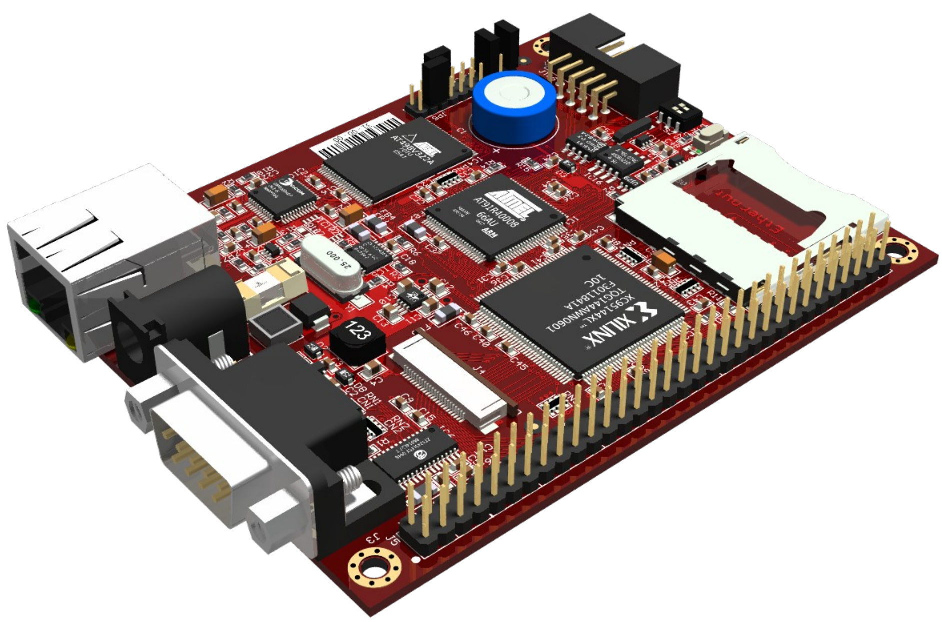
Ethernut 3.1

Hardware Manual



Manual Revision: 3.0

Issue date: November 2009

Contents

About the Ethernut 3.1 Board

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 affordable.

The board is well suited for a wide range of applications. Some areas are:

- Networked sensors
- Remote monitoring equipment
- Alarm service providers
- Remote diagnosis and service
- Industrial Ethernet applications
- Home and building control

Ethernut Features

The third generation of the Ethernut board family has been designed for fast response times at very low power consumption. When running in internal RAM, the 32-bit CPU executes 72 MIPS, while the programmable logic allows to implement special interfaces in hardware.

The key features are:

- AT91R40008 RISC microcontroller with fast 256 kByte SRAM
- 4 MByte NOR Flash ROM
- Programmable Logic Device with 144 Macrocells
- Full duplex IEEE 802.3 compliant 10/100 Mbps Ethernet interface
- RS-232 at DB-9 connector with full modem handshake
- Multimedia Card Socket
- 17+ programmable digital I/O lines
- Real time clock with backup capacitor
- Three 16-bit timer/counters
- Watchdog timer for enhanced reliability
- LED indicators for power supply and Ethernet activity
- Wide power supply range of 5V - 24V DC

Quick Start

Prerequisites for Operation

Bulk boards as well as boards included in the Ethernut starter kit are shipped with a boot loader programmed in NOR flash memory and a unique MAC address stored in non-volatile memory. JTAG jumpers are properly set to access the CPU's JTAG interface.

The following items are included in the Ethernut Starter Kit:

- Ethernut Board
- Turtelizer 2 JTAG programming adapter
- Crossed serial communication cable with DB-9 female connector at both ends
- CD with all required software tools and documents
- This manual

To run the Ethernut Board, you additionally need:

- A standard PC running Linux, Windows or Mac OS X with serial COM port and Ethernet interface
- Terminal emulation software, such as MiniTerm (Linux) or TeraTerm or Hyperterminal (Windows)
- TFTP server, such as TFTPD32
- An unregulated power supply providing a minimum of 5V, but not more than 24V on a standard 2.1 mm barrel plug
- Twisted pair cable together with 10/100 Base-T hub or switch

It is further assumed, that you have some basic knowledge about digital hardware and TCP/IP networking. This manual will not discuss 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 itself is a commercial product you paid for. You expect, that reliable and fail safe operation is guaranteed by the manufacturer. But please 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.

Be sure to take proper precautions before removing the Ethernut board from the anti-static bag. When not used, put the board back into the anti-static bag. Never pass the bare board from one person’s hand to another.

Do not use the anti-static bag as a underlying pad for Ethernut, because it’s electro-conductive. Plastic surfaces may be harmful too because of electrostatic discharge. It is advisable to put the board at least on a wooden surface or simply on a piece of paper. The optimal way is to fix stand-offs in the mounting holes.

Board Installation

1 Remove the board from the anti-static bag. Visually inspect it for any damage done during shipment. If there are visible defects, return the board for replacement.

2 Connect Ethernut`s DB-9 RS232 port to an available COM port using the serial cable included in the starter kit. Any null-modem cable should work as well.

3 Use one twisted pair cable (patch cable) to connect Ethernut's RJ-45 connector to the hub or switch. Make sure that the PC is connected to the same physical Ethernet network. Ethernut 3.1 comes with Auto-MDIX and can be connected directly to the PC with a standard patch cable. However, depending on the PC's operating system, link negotiation may not work reliably in this configuration.

4 Connect the power supply to the barrel connector on the Ethernut Board. Ethernut is equipped with its own rectifier bridge and voltage regulator. Therefore the polarity of the barrel isn't important.

5 Apply power to the Ethernut Board by connecting the power supply to an electrical outlet. When power is present, you should observe the red LED at the reset button is on. The reset button is the white push button at the board’s edge near the MMC socket.

6 Start the terminal emulation program at 115200 baud, no parity, 8 data bits, and 1 stop bit. Disable hardware (RTS/CTS) and software (XON/XOFF) flow control.

7 Reset the Ethernut board by pressing and releasing the reset button. Hold down the space-bar on the terminal emulation program and wait until the BootMon welcome message is displayed.

In the beginning it is recommended to use BootMon, which is very easy to use and much faster than any other method to upload the compiled code from the PC into the Ethernut 3 RAM. Later you may switch to the JTAG interface for flash programming and debugging.

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

Using the Boot Loader

As explained in the previous chapter, hold down the space-bar on the PC keyboard and press and release the reset button on the Ethernut Board. After a few seconds the following output should appear in the terminal emulation window:

```
BootMon 1.0.0
MAC address (000698300000):
```

The version that has been preloaded on your board may be higher and the MAC address will differ.

A MAC address, also referred to as the hardware or Ethernet address is a unique 48 bit hexadecimal 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 GmbH is 000698 hexadecimal. The lower 24 bits are the board's unique ID assigned by egnite. It is printed on the small barcode label.

The bootloader is resident, which means, that it is started each time you apply power to the board or release the reset button. However, it will normally work silently in the background, using any previously entered configuration. To change this configuration, a space character must be send to the serial port immediately after starting. Let's enter a new configuration now.

The MAC address should have been correctly set. Press enter to confirm this parameter.

BootMon will now ask for the IP Address of the board.

```
IP address (0.0.0.0):
```

If your network provides a DHCP service, you can leave the IP address at all zeros. However, for a first test it is recommended to specify an individual address, which fits to your local network. For example, let's assume, that your PC has the IP address 192.168.192.1, then 192.168.192.2 should work, if no other device in your network is using this address.

Enter the unique IP address and press enter. BootMon will prompt for the network mask.

```
Net mask (255.255.255.0):
```

You must use the same network mask as it is used with your PC. In general, all nodes within a local IP network must have the same network mask. After entering a network mask, BootMon asks for a default route.

```
Default route (0.0.0.0):
```

This should be the IP address of your router, used to connect to the Internet. This information is only required, if the Ethernut board will be accessed from or will access another node via the Internet. For now we can leave this entry at 0.0.0.0, which means, that no Internet gateway will be used.

Finally BootMon asks for the TFTP Server IP and for the name of the image to load from this server.

```
TFTP IP (192.168.192.1):
TFTP Image (threads.bin):
```

The IP address should be the one of your PC. The name of the image is actually the name of a binary file containing the application we want to run on the board.

