

FW UPGRADE SPECIFICATION



SCA10H

Doc. No.1326 Rev. 3

Table of Contents

1	Programming With the Bootloader	3
1.1	Introduction.....	3
1.2	FW Upgrade Mode Entry Sequence	3
1.3	UART Protocol.....	4
1.4	Bootloader Hardware Interface	4
2	Bootloader Protocol	4
2.1	Synchronization Sequence	4
2.2	Commands	5
2.3	Data Frame.....	5
2.3.1	Data Frame Structure	5
2.3.2	Checksum.....	6
2.3.3	Simple Programming Flow	7
2.3.4	Commands – Detailed Description.....	7
2.3.4.1	RX Data Block	7
2.3.4.2	RX Password.....	8
2.3.4.3	Mass Erase.....	8
2.3.4.4	Change Baud Rate	8
2.3.4.5	TX Data Block.....	8
2.3.4.6	TX BL version	8
2.4	Exiting the FW Upgrade Mode	9
3	Example: FW Upgrade via USB to UART Bridge	9
4	Document Change Control.....	10

1 Programming With the Bootloader

The bootloader (BL) enables firmware (FW) upgrade after the JTAG security fuse has been blown. The bootloader expects AES encrypted binary via UART. The received binary will be decrypted and flashed into the MCU.

SCA10H FW upgrade interface utilizes Texas Instruments Bootstrap Loader (BSL) interface.

www.ti.com/lit/ug/slau319i/slau319i.pdf

1.1 Introduction

The bootloader provides a method to program the flash memory on the field using the UART protocol. The FW upgrade mode is activated after reset by sending a specific entry character via UART. After activation a synchronization character followed by the command data frame initiates desired function. A bootloader session can be closed by waiting for a 10 s timeout or by resetting SCA10H, please see Figure 1 below.

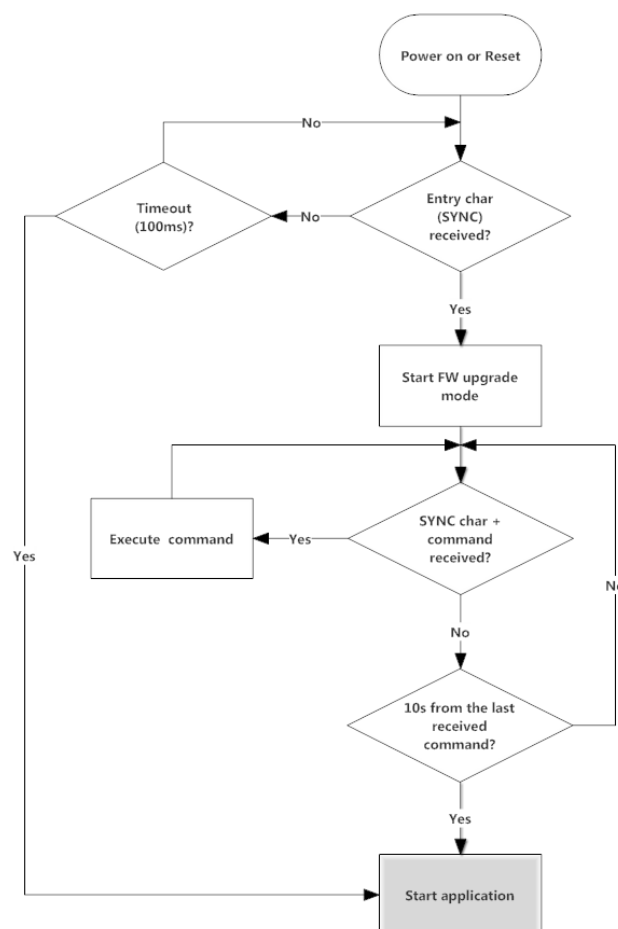


Figure 1 Bootloader workflow

1.2 FW Upgrade Mode Entry Sequence

Reset SCA10H by setting the RST/NMI pin low. If SCA10H interfaces with a serial communication port (RS232), the RST/NMI pin may be driven by the DTR signal. After

the RST/NMI rises from low to high, an entry character (SYNC = 0x80h) needs to be sent within 100ms, otherwise the application will start.

