

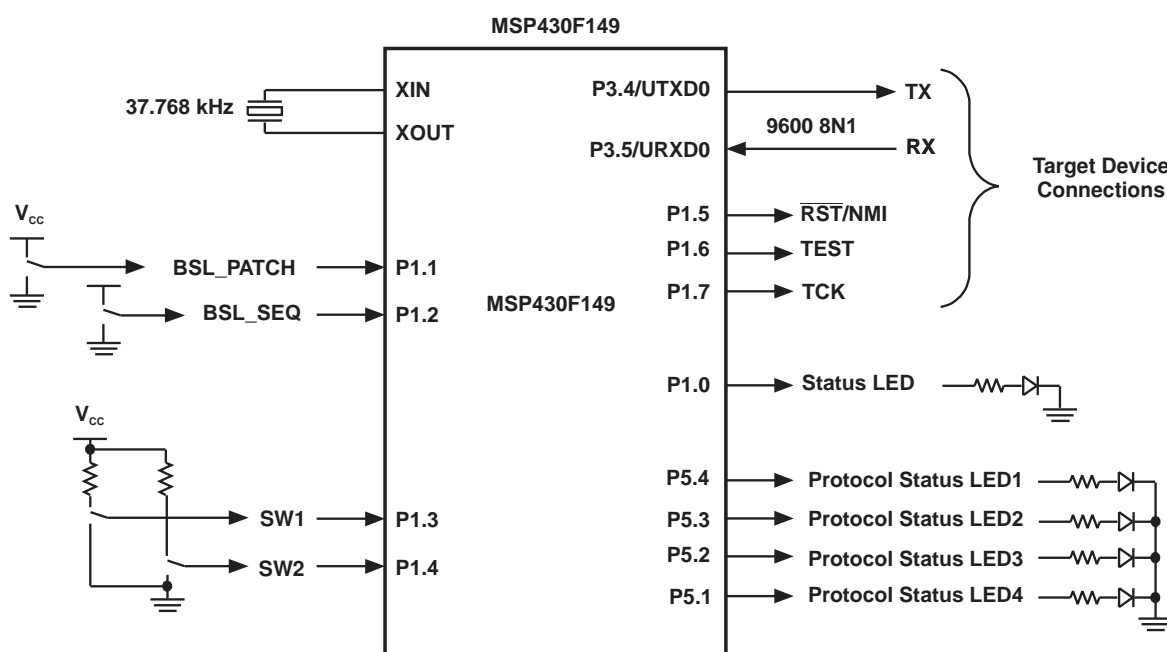
## Appendix B MSP430 BSL Replicator

Author: Greg Morton, MSP430 Applications

### B.1 BSL Replicator Overview

The BSL Replicator application, executing on a host MSP430F149 device, uses the BSL protocol to communicate with BSL (bootstrap loader) firmware residing on a target MSP430 device. Its purpose is to program other MSP430 devices. The program that is downloaded is stored locally in the host's flash memory in an array called CodeArray[] (refer to [Section B.5](#), *Building a Downloadable MSP430 Application*, for a description of how to build a target application and place it into CodeArray[]). Although not all the BSL messages are used to implement the BSL Replicator, functions to create all the BSL messages can be found in the program.

Figure B-1 shows a block diagram of the BSL Replicator.



**Figure B-1. BSL Replicator Block Diagram**

**Note:** The BSL Replicator does not currently support the MSP430X extended memory architecture devices.

The input signals shown in [Figure B-1](#):

- SW1 – momentary pushbutton switch tied to  $V_{CC}$  when open. Pressing SW1 generates a BSL entry sequence, followed by a download of the program stored in flash to a target MSP430 device. Status LED4 is illuminated if the download was successful. Otherwise, Status LED3 is illuminated, indicating a failure.
- BSL\_PATCH – static signal controlling SW1 operation. When tied to  $V_{CC}$ , a BSL firmware patch is downloaded to RAM on the target MSP430 prior to program download.
- BSL\_SEQ – static signal controlling SW1 operation. When tied to  $V_{CC}$ , the BSL entry sequence for a target device having dedicated JTAG pins is generated. When connected to GND, the BSL entry sequence for a target device having shared JTAG pins is generated.
- SW2 – momentary pushbutton switch tied to  $V_{CC}$  when open. Pressing SW2 sends a BSL message to the target, instructing it to begin executing the program downloaded to it. Status LED4 is illuminated if the target successfully received the message. Otherwise, LED3 is illuminated, indicating a failure.