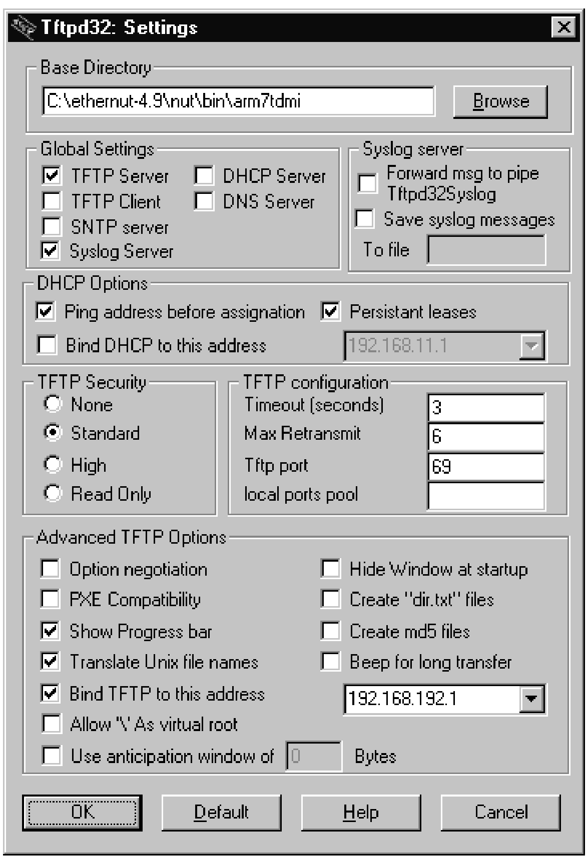
After pressing enter to confirm the TFTP image name, all parameters will be stored in non-volatile configuration memory and BootMon will immediately try to load the specified file from the specified TFTP server each time you power-up or reset the board.

TFTP Server

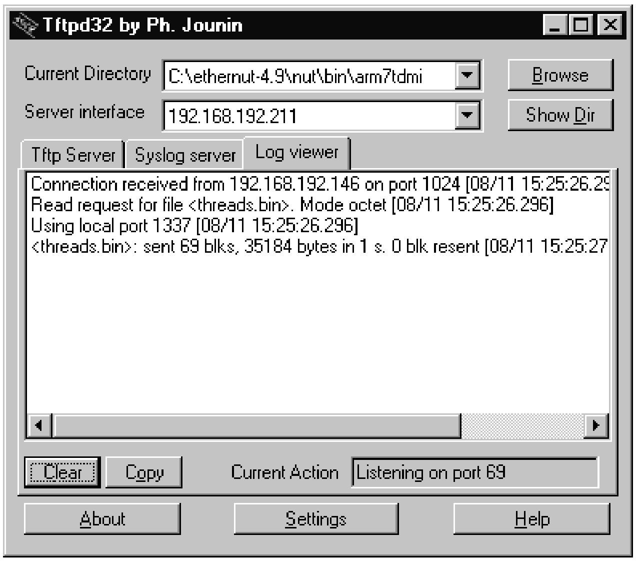
There are a number of TFTP servers available, many of them are free of charge.

With Linux and Mac OS X, a TFTP daemon is typically available as a binary package. On many Linux distributions it is installed already, it just needs to be activated in the network configuration.

For Windows, TFTPD32 is a good choice. The following screenshot shows a sample configuration of TFTPD32. The Base Directory points to the directory that contains the compiled Nut/OS application binaries.



After pressing the reset button, BootMon will request the binary image from the TFTP server running on your PC. On TFTPD32 the following should appear:



You may now follow the Nut/OS Software Manual and learn how to install the software development environment. Or you may visit the Wiki pages at

http://www.ethernut.de/nutwiki/Nut/OS_Examples

to download and try a few ready-to-use binaries.

Using the JTAG Interface

JTAG programming is required to initially burn the bootloader into flash memory or to permanently store application code. In the latter case, boot loading will be skipped and the application will immediately start after reset.

Prerequisites for Flash Programming

To connect to the Ethernut's JTAG interface, a JTAG adapter is required. Ethernut 3 uses the same JTAG connector layout as its predecessors Ethernut 1 and 2: A 10-pin, dual-row, 0.1-inch male connector. Unfortunately the same type of connector is used by Atmel and Ethernut 1 for the AVR SPI programming interface.

Warning: Ethernut 3 doesn`t support programming with SPI adapters. Connecting an AVR SPI programming adapter to the Ethernut 3 JTAG connector will at least blow the board`s fuse or in worst cases damage your Ethernut 3.

Ethernut's JTAG connectors also differ from the standard 14-pin or 20-pin connectors used on most other ARM based boards. A simple cable adapter can be used to attach any standard ARM JTAG adapter. Check the connector pinout that is shown in the board description below.

The Turtelizer JTAG Adapter is included in the Ethernut Starter Kit and has the right connectors. It is connected to the PC via USB and additionally provides a USB to COM Port Bridge, which can optionally be used to connect the PC with Ethernut's RS-232 interface.

Together with the adapter hardware you will need a related software tool, which allows to program CFI compatible flash memory devices via JTAG. The Open Source project OpenOCD offers such a tool and is available on the starter kit CD. It works with the Turtelizer and several other adapters.

A special version of OpenOCD is supplied with the Turtelizer 2. It allows to optionally use a proprietary driver based on the USB driver from FTDI Ltd. In order to conform to OpenOCD's license, the Turtelizer's USB library must be installed separately.

Installing the JTAG Adapter

Follow these steps to setup Ethernut 3 for JTAG programming:

1. If not already done, install the Ethernut software package. Details are explained in the Nut/OS software manual.
2. Install any required driver for your JTAG adapter. A special USB driver is provided with the Turtelizer 2 adapter.
3. Install OpenOCD. You may use the special version supplied with Turtelizer 2.
4. Remove the power supply from the board. Never attach or detach a JTAG adapter on a powered board.
5. Ensure that jumper JP5 is properly configured to access the CPU's JTAG interface. The jumper will have been set correctly when the board was shipped.

6 Ensure that the double slide switch (S2), located between the JTAG connector and the reset button, is in default position (both sliders moved towards the JTAG connector).

7 Connect the Turtelizer JTAG cable to J7. In case of different hardware, use a cable adapter.

8 Re-apply power to the board.

9 Connect the other interface of the JTAG adapter to the PC. For the Turtelizer, this is the USB interface.

When connecting an USB based adapter like the Turtelizer 2 to a Windows PC for the first time, the "New Hardware Detected Wizard" will pop up. Follow its instructions.

JTAG Jumper

The JTAG connector can be configured by jumpers on JP5. The most common settings are:

CPU JTAG

In the default configuration of JP5, the JTAG interface of the AT91R40008 CPU is available at the JTAG connector J7. Pins 3 and 5, 6 and 8, 15 and 16 as well as pins 19 and 20 are shortened. This configuration is also used to program the NOR flash memory.

CPLD JTAG

Routing the JTAG interface to the CPLD is provided by shortening pins 2 and 4, 7 and 9, 12 and 14 as well as pins 15 and 17. A special Xilinx compatible adapter is needed for programming the CPLD or JTAG boundary scanning.

