User's Guide
SLAU573C–November 2015–Revised October 2019

MSP-BSL Bootloader (BSL) Programmer for MSP430™ and SimpleLink™ MSP432™ Microcontrollers

The MSP-BSL (previously known as the MSP430-BSL) is a low-cost programmer in the shape of a rocket. Hence, it is nicknamed the "BSL Rocket". The MSP-BSL is designed for easy communication between a PC and the BSL of an MSP430™ or SimpleLink™ MSP432™ microcontroller (MCU) through USB. The MSP-BSL project is a collaboration between Olimex Ltd and Texas Instruments. The PCB design and firmware for the MSP-BSL are open source. As of this writing, the MSP-BSL supports UART, I^2C, and SPI communication, but it can be extended for future requirements.

Contents

List of Figures

List of Tables

Trademarks

1 Introduction

The bootloader (BSL) is an application built into MSP430 and MSP432 microcontrollers. The BSL enables the user to communicate with the device and to read and write its memory. This feature is primarily used for programming the device during prototyping phase, final production, and in service. Both the programmable memory (flash memory or FRAM) and the data memory (RAM) can be modified as required. Different BSLs offer different peripherals to communicate with (for example, UART, I^2C, SPI, or USB).

The MSP-BSL (previous knows as the MSP430-BSL) is a low-cost programmer in the shape of a rocket (see Figure 1 and Figure 2). Hence, it is nicknamed the "BSL Rocket". The MSP-BSL is designed for easy communication between a PC and the BSL of an MSP430 or MSP432 MCU through USB. The MSP-BSL project is a collaboration between Olimex Ltd and Texas Instruments. The PCB design and firmware for the MSP-BSL are open source. As of this writing, the MSP-BSL supports UART, I^2C, and SPI communication, but it can be extended for future requirements.

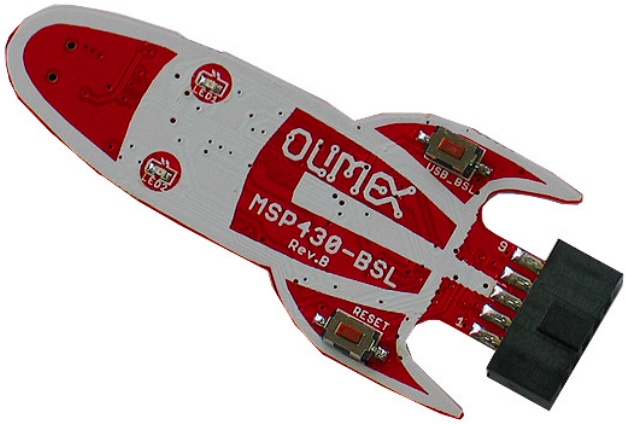

Figure 1. MSP-BSL (Top View)

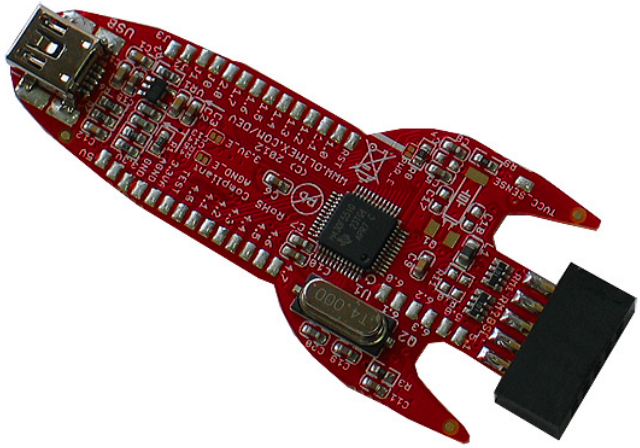

Figure 2. MSP-BSL (Bottom View)

1.1 Additional Online Information

More information on the BSL can be found in the following documents:

MSP430™ Flash Device Bootloader (BSL) User's Guide
MSP430™ FRAM Devices Bootloader (BSL) User's Guide
MSP432™ SimpleLink™ Microcontroller Bootloader (BSL) User's Guide
Bootloader for SimpleLink MSP432E Microcontrollers
Bootloader (BSL) Scripter User's Guide

The latest version of the firmware and source code for the MSP-BSL is available from http://software-dl.ti.com/msp430/msp430_public_sw/mcu/msp430/MSPBSL_Rocket_FW/latest/index_FDS.html.

