Ethernut 5.0

Hardware Manual

Preliminary

Manual Revision: 1.0
Issue date: May 2011

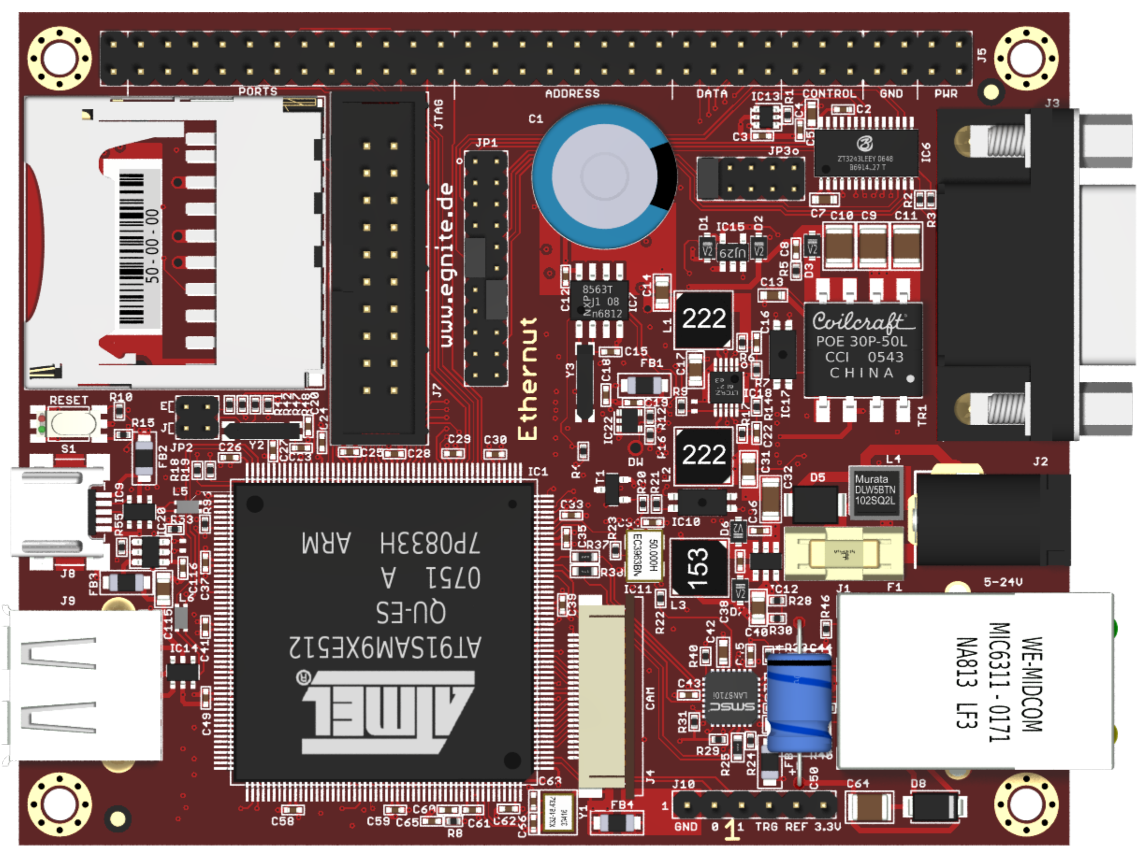
Contents

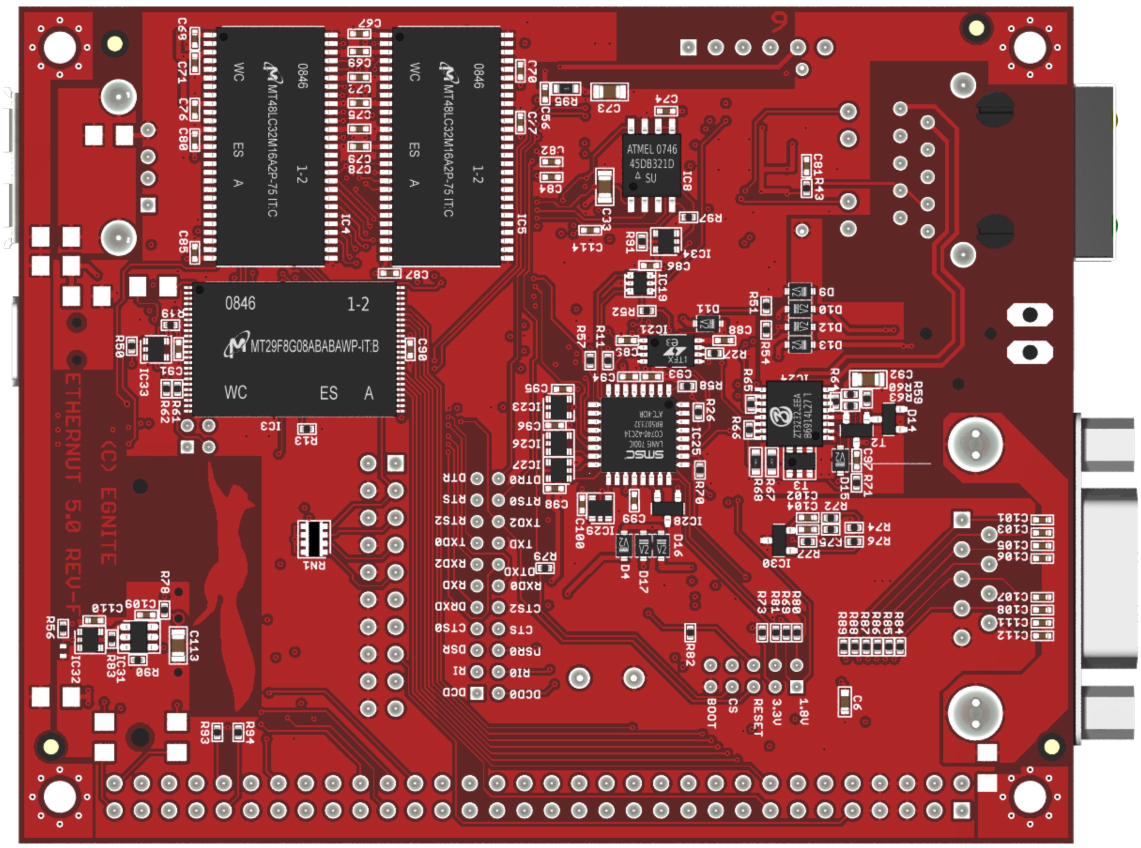
About the Ethernut 5.0 Board

The high end generation of the Ethernut board family has been designed for embedded Linux applications.

Ethernut Features

- AT91SAM9XE512 RISC microcontroller with 512 kByte protected NOR Flash, 32 kByte SRAM, 32 kByte ROM
- 1 GByte NAND Flash
- 128 MByte SDRAM
- 4 MByte serial Flash
- Multimedia/SD-Card Socket
- 10/100 Mbps Ethernet interface with Auto-MDIX
- USB 2.0 full speed device and host interface
- RS-232 at DB-9 connector with full modem handshake
- Image Sensor Interface
- 15 programmable digital I/O lines
- 2 analog input lines
- Real time clock with backup capacitor
- LED indicators for power supply and Ethernet activity
- Flexible power management including IEEE 802.3af PoE
- Lead-free and RoHS compliant
- Industrial temperature range: -25°C to +85°C





Quick Start

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

Prerequisites for Operation

Ethernut 5.0 boards are shipped with samboot boot loader programmed in internal Flash memory, U-Boot boot loader, Linux kernel and a sample Nut/OS application programmed in serial Flash memory and a Linux root file system programmed in NAND Flash memory.

