


Bootstrap Loader (BSL) Scripter 3.0.1.0

User's Guide

Copyright

Copyright © 2015 Texas Instruments Incorporated. All rights reserved. MSP430 and MSP432 are registered trademarks of Texas Instruments. Other names and brands may be claimed as the property of others.

 Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this document.

Texas Instruments
Post Office Box 655303
Dallas, TX 75265
<http://www.ti.com/msp430>



Revision Information

This is version 1 of this document, last updated on 09-February-2015

This is version 2 of this document, last updated on 06-August-2015


Table of Contents

Copyright	2
Revision Information	2
1 Introduction	5
1.1 Script File Format	5
1.2 BSL Scripter Usage	5
2 BSL Scripting Language	7
2.1 MODE	8
2.2 CHANGE_BAUD_RATE	10
2.3 CRC_CHECK	11
2.4 CRC_CHECK_32	11
2.5 DELAY	12
2.6 ERASE_SEGMENT	13
2.7 ERASE_SEGMENT_32	13
2.8 SET_PC	14
2.9 SET_PC_32	14
2.10 MASS_ERASE	15
2.11 REBOOT_RESET	16
2.12 RX_DATA_BLOCK	17
2.13 RX_DATA_BLOCK_32	18
2.14 RX_DATA_BLOCK_FAST	19
2.15 RX_PASSWORD	20
2.16 RX_PASSWORD_32	21
2.17 TOGGLE_INFO	22
2.18 TX_DATA_BLOCK	23
2.19 TX_DATA_BLOCK_32	24
2.20 TX_BSL_VERSION	25
2.21 TX_BSL_VERSION_32	26
2.22 TX_BUFFERSIZE	27
2.23 VERBOSE	28
IMPORTANT NOTICE	30

1 Introduction

The BSL Scripter is a command line tool to communicate with the Bootstrap Loader (BSL) on a MSP430® and MSP432®.

The application serves as a device programmer, a starting point for a custom BSL application (source code is included) and as a reference on how to use the BSL protocol (as sent and received data can be observed via the verbose mode).

 The BSL Scripter does not support BSLs of the 1xx/2xx and 4xx device family. To communicate with these devices use the 'BSLDEMO2.exe' command line tool found in the 'Deprecated' folder.

1.1 Script File Format

The script file is a ASCII text file. The BSL Scripter reads this text file to parse the BSL commands. Each line must contain a single BSL command with no preceding spaces. Empty lines are allowed. Some commands may have optional parameters and some may have mandatory parameters. Please be sure and check the section below for details on the BSL scripting language. Parameters in curly brackets need to be substituted by the corresponding value. Lines can be commented out using C-style '/' as first two characters.

1.2 BSL Scripter Usage

The BSL Scripter can be started from the command line by typing the application name then typing the name of the main-script-file to be read. The main-script-file directory is relative to the BSL-scripter.exe location.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the executable-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the executable-file
- Up folder: file-to-be-read is located one(or more) folder up relative the executable-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (Windows)

- Same folder: `BSL-Scripter.exe script_file.txt`
- Sub folder: `BSL-Scripter.exe SubFolder\script_file.txt`
- Up folder: `BSL-Scripter.exe ..\script_file.txt`
- Absolute folder: `BSL-Scripter.exe C:\Data\ScriptFolder\script_file.txt`

Example (Linux and Mac)

- Same folder: `BSL-Scripter.exe script_file.txt`
- Sub folder: `BSL-Scripter.exe SubFolder/script_file.txt`
- Up folder: `BSL-Scripter.exe ../script_file.txt`

- **Absolute folder:** `BSL-Scripter.exe /home/Data/ScriptFolder/script_file.txt`

When the `script_file.txt` is not inserted, the program will ask the file name after it is executed.

2 BSL Scripting Language

The following subsections describe the available commands.

Different Device families support different commands. Table 2.1 lists the commands supported by each device family.

Table 2.1: BSL commands per device family

	5xx/6xx	FRxx	P4xx
MODE	✓	✓	✓
CHANGE_BAUD_RATE	✓	✓	✓
CRC_CHECK	✓	✓	✓
CRC_CHECK_32			✓
DELAY	✓	✓	✓
ERASE_SEGMENT	✓		✓
ERASE_SEGMENT_32			✓
SET_PC	✓	✓	✓
SET_PC_32			✓
MASS_ERASE	✓	✓	✓
REBOOT_RESET			✓
RX_DATA_BLOCK	✓	✓	✓
RX_DATA_BLOCK_32			✓
RX_DATA_BLOCK_FAST	✓	✓	✓
RX_PASSWORD	✓	✓	
RX_PASSWORD_32			✓
TOGGLE_INFO	✓		
TX_DATA_BLOCK	✓	✓	✓
TX_DATA_BLOCK_32			✓
TX_BSL_VERSION	✓	✓	
TX_BSL_VERSION_32			✓
TX_BUFFER_SIZE	✓	✓	
VERBOSE	✓	✓	✓

