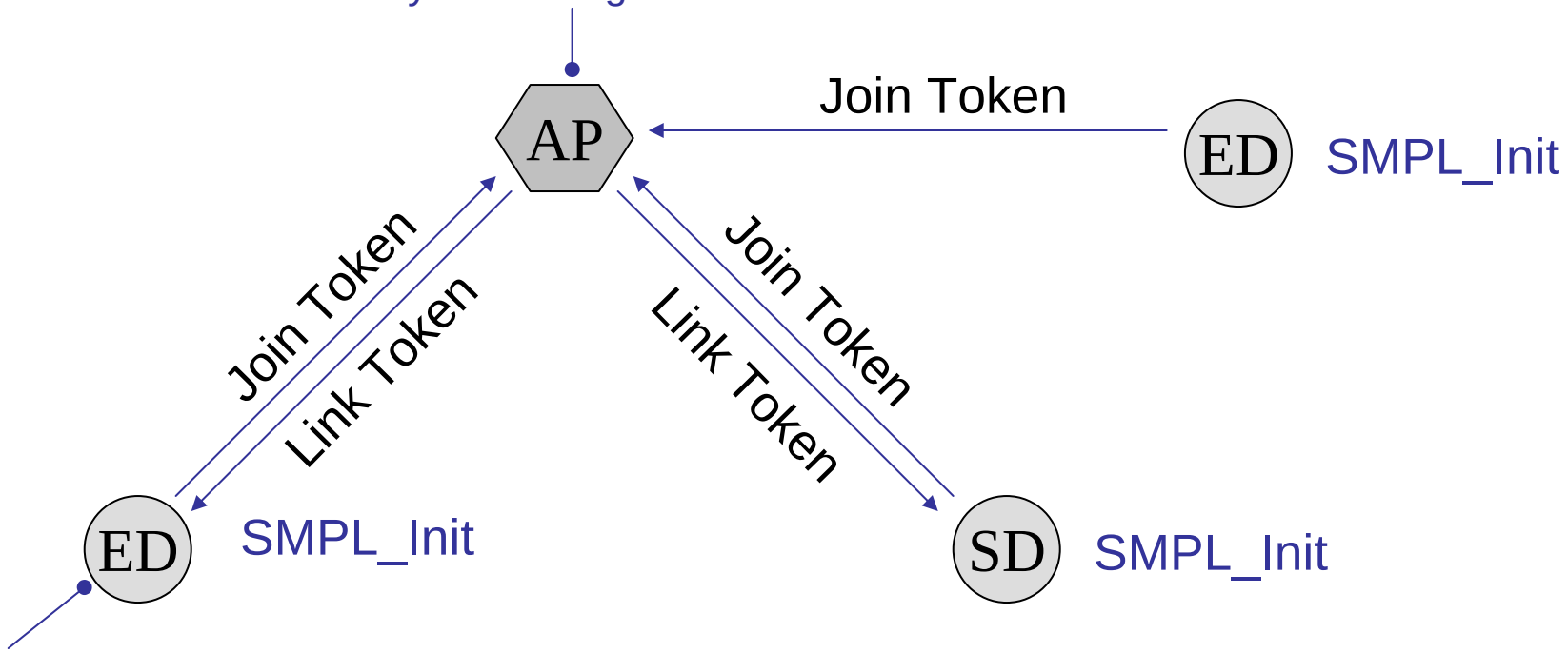
Topologies:
- Access Point Star
- Access Point Star w/ Range Extender
- Peer to Peer

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
7. The Access Point will establish Store and Forward data storage for the polling End Devices to ensure delivery of messages intended for them.

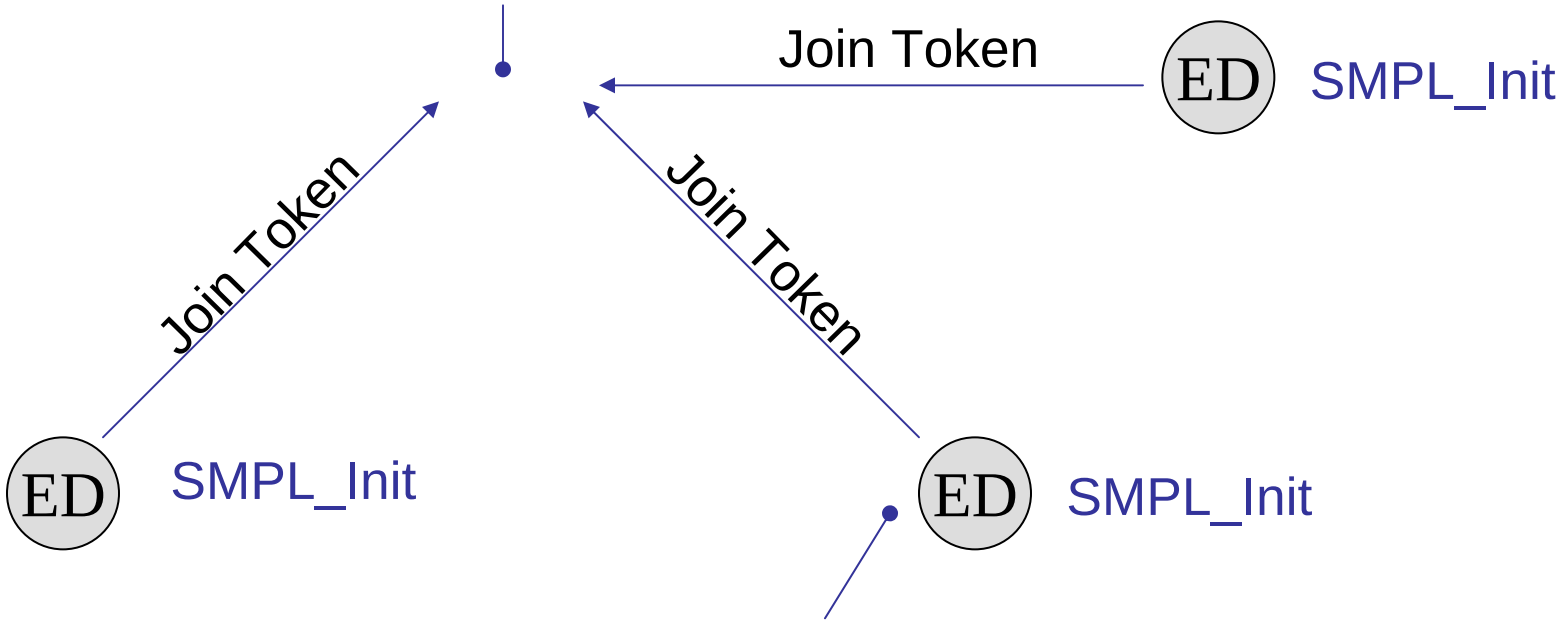


2. End Devices upon initialization, send a Join Token request to the network
4. The End Device overwrites its default Link Token w/ the new Link Token received from the AP

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.

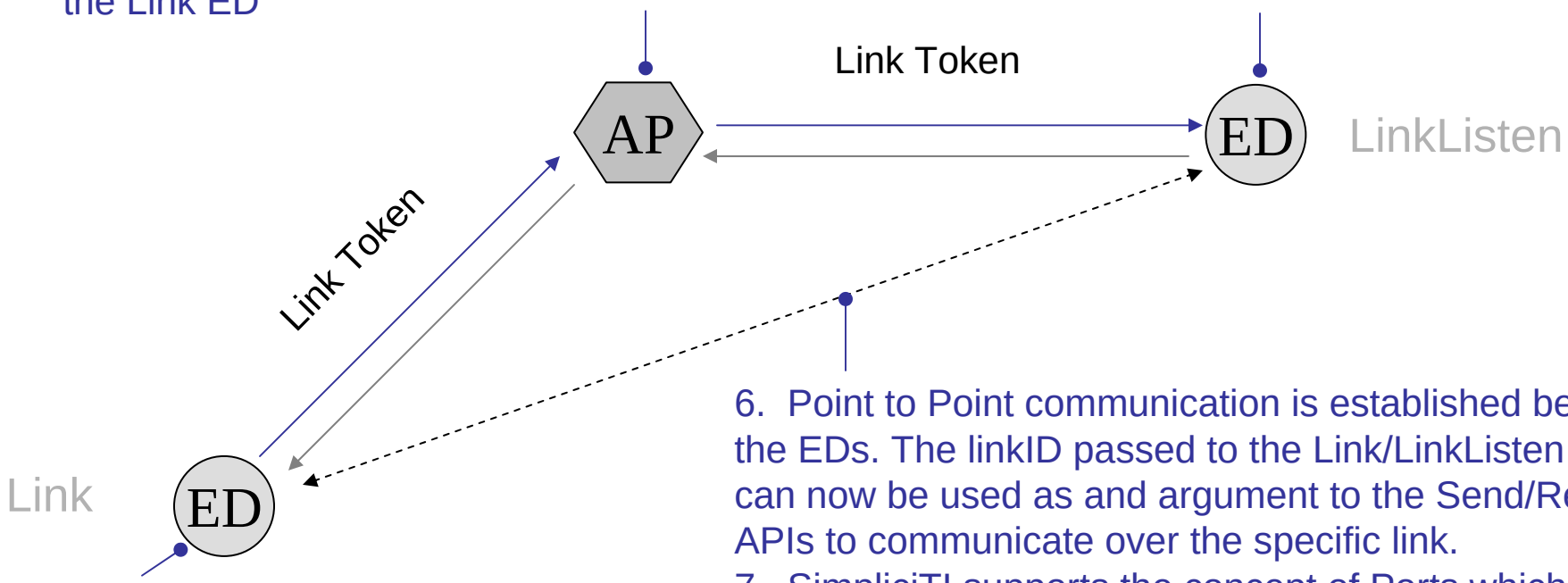


Since no Store and Forward Service exists, the polling End Devices cannot sleep as they may miss any asynchronous messages sent to them. An AP to required enable polling End Device functionality.

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



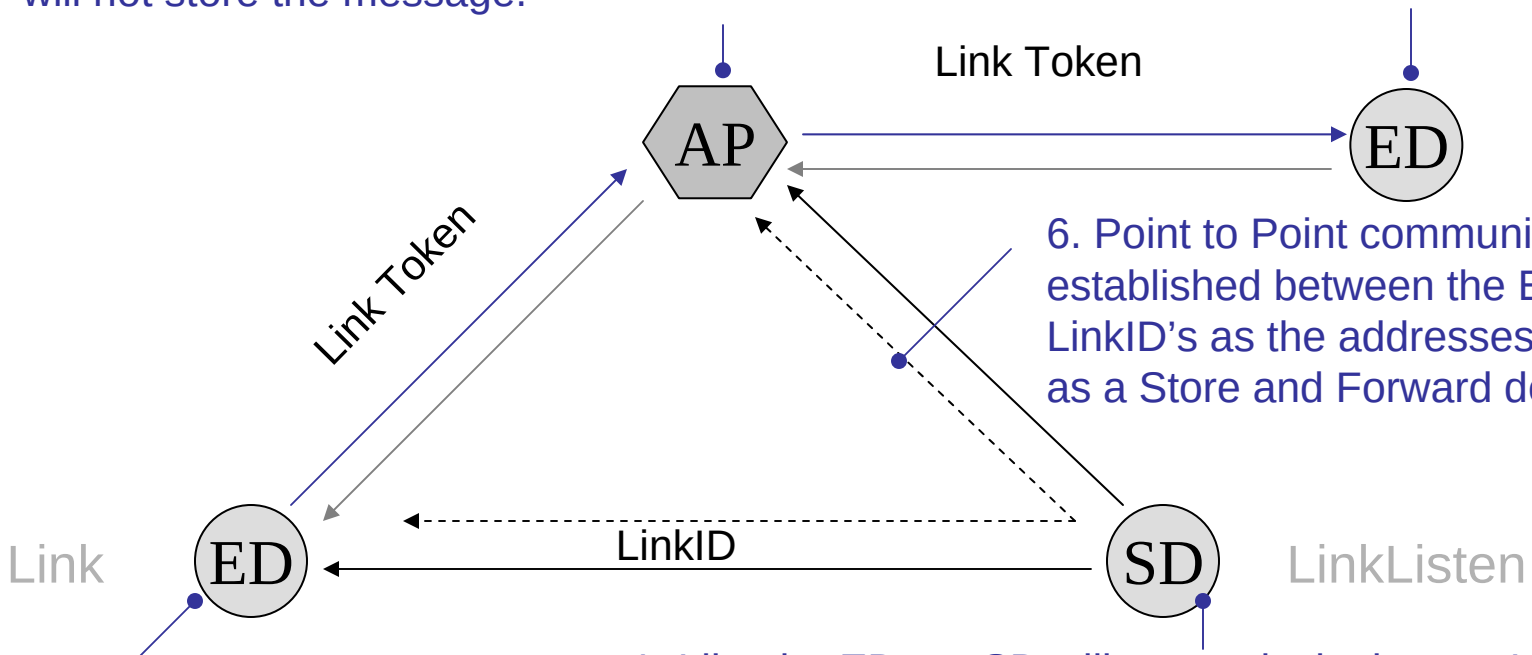
6. Point to Point communication is established between the EDs. The linkID passed to the Link/LinkListen APIs can now be used as and argument to the Send/Receive APIs to communicate over the specific link.
7. SimpliciTI supports the concept of Ports which will allow multiple connections between the same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

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

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.



