CoreModule™ 410
PC/104 CPU Module
Reference Manual

P/N 5001689A Revision A

Notice Page

NOTICE

DISCLAIMER

Ampro Computers, Incorporated makes no representations or warranties with respect to the contents of this manual or of the associated Ampro products, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Ampro shall under no circumstances be liable for incidental or consequential damages or related expenses resulting from the use of this product, even if it has been notified of the possibility of such damages. Ampro reserves the right to revise this publication from time to time without obligation to notify any person of such revisions. If errors are found, please contact Ampro at the address listed below on the Notice page of this document.

TRADEMARKS

The Ampro logo is a registered trademark, and CoreModule, EnCore, Little Board, and MiniModule are trademarks of Ampro Computers, Inc. All other marks are the property of their respective companies.

REVISION HISTORY

Revision	Reason for Change	Date
A	Initial Release	Oct/03

Ampro Computers, Incorporated
5215 Hellyer Avenue
San Jose, CA 95138-1007
Tel. 408 360-0200
Fax 408 360-0222
www.ampro.com

Audience Assumptions

This reference manual is for the person who designs computer related equipment, including but not limited to hardware and software design and implementation of the same. Ampro Computers, Inc. assumes you are qualified in designing and implementing your hardware designs and its related software into your prototype computer equipment.

Contents

Figures

Tables

Chapter 1 About This Manual

Purpose of this Manual

This manual is for designers of systems based on the CoreModule™ 410 PC/104 CPU module. This manual contains information that permits designers to create an embedded system based on specific design requirements.

Information provided in this reference manual includes:

- CoreModule 410 CPU Specifications
- Environmental requirements
- Major chips and features implemented
- CoreModule 410 CPU Module connector/pin numbers and definition
- BIOS Setup information

Information not provided in this reference manual includes:

- Detailed chip specifications
- Internal component operation
- Internal registers or signal operations
- Bus or signal timing for industry standard busses and signals

Reference Material

The following list of reference materials may be helpful for you to complete your custom design successfully. Most of this reference material is also available on the Ampro web site in the Embedded Design Resource Center. The Embedded Design Resource Center was created for embedded system developers to share Ampro's knowledge, insight, and expertise gained from years of experience.

Specifications

- PC/104 Specifications Revision 2.4, August 2001.

  For latest revision of the PC/104 specifications, contact the PC/104 Consortium, at:

  Web site: http://www.pc104.org

Chip specifications used in the CoreModule 410 CPU module:

- STMicroelectronics and the chip, STPC® Elite, used for the embedded CPU

  Web site: http://us.st.com/stonline/products/support/stpc/home.htm

- Standard Microsystems Corp and the chip, FDC37B782, used for the Super I/O controller

  Web site: http://www.smsc.com/main/datasheets/37b78x.pdf

Related Ampro Products

The following items are directly related to successfully using the Ampro product you have just purchased or plan to purchase. Ampro highly recommends that you purchase and utilize a CoreModule 410 QuickStart Kit or Development System simultaneously with the design of your product.

CoreModule 410 Support Products

- CoreModule 410 QuickStart Kit (QSK)

  The CoreModule 410 QuickStart Kit includes the CoreModule 410 CPU, a complete cable kit, the CoreModule 410 QuickStart Guide, and drivers for any Ampro supported operating systems with unique devices used on the board. The release notes, drivers, and documentation in PDF format are provided on the CoreModule 410 Documentation & Support Software CD-ROM.

- CoreModule 410 Development System

  The CoreModule 410 Development System is a benchtop system, which provides a "known good" environment for your development work. The Development System provides an integrated and easy-to-use self-hosted development environment (power supply and floppy, disk and CD-ROM drives) that lets you maximize the benefit of using off-the-shelf PC-compatible modules as the basis of your embedded system design. You can install PC/104 expansion modules or ISA bus expansion boards on the Development System chassis. The Development System is laid out to make all the components of your system accessible. Refer to the CoreModule 410 Development System Users Guide on the CoreModule 410 Documentation and Support Software (Doc & SW) CD-ROM for more information.

- CoreModule 410 Documentation and Support Software CD-ROM

  The CoreModule 410 Documentation and Support Software (Doc & SW) CD-ROM is provided with the CoreModule 410 QuickStart Kit and the Development System. The CD-ROM includes all of the CoreModule 410 documentation, including this Reference Manual, the CoreModule 410 QuickStart Guide, the CoreModule 410 Development System Users Guide in PDF format, Release notes, software utilities, and drivers.

Other CoreModule Products

- CoreModule™ 400 – This PC/104 embedded single board computer (SBC) is a state-of-the-art, high-integration x86-based computer using STMicroelectronics' new 133MHz STPC Atlas processor, which provides a complete embedded PC solution with all of the standard peripheral interfaces. In addition to the standard CoreModule features (PC/104 form factor, PC/104 bus, +5 volt power, etc.), it includes 64MB on-board SDRAM memory, Ethernet, USB, video, watchdog timer, serial console, BIOS extensions for OEM boot customization, and Advanced Power Management. The CoreModule 400 also offers two types of solid-state disk interfaces, with a Bytewide socket supporting DiskOnChip 2000 devices, as well as, a Type II Socket supporting CompactFlash devices.

- CoreModule™ 600 – This PC/104-Plus embedded single board computer (SBC) is a compact, rugged, high integration, ultra low power 400MHz ULV Celeron processor with 256kB of internal cache, and all of the standard peripheral interfaces. In addition to the standard CoreModule features (PC/104 form factor, PC/104-Plus, +5 volt power, etc.), the CoreModule 600 includes 10/100BaseT Ethernet, AGP 4X video with 32MB video memory for CRT, TFT and standard LCD flat panels, USB ports, RS232C/RS485 serial ports, and an onboard Type II CompactFlash socket, which supports up to 1GB or more of flash memory. The CoreModule 600 also supports a watchdog timer, serial console, battery-less boot, BIOS extensions for OEM boot customization, some power management features and up to 256MB of SDRAM memory.

Other Ampro Products

- Little Board™ Family – These high-performance, highly integrated single board computers use the EBX form factor (5.75x8.00 inches), and are available with Pentium III and Celeron processors. The EBX-compliant Little Board single board computers offer functions equivalent to a complete laptop or desktop PC system, plus several expansion cards. Built-in extras to meet the critical requirements of embedded applications include onboard solid state disk capability, Watchdog timer, smart power monitor, and other embedded-PC BIOS enhancements.
- MiniModule™ Family – This extensive line of peripheral interface modules, compliant with PC/104 and PC/104-Plus standard, can be used with Ampro's CoreModule and Little Board single board computers to configure embedded system solutions. Ampro's highly reliable MiniModule products currently support USB 2.0, IEEE 1394 (FireWire), CRT and flat panel display interfaces, Ethernet, PC Card expansion, analog/data acquisition, FPGA, additional RS232/RS485 serial ports, and general-purpose I/O (GPIO).
- EnCore™ Family – These high-performance, compact, modular CPU solutions use various processor technologies including Intel x86, MIPS, and PowerPC architectures to plug into your custom logic board. Each EnCore module provides standard peripherals, including IDE, floppy drive interface, PCI bus, serial, parallel, PS/2 keyboard and mouse interfaces, 10/100BaseT Ethernet, and USB ports. Some EnCore modules also provide video and AC97 sound. Depending on the model, EnCore modules can hold between 16MB and 512MB of SODIMM SDRAM memory.

Chapter 2 Product Overview

This introduction presents general information about the PC/104 architecture and the CoreModule 410 PC/104 CPU Module. After reading this chapter you should understand:

- PC/104 Concept
- CoreModule 410 architecture
- CoreModule 410 features
- Major components
- Connectors
- Specifications

PC/104 Architecture

The PC/104 architecture affords a great deal of flexibility in system design. You can build a simple system using only a CoreModule 410, input/output devices connected to the serial or parallel ports, and a solid state disk drive in the bytewide socket. To expand a simple CoreModule system, simply add self-stacking Ampro MiniModules expansion boards to provide additional capabilities, such as:

- Additional serial and parallel ports
- Analog or digital I/O
- PCMCIA interfaces
- FPGA (Field Programmable Gate Array) logic devices

PC/104 expansion modules can be stacked with the CoreModule 410 avoiding the need for large, expensive card cages and backplanes. The PC/104 expansion modules can be mounted directly to the PC/104 bus connector of the CoreModule 410. PC/104-compliant modules can be stacked with an inter-board spacing of ~0.66 inches so that a 3-module system fits in a 3.6 inch by 3.8 inch by 2.4 inch space. See Figure 2-1.

One or more MiniModule products or other PC/104 modules can be installed on the CoreModule expansion connectors, so that the expansion modules fit within the CoreModule outline dimensions. Most MiniModule products have stack through connectors compatible with the PC/104 Version 2.1 specification. Several modules can be stacked on the CoreModule headers. Each additional module increases the thickness of the package by 15mm (0.66”). See Figure 2-1.

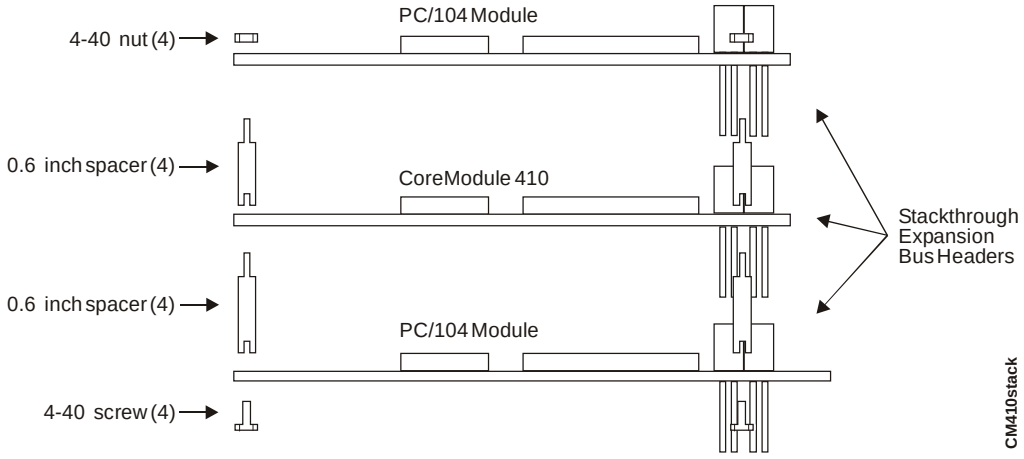

Figure 2-1. Stacking PC/104 Modules with the CoreModule 410

Product Description

The CoreModule 410 CPU is a high integration, x86-based PC/AT compatible system in the PC/104 form factor. This rugged and high quality single board system contains most of the component subsystems of a PC/AT motherboard.

In addition, the CoreModule 410 CPU includes a comprehensive set of system extensions and enhancements that are specifically designed for embedded systems. These enhancements ensure fail-safe embedded system operation, such as, a watchdog timer. It is designed to meet the size, power consumption, temperature range, quality, and reliability demands of embedded applications. The CoreModule 410 requires a single +5V power source.

The CoreModule 410 CPU is particularly well suited to either embedded or portable applications. Its flexibility makes system design quick and easy. It can be stacked with any Ampro MiniModule, any PC/104-compliant expansion boards, or it can be used as the computing engine in a fully customized application.

Module Features

- CPU
  - Uses 133MHz x86 STPC Elite microprocessor
  - Fully PC compatible architecture
  - 8KB Unified Instruction and Data Cache
  - Parallel Processing Integrated Floating Point Unit
  - Low Power and System Management Modes
- Memory
  - 16MB standard SDRAM (soldered on the board)
  - 100MHz Clock Speed
  - 32-pin bytewide memory socket
    - Supports a DiskOnChip® device
  - 1MB of flash memory (with protected records for setup data)
    - Up to 768kB available for OEM use
- PC/104 Bus Interface
  - Supports 16-bit ISA Bus
  - Clock speeds up to 8MHz
- IDE Interface
  - Supports two enhanced IDE devices
  - Supports single master mode
  - Fast ATA-capable interface supports high-speed PIO modes (PIO modes 0 to 4 and DMA modes 0 to 2)
  - Supports ATAPI and DVD peripherals
  - Supports IDE native and ATA compatibility modes
- Floppy Disk Controller
  - Supports two floppy drives
  - Supports all standard PC/AT formats: 360KB, 1.2MB, 720KB, 1.44MB, 2.88MB

- Serial Ports
  - Two buffered RS232 serial ports with full handshaking and modem capability
  - Provides 16550-equivalent controllers, each with a built-in 16-byte FIFO buffer
  - Port 2 supports RS232 or RS485 operation (Jumper selectable)
  - Supports programmable word length, stop bits, and parity
  - Supports 16-bit programmable baud-rate generator and an interrupt generator
- Parallel Port
  - Supports standard printer port
  - Supports IEEE 1284 standard protocols, and EPP, ECP outputs
  - Bidirectional data lines
  - Supports 16 byte FIFO for ECP mode
- Utility Interface
  - Keyboard and PS/2 Mouse Interface
  - Supports external battery for Real Time Clock operation
  - Supports standard speaker interface with 0.1 Watt output
  - Supports a Reset switch
- Miscellaneous
  - Battery-backed real-time clock and CMOS RAM, with support for battery-free operation
  - Watchdog Timer
  - Serial Console (or Console Redirection)
  - GPIO lines
  - Customer configured Splash screen

Block Diagram

Figure 2-2 provides the functional components of the CoreModule 410.

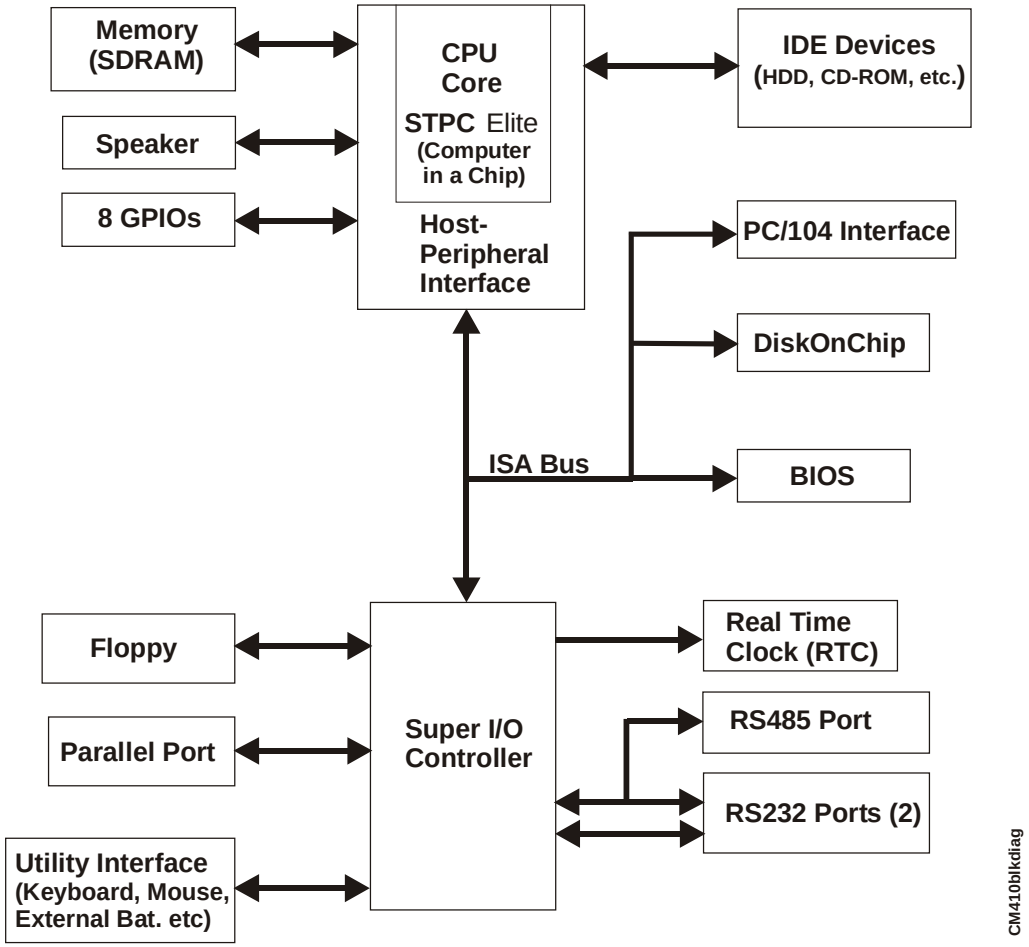

Figure 2-2. CoreModule 410 Block Diagram

Major Integrated Circuits (ICs)

Table 2-1 lists the major integrated circuits, including a brief description of each, on the CoreModule 410 and Figure 2-4 shows the location of the major chips.