Internal JTAG

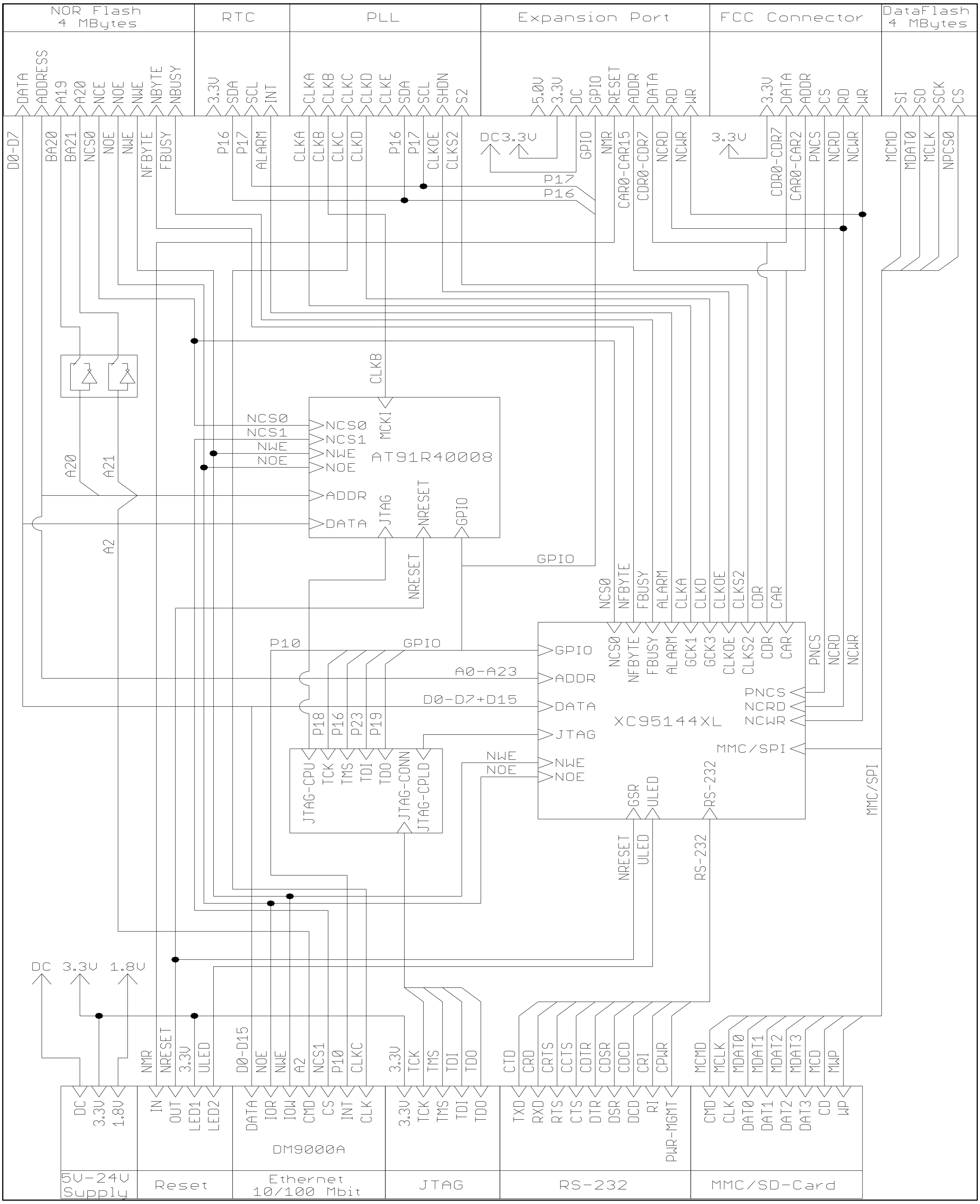
As an alternative, the CPU is able to program the CPLD. The required jumper configuration is shown on the left.

Combined CPU and Internal JTAG

This combined configuration allows you to download the CPLD programming software to the CPU via JTAG. As soon as this software starts running, it will program the CPLD.

Board Overview

The block diagram shows the interconnections between the main components. The two central parts are the microcontroller AT91R40008 and the programmable logic device XC95144XL.



AT91R40008 Microcontroller

The AT91R40008 CPU (IC1) provides 256 kBytes of high speed 32-bit on-chip SRAM, which can be used for code execution and data storage. For a detailed description please refer to the related datasheet.

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

XC59144XL Programmable Logic Device

The Ethernut 3 Board includes a Complex Programmable Logic Device (CPLD), IC3. The CPLD is accessed through the CPU's External Bus interface (EBI). Its internal functions can be controlled by the CPU through memory-mapped registers.

Initially after shipping, the CPLD provides the following functions:

- Expansion port memory bus with 8-bit data and 16-bit address
- Controlling and monitoring of the RS-232 interface
- User LED control
- SPI master for MMC and DataFlash

Please refer to the chapter "Memory Map" for more details about the initial register layout.

Re-programming the CPLD is possible by attaching a Xilinx compatible JTAG adapter to the Ethernut Board. Alternatively, a so called XSVF Executor together with an XSVF files may be uploaded to the internal RAM via the BootMon bootloader.

Almost all important internal signals of the Ethernut Board are routed to the CPLD. Thus, the chip can be used for hardware testing via JTAG boundary scan.

Power Supply

The I/O logic of the Ethernut 3 board is driven by a 3.3V power supply, while the CPU core runs at 1.8V. The board provides its own voltage regulators (IC8 and IC9). It only requires an unregulated power supply of 5V - 24V DC with a minimum current of 200 mA.

Two 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 (D1) and a rectifier bridge (D4, D5, D6 and D7).
2. 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. This input is unprotected.

As soon as power is attached to any of the inputs, the red LED at the reset button will light up.

System Reset and LEDs

A specific reset controller (IC10) is used to monitor the supply voltage and keep the system in reset state unless sufficient supply voltage is reached. A reset push button is available to manually trigger the reset state.

The Ethernut 3 board is equipped with four LEDs. Two of them are integrated in the reset button, a red LED to indicate power supply and a green LED, which is available for user applications. Another two LEDs are integrated in the RJ45 Ethernet connector (see below).

Ethernet Interface

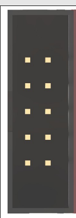
The Ethernet controller is a Davicom DM9000A (IC2). It is physically attached to the 16-bit memory bus. Ethernut provides an on-board modular RJ-45 connector with an integrated 100/10Base-T transformer/filter for its twisted pair Ethernet port. The interface supports the maximum cable length of 100 meters between the Ethernet board and a hub or switch.

Two LEDs are integrated with the RJ45 Ethernet connector. The yellow LED indicates the 10/100 Mbit link status and is lit when connected to 100 Mbit Ethernet. The green LED indicates receive and transmit activity from and to the network.

JTAG Interface

The Ethernut board has an industry standard IEEE 1149.1 Test Access Port. The JTAG port specification was initially designated as a test header. On Ethernut 3 you can perform standard boundary scan for the XC95144XL. Furthermore, the JTAG interface can be used to program the CPLD and the NOR flash memory. Finally it provides a debugging interface to the AT91R40008 CPU.

On all Ethernut Boards the same 10-pin, dual-row, 0.1-inch male connector is used for JTAG, which is the one specified by Atmel for the AVR microcontrollers and which is used on most AVR based boards.

Ethernut 10-Pin JTAG Connector				
TCK	1		2	GND
TDO	3		4	VTref
TMS	5		6	nSRST
Vsupply	7		8	NC
TDI	9		10	GND

A pin header selects routing of the JTAG signals to either of the devices by using jumper shunts. For historical reasons this jumper block is named JP5, JP1 to JP4 do not exist.

RS-232 Interface

Ethernut provides an on-board DB-9 male connector for RS-232 serial communication, wired as a DTE (data terminal equipment) port. IC6 is used to convert the required voltage levels for RS-232 from the 3.3V power supply.

Any of the two serial interfaces of the microcontroller can be routed to the RS-232 connector via selection registers within the CPLD. In the default configuration the first interface (UART0) is routed to the RS-232 connector while the second interface (UART1) is not used.

The following table shows the connector's pinout.

Pin	Signal	I/O	Alternate Function
1	DCD	In	
2	RxD	In	
3	TxD	Out	
4	DTR	Out	Secondary channel TxD
5	GND		
6	DSR	In	Secondary channel RxD
7	RTS	Out	
8	CTS	In	
9	RI	In	5V - 24V supply when R1 is removed and R101 is mounted

Use a null-modem cable to connect a PC to this port. A suitable cable will have female 9-pin D-Sub connectors on both ends. The following table shows the cable wiring:

Connector A Pin	Connector B Pin
1+6	4
2	3
3	2
4	1+6
5	5
7	8
8	7

By default 8 data bits, no parity, 1 stop bit and 115,200 baud without handshake will be used. The bootloader firmware supplied with the Ethernut board expects this configuration.

MMC/SD-Card Socket

The default CPLD configuration implements a standard SPI-mode MMC/SD Card interface, but a 9-pin MMC/SD Card socket provides access to additional signals.

Pin	Signal	I/O	SPI Function
1	CD/DAT3	Out	Card select (CS)
2	CMD	Out	Data to card (DI)
3	GND		Digital ground
4	VDD		Power Supply
5	CLK	Out	Data clock
6	GND		Digital ground
7	DAT0	In	Data from card (DO)
8	DAT1		Not used (IRQ)
9	DAT2		Not used

NOR Flash Memory

The AT91R40008 provides no on-chip flash memory. An external flash memory chip was added to permanently store program code and read-only data. This memory is organized as 4M x 8 bits and can be (re-)programmed by the CPU and via JTAG. Note, that each 32-bit access is performed by 4 successive accesses with additional wait states to the 8-bit NOR flash. Due to the low performance of flash memory accesses the image to be executed should typically be copied into RAM first.

A double slide switch (S2), located between the JTAG connector and the reset button, divides the 4MBytes into four pages, each of which may be placed at reset start address 0x00000000 (before remap). This allows to have up to 4 different boot images available. The switch selects either the original or the negated signal of address bits 20 and 21. This way the following absolute flash memory addresses are moved to the CPU address space (after remap):

SW A20	SW A21	Chip Address	Chip Address
Ori	Ori	0x00000000	0x10000000
		0x00100000	0x10100000
		0x00200000	0x10200000
		0x00300000	0x10300000
Neg	Ori	0x00100000	0x10000000
		0x00000000	0x10100000
		0x00300000	0x10200000
		0x00200000	0x10300000
Ori	Neg	0x00200000	0x10000000
		0x00300000	0x10100000
		0x00000000	0x10200000
		0x00100000	0x10300000
Neg	Neg	0x00300000	0x10000000
		0x00200000	0x10100000
		0x00100000	0x10200000
		0x00000000	0x10300000

The address line is negated, if the slide switch is moved towards the reset switch and not negated if moved towards the JTAG connector. The slider nearest to the edge of the board switches address bit 20. During flash programming the sliders should be put in default position (not negated addresses).