The following items are included in the Ethernut Starter Kit:

- Ethernut 5 Board
- Turtelizer 2 JTAG programming adapter
- Crossed serial communication cable with DB-9 female connector at both ends
- USB cable Type A to Mini B
- 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, USB port and Ethernet interface
- Terminal emulation software, such as MiniTerm (Linux) or TeraTerm or Hyperterminal (Windows)
- Twisted pair cable together with 10/100 Base-T hub or switch

For power supply the following options are available:

- Connecting the Ethernut board to a USB port of your PC.
- Using an unregulated DC power supply providing a minimum 1.5 Watt at 7V, but not more than 24V (preferably 12V) on a standard 2.1 mm barrel plug.
- Using an IEEE 802.3af capable Power over Ethernet switch.

It is further assumed, that you got some basic knowledge about digital hardware, Linux or Nut/OS and TCP/IP networking. This manual will not present any of these basics. 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 antistatic bag. When not used, put the board back into the antistatic bag. Never pass the bare board from one person’s hand to another.

Do not use the antistatic bag as a underlying pad for Ethernut, because it’s electroconductive. 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

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 antistatic bag. Do not hand the bare board to another person.

Step 1: Remove the board from the antistatic bag. Visually inspect it for any damage made during shipment. If there are visible defects, return the board for replacement.

Step 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.

Step 3: On the PC, 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.

Step 4: 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 5.0 comes with Auto-MDIX and may be connected directly to the PC with a standard patch cable. However, depending on the PC's operating system, link negotiation may not work reliable in this configuration.

Step 4.1: If not connected to a PoE switch, you can use the USB cable for power supply. Connect the Mini-B connector to the Mini USB receptacle on the Ethernut board and the Type A connector to any USB port of your PC.

Step 4.2: Alternatively you may connect an external 7-24V, 1.5W power supply to the barrel connector on the Ethernut Board. Ethernut is equipped with its own rectifier bridge. Therefore the polarity of the barrel connector isn't important.

As soon as the board is powered up, the red LED at the reset switch will be lit. Then the SAMBoot boot loader firmware will initialize the hardware and move the U-Boot boot loader from serial Flash memory to the SDRAM and start it. U-Boot messages are displayed in the terminal emulator.

See the next chapter for a detailed description of the boot loader program.

Using the Boot Loader

As soon as the board is powered up, the U-Boot boot loader will send several status messages to the serial port, which are displayed on the terminal emulator.

```
U-Boot 2011.03-rc1 (Mar 10 2011 - 15:08:32)

CPU: AT91SAM9XE
Crystal frequency:   18.432 MHz
CPU clock        :  180.634 MHz
Master clock     :   90.317 MHz
I2C:   ready
DRAM:  128 MiB
Flash: 512 KiB
NAND:  1024 MiB
MMC:   mci: 0
DataFlash:AT45DB321
Nb pages:   8192
Page Size:    528
Size= 4325376 bytes
Logical address: 0xC0000000
Area 0: C0021000 to C0041FFF (RO) setup
Area 1: C0042000 to C00C5FFF (RO) uboot
Area 2: C00C6000 to C0359FFF (RO) kernel
Area 3: C035A000 to C03DDFFF (RO) nutos
Area 4: C03DE000 to C03FEFFF      env
SF: Detected AT45DB321D with page size 528, total 16.5 MiB
In:    serial
Out:   serial
Err:   serial
Net:   macb0: PHY present at 0
macb0: Starting autonegotiation...
macb0: Autonegotiation complete
macb0: link up, 100Mbps full-duplex (lpa: 0x45e1)
macb0
Hit any key to stop autoboot:
```

In the default configuration, U-Boot will wait up to 3 seconds for any key entered via the serial port and then continue to boot the Linux kernel from serial Flash. If any key is pressed, U-Boot will display the command line prompt, waiting for a command. Enter *help* and press *Enter* to display the list of available commands.

```
U-Boot> help
```

More detailed informations about most commands are available by entering *help* followed by the specific command, e.g.

```
U-Boot> help usb
```

Beside the build-in commands, U-Boot is able to execute command scripts, which are stored in environment variables. The command *printenv* lists the contents of all environment variables.

```
U-Boot> printenv
```

Scripts stored in environment variables can be executed by entering the command *run* followed by the variable's name. The variable *bootcmd* contains the script that is executed during automatic booting.

```
U-Boot> printenv bootcmd
bootcmd=run flashbootlinux
```

This simple script calls another script contained in the variable *flashbootlinux*, which will boot the Linux kernel from serial Flash. You may alternatively boot Linux from a TFTP server by changing the bootcmd.

```
U-Boot> setenv bootcmd run tftpbootlinux
U-Boot> saveenv
Saving Environment to SPI Flash...
SF: Detected AT45DB321D with page size 528, total 16.5 MiB
Erasing SPI flash...Writing to SPI flash...done
```

The command *saveenv* will store the modified environment in serial Flash, making it available on the next power-up.

Several predefined scripts are available to boot Linux or Nut/OS or to update Flash contents:

run flashbootlinux	Boot Linux kernel from serial Flash.
run flashbootnut	Boot Nut/OS image from serial Flash.
run nfsbootlinux	Boot Linux kernel from NFS mount.
run nfsbootnut	Boot Nut/OS image from NFS mount.
run nfsinstallenv	Update U-Boot environment in serial Flash from NFS mount.
run nfsinstalllinux	Update Linux kernel in serial Flash from NFS mount.
run nfsinstallnut	Update Nut/OS image in serial Flash from NFS mount.
run nfsinstalluboot	Update U-Boot in serial Flash from NFS mount.
run tftpbootlinux	Boot Linux kernel from TFTP server.
run tftpbootnut	Boot Nut/OS image from TFTP server.
run tftpinstallenv	Update U-Boot environment in serial Flash from TFTP server.
run tftpinstalllinux	Update Linux kernel in serial Flash from TFTP server.
run tftpinstallnut	Update Nut/OS image in serial Flash from TFTP server.
run tftpinstallrootfs	Update root file system in NAND Flash from TFTP server.
run tftpinstalluboot	Update U-Boot in serial Flash from TFTP server.

When using an NFS or TFTP server, make sure to properly set the variables *nfsserver* or *tftpserver*. The U-Boot documentation will provide further details.

Booting Linux

When not interrupted by pressing any key, U-Boot will load the Linux kernel from serial Flash. Alternatively you can enter the command *boot* on the U-Boot command line.

When started, the Linux kernel will display a large number of status messages and finally show the login prompt.

```
.-------.
|       |                  .-.
|   |   |-----.-----.-----.| |   .----..-----.-----.
|       |     | __  |  ---'| '--.|  .-'|     |     |
|   |   |  |  |     |---  ||  --'|  |  |  '  | | | |
'---'---'--'--'--.  |-----''----''--'  '-----'-'-'-'
                -'  |
                '---'

The Angstrom Distribution ethernut5 ttyS0

Angstrom next ethernut5 ttyS0

ethernut5 login: root
root@ethernut5:~#
```

Enter *root* to log in, a password is not required. After successfully logged in, you will get the Linux command line prompt. For a first test you may start the HTTP server by entering *httpd*. Then try to connect the server with a web browser, using the IP address of Ethernut 5 as the URL. You can use the command *ifconfig* to query the current IP address of the Ethernut board.

