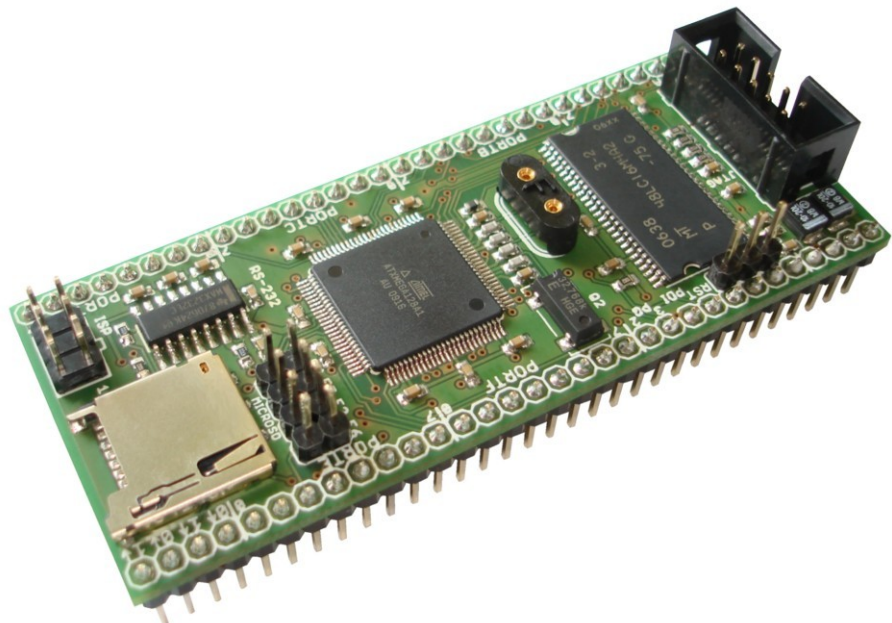
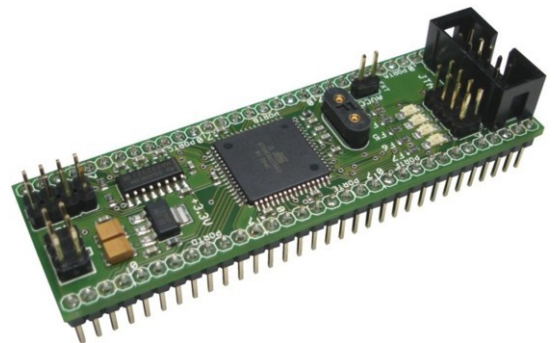


BOOTLOADER

für AVR-Entwicklungsmodule

- mit ATxmega128A1
- mit ATxmega256A3



Verzeichnis

1.	Einleitung	3
2.	Einstellungen	4
2.1.	Fuse Bits	4
2.2.	AVROSP Programm	5
2.2.1.	Erläuterung der BAT-Dateien	5
2.2.2.	Umgebungsvariablen	6
2.3.	Programm-Files	7
2.4.	Hardware	8
3.	Wiederinstallation	9
4.	Programmieren	9
5.	Links	11

1. Einleitung

In diesem Dokument stellen wir Ihnen den Bootloader von Atmel im Zusammenhang mit unserem AVR Modul AL-XAVRB vor. Mehr Information zu diesem Bootloader finden Sie im PDF-Applikation **AVR1605: XMEGA Boot Loader Quick Start Guide** (siehe Seite 11, Kapitell **5. Links**).

Dieses Dokument erläutert noch einmal die einzelne Schritte. Wir beginnen mit der Einstellung der Fuse Bits, Erläuterung des Aufbau der Programmierdatei, führen die Einstellung im Windows durch, stellen die zu ergänzende Peripherie dar und programmieren die Testdatei. Quellcode zum Bootloader können Sie auf der Atmel Seite herunterladen, den Link zum Quellcode finden Sie im Kapitell 5 dieses Dokuments. Wir stellen aber nur fertig kompilierte Hex-Files für zwei Übertragungsgeschwindigkeiten 9600 Bit/s und 115200 Bit/s zur Verfügung (siehe **2.3. Programm-Files**). Zu jedem Hex-File finden Sie auch eine selbst ausführende BAT-Datei und ein Testprogramm. Als Quarz wird der interne 2 MHz Oszillator benutzt.