Table 2-1. Major Integrated Circuit Descriptions and Function

Chip Type	Mfg.	Model	Description	Function
CPU (U1)	STMicro-electronics	STPC ELITE	Embedded CPU (U1) – The combination of features in the CPU provide more than just a processor. It also provides a PCI controller, EIDE controller, some I/O features, and power management capabilities.	Embedded CPU
Super I/O Controller (U8)	Standard Microsystems Corp.	FDC37B782	Super I/O (U8) – This chip provides the Floppy controller, Serial ports, and Parallel port.	I/O features

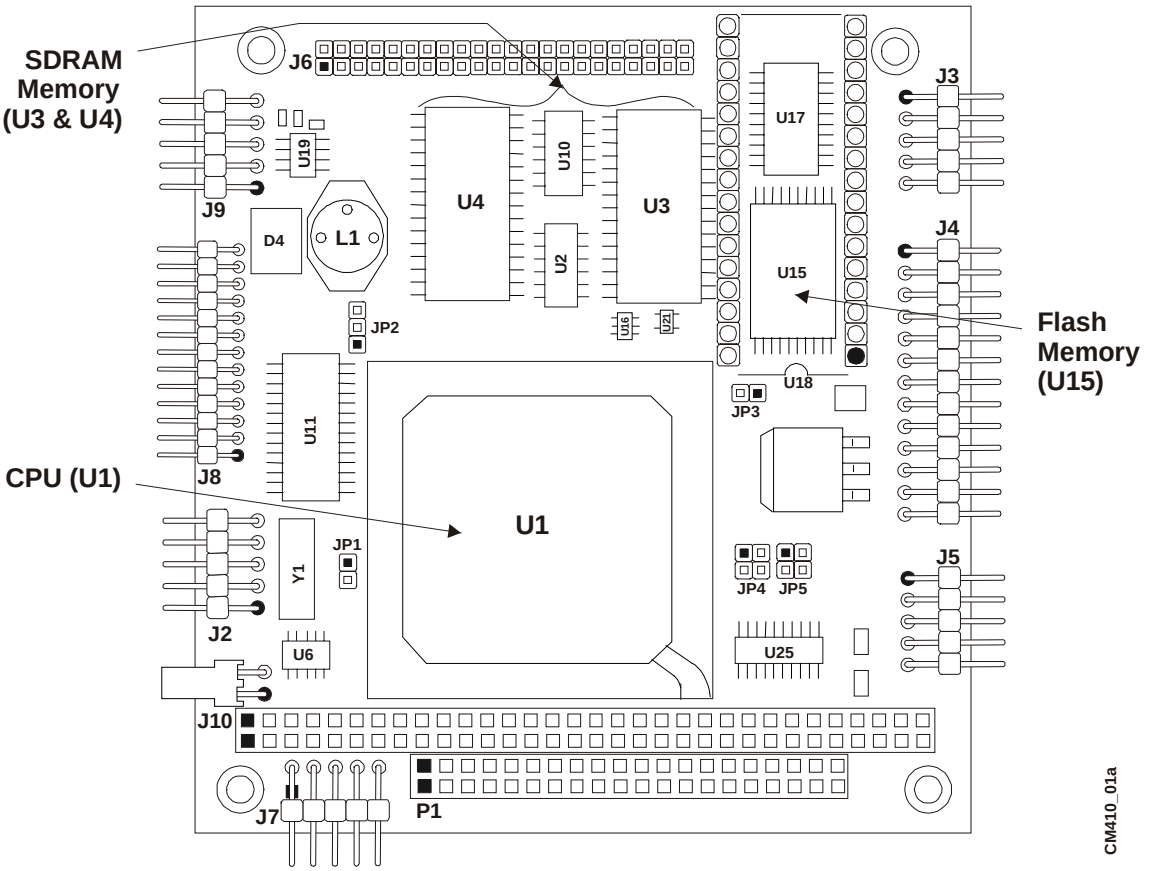

Figure 2-3. CoreModule 410 (Top View)

NOTE	Pin 1 is shown as a black square or a round black pin in connectors and jumpers in all illustrations.

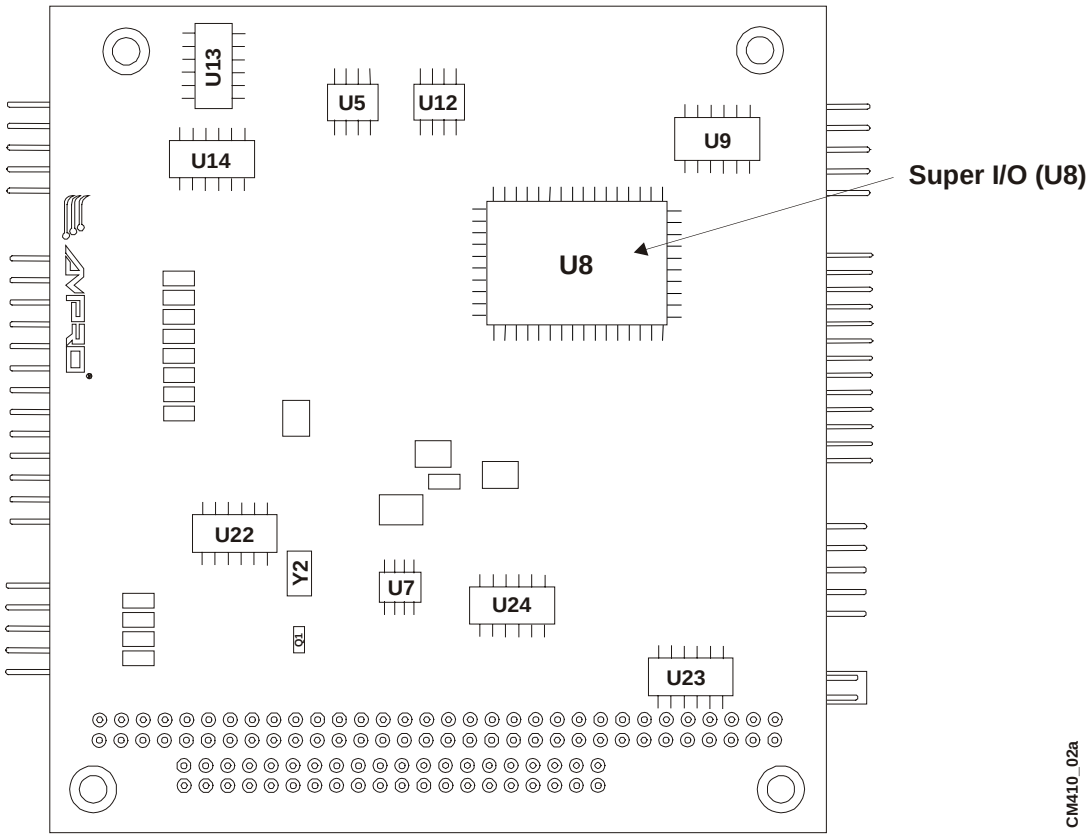

Figure 2-4. CoreModule 410 (Bottom View)

NOTE	The CoreModule 410 is in violation of the PC/104 height limitation in one place on the bottom of the board. The Super I/O chip (U8) is 0.037 inches high, exceeding the allowed height limitation by 0.025 inches. However, in future revisions to the PC/104 specifications, this height limitation may change. See Figure 2-4.

Connectors, Jumpers, and LEDs

Connector Definitions

Table 2-2 describes the connectors shown in Figures 2-4 to 2-5. All I/O connectors use 0.1" pin spacing unless otherwise indicated.

Table 2-2. Module Connector Descriptions

Jack/Plug #	Board Access	Description
P1A, P1B, P1C, & P1D – PC/104	Top/ Bottom	104-pin connector used for PC/104 bus connector
J2 – GPIO signals	Top	10-pin connector used for User GPIO signals
J3 – Serial 1 (COM1)	Top	10-pin connector used for Serial interface (Com 1)
J4 – Parallel	Top	26-pin connector provides the Parallel interface
J5 – Utility	Top	10-pin connector used for the Utility interface
J6 – IDE	Top	44-pin 2mm connector used for the IDE interface
J7 – Power	Top	10-pin connector used for the Power connection
J8 – Floppy	Top	26-pin 2mm connector used for the Floppy interface
J9 – Serial 2 (COM2)	Top	10-pin connector used for Serial interface (Com 2)
J10 – RS485	Top	2-pin connector used for RS485 interface

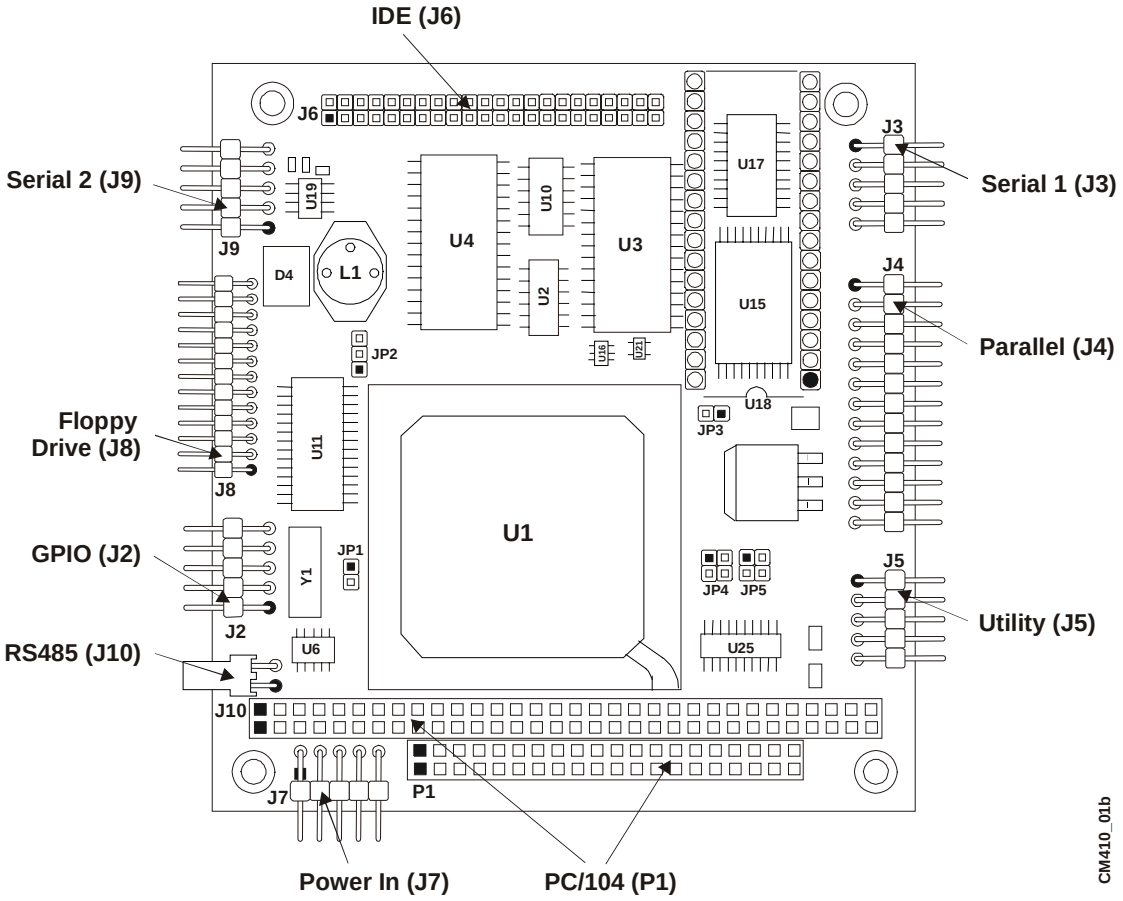

Figure 2-5. Connector Locations (Top View)

Jumper Definitions

Table 2-3 describes the jumpers shown in Figure 2-6.

Table 2-3. Jumper Settings

Jumper #	Installed	Removed
JP1 COM 2 RS485 (Serial 2) Termination	Enable RS485 Termination (Pins 1-2)	Disable RS485 Termination (No jumper) **Default** setting
JP2 RS232/RS485 Selection	Enable RS-485 (1-2)	Enable RS-232 (2-3) **Default** setting
JP3 DiskOnChip Boot Address Select	Access from DC000h-DDFFFh (Pins 1-2) **Default** setting	Access from CC000h-CDFFFh (No jumper)
JP4 & JP5 BIOS/DOC Select JP4 JP5 1 3 1 3 2 4 2 4 **BIOS/DOC Select Jumper Setting (Shown in Enable DOC (Default))**	Enable Internal BIOS – Normal operation, (JP4 & JP5, Pins 1-3)	Disabled – Won't Boot (Don't remove)
	Enable External BIOS – Used for recovery (JP4 & JP5, Pins 1-2)	Disabled (See other positions)
	Enable DOC – Boot from DiskOnChip in bytewide socket (JP4 & JP5, Pins 1-3 & 2-4) **Default** setting	Disabled (See other positions)

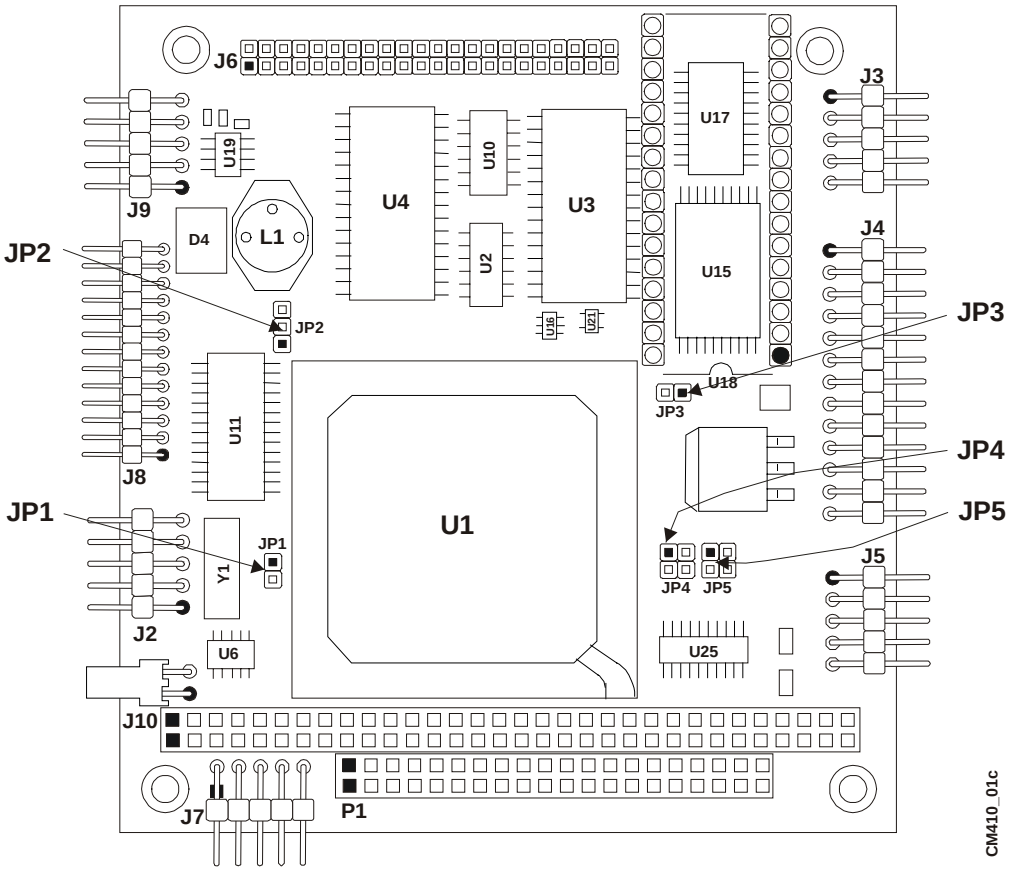

Figure 2-6. Jumper Locations (Top View)

Specifications

Physical Specifications

Table 2-4 gives the physical dimensions of the module and Figure 2-7 gives the mounting dimensions.

