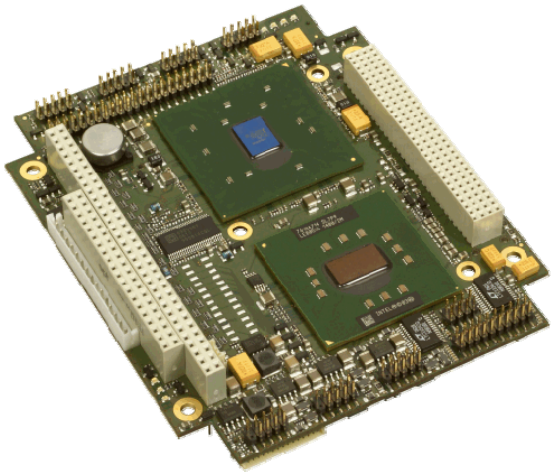
LiPPERT
THE EMBEDDED PC COMPANY

Cool RoadRunner-PM

PC/104-Plus CPU Board

Technical Manual



TME-104P-CRRPM-R3V2.doc
Revision 3.2 / August 09
© LiPPERT Embedded Computers GmbH
Hans-Thoma-Str. 11
D-68163 Mannheim
http://www.lippertembedded.com/

Technical Manual Cool RoadRunner-PM

LiPPERT Document: TME-104P-CRRPM-R3V2.doc Revision 3.2

Copyright ©2007-2008 LiPPERT Embedded Computers GmbH, All rights reserved

Contents and specifications within this manual are subject of change without notice.

Trademarks

MS-DOS, Windows, Windows 95, Windows 98, Windows NT and Windows XP are trademarks of Microsoft Corporation. PS/2 is a trademark of International Business Machines, Inc. Intel and Pentium are trademarks of Intel Corporation. Geode is a trademark of Advanced Micro Devices, Inc. M-Systems and DiskOnChip are trademarks or registered trademarks of M-Systems Flash Disk Pioneers, Ltd. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Table of Contents

1. Overview

1.1 Introduction

The Cool RoadRunner-PM is a complete, high-performance single board embedded computer build around the Intel® Pentium®M processor. The small and sturdy PC/104-Plus board is available from stock with several processor variants and clock rates: Pentium M 738 (1.4 GHz) and Pentium M 745 (1.8 GHz), and a Celeron® M 373 (1.0 GHz).

The Cool RoadRunner-PM's Intel Pentium M processor is a high performance, low power mobile processor with several micro-architectural enhancements over older Intel mobile processors. Its i82855GME chipset features a fast Intel Extreme Graphics 2 graphics engine with 2D and 3D capabilities, delivering outstanding graphics performance for embedded computer applications. The graphics and display controllers use up to 64 MB of memory, shared with the system RAM. The improved unified memory architecture (UMA) guarantees the maximum possible display performance. Display resolutions as high as 2048 x 1536 at 75 Hz can be handled. In addition to VGA-monitors, all kinds of TFT flat panel displays can be connected using the dual channel LVDS interface. Backlighting power is available.

System memory can be expanded up to 1 GByte using SODIMM-DDR-RAM modules. There is a FastEthernet (100/10BaseT) network adapter integrated on board. Six high-speed USB 2.0 ports provide sufficient connections for transportable hard disks, cameras, keyboards and other peripherals. Two serial ports can be configured to operate either with RS232 or RS85 level.

The Cool RoadRunner-PM further features an ATA-100 compliant EIDE interface with a 44 pin, 2 mm connector.

There is an AC97 codec on-board for sound I/O, providing Line-in, Line-out, Mic-in, and CD-in signals. For even better sound quality, an S/PDIF compliant digital audio output is available, too.

The Cool RoadRunner-PM handles both, PCI and ISA buses and accepts standard PC/104 or PC/104-Plus formatted peripheral cards for system extension. Troubleshooting is made easy with supervision LED's for power, watchdog, Ethernet and life signalization. These LED's are conveniently located at the edge of the module and thus are visible even with peripheral PC/104 boards mounted on top of it.

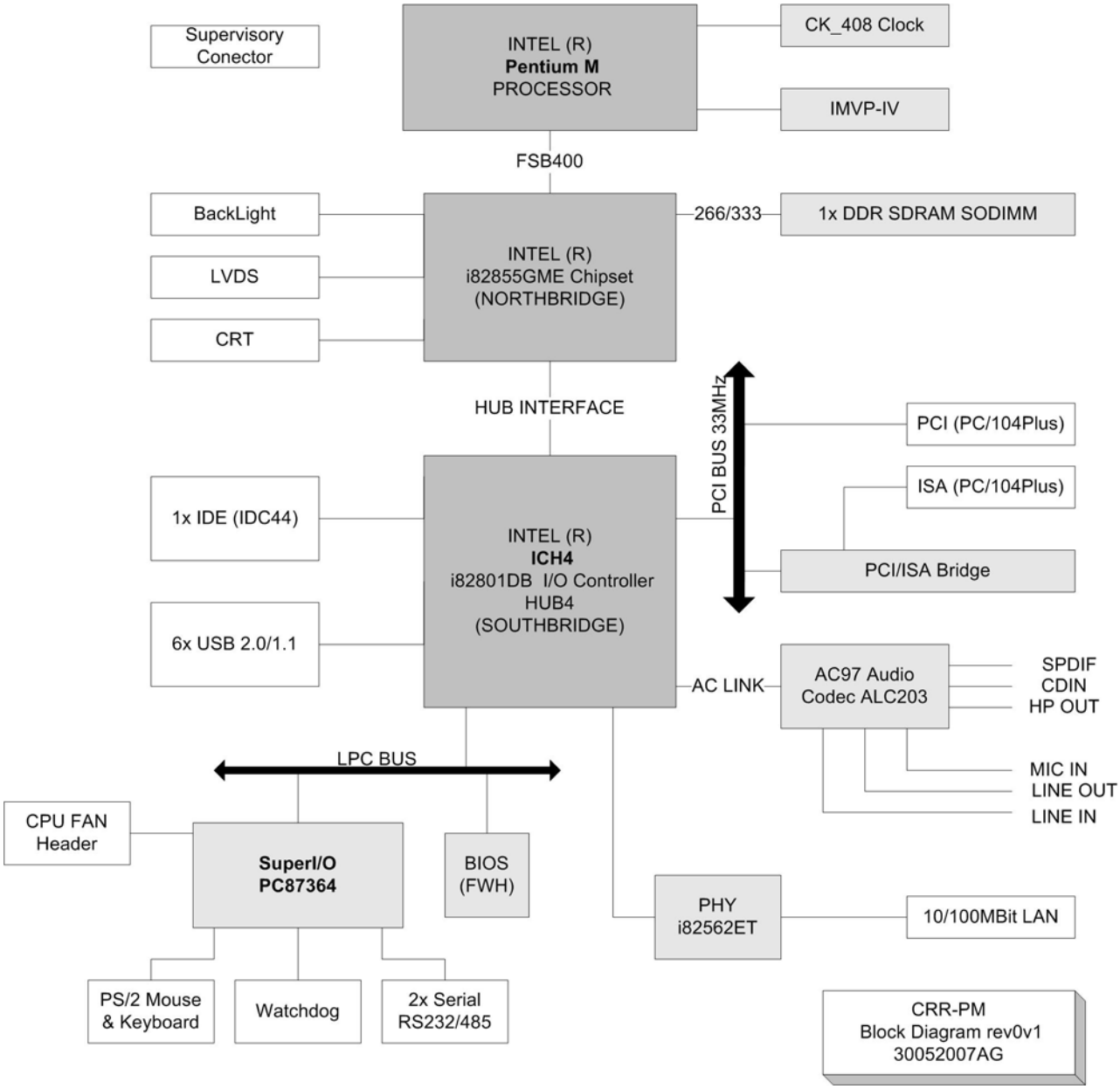
The power consumption drawn from the single 5V supply ranges from only 14 watts up to 32.5 watts, depending on the chosen processor option. The Cool RoadRunner-PM works in ambient temperatures from -20° to + 60°C; there is an extended temperature model available that withstands even -40° to +84°C without requiring an active cooling system. Complete cooling solution for the different models can ordered.

The module comes with a Phoenix FirstBIOS BIOS software in FlashPROM and supports a range of operation systems. Windows 2000, Windows XP, Windows XP Embedded, Windows CE, and Linux can be used with the Cool RoadRunner-PM, as well as some real time operating systems.

Features

- PC/104 Plus connectors for peripherals
- Ethernet 10/100BaseT
- 6 x USB 2.0
- 2 x RS232/RS485 software selectable serial ports
- EIDE (ATA100) on IDC44 header
- Analog VGA
- LVDS for displays, either single channel 18 bits or dual channel 2x18 bits, with backlight power
- AC97 Audio with line-in, line-out, mic-in and SPDIF
- PS/2 keyboard/mouse support

Block Diagram



1.2 Ordering Information

Cool RoadRunner-PM Models

Order number	Description
803-0013-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Celeron M 373 (1.0 GHz, 512kB L2 cache, 400 MHz FSB) Operating temp. range: -20°C … +60°C
803-0014-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Pentium M 738 (1.4 GHz, 2MB L2 cache, 400 MHz FSB). Operating temp. range: -20°C … +60°C
803-0015-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Pentium M 745 (1.8 GHz, 2MB L2 cache, 400 MHz FSB). Operating temp. range: -20°C … +60°C
803-0017-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Celeron M (600 MHz, 512kB L2 cache, 400 MHz FSB) Operating temp. range: -20°C … +60°C
903-0013-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Celeron M 373 (1.0 GHz, 512kB L2 cache, 400 MHz FSB) Operating temp. range: -40°C … +85°C
903-0017-10	Cool RoadRunner-PM PC/104 Plus CPU board with Intel Celeron M (600 MHz, 512kB L2 cache, 400 MHz FSB) Operating temp. range: -40°C … +85°C

Cable Sets and Accessories

Order number	Description
863-0012-10	Cable set for CRR-PM (Cool Roadrunner-PM): Power, PS/2 keyboard and mouse, audio, Ethernet, VGA-CRT, 6x USB, 2x COM, IDE (44 pin, 2mm), cable adapter 2.5" > 3.5", and adapter 3.5" > 2.5".
865-0017-10	Heat spreader
865-0018-10	Passive cooler
865-0019-10	Active cooler

1.3 Specifications

Electrical Specifications

Supply voltage	+ 5V DC
Rise time	< 5 ms
Supply voltage ripple	± 5 %
Inrush current	9.5 A

Supply current

803-0013-10 903-0013-10 1.0 GHz Celeron M	max. 2.9 A, depending on usage and operating system typ. 1.8 A (Windows XP idle mode) typ. 2.2 A (MS-DOS)
803-0014-10 1.4 GHz Pentium M	max. 4.0 A, depending on usage and operating system typ. 2.3 A (Windows XP idle mode) typ. 2.9 A (MS-DOS)
803-0015-10 1.8 GHz Pentium M	max. 4.9 A, depending on usage and operating system typ. 2.6 A (Windows XP idle mode) typ. 3.9 A (MS-DOS)
803-0017-10 903-0017-10 600 MHz Celeron M	max. 1.8 A, depending on usage and operating system typ. 1.2 A (Windows XP idle mode) typ. 1.4 A (MS-DOS)

Environmental Specifications

Temperature range	-20 … 60 °C (standard, for models with serial numbers 8xx-xxxx-xx)
	-40 … 85 °C (extended, for models with serial numbers 9xx-xxxx-xx)
Storage temperature	-40 … 85 °C
Temperature change	max. 10K / 30 minutes
Humidity (relative)	10 … 90 % (non-condensing)
Pressure	450 … 1100 hPa

