

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

Embedded PCs For Extreme Environments

*6225 CPU Card
User's Manual
5612 (0403)*

Copyright

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

Disclaimer

Copyright 2000, 2003—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.



6510 W. 91st Ave.
Westminster, CO 80031

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 but can apply any IC on the card.
- **Under rated power supply:** The board may fail to boot due to an under rated power supply. It is important that a quality power supply be used with the 6225 CPU Card that has sufficient current capacity, line and load regulation, hold up time, current limiting, and minimum ripple. It is extremely import to select a supply that ramps up in 10ms or less. This assures that all the circuitry on the CPU Cards sequences properly and avoids system lockup.
- **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 switch 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 μ F, across the switch contact. This will also eliminate radio frequency and other high frequency pickup.

Avoiding damage to the heatsink or 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 or CPU as well.

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

Table of Contents

Copyright.....	2
Disclaimer	2
Using CMOS circuitry in industrial control.....	3
Avoiding damage to the heatsink or CPU	5
Table of Contents	6
List of Figures.....	12
List of Tables.....	13
Overview: Section 1 – Installation	14
Chapter 1: Overview.....	15
Description	15
6225 CPU Card major hardware features.....	15
CPU	15
Up to 8 MB DRAM.....	15
Solid-state disks.....	15
SSD0	16
SSD1	16
Serial ports protected against ESD	16
Hard disk port.....	16
Industrial digital I/O	16
Keyboard and mouse ports.....	17
PC/104 16-bit interface	17
Ethernet	17
Multifunctional printer port.....	17
Watchdog timer added for safety	18
Real time calendar/clock with battery-backup.....	18
Hardware reset	18
5 Volt operation lowers system cost.....	18
6225 CPU Card major software features	18
Advanced power management and system management input.....	19
Diagnostic software verifies system integrity automatically	19
SETUP information stored in EEPROM for high reliability.....	19
Phoenix BIOS.....	19
“Instant DOS” operating system.....	20
On-board flash file system.....	20
Octagon BIOS extensions.....	20
Boot sequence.....	20
Rugged environmental operation.....	21
Size	21
Chapter 2: Quick start.....	22
Hardware installation	22
Using a Micro PC card cage	25
Panel mounting or stacking the 6225 CPU Card.....	28
Using the 6225 CPU Card in a passive ISA backplane	30
Establishing communications with the 6225 CPU Card.....	32

Transferring files between the 6225 CPU Card and your PC	34
Downloading files to the 6225 CPU Card using TRANSFER.EXE.....	35
Downloading files to the 6225 CPU Card using REMDISK/REMSERV	37
6225 CPU Card with no video card and one serial cable.....	38
6225 CPU Card with no video card, two PCs, and two serial cables	39
6225 CPU Card with a video card and one serial cable.....	41
Chapter 3: Setup programs.....	44
SETUP	44
Running SETUP	46
SETUP example.....	49
SETSSD.....	51
PMISETUP	52
Chapter 4: Save and run programs.....	53
Save and run your programs on the 6225 CPU Card	53
Saving programs and support files	53
Adding your application	54
Autoexecuting your application	54
Overriding the autoexecution of your application	55
Option 1.....	55
Option 2.....	55
Option 3.....	55
Option 4.....	55
Overview: Section 2 – Hardware	56
Chapter 5: Serial ports.....	57
Description	57
Serial port configurations.....	58
COM1/COM2.....	59
COM3/COM4.....	61
Function and use of serial ports	62
COM1 as serial console device	62
Mating receptacle.....	62
COM1 through COM4 as RS–232 I/O.....	62
COM2 and COM4 as RS–485 network	63
COM2 as RS–422 network	65
Optomux mode	66
RS–422/485 support functions	68
Function 00 – Initialization.....	68
Function 01 – Send message	69
Function 02 – Receive message.....	69
Function 05 – Get receiver status.....	69
Function 06 – Get transmitter status.....	69
Function 07 – Get ID	69
Function 08 – Set ID.....	69
Function 09 – Get incoming message buffer pointer	70
Function 0a – Get outgoing message buffer pointer	70
Function 0b – Set incoming message buffer pointer.....	70
Function 0c – Set outgoing message buffer pointer	70
Function 0d – Set roll call response.....	70