2.1 MODE

Command

MODE {FAMILY} {PROTOCOL} {BAUDRATE} {COM}

Description

Initializes the selected communication channel for a BSL session and invokes the BSL. This command also tells the PC side engine which communication protocol should be used. Additionally, if a MSP432Pxx device is used the initial UART baud rate can be selected. (For all other devices use the CHANGE_BAUD_RATE command to select a different UART baud rate.)

Parameters

FAMILY *[mandatory]*:

- 543x_family or msp430f543x : Indicates communication with a BSL on the following devices:
 - MSP430F5418 / MSP430F5419
 - MSP430F5435 / MSP430F5436
 - MSP430F5437 / MSP430F5438
 - NOTE: 54xxA devices are handled as '5xx'
- 5xx or msp430f5xx : Indicates communication with all other 5xx devices
- 6xx or msp430f6xx : Currently identical to '5xx' and can be used interchangeably
- FRxx or msp430frxx: Indicates communication with FRAM devices
- P4xx or msp432p4xx: Indicates communication with MSP432P4xx devices

PROTOCOL *[mandatory]*:

- UART: standard communication on most MSP430 devices, default baud rate: 9600 baud
- I2C: default baud rate: 100000 bit/s
- SPI: default baud rate: 125000 Hz
- NOTE: Due to downwards compatibility UART protocol is chosen if no protocol is given

BAUDRATE *[optional - MSP432P4xx only]*

- for UART:
 - 1200
 - 2400
 - 4800
 - 9600
 - 19200
 - 38400
 - 57600
 - 115200
- for I2C:
 - 100000
 - 400000
- for SPI:
 - 125000
 - 250000

- 1000000

COM [*mandatory*]:

- COM{x}: Indicates the PC COM port that is used, i.e. COM42
- USB: Indicates that communication will be done via USB

Examples

- MODE 543x_family COM42
- MODE msp430f543x COM42
- MODE 5xx UART COM42
- MODE msp430f5xx UART COM42
- MODE 5xx USB
- MODE msp430f5xx USB
- MODE FRxx I2C COM42
- MODE msp430frxx I2C COM42
- MODE P4xx SPI 250000 COM42
- MODE msp432p4xx SPI 250000 COM42
- MODE P4xx UART 115200 COM42
- MODE msp432p4xx UART 115200 COM42

The Scripter 3.0.1.0 handles the upper and lower case alphabets.
Use the proper serial port name in Linux and Mac Os X.

Examples (Linux and Mac)

- MODE 543x_family /dev/ttyACM0
- MODE msp430f543x /dev/tty.usbmodem1451

2.2 CHANGE_BAUD_RATE

Command

CHANGE_BAUD_RATE {SPEED}

Description

Changes the baud rate of the UART communication.

Parameters

SPEED [*mandatory*]: A new baud rate, one of the following values

■ UART

- 1200
- 2400
- 4800
- 9600 (default)
- 19200
- 38400
- 57600
- 115200

■ I2C

- 100000 (default)
- 400000

■ SPI

- 125000 (default)
- 250000
- 1000000

- NOTE: Please see individual BSL descriptions in the User's Guide for information on supported baud rates and protocols.

Examples

■ CHANGE_BAUD_RATE 115200

■ CHANGE_BAUD_RATE 9600

2.3 CRC_CHECK

Command

CRC_CHECK {ADDRESS} {LENGTH} {EXPECTED}

Description

Performs a CRC check starting at the given address over the number of bytes given by length. This command will simply output the result of the CRC operation or compare the result to a supplied value and report whether there is a match or mismatch.

Parameters

ADDRESS [*mandatory*]: The address at which the CRC will begin (hex format)

LENGTH [*mandatory*]: The number of bytes to include to CRC (hex format)

EXPECTED [*optional*]: The value to compare the CRC result with

Examples

- CRC_CHECK 0x8000 0x1000 0xCFB8
- CRC_CHECK 0x8000 0x10

2.4 CRC_CHECK_32

Command

CRC_CHECK_32 {ADDRESS} {LENGTH} {EXPECTED}

Description

Identical to CRC_CHECK except that the memory is 32-byte addressed. This command works on MSP432P4xx devices only.

