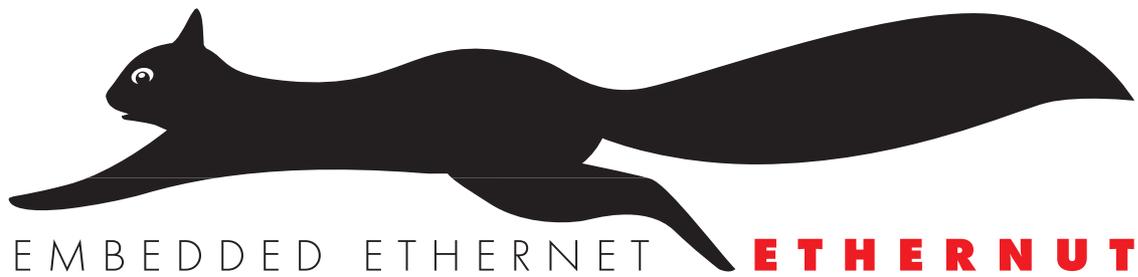


Ethernut Version 1.3
Hardware User`s Manual



Manual Revision: 1.8
Issue date: November 2005

Copyright 2001-2005 by 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

About the Ethernut 1.3 Board	1
Ethernet Features	1
Block Diagram	2
LED Indicators	3
Serial Ports	3
Ethernet Port	3
Expansion Port	3
Power Supply	4
Watchdog Timer	4
System Clock	4
Flash ROM	4
Static RAM	5
EEPROM	5
Configuration Jumpers	5
Upgrading from Previous Ethernut Revisions	5
Quick Start	6
Prerequisites for Operation	6
Precautions	6
Board Installation	7
Testing the Board	9
Ethernet Controller Read/Write Loop	10
Jump to Bootloader	10
SRAM Read/Write Loop	10
Send Broadcasts Loop	10
Exit BaseMon	10
Network Configuration	11
DHCP/BOOTP Method	12
Fixed IP Address	12
ARP Method	12
Testing Network Operation	13
Jumper Configuration	14
Jumper Overview	14
Hardware Expansion	15
Expansion Port	15
Analog Input Port	16
Troubleshooting	17
Sick Ethernuts	20
Schematic	21

About the Ethernut 1.3 Board

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

Since its introduction in the year 2000, Ethernut boards have been used to develop some of the most innovative products. Using the hardware, firmware, software and tools, developers have everything they need to develop leading networked devices rapidly and affordably. 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

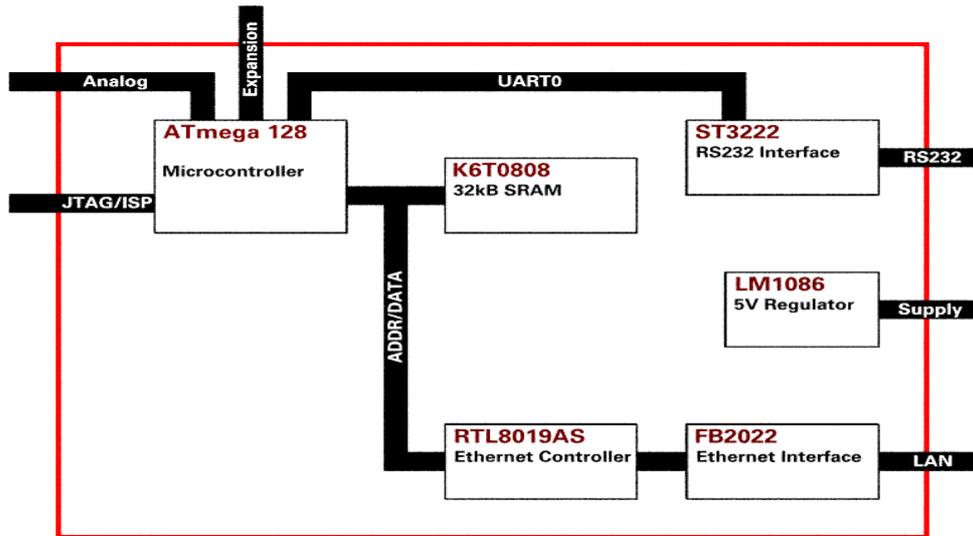
Ethernut 1.3 is a small (78 x 98 mm) board combining Atmel's ATmega128 RISC microcontroller with Realtek's RTL8019AS Ethernet controller. The main features are:

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

Block Diagram

The block diagram shows the main components.

Definitely the most important part is the ATmega 128 microcontroller. It's a quite complex chip and described in detail in Atmel's ATmega 128 data sheet. Almost all pins are routed to the Ethernut expansion port, a 64-pin connector, which can be used to add custom hardware like the Medianut MP3 decoder with LCD interface.



The microcontroller provides two UART channels, one is routed to the on-board RS-232 level shifters.

While Ethernut's software offers several bootloader capabilities over RS-232 or Ethernet, program code is initially uploaded through the JTAG or SPI interface. The connector layouts conform to Atmel's 10-pin JTAG interface. Unfortunately the same type of connector is used by Atmel for the SPI interface.



