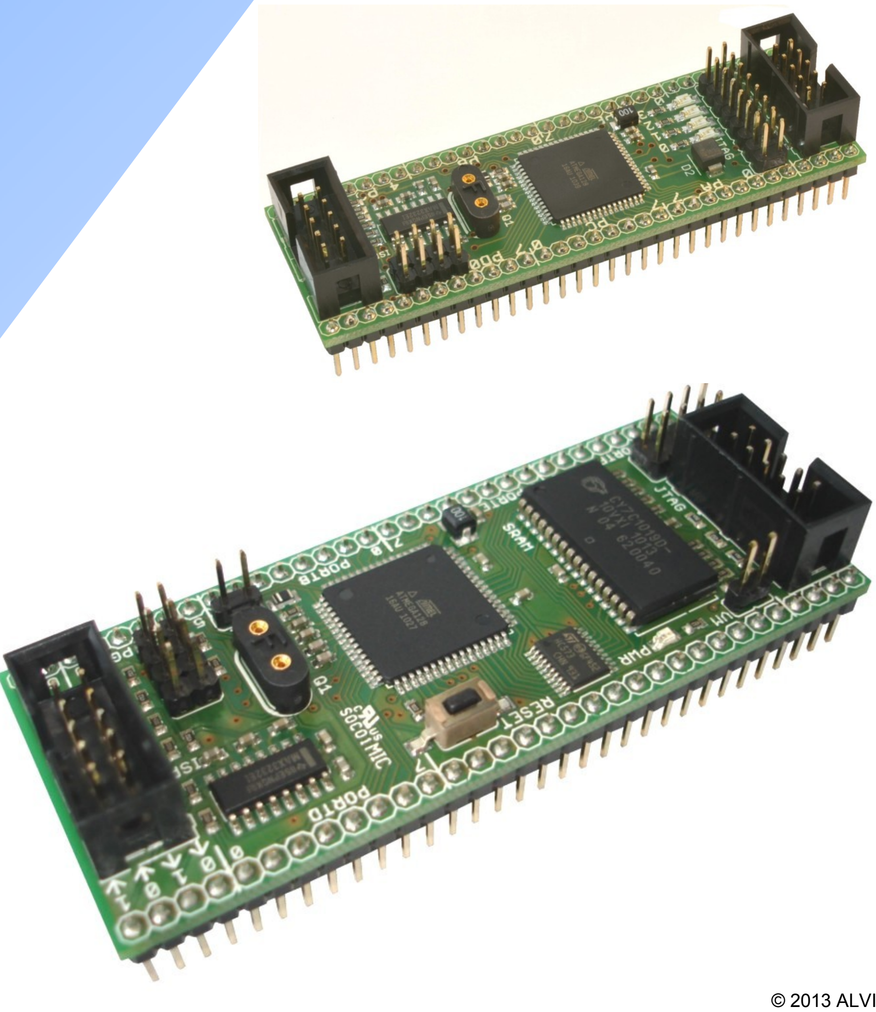
ALVIDI

BOOTLOADER

for AVR-Development Modules

- with ATmega128
- with AT90CAN128
- with ATmega2561



Directory

1. Instructions

There are several ways to program an AVR-Controller: using the ISP-Programmer, JTAG-Debugger or a boot loader.

ISP-Programmer and JTAG-Debugger are hardware solutions and are designed primarily to find sources of errors in a program code. These methods require additional hardware and a direct access to program interface on the controller.

The boot loader is a software solution. The boot loader is a program, that loads software to the memory flash. The boot loader is located in the memory of the "Boot Flash Section" and will start the boot program during the hardware or software reset, depending on the settings of Fuse Bits.

In this document we introduce your 3 boot loader: ALVIDI_BOOTLOADER, CHIP45_BOOTLOADER und FLEURY_BOOTLOADER. You will find ready compiled Hex-Files for mentioned above under following link:
http://alvidi.de/data_sheets/BOOT_FILES.zip

These boot loader could be programmed as well in key-mode (KEY_MODE) as in time-mode (TIME_MODE). FLEURY_ BOOTLOADER is only in KEY_MODE available.

KEY_MODE: in order to start boot loader in key-mode, you have to occur external Pull-Down on pin PG4 at new start or reset

TIME_MODE: boot loader in time-mode start every time for 3 seconds after new start or reset

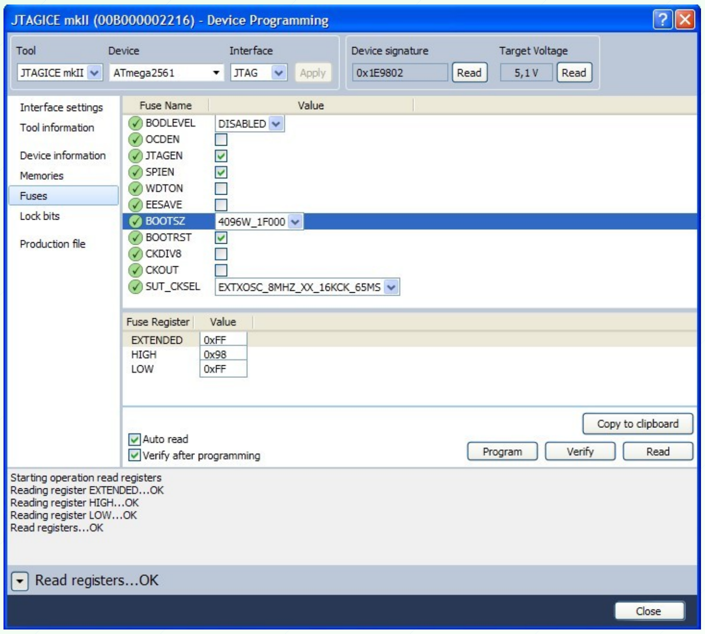
As soon boot loader started, pin PG3 get internal Pull-Down. By connection of one LED to this pin, you may observe the activity of the bootloader.

The boot loader for AVR-Development Module needs a serial connection with computer. Additionally terminal program, which supports VT100 protocol, e.g. Hyper Terminal for ALVIDI_BOOTLOADER, ***chip45boot2 GUI*** for CHIP45_BOOTLOADER and a program, which support stk500 protocol, e.g. ***AVR Studio 4*** for FLEURY_BOOTLOADER.

2. Settings

2.1. Fuse Bits

Settings of the Fuse Bits with Atmel Studio 6



JTAGICE mkII (00B000002216) - Device Programming

Tool	Device	Interface	Device signature	Target Voltage
JTAGICE mkII	ATmega2561	JTAG	0x1E9802	5,1 V

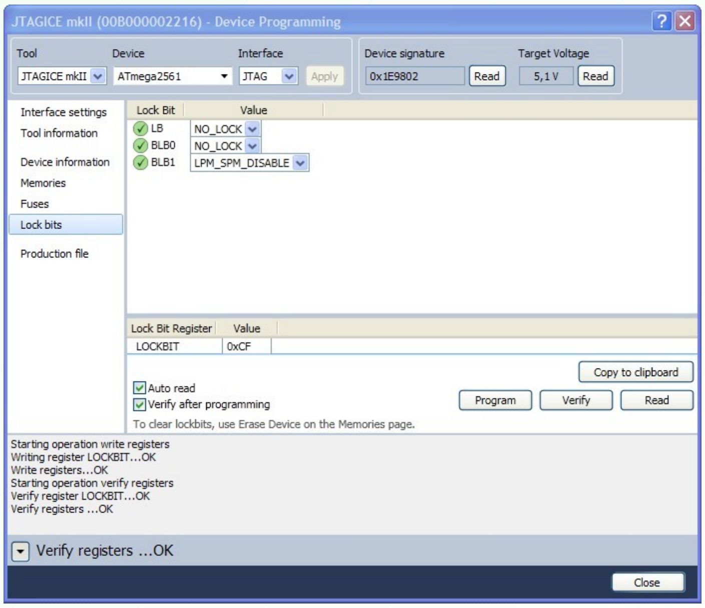
Fuse Name	Value
BODLEVEL	DISABLED
OCDEN	☐
JTAGEN	✓
SPIEN	✓
WDTON	☐
EESAVE	☐
BOOTSZ	4096W_1F000
BOOTRST	✓
CKDIV8	☐
CKOUT	☐
SUT_CKSEL	EXTXOSC_8MHZ_XX_16KCK_65MS