Board Overview

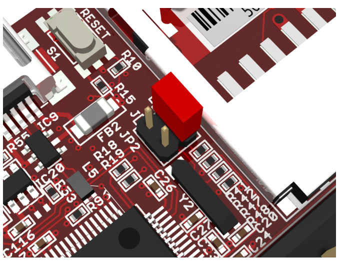
AT91SAM9XE Microcontroller

The AT91SAM9XE512 CPU (IC1) is based on an ARM926EJ-S processor and provides 512 kBytes of high speed on-chip Flash memory and 32 kBytes of SRAM, both of which can be used for code execution and data storage.

A wide rage of peripherals is embedded in the chip, including USB 2.0 full speed host and device interface, Ethernet MAC, image sensor interface, debug unit, 10-bit A/D converter, SD/SDIO/MultiMedia-Card interface, synchronous serial controller (I2S), 4 USARTs, 2 SPI buses, 2 TWI buses (I2C), and several GPIO ports plus timers and counters including a watchdog timer.

For a detailed description please refer to the AT91SAM9XE512 datasheet.

NOR Flash Memory



The AT91SAM9XE512 provides 512 kBytes of on-chip Flash memory. When enabling the security bit GPNVM0, access to the Flash is forbidden. Disabling this security bit can only be achieved by a full erase of the entire Flash. This ensures the confidentiality of the code.

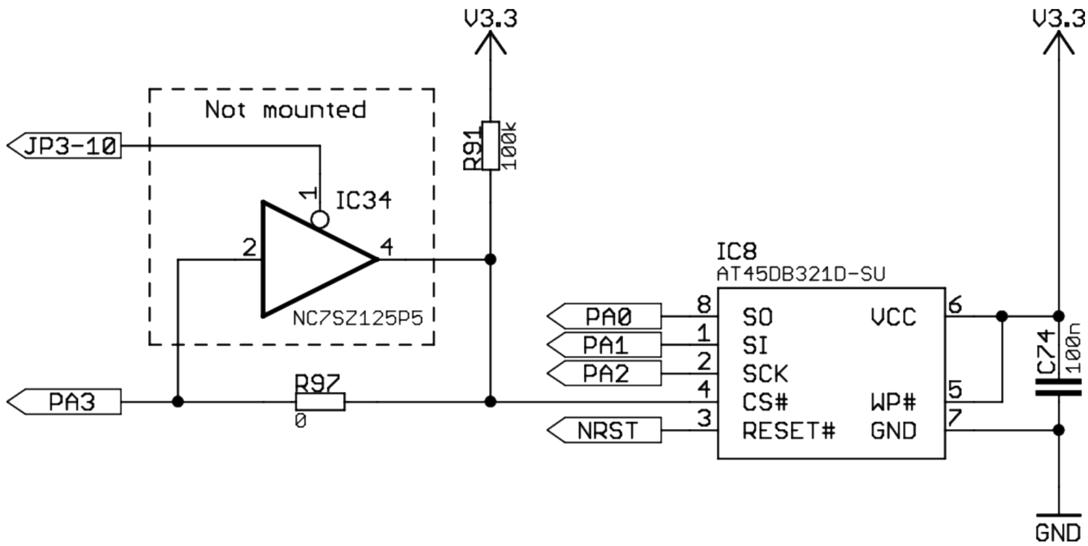
In addition, lock bits can be used to protect write/erase operation on specific Flash memory regions. This allows to protect the firmware from accidental overwrite.

Shortening pins 1 and 2 of JP2 will erase the internal Flash and disable any protection. Re-programming the internal Flash memory and enabling lock or security bits is possible by using

- the RS-232 or USB device port with the SAM-BA utility
- the JTAG interface with the OpenOCD utility.

Serial Flash Memory

The AT45DB321D DataFlash (IC8) offers 4MByte of serial Flash memory. It is accessed by the CPU via the first SPI interface using PA0 (MISO), PA1 (MOSI) and PA2 (SCK). The chip is selected, when the GPIO line PA3 is driven low. Optional mounting of IC34 is explained in chapter Flash Boot Enable.



Typically, serial Flash memory on Ethernut 5 is used to store the configuration and several boot images. Initial programming of the serial Flash can be done with

- the SAM-BA utility using the RS-232 or the USB device port.
- via JTAG by uploading U-Boot into the SDRAM and using the command *cp* or any of the pre-defined install scripts.

For the partition layout, refer to chapter "Memory Layout".

NAND Flash Memory

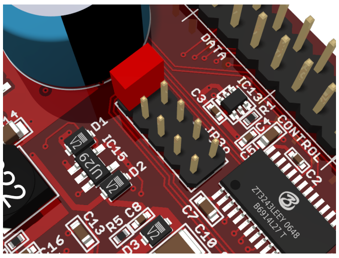
The Micron MT29F8G08ABABAWP (IC3) provides 1 GByte of NAND Flash, which is used for the Linux root file system by default.

Initial programming of this chip can be done by using

- the U-Boot command *nand*.
- the SAM-BA utility using the RS-232 or the USB device port.

NAND Flash is currently not supported by Nut/OS.

Flash Boot Enable



When removing R96 and mounting IC33, or when removing R97 and mounting IC34, pins 9 and 10 of JP3 must be shortened by a jumper to enable NAND Flash or serial Flash access, respectively.

Disabling Flash access is useful with AT91SAM9260 CPUs, where the ROM code may get stuck with bad Flash contents. On AT91SAM9XE equipped boards the ROM code will never boot automatically from external Flash and shortening JP3 pins 9 and 10 has no function.

MMC/SD-Card Socket

The interface supports the following specifications:

- MultiMedia Card Version 3.11
- SD Memory Card Version 1.0
- SDIO Version 1.1

Ethernet Interface

The MAC part of the Ethernet controller is integrated in the CPU and attached to an external PHY part, an SMSC LAN8710AI (IC2). It is physically attached via an RMII bus. The board 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 in the RJ45 Ethernet connector. The yellow LED indicates the link speed and is switched off when connected to 100 Mbit Ethernet. The green LED will go on, when a valid link has been detected and blinks to indicate receive and transmit activity from and to the network.

MAC Address: 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 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 MAC address of the Ethernut board is stored in the U-Boot environment variable *ethaddr* and in the Nut/OS CONFNET structure located in the upper page of the serial Flash.

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.

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 boot loader firmware supplied with the Ethernut board assumes this configuration.

Any of the following serial interfaces of the microcontroller can be tied to the RS-232 connector via JP1 jumper settings: DBGU, USART0 or USART2. In the default configuration the debug interface (DBGU) is attached to the connector.

Note, that USART0 handshake line may conflict with any hardware (image sensor) attached to the FFC connector J4.

DB-9 connector attached to DGBU. **Default.**

DB-9 connector attached to USART0.

DB-9 connector attached to USART0 with hardware handshake.

DB-9 connector attached to USART0 with full modem handshake.

DB-9 connector attached to USART2.

DB-9 connector attached to USART2 with hardware handshake.

The following table shows the connector's pins, that are used in a specific configuration:

Pin	Signal	I/O	DBGU	USART0	USART2
1	DCD	In		✔	
2	RxD	In	✔	✔	✔
3	TxD	Out	✔	✔	✔
4	DTR	Out		✔	
5	GND		✔	✔	✔
6	DSR	In		✔	
7	RTS	Out		✔	✔
8	CTS	In		✔	✔
9	RI	In		✔	

USB Device Port

This interface is available at an on-board Mini-B receptacle. It is mainly used to connect the Ethernut board to a PC and conforms to USB 2.0 with a maximum transfer rate of 12 Mbits. Together with the SAM-BA GUI running on the PC, the ROM code in the CPU allows reading and writing all memories as well as erasing and protecting all Flash memories via USB.

