Application Report
SLOA160–August 2011

Contactless NFC Bootstrap Loader (BSL) Using MSP430 and TRF7970A

Josh Wyatt, Eddie LaCost
Luis Reynoso

Embedded RF Applications
MSP430 Applications

ABSTRACT

Near Field Communication, or NFC, is a technology which allows contactless communication between devices providing end users with an easier and more convenient way to make transactions.

Due to its low power, inherent security, interoperability, versatility, and low cost, it is a technology that is increasingly growing in popularity, covering and expanding to areas like consumer electronics, healthcare, electric meters, eCommerce, identity, transport, etc.

The integration of NFC in next-generation cell phones, laptops, and tablets will provide end users and developers with a complete new range of possibilities to explore, and Texas Instruments provides the tools to help developers go to market faster and easier.

One of the areas of interest during development of such applications is the capability to upgrade firmware in the field, and the purpose of this application report is to explain the implementation of a bootstrap loader (BSL) based on the MSP430F5xx/F6xx devices.

Application collateral and the software described in this document can be downloaded from http://www.ti.com/lit/zip/sloa160.

Contents

1 Near Field Communication (NFC) 3
2 MSP430 BSL 4
3 System Implementation 5
4 Running the Demo 9
5 Software Description 12
6 Summary 17
7 References 17

List of Figures

1 MSP430 BSL Main Sections 4
2 NFC BSL Hardware Implementation 5
3 NFC Communication 6
4 BSL Execution Flow Diagram 7
5 MSP-EXP430F5529 Running NFC BSL Tester 9
6 MSP-EXP430F5438 Running NFC BSL 10
7 Experimenter Boards Connected for NFC Transfer Demo 10
8 LCD Screens Showing Auto Connect Process 11
9 Successful Download to Target Device 11
10 NFC PI Modules 12
11 NFC Protocol State Machine 13
12 Selecting NFC BSL Configuration in IAR 14
13 BSL Tester Project in IAR 15
14 Selecting the BSL Configuration in CCS 15

Copyright © 2011, Texas Instruments Incorporated

15 FileMaker 16

List of Tables

1 NFCIP-1 Communication Roles 3
2 NFC Communication Modes 3
3 NFC Packet Format 7
4 DEP_REQ Format 7
5 BSL Core Command Format 8
6 NFC BSL Interface Specific Commands 8
7 NFC BSL Configuration #defines 13
8 Default Images Included in BSL Tester 16

Copyright © 2011, Texas Instruments Incorporated

1 Near Field Communication (NFC)

Near Field Communication (NFC) is a combination of existing and similar technologies all using near (magnetic) field operating at 13.56 MHz, with data rates, downlink and uplink modulation methods and schemes varying according to the base standard (ISO/IEC 15693 or ISO/IEC 14443) being applied.

It is an open platform technology, first described as NFCIP-1 (Near Field Communication Interface and Protocol 1) and standardized in ISO/IEC 18092, ECMA 340, and ETSI TS 102 190. The latest addition to the standards is NFCIP-2, which is documented in ISO/IEC 21481, ECMA-352 and ETSI TS 102 312.

While this document is not intended to provide a full description of NFC, the following subsections explain some basic concepts to provide users with basic background information.

1.1 Communication Roles

NFCIP-1 defines the roles shown in Table 1 during NFC peer-to-peer communication.

Table 1. NFCIP-1 Communication Roles

Communication Role	Description
Initiator	Device generating the RF field and starter of the communication
Target	Device receiving the Initiator's communicating request and replying back

1.2 Communication Modes

An NFC device can operate in either of the following modes:

- **Active**: The device generates its own RF field
- **Passive**: The device uses the inductive coupling to transmit data

  In passive mode, a device can be powered by the RF field of an active device, thus allowing for very low power or battery-less applications, such as tag emulation for ticketing/eCommerce or communication with cell phones which are turned off.

Considering these two concepts, the communication between two NFC devices can be classified as shown in Table 2.

Table 2. NFC Communication Modes

Communication Mode	Description
Active	Two active devices communicate with each other alternating the RF field to send data
Passive	Communication takes place between an active and a passive device. The active devices provides the RF field and the passive Target responds by modulating the existing field

NOTE: Communication cannot be established with two passive devices and Initiator cannot act in passive mode until put in that mode. The NFC BSL implements *Passive* communication mode by default, which allows for lower current consumption on the Target.

1.3 Data Rate

While NFC operates at 13.56 MHz, the communication rate between devices can vary. The NFC-BSL described in this document supports the following data rates:

- 106 kbps (default)
- 212 kbps
- 424 kbps

Copyright © 2011, Texas Instruments Incorporated

2 MSP430 BSL

The MSP430 BSL is an enabling tool provided on-chip when shipped from TI, which allows users to communicate with the embedded memory of the MSP430 during prototyping, final production, and/or while in service.

While the BSL is read-only on some MSP430 devices (accessible via UART), the MSP430F5xx and MSP430F6xx families' BSL is placed in a protected region of flash memory. This not only allows the BSL to remain protected during a mass erase of "user" flash, but it also allows for BSL customization and provides access to features such as:

- Use of different interfaces and protocols, such as UART, USB, SPI, I2C, NFC, and sub-1GHz.
- Different BSL entry methods, such as entering BSL on a particular button press.
- Additional robustness features, such as adding a CRC check of the programmed contents.
- New or custom commands as required by the particular application, such as changing the NFC data rate.

The BSL software is divided into three main sections (see Figure 1):

- **Peripheral Interface (PI)**: Responsible for receiving and verifying a BSL core command. It includes the low-level implementation of the communication interface (SPI communication to the TRF7970A that implements NFC).
- **Command Interpreter**: Interprets the BSL core command bytes supplied by the PI.
- **BSL API**: Provides an abstraction layer between the command interpreter and the memory being written to or read from. It implements the flash driver, which unlocks, writes to, and reads from the flash.