Function 0e – Set state wish response	70
Function 0f – Set internal state response	71
Octagon’s command set	72
Checksum field.....	72
Examples	73
Example 1:.....	73
Example 2:.....	74
Example 3:.....	75
Chapter 6: LPT1 parallel port, LCD and Keypad	77
LPT1 parallel port	77
Printer	77
Installing a printer	77
Display	78
Installing a display.....	78
Keypad.....	79
Installing a keypad	79
LPT1 as a floppy disk drive.....	79
Chapter 7: Console devices	81
Description	81
Selecting console devices	81
Serial console	82
Keyboard	83
Mouse	83
Chapter 8: SSDs, DRAM, and battery backup	84
Description	84
SSD0 (optional).....	84
Selecting SSD0 device type	84
Selecting an SSD BIOS	86
SSD1.....	86
Battery backup for SSD0 SRAM and real time calendar clock.....	87
Installing an AT battery.....	87
Managing SSDs	87
FlashFx flash file system	88
Defining SSDs using SETSSD	88
Using FXFMT to format an SSD	90
Formatting SSD1.....	90
Using SYS to make an SSD bootable	91
Adding operating system startup files (using SYS).....	91
Changing boot SSD.....	91
Autoexecuting your applicatio	91
Using SETSSD and TESTOEM to test an SSD	92
Making copies of the 6225 SSD.....	93
Programming copies of the 6225 SSD	93
DRAM.....	94
Chapter 9: External drives.....	95
Description	95
Floppy disk controller.....	95
Power requirements	95

Installing a floppy disk drive	96
Hard disk controller	96
Bootling with ROM-DOS	97
Chapter 10: Digital I/O	98
Description	98
Chapter 11: Ethernet.....	102
Description	102
Chapter 12: PC/104 expansion.....	103
Description	103
Overview: Section 3 – System management.....	104
Chapter 13: Watchdog timer and hardware reset	105
Description	105
Watchdog function definitions using enhanced INT 17h handler	105
Enable watchdog.....	105
Strobe watchdog.....	106
Disable watchdog.....	106
Hardware reset.....	107
Chapter 14: Serial EEPROM.....	108
Description	108
Enhanced INT 17h function definitions	108
Serial EEPROM.....	109
Read a single word from the serial EEPROM.....	109
Write a single word to the serial EEPROM	109
Read multiple words from the serial EEPROM.....	110
Write multiple words to the serial EEPROM.....	111
Return serial EEPROM size	111
Chapter 15: CAMBASIC	113
Introduction	113
Major features.....	113
Event Multitasking™	113
CAMBASIC task types	114
Easy-to-write programs	114
Keyboard mode to debug hardware	114
Built-in help and error messages.....	115
Industrial commands.....	115
Downloading programs remotely.....	115
Network support.....	115
Compatibility with other programs	115
Industrial extensions of CAMBASIC.....	116
Easy-to-use programming examples	117
Example 1.....	117
Example 2.....	117
Chapter 16: CPU power management	119
Description	119
Power management overview	119
Hardware controlled modes	120
Device power management	120
System power management	121

Initiating the SUSPEND/RESUME option with system activities.....	122
Power management configuration	123
Enabling power management	123
System timers	124
IDLE timer resets	124
Suspended devices in SUSPEND mode.....	126
Resume conditions	126
Chapter 17: Troubleshooting.....	128
No screen activity—checking console serial communications	128
Garbled console screen activity.....	128
System generates a BIOS message but locks up when booting from SSD1	129
FlashFx reports a drive, but issuing a DIR generates an error message	129
FlashFx does not report the drive.....	129
System locks up on powerup; may or may not respond to reset switch.....	130
System locks up after powerdown/powerup	130
LED signaling of “beep” codes.....	130
Description	130
Technical assistance	134
Overview: Section 4 – Appendices.....	135
Appendix A: 6225 technical data	136
Technical specifications.....	136
CPU	136
Bus clock.....	136
BIOS	136
DRAM.....	136
Floppy drive	136
Hard drive	136
Solid–state disk 0.....	136
Solid–state disk 1.....	136
ROM–DOS.....	136
Serial I/O	136
Digital I/O.....	137
Parallel port	137
Watchdog timer.....	137
Bus mastering.....	137
Power requirements.....	137
Environmental specifications.....	137
Size	137
Mating connectors.....	137
Maps	138
Jumper settings	140
Connector pinouts.....	141
Appendix B: Software utilities.....	145
Introduction	145
Support commands	145
Support device drivers.....	145
COM1CON.EXE	146
FXCHK.EXE	146