6. Point to Point communication is established between the EDs using the LinkID's as the addresses and the AP acting as a Store and Forward device for the SD

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

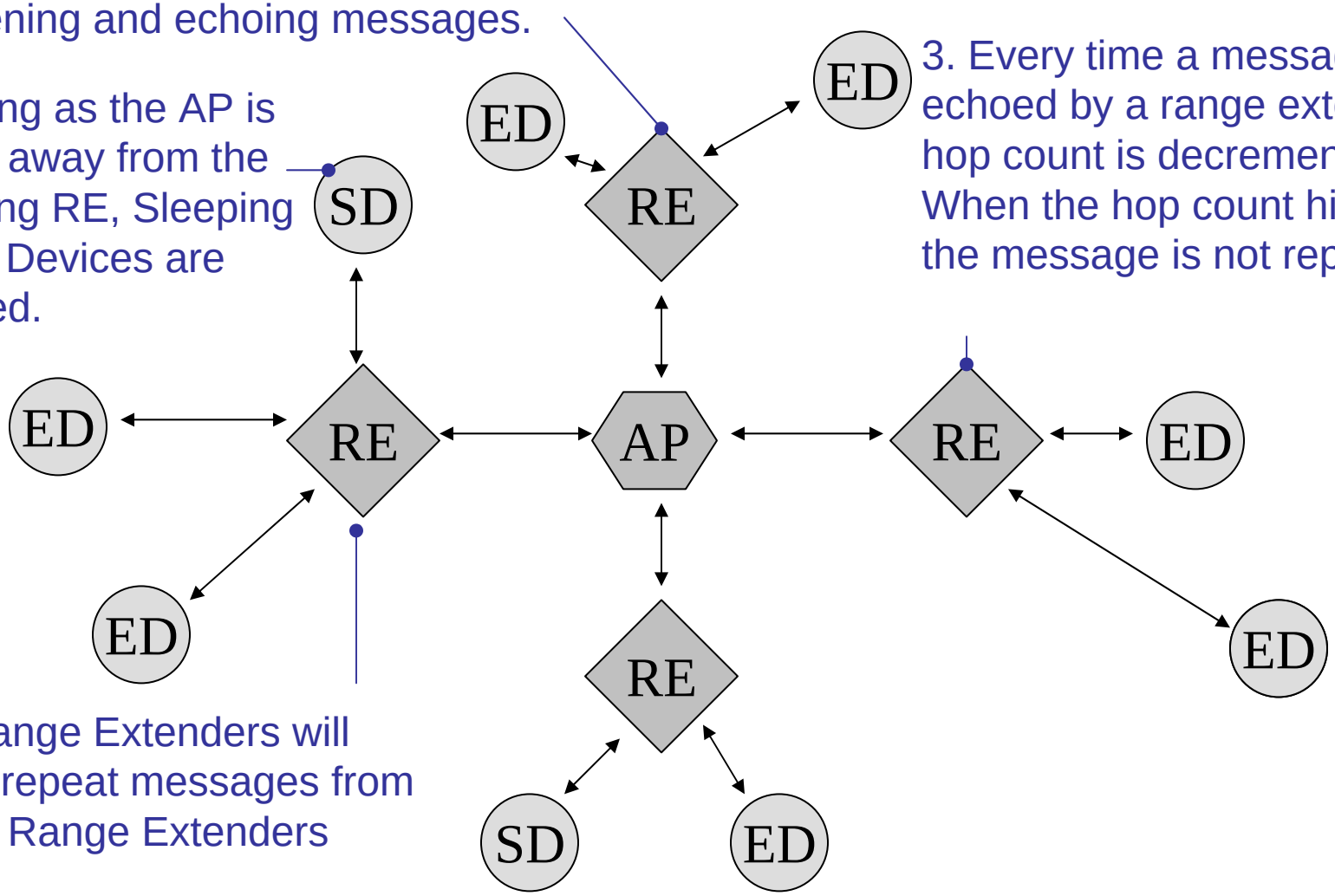
5. Upon receiving the Link message repeated by the AP (assuming Link ED is out of range) & accepting it, the SD will return a link reply message and go to sleep. It will then wake up and poll the AP for application level messages sent to its links.

Adding a Range Extender

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

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

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

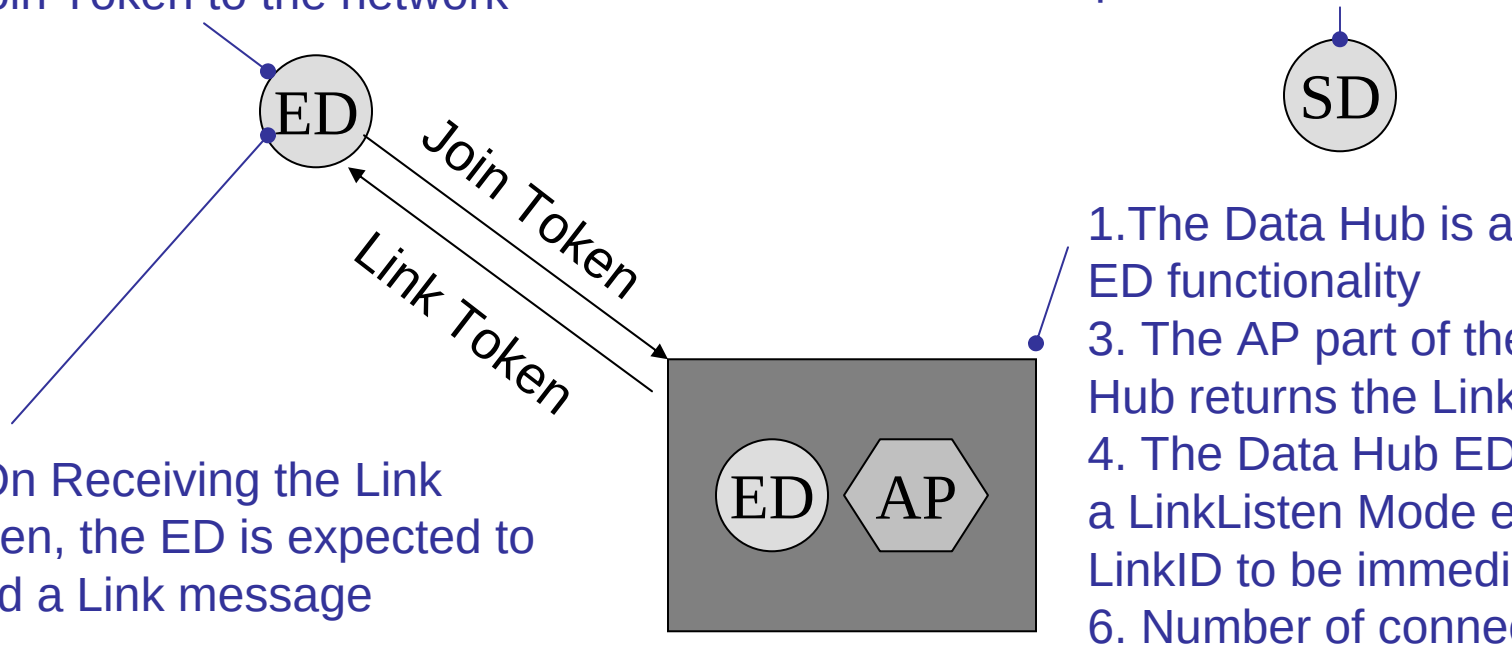


2. Range Extenders will NOT repeat messages from other Range Extenders

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

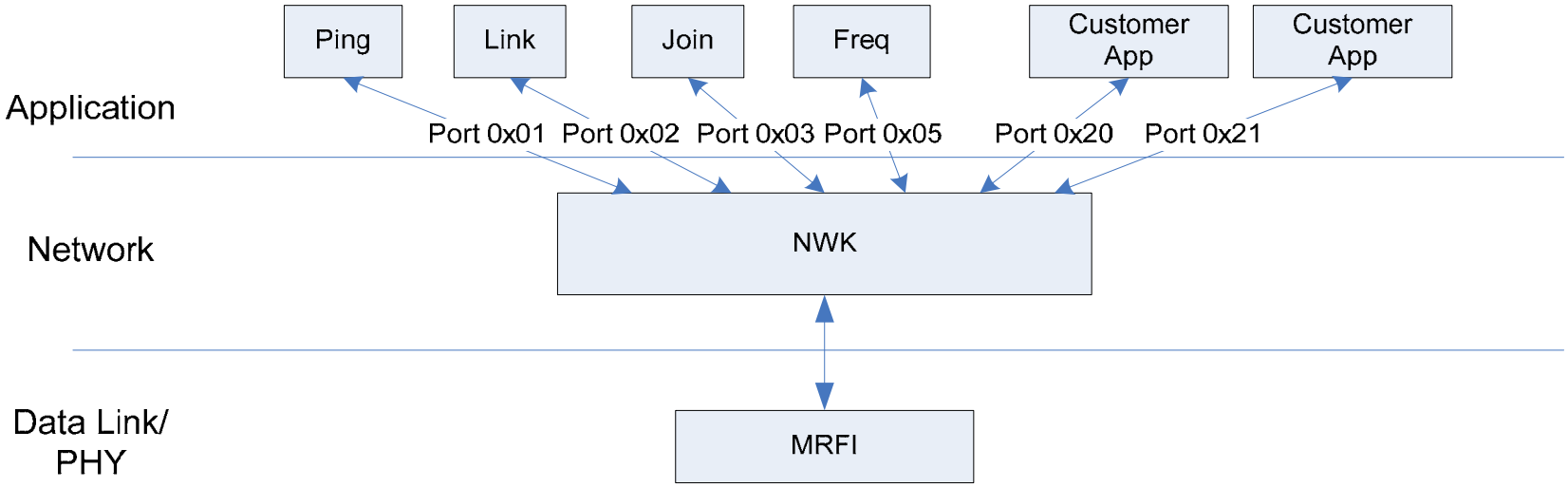
Stack Architecture and API Details

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);