WARNING: Never plug an SPI programming adapter in to the Ethernut 1.3 JTAG connector or a JTAG adapter into the SPI connector. This will result in a power supply shortcut and will at least blow the fuse of your power supply.

LED Indicators

The Ethernut 1.3 board is equipped with four LEDs, two red colored, a green and a yellow LED.

The first red LED1 is directly connected to the power supply. It is lit when power is applied to the board.

The second red LED2 is connected to the UART0 transmit line and is lit when serial data is transmitted.

A green and a yellow LED (LED3 and LED4 resp.) are used to indicate activity on the Ethernet port. The yellow LED indicates the network link status and is lit when the link status is OK. The green LED indicates receive and transmit activity from and to the network.

Serial Ports

Ethernut provides an on-board DB-9 connector for RS-232 serial communication. IC6 is used to convert the required voltage levels for RS-232 from the 5V power supply.

The second serial interface, UART1, is available on the expansion port at TTL level.

Ethernet Port

Ethernut provides an on-board modular RJ-45 connector for its twisted pair Ethernet port. This port is connected to the RTL8019AS 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.

Expansion Port

Add-on boards can be added to the expansion port. These boards may contain 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.

Power Supply

The complete logic of the Ethernut board is driven by a 5V regulator, IC8. An unregulated power supply of DC 9-12V with a minimum current of 200 mA is sufficient to run the board.

Four 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 rectifier bridge (D1, D2, D3 and D4).
- 2** Using pins 4, 5 and pins 7, 8 of the Ethernet twisted pair connector. In this case pins 1 and 3 and pins 2 and 4 of jumper JP3 must be shortened. Like the barrel connector, this input is protected by a rectifier bridge. A special power supply injector is required for the Ethernet cable. Do not set jumpers on JP3 without such an injector. You may destroy other equipment attached to the Ethernet cable.
- 3** Pin 9 of the RS232 DB-9 connector can be used to supply the Ethernut board or draw power from it. To use this option, a 0 Ohm resistor R35 must be mounted on the back side of the board.
- 4** The DC signal is routed to the Ethernut expansion connector to either supply add-on boards or to receive power supply from an add-on board.

As soon as power is attached to any of the inputs mentioned above, the red LED1 will lit.

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 (Y1), which may be replaced by a crystal of up to 16 MHz. An additional 32.768 kHz crystal (Y2) 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 (Y3).



WARNING: Note, that changing any crystal will alter the Ethernut 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 Ethernut board provides 32 kByte SRAM (IC4), which is used as read/write data storage. The lower 4352 bytes of this external memory chip are overlaid by the internal ATmega128 register and SRAM space. The required address latch is provided by 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 in-system programming. Note, that EEPROM write access is much slower (about 2.5 ms) than writing to SRAM.

Configuration Jumpers

The board is equipped with 2 jumper areas.

JP1 and JP2 configure the connection to the serial device (UART0).

JP3 enables power supply over Ethernet cable.

Upgrading from Previous Ethernut Revisions

Ethernut has undergone many changes since its initial release in the year 2000, but board dimensions and positions of main connectors remained unchanged.

Also, the software still supports all previous Ethernut Boards including revision 1.1 with the ATmega 103 microcontroller, which is no longer in production. However, there are a few things to consider.

The most important change to notice is the second programming connector for JTAG programming. Always plug the correct adapter cable into the specified connector.

Since board revision G, Ethernut 1.3 emulates an EEPROM for the Ethernet Controller. Upgrading from previous revisions of Ethernut 1 requires to link the application code with the latest Nut/OS libraries. Otherwise remove R7 (top) and stuff R37 (bottom) for full compatibility with previous revisions.

Quick Start

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

Prerequisites for Operation

The following items are necessary to run the Ethernut board:

- A standard PC equipped with Linux or Windows, an available serial COM port and a twisted pair Ethernet adapter card.
- Terminal emulation software, such as TeraTerm or Hyperterminal.
- An unregulated power supply matching your local mains. It should supply DC 9-12V, 200 mA minimum, on a standard 2.1 mm barrel connector.
- Two straight-through twisted pair cables together with a 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 Ethernut Starter Kit:

- Ethernut Board.
- SP Duo JTAG and SPI 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.
- A CD with all required software tools and documents. Additionally check www.ethernut.de for the latest releases.
- This manual.

It is further assumed, that you got some basic knowledge about digital hardware and TCP/IP networking. This manual will not present any of these basics, but you can find excellent books or web resources about these topics.

Precautions

Born out of an Open Source Project, the Ethernut board is a commercial product, from which you expect some kind of fail safe operation. But also keep in mind, that a bare electronic circuit is a fragile product, which demands careful handling. In the first place learn how to avoid problems caused by electrostatic discharge.

Moreover, no limitations are applied to chip programming, which may guard you against accidents. In particular, the ATmega 128 microcontroller may be completely disabled by misprogramming its fuses. Thus we strictly recommend not to change the fuse settings, if you are not absolutely sure what you are doing.

The following fuses had been enabled (programmed to zero) before shipping the board: JTAGEN, SPIEN, EESAVE, BOOTSZ1, BODLEVEL, BODEN and CKOPT. All other fuses remain unprogrammed (erased to one).