Parameters

ADDRESS [*mandatory*]: The address at which the CRC will begin (hex format)

LENGTH [*mandatory*]: The number of bytes to include to CRC (hex format)

EXPECTED [*optional*]: The value to compare the CRC result with

Example

- CRC_CHECK_32 0x8000 0x1000 0xCFB8
- CRC_CHECK_32 0x12345678 0x1010 0xFF31

2.5 DELAY

Command

DELAY {MS}

Description

Causes a delay of MS milliseconds.

Parameters

MS [*mandatory*]: The number of milliseconds to wait before proceeding

Example

```
■ DELAY 1000
```

2.6 ERASE_SEGMENT

Command

ERASE_SEGMENT {ADDRESS}

Description

Causes the BSL to erase the segment containing the supplied address.

Parameters

ADDRESS [*mandatory*]: An address in hex format within MSP430/MSP432 flash. The segment which contains this address will be erased.

Examples

- ERASE_SEGMENT 0x10000
- ERASE_SEGMENT 0x8000

2.7 ERASE_SEGMENT_32

Command

ERASE_SEGMENT_32 {ADDRESS}

Description

Identical to ERASE_SEGMENT except that the memory is 32-byte addressed. This command is working on MSP432 only.

Parameters

ADDRESS [*mandatory*]: An address in hex format within MSP432 flash, the segment which contains this address will be erased.

Examples

- ERASE_SEGMENT_32 0x10000
- ERASE_SEGMENT_32 0x8000

2.8 SET_PC

Command

SET_PC {ADDRESS}

Description

Sets the program counter to the given address. NOTE: This functions performs a function call to this address, so it can be returned form the call via BSL action function.

Parameters

ADDRESS [*mandatory*]: An address to which the MSP430/MSP432's Program Counter will be set and begin the program execution

Example

■ SET_PC 0x2504

2.9 SET_PC_32

Command

SET_PC_32 {ADDRESS}

Description

Identical to SET_PC except that the address will be 32-byte addressed. This command only works on MSP432 devices.

Parameters

ADDRESS [*mandatory*]: An address to which the MSP432's Program Counter will be set and begin the program execution.

Example

■ SET_PC_32 0x1FF16540

2.10 **MASS_ERASE**

Command

MASS_ERASE

Description

Causes the BSL to perform a Mass Erase.

Parameters

NONE

Example

■ **MASS_ERASE**

2.11 REBOOT_RESET

Command

REBOOT_RESET

Description

Causes the BSL to perform a Reboot Reset. This command is only working on MSP432 devices.

Parameters

NONE

Example

```
■ REBOOT_RESET
```

2.12 RX_DATA_BLOCK

Command

RX_DATA_BLOCK {FILENAME}

Description

Causes the BSL to read the supplied TI TXT file and download all data contained in this file to the MSP430/MSP432.

Parameters

FILENAME *[mandatory]*: The name of the TI TXT file to read.

The path of file that will be read is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the main-script-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-read is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (Windows)

- Same folder: `RX_DATA_BLOCK Big_File.txt`
- Sub folder: `RX_DATA_BLOCK SubFolder\Big_File.txt`
- Up folder: `RX_DATA_BLOCK ../Big_File.txt`
- Absolute folder: `RX_DATA_BLOCK C:\Data\ScriptFolder\Big_File.txt`

Example (Linux and Mac)

- Same folder: `RX_DATA_BLOCK Big_File.txt`
- Sub folder: `RX_DATA_BLOCK SubFolder/Big_File.txt`
- Up folder: `RX_DATA_BLOCK ../Big_File.txt`
- Absolute folder: `RX_DATA_BLOCK /home/Data/ScriptFolder/Big_File.txt`

2.13 RX_DATA_BLOCK_32

Command

RX_DATA_BLOCK_32 {FILENAME}

Description

Causes the BSL to read the supplied TI TXT file and download all data contained in this file to the MSP432. The command supports full 32-bit address space and works only on MSP432.

Parameters

FILENAME *[mandatory]*: The name of the TI TXT file to read.

The path of file that will be read is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the main-script-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-read is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (Windows)

- Same folder: RX_DATA_BLOCK_32 Big_File.txt
- Sub folder: RX_DATA_BLOCK_32 SubFolder\Big_File.txt
- Up folder: RX_DATA_BLOCK_32 ../Big_File.txt
- Absolute folder: RX_DATA_BLOCK_32 C:\Data\ScriptFolder\Big_File.txt

Example (Linux and Mac)

- Same folder: RX_DATA_BLOCK_32 Big_File.txt
- Sub folder: RX_DATA_BLOCK_32 SubFolder/Big_File.txt
- Up folder: RX_DATA_BLOCK_32 ../Big_File.txt
- Absolute folder: RX_DATA_BLOCK_32 /home/Data/ScriptFolder/Big_File.txt

2.14 RX_DATA_BLOCK_FAST

Command

RX_DATA_BLOCK_FAST {FILENAME}

Description

Identical to RX_DATA_BLOCK except no verification of programming is returned from the BSL. This is useful for USB programming only, the BSL Scripter can only confirm that the file was sent. This command works on MSP430 and MSP432 but covers only 24-bit address space.

