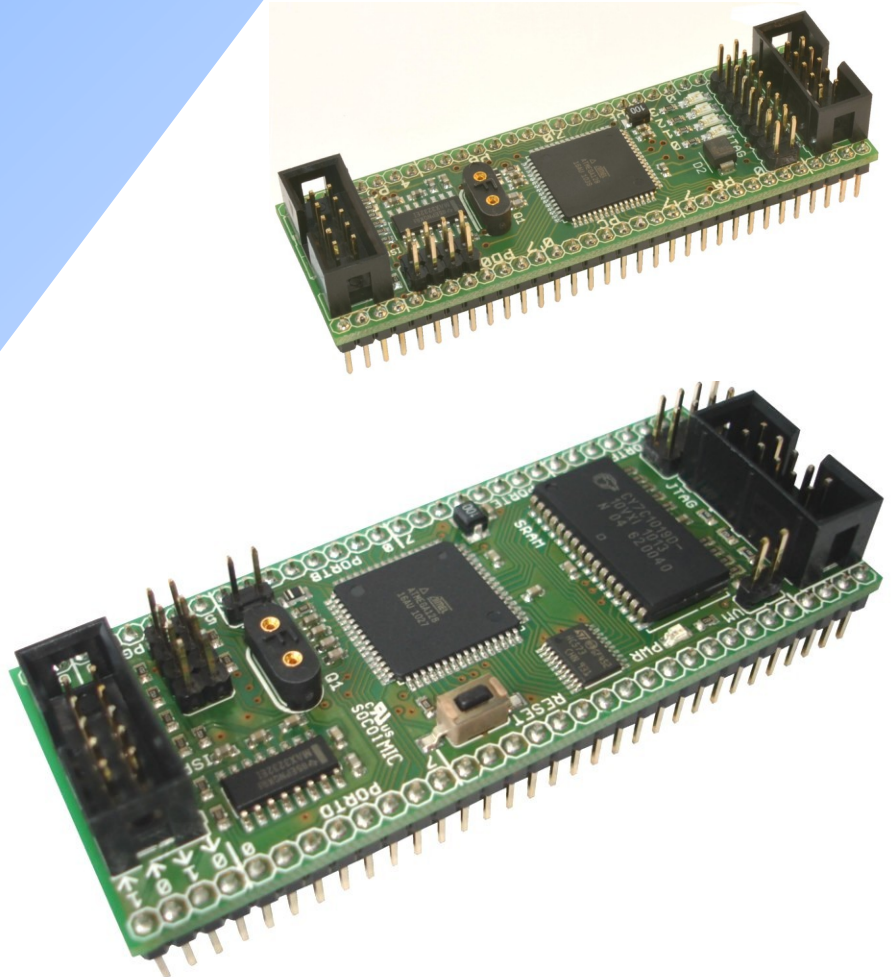


# **BOOTLOADER**

*für AVR-Entwicklungsmodule*

- mit ATmega128
- mit AT90CAN128
- mit ATmega2561



# Verzeichnis

<b>1. Einleitung</b>	<b>3</b>
<b>2. Einstellungen</b>	<b>4</b>
<b>2.1. Fuse Bits</b>	<b>4</b>
<b>2.2. Programmierumgebung</b>	<b>6</b>
2.2.1. Hyper Terminal (HT)	6
2.2.2. chip45boot2 GUI	9
2.2.3. AVR Studio4	9
<b>2.3. Hex-Files</b>	<b>10</b>
<b>2.4. Hardware</b>	<b>11</b>
<b>3. Wiederinstallation</b>	<b>12</b>
<b>4. Programmieren</b>	<b>13</b>
<b>4.1. Zeitmodus</b>	<b>13</b>
4.1.1. ALVIDI_BOOTLOADER	13
4.1.2. CHIP45_BOOTLOADER	16
<b>4.2. Tastenmodus</b>	<b>17</b>
4.2.1. ALVIDI_BOOTLOADER	17
4.2.2. CHIP45_BOOTLOADER	19
4.2.3. FLEURY_BOOTLOADER	20
<b>4.3. Problembehebung</b>	<b>21</b>
<b>5. Links</b>	<b>22</b>
<b>6. Disclaimer</b>	<b>23</b>

# 1. Einleitung

Um AVR-Controller zu programmieren gibt es mehrere Möglichkeiten: Nutzung von ISP-Programmer, JTAG-Debugger oder Bootloader.

ISP-Programmer und JTAG-Debugger sind Hardwarelösungen und in erster Linie für die Entwicklungsphase gedacht, damit Fehlerquellen im Programmcode leichter gefunden werden können. Diese Verfahren erfordern zusätzliche Hardware und einen direkten Zugang zur Programmierschnittstelle am Controller.

Bootloader ist dagegen eine reine Softwarelösung. Bootloader ist eine Software, die dem Beschreiben des Speichers mit bereits erstelltem Programmcode dient. Der Bootloader befindet sich im „*Boot Flash Section*“ und startet bei jedem Hardware- oder Software-Reset, je nach Fuse Bits Einstellung.

In diesem Dokument stellen wir Ihnen 3 Bootloader vor:

ALVIDI\_BOOTLOADER, CHIP45\_BOOTLOADER und

FLEURY\_BOOTLOADER. Sie finden fertig kompilierte Hex-Files für alle oben genannte Bootloader unter folgendem Link:

[http://alvidi.de/data\\_sheets/BOOT\\_FILES.zip](http://alvidi.de/data_sheets/BOOT_FILES.zip)

Diese Bootloader können sowohl im Tasten-Modus (KEY\_MODE) als auch im Zeit-Modus (TIME\_MODE) programmiert werden. FLEURY\_BOOTLOADER ist nur im Tasten-Modus verfügbar.

**KEY\_MODE:** damit Bootloader im Tasten-Modus gestartet werden kann, muss externe Pull-Down am Pin PG4 während Neustart oder Reset anliegen.

**TIME\_MODE:** Bootloader im Zeit-Modus startet jedes mal für 3 Sekunden nach jedem Neustart oder Reset

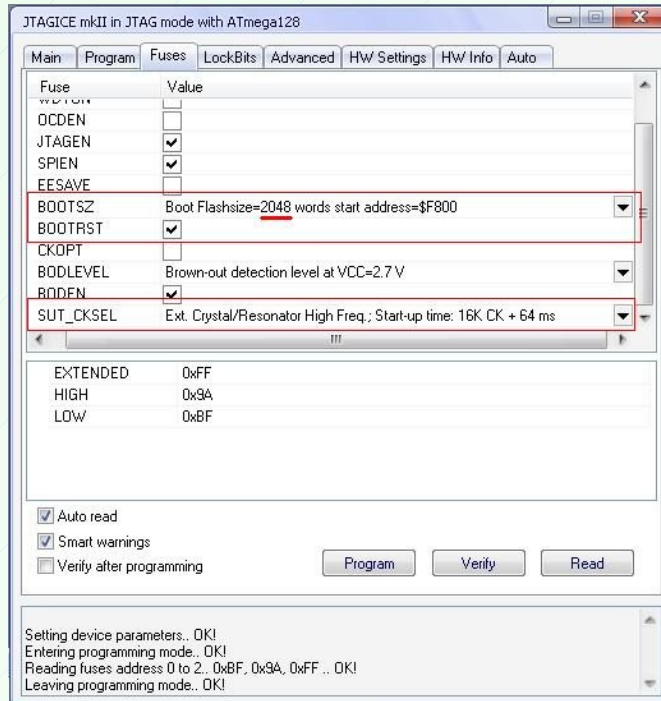
Nachdem Bootloader gestartet ist, liegt am Pin PG3 interne Pull-Down an.

