

Programming the Nut/OS httpd example application

Application Note 106

Revision 2, October 2008

This Application Note describes how to program the *httpd* example application into the XNUT-100 or XNUT-105 module. It demonstrates how compiled applications are deployed to the XNUT target unit.

Prerequisites

- WinAVR AVR-GCC compiler tool chain installed (Refer to *Application Note 102*)
- Nut/OS installed (Refer to *Application Note 103*)
- Nut/OS has been configured and compiled (Refer to *Application Note 104*)
- If you intend to use the AVR JTAGICE mkII programming adapter you need in addition AVRStudio 4 installed. (Refer to *Application Note 105*)

Preparation

Open the XNUT module and connect the JTAG programmer.



Refer to the *XNUT-100 Hardware Manual* or the *XNUT-105 Hardware Manual*, on how to open the enclosure and how to connect the JTAG programming adapter.

Please pay attention to the following:

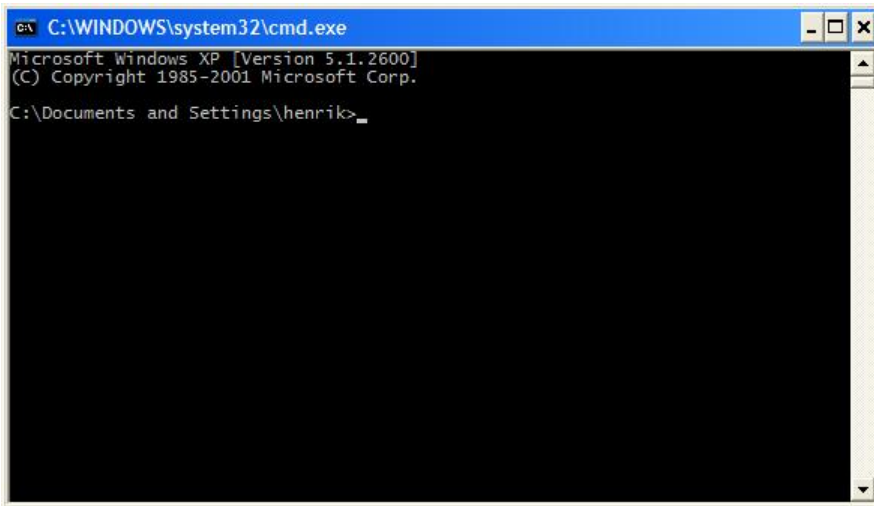
1. Care must be taken when opening the enclosure to avoid breaking the plastic clips holding the top and base shell of the enclosure together. The enclosure opens with ease, do not use force!
2. Follow the electrostatic discharge (ESD) precautions as described in the *XNUT-100 Hardware Manual* or *XNUT-105 Hardware Manual* when working with an open enclosure.
3. Do not plug-in or remove the JTAG programming adapter or touch any electronic parts inside the module while power is supplied to the module or to the programmer. Always switch off programmer first and then the module.

Connect the RS-232 serial port SER0 of the XNUT module using a null-modem cable with your computer. SER0 is the RS-232 port closest to the front of the module. In addition connect the XNUT's Ethernet port to your PC network using a CAT 5 network cable.

Apply power to module and then to the programmer.

Step by step instructions

Launch a command shell (Command Prompt) from the Windows Start menu:



Add the *Nut/OS* tools to your path environment:

```
set PATH=%PATH%;c:\ethernut-4.1.9\nut\tools\win32
```

Change to the *Nut/OS* Application Directory as it was configured during the *Nut/OS* configuration step in the *Nut/OS* Configurator:

