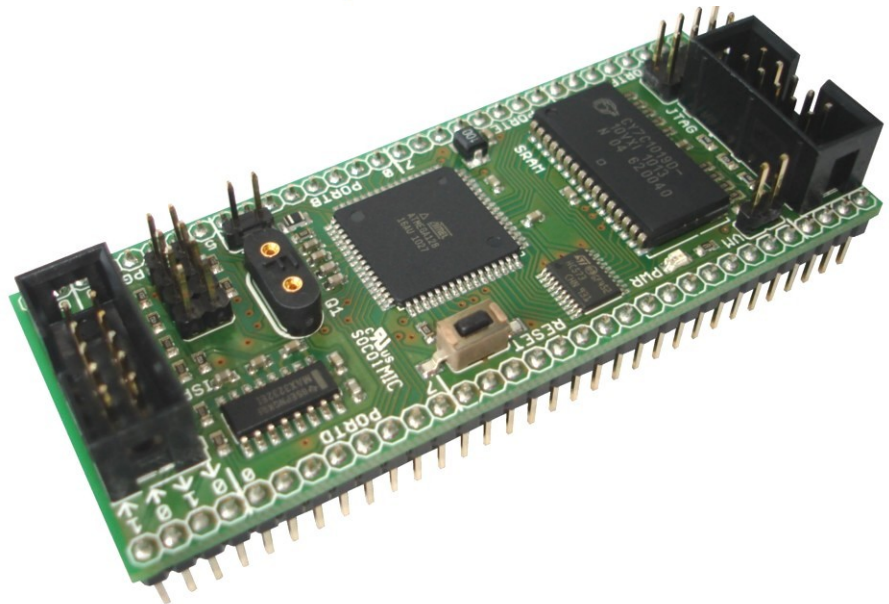
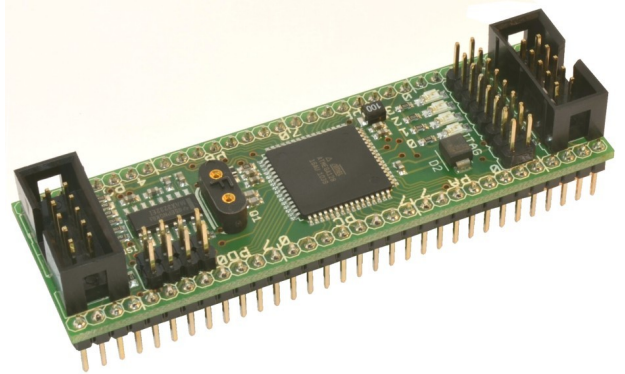


BOOTLOADER

for AVR-Development Modules

- with ATmega128
- with AT90CAN128
- with ATmega2561



Directory

1. Instructions	3
2. Settings	4
2.1. Fuse Bits	4
2.2. Program environment	6
2.2.1. Hyper Terminal (HT)	6
2.2.2. chip45boot2 GUI	9
2.2.3. AVR Studio4	9
2.3. Hex-Files	10
2.4. Hardware	11
3. Re-installation	12
4. Programming	13
4.1. Time Mode	13
4.1.1. ALVIDI_BOOTLOADER	13
4.1.2. CHIP45_BOOTLOADER	16
4.2. Key Mode	17
4.2.1. ALVIDI_BOOTLOADER	17
4.2.2. CHIP45_BOOTLOADER	19
4.2.3. FLEURY_BOOTLOADER	20
4.3. Solving the Problems	21
5. Links	22
6. Disclaimer	23

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

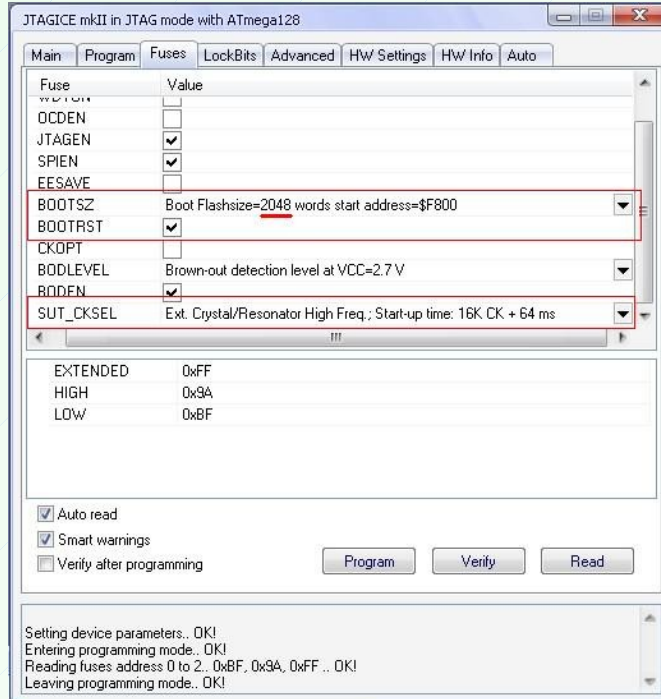
As soon boot loader started, pin PG3 get internal Pull-Down.

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.

