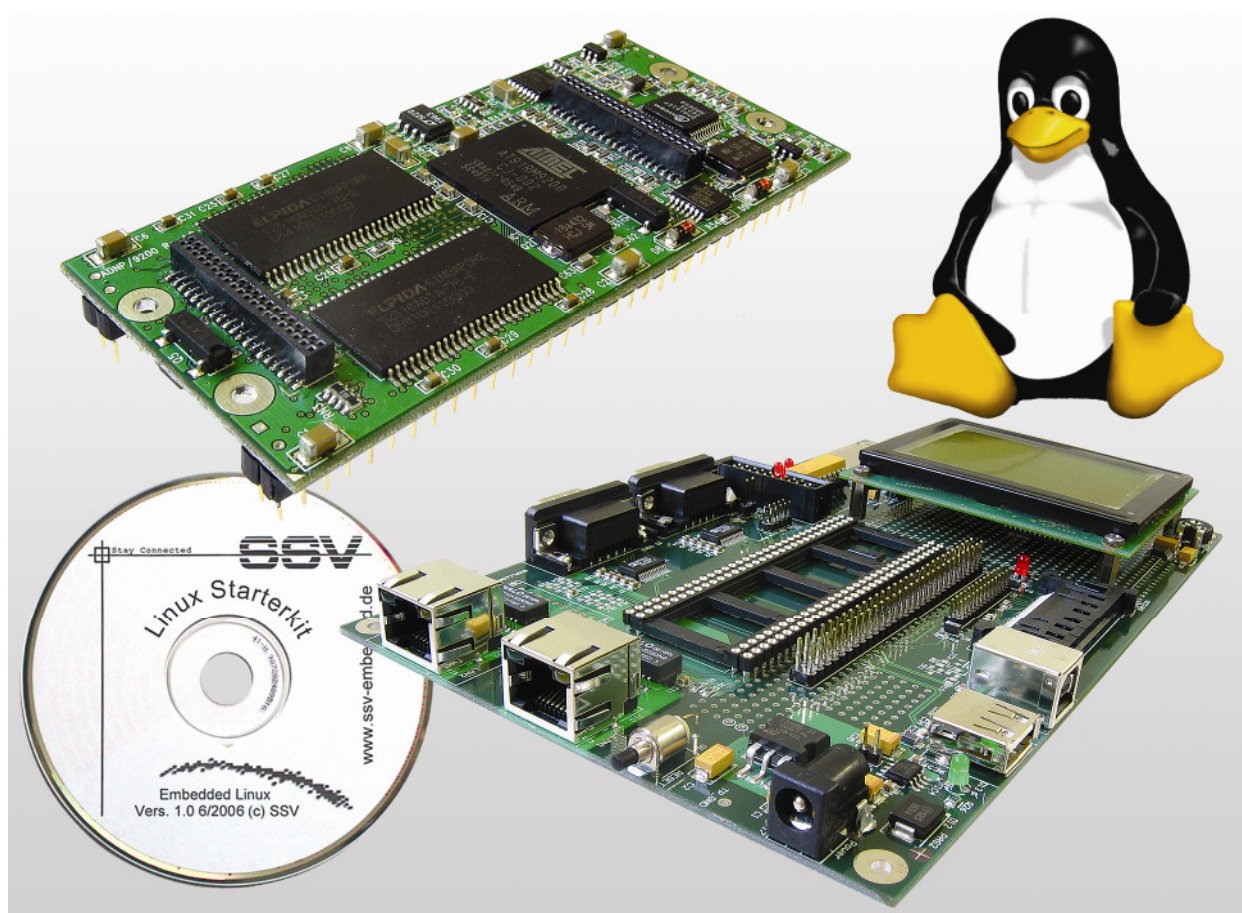


DNP/SK27

Embedded Linux Starter Kit

First Steps



SSV Embedded Systems

Heisterbergallee 72

D-30453 Hannover

Phone: +49 (0)511/40 000-0

Fax: +49 (0)511/40 000-40

E-mail: sales@ist1.de

Manual Revision: 1.0

Date: 2006-10-24

CONTENT

1	INTRODUCTION.....	3
1.1	Safety Guidelines	3
1.2	Conventions	3
1.3	Features and Technical Data	4
2	GETTING STARTED.....	5
2.1	Serial Link between DNP/SK27 and PC.....	5
2.2	Ethernet Link between DNP/SK27 and PC	6
2.3	Connecting Power Supply and Power-up the DNP/SK27	7
2.4	Using Serial Link with Terminal Program.....	8
2.5	Power-up ADNP/9200 with RCM disabled.....	9
2.6	Power-up ADNP/9200 with RCM enabled.....	10
2.7	Checking IP Address of PC	11
2.8	Checking Ethernet-based TCP/IP Communication.....	12
2.9	Using a Telnet Connection.....	13
2.10	Checking Embedded Web Server	14
2.11	Checking FTP Server	16
2.12	Checking TFTP Client	18
2.13	Changing Ex Factory IP Address (ipaddress usage)	19
3	U-BOOT COMMAND OVERVIEW	21
4	ADNP/9200 LINUX BOOT MESSAGES.....	22
5	HELPFUL LITERATURE.....	26
	CONTACT	26
	DOCUMENT HISTORY	26

1 INTRODUCTION

The DIL/NetPC ADNP/9200 Starter Kit contains everything you need to get started with your Atmel AT91RM9200 ARM9-based embedded networking application. The Starter Kit includes an ADNP/9200 module with a pre-installed U-Boot boot loader and an embedded Linux, the Evaluation Board DNP/EVA11, power supply, serial interface (null modem) cable, a CD-ROM with software and documentation and a printed user manual for the first steps with the Starter Kit.

The Starter Kit CD-ROM comes with a full GNU cross tool chain for C/C++ software development. The binary files of this pre-build tool chain run on an x86 Linux-based host (SuSE, Red Hat or other) and build executable files for the Atmel AT91RM9200 ARM9-based microcontroller.

For using the DNP/SK27 Embedded Linux Starter Kit you need a development system. The minimal configuration for this system is a Windows-based PC with the HyperTerminal terminal emulation program and a free COM port (COM1, COM2 or USB-based COMx) for the RS232 serial link between the ADNP/9200 and HyperTerminal.

For using the Ethernet link, your PC needs an Ethernet adapter with 10 Mbps or 10/100 Mbps LAN interface. This environment allows web server programming (HTML pages, Java Applets) and Linux shell script programming. For using the GNU C/C++ cross tool chain, it is necessary to run Linux on the development system.

1.1 Safety Guidelines

Please read the following safety guidelines carefully! In case of property or personal damage by not paying attention to this document and/or by incorrect handling, we do not assume liability. In such cases any warranty claim expires.



ATTENTION: Observe precautions for handling – electrostatic sensitive device!

- Discharge yourself before you work with the device, e.g. by touching a heater of metal, to avoid damages.
- Stay grounded while working with the device to avoid damage through electrostatic discharge.

1.2 Conventions

Convention	Usage
bold	Important terms
<i>italic</i>	Filenames, user inputs
monospace	Pathnames, program code, command lines

Table 1: Conventions used in this Document

1.3 Features and Technical Data

The DIL/NetPC ADNP/9200 comes with a pre-installed U-Boot boot loader and an Embedded Linux operating system. The ADNP/9200 Linux consists of two main components: 1. the Linux kernel and 2. the root file system.

The ADNP/9200 U-Boot boot loader allows the downloading of new Linux kernel versions and root file systems to the ADNP/9200 RAM and Flash. This in-system programming feature can be used by a simple serial and Ethernet link between the development system and the ADNP/9200.

- DIL/NetPC ADNP/9200 with Atmel AT91RM9200, 32 Mbytes Flash and 64 Mbytes SDRAM, 3.3 VDC Vcc
- U-Boot boot loader and Embedded Linux pre-installed in Flash memory
- Evaluation Board DNP/EVA11
- Null modem cable
- 110 VAC or 230 VAC to 5 VDC international power supply
- CD-ROM with user manual and hardware/programmers manuals
- Embedded Linux with source
- GNU cross tool chain for C/C++ software development for Linux-based PCs
- GNU gdb and gdbserver for Ethernet-based remote debugging
- Linux remote login with Telnet
- Web server setup sample
- FTP server setup sample
- Many source code samples

2 GETTING STARTED

2.1 Serial Link between DNP/SK27 and PC

Setup the serial link between the DNP/SK27 and your PC. Use a null modem cable for this connection.

