Atmel AVR2054: Serial Bootloader User Guide

Features

- **Description of the Serial Bootloader utility, including**
  - **Application overview**
  - **Instructions on how to use Serial Bootloader**
  - **The Serial Bootloader programming algorithm**

Introduction

Serial Bootloader is a stand-alone utility consisting of two parts: embedded bootstrap code that should be loaded to the flash memory of a supported MCU and the PC based application that sends data to the embedded bootstrap over serial link. Embedded bootstrap code uses the received data to program the internal flash memory and/or EEPROM of the MCU. A simple communication protocol is used to ensure proper programming. Motorola S-record (SREC) format files are supported as source images for the serial bootloader PC part.

8-bit Atmel Microcontrollers

Application Note

Rev. 2054B-AVR-05/11

1 References

[1] AVR Studio® User Guide. Available in HTML Help with the product.
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

[2] JTAGICE mkII Quick Start Guide.
http://www.atmel.com/dyn/resources/prod_documents/doc2562.pdf

2 Supported Platforms

Serial Bootloader is supported on a set of Atmel® microcontrollers and development boards as shown in Table 2-1. Note that Serial Bootloader can work with different serial interfaces. Particular serial interfaces to be supported by bootstrap firmware shall be specified in the Makefile of the embedded Serial Bootloader code at compile time.

Table 2-1. Supported MCUs and corresponding boards and modules

MCU	Supported Modules	Development Board
ATmega1281	ATZB-24-B0, ATZB-24-A2, ATZB-A24-UFL, ATZB-900-B0	MeshBean, RCB230/231/212
ATmega128RFA1	N/A	STK600, RCB128RFA1 (mounted on RCB Sensor Terminal Board, RCB Key Remote Control board, or RCB-BB), ATAVR128RFA1-EK1
ATxmega256A3, ATxmega256D3	N/A	STK600

3 Using Serial Bootloader

To install the Serial Bootloader PC application, launch the installation file and proceed with instructions. After setup is completed both GUI and console versions of the bootloader are extracted to the provided installation path.

To program a wireless device using Serial Bootloader the following steps shall be done:

1. Connect a device with embedded Serial Bootloader firmware on it to a PC via serial connection. For detail refer to documentation of the development kit or the software provided with it.
2. For the GUI version of Serial Bootloader run the application by double-clicking the `bootloadergui.exe` file. For the console version just start command line from the Serial Bootloader installation directory.
3. Specify uploading parameters.

   The console version of Serial Bootloader accepts the following command-line options (can be entered in any order):

   ```
   bootloader -f <file_name> -p <com_port>
   ```

   For example, `<com_port>` can equal `COM3` to establish connection with the device through the COM port number 3. Besides, if the `-e <eeprom_size>` option is specified the EEPROM section will be cleared. To see help information run

   ```
   bootloader -h
   ```

   The GUI version contains interface controls for the same set of options as described in Table 3-1. The GUI version window is shown on Figure 3-1.

Table 3-1. Command-line options and corresponding GUI controls

Command line option	GUI control	Description
`-f`	`Select SREC file`	Path to the firmware image of SREC format to be loaded to device
`-p`	`Select Serial port`	COM port number
`-e`	`EEPROM Erase`	Clean up EEPROM section. EEPROM size shall be specified.