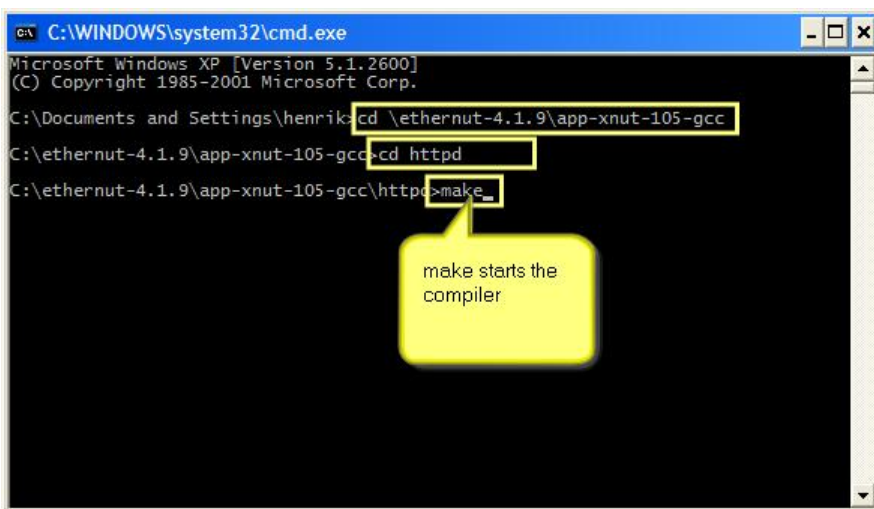
```
cd c:\ethernut-4.1.9\app-xnut-105
```

Change then into the httpd subdirectory, which contains the C source code and makefile of the httpd example:

```
cd httpd
```

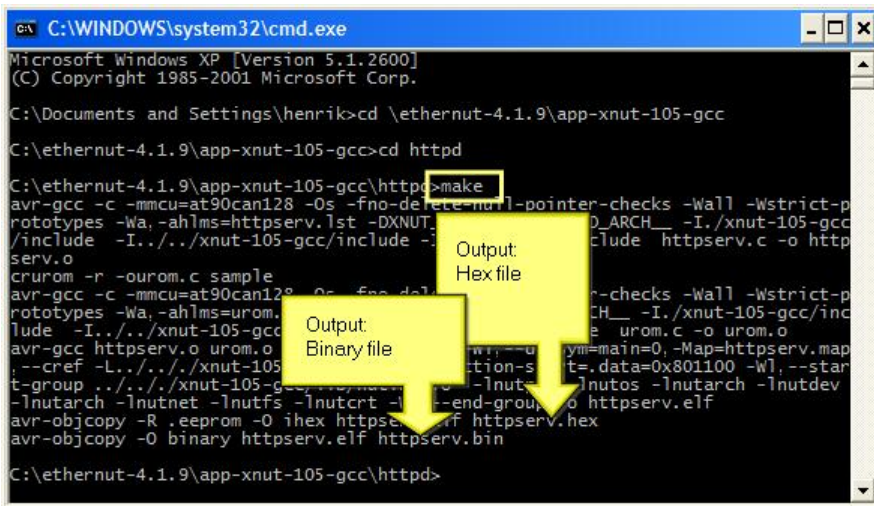
Compile the example application by running the make command:

```
make
```