FXFMT	147
FXINFO.EXE	148
FXRDIMG.EXE	148
FXREMNT.EXE.....	149
FXWRIMG.EXE.....	149
GETBIOS.EXE.....	150
GETIMG.EXE	150
GETIMG compatibility:.....	152
HIMEM.SYS	152
I17HNDLR.EXE	153
LPT1CON.COM.....	153
FXFMT.EXE	154
Example 1.....	155
PGMBIOS.EXE.....	155
Example 1.....	155
PGMIMG.EXE	156
FXDOS.SYS	157
PMISETUP.EXE.....	157
REMDISK.EXE.....	158
REMQUIT.COM	160
REMSERV.EXE.....	161
RESET.COM.....	162
SCONSOLE.EXE.....	163
SETSSD.EXE	163
SETUP.COM.....	165
TESTOEM.EXE	166
TRANSFER.EXE	167
VDISK.SYS	169
Appendix C: Third party support.....	172
Using M–Systems DiskOnChip (DOC).....	172
Booting from the DOC drive	173
Appendix D: Accessories	174
Warranty	175
Limitations on warranty	175
Service policy	176
Returning a product for repair.....	176
Returns.....	177
Governing law	177

List of Figures

Figure 2-1	6225 CPU Card component diagram	23
Figure 2-2	6225 CPU Card center-to-center hole dimensions	24
Figure 2-3	Edge connector orientation.....	26
Figure 2-4	Populated Micro PC card cage.....	27
Figure 2-5	VTC-20F cable	27
Figure 2-6	Serial communications setup	28
Figure 2-7	Panel mounting the 6225 CPU Card	29
Figure 2-8	Stacking the 6225 CPU Card	29
Figure 2-9	Using a passive ISA backplane	31
Figure 2-10	Serial console connections	36
Figure 2-11	Downloading files to the 6225 CPU Card with no video card using REMDISK/REMSERV.....	38
Figure 2-12	Downloading files to the 6225 CPU Card with no video card and two PCs	40
Figure 2-11	Downloading files to the 6225 CPU Card with a video card.....	42
Figure 5-1	Null modem adapter and VTC-20F/VTC-20M cable	62
Figure 5-2	6225 serial devices	63
Figure 5-3	RS-485 network	64
Figure 5-4	RS-422 network	66
Figure 6-1	LPT1 as a printer port	78
Figure 6-2	LPT1 as a display or keypad port.....	79
Figure 7-1	The 6225 and a serial console.....	82
Figure 7-2	VTC-20F cable and null modem adapter	83
Figure 8-1	SSD socket locations	85
Figure 8-2	M-Systems pin alignment with the 6225 SSD0 socket	86
Figure 10-1	STB-26 Terminal Board.....	101
Figure 13-1	Typical PC/104 module stack	103
Figure 21-1	Cabling diagram for a standard cable.....	159