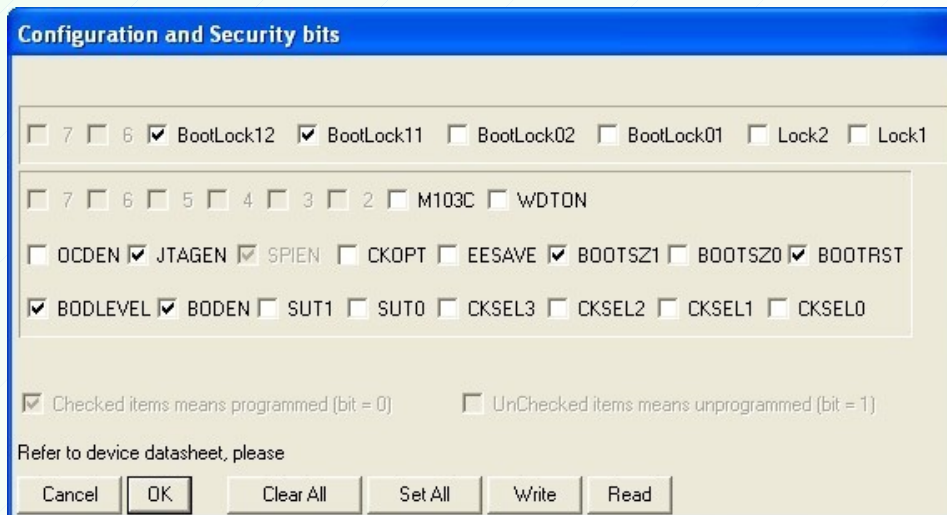
2. Settings

2.1. Fuse Bits

Settings of the Fuse Bits with AVR Studio 4



Settings of the Fuse Bits with PonyProg



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

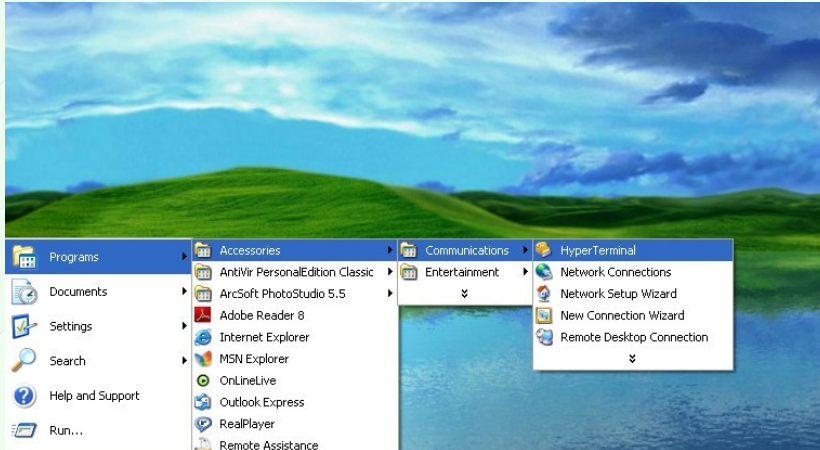
The size of the boot program is 4000 bytes, which is equal to 2000 words (1 byte=8 bit and 1 word=16 bit).

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 \$F800.

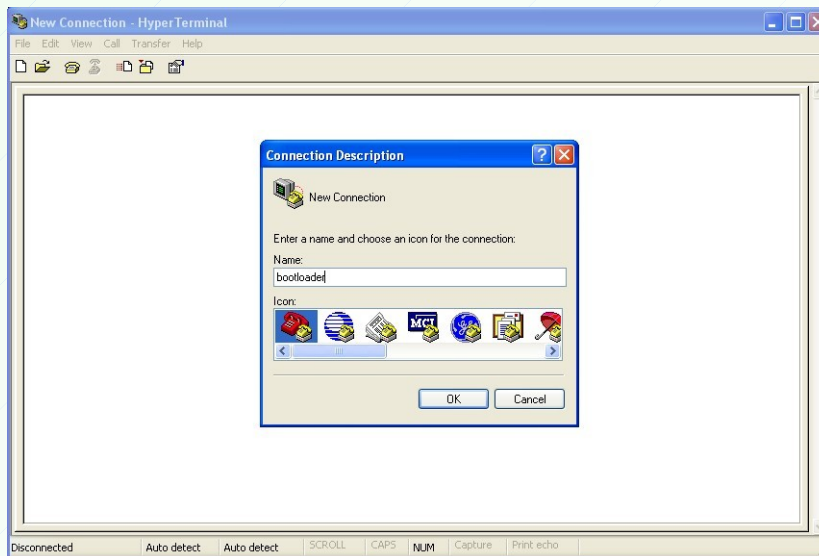
In order to protect the boot program from overwriting, you should program BootLock11 and BootLock12 (PonyProg) or „BLB1 LPM and SPM prohibited in Boot Section“ (AVR Studio).

2.2. Program environment

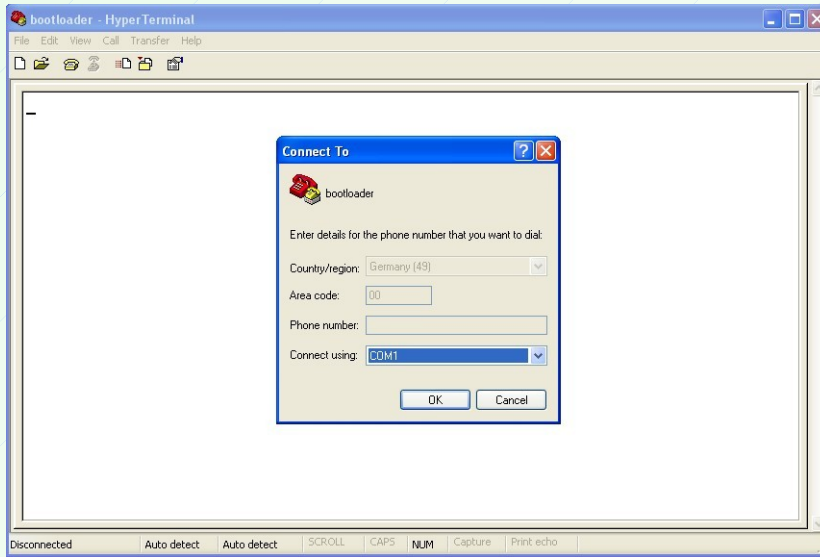
2.2.1. Hyper Terminal (ALVIDI_BOOTLOADER)