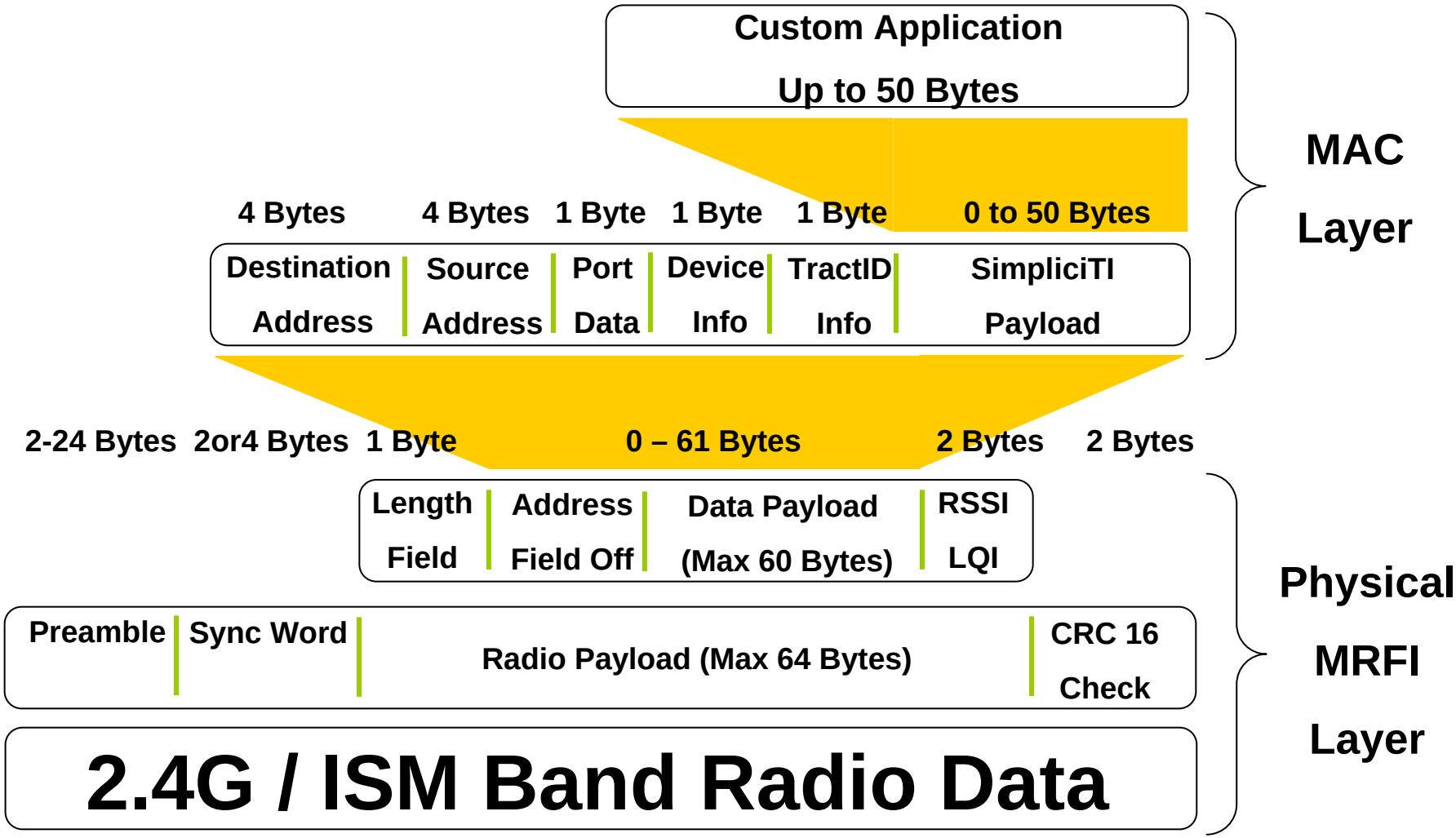
Architectural Overview

- 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

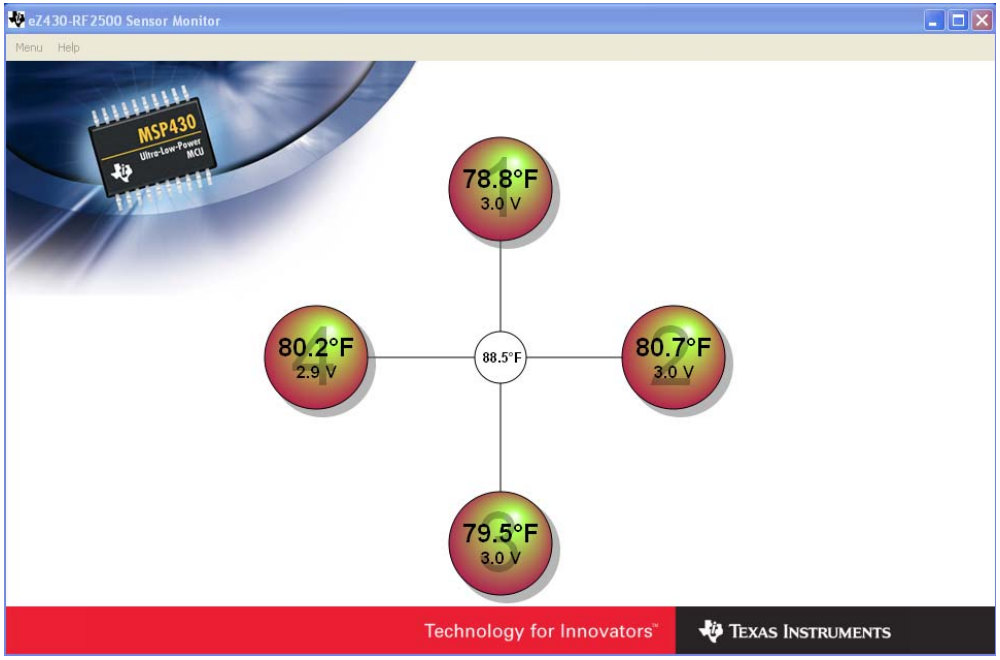
SimpliciTI™ – CC2500/CC1100

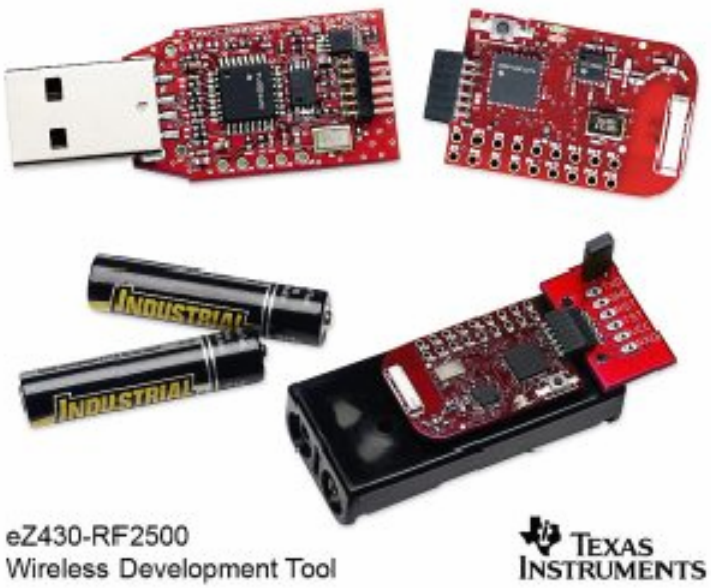


SimpliciTI™ Device Objects Overview

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-RF2500 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

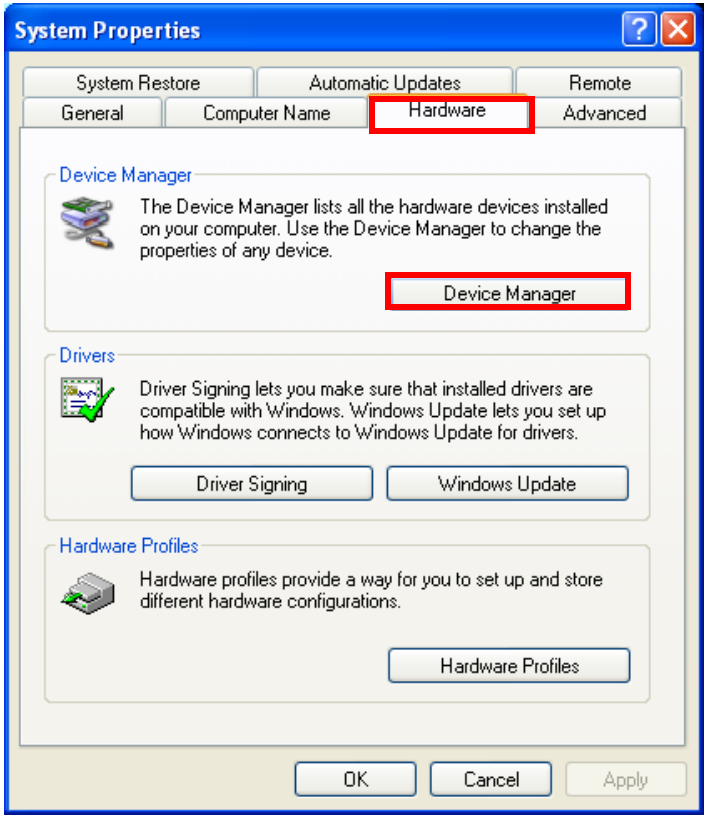
6. Install the eZ430-RF2500 Sensor Monitor Demo

