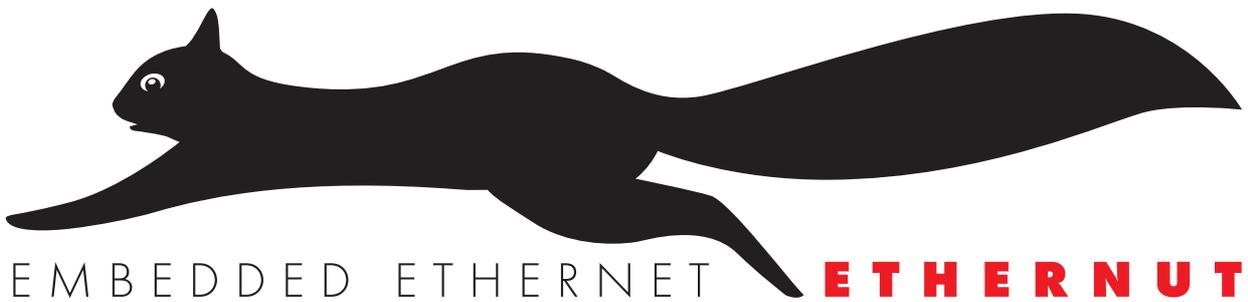


**Volume 1**  
**Hardware Manual**



Version 1.6

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

<b>1 About the Ethernut Board</b>	1
Ethernut Features	1
<b>2 Quick Start</b>	3
Prerequisites for Operation	3
Board Installation	4
<b>3 Testing the Board</b>	7
Ethernet Controller Read/Write Loop	7
Find Ethernet Controller Loop	8
Ethernet Controller Hardware Reset Loop	8
SRAM Read/Write Loop	8
Send Broadcasts Loop	8
Exit BaseMon	8
<b>4 Network Configuration</b>	11
DHCP/BOOTP Method	11
ARP Method	11
Reset to Factory Default	12
Manual Configuration	12
Testing Network Operation	12
<b>5 Functional Description</b>	13
Power Supply	13
Expansion Port	14
Serial Port	15
Ethernet Port	15
In-System Programming Port	16
LED Indicators	16
System Reset	16
Watchdog Timer	17
System Clock	17
Flash ROM	17
Static RAM	17
EEPROM	18
<b>6 Troubleshooting</b>	19
Problem Solution	19
<b>7 Schematics</b>	21
<b>8 Board Layout</b>	25



# 1 About the Ethernet Board

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

Ethernut is a small (80 x 100 mm) board combining Atmel's ATmega 128 RISC microcontroller with Realtek's 8019AS 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

## Ethernut Features

The Ethernut board provides the following features:

- ATmega 128 RISC microcontroller with up to 16 MIPS throughput
- Full duplex IEEE 802.3 compliant 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
- 32 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 and reset controller for enhanced reliability
- LED indicators for power supply, programming mode and Ethernet activity
- Single power supply DC 8-16V



## 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 8-16V, 100 mA minimum, on a standard 2.1 mm barrel connector.
- Two straight-through twisted pair cables together with 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. Make sure, that jumper JP1 has been properly set. Jumper JP2 configures the RTS/CTS lines, but this feature is not used by the BaseMon program.

### **Jumper Settings**

There are two jumpers on the Ethernut board near the DB-9 connector. If you place the board in front of you with the LEDs on the lower left and the jumpers on the upper right corner, JP1 is the left one and JP2 the one nearer to the DB-9 connector on the right.

JP1 is used to switch the serial port from DCE to DTE. Using a straight through serial cable between the Ethernut board and a PC requires DCE setting, in which case the two upper pins of JP1 should be shortened by one jumper and the two lower pins by another (holding the board as specified above).

JP2 needs not to be equipped with any jumper.

- 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.

### **WARNING**

The power supply must 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) and the green Ethernet activity LED (LED4) should go on. The Ethernut board is equipped with its own rectifier bridge and voltage regulator. Therefore the polarity of the barrel connector isn't important.