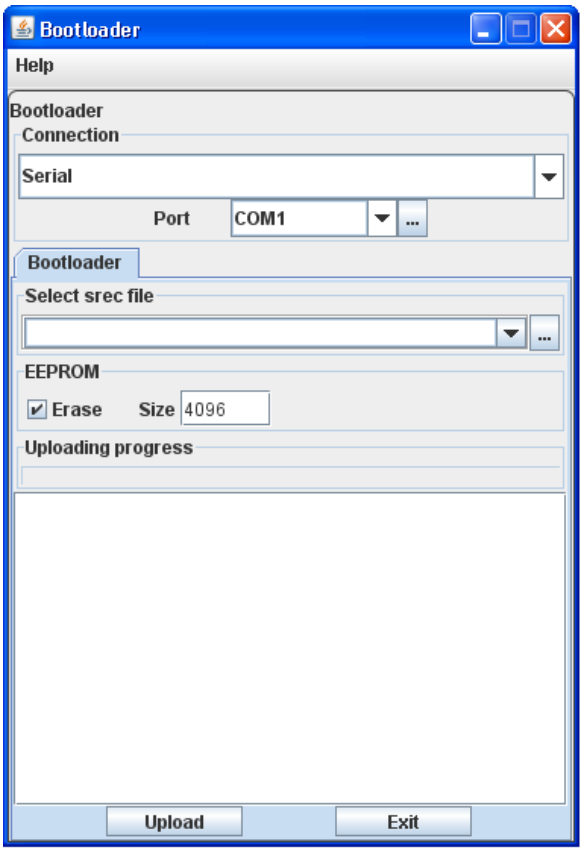
As a firmware image Serial Bootloader requires a file in the Motorola S-record hexadecimal format, also known as SREC or S19 format. Such file names have the `.srec` extension and can contain both flash memory and EEPROM images.

NOTE

There is a restriction on the size of firmware downloadable by serial booting process. Serial Bootloader cannot rewrite the area where the bootstrap code resides.

4. Press the `Upload` button if Serial Bootloader GUI is used. For the console bootloader press `Enter` on the keyboard to request for uploading process start.
5. Press the HW reset button on the device if requested. Serial Bootloader will be waiting for approximately 30 seconds for the button to be released. If this does not happen, programming will be aborted.
6. Serial Bootloader will indicate the programming progress. Once loading is finished successfully, the device will be restarted automatically. If loading fails, Serial Bootloader will indicate the reason. In rare cases, loading process could fail due to communication errors between the device and the PC. If this happens, try to program the device one more time. If loading still fails, the previous code programmed into the device could be corrupted, and the device should be reprogrammed again.

Figure 3-1. Serial Bootloader main window



4 Programming Embedded Bootstrap

If not already present on the MCU the firmware part of Serial Bootloader (`.hex` file) shall be uploaded to the device to enable programming over serial interface. AVR Studio [1] and Atmel JTAGICE mkII emulator [2] are recommended for loading embedded part of Serial Bootloader to the MCU. Note that to work correctly the embedded bootloader requires specific fuse bit settings that depend on the target platform and bootloader functionality.

CAUTION

Be careful when using JTAG for setting up the fuse bits. If you set wrong fuse bits by JTAG, your device would not work correctly.

5 Serial Bootloader Programming Algorithm

On the MCU side:

1. After the device is reset bootstrap code waits 500ms for a HANDSHAKE_REQ data sequence. If no HANDSHAKE_REQ is received, the bootstrap jumps to the entry point of the program in flash memory (if any).
2. If a HANDSHAKE_REQ sequence is received, bootstrap code sends a HANDSHAKE_CONF sequence and starts receiving SREC records of format shown in Table 5-1.

3. For every valid SREC record the bootstrap responds with an ACK data sequence.
4. In case of any error during loading process, bootstrap code sends a NACK data sequence, then proceeds to (1).

On the PC side:

1. The PC bootloader sends a HANDSHAKE_REQ data sequence for 30 seconds with 200ms interval, waiting for a HANDSHAKE_CONF data sequence between sends. Any reply except HANDSHAKE_CONF is ignored.
2. If HANDSHAKE_CONF is received, the PC bootloader starts sending data from the SREC file via the serial link. Each record from the SREC file is converted to binary representation before sending.
3. For every record sent the ACK is expected to be received over the serial link in response. If a NACK sequence is received or a timeout occurs, the PC bootloader aborts.