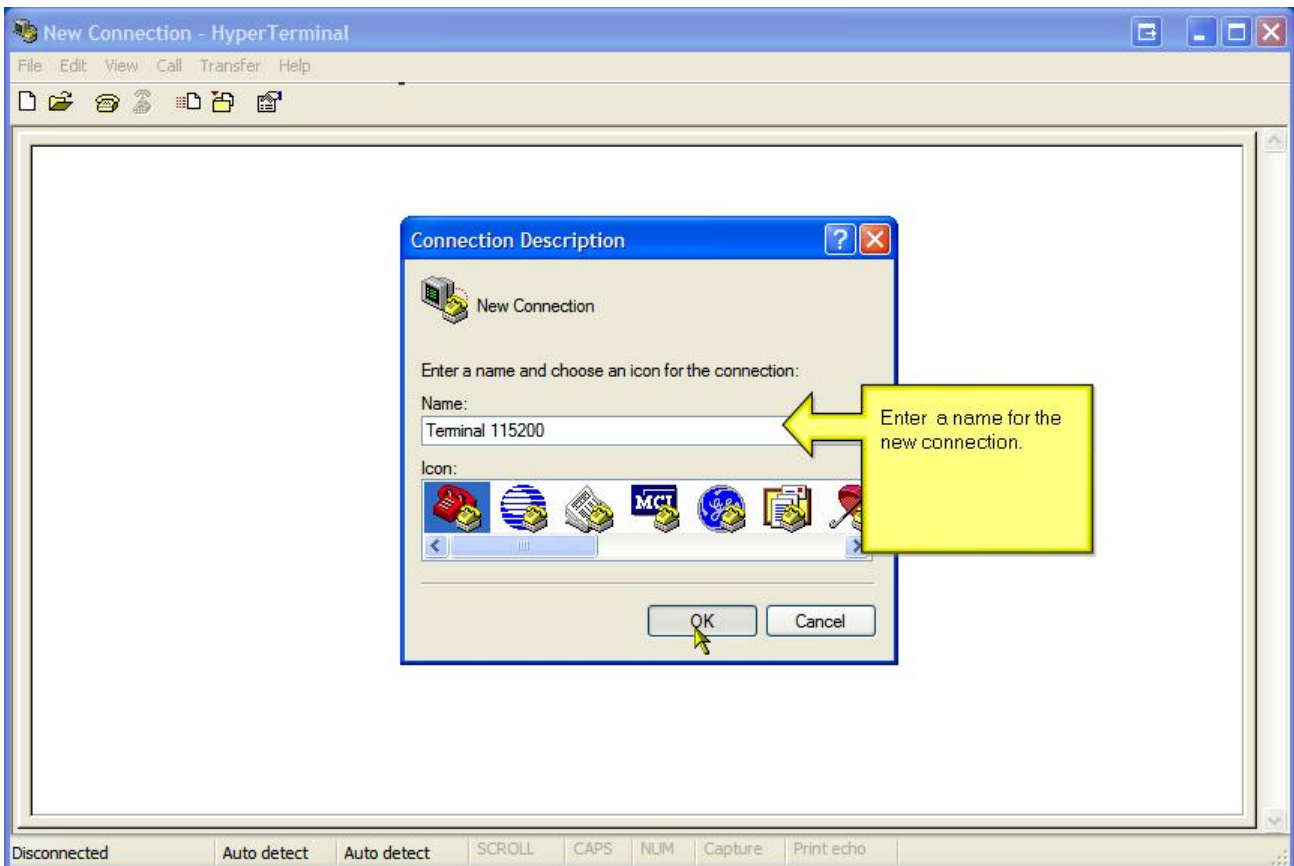


The make process should terminate without any error similar to the following output log.

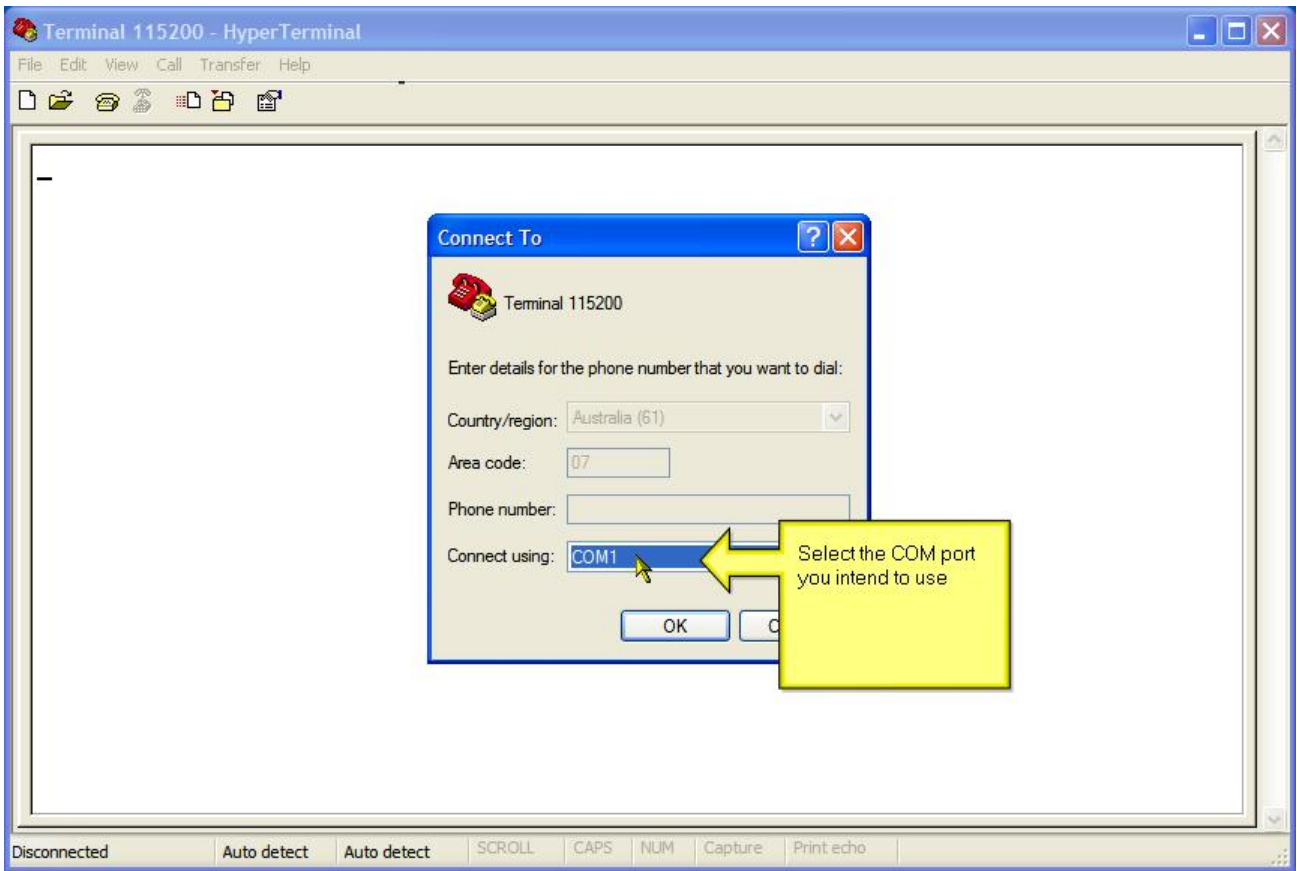
The compiler output which has been created is a so called *Intel-Hex* file and a binary File. The hex file is typically used by programming adapters like the *AVR JTAGICE mkII* or the *SPDuo*. The binary file is used by a TFTP boot loader.



