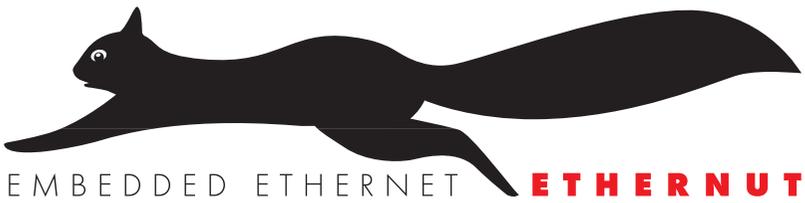


Ethernut 2
Hardware Manual



Version 1.0

Copyright 2003 egnite Software GmbH. All rights reserved.

egnite makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

egnite products are not intended for use in medical, life saving or life sustaining applications.

egnite retains the right to make changes to these specifications at any time, without notice.

All product names referenced herein are trademarks of their respective companies. Ethernut is a registered trademark of egnite Software GmbH

Contents

1 About the Ethernet 2 Board	1
Ethernet Features	1
2 Quick Start	3
Prerequisites for Operation	3
Board Installation	4
3 Testing the Board	5
Ethernet Controller Read/Write Loop	5
Find Ethernet Controller Loop	6
Ethernet Controller Hardware Reset Loop	6
SRAM Read/Write Loop	6
Send Broadcasts Loop	6
Exit BaseMon	6
4 Network Configuration	7
DHCP/BOOTP Method	7
ARP Method	7
Reset to Factory Default	8
Manual Configuration	8
Testing Network Operation	8
5 Functional Description	9
Jumper Overview	9
Power Supply	10
Expansion Port	11
Serial Ports	12
Ethernet Port	14
In-System Programming and JTAG Port	15
LED Indicators	16
Watchdog Timer	17
System Clock	17
Flash ROM	17
Static RAM	17
EEPROM	17
6 Troubleshooting	18
Problem Solution	18
7 Schematics	19
8 Index	23

1 About the Ethernet 2 Board

Low-cost Ethernet capability can be added to many embedded applications.

Ethernet 2 is a small (80 x 100 mm) board combining Atmel's ATmega128 RISC microcontroller with SMSC's LAN91C111 Ethernet controller. The board is well suited for application development in a wide range of applications.

Some areas are:

- Networked sensors
- Remote monitoring equipment
- Alarm service providing
- Remote diagnose and service
- Industrial Ethernet applications
- Home and building control

Ethernet Features

The Ethernet board provides the following features:

- ATmega 128 RISC microcontroller with up to 16 MIPS throughput
- Full duplex IEEE 802.3 compliant 10/100 Mbps Ethernet controller with on-board RJ-45 connector
- Two RS-232 serial ports, one with on-board DB-9 connector
- 128 Kbyte in-system programmable flash ROM
- 4 Kbyte in-system programmable EEPROM
- 512 Kbyte SRAM
- 22 programmable digital I/O lines
- 8-channel, 10-bit analog/digital converter
- Two 8-bit and two 16-bit timer/counters
- Watchdog timer for enhanced reliability
- LED indicators for power supply and Ethernet activity
- Single power supply DC 8-12V

2 Quick Start

This chapter will help you quickly set up and start using the Ethernet board.

Prerequisites for Operation

The following hardware items are necessary to run the Ethernet board:

- A standard PC equipped with Linux or Windows 95/98/NT/2000, an available serial COM port and a twisted pair Ethernet adapter card.
- Terminal emulation software, such as Hyperterminal.
- An unregulated power supply matching your local mains. It should supply DC 8-12V, 500 mA minimum, on a standard 2.1 mm barrel connector.
- Two straight-through twisted pair cables together with 100 or 10 Base-T hub or switch or a twisted pair cross cable, if you don't got a hub or switch.

The following items are included in the Ethernet Starter Kit:

- An STK-500 compatible programming adapter.
- A straight through serial communication cable with a DB-9 female on one end and a DB-9 male connector on the other.

Board Installation

- 1 Remove the board from the antistatic bag. Visually inspect the board to verify that it was not damaged during shipment.



WARNING: As with all computer equipment, the Ethernut board may be severely damaged by electrostatic discharge (ESD). Be sure to take proper precautions before removing the Ethernut-board from the anti-static bag.