Table 2-4. Weight and Footprint Dimensions

Item	Dimension
Weight	0.077kg. (0.17lbs.)
Height (overall) *	23.49 mm (0.925”)
Width	90.2 mm (3.6”)
Length	95.9 mm (3.8”)

Notes: *See Note on page 10 regarding the exceeded height limitation.

Mechanical Specifications

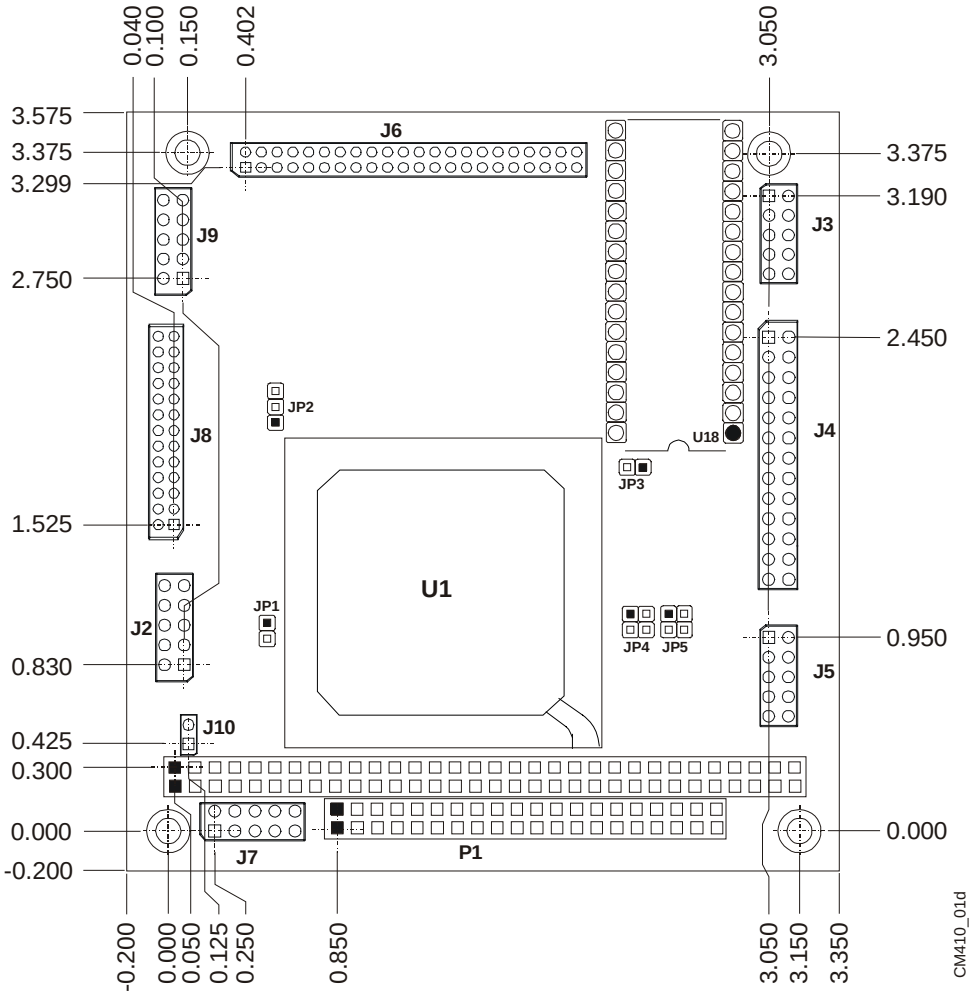

Figure 2-7. Mechanical Dimensions (Top View)

NOTE	All dimensions are given in inches. Pin 1 is shown as a black square, white square, or black round pin in connectors and jumpers in this illustration.

Power Specifications

Table 2-5 shows the power requirements.

Table 2-5. Power Supply Requirements

Parameter	Characteristics
Input Type	Regulated DC voltages
Input Power Requirements	+5 VDC +/- 5% @ 0.700 Amps*
Operating Power	3.5 Watts Continuous

Note: * This current value is determined without any other boards (video, etc.) connected to the CoreModule 410.

Environmental Specifications

Table 2-6 provides the most efficient operating and storage condition ranges required for this module.

Table 2-6. Environmental Requirements

Parameter	Conditions
Temperature	
Operating	+0° to +70° C (32° to 158° F)
Extended (Optional)	–40° to +85° C (–40° F to +185° F)
Storage	–55° to +85° C (–67° F to +185° F)
Humidity	
Operating	20% to 80% relative humidity, non-condensing
Non-operating	5% to 95% relative humidity, non-condensing

Thermal/Cooling Requirements

The CPU does not require a heatsink.

Chapter 3 Hardware Description

Overview

This chapter discusses the chips and connectors of the module features in the following order:

- CPU (U1)
- Memory
  - SDRAM (U3, U4)
  - Flash Memory (U15)
  - Bytewide socket (U18)
- PC/104 (P1A, B, C, D)
- IDE (J6)
- Floppy (J8)
- Serial (J3, J9)
- Parallel (J4)
- Utility (J5)
  - Keyboard
  - Mouse
  - Battery
  - Reset Switch
  - Speaker
- Miscellaneous
  - Time of Day/RTC
  - GPIO signals (J2)
  - Oops! Jumper
  - Serial Console
  - Watchdog timer
- Power (J7)

NOTE	Ampro Computers, Inc. only supports the features/options tested and listed in this manual. The main integrated circuits (chips) used in the CoreModule 410 may provide more features or options than are listed for the CoreModule 410, but some of these chip features/options are not supported on the board and will not function as specified in the chip documentation.

CPU (U1)

The CoreModule 410 uses an embedded microprocessor operating at 133MHz, that combines a powerful x86 core and a selection of peripheral interfaces into one chip. The STPC Elite integrates a standard 5th generation x86 core. It supports logic including PC/104, UIDE controllers and combines these with standard I/O interfaces to provide a PC compatible subsystem in a single chip.

Memory

The CoreModule 410 memory consists of the following elements:

- SDRAM
- Flash memory
- Bytewide socket

SDRAM Memory (U3, U4)

The CoreModule 410 contains two 16-bit SDRAM chips of 8MB each for a total of 16MB memory soldered into place on the module and operating at 100MHz.

Flash Memory (U15)

The CoreModule 410 uses an on-board 1MB flash device where the upper 256kB region is used for system BIOS. The remaining lower 768kB region is available for OEM use, such as splash screen.

A feature of the CoreModule 410 makes it possible to load an application in the available Flash memory and boot directly to this application. This feature implements a method of copying an application from the Flash device to a memory location . Refer to Chapter 4, *BIOS Setup*, the Release notes, and the CoreModule 410 Doc & SW CD-ROM for more information.

Bytewide Socket (U18)

The CoreModule 410 has a 32-pin DIP socket on the module used as a bytewide memory socket. This socket supports Re-Writeable Flash chips, Serial EEPROMs (SEEPs), and DiskOnChip devices, such as, DOC2000 (M-Systems DiskOnChip©). A memory device installed in the bytewide socket can be used for external BIOS or for BIOS recovery.

Memory Map

Table 3-1. Memory Map

Address	Size	Use
1 0000 0000	256KB	Flash ROM (BIOS)
FFFC 0000	4,155,482,101B	Unused
084C 600B	12B	STPC Memory Mapped registers
084C 6000	103,192KB	Unused
0200 0000	16MB	– RAM, on 32MB version – Unused, on 16MB version

Memory Map Table continued on next page

Table 3-1. Memory Map (continued)

Address	Size	Use				
		Memory hole size selected				
		8MB	4MB	2MB	1MB	0MB
0100 0000	1MB	H	H	H	H	R
00F0 0000	1MB	H	H	H	R	R
00E0 0000	1MB	H	H	R	R	R
00D0 0000	1MB	H	H	R	R	R
00C0 0000	1MB	H	R	R	R	R
00B0 0000	1MB	H	R	R	R	R
00A0 0000	1MB	H	R	R	R	R
0090 0000	1MB	H	R	R	R	R
		R = RAM H = Memory Hole, forwarded to ISA The board can be configured to have access to the 1MB Flash anywhere in the memory hole, on 1MB alignment.				
0080 0000	7MB	RAM				
0010 0000	128KB	Shadowed BIOS				
000E 0000	8KB	Unused				
000D E000	8KB	DiskOnChip, if DC000-DDFFF window selected. Unused if no DOC present.				
000D C000	56KB	Unused				
000C E000	8KB	DiskOnChip, if CC000-CDFFF window selected. Unused if no DOC present.				
000C C000	48KB	Unused				
000C 0000	128KB	– Unused, reserved for Video RAM – or in SMI mode, mapped to RAM				
000A 0000 - 0000 0000	640KB	Base memory				

Interrupt Channel Assignments

The channel interrupt assignments are shown in Table 3-2.

Table 3-2. Interrupt Channel Assignments

Device vs IRQ No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Disable
Timer	X																
Keyboard		X															
Secondary Cascade			X														
COM1				O	D	O	O	O		O	O	O	O				Z
COM2				D	O	O	O	O		O	O	O	O				Z
Floppy							D										Z
Parallel				O	O	O	O	D		O	O	O	O				Z
RTC									X								
IDE															D		Z
Math Coprocessor														X			
PS/2 Mouse				O	O	O	O	O		O	O	O	D				Z

Legend: D = Default, O = Optional, X = Fixed, Z = Disable option

NOTE	The devices listed with a "Z" in the Disable column indicate the device can be disabled, which will free the IRQ for another device in the list.

Table 3-3. DMA Map

DMA #	Use
0	LPT 1, only in ECP mode (Configurable for 0-3 or disabled)
1, 3, 5, 6, 7	
2	Floppy (can be disabled)
4	DMA 1 cascade

I/O Address Map

Table 3-4 shows the I/O address map.

Table 3-4. I/O Address Map

Address (hex)	Subsystem
0000-000F	Primary DMA Controller (#1)
0020-0021	Master Interrupt Controller (#1)
0022-0023	STPC Configuration
0040-0043	Programmable Interrupt Timer (Clock/Timer)
0060	Keyboard Controller
0061	ISA Standard Port B
0063	ISA Standard Port B alias

0064	Keyboard Controller
0065	ISA Standard Port B alias
0067	ISA Standard Port B alias
0069	ISA Standard Port B alias
006B	ISA Standard Port B alias
006D	ISA Standard Port B alias
006F	ISA Standard Port B alias
0070-0071	RTC/ NMI enable
0080-008F	DMA Page
00A0-00A1	Slave Interrupt Controller (#2)
00C0-00DF	Secondary DMA Controller (#2)
01F0-01F7	IDE 0 (can be disabled)
0201	Watchdog trigger (disabled by default)
02F8-02FF	Serial Port 2 (COM2) (base configuration @ 3F8h/**2F8h**/3E8h/2E8h/7F8h/7E8h/disabled)
0370-0371	Super I/O Configuration
0378-037B	LPT 1 (base configuration @ 378h/278h/3BCh/disabled; (3BCh does not support EPP mode)
0378-037F	LPT 1 (only in EPP modes, with default base address)
03F0-03F5	Floppy Disk Controller (can be disabled)
03F6	IDE 0 (see 1F0)
03F7	Floppy Disk Controller (see 3F0)
03F8-03FF	Serial Port 1 (COM1) (base configuration @ **3F8h**/2F8h/3E8h/2E8h/7F8h/ 7E8h/disabled)
0778-077A	LPT 1 (only in ECP modes, with default base address)
0CF8	PCI Configuration Address
0CFC-0CFF	PCI Configuration Data
E000-E007	General Purpose I/O for customer use
E400-E407	Board control
E800-E80F	IDE Bus Master registers
EC00	IDE 1 Control (IDE 1 not routed to a connector)
F000-F007	IDE 1 Command (IDE 1 not routed to a connector)

PC/104 Bus Interface (P1A,B,C,D)

The PC/104 Bus uses a 104-pin 100mm header interface. This interface header will carry all of the appropriate PC/104 signals operating at clock speeds up to 8MHz. The interface header is located on the both the top and bottom of the module.

Table 3-3. PC/104 Bus Interface Pin/Signal Descriptions (P1A)

Pin #	Signal	Description (P1 Row A)
1 (A1)	IOCHK*	I/O Channel Check – This signal may be activated by ISA boards to request that a non-maskable interrupt (NMI) be generated to the system processor. It is driven active to indicate an uncorrectable error has been detected.
2 (A2)	SD7	System Data 7 – This signal (0 to 19) provides a system data bit.
3 (A3)	SD6	System Data 6 – Refer to SD7, pin A2, for more information.
4 (A4)	SD5	System Data 5 – Refer to SD7, pin A2, for more information.
5 (A5)	SD4	System Data 4 – Refer to SD7, pin A2, for more information.
6 (A6)	SD3	System Data 3 – Refer to SD7, pin A2, for more information.
7 (A7)	SD2	System Data 2 – Refer to SD7, pin A2, for more information.
8 (A8)	SD1	System Data 1 – Refer to SD7, pin A2, for more information.
9 (A9)	SD0	System Data 0 – Refer to SD7, pin A2, for more information.
10 (A10)	IOCHRDY	I/O Channel Ready – This signal allows slower ISA boards to lengthen I/O or memory cycles by inserting wait states. This signal's normal state is active high (ready). ISA boards drive the signal inactive low (not ready) to insert wait states. Devices using this signal to insert wait states should drive it low immediately after detecting a valid address decode and an active read, or write command. The signal is released high when the device is ready to complete the cycle.
11 (A11)	AEN	Address Enable – This signal is used to degate the system processor and other devices from the bus during DMA transfers. When this signal is active, the system DMA controller has control of the address, data, and read/write signals. This signal should be included as part of ISA board select decodes to prevent incorrect board selects during DMA cycles.
12 (A12)	SA19	System Address 19 – This signal (0 to 19) provides a system address bit.
13 (A13)	SA18	System Address 18 – Refer to SA19, pin A12, for more information.
14 (A14)	SA17	System Address 17 – Refer to SA19, pin A12, for more information.
15 (A15)	SA16	System Address 16 – Refer to SA19, pin A12, for more information.
16 (A16)	SA15	System Address 15 – Refer to SA19, pin A12, for more information.
17 (A17)	SA14	System Address 14 – Refer to SA19, pin A12, for more information.
18 (A18)	SA13	System Address 13 – Refer to SA19, pin A12, for more information.
19 (A19)	SA12	System Address 12– Refer to SA19, pin A12, for more information.
20 (A20)	SA11	System Address 11 – Refer to SA19, pin A12, for more information.
21 (A21)	SA10	System Address 10 – Refer to SA19, pin A12, for more information.
22 (A22)	SA9	System Address 9 – Refer to SA19, pin A12, for more information.
23 (A23)	SA8	System Address 8 – Refer to SA19, pin A12, for more information.
24 (A24)	SA7	System Address 7 – Refer to SA19, pin A12, for more information.