The USB device port may be also used to supply power to the Ethernut board.

USB Host Port

An on-board USB Standard-A receptacle allows to attach external USB devices to the Ethernut board. The interface conforms to USB 2.0 with a maximum transfer rate of 12 Mbits.

USB host functions are supported by U-Boot and Linux, no support is currently available for Nut/OS applications.

FFC Connector

A 24-wire flexible flat cable (FFC) with 0.5mm pitch can be attached to connector J4. While providing 18 general purpose I/O pins, it includes all required hardware signals to attach an external image sensor.

The 10-bit image sensor interface offers programmable frame capture rates and conforms to ITU-R BT.601/656. It provides SAV and EAV synchronization, a preview path with scaler and YCbCr format.

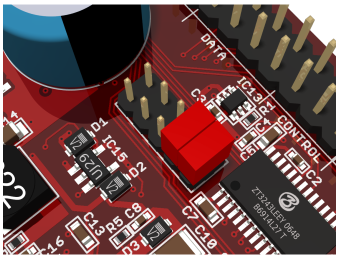
Pin	Image Sensor	GPIO	Alternate Function	Shared with
1	+3.3V Output			
2	Ground			
3		PB6	UART1 TxD	Expansion Port Pin 40
4		PB7	UART1 RxD	Expansion Port Pin 39
5	Data Bit 8 Input	PB10	UART3 TxD	
6	Data Bit 9 Input	PB11	UART3 RxD	
7	Data Bit 0 Input	PB20	SSC Receive Clock	
8	Data Bit 1 Input	PB21	SSC Receive Frams Sync	
9	Data Bit 2 Input	PB22	UART0 DSR	RS-232 Interface
10	Data Bit 3 Input	PB23	UART0 DCD	RS-232 Interface
11	Data Bit 4 Input	PB24	UART0 DTR	RS-232 Interface
12	Data Bit 5 Input	PB25	UART0 RI	RS-232 Interface
13	Data Bit 6 Input	PB26	UART0 RTS	RS-232 Interface
14	Data Bit 7 Input	PB27	UART0 CTS	RS-232 Interface
15	Pixel Clock Input	PB28	UART1 RTS	
16	Vertical Sync Input	PB29	UART1 CTS	
17	Horizontal Sync Input	PB30	Programmable Clock 0	
18	Master Clock Output	PB31	Programmable Clock 1	
19	I2C Data	PA23		On-Board I2C Bus
20	I2C Clock	PA24		On-Board I2C Bus
21	+1.8V Output			
22	+1.8V Output			
23	Ground			
24	+3.3V Output			

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.

Power Supply



The board has 3 voltage regulators (IC12, IC18 and IC24) and can be powered by using

- an unregulated 2 Watt external supply of 7-24 VDC with 2.1mm barrel connector
- IEEE 802.3af Power over Ethernet
- USB with a Mini-B Plug

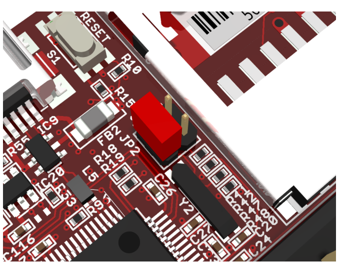
As soon as power is attached to any of the inputs, the red LED at the reset button will lit.

Internally the board uses 3 different voltages, 5V for USB and power management, 3.3V for I/Os and 1.8V for the CPU core. A dedicated power management CPU is available to select the supply source and to enable or disable consumer loads. If this CPU is not running properly or if its firmware had been erased, then the 3.3V I/O voltage and the 1.8V core voltage can be forcibly enabled by removing both control enable jumpers at JP3.

Since additional expansion boards can be connected, the voltage regulators are capable of supplying a maximum current of 500mA. Make sure that your external power supply can provide this load.

Additional information is provided in the chapter "Power Management".

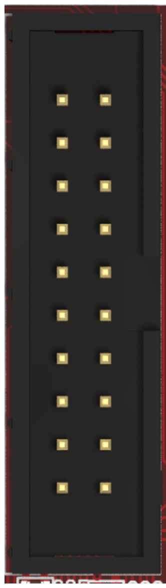
JTAG Interface



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 5 you can perform standard boundary scan when connecting pins 3 and 4 of JP2. In this case the CPU will be disabled.

Furthermore, the JTAG interface can be used to program the Flash memory. Finally it provides a debugging interface to the ARM9 CPU.

On Ethernut 5 Boards a 20-pin, dual-row, 0.1-inch male connector is used for JTAG, which is recognized as an industry standard for ARM CPUs.

Ethernut 20 pin JTAG Connector				
				
3.3V	1		2	3.3V
nTRST	3		4	GND
TDI	5		6	GND
TMS	7		8	GND
TCK	9		10	GND
RTCK	11		12	GND
TDO	13		14	GND
nRST	15		16	GND
N/C	17		18	GND
N/C	19		20	GND

Please refer to the chapter "Using the JTAG Interface" for more details.

RTC

The NXP PCF8563 Realtime Clock/Calendar Chip (IC7) is accessed via a TWI (I2C). 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.

Hardware Clocks

Three crystals and one oscillator provide the required clocks.

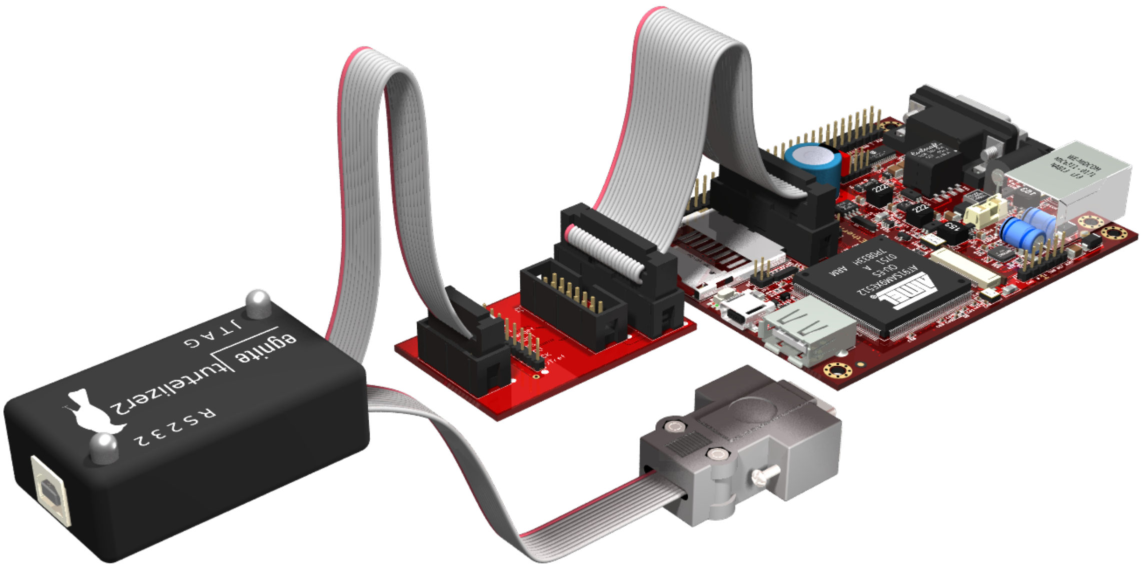
Part	Frequency	Used by
Crystal Y1	18.432 MHz	CPU main clock
Crystal Y2	32.678 kHz	CPU slow clock
Crystal Y3	32.678 kHz	RTC
Oscillator IC11	50 MHz	Ethernet PHY

The CPU is started using a 32 kHz slow clock. In normal mode, the CPU uses the internal PLL A to generate the 180 MHz master clock and a 90 MHz clock for the peripherals. Except the USB clock, which is generated by PLL B.