Alle Bootloader benötigen eine serielle Verbindung mit dem Computer. Zusätzlich: ein beliebiges Terminal-Programm, welches VT100 Protokoll unterstützt, z.B. *Hyper Terminal* für ALVIDI\_BOOTLOADER, *chip45boot2 GUI* für CHIP45\_BOOTLOADER und ein Programm, welches stk500 Protokoll unterstützt, z.B. *AVR Studio 4* für FLEURY\_BOOTLOADER.

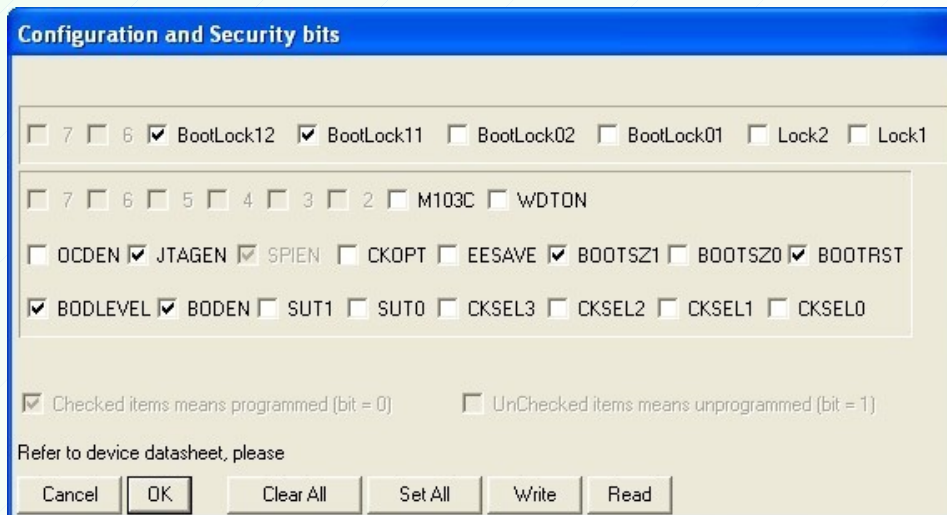
# 2. Einstellungen

## 2.1. Fuse Bits

Einstellungen der Fuse Bits mit AVR Studio 4



Einstellungen der Fuse Bits mit PonyProg



Mit den Fuse Bits BOOTSZ0 und BOOTSZ1 wird die Größe des Bootloaders festgelegt. Die unten stehende Tabelle zeigt die entsprechenden Einstellungen.

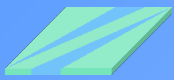
BOOTSZ1	BOOTSZ0	Boot Size	Pages	Application Flash Section	Boot Loader Flash Section	End Application section	Boot Reset Address (start Boot Loader Section)
1	1	512 words	4	\$0000 - \$FDFF	\$FE00 - \$FFFF	\$FDFF	\$FE00
1	0	1024 words	8	\$0000 - \$FBFF	\$FC00 - \$FFFF	\$FBFF	\$FC00
0	1	2048 words	16	\$0000 - \$F7FF	\$F800 - \$FFFF	\$F7FF	\$F800
0	0	4096 words	32	\$0000 - \$EFFF	\$F000 - \$FFFF	\$EFFF	\$F000

Tabelle aus dem Datenblatt Atmel AVR ATmega128.pdf Seite 284

Die Größe des Bootloadersprogramms beträgt 4000 Byte, dies entspricht 2000 Words (1 Byte=8 Bit und 1 Word=16 Bit).

Wenn das Fuse Bit BOOTRST gesetzt wird, springt der Vektor nach dem Reset auf die Adresse des Bootloaders. In unserem Fall springt der Vektor auf die Adresse \$F800.

Damit der Bootloader nicht überschrieben wird, wird der Bootloader durch das Setzen von BootLock11 und BootLock12 (PonyProg) oder „BLB1 LPM and SPM prohibited in Boot Section“ (AVR Studio 4) geschützt.

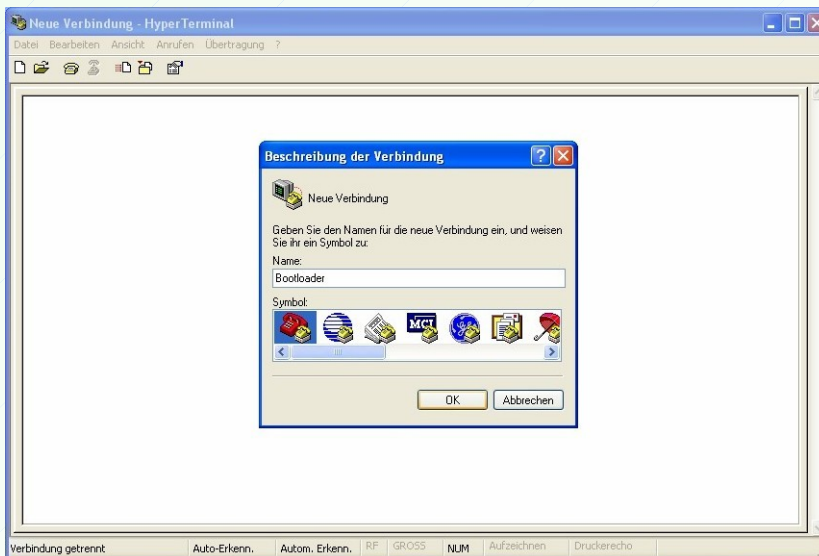


## 2.2. Programmierumgebung

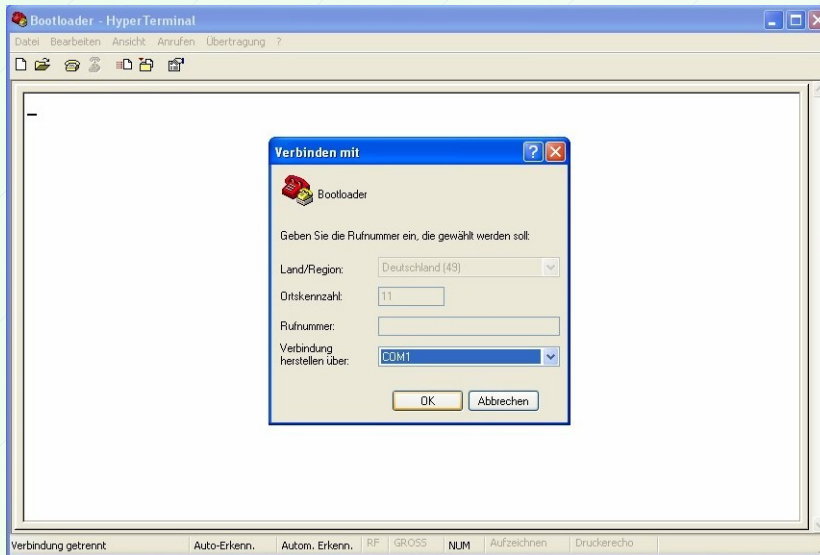
### 2.2.1. Hyper Terminal (ALVIDI\_BOOTLOADER)