Board Installation

1 Remove the board from the antistatic bag. Visually inspect the board to verify that it was not damaged during shipment. Do not use the antistatic bag as a underlying pad for Ethernet, because it's electroconductive. Put the board on a wooden surface or simply on a piece of paper. Plastic surfaces may be harmful because of electrostatic discharge.



WARNING: As with all computer equipment, the Ethernet board may be severely damaged by electrostatic discharge (ESD). Be sure to take proper precautions before removing the Ethernet board from the antistatic bag. Never pass the board from one person's hand to another.

2 Connect Ethernet's DB-9 RS232 port to an available COM port using the serial cable.

3 Use one twisted pair cable to connect Ethernet'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 Ethernet board with the PC's network adapter using a twisted pair cross cable.

4 Connect the power supply to the barrel connector on the Ethernet board. The Ethernet 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 Ethernet board.

5 Apply power to the Ethernet 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 a minute, 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 JTAG connector. 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.

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 space bar on the PC keyboard and press and release the reset button on the Ethernet board. This is the tiny push button at the board's edge near the screw terminal. After some seconds the following output should appear in the terminal emulation window:

```
BaseMon 4.1.2
Nut/OS 3.5.0.
Compiled by AVRGCC for ATmega128
Baudrate select = 23
External RAM Test... 28416 bytes
Banked RAM Test... none
Detecting NIC... RTL8019AS
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
```

Depending on the preloaded version and the baudrate, your output may slightly differ. But the amount of RAM should match and all tests should report the result OK.

The menu will not appear, if BaseMon didn't receive a space character or failed to determine the baudrate. Check again the configuration of the terminal emulation program and make sure that all handshakes are disabled. Sometimes the keyboard repeat rate of the PC is too slow, in which case BaseMon isn't able to verify the baudrate. If you don't know how to increase this rate, thumb on the space bar as fast as you can. Replacing the Beethoven sound from your MP3 player by a record from Metallica or Disturbed will be helpful.

Nevertheless, if BaseMon is unable to receive three consecutive space characters at the same baudrate, it will continue at 38,400 Baud. In this case it will not display the menu, nor wait for any further input, but display the test results and then immediately start the build-in webserver.

Unlike some other embedded monitors, BaseMon is not resident. You will later upload other sample applications or your own code, which overwrite BaseMon. Whenever you are not sure, whether any problems may arise from hardware or software failures, it's always a good idea to upload the BaseMon hex file again for running the integrated tests. The menu offers further tests, which will be described now.

Ethernet Controller Read/Write Loop

When pressing E on the BaseMon menu, the Ethernut board will enter an endless loop, trying to read the revision ID of the Ethernet controller at base address 8300 hex:

```
rev=0x7050
```

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 to 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, which is typically required before uploading a new application.

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, BaseMon 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.

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.

Changing the network configuration requires user interaction, either by keyboard/LCD interface, web browser, RS-232 communication or whatever the final system may provide. It's up to the specific application to deal with these values. Nut/OS just provides a common framework. The BaseMon application explained in the previous chapter uses RS-232, for example.

The TCP/IP configuration is stored in EEPROM and contains the following values (Nut/OS Version 3.5).

Name	Type	Default	Description
size	Byte value	31	Total size of the configuration structure. Used to check validity.
name	Character array	eth0	Name of the network interface. Maximum size is 8 characters.
mac	Byte array	000698000000	6 bytes unique MAC address.
ip_addr	IP address	0.0.0.0	Last used IP address.
ip_mask	IP address	255.255.255.0	Configured IP mask.
gateway	IP address	0.0.0.0	Default gateway IP.
cip_addr	IP address	0.0.0.0	Configured IP address.

Default values will be used by the software when the EEPROM has been previously erased. When Ethernuts are shipped, the EEPROM contains the values from the final test. The last used IP address is set to 192.168.192.x, with x varying between 100 and 254. The MAC address will be the one, that has been uniquely assigned to your board. The last six digits are written on the board. The configured IP address is set to 0.0.0.0, which automatically enables DHCP.

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.

The ATmega 128 microcontroller offers a number of programmable fuses to change general modes. One of these fuses disables EEPROM erasure during chip

erase and has been set before shipping the board. Thus, if the chip is erased in order to upload a new application, the EEPROM contents is preserved.

DHCP/BOOTP Method

This is the preferred method. As explained, the configured IP of shipped Ethernuts is set to 0.0.0.0, which enables DHCP/BOOTP. If a DHCP server exists on the network, Ethernut will automatically request its IP address, the IP address of the standard gateway, and the IP mask of the local network. If no DHCP server could be located, the board will reuse the last used address. If this is 0.0.0.0 or if the EEPROM had been erased, then Ethernut switches to the ARP method.

Fixed IP Address

If DHCP service is not available, you should assign a fixed IP address. Before you create your own applications with fancy user interfaces to set this address, you can use BaseMon. The EEPROM configuration will be preserved when reprogramming the Ethernut and will be reused by the new applications.

ARP Method

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

To set the Ethernut'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 Ethernut board.

Enter the following command to manually create an ARP entry for an Ethernut 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 Ethernut 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 set the network mask to 255.255.255.0.

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.

Refer to the Ethernut Software Manual for further information about network configuration.