Pin #	Signal	Description (P1 Row A)
25 (A25)	SA6	System Address 6 – Refer to SA19, pin A12, for more information.
26 (A26)	SA5	System Address 5 – Refer to SA19, pin A12, for more information.
27 (A27)	SA4	System Address 4 – Refer to SA19, pin A12, for more information.
28 (A28)	SA3	System Address 3 – Refer to SA19, pin A12, for more information.
29 (A29)	SA2	System Address 2 – Refer to SA19, pin A12, for more information.
30 (A30)	SA1	System Address 1 – Refer to SA19, pin A12, for more information.
31 (A31)	SA0	System Address 0 – Refer to SA19, pin A12, for more information.
32 (A32)	GND	Ground

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Table 3-4. PC/104 Bus Interface Pin/Signal Descriptions (P1B)

Pin #	Signal	Description (P1 Row B)
33 (B1)	GND	Ground
34 (B2)	RESETDRV	Reset Drive – This signal is used to reset or initialize system logic on power up or subsequent system reset.
35 (B3)	+5V	+5V power +/- 10%
36 (B4)	IRQ9	Interrupt request 9 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
37 (B5)	NC (-5V)	Not connected (-5 volts)
38 (B6)	DRQ2	DMA Request 2 – Used by I/O resources to request DMA service, or to request ownership of the bus as a bus master device. Must be held high until associated DACK2 line is active.
39 (B7)	NC (-12V)	Not connected (-12 volts)
40 (B8)	ZWS*	Zero Wait State – This signal is driven low by a bus slave device to indicate it is capable of performing a bus cycle without inserting any additional wait states. To perform a 16-bit memory cycle without wait states, this signal is derived from an address decode.
41 (B9)	+12V	+12 Volts
42 (B10)	NC	Not connected (Key Pin)
43 (B11)	SMEMW*	System Memory Write – This signal is used by bus owner to request a memory device to store data currently on the data bus and only active for the lower 1MB. Used for legacy compatibility with 8-bit cards.
44 (B12)	SMEMR*	System Memory Read – This signal is used by bus owner to request a memory device to drive data onto the data bus and only active for lower 1MB. Used for legacy compatibility with 8-bit cards.
45 (B13)	IOW*	I/O Write – This strobe signal is driven by the owner of the bus (ISA bus master or DMA controller) and instructs the selected I/O device to capture the write data on the data bus.
46 (B14)	IOR*	I/O Read – This strobe signal is driven by the owner of the bus (ISA bus master or DMA controller) and instructs the selected I/O device to drive read data onto the data bus.
47 (B15)	DACK3*	DMA Acknowledge 3 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.

Pin #	Signal	Description (P1 Row B)
48 (B16)	DRQ3	DMA Request 3 – Used by I/O resources to request DMA service. Must be held high until associated DACK3 line is active.
49 (B17)	DACK1*	DMA Acknowledge 1 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.
50 (B18)	DRQ1	DMA Request 1 – Used by I/O resources to request DMA service. Must be held high until associated DACK1 line is active.
51 (B19)	REFRESH*	Memory Refresh – This signal is driven low to indicate a memory refresh cycle is in progress. Memory is refreshed every 15.6 usec.
52 (B20)	SYSCLK	System Clock – This is a free running clock typically in the 8MHZ to 10MHz range, although its exact frequency is not guaranteed.
53 (B21)	IRQ7	Interrupt Request 7 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
54 (B22)	IRQ6	Interrupt Request 6 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
55 (B23)	IRQ5	Interrupt Request 5 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
56 (B24)	IRQ4	Interrupt Request 4 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
57 (B25)	IRQ3	Interrupt Request 3 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
58 (B26)	DACK2*	DMA Acknowledge 2 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.
59 (B27)	TC	Terminal Count – This signal is a pulse to indicate a terminal count has been reached on a DMA channel operation.
60 (B28)	BALE	Buffered Address Latch Enable – This signal is used to latch the LA23 to LA17 signals or decodes of these signals. Addresses are latched on the falling edge of BALE. It is forced high during DMA cycles. When used with AENx, it indicates a valid processor or DMA address.
61 (B29)	+5V	+5V power +/- 10%
62 (B30)	OSC	Oscillator – This clock signal operates at 14.3MHz. This signal is not synchronous with the system clock (SYSCLK).
63 (B31)	GND	Ground
64 (B32)	GND	Ground

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Table 3-5. PC/104 Bus Interface Pin/Signal Descriptions (P1C)

Pin #	Signal	Description (P1 Row C)
1 (C0)	GND	Ground
2 (C1)	SBHE*	System Byte High Enable – This signal is driven low to indicate a transfer of data on the high half of the data bus (D15 to D8).
3 (C2)	LA23	Lactchable Address 23 – This signal must be latched by the resource if the line is required for the entire data cycle.
4 (C3)	LA22	Lactchable Address 22 – Refer to LA23, pin C2, for more information.

Pin #	Signal	Description (P1 Row C)
5 (C4)	LA21	Lactchable Address 21 – Refer to LA23, pin C2, for more information.
6 (C5)	LA20	Lactchable Address 20 – Refer to LA23, pin C2, for more information.
7 (C6)	LA19	Lactchable Address 19 – Refer to LA23, pin C2, for more information.
8 (C7)	LA18	Lactchable Address 18 – Refer to LA23, pin C2, for more information.
9 (C8)	LA17	Lactchable Address 17 – Refer to LA23, pin C2, for more information.
10 (C9)	MEMR*	Memory Read – This signal instructs a selected memory device to drive data onto the data bus. It is active on all memory read cycles.
11 (C10)	MEMW*	Memory Write – This signal instructs a selected memory device to store data currently on the data bus. It is active on all memory write cycles.
12 (C11)	SD8	System Data 8 – Refer to SD7, pin A2, for more information.
13 (C12)	SD9	System Data 9 – Refer to SD7, pin A2, for more information.
14 (C13)	SD10	System Data 10 – Refer to SD7, pin A2, for more information.
15 (C14)	SD11	System Data 11 – Refer to SD7, pin A2, for more information.
16 (C15)	SD12	System Data 12 – Refer to SD7, pin A2, for more information.
17 (C16)	SD13	System Data 13 – Refer to SD7, pin A2, for more information.
18 (C17)	SD14	System Data 14 – Refer to SD7, pin A2, for more information.
19 (C18)	SD15	System Data 15 – Refer to SD7, pin A2, for more information.
20 (C19)	NC	Key Pin

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Table 3-6. PC/104 Bus Interface Pin/Signal Descriptions (P1D)

Pin #	Signal	Description (P1 Row D)
21 (D0)	GND	Ground
22 (D1)	MEMCS16*	Memory Chip Select 16 – This is signal is driven low by a memory slave device to indicates it is cable of performing a 16-bit memory data transfer. This signal is driven from a decode of the LA23 to LA17 address lines.
23 (D2)	IOCS16*	I/O Chip Select 16 – This signal is driven low by an I/O slave device to indicate it is capable of performing a 16-bit I/O data transfer. This signal is driven from a decode of the SA15 to SA0 address lines.
24 (D3)	IRQ10	Interrupt Request 10 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
25 (D4)	IRQ11	Interrupt Request 11 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
26 (D5)	IRQ12	Interrupt Request 12 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
27 (D6)	IRQ15	Interrupt Request 15 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
28 (D7)	IRQ14	Interrupt Request 14 – Asserted by a device when it has pending interrupt request. Only one device may use the request line at a time.
29 (D8)	DACK0*	DMA Acknowledge 0 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.

Pin #	Signal	Description (P1 Row D)
30 (D9)	DRQ0	DMA Request 0 – Used by I/O resources to request DMA service. Must be held high until associated DACK0 line is active.
31 (D10)	DACK5*	DMA Acknowledge 5 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.
32 (D11)	DRQ5	DMA Request 5 – Used by I/O resources to request DMA service. Must be held high until associated DACK5 line is active.
33 (D12)	DACK6*	DMA Acknowledge 6 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.
34 (D13)	DRQ6	DMA Request 6 – Used by I/O resources to request DMA service. Must be held high until associated DACK6 line is active.
35 (D14)	DACK7*	DMA Acknowledge 7 – Used by DMA controller to select the I/O resource requesting the bus, or to request ownership of the bus as a bus master device. Can also be used by the ISA bus master to gain control of the bus from the DMA controller.
36 (D15)	DRQ7	DMA Request 7 – Used by I/O resources to request DMA service. Must be held high until associated DACK7 line is active.
37 (D16)	+5V	+5V Power +/- 10%
38 (D17)	MASTER*	Bus Master Assert – This signal is used by an ISA board along with a DRQ line to gain ownership of the ISA bus. Upon receiving a -DACK a device can pull -MASTER low which will allow it to control the system address, data, and control lines. After -MASTER is low, the device should wait one CLK period before driving the address and data lines, and two clock periods before issuing a read or write command.
39 (D18)	GND	Ground
40 (D19)	GND	Ground

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

IDE Interface (J6)

The IDE device signals are provided through the standard 44-pin 2mm connector (J6).

The IDE interface supports the following features:

- Master mode PCI supporting Enhanced IDE devices
- Supports two EIDE devices
- Increased reliability using DMA transfer protocols
- Full scatter-gather capability
- Supports ATAPI compliant devices including DVD
- Supports IDE native and ATA compatibility modes

Table 3-7 gives the signals for the 44-pins of the IDE header.

Table 3-7. IDE Interface Pin/Signal Descriptions (J6)

Pin #	Signal	Description
1	RESET*	Low active hardware reset (RSTDRV inverted)
2	GND	Digital Ground
3	D7	Disk Data 7 – This signal (0 to 15) provides the disk data
4	D8	Disk Data 8 – Refer to D7, pin 3, for more information.
5	D6	Disk Data 6– Refer to D7, pin 3, for more information.
6	D9	Disk Data 9 – Refer to D7, pin 3, for more information.
7	D5	Disk Data 5 – Refer to D7, pin 3, for more information.
8	D10	Disk Data 10 – Refer to D7, pin 3, for more information.
9	D4	Disk Data 4 – Refer to D7, pin 3, for more information.
10	D11	Disk Data 11 – Refer to D7, pin 3, for more information.
11	D3	Disk Data 3 – Refer to D7, pin 3, for more information.
12	D12	Disk Data 12 – Refer to D7, pin 3, for more information.
13	D2	Disk Data 2 – Refer to D7, pin 3, for more information.
14	D13	Disk Data 13 – Refer to D7, pin 3, for more information.
15	D1	Disk Data 1 – Refer to D7, pin 3, for more information.
16	D14	Disk Data 14 – Refer to D7, pin 3, for more information.
17	D0	Disk Data 0 – Refer to D7, pin 3, for more information.
18	D15	Disk Data 15 – Refer to D7, pin 3, for more information.
19	GND	Digital Ground
20	Key	Key pin plug
21	IDRQ0	DMA Request – Used for DMA transfers between host and drive (direction of transfer controlled by DIOR* and DIOW*). Also used in an asynchronous mode with DMACK*. Drive asserts IDRQ0 when ready to transfer or receive data.
22	GND	Digital Ground
23	IOW*	Drive I/O Write – Strobe signal for write functions. Negative edge enables data from a register or data port of the drive onto the host data bus. Positive edge latches data at the host.

Pin #	Signal	Description
24	GND	Digital Ground
25	IOR*	Drive I/O Read – Strobe signal for read functions. Negative edge enables data from a register or data port of the drive onto the host data bus. Positive edge latches data at the host.
26	GND	Digital Ground
27	IORDY	I/O Channel Ready – When negated, extends the host transfer cycle of any host register access when the drive is not ready to respond to a data transfer request. High impedance if asserted.
28	Reserved	Reserved – Not used
29	IDACK0*	DMA Channel Acknowledge – Used by the host to acknowledge data has been accepted or data is available. Used in response to DMARQ asserted.
30	GND	Digital Ground
31	IRQ14	Interrupt Request 14 – Asserted by drive when it has pending interrupt (PIO transfer of data to or from the drive to the host) request.
32	NC	Not connected
33	A1	Drive Address Bus 1 – Used to indicate which byte in the ATA command block or control block is being accessed
34	NC	Not connected
35	A0	Drive Address Bus 0 – Used (0 to 2) to indicate which byte in the ATA command block or control block (register) is being accessed.
36	A2	Drive Address Bus 2 – Used (0 to 2) to indicate which byte in the ATA command block or control block (register) is being accessed.
37	CS0	Chip Select 0 – Used to select the host-accessible Command Block Register.
38	CS1	Chip Select 1 – Used to select the host-accessible Command Block Register.
39	Reserved	Reserved – Not used
40	GND	Digital Ground
41	+5V	+5 volts ±5% power supply
42	+5V	+5 volts ±5% power supply
43	GND	Digital Ground
44	Reserved	Reserved – Not used

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Floppy Drive Interface (J8)

The Super I/O chip provides the floppy controller, which is IBM PC Compatible, and supports two floppy drives. The floppy signals are provided through the 26-pin 2mm connector (J8).

Table 3-8. Floppy Drive Interface Pin/Signal Descriptions (J8)

Pin #	Signal	Description
1	NC - KEY	Not Connected – Key
2	DRVDEN0*	Drive Density Select 0 – This signal indicates a low (250/300kbps) or high (500kbps) data rate has been selected.
3	GND	Digital Ground
4	INDEX*	Index – Indicates when the head is positioned over the beginning of a track as marked by an index hole.
5	GND	Digital Ground
6	MTR0*	Motor Enable 0 – Enables the motor on floppy drive 0.
7	GND	Digital Ground
8	DS1*	Drive Select 1 – Selects floppy drive 1.
9	GND	Digital Ground
10	DS0*	Drive Select 0 – Selects floppy drive 0.
11	MTR1*	Motor Enable 1 – Enables the motor on floppy drive 1.
12	DIR*	Direction – This signal determines direction of head movement (0 = inward motion, 1 = outward motion).
13	GND	Digital Ground
14	STEP*	Step – A pulse signal moves the head to next track. Controls each track-to-track movement of the head.
15	WDATA*	Write Data – Encoded data to the drive for write operations.
16	WGATE*	Write Gate – Enables current flow in the write head just prior to a write operation to the floppy diskette.
17	GND	Digital Ground
18	TRK0*	Track 0 – Indicates when the head is positioned over track 0 (outermost track).
19	GND	Digital Ground
20	WRTPRT*	Write Protect – Indicates the inserted floppy diskette is write protected.
21	GND	Digital Ground
22	RDATA*	Read Data – Raw serial bit stream from the drive for read operations.
23	GND	Digital Ground
24	HDSEL*	Head Select – Selects the side of diskette for Read/Write operations (0 = side 1, 1 = side 0)
25	GND	Digital Ground
26	DSKCHG*	Disk Change – Senses the drive door is open or the diskette has been changed since the last drive selection.

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Parallel Port (J4)

Parallel port supports standard parallel, Bidirectional, ECP and EPP protocols. The Super I/O chip provides the parallel port interface signals to support a Standard Printer Port (SPP), Enhanced Parallel Port (EPP), and Enhanced Capabilities Port (ECP) protocols.

