Tutorial

„ICnova ADB1000“

Content

1 INTRODUCTION 3

1.1 Attaching a TFT touch screen (optional) 4

1.2 Accessing the Linux shell 6

2 FIRST STEPS WITH LINUX 8

3 ACCESS THE ICNOVA ADB1000 VIA ETHERNET 10

4 TOOLCHAIN - DEVELOPING OWN APPLICATIONS 12

4.1 Selection of software 13

4.2 Toolchain 14

4.3 The file system of ICnova AP7000 OEM(plus) 15

4.4 Transferring data to the ICnova AP7000 OEM(plus) 16

4.4.1 Single files 16

4.4.2 Transferring an entire image via JTAG 17

4.4.3 Transferring an entire image over network 17

Revision overview

Version	Date	Reason for changes	Editor
A	19.04.09	first edition	Ullrich
B	30.04.09	Added commands for OpenSuse user	Ullrich

1 Introduction

Thank you for purchasing the ICnova ADB1000.

The ICnova ADB1000 is an evaluation board for ICnova AP7000 OEM and ICnova AP7000 *OEMplus* modules with the multimedia processor AP7000 from Atmel. The ICnova AP7000 OEM and the ICnova AP7000 OEM*plus* modules are delivered with the preinstalled Linux kernel 2.6.28.

To control the system an USB mouse and a USB keyboard can be hooked up when using the board with an ICnova AP7000 OEM*plus* module.

To begin you need the following items

- an ICnova ADB1000,
- an ICnova AP7000 OEM or an ICnova AP7000 OEM*plus*
- a power adaptor with 2.1mm plug (10-36VDC)
- USB-UART-Bridge or a level transceiver for the serial interface e.g. MAX232
- an Ethernet patch cable

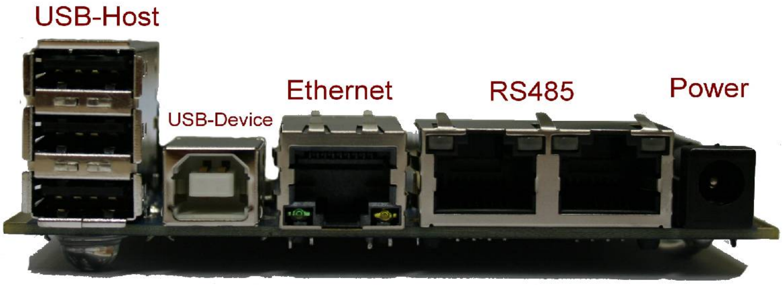


Figure 1: *Connectiors of ICnova ADB1000*

1.1 Attaching a TFT touch screen (optional)

The ICnova ADB1000 provides 2 different connectors for TFT displays.

- 1 x 18-Bit parallel RGB interface
  0.50mm Pitch Easy-On™ FFC/FPC Connector from Molex (Part. No. 5019515000)
  - compatible with TFT touch display ET035009DH6

- 1 x 24-Bit parallel RGB interface
  0.50mm Pitch Easy-On™ FFC/FPC Connector from Molex (Part. No. 5019515000)
  - compatible with TFT touch display ET050000DH6

By default, the ICnova AP7000 OEM(*plus)* is configured for a 3,5” TFT display with the 18-Bit RGB interface.

Note : *The two display sockets are connected in parallel, so only one can be used for attaching a display at a time.*

To attach the display, the connector has to be opened by tilting it horizontally to the board. After inserting the FFC cable of the display, the connector has to be tilted back to ensure a proper connection.

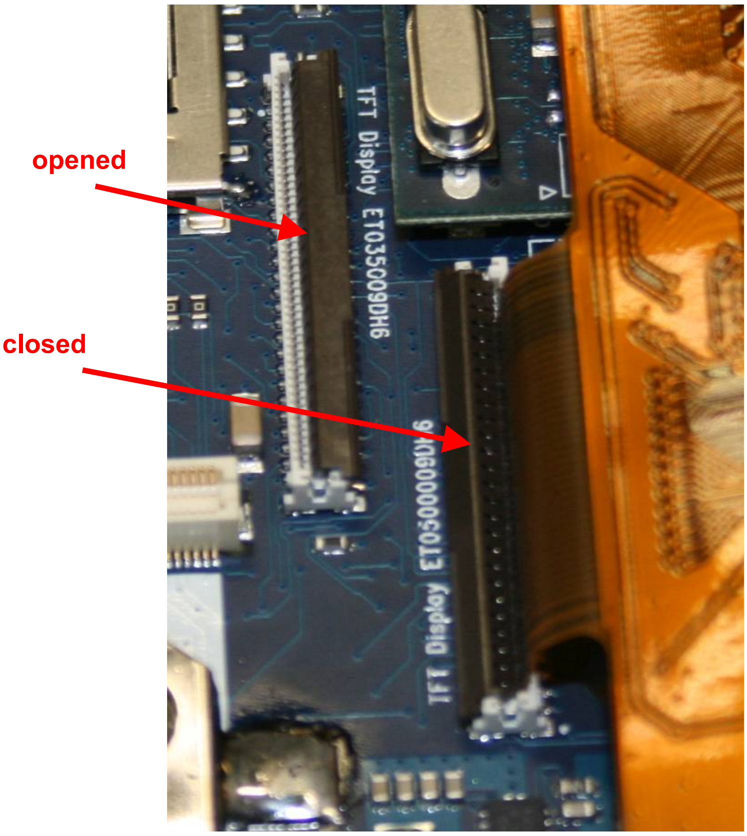


Figure 2: *opened and closed TFT connector*

Note: *When connecting the 5” TFT with the 24-Bit RGB interface, the display needs to be inserted left-aligned as shown below.*