- 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 Ethernet 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 Ethernet board by depressing and releasing the reset switch located near the LEDs. 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 Ethernut 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 Ethernut board. After some seconds the following output should appear in the emulation window:

```
Ethernut BaseMon Version 2.4.5
Baudrate select = 23
External SRAM Test...OK
NIC hardware reset...OK
NIC id detection... OK
I/O Port Test... OK
Press any of the following keys:
B - Send broadcasts
E - Ethernet controller read/write
F - Find Ethernet controller
R - Ethernet controller hardware reset (Ethernut 1.0/1.1)
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 Ethernut 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 Ethernut 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 Ethernut board will enter an endless loop, trying to read the id of the Ethernet controller at base address 8300 hex:

```
id=0x5070
```

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.

## Find Ethernet Controller Loop

When pressing F on the BaseMon menu, the program tries to locate an Ethernet controller in the address range from 8000 hex to FFE0 hex. After performing a software reset on the controller, its internal ID is checked.

If a valid ID of 5070 hex can be found, the following line is displayed:

```
NIC 8300 reset...OK id=0x5070 OK
```

The loop keeps running until a key is pressed in the terminal emulation program.

## Ethernet Controller Hardware Reset Loop

This option is for Ethernut Board prior to version 1.3 only. When pressing R on the BaseMon menu, the Ethernut board toggles bit 4 on port E in an endless loop. This port bit is tied via an inverter gate to the Ethernet controller's hardware reset line. The interval is about one second on a 3.6864 MHz CPU clock. The green LED (LED4) should flash in the same interval.

## 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:

```
SW-Reset...OK  
RTL8019AS jumper mode: IRQ0 Auto
```

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 Ethernut 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.

### **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.

## 5 Functional Description

This chapter explains the main functions of the Ethernet board.

### Power Supply

The complete logic of the Ethernet board is driven by a single 5V power supply. Because the board provides its own on-board rectifier bridge (B1) and voltage regulator (IC8), it only requires an unregulated power supply of DC 8-16V with a minimum current of 100 mA.

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

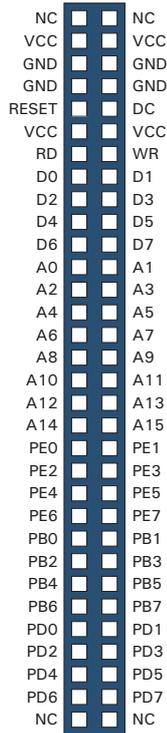
- 1 A standard 2.1 mm barrel connector.
- 2 An optional screw terminal, which is not mounted in the standard version.
- 3 Using pins 4, 5 and pins 7, 8 of the Ethernet connector. In this case pins 1 and 3 and pins 2 and 4 of jumper JP3 must be shortened.

The unregulated power supply as well as a regulated 5V supply is available at the expansion port connector. If additional components draw more than 150 mA from the regulated 5V power supply, a heat sink will be required on the on-board regulator.

## Expansion Port

External devices can be added to the expansion port. These devices may be simple I/O circuits driven by the Ethernut board, or may be equipped with their own processor, using the Ethernut board as an Ethernet I/O processor only.