MTBF

MTBF at 25°C	174.636 hours

Mechanical

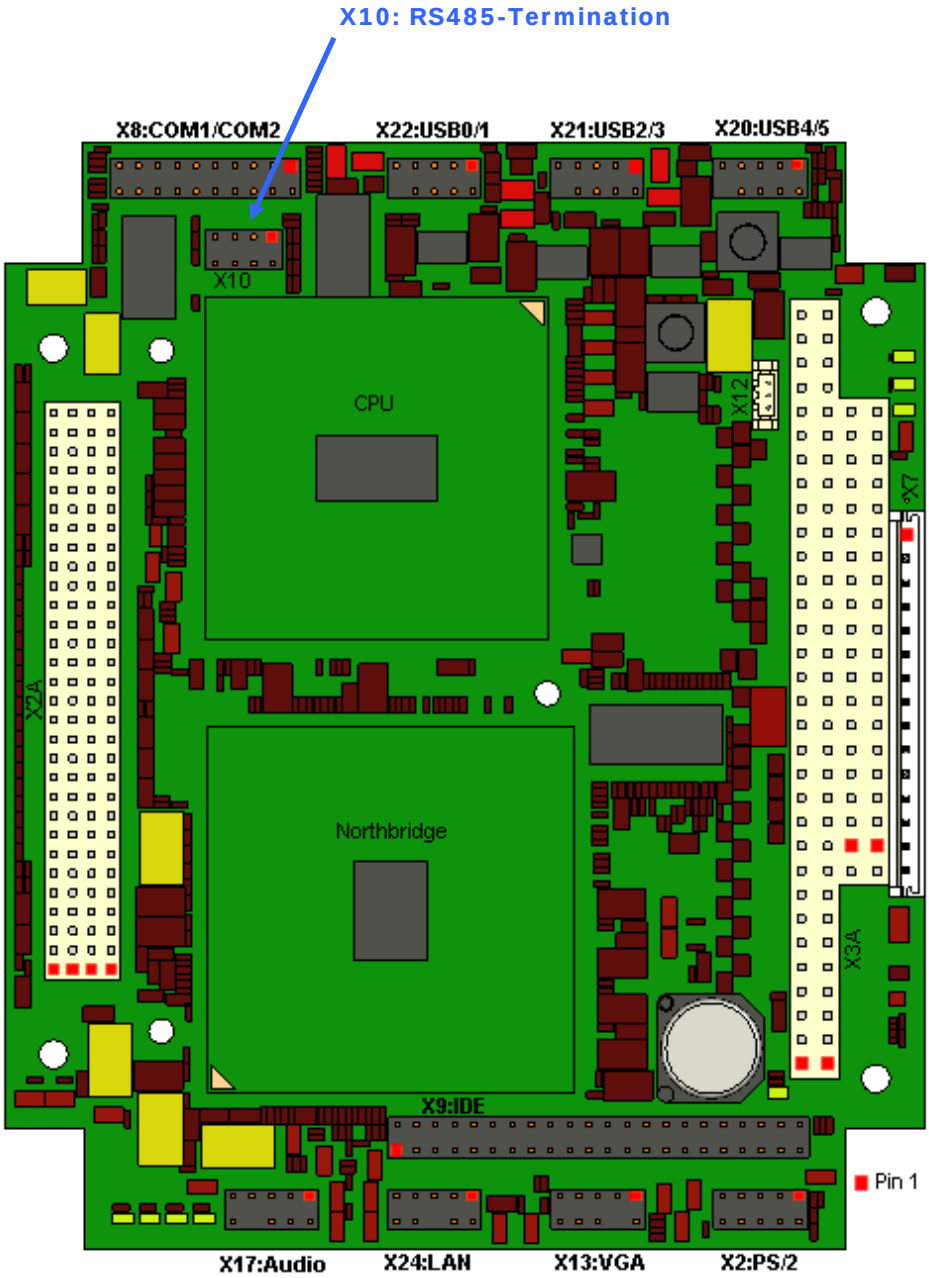
Dimensions (L x W)	95.89 mm x 102.87 mm (3.775 x 4.050-inch) 3.775" by 4.050"
Height	23 mm without cooler 50 mm with active cooler 865-0019-10
Weight	300 g
Mounting	4 mounting holes for PC/104 Plus

Note *It is strongly recommend using plastic spacers instead of metal spacers to mount the board. With metal spacers, there is a possible danger to create a short circuit with the components located around the mounting holes. This can damage the board!*

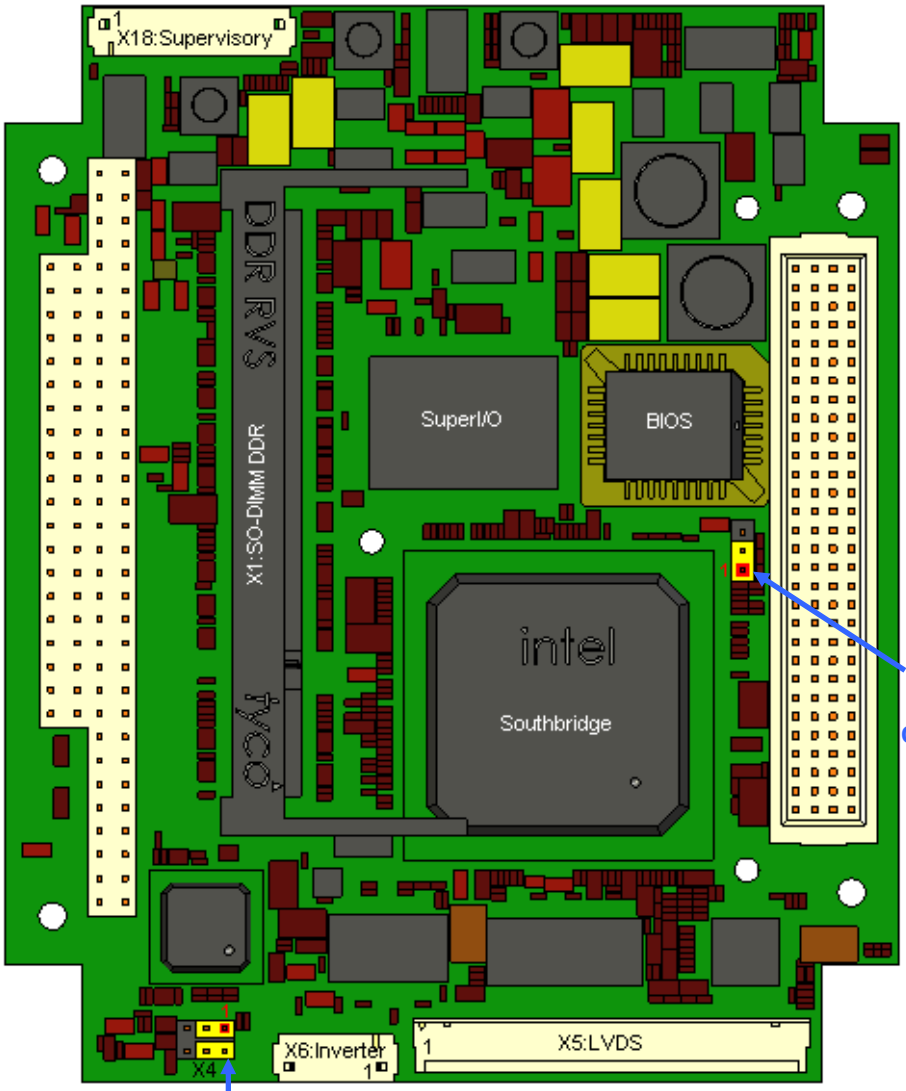
2. Getting Started

2.1 Connector and Jumper Locations

Top



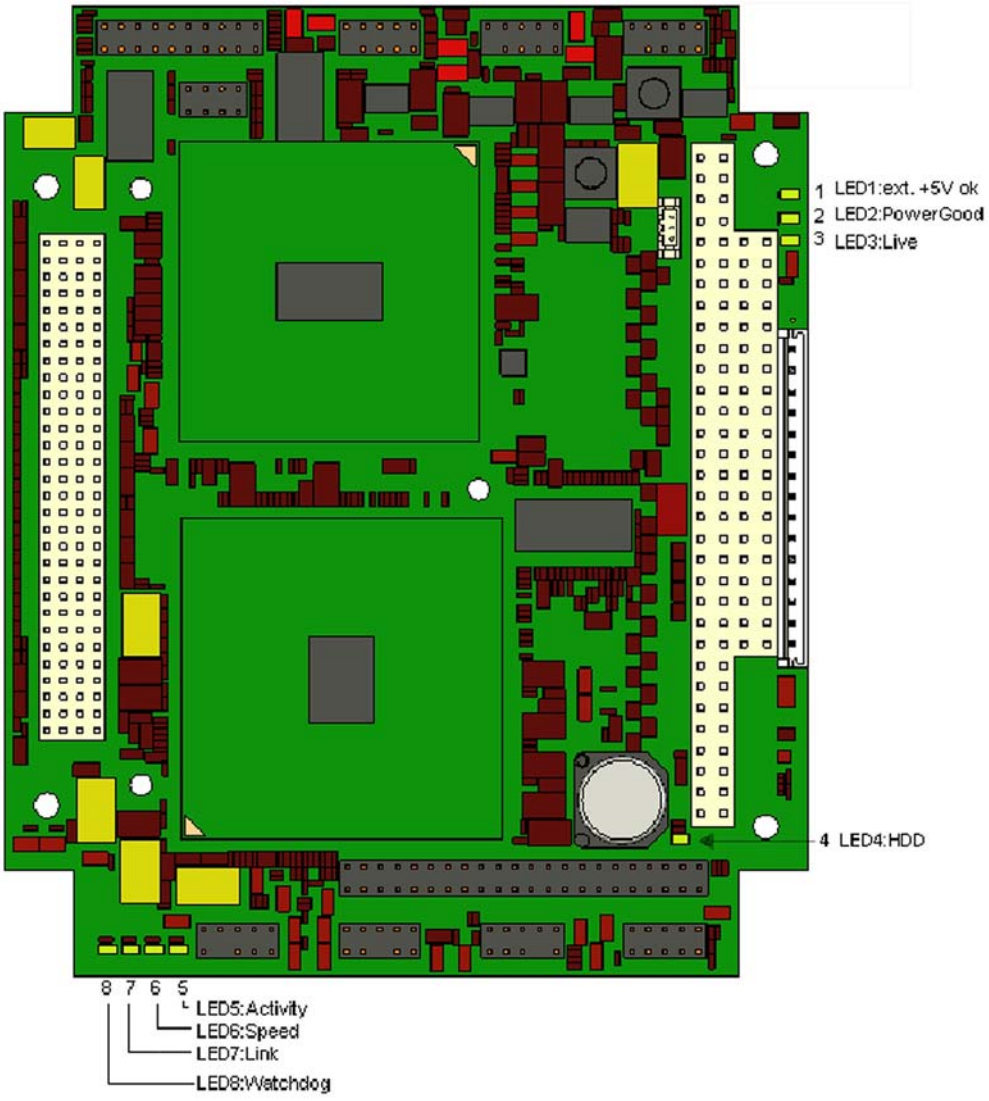
Bottom


X18:Supervisory
DDR SYS
X1:SO-DIMM DDR
SuperI/O
BIOS
intel
Southbridge
JP1:
CLR CMOS
X4
X6:Inverter
X5:LVDS
X4: Display Voltage Connector

2.2 LED Indicators

To facilitate problem solving, the Cool RoadRunner-PM provides LED indicators for the following conditions:

LED	Name	Function
1	PW	Power: external +5V power ok
2	PGD	Power Good: onboard Power Supplies are working
3	LVE	Live: Cool RoadRunner-PM was started
4	HDD	Harddisk: PATA accesses
5	ACT	LAN Activity: LAN works
6	SPD	LAN Speed: on when 100Mbit
7	LNK	LAN Link: on when connection to other device
8	WDG	Watchdog: on when watchdog has been activated



2.3 Hardware Setup

Installing the Cool RoadRunner-PM is very straightforward. First, unpack the board observing the usual electrostatic discharge (ESD) precautions.

Caution

Before you touch the board, make sure that you have discharged yourself and your gear towards a grounded terminal. Damages due to ESD are usually not immediately visible and will only show up later as failures in the field.

Mount the cooling device.

Caution

Never operate the Cool RoadRunner-PM without suitable cooling devices. Failing this can destroy the module.

Connect a display monitor to the VGA connector and keyboard and mouse to PS/2 or USB connectors. Add a suitable hard drive and/or a CD drive to the configuration.

Caution

Never connect or disconnect peripherals like hard drives while the board's power supply is connected and switched on!

Connect a standard ATX supply and switch on the power.

The display shows the BIOS messages. If you want to change the standard BIOS settings, press the <DEL> key to enter the BIOS menu. See BIOS chapter for more details.

If you need to load the BIOS default values, press the <Insert> key during startup. This forces the BIOS to load the factory settings from FlashPROM.

The Cool RoadRunner-PM can boot from CD drives, USB floppy, USB stick, harddisk, or network. Provided that any of these is connected and contains a valid operating system image, the display then shows the boot screen of your operating system.

Note

Not all USB devices are suitable to boot the Cool RoadRunner-PM.

If there are problems, please try to use another device from another manufacturer.

3. Module Description

3.1 Processor

Intel© Pentium© M Processor, 600 MHz … 1.8 GHz.

The Intel Pentium M processor is a high performance, low power mobile processor with several micro-architectural enhancements over existing Intel mobile processors.

Some of the key features of this processor are:

- Supports Intel Architecture with Dynamic Execution
- High performance, low-power core
- On-die, primary 32-kbyte instruction cache and 32-kbyte write-back data cache
- On-die, 1-MByte second level cache with Advanced Transfer Cache Architecture
- Advanced Branch Prediction and Data Prefetch Logic
- Streaming SIMD Extensions 2 (SSE2)
- 400-MHz, Source-Synchronous processor system bus
- Advanced Power Management features including Enhanced Intel® SpeedStep® technology
- Micro-FCPGA and Micro-FCBGA packaging technologies

The Intel Pentium M processor is manufactured on Intel's advanced 0.13-micron process technology with copper interconnects. The processor maintains support for MMX™ technology and Internet Streaming SIMD instructions and full compatibility with IA-32 software. The high performance core features architectural innovations like Micro-op Fusion and Advanced Stack Management that reduce the number of micro-ops handled by the processor. This results in more efficient scheduling and better performance at lower power. The on-die 32-kB Level 1 instruction and data caches and the 1-MB Level 2 cache with Advanced Transfer Cache Architecture enable significant performance improvement over existing mobile processors. The processor also features a very advanced branch prediction architecture that significantly reduces the number of mispredicted branches. The processor's Data Prefetch Logic speculatively fetches data to the L2 cache before an L1 cache requests occurs, resulting in reduced bus cycle penalties and improved performance.