1.3 UART Protocol

The UART protocol:

- Baud rate is 9600 as default.
- Half-duplex mode (one sender at a time).
- Start bit, 8 data bits (LSB first), even parity, 1 stop bit.
- Handshake is performed by an acknowledge character.

1.4 Bootloader Hardware Interface

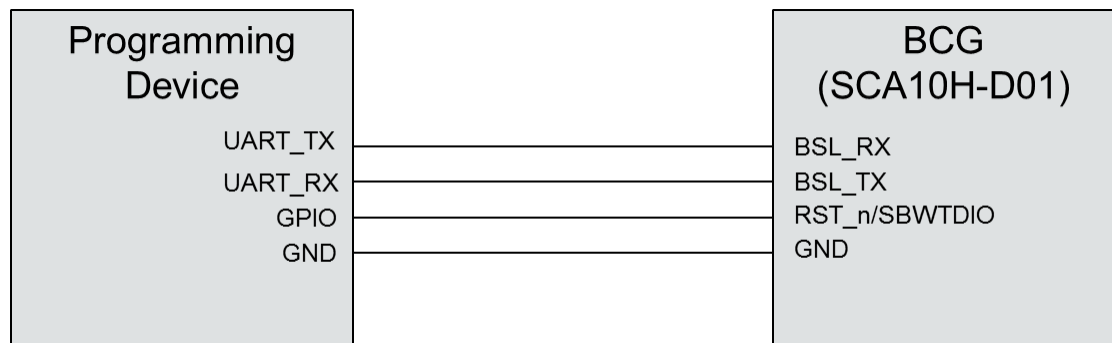


Figure 2 Bootloader hardware interface

2 Bootloader Protocol

2.1 Synchronization Sequence

A synchronization character (SYNC = 80h) must be sent to the bootloader before sending any command. When the bootloader receives the SYNC character, it will send back an acknowledge (DATA_ACK = 90h) to confirm successful reception.

This sequence must be done for every command sent to the bootloader.

2.2 Commands

Table 1 Bootloader commands

Command	Description
RX data block	The receive data block command is used for write access to the flash memory (0x9000-0xDFFF). This command erases the flash memory data block before writing a new data.
RX password	Dummy command. Returns ACK. (Password is not required)
Mass erase	Dummy command. Returns ACK. (Mass erase is not required because the receive data block command erases the flash memory data block before writing)
Change baud rate	Change the baud rate of the bootloader. Supported baud rates: 0: 9600 Baud, 1: 19200 Baud, 2: 38400 Baud, 3: 57600 Baud, 4: 115200 Baud
TX data block	The transmit data block command is used to read back the last saved encrypted data block or the boot-ROM device information (16 bytes from 0x0FF0 to 0x0FFF). The user cannot read the contents of the flash or the RAM with this command.
TX BL version	The transmit bootloader version command is used to read version information.

2.3 Data Frame

The bootloader uses the same data frame protocol as the Texas Instruments Bootstrap Loader (BSL). The following chapters are modified from TI's document *MSP430 Programming Via the Bootstrap Loader* (www.ti.com/lit/ug/slau319i/slau319i.pdf). The bootloader interface has more limited functionality than the BSL interface.

2.3.1 Data Frame Structure

Data frame protocol used by the bootloader is presented in Table 2. The bootloader is considered the receiver.

HDR:	Header
CMD:	Command identification
L1, L2:	Number of bytes consisting of AL through Dn . Restrictions: L1 = L2, L1 < 255, L1 even
AL, AH:	Block start address
LL, LH:	Number of pure data bytes (250 max)
D1 ... Dn:	Data bytes
CKL, CKH:	16-bit checksum LO and HI byte
ACK:	The acknowledge character returned by the bootloader. DATA_ACK = 90h: Frame was received correctly and command was executed successfully. DATA_NAK = A0h: Command was not executed successfully.