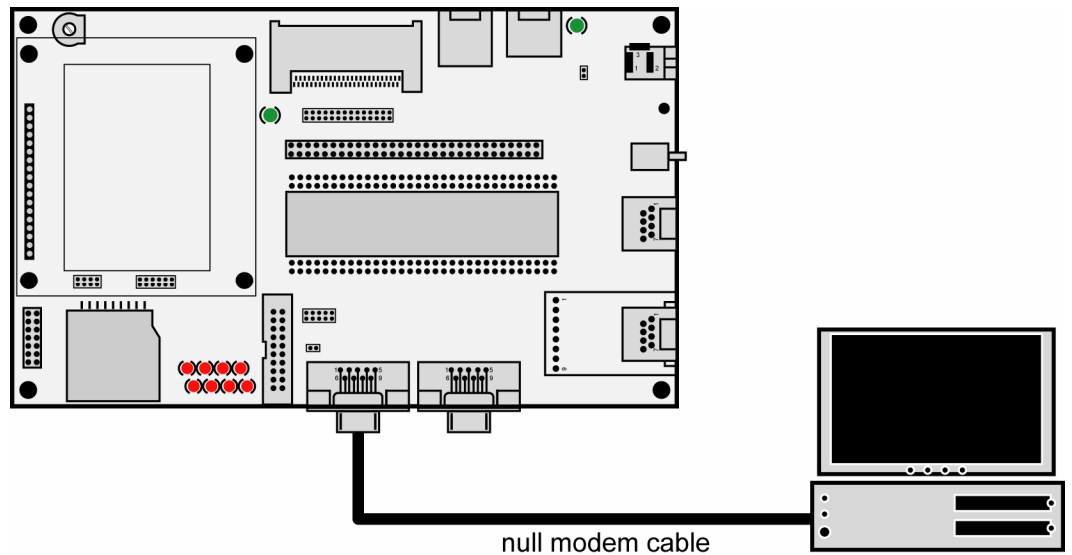


Figure 1: Serial link between DNP/SK27 and PC

Connect one end of the null modem cable with the COM1 port of the DNP/SK27 and an unused COM port of the PC. Make sure that this PC COM port supports 115.200 bps.

2.2 Ethernet Link between DNP/SK27 and PC

Setup the Ethernet LAN link between the LAN1 connector of the DNP/SK27 and your PC. Use an Ethernet cross-over cable or a switch-based infrastructure for the first LAN connection.

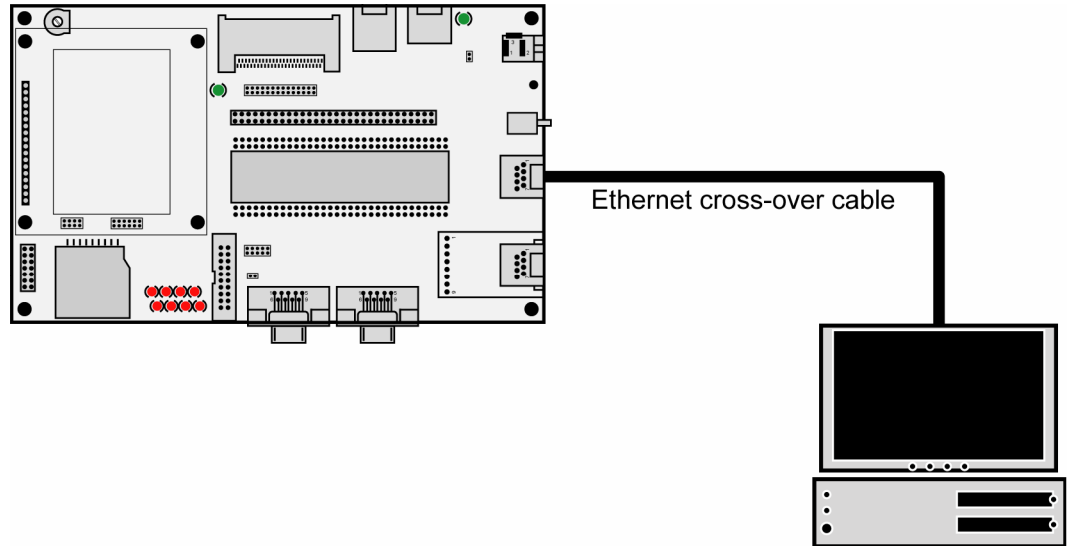


Figure 2: Ethernet link between DNP/SK27 and PC

Please note: The ADNP/9200 comes with the default IP address **192.168.0.126**. Please make sure that your PC can work with the IP address range 192.168.0.x.

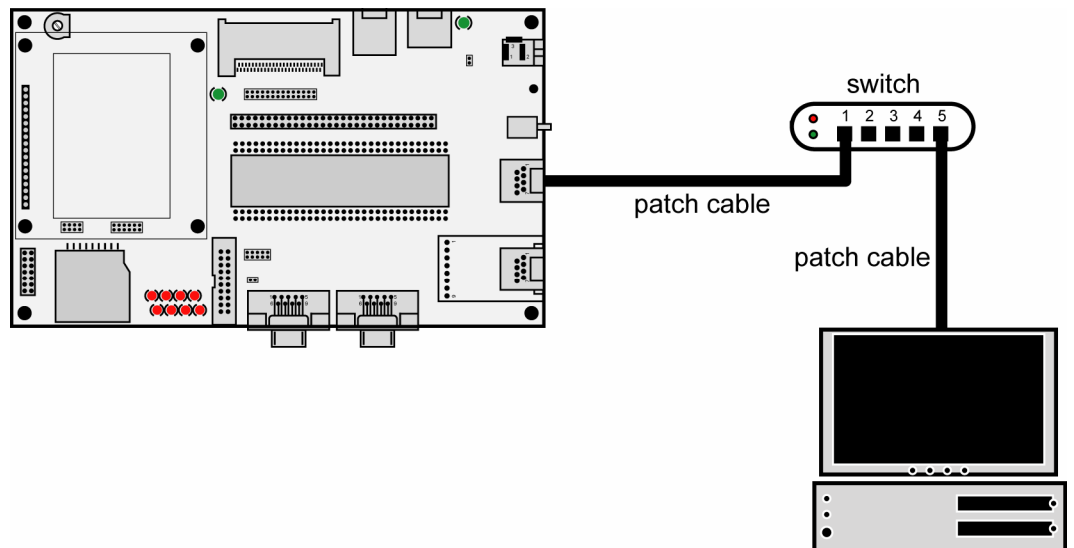


Figure 3: Switch-based Ethernet link between DNP/SK27 and PC

2.3 Connecting Power Supply and Power-up the DNP/SK27

Connect a 5 VDC power supply with a 5.5 mm x 2.5 mm jack plug to the DNP/SK27.

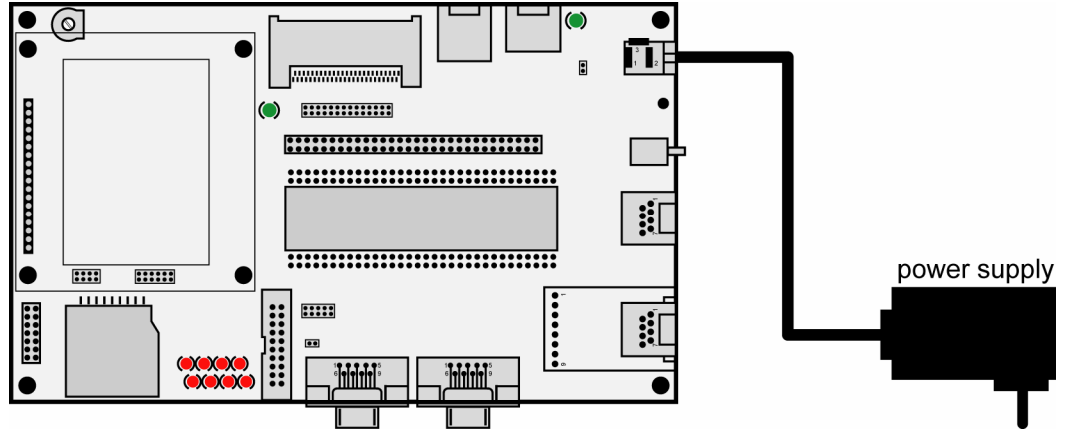


Figure 4: Power supply for the DNP/SK27



CAUTION: Providing the DNP/SK27 with a voltage higher than the regular 5 VDC $\pm 10\%$ could result in damaged board components!