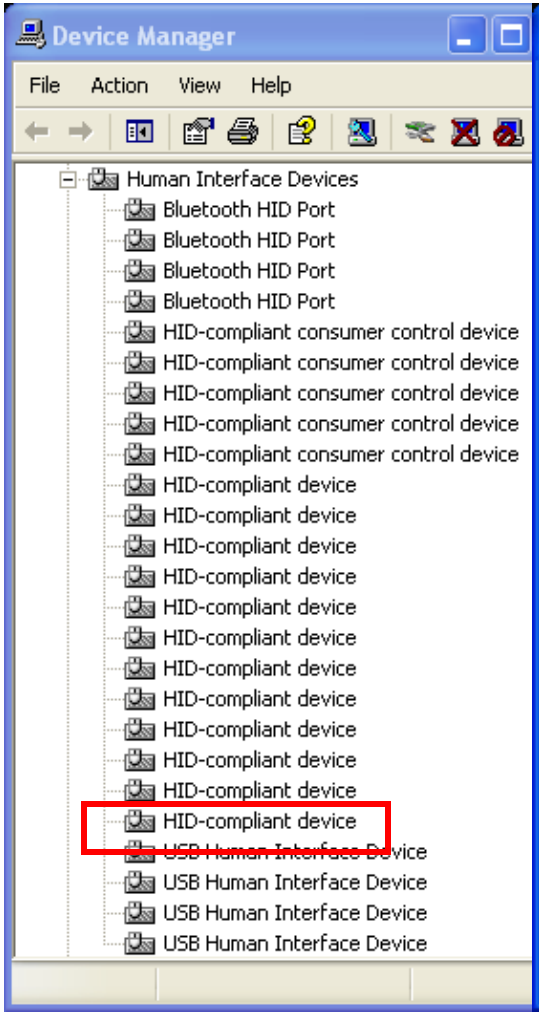
http://www.ti.com/litv/zip/slac139b

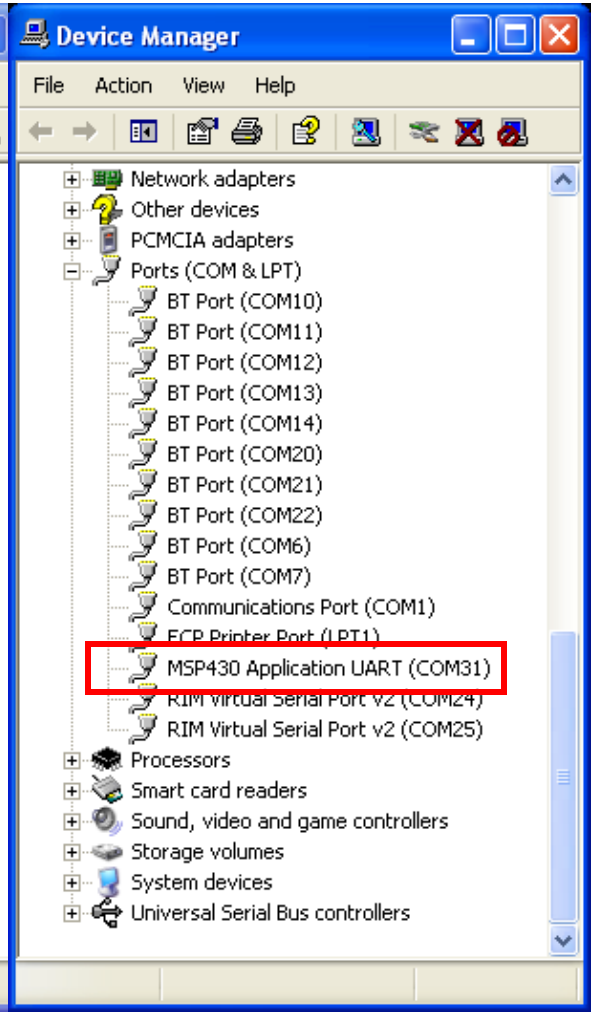
7. Install the Smart RF Studio Tool

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

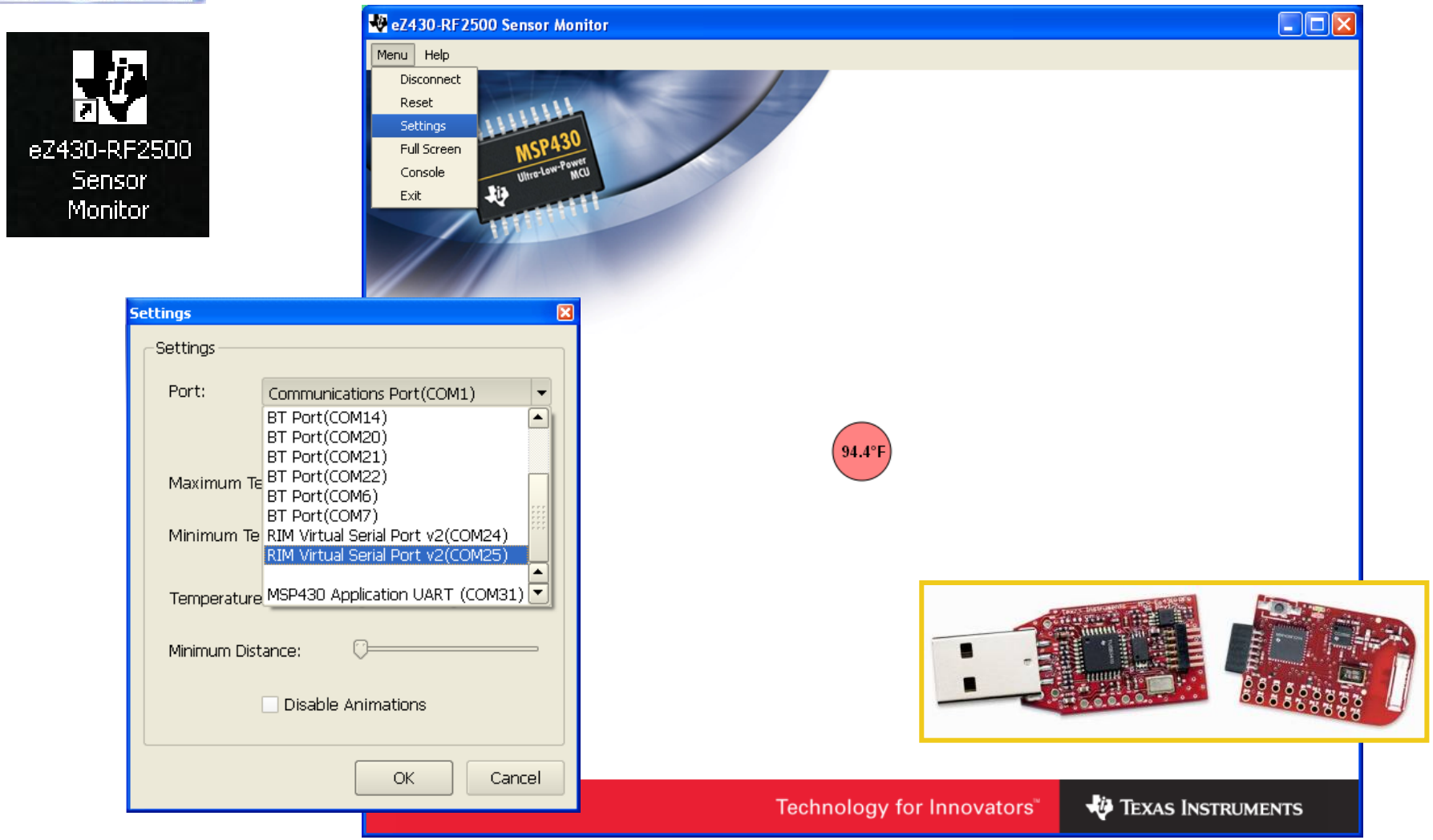
USB Driver Locations



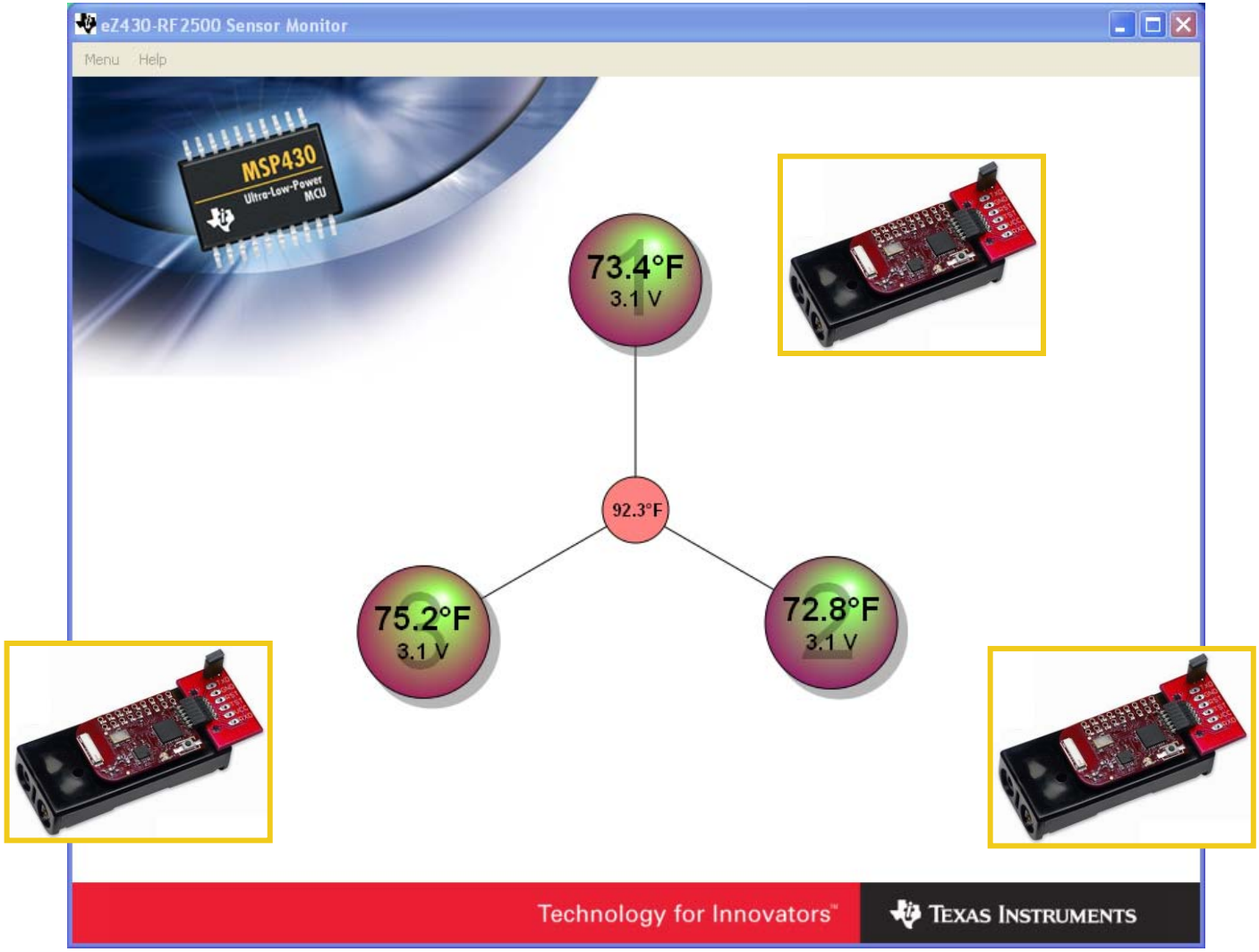




eZ430-RF2500 Sensor Monitor



Power On Battery RF2500



View of Join Token Transaction

Texas Instruments General Packet Sniffer SimpliciTI

File Help

P.nbr.	Time (us)	Length	Dest. Address	Source Address	Port		Device Info			Transaction ID	Applicaton payload						RSSI (dBm)	LQI	FCS
					Encryption	Number	Rec.Type	Send.Type	HCount			Join							
												App.Info	Token		Nbr_Conn.				
1	+0 =0	19	0x00000000	0x9E5CA29F	NO	0x03	ALWAYS_LISTEN	END_DEVICE	03	0xEF	05 08 07 06 05 02	0x05	0x5060708		0x02		-66	42	OK
2	+1395 =1395	19	0x9E5CA29F	0x865B93D4	NO	0x03	ALWAYS_LISTEN	ACCESS_POINT	01	0xEF	85 EF BE AD DE 00	0x85	0xDEADBEEF	Encryption Key	NO		-51	10	OK
3	+42312 =43707	21	0x00000000	0x9E5CA29F	NO	0x02	ALWAYS_LISTEN	END_DEVICE	03	0xF0	07 EF BE AD DE 3D 01 00	Link App.Info	Token	Local_Port	Link_numb.	RX_Type	-63	33	OK
												0x07	0xDEADBEEF	0x3D	0x01	ALWAYS_LISTEN			
4	+3538 =47245	16	0x9E5CA29F	0x865B93D4	NO	0x02	ALWAYS_LISTEN	ACCESS_POINT	00	0xF0	82 20 00	Link App.Info	Acc.Token	RX_Type			-51	15	OK
												0x82	0x20	ALWAYS_LISTEN					
5	+832706 =879951	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF1	00 01 1E	Unknown Port 0x20					-76	10	OK
6	+880847 =1760798	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF2	04 01 1E	Unknown Port 0x20					-72	10	OK
7	+879414 =2640212	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF3	04 01 1E	Unknown Port 0x20					-68	18	OK
8	+880933 =3521145	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF4	04 01 1E	Unknown Port 0x20					-72	11	OK
9	+882679 =4403824	16	0x865B93D4	0x9E5CA29F	NO	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF5	00 01 1E	Unknown Port 0x20					-69	12	OK
	+878420	Length	Dest. Address	Source Address	Port		Device Info			Transaction ID	Applicaton payload						RSSI (dBm)	LQI	FCS
					Encryption	Number	Rec.Type	Send.Type	HCount		04 01	Unknown Port							