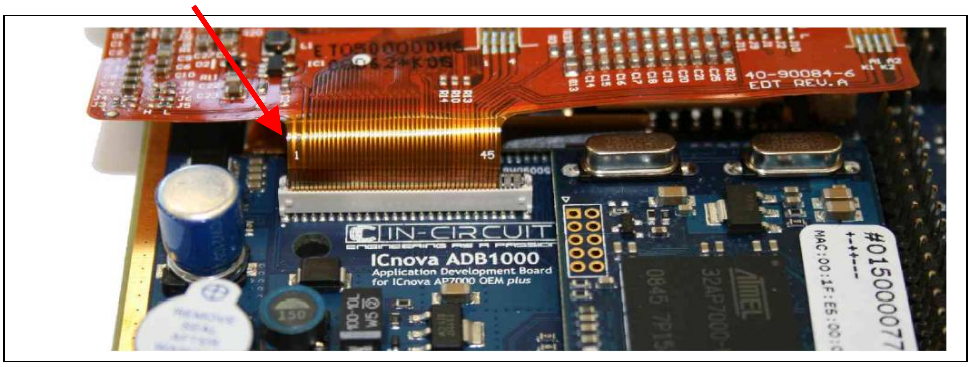


Figure 3: *display inserted left-aligned*

1.2 Accessing the Linux shell

To get access to the Linux shell the UART0 interface of the ICnova ADB1000 has to be connected to your PC. The signal level of the RX- and TX pin is 3.3V. For a proper connection between the UART and the serial port of the PC you need a level transceiver (e.g. MAX232).

Another option is to use a UART-USB transceiver. We recommend the USB-UART-Bridge of the In-Circuit GmbH, which is included in the ICnova ADB Starterkit. A detailed how-to-document for installing the drivers for CP2102 UART-USB transceivers can be found here.

After connecting the PC to the ICnova ADB1000, open a terminal program of your choice e.g. Hyperterminal or MiniCom with the following configuration:

- baud rate 115200
- parity: none
- data bits: 8
- stop bits: 1
- no flow control

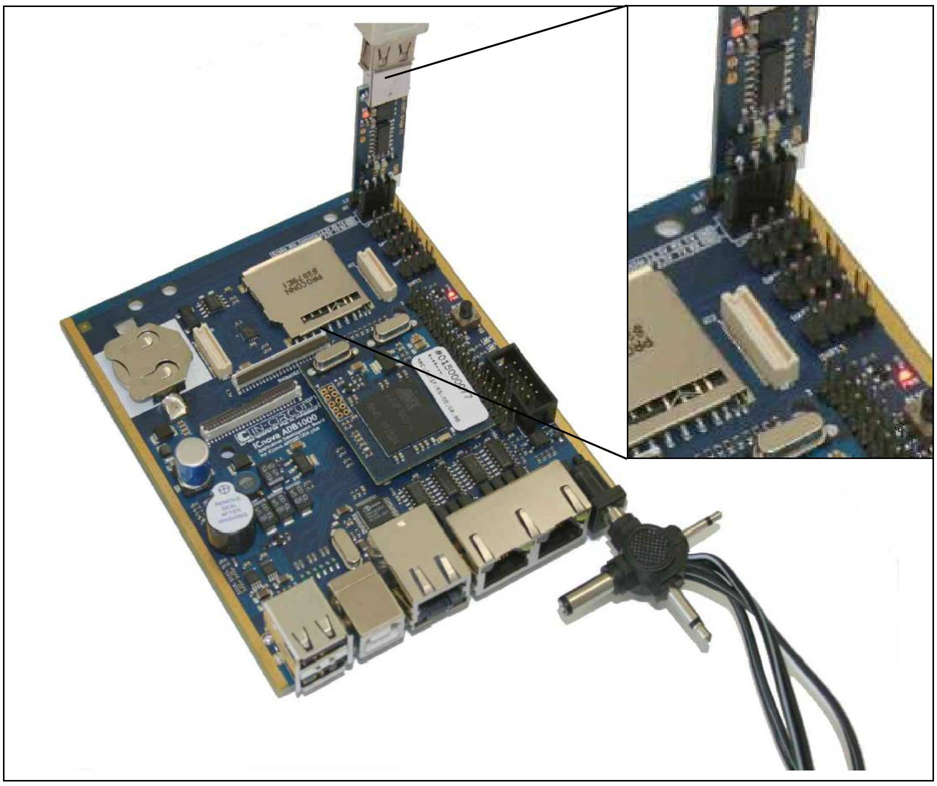
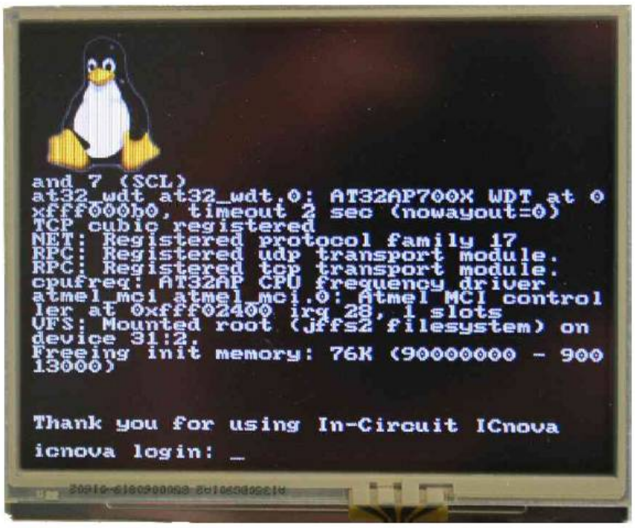