List of Tables

Table 3–1	6225 setup parameters and defaults.....	45
Table 5–1	COM port configurations	58
Table 5–2	COM port jumper settings.....	59
Table 5–3	COM1/2 and COM3/4 pinout for RS–232 (J6 and J7 connectors)	60
Table 5–4	COM2 (RS–422/485) and COM4 (RS–485) pinout (J3 and J8 connectors)	61
Table 5–5	COM2/4 mode select.....	67
Table 5–6	Definitions list for Octagon RS–485 network commands	72
Table 6–1	LPT1/floppy drive cable	80
Table 8–1	SSD0 device selection jumpers.....	84
Table 8–2	Selecting an SSD BIOS: W6[7–8].....	86
Table 8–3	Battery connector: J10.....	87
Table 10–1	Digital I/O programming examples.....	99
Table 10–2	Digital I/O pinout.....	101
Table 11–1	Ethernet LEDs	102
Table 17–1	Phoenix BIOS beep odes	132
Table 17–1	Phoenix BIOS beep (cont.).....	133
Table 17–2	Additional error codes for Phoenix BIOS.....	134
Table A–1	6225 DMA map.....	138
Table A–2	6225 I/O map	138
Table A–3	6225 interrupt map.....	139
Table A–4	6225 memory map.....	139
Table A–5	6225 jumper settings: W6 System Configuration.....	140
Table A–6	6225 jumper settings: W1, W4, W5, W6 COM port jumper settings.	140
Table A–7	6225 jumper settings: W2, W3 SSD0 device selection jumpers.....	141
Table A–8	6225 jumper settings: Selecting an SSD BIOS: W6[7–8].....	141
Table A–9	COM1/2 and COM3/4 pinout for RS–232 (J6 and J7 connectors)	141
Table A–10	COM2 (RS–422/485) and COM4 (RS–485) pinout (J3 and J8 connectors)	142
Table A–11	6225 keyboard connector pinout: J2.....	142
Table A–12	6225 PC/104 connector pinout: J5.....	143
Table A–13	6225 power connector pinout: J11.....	144
Table A–14	Battery connector: J10.....	144
Table A–15	6225 hard drive connector pinout: J7	144
Table D–1	Cables and terminal board	174
Table D–2	LCD displays and keypads	174
Table D–3	Miscellaneous part numbers	174

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 6225 CPU Card is intended for medium–performance embedded control applications. The 6225 CPU Card integrates serial communication, industrial digital I/O, floppy port, IDE hard disk port, a multifunctional parallel port, a 10BaseT Ethernet port, and a keyboard port. This single–board computer has one solid–state disk and a socket for a second.

The 6225 CPU Card can be used in a stand–alone mode, installed in a passive ISA Bus backplane, installed in a card cage, or connected through a PC/104 interface.

The 6225 CPU Card has a low–power 386SX 40 MHz CPU. This board is supplied with ROMDOS v7.1 in ROM. Since the 6225 CPU card 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.

6225 CPU Card major hardware features

CPU

The CPU is a high–performance, low–power 386SX with a clock speed of 40 MHz. The 6225 CPU Card has an ISA bus speed of 7.2 MHz.

Up to 8 MB DRAM

The 6225 CPU Card comes with 4 MB DRAM soldered on board. In OEM quantities it can also be ordered in 8 MB configurations.

Solid–state disks

Flash file system software is included which allows the SSDs to emulate hard disk operation.

SSD0

SSD0 is a 32-pin DIP socket. It accepts either an M-Systems DOC, 5V flash, 512 KB SRAM or EPROM. The socket exhibits high retention force and affords a gas tight contact. The 6225 can be configured to boot from a device in SSD0.

SSD1

SSD1 is a 2 MB SMT boot flash which contains the both a 128K BIOS drive and a DOS-compatible drive. SSD1 contains ROM-DOS 7.1 and a suite of embedded software.

Serial ports protected against ESD

The 6225 CPU Card has four serial ports with combinations of RS-232C, RS-422 and RS-485 interfaces. These serial ports have the following common specifications:

- IEC1000, level 3, ESD protection specification
 - Contact discharge ± 4 kV
 - Air-gap discharge ± 8 kV
- Backdrive protection
- 16550 compatible
- Up to 115.2K baud
- 16-byte FIFO buffers
- Jumper-selectable terminations for RS-422 and RS-485
- Enabled and disabled in SETUP

Hard disk port

The IDE hard drive port is terminated with a 44-pin, 2 mm connector. The IDE connector supplies +5V to the hard drive. The AT BIOS supports up to two IDE drives.

Industrial digital I/O

The 6225 CPU Card has 17 lines of industrial digital I/O. The lines are 5V compatible and can sink and source 15 mA.