The Streaming SIMD Extensions 2 (SSE2) enables break-through levels of performance in multimedia applications including 3-D graphics, video decoding/encoding, and speech recognition.

The new packed double-precision floating-point instructions enhance performance for applications that require greater range and precision, including scientific and engineering applications and advanced 3-D geometry techniques, such as ray tracing.

3.2 Northbridge

Intel® i82855GME with Chipset Graphics and Memory Controller Hub (GMCH)

- Processor/Host Bus at 400 MHz
- Memory System PC1600/PC2100/PC2700 DDR SDRAM (200/266/333MHz)
- System Interrupts Intel 8259
- Video Stream Decoder improved hardware motion compensation for MPEG2 and Software DVD at 60 Fields/second and 30 frames/second full screen, encoding at low CPU utilization
- Analog display support, 350-MHz integrated 24-bit RAMDAC that can drive a standard progressive scan analog monitor with pixel resolution up to 1600 × 1200 at 85 Hz and up to 2048 × 1536 at 75 Hz

- Dual independent pipe support
    - o *Concurrent*: Different images and native display timings on each display device
    - o *Simultaneous*: Same images and native display timings on each display device
- Dedicated LFP (local flat panel) LVDS interface, Single- or dual-channel LVDS panel support up to UXGA panel resolution with frequency range from 25 MHz to 112 MHz (single channel/dual channel), Supports data format up to 24 bpp
- Internal Graphics Features, 2D/3D graphics engine

For detailed information, please refer to the Intel® i82855GME datasheet.

3.3 Southbridge

Intel® i82801DB I/O Controller Hub 4 (ICH4)

The ICH4 provides extensive I/O support. Functions and capabilities include

- PCI Local Bus Specification, Revision 2.2-compliant with support for 33 MHz PCI operations.
- ACPI Power Management Logic Support
- Enhanced DMA controller, Interrupt controller, and timer functions
- Integrated IDE controller supports Ultra ATA100/66/33
- USB host interface with support for 6 USB ports; 3 UHCI host controllers; 1 EHCI high-speed USB 2.0 Host controller
- Integrated LAN controller
- System Management Bus (SMBus) Specification, Version 2.0 with additional support for I2C devices
- Supports Audio Codec ’97, Revision 2.3 specification (a.k.a., AC ’97 Component Specification, Revision 2.3) Link for Audio and Telephony codecs (up to seven channels)
- Low Pin Count (LPC) interface

3.4 Graphics Controller

The GMCH IGD provides a highly integrated graphics accelerator delivering high performance 2D, 3D, and video capabilities. With its interfaces to UMA using a DVMT configuration, an analog display, a LVDS port, and two digital display ports (e.g. flat panel), the GMCH can provide a complete graphics solution.

The GMCH also provides 2D hardware acceleration for block transfers of data (BLTs). The BLT engine provides the ability to copy a source block of data to a destination and perform raster operations (e.g., ROP1, ROP2, and ROP3) on the data using a pattern, and/or another destination. Performing these common tasks in hardware reduces CPU load, and thus improves performance. High bandwidth access to data is provided through the System Memory interface. The GMCH uses tiling architecture to increase System Memory efficiency and thus maximize effective rendering bandwidth. The Intel 855GME GMCH also improves 3D performance and quality with 3D Zone Rendering technology.

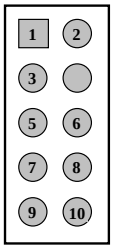
The GMCH has four display ports, one analog and three digital. These provide support for a progressive scan analog monitor, a dedicated dual channel LVDS LCD panel, and two DVO devices.

The Graphics Controller uses an integrated 350-MHz, 24-bit RAMDAC with maximum pixel resolution support up to 1600 × 1200 at 85 Hz and up to 2048 × 1536 at 72 Hz.

VGA connector

Connector type: IDC10 Pin header 2.0 mm (X13)

Pin	Signal	Pin	Signal
1	Red	2	Video-GND
3	Green	4	key
5	Blue	6	DDCCLK
7	H-Sync	8	DDCDAT
9	V-Sync	10	GND



LVDS

The Intel 855GME GMCH has an integrated dual channel LFP Transmitter interface to support LVDS LCD panel resolutions up to UXGA with center and down spread SSC support of 0.5%, 1%, and 2.5% utilizing an external SSC clock. The display pipe provides panel upscaling to fit a smaller source image onto a specific native panel size, as well as provides panning and centering support. The LVDS port is only supported on Pipe B. The LVDS port can only be driven by Pipe B, either independent or simultaneous with the Analog Display port, respectively.

The Dedicated Dual Channel LFP LVDS interface with frequency range of 25 MHz to 112 MHz (single channel/dual channel) support up to UXGA (1600x1200 at 60 Hz) LCD panel resolution with maximum pixel format of 24-bpp.

There is a connector supplying the LC-Display's inverter.

LVDS Connector

Connector type: DF14-30P-1.25H (Hirose) (X5)

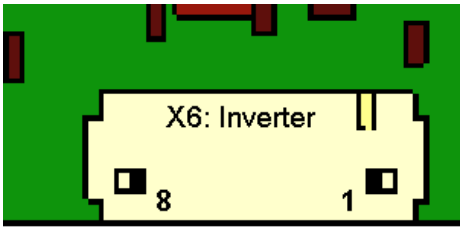
Pin	Signal
1	VDD (3.3 V, opt.5 V)
2	VDD (3.3 V, opt.5 V)
3	GND
4	GND
5	TXA3 -
6	TXA3 +
7	TXACLK -
8	TXACLK +
9	GND
10	TXA2 -
11	TXA2 +
12	TXA1 -
13	TXA1 +
14	TXA0 -
15	TXA0 +
16	GND
17	TXB3 -
18	TXB3 +
19	TXBCLK -
20	TXBCLK +
21	GND
22	TXB2 -
23	TXB2 +
24	TXB1 -
25	TXB1 +
26	TXB0 -
27	TXB0 +
28	GND
29	LVDS DDC-CLK
30	LVDS DDC-DATA

"A" and "B" in the signal names denote the two possible LVDS channels.

Display Backlight Connector

Connector type: Hirose DF13 8 Pin (X6)

Pin	Direction	Signal
1	Output	+12 V DC, max. 1A
2	Output	+12 V DC, max. 1A
3	Output	+5 V DC, max. 1A
4	Output	+5 V DC, max. 1A
5	Output	Signal: Backlight enable (level: 3.3 V)
6	Output	Switched Inverter Power, max. 1A (refer to "Display Voltage Selector" below)
7		GND
8		GND



Display Voltage Selector

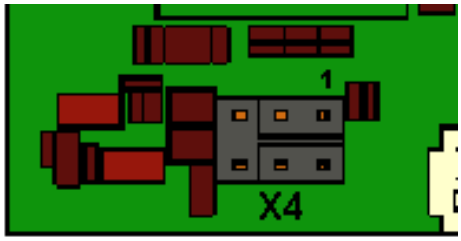
Jumper X4, Location is bottom between X6 (backlight connector) and X18 (supervisory connector).

Connector type: IDC6 jumper 2.00 mm.

Use a jumper between 1-3 or 3-5 to select the display voltage.

Use a jumper between 2-4 or 4-6 to select the backlight voltage.

Pin	Signal name	Pin	Signal name
1	+3,3V DC	2	+12V DC
3	Display voltage	4	Backlight voltage
5	+5V DC	6	+5V DC



Default setup is 3,3V for LVDS display and 12V for the backlight inverter.

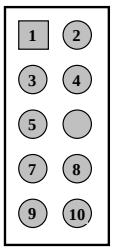
3.5 Network

Ethernet controller: Intel 82562 ET

Ethernet connector

Connector type: IDC10 Pin header 2.0 mm (X24)

Pin	Signal	Pin	Signal
1	TX-	2	TX+
3	RX-	4	RX+
5	n.c.	6	**key**
7	n.c.	8	n.c.
9	n.c.	10	n.c.



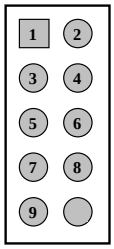
3.6 USB 2.0 Ports

There are six USB 2.0 host ports available for the application on three IDC connectors, each providing two ports.

USB connectors

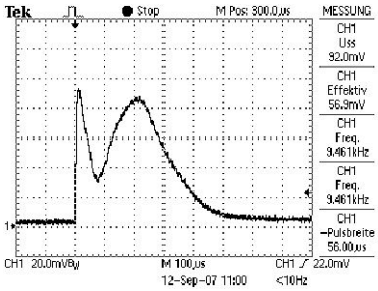
Connector type: IDC10 Pin header 2.0 mm (X20, X21, X22)

Pin	Signal	Pin	Signal
1	VCC USB*n*	2	VCC USB*m*
3	USB*n*-	4	USB*m*-
5	USB*n*+	6	USB*m*+
7	GND	8	GND
9	n.c.	10	**Key**



3.7 On Board Power Supply

The on board power supply generates all necessary voltages from the Mini ATX compliant power supply unit. To use a standard ATX 1.3 compliant power supply unit, an adapter is delivered with the board.



Inrush current, measured over 10 mΩ, with a time division of 100 µs

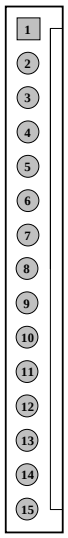
The 3.3V (also 5V, 12V, -12V) available on the PC104 Plus Connector is delivered directly from the external power supply unit, so refer to the specification of your power supply unit for information on maximum available power on the PCI104 Plus connector.

ATX or single voltage (5V DC) supplies will be supported via an adapter cable.

Power Connector

Connector type: JST B15B-EH-A connector (X7)

Pin	Signal (standard)		Signal (5 V only) Note 1
1	+5V		+5V
2	GND		GND
3	+5V		+5V
4	GND		GND
5	+5V	Note 2	+5V
6	+5V Standby		+5V
7	GND		GND
8	PSON#		n.c.
9	PWROK		n.c
10	+3.3V		n.c
11	GND		GND
12	+12V (only for PCI slot and Inverter power supply)		n.c
13	+12V (only for PCI slot and Inverter power supply)		n.c
14	GND		GND
15	-12V		n.c



Note 1: For 5 V-only operations, all +5 V and GND pins must be connected, respectively!

Note 2: Connect pins 5 and 6 when the power is applied via the PC/104 bus!

Real Time Clock Backup

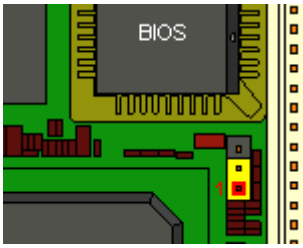
There is a large capacitor ("GoldCap") on board, used to keep the real-time clock running if the power supply is switched of. Without an external battery this GoldCap is able to buffer the date and time information for about one day.

Reset CMOS Jumper

There is a Jumper JP1 on bottom side.

1-2: **default**, route $V_{ext.Bat}$ to V_RTC

Pin	Signal
1	$V_{ext.Bat}$
2	V_RTC Output
3	1K0 pulldown to GND



2-3: CLR-CMOS, initialize clock and CMOS RAM

Pin	Signal
1	$V_{ext.Bat}$
2	V_RTC Output
3	1K0 pulldown to GND

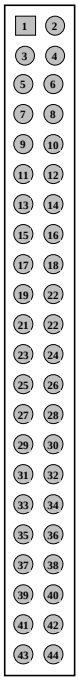
3.8 EIDE Port

An EIDE (**E**nhanced **I**ntegrated **D**rive **E**lectronics) port is provided by the chipset to connect up to two drives that integrate the controller (hard disk, CD-ROM etc.). To enhance the performance, this port has a 100 MB/s IDE controller in UDMA mode per the ATA-5 specification The EIDE port is available on a standard 44-Pin header (2 mm) for 2.5" hard disks. An adapter cable is available to connect standard EIDE devices with a 40 Pin IDC header.

EIDE Connector

Connector type: IDC44Pin header 2.0 mm (X9)

Pin	Signal	Pin	Signal
1	RST#	2	GND
3	Data7	4	Data8
5	Data6	6	Data9
7	Data5	8	Data10
9	Data4	10	Data11
11	Data3	12	Data12
13	Data2	14	Data13
15	Data1	16	Data14
17	Data0	18	Data15
19	GND	20	n.c.
21	DRQ	22	GND
23	IOW#	24	GND
25	IOR#	26	GND
27	RDY	28	CSEL
29	DACK#	30	GND
31	IRQ	32	n.c.
33	Adr1	34	PDIAG#
35	Adr0	36	Adr2
37	CS1#	38	CS3#
39	LED	40	GND
41	+5V	42	+5V
43	GND	44	GND



3.9 Control Interface

The control interface connector combines several interface signals. These are

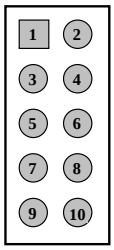
- PS2 keyboard
- PS2 mouse
- Reset input
- Power button
- External battery

The complete connector is shown below, with the individual pin description following separately.