Table 2 Data frame of bootloader commands

Received BL command	HDR	CMD	L1	L2	AL	AH	LL	LH	D1	D2...Dn	CKL	CKH	ACK*
RX data block	80	12	n	n	AL	AH	n-4	0	D1	D2...Dn-4	CKL	CKH	ACK
RX password	80	10	24	24	xx	xx	xx	xx	D1	D2...D20	CKL	CKH	ACK
Mass erase	80	18	4	4	xx	xx	6	A5	–	– – –	CKL	CKH	ACK
Change baud rate	80	20	4	4	xx	xx	D3	xx	–	– – –	CKL	CKH	ACK
TX data block	80	14	4	4	AL	AH	n	0	–	– – –	CKL	CKH	–
BL responds	80	xx	n	n	D1	D2...Dn	CKL	CKH	–
TX BL version	80	1E	4	4	xx	xx	xx	xx	–	– – –	CKL	CKH	–
BL responds	80	xx	10	10	D1	D2... D10	CKL	CKH	–

*Note: ACK is send by the bootloader.

n: Number of bytes consisting of **AL** through **Dn**
xx: Can be any data
–: No character (data byte) received or transmitted

- The first eight bytes (HDR through LH) are mandatory (xx represents dummy data).
- Data bytes D1 to Dn are optional.
- Two bytes (CKL and CKH) for checksum are mandatory.
- Acknowledge done by the bootloader is mandatory, except with the TX data block command
- The synchronization sequence is not part of the data frame.
- All numbers are bytes in hexadecimal notation.

Invalid data frame (e.g. wrong checksum, L1 ≠ L2) terminates firmware upgrade session.

2.3.2

Checksum

The 16-bit (2-byte) checksum is calculated over all received or transmitted bytes B1 to Bn in the data frame (except the checksum bytes themselves) by XORing words (two successive bytes) and inverting the result. This means that B1 is always the HDR byte and Bn is the last data byte just before the CKL byte.

Formula:

$$\text{CHECKSUM} = \text{INV} [(\text{B1} + 256 \times \text{B2}) \text{ XOR } (\text{B3} + 256 \times \text{B4}) \text{ XOR } \dots \text{ XOR } (\text{B}_{n-1} + 256 \times \text{B}_n)]$$

Or alternatively:

$$\text{CKL} = \text{INV} [\text{B1} \text{ XOR } \text{B3} \text{ XOR } \dots \text{ XOR } \text{B}_{n-1}]$$

$$\text{CKH} = \text{INV} [\text{B2} \text{ XOR } \text{B4} \text{ XOR } \dots \text{ XOR } \text{B}_n]$$