Testing Network Operation

After configuring the network parameters, you can check, that the Ethernet 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
```

Replace the shown IP address by the one you configured previously. If you receive a response without timing out, the Ethernet board is ready to try the HTTP daemon. Use any Webbrowser to query the following URL:

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

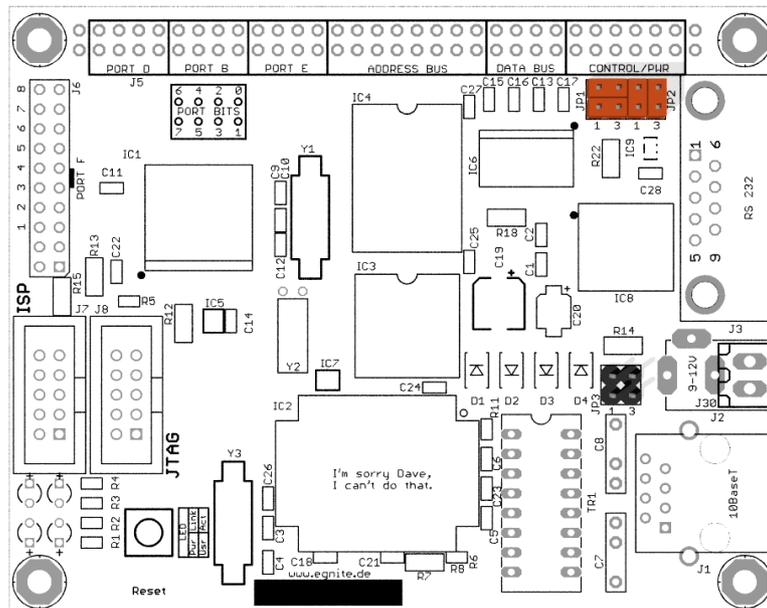
Again, instead of the above IP address use the one previously configured.

Jumper Configuration

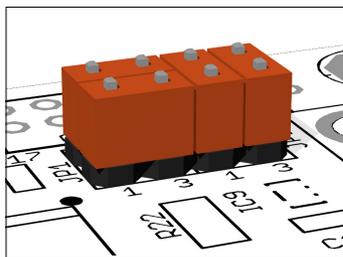
Adapting Ethernut to specific requirements.

Jumper Overview

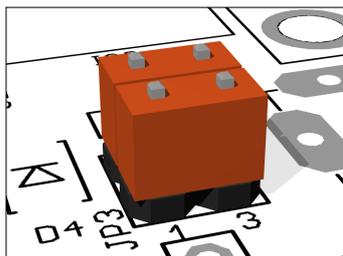
The board is equipped with 2 jumper areas. The picture below shows the default jumper configuration, set when Ethernut is shipped.



JP1 connects the serial device UART0 to the DB-9 connector.
 JP2 shortcuts hardware handshake lines on the DB-9 connector.
 JP3 enables power supply over Ethernet cable.



JP1-2 Shortening pins 1 and 3 and pins 2 and 4 of JP1 will route UART0 transmit and receive lines to the DB-9 connector. Shortening pins 1 and 2 and pins 3 and 4 of JP2 connects RTS and CTS signals on the DB-9 connector.



JP3 When pins 1 and 3 and pins 2 and 4 are shortened, then power can be supplied on pins 4, 5 and pins 7, 8 of the Ethernet twisted pair connector. A special power supply injector is required for the Ethernet cable. Do not set jumpers on JP3 without such an injector. You may destroy other equipment attached to the Ethernet cable.

Hardware Expansion

Ethernut and custom hardware.

Many applications will do just fine with nothing else than the Ethernut or external hardware may be connected to the RS-232 port. However, if more is required, the Ethernut expansion port is the first choice to add custom designed hardware.

Expansion Port

Add-on boards can be added to the expansion port. These boards may contain 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. Nearly all microcontroller pins are available at the expansion port connector, providing an interface with lots of features like PWM, I2C (2-wire), SPI (3-wire) or counter inputs, to name just a few. It is strictly recommended to consult the ATmega 128 data sheet before attaching hardware to the expansion port.

Although available at the connector, some signals are used internally by Ethernut and can't be used by external hardware. Carefully check the schematic. Bit 5 of Port E is used for LAN controller interrupts. You can switch this to bit 6 of the same port by removing R2 and mounting R31 (bottom side). Bit 4 of Port B is used as a chip select for the serial data flash.