The expansion port contains CPU data and address bus, memory read/write signals, digital I/O ports, reset signal and power supply.

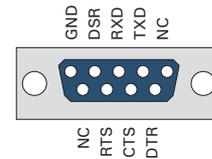


Pins	Signal	Function
1..2	NC	Reserved for future expansion, e.g. 3.3V supply.
3..4	VCC	5V regulated supply, app. 400 mA
5..8	GND	Signal ground
9	RESET	Low active reset signal.
10	DC	Unregulated supply voltage, depends on the board's supply.
11..12	VCC	5V regulated supply, same as pins 3..4.
13	RD	Low active read strobe.
14	WR	Low active write strobe.
15..22	DATA	8-bit data bus.
23..38	ADDR	16-bit address bus
39..46	PORTE	8-bit bi-directional digital I/O Port E.
47..54	PORTB	8-bit bi-directional digital I/O Port E.
55..62	PORTD	8-bit bi-directional digital I/O Port E.

## Serial Port

Ethernut provides an on-board DB-9 connector for its RS-232 serial communication port. This port is connected to the ATmega 128 first on-chip UART via a single-chip RS-232 driver/receiver interface circuit (IC6), which converts the required voltage levels from the 5V power supply. Two jumpers are provided to switch the DB-9 connector between DCE and DTE mode and two solder jumpers must be shortened to enable hardware handshake.

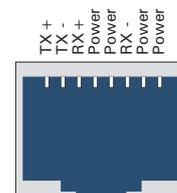
Pins	Signal	Function
2	TXD	Transmit data, if pins 2 and 4 of JP1 are shortened.
