

Tutorial

„ICnova AP7000 Base“

Content

1 INTRODUCTION.....	3
1.1 PACKAGE CONTENT.....	4
1.2 INSTALLATION.....	4
1.3 DRIVER FOR LINUX.....	5
1.4 DRIVER FOR WINDOWS.....	5
1.5 NETWORK SETTINGS.....	6
2 USAGE.....	7
2.1 SHELL ACCESS.....	7
2.2 SWITCHING LEDs.....	7
2.3 CONTROLLING GPIO-PORTS.....	8
2.4 WRITING YOUR OWN PROGRAMS.....	8
3 OWN DEVELOPMENTS FOR ICNOVA AP7000 BASE.....	8
3.1 PREMISES.....	8
3.2 SELECTION OF SOFTWARE.....	9
3.3 TOOLCHAIN.....	10
3.4 THE FILE SYSTEM OF THE ICNOVA.....	11
3.5 TRANSFERRING DATA TO THE ICNOVA.....	11
3.5.1 Single files.....	11
3.5.2 Transfer an entire image via JTAG.....	12
3.5.3 Transfer an entire image over network.....	12

Revision Overview

Version	Date	Reason for changes	Editor
A	10.04.08	original	Tietz
B	15.04.09	adaption of original to new kernel and toolchain	Ullrich

1 Introduction

Thank you for purchasing the ICnova AP7000 Base. To ease your first steps, we would like to give you a short introduction on this board.