Other pins may not be available, depending on the jumper configuration. Please refer to the previous chapter for additional information.

The ALE signal is not available at the expansion port by default and may be connected to pin 64 by mounting R30 on the bottom side.

The following table lists the expansion port's pin assignment. Pin 1 is located at the board edge on the side of the DB-9 connector.

Pin	Signal	Description	Pin	Signal	Description
1	NC	Reserved for +3.3V	2	NC	Reserved for +3.3V
3	VCC5	Regulated +5.0V	4	VCC5	Regulated +5.0V
5	GND	Signal ground	6	GND	Signal ground
7	GND	Signal ground	8	GND	Signal ground
9	MR\	Reset input	10	DC	Unregulated supply
11	VCC5	Regulated +5.0V	12	VCC5	Regulated +5.0V
13	RD\	Memory read signal	14	WR\	Memory write signal
15	D0	Data bus bit 0	16	D1	Data bus bit 1
17	D2	Data bus bit 2	18	D3	Data bus bit 3
19	D4	Data bus bit 4	20	D5	Data bus bit 5
21	D6	Data bus bit 6	22	D7	Data bus bit 7
23	A0	Address bus bit 0	24	A1	Address bus bit 1
25	A2	Address bus bit 2	26	A3	Address bus bit 3
27	A4	Address bus bit 4	28	A5	Address bus bit 5
29	A6	Address bus bit 6	30	A7	Address bus bit 7
31	A8	Address bus bit 8	32	A9	Address bus bit 9
33	A10	Address bus bit 10	34	A11	Address bus bit 11
35	A12	Address bus bit 12	36	A13	Address bus bit 13
37	A14	Address bus bit 14	38	A15	Address bus bit 15
39	PE0	Port E bit 0	40	PE1	Port E bit 1
41	PE2	Port E bit 2	42	PE3	Port E bit 3
43	PE4	Port E bit 4	44	PE5	Port E bit 5
45	PE6	Port E bit 6	46	PE7	Port E bit 7
47	PB0	Port B bit 0	48	PB1	Port E bit 1
49	PB2	Port B bit 2	50	PB3	Port E bit 3
51	PB4	Port B bit 4	52	PB5	Port E bit 5
53	PB6	Port B bit 6	54	PB7	Port E bit 7
55	PD0	Port D bit 0	56	PD1	Port E bit 1
57	PD2	Port D bit 2	58	PD3	Port E bit 3
59	PD4	Port D bit 4	60	PD5	Port E bit 5
61	PD6	Port D bit 6	62	PD7	Port E bit 7
63	NC		64	NC	ALE, if R30 stuffed

Analog Input Port

In order to avoid interference with typically noisy digital lines, a separate connector is used for the microcontroller’s analog inputs.

Pin	Signal	Description	Pin	Signal	Description
1	AVCC5	Analog supply	2	AREF	Analog reference
3	AVCC5	Analog supply	4	AGND	Analog ground
5	PF0	Analog input 1	6	AGND	Analog ground
7	PF1	Analog input 2	8	AGND	Analog ground
9	PF2	Analog input 3	10	AGND	Analog ground
11	PF3	Analog input 4	12	AGND	Analog ground
13	PF4	Analog input 5	14	AGND	Analog ground
15	PF5	Analog input 6	16	AGND	Analog ground
17	PF6	Analog input 7	18	AGND	Analog ground