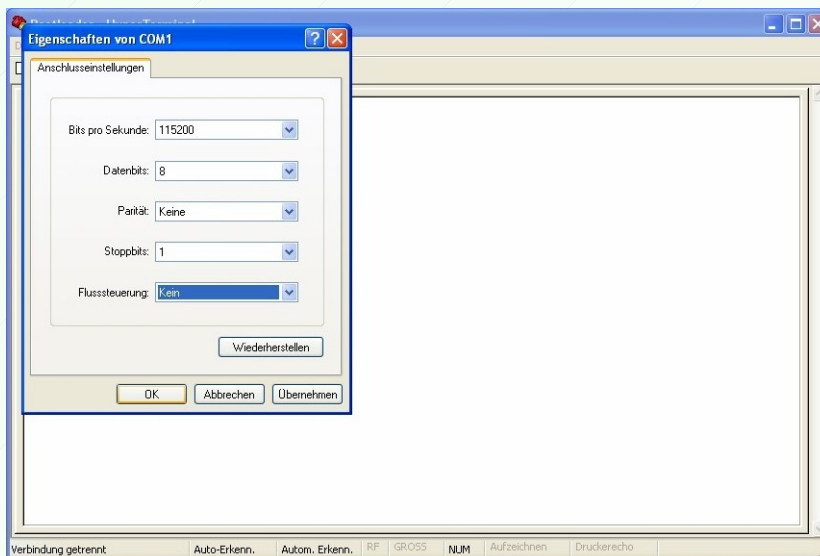
Starten Sie Hyper Terminal *Start* → *Programme* → *Zubehör* → *Kommunikation* → *Hyper Terminal*



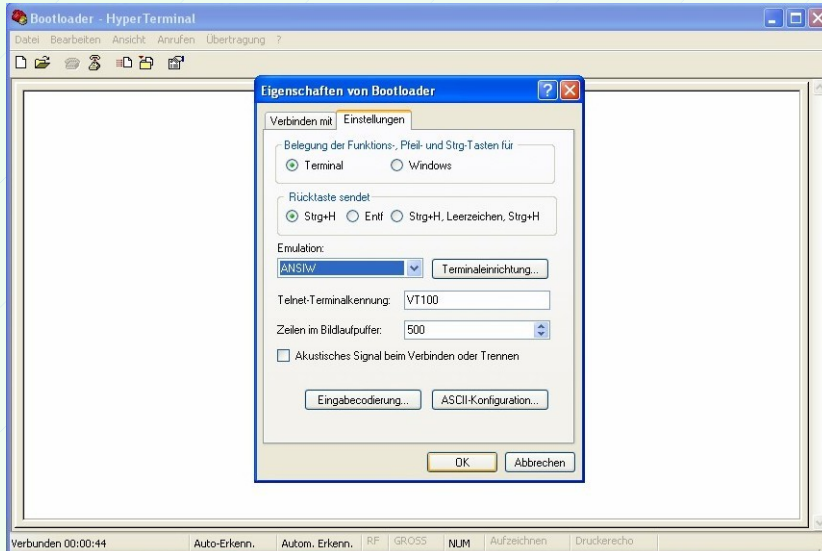
Geben Sie einen Namen, z.B. Bootloader, und klicken Sie auf die Schaltfläche „OK“.



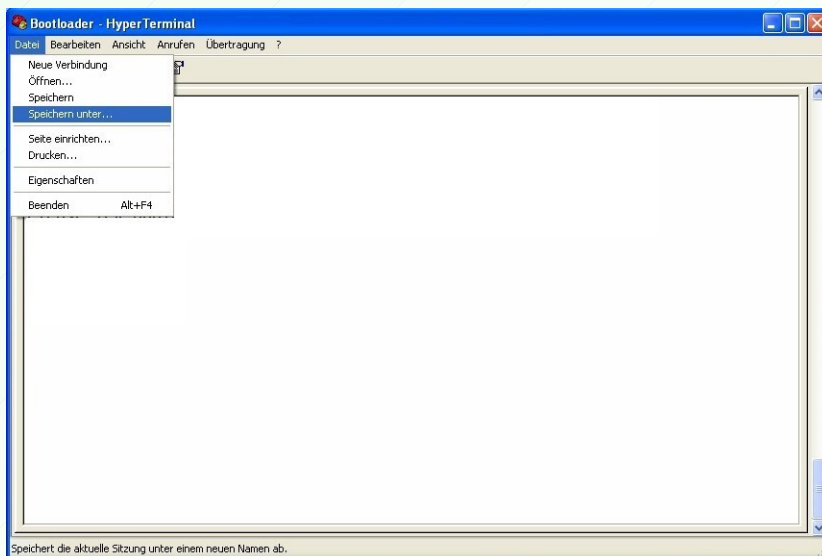
Wählen Sie die Schnittstelle, z.B. COM1, und klicken Sie auf die Schaltfläche „OK“.



Übernehmen Sie die oben abgebildeten Parameter (die „Bits pro Sekunde“ entnehmen Sie aus dem Hex File s.Kapitel 2.3. Hex-Files), und klicken Sie auf die Schaltfläche „OK“.



Im Fenster *Eigenschaften Datei* → *Eigenschaften* → *Einstellungen* wählen Sie im Feld *Emulation* → *ANSIW* oder *VT100*, und klicken Sie auf die Schaltfläche „OK“.

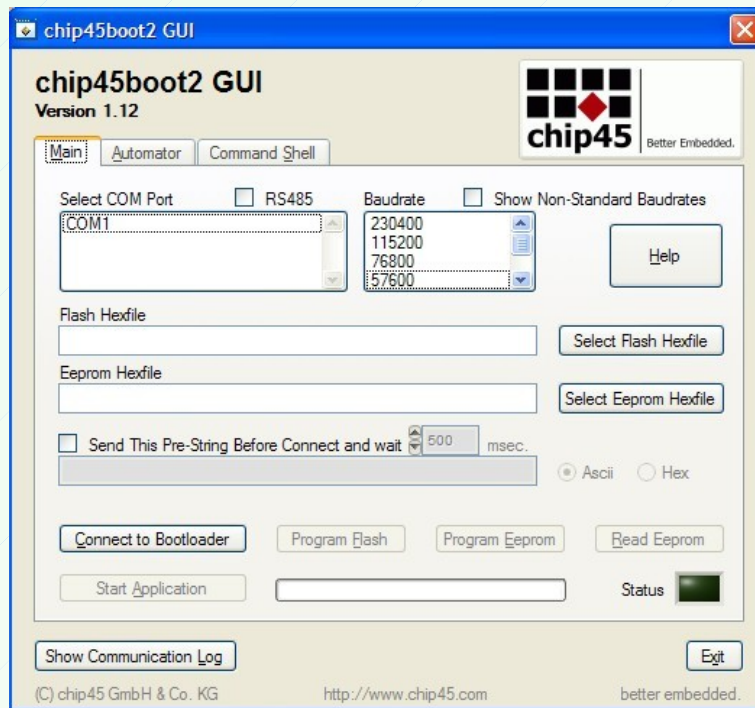


Speichern Sie die Einstellungen, z.B. auf dem Desktop unter *Bootloader.ht*.