Setup | Select fields | Packet details | Address book | Display filter | Time line | Radio Settings

Field Name: Template:

Filter condition

First And Add Remove All Apply filter Turn off filter

Filter management: Add Remove Open Save Merge

Packet count: 47 Error count: 0 Filter Off

View of End Device Interaction

Texas Instruments General Packet Sniffer SimpliciTI

File Help

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	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF6	08 01 1D	0x20	-66	16	OK
1132	+297505 =536719613	16	0x865B93D4	0xC71F648A	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xC9	04 01 1E	0x20	-68	9	OK
1133	+212661 =536932274	16	0x865B93D4	0x9E5CA29F	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x53	08 01 1E	0x20	-53	12	OK
1134	+139426 =537071700	16	0x865B93D4	0x5CE2BCD6	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x75	17 01 1E	0x20	-61	15	OK
1135	+223891 =537295591	16	0x865B93D4	0x138B4378	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF7	08 01 1D	0x20	-66	10	OK
1136	+241051 =537536642	16	0x865B93D4	0xC71F648A	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCA	04 01 1E	0x20	-70	17	OK
1137	+274649 =537811291	16	0x865B93D4	0x9E5CA29F	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x54	08 01 1E	0x20	-53	11	OK
1138	+182979 =537994270	16	0x865B93D4	0x5CE2BCD6	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x76	13 01 1E	0x20	-60	8	OK
1139	+170673 =538164943	16	0x865B93D4	0x138B4378	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF8	08 01 1D	0x20	-67	14	OK
1140	+192959 =538357902	16	0x865B93D4	0xC71F648A	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCB	08 01 1E	0x20	-69	10	OK
1141	+332509 =538690411	16	0x865B93D4	0x9E5CA29F	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x55	04 01 1E	0x20	-53	11	OK
1142	+224373 =538914784	16	0x865B93D4	0x5CE2BCD6	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0x77	17 01 1E	0x20	-61	11	OK
1143	+123656 =539038440	16	0x865B93D4	0x138B4378	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xF9	08 01 1D	0x20	-67	13	OK
1144	+136375 =539174815	16	0x865B93D4	0xC71F648A	N0	0x20	ALWAYS_LISTEN	END_DEVICE	03	0xCC	04 01 1E	0x20	-69	17	OK

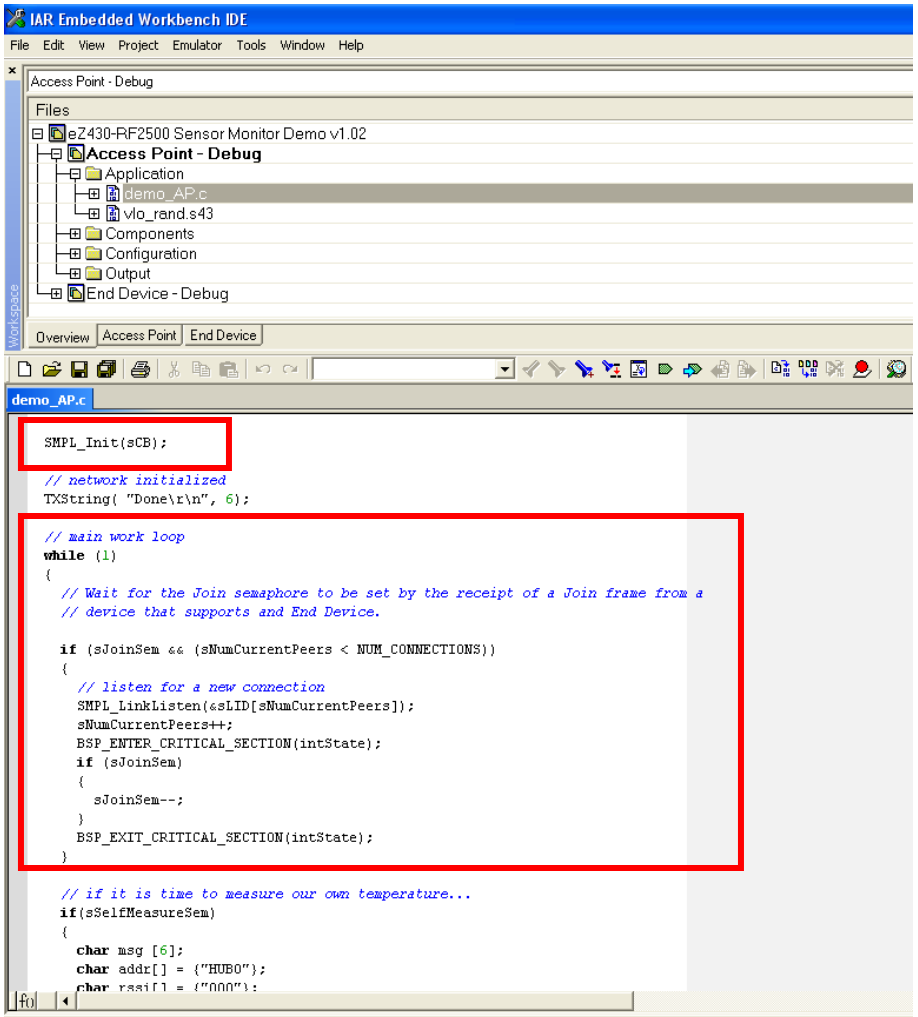
Packet count: 1145 | Error count: 10 | Filter Off

Device Objects - Access Point

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

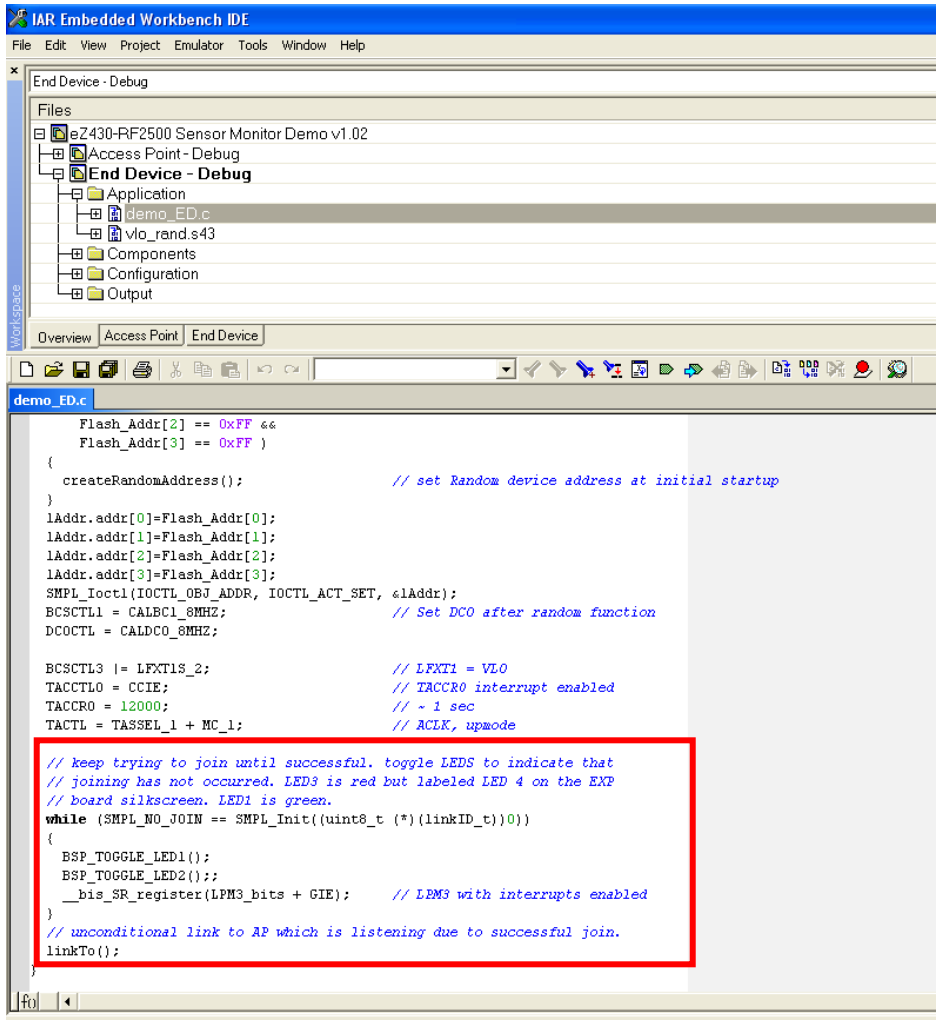
```c
/*------------------------------------------------------------------------------
* Runs in ISR context. Reading the frame should be done in the
* application thread not in the ISR thread.
------------------------------------------------------------------------------*/
static uint8_t sCB(linkID_t 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

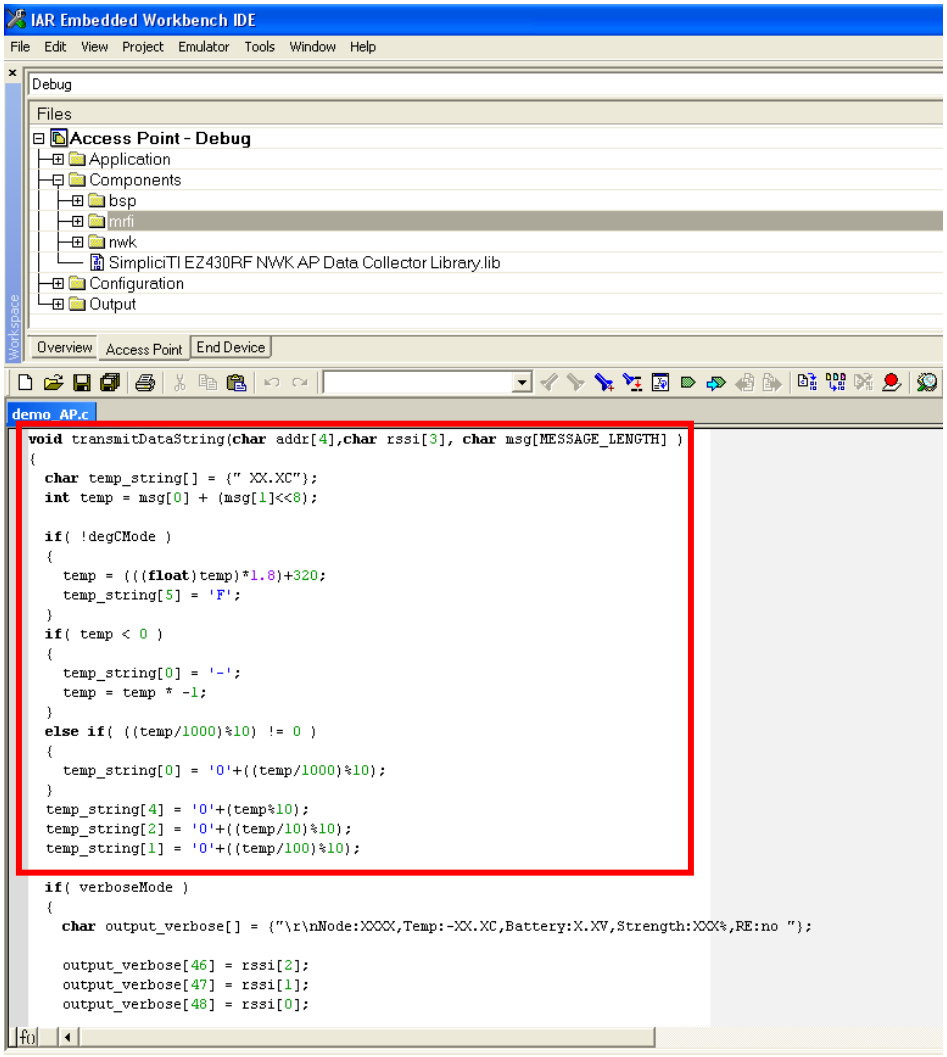
- MSP430 Initialized
- Board initialized
- SMPL_Init
  - Set up as End Device
  - Option to Sleep
  - No Call Back
  - Join Token Sent, & repeat if no reply
- Link process Initiated
  - Assumes LinkListen ED from AP
- ADC Sampled
  - Twice per Second (Temp & Vcc)
- SMPL_Send
  - Two Byte Temperature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



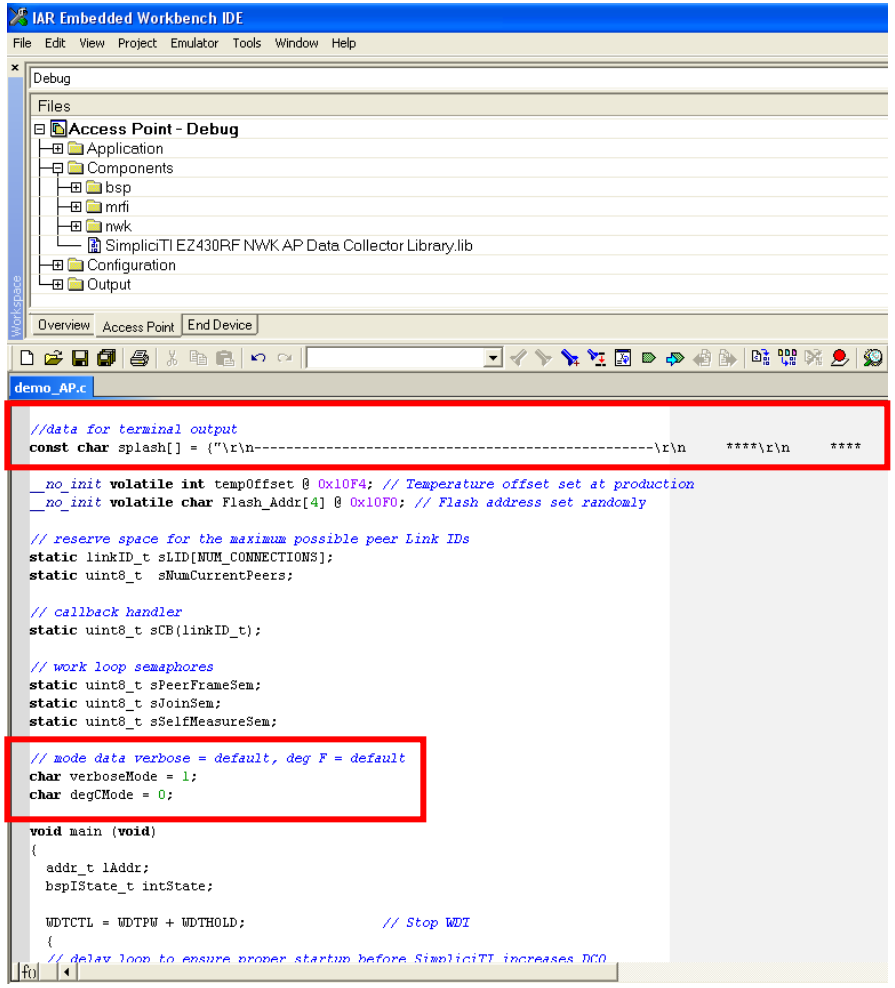
Device Objects - Access Point

- **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**

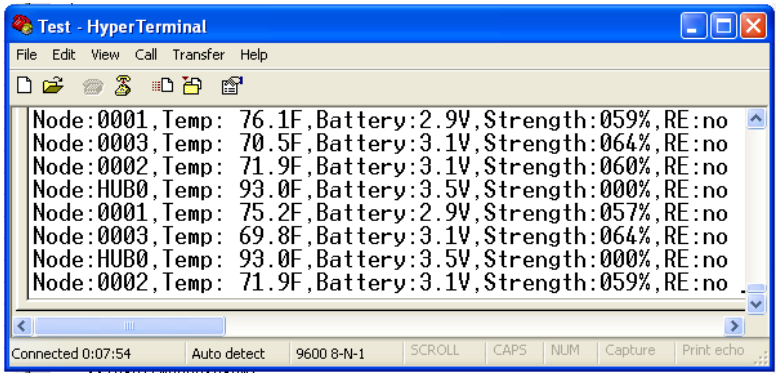
- **SimpliciTI Configuration**
  - **Star hub in the network**
    - **1 AP per net (Join Token)**
  - **Always-on**
  - **Configured as a Data Hub**
    - **AP + ED**



UART Terminal Output Types

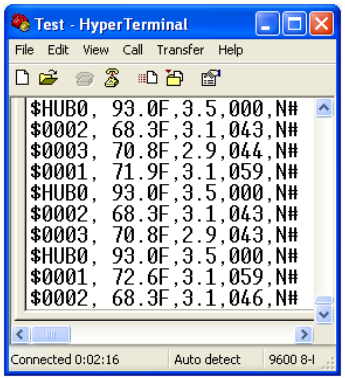



