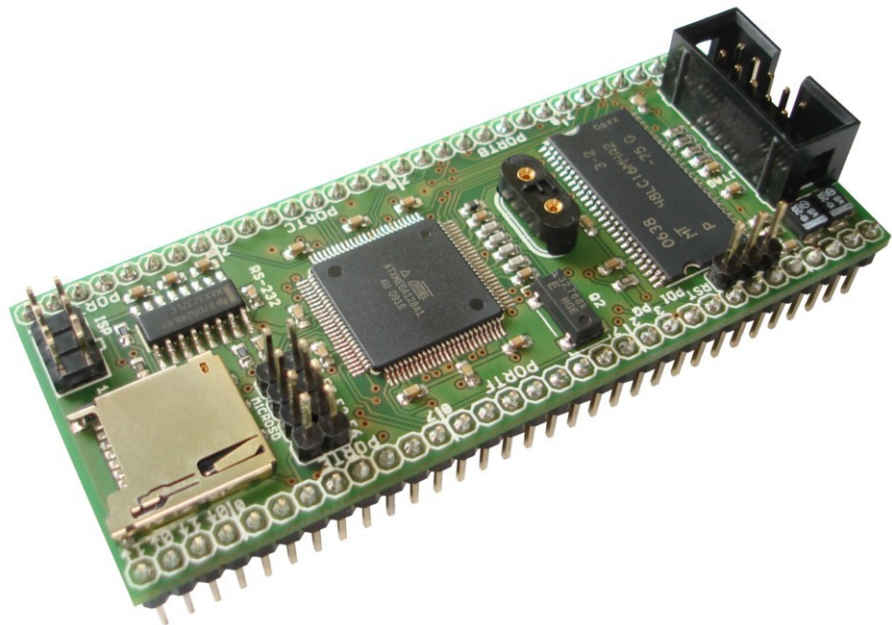
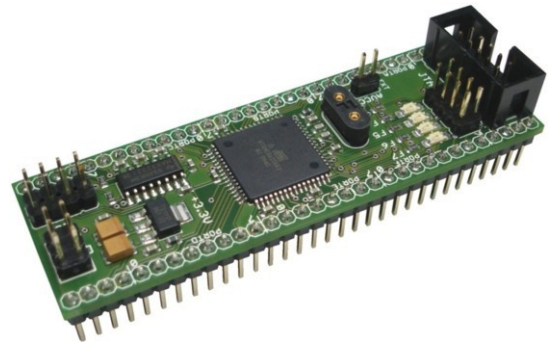


BOOTLOADER

for AVR-Development Modules

- with ATxmega128A1
- with ATxmega256A3



Directory

1.	Instructions	3
2.	Settings	4
2.1.	Fuse Bits	4
2.2.	AVROSP Program	5
2.2.1.	Explanation of BAT-Files	5
2.2.2.	Environment Variables	6
2.3.	Program-Files	7
2.4.	Hardware	8
3.	Re-installation	9
4.	Programming	9
5.	Links	11

1. Instructions

This document describes the boot loader of Atmel in connection with our AVR module AL-XAVRB. More information about this boot loader you will find in the PDF-application **AVR1605: XMEGA Boot Loader Quick Start Guide** (see page 11, chapter 5. **Links**).

This application describes every single step once again. We start with the settings of the fuse bits, explanation of the programming setup, than make the necessary settings in Windows, introduce the additional periphery and program the test file. You may download the source code for the boot loader on the Atmel Homepage, you will find the link to it in chapter 5 of this document. We offer only ready compiled hex-files with two transfer rates: 9600 bit/s and 115200 bit/s (see chapter 2.3. Program-Files). You will find self-executing BAT-file and a test-program to every hex-file. Internal frequency of 2 MHz oscillator will be used as quartz.

For programming you will need:

- AVROSP.exe - Program of Atmel

You will find this program in the ZIP-Application **AVR1605: XMEGA Boot Loader Quick Start Guide** (see page 11, chapter 5. **Links**)

- Hardware with ATxmega (128A1 or 256A3), e.g. AVR module AL-XAVRB

This module needs an additional external periphery: reset-key, pin-key with pull-up and D-SUB 9 female connector. You will get more details in the chapter **2.4. Hardware**.