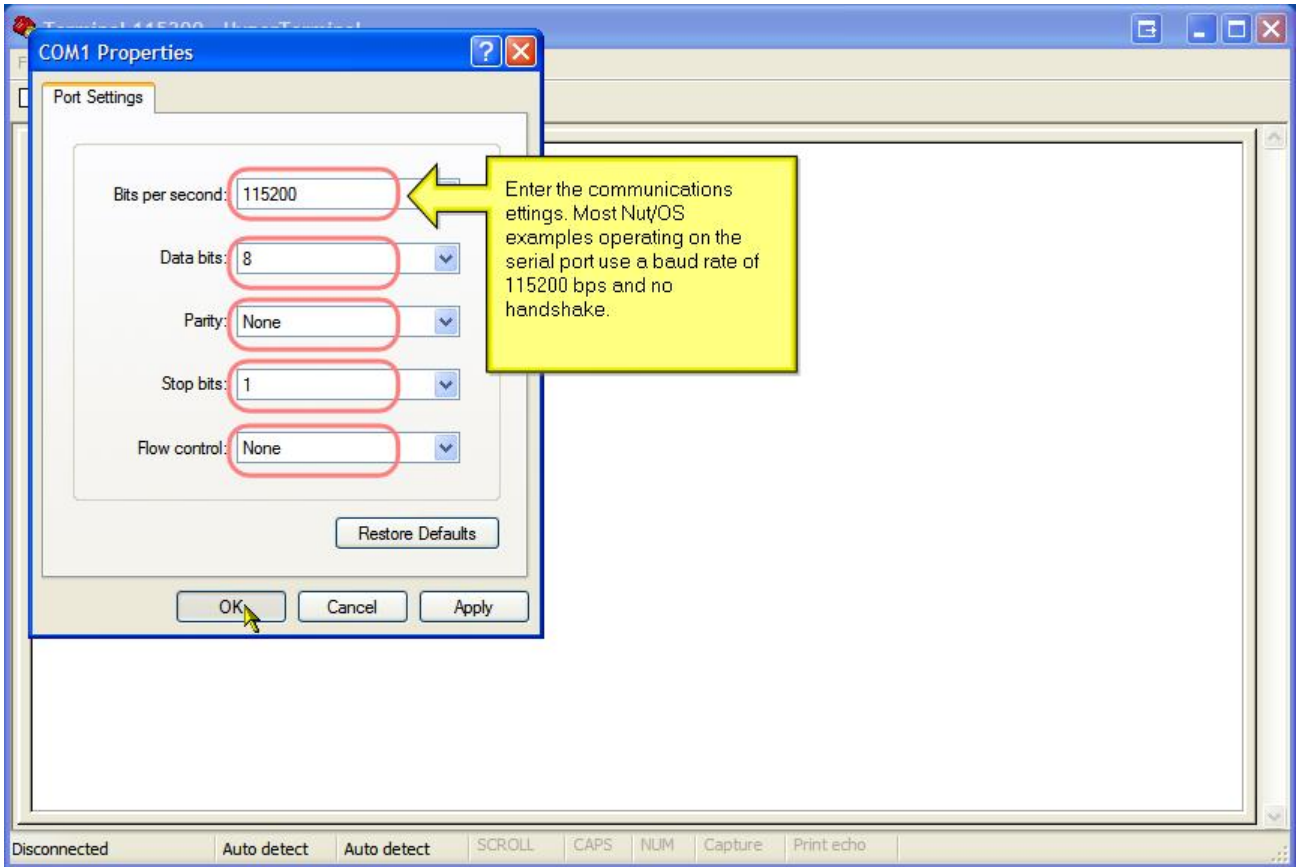
It is now time to launch a serial terminal program. On Windows launch *HyperTerminal* from the Windows Start menu. You find this program under Accessories/Communication. Enter a name for the new connection you are creating.