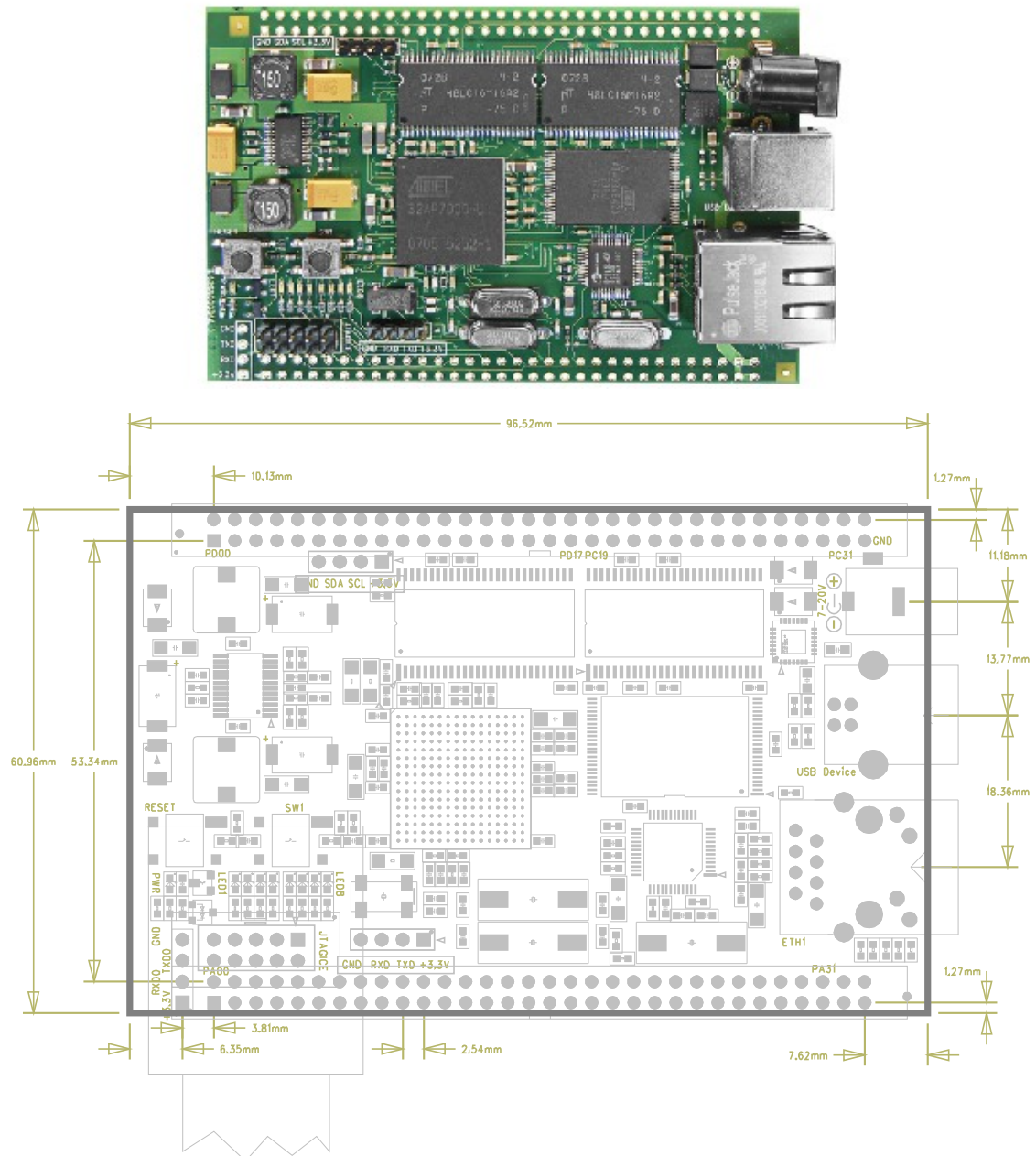


figure 1: ICnova AP7000 Base

1.1 Package Content

The package with your ICnova Base contains the following items:

- ICnova AP7000 Base with preinstalled Linux kernel 2.6.28
- this introduction paper
- a USB cable

1.2 Installation

Figure 1 shows an ICnova AP7000 Base, herein after called ICnova.

The ICnova can be powered by an external power supply (10-36V DC) or directly via the USB cord.

As soon as the ICnova has started, it tries to receive a valid IP-Address from the network via DHCP. Afterwards it provides the services Telnet and HTTP. Other services can be added on demand.

The USB connector provides access via a virtual serial interface. While booting, the boot messages are written to this interface. Set up the terminal program on your host PC with following configurations:

115200Baud, 8 data bits , no parity , 1 stop bit and no flowcontrol.

As an alternative a second debug device can be used on the 4-pole pinheader next to the JTAG connector. It is preset to 38400Baud,8,N,1 and has to be connected via a RS232 level-shifter or an USB-UART-Bridge since the signal level on these pins is only 3.3V. The debug device can also be used via a wireless connection, using an ICradio Mini 2.4G.