Zum Programmieren benötigen Sie:

- AVROSP.exe - Programm von Atmel

Sie finden dieses Programm in der ZIP-Applikation **AVR1605: XMEGA Boot Loader Quick Start Guide** (siehe Seite 11, Kapitell **5. Links**)

- Hardware mit ATxmega128A1, z.B. AVR Modul AL-XAVRB

Das Modul benötigt zusätzliche externe Beschaltung: Reset-Taster, Pin-Taster mit Pull-Up und D-SUB 9 Buchse. Die detaillierte Beschreibung finden Sie im Kapitell **2.4 Hardware**.

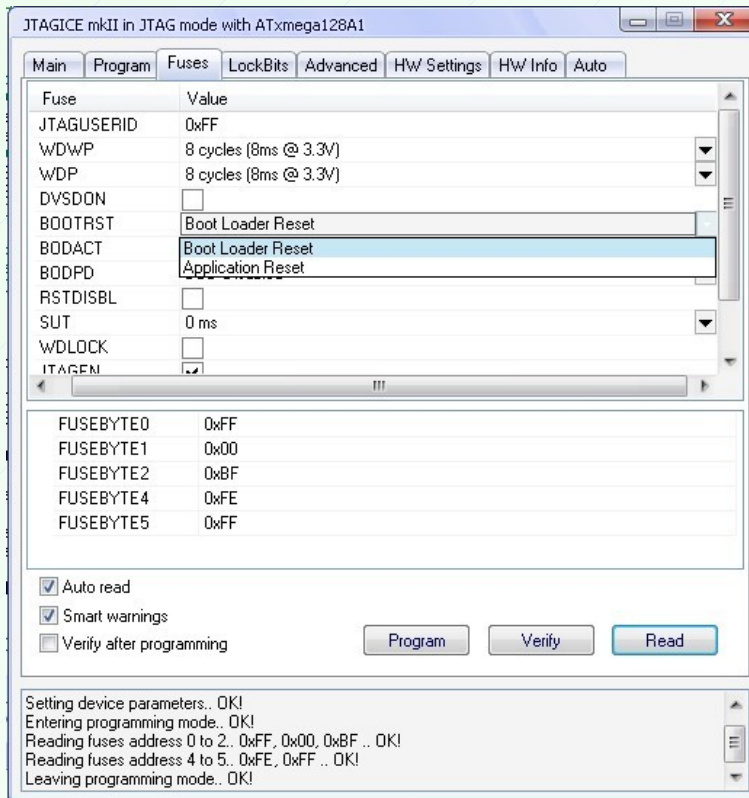
- Serielle Kommunikation mit PC, z.B. über COM1

Bei der seriellen Kommunikation wird meistens Port COM1 benutzt. Im anderen Fall soll der Port zwischen COM1 und COM8 liegen.

2. Einstellungen

2.1. Fuse Bits

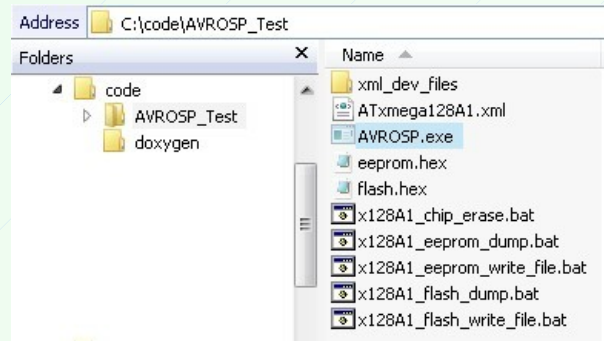
Einstellungen der Fuse Bits mit AVR Studio 4



Zum Programmieren mit Bootloader muss das Fuse Bit „**BOOTRST**“ gesetzt werden. Wie die obere Abbildung zeigt, wählt man im AVR Studio 4 unter *Tools* → *Program AVR* → *Connect ...* → *Connect ...* → *Fuses* „**Boot Loader Reset**“.