RTC

The NXP PCF8563 Realtime Clock/Calendar Chip (IC7) is accessed via a TWI (I2C). However, as the CPU doesn’t offer any TWI hardware, software bit-banging via GPIO pins P16 (data) and P17 (clock) must be used.

The chip's power supply is backed by a 0.33F double layer cap (C1). A dedicated 32.768kHz crystal (Y2) drives the reference clock.

System Clock PLL

The CY22393 PLL (IC5) uses a 25MHz crystal as a reference clock to generate all other clocks. The initial settings are stored in internal non-volatile registers, which are not in-system-programmable. On power-up, the contents are copied to RAM registers, which may be modified via the TWI (I2C) bus when the board is up and running. As the CPU doesn't offer any TWI hardware, software bit-banging via GPIO pins P16 (data) and P17 (clock) must be used to access the registers.

Warning: Playing around with the clock chip on the Ethernut 3 Board is fun, but also bears the risk of destroying your hardware due to overclocking.

When shipped with the Ethernut Board, the microcontroller clock is configured at 73.728 MHz, while the Ethernet Controller is fed by a 25 MHz clock signal. Two additional clocks are available for the CPLD, both of which are initially disabled.

The CY22393 PLL also has two control inputs, nSHDWN and S2, which are connected to the CPLD. While the first one is typically not used, the second one allows to switch the CPU clock from 73.7273MHz to 14.7455MHz (default configuration).

The following table shows the PLL configuration and usage.

Output	Connected to
Clock A	CPLD GCK1, initially disabled. The Nut/OS NPLMMC driver will re-program this output for MMC SPI clocking.
Clock B	73.7273MHz to CPU, if S2 is driven high (default). Driving S2 low will change the frequency to 14.7455MHz without any glitches.
Clock C	25MHz to Ethernet Controller, independent of S2 settings.
Clock D	To CPLD GCK3, initially disabled. It can be activated by reconfiguring the clock chip via its I2C interface.
Clock E	Not connected.
XBUF	Not connected.

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 CPU, using the Ethernut board as an Ethernet I/O processor only.

Please refer to the chapter "Hardware Expansion" for more details.

FFC Connector

A 24-wire flexible flat cable (FFC) with 0.5mm pitch can be attached to connector J4. By default it provides an 8-bit databus (CDR) with 3 address bits (CAR), read and write signals and a chip select line. These signals are shared with the related pins at the expansion port connector. In a typical application this may be connected to a front panel with LCD and a few buttons. However, all signal lines are CPLD controlled and may be redefined to meet your specific needs.

Pin	Signal
1	+3.3V
2	GND
3	CDR7
4	CDR6
5	GND
6	CDR5
7	CDR4
8	GND
9	CDR3
10	CDR2
11	GND
12	CDR1
13	CDR0
14	GND
15	nCWR
16	nCRD
17	GND
18	PNCS
19	CAR0
20	GND
21	CAR1
22	CAR2
23	GND
24	+3.3V

DataFlash Memory

The AT45DB321D serial flash offers an additional 4MBytes flash memory. The BootMon bootloader uses the last page for configuration data storage.

This chip is accessed by the CPU via the same SPI interface as the MMC socket.

Memory Map

The following table shows the memory map which is configured by the BootMon boot loader.

Addresses	Chip Select	Device
0x00000000 - 0x0003FFFF		Internal SRAM
0x10000000 - 0x103FFFFF	NCS0	External NOR Flash
0x20000000 - 0x200FFFFF	NCS1	Ethernet Controller
0x21000000 - 0x210FFFFF	NCS2	CPLD Registers
0x22000000 - 0x220FFFFF	CS4	Expansion Port Memory Bus
0xFFC00000 - 0xFFFFFFFF		On-chip Peripheral Registers

Memory Map After Reset

After reset, the external flash memory is located at address 0x00000000 and internal SRAM is located at address 0x00300000. The CPU fetches its reset vector from the first 4 bytes in flash memory and expects the exception vectors immediately thereafter. During normal operation, it is preferable to have the vectors in RAM. The internal control logic of the CPU accommodates this requirement by providing a remap command, which switches RAM location to address 0x00000000.

On the Ethernut 3 this is done by the BootMon bootloader. It is not possible to reverse this address map switching.

CPLD Registers

By default, the following registers are implemented at base address 0x21000000.

Offset	Register	Symbol	Access	Reset State
0	NPL RS232 Command Register	NPL_RSCR	NPL_RSCR	0x20
0	NPL Interrupt Mask Register	NPL_IMR	NPL_IMR	0x0000
0	NPL Signal Latch Register	NPL_SLR	NPL_SLR	0x0000
0	NPL Signal Clear Register	NPL_SCR	NPL_SCR	
0	NPL MMC Data Register	NPL_MMCDR	NPL_MMCDR	0x00
0	NPL External Enable Register	NPL_XER	NPL_XER	0x07
0	NPL Version ID Register	NPL_VIDR	NPL_VIDR	0x02

NPL RS232 Command Register

This register controls the RS232 handshake outputs and routes the specified USARTs to the RS232 driver. Two additional bits can be used to permanently switch the RS232 driver on or off.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSUSP	RSUS1	RSUS0		RSRTS	RSDTR	RSFOFF	RSFON
Reset	0	0	1	0	0	0	0	0
Access	RW	RW	RW	R	RW	RW	RW	RW

RSFON — RS232 Force On

Set this bit to disable auto shutdown of the RS232 driver.

RSFOFF — RS232 Force Off

Set this bit to force the RS232 driver into shutdown mode.

RSDTR — RS232 Data Terminal Ready

Set this bit to activate the RS232 DTR line. If both USARTs are enabled (RSUS0E and RSUS1E both set to 1), then RSDTR is ignored.

RSRTS — RS232 Request To Send

Set this bit to activate the RS232 RTS line.

RSUS0E — RS232 USART0 Enable

When this bit is set, USART0 is connected to the RS232 driver.

RSUS1E — RS232 USART1 Enable

When this bit is set, USART1 is connected to the RS232 driver.

RSUS1P — RS232 USART1 Primary

If this bit is set, USART1 becomes the primary RS232 device. Otherwise USART0 is the primary device. The primary device uses Rx, Tx, RTS and CTS lines. If a secondary device is enabled, it will use the DTR line for the transmitter and the DSR line for the receiver.

NPL Interrupt Mask Register

This register is used to enable interrupts on the specified signals.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name					NMMCD	MMCD	NRSINV AL	RSINVAL
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	RW	RW	RW	RW
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MMRDY	FBUSY		ALARM	RSRI	RSDCD	RSDSR	RSCTS
Reset	0	0	0	0	0	0	0	0
Access	R	RW	R	RW	RW	RW	RW	RW

RSCTS RS232 Clear To Send

If set to 1, IRQ0 is pulled low when the CTS line becomes active.

RSDSR RS232 Data Set Ready

If set to 1, IRQ0 is pulled low when the DSR line becomes active. If both, RSUS0E and RSUS1E in the RS232 Command register are set to 1, then the RSDTR bit is invalid.

RSDCD RS232 Data Carrier Detect

If set to 1, IRQ0 is pulled low when the DCD line becomes active.

RSRI RS232 Ring Indicator

If set to 1, IRQ0 is pulled low when the RI line becomes active.

RSINVAL RS232 Invalid

If set to 1, IRQ0 is raised when the RS232 signals become invalid.

NRSINVAL Negated RS232 Invalid

If set to 1, IRQ0 is raised when the RS232 signals become valid

MMCD Media Card Detect

If set to 1, IRQ0 is raised when inserting a card.

NMMCD Negated Media Card Detect

If set to 1, IRQ0 is raised on card eject.

NPL Signal Latch Register

This read-only register reflects latches active signal states. If a signal becomes active, the corresponding bit will be set to 1.

If the same bits in this register and in the interrupt mask register are both 1, then the IRQ0 line will be pulled low. This can be used to invoke an interrupt routine.

To clear any latched bit, 1 must be written to the same bit in the signal clear register (NPL_SCR) after the signal becomes inactive again.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Name					NMMCD	MMCD	NRSINVAL	RSINVAL
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MMRDY	FBUSY		ALARM	RSRI	RSDCD	RSDSR	RSCTS
Reset	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R

RSCTS — RS232 Clear To Send

Set to 1 when the CTS line is active.