Figure 4: *Starting the ICnova ADB1000*

Please double check that the positive pole of the power adapter is connected to the inner pin and power the board.

The flashing red LED indicates that the ICnova ADB1000 is correctly powered.

The boot-up messages are shown in your terminal program and on the display, if attached (see figure 5).

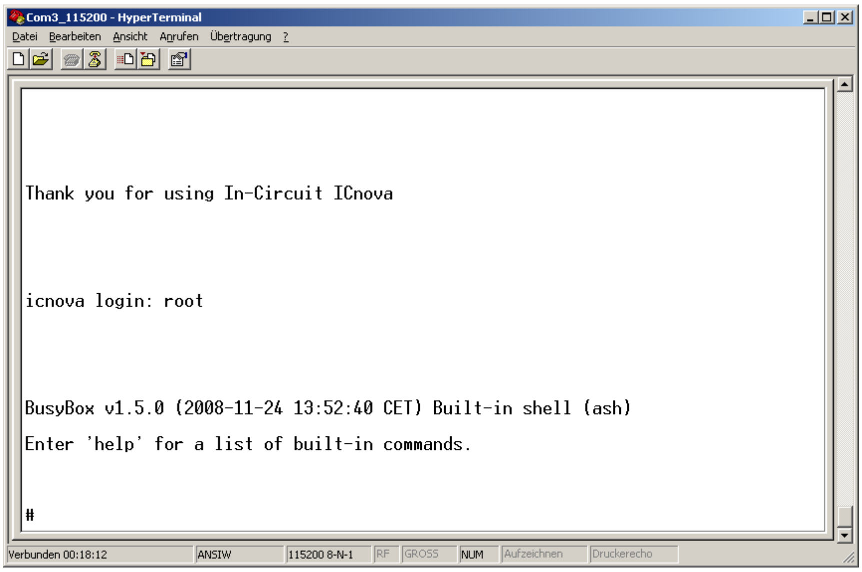


```
and 7 (SCL)
at32_wdt at32_wdt.0: AT32AP700X WDT at 0
xfff000b0, timeout 2 sec (nowayout=0)
TCP cubic registered
NET: Registered protocol family 17
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
cpufreq: AT32AP CPU frequency driver
atmel_mci atmel_mci.0: Atmel MCI control
ler at 0xfff02400 irq 28, 1 slots
VFS: Mounted root (jffs2 filesystem) on
device 31:2.
Freeing init memory: 76K (90000000 - 900
13000)

Thank you for using In-Circuit ICnova
icnova login: _
```

Figure 5: *Log in screen*

2 First steps with Linux

After booting the Linux system you can log in as „default“ or as „root“.
Logging in as „root“ grants access to all features of the system.


Com3_115200 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
Verbunden 00:18:12 ANSIW 115200 8-N-1 RF GROSS NUM Aufzeichnen Druckerecho

```
Thank you for using In-Circuit ICnova

icnova login: root

BusyBox v1.5.0 (2008-11-24 13:52:40 CET) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
```

Figure 6: *Login mask*

Now, you are in the so-called BusyBox. The kernel doesn't support the entire command set of the BusyBox. The command „`help`“ gives an overview of the supported commands.

As an example, for controlling a LED you have to do the following steps.

Note: *All GPIOs are numbered consecutively. So each alphabetically counted port has an offset of 32 e.g.: PA00 => 0, PA01 => 1, PA31 => 31, PB0 => 32, PE0 => 4*32 + 0 = 128.*
LED1 is connected to PE16 which results to GPIO pin 144 (see schematic).