2.2. AVROSP Programm

Laden Sie **AVR1605: XMEGA Boot Loader Quick Start Guide** ZIP-Datei herunter und extrahieren z.B. direkt auf C:\ In diesem Fall finden Sie das AVROSP Programm unter C:\code\AVROSP_Test



2.2.1. Erläuterung der BAT-Dateien

Im Ordner *AVROSP_Test* finden Sie außer Programm AVROSP auch einige Beispiele. *x128A1_chip_erase.bat* : der Name verrät uns, dass diese Datei führt das Löschen des kompletten Speichers (Applikation Bereich) außer Boot Loader Bereich durch.

Anhand *x128A1_flash_write_file.bat* versuchen wir nachzuvollziehen, wie die Programmierung des Flash funktioniert. Wenn wir mit Hilfe eines Editors diese Datei öffnen, z.B. Windows Editor *Start* → *Programme* → *Zubehör* → *Editor*, bekommen wir Folgendes zu sehen:

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

In der ersten Zeile konfigurieren wir die serielle Verbindung. In diesem Fall ist es Port-COM1 mit der Btrate 9600 Bit/s

Zweite Zeile gibt dem Programm AVROSP folgende Befehle weiter:

<code>-dATxmega128A1</code>	Name des Controllers
<code>-e</code>	lösche den Flash Speicher
<code>-iflash.hex</code>	das zu programmierende Hex-File „flash.hex“
<code>-pf</code>	programmiere Flash
<code>-vf</code>	vergleiche Flash mit Hex-File

pause die letzte Zeile sorgt für das Anhalten des Programmierfensters nach der Ausführung der obigen Befehle.