## 2.2.2. chip45boot2 GUI (CHIP45\_BOOTLOADER)

Diese Software können Sie auf der Webseite [www.chip45.com](http://www.chip45.com) unter folgendem Link herunterladen:

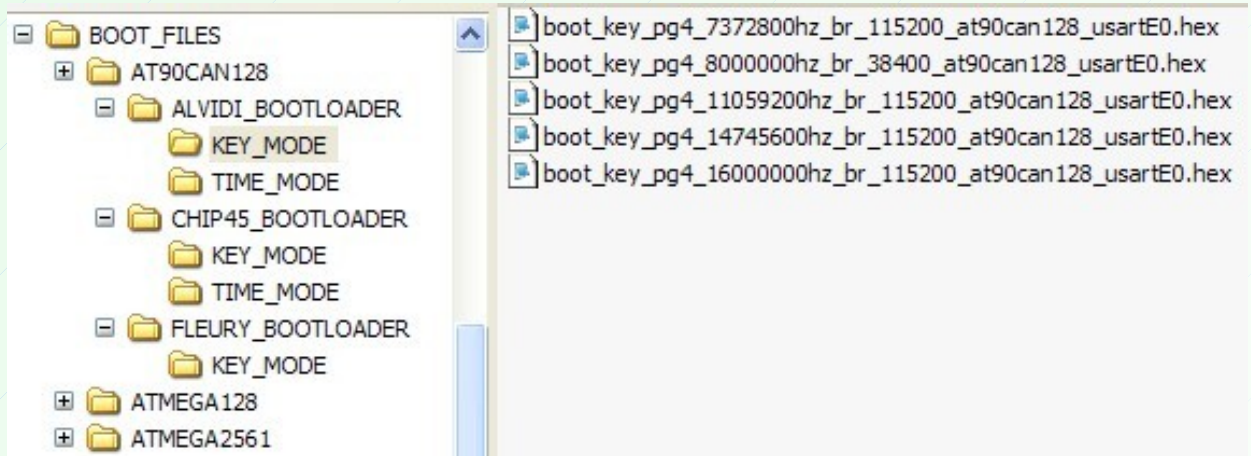
[http://download.chip45.com/chip45boot2\\_GUI\\_V1.12.zip](http://download.chip45.com/chip45boot2_GUI_V1.12.zip)



## 2.2.3. AVR Studio 4 (FLEURY\_BOOTLOADER)

Atmel bietet mit seinen Produkten kostenlose Entwicklungsumgebung AVR Studio 4 an. Sie können diese Software direkt auf Atmel-Webseite herunterladen: [http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)

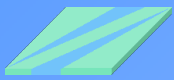
## 2.3. Hex-Files



Die bereits vorkompilierte Hex-Files für alle 3 Bootloader liegen in [BOOT\\_FILES.zip](#) Archiv, das von unserer Website heruntergeladen werden kann. Die Boot-Files liegen, wie in der obigen Abbildung dargestellt, sortiert nach dem Controller, Herkunft und der Programmierart (KEY\_MODE und TIME\_MODE) in entsprechenden Ordnern. Die Boot-Files sind für alle von uns angebotenen ATMEGA und AT90CAN128 Module in KEY\_MODE und TIME\_MODE für alle Quarzfrequenzen (16MHz, 14.7456MHz, 11.0592MHz, 8MHz, 7.3728MHz) vorhanden.

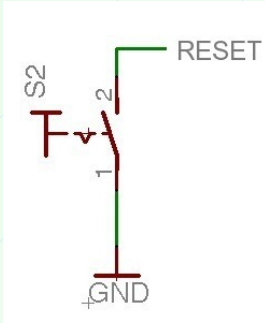
Die Bezeichnung jedes einzelnen Hex-Files enthält das Programmiermode, die Betriebsfrequenz, die Bitrate, den Controller und USART-Port.

Beispiel: `boot_key_pg4_16000000hz_br_115200_at90can128_usartE0.hex` dieses File wurde für das AVR Modul mit AT90CAN128 Controller geschrieben, das mit 16 MHz externem Quarz ausgerüstet ist. Die Übertragungsrate liegt bei 115200 bit/s am USART-Port E. Der Bootloader wird durch ein Reset und Pull-Down am Pin PG4 aktiviert.

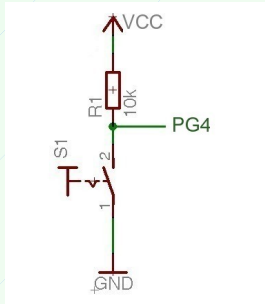


## 2.4. Hardware

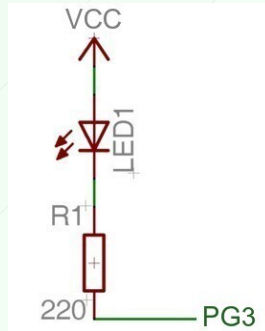
Zum Programmieren mit dem Bootloader benötigen Sie externe Beschaltung:



Beschalten Sie den Ausgang **Reset** (falls keinen Taster auf dem Modul vorhanden ist), mit z.B. einem externen Taster, so wie es in der Abbildung links dargestellt ist.

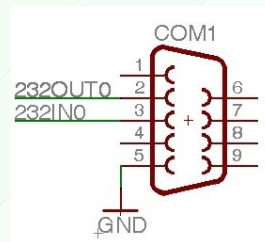


Wenn Sie den Bootloader im **Tastenmodus** benutzen wollen, müssen Sie wie im linken Abbild, PortG Pin4 (kurz **PG4**), Ihre Schaltung erweitern.



Zur Kontrolle können Sie am PortG Pin3 (kurz **PG3**) 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 (**232out0**) des AVR-Moduls, an den Pin3 der D-SUB Buchse den seriellen Eingang 0 (**232in0**) 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 das komplette Speicher, inklusive der Bootloader, gelöscht. Die Fuse Bits **BootLock11** und **Bootlock12** werden dabei zurückgesetzt.

### Wiederinstallation des Bootloaders mit AVR Studio

Vor der Installation bitte vergewissern Sie sich, dass AVR-Studio auf dem aktuellen Stand ist (mindestens ver. 4.19). Zur Wiederinstallation können Sie schon fertige Hex-File (siehe 2.3 Hex-Files) benutzen.

Nach dem der gewünschte Bootloader in den Speicher geschrieben ist, müssen die Fuse-Bits richtig eingestellt werden. Dazu führen Sie folgende Einstellungen aus: unter *Tools* → *Program AVR* → *Connect...* → *Connect...* → *Fuses* wählen Sie „**Boot Flash section size=2048 ...**“ und „**Boot Reset vector Enabled ...**“. Damit der Bootloader nicht überschrieben wird, wählen Sie unter *Tools* → *Program AVR* → *Connect...* → *Connect...* → *Lock Bits* „**BLB1 LPM and SPM prohibited in Boot Section**“

### Wiederinstallation des Bootloaders mit PonyProg

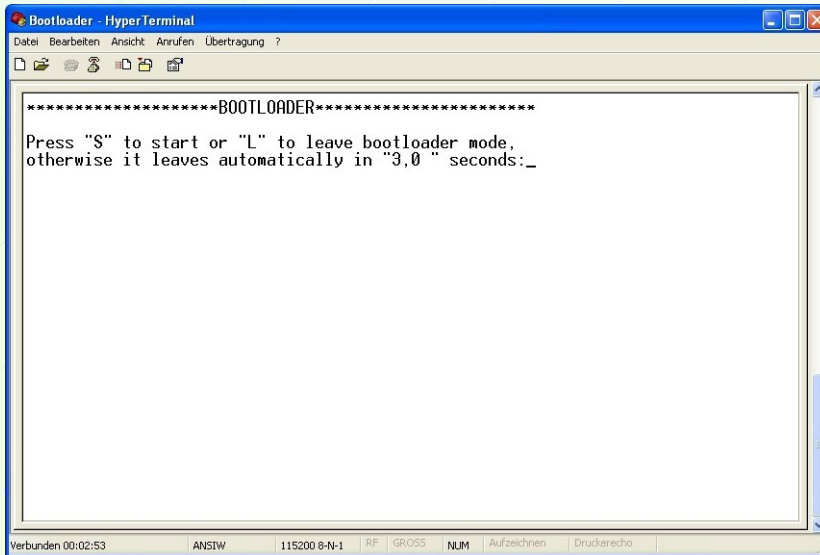
Programmieren Sie das passende Hex-File und setzen Sie die folgende Fuse Bits: **BootLock11**, **Bootlock12**, **BOOTSZ1** und **BOOTRST**.