Control Connector

Connector type: IDC10 Pin header 2.0 mm (X2)

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Bat Input	10	PowerButton IN



PS/2 Keyboard Connector

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Bat Input	10	PowerButton IN

PS/2 Mouse Connector

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Bat Input	10	PowerButton IN

Reset Button Connector

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Bat Input	10	PowerButton IN

Power Button Connector

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Bat Input	10	PowerButton IN

External Battery Connector

The $V_{ext.Bat}$ Input can be connected to a power supply (typically a battery) with nominal 3 VDC (2.5 … 3.6 VDC). It draws a current of maximum 10 µA.

Pin	Signal	Pin	Signal
1	GND	2	MouseCLK
3	Reset IN	4	MouseDAT
5	KeybDAT	6	KeybCLK
7	GND	8	+5V KBD PWR
9	$V_{ext.Bat}$ Input	10	PowerButton IN

3.10 Serial Ports

Two serial Ports are located at IDC10 Pin header (X8). A Y-adapter cable with standard DB9 male connectors is available.

The ports either work in **RS232** or **RS485** mode, selectable in the BIOS. Selecting **Special Features**, **Serial Port 1 Mode** and **Serial Port 2 Mode** can be chosen.

Termination resistors for RS485 Mode are set with Jumpers on Pin header X10.

To enable transmitters of Serial Port 1 and 2 in RS485 Mode, program the corresponding RTS signal to 1 using the standard UART programming techniques.

The serial ports can be configured in BIOS setup. Entering **Integrated Peripherals** and then **SuperIO Device** allows configuration of the serial ports.

The following settings are possible for Serial Port 1 and Serial Port 2:

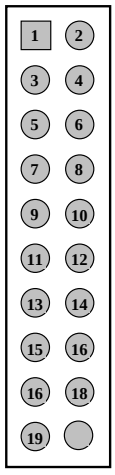
- Disabled
- 3F8 / IRQ4 (base address / interrupt channel)
- 2F8 / IRQ3 (base address / interrupt channel)
- 3E8 / IRQ4 (base address / interrupt channel)
- 2E8 / IRQ3 (base address / interrupt channel)
- Auto

The modes can be switched between RS232 and RS485.

Serial Ports Connector

Connector type: IDC20 Pin header 2.0 mm (X8)

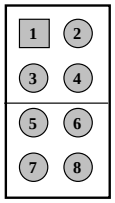
Pin	RS232 Signal	RS485 Signal		Pin	RS232 Signal	RS485 Signal
1	DCD1	Not used		2	DSR1#	RXD1+
3	RXD1	RXD1-		4	RTS1#	TXD1+
5	TXD1	TXD1-		6	CTS1#	Not used
7	DTR1#	Not used		8	RTS11#	Not used
9	GND	GND		10	+5V DC	+5V DC
11	DCD2	Not used		12	DSR2#	RXD2+
13	RXD2	RXD2-		14	RTS2#	TXD2+
15	TXD2	TXD2-		16	CTS2#	Not used
17	DTR2#	Not used		18	RTS22#	Not used
19	GND	GND		20	Key	Key



RS485-Termination Jumpers

Connector type: IDC8 Pin header 2.00 mm (X10)

Pin	Signal	Pin	Signal
1	COM1 TX+	2	COM1 TX-
3	COM1 RX+	4	COM1 RX-
5	COM2 TX+	6	COM2 TX-
7	COM2 RX+	8	COM2 RX-



Use jumpers to terminate the transmission lines correctly.

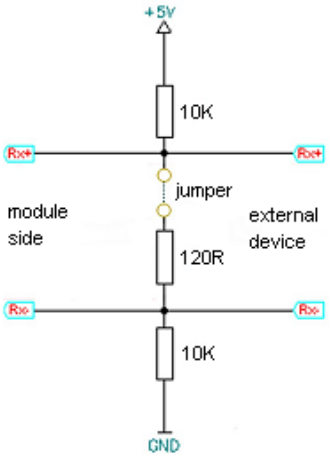
There are two jumpers for each serial port necessary.

The RS485 termination jumpers are located at the top of the printed circuit board

When the jumper is set, the differential pairs
(e.g. RX+ and RX-) are terminated with 120Ω between them.

Additionally, positive/negative receive lines are pulled up/down with 10kΩ to 5V/GND in order to protect the transceivers of the Cool RoadRunner-PM from overvoltages.

It is recommended to protect the ports of the external device in the same way!



Caution: Termination Resistors **must not** be used in RS232 Mode. Otherwise, the serial ports will not work.

3.11 Audio Interface

The Cool RoadRunner-PM features a six channel audio system with an ALC203 CODEC (Realtek).

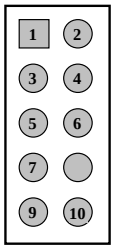
The ALC203 is an 18-bit, full duplex AC97 2.3 compatible stereo audio CODEC designed for PC multimedia systems.

This CODEC also supports an AC97 2.3 compliant S/PDIF out function which allows easy connection of the PC to consumer electronic products, such as AC3 decoder/speaker and mini disk.

Audio Connector

Connector type: IDC10 Pin header 2.0 mm (X17)

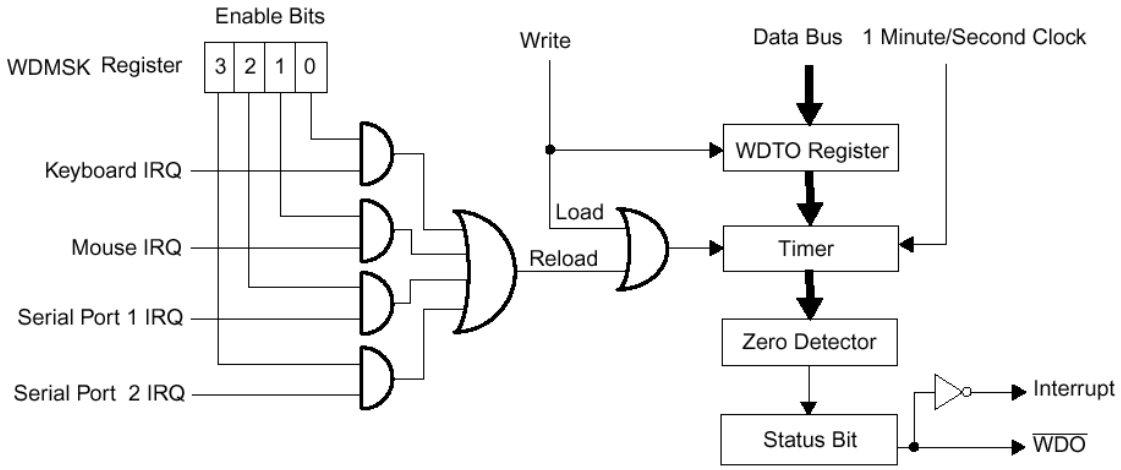
Pin	Signal	Pin	Signal
1	Line In R	2	Line In L
3	Line Out R	4	Line Out L
5	CD IN R	6	CD IN L
7	MicIn 1	8	key
9	SPDIF OUT	10	GND



3.12 Watchdog

The watchdog timer consists of an 8-bit counter and three registers: Timeout register (WDTO), Mask register (WDMSK) and Status register (WDST). The counter is an 8-bit down counter that is clocked every minute or second and is used for the timeout period countdown. The WDTO register holds the programmable timeout, which is the period of inactivity after which the watchdog timer prompts the system (1 to 255 minutes or seconds). The WDMSK register determines which system events are enabled as watchdog Timer trigger events to restart the countdown. The WDST register holds the watchdog timer status bit that reflects the value of the WDO pin and indicates that the timeout period has expired. In addition, it sets the time unit (minutes or seconds). The figure shows the functionality of the watchdog timer.

Note: The watchdog signal remains unchanged up to the next power cycle



The following registers of the Super I/O are used for controlling the watchdog function:

Index register (0x4e)	Data register (0x4f)	Action
0x07	0x0a	Select functional group 10 (Watchdog)
0x60	0x03	Set WDT base address bits (15-8)
0x61	0x70	Set WDT base address bits (7-0)
0x30	0x01	Activate Watchdog

The following table shows the registers of the WatchDog Timer (WDT):

Register	Description
0x370	**WATCHDOG Timeout Register (WDTO)** This register holds the programmable timeout period, which is between 1 and 255 minutes or seconds. Writing to this register de-asserts the WDO output and sets the WDO status bit to 1 (inactive). Additionally, writing to this register is interpreted as a command for starting or stopping the WATCHDOG Timer, according to the data written. If a non-zero value is written, the timer is activated (countdown starts). If a non-zero value is written when the counter is running, the timer is immediately reloaded with the new value and starts counting down from the new value. If 00h is written, the timer and its outputs are deactivated.
0x371	**WATCHDOG Mask Register (WDMSK)** This register is used to determine which system events (IRQ) are enabled as WATCHDOG Timer trigger events. An enabled IRQ event becomes a trigger event that causes the timer to reload the WDTO and restart the countdown.
0x372	**WATCHDOG Status Register (WDST)** This register holds the WATCHDOG Timer status, which reflects the value of the WDO pin and indicates that the timeout period has expired.

See chapter 4.2 for a programming example.

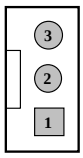
Please check the Super-I/O's datasheet (Winbond, PC87364) for details about the operation and the signals.

3.13 CPU Fan Supply

The Cool RoadRunner-PM provides a connector to power a CPU fan, if the module is actively cooled.

Connector Type: HIROSE-DF13-3Pin-1M25-S (X12)

Pin	Signal
1	Speed Signal from fan (yellow)
2	+5VDC (red)
3	GND (black)



Current at +5VDC output: max.500mA

See chapter 4.3 about how to configure, program and monitor the CPU fan.

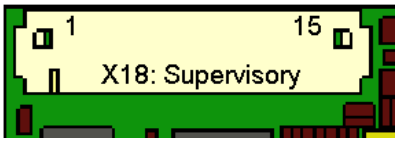
3.14 Supervisory Connector

The Cool RoadRunner-PM provides a 15-Pin Supervisory Connector on its bottom side. The table below shows the assignment of the different signals.
(*The secondary functions are available only with customized versions. Please ask for a quotation*)

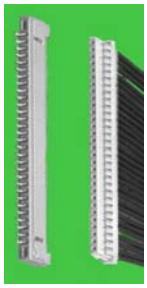
Connector type: Hirose DF13 15-Pin (X18)

Matching plug: Hirose DF13-15S-1.25C

Standard Pin usage:



Pin	Primary	*Secondary (optionally)*
1	+5V (max.25mA)	*+5V (max.25mA)*
2	+3.3V (max.40mA)	*+3.3V (max.40mA)*
3	WD-LED signal	*WD-LED signal*
4	LIVE-LED signal#	*LIVE-LED signal#*
5	IDE-LED signal	*IDE-LED signal*
6	PWRON-LED signal	*PWRON-LED signal*
7	FAN pulse (out)	*PCI_PME#*
8	SMB CLOCK	*GPIO21*
9	SMB DATA	*GPIO22*
10	EXTSMI#	*GPIO23*
11	SERIRQ	*GPIO24*
12	First LAN (X23) Speed LED	*GPIO25*
13	First LAN (X23) Active LED	*GPIO26*
14	First LAN (X23) Link LED	*GPIO27*
15	GND	*GND*



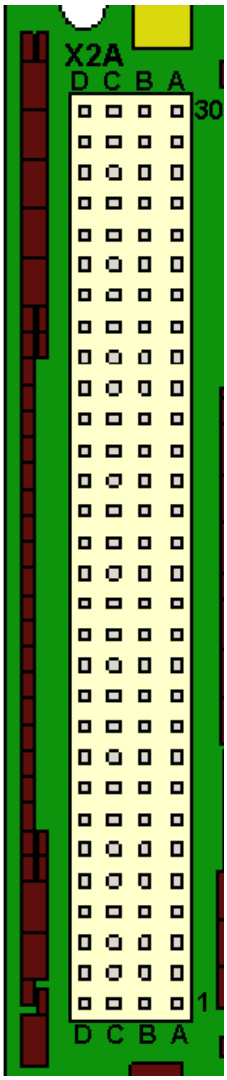
3.15 PC/104-Plus Bus Interface

The PC/104 bus is a modification of the industry standard (ISA) PC bus specified in IEEE P996. The PC/104 bus has different mechanics than P966 to allow the stacking of modules.

The following table shows the Pin assignment of the PC/104 connector.