RSDSR — RS232 Data Set Ready

Set to 1 when the DSR line is active.

RSDCD — RS232 Data Carrier Detect

Set to 1 when the DCD line is active.

RSRI — RS232 Ring Indicator

Set to 1 when the RI line is active.

RSINVAL — RS232 Invalid

Set to 1 when all RS232 signals are invalid.

NRSINVAL — Negated RS232 Invalid

Set to 1 when the RS232 signals are valid.

MMCD — Media Card Detect

Set to 1 when a card is inserted.

NMMCD — Negated Media Card Detect

Set to 1 when no card is inserted.

NPL Signal Clear Register

Writing a 1 to any bit updates the corresponding bit in the signal latch register (NPL_SLR).

Bit	15	14	13	12	11	10	9	8
Name					NMMCD	MMCD	NRSINVAL	RSINVAL
Reset	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MMRDY	FBUSY		ALARM	RSRI	RSDCD	RSDSR	RSCTS
Reset	0	0	0	0	0	0	0	0
Access	W	W	W	W	W	W	W	W

RSCTS RS232 Clear To Send

If set to 1, updates the RSCTS bit in the signal latch register.

RSDSR RS232 Data Set Ready

If set to 1, updates the RSDSR bit in the signal latch register.

RSDCD RS232 Data Carrier Detect

If set to 1, updates the RSDCD bit in the signal latch register.

RSRI RS232 Ring Indicator

If set to 1, updates the RSRI bit in the signal latch register.

RSINVAL RS232 Invalid

If set to 1, updates the RSINVAL bit in the signal latch register.

NRSINVAL Negated RS232 Invalid

If set to 1, updates the NRSINVAL bit in the signal latch register.

MMCD Media Card Detect

If set to 1, updates the MMCD bit in the signal latch register.

NMMCD Negated Media Card Detect

If set to 1, updates the NMMCD bit in the signal latch register.

NPL MMC Data Register

This register is used to receive a data byte from and send a data byte to the Multimedia Card.

Reading the register must be done first. After writing, the data byte will be shifted out and a new data byte will be shifted in.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MMD7	MMD6	MMD5	MMD4	MMD3	MMD2	MMD1	MMD60
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	R	RW	RW	RW	RW	RW

NPL External Enable Register

This register is used to enable MMC select, panel select as well as the user LED.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name					DFCS	USRLED	PANCS	MMCS
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	R	RW	RW	RW	RW	RW

MMCS Controls the Multimedia Chip Select Line

PANCS Controls the Panel Connector Chip Select Line

USRLED Controls the User LED

Set to 0 to lit the LED.

DFCS Controls the DataFlash Chip Select Line

NPL Version ID Register

This register contains the Version ID of the CPLD configuration.

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0
Reset	0	0	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R

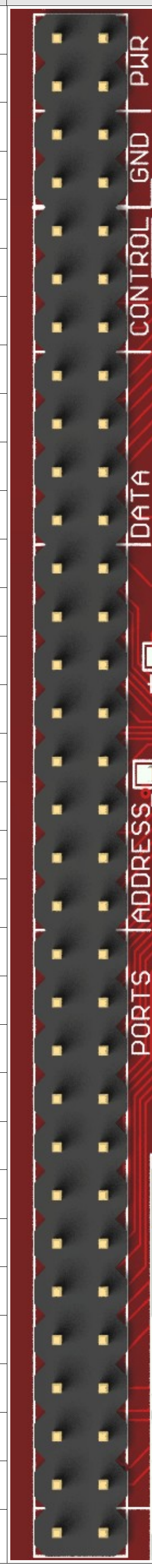
Hardware Expansion

Many applications will do just fine with nothing else than the Ethernut. If needed, external hardware may be connected to the RS-232 port. However, if more is required, the Ethernut expansion port is the primary choice to add custom designed hardware. This connector contains partial CPU data and address bus, memory read/write signals, digital I/O ports, reset signal and power supply.

Many signals are routed through the CPLD and may be redefined by re-programming this chip. Although available at the connector, some signals are used internally by Ethernut and can't be used by external hardware. Carefully check the schematic.

The following table lists the expansion port's pin assignment.

Warning: Pins 39 - 62 are not 5V tolerant and must not be connected to 5V logic without proper level translation. Pins 13 - 38 are 5V tolerant. For detailed specifications refer to the AT91R40008 and XC95144XL datasheets.



Pin	Signal	Function	Pin	Signal	Function
1	VCC3	+3.3V regulated	2	VCC3	+3.3V regulated
3	NC	Reserved	4	NC	Reserved
5	GND	Signal ground	6	GND	Signal ground
7	GND	Signal ground	8	GND	Signal ground
9	nMR	Reset input	10	DC	Unregulated supply
11	NC	Reserved	12	NC	Reserved
13	nCRD	Bus read strobe	14	nCWR	Bus write strobe
15	CDR0	Databus bit 0	16	CDR1	Databus bit 1
17	CDR2	Databus bit 2	18	CDR3	Databus bit 3
19	CDR4	Databus bit 4	20	CDR5	Databus bit 5
21	CDR6	Databus bit 6	22	CDR7	Databus bit 7
23	CAR0	Adressbus bit 0	24	CAR1	Adressbus bit 1
25	CAR2	Adressbus bit 2	26	CAR3	Adressbus bit 3
27	CAR4	Adressbus bit 4	28	CAR5	Adressbus bit 5
29	CAR6	Adressbus bit 6	30	CAR7	Adressbus bit 7
31	CAR8	Adressbus bit 8	32	CAR9	Adressbus bit 9
33	CAR10	Adressbus bit 10	34	CAR11	Adressbus bit 11
35	CAR12	Adressbus bit 12	36	CAr13	Adressbus bit 13
37	CAR14	Adressbus bit 14	38	CAR15	Adressbus bit 15
39	P15 RXD0	USART0 RxD	40	P14 TXD0	USART0 TxD
41	P13 SCK0	USART0 clock	42	P8 TIOB2	GPIO, Timer
43	P9 IRQ0	GPIO, IRQ	44	P27 NCS3	GPIO, CS
45	P11 IRQ2	GPIO, IRQ	46	P12 FIQ	GPIO, IRQ
47	P0 TCLK0	GPIO, Timer	48	P1 TIOA0	GPIO, Timer
49	P2 TIOB0	GPIO, Timer	50	P3 TCLK1	GPIO, Timer
51	P4 TIOA1	GPIO, Timer	52	P5 TIOB1	GPIO, Timer
53	P6 TCLK2	GPIO, Timer	54	P7 TIOA2	GPIO, Timer
55	P17	I2C clock	56	P16	I2C data
57	P22 RXD1	GPIO, USART1 RxD	58	P21 TXD1	GPIO, USART1 TxD
59	P23	GPIO	60	P20 SCK1	GPIO, USART1 clock
61	P19	GPIO	62	P18	GPIO
63	NC	Available	64	NC	Available

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 Nut/OS system software still supports all previous Ethernut Boards, even revision 1.1 with the ATmega103 microcontroller, which is no longer in production.

However, there are a few things to consider when moving from one board version to another.

Changes Compared to Ethernut 3.0 Rev-E

The Xicor X1286 Realtime Clock/Calendar Chip with integrated 32 kByte EEPROM has been replaced by the NXP PCF8563 and the AT45DB321D DataFlash. The latter is used by the bootloader and Nut/OS as non-volatile configuration storage. Due to this change, full binary compatibility is no longer provided. Applications written for Ethernut 3.0 must be rebuilt to work on Ethernut 3.1. In almost all cases this can be done without modifying the source code.

The Davicom DM9000(E) has been replaced by DM9000A with Auto-MDIX support. Both chips are register compatible, but LAN wakeup signal is no longer available. The related CPLD input pin is now used as DataFlash chip select output pin. Furthermore, the link indicator LED will only light up on 100 Mbit Ethernet connections. It is no longer lit at 10 Mbit networks.

A backward compatibility package is available, which allow binaries built for Ethernut 3.1 to run on Ethernut 3.0 without change.

Changes Compared to Ethernut 3.0 Rev-D

In addition to the changes compared to Rev-E listed above, the PLL clock outputs changed. Unless your code directly accesses the registers of the CY22393, this should not provide any problems.

Compared to Ethernut 3.0 Rev-E, the Xicor X1226 used on this board revision provides 512 Bytes EEPROM only.