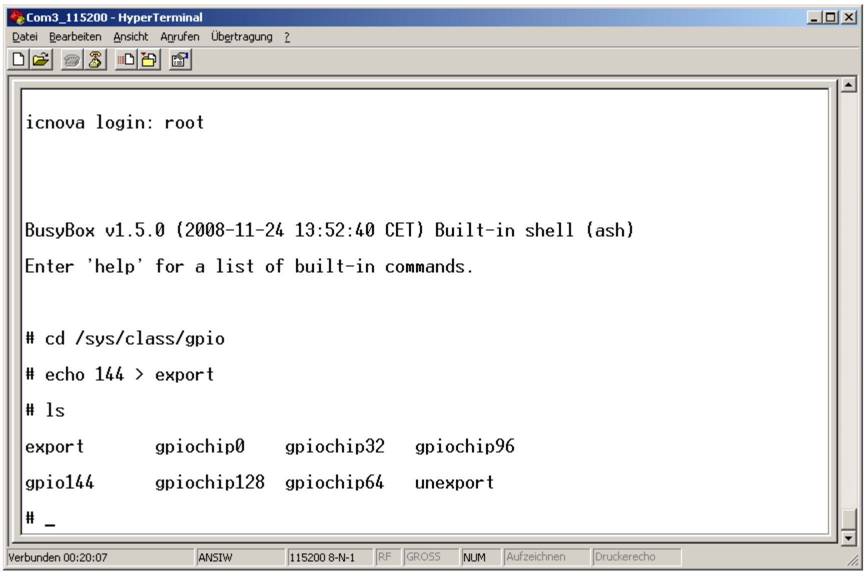
Change the directory with

```
cd /sys/class/gpio
```

By typing

```
echo 144 > export
```

a new folder named gpio144 is created.


Com3_115200 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
Verbunden 00:20:07
ANSIW
115200 8-N-1
RF
GROSS
NUM
Aufzeichnen
Druckerecho

```
icnova login: root


BusyBox v1.5.0 (2008-11-24 13:52:40 CET) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# cd /sys/class/gpio
# echo 144 > export
# ls
export       gpiochip0    gpiochip32   gpiochip96
gpio144      gpiochip128  gpiochip64   unexport
# _
```

Figure 7: *Creating a new directory*

Change into the new folder with

```
cd gpio144
```

Typing

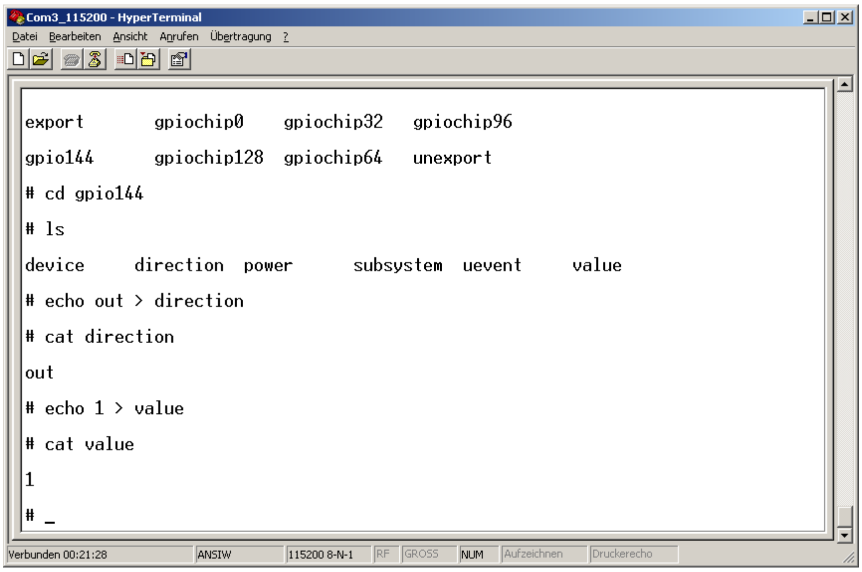
```
echo out > direction
```

defines GPIO pin 144 as an output.

The command

```
echo 1 > value
```

switches on LED1.


Com3_115200 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
Verbunden 00:21:28
ANSIW
115200 8-N-1
RF
GROSS
NUM
Aufzeichnen
Druckerecho

```
export       gpiochip0    gpiochip32   gpiochip96
gpio144      gpiochip128  gpiochip64   unexport
# cd gpio144
# ls
device     direction  power      subsystem  uevent     value
# echo out > direction
# cat direction
out
# echo 1 > value
# cat value
1
# _
```