2 Functionality and Supported Protocols

The MSP-BSL is a USB communications device class (CDC) device. It enumerates on the host PC as a virtual COM port. The data that is sent to the MSP-BSL through this serial connection appears transparently at the output, and data that is received by the MSP-BSL is forwarded transparently to the PC.

The serial communication to the MSP-BSL uses 8 data bits, no parity bit, and 1 stop bit (8N1). The baud rate selects the target communication protocol in use. Special functionality (for example applying a BSL entry sequence (see Section 2.1)) can also be triggered by baud rate changes.

Table 1 describes special-purpose baud rates used by the MSP-BSL. Each function in this table is described in detail in the following subsections.

Table 1. Supported Baud Rates of the MSP-BSL

Protocol	Baud Rate	Description
UART	4801	Invoke the USB BSL on the MSP-BSL Rocket
	4802[(1)]	Triggers the MSP-BSL Rocket to generate exit sequence in TST and RST pins for MSP430 UART BSL
	9601	Triggers the MSP-BSL Rocket to generate invoke sequence in TST and RST pins for MSP430F5xx and MSP430F6xx UART BSL
	9602	Triggers the MSP-BSL Rocket to generate invoke sequence in TST and RST pins for MSP430F1xx, MSP430F2xx, MSP430F4xx, and MSP430G2xx UART BSL
	9608, 19208, 38408, 57608, 115208	Triggers the MSP-BSL Rocket to have UART peripheral configuration for MSP432E4 family
	9600, 19200, 38400, 57600, 115200	Standard baudrate communication for MSP430 and MSP432 UART BSL
	1200, 2400, 4800, 230400, 460800, 921600	Standard baudrate communication for UART (not supported in MSP430 and MSP432 UART BSL)
I^2C	100001, 400001	Triggers the MSP-BSL Rocket to generate invoke sequence in TST and RST pins MSP430 I2C BSL
	100008, 400008	Triggers the MSP-BSL Rocket to have I2C peripheral configuration for MSP432E4 family
	100000 + (slaveAddress), 400000 + (slaveAddress)	Initialize the custom I2C slave address for I2C BSL all families. slaveAddress has to be treated in decimal format. With the implementation with BSL-Scripter, the parameter is given in hexadecimal string (for example: 0x48), and the BSL-Scripter translates this into decimal.
	100000, 400000	Standard bit rate communication for MSP430 and MSP432 I2C BSL
SPI	125008, 250008, 500008, 1000008	Triggers the MSP-BSL Rocket to have SPI peripheral configuration for MSP432E4 family
	125000, 250000, 500000, 1000000	Standard bit rate communication for MSP432P4xx and MSP432E4 SPI BSL

(1) Erratum JTAG20 applies to only the following device families: MSP430F543x, MSP430F543xA, MSP430F550x, MSP430F552x, MSP430F613x, MSP430663x.

2.1 BSL Entry Sequence

For most MSP430 devices, there are two ways to invoke the BSL: by the application software or by applying a hardware entry sequence.

The MSP-BSL can apply the entry sequence (see Figure 3) to the target. This entry sequence can be used for devices with shared or dedicated JTAG pins. The entry sequence can be triggered by setting the communication speed of the MSP-BSL to 9601 baud (for UART) or to the baud rate of any mode with a BSL entry sequence; for example, 100001 baud for I^2C mode with BSL entry sequence (see Table 1).

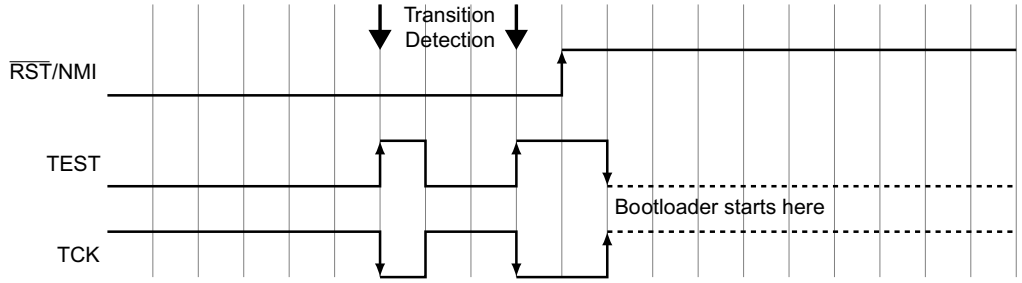

Figure 3. BSL Entry Sequence With Shared and Dedicated JTAG Pins

NOTE: For further information on the entry sequence, see the user's guides listed in Section 1.1.