Using the JTAG Interface

Prerequisites

Any standard 20-pin JTAG adapter can be used with Ethernut 5.0, provided that the accompanied software supports the AT91SAM9XE512 CPU. When using the Turtelizer 2, a 10 to 20 pin adapter is required. In any case, you should check the connector pinout that is given in the chapter "Board Overview".



In addition to the hardware you will need a related software tool, which allows to program the internal Flash memory of the CPU. We recommend the Open Source project OpenOCD, which offers such a tool. It works with the Turtelizer 2 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 5.0 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 power supply from the board. Never attach or detach a JTAG adapter on a powered board.
5. Connect the 20-pin JTAG cable to J7. In case of different hardware, use a cable adapter.
6. Re-apply power to the board.

7. Connect the other interface of the JTAG adapter to the PC. For the Turtelizer, this will be the USB interface.

Install the software that comes with your JTAG adapter.

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

Memory Layout

The following table shows the CPU memory map, which is configured by the SAMBoot boot loader.

Address Range	Usage
0x00000000 - 0x0007FFFF	Boot memory (mapped ROM, Flash or SRAM)
0x00100000 - 0x00107FFF	32 kByte internal ROM (SAM-BA firmware)
0x00200000 - 0x0027FFFF	512 kByte internal Flash (first 4 kBytes used by SAMBoot)
0x00300000 - 0x00307FFF	32 kByte internal SRAM
0x10000000 - 0x1FFFFFFF	Available, chip select 0 at expansion port when R917 mounted
0x20000000 - 0x27FFFFFF	128 MByte external SDRAM
0x40000000 - 0x4FFFFFFF	Interface to 1 GByte external NAND Flash
0x70000000 - 0x7FFFFFFF	Available, chip select 6 shared with FIQ
0x90000000 - 0xEFFFFFFF	Forbidden zone
0xF0000000 - 0xFFFFFFFF	Internal peripherals

Serial Flash Partitions

The following table shows the pre-configured partitions:

Address Range	Usage	
0x000000 - 0x020FFF	132k	Not used, available for user applications
0x021000 - 0x041FFF	132k	U-Boot Environment Image
0x042000 - 0x0C5FFF	528k	U-Boot Image
0x0C6000 - 0x359FFF	2640k	Linux Kernel Image
0x35A000 - 0x3DDFFF	528k	Nut/OS Application Image
0x3DE000 - 0x3FEFFF	132k	U-Boot Environment
0x3FF000 - 0x41FFFF 0x3FF000 - 0x3FFFFF(1	132k 4k(1)	Nut/OS Configuration

(1) When used in 512-byte page mode, the total capacity is reduced by 128 kByte. By default, the 528-byte mode is active.

NAND Flash Partitions

Planes	Blocks	Pages	Bytes	Per
2	2048	256K	1080M	chip
1	1024	128K	540M	plane
-	1	128	512K + 28K	block
-	-	1	4K + 224	page

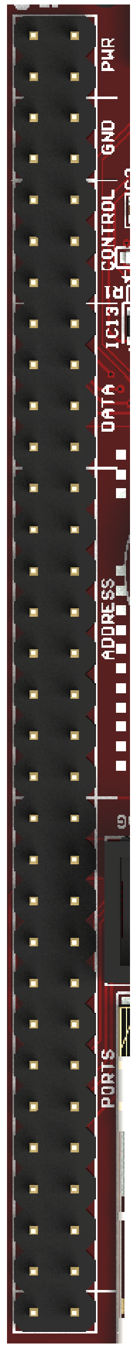
Hardware Expansion

WARNING: Expansion port pins are not 5V tolerant and must not be connected to 5V logic without proper level translation. For detailed specifications refer to the AT91SAM9XE512 datasheet.

Many applications will do just fine with nothing else than the bare Ethernut board. If needed, external hardware may be connected to the RS-232 port or the USB host port. However, if more is required, the Ethernut expansion port is the first choice to add custom designed hardware. This connector contains partial CPU data and address bus, memory read/write signals, digital I/O ports, I2C and SPI bus, reset signal and power supply.

Some signals are also used internally by Ethernut or connected to the FFC connector. Carefully check the board schematic.

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



Pin	Signal	Usage	Pin	Signal	Usage
1	V3.3	+3.3V Supply	2	V3.3	+3.3V Supply
3	V5.0	+5.0V Supply	4	V5.0	+5.0V Supply
5	GND	Signal ground	6	GND	Signal ground
7	GND	Signal ground	8	GND	Signal ground
9	nRST	Board Reset	10	DC	Unregulated supply
11	V5.0	+5.0V supply	12	V5.0	+5.0V supply
13	nRD	Memory Bus	14	nWE	Memory Bus
15	D0	Memory Bus	16	D1	Memory Bus
17	D2	Memory Bus	18	D3	Memory Bus
19	D4	Memory Bus	20	D5	Memory Bus
21	D6	Memory Bus	22	D7	Memory Bus
23	A0	Memory Bus	24	A1	Memory Bus
25	A2	Memory Bus	26	A3	Memory Bus
27	A4	Memory Bus	28	A5	Memory Bus
29	A6	Memory Bus	30	A7	Memory Bus
31	A8	Memory Bus	32	A9	Memory Bus
33	A10	Memory Bus	34	A11	Memory Bus
35	A12	Memory Bus	36	A13	Memory Bus
37	A14	Memory Bus	38	A15	Memory Bus
39	PB7	GPIO, USART, Timer	40	PB6	GPIO, USART, Timer
41	PA29	GPIO, USART	42	PC6	GPIO, Timer
43	PC7	GPIO, Timer	44	PC4	GPIO, Memory Bus, SPI
45	PC15	GPIO, Memory Bus, IRQ	46	PC13	GPIO, IRQ, Memory Bus
47	PB3	GPIO, SPI, Timer	48	PB2	GPIO, SPI, Timer
49	PB1	GPIO, SPI, Timer	50	PB0	GPIO, SPI, TImer
51	PB16	GPIO, Timer	52	PB19	GPIO, Timer
53	PB17	GPIO, Timer	54	PB18	GPIO, Timer
55	PA24	I2C	56	PA23	I2C
57	PA26	GPIO, Timer	58	PA25	GPIO, Timer
59	PA28	GPIO, Timer	60	PA27	GPIO, Timer
61	PC5	GPIO, Memory Bus, SPI	62	PA22	GPIO, ADC
63	NC	Available	64	NC	Available

Power Management

The Ethernut 5.0 CPU communicates with the power management controller via I^2C address 0x22.

The U-Boot boot loader provides the Ethernut board specific command *pwrman*.

PMM Enable Register

Register Name: PMM_REG_ENA

Access: Read-write

7	6	5	4	3	2	1	0
LED	ETHRST	ETHCLK	RS232	MMCPWR	VBUSO	VBUSI	CPUPWR
0	0	1	1	0	0	1	1

CPUPWR
0 = No effect.
1 = 1.8V and 3.3V power supply on.