Select the COM port where you connected SER0 of the XNUT module:



Configure the port settings as shown and click **OK**:



The next step would be to program the generated compiler output into the target module.

For this step it is assumed that the XNUT-100 or the XNUT-105 module has been connected to the AVR JTAGICE mkII programming adapter and that power is applied to programming adapter and to the XNUT module.

The default makefiles offer a pseudo target called burn which can be used to transfer the program into the target module. The type of programmer used for the programming process has previously been configured during the Nut/OS configuration step in the Nut/OS Configurator. For this step it is assumed that avr-jtagicemkii was selected.

Start the programming process by running *make* with the *burn* target:

```
make burn
```

```

C:\WINDOWS\system32\cmd.exe
C:\etheranut-4.1.9\app-xnut-105-gcc\httpd>make
avr-gcc -c -mmcu=at90can128 -Os -fno-delete-null-pointer-checks -Wall -Wstrict-p
rototypes -Wa,-ahlns=httpserv.lst -DXNUT_105 -D__HARVARD_ARCH__ -I./xnut-105-gcc
/include -I../xnut-105-gcc/include -I../nut/include httpserv.c -o http
serv.o
crurom -r -ourom.c sample
avr-gcc -c -mmcu=at90can128 -Os -fno-delete-null-pointer-checks -Wall -Wstrict-p
rototypes -Wa,-ahlns=urom.lst -DXNUT_105 -D__HARVARD_ARCH__ -I./xnut-105-gcc/inc
lude -I../xnut-105-gcc/include -I../nut/include urom.c -o urom.o
avr-gcc httpserv.o urom.o -mmcu=at90can128 -Wl,--defsym=main=0, -Map=httpserv.map
--cref -L../xnut-105-gcc/lib -Wl,--section-start=.data=0x801100 -Wl,--star
t-group ../xnut-105-gcc/lib/nutinit.o -lnutpro -lnutos -lnutarch -lnutdev
-lnutarch -lnutnet -lnutfs -lnutcrnt -Wl,--end-group -o httpserv.elf
avr-objcopy -R .eeprom -O ihex httpserv.elf httpserv.hex
avr-objcopy -O binary httpserv.elf httpserv.bin

C:\etheranut-4.1.9\app-xnut-105-gcc\httpd>make burn
jtagiceii -cauto -dat90can128 -e -pf -ihttpserv.hex
jtagiceii.exe v1.2.2 Copyright (C) Atmel Corporation 2004-2005
Device erased.
Programming FLASH: 100.0%
Programmed 81170 bytes (40585 words) of FLASH memory.

C:\etheranut-4.1.9\app-xnut-105-gcc\httpd>
    
```

After the programming process finishes, the *XNUT* module resets and automatically starts the programmed application. You should see the application's welcome output on your terminal window:

The screenshot shows a HyperTerminal window titled "Terminal 115200 - HyperTerminal". The window contains the following text:

```

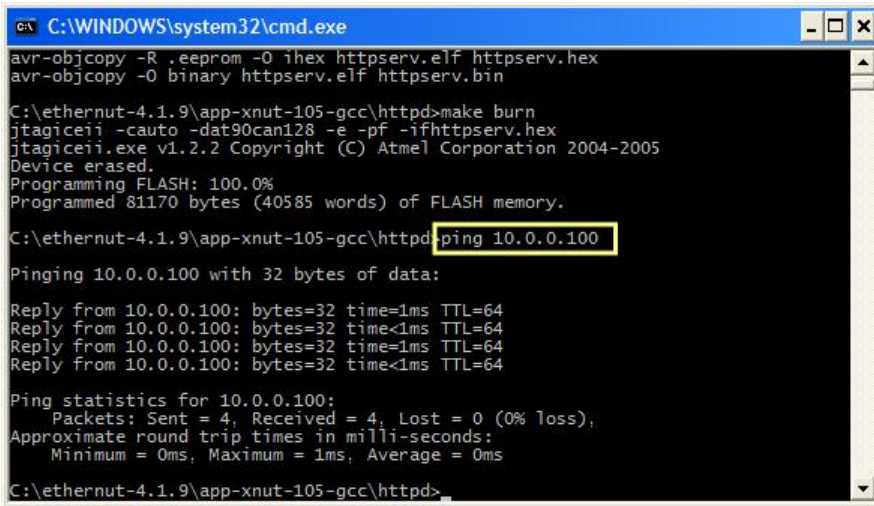
Nut/OS 4.1.9.8 rc HTTP Daemon...10.0.0.100 ready
    
```

At the bottom of the window, there is a status bar with the following information: "Connected 0:00:15", "Auto detect", "115200 8-N-1", "SCROLL", "CAPS", "NUM", "Capture", and "Print echo".

Try to ping the *XNUT* unit by executing the ping utility from the command line. Use the IP address shown as part of the welcome message in the terminal window.

```
ping 10.0.0.100
```