2.2 UART Communication

The MSP-BSL provides a mode to communicate with UART BSLs. In this mode, all data sent to the virtual COM port is output at the TX pin (RX on the target connector, see Section 4.2). All data received on the RX pin (TX on the target connector) is forwarded to the PC. The MSP-BSL acts as a transparent USB-to-UART bridge.

To start UART BSL communication, perform the following steps, depending on the device family.

For the MSP430 family:

1. The host triggers the MSP-BSL Rocket with 9601 or 9602 (refer to the invocation baud rate for different families in Table 1), so the Rocket generates invoke sequence in TEST and RST pins.
2. The BSL works in even-parity mode.
3. The host starts programming with 9600 standard baud rate.
4. After the bootloader is successfully invoked, the host is able to initiate the faster speed of 9200, 38400, 57600, or 115200 standard baud rate.

For the MSP432P4xx family:

1. BSL works in even-parity mode by default.
2. The host starts programming with 9600, 19200, 38400, 57600, or 115200 standard baud rate.

For the MSP432E4 family:

1. The host triggers the MSP-BSL Rocket with 9608, 19208, 38408, 57608, or 115208, so the Rocket initialize the peripheral without any parity.
2. The host starts programming with 9600, 19200, 38400, 57600, or 115200 standard baud rate.

2.3 I²C Communication

Similar to the UART mode, the MSP-BSL supports the I²C protocol, starting with firmware version 2.1. All data sent to the virtual COM port is output on the SDA and SCL pins (see Section 4.2). Data received from I²C is forwarded to the PC. The MSP-BSL acts as a transparent USB-to-I²C bridge, except that it returns an error code (055h) to the PC if the I²C communication fails. The MSP-BSL has a state machine that emulates this transparent behavior.

The MSP-BSL operates as an I²C master and uses 7-bit addressing mode. The slave address for the MSP430 or MSP432P4xx target is 048h, and the slave address for the MSP432E4 target is 042h.

To start I²C BSL communication, perform the following steps, depending on the device family.

For the MSP430 family:

1. The host triggers the MSP-BSL Rocket with 100001 or 400001 bit rate.
2. The slave address generated by MSP-BSL Rocket is 0x48.
3. For custom I²C address only : The host triggers the MSP-BSL Rocket with (100000 + slaveAddress) or (400000 + slaveAddress).
4. The host starts the programming with 100000 or 400000 standard bit rate.

For the MSP432P4xx family:

1. The slave address generated by MSP-BSL Rocket is 0x48.
2. For custom I²C address only : The host triggers the MSP-BSL Rocket with (100000 + slaveAddress) or (400000 + slaveAddress).
3. The host starts the programming with 100000 or 400000 standard bit rate.

For the MSP432E4 family:

1. The host triggers the MSP-BSL Rocket with 100008 or 400008 bit rate.
2. The slave address generated by MSP-BSL Rocket is 0x48.
3. For custom I²C address only : The host triggers the MSP-BSL Rocket with (100000 + slaveAddress) or (400000 + slaveAddress).
4. The host starts the programming with 100000 or 400000 standard bit rate.

The MSP-BSL supports I²C Standard Mode (Sm) with a maximum transfer rate of 100 kbps and I²C Fast Mode (Fm) with a maximum transfer rate of 400 kbps. Both of these modes are selected by changing the bit rate of the serial connection from the PC to the MSP-BSL.

NOTE: I²C communication requires pullup resistors on the SDA and SCL lines. The pullup resistors can be soldered on the MSP-BSL PCB (see Section 4.4), or they can be included in the target application design; for example, by adding jumpers to the I²C pullups on the MSP430 or MSP432 target socket board.

TI recommends pullup resistors of 2 to 4 kΩ.

2.4 SPI Communication

Similar to the UART mode, the MSP-BSL supports the SPI protocol starting from firmware version 3.0. All data send to the virtual COM port is output through the SOMI, SIMO, CLK, and STE pins (see Section 4.2). A state machine on the MSP-BSL makes the communication appear as transparent UART to the outside.

To start SPI BSL communication, perform the following steps, depending on the device family.