Please pay attention to the polarity of the power connector: the **+ pole is in the center!**

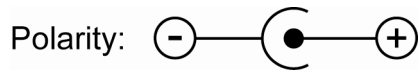


Figure 5: Polarity of the power connector

Please note: Make sure that all cable connections are OK. Then power-up the Starter Kit.

2.4 Using Serial Link with Terminal Program

Run *HyperTerminal* on your Windows-PC, *minicom* or a similar simple terminal emulation program on your Linux-based PC.



Figure 6: Direct connection setup with HyperTerminal

Setup a direct connection with the parameters of table 2. Make sure, that the PC COM port supports 115.200 bps.

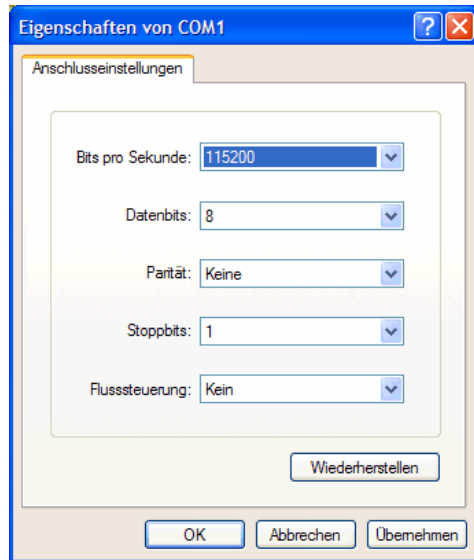


Figure 7: Parameter setup with HyperTerminal

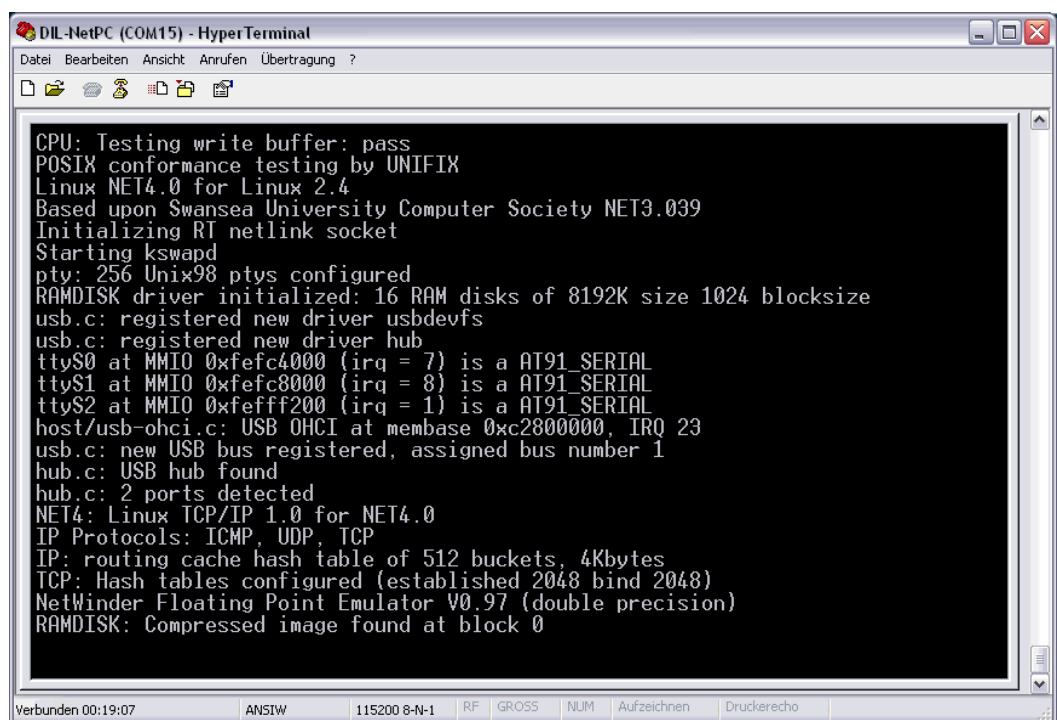
Parameter	Value
Speed	115.200 bps
Data Bits	8
Parity	None
Stop Bits	1
Protocol	No (Xon/Xoff, RTS/CTS or similar)

Table 2: Setup parameters for the serial link

2.5 Power-up ADNP/9200 with RCM disabled

After power-up the DIL/NetPC ADNP/9200 starts an automatic boot process from the on-board flash memory chip. This process consists of two steps:

1. Directly after power-up, the ADNP/9200 runs the U-Boot boot loader program for some milliseconds. U-Boot initializes the hardware components (hardware init). **With RCM disabled** (please see the *DNP/EVA11 hardware reference manual* for details), there is no U-Boot text message output over the ADNP/9200 COM1 serial interface and no **boot delay**-based¹ wait period. Direct after the hardware init, the U-Boot boot loader starts the Linux OS image.
2. Linux takes control over the ADNP/9200 hardware and runs all necessary processes for coming up to live.



```

CPU: Testing write buffer: pass
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
pty: 256 Unix98 ptys configured
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
usb.c: registered new driver usbdevfs
usb.c: registered new driver hub
ttyS0 at MMIO 0xfefc4000 (irq = 7) is a AT91_SERIAL
ttyS1 at MMIO 0xfefc8000 (irq = 8) is a AT91_SERIAL
ttyS2 at MMIO 0xfefff200 (irq = 1) is a AT91_SERIAL
host/usb-ohci.c: USB OHCI at membase 0xc2800000, IRQ 23
usb.c: new USB bus registered, assigned bus number 1
hub.c: USB hub found
hub.c: 2 ports detected
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 2048)
NetWinder Floating Point Emulator V0.97 (double precision)
RAMDISK: Compressed image found at block 0
  
```

Figure 8: Linux booting process with HyperTerminal

Please note: The U-Boot environment variable **boot delay** does not influence the ADNP/9200 boot process with RCM (Remote Console Mode) disabled.

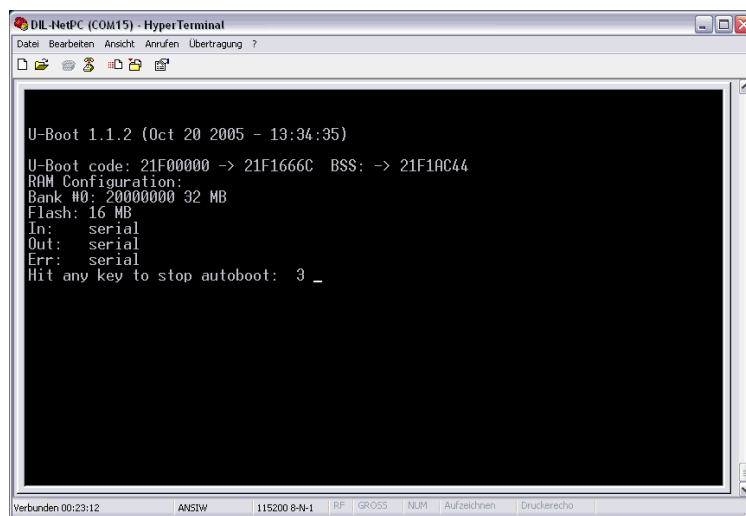
The ADNP/9200 Linux supports a serial console. It allows running a Linux-based system in a headless configuration without a monitor or keyboard. Wait until the Linux boot process finishes. Then enter the user name **root**. This user name needs no password. Just hit Enter if the ADNP/9200 Linux asks for a password.

¹ “**boot delay**” is a U-Boot environment variable. The value defines a wait time before U-Boot starts the Linux operating system.

2.6 Power-up ADNP/9200 with RCM enabled

The DIL/NetPC ADNP/9200 boot sequence with RCM enabled is similar to the boot procedure with RCM disabled. Only the first step is different:

1. The ADNP/9200 runs the U-Boot boot loader program. This software shows a wait message over the ADNP/9200 COM1 serial interface if RCM is enabled (please see the *DNP/EVA11 hardware reference manual* for details). It is possible to interrupt the boot process and switch to the U-Boot command line interface. Just hit a key of your terminal emulation program.
2. Without interruption the U-Boot boot loader starts a Linux OS image after the wait period from the ADNP/9200 Flash memory.