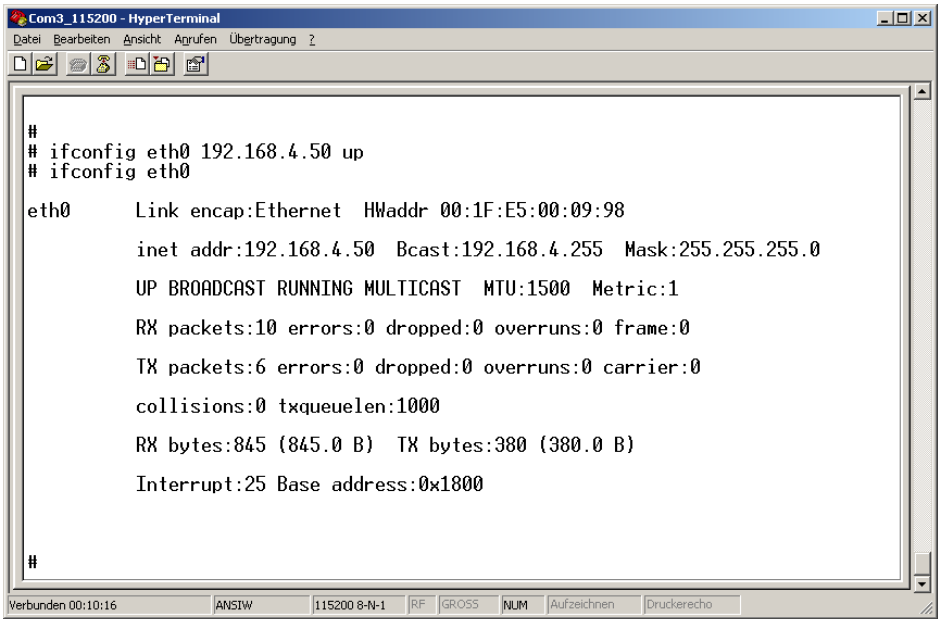
Figure 8: *Set GPIO parameters*

3 Access the ICnova ADB1000 via Ethernet

By default the ICnova gets its network configurations via DHCP on start-up. After starting the ICnova the network services HTTP and Telnet are available. More services can be easily added as desired.

The network configurations can be edited using the program ”ifconfig“.

The command ”`ifconfig eth0`“ shows the current settings including the IP address. If the automatic configuration has failed, because there was now DHCP available, the IP address can be set temporarily with the command ”`ifconfig eth0 <IP address> up`“.

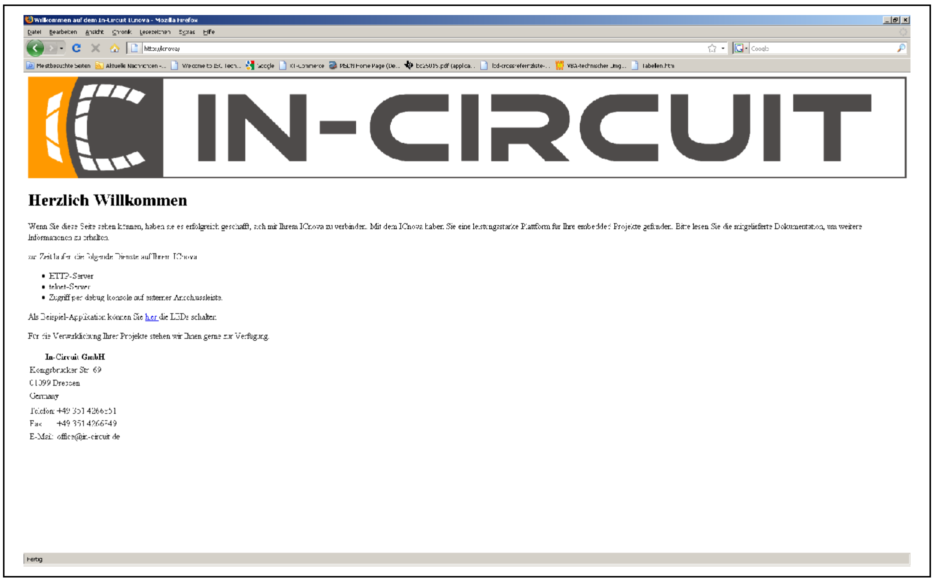

Com3_115200 - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
Verbunden 00:10:16
ANSIW
115200 8-N-1
RF
GROSS
NUM
Aufzeichnen
Druckerecho

```
#
# ifconfig eth0 192.168.4.50 up
# ifconfig eth0

eth0      Link encap:Ethernet  HWaddr 00:1F:E5:00:09:98

          inet addr:192.168.4.50  Bcast:192.168.4.255  Mask:255.255.255.0

          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

          RX packets:10 errors:0 dropped:0 overruns:0 frame:0

          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0

          collisions:0 txqueuelen:1000

          RX bytes:845 (845.0 B)  TX bytes:380 (380.0 B)

          Interrupt:25 Base address:0x1800


#
```

Figure 9: *Setting the IP address manually*

After setting the IP address, you have access to the preinstalled HTTP server.

Enter the IP address of your ICnova ADB1000 into the address line of your web browser and this screen appears.


Willkommen auf dem In-Circuit ICnova - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://icnova/
IN-CIRCUIT
Fertig

Herzlich Willkommen

Wenn Sie diese Seite sehen können, haben sie es erfolgreich geschafft, sich mit Ihrem ICnova zu verbinden. Mit dem ICnova haben Sie eine leistungsstarke Plattform für Ihre embedded Projekte gefunden. Bitte lesen Sie die mitgelieferte Dokumentation, um weitere Informationen zu erhalten.

zur Zeit laufen die folgende Dienste auf Ihrem ICnova

- HTTP-Server
- telnet-Server
- Zugriff per debug konsole auf externer Anschlussleiste.