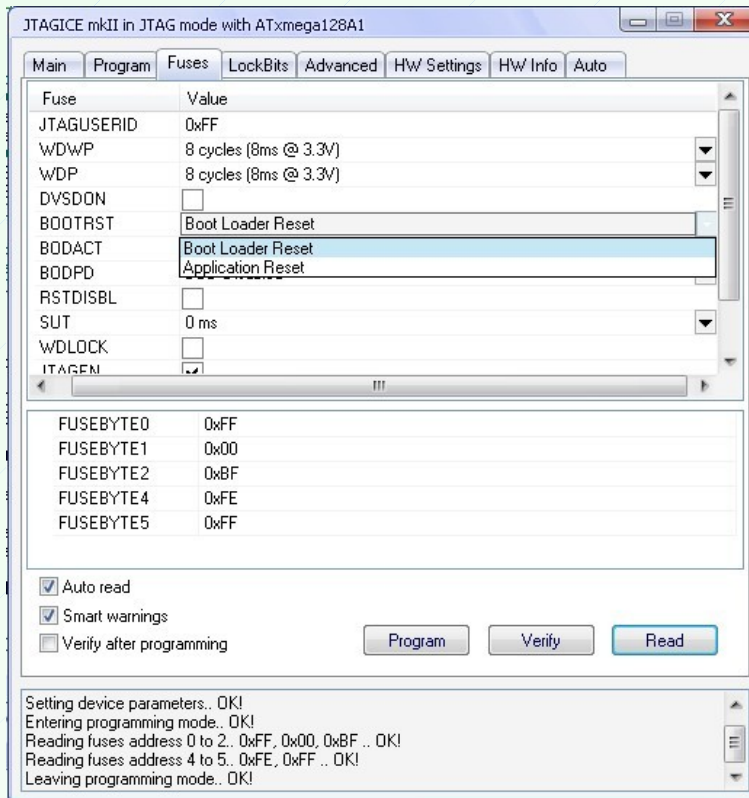
- Serial communication with PC, e.g. COM1

Port COM1 will be mostly used as standard serial communication. In other case port should be set between COM1 and COM8.

2. Settings

2.1. Fuse Bits

Settings of the Fuse Bits with AVR Studio 4

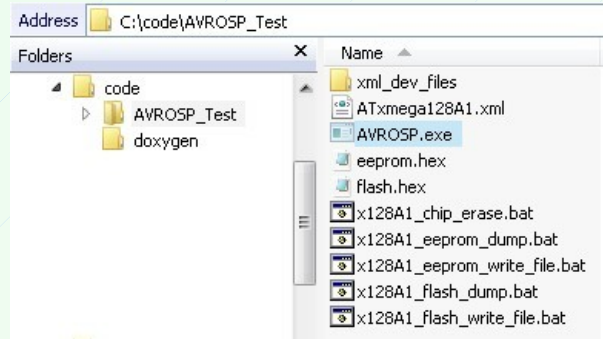


The Fuse bit „**BOOTRST**“ should be set for programming with boot loader. The picture above shows what settings you should make in AVR Studio 4:

Tools → *Program AVR* → *Connect ...* → *Connect ...* → *Fuses* „**Boot Loader Reset**“.

2.2. AVROSP Program

Download the ZIP-file: **AVR1605: XMEGA Boot Loader Quick Start Guide** and extract it e.g. on C:\ In this case you will find AVROSP Program in folder C:\code\AVROSP_Test



2.2.1. Explanation of BAT-Files

In the folder *AVROSP_Test* you will find also some examples: *x128A1_chip_erase.bat* : as the name says, this file performs deleting of the chip memory (in application section) except the boot loader section.

With help of *x128A1_flash_write_file.bat* we try to understand how the programming of the flash works. By opening the file with a notepad e.g. Windows Notepad *Start* → *Programs* → *Accessories* → *Notepad* we get the following code:

```
mode com1 Data=8 Parity=n Baud=9600 DTR=OFF RTS=OFF
AVROSP -dATxmega128A1 -e -iflash.hex -pf -vf
pause
```

The serial connection will be set in the first line. In this case port COM1 with baud rate of 9600 bit/s will be taken.