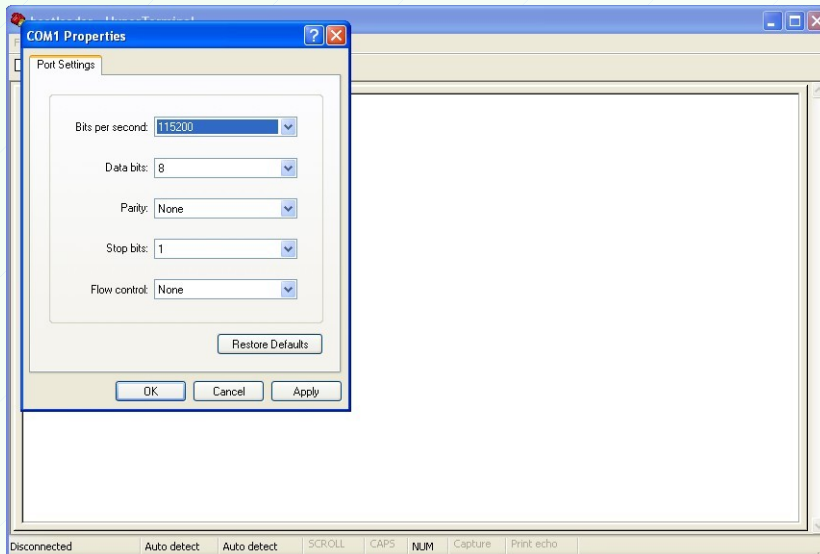
Start the Hyper Terminal *Start* → *Programs* → *Accessories* → *Communications* → *Hyper Terminal*



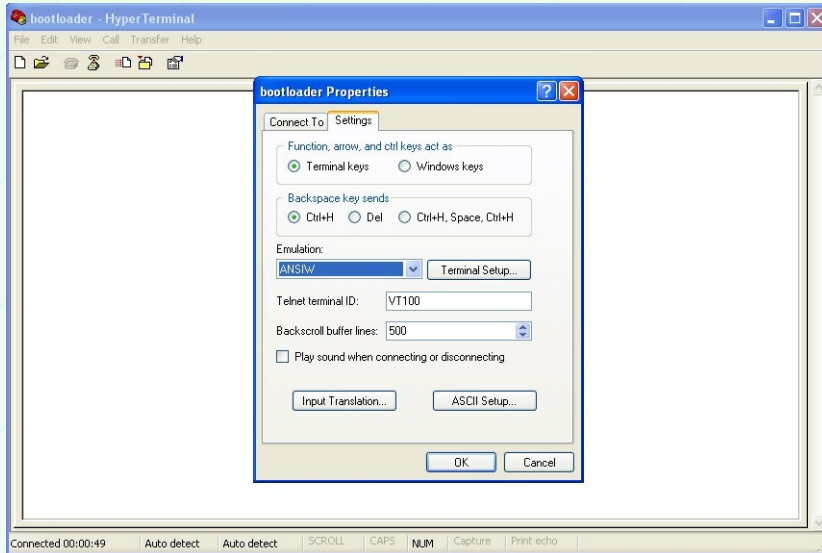
Enter the name, e.g. bootloader, and click the button „OK“.



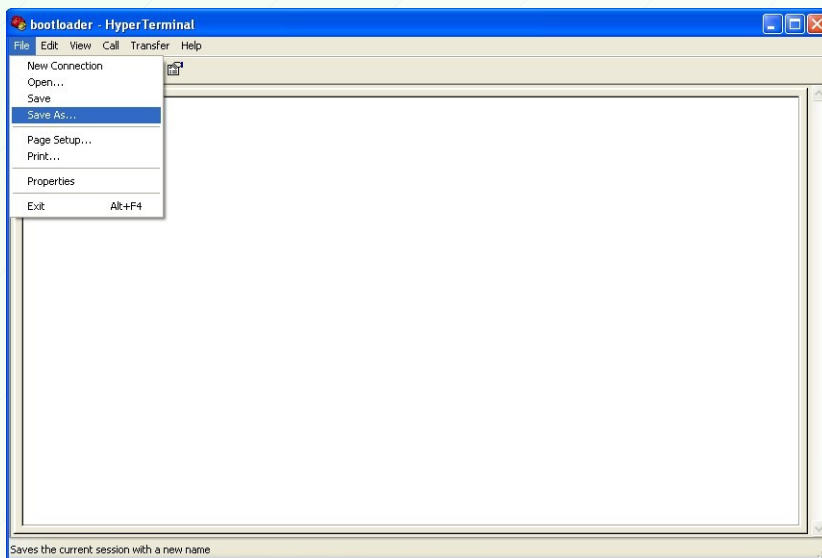
Choose your serial port, e.g. COM1, and click the button „OK“.