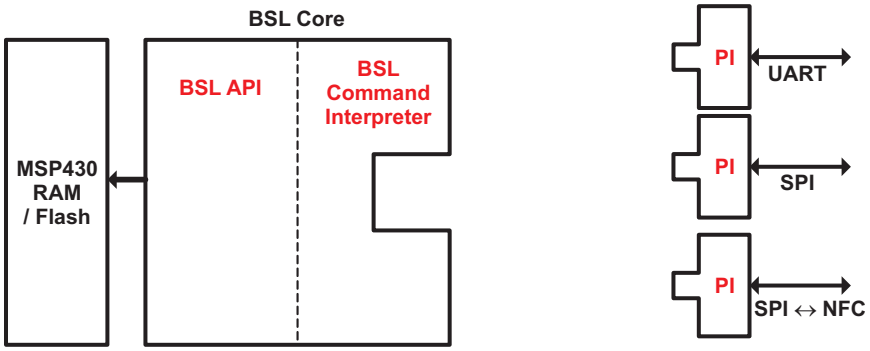


Figure 1. MSP430 BSL Main Sections

More information about the BSL can be found in these documents:

- *MSP430 Programming Via the Bootstrap Loader* (SLAU319)
- *Creating a Custom Flash-Based Bootstrap Loader (BSL)* (SLAA450)

Copyright © 2011, Texas Instruments Incorporated

3 System Implementation

This application report shows the hardware and software implementation of an NFC BSL for both the Target and an Initiator.

The Initiator is implemented with a MSP430F5529 while the Target device is a MSP430F5438A.

The MSP430F5529 is an ultra-low-power MSP430 with 128KB of flash, 8+2KB of RAM, integrated USB and PHYs supporting USB 2.0, four 16-bit timers, a high-performance 12-bit analog-to-digital converter (ADC), two universal serial communication interfaces (USCI), hardware multiplier, DMA, RTC, and 63 I/O pins.

The MSP430F5438A is an ultra-low-power MSP430 with 256KB of flash, 16KB of RAM, three 16-bit timers, a high performance 12-bit ADC, four universal serial communication interfaces (USCI), hardware multiplier, DMA, RTC, and 87 I/O pins.

Experimenter's boards for both microcontrollers (MSP-EXP430F5529 and MSP-EXP430F5438) are used for the demo, aided by TRF7970ATB boards that provide access to the TRF7970A RFID radio.

The TRF7970A is an integrated analog front end and data framing system for 13.56-MHz RFID/NFC systems. It can operate in multiple modes including RFID reader/writer, NFC peer to peer (NFCIP-1 compliant), and as a card emulator for ISO14443A/B.

Figure 2 shows the hardware implementation of the demo.

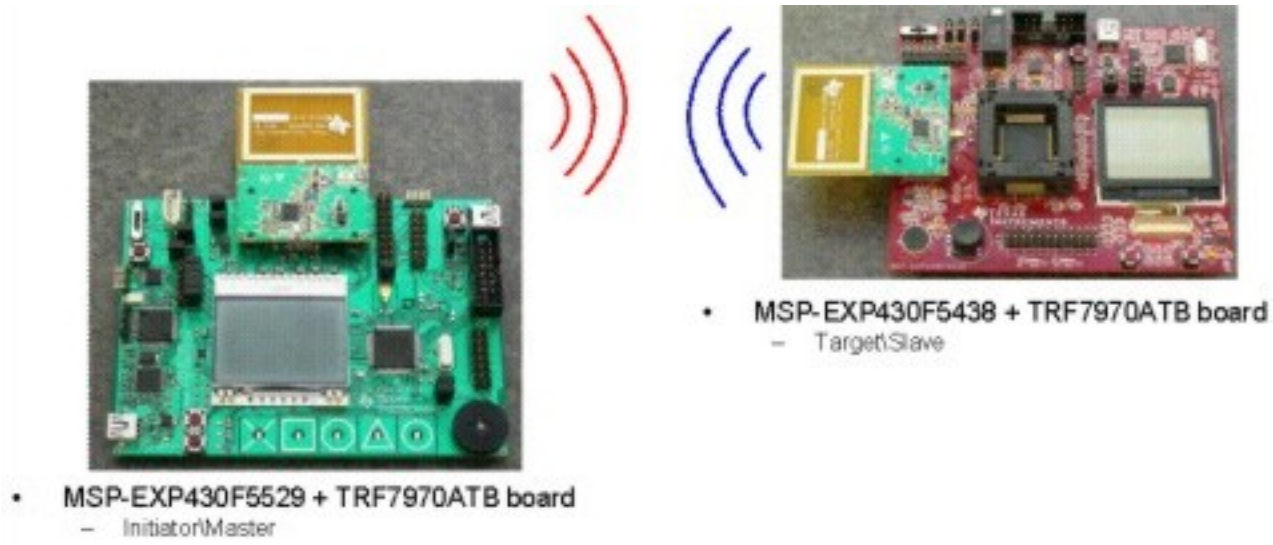


Figure 2. NFC BSL Hardware Implementation

While the provided software was implemented in the microcontrollers mentioned above, the migration effort to a different MSP430 is minimal. However, note that flash BSL is available only in MSP430F5xx and MSP430F6xx devices.

3.1 NFC Implementation

Considering the limitation of 2KB provided by the MSP430 BSL flash memory area, the NFC BSL has a limited implementation of the NFC protocol, allowing for simple transfers.

The device starts in Target passive mode at 106 kbps and makes use of the hardware Single Device Detection (SDD) feature of the TRF7970A to reduce the memory footprint during the handshaking stage.

Attribute Request (ATR_REQ) and Attribute Response (ATR_RES) frames close the handshaking process, sharing the identifiers of both devices.

Package transfer is performed using Data Exchange Protocol Requests (DEP_REQ) sent by the Initiator, and Data Exchange Protocol Responses (DEP_RES) sent by the Target (see Figure 3).