Fuse Register	Value
EXTENDED	0xFF
HIGH	0x98
LOW	0xFF

✓ Auto read
✓ Verify after programming

```
Starting operation read registers
Reading register EXTENDED...OK
Reading register HIGH...OK
Reading register LOW...OK
Read registers...OK
```

Read registers...OK



JTAGICE mkII (00B000002216) - Device Programming

Tool	Device	Interface	Device signature	Target Voltage
JTAGICE mkII	ATmega2561	JTAG	0x1E9802	5,1 V

Lock Bit	Value
LB	NO_LOCK
BLB0	NO_LOCK
BLB1	LPM_SPM_DISABLE

Lock Bit Register	Value
LOCKBIT	0xCF

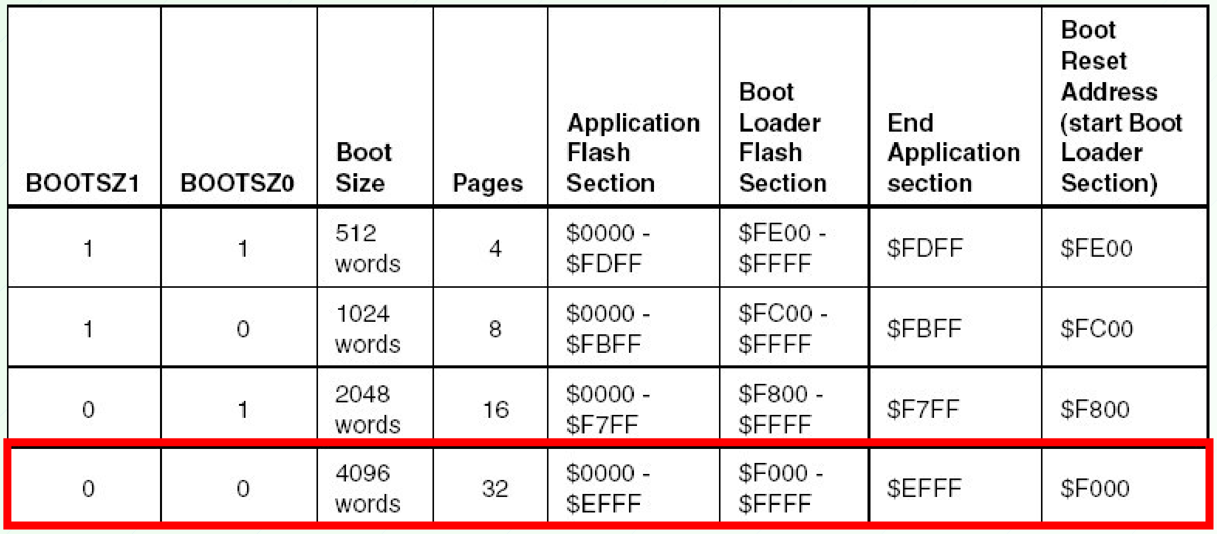
✓ Auto read
✓ Verify after programming

To clear lockbits, use Erase Device on the Memories page.

```
Starting operation write registers
Writing register LOCKBIT...OK
Write registers...OK
Starting operation verify registers
Verify register LOCKBIT...OK
Verify registers ...OK
```

Verify registers ...OK

The size of boot program can be set with the Fuse Bits BOOTSZ0 and BOOTSZ1 . The table below displays the corresponding settings.



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

The table is from the data sheet Atmel AVR ATmega128.pdf site 284

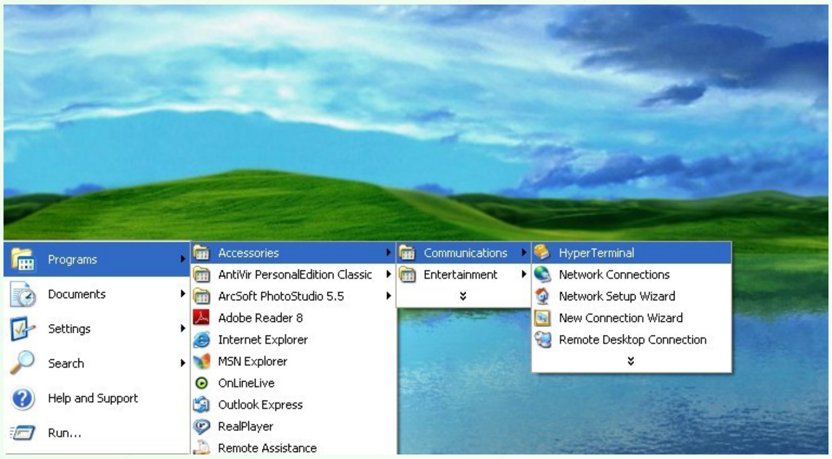
The size of the boot program is 8000 bytes, which is equal to 4000 words (1 byte=8 bit and 1 word=16 bit).

If the Fuse Bit BOOTRST is programmed, the vector will jump to the address of the boot program after reset . In this case the vector will jump to the address $F000.

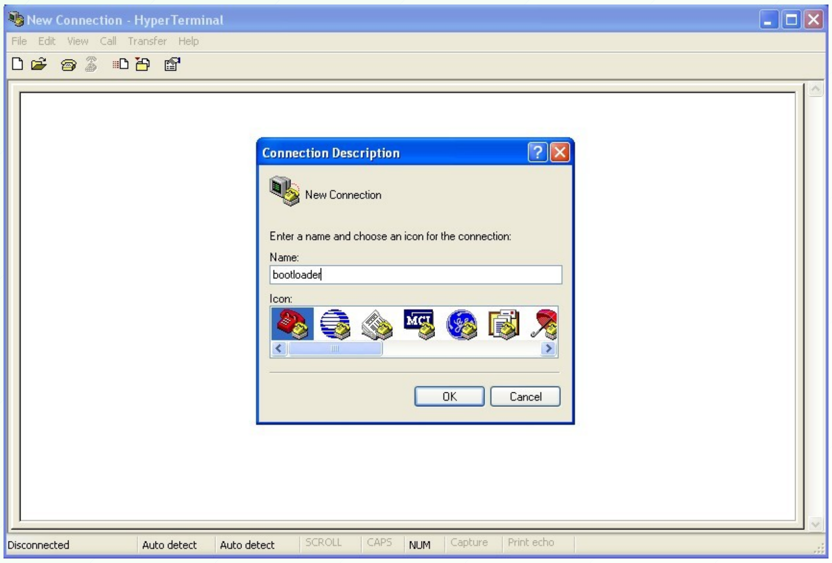
In order to protect the boot program from overwriting, you should program BootLock11 and BootLock12 (PonyProg) or „BLB1 *LPM_SPM_DISABLE*“ (Atmel Studio 6).

2.2. Program environment

2.2.1. Hyper Terminal (ALVIDI_BOOTLOADER)



Start the Hyper Terminal *Start*➔*Programs*➔*Accessories*➔*Communications*➔ *Hyper Terminal*



Connection Description

New Connection

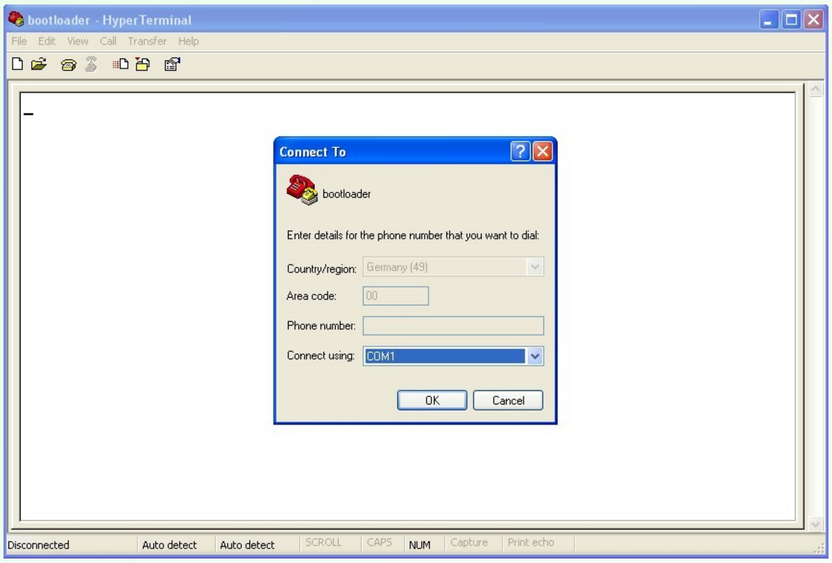
Enter a name and choose an icon for the connection:

Name:

bootloader

Icon:

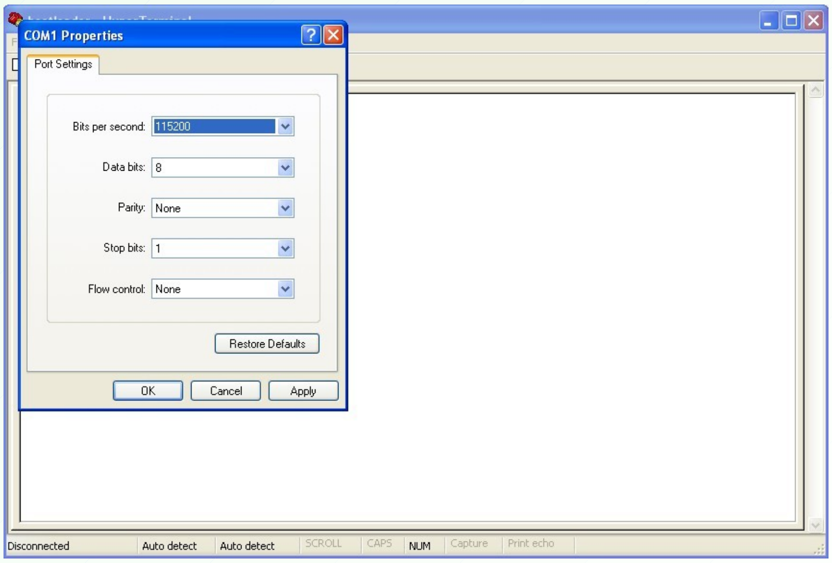
Enter the name, e.g. bootloader, and click the button „OK“.



Enter details for the phone number that you want to dial:

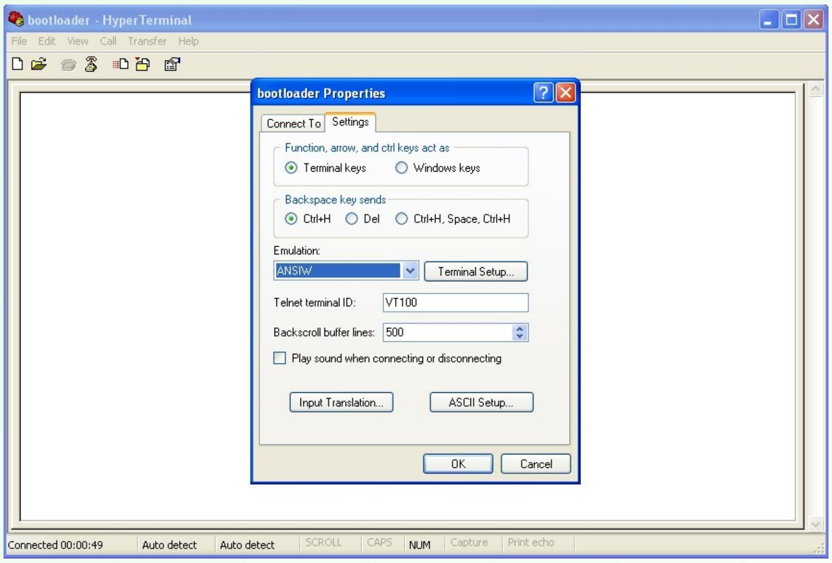
Country/region:	Germany (49)
Area code:	00
Phone number:	
Connect using:	COM1

Choose your serial port, e.g. COM1, and click the button „OK“.



Bits per second:	115200
Data bits:	8
Parity:	None
Stop bits:	1
Flow control:	None

Take over the parameters as shown above, and click the button „OK“. You will get the “Bits per Second” from the name of the Hex-File (s. Chapter 2.3 Hex-Files)


bootloader - HyperTerminal

bootloader Properties

Connect To | Settings

Function, arrow, and ctrl keys act as: [✓ Terminal keys] Windows keys

Backspace key sends: [✓ Ctrl+H] Del Ctrl+H, Space, Ctrl+H

Emulation: ANSIW

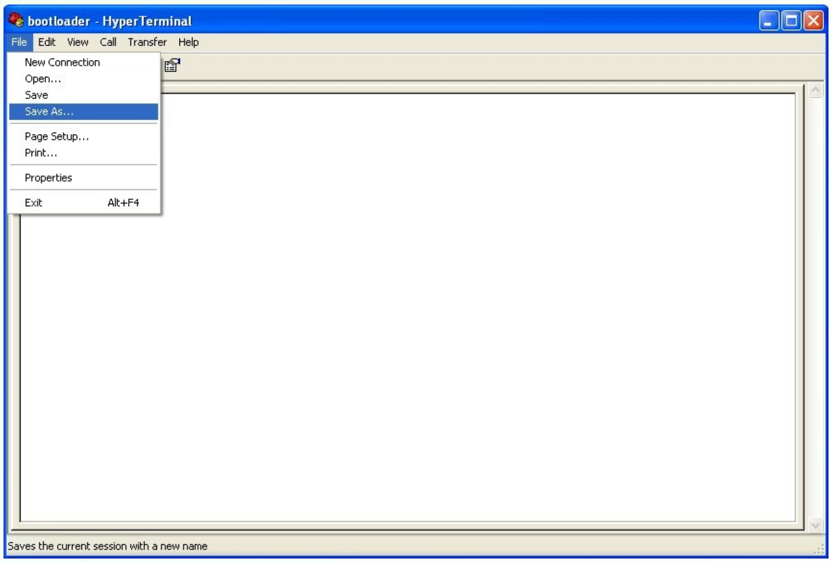
Telnet terminal ID: VT100

Backscroll buffer lines: 500

☐ Play sound when connecting or disconnecting

Connected 00:00:49 | Auto detect | Auto detect | SCROLL | CAPS | NUM | Capture | Print echo

In the window Properties *File*➔*Properties*➔*Settings* choose in the field *Emulation*➔*ANSIW or VT100*, and click the button „OK“.

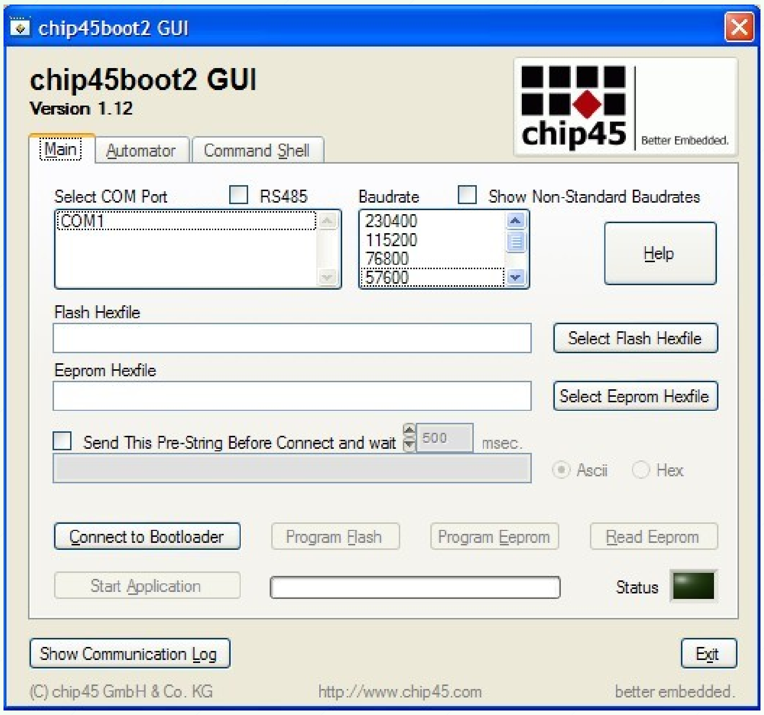

bootloader - HyperTerminal

Saves the current session with a new name

Save the settings, e.g. on the desktop, as b*ootloader.ht* .