```
----------------------------------------------------
     ****
     ****              eZ430-RF2500
     ******o****       Temperature Sensor Network
********_///_****      Copyright 2007
 ******/_//_/*****     Texas Instruments Incorporated
  ** ***(__/*****      All rights reserved.
      *********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



verboseMode = 1

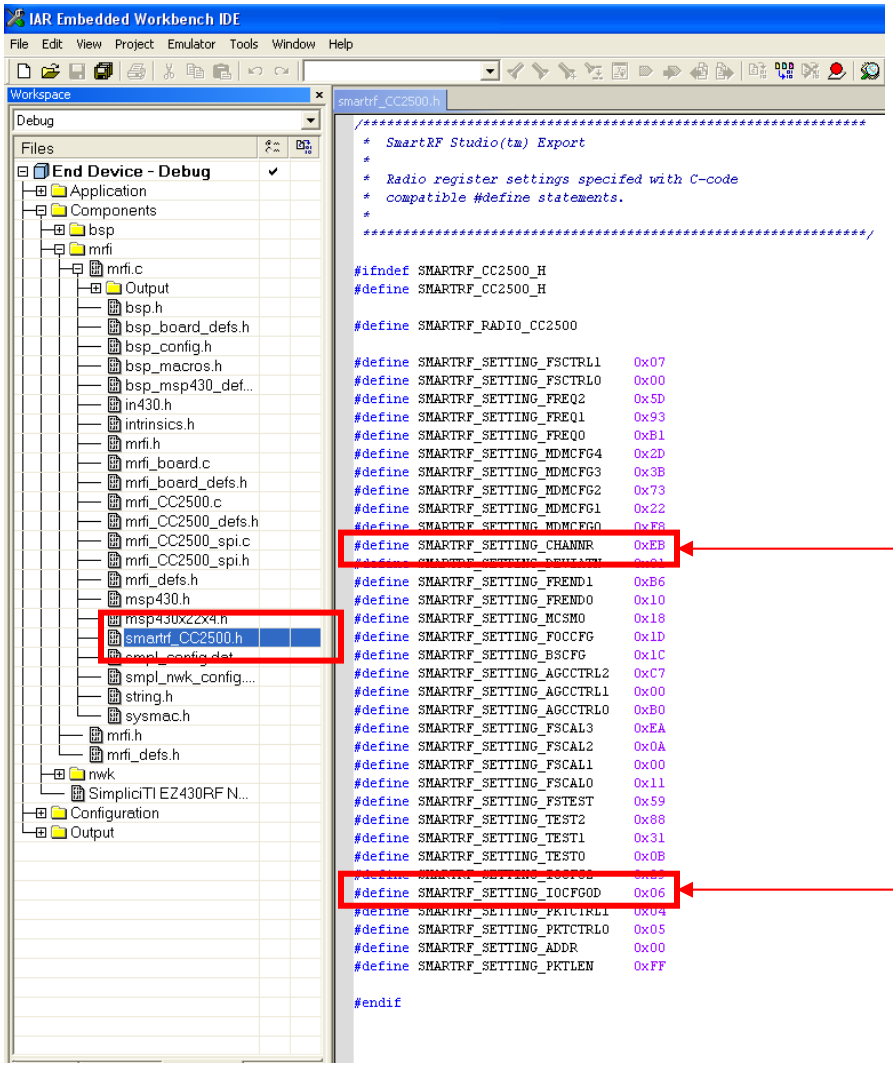
verboseMode = 0



SimpliciTI™ Overview

Radio Settings and RF Studio

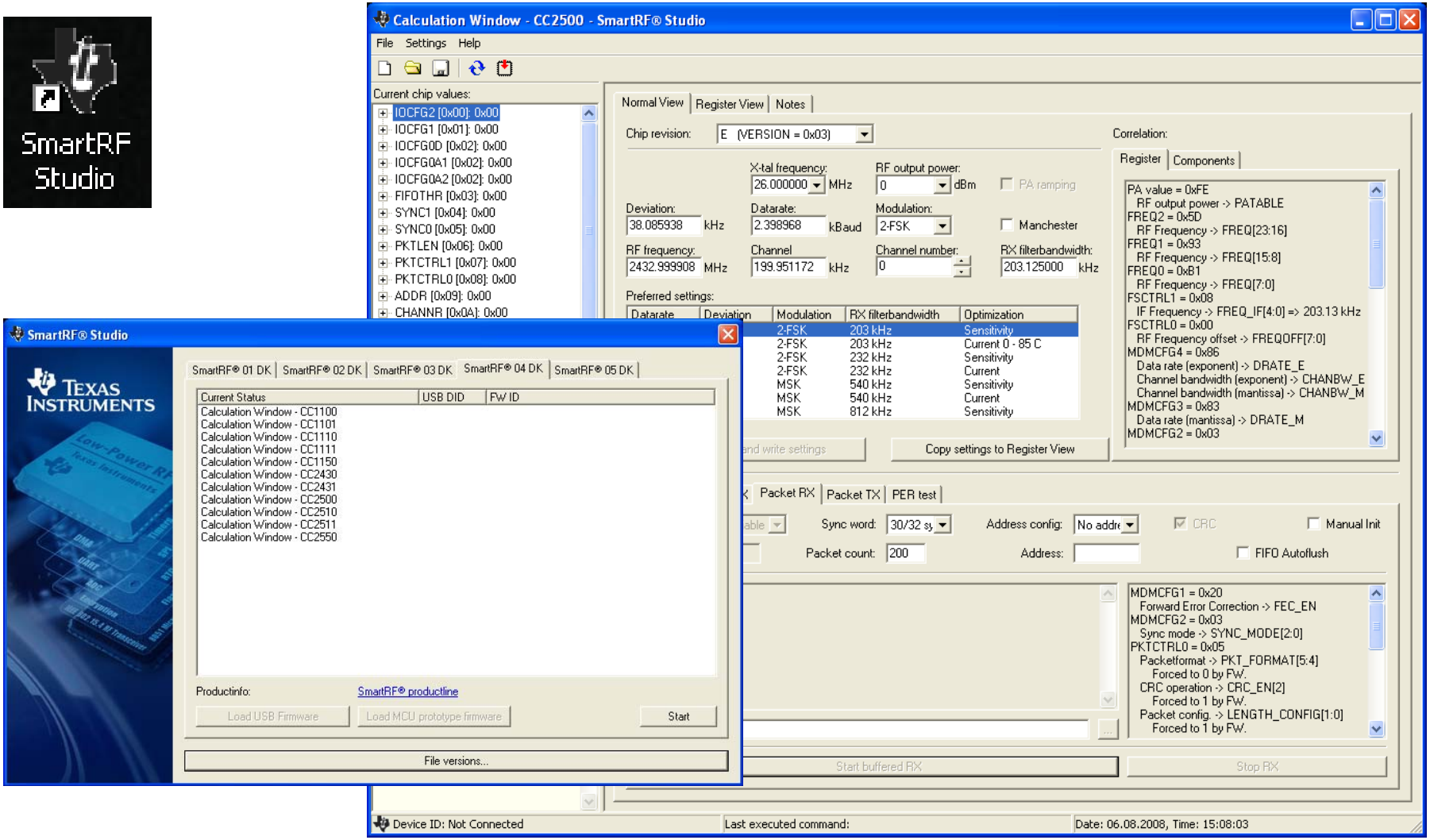
SimpliciTI™ Radio Settings



CHANNR – Sets the Channel

IOCFG0D – Sets the ISR Pin

