# NOTICE

**The drivers and utilities for Octagon products, previously provided on a CD, are now in a self-extracting zip file located at the Octagon Systems web site on the product-specific page. Download this file to a separate directory on your hard drive, then double click on it to extract the files. All references in this manual to files and directories on the CD now refer to files in the Utilities zip file.**

# O C T A G O N  S Y S T E M S

***Embedded PCs For Extreme Environments***

## *2133 PC/104 CPU*
## *User's Manual*

*5571 (0900)*

DiskOnChip® is a registered trademark of M–Systems. LynxOS® is a registered trademark of Lynx Real–Time Systems, Inc. Micro PC™, PC SmartLink™, Octagon Systems Corporation®, the Octagon logo and the Micro PC logo are trademarks of Octagon Systems Corporation. PICO FA™ is a trademark of Phoenix Technologies Ltd. QuickBASIC® is a registered trademark of Microsoft Corporation. QNX® is a registered trademark of QNX Software Systems Ltd. ROM–DOS™ is a trademark of Datalight. Windows 98™, Windows NT™, and Windows CE™ are trademarks of Microsoft Corporation.

Copyright 1999—Octagon Systems Corporation. All rights reserved. However, any part of this document may be reproduced, provided that Octagon Systems Corporation is cited as the source. The contents of this manual and the specifications herein may change without notice.

The information contained in this manual is believed to be correct. However, Octagon assumes no responsibility for any of the circuits described herein, conveys no license under any patent or other right, and makes no representations that the circuits are free from patent infringement. Octagon makes no representation or warranty that such applications will be suitable for the use specified without further testing or modification.

Octagon Systems Corporation general policy does not recommend the use of its products in life support applications where the failure or malfunction of a component may directly threaten life or injury. It is a Condition of Sale that the user of Octagon products in life support applications assumes all the risk of such use and indemnifies Octagon against all damage.

OCTAGON SYSTEMS®

6510 W. 91st Ave.
Westminster, CO 80030

Technical support: 303–426–4521

Telephone: 303–430–1500
FAX: 303–426–8126
Web site: www.octagonsystems.com

## IMPORTANT!

**Please read the following section before installing your product:**

Octagon's products are designed to be high in performance while consuming very little power. In order to maintain this advantage, CMOS circuitry is used.

CMOS chips have specific needs and some special requirements that the user must be aware of. Read the following to help avoid damage to your card from the use of CMOS chips.

# Using CMOS circuitry in industrial control

Industrial computers originally used LSTTL circuits. Because many PC components are used in laptop computers, IC manufacturers are exclusively using CMOS technology. Both TTL and CMOS have failure mechanisms, but they are different. Described below are some of the failures which are common to all manufacturers of CMOS equipment. However, much of the information has been put in the context of the Micro PC.

Octagon has developed a reliable database of customer–induced, field failures. The average MTBF of Micro PC cards exceeds 11 years, yet there are failures. Most failures have been identified as customer–induced, but there is a small percentage that cannot be identified. As expected, virtually all the failures occur when bringing up the first system. On subsequent systems, the failure rate drops dramatically.

- Approximately 20% of the returned cards are problem–free. These cards, typically, have the wrong jumper settings or the customer has problems with the software. This causes frustration for the customer and incurs a testing charge from Octagon.
- Of the remaining 80% of the cards, 90% of these cards fail due to customer misuse and accident. Customers often cannot pinpoint the cause of the misuse.
- Therefore, 72% of the returned cards are damaged through some type of misuse. Of the remaining 8%, Octagon is unable to determine the cause of the failure and repairs these cards at no charge if they are under warranty.

The most common failures on CPU cards are over voltage of the power supply, static discharge, and damage to the serial and parallel ports. On expansion cards, the most common failures are static discharge, over voltage of inputs, over current of outputs, and misuse of the CMOS circuitry with regards to power supply sequencing. In the case of the video cards, the most common failure is to miswire the card to the flat panel display. Miswiring can damage both the card and an expensive display.

- Multiple component failures: The chance of a random component failure is very rare since the average MTBF of an Octagon card is greater than 11 years. In a 7 year study, Octagon has never found a single case where multiple IC failures were not caused by misuse or accident. It is very probable that multiple component failures indicate that they were user–induced.

- Testing "dead" cards: For a card that is "completely nonfunctional", there is a simple test to determine accidental over voltage, reverse voltage or other "forced" current situations. Unplug the card from the bus and remove all cables. Using an ordinary digital ohmmeter on the 2,000 ohm scale, measure the resistance between power and ground. Record this number. Reverse the ohmmeter leads and measure the resistance again. If the ratio of the resistances is 2:1 or greater, fault conditions most likely have occurred. A common cause is miswiring the power supply.

- Improper power causes catastrophic failure: If a card has had reverse polarity or high voltage applied, replacing a failed component is not an adequate fix. Other components probably have been partially damaged or a failure mechanism has been induced. Therefore, a failure will probably occur in the future. For such cards, Octagon highly recommends that these cards be replaced.

- Other over–voltage symptoms: In over–voltage situations, the programmable logic devices, EPROMs and CPU chips, usually fail in this order. The failed device may be hot to the touch. It is usually the case that only one IC will be overheated at a time.

- Power sequencing: The major failure of I/O chips is caused by the external application of input voltage while the Micro PC power is off. If you apply 5V to the input of a TTL chip with the power off, nothing will happen. Applying a 5V input to a CMOS card will cause the current to flow through the input and out the 5V power pin. This current attempts to power up the card. Most inputs are rated at 25 mA maximum. When this is exceeded, the chip may be damaged.

- Failure on powerup: Even when there is not enough current to destroy an input described above, the chip may be destroyed when the power to the card is applied. This is due to the fact that the input current biases the IC so that it acts as a forward biased diode on powerup. This type of failure is typical on serial interface chips.

- Hot insertion: Plugging cards into the card cage with the power on will usually not cause a problem. (Octagon urges that you do not do this!) However, the card may be damaged if the right sequence of pins contacts as the card is pushed into the socket. This usually damages bus driver chips and they may become hot when the power is applied. This is one of the most common failures of expansion cards.
- Terminated backplanes: Some customers try to use Micro PC cards in backplanes that have resistor/capacitor termination networks. CMOS cards cannot be used with termination networks. Generally, the cards will function erratically or the bus drivers may fail due to excessive output currents.
- Excessive signal lead lengths: Another source of failure that was identified years ago at Octagon was excessive lead lengths on digital inputs. Long leads act as an antenna to pick up noise. They can also act as unterminated transmission lines. When 5V is switched onto a line, it creates a transient waveform. Octagon has seen submicrosecond pulses of 8V or more. The solution is to place a capacitor, for example 0.1 µF, across the switch contact. This will also eliminate radio frequency and other high frequency pickup.

## ≡ Avoiding damage to the heatsink/CPU

WARNING!
When handling any Octagon CPU card, extreme care must be taken not to strike the heatsink (if installed) against another object, such as a table edge. Also, be careful not to drop the CPU card, since this may cause damage to the heatsink/CPU as well.

Note Any physical damage to the CPU control card is not covered under warranty.

# Contents

# List of figures

# List of tables

# Abbreviations and terms used in this manual

Throughout this manual, the following symbols and terms are used:

| | |
|---|---|
| Autoexecution | Automatic execution of a program on powerup or reset. |
| BIOS | Basic Input Output System. |
| Console port | Video card or COM1 where BIOS and DOS messages appear and keyboard input is available. |
| DRAM | Dynamic Random Access Memory devices. |
| Expansion card | The expansion cards add I/O functions to the PC/104 system, such as analog input/output, digital input/output, motion control, and display. |
| Flash | Electrically erasable PROM which allows at least 100,000 write cycles. |
| h | The suffix "h" denotes a hexadecimal number. A decimal number has no prefix or suffix. For example, 1000h and 4096 are equivalent. |
| KB | Kilobyte (1,024 8–bit bytes). |
| MB | Megabyte (1,048,576 8–bit bytes). |
| Memory device | The type of static RAM, DRAM, flash memory, or EPROM specified for either volatile or nonvolatile memory. |
| PC/104 expansion | An expansion bus used for holding 8– and 16–bit expansion cards. |
| PC SmartLINK | A serial communications software package designed by Octagon for use with the 2133 PC/104 CPU. |
| Solid–state disk (SSD) | A simulated disk which uses a high speed solid–state memory device. For example, flash memory, EEPROM, or static RAM. |
| SRAM | Static Random Access Memory. |
| TTL compatible | Transistor transistor logic compatible; 0–5V logic levels. |
| W[ – ] | Denotes a jumper block and the pins to connect. |
| XMODEM | A communications protocol which allows transfer of files between two computers. |

# Conventions used in this manual

## ≡ Notes and warnings

Note A note is supplementary or background information. At other times, it is a hint or reminder that makes a task easier or quicker.

> WARNING!
>
> A warning gives vital information. Failure to heed a warning may cause system failure, equipment damage, or bodily harm to the system operator.

## ≡ Command format and procedures

Commands must be entered in a specific format. To indicate the format, this manual uses the conventions below. The conventions cover the rules for issuing all commands, including the most complex ones. The command format looks like this:

**command** [**type_this** | **or_ this**] *input* {*optional_input*}

Follow these rules and conventions:

- Information which appears on your screen is shown in a different type face, for example:

  ```
  PhoenixBIOS(TM) A486 Version 1.03
  Copyright (C) 1985-1994 Phoenix Technologies, Ltd.
  All Rights Reserved
  Octagon Systems Corp.
  ```

- Commands that you must key in are shown in **Courier Bold**, for example:

  C:> **RESET**

- Italicized refers to information that is specific to your particular system or program, e.g.,

  Enter *filename* means enter the name of your file.

- Paired angle brackets are used to indicate a specific key on your keyboard, e.g., <ESC> means the escape key; <CTRL> means the control key; <F1> means the F1 function key.

- All addresses are given in hexadecimal, for example, 328h.

# About this manual

The 2133 PC/104 CPU user's manual provides information about installing and configuring the 2133 PC/104 CPU. This manual is divided into four sections:

- Section 1 — Installation

  Chapter 1: Overview
  Chapter 2: Quick start
  Chapter 3: SETUP programs
  Chapter 4: Save and run programs

- Section 2 — Hardware

  Chapter 5: Serial ports
  Chapter 6: LPT1 parallel port
  Chapter 7: Console devices
  Chapter 8: SSDs, DRAM, and battery backup
  Chapter 9: External drives
  Chapter 10: Interpreting "beep" codes
  Chapter 11: PC/104 expansion

- Section 3 — System management

  Chapter 12: Watchdog timer and hardware reset
  Chapter 13: CPU power management
  Chapter 14: Serial EEPROM
  Chapter 15: Troubleshooting

- Section 4 — Appendices

  Appendix A: Technical data
  Appendix B: Software utilities
  Appendix C: Accessories

# *Overview:* *Section 1 – Installation*

Section 1 provides installation and programming instructions, startup options, and system configuration program examples. The following chapters are included:

Chapter 1: Overview
Chapter 2: Quick start
Chapter 3: SETUP programs
Chapter 4: Save and run programs

# *Chapter 1:* *Overview*

## ≡ Description

The 2133 is a high performance PC/104 CPU with a low–voltage 586DX 133 MHz CPU. The 2133 PC/104 CPU integrates serial communications, a multifunctional parallel port, a solid–state disk, keyboard and speaker ports, and a PC/104 interface.

This board will execute operating systems such as DOS 6.22, LynxOS®, Windows 98™, Windows NT™, Windows CE™, and QNX®. Since the 2133 PC/104 CPU uses the same functional blocks as the Micro PC™, the circuitry has been fully proven as reliable and the software is compatible with the software in the Micro PC series.

## ≡ 2133 PC/104 CPU major hardware features

### CPU

The CPU is a high–performance, low–voltage 586DX with a clock speed of 133 MHz.

### Solid–state disk

#### SSD0

The 2133 PC/104 CPU has an on–board 512 KB flash which contains the BIOS and power management software extensions.

#### SSD1

A 32–pin DIP socket accepts either an M–Systems DiskOnChip® or a 512 KB SRAM. SRAM is automatically backed up when an AT battery is connected. The socket exhibits high retention force and affords a gas tight contact.

## RAM

The 2133 PC/104 CPU has an on–board 4 MB DRAM. It can access up to 64 MB of DRAM installed in a single SO–DIMM socket. All DIMM modules from Octagon meet the full 2133 PC/104 CPU temperature range.

## Serial ports protected against ESD

The 2133 PC/104 CPU has two serial ports for 8–wire RS–232C interfaces. COM1 can also be used as a console interface. These serial ports have the following common specifications:

- IEC1000, level 3, ESD protection specification
  - — Contact discharge ±6 kV
  - — Air–gap discharge ±8 kV
- Backdrive protection
- 16C550 compatible
- Up to 115.2K baud
- 16–byte FIFO buffers
- Enabled and disabled in SETUP

## Speaker, keyboard, and mouse

A speaker and keyboard can be connected to the utility port. If a mouse is needed, it can be connected to a COM port. The keyboard controller accepts an AT style keyboard. Alternately, COM1 can be used as a console port, and all keyboard and video information is redirected through COM1. Neither the keyboard nor the mouse are required for operation.

## PC/104 16–bit interface

The PC/104 interface accepts an 8– or 16–bit PC/104 expansion board. PC/104 expansion boards are available from several manufacturers. As many as three PC/104 expansion boards may be stacked on the 2133 PC/104 CPU.

# ≡ 2133 PC/104 CPU major software features

## Advanced power management and system management input

Power management can be used to reduce power consumption or to freeze the state of the program on the occurrence of a power management interrupt. Power consumption can be reduced by more than 60%. This reduces the heat load and extends the battery life in mobile applications. A system management input (SMI) will cause the 2133 PC/104 CPU to save its current status. When the system is signaled to restart, execution resumes at the point the system last stopped. The following are functions in the advanced power management and SMI:

- Suspend/resume software — halts CPU
- Wake up through various interrupts including keyboard, SMI and serial port
- Slow CPU by duty cycling clock
- Contextual save to disk.

## Diagnostic software verifies system integrity automatically

The 2133 PC/104 CPU has built–in diagnostic software that can be used to verify on–card I/O and memory functions. On powerup, a series of tests is performed. If a problem occurs, the failed test can be identified by a series of beeps. The test is performed automatically every time the system is reset or powered up. Memory verification does not require software, test equipment, monitor, keyboard, disks, or test fixtures. See the Interpreting "beep" codes chapter for a complete listing of tests and failures and their descriptions.

## SETUP information stored in EEPROM for high reliability

Loss of SETUP data is serious in industrial applications. Most PCs store SETUP information in battery–backed CMOS RAM. If the battery fails or is replaced during routine maintenance, this information is lost. Without a keyboard and monitor in embedded applications, time consuming re–initialization is required.

The 2133 PC/104 CPU stores the SETUP information in EEPROM. If a backup battery should fail, only the system date and time are lost.

## Phoenix BIOS

The 2133 PC/104 CPU has a Phoenix AT BIOS with power management and Octagon BIOS extensions.

## Boot sequence

A 2133 PC/104 CPU can be configured to boot from an on–card, solid–state disk, an external floppy, or a hard disk.

## Multifunctional printer port

The 2133 PC/104 CPU incorporates the latest enhanced parallel port. It includes the following features:

- Unidirectional
- Bidirectional
- IEEE 1284, ECP and EPP modes
- 14 mA of drive current
- Backdrive protection
- Floppy drive mode

The following represent applications in the multifunctional parallel port:

- LPT1 for PC compatible printers
- 17 general purpose digital I/O lines
- Up to a 4 x 4 matrix keypad
- 4–line alphanumeric display
- MPB–16PC, 16–position opto–module rack

## Watchdog timer added for safety

The watchdog timer resets the system or generates an NMI (nonmaskable interrupt) if the program stops unexpectedly. The watchdog is enabled, disabled and strobed under software control. The time–out period is 1.6 seconds typical, 1.00 seconds minimum to 2.25 seconds maximum. The watchdog timer can be strobed by the I17HNDLR.EXE utility (a TSR program) to extend the timeout up to a maximum of 60 hours.

## Real time calendar/clock with battery–backup

The real time clock is fully AT compatible and uses the standard DOS calls. An optional off–card battery powers the real time clock when the 5 volt supply is removed.

## Keypad and LCD display support for low cost operator interface

For embedded applications, the parallel printer port can interface with a 16–key matrix keypad and a 2– or 4–line LCD display. The 2133 PC/104 CPU is supplied with software that provides keypad scanning and display operation. Supplied display and keypad drivers in C and Basic support these devices.

## Hardware reset

A hardware reset ensures complete reset of the system and all attached peripherals. A hardware reset can be done by any of the following methods:

- An expired watchdog timer cycle
- Cycling power
- Momentarily pulling the master reset line to ground (pin 3, connector J5)

## 5 Volt operation lowers system cost

The 2133 PC/104 CPU operates from a single 5V ±5% supply.

- 5V ±5%, approximately 930 mA (typical), 510 mA in standby
- +12V, –12V, and –5V supplied to PC/104 connector; not required for 2133 PC/104 CPU operation

## Rugged environmental operation

| | |
|---|---|
| ■ Operating temperature | –40° to 70°C |
| ■ Nonoperating temperature | –55° to 90°C, nonoperating |
| ■ Relative humidity | 5% to 95% noncondensing |
| ■ Altitude | –100m to 10,000m |
| ■ Shock | 40g, 3 axis |
| ■ Vibration | 6g, 3 axis |

## Size

3.8" x 3.6"

# *Chapter 2:* *Quick start*

This chapter covers the basics of setting up a 2133 PC/104 CPU system and tells you:

- How to set the configuration jumpers on the 2133 PC/104 CPU
- How to install the 2133 PC/104 CPU
- How to connect a power supply
- How to install a bootable disk
- How to use a serial console

> WARNING!
>
> The 2133 PC/104 CPU contains static sensitive CMOS components. Do the following to avoid damaging your card and its components:
>
> - Ground yourself before handling the 2133 PC/104 CPU card
> - Disconnect power before removing or inserting a PC/104 expansion board
> - When programming a memory device, place the device in the socket before applying power.

## ≡ Equipment required

There are several options for installing a bootable disk and loading your applications. This chapter provides procedures for installing an M–Systems DiskOnChip and using a floppy disk drive to transfer files to the DOC. The following equipment is required:

- +5V power supply
- VTC–9F, FCA–12, and null modem cables
- floppy disk drive
- M–Systems DiskOnChip, preformatted and loaded with an operating system

# ≡ Configuration jumpers

Before you continue with the installation of your 2133 PC/104 CPU, review the following tables for a list of jumper configurations to ensure you have the correct configuration.

The 2133 PC/104 CPU component diagrams are on the following page.

Table 2–1 SSD1 device configuration: W1

| Pins | Description |
|---|---|
| W1[1–3, 2–4, 6–8, 7–9] | SRAM with battery backup |
| W1[1–2, 3–4, 5–6, 9–10]* | Flash (DiskOnChip) |

* = default

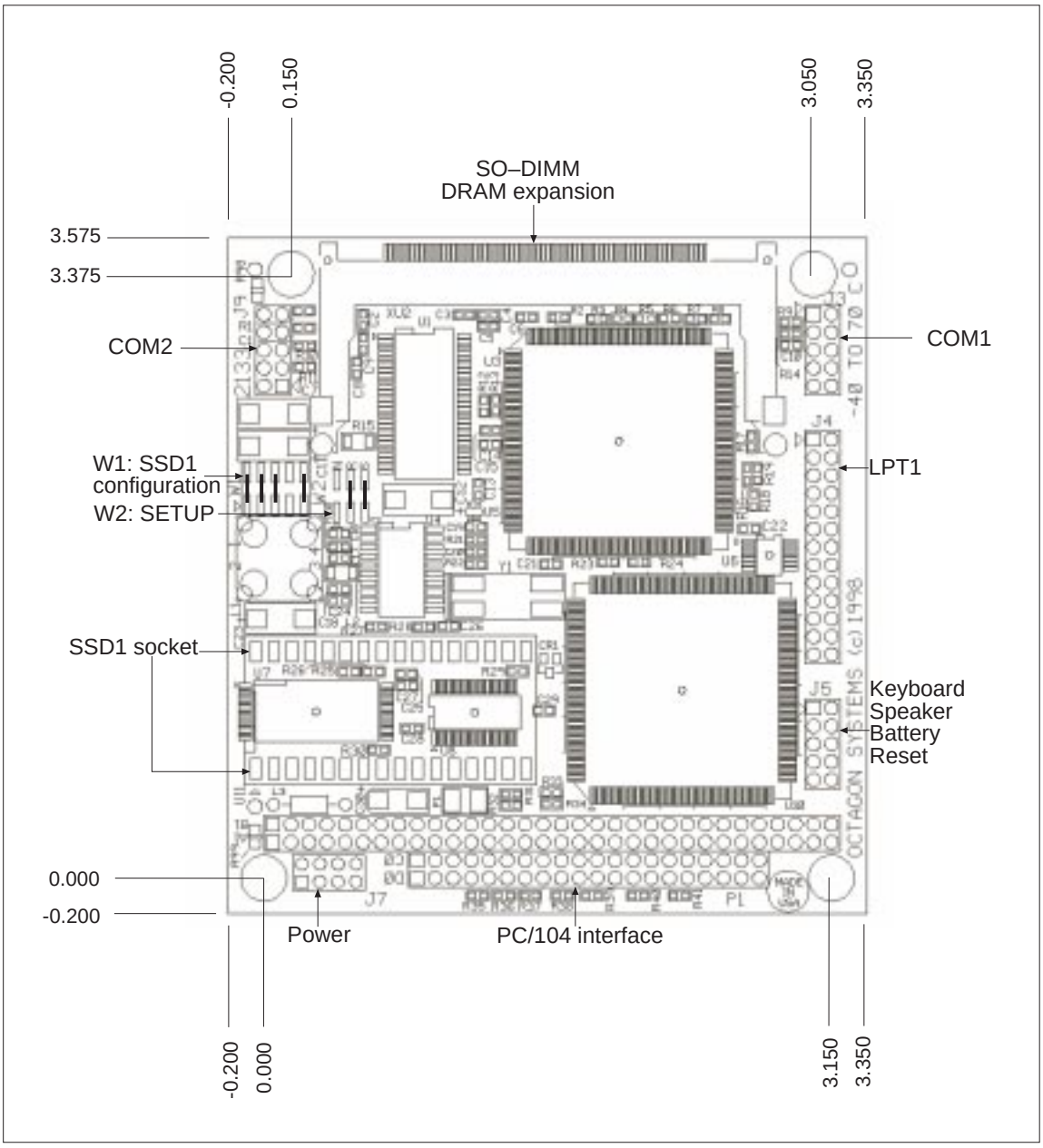
Table 2–2 SETUP: W2

| Pins | Description |
|---|---|
| 1–2 | (N)MI: Watchdog time–out connected to NMI |
| 3–4* | (R)eset: Watchdog time–out connected to reset |
| 5–6* | (S)etup: Use SETUP information:<br>On* = on–board EEPROM; Off = BIOS default |

* = default

Note When the "N" jumper W2[1–2] is on instead of the default "R" jumper W2[3–4] , the system will not reset after SETUP is run. A power cycle reset is required.