19	PF7	Analog input 8	20	AGND	Analog ground

Troubleshooting

Problem	Solution
<p>BaseMon displays positive test results but doesn't show the menu. It immediately starts the web sever. BaseMon produces garbage output and the yellow LED goes on after about a minute.</p>	<p>BaseMon was unable to detect three consecutive space characters at the same baudrate. The keyboard repeat rate of the PC may be too slow or the terminal emulator doesn't send any characters because of handshake settings or other RS-232 communication issues. See the previous problem solution.</p>
<p>Uploading new applications fails. The programming software reports, that no target has been detected.</p>	<p>Either by accident or because you refused to read any warnings you may have disabled the JTAG fuse. If you didn't disable the SPI as well, there is still hope. You can use the SPI connector and try to re-enable the JTAG fuse via SPI programming. If this fails or if you are using SPI programming, check the 14.7456 MHz crystal with an oscilloscope. If there is no sine wave, you may be still lucky having disabled the external clock only with the SPI fuse still intact. Feed an external clock signal of at least 1 MHz to the XTAL1 pin of the ATmega 128 and try again to program the board via SPI. In case you disabled both, SPI and JTAG, the ATmega128 needs to be replaced.</p>
<p>SPI or JTAG programming reports verification errors.</p>	<p>Flash memory has a limited number of erase cycles. For the ATmega 128 this limit is roughly at 10,000 cycles minimum and may be 10 times more. With normal usage this number will never be reached. However, if an application like a bootloader uses the self programming feature and runs weird, flash memory may soon wear out. Another pitfall is not to erase the chip before uploading the new code. Flash memory bits can be cleared to zero during programming but must be erased to set them back to one. Check your programming software.</p>

Problems	Solution
<p>The red LED does not go on when applying power.</p>	<p>Your power supply may not work. Remove any kind of attached hardware and remove all jumpers. Make sure the board is placed on a non conductive surface like a piece of paper. Supply the board via the barrel connector J2 only with not more than 12V DC. Best use a lab power supply with current control and carefully increase the voltage starting from 3V. The board should not draw more than 200 milliamps.</p>