Data sequences used:

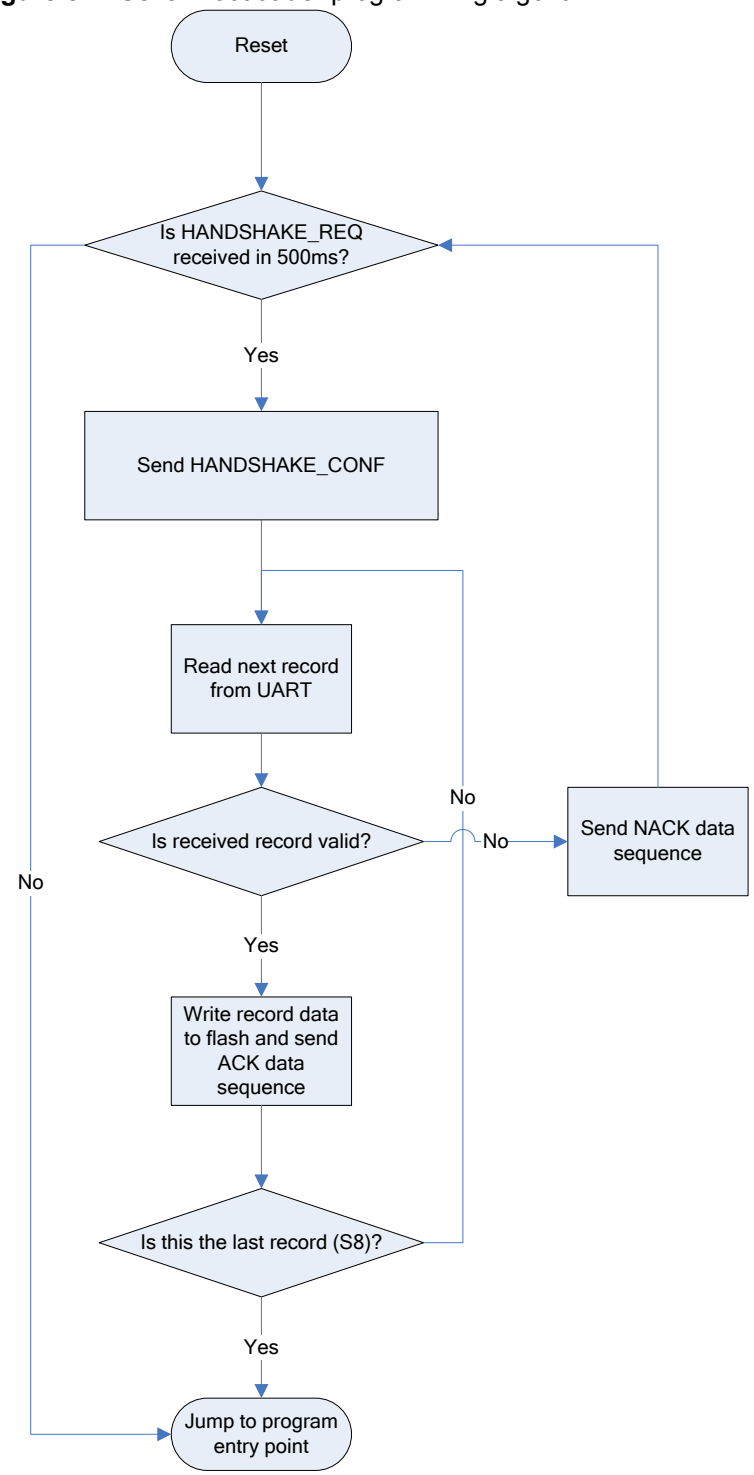
- HANDSHAKE_REQ: 0xB2, 0xA5, 0x65, 0x4B
- HANDSHAKE_CONF: 0x69, 0xD3, 0xD2, 0x26
- ACK: 0x4D, 0x5A, 0x9A, 0xB4
- NACK: 0x2D, 0x59, 0x5A, 0xB2

Table 5-1. Binary representation of the SREC record

Field name	Length (bytes)
Record type	2
Address length	1
Address	4
Record size	1
Record data	Variable
Boot section size	1024

Figure 5-1 illustrates the flowchart of the Serial Bootloader programming algorithm.

Figure 5-1. Serial Bootloader programming algorithm



Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Milennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chou-ku, Tokyo 104-0033
JAPAN
Tel: (+81) 3523-3551
Fax: (+81) 3523-7581