The second line sends the following commands to the AVROSP program :

<code>-dATxmega128A1</code>	name of the controller
<code>-e</code>	chip erase
<code>-iflash.hex</code>	program with hex-file „flash.hex“
<code>-pf</code>	program the flash
<code>-vf</code>	compare the flash with hex-file

pause After executing the commands above, the last line keeps the program window opened.

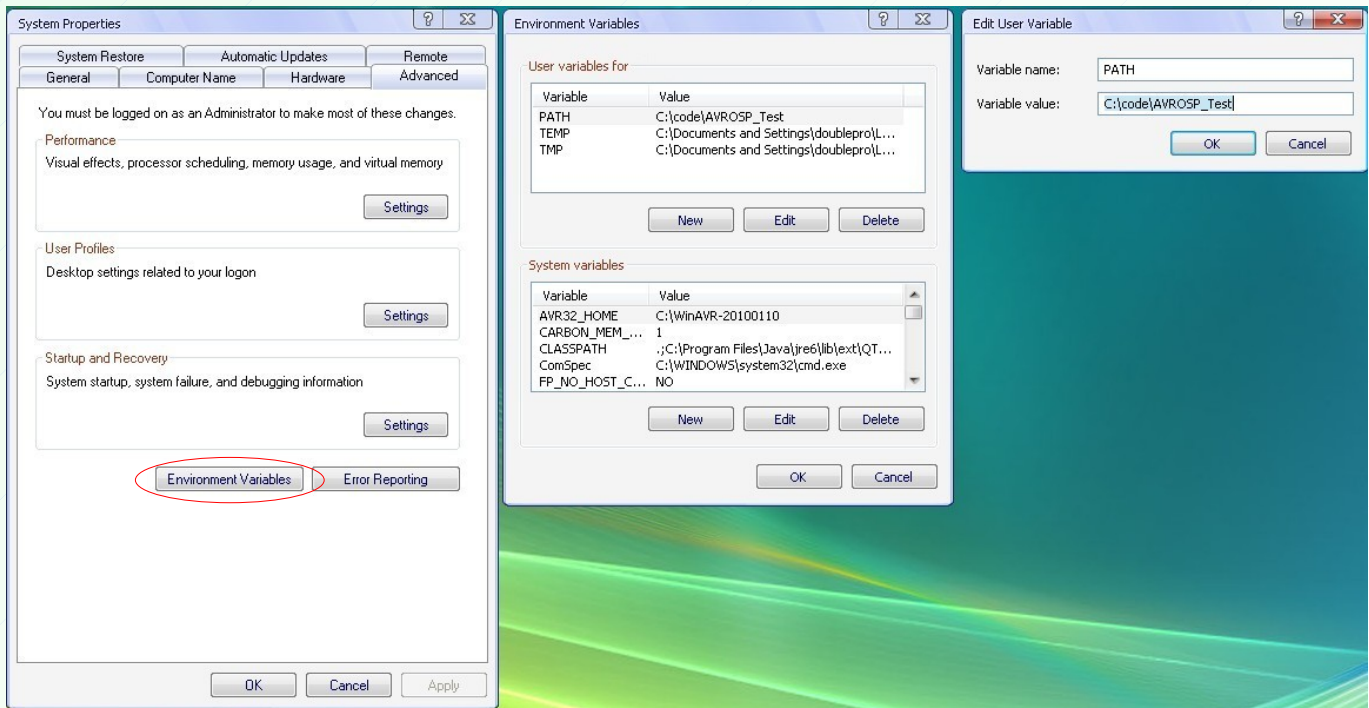
2.2.2. Environment Variables

With the setting of the environment variables we make sure that the programming of the hex- file started in the project folder and the hex-file will not be transferred into the folder **C:\code\AVROSP_Test** with every new programming of the flash.