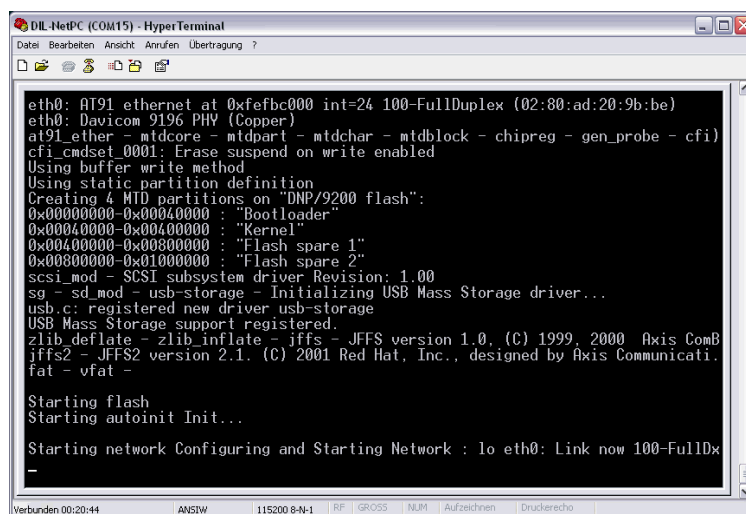
```

U-Boot 1.1.2 (Oct 20 2005 - 13:34:35)
U-Boot code: 21F00000 -> 21F1666C BSS: -> 21F1AC44
RAM Configuration:
Bank #0: 20000000 32 MB
Flash: 16 MB
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 3 _

```

Figure 9: U-Boot wait message

Please note: The U-Boot command line interface allows you to change the wait time of the first step. Please see the U-Boot environment variable **boot delay** for details.



```

eth0: AT91 ethernet at 0xfefbc000 int=24 100-FullDuplex (02:80:ad:20:9b:be)
eth0: Davicom 9196 PHY (Copper)
at91_ether - mtdcore - mtdpart - mtdchar - mtdblock - chipreg - gen_probe - cfi)
cfi_cmdset_0001: Erase suspend on write enabled
Using buffer write method
Using static partition definition
Creating 4 MTD partitions on "DNP/9200 flash":
0x00000000-0x00040000 : "Bootloader"
0x00040000-0x00040000 : "Kernel"
0x00040000-0x00080000 : "Flash spare 1"
0x00080000-0x01000000 : "Flash spare 2"
scsi_mod - SCSI subsystem driver Revision: 1.00
sg - sd_mod - usb-storage - Initializing USB Mass Storage driver...
usb.c: registered new driver usb-storage
USB Mass Storage support registered.
zlib deflate - zlib inflate - iffs - JFFS version 1.0. (C) 1999, 2000 Axis ComB
iffs2 - JFFS2 version 2.1. (C) 2001 Red Hat, Inc., designed by Axis Communicati.
fat - vfat -

Starting flash
Starting autoinit Init...

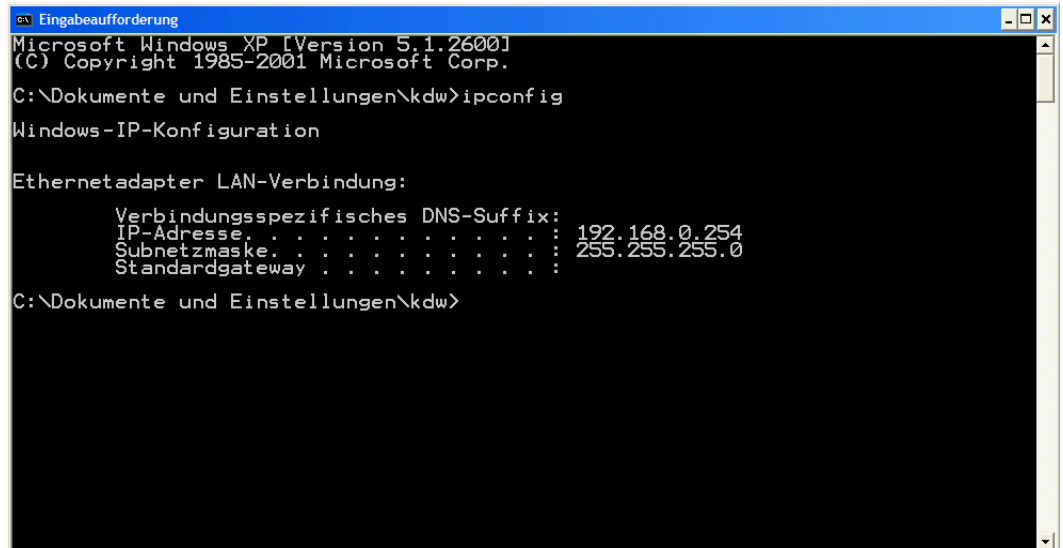
Starting network Configuring and Starting Network : lo eth0: Link now 100-FullDx

```

Figure 10: Linux booting process after the U-Boot boot delay

2.7 Checking IP Address of PC

Make sure that your PC is using the right IP address for the Ethernet-based TCP/IP communication with the DIL/NetPC. Use 192.168.0.1 or 192.168.0.254 for your PC and 192.168.0.126 for the ADNP/9200.



```
Eingabeaufforderung
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Dokumente und Einstellungen\kdw>ipconfig

Windows-IP-Konfiguration

Ethernetadapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix:
    IP-Adresse. . . . . : 192.168.0.254
    Subnetzmaske. . . . . : 255.255.255.0
    Standardgateway . . . . . :

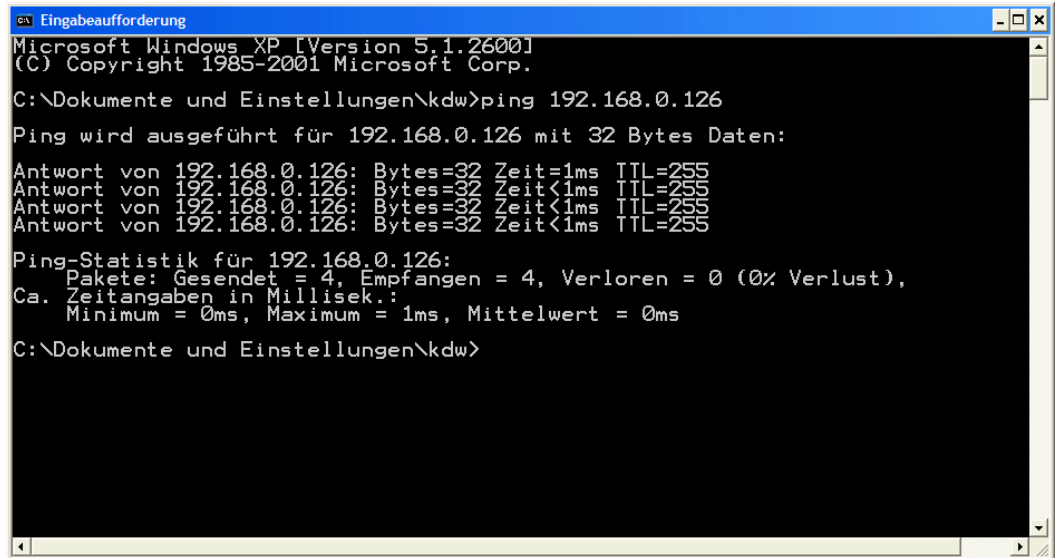
C:\Dokumente und Einstellungen\kdw>
```

Figure 11: Windows-PC IP address check with *ipconfig*

Talk to your network administrator if you have problems with the IP address understanding.

2.8 Checking Ethernet-based TCP/IP Communication

Check the Ethernet-based TCP/IP communication between the ADNP/9200 and the PC with a simple *ping* command.



```
Eingabeaufforderung
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Dokumente und Einstellungen\kdw>ping 192.168.0.126

Ping wird ausgeführt für 192.168.0.126 mit 32 Bytes Daten:

Antwort von 192.168.0.126: Bytes=32 Zeit=1ms TTL=255
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=255
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=255
Antwort von 192.168.0.126: Bytes=32 Zeit<1ms TTL=255

Ping-Statistik für 192.168.0.126:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Dokumente und Einstellungen\kdw>
```

Figure 12: Windows-PC TCP/IP communication check with *ping*

First check the cable connections and then the IP addresses if your ping does not work. Then check the TCP/IP setup of your PC.

2.9 Using a Telnet Connection

Run a Telnet client program on your PC with the IP address of the ADNP/9200. You can use a Telnet session for remote entering Linux commands.