This tests the Ethernet link. If the Ethernet link is active, ping should show successful replies from the unit as shown in the following log:



```
C:\WINDOWS\system32\cmd.exe
avr-objcopy -R .eeprom -O ihex httpserv.elf httpserv.hex
avr-objcopy -O binary httpserv.elf httpserv.bin

C:\ethernut-4.1.9\app-xnut-105-gcc\httpd>make burn
jtagiceii -cauto -dat90can128 -e -pf -ifhttpserv.hex
jtagiceii.exe v1.2.2 Copyright (C) Atmel Corporation 2004-2005
Device erased.
Programming FLASH: 100.0%
Programmed 81170 bytes (40585 words) of FLASH memory.

C:\ethernut-4.1.9\app-xnut-105-gcc\httpd>ping 10.0.0.100

Pinging 10.0.0.100 with 32 bytes of data:

Reply from 10.0.0.100: bytes=32 time=1ms TTL=64
Reply from 10.0.0.100: bytes=32 time<1ms TTL=64
Reply from 10.0.0.100: bytes=32 time=1ms TTL=64
Reply from 10.0.0.100: bytes=32 time<1ms TTL=64

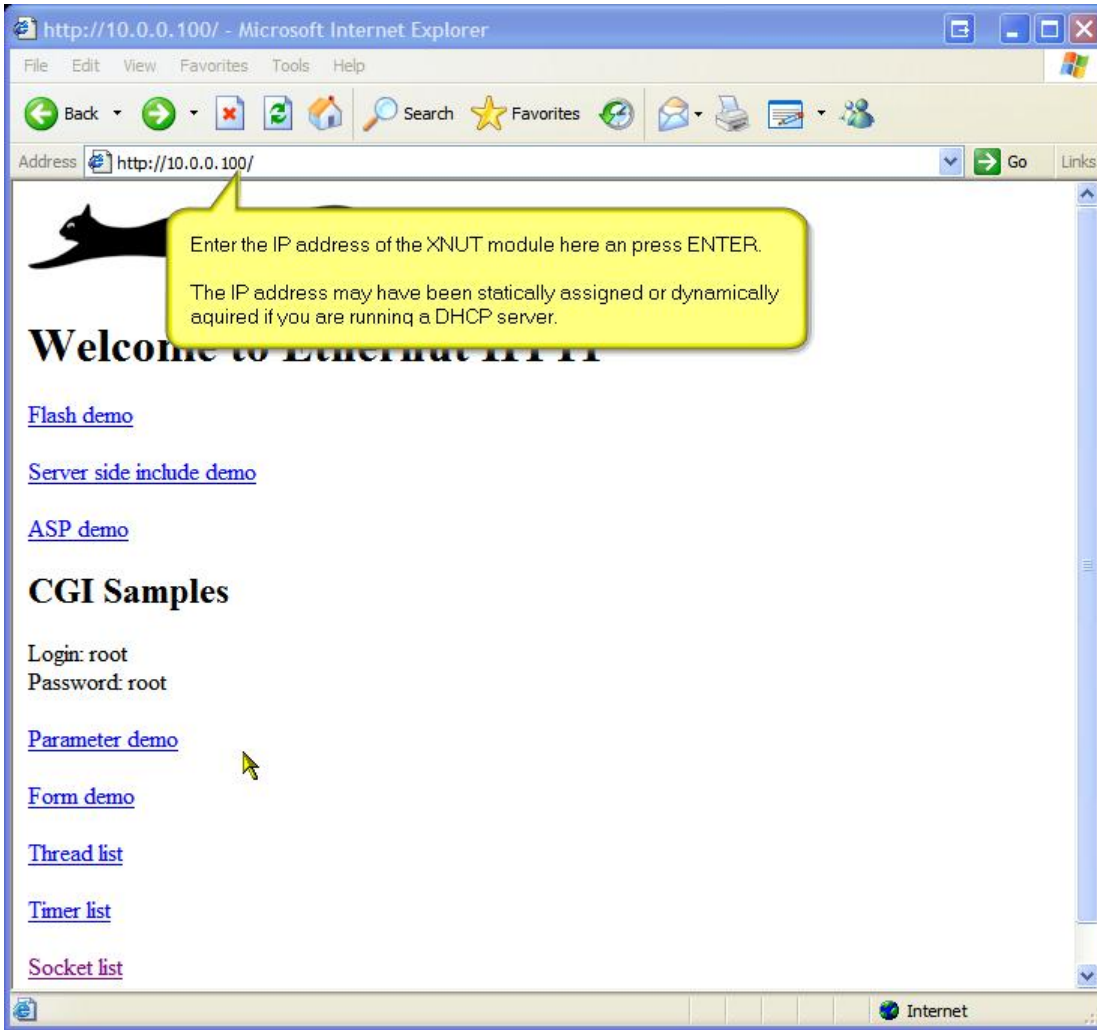
Ping statistics for 10.0.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\ethernut-4.1.9\app-xnut-105-gcc\httpd>
```

If ping fails, check the following:

- Network cable connection between *XNUT* module and switch as well as switch and your PC
- Firewall settings. If in doubt disable the firewall temporarily until the problem has been located.
- Your PC and the module must be on the same subnet. In most cases this means that the first three parts of your PC's and the *XNUT* module's IP address must be identical. Example: *XNUT* IP address is 10.0.0.100, PC IP address is 10.0.0.152.

Start the web browser and enter the *XNUT* module's IP address as URL. The browser should connect and show the following demo home page:



proconX and *FieldTalk* are trademarks of *proconX* Pty Ltd. All other product and brand names mentioned in this document may be trademarks or registered trademarks of their respective owners.

Disclaimer

proconX Pty Ltd makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in the Terms and Conditions located on the Company's Website. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of *proconX* are granted by the Company in connection with the sale of *proconX* products, expressly or by implication. *proconX* products are not authorized for use as critical components in life support devices or systems.

Support & product feedback

We provide an electronic support and feedback system for our *proconX* products. It can be accessed through the following web link:

<http://www.proconx.com/support>

Your feedback and comments are always welcome. It helps improving this product.

Contact

For further information about this document please contact us at:

proconX Pty Ltd
PO Box 791
Sumner QLD 4074
Australia
Website: <http://www.proconx.com>