- 2 Connect Ethernut's DB-9 RS232 port to an available COM port using the serial cable.
- 3 Use one twisted pair cable to connect Ethernut's RJ-45 connector to the hub or switch and the other twisted pair cable to connect the hub or switch with the network adapter in the PC. If you are not using a hub or switch, then directly connect the Ethernut board with the network adapter using a twisted pair cross cable.
- 4 Connect the power supply to the barrel connector on the Ethernut board. The Ethernut board is equipped with its own rectifier bridge and voltage regulator. Therefore the polarity of the barrel connector isn't important.



WARNING: The power supply should not be plugged into an electrical outlet before connecting it to the Ethernut board.

- 5 Apply power to the Ethernut board by connecting the power supply to an electrical outlet. When the board is powered up, the red power LED (LED1) should go on.
- 6 Start the terminal emulation program at 38400 baud or any higher rate up to 115200 baud, no parity, 8 data bits, and 1 stop bit. Disable hardware (RTS/CTS) and software (XON/XOFF) flow control.

Baudrate

The baudrate of the Ethernut board is specified by the CPU crystal (Q1, 14.7456 MHz by default) and a baudrate selector ranging from 0 to 255. The actual baudrate can be calculated by

$$\text{baudrate} = \text{crystal frequency} / (16 * (\text{selector} + 1))$$

Running at 14.7456 MHz, a selector value of 23 gives a baudrate of 38400 Baud:

$$38400 = 14745600 / (16 * (23 + 1))$$

The BaseMon program provides a simple automatic baudrate selection by changing the selector from zero to 71, while trying to receive a space character. If no space character could be received within about 2 minutes, then the default selector 23 is set (38400 Baud at 14.7456 MHz).

- 7 Reset the Ethernut board by depressing and releasing the reset switch located near the SMC chip. Hold down the spacebar on the terminal emulation program and wait until the BaseMon welcome message is displayed.

See the next chapter for a detailed description of the BaseMon program.

3 Testing the Board

Using the preloaded BaseMon firmware to test the Ethernet hardware.

When using a terminal emulation program like described in the previous chapter, hold down the spacebar on the PC keyboard after pressing the reset button on the Ethernet board. After some seconds the following output should appear in the terminal emulation window:

```
BaseMon 4.0.2
Nut/OS 3.3.2.1
Baudrate select = 23
External RAM Test... 44800 bytes
Banked RAM Test... 30 banks, 16384 bytes ea.
Detecting NIC... LAN91C111
Testing NIC... OK
I/O Port Test... OK

Press any of the following keys:
B - Send broadcasts
E - Ethernet controller read/write
J - Jump to bootloader
S - SRAM read/write
X - Exit BaseMon, configure network and start WebServer
```

The baudrate selector may differ, depending on the CPU crystal (Q1) on the Ethernet board and the baudrate used with the terminal emulation program. If any of the above tests fails, you may try one of the keys shown in the BaseMon menu to enter a specific test loop. If after a few minutes the Ethernet board doesn't receive a space character, it will skip the BaseMon program and initialize the Nut/OS operating system as well as the Nut/Net TCP/IP stack and finally enter a sample HTTP daemon application.

Ethernet Controller Read/Write Loop

When pressing E on the BaseMon menu, the Ethernet board will enter an endless loop, trying to read the revision ID of the Ethernet controller at base address C000 hex:

```
rev=0x91
```

The loop keeps running until a key is pressed in the terminal emulation program and may be used to check the board's address and data bus signals with an oscilloscope or logic analyzer.

Jump Bootloader

When pressing J on the BaseMon menu, the program will jump to flash memory location 1F000 hex. Fully assembled Ethernut boards are delivered with a preloaded bootloader, which uses DHCP/BOOTP/TFTP to load a new flash ROM image. Note, that the bootloader will be deleted by the chip erase command. Thus, if you uploaded BaseMon with an ISP adapter, there will be no bootloader available.

SRAM Read/Write Loop

When pressing S on the BaseMon menu, the Ethernut board enters an endless loop, doing a walking bit test on all address and data bus lines. The loop keeps running until a key is pressed in the terminal emulation program and may be used to check the board's address and data bus signals with an oscilloscope or logic analyzer.

Send Broadcasts Loop

When pressing B on the BaseMon menu, the Ethernut board will initialize the Ethernet Controller and start sending Ethernet broadcasts in an endless loop. The yellow link LED will lit and the green activity LED will start flashing. The terminal emulation window will show the progress.