Copyright © 2011, Texas Instruments Incorporated

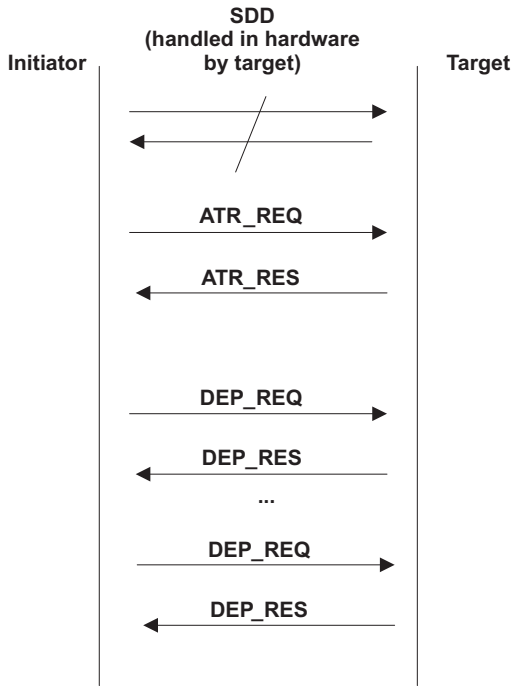


Figure 3. NFC Communication

3.2 BSL Implementation

The communication shown in Section 3.1 reduces the footprint of NFC BSL allowing for a minimal implementation of BSL, but it is not enough to allow the implementation of the full BSL protocol plus NFC in the reserved 2KB of flash.

Following the implementation of similar BSLs implemented for MSP430, such as for USB, the NFC BSL uses a "RAM Write-only" version of the BSL, known in this document as "MiniBSL", to load the full BSL to RAM and execute from there (see Figure 4).

Copyright © 2011, Texas Instruments Incorporated

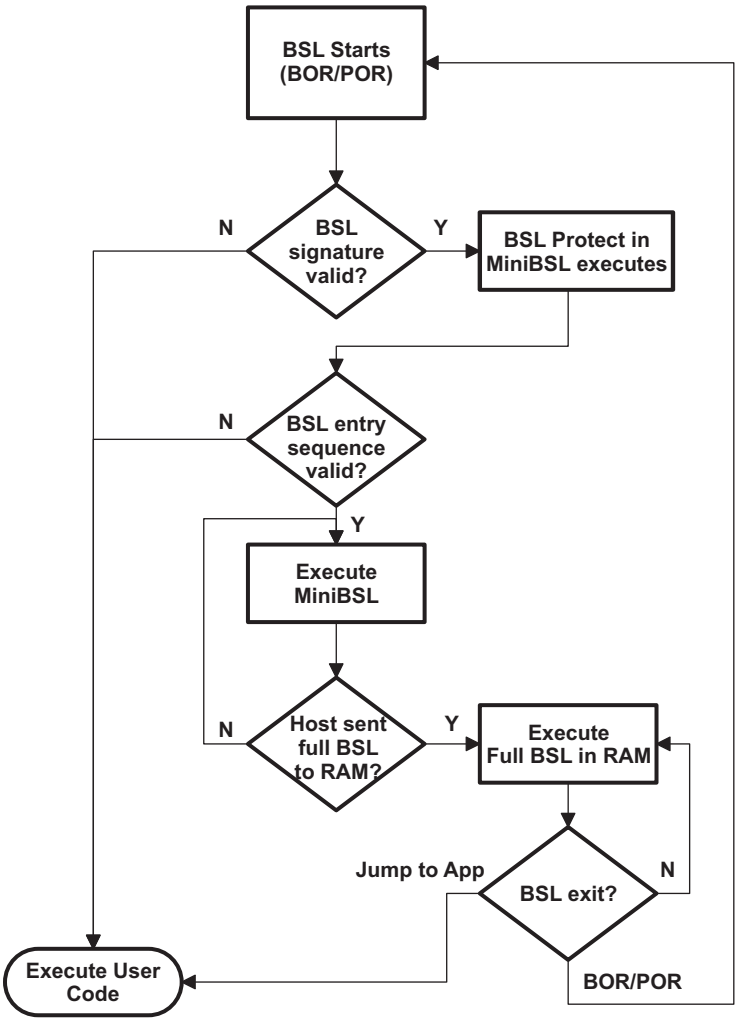


Figure 4. BSL Execution Flow Diagram

Note that the MiniBSL is always available in the protected Flash area, so there's no risk of losing the microcontroller; and the full BSL is loaded to and executes from RAM temporarily, meaning that application can still use the whole RAM space.

3.2.1 BSL NFC PI Implementation

The NFC BSL Peripheral Interface has the format shown in Table 3.

Table 3. NFC Packet Format

Preamble	SYNC	Length	NFC Payload	CRC

Note that the TRF7970A handles all the shaded fields and the NFC Payload is handled by the NFC layer.

As mentioned in Section 3.1, data is transferred using DEP_REQ packets, which have the format shown in Table 4.

Table 4. DEP_REQ Format

0	1:2	3	4	5	6+
LEN	CMD (0xD406)	PFB	DID	NAD	BSL_Payload

Copyright © 2011, Texas Instruments Incorporated

A detailed description of NFC-related fields is found in NFCIP-1, but note that the BSL payload is passed to the BSL Command Interpreter following the format described in SLAU319 as shown in Table 5.

Table 5. BSL Core Command Format

CMD	AL	AM	AH	Data

All of the core commands described in SLAU319 are supported in NFC BSL with the addition of the interface-specific command shown in Table 6.

Table 6. NFC BSL Interface Specific Commands

BSL Command	CMD	AL	AM	AH	Data
Change Data Rate	0x52	-	-	-	D1(1)

(1) D1 can have the following values:
- 0x01: 212 kbps
- 0x02: 424 kbps
- Other values: 106 kbps

Copyright © 2011, Texas Instruments Incorporated

4 Running the Demo

Follow these steps when running the default demo for the first time:

1. Connect TRF7970ATB board to the RF sockets in both the MSP-EXP430F5438 and MSP-EXP430F5529 boards.
2. Make sure the boards are powered properly (battery, USB, JTAG, etc.) and that the RF boards get power through RF_PWR.
3. Build the MiniBSL project for the Target device as explained in Section 5.1.2.
4. Flash the MiniBSL to MSP-EXP430F5438.
5. Build the NFC BSL Tester project for the Initiator as explained in Section 5.2.1.
6. Flash the MSP-EXP430F5529 with the NFC BSL Tester image.
7. Reset the MSP-EXP430F5529 board which should execute the Initiator demo as shown in Figure 5.