Parameters

FILENAME *[mandatory]*: The name of the TI TXT file to read.

The path of file that will be read is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the main-script-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-read is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (Windows)

- Same folder: RX_DATA_BLOCK_FAST Big_File.txt
- Sub folder: RX_DATA_BLOCK_FAST SubFolder\Big_File.txt
- Up folder: RX_DATA_BLOCK_FAST ../Big_File.txt
- Absolute folder: RX_DATA_BLOCK_FAST C:\Data\ScriptFolder\Big_File.txt

Example (Linux and Mac)

- Same folder: RX_DATA_BLOCK_FAST Big_File.txt
- Sub folder: RX_DATA_BLOCK_FAST SubFolder/Big_File.txt
- Up folder: RX_DATA_BLOCK_FAST ../Big_File.txt
- Absolute folder: RX_DATA_BLOCK_FAST /home/Data/ScriptFolder/Big_File.txt

2.15 RX_PASSWORD

Command

RX_PASSWORD {FILENAME}

Description

Causes the BSL to read the supplied TI TXT file and submit this data to the BSL as a password to unlock the device if the password is correct. If the password is wrong, a mass erase will be done.

Note: Although the same command is used to supply the password for the 543x family and other 5xx devices, the password file needs to be handled differently for these devices due to smaller password size in the 543x family. For more details, please see the Bootstrap Loader User's Guide.

Note: For USB BSL (on the device, not the full-featured RAM BSL for USB) without built in MASS ERASE command, this command can be used with an incorrect password to trigger a mass erase.

FILENAME *[mandatory]*: The name of the TI TXT file to read.

The path of file that will be read is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the main-script-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-read is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (without input file)

- RX_PASSWORD

Example (Windows)

- Same folder: RX_PASSWORD app_pass.txt
- Sub folder: RX_PASSWORD SubFolder\app_pass.txt
- Up folder: RX_PASSWORD ..\app_pass.txt
- Absolute folder: RX_PASSWORD C:\Data\ScriptFolder\app_pass.txt

Example (Linux and Mac)

- Same folder: RX_PASSWORD app_pass.txt
- Sub folder: RX_PASSWORD SubFolder/app_pass.txt
- Up folder: RX_PASSWORD ../app_pass.txt
- Absolute folder: RX_PASSWORD /home/Data/ScriptFolder/app_pass.txt

2.16 RX_PASSWORD_32

Command

RX_PASSWORD_32 {/FILENAME}

Description

Identical to RX_PASSWORD except that the default password has a length of 256 byte and can only be used for MSP432Pxx devices. The memory is 32 byte addressed.

Parameters

FILENAME *[mandatory]*: The name of the TI TXT file to read.

The path of file that will be read is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-read is located on the same folder as the main-script-file
- Sub folder: file-to-be-read is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-read is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-read is located in specific absolute folder

Example (without input file)

- RX_PASSWORD_32

Example (Windows)

- Same folder: RX_PASSWORD_32 app_pass.txt
- Sub folder: RX_PASSWORD_32 SubFolder\app_pass.txt
- Up folder: RX_PASSWORD_32 ..\app_pass.txt
- Absolute folder: RX_PASSWORD_32 C:\Data\ScriptFolder\app_pass.txt

Example (Linux and Mac)

- Same folder: RX_PASSWORD_32 app_pass.txt
- Sub folder: RX_PASSWORD_32 SubFolder/app_pass.txt
- Up folder: RX_PASSWORD_32 ../app_pass.txt
- Absolute folder: RX_PASSWORD_32 /home/Data/ScriptFolder/app_pass.txt

2.17 TOGGLE_INFO

Command

TOGGLE_INFO

Description

This command causes the INFO_A lock to toggle to either protect or unlock the INFO_A segment on MSP430F5xx and MSP430F6xx devices. For more detail on this lock please see the corresponding User's Guide (SLAU208).

Parameters

NONE

Example

■ TOGGLE_INFO

2.18 TX_DATA_BLOCK

Command

TX_DATA_BLOCK {ADDRESS} {LENGTH} {/FILENAME}

Description

Writes a block of data in TI TXT format to given file.