2.2.2. chip45boot2 GUI
(CHIP45_BOOTLOADER)

In order to program with chip45 boot loader, download chip45boot2 GUI software on the home page www.chip45.com under following link:
http://download.chip45.com/chip45boot2_GUI_V1.12.zip

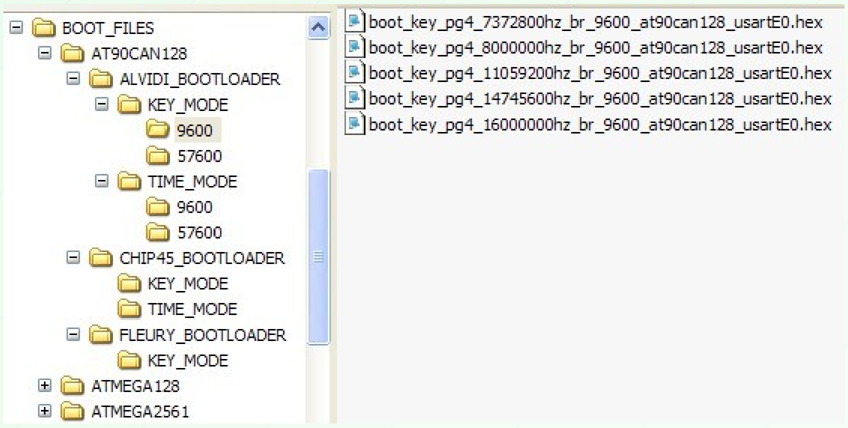

chip45boot2 GUI
Version 1.12

Select COM Port: COM1
RS485 [ ]
Baudrate: 230400, 115200, 76800, [✓ 57600]
Show Non-Standard Baudrates [ ]
Flash Hexfile:
Eeprom Hexfile:
[ ] Send This Pre-String Before Connect and wait 500 msec.
[✓ Ascii] Hex

2.2.3. AVR Studio 4
(FLEURY_BOOTLOADER)

Fleuery boot loader works with STK500 protocol. This protocol support Atmel free development environment: AVR Studio 4. You may download this software directly on Atmel home page:
http://www.atmel.com/tools/STUDIOARCHIVE.aspx

2.3. Hex-Files



```
BOOT_FILES
  AT90CAN128
    ALVIDI_BOOTLOADER
      KEY_MODE
        9600
        57600
      TIME_MODE
        9600
        57600
    CHIP45_BOOTLOADER
      KEY_MODE
      TIME_MODE
    FLEURY_BOOTLOADER
      KEY_MODE
  ATMEGA128
  ATMEGA2561
```

```
boot_key_pg4_7372800hz_br_9600_at90can128_usartE0.hex
boot_key_pg4_8000000hz_br_9600_at90can128_usartE0.hex
boot_key_pg4_11059200hz_br_9600_at90can128_usartE0.hex
boot_key_pg4_14745600hz_br_9600_at90can128_usartE0.hex
boot_key_pg4_16000000hz_br_9600_at90can128_usartE0.hex
```

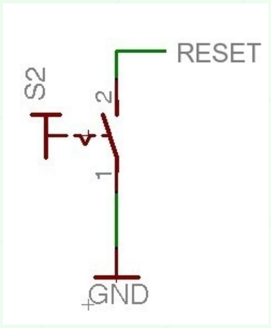
Hex files for boot loaders are available in archive BOOT_FILES.zip, which can be downloaded on our website. The boot files are located, as shown on the picture above, sorted by the controller, origin and the program mode (KEY_MODE and TIME_MODE) in appropriate folders. The boot files are available for every ATMEGA and AT90CAN128 Module in KEY_MODE and TIME_MODE for all quartz frequencies (16 MHz, 14.7456 MHz, 11.0592 MHz, 8 MHz, 7.3728 MHz).

The name of every single Hex file contains the program mode, the operating frequency, the baud rate, the controller and USART-port.

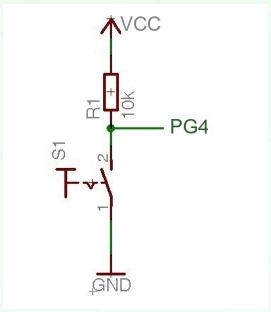
Example: boot_key_pg4_16000000hz_br_57600_at90can128_usartE0.hex - this file was written for the AVR module with AT90CAN128 controller which is equipped with 16 MHz external quartz. The baudrate is 57600 bits / s on USART-port E. The boot loader is activated by a reset and Pull-Down on the Pin PG4.

2.4. Hardware

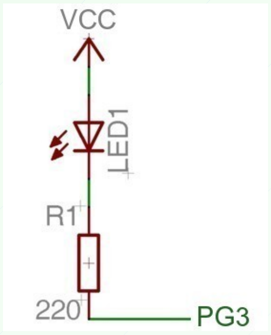
You will need an external circuit for programming with the boot program:



Connect the reset pin of module (if don't any key internal all ready installed), with a e.g. external key, the way as it is shown on the left picture.

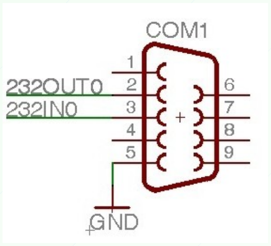


If you would like to use the boot program in key mode, you should take over the settings on the left picture, e.g. Port G Pin 4 (PG4), to your circuit.



As a check you may install a LED on PortG pin 3 (**PG3**) 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 . The Fuse Bits **BootLock11** and **Bootlock12** will be unprogrammed.

Re-installation of boot programs with AVR Studio 4

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 (see 2.3 Hex Files).

You should program the appropriate hex-file and set the following settings: ***Tools➔Program AVR➔Connect...➔ Connect... ➔ Fuses*** choose „**Boot Flash section size=2048 ...**“ and „**Boot Reset vector Enabled ...**“. In order to protect the boot program from overwriting, you should program ***Tools➔Program AVR➔Connect...➔ Connect... ➔ Lock Bits*** „BLB1 *LPM and SPM prohibited in Boot Section*“

Re-installation of boot programs with PonyProg

You should program the appropriate hex-file and set the following Fuse Bits: **BootLock11, Bootlock12, BOOTSZ1 and BOOTRST.**

Re-installation of boot programs with Atmel Studio 6

1. set boot loader size: ***Tools➔Device Programming➔Fuse*** **BOOTSZ 4096_1F00 und BOOTRST** choose.

2. select and program boot loader : ***Tools➔Device Programming➔ Memories➔Flash.*** With „**…**“-button choose your Hex-File and press „**Program**“-button.

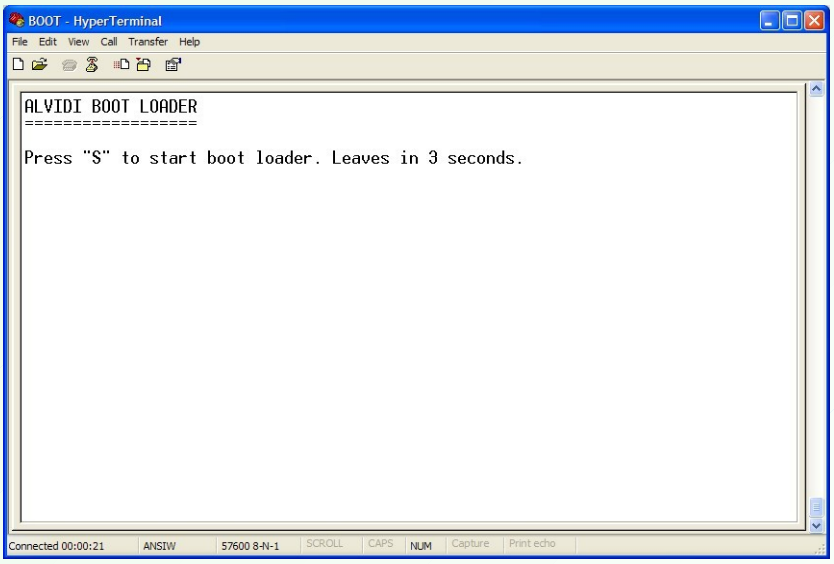
3. finally protect boot loader: ***Tools➔Device Programming➔Lock bits*** **BLB1 LPM_SPM_DISABLE**