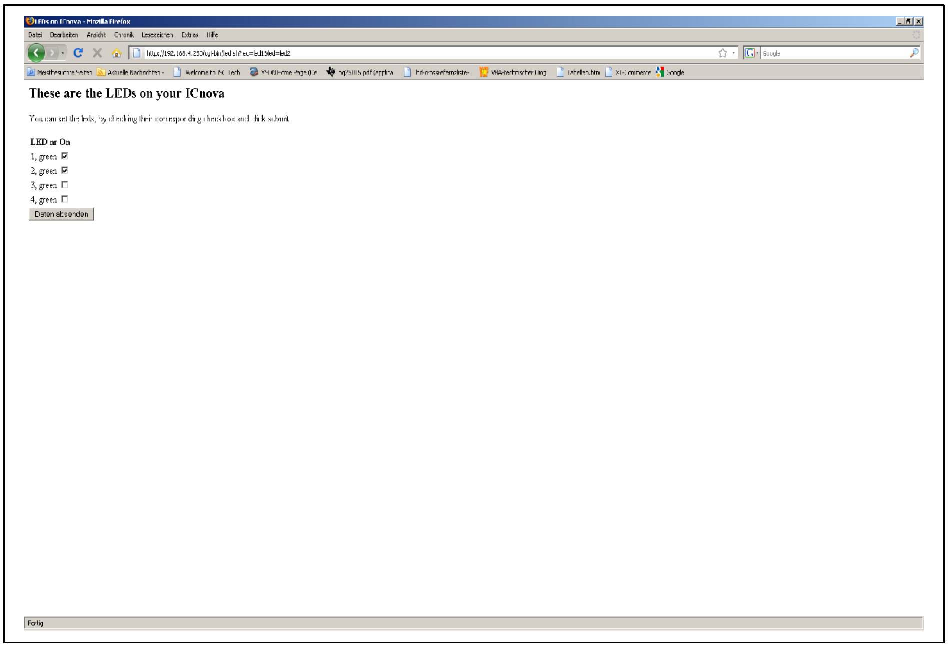
Als Beispiel-Applikation können Sie hier die LEDs schalten.

Für die Verwirklichung Ihrer Projekte stehen wir Ihnen gerne zur Verfügung.

In-Circuit GmbH
Königsbrücker Str. 69
01099 Dresden
Germany
Telefon: +49 351 4266251
Fax +49 351 4266549
E-Mail: office@in-circuit.de

Figure 10: *HTTP web server*

The link leads to a website providing control over the LEDs of the ICnova ADB1000. Clicking the submit button after the corresponding check boxes have been selected, changes the state of the LEDs.


LEDs on ICnova - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://192.168.4.250/cgi-bin/led.sh?led=led1&led=led2
Daten absenden
Fertig

These are the LEDs on your ICnova

You can set the leds, by checking their corresponding checkbox and click submit.

LED nr On
1, green ☑
2, green ☑
3, green ☐
4, green ☐

Figure 11: *LED demo website*

4 Toolchain - developing own applications

The entire toolchain to build the Linux kernel version 2.6.28 (buildroot 2009.02) can be found on the support CD. The GCC compiler 4.2.2 is also located on the CD. Copy the entire ICnova directory from the CD to your PC using "`cp -a /media/cdrom/ICnova .`". Don't use drag and drop for copying the files.

Adapting software for ICnova AP7000 OEM(*plus*), requires an applicable toolchain for your PC.
It's strongly recommended to use a Linux OS, since it fully supports the provided software.
The required software packages are already included in most of the Linux distributions. Please check with your package manager (aptitude, synaptic, yum, yast) if the following packages are installed. The exact names may vary from one distribution to another.
The required packages are:

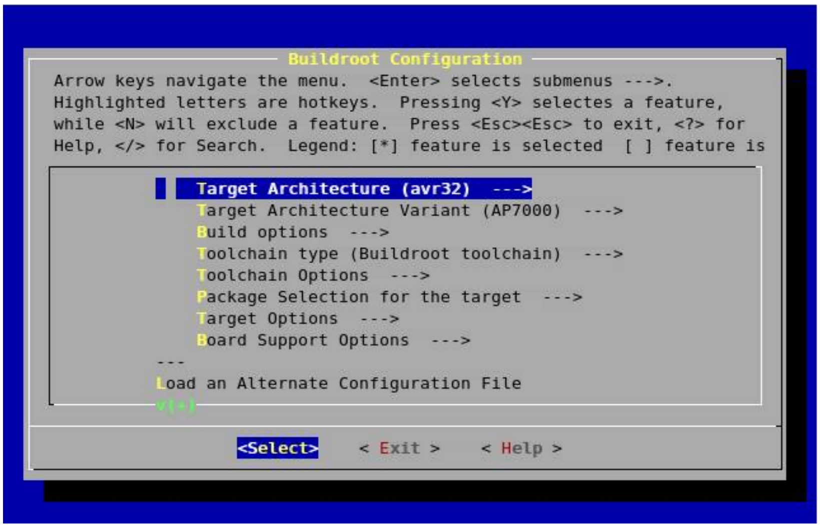
- gcc,
- make,
- automake,
- autoconf,
- libtool,
- flex,
- bison,
- texinfo
- header files for zlib, (usually zlib-dev or zlib1g-dev)
- header files for lzo2 (usually liblzo2-dev)
- header files for ncurses (usually ncurses-dev)

Additionally, the program "avr32program" is needed for writing data to your ICnova via a JTAG interface. This tool can be downloaded from the Atmel website.

4.1 Selection of software

After copying the folder ”ICnova“ to your hard drive call ”`make menuconfig`“ within this directory.

Now, the software can be selected which should be included in your toolchain.