2.3.3 Simple Programming Flow

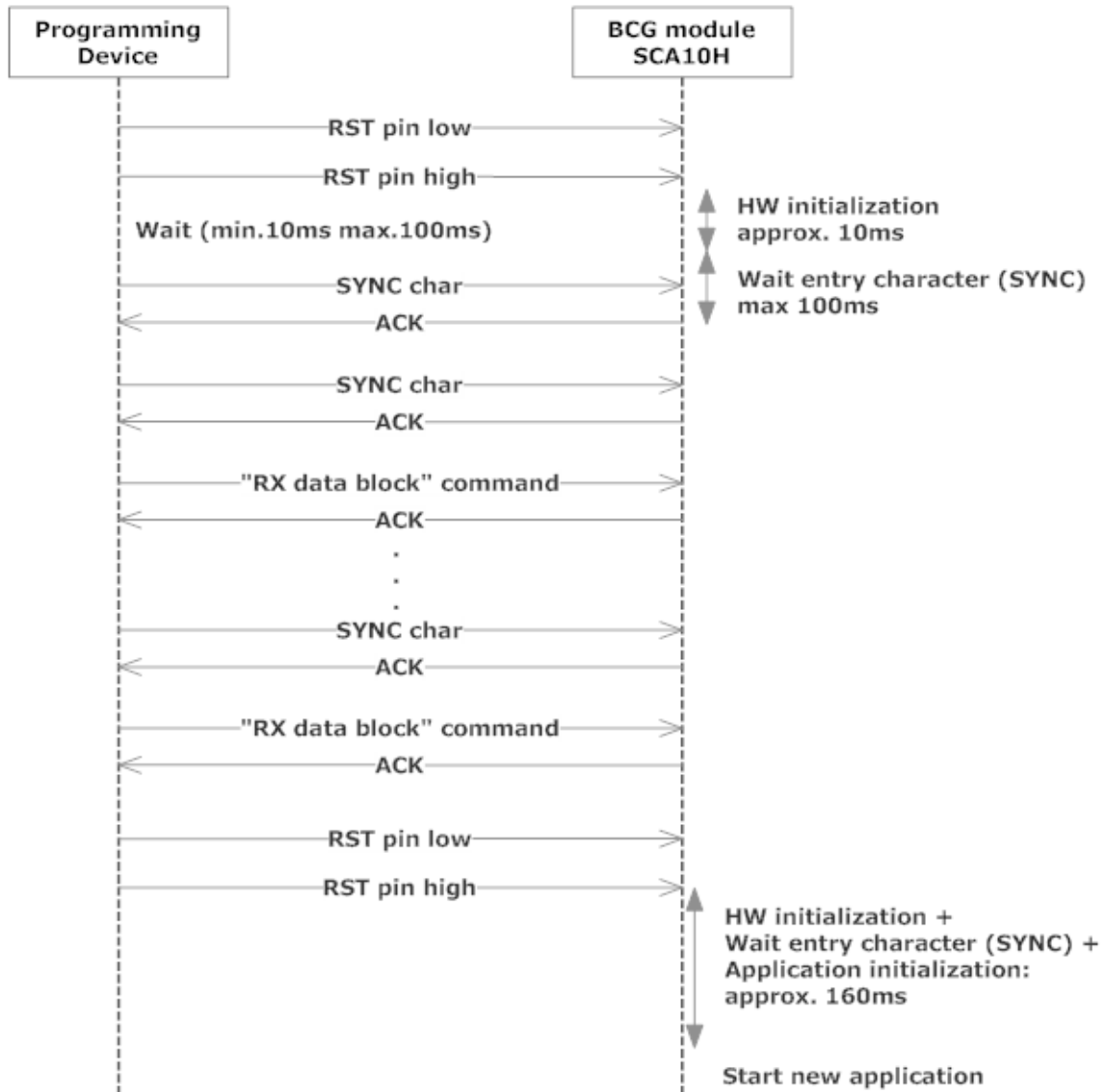


Figure 3 Simple programming flow

2.3.4 Commands – Detailed Description

2.3.4.1 RX Data Block

The receive data block command is used for write access to the flash memory (0x9000-0xDFFF). Bootloader requires data blocks in exact order from 0x9000 to 0xDFFF. It increases internal start address every time it receives a new data block. Bootloader does not use AL (low byte) and AH (high byte) to determine the start address.

The 16-bit even-numbered block length is defined in LL (low byte) and LH (high byte). Because pure data bytes are limited to a maximum of 250, LH is always 0. The following data bytes are succeeded by the checksum bytes CKL (low byte) and CKH (high byte).

If programming of the appropriate data block was successful, an acknowledge character DATA_ACK is sent back by the bootloader.

2.3.4.2 RX Password

Dummy command. This command is supported to maintain compatibility with the BSL interface. If the password was received correctly, a positive acknowledge DATA_ACK is sent back by the bootloader. The bootloader does not verify the password.

2.3.4.3 Mass Erase

Dummy command. This command is supported to maintain compatibility with the BSL interface.

All parameters shown in Table 2 are mandatory. An acknowledge character DATA_ACK is sent back by the bootloader.

2.3.4.4 Change Baud Rate

The change baud rate command offers the capability of transmissions at higher baud rates than the default 9600 baud.