The output signals shown in [Figure B-1](#):

- TX/RX – 9600 baud, 8N1 UART.
- $\overline{\text{RST}}$ /NMI – connects to target's  $\overline{\text{RST}}$ /NMI pin.
- TEST – connects to target's TEST pin for MSP430 devices having shared JTAG pins
- TCK – connects to target's TCK pin for MSP430 devices having dedicated JTAG pins.
- Status LED – optional status LED. Indicates that the host's DCO was successfully calibrated when illuminated at power-up. Later, the LED is illuminated whenever timer is active.
- Protocol Status LED1 – optional status LED. Illuminated whenever an ACK character or a message is received from the target.
- Protocol Status LED2 – optional status LED. Illuminated after a transmit message is sent to the target.
- Protocol Status LED3 – optional status LED. Illuminated whenever a NACK is received from the target. Also illuminated to indicate a failure.
- Protocol Status LED4 – optional status LED. Illuminated to indicate a successful operation.

## **B.2 Implementation**

[Figure B-2](#) and [Figure B-3](#) contain a flow chart for the BSL Replicator application. It illustrates the logic flow for the main program loop. In [Figure B-2](#), an exit from low-power mode 0 (LPM0) occurs after either a Timer\_A interrupt or receiving a character from the target BSL.

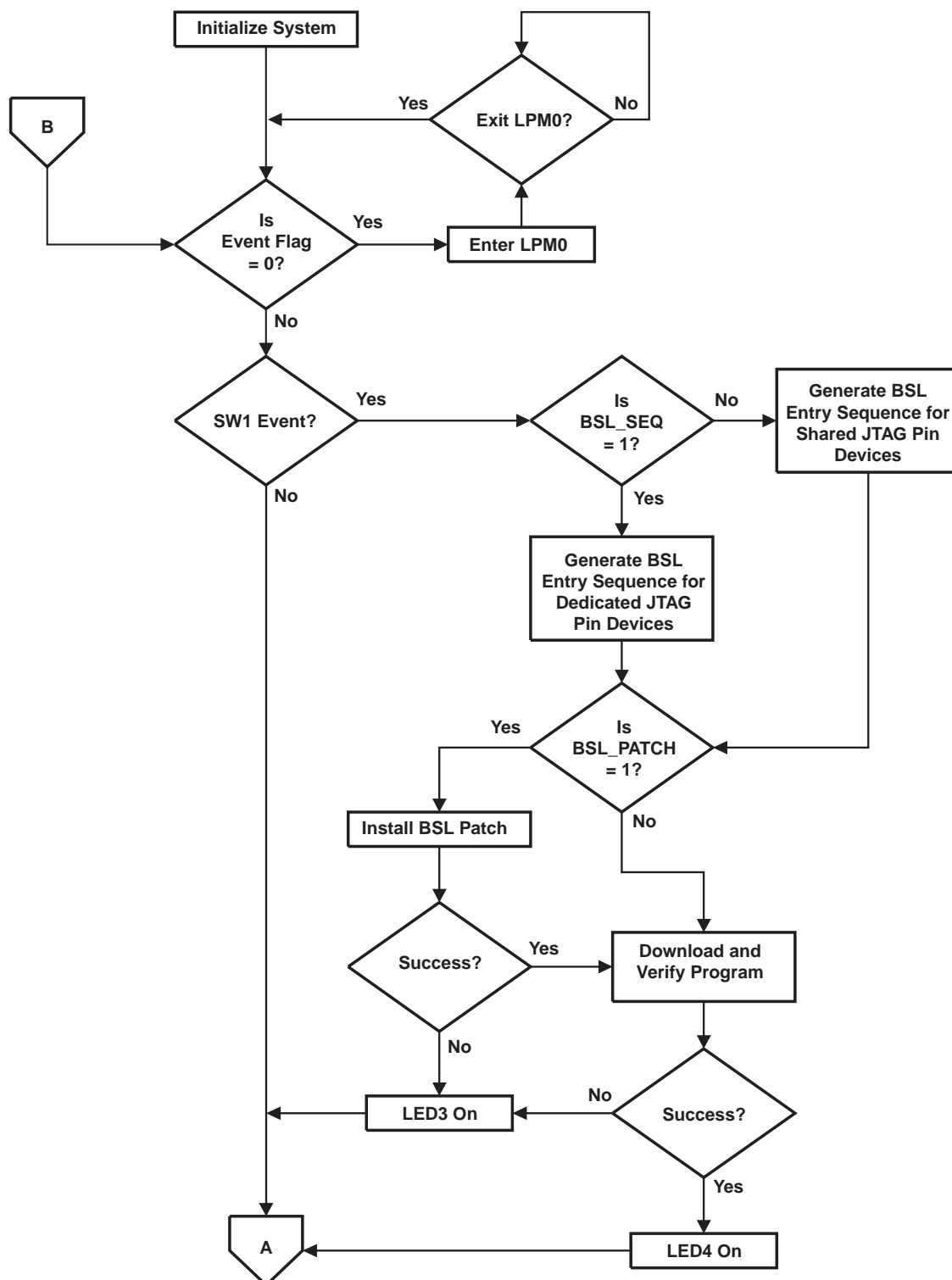


Figure B-2. BSL Replicator Flow Chart

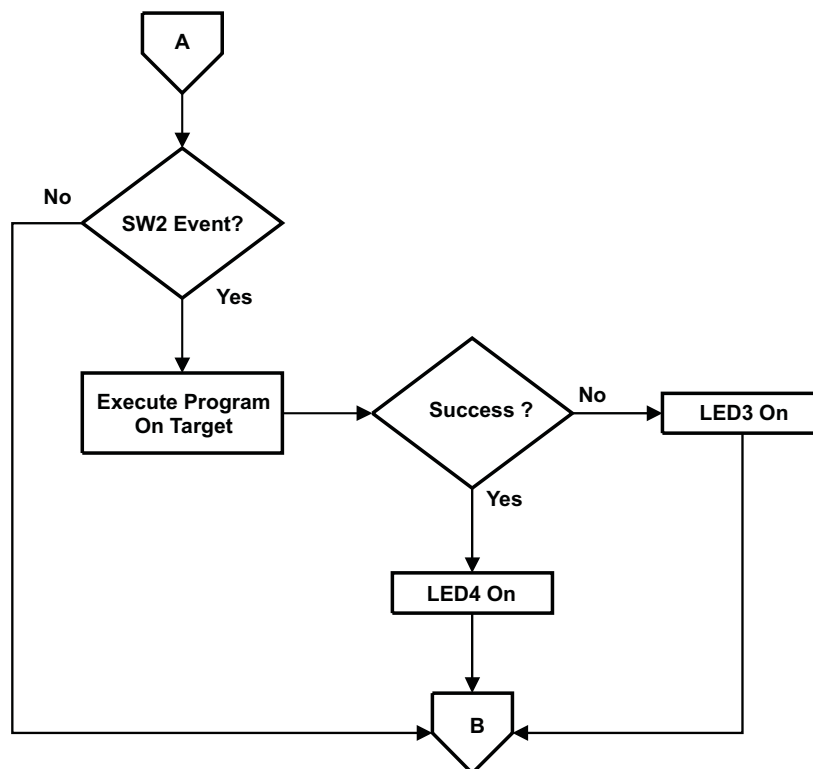


Figure B-3. BSL Replicator Flow Chart (continued)

### B.3 BSL Message Functions

The following functions implement the messages defined in the BSL protocol:

- **bslMsgRxDataBlk(BslMsg\* pMsg, int addr, int len, unsigned int\* pData)** – creates message for the BSL command, RX Data Block (CMD ID 0x12). Writes block of data to target. Parameters:
  - pMsg – pointer to BslMsg data structure
  - addr – 16-bit block start address
  - len – number of bytes to write. Must be an even number. Maximum value is 250.
  - pData – pointer to host data to write to target device
- **bslMsgRxPassword(BslMsg\* pMsg)** – creates message for the BSL command, RX Password (CMD ID 0x10). Sends password to target BSL. Password is 32 bytes long and is stored in the global array, BslPassword[]. Password MUST match the 32 bytes starting at address 0xFFE0 in the target's flash memory. Default password is all 0xFF's which matches the contents of the target's memory after erasing flash. Parameter:
  - pMsg – pointer to BslMsg data structure
- **bslMsgEraseSeg(BslMsg\* pMsg, int addr, int len)** – creates message for the BSL command, Erase Segment (CMD ID 0x16). Erases specified flash segment. Parameters:
  - pMsg – pointer to BslMsg data structure
  - addr – any even-numbered address within the flash segment to be erased
  - len – value of 0xA504 performs erasure of main flash memory ?????
- **bslMsgErase(BslMsg\* pMsg)** – creates message for the BSL command, Mass Erase (CMD ID 0x18). Erases entire flash area, both Main and Information flash. Parameter:
  - pMsg – pointer to BslMsg data structure
- **bslMsgEraseChk(BslMsg\* pMsg, int addr, int len)** – creates message for the BSL command, Erase Check (CMD ID 0x1C). Verifies flash erasure within specified address range. Parameters:
  - pMsg – pointer to BslMsg data structure
  - addr – any 16-bit block start address (odd or even)
  - len – length of block (odd or even)
- **bslMsgBaudRate(BslMsg\* pMsg, int dco, int bcs, int baudRate)** – creates message for the BSL command, Changes Baud Rate (CMD ID 0x20). Parameters:
  - pMsg – pointer to BslMsg data structure
  - dco – DCO Control Register (DCOCTL) setting for a 1xx device or System Clock Frequency Integrator Register 0 (SCFI0) setting for a 4xx device.
  - bcs – Basic Clock System Control Register 1 (BCSCTL1) setting for a 1xx device or System Clock Frequency Integrator Register 1 (SCFI1) setting for a 4xx device.
  - baudRate – 0 is 9600 baud, 1 is 19200 baud, and 2 is 38400 baud
- **bslMsgLoadPC(BslMsg\* pMsg, int addr)** – creates message for the BSL command, Load PC (CMD ID 0x1A). Sets target's program counter register to specified value. Parameters:
  - pMsg – pointer to BslMsg data structure
  - addr – value for target's program counter register (R0)
- **bslMsgTxDataBlk(BslMsg\* pMsg, int addr, int bytes)** – creates message for the BSL command, TX Data Block (CMD ID 0x14). Reads block of data from target. Parameters:
  - pMsg – pointer to BslMsg data structure
  - addr – 16-bit block start address
  - bytes – length of block. Maximum value is 250
- **bslMsgVer(BslMsg\* pMsg)** – creates message for the BSL command, BSL Version (CMD ID 0x1E). Reads BSL version number and MSP430 device information from target. Parameter:
  - pMsg – pointer to BslMsg data structure