PC/104 Bus Connector

Pin	D	C	B	A
30	GND	GNT3#	REQ3#	-12 V
29	INTC#	INTB#	INTA#	+12 V
28	RST#	+5 V	INTD#	GND
27	GND	CLK3	+5 V	CLK2
26	CKL1	GND	CLK0	+5 V
25	GND	GNT2#	VI/O	GNT1#
24	GNT0#	+5 V	REQ2#	GND
23	VI/O	REQ1#	GND	REQ0#
22	AD31	GND	AD30	+5 V
21	AD27	AD28	+5 V	AD29
20	GND	AD25	AD26	GND
19	IDSEL3	VI/O	C/BE3#	AD24
18	IDSEL2	IDSEL1	GND	IDSEL0
17	+3.3 V	AD22	AD23	+3.3 V
16	AD19	GND	AD20	AD21
15	GND	AD17	+3.3 V	AD18
14	C/BE2#	+3.3 V	AD16	GND
13	+3.3 V	IRDY#	GND	FRAME#
12	DEVSEL#	GND	TRDY#	+3.3 V
11	GND	LOCK#	+3.3 V	STOP#
10	SDONE	+3.3 V	PERR#	GND
9	PAR	SB0#	GND	SERR#
8	+3.3 V	AD15	C/BE1#	+3.3 V
7	AD12	GND	AD13	AD14
6	M66EN	AD10	VI/O	AD11
5	GND	AD08	AD09	GND
4	AD06	GND	AD07	C/BE0#
3	AD03	AD04	GND	AD05
2	+5 V	AD01	AD02	VI/O
1	AD00	+5 V	Reserved	GND



Note: All VIO Pins are connected to +5V. The voltages +3.3V, +5V, 12V and –12 V are not generated by the onboard power-supply but routed from the Micro ATX Connector. The maximum current is limited to 1.0 amp each

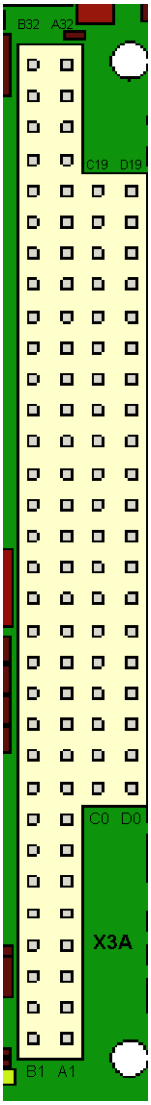
3.16 PC/104 Bus Interface

The PC/104 bus is a modification of the industry standard (ISA) PC bus specified in IEEE P996. The PC/104 bus has different mechanics than P966 to allow the stacking of modules. The main features are:

- Supports programmable extra wait state for ISA cycles
- Supports I/O recovery time for back-to-back I/O cycles

PC/104 Bus Connector

Pin	B	A		Pin	C	D
32	GND	GND				
31	GND	A0				
30	OSC	A1				
29	+5V	A2				
28	BALE	A3		19	n.c. (KEY)	GND
27	TC	A4		18	SD15	GND
26	DACK2#	A5		17	SD14	MASTER#
25	IRQ3	A6		16	SD13	+5V
24	IRQ4	A7		15	SD12	DRQ7
23	IRQ5	A8		14	SD11	DACK7#
22	IRQ6	A9		13	SD10	DRQ6
21	IRQ7	A10		12	SD9	DACK6#
20	SYSCLK	A11		11	SD8	DRQ5
19	REFRESH#	A12		10	MEMW#	DACK5#
18	DRQ1	A13		9	MEMR#	DRQ0
17	DACK1#	A14		8	LA17	DACK0#
16	DRQ3	A15		7	LA18	IRQ14
15	DACK3#	A16		6	LA19	IRQ15
14	IOR#	A17		5	LA20	IRQ12
13	IOW#	A18		4	LA21	IRQ11
12	SMEMR#	A19		3	LA22	IRQ10
11	SMEMW#	AEN		2	LA23	IOCS16#
10	n.c. (KEY)	IOCHRDY		1	SBHE#	MEMCS16#
9	+12V	D0		0	GND	GND
8	n.c.	D1				
7	-12V	D2				
6	DRQ2	D3				
5	n.c.	D4				
4	IRQ9	D5				
3	+5V	D6				
2	RESET	D7				
1	GND	IOCHCK#				



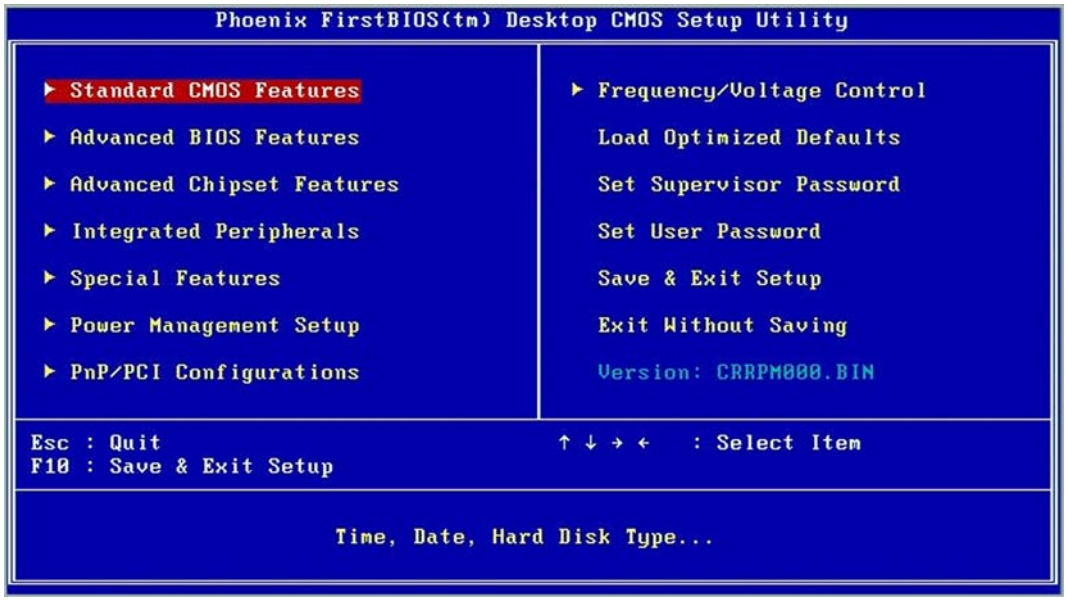
4. Using the Module

4.1 BIOS

The Cool RoadRunner-PM is delivered with a standard PC BIOS. By default, all setup settings are done to have a "ready to run" system, even without a BIOS setup backup battery. The BIOS is located in a Flash PROM and can be easily updated on-board.

Setup

Pressing <**F2**> or <**DEL**> at power-up starts the setup utility.



Initialize BIOS at first startup

It is important to initialize the BIOS setting at first startup of the board.

Call setup by pressing <**F2**> or <**DEL**> at power-up and executed **Load Optimized Defaults**. Then use **Save & Exit Setup** to save and activate the new settings.

The "Optimized Defaults" is the optimized BIOS setup for the Cool RoadRunner-PM

Booting from alternative device

Pressing the <**ESC**> key at power-up starts the Boot Menu. Choose one of the listed bootable devices for booting.

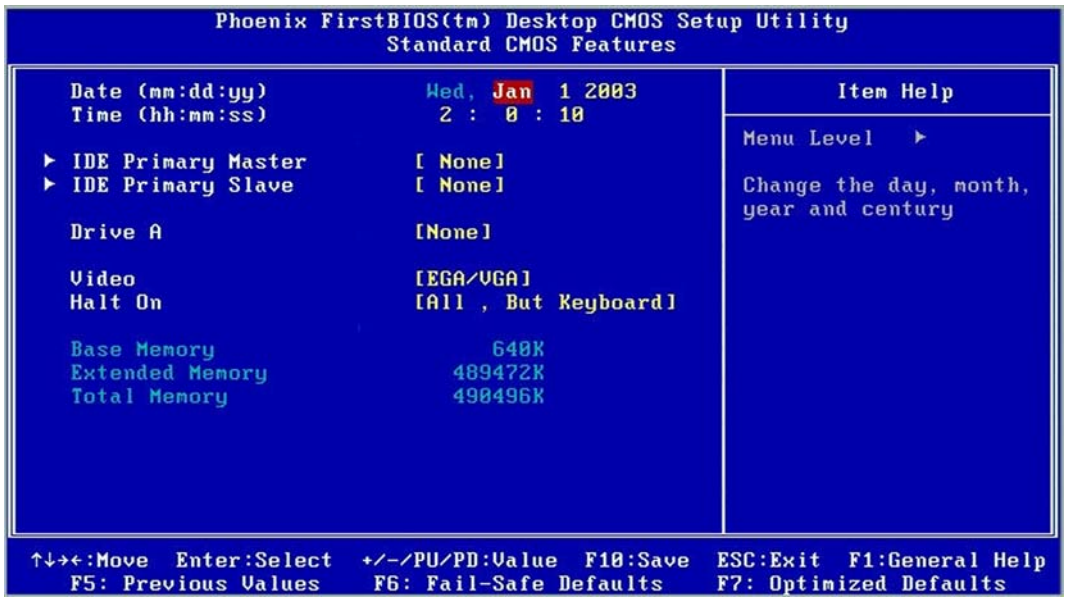
Reload default BIOS values

The default values of the BIOS can be automatically reloaded at boot time. Press <0/INSERT> on the NUM pad before the system is turned on. Holding this key and turning the board on, the default values will be loaded.

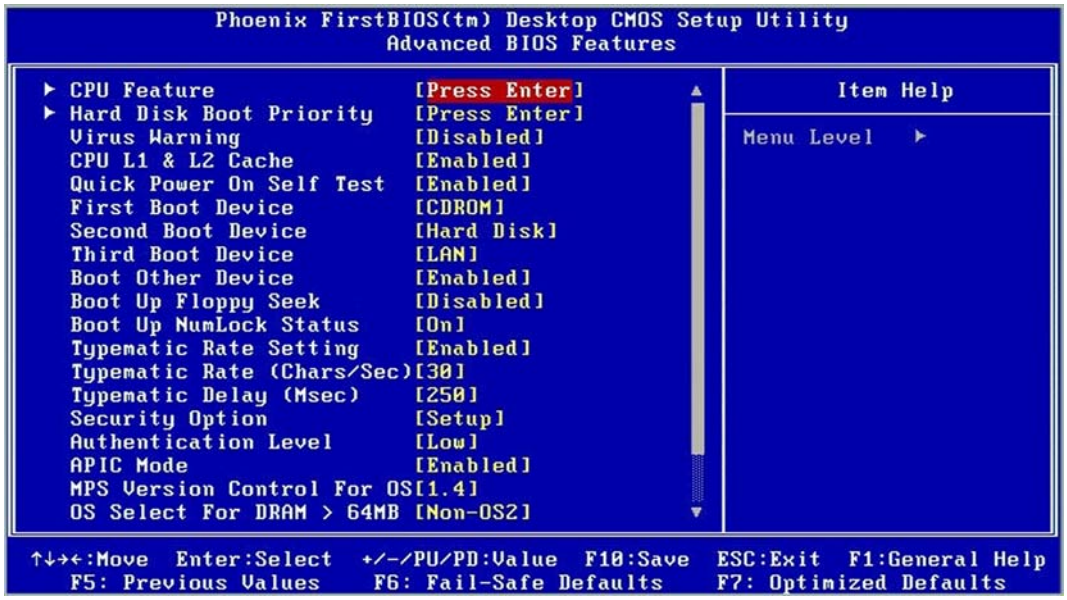
BIOS Screens

The BIOS setup utility allows setting of various board parameters. The following pictures show the different setup menus. The Cool RoadRunner-PM specific settings are explained here.