# 4. Programmieren

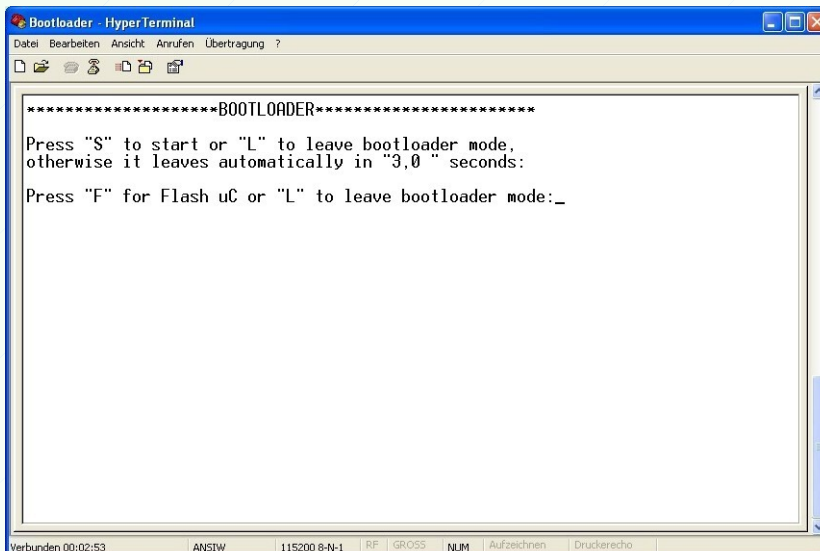
## 4.1. Zeitmodus

### 4.1.1. ALVIDI\_BOOTLOADER

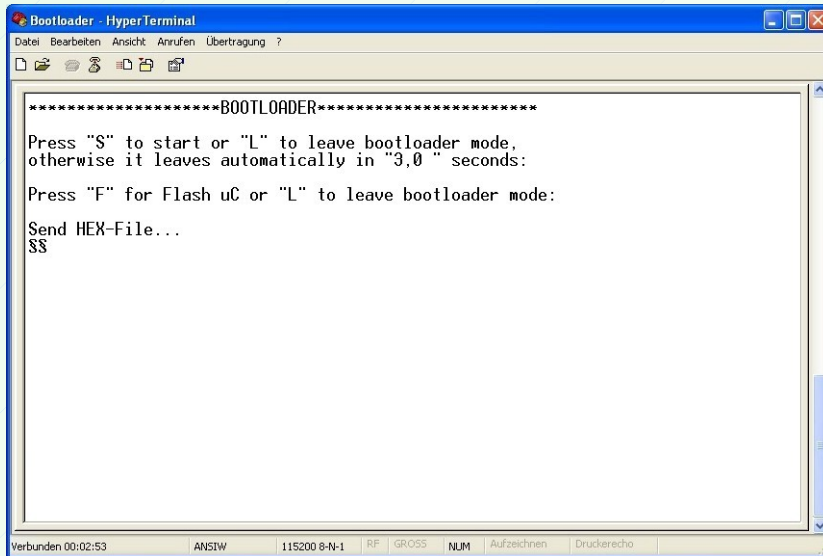
Starten Sie die gespeicherte Einstellungen des Hyper Terminals, z.B. Bootloader.ht (siehe Kapitel 2.2. Hyper Terminal)



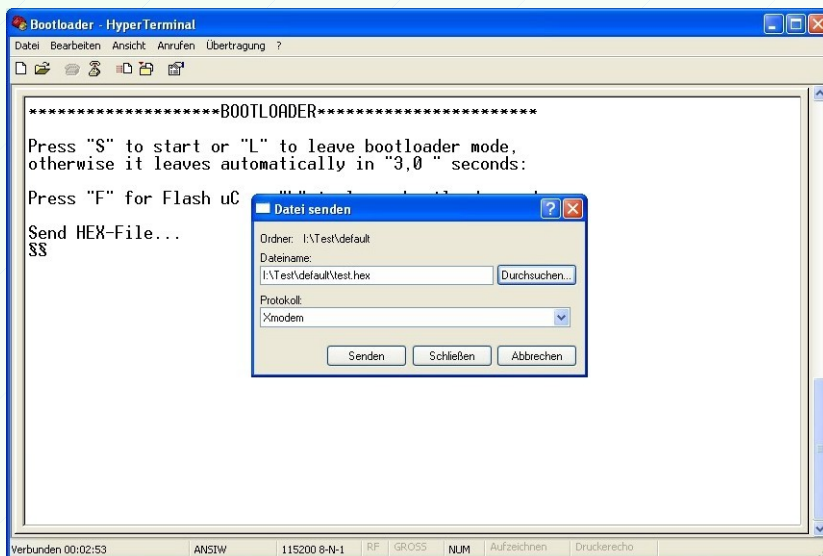
Drücken Sie die Reset-Taste, um oberes Abbild im Fenster des Hyper Terminals zu bekommen. Der Bootloader ist gestartet. Das Drücken der „S“-Taste führt Sie in das Bootloader-Menü. Zum Verlassen drücken Sie „L“-Taste. Falls kein „S“ Zeichen innerhalb von 3 Sekunden empfangen wurde, startet der Applikationsbereich automatisch.



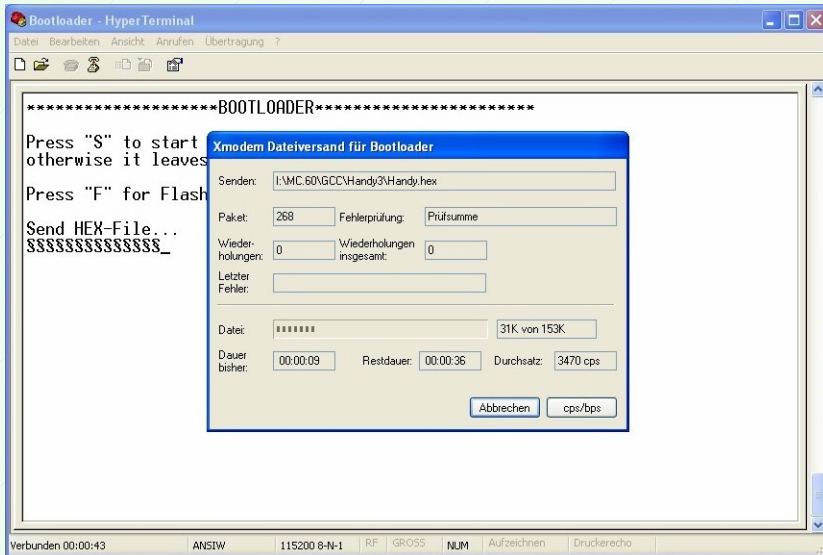
Im Bootloader-Menü drücken Sie die „F“-Taste zum Programmieren des Mikrocontrollers oder die „L“-Taste zum Verlassen des Bootloaders.



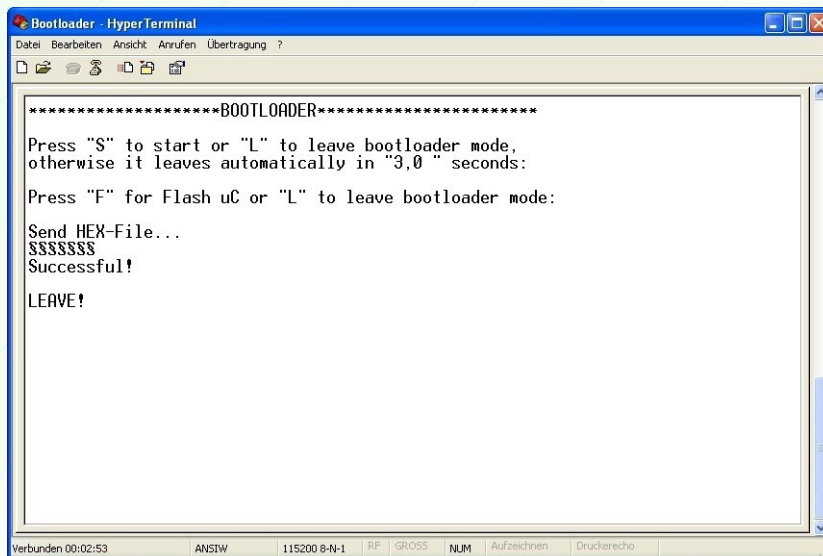
Nachdem die „F“-Taste gedrückt wurde, erscheint „Send Hex-File...“. Sie haben jetzt 60 Sekunden die Hex-Daten an den Controller zu senden, sonst startet nach Ablauf dieser Zeit der Applikationsbereich.