*Figure 2–1 Component locations and mounting dimensions*



# ≡ Installing the 2133 PC/104 CPU board

The 2133 PC/104 CPU can be installed two ways – mounted on a flat surface, or connected to a PC/104 interface. The following equipment is included with the 2133:

- 4 threaded hex stainless steel standoffs (4–40 x 3/8")
- 4 screws (4–40 x 1/4")
- 4 cap nuts (4–40)
- 4 internal star lock washers (4–40)

## Mounting on a flat surface

To mount the 2133 PC/104 CPU on a flat surface, use the #4 standoffs, washers, and screws and place them in the four holes on the 2133 PC/104 CPU board. Refer to Figure 2–1 for the center–to–center mounting hole dimensions and for the location of the designated holes used for mounting the hardware. Fasten the board securely to the mounting surface.

> WARNING!
>
> Before the 2133 PC/104 CPU is powered on for bench testing, all four standoffs, screws and washers should be secured to the board. The standoff pieces will ensure full support of the 2133 PC/104 CPU board. These hardware pieces will reduce the circuit board flex when a PC/104 expansion board and/or the SSD1 are inserted. Flexing of the 2133 PC/104 CPU board should be avoided, since it can cause problems with the copper circuit traces, keyboard, monitor, and surface mounted components.

## Connecting to a PC/104 interface

The 2133 PC/104 CPU can be connected to an existing PC/104 system. Use the #4 standoffs, washers, and screws and place them in the four holes on the 2133 PC/104 CPU board, with the male end of the stand-off facing down. Align the PC/104 pins with the connectors below, and the standoffs with the mounting holes below, then firmly press the boards together. Secure the boards together with the cap nuts.

# ≡ Power supply

The 2133 PC/104 CPU requires 5V ±5%, approximately 930 mA (typical). It is designed to operate from a single +5 VDC supply, typically supplied at connector J7. In some configurations where the 2133 is connected to another PC/104 system, the power required by the 2133 can be drawn from the system through the PC/104 connector. If you are using the PC/104 interface, you may also require ±12 VDC. Make sure that you utilize both +5 VDC conductors and both ground conductors. Refer to Figure 2–2 and Table 2–3.

> WARNING!
>
> Accidently crossing the wires, i.e., plugging +5V wires into the ground connector or the ground wires into the +5V connector will damage the 2133 PC/104 CPU.

The maximum current requirements for the 2133 PC/104 CPU is approximately 1.5A. It is important that a quality power supply be used with the 2133 PC/104 CPU. For example, when a particular application calls for a custom power supply, there are several internal issues to consider such as current capacity, line and load regulation, maximum ripple, hold up time, efficiency, and current limiting. You should also consider the power devices and equipment such as the power cable conductor gauge, number and length of conductors, mating connectors, and the power supply to external PC/104 devices.

The proper selection of a quality power supply ensures reliability and proper functioning of the 2133 PC/104 CPU.

> WARNING!
>
> Make sure the power supply is OFF when connecting the power cable to the 2133 PC/104 CPU board. Damage to the 2133 PC/104 CPU may occur if the power is ON when connecting the power cable.

Figure 2–2 Power connector

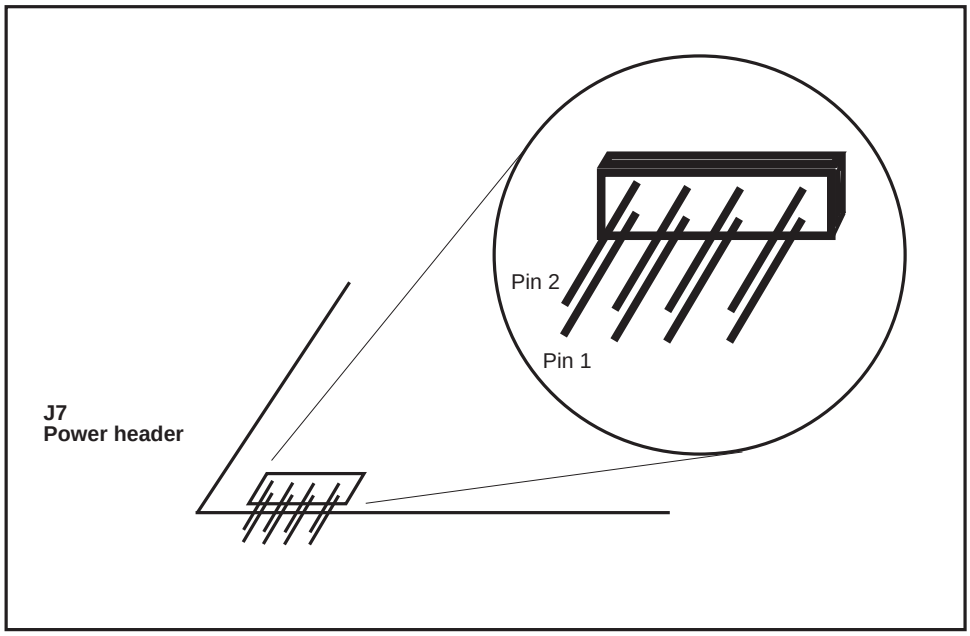



Table 2–3 Power pinout: J7

| Pin | Function |
|---|---|
| 1 | Gnd |
| 2 | +5 VDC |
| 3 | NC |
| 4 | +12 VDC to PC/104 |
| 5 | –5 VDC to PC/104 |
| 6 | –12 VDC to PC/104 |
| 7 | Gnd |
| 8 | +5 VDC |

# ≡ Bootable disks

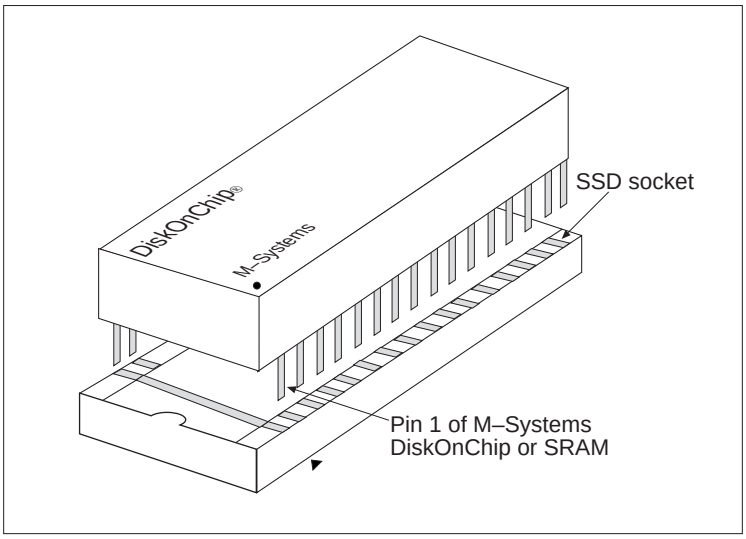
The 2133 PC/104 CPU has an on–board 512 KB flash which contains the BIOS and power management software extensions. It does not contain an operating system or a bootable disk.

There are three choices for supplying a bootable disk: a floppy drive (through LPT1 or a PC/104 card), a PC/104 hard drive, or an M–Systems DiskOnChip (DOC). The example in this chapter shows how to use a DOC to boot your system and a floppy drive connected to LPT1 to install your applications.

When ordering an M–Systems DiskOnChip, you can specify that you want it preformatted and with an operating system already installed. This simplifies your startup procedures. Refer to your M–Systems utility disk for more information on using DiskOnChip.

Before installing the DOC, remove power from the 2133. Make sure the SSD socket is configured as flash W1[1–2, 3–4, 5–6, 9–10]. Align pin one of the DiskOnChip with pin one of the SSD1 socket, then firmly press the chip into the socket. Refer to Figure 2–3 for correct SSD alignment.

Figure 2–3 M–Systems pin alignment with the 2133 SSD1 socket

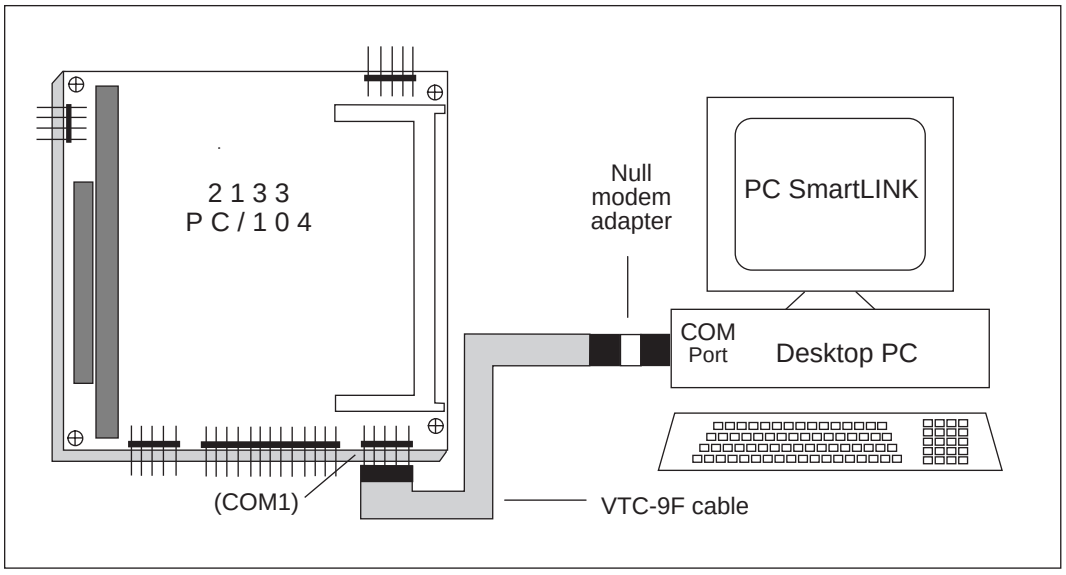


# ≡ Serial console

COM1 can serve as a console device, allowing you to use your desktop PC to communicate with the 2133. The default configuration stored in the BIOS SETUP is for COM1 to be a console device. Use the figure below to connect a desktop computer to the 2133 as a console device. This connection requires a VTC–9F cable and a null modem adapter. It also requires PC SmartLINK, available from Octagon, or an equivalent terminal emulator.

Note When interfacing the 2133 PC/104 CPU to your desktop PC, you must use a null modem adapter.

Figure 2–4 Serial console connections




Power up the 2133. You should see the BIOS sign–on messages for the 2133 on your monitor. If the boot process completes and the C:\> prompt appears, you have a bootable disk. Go to the Floppy disk section for information on transferring files.

If the message "press C to try again or S for setup" appears, the boot did not complete. The most likely reason is that the DOC does not have an operating system installed. Press S to enter Setup, then configure LPT1 as a floppy drive, number of floppy drives to 1, and floppy drive size to the size of the floppy you will be installing. Save the Setup changes, then power off the 2133. Ensure that the S jumper is on so that the next time the 2133 is powered, it will use the parameters from the Setup stored in EEPROM. Proceed to the next section, Floppy drive.

# ≡ Floppy drive

This section describes how to install a floppy drive. The floppy drive can be used as a boot device, or to transfer programs to a DOC.

Note If you have a DOC installed on the 2133 and have booted, you may have the utilities TRANSFER.EXE or REMDISK/REMSERVE (ROM DOS) or INTERLINK/INTERSVR (MS DOS) on the DOC. You can use these utilities to transfer files from your host computer to the 2133 over the serial console. Refer to the documentation included with your operating system for information on how to use these utilities.

## Configuring setup

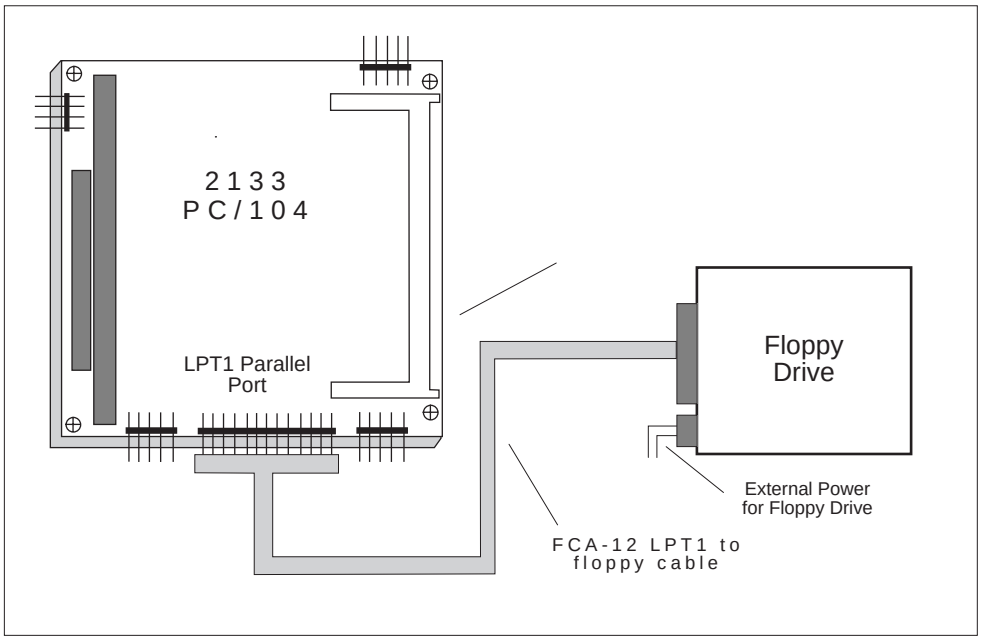
The default in the BIOS SETUP is for LPT1 in bidirectional mode. If you have previously run SETUP and changed the settings for LPT1 to floppy drive, and saved those settings, ensure that the SETUP jumper is on so that the BIOS uses the settings from the EEPROM.

You can enter SETUP by pressing the "backspace" key followed by the "S" key during BIOS POST sequence (this occurs between the memory test and boot). Configure LPT1 as a floppy drive, number of floppy drives to 1, and floppy drive size to the size of the floppy you will be installing. Save the Setup changes, then power off the 2133. Ensure that the SETUP jumper is on so that the BIOS uses the settings from the EEPROM during the next boot.

## Installing a floppy disk drive

1. Disconnect power to the 2133 PC/104 CPU.
2. Insert one end of an Octagon FCA–12 cable into the rear of the floppy drive. Make sure pin 1 on the cable is connected to pin 1 on the drive.
3. Insert the other end of the cable into J4 on the 2133 PC/104 CPU.
4. Connect power to the floppy drive.
5. Power up the system. If you have a bootable disk in SSD1, the system should boot up. If you do not have a bootable disk in SSD1, you will need to insert a boot disk into the floppy disk drive. Refer to the next section for information on creating a boot disk.

Figure 2–5 LPT1 and a Floppy Drive



### Creating a boot disk

You can create a floppy boot disk on any PC that has DOS installed. This floppy can then be inserted in the floppy disk drive connected to the 2133 and used to boot the system.

To create a boot disk, insert a blank, formatted floppy into the floppy drive of your PC. If you are running Windows on your PC, exit Windows and open up the DOS shell. Type the command **SYS A:**, then press Enter.

The SYS command copies three files to the floppy disk. Two of the files are hidden (not shown by the DIR command), and the third file is COMMAND.COM. These three files are sufficient to boot the system.

Insert the boot disk in the 2133 floppy drive, then power on the system. The console monitor should return the A:\ prompt. Type **CD C:** to change to the C drive (if you have installed a DiskOnChip). Type **DIR** then press Enter.

If the CD C:\ or DIR commands return the message "No drive found" or "Abort Retry Fail", the DiskOnChip is probably not formatted. Refer to your M–Systems documentation for information on how to format the DOC.

If the DIR command returns a list of files from the DOC, and the list does not contain the file COMMAND.COM, then the DOC most likely does not contain an operating system. Return to the A: drive, type the command **SYS C:**, then press Enter. This should copy the boot files to the DOC. Remove the floppy disk from the floppy disk drive and cycle power to the 2133. The 2133 should boot from the DOC.

## ≡ Copying files

Once you have a bootable disk in SSD1, or a bootable PC/104 hard drive, you can use the floppy disk drive to transfer files to the bootable disk. Use the COPY command to copy files (such as, **COPY MYPRGRM.COM C:**). Once the files are copied, you can remove the floppy disk drive.

When DOS starts up, it processes the files CONFIG.SYS and AUTOEXEC.BAT. If you add the command line for your application to the AUTOEXEC.BAT file, your application will automatically start when the system boots. Refer to your operating system documentation for information on modifying these two files.

# Chapter 3: SETUP programs

## Introduction

Two system configuration programs exist for the 2133 PC/104 CPU. They are:

- PMISETUP — Configures power management options at a more detailed level than SETUP
- SETUP — Configures devices set up by the BIOS such as serial ports and floppy drives

## PMISETUP

PMISETUP allows the user to customize the power management features of the 2133 PC/104 CPU. Refer to the CPU power management chapter. See also the Software utilities appendix for details.

## SETUP

The SETUP program defines the 2133 PC/104 CPU system parameters. This program is shipped with default configuration parameters stored in the serial EEPROM. Changes are made by running the SETUP program. The SETUP program is stored on the 2133 utility disk.

SETUP can be entered in one of two ways:

- Run SETUP.COM. To run SETUP.COM, this program must be resident on either the floppy disk drive and/or hard drive, or on a flash device such as the M–Systems DiskOnChip. SETUP.COM is on the 2133 utility disk.
- Press the "backspace" key followed by the "S" key during BIOS POST sequence (this occurs between the memory test and boot).

The system will display the 2133 setup parameters and available options. Select the option by pressing the space bar until the correct information appears, then press <ENTER>. Press <ESC> twice if you want to exit setup without saving your responses.

By removing the SETUP jumper from the "S" position at W2[5–6], you force the setup to revert to the defaults stored in BIOS. The default settings allow you to restart the system in a known configuration.

Table 3-1 SETUP Jumpers: W2

| Pins | Description |
|---|---|
| 1–2 | (N)MI: Watchdog time–out connected to NMI |
| 3–4* | (R)eset: Watchdog time–out connected to reset |
| 5–6* | (S)etup: Use SETUP information:<br>On* = on–board EEPROM; Off = BIOS default |

* = default

Note Options having an * are default settings.

- Serial Console on COM1:
  Enabled*
  Disabled

Note When this option is disabled, it may be overridden by connecting the COM1 RTS line to the COM1 RI line.

- COM1 Console Baud Rate:
  1200
  2400
  4800
  9600*
  14400
  19200
  28800
  38400
  57600
  115200

- Power on memory test:
  Enabled*
  Disabled

You may want to disable the memory test to speed up the boot process. You may also press the space bar to cancel the memory test while in progress.

- Boot Sequence:
  C: Only*
  A: Then C:

Note The M–Systems DiskOnChip software typically overrides the "C: Only" option.

- Serial Port A:
  Enabled*
  Disabled

- Serial Port A address:
  3F8H*
  2F8H
  338H
  3E8H
  2E8H
  220H
  238H
  2E8H
  2E0H
  228H

- Serial Port B:
  Enabled*
  Disabled

- Serial Port B address:
  2F8H*
  338H
  3E8H
  2E8H
  220H
  238H
  2E8H
  2E0H
  228H

Note We strongly recommend 3F8H for COM1 and 2F8H for COM2.

Note The choices for COM port addresses are mutually exclusive; therefore, you may not see all the choices listed above.

- Parallel (LPT) Port:
  Enabled*
  Disabled

- Parallel Port Mode:
  Bidirectional mode*
  EPP mode
  ECP mode
  Floppy disk mode
  Standard (Unidirectional) mode

- Parallel Port Address:
  378h*
  278h
  3BCh

- Parallel Port Interrupt:
  IRQ5
  IRQ7*

Note Standard mode is provided for compatibility only. We recommend the use of bidirectional mode. EPP and ECP modes are provided for equipment that has the capability to operate in these modes for enhanced performance.

- Number of floppy drives:
  0*, 1, 2
- Floppy drive 1 size:
  5.25", 360KB
  5.25", 1.2 MB
  3.5", 720KB
  3.5", 1.44 MB*
- Floppy drive 2 size:
  5.25", 360KB
  5.25", 1.2 MB
  3.5", 720KB
  3.5", 1.44 MB*
- Number of hard drives:
  0*, 1, 2
- Auto drive configuration:
  Enabled*
  Disabled
- Drive 0 parameters:
  Cylinders (xxx):
  Heads (x):
  Sectors (xx):
- Drive 1 parameters:
  Cylinders (xxx):
  Heads (x):
  Sectors (xx):
- Internal CPU cache:
  Enabled*
  Disabled
- Limit CPU to half speed:
  No*
  Yes
- Setup entry via hotkey:
  Enabled*
  Disabled
- Power Management:
  Enabled
  Disabled*
- Doze clock:
  Slow*
  Stop
- Time update after Suspend:
  Enabled*
  Disabled
- DIMM module type:
  EDO*
  Standard

- Shadow video BIOS:
  Enabled
  Disabled*

- Shadow C8000H - CFFFFH
  Disabled*
  Enabled

- Shadow D0000H - D7FFFH
  Disabled*
  Enabled

- Shadow D8000H - DFFFFH
  Disabled*
  Enabled

```
Press ENTER to SAVE the changes or
Press ESC to EXIT without saving the changes.
Saving options.
Options saved.
```

Depending on the options you have selected, the system may display the following message:

```
You must reset for these options to take effect.
```

If you entered SETUP with the hotkeys (i.e., "backspace" and "S" keys), the system will reboot automatically. However, if the watchdog jumper is set to NMI, then the I/O parity check error appears and the board will halt. A power cycle reset is then required.

## Running SETUP over the console port

1. To run SETUP make sure you have established a communications link between a keyboard and monitor (via a PC/104 video card) with the 2133 PC/104 CPU or a serial communications link between the 2133 PC/104 CPU and your PC. See the Console devices chapter for more information on these two communication links. Ensure the SETUP.EXE file has been copied from the 2133 utility disk to the boot drive, or that a floppy drive is connected to the system.

2. Enter:

   C:\> **SETUP**

Note If you are not booting from the SSD1 drive, the drive designator may differ.