Standard CMOS Features



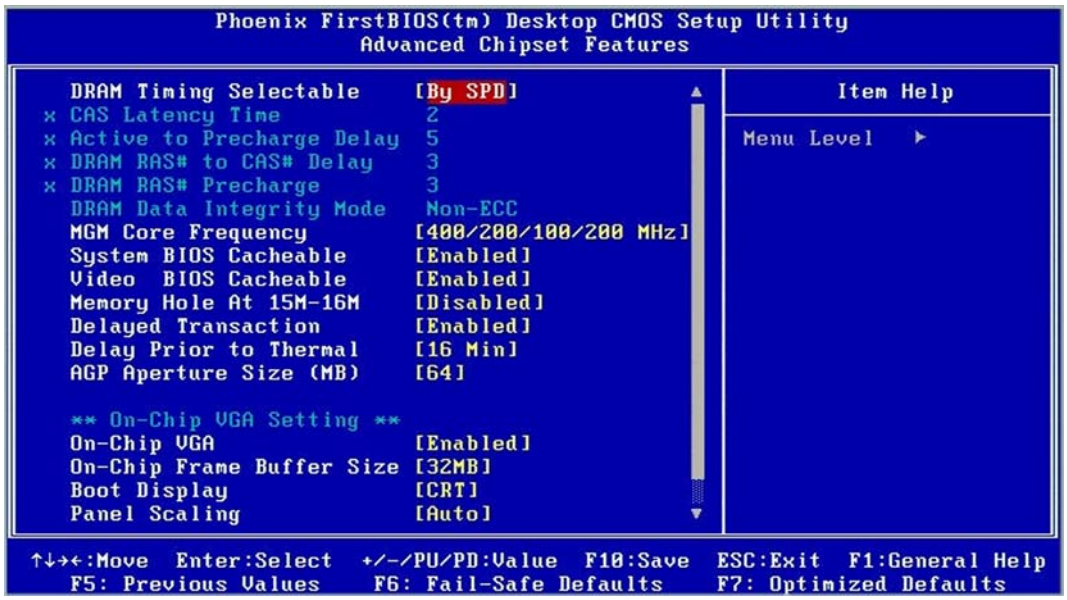
Advanced BIOS Features



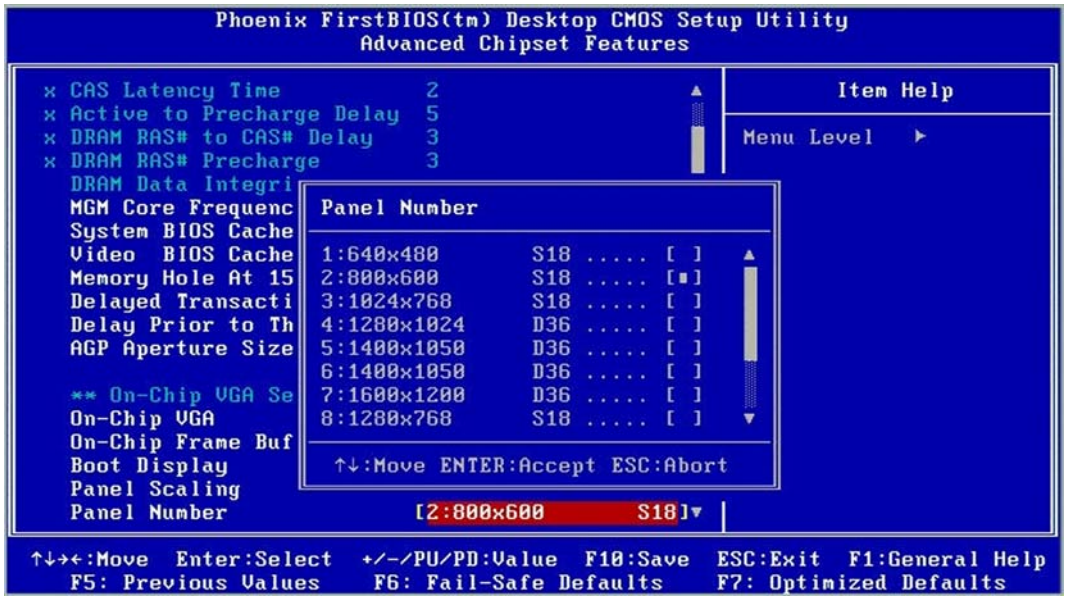
Advanced BIOS Features – CPU Feature

Using TM2 Bus Ratio, the processor's clock frequency can be decreased n order to avoid overheating. TM2 Bus VID changes the core voltage, also to avoid overheating.

Advanced Chipset Features

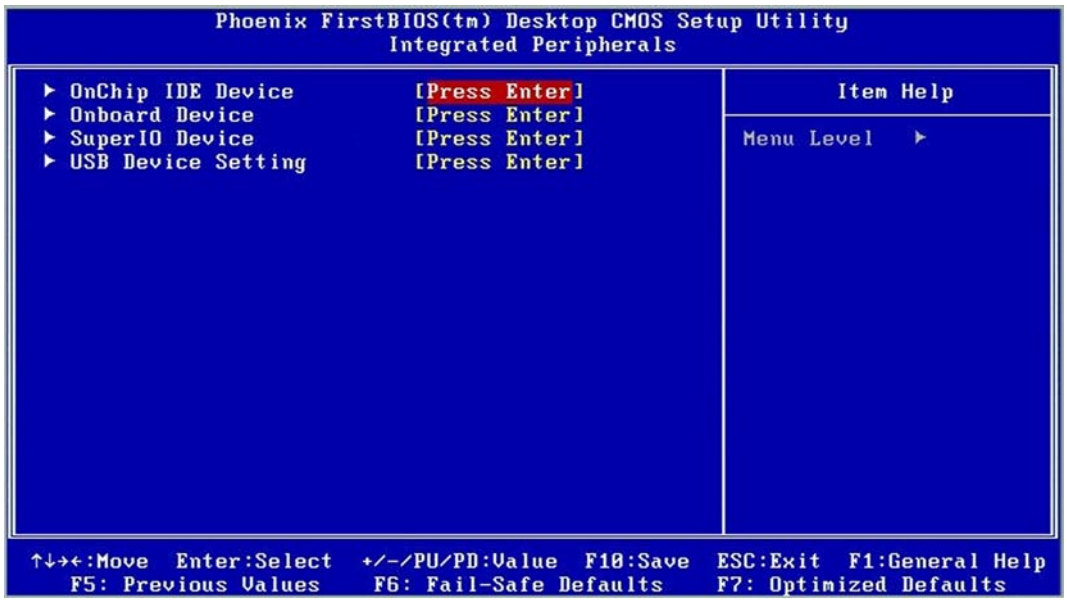


Advanced Chipset Features – Panel Number

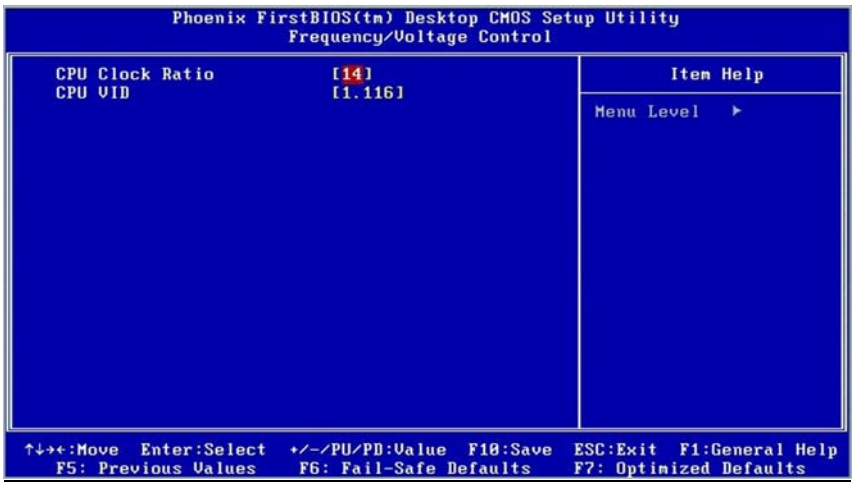


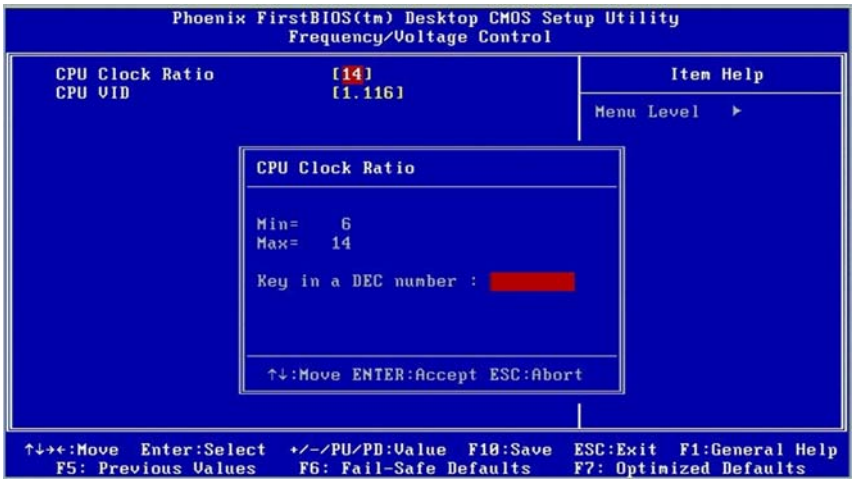
With the Panel Number settings, the LVDS interface can be configured. States marked with "S" are for single channel, states marked with "D" for dual channel operation. The numbers give the screen resolution used for the display.

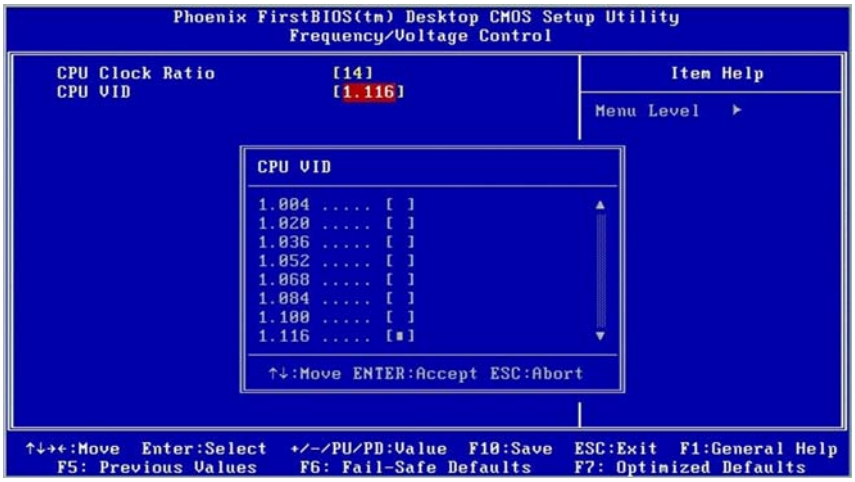
Integrated Peripherals



Frequency/Voltage Control





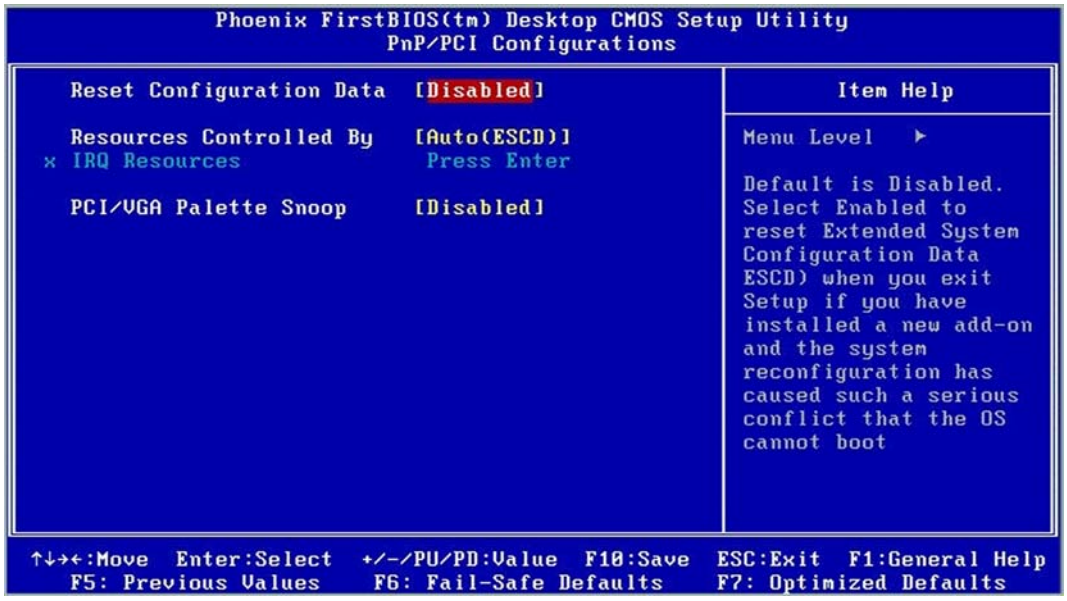


The CPU clock ratio is set by the BIOS, depending on the processor version used. Usually, this is the best setting possible. Changing these values is only recommended for very experienced users and should only be attempted after studying the processor's datasheet.

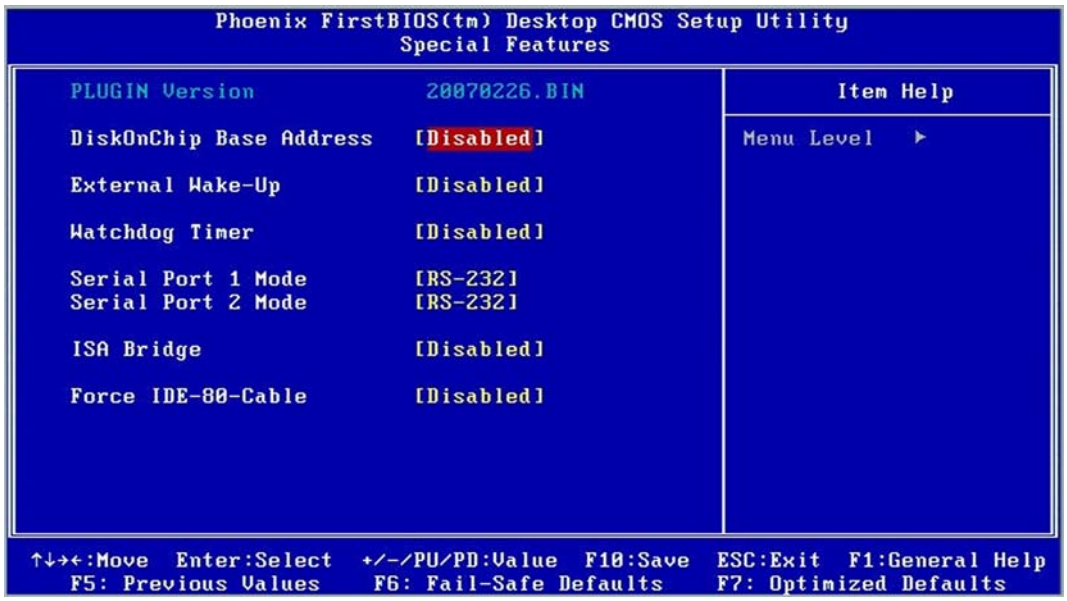
Caution

Use this feature on your own risk.