Gehen Sie auf **Übertragung** → **Datei senden...** und wählen Sie im Feld **Dateiname:** zu programmierendes Hex-File. Im Feld **Protokoll:** wählen Sie **Xmodem** und klicken Sie anschließend auf die Schaltfläche „**Senden**“

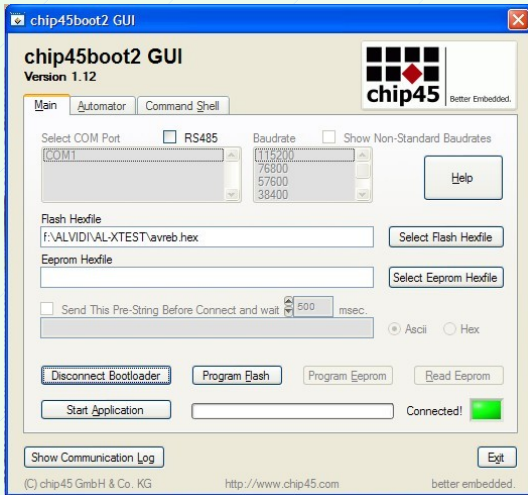


Während der Datenübertragung an den Mikrocontroller erscheint im Fenster „*Xmodem Dateiversand für Bootloader*“ die Einzelheiten der Übertragung, siehe obere Abbildung.

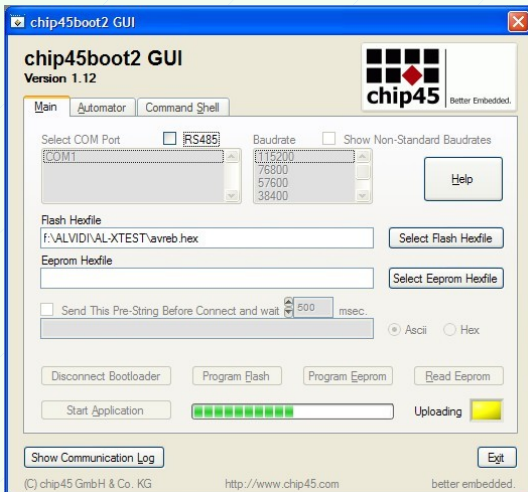


Wenn die Datenübertragung erfolgreich war, erscheint „**Successful**“ und der Bootloader wird mit dem Hinweis „**LEAVE!**“ verlassen.

## 4.1.2. CHIP45\_BOOTLOADER



1. Starten Sie die heruntergeladene Software chip45boot2 GUI
2. Wählen Sie im linken Fenster „Select COM Port“ mit dem Modul angeschlossene Serielle Schnittstelle aus.
3. Im rechten Fenster „Baudrate“ können Sie die Bitrate bestimmen. Empfohlen wird am Anfang mit niedriger Bitrate (19200) zu beginnen.
4. Drücken Sie die interne (falls vorhanden) oder externe Reset-Taste auf dem AVR-Modul
5. Wenn Sie innerhalb von 3 Sekunden Button „Connect to Bootloader“ gedrückt haben, erscheint das linke Bild.

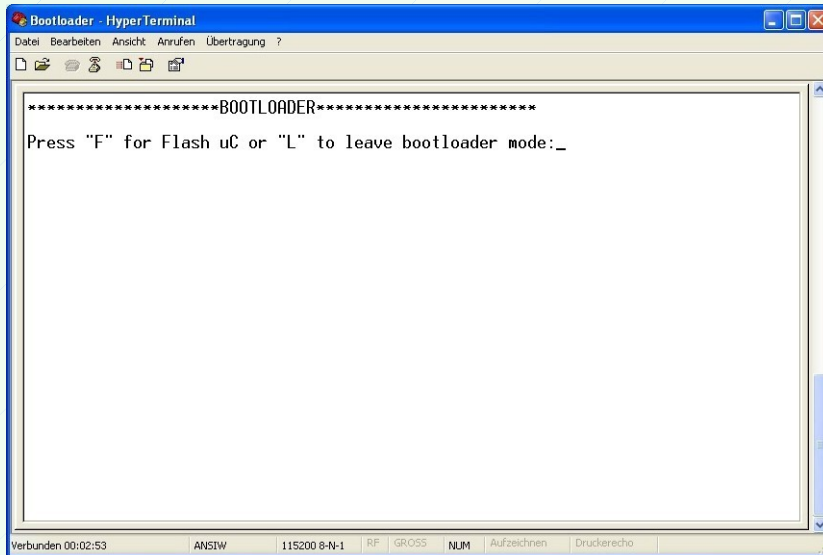


6. Klicken Sie auf Schaltfläche „Select Flash Hexfile“ und wählen Sie zu programmierende Hex-File
7. Zum Programmieren bzw. zum Beschreiben des Flash klicken Sie auf den Button „Program Flash“, es erscheint das linke Abbild
8. Sobald das Beschreiben des Controllers abgeschlossen ist, starten Sie Ihr Programm indem Sie die Schaltfläche „Start Application“ betätigen.