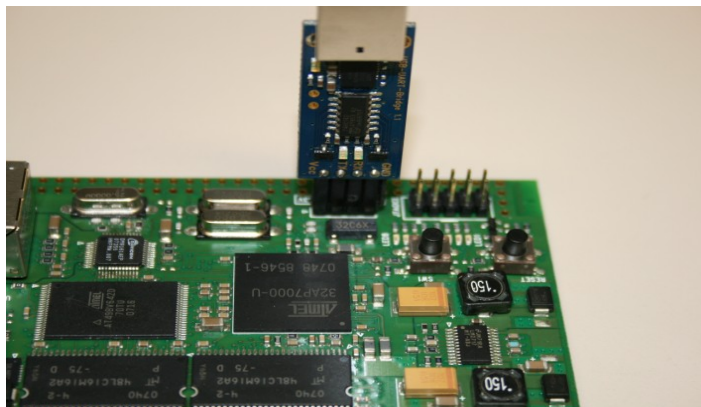


figure 2: USB-UART-Bridge on ICnova

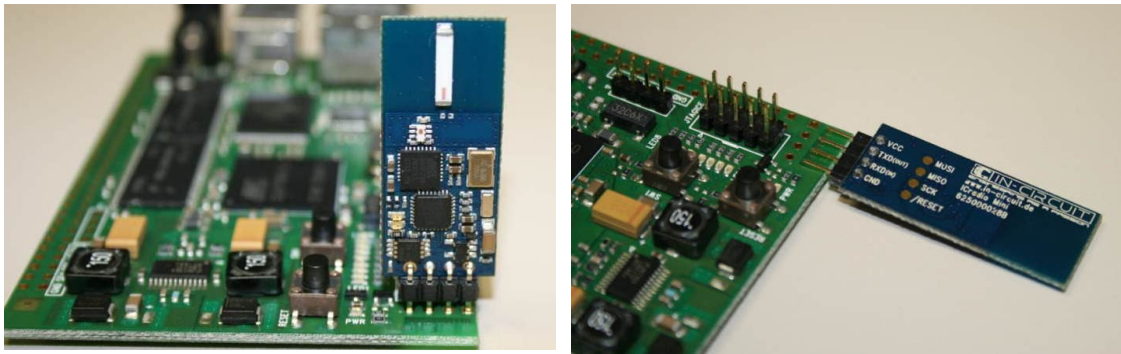


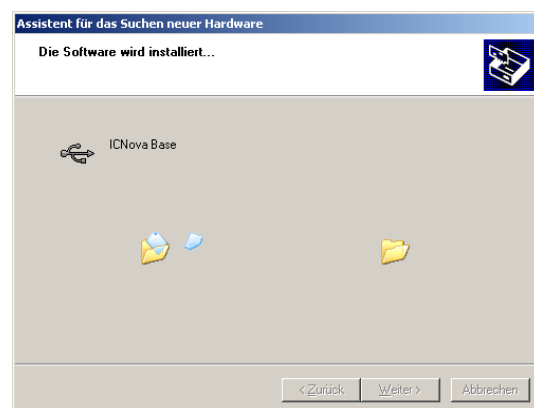
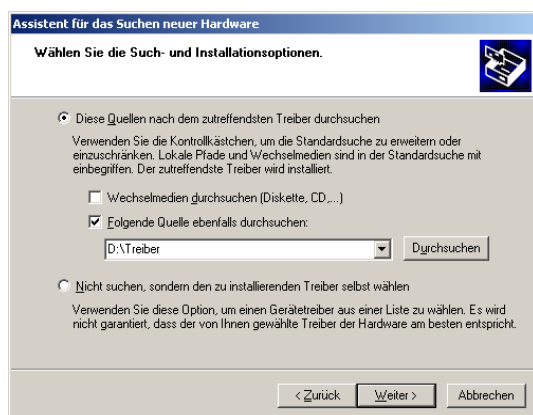
figure 3: ICnova wireless communication with ICradio Mini

1.3 Driver for Linux

Linux kernels ($\geq 2.6.12$) provides support for the ICnova Base. After connecting, the ICnova should be available via an USB-serial device. Depending on your specific configuration it should be called like `/dev/tty-USB0`.

1.4 Driver for Windows

When you connect the ICnova Base to Windows PC for the first time, the system should detect new hardware. After a short while, there should pop up the Windows hardware manager. After checking the box "Choose the software from a list" you can install the driver by clicking "Next".



In the following dialog the option "also search following directories" should be enabled and set to "D:\Treiber". Please substitute "D:" by the letter of your CD-ROM drive.

1.5 Network Settings

While booting the ICnova tries to get a network address via DHCP. The configuration can be viewed and set using the tool "ifconfig". Calling "ifconfig eth0" the settings for the network interface are shown. If there were no network address was found, because there is no DHCP-Server present, it can be set by typing "ifconfig eth0 <IP-address> up".

To set up the ICnova with a static IP on bootup, the file /etc/network/interfaces must be edited. To do this, use the command "vi /etc/network/interfaces". In this file the line "iface eth0 inet dhcp" must be substituted by the following block (adapted to your network environment):

```
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.254
```

Help for vi:

- *the insertion mode is entered by pressing "i"*
- *pressing <ESC> ends insertion mode*
- *to save your modifications and leave the insertion-mode and type ":wq"*

To supply the domain name resolution, the file "/etc/resolv.conf" has to be edited, too.

After starting "vi /etc/resolv.conf" change into the insertion-mode by pressing "i". Now add the following two lines (replace the values with those, suitable for your environment).

```
domain example.com
nameserver 192.168.0.254
```

End insertion mode with <ESC> and save your changes typing ":wq".

The text following the keyword domain will be interpreted as the domain name of your network. So, the computer named foo.example.com can be accessed by simply typing foo. The second line defines a name server, doing the domain name resolution in your network. If there is more than one server, you can insert multiple lines, using different IP addresses.

2 Usage

2.1 Shell Access

The easiest way to access the ICnova via a shell is using the USB port or the debug device on the ICnova. The connection can be provided using any terminal program, like Hyperterminal for Windows or minicom for Linux.

Furthermore the ICnova provides access via ethernet with an integrated telnet server. To use the connection must be established using a suitable program (e.g. „telnet“ for Linux or „putty“ for Windows).

The user name to log in is „default“. If there is a need for super user privileges, they can be granted using the command „su“. Afterwards the user is working as „root“. The password is empty in both cases. Now there is a fully functional shell available.

2.2 Switching LEDs

The ICnova provides 8 green and one red LED. The red LED is used to show the power status, but can also be used for other purposes. Toggling one of these LED can be done following these instructions:

- log in
- change the directory with `cd /sys/class/led`
- to view the available LEDs, use `ls`
- use `cd` to change into the directory of the LED
- The LED can be switched on using `echo 1 >> brightness`
- or switched off using `echo 0 >> brightness`

The LEDs can also be switched using the web interface. As soon as the network is set up correctly, this interface can be used via any browser. By typing the IP address of your ICnova into the address field of your browser you have access to the main page which leads you to the LED demo website. Now the LEDs can be switched by selecting the checkboxes and clicking „submit“.