Keyboard and mouse ports

The keyboard controller accepts an AT style keyboard; the connector is on an 8-pin header. The 6225 CPU Card does not have a mouse port; if a mouse is required for your application, one of the COM ports can be used for a serial mouse. Neither the keyboard nor the mouse is 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. Up to two PC/104 expansion boards may be stacked on the 6225 CPU Card.

Ethernet

The 6225 CPU Card provides a 10BaseT Ethernet port and supports the IEEE 802.3 Ethernet standard. The Ethernet controller IC chip provides the following:

- ISA interface
- Integrated 10BaseT transceiver interface
- Two LEDs for link and traffic status
- NE2000 compatible

Multifunctional printer port

The 6225 CPU Card incorporates the latest enhanced parallel port. It includes the following features:

- Unidirectional
- Bidirectional
- IEEE 1284, ECP and EPP modes
- 12 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

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 programmable from 0.5 seconds to 64 seconds.

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. A connector is provided for the external battery.

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
- Depressing the reset switch
- Cycling power
- Power supervisor reset

5 Volt operation lowers system cost

The 6225 CPU Card operates from a single 5V $\pm 5\%$ supply.

- 5V $\pm 5\%$, approximately 650 mA (dependent upon speed and I/O devices)
- +12V and -12V supplied to PC/104 connector; not required for 6225 CPU Card operation

6225 CPU Card 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 6225 CPU Card 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:

- Wake up through various interrupts including keyboard, SMI and serial port
- Power down of peripherals such as hard disk drive.

Diagnostic software verifies system integrity automatically

The 6225 CPU Card 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 flashing LED. 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 “*Troubleshooting*” 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 6225 CPU Card stores the SETUP information in EEPROM with 512 bytes available to the user. Software routines to use this available memory come with the 6225 CPU Card.

Phoenix BIOS

The 6225 CPU Card has a Phoenix AT BIOS with power management and Octagon BIOS extensions. The AT BIOS supports up to two IDE

drives.

“Instant DOS” operating system

Datalight ROM-DOS v7.1 operating system is in flash. This means that this version is always present on powerup. The system boots and operates the same way as a desktop PC. Since all software and hardware are included, the system is fully operational “out of the box.”

On-board flash file system

The flash file system controls the on-board SSDs, which allows read/write DOS access to the flash/SRAM devices. For certain types of flash, the flash file system uses “wear leveling” to spread the usage and maximize the lifetime of these devices.

Octagon BIOS extensions

On-board BIOS extensions allow easy access to digital I/O, power management, watchdog timer functions, etc.

Boot sequence

A 6225 CPU Card can be configured to boot from the on-card, solid-state disk in SSD1, a device in SSD0, an external floppy, or a hard disk.

Rugged environmental operation

- Operating temperature – 40° to 85°C, operating with adequate airflow
- Nonoperating temperature –55° to 95°C, nonoperating
- Relative humidity 5% to 95% noncondensing
- Altitude –100 to 10,000 m
- Shock 40g, 3 axis
- Vibration 5g, 3 axis

Size

4.5" x 4.9" x 0.8"

Chapter 2: **Quick start**

This chapter covers the basics of setting up a 6225 CPU Card system. Refer to the 6225 CPU Card component diagram (Fig. 2-1) for the location of the various connectors. The following topics are discussed:

- Panel mounting, stacking, or installing the 6225 CPU Card into an Octagon card cage
- Setting up a serial communications console I/O link between the 6225 CPU Card and your desktop PC
- Downloading files to the 6225 CPU Card and running a program from the virtual drive.

WARNING!

The 6225 CPU Card may not be installed in a PC. These cards are designed to be independent CPU cards only, not accelerators or coprocessors.

Hardware installation

WARNING!

The 6225 CPU Card contains static-sensitive CMOS components. The card is most susceptible to damage when it is plugged into a card cage. The 6225 CPU Card becomes charged by the user, and the static discharges to the backplane from the pin closest to the card connector. If that pin happens to be an input pin, even TTL inputs may be damaged. To avoid damaging your card and its components:

- **Ground yourself before handling the card**
- **Disconnect power before removing or inserting the card.**

WARNING!