The loop keeps running until a key is pressed in the terminal emulation program and may be used to check the board's Ethernet output with an oscilloscope.

Exit BaseMon

Pressing X on the BaseMon menu will quit the BaseMon program, initialize the Nut/OS operating system and Nut/Net TCP/IP stack and finally enter a sample HTTP daemon application. However, before that is done, BaseMon queries a MAC address, IP address, network mask and default route:

```
MAC address (000698000000):  
IP address (0.0.0.0):  
Net mask (255.255.255.0):  
Default route (0.0.0.0):
```

The last six digits of the MAC address are written on the board. Enter these six digits on the MAC address prompt. On all prompts, you may simply press enter to confirm the default shown in brackets, or enter other values in their decimal dotted form. If the IP address is 0.0.0.0, Ethernut will not query the network mask and default route, but request these values from a DHCP server. This requires of course, that a DHCP server is running in your local network.

Network configuration is discussed in more detail in the next chapter.

4 Network Configuration

This chapter shows different methods to configure Ethernet's network parameters.

In order to communicate over a TCP/IP network, the Ethernet board needs a unique IP address. It is important, that this address is not used by any other node on the network.

DHCP/BOOTP Method

The Ethernet board ships with a blank EEPROM. If a DHCP server exists on the network, the Ethernet board will automatically request its IP address, the IP address of the standard gateway, and the IP address mask of the local network. If no DHCP server could be located, the Ethernet board switches to the ARP method.

ARP Method

If the Ethernet's EEPROM contains no configuration data and no DHCP server is available on the network, then the ARP method can be used to set the board's IP address. In this mode the Ethernet board set its address from the first ICMP packet it receives.

To set the Ethernet's IP address by the ARP method, an ARP entry can be manually created on the PC and then a ping packet is sent from the PC to the Ethernet board.

Enter the following command to manually create an ARP entry for an Ethernet board with a MAC address of 00:06:98:00:00:00 and an IP address of 192.168.171.5 on a LINUX command line shell:

```
arp -s 192.168.171.5 00:06:98:00:00:00
```

On a Windows DOS prompt this command is slightly different:

```
arp -s 192.168.171.5 00-06-98-00-00-00
```

The next command to enter is the same on both systems:

```
ping 192.168.171.5
```

The first ping packet that arrives at the Ethernet board with the MAC address of 00:06:98:00:00:00 sets the IP address of that board to 192.168.171.5. Note, that the ARP method will not configure a default gateway and will fix the network mask to 255.255.255.0.

A MAC address, also referred to as the hardware or Ethernet address is a unique 48 bit number assigned to every Ethernet node. The upper 24 bits are the manufacturer's ID, assigned by the IEEE Standards Office. The ID of Ethernet boards manufactured by egnite Software GmbH is 000698 hexadecimal. The lower 24 bits are the board's unique ID assigned by the manufacturer of the board. Boards produced by egnite do have a unique ID, which is written on the board.

Reset to Factory Default

The ARP method will be used on blank EEPROMs only. After having set it once, the configuration will be stored in the EEPROM and used in the next system start. To enable the ARP method again, you must use your ISP Software to clear the EEPROM contents. Refer to the Ethernut Software Manual for further information about ISP Software.

Manual Configuration

The preloaded BaseMon firmware may be used to manually configure the boards IP address and network mask. These settings will be stored in the EEPROM and will be preserved when uploading a new firmware into the Ethernut Board via ISP or JTAG, when the EESAVE has been programmed (set to 0).

Testing Network Operation

You can now check, that the Ethernut board is properly hooked up to the network by running ping from your PC. On a DOS prompt or command line shell, type:

```
ping 192.168.171.5
```

Instead of the above IP address use the one you configured previously. If you receive a response without timing out, the Ethernut board is ready to try the HTTP daemon.