Changes Compared to Ethernut 1 and Ethernut 2

The most important change to notice is the different CPU used on Ethernut 3. Existing applications must be recompiled. Depending on the hardware functions used by your application, further source code changes may be required. Note, that Nut/OS provides many API functions for writing fully portable applications. Using these functions allows to create one source code running on all supported platforms.

You can't use programming adapters shipped with previous starter kits, because Ethernut 3 needs a specific JTAG adapter for an ARM CPU. SPI programming is not supported. Never plug your SPI adapter into Ethernut 3 programming socket.

Also note, that the ARM7TDMI used on Ethernut 3 has a different I/O port layout compared to the AVR-based boards. Care has been taken to keep the expansion port as compatible as possible. However, the AT91R40008 outputs use 3.3V switching levels, and the inputs are not 5V-tolerant. They cannot withstand 5V switching levels. The XC95144XL CPLD outputs use 3.3V levels as well, but its inputs are 5V-tolerant.

Troubleshooting

At some time in the life of your Ethernut it may suddenly cease functioning. This can happen due to bad configuration or hardware trouble. This chapter provides some hints on tracking down the problem.

Before proceeding, remove any hardware attached to the expansion port and re-check the precautions in the chapter "Quick Start".

Basic Checks

If the red LED is not lit, make sure that the mains adapter used for power supply is working. Some adapters expect a 2.5mm center pin at the barrel connector, while the one mounted on Ethernut is 2.1mm. This may give an unreliable contact.

If the power supply is working, the fuse may be blown. The fuse is manufactured by Littelfuse, it's catalog number is 0453.500. The boards are shipped with a spare fuse. To avoid blowing the next fuse again, move forward to the next chapter, describing some advanced checks.

Even if the red LED is lit, you should still replace the power supply in the first place. Some of them have too high ripple voltage, others do not output the voltage level they claim to do. In rare cases the start-up ramp of the output voltage may let the CPLD initialization fail.

If you can't get any output from BootMon, check the LEDs at the Ethernet connector. When attached to a 100 Mbit network, the yellow LED should light up a few seconds after powering up the board. Even on a 10 Mbit network, you should notice the green LED blinking in short intervals. If this is the case, then the problem is most probably the RS-232 cable or the terminal emulator. Ensure that the COM port parameters are set to 115200 baud, no parity, 8 data bits, and 1 stop bit and that all handshakes are disabled. If all is configured correctly and if it still fails, try a different emulator program, replace the cable and finally try another PC.

If the yellow LED at the Ethernet connector is lit, or, in case of 10 Mbit Ethernet, the green LED is lit sporadically, then the Ethernet hardware is most probably working. If BootMon is not connecting the TFTP server, verify the network configuration. Consider to install the Open Source network sniffer Wireshark on your PC. It will be most useful when tracking down network problems.

Advanced Checks

These checks require additional equipment, which may not be available for you or may require specific knowledge you do not have. If this is the case and if your board is still showing no signs of life, contact either your distributor or egnite directly.

In cases of power supply problems, use a lab power supply with current control. Replace the fuse on the Ethernut board and carefully increase the voltage, starting at the minimum. The board should never draw more than 150 mA. Higher currents must be considered as shortcuts. Several chips are separated from the power supply plane by ferrite beads. Replacing them one after the other may help to find the location of the problem.

If the current is within limits, you can check the board's internal voltages at the two test points near IC9. The one that is nearer to the expansion port should provide 1.8V, while 3.3V should be available on the other.

If the power supply is OK, check the level of the NRESET signal (see schematic page 4).

An oscilloscope may be used to check crystal clocks and the PLL outputs. See the board's hardware description in this manual for the expected frequencies.

Warranty

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 this 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.
- BootMon 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 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.

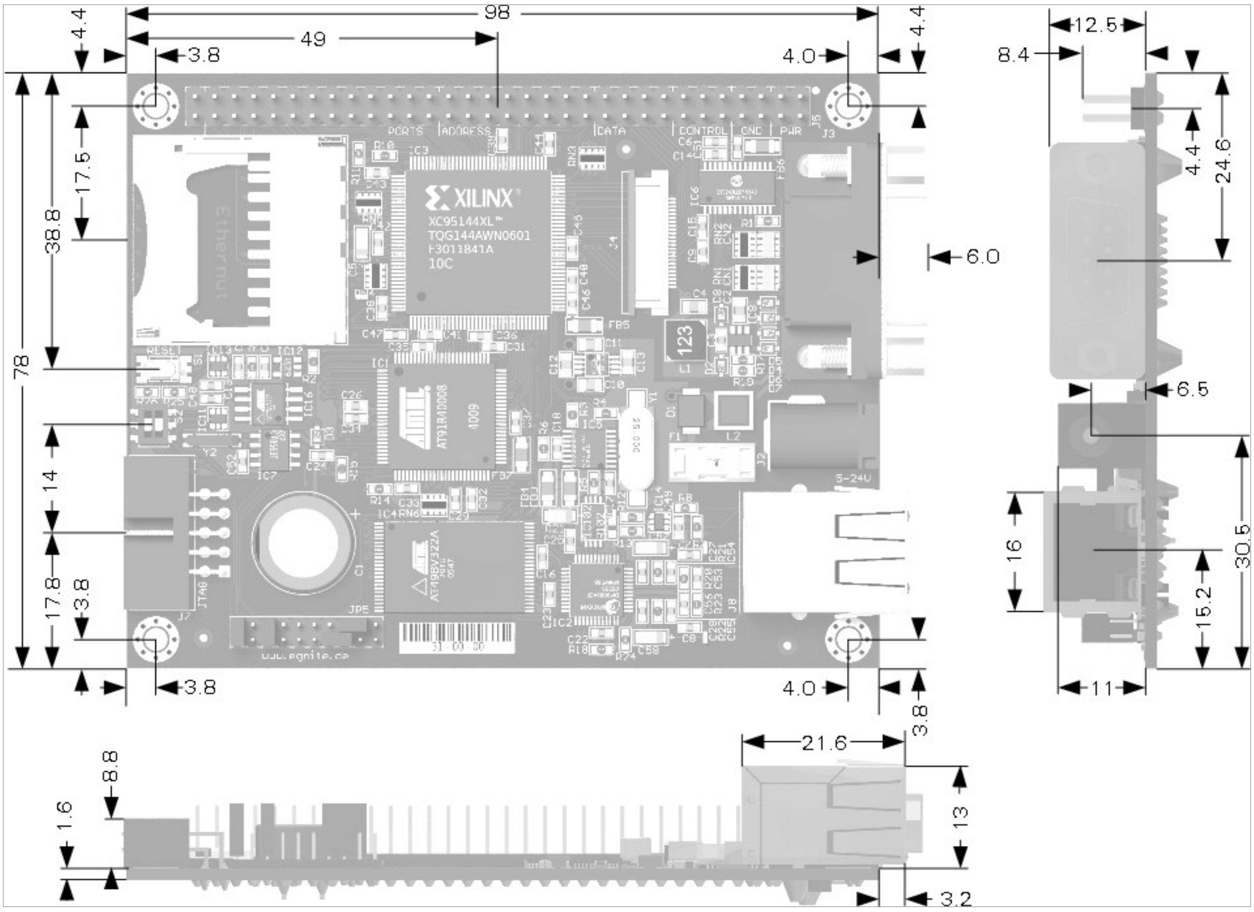
Anyway, whatever happened, we will do anything possible to revitalize your Ethernut. Or, if it finally passed away, let it rest in peace and send a replacement back to you at the least possible costs.