Table 3-9. Parallel Interface (SPP) Pin/Signal Descriptions (J4)

Pin #	Signal	Description
1	Strobe*	Strobe* – This is an output signal used to strobe data into the printer. I/O pin in ECP/EPP mode.
2	AUTOFDX*	Auto Feed * – This is a request signal into the printer to automatically feed one line after each line is printed.
3	PD0	Parallel Port Data 0 – This pin (0 to 7) provides parallel port data signals.
4	ERR*	Error – This is a status output signal from the printer. A Low State indicates an error condition on the printer.
5	PD1	Parallel Port Data 1 – This pin (0 to 7) provides parallel port data signals.
6	INIT*	Initialize * – This signal used to Initialize printer. Output in standard mode, I/O in ECP/EPP mode.
7	PD2	Parallel Port Data 2 – This pin (0 to 7) provides parallel port data signals.
8	SLCTIN	Select In – This output signal is used to select the printer. I/O pin in ECP/EPP mode.
9	PD3	Parallel Port Data 3 – This pin (0 to 7) provides parallel port data signals.
10	GND	Digital Ground
11	PD4	Parallel Port Data 4 – This pin (0 to 7) provides parallel port data signals.
12	GND	Digital Ground
13	PD5	Parallel Port Data 5 – This pin (0 to 7) provides parallel port data signals.
14	GND	Digital Ground
15	PD6	Parallel Port Data 6 – This pin (0 to 7) provides parallel port data signals.
16	GND	Digital Ground
17	PD7	Parallel Port Data 7 – This pin (0 to 7) provides parallel port data signals.
18	GND	Digital Ground
19	ACK*	Acknowledge * – This is a status output signal from the printer. A Low State indicates it has received the data and is ready to accept new data.
20	GND	Digital Ground
21	BUSY	Busy – This is a Status output signal from the printer. A High State indicates the printer is not ready to accept data.
22	GND	Digital Ground

Pin #	Signal	Description
23	PE	Paper End – This is a status output signal from the printer. A High State indicates it is out of paper.
24	GND	Digital Ground
25	SLCT	Select – This is a status output signal from the printer. A High State indicates it is selected and powered on.
26	NC (Key)	Not connected - Key

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Serial Ports (J3, J9)

The Super I/O chip contains the circuitry for the two serial ports, which support the following features:

- Both ports are 16550 compatible
- Programmable word length, stop bits and parity
- 16-bit programmable baud rate generator
- Interrupt generator
- Loop-back mode
- Two 16-bit FIFOs
- Two DMA handshake lines
- Serial 1 (J3, COM1) supports RS232 and full modem support
- Serial 2 (J9, COM2) supports RS232/RS485 with full modem support

NOTE	The RS232/RS485 mode for Serial Port 2 (COM2) is selected using a jumper (JP2) on the board rather than in BIOS. However, the RS232 mode is the default selection for Serial Port 2 (COM2).

Table 3-10 shows signals for corresponding pins in either serial interface.

Table 3-10. Serial Ports Pin/Signal Descriptions (J3, J9)

Pin #	Signal	DB9 #	Description
1	DCD	1	Data Carrier Detect – Indicator to the serial port that external modem is detecting a carrier signal (i.e., a communication channel is currently open). In direct connect environments, this input will be driven by DTR as part of the DTR/DSR handshake.
2	DSR	6	Data Set Ready – Indicator to the serial port that external serial communications device is powered, initialized, and ready. Used as hardware handshake with DTR for overall readiness to communicate.
3	RXD	2	Receive Data – Serial port receive data in
4	RTS	7	Request To Send – Indicator to serial output port is ready to transmit data. Used as hardware handshake with CTS for low level flow control.
5	TXD	3	Transmit Data – Serial port transmit data out

Pin #	Signal	DB9 #	Description
6	CTS	8	Clear To Send – Indicator to the serial port that external serial communication device is ready to receive data. Used as hardware handshake with RTS for low level flow control.
7	DTR	4	Data Terminal Ready – Serial port indicator that port is powered, initialized, and ready. Used as hardware handshake with DSR for overall readiness to communicate.
8	RI	9	Ring Indicator – Indicator to serial port that external modem is detecting a ring condition. Used by software to initiate operations to answer and open the communications channel.
9	GND	5	Digital Ground
10	NC (Key)	NC	Not connected - Key

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

RS485 Port (J10)

The RS485 signals do not pass through the serial transceiver chips, but go directly through the RS485 circuitry to the RS485 connector (J10) from the Super I/O chip (FDC37B782), when enabled by the RS232/RS485 selection jumper (JP2).

- Supports a three-pin RS232/RS485 selection jumper (JP2)
- Supports a two-pin termination jumper (JP1); terminated (default) when jumper installed.
- Supports a two-pin connector (J10) for the RS485 port.

Table 3-11. Serial Ports Pin/Signal Descriptions (J10)

Pin #	Signal	Description
1	I/O +	Rx Data +/ Tx Data +
2	I/O-	RX Data –/ Tx Data –

Utility Interface (J5)

The Utility interface consists of the 10-pin header on the module and is used as the interface for various utility signals. The Super I/O chip drives most of the device interfaces on the Utility interface. Table 3-12 shows the meaning of the interface signals for the utility interface.

- Keyboard and PS/2 Mouse
- Battery
- Reset Switch
- Speaker

Keyboard

The signal lines for the keyboard are provided through the Utility interface (J5).

Mouse

The signal lines for a PS/2 mouse are provided through the Utility interface (J5).

Battery

An external battery input connection is provided through the Utility interface (J5) to provide battery backup for the CMOS RAM and the RTC (Real Time Clock).

Reset Switch

The signal lines for a reset switch are provided through the Utility interface (J5).

Speaker

The signal lines for a speaker with 0.1-watt drive are provided through a Utility interface (J5).

Table 3-11. Utility Interface Pin/Signal Descriptions (J5)

Pin #	Signal	Description
1	SPKR	Speaker Output
2	BATV-	Ground return
3	RESET*	Reset Switch
4	MDATA	Mouse Data
5	KBDATA	Keyboard Data
6	KBCLK	Keyboard Clock
7	GND	Digital Ground
8	KBPWR	Keyboard power (+5V)
9	BATV+	Real time battery voltage (3.6V Type/ 4.0V Max)
10	MCLK	Mouse Clock

Notes: The shaded area denotes power or ground. The signals marked with * indicate active low.

Miscellaneous

Real Time Clock (RTC)

The CoreModule 410 contains a Real Time (time of day) Clock (RTC), which can be backed up with a Lithium Battery. The CoreModule 410 will function without a battery in those environments that prohibit batteries. The CoreModule 410 will also continue to operate after the battery life has been exhausted. Under these conditions all setup information is restored from the onboard serial EEPROM during POST along with the default date and time information.

NOTE	Some operating systems require a valid default date and time to function.

User GPIO Signals

The CoreModule 410 provides GPIO pins for custom use and the signals are routed to connector J2. There is an example of how to use the slave GPIO pins on the CoreModule 410 Doc & SW CD-ROM in the directory, *\examples\CM410\GPIO*.

CAUTION	To prevent a system crash, or render the CoreModule 410 BIOS unusable, do not attempt to use the *master* GPIO pins. The STPC Elite processor has two GPIO blocks, master and slave. The slave GPIO pins are reserved for customer applications. The *master* GPIO pins are dedicated for BIOS use only, where the master pins are used for controlling on-board peripherals. The *master* GPIO pins can not be used for customer applications.

The example program can be built by using the *make.bat* file. This produces a 16-bit DOS executable application, *gpio.exe*, which can be run on the CoreModule 410 to demonstrate the use of GPIO pins. For more information about the GPIO pin operation, refer to the Programming Manual for the STPC Elite processor at:

http://www.stmcu.com/devicedocs-Elite-75.html

Table 3-12. User GPIO Signals Pin/Signal Descriptions (J2)

Pin #	Signal	Description
1	GPIO8	User defined
2	GPIO9	User defined
3	GPIO10	User defined
4	GPIO11	User defined
5	GPIO12	User defined
6	GPIO13	User defined
7	GPIO14	User defined
8	GPIO15	User defined
9	GND	Ground
10	GND	Ground

Notes: The shaded area denotes ground.

Oops! Jumper (BIOS Recovery)

The Oops! jumper is provided in the event you've selected BIOS settings that prevent you from booting the system. By using the Oops! jumper you can stop the current BIOS settings from being loaded, allowing you to proceed, using the default settings. Connect the DTR pin to the RI pin on Serial port 1 (COM 1) prior to boot up to prevent the present BIOS settings from loading. After booting with the Oops! jumper in place, remove the Oops! jumper and go into BIOS Setup.

To restore your BIOS setting changes without the errors, you must first select *Reset CMOS to factory defaults*, which will automatically load and save the defaults and reboot the system. Then you can modify the default settings to your desired values. Ensure you save the changes before rebooting the system.

NOTE	The CoreModule 410 Serial Port 1 (J3) is a 10-pin header and uses pin 7 = DTR and pin 8 = RI. At Serial Port 1, short pin 7 to 8, as shown in Figure 3-2. Alternatively, you may short the equivalent pins on a DB9 connector attached to Serial Port 1 as shown in Figure 3-1.

Figure 3-1. Oops! Jumper

Serial Console

The CM 410 BIOS supports the serial console (or console redirection) function. This function is used with an ANSI-compatible serial terminal, or the equivalent terminal emulation software running on another system. This can be very useful when setting up the BIOS on a production line for systems that are not connected to a keyboard and display.

Serial Console BIOS Setup

The serial console feature may be invoked by entering the appropriate option (port selected) in the Serial Console field of the Custom Configuration Screen in BIOS setup. A standard null modem serial cable is used to connect the chosen serial port on the CM 410 (J3 or J9) to a serial terminal or PC. The serial terminal, or PC with communications software, must be set to the following values:

- 9600 baud
- 8 bits
- One stop bit
- No parity
- No hardware handshake

Hot (Serial) Cable

The BIOS uses the serial console settings unless it detects a "hot cable" connected to one of the serial ports. If a Hot Cable is connected to one of the serial ports, it has the affect of overriding the serial console settings in BIOS. The modified serial cable (or "Hot Cable") is connected between either serial port (J3 or J9) and the serial terminal, or the PC with communications software, which is set to the appropriate values outlined above.

To convert a standard serial cable to a Hot Cable, certain pins must be shorted together at the Serial port connector or at the DB9 connector. Short together the RTS (4) and RI (8) pins on either serial port (J3 or J9) header. As an alternate, you can short the equivalent pins (pins 7 and 9) on the respective DB9 port connector as shown in Figure 3-2.

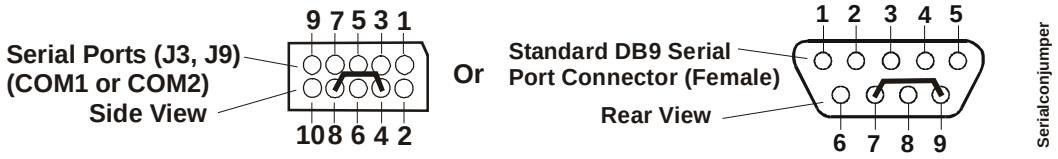

Figure 3-2. Hot Cable Jumper

Watchdog Timer (WDT)

The watchdog timer (WDT) restarts the system if a mishap occurs, ensuring proper start-up after the interruption. Possible problems include failure to boot properly, the application software's loss of control, failure of an interface device, unexpected conditions on the bus, or other hardware or software malfunctions.

The WDT (watchdog timer) can be used both during the boot process and during normal system operation.

- During the Boot process – If the OS fails to boot in the time interval set in the BIOS, the system will reset.

  Enable the WDT in the Custom Configuration Screen of BIOS Setup. Set the WDT for a time-out interval in seconds, between 1 and 255, in one second increments. Ensure you allow enough time for the boot process to complete and for the OS to boot. The OS or application must trigger or turnoff the WDT as soon as it comes up. This can be done by accessing the hardware directly or through a BIOS call.

- During System Operation – An application can set up the WDT hardware through a BIOS call, or by accessing the hardware directly. Some Ampro Board Support Packages provide an API interface to the WDT. The application must trigger (turnoff) the WDT in the time set when the WDT is initialized or the system will be reset. You can use a BIOS call to tigger the WDT or access the hardware directly.

The BIOS implements interrupt 15 function 0C3h to manipulate the WDT. The following machine language routines illustrate how to control the WDT:

NOTE	Some of the following code examples can be used across multiple Ampro board products and whenever applicable will be so noted. Refer to CoreModule 410 Doc & SW CD-ROM for more information.

The following text is provided as code examples to enable, disable and trigger the watchdog timer. Refer to the CoreModule 410 Doc & SW CD-ROM in the *\examples\CM410\watchdog\dos* directory.

The following example can be used across multiple Ampro board products.

Enable watchdog timer

```
mov     ax, 0C301h
mov     bx, 10      ; bx should contain the number of seconds
                    ; before expire. The value has to be between
                    ; 1 and 255 inclusive.
                    ; 10 seconds in this example.
int     15h
```

Disable watchdog timer

```
mov     ax, 0C300h
mov     bx, 0
int     15h
```

Trigger watchdog timer

```
mov     dx,201h
in      al,dx
```

The following code examples, specific to the CoreModule 410, enable, disable and trigger the watchdog timer. These examples are intended for device driver developers who wish to implement watchdog support for Operating Systems, which can not use the proceeding examples:

```
/*
 * Note: parameter validation and/or error checking may not be
 * performed in these examples.
 */

      /*
      * section: non-portable code
      *
      * The following code needs to be ported to your platform/
      * operating system/compiler.
      *
      * Explanation of this code is beyond the scope of these
      * examples.
      */

/* for 32-bit platforms: */
typedef unsigned char  u8;       /* unsigned 8-bit  */
typedef unsigned short u16;      /* unsigned 16-bit */
typedef unsigned long  u32;      /* unsigned 32-bit */

/* 8-bit input cycle from a port on x86 platform */
u8 in_port_8( u16 port )
{
u8 val;
       _asm {
                    push  dx
                    push  ax
                    mov   dx,  port
                    in    al,  dx
                    mov   val, al
                    pop   ax
                    pop   dx
       }
       return val;
}

/* 8-bit output cycle from a port on x86 platform, note parameter
order */
void out_port_8( u16 port, u8 val )
{
       _asm {
                    push      dx
                    push      ax
                    mov       dx, port
                    mov       al, val
```

```
                        out         dx, al
                        pop         ax
                        pop         dx
                }
        }

        /*
         * this function saves the current state of the flags in the
         * supplied parameter and then disables the interrupts.
         */
        void disable_ints( u32 *save_flags )
        {
        u32 flags;
                _asm {
        .386
                        pushfd
                        pop         eax
                        mov         flags, eax
                        cli
                }
                *save_flags = flags;
        }
        /*
         * This function restores the flags so that interrupts are reenabled
         * if they were in a previous call to disable_ints
        */
        void restore_ints( u32 flags )
        {
                _asm {
        .386
                        mov         eax, flags
                        push        eax
                        popfd
                }
        }

        /*
         * This function generates a software SMI on the STPC Elite. The SMI
         * code expects the value in BX to be the watchdog timer reload value.
         * In version R1.00 of the BIOS the SMM code does not clear the
         * software SMI, so it's done in this routine. In the next release
         * of the BIOS, the SMM code does clear the software SMI bit.
        */
        void generate_smi( u16 bx_val )
        {
                _asm {
                        pushf
                        cli
                        push        bx
                        push        ax
                        mov         al, 71h
                        out         22h, al
                        in          al, 23h
                        or          al, 00000010b
                        mov         bx, bx_val
                        out         23h, al
                        and         al, 11111101b
                        out         23h, al
                        pop         ax
```

```
                    pop     bx
                    popf
            }
    }

            /*
             * section: Support routines
             *
             * The following routines support the watchdog functions.
             * They are written to be portable across platforms.
             */

    u8 get_stpc( u8 index )
    {
            out_port_8( 0x22, index );
            return in_port_8( 0x23 );
    }

    void set_stpc( u8 index, u8 val )
    {
            out_port_8( 0x22, index );
            out_port_8( 0x23, val );
       }

            /*
             * section: Ampro Watchdog example
             *
             * The following code should allow you to create drivers for
             * any platform to support the watchdog timer functionality.
             *
             * Theory of Operation:
             * The 'enable_watchdog' function sets the timer reload value
             * and starts the timer. The timer counts down, starting at
             * the timer reload value.
             * If the timer reaches zero then the board will be reset.
             * If the 'kick_watchdog' function is called before the
             * timer reaches zero, then the timer is loaded with the
             * timer reload value and starts counting down from the
             * newly loaded value.
             * If the 'disable_watchdog' function is called before the
             * timer reaches zero, then the timer is disabled.
             * In this state the timer will not reach zero and the
             * board is not reset.
             */

       void disable_watchdog( void )
       {
       u32 flags;
            disable_ints( &flags );

            /* Disable Doze timer */
            set_stpc( 0x8D, get_stpc( 0x8D ) & 0x1F );

            /* Disable Activity detection on IO range */
            set_stpc( 0x69, get_stpc( 0x69 ) & 0xFB );

            restore_ints( flags );
    }
```

```
void enable_watchdog( unsigned seconds )
{
u32 flags;
u16 hseconds;
        if( (seconds < 1) || (seconds > 255) )
              return;

        disable_watchdog();

        disable_ints( &flags );


        /* convert seconds to 500 millisecond units */
        hseconds = seconds * 2;

        /*
         * adjust for ISA clock difference. Timer assumes 7.16MHz
         * clock for 500ms ticks. CM410 uses 8.25MHz clock
         * by default.
         */
        if( get_stpc( 0x50 ) & 0x10 ) {
              hseconds = (u32)hseconds * 8250L / 7160L;
        }

       /* generate software SMI, which loads the timer reload value */
        generate_smi( hseconds );

        /* set trappable address range to 0x200 for 4 bytes */
        set_stpc( 0x6B, 0x02 );
        set_stpc( 0x6C, 0x00 );

        /* Enable Activity detection on IO range */
        set_stpc( 0x69, get_stpc( 0x69 ) | 0x04 );


        /* Set Doze timer timeout to generate SMI */
        set_stpc( 0x71, get_stpc( 0x71 ) | 0x80 );

        /* Enable Doze timer with 500millisec interval */
        set_stpc( 0x8D, (get_stpc( 0x8D ) & 0x1F) | 0x60 );

        restore_ints( flags );
}

void kick_watchdog( void )
{
        /* This triggers an SMI which will reset the countdown timer
        /* to the value provided in enable_watchdog.
         */
        in_port_8( 0x201 );
}
```

Power Interface (J7)

The CoreModule 410 requires one +5 volt power source and uses a 10-pin header with 0.100” spacing. The power input connector (J7) supplies the following voltage directly to the module:

- 5.0VDC +/- 5% @ 0.700 Amps

Table 3-13 gives the pin and signals for the power interface connector (J7) and Table 3-14 gives the pin arrangement of the connector.