<p>The yellow LED does not go on after entering the BaseMon HTTP server or similar network enabled software.</p>	<p>The yellow LED will go on only if Ethernut is connected to an Ethernet network and the Ethernut software properly initialized the LAN controller hardware on the Ethernut. Load the board with BaseMon to make sure, that the CPU is working. Replace the Ethernet cable and try the same connection with your PC to make sure that the network link is working.</p>
<p>BaseMon reports errors during hardware checks.</p> <p>The board seems to work unreliable.</p> <p>I'm not able to program the microcontroller.</p>	<p>This problem is typically caused by a wrong power supply. Make sure to use one with 8-12 V and a minimum DC current of 200 mA. Although equipped with a rectifier bridge, Ethernut will not work with AC supply.</p> <p>BaseMon may report port problems when additional hardware is attached to the board. The program enables the internal pullups and expect the port signals to become high. It will then set a single port bit to low and check whether the all other port bits remain high. Depending on the hardware attached, this may not be considered a failure. It is generally a good idea to remove external hardware before running BaseMon.</p>

Problem	Solution
<p>Ethernut doesn't respond to pings. The green LED does not go on.</p>	<p>Configuring TCP/IP looks generally simple after one has understood the principle, but may still become confusing under some circumstances. For example, changing Ethernut's MAC address can disable a link, which had been running fine before the change. This happens, because the PC remembers the MAC/IP relations for some minutes.</p> <p>Check your configuration again. Make sure, that Ethernut and the PC are located in the same network, sharing the same IP mask and network IP address. If you don't know what all this means, check the WWW for some excellent TCP/IP tutorials.</p>
<p>Ethernut works fine after pressing reset, but not after switching on the power supply.</p>	<p>The LAN controller's power on reset requires a minimum supply raise time, while some power supplies do have an intentionally slow rise. The Ethernut 1.3 board is equipped with an MIC2775 reset controller to keep the LAN controller in reset state during power on. A broken reset controller results in the failure described.</p>