SmartRF Studio

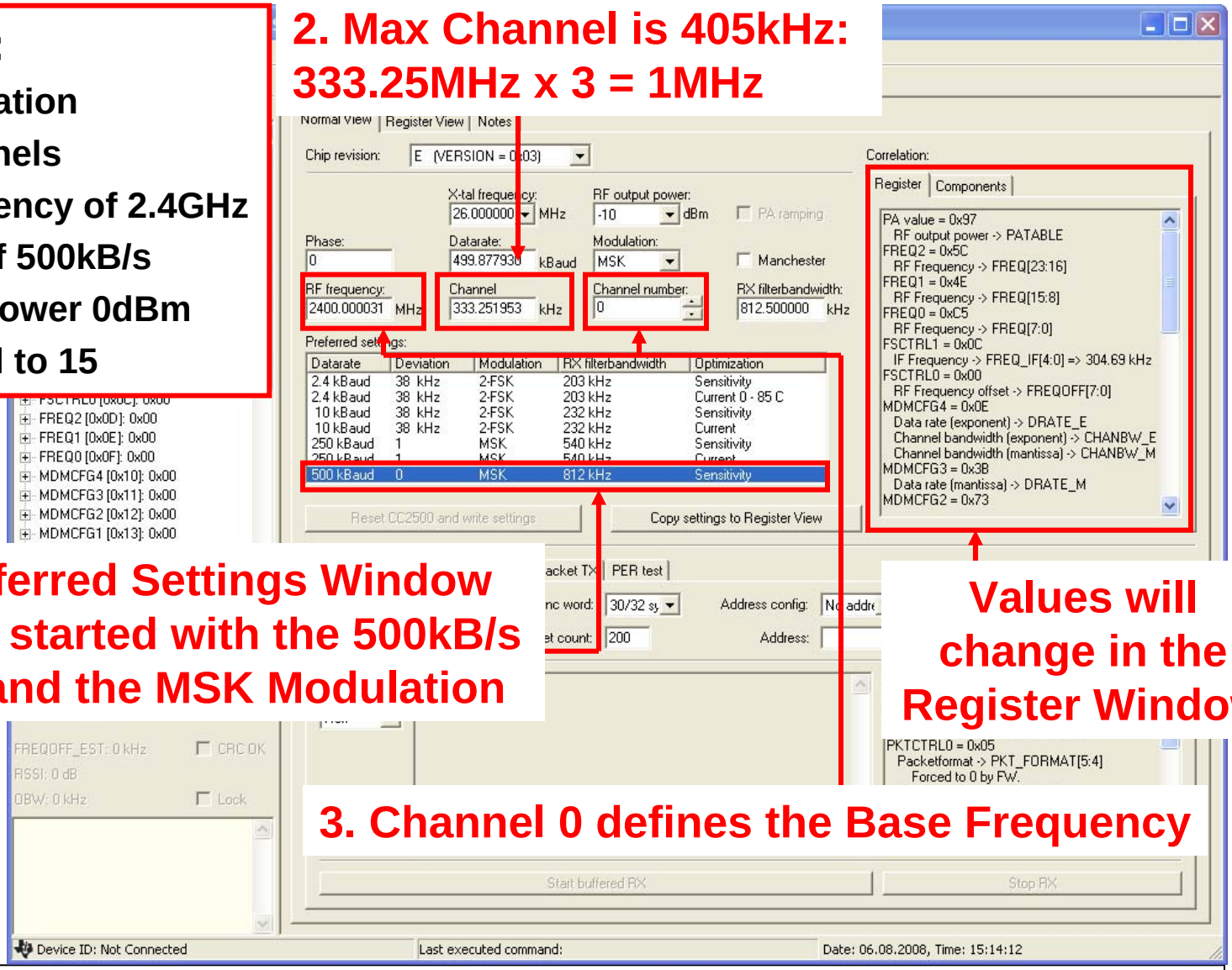


Example

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

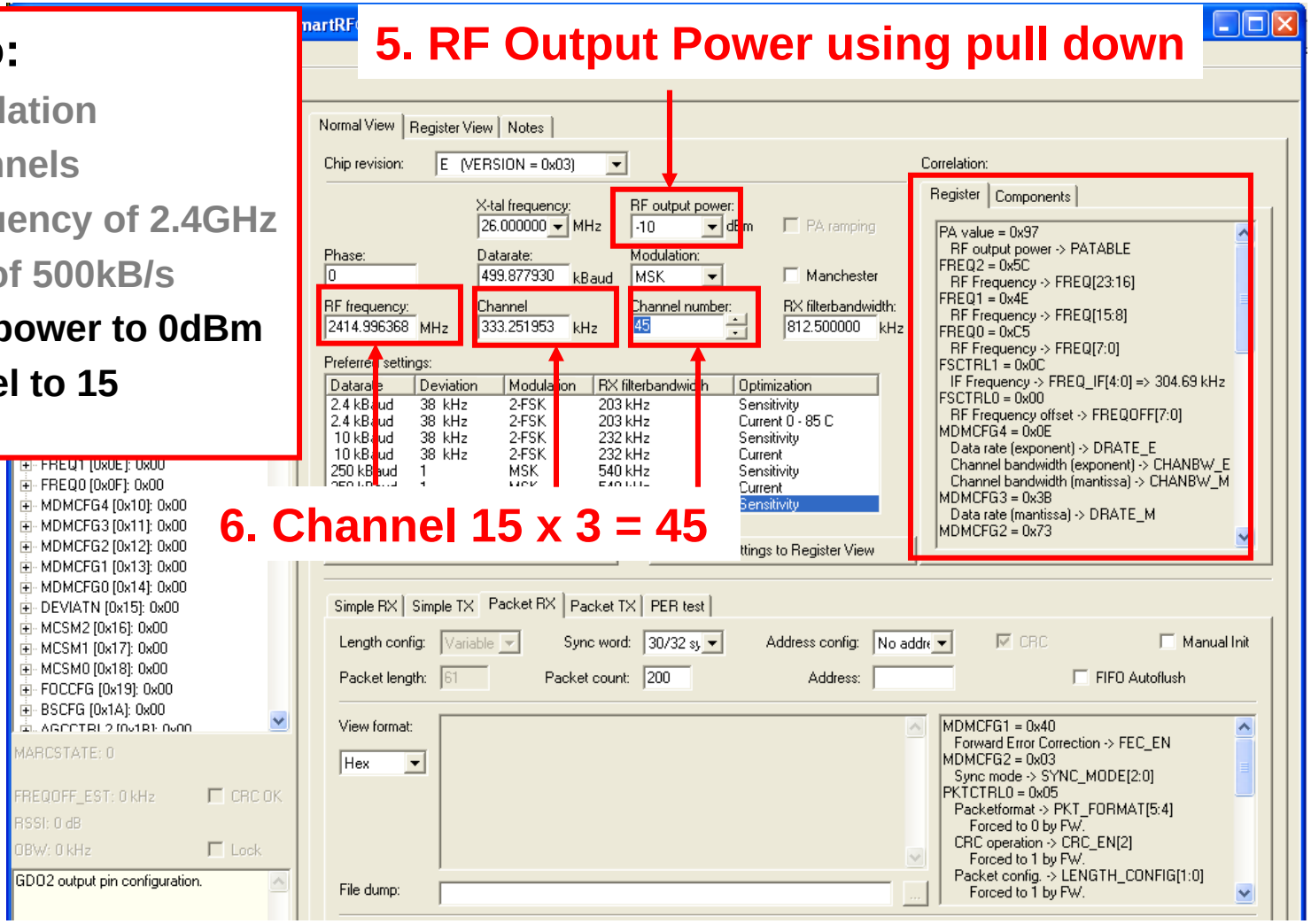
Values will change in the Register Window

3. Channel 0 defines the Base Frequency

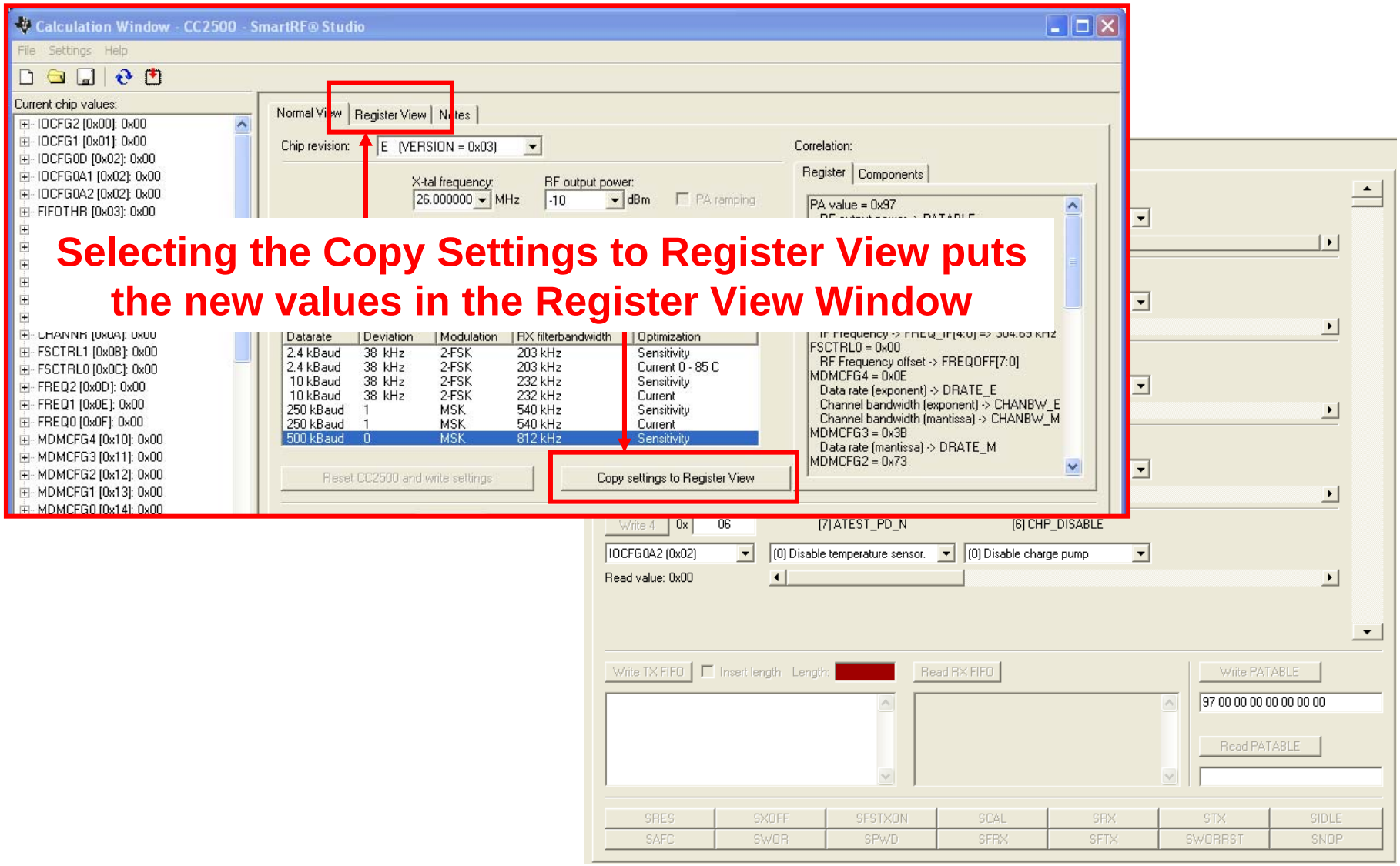
Example

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

Calculation Window - CC2500 - SmartRF® Studio

File Settings Help

Reset configuration...
Open configuration...
Save configuration...
Load CC2500 state...
Save CC2500 state...
Export CC2500 registers...
Import CC2500 registers...
Export CC2500 code...
Close