Note You may also enter SETUP after the memory test and before the system has booted by pressing the "backspace" key followed by the "S" key.

3. The system will display the 2133 PC/104 CPU setup parameters and available options. Select the option by pressing the space bar until the correct information appears, then press <ENTER>. Press <ESC> twice if you want to exit SETUP without saving your responses.

## SETUP example

The following example configures a system with no memory test, 9600 baud, printer at 378h, and booting from C:

```
                OCTAGON SYSTEMS CORPORATION
             2133 PC/104 CPU SETUP UTILITY Vx.x
          (c) Phoenix Technologies, Ltd. 1985, 1995
 ____________________________________________________________
(Press SPACE to CHANGE, ENTER to ACCEPT, ESC to EXIT)


Serial Console on COM1:          Enabled
COM1 Console Baud Rate:          9600
Power on memory test:            DISABLED
Boot Sequence:                   C: ONLY
Serial Port COM1:                ENABLED
 Serial Port COM1 Address:       3F8h (default)
Serial Port COM2:                ENABLED
 Serial Port COM2 Address:       2F8h (default)
Parallel (LPT) Port:             ENABLED
    Parallel Port Mode:          Bidirectional Printer
                                 Port
    Parallel Port Address:       378h
Number of floppy drives:         1
Floppy drive 1 size:             3.5", 1.44 MB
Number of hard drives:           0
Internal CPU cache:              ENABLED
Limit CPU to half speed:         NO
SETUP Entry via Hotkey:          ENABLED
Power Management:                ENABLED
 Doze Clock (slow, stop):        SLOW
 Time Update after Suspend:      ENABLED
DIMM Module Type:                EDO
Enable ROM at C0000h-C7FFFh:     ENABLED
Enable ROM at C8000h-CFFFFh:     ENABLED
Enable ROM at C8000h-DFFFFh:     ENABLED
Shadow D0000h-C7FFFh:            ENABLED
Shadow C8000h-CFFFFh:            ENABLED
Shadow D0000h-D7FFFh:            DISABLED
Shadow D8000h-DFFFFh:            DISABLED

Press ENTER to SAVE the changes
Press R to RESTART with original values or
Press ESC to EXIT without saving the changes:

Options Saved.

You must reset for these options to take effect.
PC2133 C:\>
```

Note Executing SETUP /D will change all setup parameters to default values.

# *Chapter 4:* *Save and run programs*

## ≡ Save and run your programs on the 2133 PC/104 CPU

Once you have written, tested and debugged your application, you can then save it in SSD1 if you have a DOC, or to a PC/104 hard disk. When you reboot the 2133 PC/104 CPU, your program can automatically load into DOS memory and execute. This assumes SSD1 contains a bootable DOS.

This chapter describes the following:

- Saving an application program
- Autoexecuting the program from the 2133 PC/104 CPU
- Overriding autoexecution of your program

The information in this chapter assumes you are using ROM–DOS or MS–DOS in your application. Some Microsoft programs make undocumented DOS calls. With ROM–DOS, an error returns when an undocumented DOS call is made, causing your program to operate erratically.

### Adding your application

To add your application to your SSD or hard disk, do the following:

1. Depending on the operating system you have installed, you may have the utilities TRANSFER.EXE or REMDISK/REMSERVE (ROM–DOS) or INTERLINK/INTERSVR (MS–DOS) on the DOC. You can use these utilities to transfer files from your host computer to the 2133 over a serial console. Refer to the documentation included with your operating system for information on how to use these utilities.

   From a floppy drive on the 2133 PC/104 CPU, issue the COPY command.

2. Add or remove any device drivers for your application. You may want to do the same for the CONFIG.SYS file on your bootable drive.

3. To autoexecute your application, add your application name to the AUTOEXEC.BAT file.

## Overriding the autoexecution of your application

You may stop the autoexecution of your application by doing one of the following options:

### Option 1

1. Press F5 or F8 on a local keyboard (connected to the 2133). For more information, see your ROM–DOS or MS–DOS manual.

### Option 2

1. Press Ctrl-C when the system is first starting. This halts all batch files.
2. Change AUTOEXEC.BAT and/or CONFIG.SYS to not call out your program.

### Option 3

1. Install a floppy.
2. Change SETUP option "Boot Sequence" to "A: THEN C:"
3. Change SETUP to enable the floppy.
4. Boot from floppy.
5. Change AUTOEXEC.BAT on C:.

# *Overview:* *Section 2 – Hardware*

Section 2 discusses usage, functions, and system configurations of the 2133's major hardware features. The following chapters are included:

Chapter 5: Serial ports
Chapter 6: LPT1 parallel port
Chapter 7: Console devices
Chapter 8: SSDs, DRAM, and battery backup
Chapter 9: External drives
Chapter 10: Interpreting "beep" codes
Chapter 11: PC/104 expansion

# *Chapter 5:* *Serial ports*

## ≡ Description

The 2133 PC/104 CPU has two serial ports, COM1 and COM2. These serial ports interface to a printer, terminal, or other serial device. Both ports support 5–, 6–, 7–, or 8–bit word lengths, 1, 1.5, or 2 stop bits, and baud rates up to 115.2K. COM1 can be used as a serial console or an 8–wire RS–232; COM2 is a dedicated 8–wire RS–232.

The serial ports have the following specifications:

- 16C550 compatible
- 16–byte FIFO buffers
- IEC 1000, level 3, ESD protection
  - — Contact discharge ±6 kV
  - — Air–gap discharge ±8 kV
- Backdrive protection
- Up to 115.2K baud operation

The following sections describe these ports in more detail.

# ≡ Serial port configurations

## COM1

COM1 is a full 8–wire RS–232. The I/O address for COM1 is 3F8h. IRQ4 is dedicated to COM1. If COM1 is not used under interrupt control, IRQ4 may be used by other resources. If COM1 is the console, COM1 uses IRQ4. COM1 can be used for console I/O or RS–232 I/O. COM1 uses the J3, 10–pin header.

When a video card is not connected to the PC/104, COM1 can be used as the console port. All video and keyboard information is redirected through the console port.

COM1 can be enabled as a serial console by two methods. SETUP allows you to select COM1 as a serial console. This is the default configuration stored in the BIOS. Also, when COM1 RTS is connected to COM1 RI the serial console on COM1 is enabled. This hardware configuration overrides the COM1 setting in SETUP. Figure 5–1 shows a custom null modem cable that performs this configuration.

Table 5–1 shows the pinout for COM1 and COM2.

## COM2

COM2 is a dedicated full 8–wire RS–232 configuration. RI on COM2 can be used as a method of "waking up" the 2133 PC/104 CPU from a power management SUSPEND operation. Refer to the CPU power management chapter for more information. The I/O address for COM2 is 2F8h.

IRQ3 is dedicated to COM2. If COM2 is not used under interrupt control, IRQ3 may be used by other resources. COM2 uses the J9, 10–pin header.

Table 5–1 shows the pinout for COM1 and COM2.

Note See the Accessories appendix for mating information on COM1 and COM2.

Table 5–1 COM1 and COM2 pinouts (J3 and J9 connectors)

| Pin | Signal | Pin | Signal |
|---|---|---|---|
| 1 | DCD | 2 | DSR |
| 3 | RxD | 4 | RTS |
| 5 | TxD | 6 | CTS |
| 7 | DTR | 8 | RI |
| 9 | Gnd | 10 | Gnd |

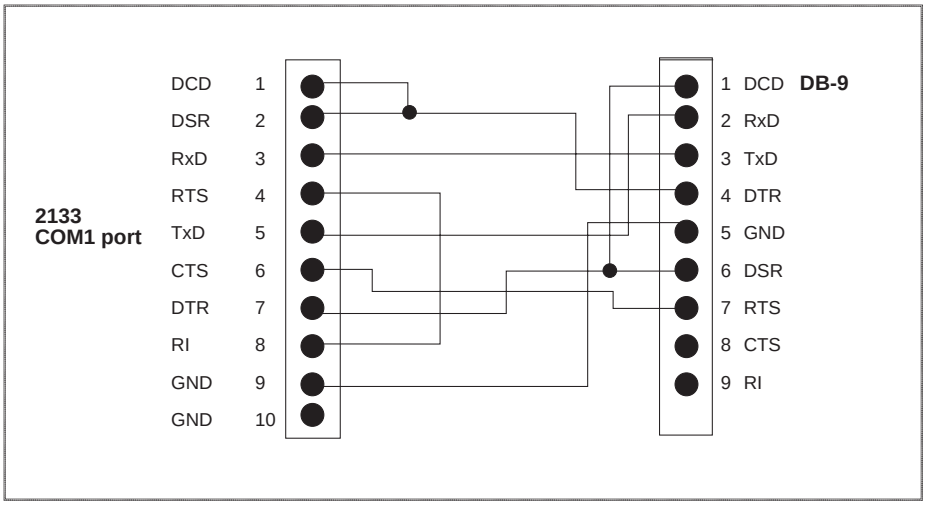
# ≡ Function and use of serial ports

## COM1 as the serial console device

Instead of using a PC/104 video card to connect a monitor, you can use COM1 as a console device. COM1 can be enabled as a serial console by two methods. SETUP allows you to select COM1 as a serial console. This is the default configuration stored in BIOS. Also, when COM1 RTS is connected to COM1 RI the serial console on COM1 is enabled. This hardware configuration overrides the COM1 setting in SETUP. Figure 5–1 shows a custom null modem cable that performs this configuration.

Note When interfacing the 2133 PC/104 CPU to your desktop PC, you must use a null modem cable. Commercial null modem adapters do not connect RI to RTS; they can be used if COM1 is enabled as a serial console in SETUP. The custom null modem below configures COM1 as a serial console regardless of the settings in SETUP. You also need PC SmartLINK, available from Octagon, or an equivalent terminal emulator.

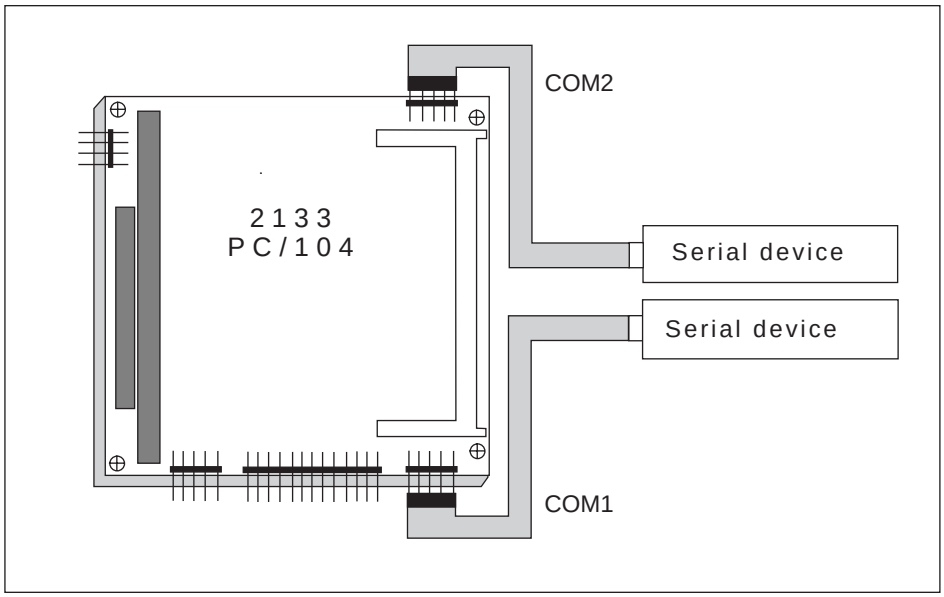
Figure 5–1 Custom null modem adapter with RTS and RI connected



## COM1 and COM2 as RS–232 I/O

COM1 and COM2 can be used as RS–232 serial ports. COM1 and COM2 support 8–wire RS–232 configurations. Use a VTC–9F or VTC–9M cable to interface from the 2133 PC/104 CPU to the serial device.

Figure 5-2 2133 PC/104 CPU serial devices



# Chapter 6: LPT1 parallel port

## ≡ LPT1 parallel port

The LPT1 port has a 26–pin connector. It supports the unidirectional standard mode, bidirectional mode, enhanced parallel port (EPP) mode, and extended capabilities port (ECP) mode. The default I/O address is 378h (IRQ7). You can also select I/O address 278h or 3BCh, or IRQ5, in the SETUP menu.

The LPT1 port supports a number of devices including a PC compatible printer, a floppy drive, a multiline display, a matrix keypad or an opto rack with opto–isolated digital I/O modules.

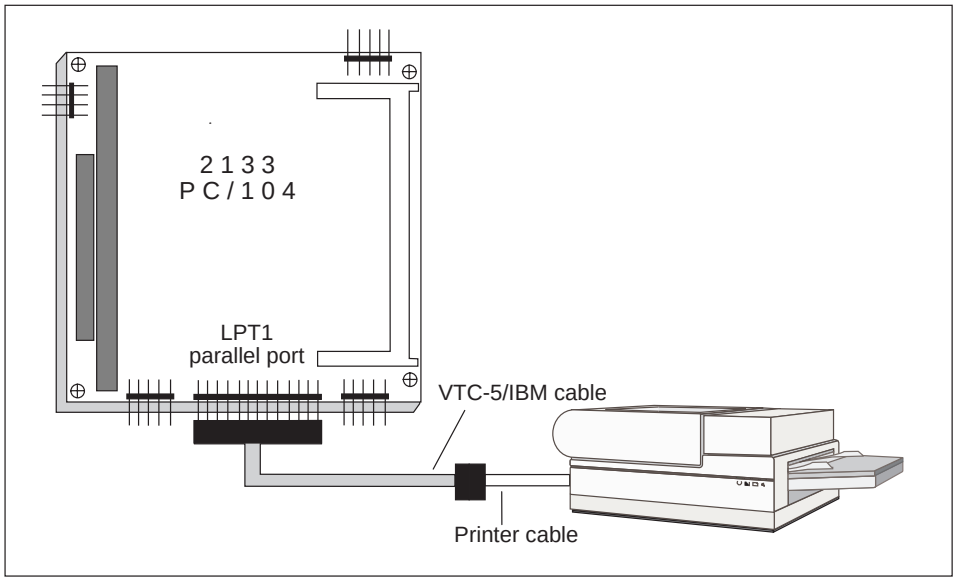
## ≡ Printer

Note See the Accessories appendix for mating information on the LPT1 printer port connector.

To install a printer:

1. Make sure that the LPT1 port is in standard or bidirectional mode.
2. Connect an Octagon VTC–5/IBM cable from the LPT1 port to the 25–pin connector on your printer cable. You can also connect a PCA–36 cable directly from LPT1 to a Centronics interface on a printer.
3. Connect the cable to your printer.

Figure 6–1 LPT1 as a printer port



## ≡ Floppy drive

The LPT1 parallel port can be used as a floppy disk drive port. Refer to the External Drives chapter for connection information.

## ≡ Display

The LPT1 port supports either a 4 x 20 or a 4 x 40 liquid crystal display (LCD). To interface the displays to the 2133 PC/104 CPU, use the Octagon 2010 interface board. A CMA–26 cable is required to connect the interface board to the 2133 PC/104 CPU. The program DISPLAY.EXE (found on the 2133 PC/104 CPU utility disk) provides an easy method to use the display. Refer to the file DISPLAY.DOC on the 2133 PC/104 CPU utility disk for information on initializing and using the display. Also, refer to the 2010 product sheet for more information on the interface board.

To install a display:

1. Connect a CMA–26 cable from the LPT1 port on the 2133 PC/104 CPU (J4) to J1 on the 2010. See Figure 6–2 (next page).
2. Connect the display cable to either the 14–pin or 16–pin header on the 2010. The size of the display will determine which header to use.
3. Refer to the file DISPLAY.DOC for more information on initializing and using the display.

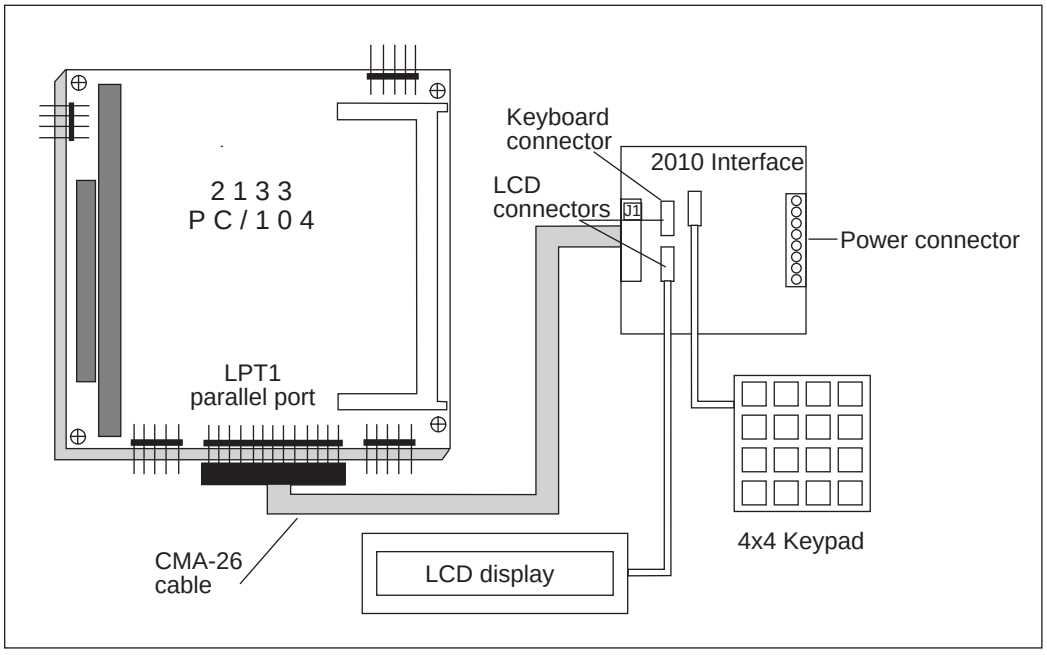
# ≡ Keypad

LPT1 also supports 4 x 4 matrix keypads. To interface the keypad to the 2133 PC/104 CPU, use the Octagon 2010 interface board. A CMA–26 cable is required to connect the interface board to the 2133 PC/104 CPU. The program DISPLAY.EXE (found on the 2133 PC/104 CPU utility disk) provides an easy method to use the keypad. Refer to the file DISPLAY.DOC on the 2133 PC/104 CPU utility disk for information on initializing and using the keypad. Also, refer to the 2010 product sheet for information on the interface board.

To install a keypad:

1. Connect a CMA–26 cable from the LPT1 port on the 2133 PC/104 CPU (J4) to J1 on the 2010. See Figure 6–2.
2. Connect the keypad cable to the 10–pin header on the 2010.
3. Refer to the DISPLAY.DOC file for more information on reading the keypad.

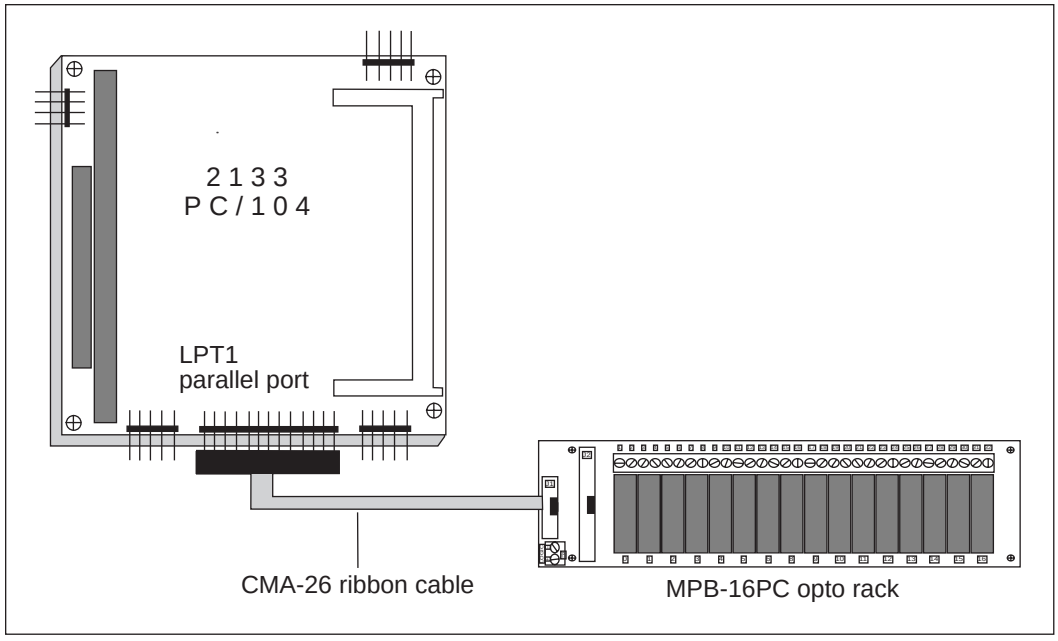
Figure 6–2 LPT1 as a display or keypad port



# ≡ Opto rack

The Octagon MPB–16PC opto rack interfaces directly to the parallel printer port and can control high voltage/high current G4 opto–isolated modules. Of the sixteen positions available, eight can be either input or output, four are dedicated as inputs and the other four are dedicated as outputs. Refer to the MPB–16PC opto module rack product sheet for more information.

Figure 6–3 LPT1 and an opto rack



# *Chapter 7:* *Console devices*

## ≡ Description

The 2133 PC/104 CPU has three console options. You can use a PC/104 video card with a monitor and a keyboard as your console. You can also use COM1 as the console, or you can run the system without a console device.

## ≡ Selecting console devices

The following represent the 2133 PC/104 CPU's three options for console devices:

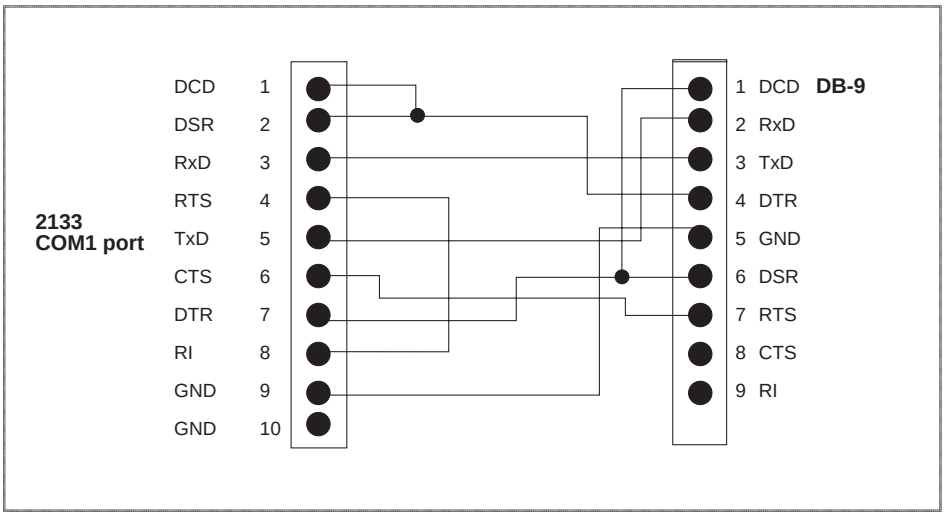
- A standard PC/104 video card, such as an Octagon 2430 SVGA card, and a keyboard.
- Serial console from COM1. A serial cable/null modem adapter plugged into a host PC running SmartLINK provides both input and output. The local keyboard also allows input.
- No console device means no video output, either from a PC/104 card or the serial console. The local keyboard allows input.