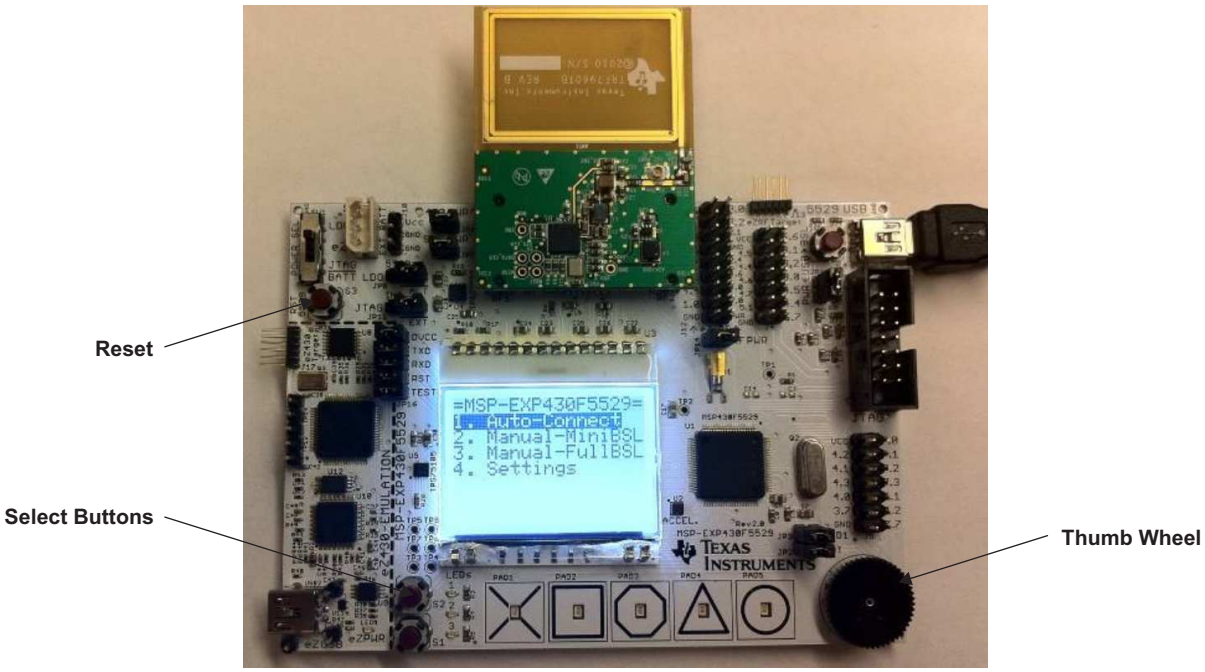


Figure 5. MSP-EXP430F5529 Running NFC BSL Tester

The Initiator demo uses the LCD, select buttons, and thumb wheel for easier control and display. The initial screen shows the following options:

- Auto-Connect: Attempts to connect to the Target Mini-BSL and, if successful, downloads the full RAM BSL and then attempts to connect to it in a single step. If successful, offers the option to send some code images.
- Manual-MiniBSL: Attempts to connect to the MiniBSL and, if successful, offers the option to send single commands in MiniBSL mode.
- Manual-FullBSL: Attempts to connect to the Full RAM BSL and, if successful, offers the option to send single commands in Full BSL mode.
- Settings: Settings menu from MSP-EXP430F5529 User's Experience code. Can be used to adjust LCD Contrast and Backlight.

8. Press and hold S1 while pressing the Reset button in the MSP-EXP430F5438. This forces the part to go into BSL mode during the entry sequence. The MSP-EXP430F5438 does not show any signs of activity, but it does wait for the Initiator's command (see Figure 6).