<p>BaseMon doesn't produce any output on the terminal emulation window. However, after a few minutes the yellow LED lits.</p>	<p>Probably an RS-232 communication problem. Put all jumpers on the Ethernut to their default positions. Check the settings of the terminal emulator. Use 8 data bits, 1 stop bit and no parity, all handshakes should be disabled. Set the baudrate to 38400 baud. Test the PC's serial port with the terminal emulation and some other hardware or a second PC (requires a null modem cable).</p> <p>If it takes several minutes until the yellow LED goes on, you may have misprogrammed your ATmega 128 fuses. The CPU may be driven by the slow internal clock instead by the external 14.7465 MHz crystal.</p>

Sick Ethernuts

Is there still life in it?

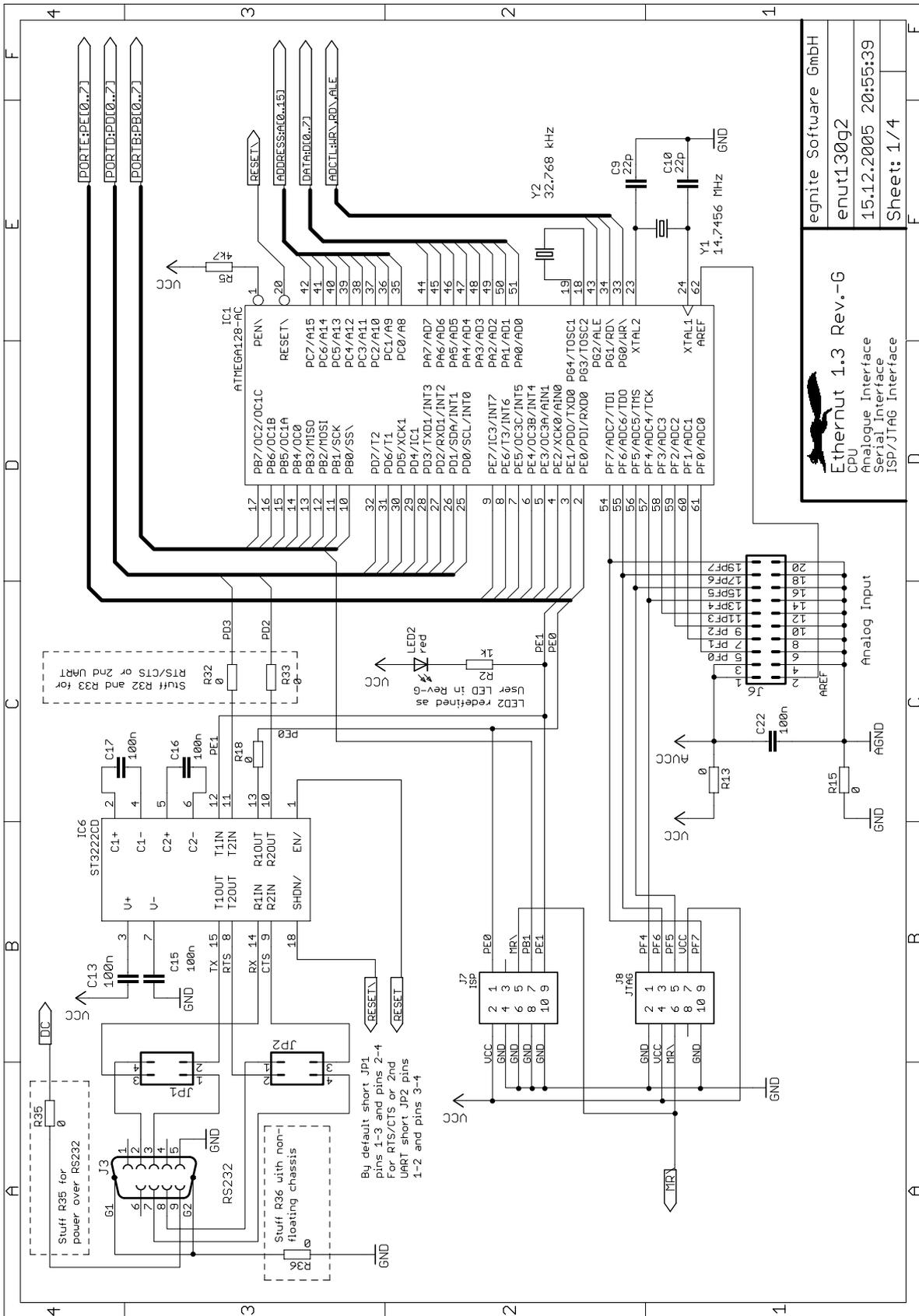
Our warranty scheme is simple. All boards have been extensively tested before shipment and we feel responsible, that it continues to work reliable after passing it to you.

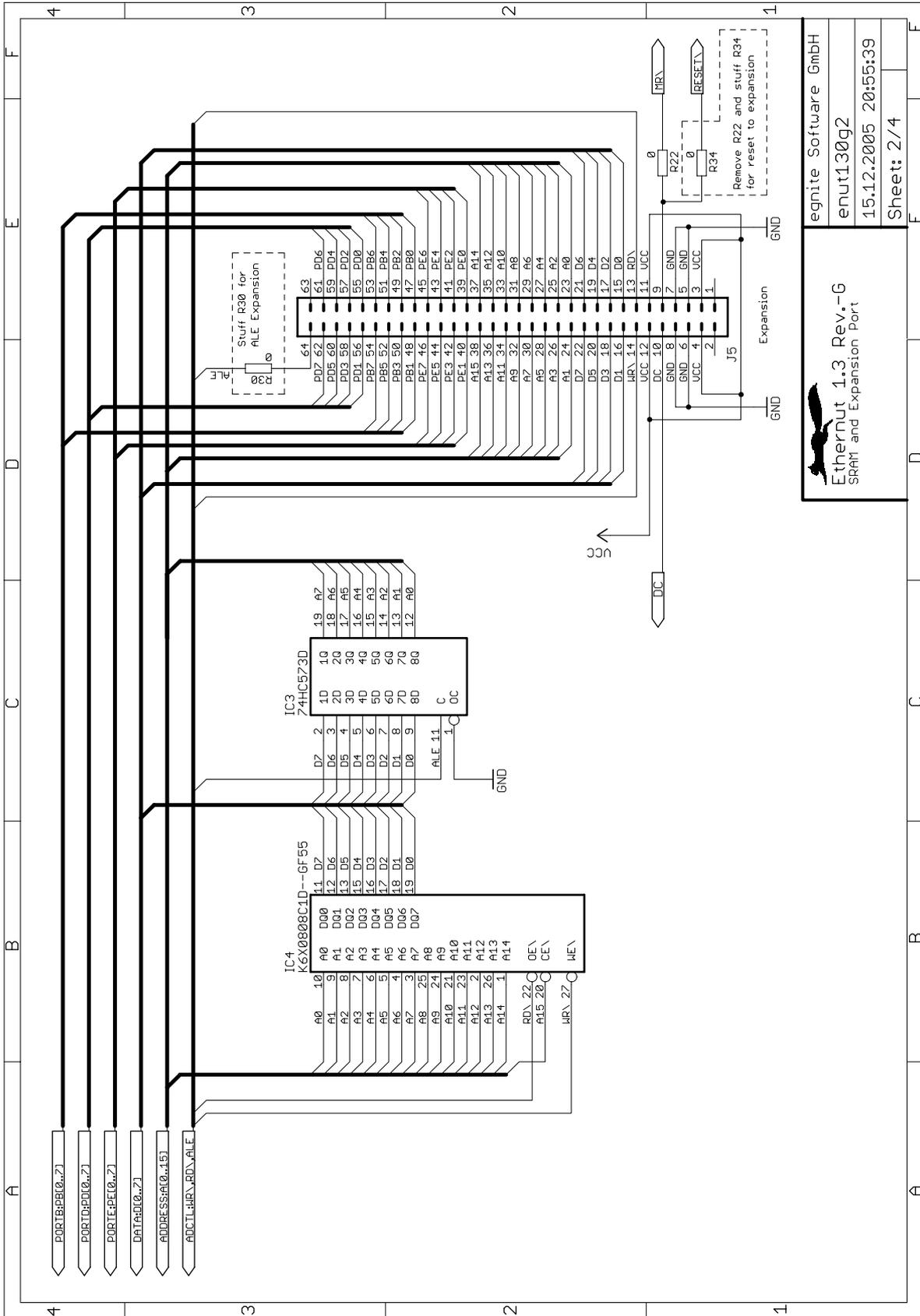
If the trouble shooting guide doesn't help or if it results in the conclusion, that your Ethernut is broken, you should send an email to info@egnite.de, including the following information:

- Ethernut Revision, printed on the back side of the board.
- MAC address of your Ethernut, written on top of the board and on the invoice.
- BaseMon output, if applicable. Or software revision you're using, noted on the first page of the API documentation.
- Description of your problem. You may keep it simple, we may request details later.

Please understand, that we are not able to provide any warranty, if you misprogrammed the fuses, destroyed the board because of ignoring our ESD precautions advises or attaching badly designed hardware. In such cases we may ask at least for a refund of our shipping costs.

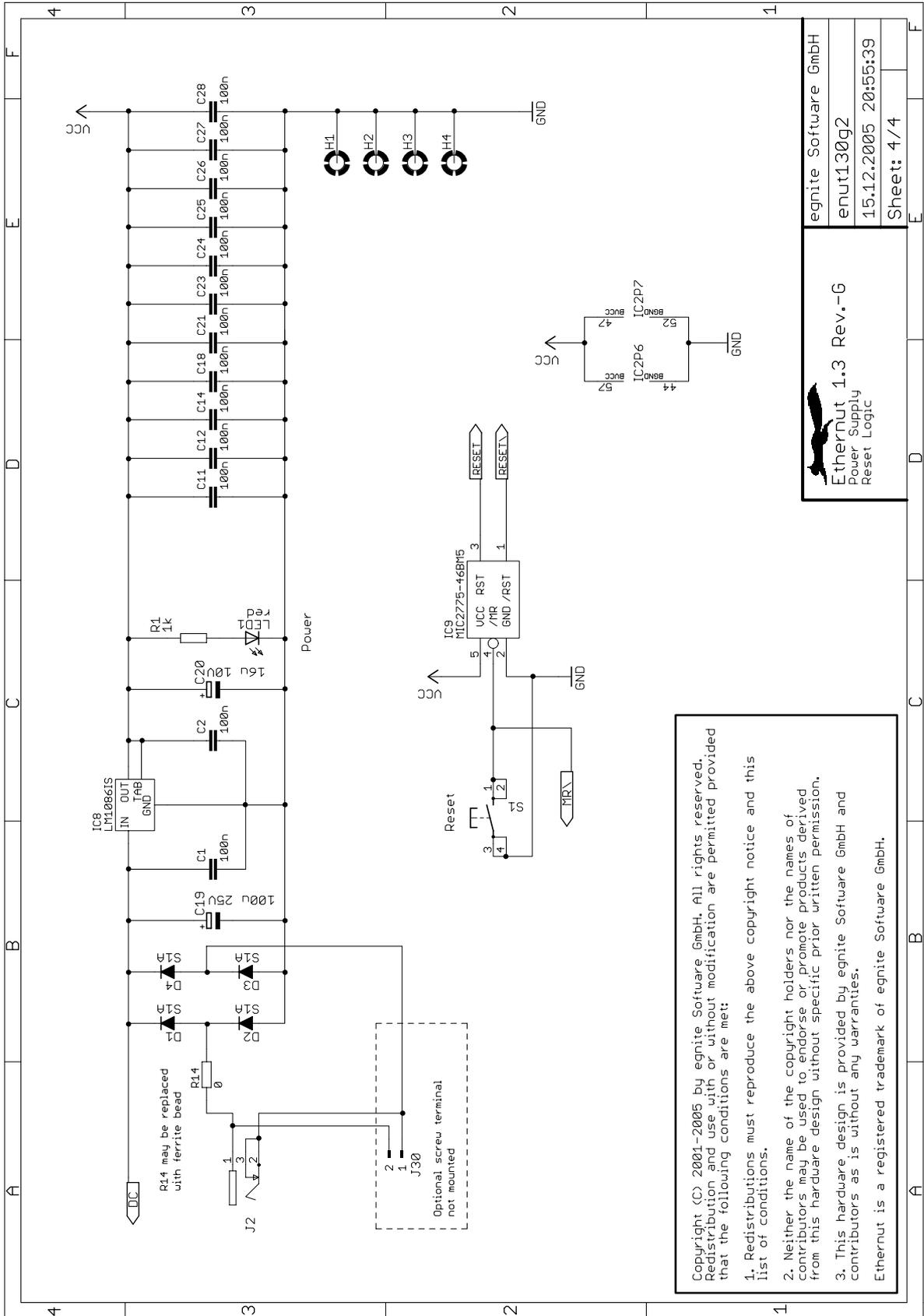
Anyway, whatever happened, we will do anything possible to revitalize your Ethernut. Or, if it finally passed away, let it rest in peace and send a replacement back to you at the least possible costs.





egnite Software GmbH
 enu130g2
 15.12.2005 20:55:39
 Sheet: 2/4

 Ethernet 1.3 Rev.-G
 SRAM and Expansion Port



Copyright (C) 2001-2005 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 reproduce the above copyright notice and this list of conditions.
2. 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.
3. This hardware design is provided by egnite Software GmbH and contributors as is without any warranties.

Ethernut is a registered trademark of egnite Software GmbH.

 Ethernet 1.3 Rev.-6 Power Supply Reset Logic	
egnite Software GmbH	
enut130g2	
15.12.2005 20:55:39	
Sheet: 4/4	



egnite Software GmbH
Erinstr. 9
44575 Castrop-Rauxel
Germany

Phone +49 (0)23 05-44 12 56
Fax +49 (0)23 05-44 14 87

Email info@egnite.de

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