## **B.4 BSL Implementation Functions**

The following functions assist with the implementation of the BSL protocol.

- **bslEntrySeq(void)** – generates BSL entry sequence. If the input signal BSL\_SEQ is tied to VCC, then the signals required to place a MSP430 device having dedicated JTAG pins into BSL mode will be generated. If BSL\_SEQ is tied low, then the signals required to place a MSP430 device having shared JTAG pins into BSL mode will be generated.
- **bslChecksum(BslMsg\* pMsg)** – generates check sum for BSL message. Parameter:
  - pMsg – pointer to BslMsg data structure
- **bslTxMsg(BslMsg\* pMsg)** – transmits message to target BSL. Parameter:
  - pMsg – pointer to BslMsg data structure
- **bslTxSync(void)** – transmits SYNC character to target BSL.
- **bslSendChar(unsigned char val)** – transmits a single character to target BSL. Parameter:
  - val – any 8-bit value
- **bslWaitForReply(unsigned char type)** – waits for reply from target BSL. The BSL can respond in two ways. It either sends an ACK character or it responds with a message having a specified length. If target BSL does not reply, function returns a timeout error. Parameter:
  - type – BSL reply type, BSL\_ACK\_REPLY or BSL\_MSG\_REPLY
- **bslStateReset(void)** – resets host to restart communications with target BSL.
- **bslDownloadCode(unsigned int\* pCodeArray, unsigned char patchFlag)** – downloads code to target BSL. Code can be for BSL patch or target program. Parameters:
  - pCodeArray – pointer to host array containing code to download to target
  - patchFlag – !0 value indicates BSL patch code is installed on target
- **bslDownloadProgram(void)** – downloads program to target. Program is stored in flash memory on host in global array called CodeArray[].
- **bslInstallPatch(void)** – installs BSL patch on target. Patch is stored in flash memory on host in global array called PatchArray[].

## **B.5 Building a Downloadable MSP430 Application**

Perform the following steps to build an application that can be downloaded by the BSL Replicator to program another MSP430 device. The BSL Replicator stores the application locally in the array, CodeArray[].

### **Creating the Target Source**

1. Use IAR's IDE to create a MSP430 project for the target device.
2. Open the "Options" dialog box for the project by selecting "Project->Options..." from the main menu or by right-clicking on the project and selecting "Options..." on the pop-up menu.
3. Select "Linker" under the section labeled "Category".
4. Select the "Output" tab.
5. Under the section labeled "Format", select the radio button labeled "Other".
6. Select the "msp430-txt" option for the text box labeled "Output format".
7. Select "OK" and close the "Options" dialog box
8. Build the project.
9. Convert the "\*.txt" project output file into a "\*.s43" assembly file using the FileMaker.exe conversion program provided with this application note (Note: when using FileMaker, make sure to select the .s43 output file format).

---

**Note:** This version of FileMaker does not support placement of program code or data at memory addresses beyond address 0xFFFFh.

---

### **Creating the BSL Replicator Source**

1. Use IAR's IDE to create a MSP430 project for the Host device (BSL Replicator).
2. Add the bsl\_replicator.c source file to the BSL Replicator project.
3. Add the PATCH.s43 source file to the BSL Replicator project if using the BSL patch for the target.
4. Add the converted assembly .s43 output file from Step 9 to the BSL Replicator project.
5. Build the BSL Replicator project and download to the MSP430F149 host.

### **Programming the Target Device**

1. Connect the BSL Replicator to the Target device as shown in [Figure B-1](#).
2. Press SW1 to program the Target device.
3. After successful programming, pressing SW2 will release the Target to execute the target program.