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)



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

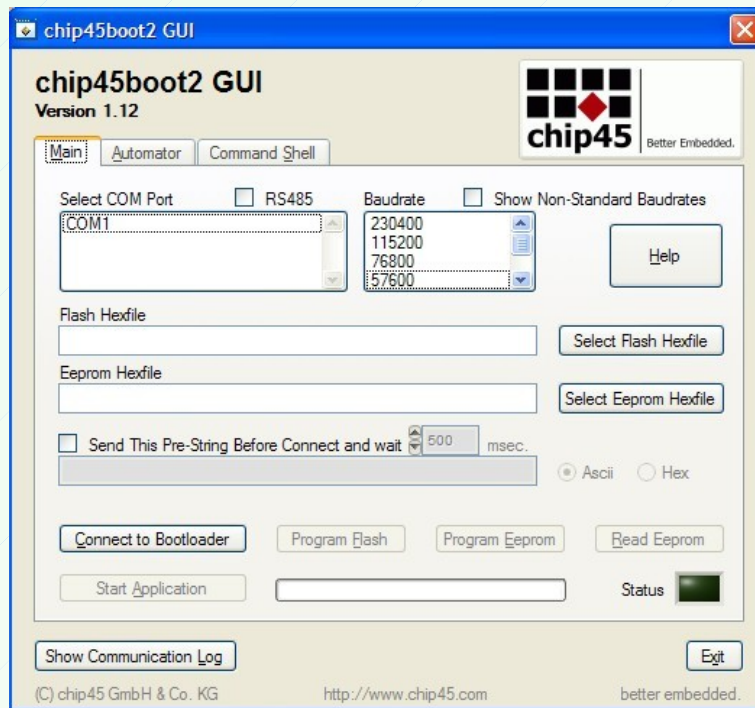


Save the settings, e.g. on the desktop, as *bootloader.ht* .

2.2.2. chip45boot2 GUI (CHIP45_BOOTLOADER)

You may download this software on the home page www.chip45.com under following link:

http://download.chip45.com/chip45boot2_GUI_V1.12.zip

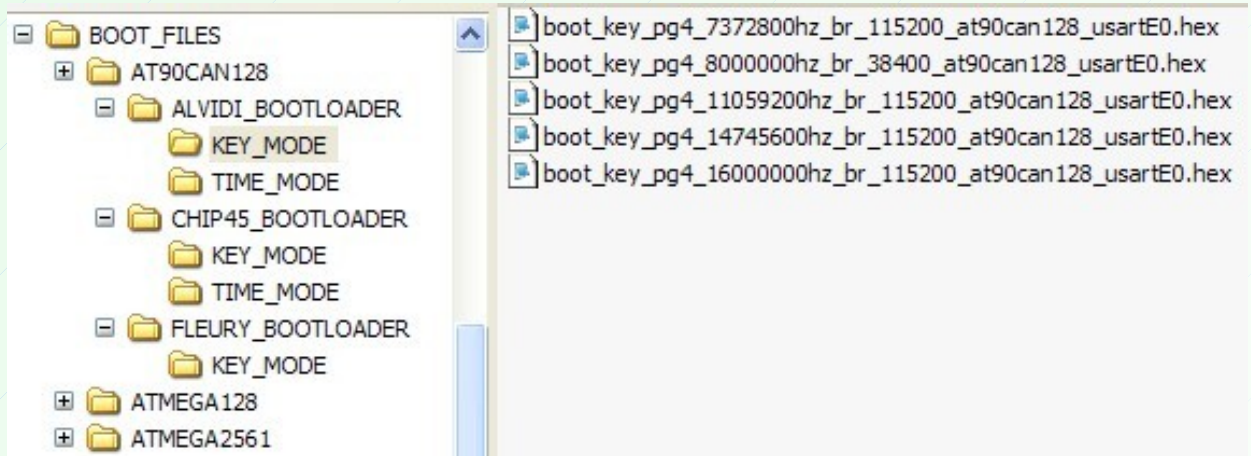


2.2.3. AVR Studio 4 (FLEURY_BOOTLOADER)

Atmel offer with own products free development environment: AVR Studio 4. You may download this software directly on Atmel home page:

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

2.3. Hex-Files



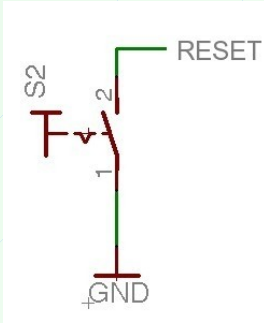
Hex files for boot loaders are available in archive [BOOT_FILES.zip](#), which can be downloaded on our website. The boot files are located, as shown on the picture above, sorted by the controller, origin and the program mode (KEY_MODE and TIME_MODE) in appropriate folders. The boot files are available 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.

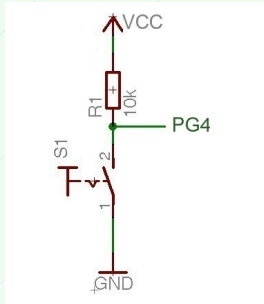
Example: `boot_key_pg4_16000000hz_br_115200_at90can128_usartE0.hex` - this file was written for the AVR module with AT90CAN128 controller which is equipped with 16 MHz external quartz. The baudrate is 115200 bits / s on USART-port E. The boot loader is activated by a reset and Pull-Down on the Pin PG4.

2.4. Hardware

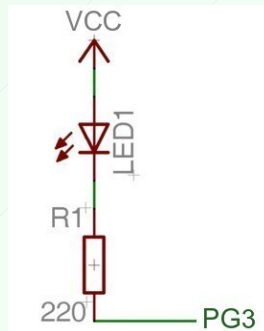
You will need an external circuit for programming with the boot program:



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.

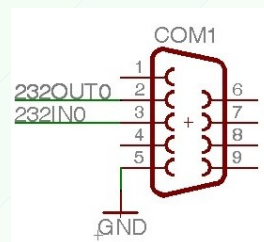


If you would like to use the boot program in key mode, you should take over the settings on the left picture, e.g. Port G Pin 4 (PG4), to your circuit.