Control byte D3 determines the selected baud rate:

- 0: 9600 Baud,
- 1: 19200 Baud,
- 2: 38400 Baud,
- 3: 57600 Baud,
- 4: 115200 Baud

After receiving the data frame, an acknowledge character DATA_ACK is sent back, and the UART is initialized to use the selected baud rate.

2.3.4.5 TX Data Block

The transmit data block command is used to read back the last saved encrypted data block or the boot-ROM device information (16 bytes from 0x0FF0 to 0x0FFF). The user cannot read the content of the flash or the RAM with this command.

The 16-bit block start address is defined in AL (low byte) and AH (high byte). When reading back the last saved encrypted data block, AL and AH must be same as used in the last RX Data Block command.

The 16-bit block length is defined in LL (low byte) and LH (high byte). Because pure data bytes are limited to a maximum of 250, LH is always 0. The checksum bytes CKL (low byte) and CKH (high byte) immediately follow this information.

The bootloader responds with the requested data block. After transmitting HDR, dummy CMD, L1 and L2, the bootloader sends data bytes D1 through Dn, followed by the checksum bytes CKL (low byte) and CKH (high byte). No acknowledge character is sent.

2.3.4.6 TX BL version

The transmit bootloader version command is used to read version information.

The values for AL, AH, LL, and LH can be any data, but must be transmitted to meet the protocol requirements. The checksum bytes CKL (low byte) and CKH (high byte) follow this information.

The bootloader responds with a 16-byte data block. After transmitting HDR, dummy CMD, L1 and L2, the bootloader sends data bytes D1 through D16 (decimal), followed by the checksum bytes CKL (low byte) and CKH (high byte). No acknowledge character is sent.

- D11: Bootloader version (high byte)
- D12: Bootloader version (low byte)
- D1-D10,D13-D16: Reserved

2.4 Exiting the FW Upgrade Mode

To exit the FW upgrade mode, two possibilities are provided:

- Applying the standard RESET sequence to RST/NMI pin. (Recommended)
- Waiting for a 10 s timeout to occur after the last sent command.

3 Example: FW Upgrade via USB to UART Bridge

SCA10H FW can be upgraded using a PC and a simple USB to UART bridge. In this example a Silabs cp2102 USB to UART bridge is used.

Requirements:

- SCA10H
- PC
- USB to UART bridge (Silabs cp2102)
- USB cable
- SCA10HBootloaderDemo.exe command line software

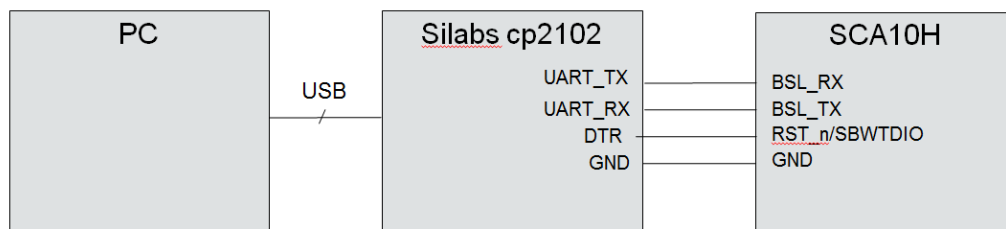


Figure 4 HW interface

1. Connect the Silabs cp2102 to SCA10H as shown in Figure 3.
2. Connect USB cable between the PC and the Silabs cp2102
3. Open the command prompt in PC
4. Run the SCA10HBootloaderDemo.exe

To upgrade the FW:

```
SCA10HBootloaderDemo.exe -cCOM{port number} -s{baud rate num} +pr
program_file.txt
```