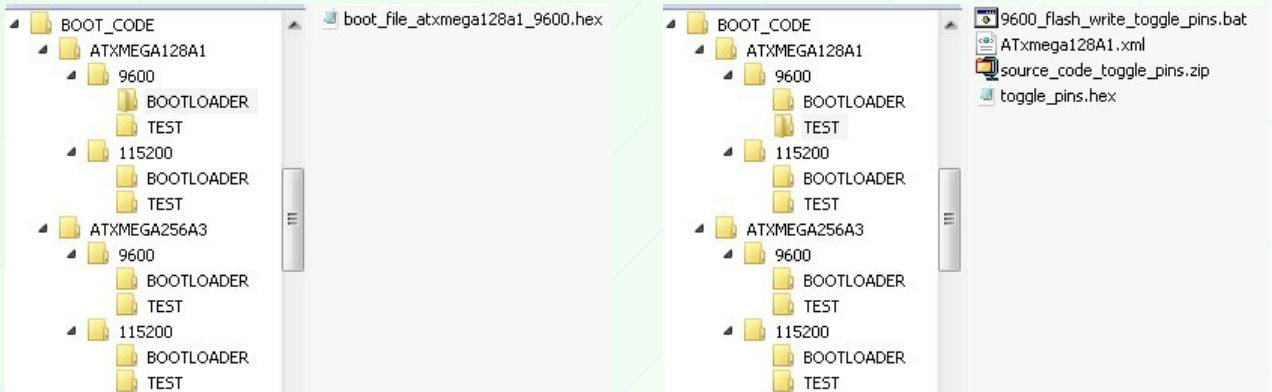
You have to make the following settings to get this result:

- click with the right mouse key on *My Computer* → *Properties Advanced* → *Environment Variables*
- set or add in the window Environment Variables the "PATH" variable with the link to the AVROSP program. In our case it is **C:\code\AVROSP_Test**
- when the PATH already exists, you can enter the link to the AVROSP program behind the existing link with the ; - semicolon sign.

The lower picture shows the whole setting process.



2.3. Program-Files



[BOOT_CODE.zip](#) The archive contains two sub-folders ATXMEGA128A1 and ATXMEGA256A3. In every folder there are two other sub-folders: 9600 and 115200. The name points out to the transfer rate. In every transfer rate-folder there are two more folders: the one with the boot loader (see the left upper picture) and the other with a test program (right picture).

BOOT_CODE\ATXMEGA128A1\9600\BOOTLOADER: - in this folder you will find the hex-file „*boot_file_atxmega128a1_9600.hex*“. The name of the hex-files describes the type of the file, controller compatibility and baud rate.

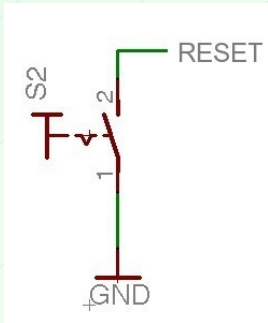
BOOT_CODE\ATXMEGA128A1\9600\TEST: - this folder contains: the test-program „*toggle_pins.hex*“ with the related source code in ZIP-format „*source_code_toggle_pins.zip*“ and a self-executing file „*9600_flash_write_toggle_pins.bat*“ with XML-file „*ATxmega128A1.xml*“. BAT-file was adapted to the test program „*toggle_pins.hex*“ and the boot loader „*boot_file_atxmega128a1_9600.hex*“:

```
mode com1 Data=8 Parity=n Baud=9600 DTR=OFF RTS=OFF
AVROSP -dATxmega128A1 -e -iftoggle_pins.hex -pf -vf
pause
```

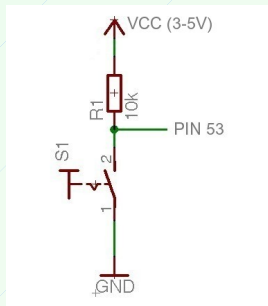
The explanation of every single line you will find in the chapter **2.2.1. Explanation BAT-Files**

2.4. Hardware

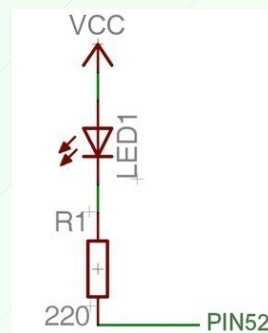
You will need an external circuit for programming with the boot program:



Connect the reset pin of module, with a e.g. key, the way as it is shown on the left picture.