## COM1 as serial console device

Instead of using a PC/104 video card to connect a monitor, you can use COM1 as a console device. COM1 can be enabled as a serial console by two methods. SETUP allows you to select COM1 as a serial console. This is the default configuration stored in BIOS. Also, when COM1 RTS is connected to COM1 RI the serial console on COM1 is enabled. This hardware configuration overrides the COM1 setting in SETUP. Figure 7–1 shows a custom null modem cable that performs this configuration.

Note When interfacing the 2133 PC/104 CPU to your desktop PC, you must use a null modem cable. Commercial null modem adapters do not connect RI to RTS; they can be used if COM1 is enabled as a serial console in SETUP. The custom null modem below configures COM1 as a serial console regardless of the settings in SETUP. You also need PC SmartLINK, available from Octagon, or an equivalent terminal emulator.

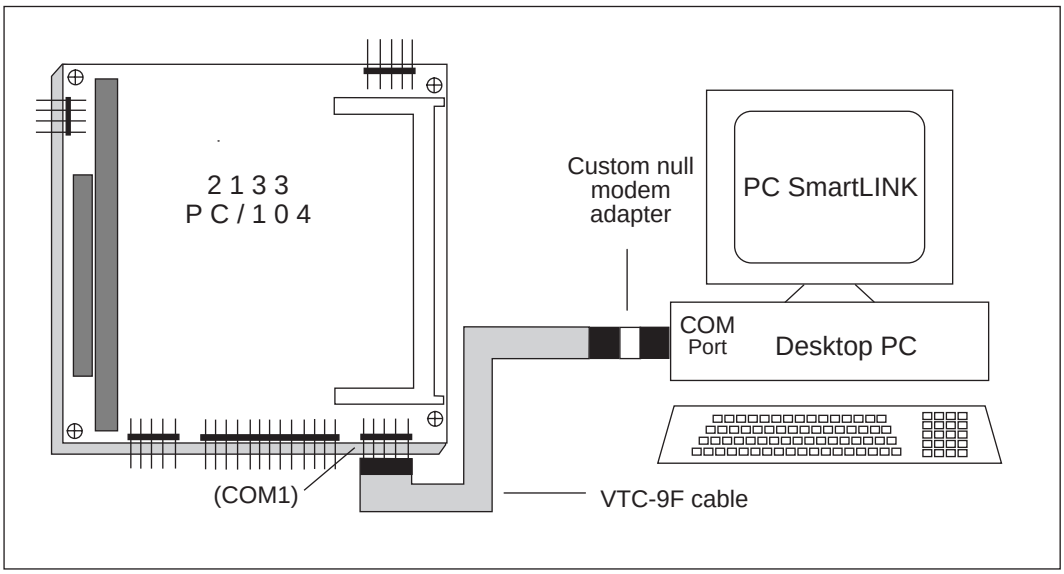
Figure 7–1 Custom null modem adapter with RTS and RI connected




You can use a commercial null modem adapter instead of the custom null modem adapter shown above if the SETUP option "Serial Console on COM1" is enabled (default configuration). If this option is disabled in SETUP, you can get the serial console back by any of the following:

1. Using the custom null modem adapter shown above.
2. Removing the "S" jumper W2[5-6] and rebooting. This causes the 2133 to use the default SETUP settings from the BIOS.
3. Installing a PC/104 video card/keyboard to run SETUP and enable the serial console.

Figure 7–2 The 2133 PC/104 CPU and a serial console



## Keyboard, speaker, and mouse

You can add a keyboard and a local speaker with any of the console options listed above in the Selecting console devices section. Interface the keyboard and speaker via the 10–pin connector at J5. You may use any external speaker from 8–50 ohms. Table 7–1 shows the pinout for connector J5.

If your application requires a mouse, you can connect a serial mouse to COM1 or COM2, and load a mouse driver. The 2133 PC/104 CPU does not require a keyboard, speaker, or mouse for operation.

Table 7–1 Speaker, battery, keyboard, and reset: J5

| Pin | Signal | Function |
|---|---|---|
| 1 | +Speaker | +5V in series with 33 Ω |
| 2 | GND | 3.6V external battery, negative |
| 3 | Reset SW | External reset |
| 4 | KBD SW | Inhibit switch to disable keyboard |
| 5 | KBD Data | Keyboard data |
| 6 | KBD CLK | Keyboard clock |
| 7 | Ground | Signal and power ground |
| 8 | KBD PWR | +5V for keyboard |
| 9 | BATV+ | 3.6V external battery, positive |
| 10 | PWR good | Power good input |

# ≡ Transferring files between the 2133 PC/104 CPU and your PC

Once you have established communications between your PC and the 2133 PC/104 CPU, you can serially download files to any read/write drive used by the 2133 PC/104 CPU. You can also upload files from the 2133 PC/104 CPU to your desktop PC for editing and debugging.

There are two methods to download files through the serial port to the 2133 PC/104 CPU. Depending on the operating system you have installed, you may have the utilities TRANSFER.EXE or REMDISK/REMSERVE (ROM–DOS) or INTERLINK/INTERSVR (MS–DOS) on the DOC. You can use these utilities to transfer files from your host computer to the 2133 over a serial console. Refer to the documentation included with your operating system for additional information on how to use these utilities.

## ROM–DOS operating system

The TRANSFER utility is used to download files, one at a time, to the 2133 PC/104 CPU using the XMODEM protocol. TRANSFER.EXE is used to send or receive files via the serial port (e.g., COM1). TRANSFER.EXE uses the XMODEM protocol, as does PC SmartLINK. (See the note below on XMODEM).

Note XMODEM only transfers files in which the file size is exactly on a 128–byte boundary. If the file size does not fall exactly on the boundary, XMODEM automatically rounds the file size up to the next 128–byte boundary with padding characters. For example, a file with a size of 10,000 bytes, will be rounded up to 10,112 bytes, transferred, and written with the new file size. In most cases, this is not a concern, but in some instances the XMODEM padding causes problems. The padding problems become apparent when an application program is expecting a specific file size or is expecting characters other than the padding characters to be at the end of the file.

REMDISK/REMSERV utilities allow access to all of the files on a remote disk drive. Once these programs are executed, single or multiple files can then be transferred to and from the 2133 PC/104 CPU using DOS COPY or XCOPY commands.

## MS–DOS operating system

For information on using INTERLINK/INTERSVR utilities, refer to the documentation included with your operating system.

# ≡ Transferring files to the 2133 PC/104 CPU

The following steps detail the procedures for transferring files from your PC to a virtual drive on the 2133 PC/104 CPU. This procedure assumes you are using ROM–DOS, and have the TRANSFER utility available. In order to transfer files from your PC to the 2133 PC/104 CPU, you must execute the TRANSFER program from both the 2133 PC/104 CPU and your PC.

1. Connect a 9-pin serial cable with a null modem adapter between COM1 of your PC to COM1 of the 2133 PC/104 CPU, using a VTC–9F cable.

2. Execute the TRANSFER program from the 2133 PC/104 CPU to receive a file from your PC.

   **TRANSFER /COM1 /R /V <drive>*filename.ext***

   <drive> is the drive on the 2133 PC/104 CPU where the file will be transferred.

   /R specifies to receive a file (default).

   *filename.ext* is the name of the file on the 2133 PC/104 CPU which you are receiving from your PC.

   COM1 specifies the serial port on the 2133 PC/104 CPU.

   /V enables "R" characters upon receiving a block and "T" upon transferring a block. Do not use /V when using a serial console.

3. Execute the TRANSFER program from your PC to send a file to the 2133 PC/104 CPU. /S specifies send to file.

   **TRANSFER /COM1 /S /V <drive><path>*filename.ext***

   *filename.ext* is the name of the file on the PC which you are sending to the 2133 PC/104 CPU.

Note An alternate method of transferring a file is to press <ALT><D>, when you use PC SmartLINK.

Note Transfer will time-out if the program has not been started after approximately 40 seconds. It displays the following message:

```
Failed to receive <drive>filename.ext
Deleting  <drive>filename.ext
```

Also, you may speed up the transfer using the /Bnnnn switch to increase the baud rate. Example: /B57600. When you use a serial console, do not use the /B option on the 2133 PC/104 CPU. Instead, change the serial console baud rate in SETUP.

# ≡ Transferring files from the 2133 PC/104 CPU

This procedure assumes you are using ROM–DOS, and have the TRANSFER utility available. In order to transfer files from your PC to the 2133 PC/104 CPU, you must execute the TRANSFER program from both the 2133 PC/104 CPU and your PC.

1. Connect a 9-pin serial cable with a null modem adapter between COM1 of your PC to COM1 of the 2133 PC/104 CPU, using a VTC–9F cable.

2. Execute the TRANSFER program from the 2133 PC/104 CPU to send a file to your PC.

   **TRANSFER /COM1 /S /V *filename.ext***

   *filename.ext* is the name of the file on the 2133 PC/104 CPU which you are sending to your PC.

   /V enables "R" characters on receiving a block and "T" on transferring a block.

   **/COM1 /S /V: Send in verbose mode**

3. Execute the TRANSFER program from your PC to receive a file from the 2133 PC/104 CPU.

   **TRANSFER /COM1 /R /V *filename.ext***

   *filename.ext* is the name of the file on the PC which you are receiving from the 2133 PC/104 CPU.

   **/COM1 /R /V: Receive in verbose mode**

Note Transfer will time-out if the program has not been started after approximately 40 seconds. It displays the following message:

```
Failed to receive <drive>filename.ext
Deleting  <drive>filename.ext
```

Also, you may speed up the transfer using the /Bnnnn switch to increase the baud rate. Example: /B57600. When you use a serial console, do not use the /B option on the 2133 PC/104 CPU. Instead, change the serial console baud rate in SETUP.

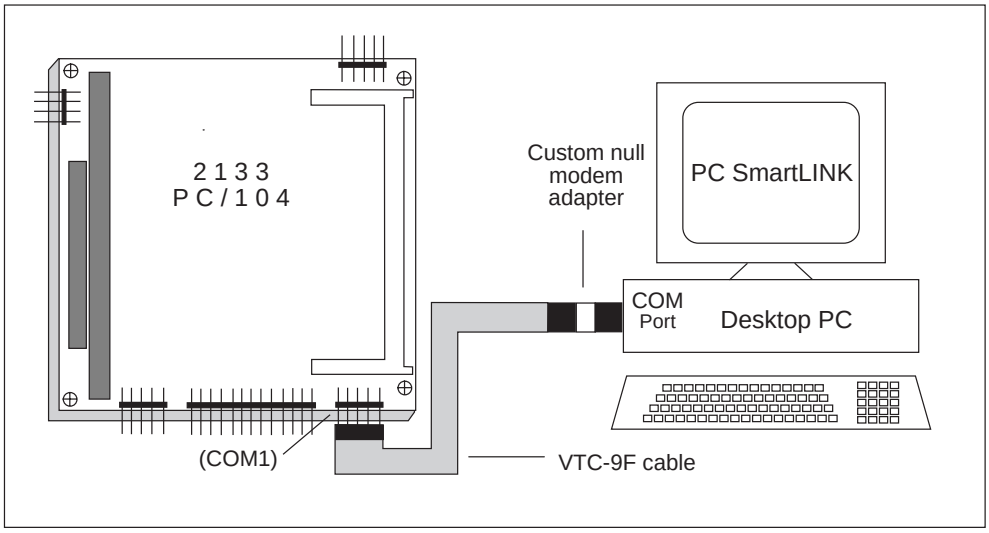
## ≡ Downloading files to the 2133 PC/104 CPU using PC SmartLINK

This procedure assumes you are using ROM–DOS, and have the TRANSFER utility available. The following information shows how to download files between the 2133 PC/104 CPU and your PC using Octagon's PC SmartLINK. For other communication programs, refer to those program instructions for performing Xmodem file transfer to a target system. The hardware and software requirements are:

- Desktop PC, running PC SmartLINK, connected by a VTC–-9F cable and a null modem adapter to COM1 of the 2133 PC/104 CPU.
- 2133 PC/104 CPU with ROM–DOS operating system.

1. Connect the equipment according to the following diagram:

Figure 7–3 Downloading files using PC SmartLINK and TRANSFER.EXE




2. Start PC SmartLINK and power on the 2133 PC/104 CPU.

Note PC SmartLINK is a DOS application. File transfer problems will occur when running in Windows. If you are using Windows on your PC, restart the PC in DOS mode before running SmartLINK.

3. Execute the TRANSFER.EXE program from the 2133 PC/104 CPU by entering:

   `C:\> TRANSFER filename.ext`

   *filename.ext* is the name of the file on your PC which you are sending to 2133 PC/104 CPU.

   The following message is displayed from the 2133 PC/104 CPU:

   `Receiving filename.ext . . .`

4. Execute the following steps using PC SmartLINK:
   - Press <ALT><D> to enter the download screen.
   - Type in the name of the file to transfer, e.g.:
     **C:\MPC\DEMO*filename.ext***
   - To begin the transfer:
     — either press ENTER (default download START);
     — tab to START;
     — click on the START button in the download screen.
   - When the file transfer is completed, press <ESC> twice to return to the main PC SmartLINK screen.

Note TRANSFER.EXE will time-out if the program has not been started after approximately 40 seconds. If the time-out occurs, the following message from the 2133 PC/104 CPU is displayed:

```
Failed to receive filename.ext!
Deleting filename.ext
```

6. When the file transfer is complete, type the following DOS command to view the drive directory and confirm that your file has been transferred to the 2133 PC/104 CPU:

```
C:\> DIR
```

The system will display the contents of drive contents:

```
Volume in drive is <label>
Directory of <drive>:\

filename ext   27264 06-07-96   2:57p

     1 file(s) 27264 bytes
```

# Chapter 8: SSDs, DRAM, and battery backup

## ≡ Description

The 2133 PC/104 CPU contains one solid–state disk and a socket for a second solid–state disk.

## ≡ Booting

The 2133 PC/104 CPU can boot from an external drive such as a floppy connected to LPT1 or a PC/104 hard drive, or from an M–Systems DiskOnChip (DOC) in SSD1. For an external drive, the boot sequence is specified in SETUP. For an M–Systems DOC, the BIOS in SSD0 automatically looks for BIOS extensions in the DOC during the boot process.

## ≡ SSD0

SSD0 contains an on–board 512 KB flash. 128 KB contains the BIOS drive and power management. 384 KB is unused, but is not accessable.

## ≡ SSD1

SSD1 is a 32–pin DIP socket for SRAM or M–Systems DOC. SRAM is automatically backed up when an AT battery is connected. If a DOC is installed in SSD1 it can be programmed as a boot drive. The socket exhibits high retention force and affords a gas tight contact.

> WARNING!
>
> Incorrect installation of an SSD1 device will destroy the chip. To install a device in SSD1, be sure to match the notch in the device with the notch in the SSD1 marked silkscreen on the 2133.

Note SRAM contents are sometimes affected by system noise. Therefore, the use of SRAM is not recommended in electrically noisy environments, especially when systems are critical. Also note that SRAM devices require a battery to back up the SRAM files (see "Battery backup for SSD1 SRAM and real time calendar clock" in this chapter.

### Jumpers

Table 8–1 shows the jumper configurations for devices in SSD1. An M–Systems DOC is configured as flash.

Table 8–1 SSD1 device configuration: W1

| Pins | Description |
|---|---|
| W1[1–3, 2–4, 6–8, 7–9] | SRAM with battery backup |
| W1[1–2, 3–4, 5–6, 9–10] | *Flash (DiskOnChip) |

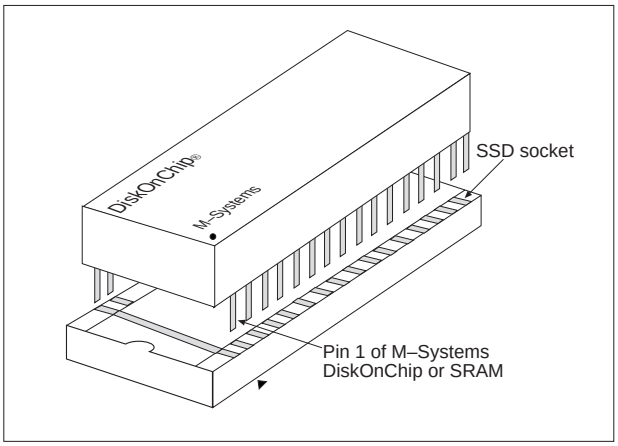
* = default

## ≡ M–Systems DiskOnChip

The 2133 supports a preprogrammed M–Systems DOC. Refer to your M–Systems utility disk for more information on using a DOC.

Before installing M–Systems DOC, make sure the SSD socket is configured as flash W1[1–2, 3–4, 5–6, 9–10]. Also, align pin one of the DOC with pin one of the SSD1 socket. Refer to Figure 8–1 for correct SSD alignment.

Figure 8–1 M–Systems pin alignment with the 2133 SSD1 socket



# ≡ SSD1 as SRAM space

## Selecting the SSD and address lines

The upper address lines are routed to the SSD1 socket via Octa/ Superglue ports. The following port controls the address lines. This allows access via a 32K window. This window is at segment E800–EFFFh.

Table 8–2 SSD access: SSD control at address 208h

| Register 0x208 | Select 1 | Select 0 | A20 | A19 | A18 | A17 | A16 | A15 |
|---|---|---|---|---|---|---|---|---|
| Memory Select | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2* | Bit 1 | Bit 0 |
| No SSD (default after reset) | 0 | 0 | | | | | | |
| SSD0 | 0 | 1 | WARNING! BIOS Area, Do Not Use. | | | | | |
| SSD1 | 1 | 0 | | | | | | |
| Bank 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bank 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Bank 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Bank 3 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Bank 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Bank 5 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Bank 6 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| Bank 7 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Bank 8 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Bank 9 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| Bank 10 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| Bank 11 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| Bank 12 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| Bank 13 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Bank 14 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| Bank 15 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

* Note = A17 must be set to 1 when accessing 128K SRAM devices.

> WARNING!
>
> Do not set 208h bits 7 and 6 to 01. This will allow access to the BIOS area of SSD0. Potential damage can occur to the BIOS and render the 2133 unbootable.

## Accessing SSD contents

> WARNING!
>
> Because of the power management features on this card (including automatic cool down clocking) if any critical timings are necessary (e.g. during write to a flash) your SW must be capable of running at multiple speeds or you must disable the feature via PMISETUP program (default). If the cool down clock feature is disabled, the CPU must be kept within its proper operating temperature or damage to the CPU will result.

Follow the steps below to access the SSD contents:

1. Determine the address from which you want to poke or peek data. Divide by 32768 to get the bank address

Example: Assume the 512K memory goes from 0 to 524288 (decimal). If you wanted to write to 33280, follow the steps below:

   a. Divide 33280 by 32768, which equals 1.015625. 1.015625 equals Bank 1 (truncate to the next lower whole number).

   b. Find the offset by multiplying the fractional part (.015625) by 32768. In this case, the offset is 512.

2. OR 80h with the bank address to set the top two bits to 10xxxxxx to select SSD1.

3. If you are using a 128K SRAM part also OR in 04h.

4. Output the result to 208h.

5. The bank will now be available at segment E800h–EFFFh.

> WARNING!
>
> There is no mechanism in place to prevent the destruction of valid data poked into SSD1.

Use the following C code with 512K SRAM to select banks:

```
outportb(0x208,0x80 | 0); // selects bank 0
(1st 32K) of 512K SRAM
outportb(0x208,0x80 | 1); // selects bank 1
(2nd 32K) of 512K
outportb(0x208,0x80 | 2); // selects bank 2
(3rd 32K) of 512K
outportb(0x208,0x80 | 3); // selects bank 3
(4th 32K) of 512K
      etc...
outportb(0x208,0x80 | F); // selects bank 15
(last 32K) of 512K
```

When using 128K SRAM use the following code:

(bit 2 needs to be 1 with 128K SRAM)

```
outportb(0x208,0x80 | 0 | 4); // selects bank 0
(1st 32K of 128K)
outportb(0x208,0x80 | 1 | 4); // selects bank 1
(2nd 32K of 128K)
outportb(0x208,0x80 | 2 | 4); // selects bank 2
(3rd 32K of 128K)
outportb(0x208,0x80 | 3 | 4); // selects bank 3
(4th and last 32K of 128K)
```

**Borland/Turbo C++ example to access 2133 SSD1, bank 4 of a 512K SRAM**

```
// disable interrupts
     disable();

// select bank 4 (5th 32K) of 512K SRAM
     outportb(0x208,0x80 | 4);

// now access the device
// your code goes here
// data can be written to locations E800:0000 -
     E800:7FFFh

// when finished accessing, deselect SSD1
     outportb(0x208,0x00);

// enable interrupts
     enable();
```

# ≡ DRAM

The standard version of the 2133 PC/104 CPU is shipped with 4 MB of DRAM on–card. The on–card DRAM socket holds up to a 64 MB dual in–line memory module (DIMM). The card supports fast page mode (FPM) and extended data out (EDO) memory. These can be symmetric or asymmetric configurations. You may order EDO DIMM memory modules from Octagon Systems. See the following for ordering information:

| Description | Octagon Part Number |
|---|---|
| 4 MB EDO DIMM memory module | 4583 |
| 8 MB EDO DIMM memory module | 4584 |
| 16 MB EDO DIMM memory module | 4582 |
| 32 MB EDO DIMM memory module | 4990 |
| 64 MB EDO DIMM memory module | 5364 |

## ≡Battery backup for SSD1 SRAM and real time calendar clock

If SRAM is used in SSD1 you need an AT battery for battery backup of the SRAM files. An AT battery also backs up the CMOS real time clock. The 2133 PC/104 CPU does not have a battery installed when it is shipped. To install a battery:

1. Power off the 2133 PC/104 CPU.
2. Install the 3.6V AT clock battery at the J5 connector. Refer to the component diagram in the Quick start chapter for the location of J5.