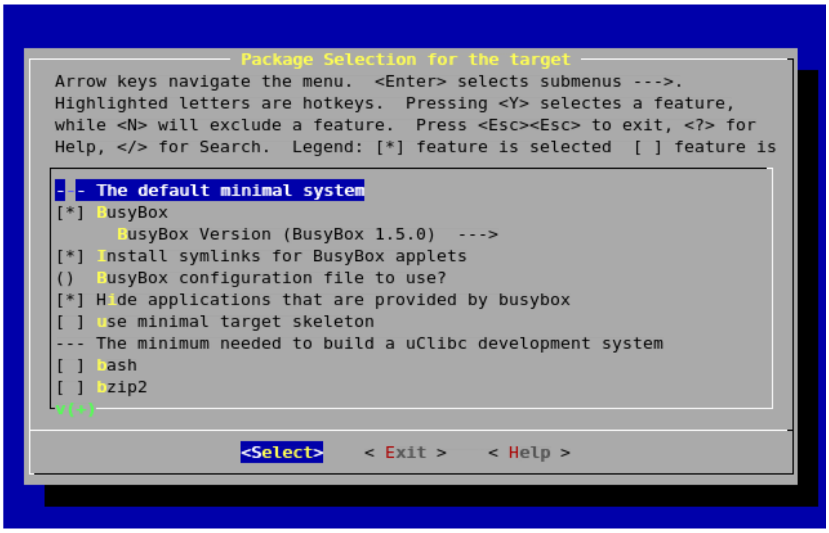


```
                       Buildroot Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus --->.
 Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
 while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
 Help, </> for Search.  Legend: [*] feature is selected  [ ] feature is

            Target Architecture (avr32)  --->
            Target Architecture Variant (AP7000)  --->
            Build options  --->
            Toolchain type (Buildroot toolchain)  --->
            Toolchain Options  --->
            Package Selection for the target  --->
            Target Options  --->
            Board Support Options  --->
        ---
        Load an Alternate Configuration File
        v(+)

                 <Select>    < Exit >    < Help >
```

Figure 12: *Menu Config*

Caution! *Not all options listed there, are supported for the AP7000 and therefore not executable on the ICnova AP7000 OEM(plus)! For more information on which programs are supported and how to port programs for the AP7000 visit www.avr32linux.org.*

Choosing ”Package Selection for the target --->“ shows a list with all programs that are provided. For installing a program select it by using the cursor and hit <Enter>. Do not use the option “`minimal target skeleton`“



```
                    Package Selection for the target
 Arrow keys navigate the menu.  <Enter> selects submenus --->.
 Highlighted letters are hotkeys.  Pressing <Y> selectes a feature,
 while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
 Help, </> for Search.  Legend: [*] feature is selected  [ ] feature is

 --- The default minimal system
 [*] BusyBox
       BusyBox Version (BusyBox 1.5.0)  --->
 [*] Install symlinks for BusyBox applets
 ()  BusyBox configuration file to use?
 [*] Hide applications that are provided by busybox
 [ ] use minimal target skeleton
 --- The minimum needed to build a uClibc development system
 [ ] bash
 [ ] bzip2
 v(+)

                 <Select>    < Exit >    < Help >
```

Figure 13: *Menu config packages*

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". Please don't enable the Java compiler to avoid trouble during compiling.

After selection, the program can be closed using the ”`<Exit>`“ button. Being back in the command line, the ”`make`“-command generates 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/oem/rootfs.avr32.jffs2`” *(command needs to be in a single line)*

or respectively

“`avr32program program -v -e -f cfi@0 -F bin -O 0x00030000 binaries/oemplus/rootfs.avr32.jffs2`” *(command needs to be in a single line)*

After transfering the new image, restart the ICnova by pressing the reset button on the board.
Alternatively the Image can be transmitted over Ethernet using the UBOOT bootloader (see 4.4.3).
A major part of the functionality is provided by BusyBox. To configure the BusyBox use ”`make busybox-menuconfig`“ before building the file system image.

4.2 Toolchain

Building software for the ICnova, requires a suitable cross toolchain. 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/oem/root`“ or respectively ”`project_build_avr32/oemplus/root`“ it will also be merged into the next image you create and so it will be available on the ICnova.

4.3 The file system of ICnova AP7000 OEM(*plus*)

If a file is needed only once, it can be saved in the directory /tmp. All files in this directory are saved only in RAM and are not available after a reboot. Since the ICnova modules provide more RAM than flash memory it is possible to store large files in RAM.

If you have an own program, it should be saved in the directory

```
project_build_avr32/oem/root/usr/bin
```

or respectively

```
project_build_avr32/oemplus/root/usr/bin
```

to make them accessible from every location in the system.

Software libraries should be stored in

```
project_build_avr32/oem/usr/lib
```

or respectively in

```
project_build_avr32/oemplus/usr/lib.
```

4.4 Transferring data to the ICnova AP7000 OEM(*plus*)

4.4.1 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: Transferring 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.
Before downloading a file, it should be made available on an arbitrary computer and the correct URL should be known. After logging 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 share your files by adding a line into /etc/exports like this “/path/to/the/files *(rw,sync,subtree_check)”
Restart your NFS server 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“.
OpenSuse users use

```
"/etc/init.d/nfs-kernel-server stop"
```

to stop the NFS server. If an image is already mounted, unmount it with

```
"umount /shared_folder"
```

Mount the new image using

```
"mount -o loop .../binaries/oemplus/rootfs.avr32.ext2 /shared_folder"
```

and restart your NFS server using the command

```
"/etc/init.d/nfs-kernel-server start".
```

Get access to this directory on the ICnova by the following command line:
“mount -t nfs <your-NFS-Server-IP>:/path/to/files /mnt”
Using
“cp /mnt/<source file> <dest. file>”.

stores this file permanently into flash memory.