3	RXD	Transmit data, if pins 1 and 3 of JP1 are shortened.
4	DSR	Data set ready, not connected
5	GND	Signal ground
6	DTR	Data terminal ready.
7	CTS	Clear to send.
8	RTS	Ready to send.



## Ethernet Port

Ethernut provides an on-board modular RJ-45 connector for its twisted pair Ethernet port. This port is connected to the Realtek 8019AS Ethernet controller via a 10Base-T transformer/filter. The interface supports the maximum cable length of 100 meters between the Ethernet board and a hub.

Pins	Signal	Function
1, 2	TX	Differential transmit output pair.
3, 6	RX	Differential receive input pair.
4, 5	Power	Power supply.
7, 8	GND	Power supply.



## In-System Programming Port

A multiplexer (IC7) switches the microcontroller pins during programming, which are used by the serial port during normal operation. An on-board LED (LED2) indicates programming activity.



Pins	Signal	Function
1	MOSI	Slave serial data input.
2	VCC	5V regulated supply for programming adapter.
3	PROG	Low active programming mode. Must be held low during programming.
5	RESET	Low active reset signal. Must be held low during programming.
7	SCK	Slave input for serial clock.
9	MISO	Slave serial data output.
4, 6, 8, 10	GND	Signal ground.

### WARNING

Do not plug in or remove the programming adapter while power is supplied to the Ethernut board. You may damage the Ethernut board or the parallel port on the PC. This port allows serial programming of the ATmega 128 non-volatile Flash ROM and EEPROM without physical removal of the microcontroller from the system.

### LED Indicators

The Ethernut board is equipped with four LEDs.

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

A second red LED (LED2) is lit during in-system programming.

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.

## System Reset

System reset is controlled by a push button and a voltage supply supervisor (IC9). This device protects the EEPROM from corruption by generating a reset signal, when power supply from the on-board regulator (IC8) falls below 4.75 volts.

## Watchdog Timer

Software bugs, temporary hardware failures caused by electrical transients or interference and many other problems might cause the system to malfunction. The ATmega 128 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 32 kByte SRAM (IC4), which is used as read/write data storage. However, the lower 4 kBytes are overlaid by the ATmega128 internal register and SRAM space. The required address latch is provided by a 74HC573 (IC3).

## **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

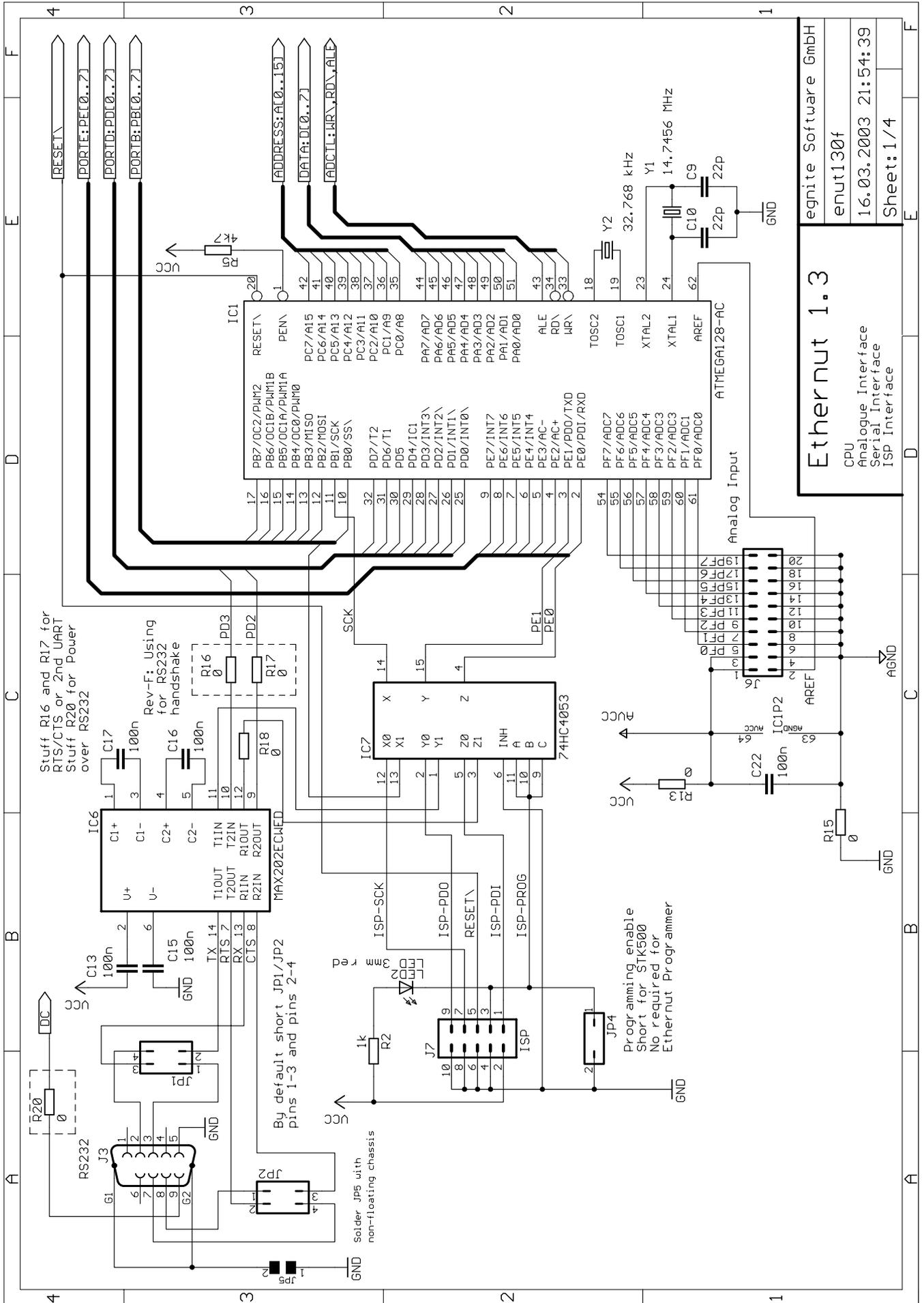
This chapter will help you in case of problems.

### Problem Solution

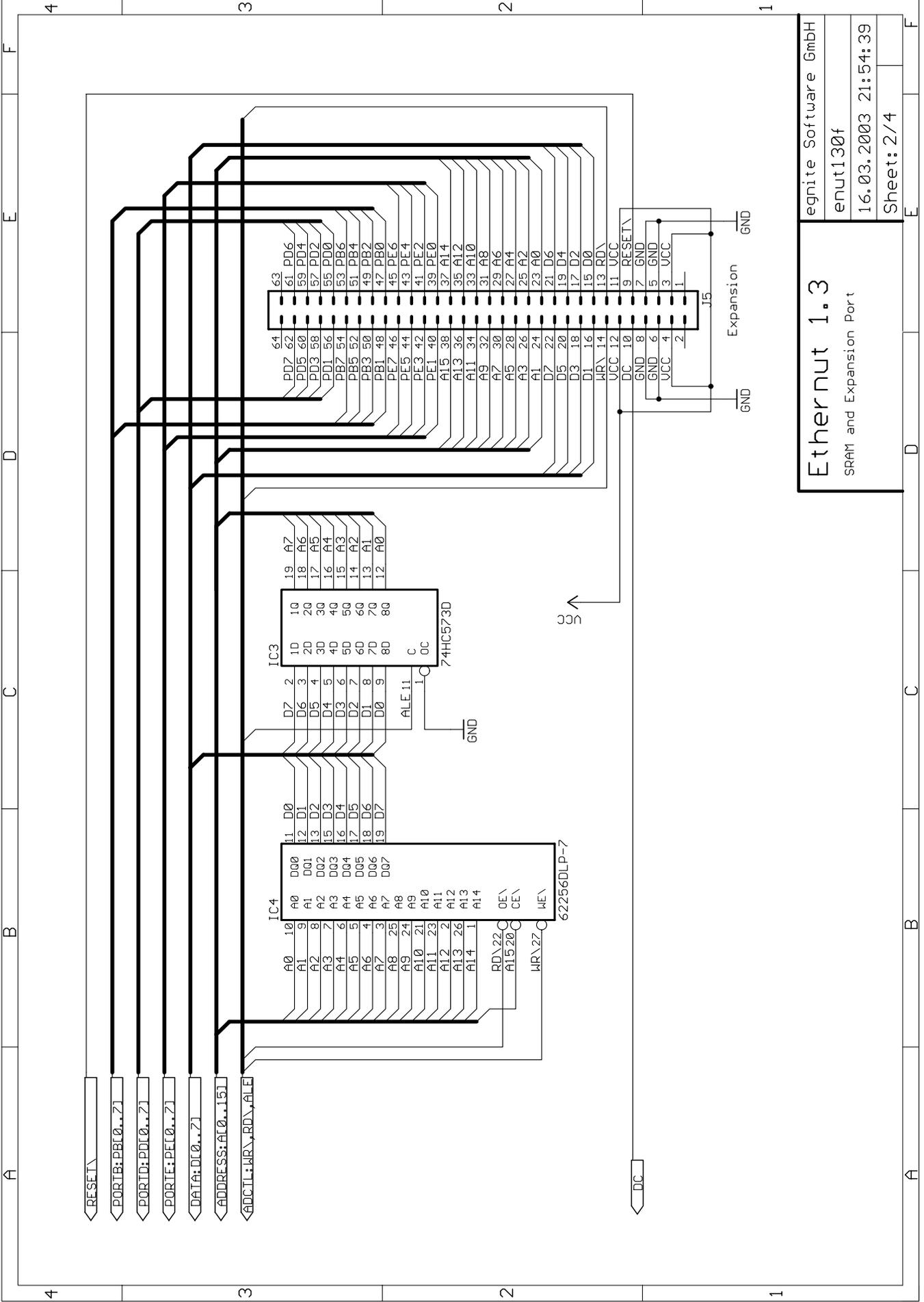