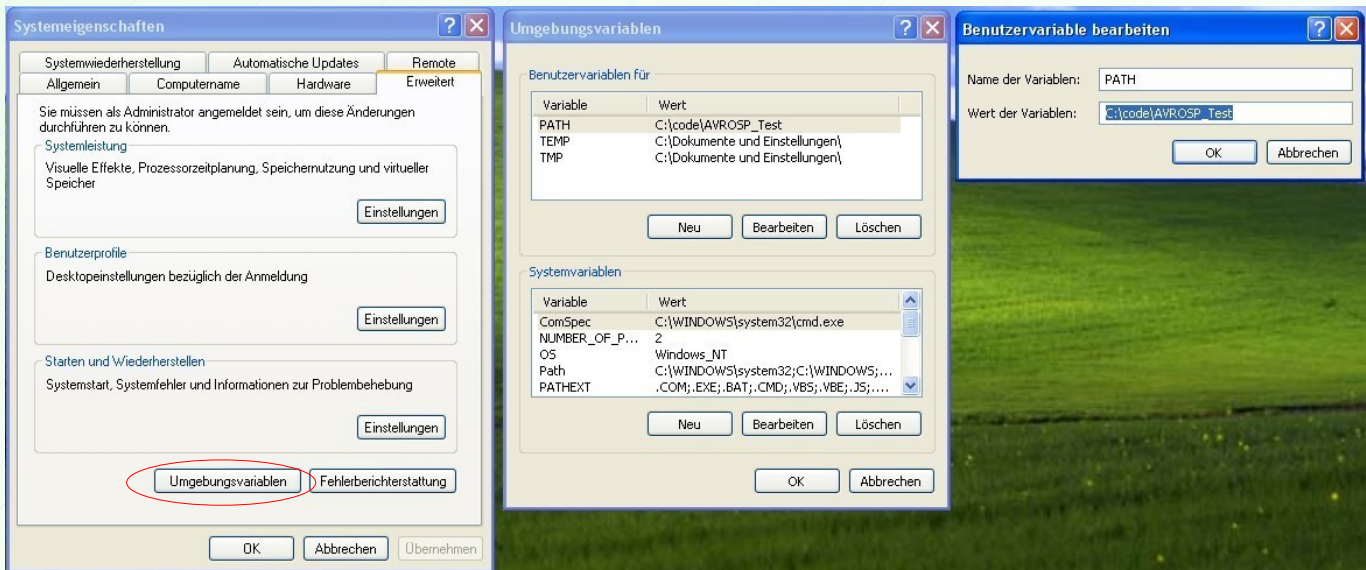
2.2.2. Umgebungsvariablen

Mit der Einstellung der Umgebungsvariablen sorgen wir dafür, dass die Programmierung des Hex-Files vom Projektordner gestartet wird und das Hex-File nicht bei jeder neuen Beschreibung des Flash in den Ordner **C:\code\AVROSP_Test** übertragen wird.

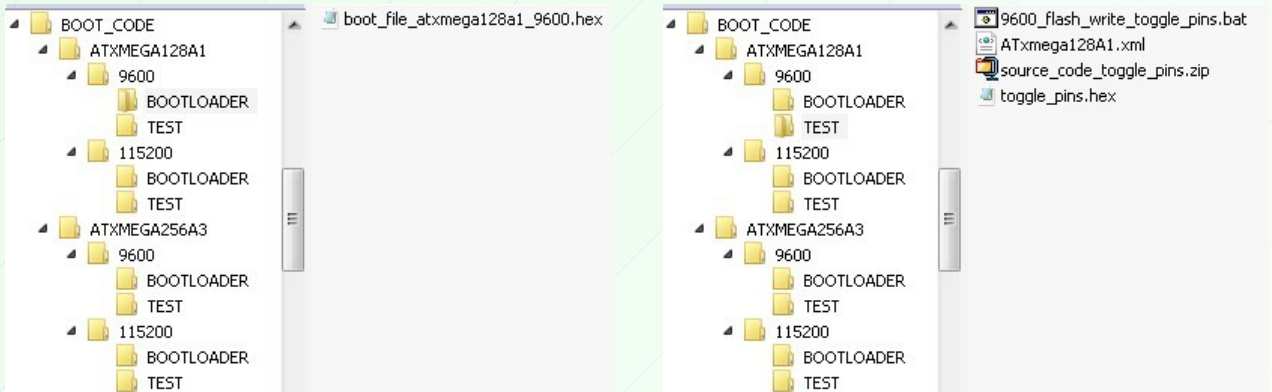
Dafür führen Sie die folgenden Einstellungen durch:

- klicken Sie mit der rechten Maustaste auf das *Arbeitsplatz* → *Eigenschaften* → *Erweitert* → *Umgebungsvariablen*
- erstellen bzw. ergänzen Sie im Fenster Umgebungsvariablen die Variable „*PATH*“ mit dem Link zum AVROSP Programm. In unserem Fall ist es **C:\code\AVROSP_Test**
- sollte ein *PATH* schon vorhanden sein, so können Sie hinter dem vorhandenen Link durch ;-Semikolon Zeichen das Link zum AVROSP Programm eingeben.

Das untere Abbild stellt den gesamten Ablauf dar.



2.3. Programm-Files



[BOOT_CODE.zip](#) Archiv beinhaltet zwei Unterordner ATXMEGA128A1 und ATXMEGA256A3. In jedem Ordner finden sich weitere zwei Unterordner 9600 und 115200. Die Bezeichnung weist auf die Übertragungsrates hin. In jedem Übertragungsrates-Ordner liegen zwei Ordner: mit dem Bootloader als Inhalt (siehe linke obere Abbildung) und mit Testprogramm (rechtes Abbild).

BOOT_CODE\ATXMEGA128A1\9600\BOOTLOADER: - in diesem Ordner liegt ein Hex-File „*boot_file_atxmega128a1_9600.hex*“. Der Name des Hex-Files bezeichnet Typ des Files, Controller Kompatibilität und Übertragungsgeschwindigkeit.