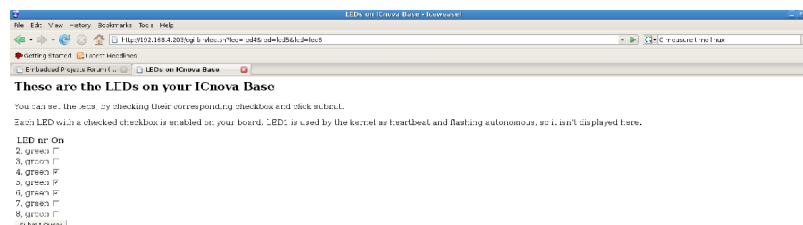


figure 4: controlling LEDs via web interface

2.3 Controlling GPIO-Ports

On the sides of the board almost all GPIO lines of the AP7000 are accessible. To control those via user space, the following instructions should be used.

All GPIOs are numbered consecutively

*e.g.: PA00 => 0, PA01 => 1, PA31 => 31, PB0 => 32, PE0 => 4*32 + 0 = 128.*

Change the directory with `cd /sys/class/gpio`.

Type `echo 1 > export` to create a new folder named gpio1.

Change into this new directory with `cd gpio1`.

Entering the commando `echo out > direction` configures GPIO pin 1 as an output.

`echo 1 > value` sets the GPIO pin to logical 1.

2.4 Writting your own programs

Since the ICnova provides a complete Linux, it can run programs like any PC. So programs can be written like for a PC.

3 Own developments for ICnova AP7000 Base

To change the software on the ICnova, a toolchain for the PC is needed. **It is strongly recommended to use Linux as OS, since only Linux supports all needed programs completely.** On the Support-CD you will find the complete toolchain to build a Linux Kernel 2.6.28 with the Buildroot 2009.02.

3.1 Premises

The most Linux distributions already contain the needed software. Commonly the software is installed already. Please check the state of the following packages using your favourite package manager (aptitude, synaptic, yum or yast). The exact names can vary from one distribution to another.

The following packages will be needed:

- gcc,
- make,
- automake,
- autoconf,
- libtool,
- flex,

- bison,
- texinfo
- header files of zlib (usually zlib-dev or zlib1g-dev)
- header files of lzo2 (usually liblzo2-dev)
- header files of ncurses (usually ncurses-dev)

Additionally the program `avr32program` is needed to program the ICnova using a JTAG-interface. You can download the tool on the Atmel website.

3.2 Selection of software

To start a new collection, copy the whole directory `"ICnova_base"` from CD to your hard drive. Now change into this directory and run the command `"make menuconfig"`. An interface is shown where you can choose the software and change the options for the cross toolchain to be used.

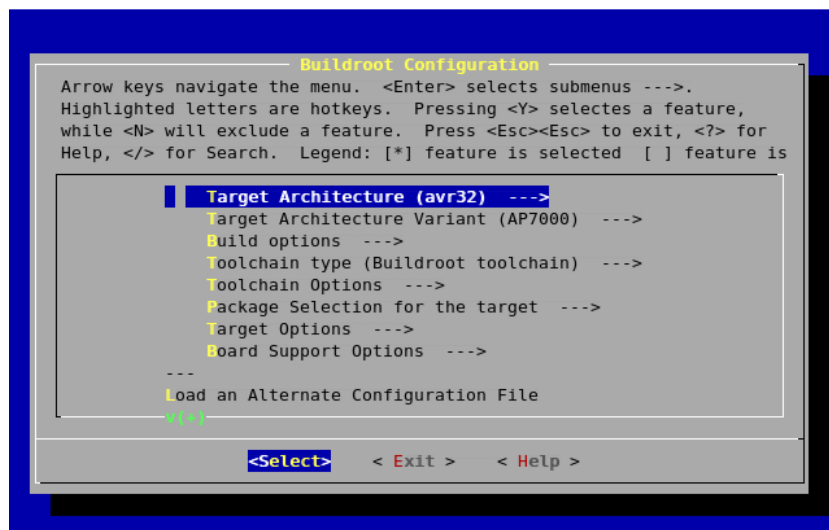


figure 5: *Menue Config*

Attention! Not all programs shown here, are already supported on the AVR32-architecture and thus not suitable for the ICnova. For more information take a look on www.avr32linux.org.