VBUSI
0 = No effect.
1 = Enables 5V power supply drawn from USB device connector.

VBUSO
0 = No effect.
1 = Enables USB host connector power supply.

MMCPWR
0 = No effect
1 = Enables MMC/SD-Card power supply.

RS232
0 = No effect
1 = Enables RS-232 interface.

ETHCLK
0 = No effect
1 = Enables Ethernet clock oscillator.

ETHRST
0 = No effect
1 = Activates Ethernet PHY reset.

LED
0 = No effect
1 = Switches on green LED.

Disable Register

Register Name: PMM_REG_DIS

Access: Read-write

7	6	5	4	3	2	1	0
LED	ETHRST	ETHCLK	RS232	MMCPWR	VBUSO	VBUSI	CPUPWR
1	1	0	0	1	1	0	0

CPUPWR

0 = No effect.
1 = 1.8V and 3.3V power supply on.

VBUSI

0 = No effect.
1 = Disables 5V power supply drawn from USB device connector.

VBUSO

0 = No effect.
1 = Disables USB host connector power supply.

MMCPWR

0 = No effect
1 = Disables MMC/SD-Card power supply.

RS232

0 = No effect
1 = Disables RS-232 interface.

ETHCLK

0 = No effect
1 = Disables Ethernet clock oscillator.

ETHRST

0 = No effect
1 = Deactivates Ethernet PHY reset.

LED

0 = No effect
1 = Switches off green LED.

Status Register

Register Name: PMM_REG_STA

Access: Read-only

7	6	5	4	3	2	1	0
ALARM	-	-	RS232	MMCPWR	VBUSO	VBUSI	CPUPWR
0	0	0	1	1	1	1	1

CPUWR

0 = 1.8V and 3.3V power supply inactive.
1 = 1.8V and 3.3V power supply active.

VBUSI

0 = USB device connector supply inactive.
1 = USB device connector supply active.

VBUSO

0 = USB host connector supply inactive.
1 = USB host connector supply active.

MMCPWR

0 = Memory card supply inactive.
1 = Memory card supply active.

RS232

0 = RS-232 disconnected.
1 = RS-232 connected.

ALARM

0 = No RTC alarm pending.
1 = RTC alarm pending.

Temperature Register

Register Name: PMM_REG_TEMP

Access: Read-only

7	6	5	4	3	2	1	0
ADC_TEMP							
0	0	0	0	0	0	0	0

ADC_TEMP

ADC value of the temperature sensor.

The temperature is measured with an LM50C sensor, connected to an ADC input. The sensor is located at the bottom side of the board near to the ATmega168. Its output voltage is linear to the measured temperature in degrees Celsius and covers the range from -40°C to +125°C. The output voltage can be calculated with the following formula:

$U_{OUT} = 0.5V + T * 0.01V$

To calculate the temperature (°C) from the ADC value, use

```
T = ADC_TEMP * 500 / 256 - 50
```

Auxiliary Voltage Register

Register Name: PMM_REG_VAUX

Access: Read-only

7	6	5	4	3	2	1	0
ADC_VAUX							
0	0	0	0	0	0	0	0

ADC_VAUX

ADC value of the auxiliary voltage sensor.

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.

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

Changes Compared to Ethernut 1 and Ethernut 2

The most important change to notice is the different CPU used on Ethernut 5. 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 will allow to create one source code running on all supported platforms.

You can't use programming adapters shipped with previous AVR starter kits, because Ethernut 5 needs a specific JTAG adapter for an ARM CPU. Note, that Ethernut 5 allows re-programming via RS-232 and USB. The JTAG adapter is required for debugging only.

Also note, that the ARM9 used on Ethernut 5 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 AT91SAM9XE outputs use 3.3V switching levels, and the inputs are not 5V-tolerant. They cannot withstand 5V switching levels.

Changes Compared to Ethernut 3

Although Ethernut 3 is also based on an ARM CPU, it is still required to re-compile existing code for Ethernut 5.

While Ethernut 3 allows 5V logic being attached to some of the expansion port pins, Ethernut 5 does not provide any 5V tolerant pins.

The expansion port memory bus on Ethernut 3 is implemented in the CPLD for backward compatibility with Ethernut 1 and 2. However, on Ethernut 5 it is directly connected to the ARM9 CPU. Related timing constraints are explained in the AT91SAM9XE datasheet.

The Turtelizer 2 programming adapter, that is shipped with the Ethernut 3 starter kit, can be used with Ethernut 5, but requires a 10 to 20 pin JTAG connector adapter.

Troubleshooting

At some time in the life of your Ethernut may suddenly cease to function. This may 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 check again 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 board initialization fail.

If you can't get any output from U-Boot, then the problem is most probably with 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 had been configured correctly and if it still fails, try a different emulator program, replace the cable and finally try another PC.

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 GmbH 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 200 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 problem.

If the current is within limits, you can check the board's internal voltages. Expansion port pins 5 to 8 provide the ground level, while 5.0V and 3.3V are available at pins 3 and 1, respectively. The 1.8V supply can be checked at the test point near C14.

If the power supply is OK, check that the NRST signal (see schematic page 4) is at high level.

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:

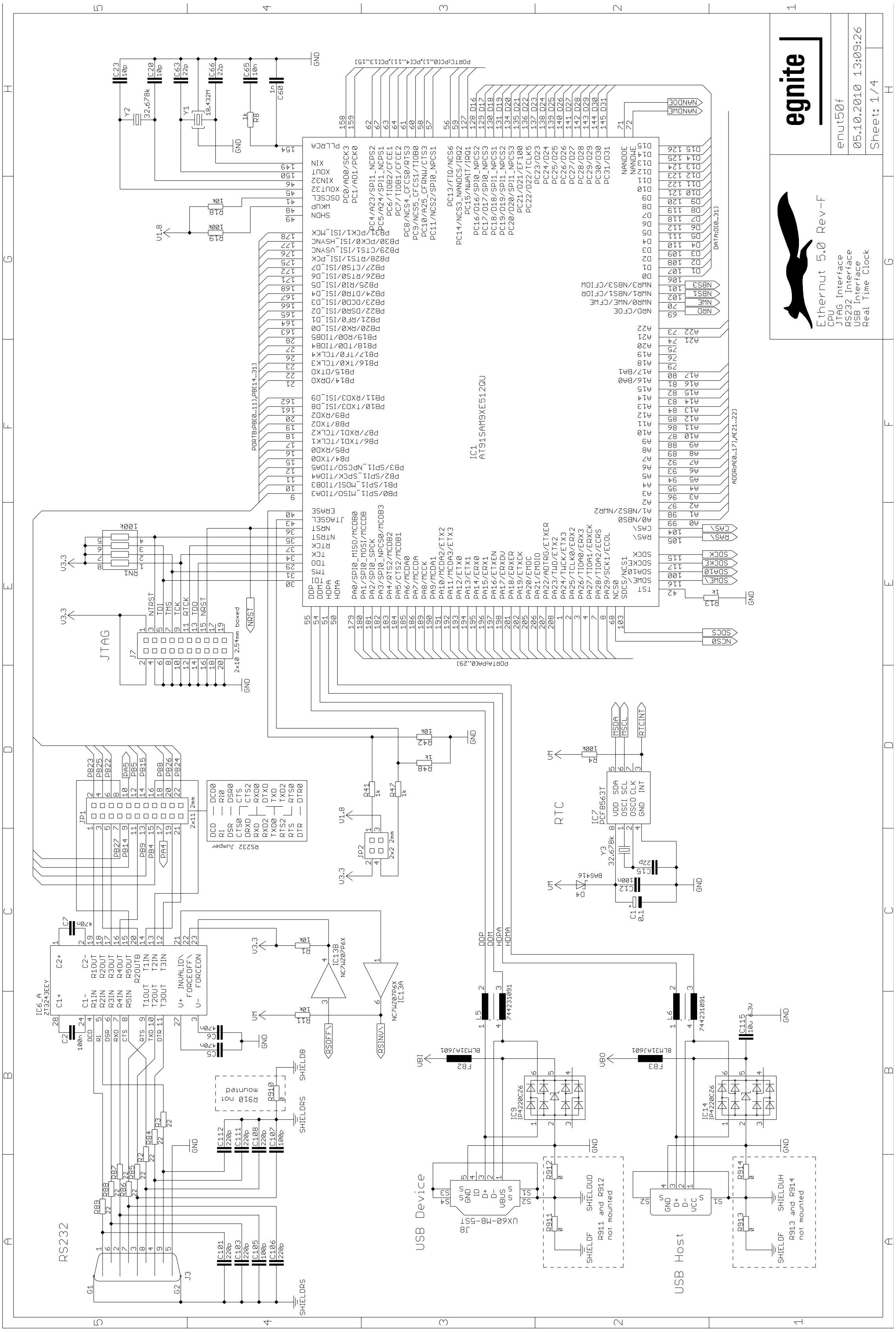
- Ethernut Revision, printed on the back side of the board.
- MAC address of your Ethernut, written on top of the board and on the invoice.
- Description of your problem. You may keep it simple, we may request details later.