4. Programming

4.1. Time Mode

4.1.1. ALVIDI_BOOTLOADER

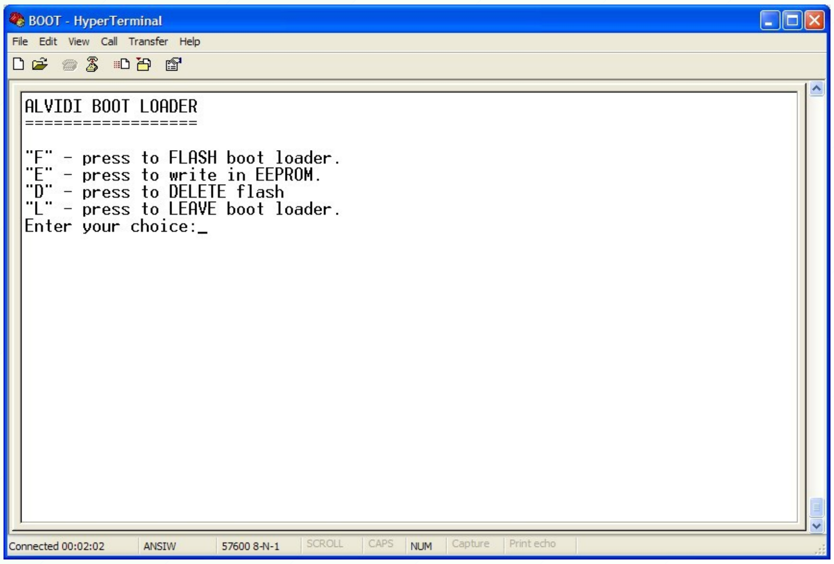
Start the saved settings of Hyper Terminals, e.g. bootloader.ht (see chapter **2.2.Hyper Terminal**)



```
ALVIDI BOOT LOADER
==================

Press "S" to start boot loader. Leaves in 3 seconds.
```

BOOT - HyperTerminal
File Edit View Call Transfer Help
Connected 00:00:21 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

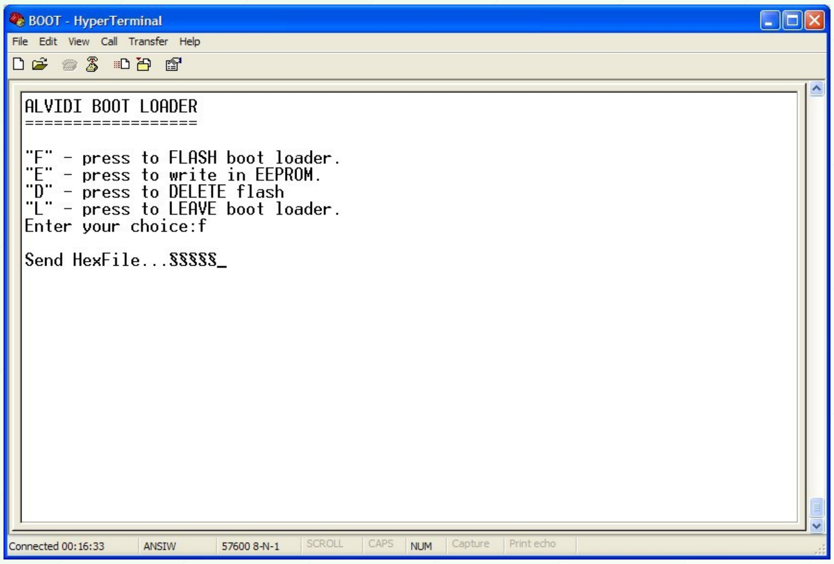
Press the reset button, in order to get the picture above in Hyper Terminal window. The boot program starts. Press the „S"-key to go to the boot program menu or press the „L"-key for leaving the boot program. Otherwise the boot program will be left automatically in 3 seconds.



```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:_
```

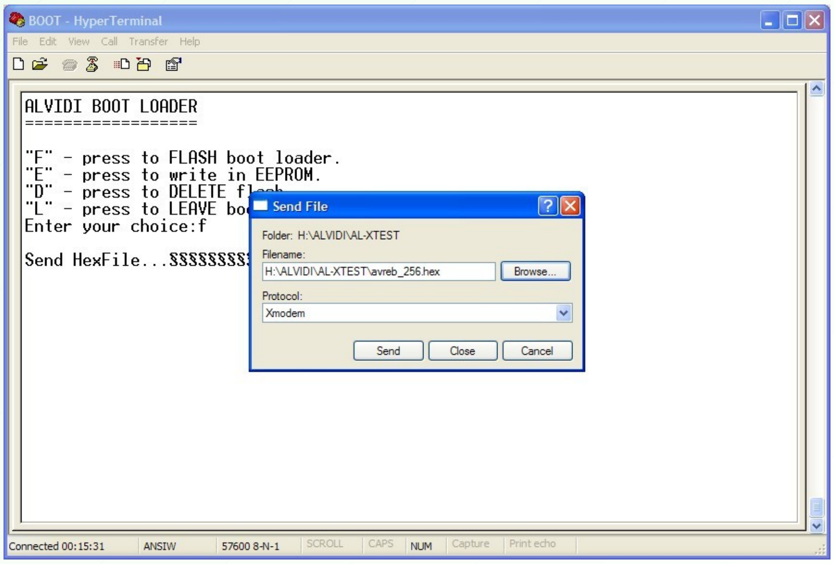
BOOT - HyperTerminal
File Edit View Call Transfer Help
Connected 00:02:02 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

In the boot program menu press the „F“-key for flash programming. With „E“-key you may write to the EEPROM, with „D“-key will be complete apliccation area deleted and with the „L“-key you may leave the boot loader program.



```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:f

Send HexFile...§§§§§_
```

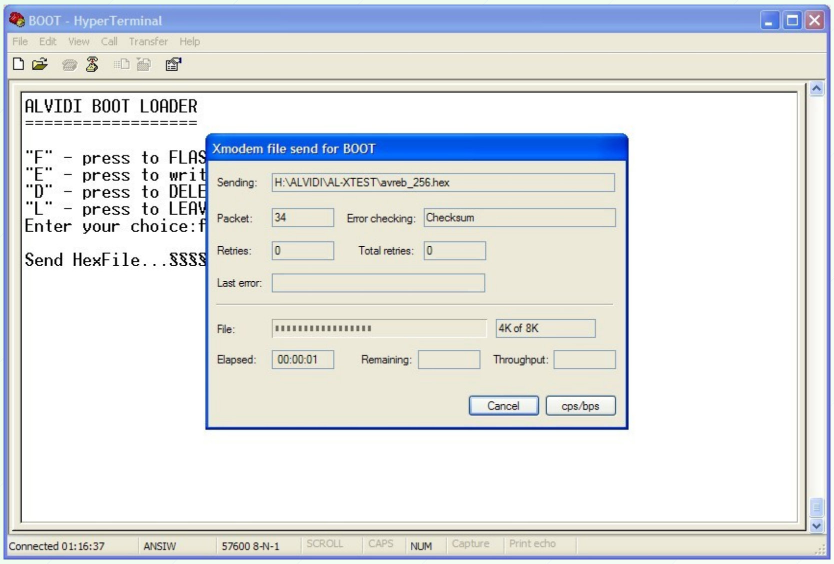
After you have pressed the „F”-key, „Send Hex-File...“ appears. You have now 60 seconds to send the hex-file to the controller, otherwise the boot program will be left.



```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE f
"L" - press to LEAVE bo
Enter your choice:f

Send HexFile...§§§§§§§§§
```

Folder: H:\ALVIDI\AL-XTEST
Filename: H:\ALVIDI\AL-XTEST\avreb_256.hex
Protocol: Xmodem

Click ***Transfer➔Send File...*** in the menu bar and choose your hex-file in the field **Filename:**. Choose **<u>Xmodem</u>** in the field **Protocol:** and click the button „**Send**“