Parameters

ADDRESS[mandatory]: The address at which the read should begin (hex format).

LENGTH[mandatory]: The number of bytes to read (hex format).

FILENAME[mandatory]: The file to which the read bytes will be written.

The path of file that will be written is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-written is located on the same folder as the main-script-file
- Sub folder: file-to-be-written is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-written is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-written is located in specific absolute folder

Example (Windows)

- Same folder: `TX_DATA_BLOCK 0x8000 0x1000 Data_Read.txt`
- Sub folder: `TX_DATA_BLOCK 0x8000 0x1000 SubFolder\Data_Read.txt`
- Up folder: `TX_DATA_BLOCK 0x8000 0x1000 ../Data_Read.txt`
- Absolute folder: `TX_DATA_BLOCK 0x8000 0x1000 C:\Data\ScriptFolder\Data_Read.txt`

Example (Linux and Mac)

- Same folder: `TX_DATA_BLOCK 0x8000 0x1000 Data_Read.txt`
- Sub folder: `TX_DATA_BLOCK 0x8000 0x1000 SubFolder/Data_Read.txt`
- Up folder: `TX_DATA_BLOCK 0x8000 0x1000 ../Data_Read.txt`
- Absolute folder: `TX_DATA_BLOCK 0x8000 0x1000 /home/Data/ScriptFolder/Data_Read.txt`

2.19 TX_DATA_BLOCK_32

Command

TX_DATA_BLOCK_32 {ADDRESS} {LENGTH} {/FILENAME}

Description

Identical to TX_DATA_BLOCK except that memory accessed by 32-byte addresses. This command works on MSP432 devices only.

Parameters

ADDRESS *[mandatory]*: The address at which the read should begin (hex format).

LENGTH *[mandatory]*: The number of bytes to read (hex format).

FILENAME_[mandatory]: The file to which the read bytes will be written.

The path of file that will be written is relative to the location where the main-script-file is located.

Four directory approaches are:

- Same folder: file-to-be-written is located on the same folder as the main-script-file
- Sub folder: file-to-be-written is located one(or more) folder down relative the main-script-file
- Up folder: file-to-be-written is located one(or more) folder up relative the main-script-file
- Absolute folder: file-to-be-written is located in specific absolute folder

Example (Windows)

- Same folder: TX_DATA_BLOCK_32 0x8000 0x1000 Data_Read.txt
- Sub folder: TX_DATA_BLOCK_32 0x8000 0x1000 SubFolder\Data_Read.txt
- Up folder: TX_DATA_BLOCK_32 0x8000 0x1000 ..\Data_Read.txt
- Absolute folder: TX_DATA_BLOCK_32 0x8000 0x1000 C:\Data\ScriptFolder\Data_Read.txt

Example (Linux and Mac)

- Same folder: TX_DATA_BLOCK_32 0x8000 0x1000 Data_Read.txt
- Sub folder: TX_DATA_BLOCK_32 0x8000 0x1000 SubFolder/Data_Read.txt
- Up folder: TX_DATA_BLOCK_32 0x8000 0x1000 ../Data_Read.txt
- Absolute folder: TX_DATA_BLOCK_32 0x8000 0x1000 /home/Data/ScriptFolder/Data_Read.txt

2.20 TX_BSL_VERSION

Command

TX_BSL_VERSION

Description

Returns the BSL version of the MSP430 device.

Vendor: [xx], CI: [xx], API: [xx], PI: [xx]

Parameters

NONE

Example

■ TX_BSL_VERSION

2.21 TX_BSL_VERSION_32

Command

TX_BSL_VERSION_32

Description

Returns the BSL version of the MSP432 device.

Vendor: [xxxx], CI: [xxxx], API: [xxxx], PI: [xxxx], BID: [xxxx]

Parameters

NONE

Example

■ TX_BSL_VERSION_32

2.22 TX_BUFFER_SIZE

Command

TX_BUFFER_SIZE

Description

The BSL transmits a value that represents the number of bytes available in its data buffer for sending or receives BSL core data packets. NOTE: This command is not supported with MSP432 devices.

Parameters

NONE

Example

■ TX_BUFFER_SIZE

2.23 VERBOSE

Command

VERBOSE

Description

Causes the PC application to toggle output for all transmitted and received bytes on or off.

Parameters

NONE

Example

■ VERBOSE

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

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Medical	www.ti.com/medical
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
Security	www.ti.com/security
Telephony	www.ti.com/telephony
Video & Imaging	www.ti.com/video
Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2015, Texas Instruments Incorporated