Table 8–3 Speaker, battery, keyboard, and reset: J5

| Pin | Signal | Function |
|---|---|---|
| 1 | +Speaker | +5V in series with 33 Ω |
| 2 | GND | 3.6V external battery, negative |
| 3 | Reset SW | External reset |
| 4 | KBD SW | Inhibit switch to disable keyboard |
| 5 | KBD Data | Keyboard data |
| 6 | KBD CLK | Keyboard clock |
| 7 | Ground | Signal and power ground |
| 8 | KBD PWR | +5V for keyboard |
| 9 | BATV+ | 3.6V external battery, positive |
| 10 | PWR good | Power good input |

# *Chapter 9: External drives*

## ≡ Description

The LPT1 parallel port can be used as a floppy disk drive port. The 2133 PC/104 CPU is compatible with all common floppy disk drives used on desktop PCs. It can also be used with a PC/104 floppy controller. The floppy drives use DMA channel 2.

The 2133 PC/104 CPU can also interface with any standard IDE or EIDE hard drives that have a PC/104 interface.

## ≡ Floppy disk controller

The 2133 PC/104 CPU uses an Octagon FCA–12 cable to connect directly to one or two 3.5 in. or 5.25 in. floppy drives via the LPT1 connector at J4.

Note See the Accessories appendix for mating information on the floppy disk connector.

Note If you wish to add a second floppy drive to your system, you must use a floppy drive cable which has two connectors.

### Power requirements

You must supply power to the floppy drive(s) through an external source. Refer to your floppy drive manual for specific instructions.

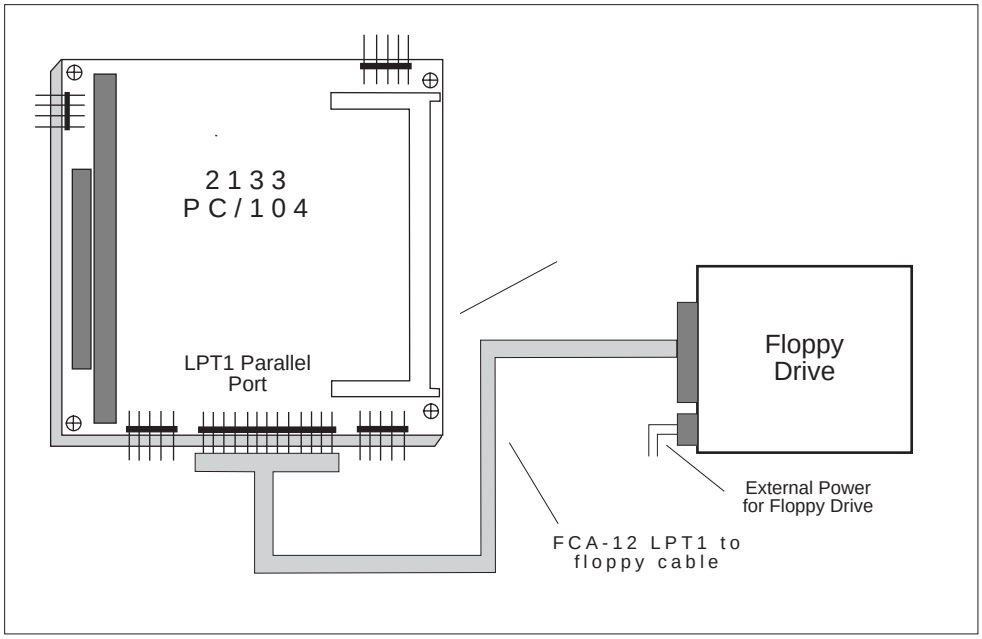
## Installing a floppy disk drive

You can connect a floppy disk drive to LPT1, or use a PC/104 floppy disk controller card. This procedure shows how to connect a floppy disk drive to LPT1.

1. Disconnect power to the 2133 PC/104 CPU.
2. Insert one end of an Octagon FCA–12 cable into the rear of the floppy drive. Make sure pin 1 on the cable is connected to pin 1 on the drive.
3. Insert the other end of the cable into J4 on the 2133 PC/104 CPU.
4. Connect power to the floppy drive.
5. Run SETUP in the \UTILS subdirectory to set up the floppy drive. You can execute this program either by pressing <BACKSPACE><S> during system bootup or by executing the file SETUP.EXE. The system steps you through the configuration. Also, refer to the SETUP programs chapter for more information on the AT BIOS SETUP program. Reboot the system after running BIOS SETUP.

Note Ensure that the Setup jumper (W2[5–6]) is installed so that the system will use the Setup settings in EEPROM when the system is rebooted.

Figure 9–1 LPT1 and a Floppy Drive



# ≡ Hard disk controller

The 2133 PC/104 CPU will drive two PC/104 hard drives that have 16–bit IDE interfaces.

To install a hard drive:

1. Disconnect power to the 2133 PC/104 CPU.
2. Connect the PC/104 hard drive to the PC/104 connector on the 2133. Refer to the instructions included with the hard drive.
4. Execute the SETUP program to configure your system for a hard drive. You can execute this program either by pressing <BACK–SPACE><S> during system bootup or by executing the file SETUP.COM. This file is found on the 2133 PC/104 utility disk. The system steps you through the configuration. Also, refer to the SETUP programs chapter for more information on the BIOS SETUP program. Reboot the system after running BIOS SETUP.

Note Ensure that the Setup jumper (W2[5–6]) is installed so that the system will use the Setup settings in EEPROM when the system is rebooted.

5. Run FDISK on the hard drive. This partitions the hard drive, and assigns a drive letter. By default, it will assign the drive letter which is subsequent to your currently installed drives. When FDISK is finished, reboot the system.

Note Some hard drives are shipped from the factory pre–fdisked, and some are pre–formatted. Refer to the documentation included with your hard drive.

6. Format the hard drive using the DOS FORMAT command. When the drive is formatted, reboot the system.
7. If you want to boot the system from the hard drive, run the SYS command. The system copies COMMAND.COM as well as hidden files to the hard drive.
8. Reboot the system.

# *Chapter 10: Interpreting "beep" codes*

## ≡ Description

The 2133 PC/104 CPU BIOS generates system status codes during POST. If a speaker is installed at J5, the beep codes are audible. The following table interprets these beep codes.

Table 10–1 Phoenix BIOS beep codes

| Diagnostic port output | Beep codes | Description of test or failure |
|---|---|---|
| 01h | | 80286 register test in-progress |
| 02h | 1-1-3 | CMOS write/read test in-progress |
| 03h | 1-1-4 | BIOS ROM checksum in-progress |
| 04h | 1-2-1 | Programmable interval timer test in-progress or failure |
| 05h | 1-2-2 | DMA initialization in-progress or failure |
| 06h | 1-2-3 | DMA page register write/read test in-progress or failure |
| 08h | 1-3-1 | RAM refresh verification in-progress or failure |
| 09h | | 1st 64K RAM test in-progress |
| 0Ah | 1-3-3 | 1st 64K RAM chip or data line failure multi-bit |
| 0Bh | 1-3-4 | 1st 64K RAM odd/even logic failure |
| 0Ch | 1-4-1 | 1st 64K RAM address line failure |
| 0Dh | 1-4-2 | 1st 64K RAM parity test in-progress or failure |
| 10h | 2-1-1 | 1st 64K RAM chip or data line failure-bit 0 |
| 11h | 2-1-2 | 1st 64K RAM chip or data line failure-bit 1 |
| 12h | 2-1-3 | 1st 64K RAM chip or data line failure-bit 2 |
| 13h | 2-1-4 | 1st 64K RAM chip or data line failure-bit 3 |
| 14h | 2-2-1 | 1st 64K RAM chip or data line failure-bit 4 |
| 15h | 2-2-2 | 1st 64K RAM chip or data line failure-bit 5 |
| 16h | 2-2-3 | 1st 64K RAM chip or data line failure-bit 6 |
| 17h | 2-2-4 | 1st 64K RAM chip or data line failure-bit 7 |
| 18h | 2-3-1 | 1st 64K RAM chip or data line failure-bit 8 |
| 19h | 2-3-2 | 1st 64K RAM chip or data line failure-bit 9 |
| 1Ah | 2-3-3 | 1st 64K RAM chip or data line failure-bit A |
| 1Bh | 2-3-4 | 1st 64K RAM chip or data line failure-bit B |
| 1Ch | 2-4-1 | 1st 64K RAM chip or data line failure-bit C |
| 1Dh | 2-4-2 | 1st 64K RAM chip or data line failure-bit D |
| 1Eh | 2-4-3 | 1st 64K RAM chip or data line failure-bit E |
| 1Fh | 2-4-4 | 1st 64K RAM chip or data line failure-bit F |
| 20h | 3-1-1 | Slave DMA register test in-progress or failure |
| 21h | 3-1-2 | Master DMA register test in-progress or failure |
| 22h | 3-1-3 | Master interrupt mask reg. test in-progress or failure |
| 23h | 3-1-4 | Slave interrupt mask reg. test in-progress or failure |
| 25h | N/A | Interrupt vector loading in-progress |
| 27h | 3-2-4 | Keyboard controller test in-progress or failure |

Table 10–1 Phoenix BIOS beep codes (cont'd)

| Diagnostic port output | Beep codes | Description of test or failure |
|---|---|---|
| 28h | N/A | CMOS power-fail and checksum checks in-progress |
| 29h | N/A | CMOS config info. validation in-progress |
| 2Bh | 3-3-4 | Screen memory test in-progress or failure |
| 2Ch | 3-4-1 | Screen initialization in-progress or failure |
| 2Dh | 3-4-2 | Screen retrace tests in-progress or failure |
| 2Eh | N/A | Search for video ROM in-progress |
| 30h | N/A | Screen believed operable: |
| 30h | N/A | Screen believed running w/video ROM |
| 31h | N/A | Monochromatic screen believed operable |
| 32h | N/A | 40-column color screen believed operable |
| 33h | N/A | 80-column color screen believed operable |
| 34h | 4-2-1 | Timer tick interrupt test in-progress or failure |
| 35h | 4-2-2 | Shutdown test in-progress or failure |
| 36h | 4-2-3 | Gate A20 failure |
| 37h | 4-2-4 | Unexpected interrupt in protected mode |
| 38h | 4-3-1 | RAM test in-progress or failure above address 0FFFFh |
| 3Ah | 4-3-3 | Interval timer channel 2 test in-progress or failure |
| 3Bh | 4-3-4 | Time-of-day clock test in-progress or failure |
| 3Ch | 4-4-1 | Serial port test in-progress or failure |
| 3Dh | 4-4-2 | Parallel port test in-progress or failure |
| 3Eh | 4-4-3 | Math coprocessor test in-progress or failure |
| 50h | N/A | Beginning of CSET_INIT |
| 51h | N/A | Loading the RCM table |
| 52h | N/A | Loading the FCM table, doing DMC |
| 53h | N/A | Entering CSET_BFR_VIDROM (before video ROM) |
| 54h | N/A | Entering CSET_BFR_SIZMEM (before memory sizing) |
| 55h | N/A | Entering CSET_AFT_MTEST (before memory test) |
| 56h | N/A | Entering CSET_AFT_CMCFG (before CMOS configuration check) |
| 57h | N/A | Entering CSET_BFR_OPROM (before option ROM scan) |

Table 10–2 Additional error codes for Phoenix BIOS

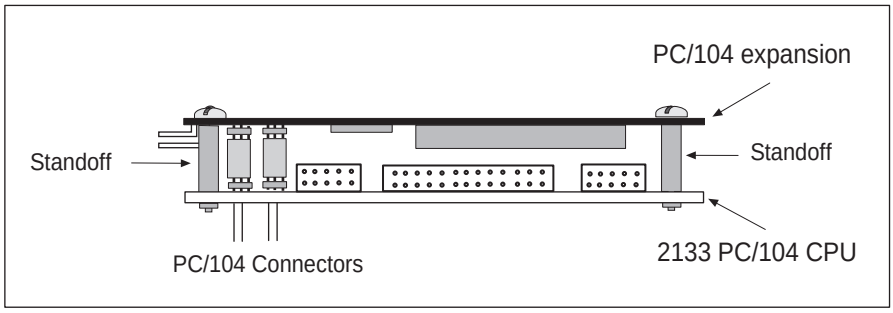
| Diagnostic port output | Beep codes | Description of test or failure |
|---|---|---|
| C0h | N/A | Entry to power management initialization |
| C1h | N/A | Return from power management initialization |
| C2h | N/A | Entry to cache initialization |
| C3h | N/A | Return from cache initialization |
| D7h | N/A | Using defaults from ROM |
| D9h | N/A | Using EEPROM values |
| DFh | N/A | Exit CMOS initialization |
| E0h | N/A | Reset |
| E1h | N/A | BIOS determined it is an actual reset |
| E7h | N/A | Going to CMOS initialization |
| E8h | N/A | Returned from CMOS initialization |
| E9h | N/A | Entry to chipset initialization |
| EAh | N/A | Exit from chipset initialization |
| F0h | N/A | Loading chipset from EEPROM or defaults |
| F1h | N/A | Completed chipset load |
| F3h | N/A | Loading mvb specific values |
| F4h | N/A | Completed mvb load |
| F9h | N/A | Starting memory autosizing |
| FAh | N/A | Completed memory autosizing |

# *Chapter 11:* PC/104 expansion

## ≡ Description

The 2133 PC/104 CPU can interface to up to three additional PC/104 form factor modules including A/D converters, digital I/O, serial ports, etc. The 2133 supports 8– and 16–bit cards and provides ±12V from the power supply at J7. These modules can be stacked on top of the 2133 to form a highly integrated control system.

Figure 11–1 Typical PC/104 module stack




WARNING!

When installing any PC/104 module, avoid excessively flexing the board. Mate pins correctly and use the required mounting hardware.

Note See the Accessories appendix for mating information on the PC/104 connector.

# *Overview:* *Section 3 – System management*

Section 3 provides information on managing the 2133 in the areas of internal control, CPU power management, and troubleshooting. The following chapters are included:

Chapter 12: Watchdog timer and hardware reset

Chapter 13: CPU power management

Chapter 14: Serial EEPROM

Chapter 15: Troubleshooting

# Chapter 12: Watchdog timer and hardware reset

## ≡ Description

The watchdog timer is a fail–safe against program crashes or processor lockups. It times out every 1.6 seconds (1.6 sec. typical, 1.00 sec. min., 2.25 sec. max.) unless reset by the software, or extended by the INT17 functions setWDTimer or incWDTimer. The watchdog timer is software controlled. It can be controlled through the enhanced INT 17h handler (I17HNDLER.EXE).

## ≡ Watchdog function definitions using Borland C code

This section provides definitions for the following functions: Enable watchdog, Strobe watchdog, and Disable watchdog.

### Enable watchdog

To "enable" the watchdog timer, set bit 7 of port 20C in the following Borland C code:

```
int port20c;


outportb(0xae, 0xff)          // Unblock watchdog
                              // enable port
port20c = inportb(0x20c)
outportb(0x20c,port20c | 0x80); // Enable watchdog
```

### Strobe watchdog

The watchdog must be strobed before the timeout period expires. To "strobe" the watchdog timer use the following Borland C code:

```
inportb(0x20c)
```

### Disable watchdog

To "disable" the watchdog use the following Borland C code:

```
int port20c;
outportb(0xae,0xff);
port20c = inportb(0x20c)
outportb(0x20c,port20c & 0x7f)
```

# ≡ Watchdog function definitions using enhanced INT 17h handler

This section provides definitions for the watchdog functions using the Int17 handler (I17HNDLR.EXE). I17HNDLR.EXE is a TSR program. Once executed it is active, but it must be executed each time the system is rebooted. If the Int17 functions will be used by your application, copy the utility to your hard drive and add it to your AUTOEXEC.BAT.

## Enable watchdog

Function: fdh

Subfunction: 01h

Purpose: To enable the watchdog.

Calling registers:
AH fdh
AL 01h
DX ffffh

Return registers: None

Comments: This function enables the watchdog. Once the watchdog is enabled, it has to be strobed at a period of not less than 1.6 seconds or until the watchdog is disabled. Otherwise, a system reset will occur.

Programming example:

```
/* Inline assembly code for Borland C++ 3.1 */
asm {
mov  ax,0fd01h
mov  dx,0ffffh
int  17h
}
```

## Strobe watchdog

Function: fdh

Subfunction: 02h

Purpose: To strobe the watchdog.

Calling registers:
AH fdh
AL 02h
DX ffffh

Return registers: None

Comments: This function strobes the watchdog. Once the

watchdog is enabled, it has to be strobed at a period of not less than 1.6 seconds or until the watchdog is disabled. Otherwise, a system reset will occur.

Programming example:

```
/* Inline assembly code for Borland C++ 3.1 */
asm {
mov ax,0fd02h
mov dx,0ffffh
int 17h
    }
```

The watchdog timer can also be strobed by reading address 20Ch. This may be faster than strobing the watchdog timer with an interrupts function call, for example:

```
A=INP(20Ch)
```

## Disable watchdog

Function: fdh

Subfunction: 03h

Purpose: To disable the watchdog.

Calling registers:

| | |
|---|---|
| AH | fdh |
| AL | 03h |
| DX | ffffh |

Return registers: None

Comments: This function disables the watchdog. Once the watchdog is enabled, it has to be strobed at a period of not less than 1.6 seconds or until the watchdog is disabled. Otherwise, a system reset will occur.

Programming example:

```
/* Inline assembly code for Borland C++ 3.1 */
asm {
     mov   ax,0fd03h
     mov   dx,0ffffh
     int   17h

     }
```

## Set WDTimer

Function: fdh

Subfunction: 04h

Purpose: To set the watchdog timer timeout period.

Calling registers:
AH fdh
AL 04h
BX count in ticks
DX ffffh

Return registers: BX remaining count

Comments: This function sets the counter for the software/watchdog timer. The software/watchdog strobe interrupt handler is installed in the user timer tick interrupt (INT 1C) chain. At each tick (approximately 54 ms or 18.2 times per second) the handler wakes up and checks the counter. If the counter is not zero, it decrements the counter.

Every eight ticks of the counter (or a little less than 1/2 second), the handler strobes the watchdog. Once the counter is down to 0, no further strobing occurs, until a new count is set. If the counter is set to 182, the handler will then strobe the watch dog automatically every 8 ticks for about 10 seconds (18.2 * 10) when the counter has decremented to 0. If no other strobing is done, the watchdog will reset in about 1 second after that.

A typical application would enable the watchdog and then do manual strobing with SetWDTimer. Before calling functions such as the flash write procedure (which may take longer than 1.2 seconds), the subroutine would increase the counter for that subroutine using IncrementWDTimer.

Notes: 1. The application is responsible for enabling the watchdog before using these functions.

2. The application may continue to strobe the watchdog manually. While the counter is not zero, strobing will occur from both places.

3. Setting the counter to 0 disables the automatic strobing.

4. If interrupts such as IRQ0 are disabled for a period longer than 1/2 second, the handler might not strobe the watchdog before the system resets.

5. If the period of IRQ0 is changed, this may change the INT 1C frequency, thereby changing the watchdog strobe frequency.

Programming example:

```
/* Inline assembly code for Borland C++ 3.1 */
asm {
     mov   ax,0fd04h
     mov   dx,0ffffh
     mov   bx,182          ; 10 seconds = 18.2*10
     int   17h             ; set counter to 10 seconds
     }
```

Additional example

```
asm {
     mov   ax, 0fd04h
     mov   dx,0ffffh
     mov   bx,0            ; 0 disables counter
     int   17h             ; disable counter
  }
```

## IncrementWDTimer

Function: fdh

Subfunction: 05h

Purpose: To increment the watchdog timer timeout period.

Calling registers:

| | |
|---|---|
| AH | fdh |
| AL | 05h |
| BX | increment count in ticks |
| DX | ffffh |

Return registers:

| | |
|---|---|
| BX | remaining count |

Comments: See SetWDTimer comments

Programming example:

```
/* Inline assembly code for Borland C++ 3.1 */
asm {
     mov   ax,0fd05h
     mov   dx,0ffffh
     mov   bx,91           ; 5 seconds = 18.2*5
     int   17h             ; increment counter by 5s
     }
```

## ≡ Hardware reset

The 2133 PC/104 CPU does not have a reset button. However, pin 3 on connector J5 can be connected to a switch which momentarily pulls the pin to ground, causing a hardware reset. Also, the RESET command accomplishes the same thing as a reset button. This provides a more complete reset than the <CTRL><ALT><DEL> method.

> WARNING!
>
> When using COM1 as the console, the <CTRL><ALT> <DEL> commands on the host system keyboard only reset the host system. Use the RESET command to issue a hardware reset on the 2133.

# Chapter 13: CPU power management

## ≡ Description

The power demands of a system can severely limit an application due to thermal constraints or the raw power usage in a battery–operated application. To maintain speed and efficiency, a software–controlled, power management system must be tailored to the application. Even if your application is operating within specified limits, a power management system may improve the life and reliability of your system by reducing thermal stress to the CPU.

The advance power management functions include:

- SUSPEND through software control, RESUME operation via hardware interrupts
- Slowing down the CPU by dividing the clock
- Contextual Save to Disk

Power management can be enabled in the 2133 SETUP program and is adjusted with the PMISETUP program. DOS–supplied advanced power management (APM) programs, such as POWER.EXE are also supported. See the 2133 PC/104 CPU utility disk for a list of example programs located in the \EXAMPLES directory. For more information on using the SETUP utility, refer to the SETUP programs chapter. For more information on using the PMISETUP utility, see the PMISETUP section later in this chapter.

## ≡ Power management overview

Power management is implemented via the software management interface (SMI) function, and provides multiple levels of management. The firmware is also capable of cooperative power management with an APM compatible driver or application, such as POWER.EXE. Cooperative power management allows power aware applications to control the power state of the system without depending on interrupts or device access to indicate that the CPU is actively executing application code. At the hardware level, the power management system cannot detect CPU activity except by monitoring bus activity such as interrupts or access to specific memory or I/O address ranges.

The hardware is capable of minimal levels of power management without interacting with the firmware at all. Once configured by the firmware, the DOZE timer and the CPU activity monitor (cool–down clocking mechanism) can slow the system clock to reduce power consumption. The DOZE timer monitors specific bus activity and reduces the system clock after periods of inactivity. The CDC mechanism simply guarantees that the CPU spends some specified portion of time at a reduced clock speed, either due to DOZE mode of firmware controlled reduced clock mode, or by forcing cool–down periods if neither of these occur.