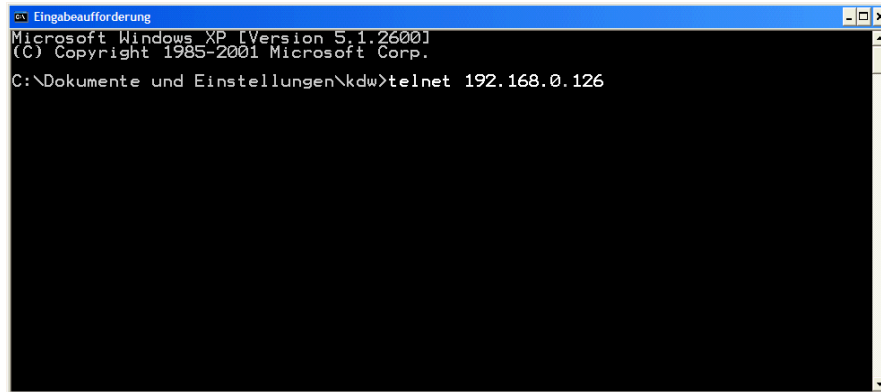


Figure 13: Run the Windows telnet client program

Wait until the ADNP/9200 Linux requests a user name. Please enter the user name *root*. This user name needs no password. Just hit Enter if the ADNP/9200 Linux requests a password.

Please note: The ADNP/9200 Linux comes with *BusyBox*. All Linux command line commands are implemented in *BusyBox*. *BusyBox* combines tiny versions of many common UNIX utilities into a single small executable. It provides replacements for most of the utilities you usually find in GNU *fileutils*, *shellutils*, etc. The utilities in *BusyBox* generally have fewer options than their full-featured GNU cousins; however, the options that are included provide the expected functionality and behave very much like their GNU counterparts. *BusyBox* provides a fairly complete environment for any small or embedded system.

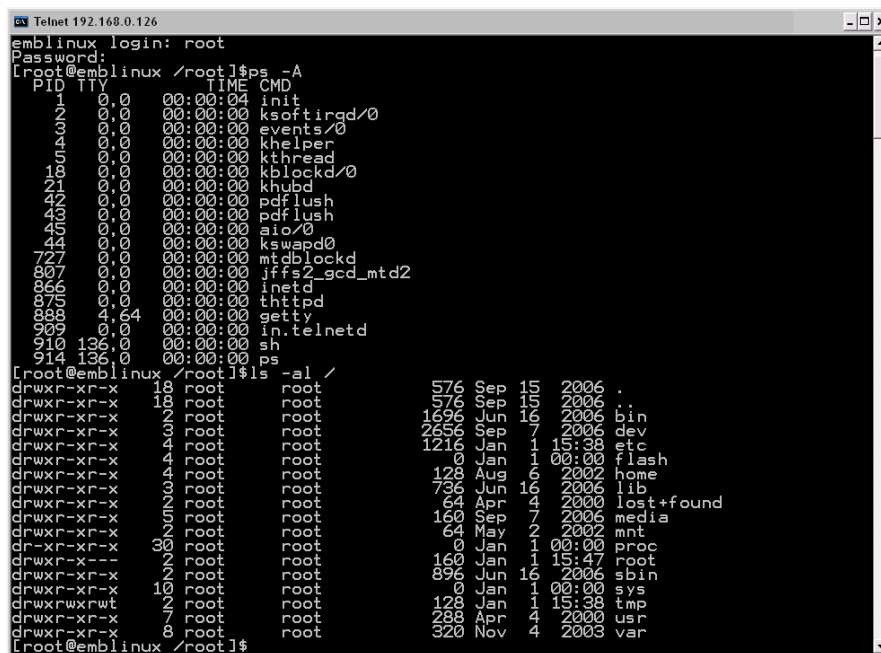


Figure 14: Using Linux commands within a Telnet client window

2.10 Checking Embedded Web Server

The ADNP/9200 default Linux configuration comes with a pre-installed embedded web server. The object storage space (HTML pages, pictures, CGI programs, Java Applets, ...) for this web server is located within the ADNP/9200 Flash memory.

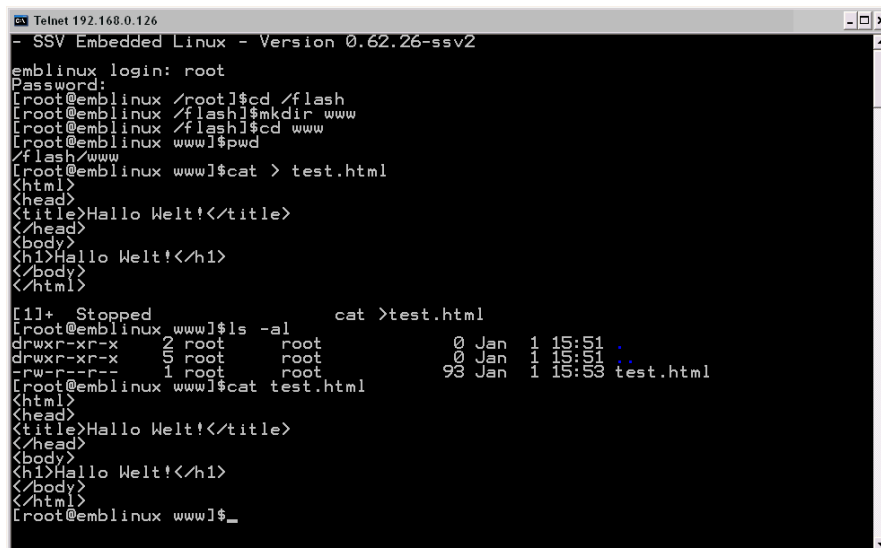
Setup a Telnet session with root user rights from your PC to the ADNP/9200 (see chapter 2.9 if necessary). Enter the following command lines within this Telnet session:

```
cd /flash
mkdir www
cd www
cat > test.html
<html>
<head>
<title>Hallo Welt!</title>
</head>
<body>
<h1>Hallo Welt!</h1>
</body>
</html>
```

To stop the Linux cat command press two times (!) CTRL-D.

These command lines create an HTML file `/flash/www/test.html` within the ADNP/9200 Flash memory with the German version of “Hello World” (“Hallo Welt!”).

Reboot your ADNP/9200. This reboot defines `/flash/www/` as default directory for the web server.



```
Telnet 192.168.0.126
- SSV Embedded Linux - Version 0.62.26-ssv2
emblinux login: root
Password:
root@emblinux /root]#cd /flash
root@emblinux /flash]#mkdir www
root@emblinux /flash]#cd www
root@emblinux www]#pwd
/flash/www
root@emblinux www]#cat > test.html
<html>
<head>
<title>Hallo Welt!</title>
</head>
<body>
<h1>Hallo Welt!</h1>
</body>
</html>
[!]+ Stopped
root@emblinux www]#ls -al
drwxr-xr-x  2 root  root    0 Jan  1 15:51 .
drwxr-xr-x  5 root  root    0 Jan  1 15:51 ..
-rw-r--r--  1 root  root   93 Jan  1 15:53 test.html
root@emblinux www]#cat test.html
<html>
<head>
<title>Hallo Welt!</title>
</head>
<body>
<h1>Hallo Welt!</h1>
</body>
</html>
root@emblinux www]#_
```

Figure 15: Create an HTML file within a Telnet session

Run your PC web browser and access the HTML file `test.html` with your browser. Use the URL `http://192.168.0.126/test.html`. This URL assumes that your ADNP/9200 is using the IP address 192.168.0.126 for the Ethernet LAN interface LAN1. Change this IP address if necessary.

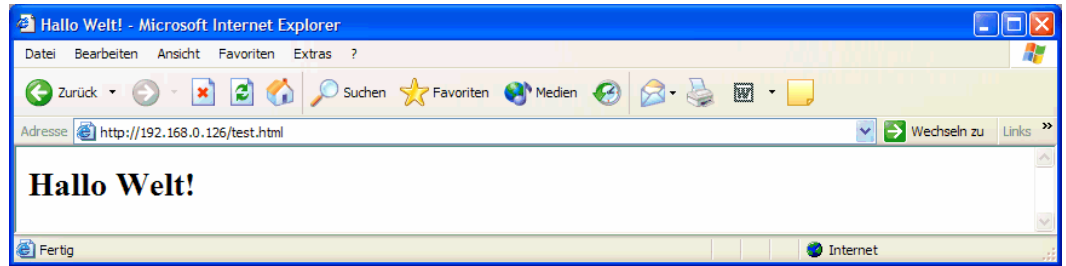


Figure 16: Check the embedded web server with Internet Explorer

Please note: `/flash/www` is the default directory for the ADNP/9200 embedded web server.