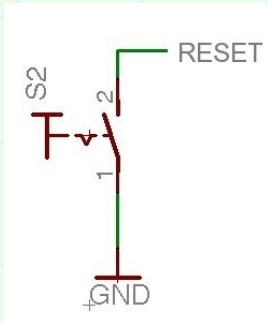
BOOT_CODE\ATXMEGA128A1\9600\TEST: - dieser Ordner beinhaltet: Test Programm „*toggle_pins.hex*“ mit der zugehörigen Quellcode im ZIP-Format „*source_code_toggle_pins.zip*“ und ein selbst ausführendes File „*9600_flash_write_toggle_pins.bat*“ mit XML-File „*ATXmega128A1.xml*“. BAT-Datei wurde dem Test Programm „*toggle_pins.hex*“ und dem Bootloader „*boot_file_atxmega128a1_9600.hex*“ angepasst:

```
mode com1 Data=8 Parity=n Baud=9600 DTR=OFF RTS=OFF
AVROSP -dATXmega128A1 -e -iftoggle_pins.hex -pf -vf
pause
```

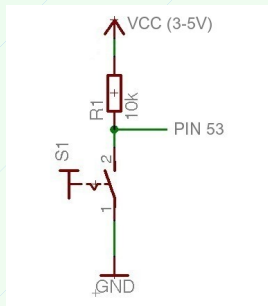
Die Erläuterung jeder einzelnen Zeile finden Sie im Kapitell **2.2.1. Erläuterung der BAT-Dateien**

2.4. Hardware

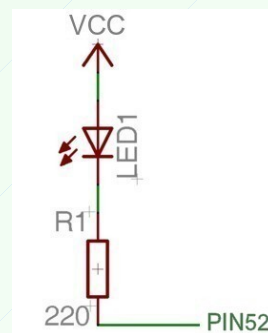
Zum Programmieren mit dem Bootloader benötigen Sie externe Beschaltung:



Beschalten Sie den Ausgang **Reset**, mit z.B. einem Taster, so wie es in der Abbildung links dargestellt ist.

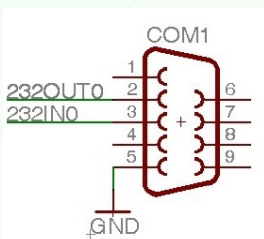


Schließen Sie an Pin 53 (**PQ2**- bei AL-XAVRB und **PF0** bei AL-XSLED) einen Pull-Up Taster wie im linken Abbild vorgeführt ist .



Zur Kontrolle können Sie an Pin52 (**PF7**- bei AL-XA-VRB und **PE7** bei AL-XSLED) eine LED, wie im linken Abbild dargestellt, anschließen.

Sobald Bootloader gestartet ist, leuchtet die angeschlossene LED.



Verbinden Sie **D-SUB 9 polige Buchse** mit dem Modul. Schließen Sie an den Pin2 der D-SUB Buchse den seriellen Ausgang 0 (**RS232out0**) des AVR-Moduls, an den Pin3 der D-SUB Buchse den seriellen Eingang 0 (**RS232in0**) des AVR-Moduls und an den Pin5 der D-SUB Buchse die Masse.

3. Wiederinstallation

Nach jedem „*Chip Erase*“ mit ISP-Programmer oder JTAG-Programmer wird der komplette Speicher, inklusive dem Bootloader, gelöscht.

Wiederinstallation des Bootloaders mit AVR Studio

Vor der Installation bitte vergewissern Sie sich, dass AVR-Studio auf dem aktuellen Stand ist (mindestens ver. 4.18). Zur Wiederinstallation können Sie schon fertige Hex-Files aus dem Ordner BOOTLOADER (siehe **2.3 Programm-Files**) benutzen.