PKTCTRL1 [0x07]: 0x00
PKTCTRL0 [0x08]: 0x00
ADDR [0x09]: 0x00
CHANNR [0x0A]: 0x00
FSCTRL1 [0x0B]: 0x00
FSCTRL0 [0x0C]: 0x00
FREQ2 [0x0D]: 0x00
FREQ1 [0x0E]: 0x00
FREQ0 [0x0F]: 0x00
MDMCFG4 [0x10]: 0x00
MDMCFG3 [0x11]: 0x00
MDMCFG2 [0x12]: 0x00
MDMCFG1 [0x13]: 0x00
MDMCFG0 [0x14]: 0x00
DEVIATN [0x15]: 0x00
MCSM2 [0x16]: 0x00
MCSM1 [0x17]: 0x00
MCSM0 [0x18]: 0x00
FOCCFG [0x19]: 0x00
BSCFG [0x1A]: 0x00

MARCSTATE: 0
FREQOFF_EST: 0 kHz
CRC OK
RSSI: 0 dB
OBW: 0 kHz
Lock

Normal View

Write 0 0x
IOCFG2 (0x00)
Read value: 0x00
Write 1 0x
IOCFG1 (0x01)
Read value: 0x00
Write 2 0x
IOCFG0D (0x02)
Read value: 0x00
Write 3 0x
IOCFG0A1 (0x02)
Read value: 0x00
Write 4 0x
IOCFG0A2 (0x02)
Read value: 0x00

Write TX FIFO

SRES SXOFF SFSTXON SCAL SRX STX SIDLE

SmartRF® Studio Code Export

Export format

Template name: SimpliciTI settings Save

Make chip specific

Normal View summary Comment delimiters: /* - */

Header:

```
/*****************************************************
 * SmartRF Studio(tm) Export
 *
 * Radio register settings specifed with C-code
 * compatible #define statements.
```

RN rn RD Rd rd AH ah AD VH vh VD <<

For each register: #define SMARTRF_SETTING_@RN@ @<<@ 0x@VH@

Footer:

```
#endif
```

Templates

Refresh list Delete

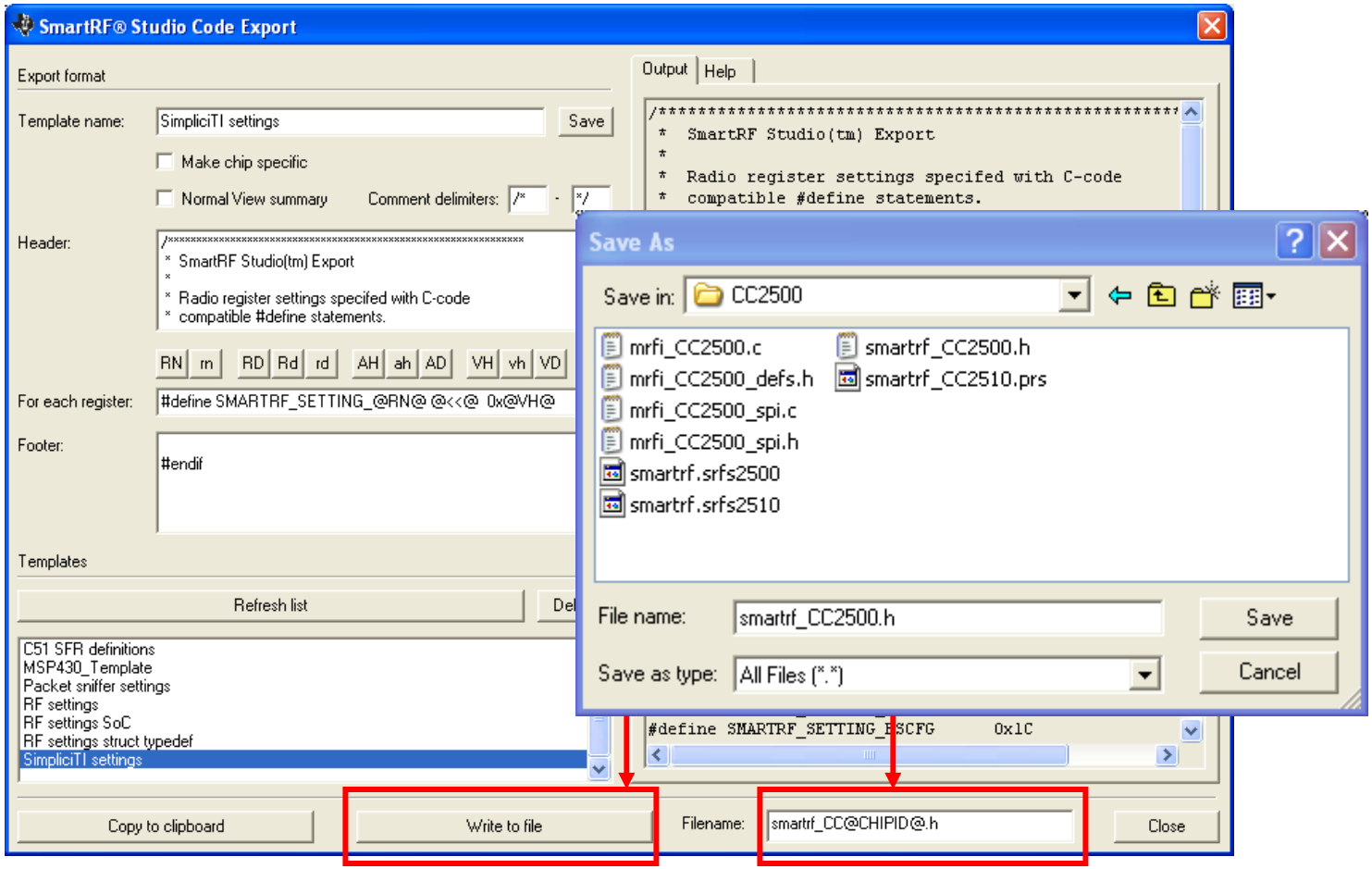
C51 SFR definitions
MSP430_Template
Packet sniffer settings
RF settings
RF settings SoC
RF settings struct typedef
SimpliciTI settings

Output Help

```
/*****************************************************
 *  SmartRF Studio(tm) Export
 *
 *  Radio register settings specifed with C-code
 *  compatible #define statements.
 *
 *****************************************************

#ifndef SMARTRF_CC2500_H
#define SMARTRF_CC2500_H

#define SMARTRF_RADIO_CC2500

#define SMARTRF_SETTING_FSCTRL1    0x0C
#define SMARTRF_SETTING_FSCTRL0    0x00
#define SMARTRF_SETTING_FREQ2      0x5C
#define SMARTRF_SETTING_FREQ1      0x4E
#define SMARTRF_SETTING_FREQ0      0xC5
#define SMARTRF_SETTING_MDMCFG4    0x0E
#define SMARTRF_SETTING_MDMCFG3    0x3B
#define SMARTRF_SETTING_MDMCFG2    0x73
#define SMARTRF_SETTING_MDMCFG1    0x43
#define SMARTRF_SETTING_MDMCFG0    0xA4
#define SMARTRF_SETTING_CHANNR     0x2D
#define SMARTRF_SETTING_DEVIATN    0x00
#define SMARTRF_SETTING_FREND1     0xB6
#define SMARTRF_SETTING_FREND0     0x10
#define SMARTRF_SETTING_MCSM0      0x18
#define SMARTRF_SETTING_FOCCFG     0x1D
#define SMARTRF_SETTING_BSCFG      0x1C
```

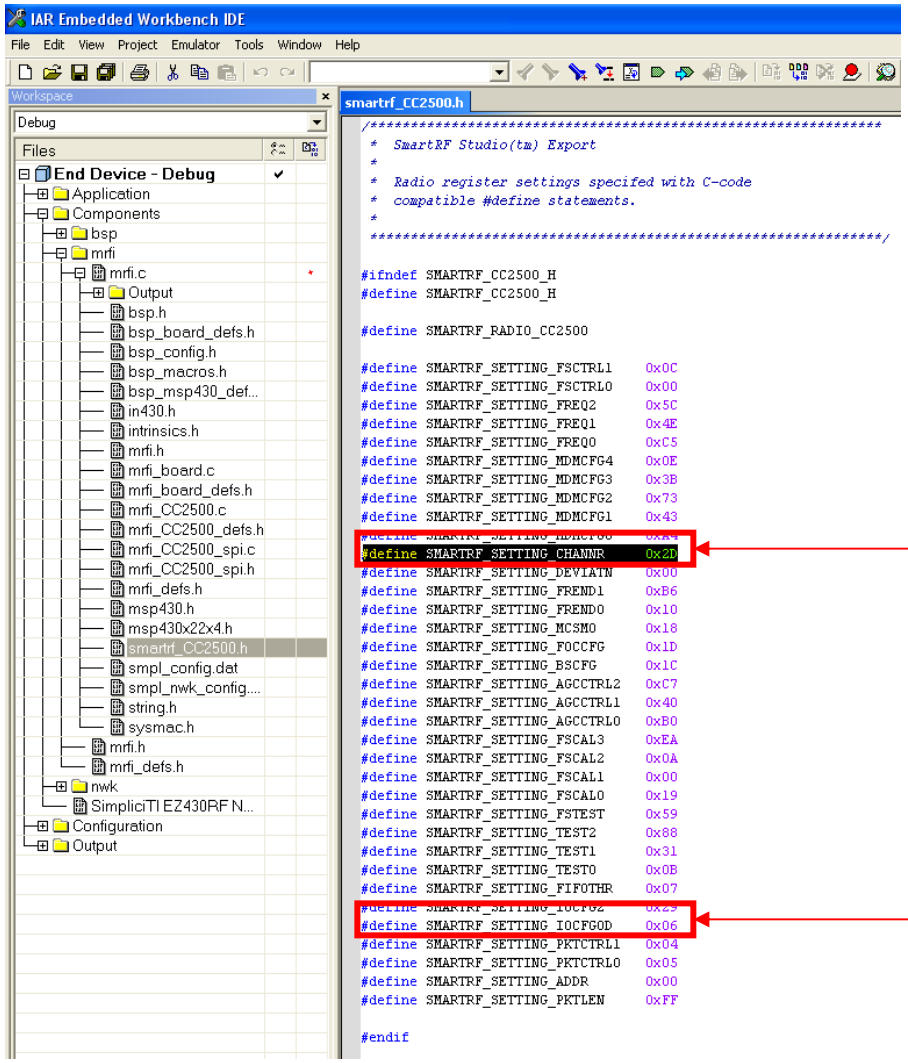
Copy to clipboard Write to file Filename: smartrf_CC@CHIPID@.h Close

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?