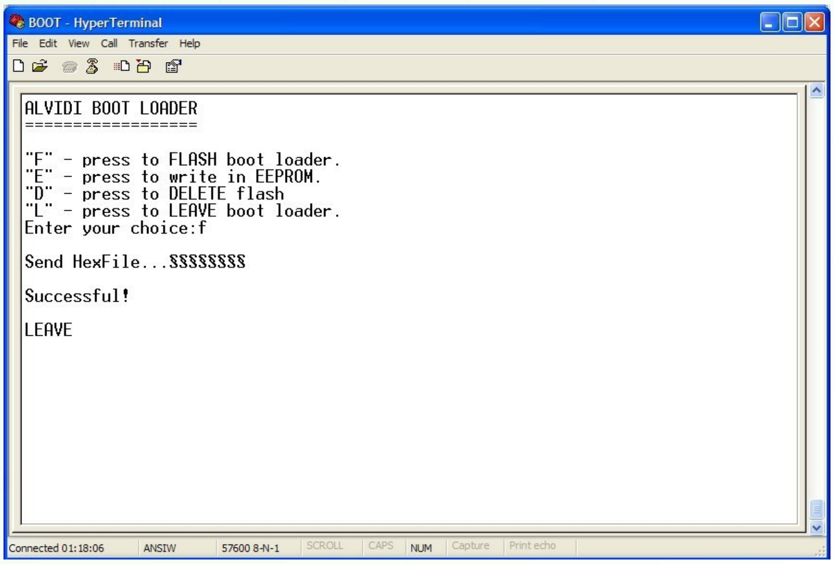


```
ALVIDI BOOT LOADER
==================

"F" - press to FLAS
"E" - press to writ
"D" - press to DELE
"L" - press to LEAV
Enter your choice:f

Send HexFile...§§§§
```

Sending: H:\ALVIDI\AL-XTEST\avreb_256.hex
Packet: 34 Error checking: Checksum
Retries: 0 Total retries: 0
Last error:
File: 4K of 8K
Elapsed: 00:00:01 Remaining: Throughput:

Connected 01:16:37 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

While the data is being sent to microcontroller, the details of the transmitting appear in window **„*Xmodem file send bootloader*”** (see the picture above)

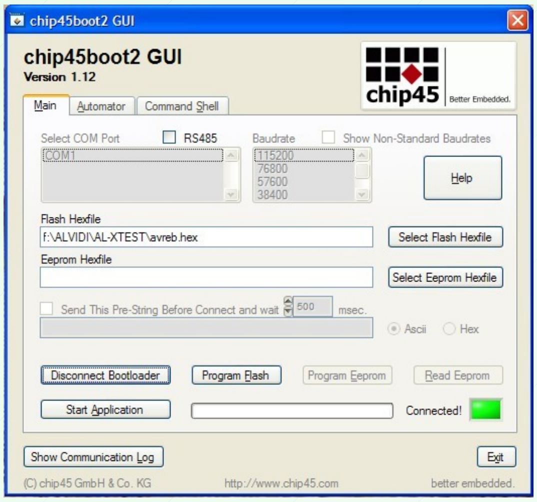


```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:f

Send HexFile...§§§§§§§§

Successful!

LEAVE
```

Connected 01:18:06 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

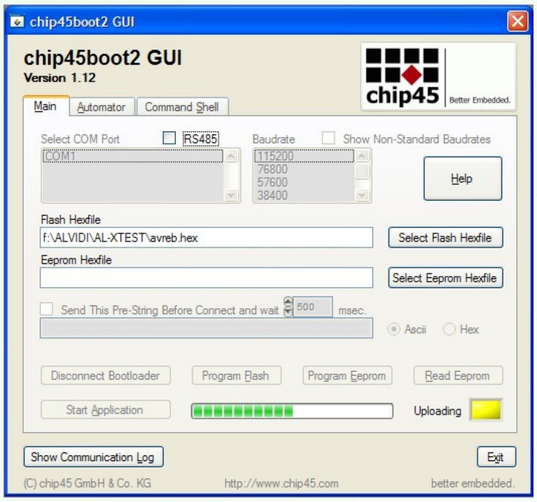
If the data transfer was successful, „**Successful!**“ appears and boot program will be left with the cue „**LEAVE**“.

4.1.2. CHIP45_BOOTLOADER



Select COM Port: COM1 — RS485 [ ]
Baudrate: 115200, 76800, 57600, 38400 — Show Non-Standard Baudrates [ ]
Flash Hexfile: f:\ALVIDI\AL-XTEST\avreb.hex
Eeprom Hexfile:
[ ] Send This Pre-String Before Connect and wait 500 msec.
Connected!

1. Start downloaded software chip45boot2 GUI
2. Choose in left window „Select COM Port“ with the module connected serial interface
3. In right window „Baudrate“ set your trans-mission rate. Recommended to start with low baud rate (19200).
4. Push the internal (if any) or external reset-key on AVR-module
5. If you within 3 seconds type the button „Connect to Bootloader“, you will get the left picture.



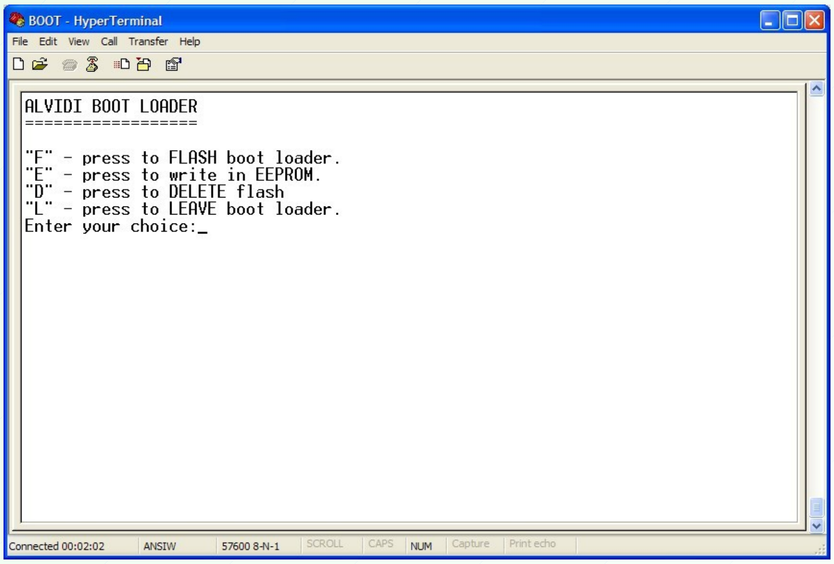
Select COM Port: COM1 — RS485 [ ]
Baudrate: 115200, 76800, 57600, 38400 — Show Non-Standard Baudrates [ ]
Flash Hexfile: f:\ALVIDI\AL-XTEST\avreb.hex
Eeprom Hexfile:
[ ] Send This Pre-String Before Connect and wait 500 msec.
Uploading

6. Press the button „Select Flash Hexfile“ and choose your Hex-File
7. In order to program or to write the flash click the button „Program Flash“, , you will get the left picture.
8. As soon as writing of the controller finished, you may start your program with the button „Start Application“.

4.2. Key Mode

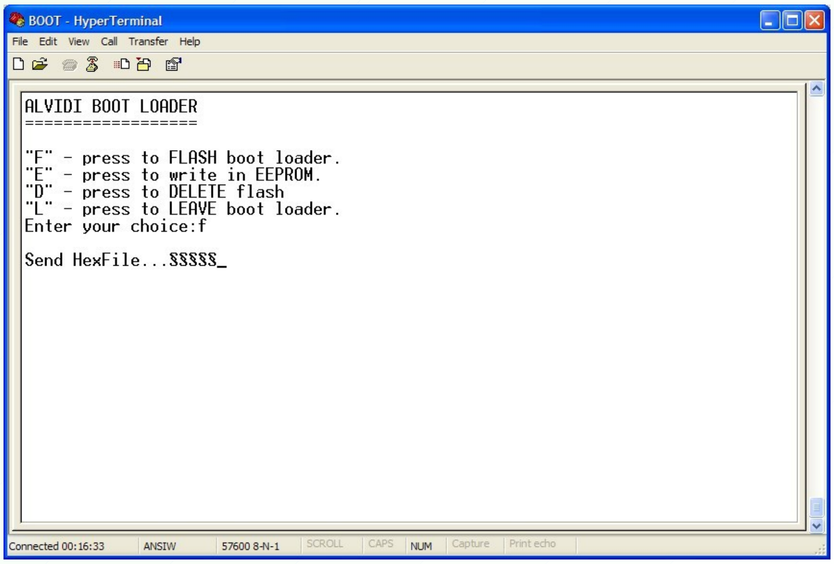
4.2.1. ALVIDI_BOOTLOADER

Start the saved settings of Hyper Terminals, e.g. bootloader.ht (see chapter **2.2.Hyper Terminal**)