PnP/PCI Configurations



Special Features

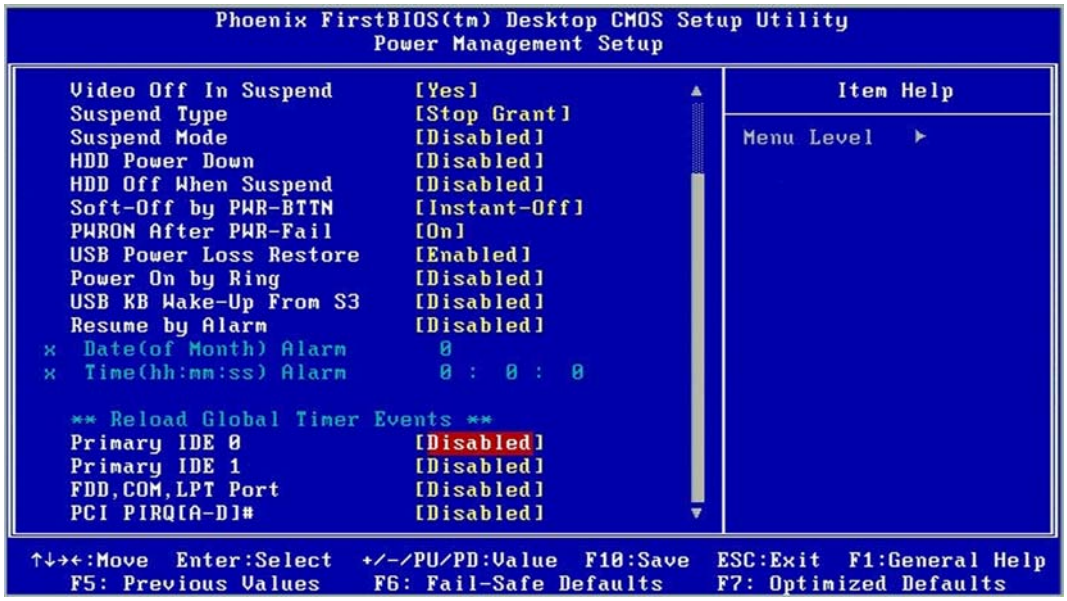


Power Management Setup, part 1

S1: POS – Power On Suspend
No instructions are executed by the processor, RAM contents are preserved

S3: Suspend To RAM (STR), Standby
The current processor context is saved to RAM, the processor itself and most peripherals are switched off. RAM content is preserved by hardware.

Power Management Setup, part 2



When enabled, the Reload Global Timer Events allow restarting the global standby timer when such an event occurs.

4.2 Watchdog Programming

The following example show how to configure and handle the watchdog.

It is meant to be compiled using gcc under Linux.

```
#include <stdio.h>
#include <sys/io.h>
#include <unistd.h>

#define GPIO_ADDR  0x4E
#define GPIO_DATA  0x4F

int main()
{
  unsigned char i;
  iopl(3);  //get all I/O access rights
  // watchdog trigger sequence
  outb(0x07, GPIO_ADDR);       // select functional group
  outb(0x0a, GPIO_DATA);       // set functional group 10 (watchdog)
  outb(0x60, GPIO_ADDR);       // select WDT base address bits (15-8)
  outb(0x03, GPIO_DATA);       // set WDT base address bits (15-8) to 0x03
  outb(0x61, GPIO_ADDR);       // select WDT base address bits (7-0)
  outb(0x70, GPIO_DATA);       // set WDT base address bits (7-0) to 0x70
  outb(0x30, GPIO_ADDR);       // select enable/disable register
  outb(0x01, GPIO_DATA);       // enable watchdog function device

  //pci device: (bus:0, dev:1f, function:0)
  outl(0x8000f858, 0xcf8); //select decode range of LPC bridge
  outw(0x0371, 0xcfc);  // write 128bit adress range beginning at 0x800 to address 0xe4-0xe5 of
LPC bridge device
                        // bit0 = 1 -> enable

  outb(0x00, 0x371);    // disable all IRQs as WDT trigger event
  outb(0x80, 0x372);    // set time unit to seconds
  outb(0x84, 0x372);    // set time unit to seconds
  outb(0x03, 0x370);    // set timeout to 3 seconds and activate WDT
  printf("Watchdog aktivated. Press CTRL+C within 5 seconds to stop resetting the
watchdog.\n");
  for(i=0; i<5; i++)
  {
    outb(0x03, 0x370);  // reset WDT
    printf(".");
    fflush(stdout);
    sleep(1);
  }
  outb(0x00, 0x370);    // deactivate WDT
  printf(" watchdog disabled\n");
  return 0;
}
```

4.3 CPU Fan Programming

Fan control

The example shows how to control the CPU fan speed in a range from 0...255. It meant to be compiled using gcc under Linux.

```
#include <stdio.h>
#include <sys/io.h>
#include <unistd.h>

#define SIO_BASE 0x4E
#define SIO_DATA 0x4F

int main(int argc, char *argv[])
{
  iopl(3);              // gives you access to all ports -> as root !!!!
// initialize SIO
  outb( 0x07,SIO_BASE); // FAN LDN
  outb( 0x09,SIO_DATA);
  outb( 0x30,SIO_BASE); // disable ACB
  outb( 0x00,SIO_DATA);
  outb( 0x60,SIO_BASE); // FAN MSB
  outb( 0x08,SIO_DATA);
  outb( 0x61,SIO_BASE); // FAN LSB
  outb( 0x10,SIO_DATA);
  outb( 0x22,SIO_BASE); // Pin = FANOUT0
  outb( 0x31,SIO_DATA);
  outb( 0xF0,SIO_BASE); // enable fan
  outb( 0x0C,SIO_DATA);
  outb( 0x30,SIO_BASE); // enable ACB
  outb( 0x01,SIO_DATA);

  //pci device: (bus:0, dev:1f, function:0)
  outl(0x8000f8e4, 0xcf8); //select decode range of LPC bridge
  outw(0x0801, 0xcfc);     // write 128bit adress range beginning at 0x800 to
                           //address 0xe4-0xe5 of LPC bridge device
                           // bit0 = 1 -> enable
  if(argc == 2)
  {
    outb(0xc0,0x810);   // set fan speed
    outb((unsigned char)strtol(argv[1],(char**)NULL,16),0x811);
  }
  else
    printf("USAGE: ./fanctrl <value: 00..ff>\n");
  iopl(0);              // disable access
  return 0;
}
```

Fan monitoring

The CPU fan's speed can be checked with the following routine. It also is meant to be compiled using gcc under Linux.

```
#include <stdio.h>
#include <sys/io.h>
#include <unistd.h>

#define SIO_BASE 0x4E
#define SIO_DATA 0x4F

int main(int argc, char *argv[])
{
  unsigned char speed, ctrl, ctrlout;
  iopl(3);              // gives you access to all ports -> as root !!!!
// initialize SIO
  outb(0x07, SIO_BASE); // FAN LDN
  outb(0x09, SIO_DATA);
  outb(0x60, SIO_BASE); // FAN MSB
  outb(0x08, SIO_DATA);
  outb(0x61, SIO_BASE); // FAN LSB
  outb(0x10, SIO_DATA);
  outb(0x22, SIO_BASE); // Pin = FANOUT0
  outb(0x31, SIO_DATA);

  outb(0x30, 0x818);  //monitor 0: Control & Status

  while(1)
  {
    ctrlout = 0x20;
    ctrl = inb(0x818);
    if((ctrl & 0x4) == 0x4)
      ctrlout |= 0x4;
    outb(ctrlout, 0x818);             //reset OVER THRESHOLD
    if((ctrl & 0x1) == 0x1)
    {
      speed = inb(0x817);      // monitor 0 -> offset 0x07
      printf("FANSPEED:%d          \r",(unsigned int)(60.0 * 8000.0 / ((float)speed * 4.0)));
      fflush(stdout);
    }
  }
  return 0;
}
```

4.4 CPU and Ambient Temperature Measurement

This example shows how to read the two temperature sensors, which are connected to the chipset via the I²C bus. Again, this is meant to be compiled using gcc under Linux.

```
#include <stdio.h>
#include <sys/perm.h>
#include <unistd.h>
#include <sys/io.h>

#define SIO_BASE                0x4E    // fix
#define SIO_DATA                0x4F    // fix
#define IDX_DEVICE_SEL          0x7     // fix
#define IDX_SIOCF3              0x23    // SIO Config register 3
#define IDX_DEVICE_ENABLE       0x30    // Device Enable Register
#define IDX_GPIO_BASE_HIGH      0x60    // fix
#define IDX_GPIO_BASE_LOW       0x61    // fix
#define IDX_GPIO_PIN_SEL        0xf0    // GPIO Pin Select register
#define IDX_GPIO_PIN_CFG        0xf1    // GPIO Pin Configuration register
#define GPIO_DEVICE_NUM         0x7     // Device Number of GPIO

// hardcoded address byte.
#define DAB             0x98
#define SDA                     3      // GPIO13 (Port 1, Pin 3)
#define SCL                     2      // GPIO12 (Port 1, Pin 2)

// Globals.
unsigned int GPIOBase = 0;


void WriteGPIOPort1(unsigned char GPIOBitSel, unsigned char bitVal)
{
  if (GPIOBitSel == SDA)
  { // Configure SDA as output.
    outb(IDX_GPIO_PIN_CFG,SIO_BASE);     // GPIO Pin Configuration register
    outb(0x45,SIO_DATA);                 // Set SDA as output, open-drain, no internal pull-up
  }
  if (bitVal) // Set or clear the corresponding bit.
    outb(inb(GPIOBase+0x05) | (1 << GPIOBitSel),GPIOBase+0x04);
  else
    outb(inb(GPIOBase+0x05) & ~(1 << GPIOBitSel),GPIOBase+0x04);
}


unsigned char ReadGPIOPort1(unsigned char GPIOBitSel)
{
  unsigned char retVal;
  if (GPIOBitSel == SDA)
  { // Configure SDA as input.
    outb(IDX_GPIO_PIN_CFG,SIO_BASE);       // GPIO Pin Configuration register
    outb(0x44,SIO_DATA);                   // Set SDA as input, open-drain, no internal pull-up
  }
  retVal = inb(GPIOBase+0x05);         // Read the GPIO pins.
  return (retVal >> GPIOBitSel) & 0x1; // Return the bit value.
}


void SendSTART()
{
  WriteGPIOPort1(SDA, 1);  // Change the SDA line from HIGH -> LOW while the SCL line is HIGH.
  WriteGPIOPort1(SCL, 1);
  WriteGPIOPort1(SDA, 0);
  WriteGPIOPort1(SCL, 0);
}


void SendSTOP()
{
  WriteGPIOPort1(SDA, 0);  // Change the SDA line from LOW -> HIGH while the SCL line is HIGH.
  WriteGPIOPort1(SCL, 1);
  WriteGPIOPort1(SDA, 1);
}


void SendACK(unsigned char ucVal)
```

```
{   // ucVal=0 -> ACK ; ucVal=1 -> NACK
  WriteGPIOPort1(SDA, ucVal & 0x1);
  WriteGPIOPort1(SCL, 1);
  WriteGPIOPort1(SCL, 0);
}


unsigned char RecACK()
{
  unsigned char sdaVal;
  WriteGPIOPort1(SDA, 1);
  WriteGPIOPort1(SCL, 1);
  sdaVal = ReadGPIOPort1(SDA);
  WriteGPIOPort1(SCL, 0);
  return !sdaVal;
}


void SendI2CByte(unsigned char byteVal)
{
  unsigned char i;
  for(i = 8; i > 0; i--) // Send the byte.
  {
    WriteGPIOPort1(SDA, (byteVal >> (i - 1)) & 0x1);
    WriteGPIOPort1(SCL, 1);
    WriteGPIOPort1(SCL, 0);
  }
  WriteGPIOPort1(SDA, 1);// Release the SDA line by setting to HIGH.
}


unsigned char RecvI2CByte()
{
  unsigned char byteVal = 0, i;
  for (i = 8; i > 0; i--)
  {
    WriteGPIOPort1(SCL, 1);
    byteVal |= ReadGPIOPort1(SDA) << (i - 1);
    WriteGPIOPort1(SCL, 0);
  }
  return byteVal;
}


void WriteRegUchar(unsigned char regIndex,unsigned char* byteVal,unsigned char NumberOfBytes)
{
  unsigned char error;
  SendSTART();                        // Send START condition.
  SendI2CByte(DAB | 0);               // Send Device address byte.
  error = RecACK();                   // Send an ACK pulse.
  if (!error)
    printf("Error sending DAB on I2C.\n");
  SendI2CByte(regIndex);              // Send register address byte.
  error = RecACK();                   // Send an ACK pulse.
  if (!error)
    printf("Error sending DAB on I2C.\n");
  while (NumberOfBytes-- && error!=0) // Send the register value(s).
  {
    SendI2CByte(*byteVal++);
    error = RecACK();                 // Send an ACK pulse.
    if (!error)
      printf("Error sending register value on I2C.\n");
  }
  SendSTOP();                         // Send STOP.
}


void ReadRegUchar(unsigned char regIndex, unsigned char* byteVal, unsigned char NumberOfBytes)
{
  unsigned char error;
  SendSTART();                // Send the START condition.
  SendI2CByte(DAB | 0);       // Send Device address byte.
  error = RecACK();           // Send an ACK pulse and record error.
  if (!error)
    printf("Error sending DAB (write) on I2C.\n");
  SendI2CByte(regIndex);      // Send Register address byte.
```

```
  error = RecACK();
  if (!error)
    printf("Error sending RAB on I2C.\n");
  SendSTART();                 // Send a RESTART condition.
  SendI2CByte(DAB | 1);        // Send Device address byte (with READ bit set).
  error = RecACK();
  if (!error)
    printf("Error sending DAB (read) on I2C.\n");
  while (NumberOfBytes-- && error!=0)
  {
    *byteVal++ = RecvI2CByte();// Read the data byte.
    if (NumberOfBytes)
      SendACK(0);
    else
      SendACK(1);
  }
  SendSTOP();                  // Send STOP condition.
}


void I2CInit()
{
// select and enable GPIO device 7:
  outb(IDX_DEVICE_SEL,SIO_BASE);
  outb(GPIO_DEVICE_NUM,SIO_DATA);        // Select device 7, GPIO
  outb(IDX_DEVICE_ENABLE,SIO_BASE);     // Device Enable Register
  outb(0x01,SIO_DATA);                   // Enable device 7, GPIO
// Get GPIO base:
  outb(IDX_GPIO_BASE_HIGH,SIO_BASE);
  GPIOBase = inb(SIO_DATA) << 8;
  outb(IDX_GPIO_BASE_LOW,SIO_BASE);
  GPIOBase |= inb(SIO_DATA) & 0xFF;
  if (!GPIOBase)
  { // Set IO-Base of GPIOs to 0x600
    GPIOBase = 0x600;
    outb( IDX_GPIO_BASE_HIGH,SIO_BASE);
    outb( GPIOBase >> 8,SIO_DATA);
    outb( IDX_GPIO_BASE_LOW,SIO_BASE);
    outb( GPIOBase & 0xFF,SIO_DATA);
  }
  // Configure SDA and SCL GPIO pins.
  outb(IDX_SIOCF3,SIO_BASE);                  // SIO Config register 3
  outb(0x03,SIO_DATA);                        // Configure Pin 54, 55 as GPIOs
  outb(IDX_GPIO_PIN_SEL,SIO_BASE);            // GPIO Pin Select register
  outb(0x10|SCL,SIO_DATA);                    // Select SCL Pin
  outb(IDX_GPIO_PIN_CFG,SIO_BASE);            // GPIO Pin Configuration register
  outb(0x45,SIO_DATA);                        // Set SCL as output, open-drain, no
                                              //internal pull-up
  outb(IDX_GPIO_PIN_SEL,SIO_BASE);            // GPIO Pin Select register
  outb(0x10|SDA,SIO_DATA);                    // Select SDA Pin
  outb(IDX_GPIO_PIN_CFG,SIO_BASE);            // GPIO Pin Configuration register
  outb(0x44,SIO_DATA);                        // Set SDA as input, open-drain,
                                              // no internal pull-up
  SendSTOP();                                 // Send initial STOP condition.
}


void LM90Init()
{
  unsigned char data;
  data = 0x07;          // 8 Conversions/Sec
  WriteRegUchar(0x0a, &data ,1);
  data = 0x02;          // Filter Level 1
  WriteRegUchar(0xbf, &data ,1);
}


signed char GetTemp_AMBIENT()
{
  unsigned char data[2];
  ReadRegUchar(0x00, data, 1);
  return (-1 - ~data[0]);
}


float GetTemp_CPU()
```

```
{
  unsigned char data[2], low;
  signed char CPUTemp;
  ReadRegUchar(0x01, &data[0] ,1);
  ReadRegUchar(0x10, &data[1] ,1);
  low = ((int)(data[1]>>5) * 125 + 50) / 100;
  CPUTemp = -1 - ~data[0];
  return CPUTemp + (float)low/10;
}


int main()
{
  iopl(3);              // gives you access to all ports -> as root !!!!
  I2CInit();
  outl(0x8000f8e4, 0xcf8);
  outw(GPIOBase+1, 0xcfc);
  LM90Init();
  printf("Press CTRL+C to cancel!\n");
  printf("Temp:  CPU   AMBIENT\n");
  while(1)
  {
    printf("       %.1f    %d \r", GetTemp_CPU(), GetTemp_AMBIENT());
    fflush(stdout);
    usleep(100);
  }
  return 0;
}
```