Choosing the menu item `"Package Selection for the target --->"` a list of available software will be shown. To install another part of software, move the cursor above the corresponding item and press `"<Enter>"`.

Please ensure that you build your toolchain with C++ support. To do so, you have to select `"Build/Install C++ Compiler and libstdc++"` under `"Toolchain options"`. To avoid trouble during compiling, don't enable the Java compiler.

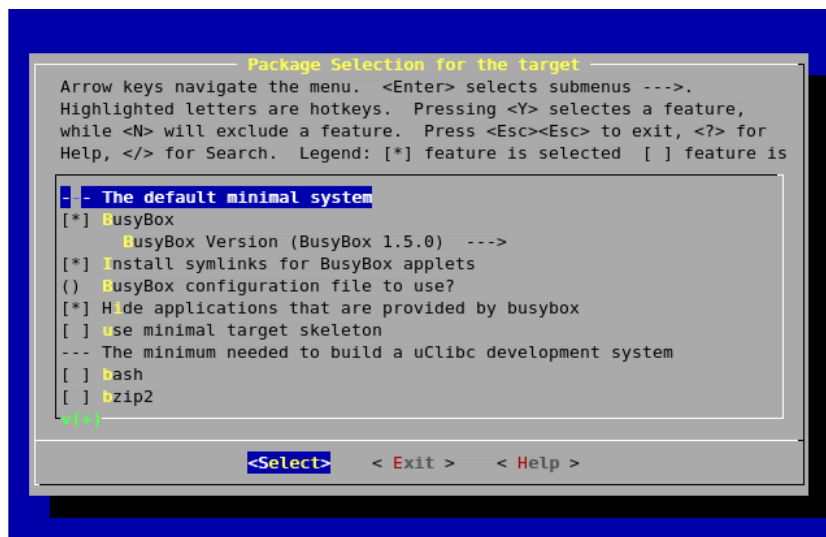


figure 6: Menue Config Pakage

After the selection the program can be closed using the "<Exit>" button. Being back on the command-line, the "make"-command will generate a cross toolchain with the selected software and a file system image used to program the ICnova. This image can be written via JTAG using the command

```
"avr32program program -v -e -f cfi@0 -F bin -O 0x00030000
binaries/base/rootfs.avr32.jffs2"
```

To restart the ICnova with the new image, press the reset button on the board afterwards. Alternatively the Image can be transmitted via ethernet using the UBOOT bootloader (see 3.5.2).

A major part of the functionality is provided by busybox.

To configure the busybox use "make busybox-menuconfig".

3.3 Toolchain

To build software for the ICnova, a suitable cross toolchain is needed. It's being created the first time the image is generated. It contains all the programs needed to create software for the ICnova.

After generation, the toolchain can be found in the directory "build_avr32/staging_dir/bin". You can use it to create own programs, too. If you move the compilation of your program to

"project_build_avr32/base/root" it will also be merged into the next image you create and so it will be available on the ICnova.

3.4 The file system of the ICnova

The ICnova provides 8MByte of flash memory. Since the files are packed before being saved in the flash, more space is available. If a file is needed only once, it could be saved in the directory /tmp. All files shown here aren't saved in Flash but in the systems RAM. Within this directory 32MB space are available. Just after a reboot, all files saved here are lost. If you have an own program, it should be saved in the directory `project_build_avr32/base/root/usr/bin`. This way the system will automatically find it and which makes it available all over the system. To provide libraries, put them in `project_build_avr32/base/root/usr/lib`.

3.5 Transferring data to the ICnova

3.5.1 Single files

The easiest way to transfer data to the ICnova is to make the files accessible via network. You can do this with the following protocols:

- a) HTTP/FTP: To transfer a file fast and easy to the ICnova, even from the internet, the HTTP and the FTP protocol are appropriated. They are common and easy to use.

To download a file, it should be made available on an arbitrary Computer and the correct URL should be known. After you have logged in on the ICnova type the following command:

```
"wget <URL>", e.g.
```

```
" wget http://192.168.4.200/public/demo "
```

Now this file is going to be downloaded into the current working directory.

- b) NFS: To get access to files via NFS, the portmap-daemon is needed on the ICnova. It can be started using the command "portmap". On the host computer you have to share your files by adding a line into `/etc/exports` like the this "`/path/to/the/files *(rw, sync, subtree_check)`"

Now you have to restart your NFS server should be restarted on your host system. On a Debian or Ubuntu based system this can be done by using "`sudo /etc/init.d/nfs-kernel-server restart`".