Use any Webbrowser to query the following URL:

```
http://192.168.171.5/index.html
```

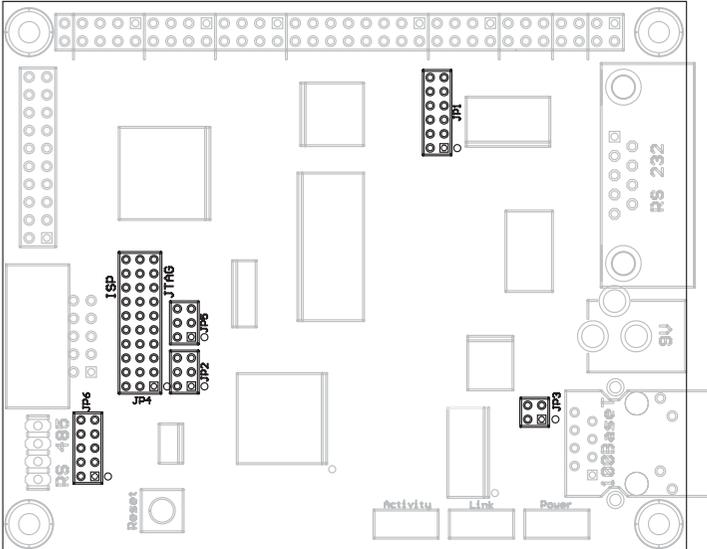
Again, instead of the above IP address use the one previously configured. MAC address

5 Functional Description

This chapter explains the main functions of the Ethernetut board.

Jumper Overview

For maximum flexibility, the board is equipped with 6 jumper areas:



JP1 configures the connection to the two serial devices (UARTS),

JP2 is used to select the port bit of the RS485 direction signal.

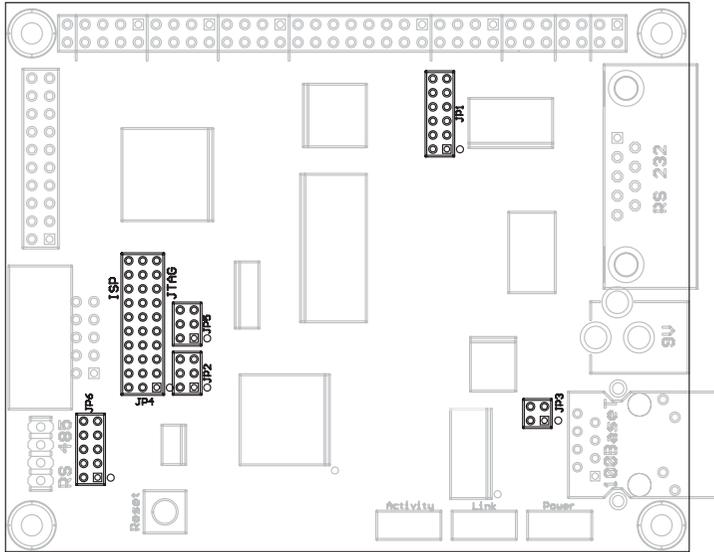
JP3 enable power supply over Ethernet cable.

JP4 selects the programming adapter mode, either ISP or JTAG.

JP5 is used to specify the JTAG chain.

JP6 configures the RS485 screw terminal.

Power Supply

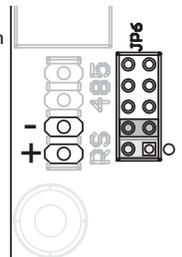


The complete logic of the Ethernut board is driven by 5V and 3.3V power supply. The board provides its own voltage regulators (IC8 and IC9). It only requires an unregulated power supply of DC 8-12V with a minimum current of 500 mA.

Three different methods may be used to connect an external power supply.

1 A standard 2.1 mm barrel connector. This input is protected by a fuse (F1), a fast transient voltage suppressor (D3) and a rectifier bridge (B1).

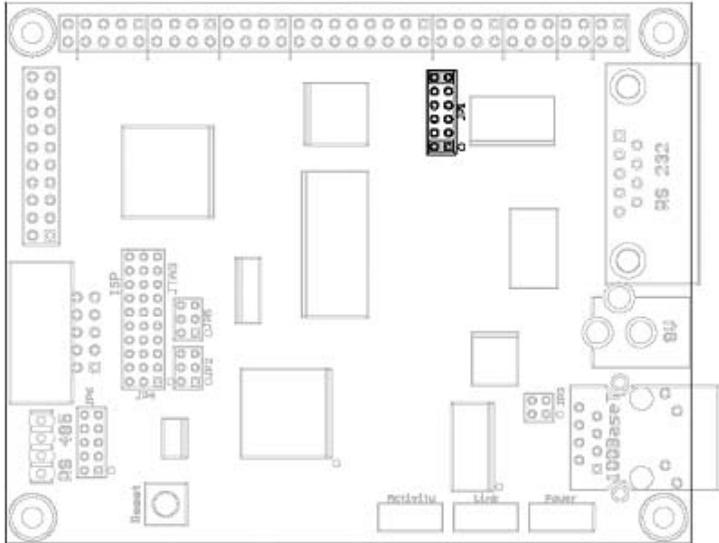
2 A screw terminal. Pins 1 and 2 and pins 2 and 3 of jumper JP6 must be shortened. When both jumpers are set, this is an unprotected input. Applying a wrong polarity or a too high voltage will destroy the Ethernut board. Typically this is used to supply external RS485 devices, in which case only pins 2 and 3 are shortened, but not pins 1 and 2