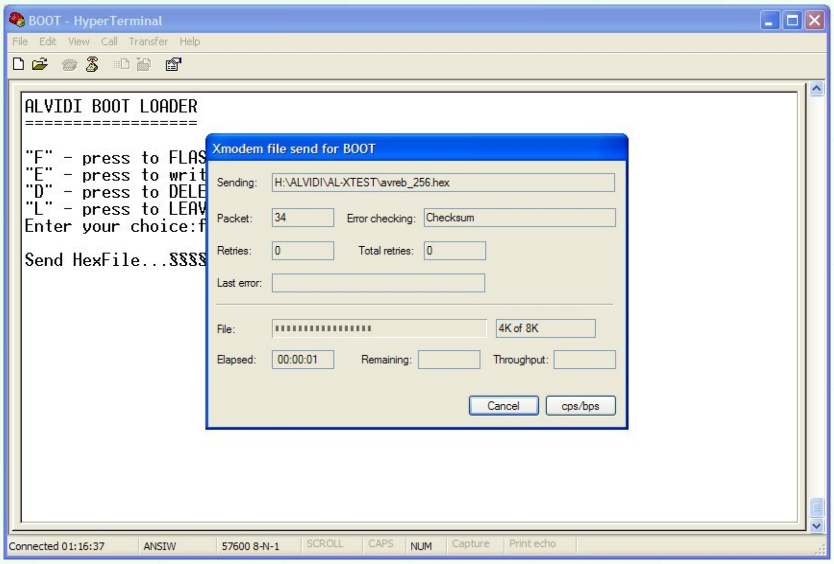
```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:_
```

Press the boot key (PG4) and the reset button together in order to get the picture above in Hyper Terminal window. In the boot program menu press the „F"-key for flash programming. With „E"-key you may write to the EEPROM, with „D"-key will be complete apliccation area deleted and with the „L"-key you may leave the boot loader program.



```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:f

Send HexFile...§§§§§_
```

After you have pressed the „F”-key, „Send Hex-File...“ appears. You have now 60 seconds to send the hex-file to the controller, otherwise the boot program will be left after 60 seconds.



BOOT - HyperTerminal
File Edit View Call Transfer Help

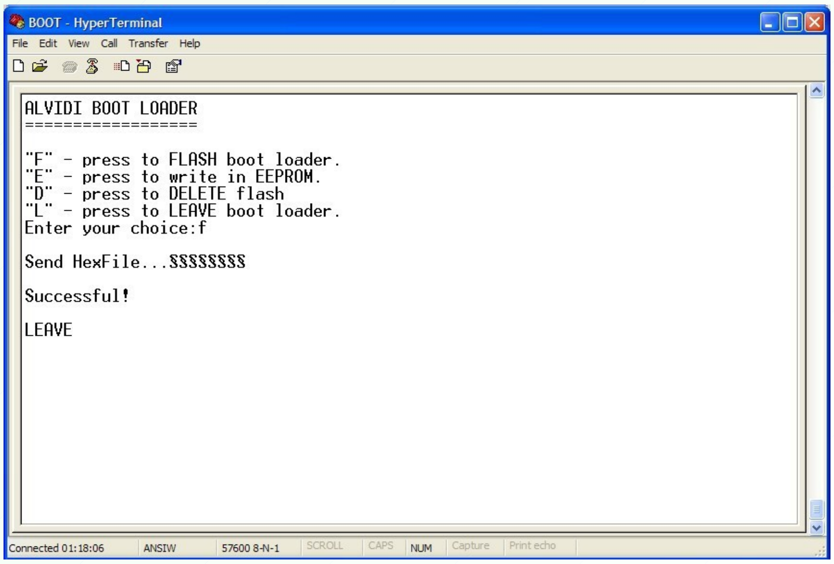
```
ALVIDI BOOT LOADER
==================

"F" - press to FLAS
"E" - press to writ
"D" - press to DELE
"L" - press to LEAV
Enter your choice:f

Send HexFile...§§§§
```

Xmodem file send for BOOT

Sending: H:\ALVIDI\AL-XTEST\avreb_256.hex
Packet: 34 Error checking: Checksum
Retries: 0 Total retries: 0
Last error:
File: 4K of 8K
Elapsed: 00:00:01 Remaining: Throughput:

Cancel
cps/bps
Connected 01:16:37 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

When the data to microcontroller is being sent, the details of the transmitting appear in the window ***„Xmodem file send bootloader”*** (see the picture above)



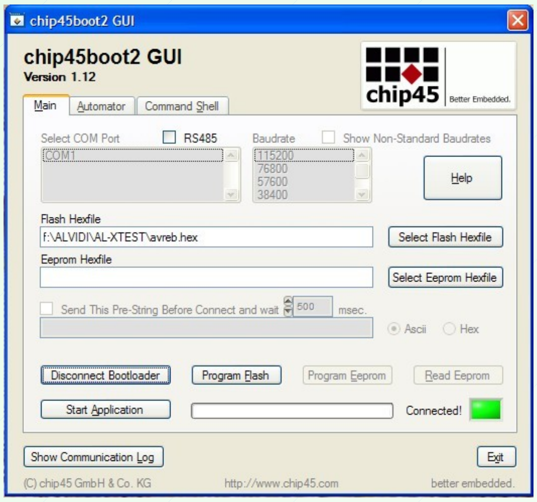
BOOT - HyperTerminal
File Edit View Call Transfer Help

```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:f

Send HexFile...§§§§§§§§

Successful!

LEAVE
```

Connected 01:18:06 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

If the data transfer was successful, „**Successful!**“ appears and the boot program will be left with the cue „**LEAVE!**“.

4.2.2. CHIP45_BOOTLOADER



Select COM Port: COM1 ☐ RS485

Baudrate: 115200, 76800, 57600, 38400 ☐ Show Non-Standard Baudrates

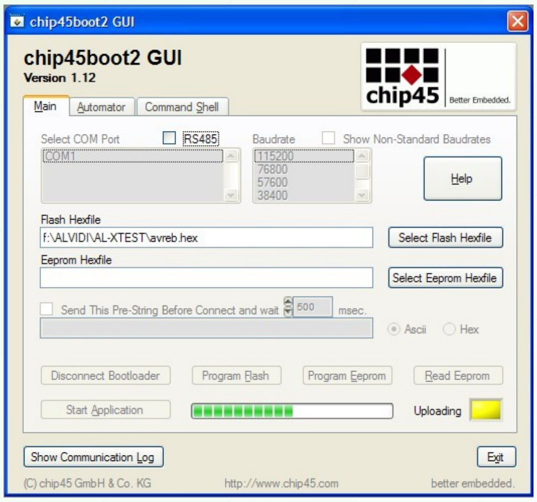
Flash Hexfile: f:\ALVIDI\AL-XTEST\avreb.hex

Eeprom Hexfile:

☐ Send This Pre-String Before Connect and wait 500 msec. ● Ascii ○ Hex

Connected! 🟩

1. Start downloaded software chip45boot2 GUI
2. Choose in left window „Select COM Port“ with the module connected serial interface
3. In right window „Baudrate“ set your transmission rate. Recommended to start with low baud rate (19200).
4. Push the internal (if any) or external reset-key on AVR-module with external pin-key (**PG4**) at the same time. After that loose the reset key at first.
5. Type the button „Connect to Bootloader“, you will get the left picture.



Select COM Port: COM1 ☐ RS485

Baudrate: 115200, 76800, 57600, 38400 ☐ Show Non-Standard Baudrates

Flash Hexfile: f:\ALVIDI\AL-XTEST\avreb.hex

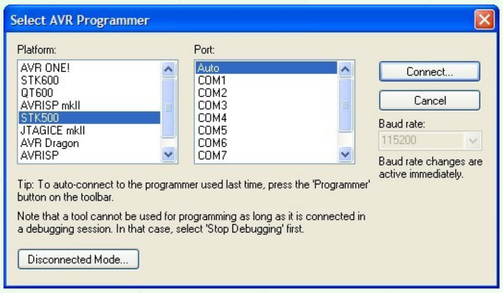
Eeprom Hexfile:

☐ Send This Pre-String Before Connect and wait 500 msec. ● Ascii ○ Hex

Uploading 🟨

6. Press the button „Select Flash Hexfile“ and choose your Hex-File
7. In order to program or to write the flash click the button „Program Flash“, , you will get the left picture.
8. As soon as writing of the controller finished, you may start your program with the button „Start Application“.