4.4.2 Transferring an entire image via JTAG

For transferring an entire image to the ICnova AP7000 OEM(*plus*) a JTAG adaptor, e.g. the JTAG-ICE MKII, can be used. To write the image via JTAG, the program avr32program is needed, which can be downloaded from the Atmel website.

The flash memory is compatible to the CFI standard and is mapped to address 0. The image for the boot-loader should be written to address 0 and the image for the file system on address 0x30000.

This way, one of the following command lines should be used:

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

and

```
avr32program program -v -e -f cfi@0 –F bin -O 0x00030000
binaries/oem/rootfs.avr32.jffs2 (command has to be in a single line)
```

Important remark for ICnova AP7000 OEM*plus* modules:

Since JTAG does not support NAND flash, the root file system can't be downloaded via JTAG.

The bootloader of the ICnova AP7000 OEM*plus* is stored in the NOR flash which is also not supported.

Therefore, the only option changing the bootloader is downloading it with a programmer like the USBprog.

4.4.3 Transferring an entire image over network

If you don't want to flash the image via JTAG, the image could be downloaded over the network. By default, only mtdblock2 can be written preventing an accidentally overwriting of the bootloader section.

Option A:

A save way flashing a new image to the ICnova is to boot the system over NFS.

Only a new created ext2-image can be made bootable via NFS.

The image can be found under

```
binaries/oem/rootfs.avr32.ext2
```

or respectively

```
binaries/oemplus/rootfs.avr32.ext2
```

To boot over NFS, adapt the bootloader as described below.

After making your image accessible over NFS as described earlier, the bootloader can be changed by pressing <SPACE> during the boot-up sequence.

A list of all environment variables is displayed using the command "`printenv`". It is recommended to make notes of the default settings allowing a restore of the settings if needed.

Using "`setenv`" creates, “`askenv`" modifies and “`saveenv`" saves all environment variables permanently in the flash memory.

For booting via NFS set the environment variables as follows:

```
“setenv ipaddr <IP-Address-of-ICnova>„
e.g. „setenv ipaddr 192.168.4.250“

„setenv serverip <IP-address-of-server>”
e.g.: “setenv serverip 192.168.4.18”

“setenv bootcmd nfs 10400000 $(serverip):<path to kernel image>;bootm”
e.g.: “setenv bootcmd nfs 10400000 $(serverip):/icnova/boot/uImage;bootm”
```

If using a static IP address set the boot arguments as shown.

```
"setenv bootargs nfsroot=$(serverip):/icnova"
"setenv bootargs $(bootargs) ip=$(ipaddr):$(serverip)::::eth0:none"
```

(*command needs to be in a single line*)

```
e.g.:"setenv bootargs nfsroot=$(serverip):/icnova"
"setenv bootargs $(bootargs)
ip=$(ipaddr):$(serverip)::255.255.255.0::eth0:none"
```

To ensure the settings made in the bootloader doesn't get overwritten by the target system during booting, comment the line "auto eth0" in the file `icnova/project_build_avr32/oem(plus)/root/etc/network/interfaces` out before recompiling the system.

If DHCP is used the settings should look like this.

```
"setenv bootargs nfsroot=$(serverip):/icnova"
"setenv bootargs $(bootargs) ip=dhcp"
```

The command “`boot`“ reboots the ICnova with the new settings.

After rebooting, copy the image by typing:

```
“dd if=<path to the image> of=/dev/mtdblock2 bs=65535”
```

On ICnova OEM modules the mtdblock0 contains the bootloader, mtdblock1 contains the environment of the bootloader and mtdblock2 contains the root file system.

Note: *The root file system on the AP7000 OEMplus is located in mtdblock1. Erase this block before copying the new image.*

```
“flash_eraseall /dev/mtd1”
“dd if=<path to the image> of=/dev/mtdblock1 bs=131072”
```

Option B:

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

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

To flash an image, start the ICnova and boot up Linux. Then log in and become root by typing ”`su`“. Since we want to reprogram the root file system, 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, the image can be copied to the directory /tmp on the ICnova as described above using the command “`wget`”.

After making the image available, the command

“`dd if=<path to the image> of=/dev/mtdblock2 bs=65535`”

or respectively

“`dd if=<path to the image> of=/dev/mtdblock2 bs=131072`”

copies it into the flash.

Note: *When updating the image in the ICnova AP7000 OEMplus this sequence has to be followed:*

```
flash_eraseall /dev/mtd0
flash_eraseall /dev/mtd1
dd if=rootfs.avr32.jffs2 of=/dev/mtdblock1 bs=131072
dd if=uImage of=/dev/mtdblock0 bs=131072
```