Serial Ports

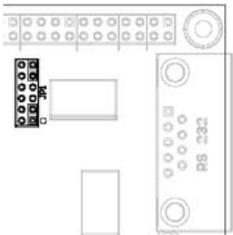
Ethernet provides an on-board DB-9 connector for RS-232 serial communication as well as a screw terminal for RS-485 half duplex communication. IC6 is used to convert the required voltage levels for RS-232 from the 5V power supply, while IC7 provides the signal conversion for the RS-485 interface.

Both, the RS-232 or the RS-485 connector may be connected to the first (UART0) or the second (UART1) serial device of the ATmega128. This is configured by jumper JP1.

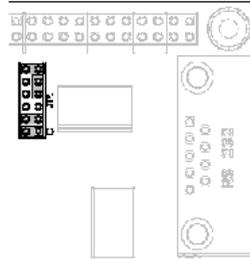


Shortening pins 1 and 2 and pins 11 and 12 will route UART0 transmit and receive lines to the DB-9 connector.

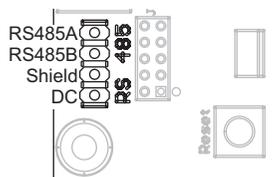
When pins 1 and 3 and pins 9 and 11 are shortened, then UART1 is connected to the DB-9 connector



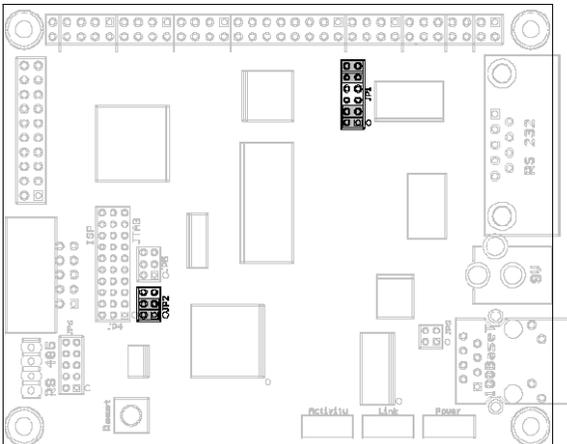
Even both UARTs can be routed to the DB-9 connector when shortening pins 1 and 2, 2 and 3, 7 and 9 and 11 and 12. However, a special adapter is required to route the UART1 transmit line from the onboard DB-9 pin 7 to a second DB-9 pin 3 and the receive line from the onboard DB-9 pin 8 to pin 2 of the second connector.



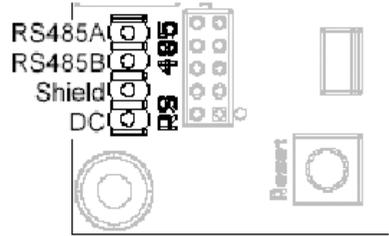
If your application requires hardware handshake via RTS/CTS, then connect JP1 pins 4 and 6 and pins 5 and 7.



Most applications will probably use UART0 for RS-232 and UART1 for RS-485. This configuration is shown below. Pins 1 and 2, 3 and 4, 9 and 10 and 11 and 12 of JP1 are shortened.

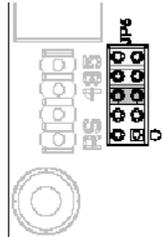


The RS-485 interface is half duplex. Only two wires are needed for communication, but an additional port bit is required to switch between receive and transmit mode. Thus pins 1 and 2 as well as pins 3 and 4 of JP2 must be shortened to control the data direction via ATmega port bit PD5. Shortening pins 5 and 6 instead of pins 1 and 2 will use port bit PD4.