5. Address Maps

This section describes the mapping of the CPU memory and I/O address spaces. Also covered in this section is the PCI configuration space mapping.

Note - Depending on enabled or disabled functions in the BIOS, other or more resources may be used.

Memory Address Map

Address Range (Hex)	Description
000000–09FFFF	Conventional Memory
0A0000–0AFFFF	VGA Adapter
0B0000–0BFFFF	VGA Adapter
0C0000–0DFFFF	Adapter ROM
0E0000–0EFFFF	System
0F0000–0FFFFF	System Bios
100000–1DEDFFFF	Extended Memory
E0000000-E7FFFFFF	Graphics Controller
EC000000-EC01FFFF	Ethernet Controller
EC020000-EC02FFFF	Ethernet Controller
EC021000-EC021FFF	Ethernet Controller
EC100000-EC17FFFF	Graphics Controller
EC180000-EC1803FF	USB2 Controller
EC181000-EC1811FF	AC97 Audio
EC182000-EC1820FF	AC97 Audio
F0000000-F7FFFFFF	Graphics Controller
FEB00000-FEB7FFFF	Graphics Controller
FEBFFC00-FEBFFFFF	IDE Controller
FFB80000-FFBFFFFF	Firmware Hub

I/O Address Map

The system chipset implements a number of registers in I/O address space. These registers occupy the following map in the I/O space (depending on enabled or disabled functions in the BIOS other or more resources may be used).

Address Range (Hex)	Description
0000-000F	DMA Controller 1 (8237)
0020-0021	Interrupt Controller 1 (8259)
0040-0043	Timer Controller (8254)
0060	Keyboard Controller Data Byte
0061	Speaker Control
0064	Kbd Ctlr, CMD,STAT Byte
0070-0073	Real Time Clock
0078	internal
0079	Watchdog
0080-009F	DMA Page Registers
00A0-00A1	Interrupt Controller 2 (8259)
00C0-00DF	DMA Controller 1 (8237)
00F0-00FF	Math Coprocessor
01F0-01F7	Primary IDE Channel
0200-020F	GPIO Registers
0274-0279	ISAPnP Data port
02F8-02FF	Serial Port 2
0300-031F	User Area for Prototype Cards
0370-0373	Watchdog
0376	IDE Controller
0378-037F	Parallel Port 1
03B0-03BB	VGA Registers
03C0-03DF	VGA Registers
03F0-03F5	Floppy Controller Registers
03F6	IDE Command Port
03F7	Floppy Command Port
03F8-03FF	Serial Port 1
0400-04BF	PCI Bus
04D0-04D1	Interrupt Controller
0500-051F	SMBus Controller
0800-087F	Motherboard Resource
0A79-A79	ISAPnP Data port
0CF8-0CFF	PCI Bus
14F0-14FF	Bus Master IDE I/O Registers

Address Range (Hex)	Description
9000-9FFF	PCI-PCI Bridge
A000–A01E	USB Controller
A400–A41E	USB Controller
A800–A81E	USB Controller
AC00–AC06	VGA Adapter
B400-B4FF	Audio Mixer
B800-B8FF	Audio Bus
D000-D03F	Ethernet Controller
D100-D13F	Ethernet Controller
E000-E0FF	AC´97 Audio
E800-E81F	USB Controller
E900-E907	Graphics Controller
EB00-EB1F	USB Controller
EC00-EC3F	AC´97 Audio
ED00-ED1F	USB Controller
F000–F00F	IDE Controller

Interrupts

IRQ (Bus)	System Resource
NMI	Parity Error
0 (ISA)	Timer
3 (ISA)	Serial Port 2
4 (ISA)	Serial Port 1
6 (ISA)	Floppy
7 (ISA)	Not used
8 (ISA)	Real Time Clock
9 (ISA)	User available (PCI)
13 (ISA)	Math coprocessor
14 (ISA)	IDE Controller
15 (PCI)	Ethernet Controller
16 (PCI)	USB Controller
16 (PCI)	Graphics Controller
17 (PCI)	AC´97 Audio
18 (PCI)	USB Controller
19 (PCI)	USB Controller
20 (PCI)	Ethernet Controller
23 (PCI)	USB Controller

DMA Channels

DMA	System Resource
0	User available
1	User available
2	User available
3	User available
4	DMA Controller
5	User Available
6	User Available
7	User Available

6. Troubleshooting

First steps if the Board does not boot:

- Check the status LED's on the board. Are all voltages properly available?
- Check the power connectors to the board, monitor and additional devices.
- Are all cables plugged on the correct connector and in the correct orientation? The board may not boot if some of the cables are not plugged in correctly!
- Is a RAM module inserted on CPU Board?
- Check the power supply. Is the supply voltage correct for the board? If you are not sure, read the manual. Try plugging in a different power supply or multi-meter to check the power a wrong supply voltage can easily FRY a computer and other electrical devices.
- Is your display ok? Is the monitor powered on? Is the monitor's video cable plugged into the video connector? Double-check the brightness and contrast settings. Plug the monitor into another computer if possible to verify the monitor isn't the problem.
- Remove all additional devices from the system. Only the processor board, power supply, monitors and the keyboard should remain in the system.
- Replace the system RAM
- Assure your cooling measures work correctly and keep the processor at a reasonable temperature.
- If all else has failed, replace the CPU Board itself.
- If system comes up then load at first the OPTIMIZED DEFAULTS in the BIOS setup and reboot.

If you need to send the board to LiPPERT for repair, be sure you get a Return Material Authorization number (RMA) first.
Check also Appendix B (Getting Help).

Appendix A, Contact Information

Headquarters

LiPPERT Embedded Computers GmbH

Hans-Thoma-Straße 11

68163 Mannheim

Germany

Phone	+49 621 432140
Fax	+49 621 4321430
E-mail	sales@lippertembedded.com
	support@lippertembedded.com
Website	www.lippertembedded.com

US Office

LiPPERT Embedded Computers, Inc.

5555 Glenridge Connector, Suite 200

Atlanta, GA 30342

USA

Phone	+1 (404) 459 2870
Fax	+1 (404) 459 2871
E-mail	ussales@lippertembedded.com
	support@lippertembedded.com
Website	www.lippertembedded.com

Appendix B, Getting Help

Should you have technical questions that are not covered by the respective manuals, please contact our support department at **support@lippertembedded.com**.

Please allow one working day for an answer!

Technical manuals as well as other literature for all LiPPERT products can be found in the *Products* section of LiPPERT's website www.lippertembedded.com. Simply locate the product in question and follow the link to its manual.

Returning Products for Repair

To return a product to LiPPERT for repair, you need to get a Return Material Authorization (RMA) number first.
Please print the RMA Request Form from http://www.lippertembedded.com/service/repairs.html fill in the blanks and fax it to +49 621 4321430. We'll return it to you with the RMA number.

Deliveries without a valid RMA number are returned to sender at his own cost!

LiPPERT has a written Warranty and Repair Policy, which can be retrieved from http://www.lippertembedded.com/service/warranty.html

It describes how defective products are handled and what the related costs are. Please read this document carefully before returning a product.

Appendix C, Further Resources

http://www.lippertembedded.com

LiPPERT Embedded Computers' website with news and detailed information.

http://www.intel.com

Datasheet of the CPU, Chipset and Ethernet-Controller.

http://www.pc104.org/

The PC/104 Consortium.
A copy of it and the latest PC/104 and PC104-Plus specifications can be obtained from this website.

http://www.smbus.org

Information about the System Management Bus (SMBus)

http://www.phoenix.com/en/customer+services/bios/awardbios

Additional BIOS information.

http://www.winbond.com/

The manufacturer of the Super I/O chip.
Check there for the datasheet of the PC87364.

Appendix D, Revision History

Filename	Date	Edited by	Change
TME-104P-CRRPM-R0V0	2007-03-28	Alfred Glass	Prerelease/Draft for PCB 1.0
TME-104P-CRRPM-R0V1	2007-05-30	Alfred Glass J. Stauffer	Watchdog, I/O-Address-Map, block diagram updated, Trouble Shooting
TME-104P-CRRPM-R1V0	2007-07-05	PK	Accessory list completed Technical data completed Watchdog description Various editorial changes
TME-104P-CRRPM-R1V1	2007-12-07	PK	Code examples Inrush current Jumper location Editorial changes
TME-104P-CRRPM-R1V2	2007-12-28	MF	Correction Power connector type
TME-104P-CRRPM-R1V3	2008-03-04	PK	Current consumption 600MHz added LAN connector corrected
TME-104P-CRRPM-R1V4	2008-05-15	PK	Power connector description improved
TME-104P-CRRPM-R2V0	2009-04-02	AG	Reset CMOS Jumper JP1, Jumper Locations on bottom side, PCB 2V0, X8/pin20 removed
TME-104P-CRRPM-R3V0	2009-04-28	AG	PCB 3V0
TME-104P-CRRPM-R3V1	2009-07-07	MF	Text description of X4 corrected
TME-104P-CRRPM-R3V2	2009-08-25	PK	Phone number and links corrected