Copyright © 2011, Texas Instruments Incorporated

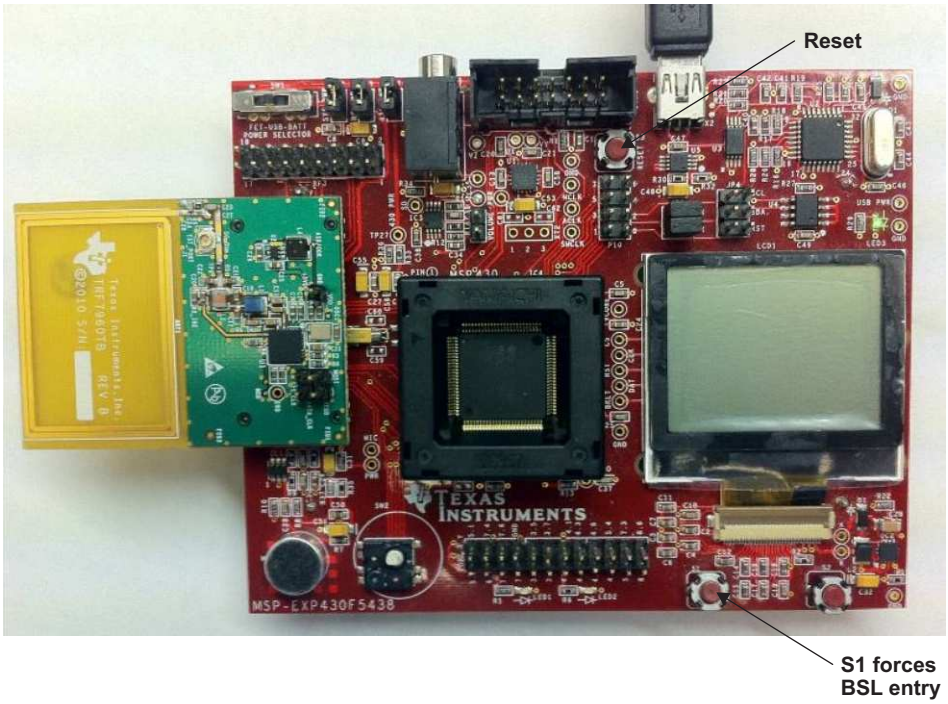


Figure 6. MSP-EXP430F5438 Running NFC BSL

9. Attach the TRF7970ATB boards of both experimenters board as shown in Figure 7.

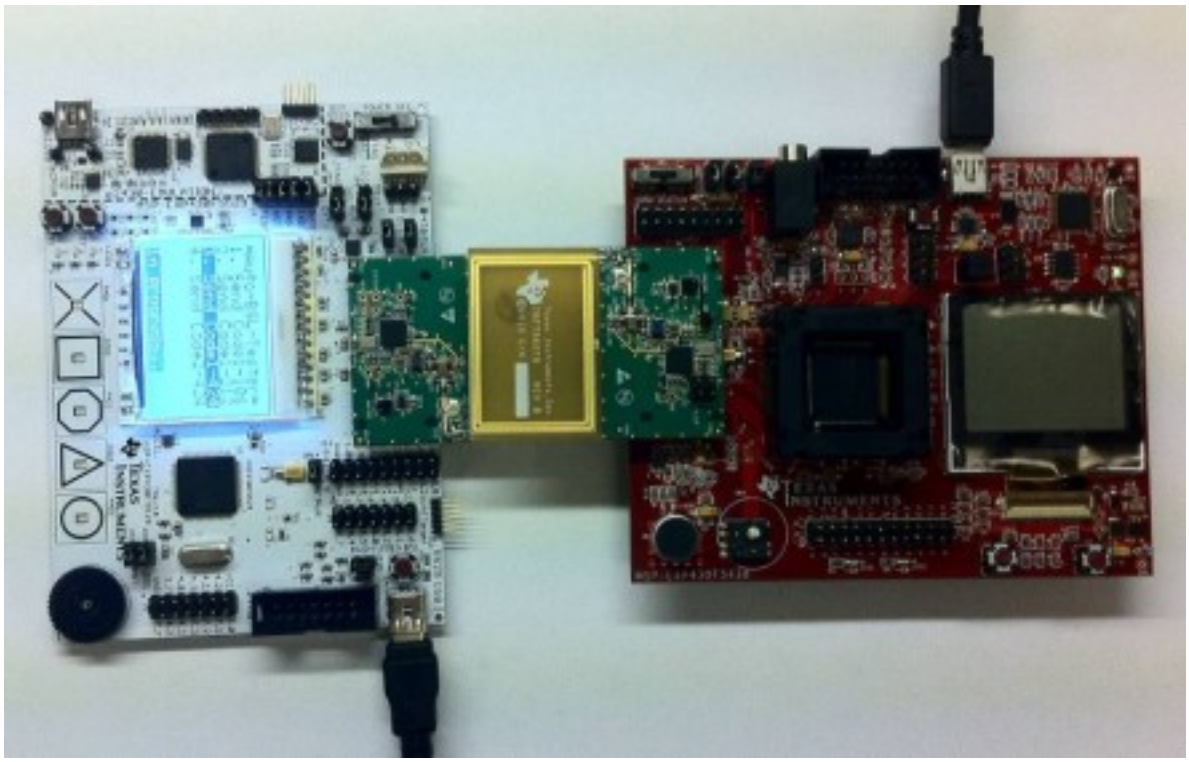


Figure 7. Experimenter Boards Connected for NFC Transfer Demo

10. Using the MSP-EXP430F5529 wheel and select buttons, scroll to option 1. Auto-Connect (see Figure 8).

    The Initiator tries to establish connection with the Target and, if successful, sends the Full RAM BSL

Copyright © 2011, Texas Instruments Incorporated

image. It then sends a BSL command asking for the version number, which verifies that the device is executing the Full RAM BSL image.

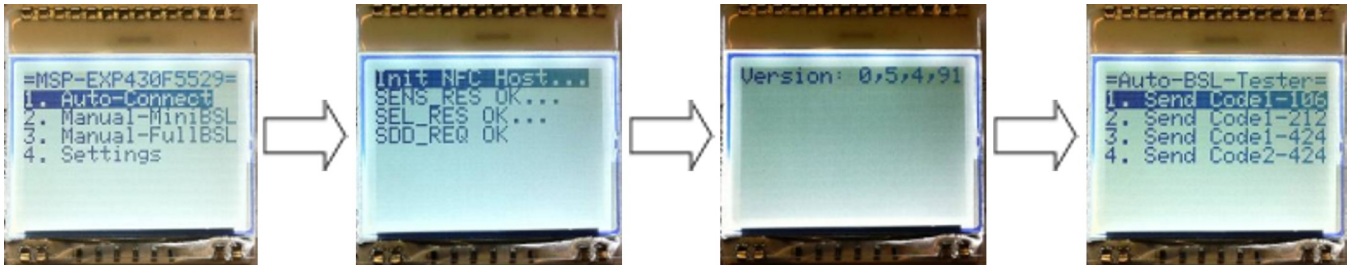


Figure 8. LCD Screens Showing Auto Connect Process

11. Select an image to send: Code1 downloads the MSP-EXP430F5438 User Experience code, while Code2 downloads a demo using the TRF7970A to read ISO15693 tags. Different data rates can be selected for Code1 for demo purposes.
12. The MSP-EXP430F5438 executes the new application if the transfer is successful (see Figure 9).