The direction signal is pulled low by R27. If pins 3 and 4 are shortened only, the RS-485 interface is configured as a receiver and no port bit for direction control is required.

Both ends of a RS-485 communication line must be terminated by a resistor. If the Ethernet board is located at one end of the line, you must short pins 5 and 6 of JP6 to enable the 120 Ohm termination resistor (R24).



During switching from transmit to receive mode, the condition of the RS-485 line becomes undetermined and may introduce problems on connected receivers. Shortening pins 7 and 8 and pins 9 and 10 will enable two 560 Ohm failsafe resistors (R23 and R25). These resistors should be enabled on one end only.

Pin 3 of the screw terminal is connected to Ethernet's signal ground via a 100 Ohm resistor (R26) and can be used to connect a cable shield. Alternatively pins 3 and 4 of the screw terminal may be used to provide power to the external RS-485 device. In this case shorten pins 3 and 4 of JP6 to disable R26 and connect screw terminal pin 3 directly to ground. Pin 4 of the screw terminal is connected to the Ethernet's unregulated DC line via a diode (D2).

Ethernet Port

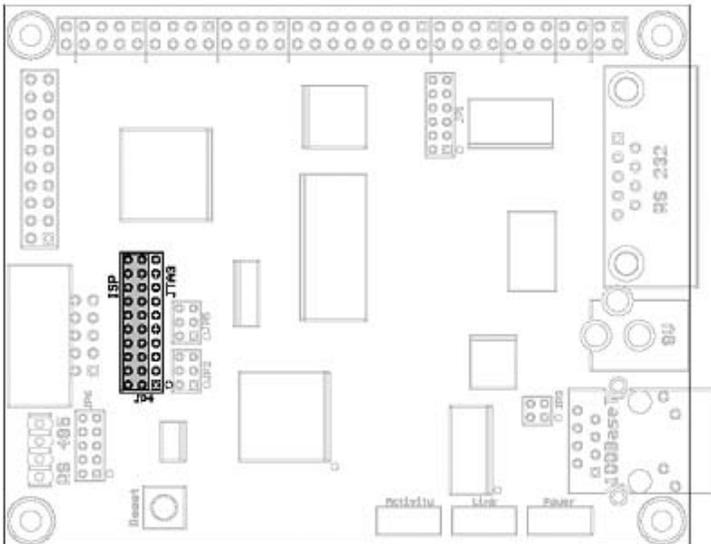
Ethernut provides an on-board modular RJ-45 connector for its twisted pair Ethernet port. This port is connected to the SMSC LAN91C111 Ethernet controller via a 100/10Base-T transformer/filter. The interface supports the maximum cable length of 100 meters between the Ethernet board and a hub.

In-System Programming and JTAG Port

This 10-pin header port allows serial programming of the ATmega128 non-volatile Flash ROM and EEPROM as well as the CPLD without physical removal of the chips from the system.

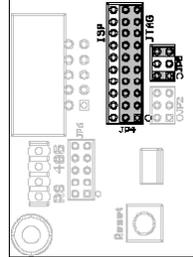
Ethernet 2 offers two methods of in-system programming, via ISP-Adapter or via JTAG. While the JTAG interface can be used to program both, the ATmega128 and the CPLD, the ISP method is available for the ATmega128 only. In addition, in circuit debugging is provided by the JTAG interface.

Jumper JP4 is used to select between ISP and JTAG. If the 20-pin jumper is inserted towards the ISP/JTAG connector, ISP mode is selected.

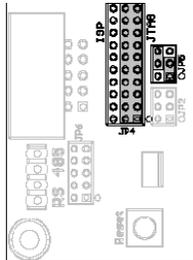


WARNING: When using a parallel port adapter, do not plug in or remove the programming adapter while power is supplied to the Ethernet board. You may damage the Ethernet board or the parallel port on the PC.

Moving the 20-pin jumper on the other side of JP4 enables the JTAG interface. JP5 is then used to route the JTAG signals to the ATmega128, if pins 1 and 2 and pins 5 and 6 are shortened.



Routing the JTAG interface to the CPLD is provided by shortening pins 1 and 3 and pins 4 and 6 of JP5. It is also possible to daisy chain the JTAG signals through both chips, but there's currently no programming software supporting this.