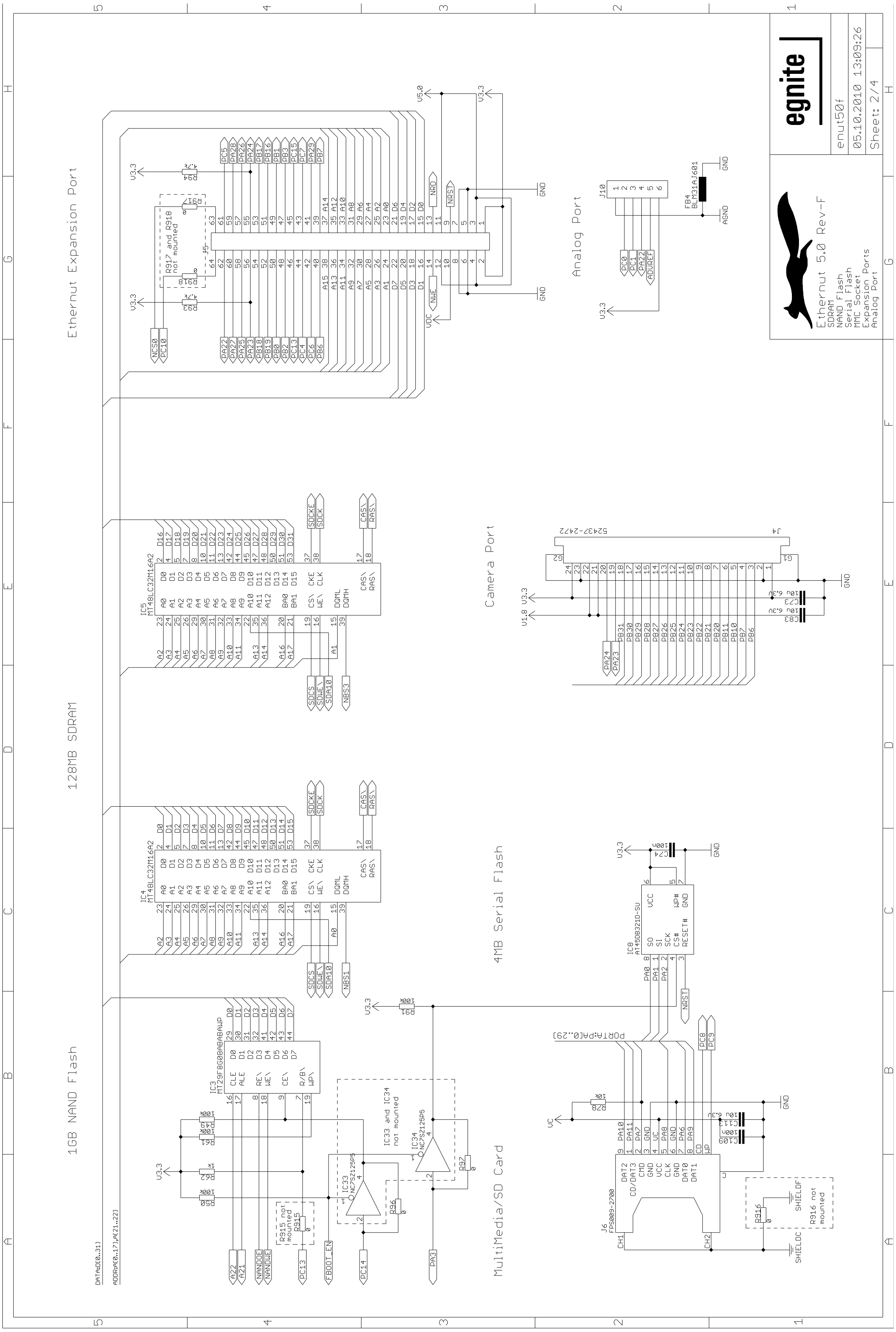
Please understand, that we are not able to provide any warranty, if you destroyed the board because of ignoring our ESD precautions advises or attaching badly designed hardware. In such cases we may ask at least for a refund of our shipping costs.