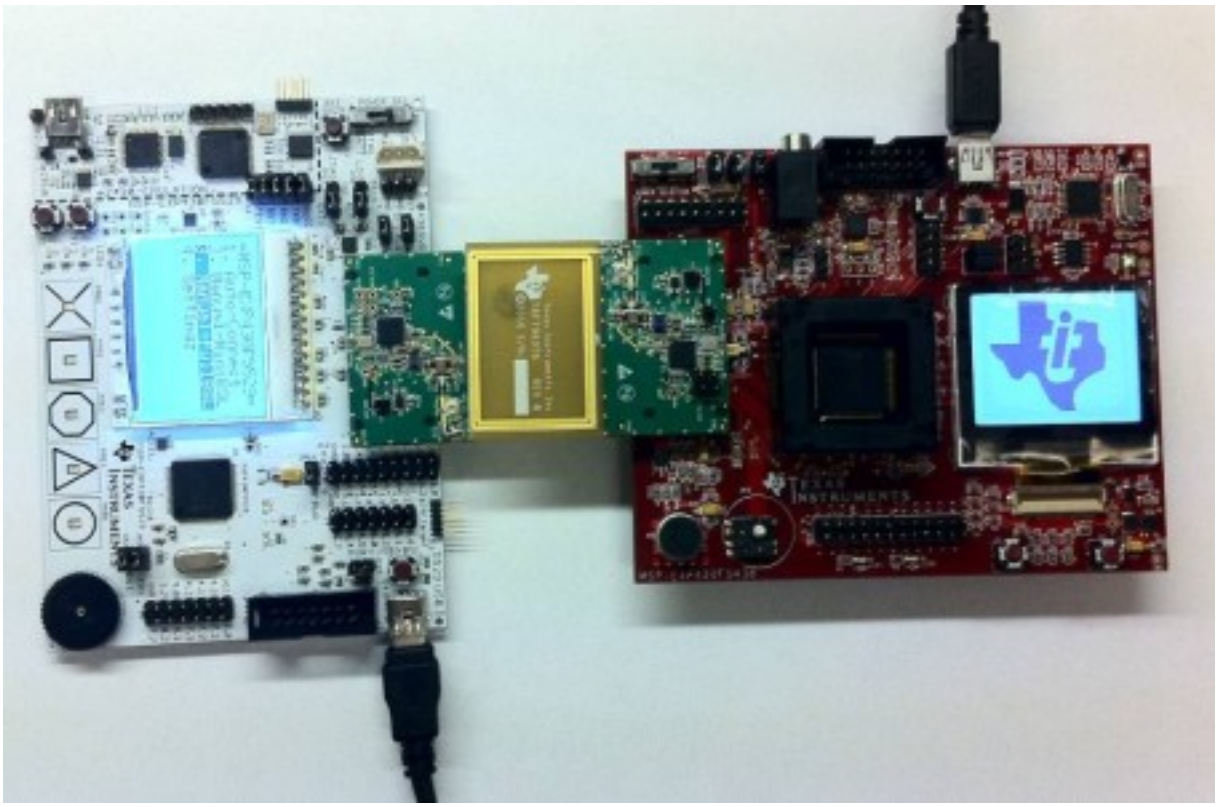


Figure 9. Successful Download to Target Device

Copyright © 2011, Texas Instruments Incorporated

5 Software Description

Software comprises two projects:

- BSL_NFC: Implementation of NFC-BSL for the MSP430F5438A acting as a passive Target device.
- NFC-BSL_tester: Implementation of the BSL Initiator using a MSP430F5529. The Initiator can reprogram the Target device with application firmware.

5.1 BSL_NFC

The BSL NFC is based on the MSP430 BSL code described in Section 2.

To implement NFC communication, a new Peripheral Interface (PI) was implemented to use SPI in master mode communicating with the TRF7970A transceiver. The BSL's Command Interpreter and API do not require changes.

5.1.1 NFC Peripheral Interface (PI)

The NFC PI has three layers:

- SPI Driver: Low-level SPI driver
- RFID HAL: Driver interfacing with TRF7970A transceiver
- NFC: NFC protocol implementation

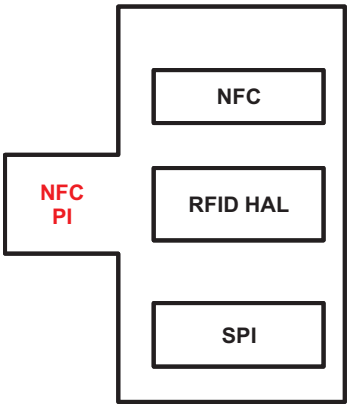


Figure 10. NFC PI Modules

5.1.1.1 SPI Driver

The SPI driver is a simple driver that uses USCI_B0 and polling mode. BSL drivers are restricted from using interrupts, because interrupt vectors are erased and written for application purposes.

5.1.1.2 RFID HAL

This module contains a driver to handle the TRF7970A transceiver. Its most important functionality is to detect when the radio signals an IRQ interrupt using polling, and it checks the source of such interrupt.

5.1.1.3 NFC

The NFC module handles the NFC protocol following the state machine shown in Figure 11.