For the MSP430 family:

1. The SPI BSL is not supported for MSP430 BSL

For the MSP432P4xx family:

1. The SPI BSL operates with the default configuration of:
   a. Data is changed on first clock edge and captured on the following edge (CKPH = 0).
   b. The clock is high when inactive (CKPL = 1).
   c. The slave transmit enable is active low.
   d. Uses 8-bit serial data character format.
2. The host starts the programming with 125000, 250000, 500000, or 1000000 standard bit rate.

For the MSP432E4 family:

1. The host triggers the MSP-BSL Rocket with 125008, 250008, 500008, or 1000008 bit rate.
2. The SPI BSL operates with the default configuration of:
   a. Data is changed on first clock edge and captured on the following edge (CKPH = 0).
   b. The clock is low when inactive (CKPL = 0).
   c. The slave transmit enable is active low.
   d. Uses 8-bit serial data character format.
3. Host starts the programming with 125000, 250000, 500000, or 1000000 standard bit rate.

3 Firmware Update

To update the firmware of the MSP-BSL, the USB BSL of the MSP430F5510 on the MSP-BSL can be used. To update the firmware:

1. Download the latest firmware for the MSP-BSL from the BSL tool folder.
2. Download and install the MSP430 USB Firmware Upgrade Example. It is part of the MSP430 USB Developers Package.
3. Start the MSP430 USB Firmware Upgrade Example. It shows 'No device connected' at this time.
4. Before starting the update, the USB BSL of the MSP-BSL must be invoked.
    1. Disconnect the MSP-BSL from the USB cable.
    2. Hold the USB BSL button down while connecting the USB cable.
    3. Alternatively, switch to baud rate 4801 to invoke the BSL on the MSP-BSL.

    The BSL is now invoked, and the Firmware Upgrade Example shows 'Found 1 device' (see Figure 4).

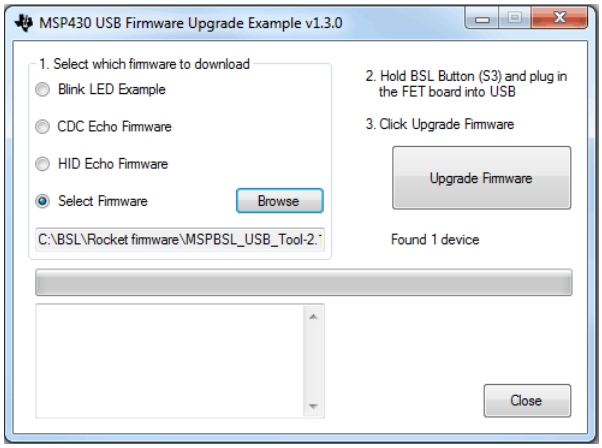

Figure 4. Firmware Upgrade Example With Selected Firmware and Connected MSP-BSL (USB BSL Invoked)

5. Now the firmware can be updated.
    1. Click ‘Select Firmware’ and ‘Browse’ to select the TI-TXT firmware image for the MSPBSL.
    2. Click ‘Upgrade Firmware’.

    The image is loaded, and the MSP-BSL restarts automatically. The new firmware is now ready to use (see Figure 5).

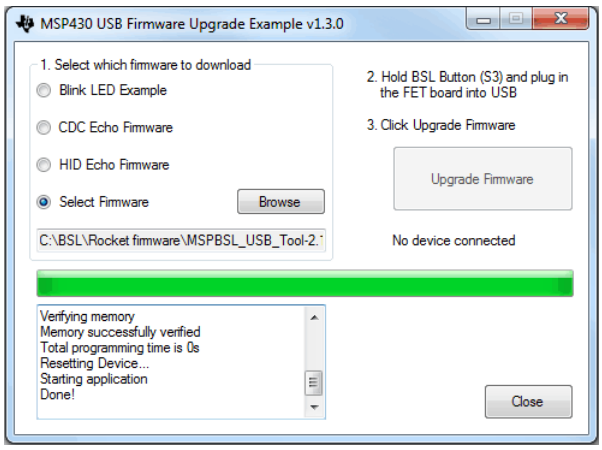

Figure 5. Firmware Upgrade Example Showing Successful Firmware Update

4 Hardware

The hardware of the MSP-BSL is open source. The design files are available on the Olimex website.

The MSP-BSL programmer is based on MSP430F5510. All I/Os that are not used by the standard BSL target connector (see Section 4.2) are made available as pads on the bottom of the PCB. Furthermore, there are two status LEDs (green and yellow) and two push buttons, a reset button and a button to invoke the USB BSL of the MSP430F5510.