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

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

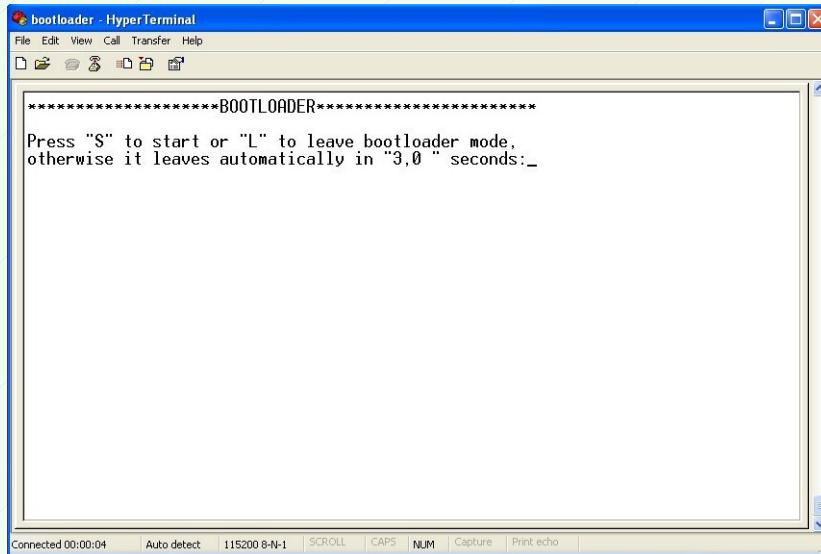
You should program the appropriate hex-file and set the following Fuse Bits: **BootLock11, Bootlock12, BOOTSZ1 and BOOTRST.**

4. Programming

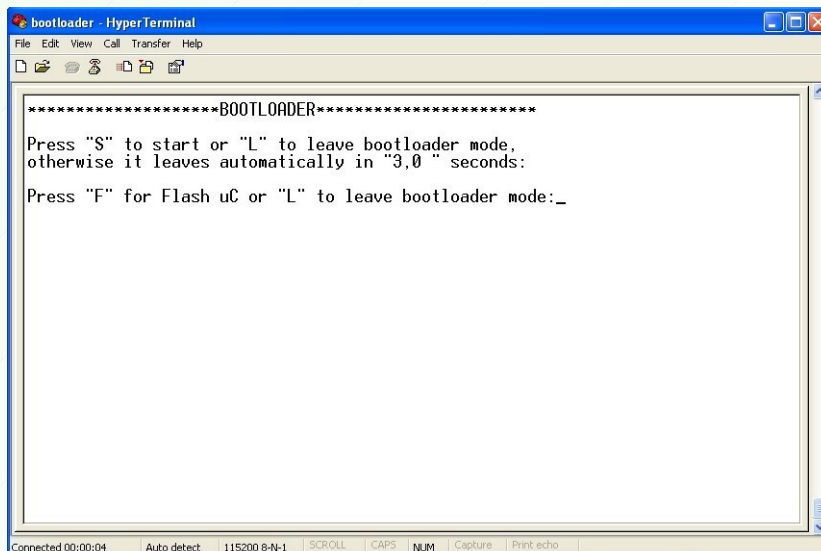
4.1. Time Mode

4.1.1. ALVIDI_BOOTLOADER

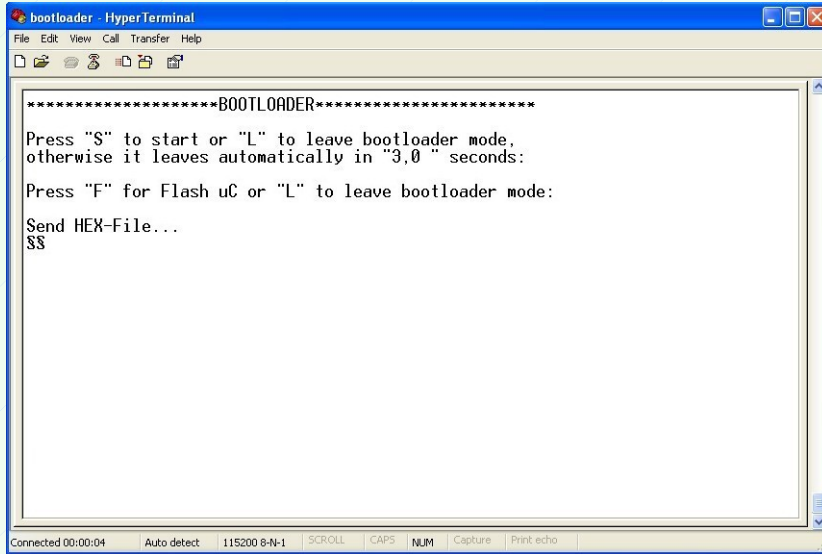
Start the saved settings of Hyper Terminals, e.g. bootloader.ht (see chapter 2.2.Hyper Terminal)