You get access to this directory on the ICnova using the following command line:

```
"mount -t nfs <your-NFS-Server-IP>:/path/to/files /mnt"
```

To store this file permanently into flash memory use "`cp /mnt/<source file> <dest. file>`".

3.5.2 Transfer an entire image via JTAG

To transfer an entire image to the ICnova, a JTAG adaptor, e.g. the JTAG-ICE MKII, could be used.

To download the image the program `avr32program` can be used, which can be downloaded on the Atmel website.

The flash memory is compatible with the CFI standard and is mapped on address 0. The image for the bootloader should be written to address 0 and the image for the filesystem on address 0x30000.

This way, one of the following commandlines should be used:

```
avr32program program -v -e -f cfi@0 binaries/base/u-boot.bin
```

or

```
avr32program program -v -e -f cfi@0 -F bin -O 0x00030000
```

```
binaries/base/rootfs.avr32.jffs2 (command needs to be in a single line)
```

3.5.3 Transfer an entire image over network

As alternative to flashing the image with a JTAG, you can download the image via network. By default only `mtdblock2` can be written to prevent an accidentally overwriting of the bootloader.

Option A:

The safer way to flash a new image into the ICnova is to boot the system over NFS. To do so, you have to adapt the bootloader as described below.

After you made your image accessible over NFS as described earlier, you can change your bootloader by pressing `<SPACE>` during the bootup sequence.

To list all environment variables use the command `"printenv"`. It is recommended to write the default settings down, so you can restore the settings if needed.

By using `"setenv"` you can create, with `"askenv"` you can modify and with `"saveenv"` you can save all environment variables permanently in the flash.

To boot via NFS set the environment variables as following:

```
"setenv ipaddr <IP-Adresse-ICnova> „  
e.g.: „setenv ipaddr 192.168.4.250“  
  
„setenv serverip <IP-Adresse-Server>“  
e.g.: „setenv serverip 192.168.4.18“  
  
"setenv bootcmd nfs 10400000 $(serverip):<Pfad zum Kernelimage>;bootm"  
z.B.: "setenv bootcmd nfs 10400000 $(serverip):/icnova/boot/uImage;bootm"
```

If you are using a static IP address set the boot arguments.

```
"setenv bootargs nfsroot=$(serverip):/<NFS-pfad>/ip=$(ipaddr):$(  
(serverip)::255.255.255.0::eth0:none"  
(command needs to be in a single line)  
e.g.: "setenv bootargs nfsroot=$(serverip):/icnova/ ip=$(ipaddr):$(  
(serverip)::255.255.255.0::eth0:none"
```

To ensure that the bootloader settings doesn't get overwritten by the new target system you should comment out the line "auto eth0" in the file "ICnova/project_build_avr32/base/root/etc/network/interfaces" before you compile your new kernel.

If you want to use DHCP your settings should look like this.

```
"setenv bootargs nfsroot=$(serverip):/<NFS path>/ ::$(server  
IP)::255.255.255.0::eth0:dhcp" (command needs to be in a single line)
```

With the command "boot" you can reboot your ICnova with the new settings.

Now you can copy the image by using:

```
"dd if=<Pfad zum Image> of=/dev/mtdblock2 bs=65535"
```

mtddblock0 contains the bootloader, mtdblock1 contains the environment of the bootloader and mtdblock2 contains the root file system.

Option B:

Another option is to copy the new image into the flash during runtime.

Caution: *It is possible to corrupt the whole system when flashing the new image. If that happens, the system can only be restored by using the JTAG interface.*

To flash an image, start the ICnova and boot up linux. Now log in and become root by typing "su". Since we want to reprogram the root filesystem, we have to make the running file system *read only* by using the command "mount -o remount,ro /". This way no file can be written during the upgrade procedure and the possibility of a corrupted file system is way lower.

If the image isn't accessible via NFS, you can copy the image as described earlier with the command "wget" to the directory /tmp on the ICnova. Files in this directory are only stored in RAM and therefore you can copy data larger than 8MByte.

After making the image available, the command

```
"dd if=<Pfad zum Image> of=/dev/mtdblock2 bs=65535"
```

can be used to copy it into the flash.

The file /dev/mtdblock2 contains the filesystem, /dev/mtdblock0 represents the bootloader and /dev/mtdblock1 the environment of the bootloader.