The MSP-BSL features also an onboard 3.3-V voltage regulator that can supply up to 150 mA to the target (see Section 4.3).

4.1 Schematic

Figure 6 shows the schematic of the MSP-BSL.

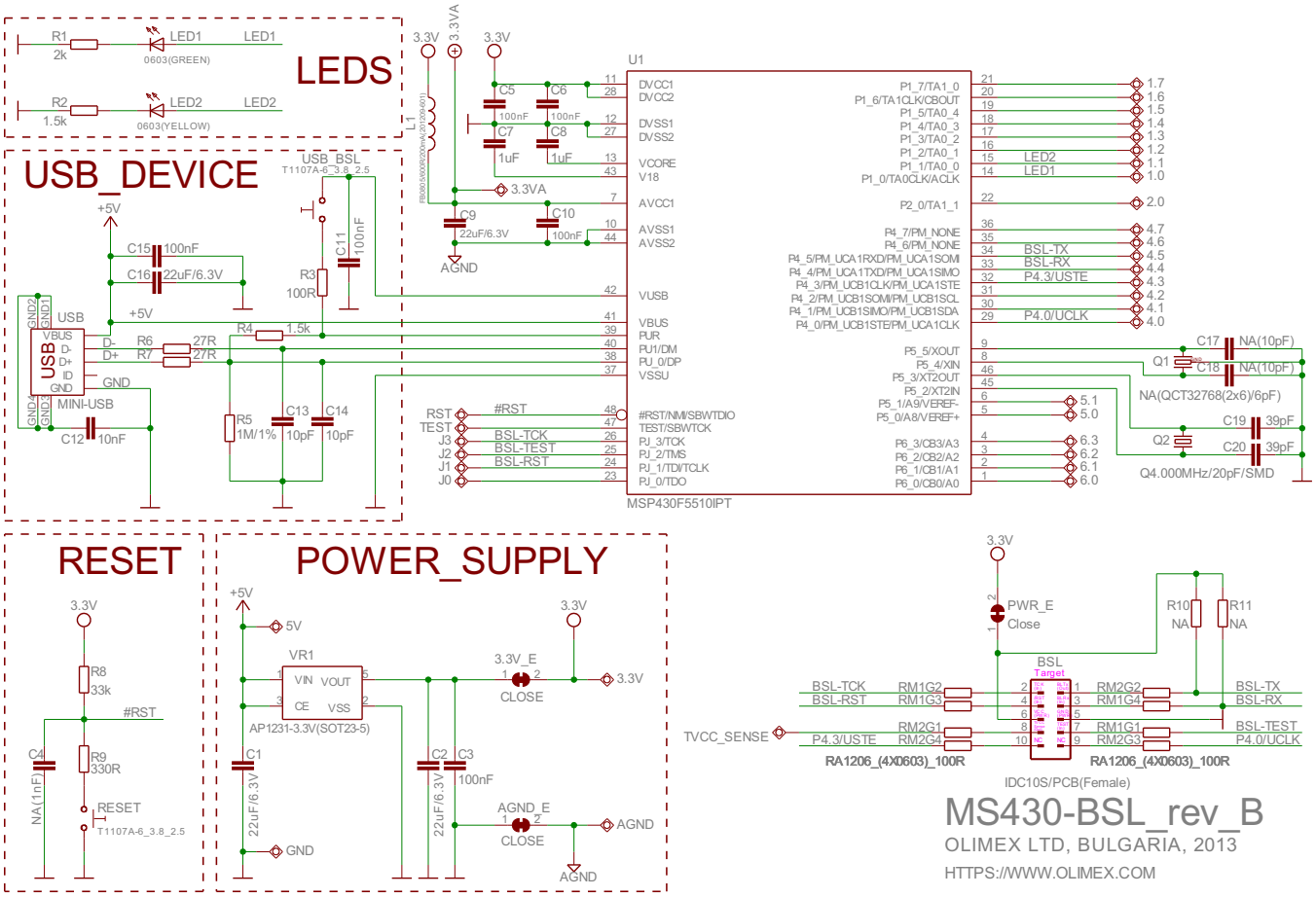

Figure 6. Schematic of the MSP-BSL

4.2 BSL Connector and Available Pins

Figure 7 shows the BSL Connector as seen from the target side.