Copyright © 2011, Texas Instruments Incorporated

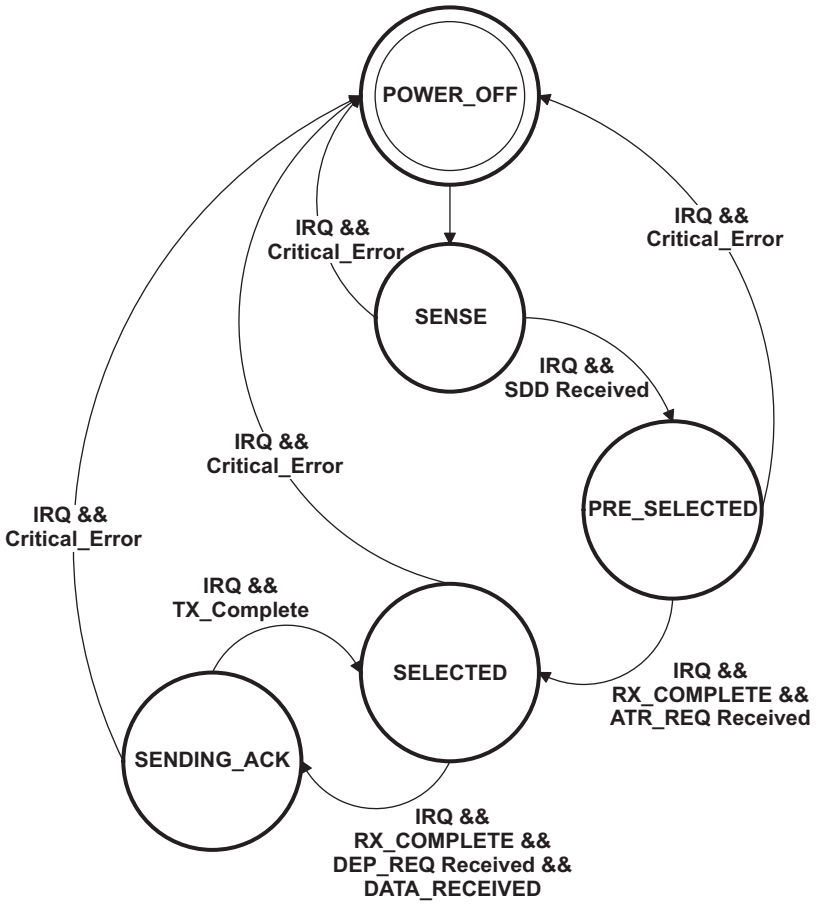


Figure 11. NFC Protocol State Machine

Note that this is a limited NFC implementation. It handles SDD requests in hardware, uses ATR_REQ/ATR_RES to complete the handshake process, and uses DEP_REQ and DEP_RES to exchange data packets, which include the BSL payload.

5.1.2 Building and Customizing the NFC BSL

The NFC BSL source code and an IAR project are included with this application report.

The default demo can be built in IAR by opening this workspace: ~\BSL_NFC\BSL.eww

As explained in Table 7, two binaries are needed to run the NFC BSL:

- MiniBSL: RAM-write-only limited BSL residing in Flash BSL area
- Full RAM BSL: Full BSL sent by the initiator, loaded to and executed from RAM

The MSP430 BSL uses predefined macros in BSL_Device_File.h to configure the BSL as shown in Table 7.

Table 7. NFC BSL Configuration #defines(1)

	RAM_WRITE_ONLY_BSL	FULL_FLASH_BSL	RAM_BASED_BSL
Full RAM BSL	✓	✗	✗
MiniBSL	✗	✓	✓

(1) ✓ = Defined, ✗ = Undefined

Copyright © 2011, Texas Instruments Incorporated

To facilitate development and testing, the IAR project includes two build configurations allowing for an easier selection. These can be selected by clicking in the window shown in Figure 12.

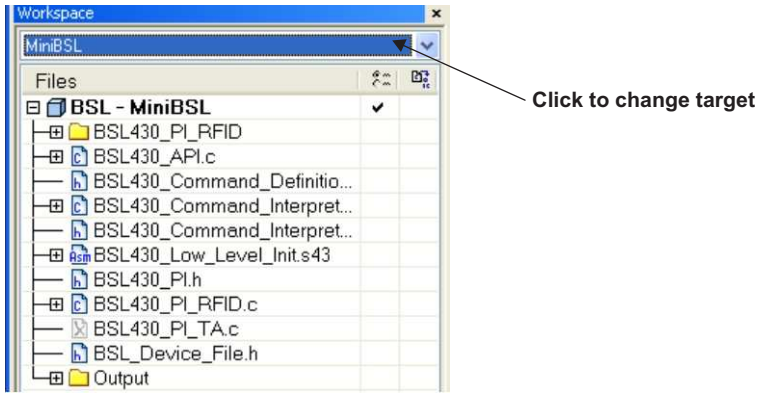


Figure 12. Selecting NFC BSL Configuration in IAR

After selecting the proper configuration, build the application using IAR, and the corresponding binary file is generated.

These binary files can be loaded as follows:

- MiniBSL can be loaded directly to flash using the IDE and JTAG, because the MiniBSL configuration is already configured to unlock the BSL area.
- The Full_RAM BSL configuration generate an output here: ~\BSL_NFC\Full_RAM\Exe\BSL_RAM.txt

  This file is used by the Initiator to send the full RAM BSL to the Target device. Follow the steps described in Section 5.2.2 to load a new Full RAM BSL to the NFC-BSL Tester.

NOTE: Source code for NFC BSL can be modified and adapted according to the application requirements. Changes to the source code might affect one or both configurations and binaries should be rebuilt following the procedure described above to apply the changes.

5.2 NFC_BSL_Tester

This application report includes sample code for an NFC Initiator device that can be used to reprogram the NFC Target device.

The code is based on the MSP-EXP430F5529 User Experience code, which contains drivers and sample code for multiple functions such as USB, LCD, CapTouch, and SmartCard.

The demo adds drivers to act as an NFC Active Initiator using the TRF7970A through the SPI interface. It uses existing LCD, buttons, and wheel drivers to provide easy access to commands, status, and data.

5.2.1 Building and Customizing the NFC BSL Tester

Source code and projects for both CCS and IAR are included in this application report. These files can be downloaded from http://www.ti.com/lit/zip/sloa160.

5.2.1.1 IAR

The default demo can be built in IAR by opening this workspace:
~NFC_BSL_Tester\IAR\NFC_BSL_Tester.eww

Copyright © 2011, Texas Instruments Incorporated

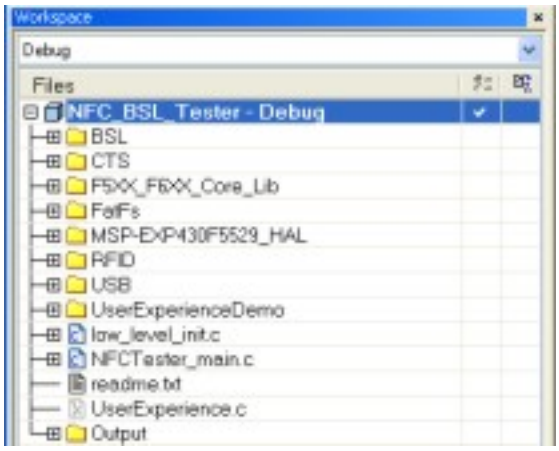


Figure 13. BSL Tester Project in IAR

The project can be built and loaded to the Target device using the same IAR IDE and a JTAG/ezFET.

5.2.1.2 CCS

The project can be imported to CCS from this folder: ~NFC_BSL_Tester\CCS\

The Active configuration can be built then by right-clicking the project and selecting the "BSL Tester" Target as shown in Figure 14.