NOTE	If the +5 power drops below ~4.0V, a low voltage reset trigger activates a system interrupt.

Table 3-13. Power Interface Pins/Signals (J7)

Pin #	Signal	Descriptions
1	GND	Ground
2	+5V	+5 Volts
3	GND	Ground
4	+12V	+12 volts is not used.
5	GND	Ground
6	NC	Not connected - Key
7	GND	Ground
8	+5V	+5 Volts
9	GND	Ground
10	+5V	+5 Volts

Note: The shaded area denotes power or ground.

Table 3-14. Power Interface Pin Arrangement (J7)

Pin #	Signal	Pin #	Signal
1	GND	2	+5V
3	GND	4	+12V
5	GND	6	NC - Key
7	GND	8	+5V
9	GND	10	+5V

Note: The shaded area denotes power or ground.

Chapter 4 BIOS Setup

Introduction

This chapter describes the BIOS Setup menus and the various screens used for configuring the CoreModule 410. Some features in the Operating System or application software may require configuration in the BIOS Setup screens.

This section assumes the user is familiar with general BIOS setup and does not attempt to describe the BIOS functions. Refer to the appropriate PC reference manuals for information about the software interface of the onboard ROM-BIOS. If Ampro has added to or modified the standard functions, these functions will be described.

The options provided for the CoreModule 410 are controlled by the BIOS Setup Utility. BIOS Setup is used to configure the CoreModule 410 features, modify the fields in the Setup screens, and save the results in the onboard configuration memory. Configuration memory consists of portions of the CMOS RAM in the battery-backed real-time clock chip and the flash memory.

The Setup information is retrieved from configuration memory when the module is powered up or when it is rebooted. Changes made to the Setup parameters, with the exception of the time and date settings, do not take effect until the module is rebooted.

Setup is located in the ROM BIOS and can be accessed, when prompted using the <Del> key, while the module is in the Power-On Self Test (POST) state, just before completing the boot process. The screen displays a message indicating when you can press <Del>, as described in the following topics.

The CoreModule 410 Setup is used to configure items in the BIOS using the following menus:

- Basic CMOS Configuration
- Features Configuration
- Custom Configuration
- PnP Configuration
- Reset CMOS to last known values
- Reset CMOS to Factory Defaults
- Write to CMOS and Exit
- Exit without changing CMOS

Table 4-2 summarizes the list of BIOS menus and some of the features available for CoreModule 410. The BIOS Setup menu offers the menu choices listed above and the related topics and screens are described on the following pages.

Accessing BIOS Setup (Serial Console)

Entering the BIOS Setup, in serial console mode, is very similar to the steps you would use to enter BIOS Setup with an optional video card (video output and keyboard connection), except for the actual keys used.

1. Connect the serial console, or the PC with serial terminal emulation, to Serial Port 1 (J3) or Serial Port 2 (J9) of the CoreModule 410.
   - If the BIOS option, *Serial Console Mode* is set to [Always], use a standard null-modem serial cable.
   - If the BIOS option, *Serial Console Mode* is set to [Hot Cable], use the modified serial cable described in Chapter 3, under *Hot (Serial) Cable*.
2. Turn on the serial console or the PC with serial terminal emulation and the power supply to the CoreModule 410.
3. Start Setup by pressing the Ctl–c keys, when the following message appears on the boot screen.

   ```
   Hit ^C if you want to run SETUP
   ```

4. Use the <Enter> key to access the Setup screen menus listed in the BIOS opening screen.

5. Depending on your terminal, you may have to use the substitute key equivalents for the "Arrow" keys.

   The "Arrow" keys are simulated using the following substitute key equivalents. See Table 4-1.

Table 4-1. Substitute Arrow Key Equivalents

Function	Substitute Keys
Up Arrow (^)	Ctrl-e
Down Arrow (v)	Ctrl-x

Accessing BIOS Setup (Optional VGA Display)

To access BIOS Setup using a VGA display for the CoreModule 410:

1. Turn on the VGA monitor and the power supply to the CoreModule 410.
2. Start Setup by pressing the [Del] key, when the following message appears on the boot screen.

```
Hit <Del> if you want to run SETUP
```

NOTE	If the setting for *Memory Test* is set to FastHigh, you may not see this prompt appear on screen if the monitor is too slow to display it on start up. If this happens, use the <Del> key early in the boot sequence to enter BIOS Setup.

3. Use the <Enter> key to access the screen menus listed in the BIOS opening screen.
4. Follow the instructions at the bottom of each screen to navigate through the selections and modify any settings.

Table 4-2. BIOS Setup Menus

BIOS Setup Menu	Item/Topic
Basic CMOS Configuration	Date and Time Drive Assignment Order Boot order Keyboard settings Boot device Memory settings IDE Drive Settings Floppy Drive types
Features Configuration	Power management, splash screen, and Memory management
Custom Configuration	Memory Settings On-Board settings (Floppy, IDE, COM ports, etc.) Serial Console and Watchdog Timer settings
PnP Configuration	Plug and Plug setup or IRQ assignments
Reset CMOS to last known values	Resets the (BIOS) CMOS to the last boot settings of the BIOS prior to any changes you could have made.
Reset CMOS to Factory Defaults	Option to reset CMOS
Write to CMOS and Exit	Option to write to CMOS and exit
Exit without change CMOS	Option to exit CMOS

BIOS Setup Opening Screen

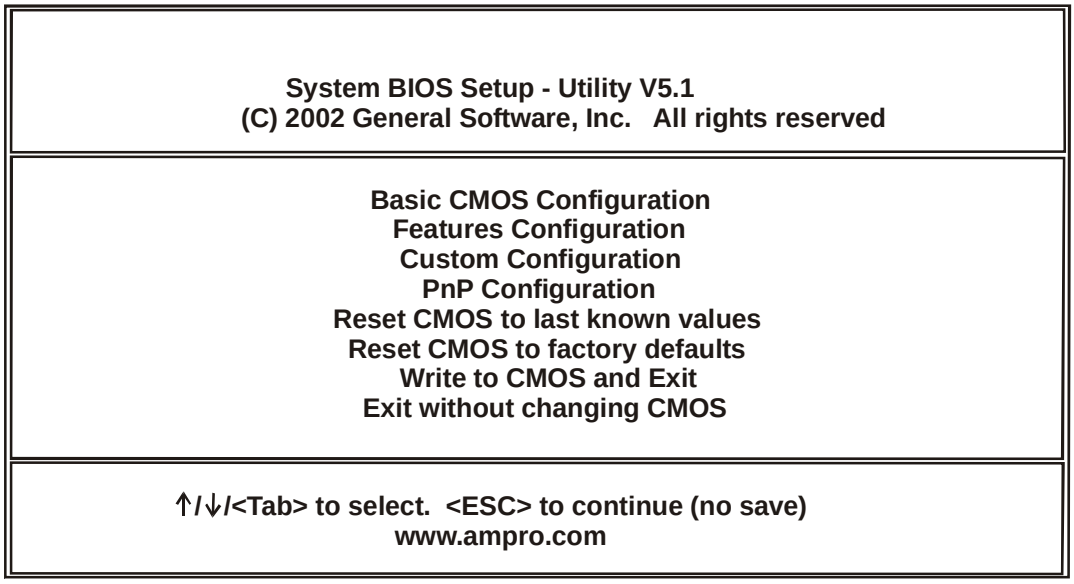

Figure 4-1. BIOS Setup Opening Screen

NOTE	The default values or the typical settings are shown highlighted (**bold text**) in the list of options for each menu selection. Refer to the bottom of the BIOS screens for the navigation instructions when making selections.

Basic CMOS Configuration Screen

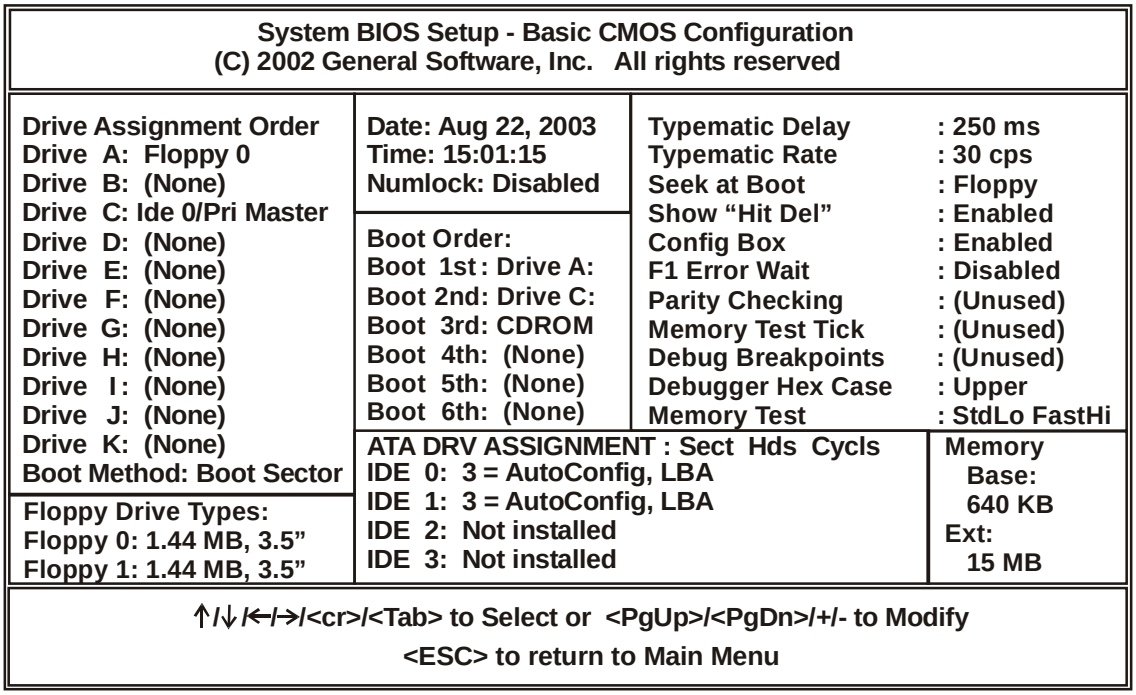

Figure 4-2. Basic CMOS Configuration Screen

- Left Column Display, Drive Assignment Order – This section provides up to two supported floppy drives, and lists up to 9 IDE drives, but only two IDE drives are supported.
  - Drives A & B – [None], [**Floppy 0**], or [Floppy 1]
  - Drives C-D – [None], [Floppy 0], [Floppy 1], [**Ide 0/Pri Master**], or [Ide 1/Pri Slave]

CAUTION	To prevent difficulties in reading from the CD-ROM, do not attempt to configure the CD-ROM in the Drive Assignment Order. The CD-ROM is not configured in Drive Assignment Order, but is configured in the ATA DRV ASSIGNMENT and listed in the boot order. A DiskOnChip is also not listed as a drive and is not in the boot order.

  - Drives E-K – [**None**], [Ide 0/Pri Master], or [Ide 1/Pri Slave]
  - Boot Method – [**Boot Sector**] or [Windows CE]
  - Floppy Drive Types – [Not installed], [360KB, 5.25"], [1.2MB, 5.25"], [720KB, 3.5"], [**1.44MB, 3.5"**], or [2.88MB, 3.5"]
- Middle Column Display, Date, Time, & Numlock – Allows you to set the time, date, and Numlock option before leaving BIOS Setup.
  - Date fields allow for the month (**Jan** to Dec), day (**1** to 31), and year (**1980** to 2099)
  - Time fields allow for a 24 hour clock in Hours, Minutes, and Seconds
  - Numlock: – [**Disabled**] or [Enabled]
- Middle Column Display, Boot Order – Change the boot order of 6 devices and selections are:

- [None], [**(1st) Drive A**], [Drive B], [**(2nd) Drive C**], [Drive D], [Alarm], [Browser], [RAS], [Power Off], [Reboot], [High ROM], [DOS-ROM], [Mfg Mode], [Debugger], [(3rd) **CDROM**], or [CLI]

NOTE	The default Boot order is, A, C, CD-ROM, and the BIOS will start its search for a bootable device in drive A, then C, then the CD-ROM. If no bootable device is found, the screen will display "No Bootable Device Available" and the boot process will stop, allowing you to select from: R – for Reboot, or S – for Setup. If you do not choose R or S, the boot process stops, but will reboot after 5 seconds.

- Middle-Bottom Column, ATA DRV ASSIGNMENT:
  - Ide 0-3: [Not installed], [1 = User Type, Set 01 Hds 1 Cyls 0], [2 = Autoconfig, Physical], [**3 = Autoconfig, LBA**], [4 = Autoconfig, Phoenix], or [5 = IDE CDROM]

    This feature determines the IDE configuration of each IDE device connected to the CoreModule 410. The ATA DRV ASSIGNMENT list identifies up to four IDE devices (two channels with two devices per channel), but the CoreModule 410 only uses one IDE channel. Therefore, the Secondary channel appears as [Not installed]. The IDE configuration order is listed below:
    - Ide 0 = Primary Master
    - Ide 1 = Primary Slave
    - Ide 2 = Secondary Master
    - Ide 3 = Secondary Slave

    This feature also defines the setup information for each IDE device. However, newer IDE hard drives only use [3 = Autoconfig, LBA] for configuration setup. Older IDE hard drives and older OSs use the other settings.

NOTE	To boot from the CD-ROM, this item [5 = IDE CDROM] must be selected and appear in the ATA DRV ASSIGNMENT list as primary master or slave. However, if this item [5 = IDE CDROM] is in the ATA DRV ASSIGNMENT list and there is no CD-ROM connected, the boot process will be slowed down considerably.