The red power LED is not on.	Connect the DC power cable. Check that the power supply is of DC type 8-16V min. 100 mA.
Nothing happens when pushing the RESET button.	Sometimes it is difficult to make a good connection when pushing the small RESET button. Try removing the power supply from the electrical outlet and disconnecting and reconnecting the power supply.
The PC does not show the BaseMon software prompt.	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 power, 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.
The terminal emulation software displays unreadable characters.	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.
AVRISP does not detect the ATmega device.	Disable the signature check in the option menu of the AVRISP.
The AVR device cannot be programmed.	Check that the notch on the Ethernet socket matches the notch on the programming device. The memory lock bits may be cleared. Completely erase the memory before programming.
There is a problem you cannot resolve.	Contact your dealer for assistance.



# 7 Schematics



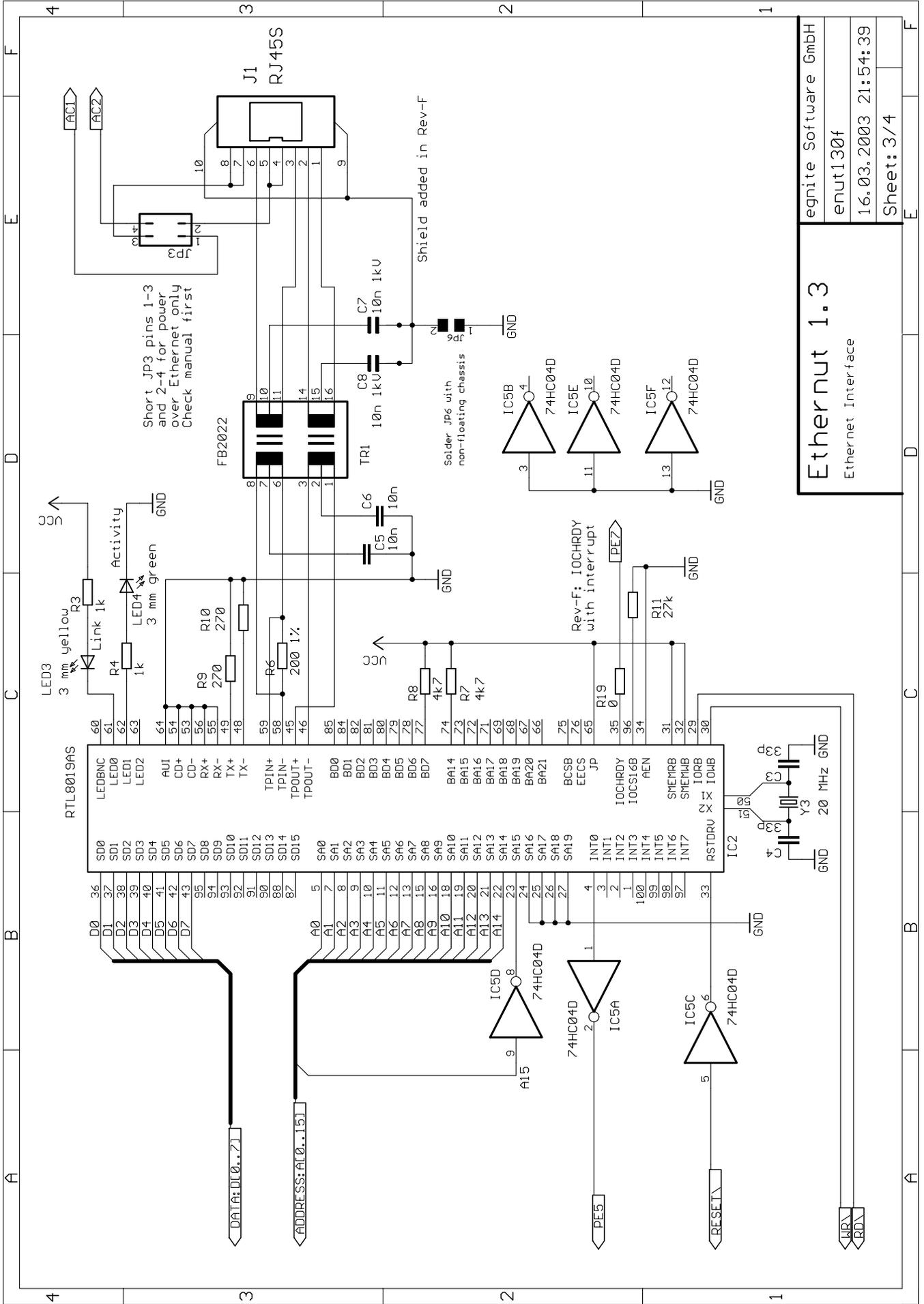
<b>Ethernet 1.3</b>	
CPU	egnite Software GmbH
Analogue Interface	enut130f
Serial Interface	16.03.2003 21:54:39
ISP Interface	Sheet: 1/4



# Ethernet 1.3

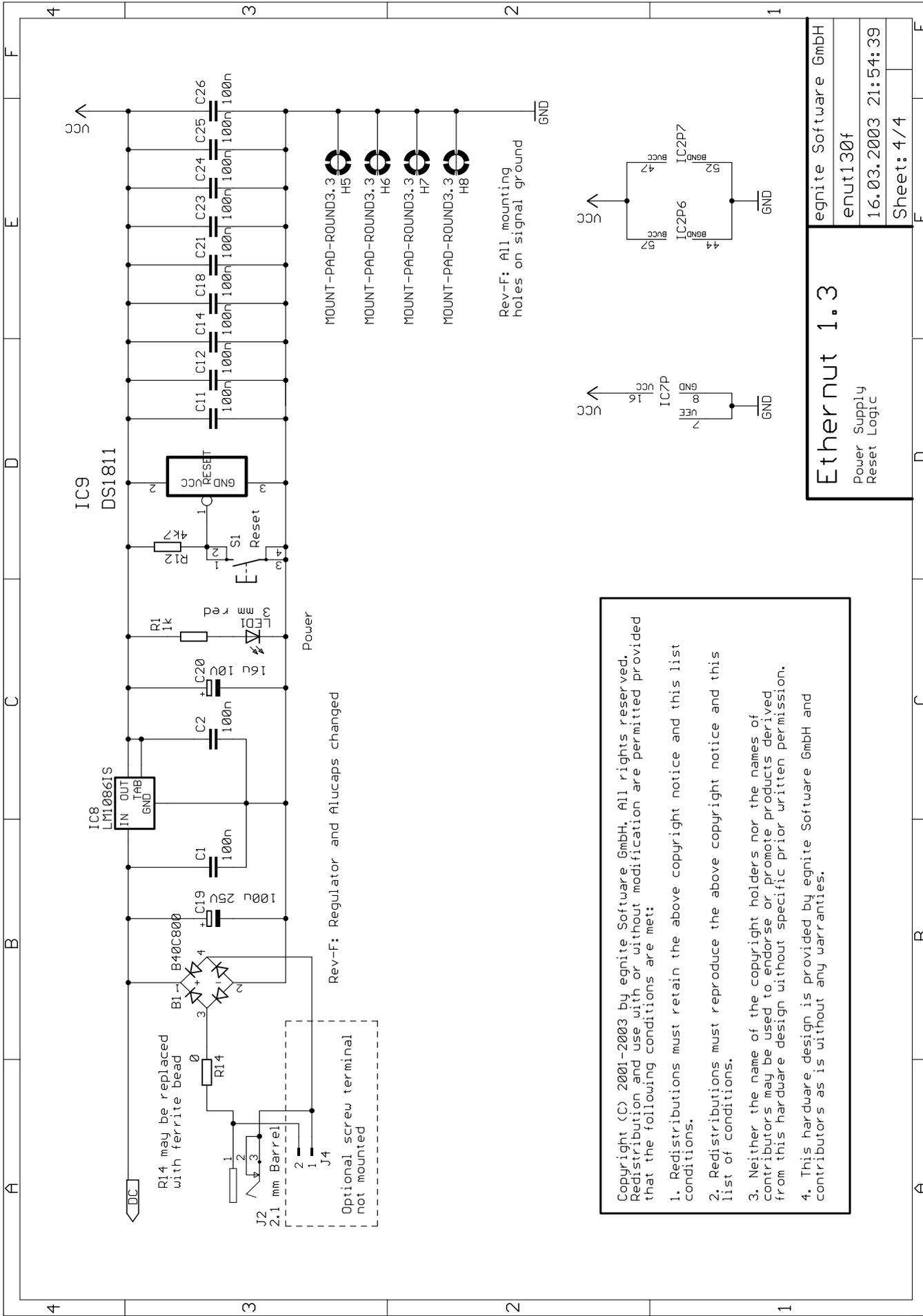
SRAM and Expansion Port

egnite Software GmbH  
 enut130f  
 16.03.2003 21:54:39  
 Sheet: 2/4