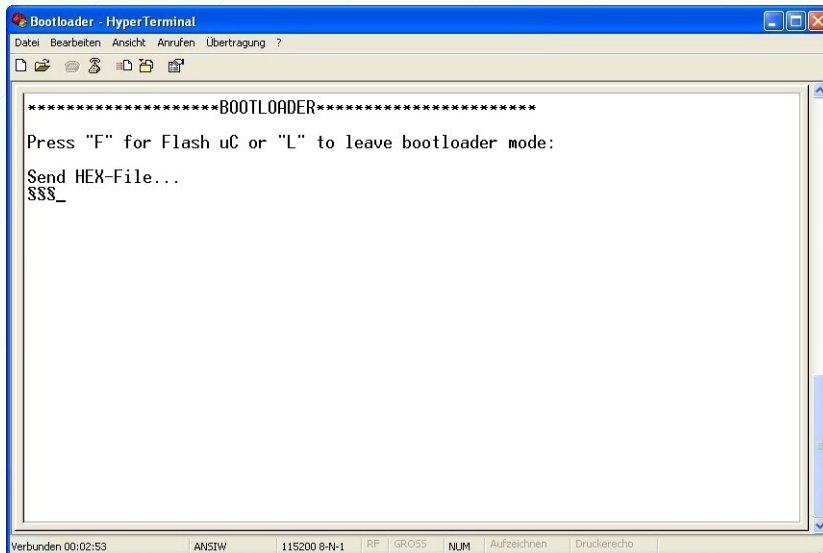
## 4.2. Tastenmodus

### 4.2.1. ALVIDI\_BOOTLOADER

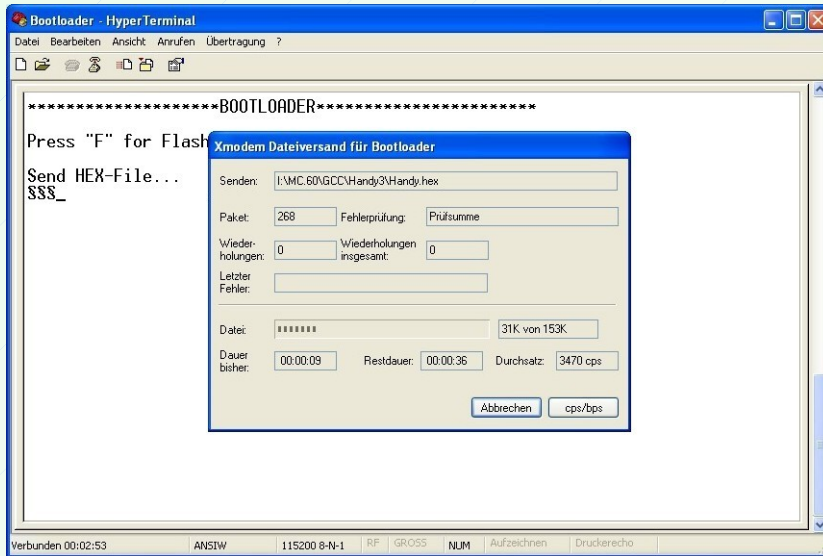
Starten Sie die gespeicherte Einstellungen des Hyper Terminals, z.B. Bootloader.ht (siehe Kapitel 2.2. Hyper Terminal)



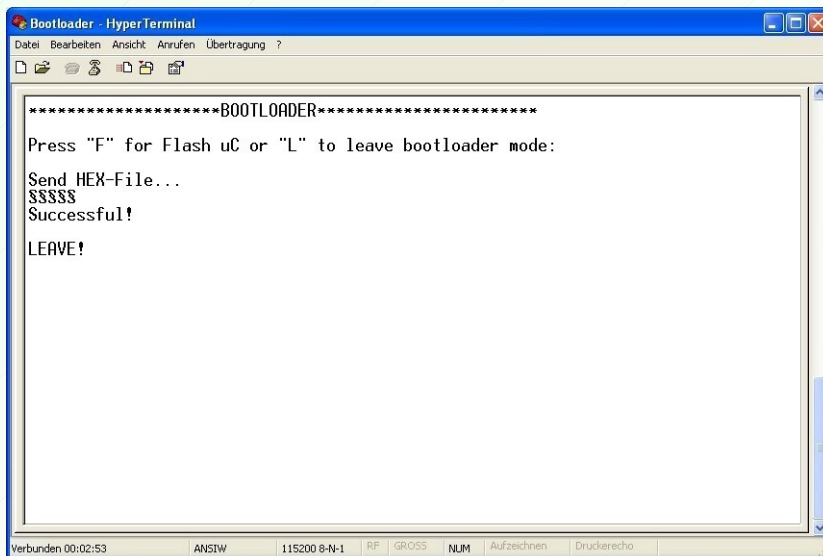
Drücken Sie die Pin-Taste (**PG4**) und die Reset-Taste zusammen, um oberes Abbild im Fenster des Hyper Terminals zu bekommen. Sie sind im Bootloader-Menü. Dort drücken Sie die „F“-Taste zum Programmieren des Mikrocontrollers oder die „L“-Taste zum Verlassen des Bootloaders.



Nachdem die „F“-Taste gedrückt wurde, erscheint „Send Hex-File...“. Sie haben jetzt 60 Sekunden die Hex-Daten an den Controller zu senden, sonst startet nach Ablauf dieser Zeit der Applikationsbereich.

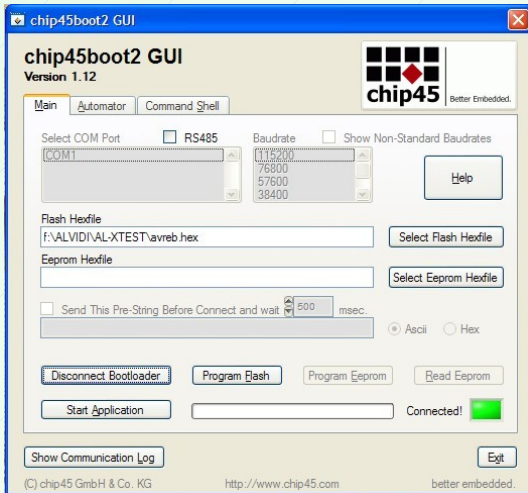


Während der Datenübertragung an den Mikrocontroller erscheinen im Fenster „*Xmodem Dateiversand für Bootloader*“ die Einzelheiten der Übertragung, siehe obere Abbildung.

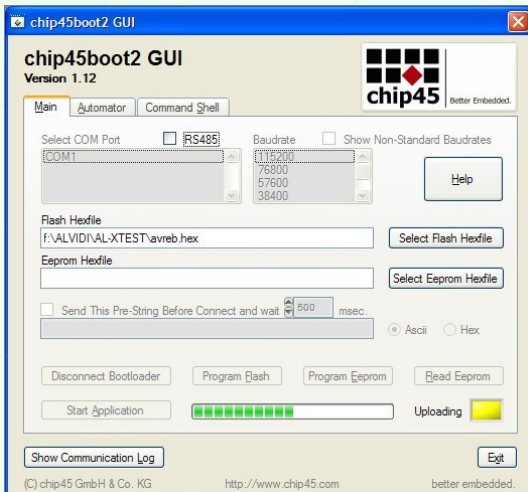


Wenn die Datenübertragung erfolgreich war, erscheint „**Successfull**“ und der Bootloader wird mit dem Hinweis „**LEAVE!**“ verlassen.

## 4.2.2. CHIP45\_BOOTLOADER

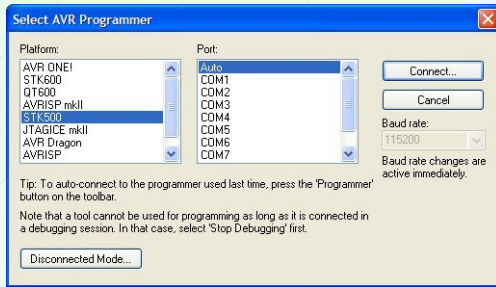


1. Starten Sie die heruntergeladene Software chip45boot2 GUI
2. Wählen Sie im linken Fenster „Select COM Port“ mit dem Modul angeschlossene Serielle Schnittstelle
3. Im rechten Fenster „Baudrate“ können Sie die Bitrate bestimmen. Empfohlen wird am Anfang mit niedriger Bitrate (19200) zu beginnen.
4. Drücken Sie die interne (falls vorhanden) oder externe Reset-Taste auf dem AVR-Modul mit externer Pin-Taste (**PG4**) gleichzeitig. Danach lassen Sie die Reset-Taste als Erste los.
5. Drücken Sie Button „Connect to Bootloader“, es erscheint das linke Bild.



6. Klicken Sie auf Schaltfläche „Select Flash Hexfile“ und wählen Sie zu programmierende Hex-File
7. Zum Programmieren bzw. zum Beschreiben des Flash klicken Sie auf den Button „Program Flash“, es erscheint das linke Abbild
8. Sobald das Beschreiben des Controllers abgeschlossen ist, starten Sie Ihr Programm indem Sie die Schaltfläche „Start Application“ betätigen.

## 4.2.3. FLEURY\_BOOTLOADER



1. Drücken Sie die interne (falls vorhanden) oder externe Reset-Taste auf dem AVR-Modul mit externer Pin-Taste (PG4) gleichzeitig. Danach lassen Sie die Reset-Taste als Erste los.