Technical Data

Processor	
CPU	AT91R40008, 73.728 MHz Clock
Flash memory	External 4 MByte
Static RAM	Internal 256 kByte
Serial Flash	External 4 MByte
RTC	Hardware (PCF8563T)
Interfaces	
Ethernet	RJ-45 10/100BaseT (DM9000A)
RS-232	1 x 9-pin DCE, 8-Wire
Digital I/O	17 configurable GPIO lines with alternate functions
Analog I/O	None available
Programming	10-pin JTAG
Indicators	Power (red), link (yellow), activity (green), user (green)
Power supply	
Regulator	600 mA Switcher (LT1616)
Input	2.1 mm barrel connector, unregulated 5 to 24 V DC
Expansion Port	5 to 24 V unregulated or 3.3 V regulated, output > 1 W
Consumption	< 1.2 W at 12 V
Battery backup	None
Protection	
RS-232	15 kV ESD protection
Ethernet	Transformer isolation
Power supply	1 A replaceable fuse, rectifier bridge, current limiter, thermal shutdown
Environmental	
Operating temperature	0 to 70 °C (32 to 158 °F)
Storage temperature	-65 to 140 °C (-85 to 284 °F)
Humidity	5 to 95 %, non-condensing
Approvals	
Safety	PCB flammability rating UL94-V-0
RoHS compliance	EU directive 2002/95/EC
Metrics	
Dimensions (L x W x H)	98 x 78 x 17 mm (3.86 x 3.07 x 0,67 in)
Weight	61 g (0.134 lb)
Product identification	
PCB revision	Written in copper on the PCB's backside
Serial number	IEEE registered MAC Address on barcode sticker label (Code 128C)

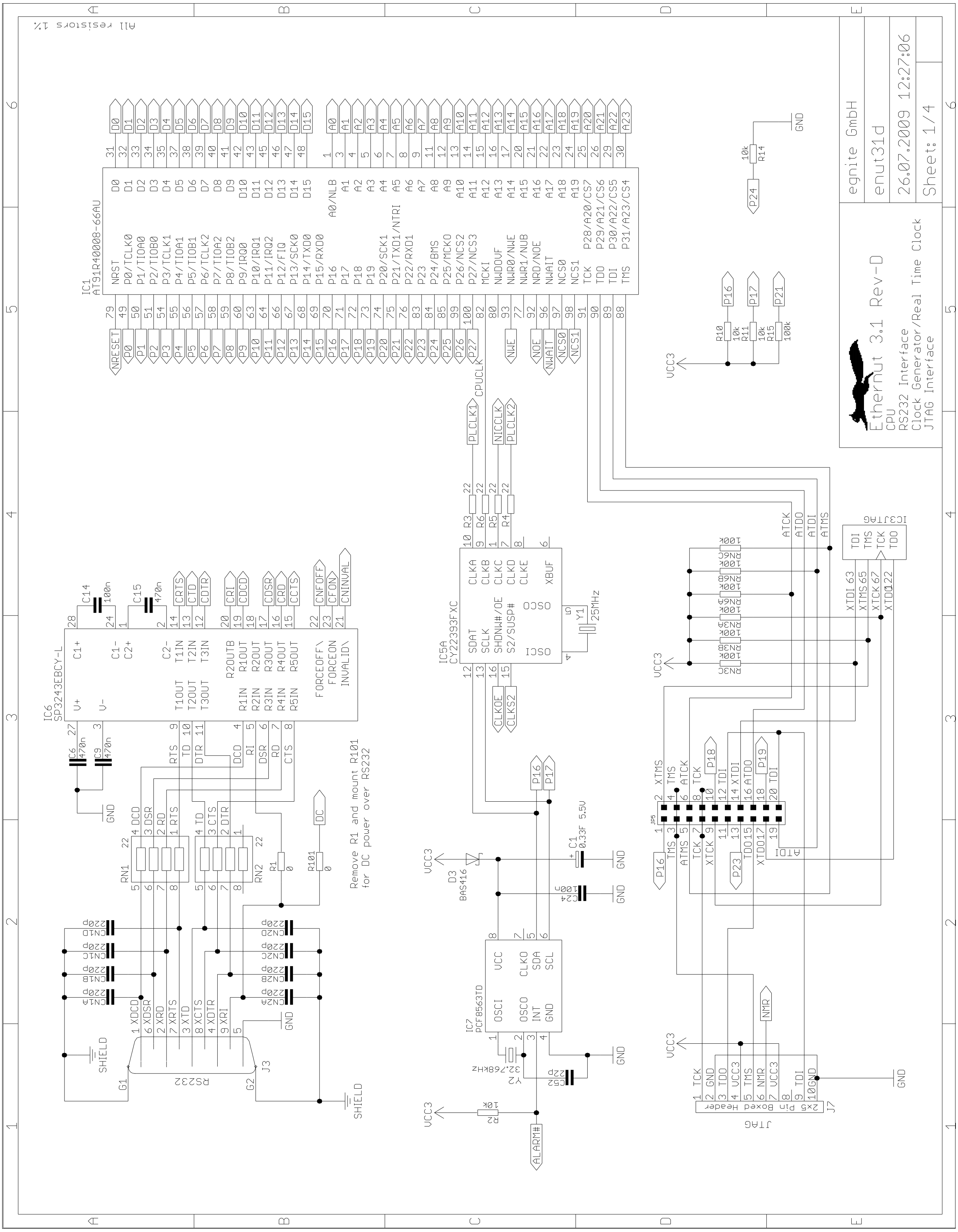
Board Layout

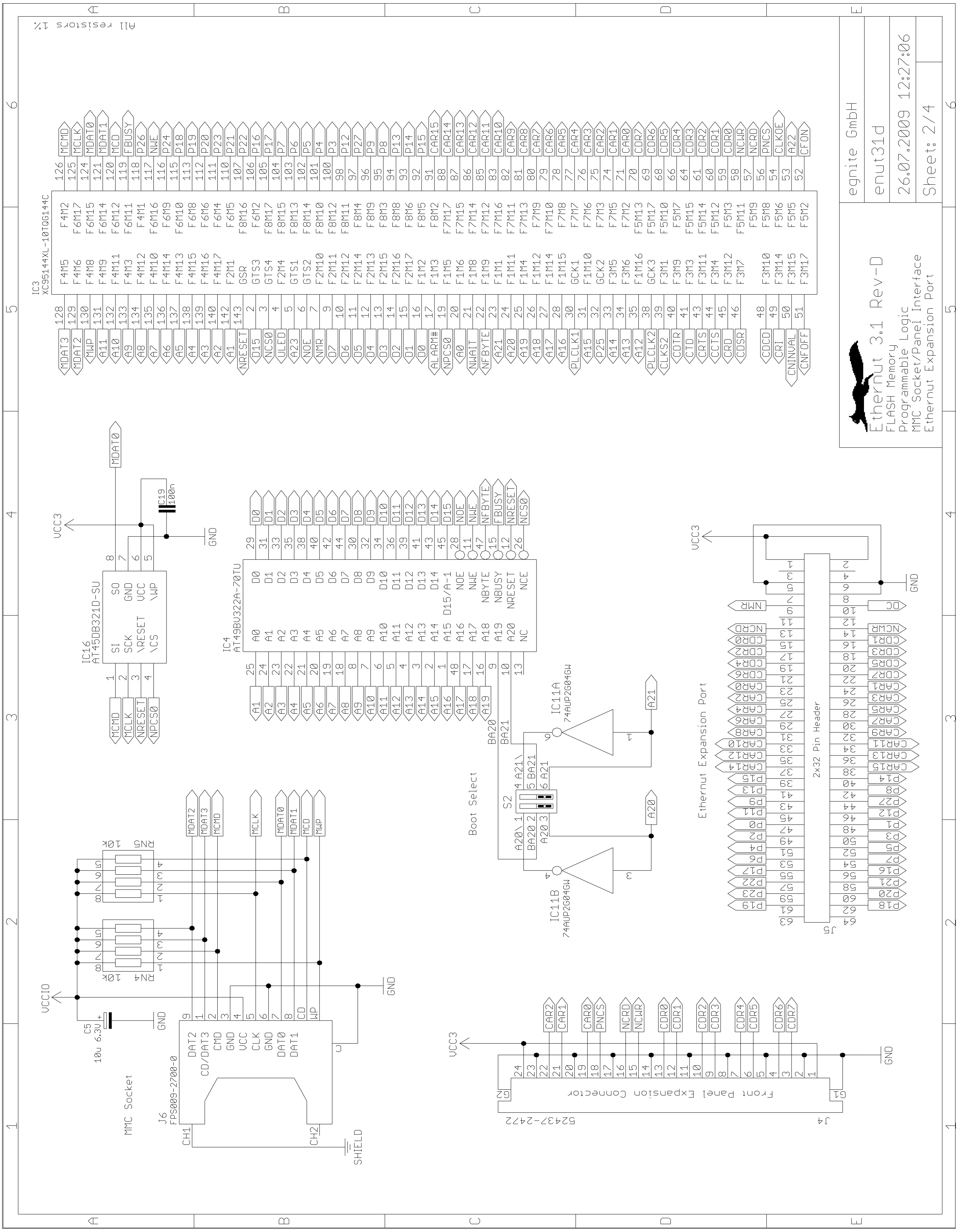
The Ethernut 3.1 printed circuit board is a 4-layer board using 6mil clearance and 5mil tracks. The following drawing shows the physical dimensions of the mounted board. Measures are in mm.