4.2.3. FLEURY_BOOTLOADER

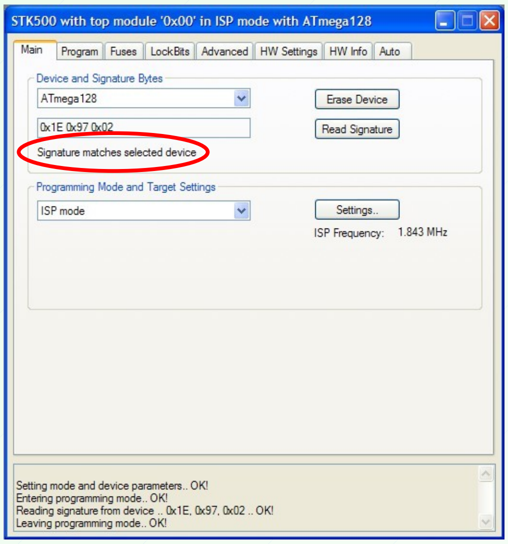


Baud rate: 115200

Baud rate changes are active immediately.

Tip: To auto-connect to the programmer used last time, press the 'Programmer' button on the toolbar.

Note that a tool cannot be used for programming as long as it is connected in a debugging session. In that case, select 'Stop Debugging' first.


STK500 with top module '0x00' in ISP mode with ATmega128
Main
Program
Fuses
LockBits
Advanced
HW Settings
HW Info
Auto
Device and Signature Bytes
Erase Device
Read Signature
Programming Mode and Target Settings
Settings..

ATmega128

0x1E 0x97 0x02

Signature matches selected device

ISP mode

ISP Frequency: 1.843 MHz

```
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading signature from device .. 0x1E, 0x97, 0x02 .. OK!
Leaving programming mode.. OK!
```

1. Push the internal (if any) or external reset-key on AVR-module with external pin-key (**PG4**) at the same time. After that loose the reset key at first.

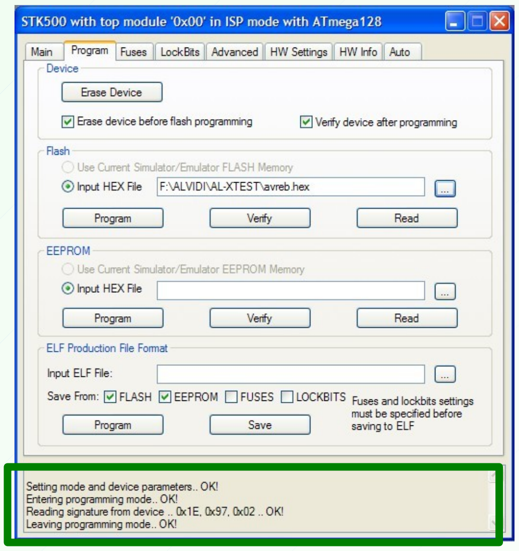
2. Start AVR Studio 4.

3. Click in taskbar ***Tools➔Program AVR➔Connect...***

4. Choose in left window „Platform“ **stk500** and press the button ***Connect...***

5. In sub-window *„Main > Device and Signature Bytes“* choose your controller.

6. In order to control your settings click the button *„Read Signature“*, if you took the right controller, you get ***„Signature matches selected device“***, see left picture



☑ Erase device before flash programming ☑ Verify device after programming

Flash: ○ Use Current Simulator/Emulator FLASH Memory ● Input HEX File F:\ALVIDI\AL-XTEST\avreb.hex

EEPROM: ○ Use Current Simulator/Emulator EEPROM Memory ● Input HEX File

Input ELF File:

Save From: ☑ FLASH ☑ EEPROM ☐ FUSES ☐ LOCKBITS Fuses and lockbits settings must be specified before saving to ELF

```
Setting mode and device parameters.. OK!
Entering programming mode.. OK!
Reading signature from device .. 0x1E, 0x97, 0x02 .. OK!
Leaving programming mode.. OK!
```

7. In sub-window “Program” press the button “...” opposite flash-enter-link-fenster and put your Hex-File.

8. With the button „Program“ under flash-enter-link-fenster you may flash controller with your program.

9. In lower window you get current information (mark with green line)

10. As soon you are ready with programming, press reset-key in order to start your program on the module.

4.3. Solving the Problems

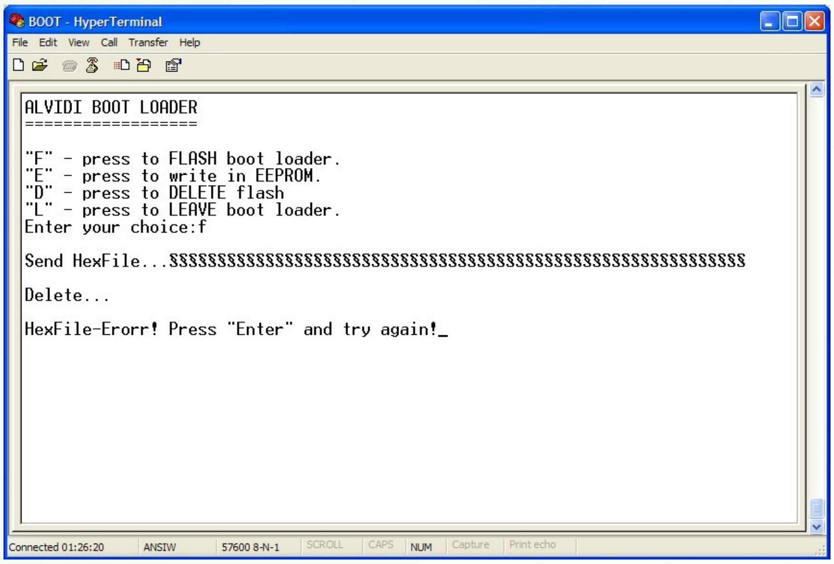
There is no perfect system in the world, various problems can take place. That's why we made a list of the most often happening mistakes.

1. Make sure, the jumpers of AVR-Development Module (all models: AL-AVREB JP2-1 and JP2-3) (all models: AL-ERAM128 JP5-1 and JP5-3) are set.

2. The pins PE0 and PE1 shouldn't have any connections.

3. Inspect hardware for the right connection (see chapter **2.4. Hardware**).

4. If a software problem occurs, reinstall the boot loader (see chapter **3. Re-installation**)



```
ALVIDI BOOT LOADER
==================

"F" - press to FLASH boot loader.
"E" - press to write in EEPROM.
"D" - press to DELETE flash
"L" - press to LEAVE boot loader.
Enter your choice:f

Send HexFile...SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS

Delete...

HexFile-Erorr! Press "Enter" and try again!_
```

Connected 01:26:20 ANSIW 57600 8-N-1 SCROLL CAPS NUM Capture Print echo

5. If the hex-file wasn't sent to controller in 60 seconds, the picture above will appear.
6. If the data transfer was stopped during programming, inspect **<u>Point 2 (see above)</u>.**
7. If after the re-installation the boot loader menu is not visible in the hyper-terminal, the installation of the newest version the AVR studio software could be helpful.

5. Links

○ Bootloader Hex-Files BOOT_FILES.zip (0,32 MB, 5/2013)
http://alvidi.de/data_sheets/BOOT_FILES.zip

○ AVR Studio 4.19 (124 MB, revision build 730, updated 9/11)
http://www.atmel.com/tools/STUDIOARCHIVE.aspx

○ Atmel Studio 6.1 (build 2562) Installer – Full
http://www.atmel.com/tools/ATMELSTUDIO.aspx

○ Frei zu verwendender chip45boot2 Bootloader für AVR ATmega und Xmega µC
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

○ 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

We are not liable for the problems, which occur during incorrect usage of our products.

We don't take liability for mistakes, which may occur while using our products.

We don't take responsibility for possible damages, which may happen while using our products.

The usage of the software is permitted without limitation only in correspondence to the distributed hardware. The usage of the software with different hardware without our written approval is prohibited.

All rights reserved.