- Right Column Display
  - Typematic Delay – [Disabled], [**250ms**], [500ms], [750ms], or [1000ms]

    This feature is used for the keyboard and determines the typing delay.
  - Typematic Rate – [**30cps**], [24cps], [20cps], [15cps], [12cps], [10cps], [8cps], or [6cps]

    This feature is used for the keyboard and determines the typing rate.
  - Seek at Boot – [None], [**Floppy**], [Ide], or [Both]

    This feature determines the type of boot device the Setup will search for during the boot process.
  - Show "Hit Del" – [Disabled] or [**Enabled**]

    This feature, if Enabled, will place "Hit Del" on screen during the boot process, to indicate when you may Hit Del to enter the BIOS Setup menus.
  - Config Box – [Disabled] or [**Enabled**]

    This feature, if Enabled, displays the Configuration Summary Box, which list all of the configuration information for the system, at the completion of POST, but before the Operating System is loaded.

- F1 Error Wait – [**Disabled**] or [Enabled]

  This feature, if Enabled, will display an Error message indicating when an error has occurred and wait for you to respond by hitting the F1 key.

- Parity Checking – (**Unused**)
- Memory Test Tick – (**Unused**)
- Debug Breakpoints – (**Unused**)
- Debugger Hex Case – [**Upper]** or [Lower]

  This feature is typically not used, even though the default setting is Upper.

- Memory Test – [**StdLo FastHi**], [StdLo StdHi], [StdLo FullHi], [FullLo FastHi], [FullLo StdHi], [FullLo FullHi], [FastLo FastHi], [FastLo StdHi], or [FastLo FullHi]

  The memory type and speed are represented by the following abbreviations: Lo = Base memory, Hi = Extended memory, Std = standard test, Full = full (or complete) and extensive memory test, Fast = fast test. Where StdLo FastHi = standard test of base memory and a fast test of extended memory.

Features Configuration Screen

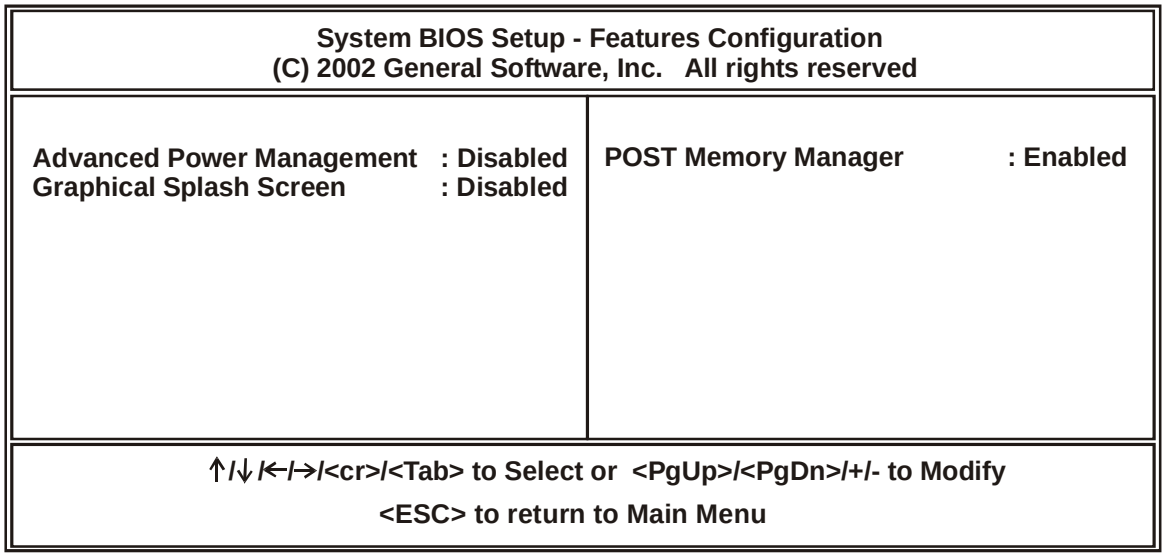

Figure 4-3. Features Configuration Screen

- Left Column
  - Advanced Power Management – [**Disabled**] or [Enabled]
  - Graphical Splash Screen – [**Disabled**] or [Enabled]

    This feature enables the Splash Screen and displays a default or customized splash screen. Refer to the Splash Screen Customization topic later in this chapter for instructions on how to customize the splash screen.

NOTE	If you provide a graphical image for the splash screen, it will not appear on the serial console. However, if you use an external video board to drive a CRT or flat panel, the splash screen will appear on screen depending on the options enabled/disabled.

- Right Column
  - POST Memory Manager – [Disabled] or [**Enabled**]

Custom Configuration Screen

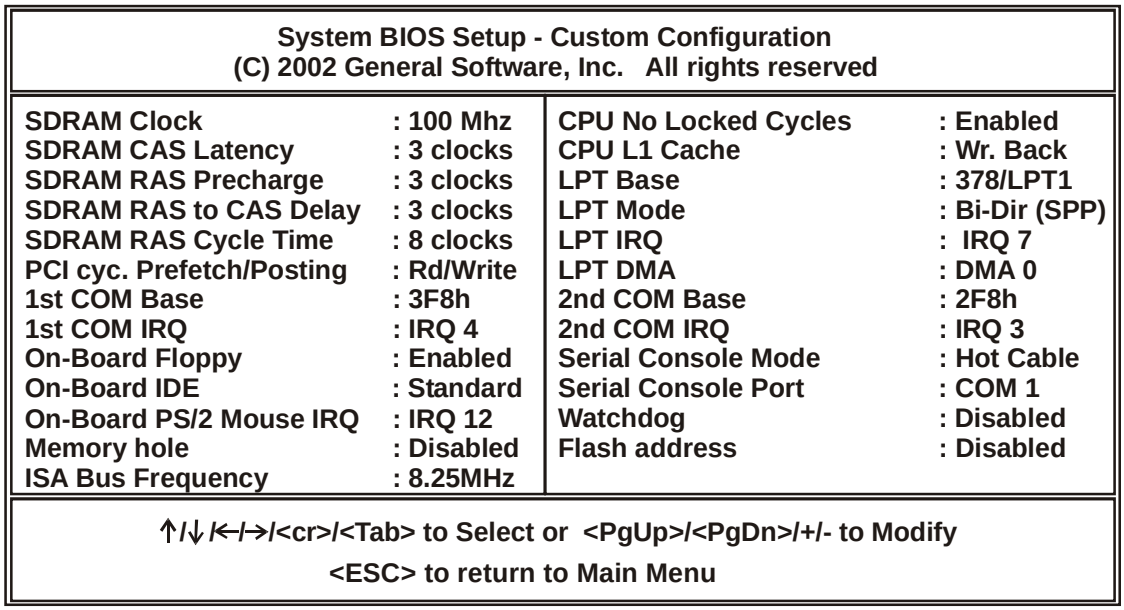

Figure 4-4. Custom Configuration Screen

- Left Column Display
  - SDRAM Clock – [45 MHz], [50 MHz], [55 MHz], [60 MHz], [66 MHz], [75 MHz], [80 MHz], [85 MHz], [90 MHz], [95 MHz], or [**100** MHz]
  - SDRAM CAS Latency – [1 clock], [2 clocks], or [**3 clocks**]
  - SDRAM RAS Precharge – [1 clock], [2 clocks], [**3 clocks**], or [4 clocks]
  - SDRAM RAS to CAS Delay – [1 clock], [2 clocks], [**3 clocks**], or [4 clocks]
  - SDRAM Cycle Time – [6 clocks], [7 clocks], [**8 clocks**], or [9 clocks]
  - PCI cyc. Prefetch/Posting – [Disabled], [Write], [Read], or [**Rd/Write**]
  - 1st COM Base – [Disabled], [**3F8h**], [2F8h], [3E8h], [2E8h], [7F8h], or [7E8h]
  - 1st COM IRQ – [IRQ 3], [**IRQ 4**], [IRQ 5], [IRQ 6], [IRQ 7], [IRQ 9], [IRQ 10], [IRQ 11], or [IRQ 12]

    This field selects the IRQ for COM 1 (Serial 1) and is ignored if 1st COM Base is set to [Disabled]
  - On-Board Floppy – [Disabled] or [**Enabled**]
  - On-Board IDE – [Disabled], [**Standard**], or [Sec Nativ]
    - If this field is set to [Disabled], then the On-Board IDE controller and all its resources are available for other devices, including IRQ 14 and IRQ 15.
    - If this field is set to [Standard], then the IDE controller uses all of the resources available for the primary and secondary IDE interface, including IRQ 14 and IRQ 15.

- If this field is set to [Sec Nativ (secondary Native mode], then the IDE controller is only configured for the primary IDE interface with its standard resources, including IRQ 14. The second IDE channel is configured for native mode, making the resources normally assigned to the secondary IDE controller available for other devices, including IRQ 15.

NOTE	Using [Sec Nativ] mode allows the use of IRQ 15 by an additional PC/104 adapter board.

- On-Board PS/2 Mouse IRQ – [Disabled], [IRQ 3], [IRQ 4], [IRQ 5], [IRQ 6], [IRQ 7], [IRQ 9], [IRQ 10], [IRQ 11], or [**IRQ 12**]
- Memory Hole – [**Disabled**], [15-16MB], [14-16MB], [12-16MB], or [8-16MB]

  This field enables the memory hole below 16MB of the size specified. This field is used for some legacy ISA devices and is not normally used. The available extended memory is decreased by the amount specified in this field. If you have difficulty using this feature, refer to the release notes concerning this feature.
- ISA Bus Frequency – [7.16MHz] or [**8.25MHz**]

- Right Column Display
  - CPU No Locked Cycles – [Disabled] or [**Enabled**]
  - CPU L1 Cache – [Disabled] and [Wr. Thru] or [**Wr. Back**]
  - LPT Base – [Disabled], [**378/LPT1**], 278/LPT2], or [3BC/LPT3]
  - LPT Mode – [Standard], [**Bi-Dir SPP**], [EPP1.9 +SPP], [EPP1.7 +SPP], [ECP], [ECP+EPP1.9], [ECP+EPP1.7],

    This field is ignored if LPT Base is set to [Disabled].
  - LPT IRQ – [IRQ 3], [IRQ 4], [IRQ 5], [IRQ 6], [**IRQ 7**], [IRQ 9], [IRQ 10], [IRQ 11], or [IRQ 12]

    This field is ignored if LPT Base is set to [Disabled].
  - LPT DMA – [**DMA 0**], [DMA 1], [DMA 2], or [DMA 3]

    This field is only used if LPT Mode is set to [ECP].
  - 2nd COM Base – [Disabled], [3F8h], [**2F8h**], [3E8h], [2E8h], [7F8h], or [7E8h]
  - 2nd COM IRQ – [**IRQ 3**], [IRQ 4], [IRQ 5], [IRQ 6], [IRQ 7], [IRQ 9], [IRQ 10], [IRQ 11], or [IRQ 12]

    This field selects the IRQ for COM 2 (Serial 2) and is ignored if 2nd COM Base is set to [Disabled].
  - Serial Console Mode – [**Hot Cable**] or [Always]
    - The Hot Cable option only allows console redirection when a Hot Cable is actually connected to COM 1 or 2. Use the modified serial cable described in Chapter 3, under *Hot (Serial) Cable*.
    - The Always option instructs the BIOS to operate in the console redirection mode at all times with the serial port selected in the Serial Console Port field. Use a standard null-modem serial cable.
    - However, connecting a Hot Cable to the other port (port not selected) overrides the setting of this field [Always] and the Serial Console Port field.

- Serial Console Port – [**COM 1** (Serial 1)] or [COM 2 (Serial 2)]
  - This field selects the COM (Serial) port used for console redirection when [Always] has been selected in Serial Console Mode. Use a standard null-modem serial cable.
  - However, connecting a Hot Cable to the other port (port not selected) overrides this field setting and activates the connected port.
- Watchdog Timer – [**Disabled**] or [select whole number between 1 and 255 seconds, in 1 second increments]

  This feature, if enabled by selecting a timer interval, will direct the watchdog timer to reset the system if it fails to boot the OS properly.
- Flash Address – [**Disabled**], [8MB], [9MB], [10MB], [11MB], [12MB], [13MB], [14MB], or [15MB]

  This options sets the base address of the Flash memory. Flash memory can be accessed at the settings of *Flash Address*, if the *Memory Hole* setting (below 16MB) is enabled for the respective region. However, use of the Memory Hole for this purpose is not recommended. Refer to the Release Notes and Virtual Technician on the Ampro web site for more information.

  Ampro recommends using Flash memory at 128MB intervals above the base address (with the exception of 256MB). For example, if the Flash address is set to 8MB, then the Flash memory can be accessed at 136MB, 392MB, 520MB etc. It is also possible to load an application in the available lower Flash memory and boot directly to this application. For more information and details on how to use On-Board Flash, refer to *the On-Board Flash Access and Use* later in this chapter.

CAUTION	To prevent a system crash or an unusable BIOS, do not overwrite the BIOS. The entire 1MB of Flash is accessible, but only the lower 768kB region is available for custom applications. The higher 256kB region is reserved for the BIOS and can be overwritten, **rendering the CoreModule 410 BIOS unbootable!**

PnP Configuration Screen

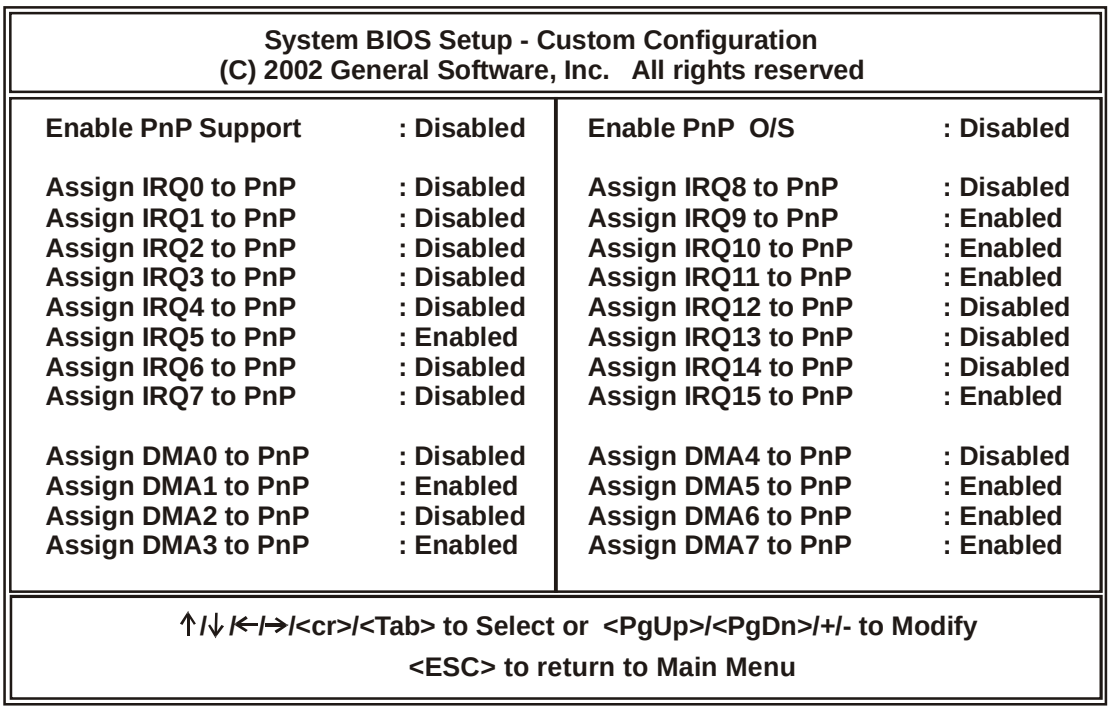

Figure 4-5. PnP Configuration Screen

This screen allows BIOS support of Plug and Play (PnP) ISA cards. If Enable PnP Support is set to [Enabled], then the BIOS can use the following settings of IRQ and DMA, depending how each one is set (Disabled or Enabled), to assign to a PnP ISA card.