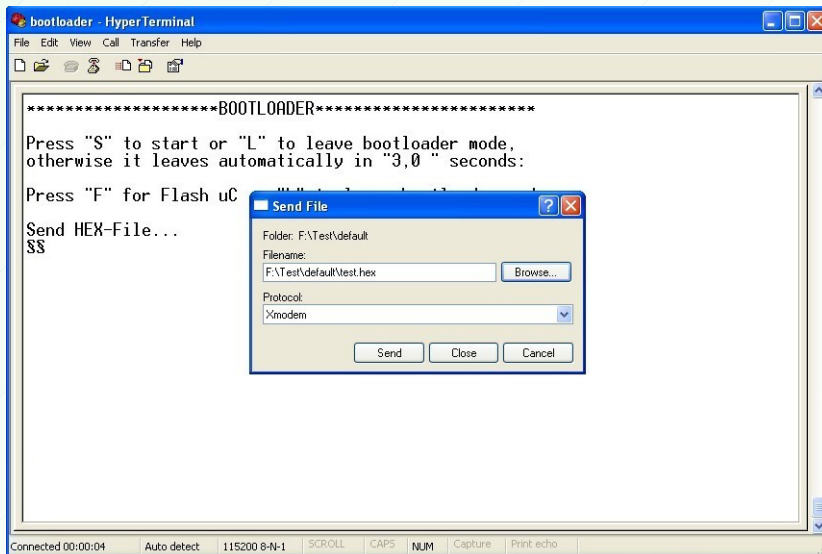
Press the reset button, in order to get the picture above in Hyper Terminal window. The boot program starts. Press the „S”-key to go to the boot program menu or press the „L”-key for leaving the boot program. Otherwise the boot program will be left automatically in 3 seconds.



In the boot program menu press the „F“-key for flash programming or the „L“-key to leave boot program.

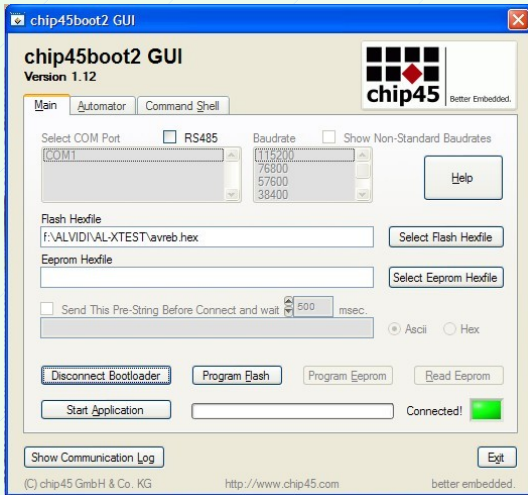


After you have pressed the „F“-key, „Send Hex-File...“ appears. You have now 60 seconds to send the hex-file to the controller, otherwise the boot program will be left.

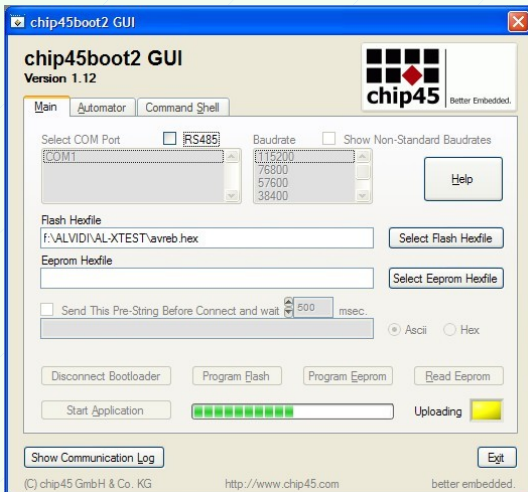


Click **Transfer** → **Send File...** in the menu bar and choose your hex-file in the field **Filename:**. Choose **Xmodem** in the field **Protocol:** and click the button „**Send**“

4.1.2. CHIP45_BOOTLOADER



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
5. If you within 3 seconds type the button „Connect to Bootloader“, you will get the left picture.

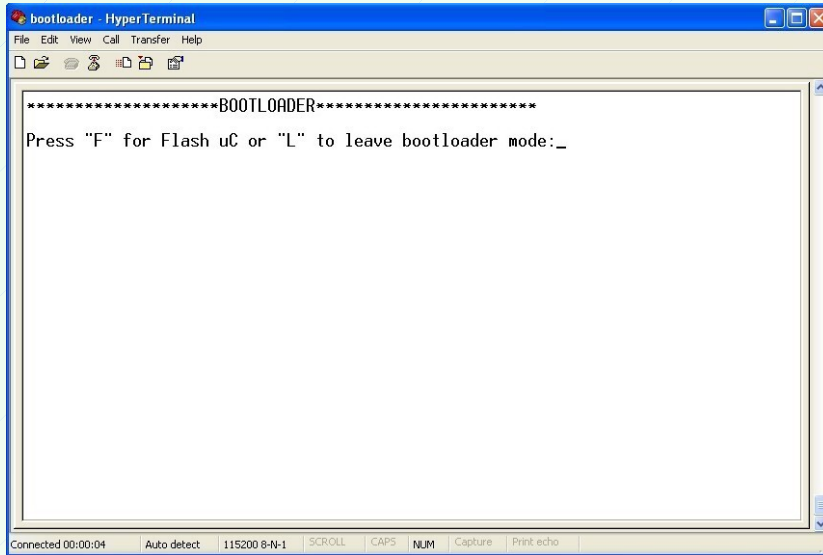


6. Press the button „Select Flash Hexfile“ and choose your Hex-File
7. In order to program or write the flash click the button „Program Flash“, you will get the left picture.
8. As soon as writing of the controller finished, you may start your program with the button „Start Application“.