Restart the ADNP/9200 Linux after the creation of `/flash/www/test.html` and before the first access with a web browser.

The ADNP/9200 embedded web server looks out for `/flash/www` at boot time. If the directory `/flash/www` does not exist at boot time, the ADNP/9200 embedded web server works with the RAM disk-based directory `/usr/local/www`.

2.11 Checking FTP Server

The DIL/NetPC ADNP/9200 Linux comes with a pre-installed FTP server. This server allows the file transfer between a PC and the ADNP/9200.

Run an FTP client program on your PC. Set the session parameters for your FTP client to IP address **192.168.0.126**, user name **guest** and password **guest**. The ADNP/9200 default directory for the FTP user name guest is /home/guest. This directory is located within the ADNP/9200 RAM disk.

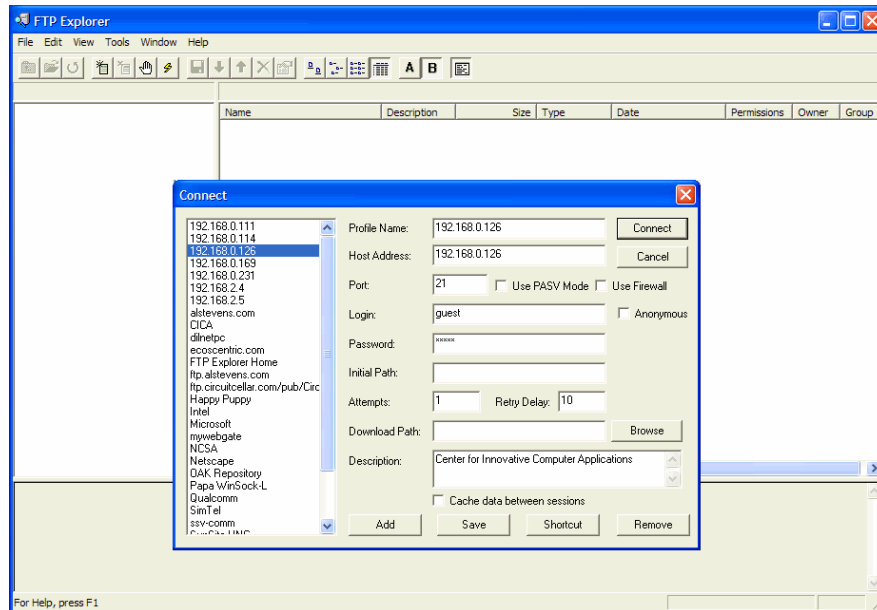


Figure 17: Set the session parameters for an FTP client program

Connect your FTP client with the ADNP/9200 FTP server. Please watch the ADNP/9200 default FTP directory. Transfer some files from your PC to the ADNP/9200. The ADNP/9200 stores these files in the ADNP/9200 default FTP directory.

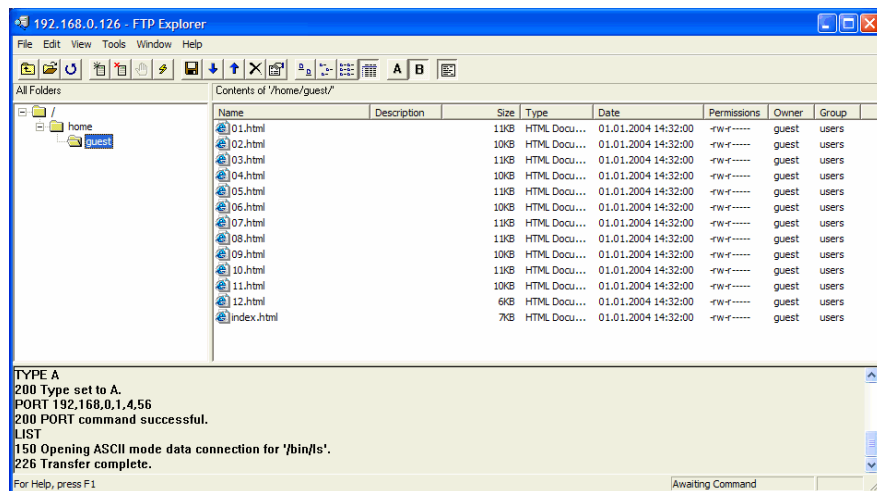


Figure 18: Transfer some files to the ADNP/9200

Check the new files with a Telnet session. Change to the ADNP/9200 default FTP directory /home/guest within your Telnet session.

```

Telnet 192.168.0.126
- SSV Embedded Linux - Version 0.20
emblinux login: root
Password:
[root@emblinux /root]#cd /home/guest
[root@emblinux guest]#ls -al
drwxr-xr-x  2 guest  users      480 Jan  1 14:32 .
drwxr-xr-x  4 root   root       128 Aug  6 14:20 ..
-rw-r----- 11 guest  users     6436 Jan 14 00:00 .
-rw-r----- 10296  guest  users     9796 Jan 14 00:00 .
-rw-r----- 10315  guest  users     9761 Jan 14 00:00 index.html
-rw-r----- 10329  guest  users     9328 Jan 14 00:00 01.htm
-rw-r----- 10306  guest  users     9306 Jan 14 00:00 02.htm
-rw-r----- 10317  guest  users     9317 Jan 14 00:00 03.htm
-rw-r----- 10327  guest  users     9327 Jan 14 00:00 04.htm
-rw-r----- 10356  guest  users     9356 Jan 14 00:00 05.htm
-rw-r----- 10372  guest  users     9372 Jan 14 00:00 06.htm
-rw-r----- 10388  guest  users     9388 Jan 14 00:00 07.htm
-rw-r----- 10404  guest  users     9404 Jan 14 00:00 08.htm
-rw-r----- 10420  guest  users     9420 Jan 14 00:00 09.htm
-rw-r----- 10436  guest  users     9436 Jan 14 00:00 10.htm
-rw-r----- 10452  guest  users     9452 Jan 14 00:00 11.htm
-rw-r----- 10468  guest  users     9468 Jan 14 00:00 12.htm
[root@emblinux guest]#

```

Figure 19: The new files within /home/guest

2.12 Checking TFTP Client

The DIL/NetPC ADNP/9200 Linux offers also a pre-installed TFTP client. This program allows the TFTP-based file transfer between a PC and the ADNP/9200.

First run a TFTP server program on your PC system. Linux-based PCs offer a TFTP server as part of the standard networking feature set. For Windows-based PCs please use the free *Win32 TFTP* server program from the Starter Kit CD-ROM (see directory /TFTP-Server-Win32 at the Starter Kit CD-ROM).

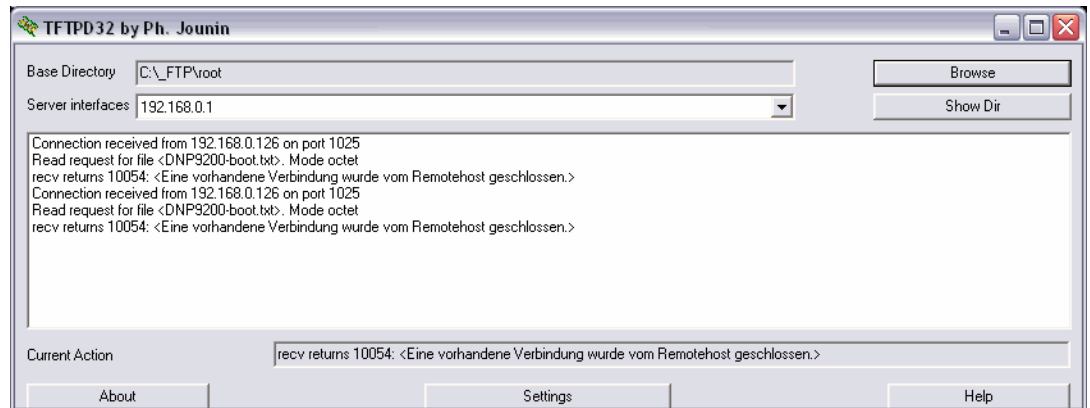


Figure 20: Running the Win32 TFTP server

Figure 20 shows the free Win32 TFTP server in action. This server allows file transfers to and from any ADNP/9200 directory.