LED Indicators

The Ethernut 2 board is equipped with three LEDs. For historical reason, these LEDs are LED1, LED3 and LED4.

One red LED (LED1) is directly connected to the power supply. It is lit when power is applied to the board.

A yellow and a green LED (LED3 and LED4 resp.) are used to indicate activity on the Ethernet port. The yellow LED indicates the 10BASE-T link status and should be lit, if the link status is OK. The green LED indicates receive and transmit activity from and to the network.

Watchdog Timer

Software bugs, temporary hardware failures caused by electrical transients or interference and many other problems might cause the system to malfunction. The ATmega128 microcontroller (IC1) provides an on-chip watchdog timer, which forces a system reset, if the application program fails to periodically update this timer.

System Clock

The ATmega 128 microcontroller clock is generated by a 14.7456 MHz crystal (Q1), which may be replaced by a crystal of up to 16 MHz. An additional 32.768 kHz crystal (Q2) drives an on-chip asynchronous timer, which is typically used for a software realtime clock. The Ethernet controller is driven by a separate 20-MHz crystal (Q3).



WARNING: Note, that changing any crystal will alter the Ethernet board's EMC characteristics and require re-testing.

Flash ROM

The ATmega 128 provides 128 kBytes of on-chip, non-volatile flash memory space, which is used for program code and read-only data storage. This memory is organized as 64K x 16 bits and can be (re-)programmed through in-system programming.

Static RAM

The Ethernet board provides 512 kByte SRAM (IC4), which is used as read/write data storage. The lower 4 kBytes are overlaid by the ATmega128 internal register and SRAM space. The required address latch is provided by the CPLD (IC3). With its 16 bit address bus the ATmega128 can address upto 64 kByte RAM only. The lower 32 kByte on the Ethernet board are fixed.

EEPROM

The ATmega 128 provides 4 kBytes of on-chip, non-volatile, electrically erasable memory, typically used for configuration data storage. This memory provides read/write access under program control as well as through insystem programming. Note, that EEPROM write access is much slower (about 2.5 ms) than writing to SRAM.

6 Troubleshooting

This chapter will help you in case of problems.

Problem Solution

Problem

The red power LED is not on.

Nothing happens when pushing the RESET button.

The terminal emulation on the PC does not show the BaseMon software prompt.

The terminal emulation software displays unreadable characters

The AVR device cannot be programmed.

Fix

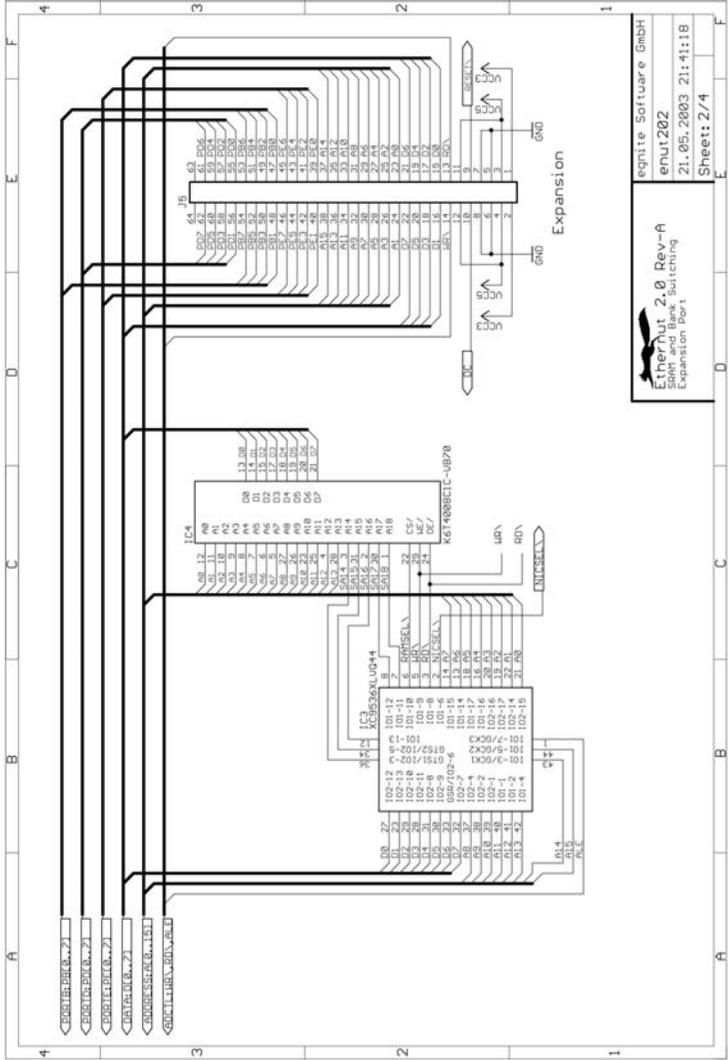
Connect the DC power cable. Check that the power supply is of DC type 8-12V min. 500 mA. Check the fuse.