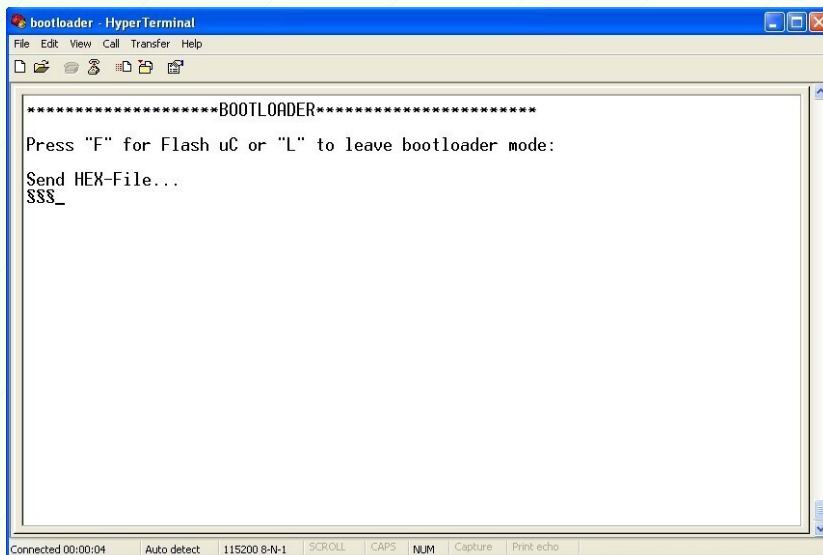
4.2. Key Mode

4.2.1. ALVIDI_BOOTLOADER

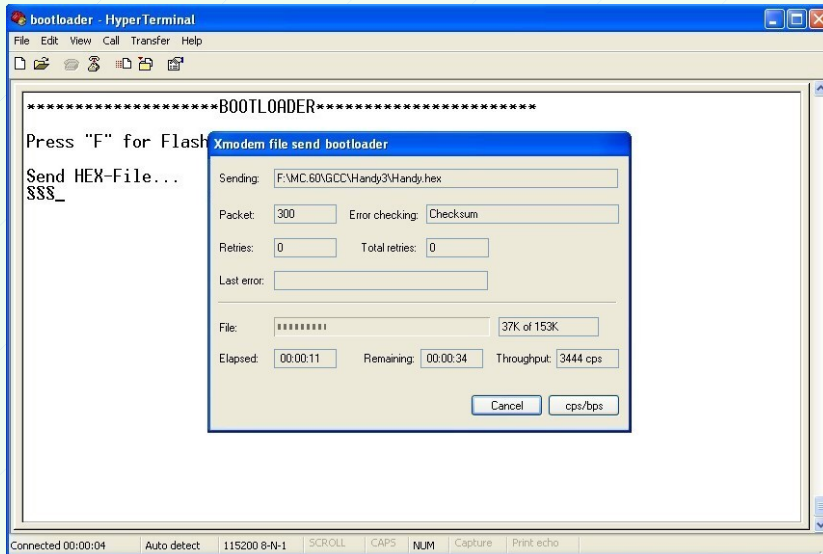
Start the saved settings of Hyper Terminals, e.g. bootloader.ht (see chapter 2.2.Hyper Terminal)



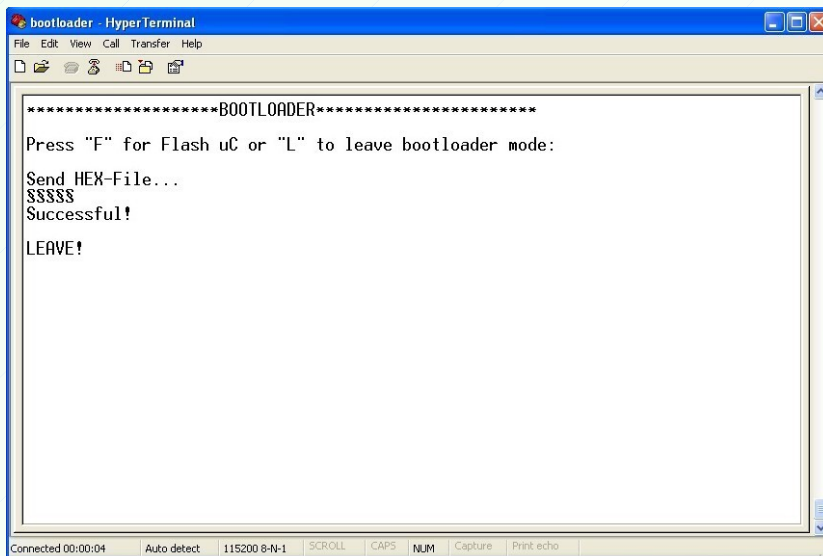
Press the boot key (PG4) and the reset button together in order to get the picture above in Hyper Terminal window. Press the „F“-key in the boot program menu for flash programming or the „L“-key to leave boot program.



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.

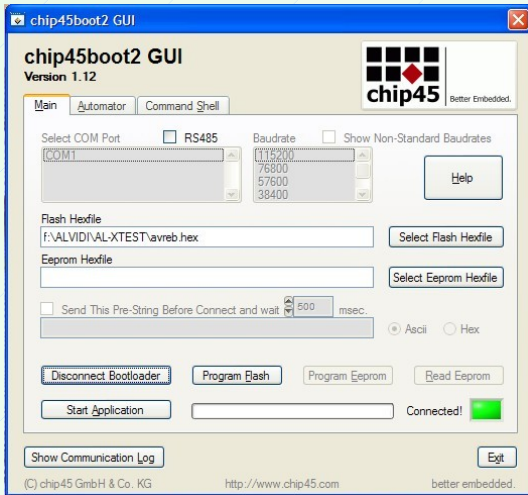


When the data to microcontroller is being sent, the details of the transmitting appear in the window „*Xmodem file send bootloader*” (see the picture above)