In a stand–alone environment (no APM software active), the firmware works in conjunction with the hardware timers and monitoring functions to identify periods when certain devices or the entire system are inactive. Individual timers are supported for specific devices, including the hard disk, floppy disk and serial ports. Whenever these devices are not accessed for a specified period, they are powered down to reduce system power consumption. Whenever none of the monitored system devices have been accessed for a specified period of time, the performance of the system is reduced or the system is stopped altogether to further reduce power consumption.

In a cooperative environment, devices are still controlled by the firmware, but the CPU is never slowed or stopped without the consent of the APM software. Rather, the firmware notifies the software when a timer has expired or some other event has occurred which should place the system in a reduced power mode. The APM software polls the firmware for such events. Once an event has occurred, the software initiates the reduced power mode by acknowledging the event back to the firmware. The firmware then initiates the reduced power mode. The APM software can inquire "APM aware" applications to ensure that the reduced power mode is acceptable.

## ≡ Hardware controlled modes

The firmware assumes that keyboard, disk, and video access should prevent the system from entering DOZE mode. In addition, specific interrupts can be configured to reset the DOZE timer. DOZE mode is typically a reduced–clock operation, but a stopped–clock mode is also supported. If the slow–clock mode is selected, the CPU clock is slowed from 33 MHz to 8 MHz. If stopped–clock mode is selected, the secondary DOZE timer is configured to restart the CPU clock whenever the timer interrupt occurs. This allows the CPU to maintain the system clock while in stopped–clock mode even though the CPU clock remains stopped for more than 90% of the time.

The CPU activity monitor (cool–down clocking mechanism) is intended to prevent thermal run–away in low airflow environments. It essentially counts the time that the CPU operates in full–speed mode and the time spent in reduced clock mode. If full–speed mode is maintained for a period of time that could cause excessive heat build–up, then the clock is reduced for a cool–down period to maintain an equilibrium level. The CPU clock is shifted from 33 MHz to 16.5 MHz for a pre–determined amount of time, back to 33 MHz, and then the CDC cycle begins again.

## ≡ Device power management

The hard disk, floppy disk, and serial ports are power managed on an individual basis. For each device, the firmware configures a hardware timer that is reset each time the device is accessed. When the device goes unaccessed for the duration of the timer, an SMI is generated to notify the firmware that the device is inactive. The firmware can then SUSPEND that device.

Before a device is SUSPENDed, or powered down, its context is saved in memory so that it can be restored to its active state when it is powered up. The timer is disabled on SUSPEND to prevent repetitive triggering, and the chipset is configured to generate another SMI when the device is accessed again. That SMI allows the firmware to restore power to the device and restore its context.

After the device power is restored by a triggering event, such as a keyboard stroke, the access SMI is disabled and the timer is restarted. This cycle may then be repeated. The status of each device is maintained to ensure that a powered–down device is not accessed for power–down repeatedly.

Devices other than the disks and serial ports are not monitored for activity, but can be individually powered down in reduced power states at the system level.

## ≡ System power management

At the system level, power management is very similar to the device level management, with a couple of exceptions. Cooperative management is supported, allowing an APM driver, such as POWER.EXE, to control the actual power state transitions. This is done by identifying power management events and reporting them to the APM driver via a polling mechanism. Power state transitions then occur at the request of the APM driver. Individual device states in the various system power states can be configured via CMOS locations to the extent at which the firmware has the ability to control them. For example, the video can remain ON during STANDBY, which makes STANDBY mode more transparent to the user, or it can be powered OFF during STANDBY to further reduce power. The parallel port may be powered OFF in STANDBY, or remain ON in STANDBY and powered OFF in SUSPEND, but on return to full speed it must be powered ON because there is no access SMI available. The disks and serial ports, on the other hand, may remain powered OFF after the system RESUMEs and can be powered ON when accessed.

The IDLE timer can be reset by numerous sources, including device accesses and interrupts. Note that it is possible for the IDLE timer configuration to be of shorter duration than the device timers. This means that the system can be deemed IDLE even though some of the devices are still active. When this occurs, the device power states are set according to their configuration in CMOS.

Note that the APM interface prevents the system from entering STANDBY or SUSPEND modes directly. These modes will be entered, but that will occur through the APM interface (INT 15h) at the request of the APM driver.

SUSPEND mode is the lowest power state that the system can attain while still powered. The CPU clock is stopped and all controllable devices are powered down. Because the devices are powered down and the CPU is not running, only an external event can cause the

system to RESUME normal operation. These events include the COM2 RI signal (Ring Indicator), and interrupts from selected sources located in the .PMI file. The devices which are powered ON when the system RESUMEs are specified in CMOS, loaded from the .PMI file. Devices which do not have associated access SMIs, must be powered up. In addition, since the CPU was stopped, the system time must be updated. If an APM driver is operating, it has the responsibility of updating the time when notified to do so. Otherwise, the firmware will update the DOS compatible system time if configured to do so. For operating systems with DOS compatible system clocks, this function should be disabled in CMOS. Since the clock does not run in SUSPEND mode and the system is not restarted by IRQ0 to maintain the time of day, the time must be reset when the system resumes. Normally, the BIOS can read the actual time from the real time clock and restore the operating system's timer from that value. However, some operating systems do not support the update methods utilized by the BIOS for this function. The time update can be enabled or disabled using the 2133 SETUP program. In SETUP, the following options are available:

| | |
|---|---|
| Power Management: | ENABLED or DISABLED |
| DOZE Clock (Slow, Stop): | SLOW or STOP |
| Time Updated After Suspend: | ENABLED or DISABLED |

## How to initiate the SUSPEND/RESUME option

1. In 2133 SETUP and in a .PMI file, enable power management and select the following options:

   SETUP:

   | | |
   |---|---|
   | Power Management: | ENABLED |
   | DOZE Clock (Slow, Stop): | SLOW |
   | Time Updated After Suspend: | ENABLED |

   TEST.PMI file:

   | | |
   |---|---|
   | pmi–enable=Y | Enables power management |

2. If SUSPEND is to be generated by a time–out, then in the same .PMI file, set the time delay, in minutes, for the "suspend–delay" option.

   TEST.PMI file:

   | | |
   |---|---|
   | suspend–delay=xx | Sets delay time before SUSPEND (x=0–31 minutes) |

3. In the same .PMI file, select any signal that will cause the system to resume. The options include IRQs and COM2 RI.

   TEST.PMI file:

   | | |
   |---|---|
   | irq1–resume–reset=Y | Enables RESUME if IRQ1 occurs (keyboard input) |
   | irq4–resume–reset=Y | Enables RESUME if IRQ4 occurs (COM1 input) |
   | RI–resume–reset=Y | Enables RESUME if RI occurs (COM2 ring indicator) |
   | RI–transitions=5 | Specifies number of RI transitions (n) for RESUME (COM2 ring indicator) |

4. Load the .PMI file changes by including the .PMI file on the PMISETUP command line. PMISETUP is located in the \UTILS directory:

   ```
   C:\> PMISETUP TEST.PMI
   ```

5. Reset the system for the PMISETUP options to take effect. The 2133 PC/104 CPU system is now ready for SUSPEND/RESUME.

6. Initiate a SUSPEND by allowing the "suspend–delay" timer to expire.

7. The 2133 PC/104 CPU system enters the SUSPEND mode, powering down any devices that can be powered down.

8. A system RESUME is generated by any event defined in the .PMI file. If a monitor is attached, the video will return. If a hard drive is attached, the drive will not spin up until accessed.

   WAKEIRQ8.EXE is a sample TSR that wakes the CPU, using the CMOS clock, from SUSPEND mode after a 30 second delay. WAKEIRQ8.EXE and WAKEIRQ8 files are available in the \EXAMPLES directory on the 2133 PC/104 CPU utility disk.

   Refer to the default .PMI file in the Power management configuration section in this chapter for all of the power management options.

   STANDBY mode is similar to hardware DOZE mode, except that it is firmware controlled and devices may be power managed as well as the CPU clock. STANDBY is controlled by software, such as POWER.EXE. Since the CPU may still be executing in STANDBY mode, access events may RESUME the system in addition to external inputs, such as interrupts. The firmware must configure the chipset to report these events to the firmware via an SMI. This allows the hardware to restore the CPU clock while notifying the firmware to restore power to the managed devices. On RESUME from STANDBY mode, all devices which cannot generate an access SMI are powered up and those which can generate an access SMI are powered up if specified in CMOS. Any device access or specified interrupt can cause the system to RESUME.

## How to initiate the STANDBY option

1. In the 2133 SETUP and in a .PMI file, enable power management and select the following options:

   SETUP:

   | | |
   |---|---|
   | Power Management: | ENABLED |
   | DOZE Clock (Slow, Stop): | SLOW |
   | Time Updated After Suspend: | ENABLED |

   TEST.PMI:

   | | |
   |---|---|
   | pmi–enable=Y | Enables the power management |
   | standby–delay=xx | Sets delay time before STANDBY (x=0–31 minutes) |
   | fdd–to=xx | Sets the on–card (LPT1) floppy disk drive time–out value (x=0–31 minutes) |
   | hdd–to=xx | Sets the hard drive time–out value (x=0–31 minutes) |
   | COM1–to=xx | Sets the COM1 time–out value (x=0–31 minutes) |
   | COM2–to=xx | Sets the COM2 time–out value (x=0–31 minutes) |

   Set the time delay, in minutes, for the "standby–delay" option and devices to be powered down, such as drives and serial ports.

Note The hard drive and floppy drives are actually controlled by firmware and not by software such as POWER.EXE. The floppy must be on–card, controlled from the LPT1 port.

2. In the same .PMI file, select any signal that will cause the system to resume. The options only include IRQs.

   TEST.PMI file:

   | | |
   |---|---|
   | irq1–standby–reset=Y | Enables reset of STANDBY mode if IRQ1 occurs |
   | irq3–standby–reset=Y | Enables reset of STANDBY mode if IRQ3 occurs |
   | irq4–standby–reset=Y | Enables reset of STANDBY mode if IRQ4 occurs |
   | irq5–standby–reset=Y | Enables reset of STANDBY mode if IRQ5 occurs |
   | irq6–standby–reset=Y | Enables reset of STANDBY mode if IRQ6 occurs |
   | irq7–standby–reset=Y | Enables reset of STANDBY mode if IRQ7 occurs |

| | |
|---|---|
| irq8–standby–reset=Y | Enables reset of STANDBY mode if IRQ8 occurs |
| irq14–standby–reset=Y | Enables reset of STANDBY mode if IRQ14 occurs |

3. Load the .PMI file changes by including the .PMI file on the PMISETUP command line. PMISETUP is located in the \UTILS directory:

   ```
   C:\> PMISETUP TEST.PMI
   ```

4. Reset the system for the PMISETUP options to take effect. The 2133 is now ready to enter the STANDBY mode.
5. Initiate APM software, such as POWER.EXE.
6. The 2133 enters the SUSPEND mode as determined by the APM software, powering down any devices that can be powered down.
7. A system RESUME is generated by any event defined in the .PMI file or when an external RESUME signal is applied. If a monitor is attached, the video will return. If a hard drive is attached, the drive will not spin up until accessed.

   Refer to the default .PMI file in the Power management configuration section in this chapter for all of the power management options.

## ≡ Save to disk

The firmware provides an additional SUSPEND mode which supports removal of system power. This mode is initiated through software or by the external power management input. In this mode, the context of the entire system is saved on the hard disk so that it can be restored completely when power is restored to the system. The system context includes all of the system memory, video memory, and the states of the hardware registers in all devices (interrupt controllers, DMA controllers, serial ports, keyboard controller, etc.). The firmware can only save and restore devices of which it is "aware", meaning those that are on–board and most standard VGA video controllers. This function may not be fully functional for some extended system configurations. This function will only work with a hard drive and not with the on–board SSD flash drive or with a flash drive card.

After a Save To Disk function has been completed, the system must be reset to restore the context. After saving the context of the system to disk, all devices are powered down and the CPU is SUSPENDed. The validity of the system context on the hard disk is identified by marking a bit in the configuration EEPROM. On the next system reset, the validity of the context is checked in EEPROM and, if valid, is restored to the system memory and devices.

In order for the Save to Disk function to execute, the hard disk must be prepared using the PHDISK.EXE program. A special partition is saved on the disk which must be big enough to hold the system context. Generally speaking, this means the size of the installed system memory (typically 1 to 17 MB DRAM) plus the size of the video memory (typically 512 KB to 1 MB). However, a larger partition may be created in order to support additional system memory to be added at a later time.

> WARNING!
>
> PHDISK.EXE provides the same function as FDISK but also creates a special partition for the Save to Disk option. Only use a hard drive on which any existing data can be destroyed.

The Save To Disk function may be disabled. If it is disabled, then the power management firmware will, upon detection of a SUSPEND event, execute an interrupt 15h with register AX set to 0AA55h and register DX set to 1234h. This indicates a power down request when the PMI input signal is generated. PFHINT15.EXE is a sample TSR that prints "PFH" to the screen when the PMI signal is generated but the "save to disk" option is disabled. The PFHINT15.EXE and PFHINT15.CPP files are located in the \EXAMPLES directory on the 2133 utility disk.

## How to initiate the save to disk option

1. Use PHDISK.EXE, located in the \UTILS directory on the 2133 utility disk, to prepare the hard drive.
2. In 2133 SETUP and in a .PMI file, enable power management and select the following options:

   SETUP:

   | | |
   |---|---|
   | Power Management: | ENABLED |
   | DOZE Clock (Slow, Stop): | SLOW |
   | Time Updated After Suspend: | ENABLED |

   TEST.PMI:

   | | |
   |---|---|
   | pmi–enable=Y | Enables the power management |
   | save–to–disk=Y | Enables save to disk feature |
   | pfh–enable=Y | Enables power fail option |
   | pfh–reset=N | Disables power fail reset |

3. Load the .PMI file changes by including the .PMI file on the PMISETUP command line. PMISETUP is located in the \UTILS directory:

   ```
   C:\> PMISETUP TEST.PMI
   ```

4. Reset the system for the PMISETUP options to take effect. The 2133 is now ready to Save to Disk.
5. Initiate a SUSPEND by allowing the "suspend–delay" timer to expire.
6. The system now performs a contextual save, saving all relevant information to the special disk partition.
7. The system will respond with "Please turn off your computer." Power down the system. The CPU enters SUSPEND mode. The validity of the system context on the hard disk is identified by marking a bit in the configuration EEPROM.
8. When the system is powered on, the validity of the context is checked in EEPROM. If valid, the system's context is restored to the system memory and devices, back to where it was before the PMI signal was generated.

# ≡ Power management configuration

The power management functions can be globally enabled or disabled in CMOS. The 2133 SETUP.EXE and PMISETUP.EXE utilities provide an option for enabling or disabling power management.

## Enabling power management

Whichever utility, SETUP or PMISETUP, is configured and saved (or loaded) last, that configuration is used for enabling or disabling power management. In other words, if the power management option is disabled in SETUP and then later a .PMI file which shows pmi–enable = Y is loaded with PMISETUP, the power management option in SETUP will now show power management as being enabled.

In 2133 SETUP, the following options are available:

| | |
|---|---|
| Power Management: | ENABLED or DISABLED |
| DOZE Clock (Slow, Stop): | SLOW or STOP |
| Time Updated After Suspend: | ENABLED or DISABLED |

In a .PMI file, the following option is available:

| | |
|---|---|
| pmi–enable=Y/N | Disables/enables the power management code |

## System timers

The system timers define the delays associated with power state transitions in the system. Three managed system power modes (in addition to full on) are supported, DOZE, STANDBY, and SUSPEND. Individual timers for system devices, including floppy disk, hard disk, and serial ports are also provided. The timers are described below:

In a .PMI file, the following options are available:

| | |
|---|---|
| fdd–to=xx | Sets the on–card (LPT1) floppy disk drive time–out value (x=0–31 minutes) |
| hdd–to=xx | Sets the hard drive time–out value (x=0–31 minutes) |
| COM1–to=xx | Sets the COM1 time–out value (x=0–31 minutes) |
| COM2–to=xx | Sets the COM2 time–out value (x=0–31 minutes) |
| doze–delay=x | Sets DOZE time–out before STANDBY (x=0,2,8 seconds) |
| standby–delay=xx | Sets delay time before STANDBY (x=0–31 minutes) |
| suspend–delay=xx | Sets delay time before SUSPEND (x=0–31 minutes) |

## Doze timer enable and resets

The DOZE mode may be selected to be a SLOW clock or STOP clock mode and is determined in power management SETUP. Numerous sources may reset the DOZE timer, preventing entry to DOZE mode. The BIOS assumes that video, floppy, hard disk, and keyboard accesses should all reset the DOZE timer. In addition, various interrupts can be configured to reset the timer. DOZE timer 0 is used for all DOZE reset selections. DOZE timer 1 is used for the special case of the timer interrupt in a STOP clock mode. In this mode, DOZE timer 1 is configured for a 4 millisecond time–out, with only IRQ0 configured to reset DOZE timer 1. This allows the CPU to resume normal operation for 4 mS on each IRQ0 in order to maintain the time of day.

In the 2133 PC/104 CPU SETUP, the following options are available:

| | |
|---|---|
| Power Management: | ENABLED or DISABLED |
| DOZE Clock (Slow, Stop): | SLOW or STOP |
| Time Updated After Suspend: | ENABLED or DISABLED |

Interrupts that reset DOZE are configured in a .PMI file:

| | |
|---|---|
| irq0–reset–doze=Y/N | Enables reset of DOZE clock if IRQ0 occurs |
| irq3–reset–doze=Y/N | Enables reset of DOZE clock if IRQ3 occurs |
| irq4–reset–doze=Y/N | Enables reset of DOZE clock if IRQ4 occurs |

| | |
|---|---|
| irq5–reset–doze=Y/N | Enables reset of DOZE clock if IRQ5 occurs |
| irq7–reset–doze=Y/N | Enables reset of DOZE clock if IRQ7 occurs |
| irq8–reset–doze=Y/N | Enables reset of DOZE clock if IRQ8 occurs |
| irq12–reset–doze=Y/N | Enables reset of DOZE clock if IRQ12 occurs |
| irq13–reset–doze=Y/N | Enables reset of DOZE clock if IRQ13 occurs |

## IDLE timer resets

The IDLE timer monitors system activity to prevent the system from entering STANDBY or SUSPEND modes if bus activity indicates that the system is busy. Access to these devices will also cause the system to RESUME from STANDBY mode. The bus activities that are monitored are configured in a .PMI file:

| | |
|---|---|
| LCD–reset–idle=Y/N | LCD/VGA access resets IDLE timer |
| COM1–reset–idle=Y/N | Enables reset of IDLE timer if COM1 access occurs |
| COM2–reset–idle=Y/N | Enables reset of IDLE timer if COM2 access occurs |
| LPT–reset–idle=Y/N | Enables reset of IDLE timer if LPT access occurs |
| KBD–reset–idle=Y/N | Enables reset of IDLE timer if Keyboard access occurs |
| FDD–reset–idle=Y/N | Enables reset of IDLE timer if Floppy Disk Drive access occurs |
| HDD–reset–idle=Y/N | Enables reset of IDLE timer if Hard Disk Drive access occurs |

Interrupts in the system can also reset the IDLE timer to prevent entry into reduced power modes. These interrupts should be enabled to reset the IDLE timer if they indicate that the system is active. The interrupts to reset the IDLE timer are configured in a .PMI file:

| | |
|---|---|
| irq0–reset–idle=Y/N | Enables reset of IDLE clock if IRQ0 occurs |
| irq1–reset–idle=Y/N | Enables reset of IDLE clock if IRQ1 occurs |
| irq3–reset–idle=Y/N | Enables reset of IDLE clock if IRQ3 occurs |
| irq4–reset–idle=Y/N | Enables reset of IDLE clock if IRQ4 occurs |
| irq5–reset–idle=Y/N | Enables reset of IDLE clock if IRQ5 occurs |
| irq6–reset–idle=Y/N | Enables reset of IDLE clock if IRQ6 occurs |
| irq7–reset–idle=Y/N | Enables reset of IDLE clock if IRQ7 occurs |
| irq8–reset–idle=Y/N | Enables reset of IDLE clock if IRQ8 occurs |
| irq9–reset–idle=Y/N | Enables reset of IDLE clock if IRQ9 occurs |
| irq10–reset–idle=Y/N | Enables reset of IDLE clock if IRQ10 occurs |
| irq11–reset–idle=Y/N | Enables reset of IDLE clock if IRQ11 occurs |
| irq12–reset–idle=Y/N | Enables reset of IDLE clock if IRQ12 occurs |

| | |
|---|---|
| irq13–reset–idle=Y/N | Enables reset of IDLE clock if IRQ13 occurs |
| irq14–reset–idle=Y/N | Enables reset of IDLE clock if IRQ14 occurs |
| irq15–reset–idle=Y/N | Enables reset of IDLE clock if IRQ15 occurs |

## RESUME from STANDBY

The events which can resume the system from STANDBY mode to full power can also be selected. Access to the primary devices can resume the system if access to those devices are selected to reset the IDLE timer. In addition, interrupts can be selected to resume the system from STANDBY mode. The interrupts to resume from STANDBY are configured in a .PMI file:

| | |
|---|---|
| irq0–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ0 occurs |
| irq1–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ1 occurs |
| irq3–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ3 occurs |
| irq4–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ4 occurs |
| irq5–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ5 occurs |
| irq6–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ6 occurs |
| irq7–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ7 occurs |
| irq8–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ8 occurs |
| irq9–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ9 occurs |
| irq10–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ10 occurs |
| irq11–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ11 occurs |
| irq12–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ12 occurs |
| irq13–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ13 occurs |
| irq14–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ14 occurs |
| irq15–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ15 occurs |

When the system resumes from STANDBY mode, the CPU clock is restored to full speed. In addition, some or all of the devices are restored to full power. The video (if suspended in STANDBY mode) and parallel port, are always powered up. The following devices may be held in SUSPEND until they are accessed:

- COM1
- COM2
- On–board floppy disk (LPT1 port)
- Hard disk

## RESUME from SUSPEND

SUSPEND mode is the lowest power mode supported by power management. In this mode all devices are powered down and the CPU is placed

in SUSPEND mode and the clock is stopped. In addition to the SUSPEND/RESUME input, the activities that may resume the system from SUSPEND mode can be selected in the .PMI file.