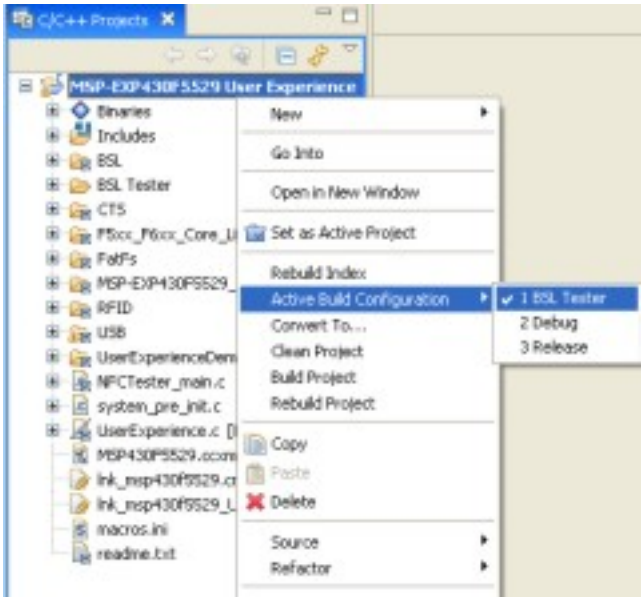


Figure 14. Selecting the BSL Configuration in CCS

The project can be built and loaded to the Target device using the same CCS IDE and a JTAG/ezFET.

5.2.2 Adding a New Image to the BSL Tester

The BSL Tester project includes a pre-built image of the Target's Full RAM BSL (see Section 5) and two application examples that can be sent to the Target device for demo purposes (see Table 8).

Copyright © 2011, Texas Instruments Incorporated

Table 8. Default Images Included in BSL Tester

File/Array	Description
bsl_ram_image.h Array: BSLCodeArray[]	Full RAM BSL for MSP-EXP430F5438. Sent from Initiator to Target device to execute full BSL functionality.
MSP-EXP430F5438_User_Experience.h Array: CodeArray1[]	MSP-EXP430F5438 User's experience demo. Sent from Initiator to Target as Code Example 1.
TRF7970_Demo_EXP543xA_IAR.h Array: CodeArray2[]	Demo using MSP-EXP430F5438 with TRF7970A reading tags. Sent from Initiator to Target as Code Example 2.

Changes to the BSL RAM image or new demos can be loaded following these steps:

1. Configure your IAR/CCS project to generate an MSP430-TXT output following the steps described in this link:

   http://processors.wiki.ti.com/index.php/Generating_and_Loading_MSP430_Binary_Files
2. Run FileMaker.exe, which is installed to this directory: ~\NFC_BSL_Tester\BSL\FileMaker.exe

   Or, it can also be found in the zip file that accompanies the *MSP430 Programming Via the Bootstrap Loader (BSL) User's Guide* (slau319.zip).
3. Use FileMaker.exe to convert the .txt output to .h.

   Note that FileMaker.exe uses CodeArray to name the output array. Rename the array as needed.

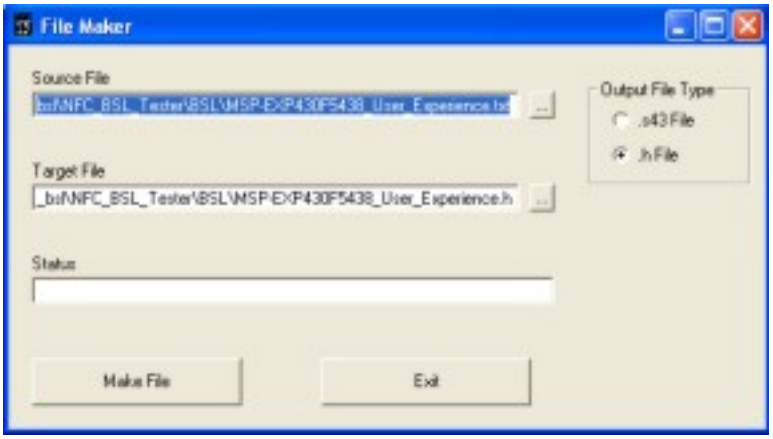


Figure 15. FileMaker

4. Copy the .h file to this folder: ~\NFC_BSL_Tester\BSL\
5. Include the .h file in BSL_App_cmd.c following the same approach used for default images
6. Use BSL_SendCode( ) to send the code to the Target following the same approach used for the default images.

Copyright © 2011, Texas Instruments Incorporated

6 Summary

NFC is a technology that is opening new possibilities in diverse markets. This application report shows the implementation of a small NFC bootstrap loader that can be used to upgrade firmware, change functionality, and communicate with an NFC-enabled device wirelessly.

A demo that can be built using readily available TI tools and software for both the Target and Initiator is provided as a guide to help the design of an end-user application.

7 References

- *MSP430x5xx/MSP430x6xx Family User's Guide* (SLAU208)
- TRF7970A data sheet (SLOS743)
- MSP430F5438A data sheet (SLAS655)
- MSP430F5529 data sheet (SLAS590)
- *MSP-EXP430F5438 User's Guide* (SLAU263)
- *MSP-EXP430F5529 User's Guide* (SLAU330)
- *MSP430 Programming Via the Bootstrap Loader User's Guide* (SLAU319)
- *Creating a Custom Flash-Based Bootstrap Loader (BSL)* (SLAA450)
- Generating and Loading MSP430 Binary Files (http://processors.wiki.ti.com/index.php/Generating_and_Loading_MSP430_Binary_Files)
- International Organization for Standardization (http://www.iso.org/iso/home.html)
- ECMA-340 (http://www.ecma-international.org/publications/standards/Ecma-340.htm)
- ECMA-352 (http://www.ecma-international.org/publications/standards/Ecma-352.htm and http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-352.pdf)

Copyright © 2011, Texas Instruments Incorporated

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com	Wireless	www.ti.com/wireless-apps
RF/IF and ZigBee® Solutions	www.ti.com/lprf		
	TI E2E Community Home Page		e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2011, Texas Instruments Incorporated