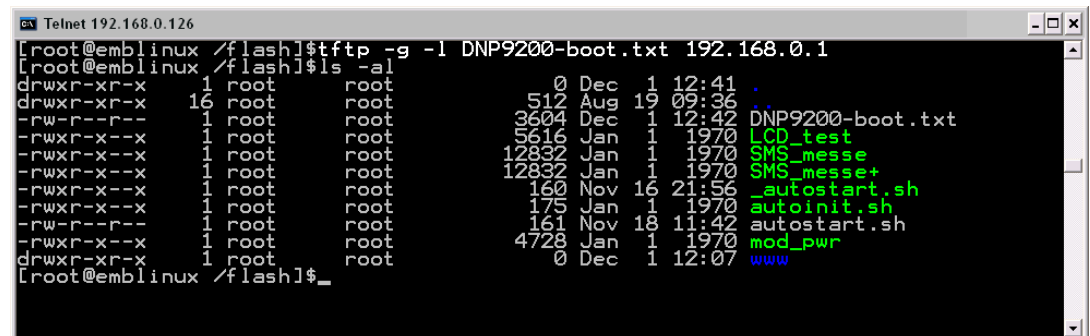


Figure 21: Download with TFTP *get* command

Enter your *get* and *put* commands within a Telnet session. Figure 21 shows a sample for a TFTP *get* command (getting a file from the PC to the ADNP/9200). Here we enter:

```
tftp -g -l DNP9200-boot.txt 192.168.0.1
```

DNP9200-boot.txt is the filename which is requested from the TFTP server. **192.168.0.1** is the TFTP server IP address.

2.13 Changing Ex Factory IP Address (ipaddree usage)

The DIL/NetPC ADNP/9200 is automatically assigned an IP address on DHCP-enabled networks as it is DHCP-enabled by default. If DHCP does not work (i.e. no DHCP server available), the ADNP/9200 takes its default IP address.

The ex factory value for this default IP address is **192.168.0.126**. The network mask for this address is **255.255.255.0**. This IP address can be changed over an RS232-based serial link with help of the program called *ipaddree*.

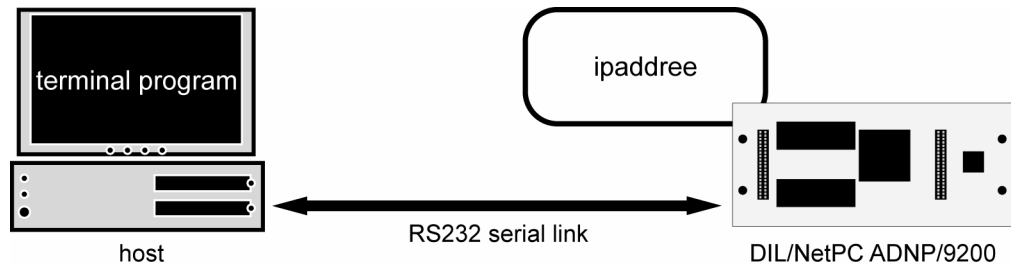
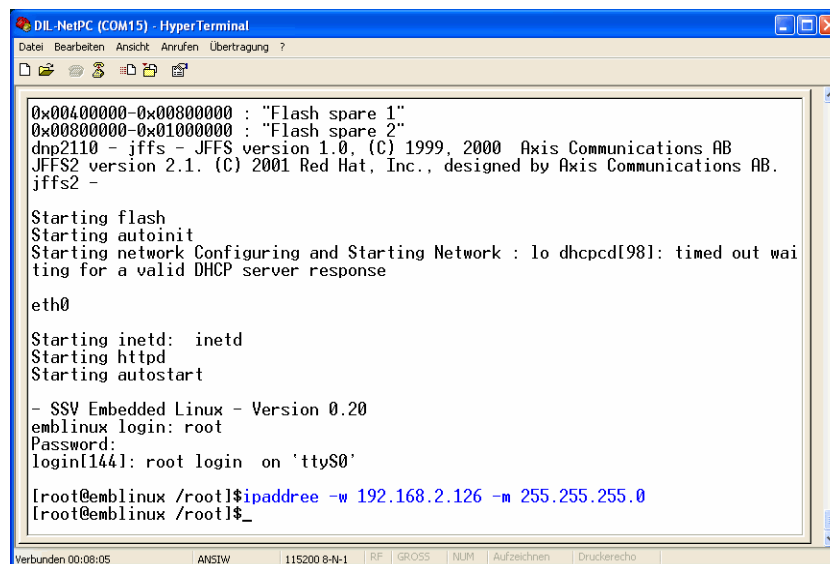


Figure 22: Environment for ipaddree usage

Use a serial console and enter the user name *root*. This user name needs no password. Just hit Enter if the ADNP/9200 Linux requests a password. Then execute the following command:

```
ipaddree -w 192.168.2.126 -m 255.255.255.0
```

“ipaddree” is the name of a command line IP setup tool. “192.168.2.126” is an IP address. Use the IP address of your choice for your setup on this position. “255.255.255.0” is a network mask. Use a valid network mask for your IP address.



```
DIL-NetPC (COM15) - HyperTerminal
Datei Bearbeiten Ansicht Anrufen Übertragung ?
0x00400000-0x00800000 : "Flash spare 1"
0x00800000-0x01000000 : "Flash spare 2"
dnp2110 - jffs - JFFS version 1.0. (C) 1999, 2000 Axis Communications AB
JFFS2 version 2.1. (C) 2001 Red Hat, Inc., designed by Axis Communications AB.
jffs2 -
Starting flash
Starting autoinit
Starting network Configuring and Starting Network : lo dhcpd[98]: timed out waiting for a valid DHCP server response
eth0
Starting inetd: inetd
Starting httpd
Starting autostart
- SSV Embedded Linux - Version 0.20
emlinux login: root
Password:
login[144]: root login on 'ttyS0'
[root@emlinux /root]$ipaddree -w 192.168.2.126 -m 255.255.255.0
[root@emlinux /root]$_
```

Figure 23: Serial console with ipaddree command line

Reboot the ADNP/9200. The new IP address and network mask are valid after the next boot process if no DHCP server is available or if DHCP does not work. Check the new IP address with *ping*.



```
Eingabeaufforderung
C:\>ping 192.168.2.126
Ping wird ausgeführt für 192.168.2.126 mit 32 Bytes Daten:
Antwort von 192.168.2.126: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.126: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.126: Bytes=32 Zeit<1ms TTL=64
Antwort von 192.168.2.126: Bytes=32 Zeit<1ms TTL=64
Ping-Statistik für 192.168.2.126:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
C:\>
```

Figure 24: Check the new IP address with *ping*

Please note: Do not forget to change the IP address of your PC to 192.168.2.1 or similar.

The program *ipaddree* comes as version for the DNP/9200. This is the reason why it supports only the Ethernet interface **eth0**.

3 U-BOOT COMMAND OVERVIEW

The user interface to U-Boot consists of a command line interpreter (CLI), much like a Linux shell prompt. When connected via a serial line you can interactively enter commands and see the results. The following table shows the available U-Boot commands for the DIL/NetPC ADNP/9200.

Command	Function
autoscr	run script from memory
base	print or set address offset
bdinfo	print Board Info structure
bootm	boot application image from memory
bootp	boot image via network using BootP/TFTP protocol
bootd	boot default, i.e., run 'bootcmd'
cmp	memory compare
cp	memory copy
crc32	checksum calculation
echo	echo args to console
erase	erase FLASH memory
flinfo	print FLASH memory information
go	start application at address 'addr'
help	print online help
iminfo	print header information for application image
loadb	load binary file over serial line (kermit mode)
loadc	load binary file over serial line (ymodem-c mode)
loadg	load binary file over serial line (ymodem-g mode)
loads	load S-Record file over serial line
loop	infinite loop on address range
md	memory display
mm	memory modify (auto-incrementing)
mtest	simple RAM test
mw	memory write (fill)
nm	memory modify (constant address)
printenv	print environment variables
protect	enable or disable FLASH write protection
rarpboot	boot image via network using RARP/TFTP protocol
reset	perform RESET of the CPU
run	run commands in an environment variable
saveenv	save environment variables to persistent storage
setenv	set environment variables
sleep	delay execution for some time
tftpboot	boot image via network using TFTP protocol and env variables ipaddr and serverip
version	print monitor version
?	alias for 'help'