Sometimes it is difficult to make a good connection when pushing the small button. Try pushing it in the center.

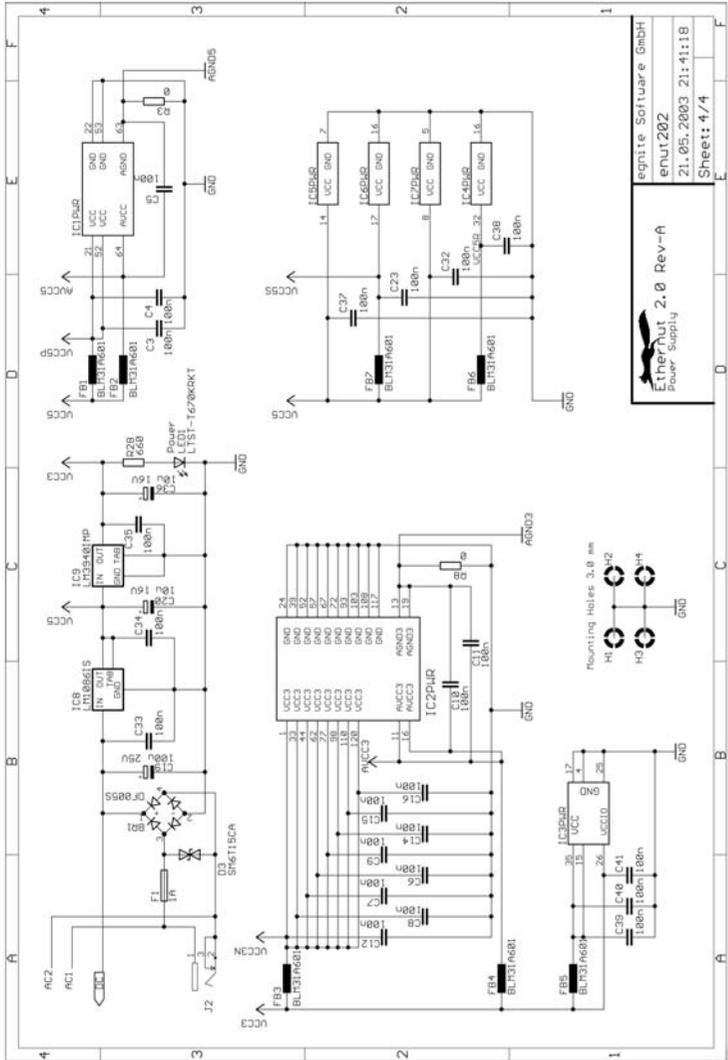
Check the COM port connection with the Ethernet board. Make sure that the same COM port is selected in the terminal emulation software. Verify the check the cables, etc. Try using a different baudrate and make sure, that the terminal emulation repeatedly transmits space characters while the spacebar is held down. If not sure, try another program. Make sure, that any flow control is switched off and jumper JP1 is properly set.

Check the word length, number of stop bits and parity setting of the terminal emulation software. It should be set to 8 data bits, no parity and 1 stop bit.

You need to completely erase the device before programming.



egnite Software GmbH
 emut202
 21.05.2003 21:41:18
 Sheet: 2/4



Index

A About the Ethernet 2 Board	1
B Quick Start	3
C Testing the Board	5
D Network Configuration	7
E Functional Description	9
F Troubleshooting	18
G Schematics	19



egnite Software GmbH
Westring 303
44629 Herne
Germany

Fon +49 (0)23 23-92 53 75
Fax +49 (0)23 23-92 53 74
Email info@egnite.de

<http://www.egnite.de>
<http://www.ethernut.de>