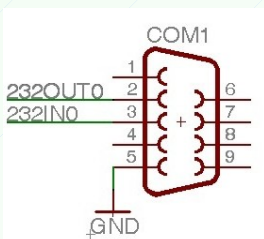


Connect the pull-up key to the pin 53 (**PQ2** by AL-XAVRB and **PF0** by AL-XSLED) as shown on the left picture.



As a check you may install a LED on pin 52 (**PF7** by AL-XAVRB and **PE7** by AL-XSLED) see the left picture.

As soon boot loader started, connected LED will be on.



Connect **D-Sub 9-pins female connector** with Module, pin 2 of D-Sub with output 0 (232out0) on the AVR module, pin3 of D-Sub with input 0 (232in0) and pin5 of D-Sub with the ground.

3. Re-installation

After each „*Chip Erase*“ with ISP-Programmer or JTAG-Debugger the whole memory inclusive the boot program will be deleted.

Re-installation of boot programs with AVR Studio

Please make sure before the installation that AVR studio is updated (at least. 4.18). For re-installation you can use the ready hex-files in the folder BOOTLOADER (see **2.3 Program-Files**).

After programming the favoured boot loader to the flash memory you have to set the fuse bit BOOTRST (see chapter **2.1. Fuse Bits**)

4. Programming

All the settings which we have made up to this chapter should be performed only once. After we have programmed the hex file of the boot loader in the boat loader section of the controller, set the fuse bit BOOTRST and equipped the AVR module AL-XAVRB with the external periphery, we can start with the programming.

- Connect the module with the serial interface of the PC e.g. COM1.
- Push pin 53 and reset key at the same time (external periphery see **2.4 Hardware**), after that let off the reset key before the pin 53 key .

The next step requires the execution of the chapter 2.2.2. Environment Variables

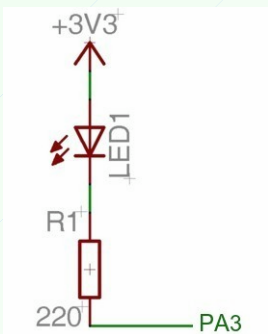
- open the compatible sub-folder to the programmed boot loader 9600\TEST bzw. 115200\TEST and make a double-click on the BAT-file.

After that you will see the following window:

```
ex C:\WINDOWS\system32\cmd.exe
Serial port timeout set to 10 sec.
Scanning COM ports for supported programmer...
COM1...
Found AVRBOOT on COM1!
Entering programming mode...
Parsing XML file for device parameters...
Parsing '.\ATxmega128A1.xml'...
#####
Saving cached XML parameters...
Signature matches device!
Erasing chip contents...
Reading HEX input file for flash operations...
#####
Programming Flash contents...
Using block mode...
##
Reading Flash contents...
Using block mode...
##
Comparing Flash data...
Equal!
Leaving programming mode...
C:\BOOT_CODE\115200\TEST>pause
Press any key to continue . . .
```

„Equal!“ means, that the flash-memory of the controller is equal to the content of the hex-file / the test program was transferred successfully.

If you want to make sure, that the test program is working properly, connect the resistor 220 Ω and the LED in a row to port A pin 3, as you can see on the lower picture. If the LED is on/off every 5 seconds, you have made everything right.



If you want to program your individual projects, copy BAT-, and XML-file from TEST folder and paste them to the hex-file folder of your project. Open the BAT-file with Notepad (see chapter 2.2.1. **Explanation of BAT-Files**) and make your settings.

Good luck!

5. Links

The necessary source code, libraries and descriptions of Atmel you will find under the following links:

- **AVR1316: Using the XMEGA Self-programming** [[PDF](#)] [[ZIP](#)]

This application describes the basic functionality of the XMEGA self-programming and presents the programming with the help of source code examples.

- **AVR1605: XMEGA Boot Loader Quick Start Guide** [[PDF](#)] [[ZIP](#)]

This application describes, how to use the boot loader with the XMEGA controller (e.g. ATxmega-128A1), and how an AVR with the Store Program Memory (SPM) instruction can be configured for Self-programming.