- UART communication is handled through pin 1 (TX) and pin 3 (RX).
- I^2C communication is handled through pin 1 (SDA) and pin 9 (SCL).
- SPI communication is handled through pin 1 (SOMI), pin 3 (SIMO), pin 9 (CLK), and pin 10 (STE).
- The entry sequence can be generated using pin 4 (RST) and pin 7 (TEST) for devices with shared JTAG pins, or using pin 2 (TCK) and pin 4 (RST) for devices with dedicated JTAG pins.
- Power is supplied through pin 6 (VCC), and electrical ground is supplied through pin 5 (GND).

The connector uses 0.1-inch spacing and is a 10-pin male header on the target board.

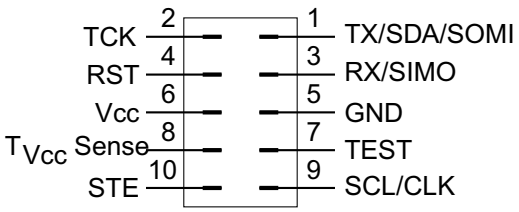

Figure 7. Pinout of the BSL Connector (Target Side)

4.3 Target Power Supply

The MSP-BSL has a built-in 3.3-V power supply for the target board. It can supply up to 150 mA of current. The power is supplied on pin 6 of the BSL connector but can be cut (by opening PWR_E, see Figure 8) if not needed.

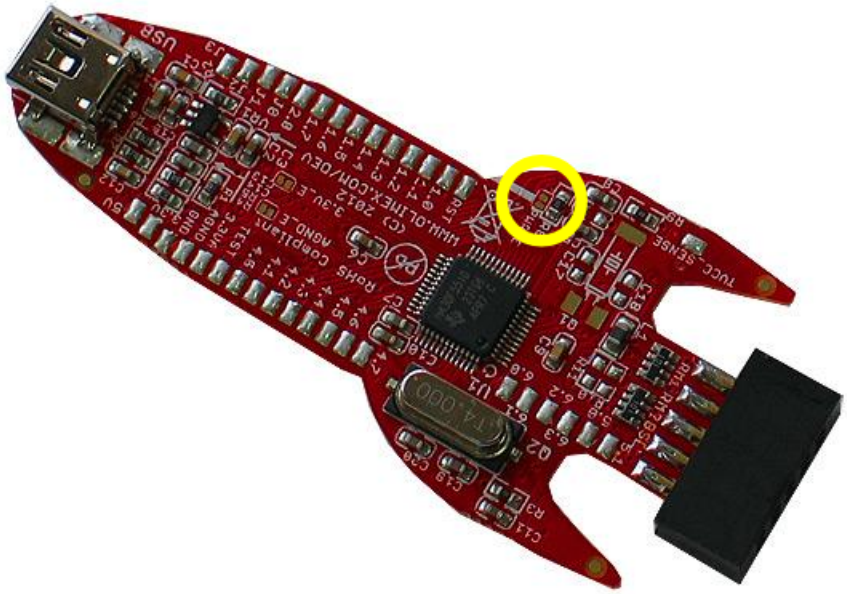

Figure 8. Cuttable Power Supply Trace (PWR_E)

NOTE: Cutting the power supply at PWR_E also prevents the onboard pullup resistors (R10, R11) from being pulled high.

4.4 Pullups for I²C Operation

I²C communication requires pullups on the SDA and SCL lines. These pullup resistors can be included in the target application design or onboard the MSP-BSL. Newer MSP430 and MSP432 target socket boards have I²C pullup resistors already included.

5 Firmware Revision History

Table 2 lists the revision history of the firmware.

Table 2. Firmware Releases

Version	Changes
3.1	Improved the FSM for SPI communication. Generalized the entry sequence timing requirements. Implemented the exit sequence.
3.0	Added SPI support Updated USB stack
2.1	Added I²C support
1.0	Initial version

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from October 28, 2017 to October 4, 2019 **Page**

- Added option 9602 for the UART protocol in Table 1, *Supported Baud Rates of the MSP-BSL* 3
- Added the invocation baud rate of 9602 to step 1 for the MSP430 family in Section 2.2, *UART Communication* 4
- Removed baud rate options except for 9600 in step 3 for the MSP430 family in Section 2.2, *UART Communication* 4
- Added step 4 for the MSP430 family in Section 2.2, *UART Communication* 4

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265