- Left Column
  - Enable PnP Support – [**Disabled**] or [Enabled]
    - If Enable PnP Support is set to [Enabled], the BIOS can assign available resources to PnP ISA cards and dynamically configure the PnP ISA cards, very much like Plug and Play PCI cards.
    - If Enable PnP Support is set to [Disabled], then the following fields and values are not valid for any Plug and Play ISA cards.
  - Assign IRQ0 to PnP – [**Disabled**] or [Enabled] (Typically Timer)
  - Assign IRQ1 to PnP – [**Disabled**] or [Enabled] (Typically Keyboard)
  - Assign IRQ2 to PnP – [**Disabled**] or [Enabled] (Typically Secondary Cascade)
  - Assign IRQ3 to PnP – [Disabled] or [**Enabled**] (Typically COM2)
  - Assign IRQ4 to PnP – [Disabled] or [**Enabled**] (Typically COM1)
  - Assign IRQ5 to PnP – [Disabled] or [**Enabled**]
  - Assign IRQ6 to PnP – [**Disabled**] or [Enabled] (Typically Floppy Disk)
  - Assign IRQ7 to PnP – [Disabled] or [**Enabled**] (Typically LPT1)
  - Assign DMA0 to PnP – [**Disabled**] or [Enabled]
  - Assign DMA1 to PnP – [**Disabled**] or [Enabled]
  - Assign DMA2 to PnP – [**Disabled**] or [Enabled]

- Assign DMA2 to PnP – [Disabled] or [**Enabled**]

- Right Column
  - Enable PnP O/ S – [**Disabled**] or [Enabled]
  - Assign IRQ8 to PnP – [**Disabled**] or [Enabled] (Typically RTC)
  - Assign IRQ9 to PnP – [Disabled] or [**Enabled**] (Typically unused)
  - Assign IRQ10 to PnP– [Disabled] or [**Enabled**] (Typically unused)
  - Assign IRQ11 to PnP– [Disabled] or [**Enabled**] (Typically ISA Bridge/Native IDE)
  - Assign IRQ12 to PnP– [**Disabled**] or [Enabled] (Typically PS/2 Mouse)
  - Assign IRQ13 to PnP – [Disabled] or [**Enabled**] (Typically Coprocessor)
  - Assign IRQ14 to PnP – [**Disabled**] or [Enabled] (Typically Hard Disk)
  - Assign IRQ15 to PnP – [**Disabled**] or [Enabled] (Typically Hard Disk)
  - Assign DMA4 to PnP – [Disabled] or [**Enabled**]
  - Assign DMA5 to PnP – [Disabled] or [**Enabled**]
  - Assign DMA6 to PnP – [Disabled] or [**Enabled**]
  - Assign DMA7 to PnP – [Disabled] or [**Enabled**]

Splash Screen Customization

The CoreModule 410 BIOS supports a graphical splash screen, which can be customized by the user and displayed on screen when enabled through the BIOS Setup Utility. The graphical image can be a company logo or any custom image the user wants to display during the boot process. The custom image can be displayed as the first image displayed on screen during the boot process and remain there, depending on the options selected in BIOS Setup, while the OS boots.

Splash Screen Image Requirements

The user's image may be customized with any bitmap software editing tool, but must be converted into an acceptable format with the tools (files and utilities) provided by Ampro. If the custom image is not converted with the utilities provided, then the image will not display properly when this feature is selected in BIOS Setup.

NOTE	Do not use other splash screen conversion tools, as these will render an image that is not compatible with the CoreModule 410 BIOS.

The splash screen image supported by the CoreModule 410 BIOS should be:

- Bitmap image
- Exactly 640x480 pixels
- Exactly 16 colors
- A converted file size of not greater than 100KB (102,400 bytes)

Converting the Splash Screen File

The following files are provided by Ampro on the CoreModule 410 Doc & SW CD-ROM and are required for converting a custom splash screen file. Refer to the CD-ROM for the utilities and an example of how to load a custom image in the *\examples\CM410\splash* directory.

- splash.bmp
- convert.exe
- convert.idf
- resplash.com
- cm410.bin

The process of converting and loading a custom image onto the CoreModule 410 involves the following sequence of events:

- Prepare directory for conversion (create directory and copy files into it)
- Obtain the CoreModule 410 BIOS binary
- Prepare the custom image file
- Convert the image to an acceptable BIOS format
- Merge the image with BIOS binary to create new BIOS binary
- Load the new BIOS binary onto the CoreModule 410 board

NOTE	You can use any Windows PC to convert the custom image, but your PC must have an internet browser to access, view, and make selections in the main menu of the CoreModule 410 Doc & SW CD-ROM. For example: Microsoft Internet Explorer 4.x, or greater, Netscape Navigator version 4.x, or greater, or the equivalent.

Use the following steps to convert and load your custom image onto the CoreModule 410.

1. Copy the files from the *\examples\CM410\splash* directory on the CD-ROM to a new directory (conversion directory) on your PC.

   This new conversion directory is where you intend to do the conversion and save the file.

2. Ensure you remove the read-only attributes from all the files as part of the file copying process.

3. Copy the CoreModule 410 BIOS binary file (cm410.bin) to the new conversion directory on your PC where the other files and utilities are located.

   If this file is not on the CoreModule 410 Doc & SW CD-ROM, you will have to obtain it from Ampro.

NOTE	Ampro recommends keeping a copy of this original cm410.bin file, just in case you encounter problems with your new file or have difficulty updating the BIOS with the new image.

4. Prepare your custom image file with any Windows bitmap software editing tool.

   - For example, Corel Photo-Paint, Adobe Photoshop, or the Windows Paint program provided with Windows. You can insert a desired graphic image, logo, text, etc. into the file.
   - The custom image must be a bitmap image in .bmp format at 640x480 pixels and it must be 16 colors. The file should be about 153,718 bytes. Refer to the example file splash.bmp.

5. Save your custom image file as splash.bmp at 640x480 pixels by 16 colors.

   - If your custom image file is not approximately 153,718 bytes in size it is probably not in the right format or is to complex to be used in the BIOS. You will have to edit it down in size until you have reached an acceptable file size.
   - If you are doubtful about the conversion process, due to the file size, Ampro recommends making a copy of your new splash.bmp, so that you can edit it later if the conversion does not yield a small enough file. Otherwise, you may have to re-create your custom image before you can edit it down to an acceptable file size.

6. If your custom image file is not on the conversion PC, copy the new splash.bmp file to the conversion directory.

7. Run the following command from DOS, or a Windows DOS pop-up screen to convert your new splash.bmp file.

   ```
   C:\splash>convert convert.idf
   ```

   This conversion should yield a *splash.rle* file of approximately 100kB in size or less, depending on the complexity of your image.

8. If the splash.rle file size is greater than 100kB (102,400 bytes), go back to the unconverted image file and edit it to a smaller file size.

   Try to make it smaller, down to less than or equal to 100kB, by reducing the image's complexity.

9. Run the following command to merge the converted image with the BIOS binary file.

   ```
   C:\splash>resplash cm410.bin splash.rle cm410n.bin
   ```

   This creates a new BIOS named cm410n.bin, which has the new splash image. This new BIOS is ready to be loaded onto the CoreModule 410.

10. Copy the files update.bat, aflash.exe, and cm410n.bin to a DOS boot floppy.

11. Boot the CoreModule 410 from the floppy and run update.bat.

12. Cycle the power to the CoreModule 410 and enter BIOS Setup.

13. Go to the Features Configuration screen and set *Graphical Splash Screen* to **Enabled**.
14. Save changes and exit, which will reboot the system.

    The new custom image should appear on screen during the boot process, depending on your BIOS settings.
15. As an option, go to the Basic CMOS Configuration screen and set *Config Box* to **Disabled**.
    - If the Config box (Basic CMOS Configuration Screen) is [Disabled] prior to the boot process, then the splash screen remains on the display while the OS boots. The splash screen will stay on the display until the OS changes it, or the OS changes video modes.
    - If the Config box (Basic CMOS Configuration Screen) is [Enabled] prior to the boot process, then the splash screen appears on the display until the BIOS displays the Config box just before the OS boots

On-Board Flash Access and Use

This section describes how to use the on-board flash memory and load applications to it in the available lower 768kB region of the 1MB of Flash Memory. The applications loaded into the on-board flash memory can also be used to boot directly from.

The Flash memory can be accessed at 128MB intervals above the base address (with the exception of 256MB). For example, if the Flash address is set to 8MB, then the Flash memory can be accessed at 136MB, 392MB, 520MB etc.

CAUTION	To prevent a system crash or an unusable BIOS, do not overwrite the BIOS. The entire 1MB of Flash is accessible, but only the lower 768kB region is available for custom applications. The higher 256kB region is used for the BIOS and can be overwritten, rendering the CoreModule 410 BIOS unbootable!!

Flash Programming Requirements

To build an example application under DOS or in a Windows DOS pop-up screen, you need to have one of the following tools.

- Microsoft Visual C++ 7.0 – This is a commercial product and is available from Microsoft. It can be downloaded as part of the .NET Framework from http://msdn.microsoft.com. The compiler is part of the Microsoft .NET Framework V1.1 Software Development Kit and the NET Framework Redistributable Package V1.1. Both of these need to be downloaded and installed.
- Open Watcom C/C++32 1.1 – This is a commercial compiler product available from http://www.openwatcom.org. It is also included on the *CoreModule 410 Doc & SW CD-ROM* in the *\examples\tools\watcom* directory.
- Other versions of the above tools may also work.

The following example application is also necessary and is provided by Ampro.

- Example application – This application can be found in the *\examples\CM410\flash\watcom* or *\examples\CM410\flash\msvc* directory for the Watcom compiler or Visual C++ compiler respectively. This example application will be described in more detail later.

Building the Example

Ampro provides an example for flash programming found on the *CoreModule 410 Doc & SW CD-ROM*. The example actually consists of two parts:

- Example application – The example application shows how a C++ compiler can be used to generate a 32-bit application, which runs without an Operating System. However, the example application has to be built first since the build process of the bootloader requires the binary. The application can be built using the *make.bat* file. The *make.bat* file will build *app.exe*.
- Bootloader – The bootloader can be found in *bootsec.asm* and the final Flash image is built with *image.asm*. You can use *make.bat* to build the bootloader and Flash image (in the *\examples\CM410\flash*).

Example Assumptions

The following assumptions have been made concerning the application and certain functionality has not implemented.

- The application is located at the fixed address of 1MB
- The bootloader has to load the application at the fixed address of 1MB
- The startup code is incomplete

  For example, early initialization functions and constructors normally called before *main*, are not called at all.
- In general, the standard libraries can NOT be used
- C++ exception handling is not supported.
- The bootloader makes certain assumptions, which are documented in the source code.

Installing the Example Application

To install the example application, the generated Flash image needs to be programmed into Flash memory.

1. Copy the files *aflash.exe*, *image* and *updimg.bat* to a floppy.
2. Turn on power to the CoreModule 410 and enter BIOS Setup.
3. Go to the *Custom Configuration* screen and set **Flash Address** to [8MB].
4. Select *Esc* to exit to the main menu.
5. Exit BIOS Setup using *Write to CMOS and Exit* option.
6. Reboot the CoreModule 410 from a MS-DOS 6.22 floppy diskette, without a *config.sys* and *autoexec.bat* and then remove the diskette.
7. Insert the floppy diskette into the drive with *aflash.exe*, *image* and *updimg.bat* previously copied to it.
8. Change the current directory to the floppy, by typing **a:**
9. Run the *updimg.bat* file from the diskette.

   This bat file will program the file *image* into the Flash memory.
10. Reboot the CoreModule 410 and enter BIOS Setup again.
11. Go to the *Basic CMOS Configuration*, set **Boot 1st** to [High ROM] to boot from Flash.
12. Select *Esc* to exit to the main menu.
13. Exit BIOS Setup using the *Write to CMOS and Exit* option.

14. After system reboots from the Flash, the example application sends a message to the screen, if a color VGA compatible adapter is plugged in.

Flash Boot API

The BIOS implements an API call to assist in booting from Flash. This API allows bootloaders to call the BIOS to copy memory anywhere in the 32-bit address range. All addresses are treated as linear, physical addresses.

Function 1: Copy memory within 4GB address space

This function will copy memory anywhere in the 4GB address space. The caller has to insure A20 is turned on.

The API is called by performing an INT 15h instruction, with the registers set as follows:

```
EAX = 0A000E821h
BX = 1
ECX = bytes to copy
ESI = 32-bit flat/linear source address
EDI = 32-bit flat/linear destination address
```

The call returns with:

```
EAX = 057AA17h
```

The call uses the following registers:

```
None, except the 'hidden' part of the segment registers DS & ES.
The real mode part of DS & ES are preserved.
```

Example:

```
mov     eax, 0A000E821h
mov     bx, 1
mov     ecx, 64*1024                    ; copy 64 KB
mov     esi, (384+8)*1024*1024 + 512    ; Source address:
                                        ; 512 bytes into Flash
mov     edi, 1024*1024                  ; Destination address: 1MB
int     015h
```

Appendix A Technical Support

Ampro Computers, Inc. provides a number of methods for contacting Technical Support listed below in Table A-1. Requests for support through the Virtual Technician are given the highest priority, and usually will be addressed within one working day.

- Ampro Virtual Technician – This is a comprehensive support center designed to meet all your technical needs. This service is free and available 24 hours a day through the Ampro web site at http://ampro.custhelp.com. This includes a searchable database of Frequently Asked Questions, which will help you with the common information requested by most customers. This is a good source of information to look at first for your technical solutions. However, you must register and log in to access this service.

  Personal Assistance – You may also request personal assistance by going to the "Ask a Question" area in the Virtual Technician. Requests can be submitted 24 hours a day, 7 days a week. You will receive immediate confirmation that your request has been entered. Once you have submitted your request you can go to the "My Stuff" area and log in to check status, update your request, and access other features.

- Embedded Design Resource Center – This service is also free and available 24 hours a day at the Ampro web site at http://www.ampro.com. However, you must log in to access this service.

  The Embedded Design Resource Center was created as a resource for embedded system developers to share Ampro's knowledge, insight, and expertise gained from years of experience. This page contains links to White Papers, Specifications, and additional technical information.

Table A-1. Technical Support Contact Information

Method	Contact Information
Virtual Technician	http://ampro.custhelp.com
Web Site	http://www.ampro.com
Standard Mail	Ampro Computers, Incorporated 5215 Hellyer Avenue San Jose, CA 95138-1007, USA

Appendix B Connector Part Numbers

These connectors are used on the CoreModule 410 and can be used to determine the mating connectors, if you want to make your own cables.

Table B-1. Connector and Manufacturer's Part Numbers

Connector	Pin Number/Pin Spacing/ Orientation	Manufacturer	Manufacturer's PN
J2 – GPIO	10-pin, 0.1", right angle	Samtech	ASP-16939-02M
J3 – Serial 1	10-pin, 0.1", right angle	Molex	10-89-1106
J4 – Parallel	26-pin, 0.1", right angle	T&B Ansley or Spectra-Strip	609-2600M 812-2622-134
J5 – Utility	10-pin, 0.1", right angle	AMP or Molex	102387-1 22-55-3101
J6 – IDE	44-pin, 2mm, straight	Standard IDE connector	
J7 – Power	10-pin, 0.1", right angle	• Housing = AMP or Molex • Contact = AMP or Molex	87456-5 22-55-2101 87523-6 16-02-0103
J8 – Floppy	26-pin, 2mm, right angle	Adam-Tech or Astron	2PH2R26SGA AT-PH2-26-2-1-GF
J9 – Serial 2	10-pin, 0.1", right angle	Molex	10-89-1106
J10 – RS485	2-pin, 0.1", right angle	Molex	22-12-2024

Index