The Ring Indicator from COM2 can RESUME the system after a specified number of transitions. This function may be enabled and the number of transitions specified in the .PMI file.

| | |
|---|---|
| RI–resume–reset=Y/N | Enables RESUME if RI occurs (COM2 ring indicator) |
| RI–transitions=n | Specifies number of RI transitions (n) for RESUME (COM2 ring indicator) |

Interrupts may cause the system to RESUME. This is particularly useful for activities such as a key on the keyboard being pressed. If any interrupt is to cause the system to RESUME, then the interrupt must be selected in the .PMI file.

| | |
|---|---|
| resume–irqs–enable=Y/N | Enables RESUME from RESUME IRQs |
| irq1–resume–reset=Y/N | Enables RESUME if IRQ1 occurs |
| irq3–resume–reset=Y/N | Enables RESUME if IRQ3 occurs |
| irq4–resume–reset=Y/N | Enables RESUME if IRQ4 occurs |
| irq5–resume–reset=Y/N | Enables RESUME if IRQ5 occurs |
| irq7–resume–reset=Y/N | Enables RESUME if IRQ7 occurs |
| irq8–resume–reset=Y/N | Enables RESUME if IRQ8 occurs |
| irq9–resume–reset=Y/N | Enables RESUME if IRQ9 occurs |
| irq10–resume–reset=Y/N | Enables RESUME if IRQ10 occurs |
| irq11–resume–reset=Y/N | Enables RESUME if IRQ11 occurs |
| irq12–resume–reset=Y/N | Enables RESUME if IRQ12 occurs |
| irq14–resume–reset=Y/N | Enables RESUME if IRQ14 occurs |
| irq15–resume–reset=Y/N | Enables RESUME if IRQ15 occurs |

## Remote suspend/resume inputs

If the required options are selected in PMISETUP.EXE, the ring indicator from a modem connected to COM2, will cause the 2133 to resume.

## External event

An external event can be used to generate a software interrupt. The power management code is capable of properly resetting the system, saving the state of the system to disk, or generating an INT 15h. Once activated, the system either performs a System Reset, performs a Save to Disk operation, or generates an INT 15h. These functions are enabled in a .PMI file.

1. System Reset

   The PMI file contains:

   | | |
   |---|---|
   | pfh–enable=Y | Enables power fail options |
   | pfh–reset=Y | Enables system reset |

2. Save to Disk

   The PMI file contains:

   | | |
   |---|---|
   | pfh–enable=Y | Enables power fail options |
   | pfh–reset=N | Disables system reset |
   | save–to–disk=Y | Enables save to disk feature |

Refer to the Save to disk section in this chapter for more information.

3. INT 15h

   The PMI file contains:

   | | |
   |---|---|
   | pfh–enable=Y | Enables power fail options |
   | pfh–reset=N | Disables system reset |
   | save–to–disk=N | Disables save to disk feature |

4. To disable all three options:

   The PMI file contains:

   | | |
   |---|---|
   | pfh–enable=N | Disables power fail options |

## Thermal management

Automatic thermal management of the CPU is provided by measuring the full speed operation of the CPU verses a thermal equilibrium value. During thermal management, CPU activity is monitored and the activity is considered against other criteria. When determined that the activity level has exceeded a certain level, the CPU speed is divided in half from 33 MHz to 16.5 MHz for about 30 seconds, the CPU speed is increased to 33 MHz and the thermal management cycle begins again. If thermal management is disabled, the CPU will run "full on" without any intervention from the firmware. Default configuration from Octagon has thermal management enabled.

The option for thermal management is in a .PMI file:

| | |
|---|---|
| thermal–management=Y/N | Enables thermal management features (cool–down clocking ) |

# ≡ PMISETUP

The PMISETUP program allows modification of the 82C465 registers and optionally the serial EEPROM data associated with the 82C465/ CMOS registers. The command line format for the PMISETUP program is as follows:

**PMISETUP** *inputfilename* [**/SHOWALL**] [**/DEFAULT**] [**/P**] [**/?**]

PMISETUP will parse a file using KEYWORD=nn[type] options. One keyword can be specified per line, characters after a semicolon ";" will be regarded as comments. Blank lines are ignored. If multiple lines contain the same keyword, the last value will be used. Keywords and values are not case sensitive.

A sample input file, TEST.PMI, is as follows:

| | |
|---|---|
| pmi–enable=Y | Disables/enables the power management code |
| doze–delay=2 | Sets DOZE time–out to 2 seconds before standby |
| hdd–to=10 | Sets the hard drive time–out to 10 minutes. |

Load the .PMI file changes by including the .PMI file on the PMISETUP command line. PMISETUP is located in the \UTILS directory:

```
C:\> PMISETUP TEST.PMI
```

The output from PMISETUP will show all selectable options and their current values even when not specified when the /SHOWALL option is used.

When the /DEFAULT option is used, default settings are first loaded before the keywords are parsed. Refer to the default PMI configuration later in this chapter.

When the /P option is used with the /SHOWALL, the display pauses between each displayed page of information.

When the /? option is used, all other options are ignored and a "help display" is presented to the user. A <CR> keystroke between each page is necessary to continue to the next page of help.

If an error occurs during parsing, the line in question will be displayed and an ERRORLEVEL returned. For example:

```
>> Parameter out of range error. Line follows:
hdd–to=50
The correct range is 0–31 minutes
or
>> Unknown keyword error in input file line 7. Line
follows:
standbuy–delay=10
Standby was spelled incorrectly.
```

The following ERRORLEVELs will be returned:

0 — No errors occurred.

1 — Unable to save to serial EEPROM.

2 — Configuration file not found.

3 — Parsing error or out of range error.

In PMISETUP, the following options are available and can be included in a .PMI file which is loaded by PMISETUP.EXE. PMISETUP.EXE is located in the \UTILS directory:

The following are the keywords which are used. To disable options, either an "N" or "0" is used, depending on the option.

| | |
|---|---|
| pmi–enable=Y/N | Disables/enables the power management code |
| fdd–to=xx | Sets the on–card (LPT1) floppy disk drive time–out value (x=0–31 minutes) |
| hdd–to=xx | Sets the hard drive time–out value (x=0–31 minutes) |
| COM1–to=xx | Sets the COM1 time–out value (x=0–31 minutes) |
| COM2–to=xx | Sets the COM2 time–out value (x=0–31 minutes) |
| doze–delay=x | Sets DOZE time–out before STANDBY (x=0,2,8 seconds) |
| standby–delay=xx | Sets delay time before STANDBY (x=0–31 minutes) |
| suspend–delay=xx | Sets delay time before SUSPEND (x=0–31 minutes) |

| | |
|---|---|
| irq0–reset–doze=Y/N | Enables reset of DOZE clock if IRQ0 occurs |
| irq3–reset–doze=Y/N | Enables reset of DOZE clock if IRQ3 occurs |
| irq4–reset–doze=Y/N | Enables reset of DOZE clock if IRQ4 occurs |
| irq5–reset–doze=Y/N | Enables reset of DOZE clock if IRQ5 occurs |
| irq7–reset–doze=Y/N | Enables reset of DOZE clock if IRQ7 occurs |
| irq8–reset–doze=Y/N | Enables reset of DOZE clock if IRQ8 occurs |
| irq12–reset–doze=Y/N | Enables reset of DOZE clock if IRQ12 occurs |
| irq13–reset–doze=Y/N | Enables reset of DOZE clock if IRQ13 occurs |
| irq0–reset–idle=Y/N | Enables reset of IDLE clock if IRQ0 occurs |
| irq1–reset–idle=Y/N | Enables reset of IDLE clock if IRQ1 occurs |
| irq3–reset–idle=Y/N | Enables reset of IDLE clock if IRQ3 occurs |
| irq4–reset–idle=Y/N | Enables reset of IDLE clock if IRQ4 occurs |
| irq5–reset–idle=Y/N | Enables reset of IDLE clock if IRQ5 occurs |
| irq6–reset–idle=Y/N | Enables reset of IDLE clock if IRQ6 occurs |
| irq7–reset–idle=Y/N | Enables reset of IDLE clock if IRQ7 occurs |
| irq8–reset–idle=Y/N | Enables reset of IDLE clock if IRQ8 occurs |
| irq9–reset–idle=Y/N | Enables reset of IDLE clock if IRQ9 occurs |

| | |
|---|---|
| irq10–reset–idle=Y/N | Enables reset of IDLE clock if IRQ10 occurs |
| irq11–reset–idle=Y/N | Enables reset of IDLE clock if IRQ11 occurs |
| irq12–reset–idle=Y/N | Enables reset of IDLE clock if IRQ12 occurs |
| irq13–reset–idle=Y/N | Enables reset of IDLE clock if IRQ13 occurs |
| irq14–reset–idle=Y/N | Enables reset of IDLE clock if IRQ14 occurs |
| irq15–reset–idle=Y/N | Enables reset of IDLE clock if IRQ15 occurs |
| save–to–disk=Y/N | Enables save to disk feature |
| thermal–management=Y/N | Enables thermal management features (CDC) |
| irq0–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ0 occurs |
| irq1–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ1 occurs |
| irq3–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ3 occurs |
| irq4–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ4 occurs |
| irq5–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ5 occurs |
| irq6–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ6 occurs |
| irq7–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ7 occurs |
| irq8–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ8 occurs |
| irq9–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ9 occurs |
| irq10–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ10 occurs |
| irq11–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ11 occurs |
| irq12–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ12 occurs |
| irq13–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ13 occurs |
| irq14–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ14 occurs |
| irq15–standby–reset=Y/N | Enables reset of STANDBY mode if IRQ15 occurs |
| resume–irqs–enable=Y/N | Enables RESUME from RESUME IRQ group |
| irq1–resume–reset=Y/N | Enables RESUME if IRQ1 occurs |
| irq3–resume–reset=Y/N | Enables RESUME if IRQ3 occurs |
| irq4–resume–reset=Y/N | Enables RESUME if IRQ4 occurs |
| irq5–resume–reset=Y/N | Enables RESUME if IRQ5 occurs |
| irq7–resume–reset=Y/N | Enables RESUME if IRQ7 occurs |
| irq8–resume–reset=Y/N | Enables RESUME if IRQ8 occurs |
| irq9–resume–reset=Y/N | Enables RESUME if IRQ9 occurs |
| irq10–resume–reset=Y/N | Enables RESUME if IRQ10 occurs |
| irq11–resume–reset=Y/N | Enables RESUME if IRQ11 occurs |
| irq12–resume–reset=Y/N | Enables RESUME if IRQ12 occurs |
| irq14–resume–reset=Y/N | Enables RESUME if IRQ14 occurs |
| irq15–resume–reset=Y/N | Enables RESUME if IRQ15 occurs |

| | |
|---|---|
| RI–resume–reset=Y/N | Enables RESUME if RI occurs (COM2 ring indicator) |
| RI–transitions=n | Specifies number of RI transitions (n) for RESUME (COM2 ring indicator) |
| LCD–reset–idle=Y/N | LCD/VGA access resets IDLE timer |
| COM1–reset–idle=Y/N | Enables reset of IDLE timer if COM1 access occurs |
| COM2–reset–idle=Y/N | Enables reset of IDLE timer if COM2 access occurs |
| LPT–reset–idle=Y/N | Enables reset of IDLE timer if LPT access occurs |
| KBD–reset–idle=Y/N | Enables reset of IDLE timer if Keyboard access occurs |
| FDD–reset–idle=Y/N | Enables reset of IDLE timer if Floppy Disk access occurs |
| HDD–reset–idle=Y/N | Enables reset of IDLE timer if Hard Disk Drive access occurs |
| pfh–enable=Y/N | Enables power fail options |
| pfh–reset=Y/N | Enables power fail reset |

PMISETUP default PMI configuration

```
pmi–enable = Y
fdd–to = 0
hdd–to = 0
COM1–to = 0
COM2–to = 0
doze–delay = 8
standby–delay = 0
suspend–delay = 0
irq0–reset–doze = N
irq3–reset–doze = N
irq4–reset–doze = Y
irq5–reset–doze = N
irq7–reset–doze = N
irq8–reset–doze = N
irq12–reset–doze = N
irq13–reset–doze = Y
irq0–reset–idle = N
irq1–reset–idle = Y
irq3–reset–idle = Y
irq4–reset–idle = Y
irq5–reset–idle = Y
irq6–reset–idle = N
irq7–reset–idle = Y
irq8–reset–idle = Y
irq9–reset–idle = N
irq10–reset–idle = N
irq11–reset–idle = N
```

```
irq12–reset–idle = N
irq13–reset–idle = Y
irq14–reset–idle = Y
irq15–reset–idle = N
save–to–disk = N
thermal–management = Y
irq0–standby–reset = N
irq1–standby–reset = Y
irq3–standby–reset = Y
irq4–standby–reset = Y
irq5–standby–reset = Y
irq6–standby–reset = Y
irq7–standby–reset = Y
irq8–standby–reset = Y
irq9–standby–reset = N
irq10–standby–reset = N
irq11–standby–reset = N
irq12–standby–reset = N
irq13–standby–reset = N
irq14–standby–reset = Y
irq15–standby–reset = N
resume–irqs–enable = Y
irq1–resume–reset = Y
irq3–resume–reset = Y
irq4–resume–reset = Y
irq5–resume–reset = N
irq7–resume–reset = N
irq8–resume–reset = Y
irq9–resume–reset = N
irq10–resume–reset = N
irq11–resume–reset = N
irq12–resume–reset = N
irq14–resume–reset = N
irq15–resume–reset = N
RI–resume–reset = Y
RI–transitions = 5
EPMI–resume–reset = Y
EPMI–reset–idle = Y
LCD–reset–idle = Y
COM1–reset–idle = Y
COM2–reset–idle = Y
LPT–reset–idle = Y
KBD–reset–idle = Y
FDD–reset–idle = Y
HDD–reset–idle = Y
CDC–beep = Y
COM1–suspend=Y
COM2–suspend=Y
```

# *Chapter 14:* *Serial EEPROM*

## ≡ Description

Up to 768 words of user-definable data can be saved in the serial EEPROM. The serial EEPROM does not require battery backup to maintain the data when the system power is off. The serial EEPROM is easily accessible via software interrupts by most programming languages.

The serial EEPROM definitions include the following functions: Read a single word from serial EEPROM, Write a single word to serial EEPROM, Read multiple words from serial EEPROM, Write multiple words to serial EEPROM, and Return serial EEPROM size.

These functions require the INT17 handler (I17HNDLR.EXE). I17HNDLR.EXE is a TSR program. Once executed it is active, but it must be executed each time the system is rebooted. If the INT17 functions will be used by your application, copy the utility to your hard drive and add it to your AUTOEXEC.BAT.

## ≡ Serial EEPROM

### Read a single word from the serial EEPROM

Function: fch

Subfunction: 00h

Purpose: To read a single word from the on-board serial EEPROM.

Calling registers:

| | |
|---|---|
| AH | fch |
| AL | 00h |
| BX | Word address (zero based) |
| DX | ffffh (relative to user area) |

Return registers:

Carry flag cleared if successful

| | |
|---|---|
| AX | Word read |

Carry flag set if error

| | |
|---|---|
| AL | Error code |

| Error code | Meaning |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function reads a word from the user area of the serial EEPROM.

Programming example:

```
/* Read word 2 */
unsigned int seeData;
/* Inline assembly code for Borland C++ 3.1 */
asm {
mov  ax,0fc00h
mov  bx,02h     /* Read word 2 */
mov  dx,0ffffh
int  17h
mov  seeData,ax /* store data in c environment */
    }
```

## Write a single word to the serial EEPROM

Function: fch

Subfunction: 01h

Purpose: To write a single word to the on-board serial EEPROM.

Calling registers:

| | |
|---|---|
| AH | fch |
| AL | 01h |
| BX | Word address (zero based) |
| CX | Data word to write |
| DX | ffffh (relative to user area) |

Return registers:

Carry flag cleared if successful
Carry flag set if error
AL Error code

| Error code | Meaning |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function writes a word to the user area of the serial EEPROM.

Programming example:

```
/* Write 0x1234 to word 3*/
unsigned int seeData = 0x1234;
/* Inline assembly code for Borland C++ 3.1 */
asm {
     mov  ax,0fc01h
     mov  bx,03h     /* Write word 3 */
     mov  cx,seeData /* Get write data from
                        c environment */
     mov  dx,0ffffh
     int  17h
    }
```

## Read multiple words from the serial EEPROM

Function: fch

Subfunction: 02h

Purpose: To read multiple words from the on-board serial EEPROM.

Calling registers:

| | |
|---|---|
| AH | fch |
| AL | 02h |
| BX | Word address (zero based) |
| CX | Word count |
| DX | ffffh (relative to user area) |
| ES:DI | Destination pointer |

Return registers:

Carry flag cleared if successful

| | |
|---|---|
| AX | Word read |

Carry flag set if error

| | |
|---|---|
| AL | Error code |

| Error Code | Meaning |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function reads multiple words from the user area of the serial EEPROM.

Programming example:

```
/* Read 10 words starting at word 5 */
unsigned int far *seeDataPtr = new unsigned int[10];
/* Allocate storage*/
/* Inline assembly code for Borland C++ 3.1 */
asm {
    mov   ax,0fc02h
    mov   bx,05h     /* Read starts at word 5 */
    mov   cx,10      /* Read 10 words */
    mov   dx,0ffffh
    les   di,seeDataPtr
    int   17h
    }
```

## Write multiple words to the serial EEPROM

Function: fch

Subfunction: 03h

Purpose: To write multiple words to the on-board serial EEPROM.

Calling registers:

| | |
|---|---|
| AH | fch |
| AL | 03h |
| BX | Word address (zero based) |

CX Word count
DX ffffh (user area relative address)
DS:SI Source pointer

Return registers: Carry flag cleared if successful
Carry flag set if error
AL Error code

| Error Code | Meaning |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function writes multiple words to the user area of the serial EEPROM.

Programming example:

```
/* Write 8 words starting at word 6*/
unsigned int far *seeDataPtr = new unsigned int[8];
/* Allocate storage*/
    unsigned int far* tmpPtr = seeDataPtr;
    for(int i=0;i<8;i++)
    *seeDataPtr = i;/* initialize data */
/* Inline assembly code for Borland C++ 3.1 */
    asm {
    push ds
    mov  ax,0fc03h
    mov  bx,06h     /* Write starts at word 6 */
    mov  cx,8       /* Write 8 words */
    mov  dx,0ffffh
    lds  si,seeDataPtr
    int  17h
    pop  ds
         }
```

## Return serial EEPROM size

Function: fch

Subfunction: 04h

Purpose: To obtain the size of the on-board serial EEPROM.

Calling registers: AH fch
AL 04h
DX ffffh

Return registers: Carry flag cleared if successful
AX Size of the serial EEPROM (in words)
BX Size available to user (in words)
Carry flag set if error
AL Error code

| Error code | Meaning |
|---|---|

| | |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function returns the size (in words) of the serial EEPROM. Since the user cannot access all of the serial EEPROM, this function determines how much space is available to the user. This avoids the user from accessing unavailable addresses.

Programming example:

```
unsigned int seeUserSize;
/* Inline assembly code for Borland C++ 3.1 */
asm {
      mov   ax,0fc04h
      mov   dx,0ffffh
      int   17h
      mov   seeUserSize,bx
      }
```

## Check CMOS battery

Function: fbh

Subfunction: 08h

Purpose: To check CMOS battery condition.

Calling registers:

| | |
|---|---|
| AH | fbh |
| AL | 08h |
| DX | ffffh |

Return registers: Carry flag cleared if successful
ZF set = battery okay, ZF clear = battery bad
AL copy of CMOS register 0Eh at powerup time
Carry flag set if error
AL Error code

| Error Code | Meaning |
|---|---|
| ffh | Unknown error |
| 01h | Function not implemented |
| 02h | Defective serial EEPROM |
| 03h | Illegal access |

Comments: This function reports the condition of the CMOS battery. This is useful to determine if extended CMOS data (contents) should be relied upon or refreshed from EEPROM.

Programming example:

```
/* Reports the condition of the CMOS battery */
   unsigned int cmosflag;
/* Inline assembly code for Borland C++ 3.1 */
   asm {
         mov   ax,0fb08h
```

```
            mov   dx,0ffffh
            int   17h
            mov   cmosflag,al
          }
printf("The CMOS byte 0E at powerup time  = %02x\n",
cmosflag)
```

# Chapter 15: Troubleshooting

If your system is not working properly, check the following items:

## No screen activity – checking console serial communications

If the message "press C to try again or S for setup" appears, the boot did not complete. The most likely reason is that the DOC does not have an operating system installed. Press S to enter Setup, then configure LPT1 as a floppy drive, number of floppy drives to 1, and floppy drive size to the size of the floppy you will be installing. Save the Setup changes, then power off the 2133. Ensure that the S jumper is on so that the next time the 2133 is powered, it will use the parameters from the Setup stored in EEPROM. Install a floppy disk drive on LPT1, insert a boot disk in the floppy disk drive, then reboot. Refer to you DOC manual for information on how to load an operating system on the DOC.

If you do not get the sign-on message after bootup:

1. Make sure all PC/104 expansion cards are removed from the 2133 PC/104 CPU. This ensures that other cards are not interacting with the 2133 PC/104 CPU.
2. The VTC-9F serial cable turns the 2133 PC/104 CPU serial port into a 9-pin AT serial port. Make sure a null modem adapter is installed on the other end, and that the assembly is inserted into the proper serial port on the PC. Make sure the VTC-9F serial cable is connected to J3 of the 2133 PC/104 CPU. Ensure that the SETUP jumper (W2 [5:6]) is removed, so that the 2133 is using the BIOS default settings. (The SETUP default settings specify COM1 as a serial console).
3. Make sure your power module provides +5V (+/-0.25V) and at least 1.5A of current.
4. After verifying the above conditions, you can monitor voltage levels by connecting an oscilloscope between the TxD* line on J3 (pin 5) and ground. After powerup, you should see a burst of activity on the oscilloscope screen. The voltage level should switch between +/-8V. This test verifies that the CPU is active and that the transmit from COM1 is functional.

## Garbled serial console screen activity

If you do get activity on your console screen but the message is garbled, check the following:

1. Remove SETUP W2[5:6] to force 9600, N, 8, 1 for COM1.

2. If you are using PC SmartLINK, make sure you have configured the software for 9600 baud and have selected the correct serial port for communicating with your PC. Refer to the PC SmartLINK manual for information on selecting the baud rate.

3. If you are using communications software other than PC SmartLINK, Octagon cannot guarantee the operation. Make sure that the software parameters are set to match those of the 2133 PC/104 CPU: 9600 baud, 8 bits, 1 stop bit, no parity.

### System generates a BIOS message but locks up when booting from SSD1

The most likely cause is that the DOC does not have boot files. Connect a floppy disk drive, insert a boot disk in the floppy disk drive, and reboot the system. Use the Use the DOS SYS command to install boot files on the DOC.

### System will not recognize hard drive

1. Run SETUP. Change Primary Master Fixed Disk to User and specify Heads, Sectors, and Cylinders.

2. Check hard drive Master/Slave jumpers.

### System locks up after powerdown/powerup

If the power supply does not drain below 0.7V, the CMOS components on the card will act like diodes and forward bias. This is typically caused by using power supplies that have large output capacitors. Either use a different power supply that discharges faster, leave the power off until the supply has adequate time to discharge or place a 100 ohm, large wattage resistor across the output capacitor.