Nachdem der gewünschte Bootloader in den Speicher geschrieben wurde, muss das Fuse-Bit BOOTRST gesetzt werden. Siehe Kapitel **2.1. Fuse Bits**

4. Programmieren

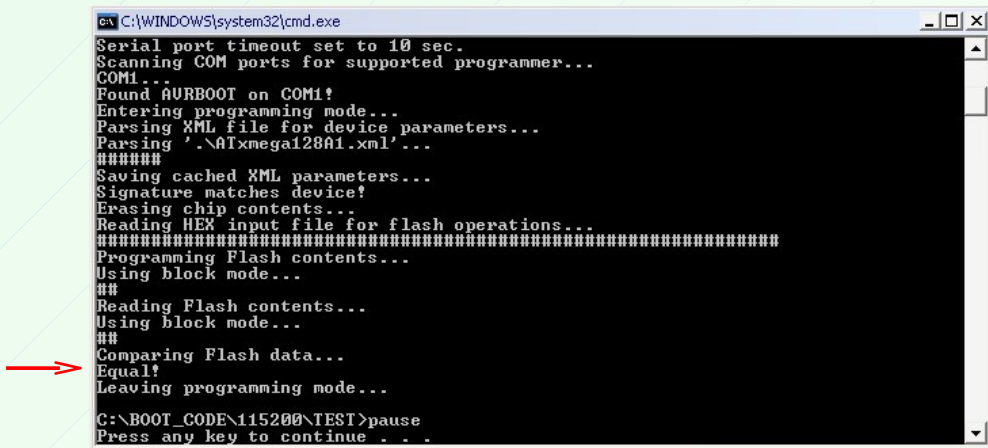
Alle Einstellungen, die wir bis zum diesen Kapitel durchgeführt haben, werden nur einmal gemacht. Nachdem wir das Bootloader Hex-File in das Boot Loader Section des Controllers eingespielt, BOOTRST Fuse Bit gesetzt und das AVR Modul AL-XAVRB mit der externen Beschaltung ausgestattet haben, können wir mit der Programmierung beginnen.

- Verbinden Sie das Modul mit dem Computer über die serielle Schnittstelle, z.B. COM1.
- Drücken Sie gleichzeitig Pin 53 und Reset Taster (externe Erweiterung s. **2.4 Hardware**), danach soll der Reset Taster vor dem Pin 53 Taster losgelassen werden.

Der nächste Schritt setzt voraus, dass der Kapitel 2.2.2. Umgebungsvariablen durchgeführt wurde.

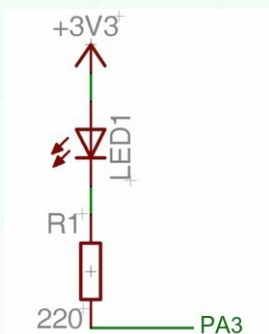
- öffnen Sie einen passenden Unterordner zum einprogrammierten Bootloader 9600\TEST bzw. 115200\TEST und klicken Sie doppelt auf die BAT-Datei.

Dabei erscheint folgendes Fenster:



```
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!“ deutet darauf hin, dass der Flash-Inhalt des Controllers gleich dem Inhalt des Hex-Files ist, bzw. das Test Programm wurde erfolgreich übertragen. Um sich zu vergewissern, dass das Test Programm richtig funktioniert, schließen Sie z.B. an Port A Pin 3 in einer Reihe Widerstand 220 Ω und ein LED an, wie im unteren Schaltplan. Wenn das LED im Takt von ca. 5 Sekunden blinkt, dann haben Sie alles richtig gemacht.



Zum Programmieren individueller Projekte kopieren Sie aus dem Ordner TEST BAT-, und XML-File und fügen Sie sie in den Hex-File Ordner Ihres Projektes ein. Öffnen Sie BAT-Datei mit einem Texteditor (siehe Kapittel 2.2.1. **Erläuterung der BAT-Dateien**) und passen Sie sie Ihrem Projekt an.

Viel Erfolg!

5. Links

Die notwendigen Quellcode, Bibliotheken und Beschreibungen von Atmel finden Sie unter diesen Links :

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

Diese Applikation beschreibt die grundlegende Funktionalität der XMEGA Selbst-Programmierung und stellt anhand der Quellcode-Beispiele die Programmierung dar.

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

Diese Applikation beschreibt, wie man den Bootloader mit einem der XMEGA Controller verwendet (z.B. ATxmega128A1), und wie ein AVR mit der Store Program Memory (SPM) Instruktion für die Selbstprogrammierung konfiguriert werden kann.