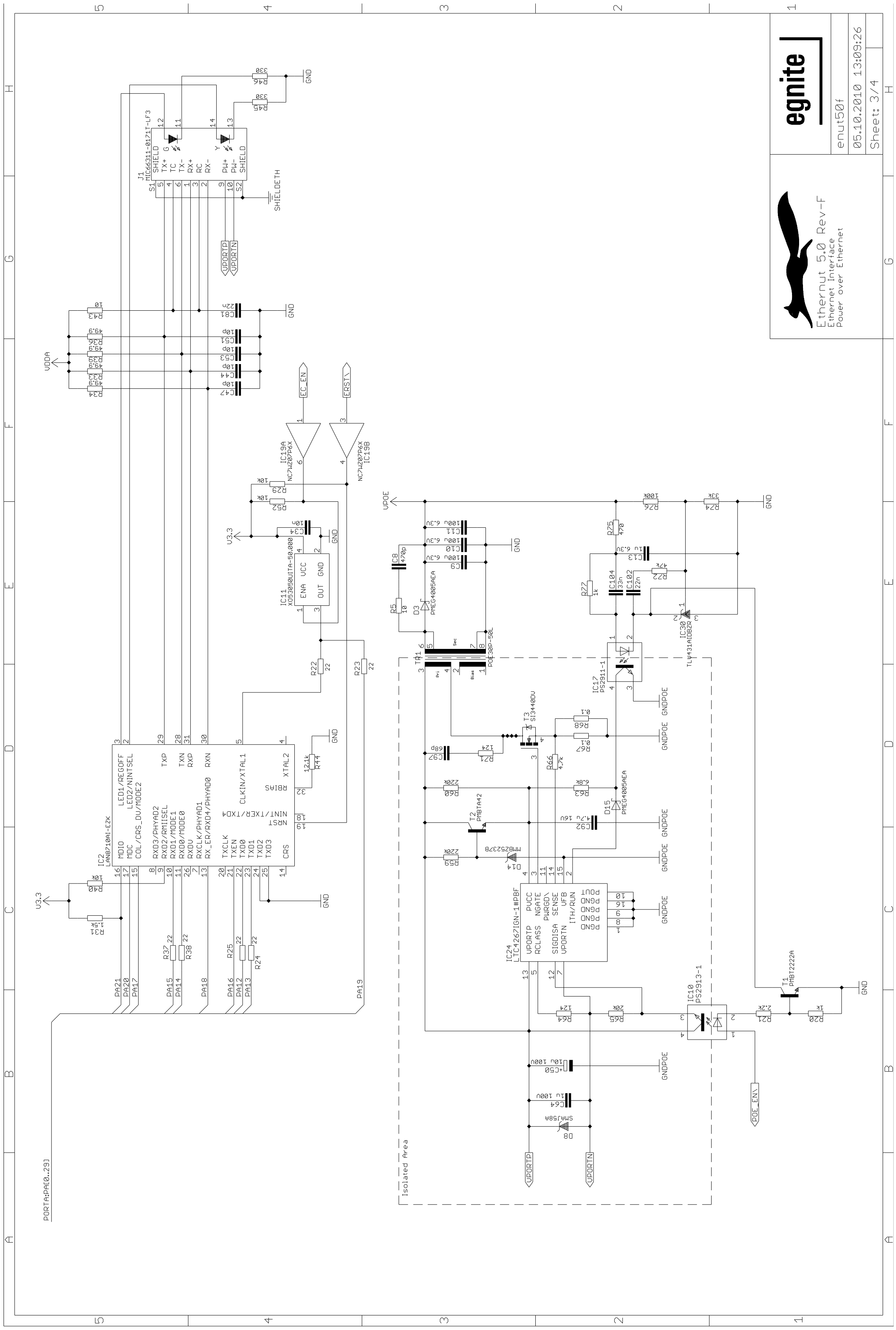
Anyway, whatever happened, we will do anything possible to revitalize your Ethernut. Or, if it finally passed away, let it rest in peace and send a replacement back to you at the least possible costs.

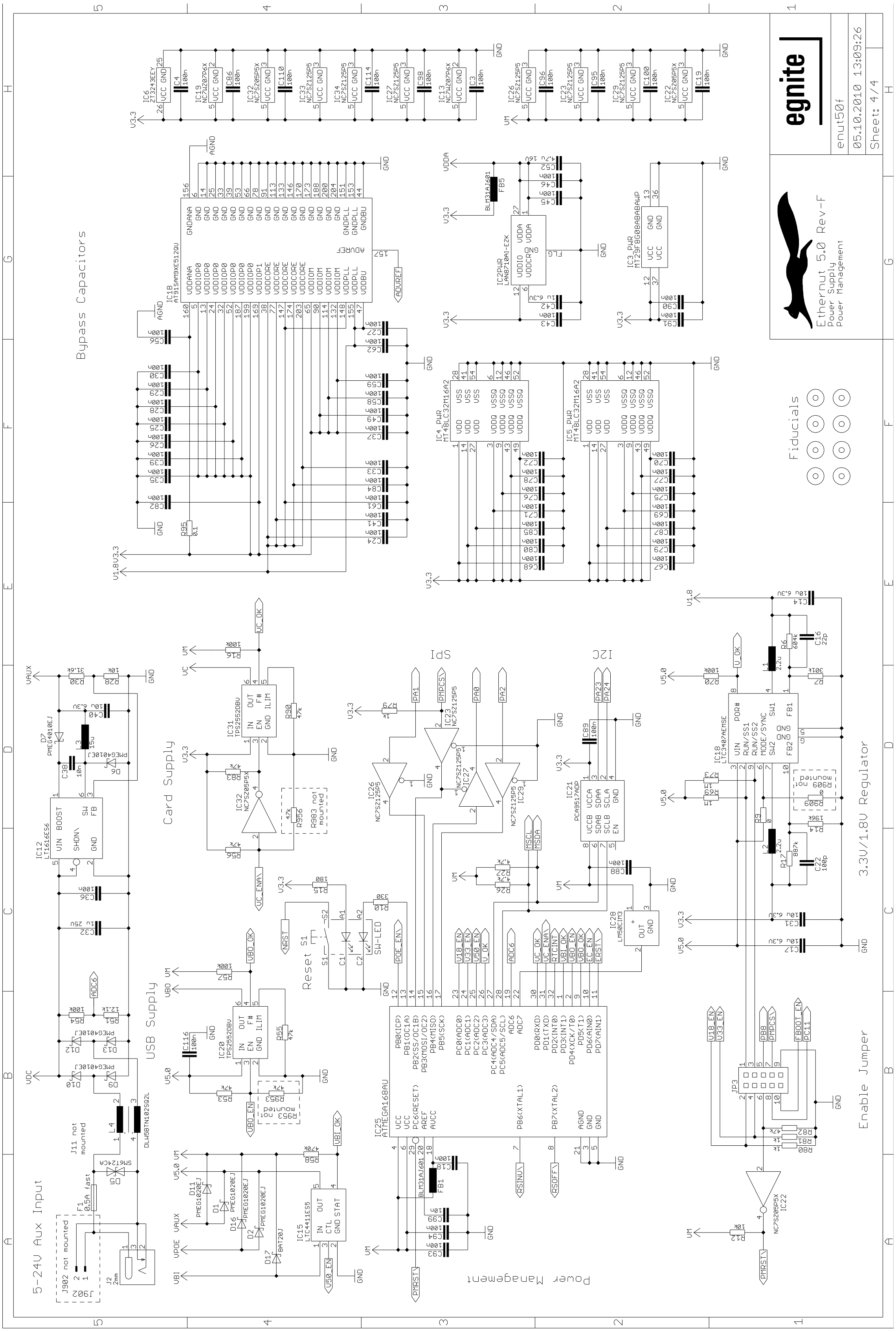
Schematics

Full schematics are provided on the next 4 pages.


egnite
Ethernut 5.0 Rev-F
CPU
JTAG Interface
RS232 Interface
USB Interface
Real Time Clock
enut50f
05.10.2010 13:09:26
Sheet: 1/4


Ethernut Expansion Port
128MB SDRAM
1GB NAND Flash
Analog Port
Camera Port
4MB Serial Flash
MultiMedia/SD Card
egnite
enut50f
05.10.2010 13:09:26
Sheet: 2/4
Ethernut 5.0 Rev-F
SDRAM
NAND Flash
Serial Flash
MMC Socket
Expansion Ports
Analog Port


egnite
enut50f
05.10.2010 13:09:26
Sheet: 3/4
Ethernut 5.0 Rev-F
Ethernet Interface
Power over Ethernet
Isolated Area


Bypass Capacitors
Card Supply
USB Supply
5-24V Aux Input
Power Management
SPI
I2C
3.3V/1.8V Regulator
Enable Jumper
Fiducials
egnite
Ethernut 5.0 Rev-F
Power Supply
Power Management
enut50f
05.10.2010 13:09:26
Sheet: 4/4

egnite GmbH
Erinstrasse 9
44575 Castrop-Rauxel
Germany

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

E-Mail: info@egnite.de

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