## ≡ Technical assistance

Carefully recheck your system before calling Technical Support. Run as many tests as possible; the more information you can provide, the easier it will be for the Technical Support staff to help you solve the problem. For additional technical assistance, try the following:

Technical Support telephone: 303-426-4521

E-mail Technical Support: Support@octagonsystems.com

Applications Notes (via web): http://www.octagonsystems.com/Contact%20Us/Application%20Notes/application%20notes.html

FAQ (via web):http: //www.octagonsystems.com/Contact%20Us/FAQ/faq.html

# *Overview:* *Section 4 – Appendices*

Section 4 contains a series of appendices which provides additional information about the 2133.

Appendix A: Technical data

Appendix B: Software utilities

Appendix C: Accessories

# *Appendix A:* *Technical data*

## ≡ Technical specifications

**CPU**
133 MHz 586DX

**Local bus clock**
66MHz

**BIOS**
AT compatible with industrial extensions

**DRAM**
4 MB installed on–board
DIMM expandable to 68 MB (64 MB addressable)

**Floppy drive**
Floppy drive support via the LPT1 port or off–card floppy controller

**Hard drive**
Hard drive support via PC/104 hard drive

**SSD0**
On–board 512 KB flash

**SSD1**
Supports a 512K SRAM or an M–Systems DiskOnChip

**ROM–DOS**
None supplied

**Serial I/O**
COM1 and COM2 are 16550 compatible, with 16–byte FIFO

**Parallel port**
LPT1 is PC compatible with multifunctional capability
Supports bi–directional, unidirectional, ECP, and EPP modes

### Battery backup

AT style battery (not included)

### Power requirements

5V ±0.25V
586DX microprocessor

| | |
|---|---|
| Full speed: | 930 mA typical, 1.56A maximum |
| Slow clock: | 650 mA typical |
| Stop clock: | 770 mA typical |
| Suspend: | 406 mA typical |

### Voltage supervisor

Reset threshold: +5V supply > 4.65V typical

### Environmental specifications

–40° to 70° C operating
–55° to 90° C nonoperating
RH 5% to 95%, noncondensing

### Size

3.8" x 3.6"

### Watchdog timer

Default time-out is typically 1.6 seconds (1.0 seconds minimum) software enabled and strobed. Disabled on powerup and reset.

### Bus mastering

Bus mastering is supported

Table A–1 2133 PC/104 CPU memory map

| Hex range | Function |
|---|---|
| 00000h—9FFFFh | System memory (640KB base RAM) |
| A0000h—BFFFFh | Off–card memory (Normally reserved for video memory) |
| C0000h—C7FFFh | Off–card memory. Reserved for video BIOS. Shadow enable/disable* option in SETUP. |
| C8000h—CFFFFh | Off–card memory. Shadow enable/disable* option in SETUP. |
| D0000h—D7FFFh | Off–card memory. Shadow enable/disable* option in SETUP |
| D8000h—DFFFFh | Reserved system area. |
| E0000h—E7FFFh | Off–card memory. |
| E8000h—EFFFFh | 32KB SSD memory paging window. Shadow always disabled. |
| F0000h—FFFFFh | 64KB BIOS area. Shadow always enabled. |
| 100000h—3FFFFFFh | 63MB addressable extended memory |

* = default

Table A–2 2133 PC/104 CPU I/O map

| Hex range | Function |
|---|---|
| 000h—0A7h | System I/O functions |
| 0A8h—0AFh | General purpose status registers (reads jumpers on IOR cycle) |
| 0B0h—0FFh | System I/O functions |
| 100h—207h | Off–card I/O space |
| 208h—20Bh | System control register 0 (R/W) (no SEEP CLK) |
| 20Ch—20Fh | System control register 1 (R/W) (WD IOR strobe) (no SEEP CLK) |
| 210h—213h | System control register 0 (RO) (SEEP CLK) |
| 214h—217h | System control register 1 (RO) (WD IOR strobe) (serial EEPROM R/W) |
| 2F8h—2FFh | COM2 |
| 378h—37Fh | Bi–directional parallel port (LPT1) |
| 300h—3F7h | Off–card I/O space |
| 3F8h—3FFh | COM1 |

Table A–3 I/O map (Configuration inputs via OctaGlue: Address at 0A9h)

| Function | Port 0A9H |
|---|---|
| Not used | Bit 7, I/O read: 0 = on, 1 = off |
| Not used | Bit 6, I/O read: 0 = on, 1 = off |
| Not used | Bit 5, I/O read: 0 = on, 1 = off |
| Not used | Bit 4, I/O read: 0 = on, 1 = off |
| Not used | Bit 3, I/O read: 0 = on, 1 = off |
| Not used | Bit 2, I/O read: 0 = on, 1 = off |
| Not used | Bit 1, I/O read: 0 = on, 1 = off |
| USESETUP | Bit 0, I/O read: 0 = on, 1 = off |

Table A–4 DMA map

| Channel | Function |
|---|---|
| Channel 0 | Reserved for bus memory refresh |
| Channel 1 | Reserved for ECP parallel port |
| Channel 2 | Reserved for floppy drive |
| Channel 3 | Available |
| Channel 4 | Slave |
| Channel 5 | Available |
| Channel 6 | Available (16 bit) |
| Channel 7 | Available (16 bit) |

Table A–5 Interrupt map

| Interrupt | Function |
|---|---|
| IRQ0 | System timer |
| IRQ1 | Keyboard |
| IRQ2 | Cascade |
| IRQ3 | COM2 |
| IRQ4 | COM1 |
| IRQ5 | Parallel port secondary IRQ |
| IRQ6 | Free (normally reserved for the floppy drive) |
| IRQ7 | Parallel port primary IRQ |
| IRQ8 | RTC alarm |
| IRQ9 | Free |
| IRQ10 | Free |
| IRQ11 | Free |
| IRQ12 | Free (Normally reserved for the mouse port) |
| IRQ13 | Floating point unit |
| IRQ14 | Free (Normally rerserved for hard disk drive) |
| IRQ15 | Free |

Table A–6 COM1 available addresses

| COM1 available addresses | |
|---|---|
| 220h | 2F8h |
| 228h | 338h |
| 238h | 3E8h |
| 2E0h | 3F8h* |
| 2E8h | |

* = default

Table A–7 COM2 available addresses

| COM2 available addresses | |
|---|---|
| 220h | 2F8h* |
| 228h | 338h |
| 238h | 3E8h |
| 2E0h | 3F8h |
| 2E8h | |

* = default

Table A–8 Available LPT1 port addresses

| LPT1 port addresses |
|---|
| 278h |
| 378h* |
| 3BCh |

* = default

# ≡ Connector pinouts

Table A–16 PC/104 signal assignments

| Pin | Row A | Row B | Row C | Row D |
|---|---|---|---|---|
| 0 | — | — | Gnd | Gnd |
| 1 | IOCHK* | Gnd | SBHE* | MEMCS16* |
| 2 | SD7 | RESETDRV | LA23 | IOCS16* |
| 3 | SD6 | +5V | LA22 | IRQ10 |
| 4 | SD5 | IRQ9 | LA21 | IRQ11 |
| 5 | SD4 | –5V | LA20 | IRQ12 |
| 6 | SD3 | DRQ2 | LA19 | IRQ15 |
| 7 | SD2 | –12V | LA18 | IRQ14 |
| 8 | SD1 | 0 WS** | LA17 | DACK0* |
| 9 | SD0 | +12VDC | MEMR* | DRQ0 |
| 10 | IOCHRDY | Key | MEMW* | DACK5* |
| 11 | AEN | SMEMW* | SD8 | DRQ5 |
| 12 | SA19 | SMEMR* | SD9 | DACK6* |
| 13 | SA18 | IOW* | SD10 | DRQ6 |
| 14 | SA17 | IOR* | SD11 | DACK7* |
| 15 | SA16 | DACK3* | SD12 | DRQ7 |
| 16 | SA15 | DRQ3 | SD13 | +5V |
| 17 | SA14 | DACK1* | SD14 | Master* |
| 18 | SA13 | DRQ1 | SD15 | Gnd |
| 19 | SA12 | Refresh* | Key | Gnd |
| 20 | SA11 | SYSCLK | — | — |
| 21 | SA10 | IRQ7 | — | — |
| 22 | SA9 | IRQ6 | — | — |
| 23 | SA8 | IRQ5 | — | — |
| 24 | SA7 | IRQ4 | — | — |
| 25 | SA6 | IRQ3 | — | — |
| 26 | SA5 | DACK2* | — | — |
| 27 | SA4 | TC | — | — |
| 28 | SA3 | Bale | — | — |
| 29 | SA2 | +5V | — | — |
| 30 | SA1 | 14 MHz | — | — |
| 31 | SA0 | Gnd | — | — |
| 32 | Gnd | Gnd | — | — |

* = active low; ** = wait state

Table A–17 Speaker, battery, keyboard, and reset: J5

| Pin | Signal | Function |
|---|---|---|
| 1 | +Speaker | +5V in series with 33 Ω |
| 2 | GND | 3.6V external battery, negative |
| 3 | Reset SW | External reset |
| 4 | KBD SW | Inhibit switch to disable keyboard |
| 5 | KBD Data | Keyboard data |
| 6 | KBD CLK | Keyboard clock |
| 7 | Ground | Signal and power ground |
| 8 | KBD PWR | +5V for keyboard |
| 9 | BATV+ | 3.6V external battery, positive |
| 10 | PWR good | Power good input |

Table A–19 Power: J7

| Pin | Function |
|---|---|
| 1 | Gnd |
| 2 | +5 VDC |
| 3 | NC |
| 4 | +12 VDC to PC/104 |
| 5 | –5 VDC to PC/104 |
| 6 | –12 VDC to PC/104 |
| 7 | Gnd |
| 8 | +5 VDC |

Table A–31 LPT1 printer connector: J4

| Pin | DB-25 pin | Function | Pin | DB-25 pin | Function |
|---|---|---|---|---|---|
| 1 | 1 | STB* | 14 | 20 | Gnd |
| 2 | 14 | AFD* | 15 | 8 | DATA6 |
| 3 | 2 | DATA0 | 16 | 21 | Gnd |
| 4 | 15 | ERR* | 17 | 9 | DATA7 |
| 5 | 3 | DATA1 | 18 | 22 | Gnd |
| 6 | 16 | INIT* | 19 | 10 | ACK* |
| 7 | 4 | DATA2 | 20 | 23 | Gnd |
| 8 | 17 | SLIN* | 21 | 11 | BUSY |
| 9 | 5 | DATA3 | 22 | 24 | Gnd |
| 10 | 18 | Gnd | 23 | 12 | PE |
| 11 | 6 | DATA4 | 24 | 25 | Gnd |
| 12 | 19 | Gnd | 25 | 13 | SLCT |
| 13 | 7 | DATA5 | 26 | NC | +5V Safe |

* = active low

# *Appendix B:* Software utilities

## ≡ Introduction

The 2133 PC/104 CPU Software Utility Disk comes with the utilities listed below. This appendix describes the utilities and their use.

### Support commands:

- COM1CON.EXE
- I17HNDLR.EXE
- LPT1CON.COM
- PGMBIOS.EXE
- PMISETUP.EXE
- RESET.COM
- SETUP.COM
- SCONSOLE.COM

### Display commands:

In addition to the system support utilities, there are utilities to handle displays and keypads. These are located in the Display directory. Refer to Display.txt for information on these utilities.

- DISPLAY.EXE
- KPTEST.EXE
- DSPTEST.EXE
- DSQBTEST.EXE
- KPQBTEST.EXE
- KPOFF.COM
- KPON.COM

# ≡ COM1CON.EXE

## Purpose

This support command enables COM1 as the console device when the system uses a video card.

## Syntax

**COM1CON.EXE**

## Parameter

/Ux specifies to revert to the video card.

## Remarks

The memory for COM1CON is not released using the /U parameter. Only the interrupt vectors are restored to the previous state.

# ≡ I17HNDLR.EXE

## Purpose

This support command allows the system to use the INT 17 functions.

## Syntax

**I17HNDLR**

## Remarks

The I17HNDLR allows the system to use the INT 17 functions. I17HNDLR is a TSR program. If you are using it, add the command to your AUTOEXEC.BAT file.

# ≡ LPT1CON.COM

## Purpose

This support command redirects the video to the LPT1 port.

## Syntax

**LPT1CON**

## Remarks

If you have a 2010 interface board and an LCD display connected to the LPT1 port, executing the DISPLAY.EXE and LPT1CON.COM programs allow you to use the display as the system console. You must reset your system to change the video to the original parameters.

# ≡ PGMBIOS.EXE

## Purpose

This support command programs a new system BIOS into the 2133 PC/104 CPU.

## Syntax

**PGMBIOS** [*filename*] [**/Y**] [**/C**] [**/?**]

## Parameters

- filename specifies the BIOS .DAT file to program into flash.
- /Y specifies to perform the programming without first prompting "Are you sure?".
- /C specifies PGMBIOS is to allow programming even when the checksum is bad.
- /? requests a help menu.

## Example

To program the BIOSFILE.BIN files into the SSD0 BIOS area, enter:

```
PGMBIOS 4985597.104 /Y
```

# ≡ PMISETUP.EXE

## Purpose

This support command allows modification of the power management options.

## Syntax

**PMISETUP** *inputfile* [**/SHOWALL**] [**/DEFAULT**] [**/P**] [**/?**]

## Parameters

- inputfile specifies the file containing PMISETUP commands.
- /SHOWALL specifies to show all of the power management options on screen.
- /DEFAULT specifies the BIOS defaults that are to be loaded first before the inputfile is used.
- /P specifies to enable pausing between screens for viewing when using the /SHOWALL option.
- /? displays a short help screen for the PMISETUP program. No other arguments are to be included on the command line when the /? is used.

## See also

The PMISETUP section in the SETUP programs chapter.

# ≡ RESET.COM

## Purpose

This support command enables the watchdog timer and allows time–out to expire, thus restarting the system.

## Syntax

**RESET**

## Remarks

The RESET command also restarts all the expansion I/O cards on the bus. This differs from a <CTRL><ALT><DEL> reboot of the system which only restarts the system but not the expansion cards.

# ≡ SETUP.COM

## Purpose

This support command configures various system parameters, including serial ports, a parallel port, and a floppy and hard drive.

## Syntax

**SETUP** [**/D**]

## Parameter

- /D returns all setup values to default values.

## Remarks

From the directory where this utility file is located, enter:

**SETUP**

After the copyright message displays, the main menu appears:

```
                    OCTAGON SYSTEMS CORPORATION
                      PC680 SETUP UTILITY Vx.x
                (c) Phoenix Technologies, Ltd. 1985, 1995
 ____________________________________________________________
 (Press SPACE to CHANGE, ENTER to ACCEPT, ESC to EXIT)

Serial Console on COM1:          ENABLED
 COM1 Console Baud Rate:         9600
Power on memory test:            ENABLED
Boot Sequence:                   C: ONLY
Serial Port COM1:                ENABLED
 Serial Port COM1 Address:       3F8h (default)
COM2:                            ENABLED
 Serial Port COM2 Address:       2F8h
Parallel (LPT) Port:             ENABLED
 Parallel Port Mode:             Bidirectional Printer
                                 Port
 Parallel Port Address:          378h
Number of floppy drives:         1
Floppy drive 1 size:             3.5", 1.44 MB
Number of hard drives:           0
Internal CPU cache:              ENABLED
Limit CPU to half speed          NO
SETUP Entry via Hotkey:          ENABLED
Power Management:                ENABLED
 Doze Clock (slow, stop):        SLOW
 Time Update after Suspend:      ENABLED
DIMM Module Type:                EDO
Enable ROM at C0000h–C7FFFh:     ENABLED
Enable ROM at C8000h–CFFFFh:     ENABLED
Enable ROM at D8000h–DFFFFh:     ENABLED
Shadow C0000h–C7FFFh:            ENABLED
Shadow C8000h–CFFFFh:            ENABLED
Shadow D0000h–D7FFFh:            DISABLED
Shadow D8000h–DFFFFh:            DISABLED

Press ENTER to SAVE the changes or
Press ESC to EXIT without saving the changes.

Options Saved.

You must reset for these options to take effect.
PC680 C:\>
```

Note Executing SETUP /D will change all setup parameters to default values.

## See also

See the SETUP programs chapter for more information. You may also enter SETUP at post time by entering the “backspace” and “s” keys.

# ≡ SCONSOLE.EXE

## Purpose

This support command checks whether the system is running on a serial console.

## Syntax

**SCONSOLE**

## Remarks

This command is useful in batch programs to detect if the serial console is in use. It returns an error level of 0 if the serial console is enabled, allowing a DOS batch file to "react" to the serial console being enabled.

# *Appendix C:* Accessories

Table C–1 Cables and terminal board

| Product | Description | Octagon part number |
|---|---|---|
| VTC-9F | Serial cable - Female | 2746 |
| VTC-9M | Serial cable - Male | 2472 |
| Null modem adapter | 9-pin to 9-pin | 2470 |
| FCA-12 | LPT1 to floppy cable | 4809 |
| VTC-5/IBM | LPT1 to DB25 printer cable | 1237 |
| PCA-36 | LPT1 to Centronics printer cable | 4808 |
| CMA-26-12 | 12" cable for LPT1 port | 2776 |
| CMA-26-24 | 24" cable for LPT1 port | 1257 |
| STB-26 | Terminal board, 26-position | 2905 |

Table C–2 Memory devices

| Product | Description | Octagon part number |
|---|---|---|
| 4MB DRAM | EDO DIMM memory module | 4583 |
| 8MB DRAM | EDO DIMM memory module | 4584 |
| 16MB DRAM | EDO DIMM memory module | 4582 |
| 32MB DRAM | EDO DIMM memory module | 4990 |
| 64MB DRAM | EDO DIMM memory module | 5364 |
| 512K SRAM | 512K SRAM, 32 pin | 2915 |

Table C–3 LCD displays and keypads

| Product | Description | Octagon part number |
|---|---|---|
| LCD–4 x 20 | LCD display w/cable, 40 character | 2783 |
| LCD–4 x 40 | LCD display w/cable, 80 character | 2784 |
| 2010 | LCD display/keypad interface | 3909 |
| KP-1 | Keypad w/cable, 16-key, low cost | 1218 |
| KP-2-16 | Keypad w/cable, 16-key, relegendable | 1736 |
| KP-3 | Keypad w/cable, 16-key, NEMA 2 rated | 1737 |

Table C–4 Opto rack and modules

| Product | Description | Octagon part number |
|---|---|---|
| MPB-16PC | PC opto rack, 16-position | 3389 |
| G4-IAC5 | AC input, 90-140 VAC | 2395 |
| G4-IAC5A | AC input, 180-280 VAC | 2396 |
| G4-IDC5 | DC input, 15-32 VDC | 2397 |
| G4-IDC5B | DC input, 4-16 VDC | 2511 |
| G4-IDC5D | DC input, 2.5-28 VDC | 2529 |
| G4-OAC5 | AC output, 12-140 VAC | 2398 |
| G4-OAC5A | AC output, 12-280 VAC | 2399 |
| G4-ODC5 | DC output, 5-60 VDC | 2400 |
| G4-ODC5A | DC output, 5-200 VDC | 2503 |
| G4-ODC5R | DC output, dry contact output | 3013 |

Table C–5 Miscellaneous part numbers

| Product | Description | Octagon part number |
|---|---|---|
| AT battery | Calendar/clock battery backup | 3186 |
| PC SmartLINK IV | Terminal emulation software | 3447 |
| CAMBASIC | Multitasking, industrial control programming language | 4059 |

# Index

## R



## S



## T



## W

# *Warranty*

Octagon Systems Corporation (Octagon), warrants that its standard hardware products will be free from defects in materials and workmanship under normal use and service for the current established warranty period. Octagon's obligation under this warranty shall not arise until Buyer returns the defective product, freight prepaid to Octagon's facility or another specified location. Octagon's only responsibility under this warranty is, at its option, to replace or repair, free of charge, any defective component part of such products.

## Limitations on warranty

The warranty set forth above does not extend to and shall not apply to:

1. Products, including software, which have been repaired or altered by other than Octagon personnel, unless Buyer has properly altered or repaired the products in accordance with procedures previously approved in writing by Octagon.
2. Products which have been subject to power supply reversal, misuse, neglect, accident, or improper installation.
3. The design, capability, capacity, or suitability for use of the Software. Software is licensed on an "AS IS" basis without warranty.

The warranty and remedies set forth above are in lieu of all other warranties expressed or implied, oral or written, either in fact or by operation of law, statutory or otherwise, including warranties of merchantability and fitness for a particular purpose, which Octagon specifically disclaims. Octagon neither assumes nor authorizes any other liability in connection with the sale, installation or use of its products. Octagon shall have no liability for incidental or consequential damages of any kind arising out of the sale, delay in delivery, installation, or use of its products.

## Service policy

1. If a product should fail during the warranty period, it will be repaired free of charge. For out of warranty repairs, the customer will be invoiced for repair charges at current standard labor and materials rates.
2. Customers that return products for repairs, within the warranty period, and the product is found to be free of defect, may be liable for the minimum current repair charge.

## Returning a product for repair

Upon determining that repair services are required, the customer must:

1. Obtain an RMA (Return Material Authorization) number from the Customer Service Department, 303-430–1500.
2. If the request is for an out of warranty repair, a purchase order number or other acceptable information must be supplied by the customer.
3. Include a list of problems encountered along with your name, address, telephone, and RMA number.
4. Carefully package the product in an antistatic bag. (Failure to package in antistatic material will VOID all warranties.) Then package in a safe container for shipping.
5. Write RMA number on the outside of the box.
6. For products under warranty, the customer pays for shipping to Octagon. Octagon pays for shipping back to customer.
7. Other conditions and limitations may apply to international shipments.

Note PRODUCTS RETURNED TO OCTAGON FREIGHT COLLECT OR WITHOUT AN RMA NUMBER CANNOT BE ACCEPTED AND WILL BE RETURNED FREIGHT COLLECT.

## Returns

There will be a 15% restocking charge on returned product that is unopened and unused, if Octagon accepts such a return. Returns will not be accepted 30 days after purchase. Opened and/or used products, non-standard products, software and printed materials are not returnable without prior written agreement.

## Governing law

This agreement is made in, governed by and shall be construed in accordance with the laws of the State of Colorado.

The information in this manual is provided for reference only. Octagon does not assume any liability arising out of the application or use of the information or products described in this manual. This manual may contain or reference information and products protected by copyrights or patents. No license is conveyed under the rights of Octagon or others.