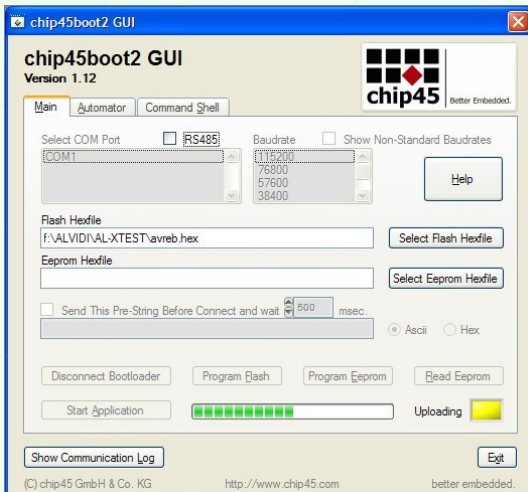


If the data transfer was successful, „**Successful**” appears and the boot program will be left with the cue „**LEAVE!**”.

4.2.2. CHIP45_BOOTLOADER

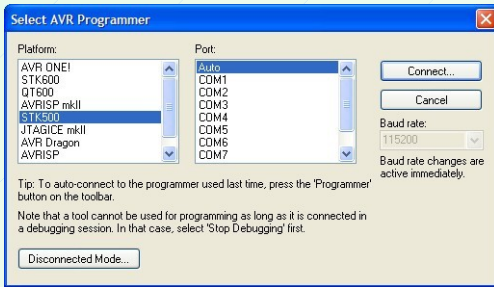


1. Start downloaded software chip45boot2 GUI
2. Choose in left window „Select COM Port“ with the module connected serial interface
3. In right window „Baudrate“ set your transmission rate. Recommended to start with low baud rate (19200).
4. Push the internal (if any) or external reset-key on AVR-module 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.



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



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.

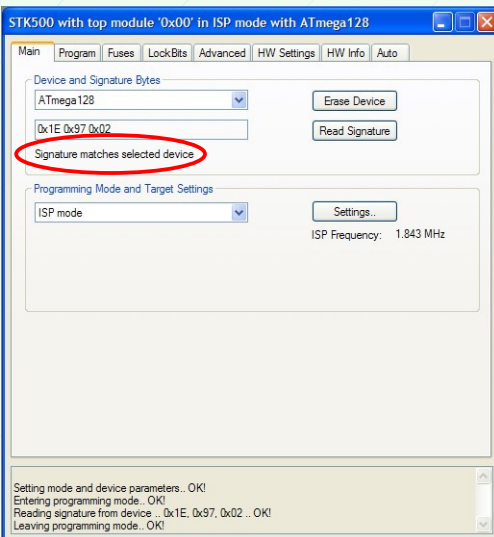
2. Start AVR Studio 4.

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

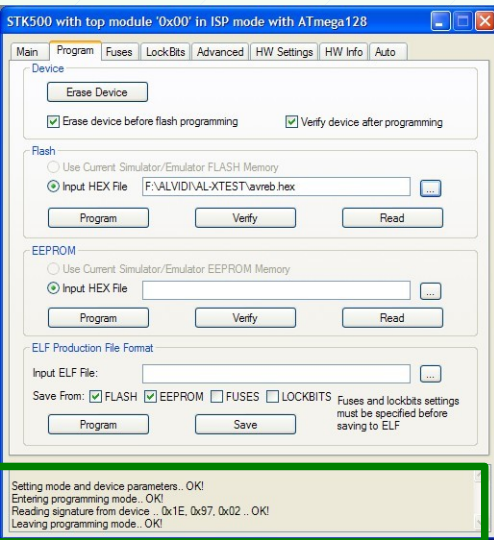


7. In sub-window “Program” press the button “...” opposite flash-enter-link-fenster and put your Hex-File.

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)

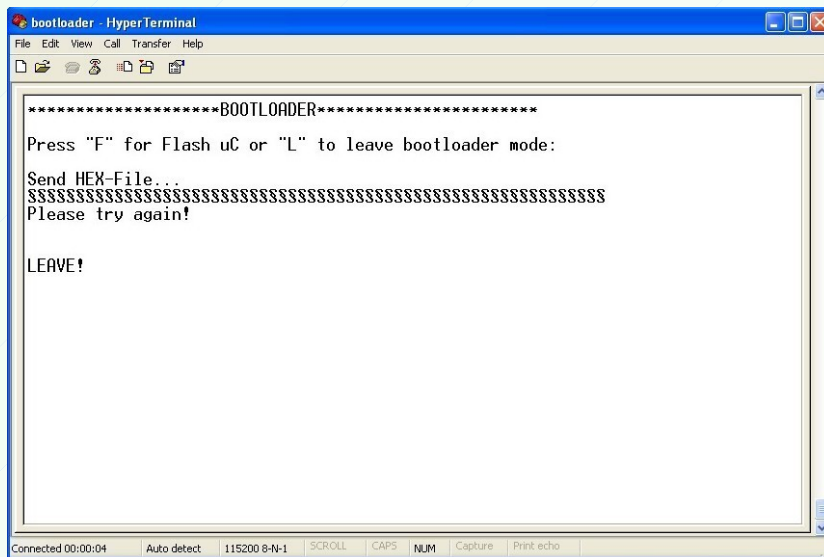
10. As soon you are ready with programming, press reset-key in order to start your program on the module.



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**)



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 **Point 2 (see above)**.
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 (0,18 MB September 2011)
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
- 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
- Royalty Free chip45boot2 Bootloader for AVR ATmega and Xmega μ 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
- 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.