Table 3: U-Boot command overview

4 ADNP/9200 LINUX BOOT MESSAGES

Please note: This symbol “↵” shows that the following line belongs to the previous line.

```

U-Boot 1.1.2 (Sep 12 2006 - 13:14:57)

U-Boot code: 21F00000 -> 21F1666C BSS: -> 21F1AC44
RAM Configuration:
Bank #0: 20000000 64 MB
Flash: 16 MB
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 3 ... 2 ... 1 ... 0
## Booting image at 10040000 ...
   Image Name: LINUXIMAGE-ADNP9200-20060915
   Image Type: ARM Linux Multi-File Image (uncompressed)
   Data Size: 3416352 Bytes = 3.3 MB
   Load Address: 20008000
   Entry Point: 20008000
   Contents:
     Image 0: 1690464 Bytes = 1.6 MB
     Image 1: 1725876 Bytes = 1.6 MB
   Verifying Checksum ... OK
OK
Loading Ramdisk at 101dcbac - 10382160, to 20408000 - 205ad5b4 ... OK

Starting kernel ...

Linux version 2.6.16.20-at91-ssv2 (root@linux) SSV20060616 (gcc version 3.4.4) ↵
#1 Fri Sep 15 10:17:46 CEST 2006
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
Machine: SSV DNP/9200
Memory policy: ECC disabled, Data cache writeback
Clocks: CPU 179 MHz, master 59 MHz, main 18.432 MHz
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets

CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Built 1 zonelists
Kernel command line: console=ttyS0,115200 root=/dev/ram
AT91: 128 gpio irqs in 4 banks
PID hash table entries: 512 (order: 9, 8192 bytes)
Console: colour dummy device 80x30
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 64MB = 64MB total

Memory: 61384KB available (1348K code, 303K data, 76K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
checking if image is initramfs...it isn't (no cpio magic); looks like an initrd
Freeing initrd memory: 1685K
NET: Registered protocol family 16
usbcore: registered new driver usbfs
usbcore: registered new driver hub
NetWinder Floating Point Emulator V0.97 (double precision)
Initializing Cryptographic API
io scheduler noop registered (default)

```

```
at91_usart.0: ttyS0 at MMIO 0xffffc4000 (irq = 7) is a AT91_SERIAL
at91_usart.1: ttyS1 at MMIO 0xffffc8000 (irq = 8) is a AT91_SERIAL
at91_usart.2: ttyS2 at MMIO 0xffffc0000 (irq = 6) is a AT91_SERIAL
at91_usart.3: ttyS3 at MMIO 0xffeff200 (irq = 1) is a AT91_SERIAL
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
eth0: Link down.

eth0: AT91 ethernet at 0xfefbc000 int=24 10-HalfDuplex (02:80:ad:20:9b:c4)
eth0: Davicom 9196 PHY (Copper)
usbmon: debugfs is not available
at91_ohci at91_ohci: AT91 OHCI
at91_ohci at91_ohci: new USB bus registered, assigned bus number 1
at91_ohci at91_ohci: irq 23, io mem 0x00300000

usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
hub 1-0:1.0: 2 ports detected
mice: PS/2 mouse device common for all mice
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 2, 16384 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered

TCP bic registered
RAMDISK: Compressed image found at block 0
VFS: Mounted root (minix filesystem).
Freeing init memory: 76K

INIT: version 2.74 booting

IP=
ipaddree info: No IP configuration in EEPROM

INIT: Entering runlevel: 3

Starting modutils Loading modules:
intermodule - zlib_deflate - zlib_inflate - af_packet - NET: Registered protocol ↵
family 17
unix - NET: Registered protocol family 1
mtdcore - mtdpart - mtdchar - mtd_blkdevs - mtdblock - chipreg - gen_probe - ↵
cfi_util - cfi_probe - cfi_cmdset_0001 - dnp9200 - Probing DNP/9200 flash at ↵
physical address 0x10000000 (16-bit buswidth)
DNP/9200 flash: Found 1 x16 devices at 0x0 in 16-bit bank
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
cfi_cmdset_0001: Erase suspend on write enabled
Using static partition definition
Creating 4 MTD partitions on "DNP/9200 flash":
0x00000000-0x00040000 : "Bootloader"
0x00040000-0x00400000 : "Kernel"
0x00400000-0x00800000 : "Flash spare 1"
0x00800000-0x01000000 : "Flash spare 2"
Probing ADNP/9200 flash at physical address 0x30000000 (16-bit buswidth)
ADNP/9200 flash: Found 1 x16 devices at 0x0 in 16-bit bank
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method

cfi_cmdset_0001: Erase suspend on write enabled
Using static partition definition
Creating 1 MTD partitions on "ADNP/9200 flash":
0x00000000-0x01000000 : "spare Flash"
```

```
scsi_mod - SCSI subsystem initialized
sg - sd_mod - usb-storage - Initializing USB Mass Storage driver...
usbcore: registered new driver usb-storage
USB Mass Storage support registered.
nls_base - nls_cp437 - nls_iso8859-1 - jffs2 - JFFS2 version 2.2. (NAND) (C) ↵
2001-2003 Red Hat, Inc.
fat - vfat - at91_wdt - AT91 Watchdog Timer enabled (5 seconds, nowayout=1)
```

```
Starting flash
Starting autoinit Init...
```

```
Starting network Configuring and Starting Network : lo eth0: Link down.
dhcpcd[840]: timed out waiting for a valid DHCP server response
```

```
eth0: Link down.
eth0
```

```
Starting inetd: inetd
Starting httpd
Starting autostart Starting...
```

```
dm9000 Ethernet Driver
eth1: Invalid ethernet MAC address. Please set using ifconfig
eth1: dm9000 at c4806000,c480d004 IRQ 101 MAC: 00:00:00:00:00:00
Configuring eth1 with MAC-addr: 02:80:ad:20:9b:c5
Configuring eth1 with IP-addr: 192.168.1.126
eth1: link down
```

```
- SSV Embedded Linux - Version 0.62.26-ssv2
emblinux login:
```

Listing 1: Boot messages without Ethernet connection

```
Starting network Configuring and Starting Network : lo eth0: Link now 100-↵
FullDuplex
dhcpcd[840]: timed out waiting for a valid DHCP server response
```

```
eth0: Link now 100-FullDuplex
eth0
```

```
Starting inetd: inetd
Starting httpd
Starting autostart Starting...
```

```
dm9000 Ethernet Driver
eth1: Invalid ethernet MAC address. Please set using ifconfig
eth1: dm9000 at c4806000,c480d004 IRQ 101 MAC: 00:00:00:00:00:00
Configuring eth1 with MAC-addr: 02:80:ad:20:9b:c7
Configuring eth1 with IP-addr: 192.168.1.126
eth1: link down
```

```
- SSV Embedded Linux - Version 0.62.26-ssv2
emblinux login:
```

Listing 2: Boot messages with eth0 Ethernet connection


```
Starting network Configuring and Starting Network : lo eth0: Link now 100-↵  
FullDuplex  
dhcpcd[840]: timed out waiting for a valid DHCP server response
```

```
eth0: Link now 100-FullDuplex  
eth0
```

```
Starting inetd: inetd  
Starting httpd  
Starting autostart Starting...
```

```
dm9000 Ethernet Driver  
eth1: Invalid ethernet MAC address. Please set using ifconfig  
eth1: dm9000 at c4806000,c480d004 IRQ 101 MAC: 00:00:00:00:00:00  
Configuring eth1 with MAC-addr: 02:80:ad:20:9b:c7  
Configuring eth1 with IP-addr: 192.168.1.126  
eth1: link down
```

```
- SSV Embedded Linux - Version 0.62.26-ssv2  
emblinux login:
```

Listing 3: Boot messages with eth0/eth1 Ethernet connection

5 HELPFUL LITERATURE

- DIL/NetPC ADNP/9200 hardware reference manual (SSV Starter Kit item)
- Evaluation Board DNP/EVA11 hardware reference manual (SSV Starter Kit item)

CONTACT

SSV Embedded Systems

Heisterbergallee 72

D-30453 Hannover / Germany

Phone: +49 (0)511/40 000-0

Fax: +49 (0)511/40 000-40

E-mail: sales@ist1.de

Internet: www.dilnetpc.com

DOCUMENT HISTORY

Revision	Date	Remarks	Name
1.0	2006-10-24	first version	WBU

The content of this document can change any time without announcement. There is taken over no guarantee for the accuracy of the statements. The user assumes the entire risk as to the accuracy and the use of this document. Information in this document is provided 'as is' without warranty of any kind. Some names within this document can be trademarks of their respective holders.

© 2006 SSV EMBEDDED SYSTEMS. All rights reserved.