**Ethernet 1.3**  
Ethernet Interface

egnite Software GmbH  
enut130f  
16.03.2003 21:54:39  
Sheet: 3/4



Rev-F: Regulator and Alucaps changed

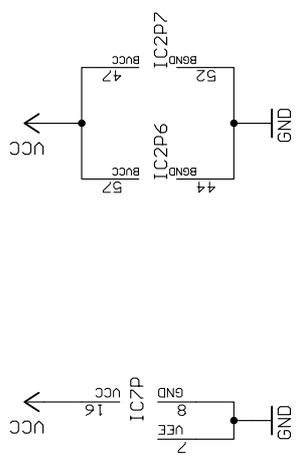
MOUNT-PAD-ROUND3.3 H5  
 MOUNT-PAD-ROUND3.3 H6  
 MOUNT-PAD-ROUND3.3 H7  
 MOUNT-PAD-ROUND3.3 H8

Rev-F: All mounting holes on signal ground

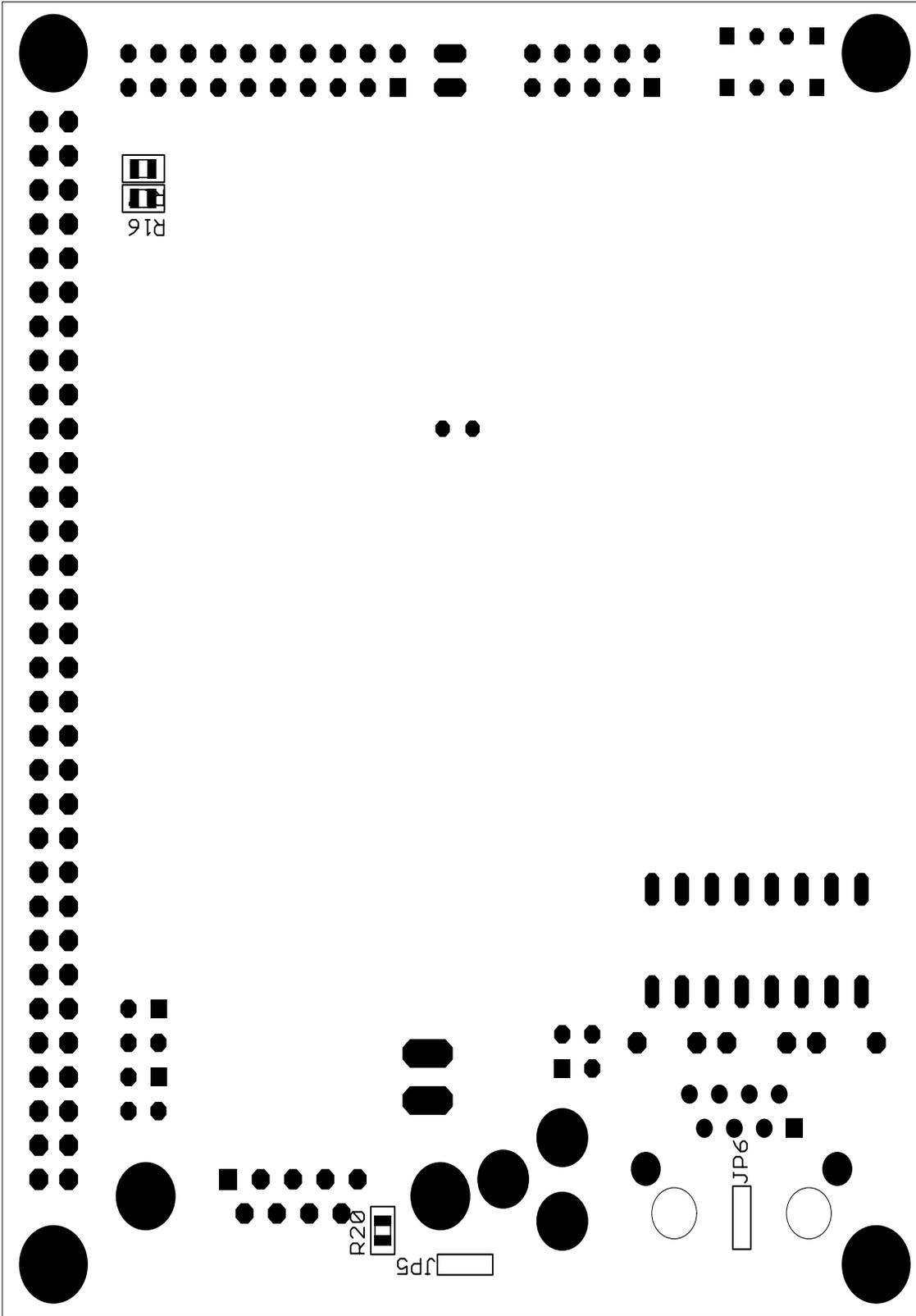
Copyright (C) 2001-2003 by egnite Software GmbH. All rights reserved. Redistribution and use with or without modification are permitted provided that the following conditions are met:

1. Redistributions must retain the above copyright notice and this list of conditions.
2. Redistributions must reproduce the above copyright notice and this list of conditions.
3. Neither the name of the copyright holders nor the names of contributors may be used to endorse or promote products derived from this hardware design without specific prior written permission.
4. This hardware design is provided by egnite Software GmbH and contributors as is without any warranties.

<b>Ethernut 1.3</b>	
Power Supply Reset Logic	
egnite Software GmbH	
enut130f	
16.03.2003 21:54:39	
Sheet: 4/4	











egnite Software GmbH  
Westring 303  
44629 Herne  
Germany

Phone + 49 (0)2323-925 375  
Fax + 49 (0)2323-925 374  
Email: [info@egnite.de](mailto:info@egnite.de)

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