Show Help:

```
SCA10HBootloaderDemo.exe -h
```

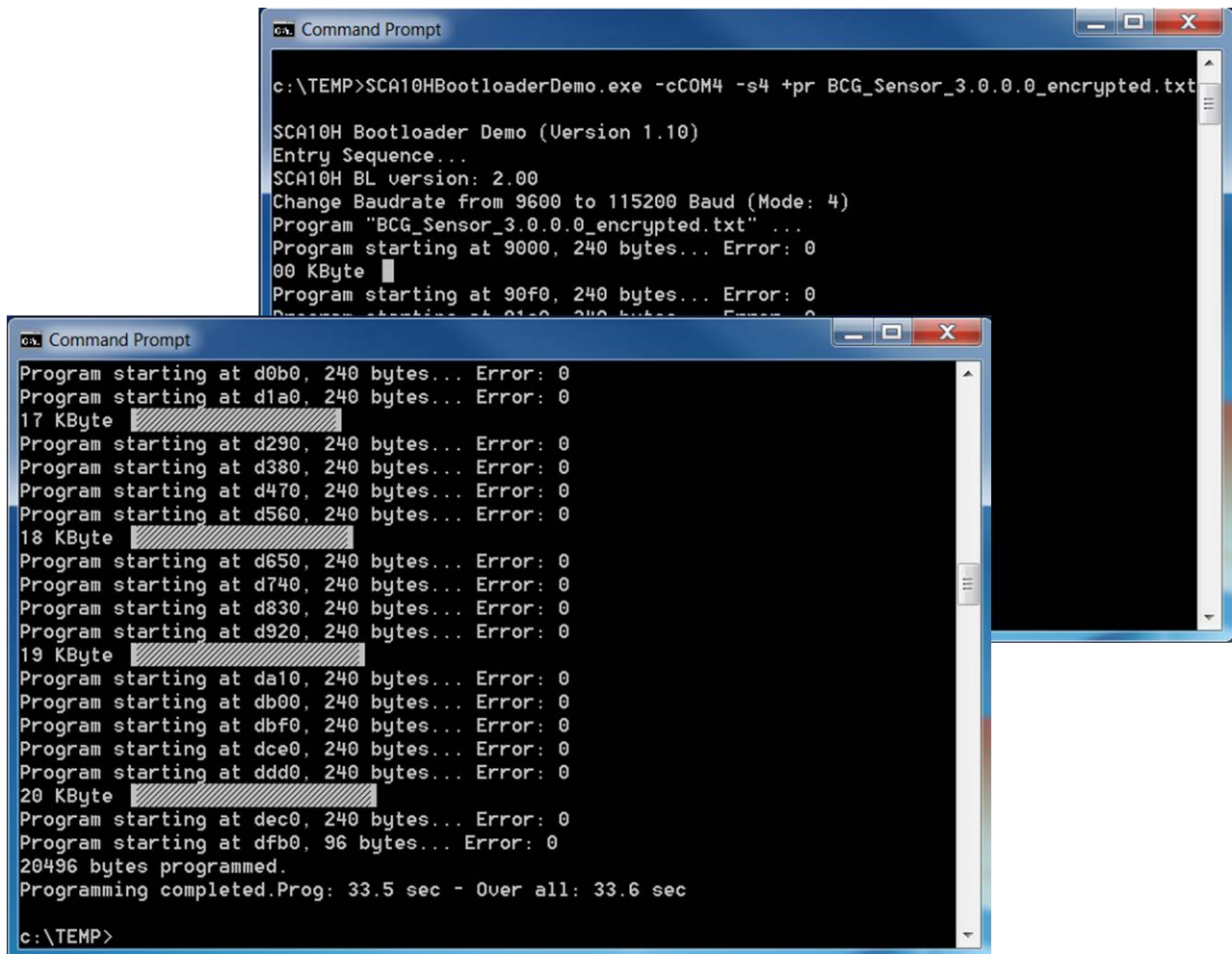


Figure 5 Programming view

4 Document Change Control

Rev.	Date	Change Description
1	10-Sep-15	Document moved to new control system and template. Added chapter "Example: FW Upgrade via USB to UART Bridge".
2	26-Jan-16	Bootloader example updated. RX Data Block and TX Data Block description updated.
3	7-Sep-16	Documentation for TX BL version command added.