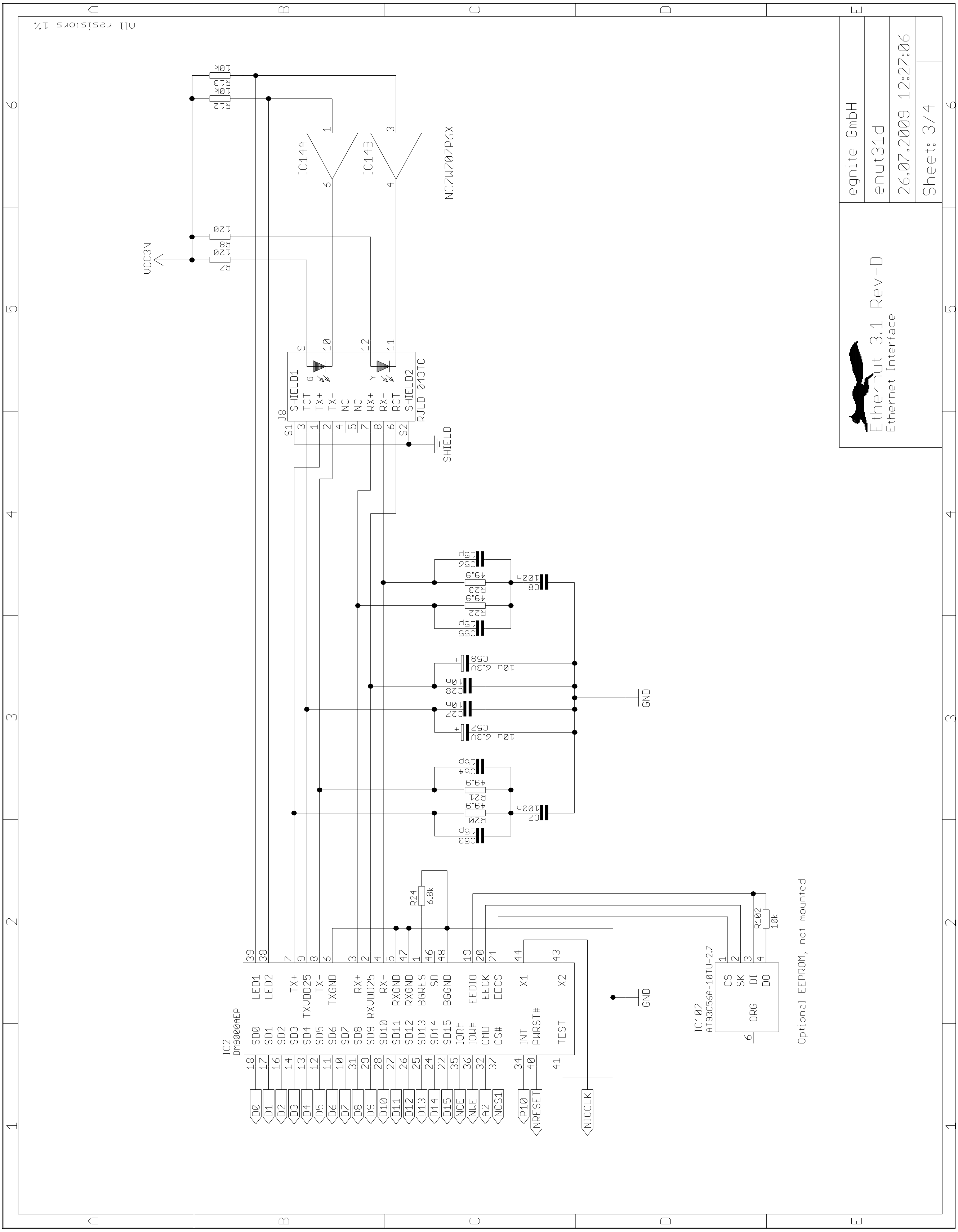
Ethernut has four 3mm mounting holes. Except for the mounting hole close to the Ethernet connector, all remaining holes are internally connected to the Ethernet, RS-232 and MMC socket shield. Conductive mounting bolts may be used in these positions to provide a good connection to a metal housing. This will minimize electrical and magnetic radiation.

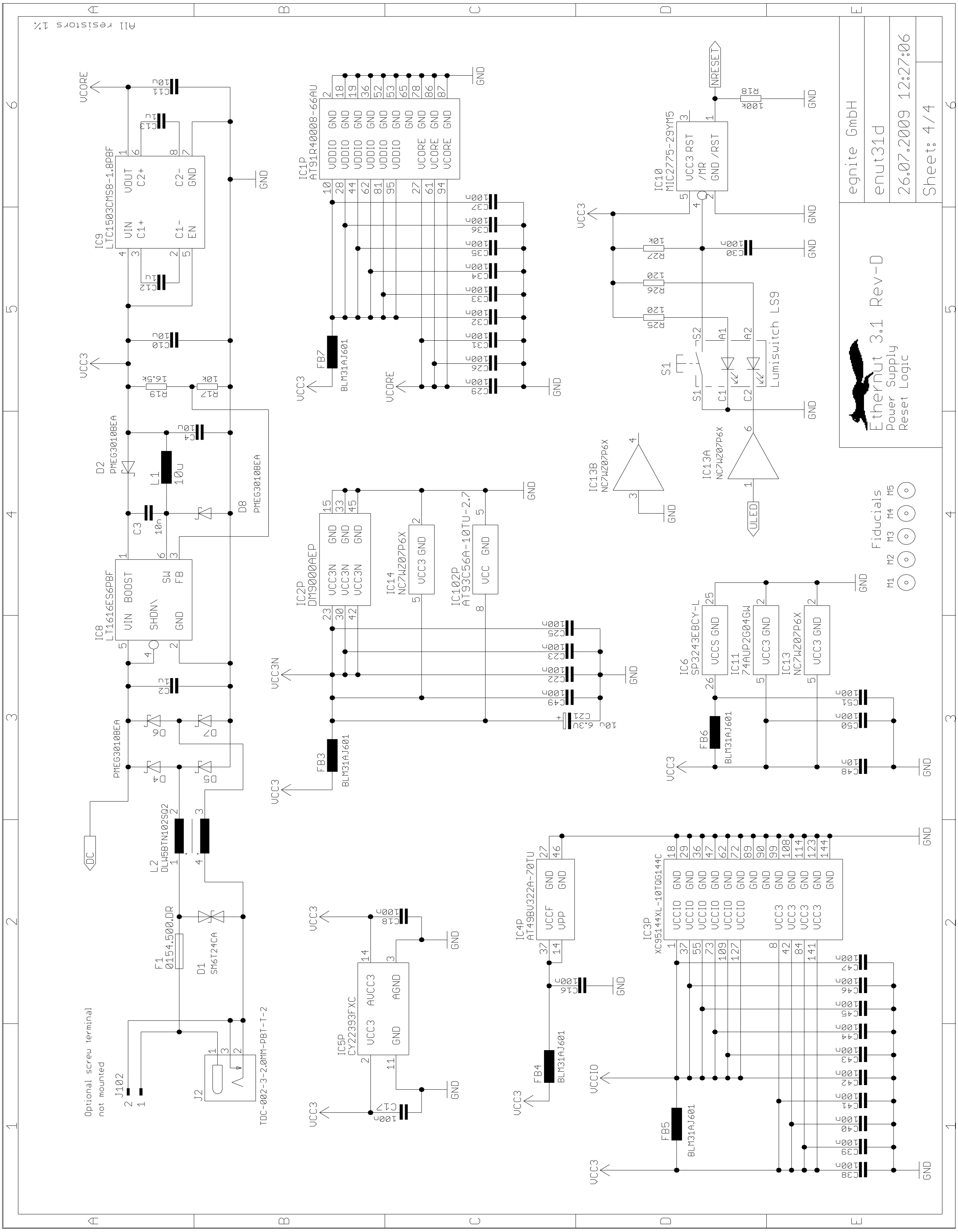
Schematics


All resistors 1%
egnite GmbH
enut31d
26.07.2009 12:27:06
Sheet: 1/4
Ethernut 3.1 Rev-D
CPU
RS232 Interface
Clock Generator/Real Time Clock
JTAG Interface
Remove R1 and mount R101
for DC power over RS232



Ethernut 3.1 Rev-D
FLASH Memory
Programmable Logic
MMC Socket/Panel Interface
Ethernut Expansion Port
egnite GmbH
enut31d
26.07.2009 12:27:06
Sheet: 2/4
All resistors 1%
MMC Socket
Boot Select
Ethernut Expansion Port
Front Panel Expansion Connector


egnite GmbH
enut31d
26.07.2009 12:27:06
Sheet: 3/4
Ethernut 3.1 Rev-D
Ethernet Interface
All resistors 1%
Optional EEPROM, not mounted


egnite GmbH
enut31d
26.07.2009 12:27:06
Sheet: 4/4
Ethernut 3.1 Rev-D
Power Supply
Reset Logic
All resistors 1%

egnite GmbH
Erinstrasse 9
44575 Castrop-Rauxel
Germany

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

info@egnite.de
www.egnite.de
www.ethernut.de