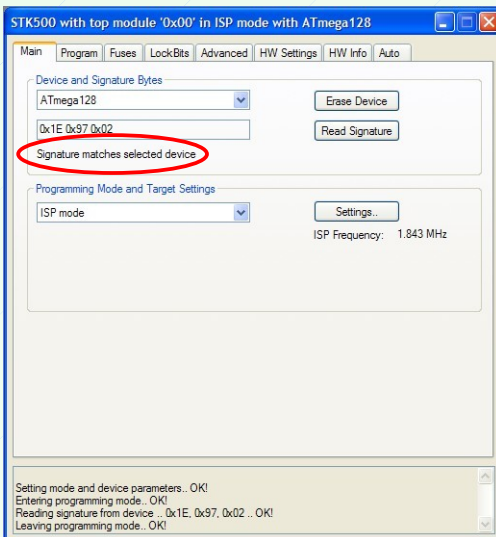
2. Starten Sie AVR Studio 4.

3. Klicken Sie in der Taskleiste **Tools** → **Program AVR** → **Connect...**

4. Wählen Sie im linken Fenster „Platform“ **stk500** und betätigen Sie die Schaltfläche **Connect...**

5. Im Unterfenster „Main > Device and Signature Bytes“ wählen Sie zutreffende Controller aus.

6. Zur Kontrolle klicken Sie auf die Schaltfläche „Read Signature“, wenn Sie den richtigen Controller ausgesucht haben, erscheint „**Signature matches selected device**“, siehe linkes Abbild.

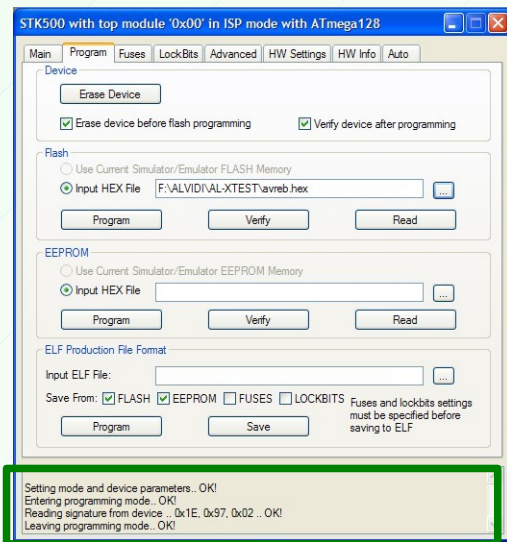


7. Im Unterfenster „Program“ klicken Sie den Button „...“ gegenüber Flash-Eingabe-Link-Fenster und fügen Sie das zu programmierende Hex-File ein.

8. Mit dem Klick auf die Schaltfläche „Program“ unterhalb Flash-Eingabe-Link-Fenster beschreiben Sie den Controller.

9. Im unterem Fenster können Sie den gesamten Ablauf beobachten (mit grüner Linie gekennzeichnet).

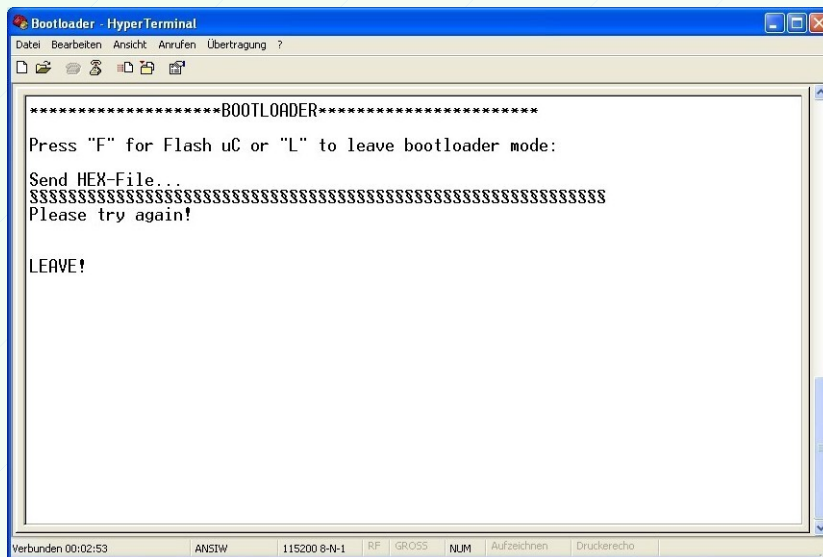
10. Sobald die Programmierung abgeschlossen ist, drücken Sie die Reset-Taste.



## 4.3. Problembhebung

Kein System ist perfekt, daher könnten auch hier Probleme stattfinden. Damit Sie schnell Fehler beheben können, stellen wir Ihnen oft auftretende Missgeschicke dar.

1. Vergewissern Sie sich, dass die Jumper des AVR-Entwicklungsmoduls (alle AL-AVREB Module JP2-1 und JP2-3) (alle AL-ERAM128 Module JP5-1 und JP5-3) gesetzt sind.
2. Stellen Sie fest, dass an den folgenden Pins keine Last angeschlossen ist: PE0 und PE1.
3. Prüfen Sie, ob die Hardware richtig angeschlossen ist (siehe Kapitel **2.4. Hardware**).
4. Im Falle der Softwarefehler führen Sie die Wiederinstallation durch (siehe Kapitel **3. Wiederinstallation**).



5. Falls die Hex-Daten innerhalb von 60 Sekunden nicht an den Mikrocontroller gesendet wurden, erscheint die obere Abbildung.
6. Falls es während der Datenübertragung zum Abbruch des Datentransfers kommt, überprüfen Sie **Punkt 2 (siehe oben)**
7. Sollte Bootloader-Menü nach der Wiederinstallation im Hyper-Terminal nicht sichtbar sein, könnte die Installation der neuesten Version der AVR-Studio Software erforderlich sein.

## 5. Links

- Bootloader Hex-Files BOOT\_FILES.zip (0,18 MB September 2011)  
[http://alvidi.de/data\\_sheets/BOOT\\_FILES.zip](http://alvidi.de/data_sheets/BOOT_FILES.zip)
- AVR Studio 4.19 (124 MB, revision build 730, updated 9/11)  
[http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)
- AVR Toolchain 3.3.0 Installer (94 MB, revision 3.3.0.710, updated 9/11)  
[http://www.atmel.com/dyn/products/tools\\_card.asp?tool\\_id=2725](http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725)
- Frei zu verwendender chip45boot2 Bootloader für AVR ATmega und Xmega  $\mu$ C  
<http://shop.chip45.com/AVR-Mikrocontroller-Software/AVR-ATmega-Xmega...>
- chip45boot2 GUI PC/Windows Programm (9 MB, Version V1.12 )  
[http://download.chip45.com/chip45boot2\\_GUI\\_V1.12.zip](http://download.chip45.com/chip45boot2_GUI_V1.12.zip)
- Home page Peter Fleury  
<http://homepage.hispeed.ch/peterfleury/index.html>
- AVR Studio compatible Boot Loader (20kB, Version V1.15 Mai 2008)  
<http://homepage.hispeed.ch/peterfleury/stk500v2bootloader.zip>

## **6. Disclaimer**

Wir haften nicht für die Eignung der Ware für einen bestimmten, vom Kunden beabsichtigten Verwendungszweck.

Wir übernehmen keine Haftung für die Fehler, die im Laufe der Nutzung auftreten können.

Wir übernehmen keine Verantwortung für mögliche Schäden, die bei der Nutzung des Programms entstehen können.

Nutzung der Software ist nur im Zusammenhang mit von uns vertriebenen Hardware uneingeschränkt gestattet. Nutzung der Software auf fremder Hardware ist ohne schriftliche Genehmigung untersagt.

Alle Rechte sind vorbehalten