Take care to correctly position the 6225 CPU Card in the card cage. The VCC and ground signals must match those on the backplane. Figure 2-3 shows the relative positions of the 6225 CPU Card as it is installed in the card cage.

Figure 2-1 6225 CPU Card component diagram

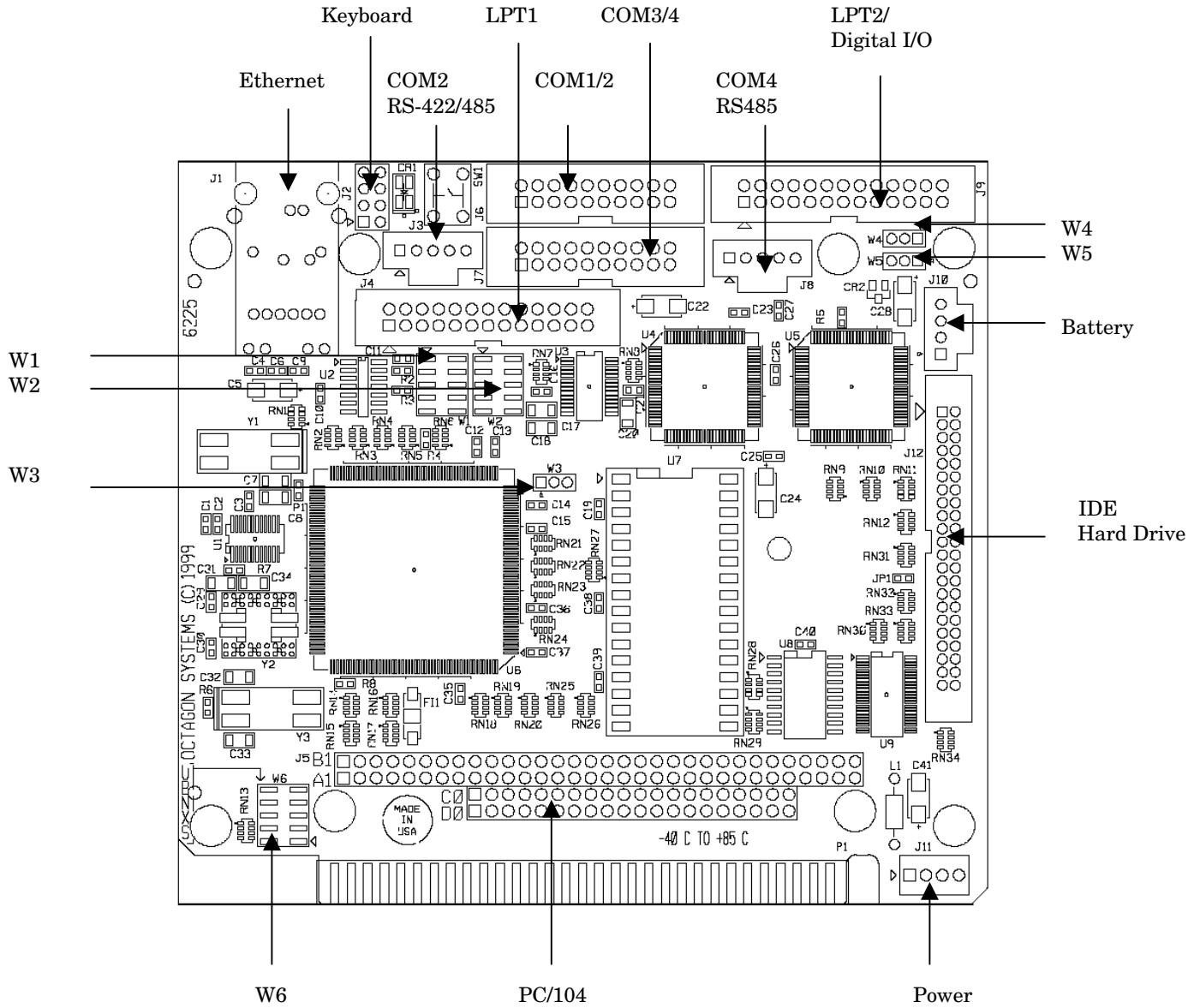
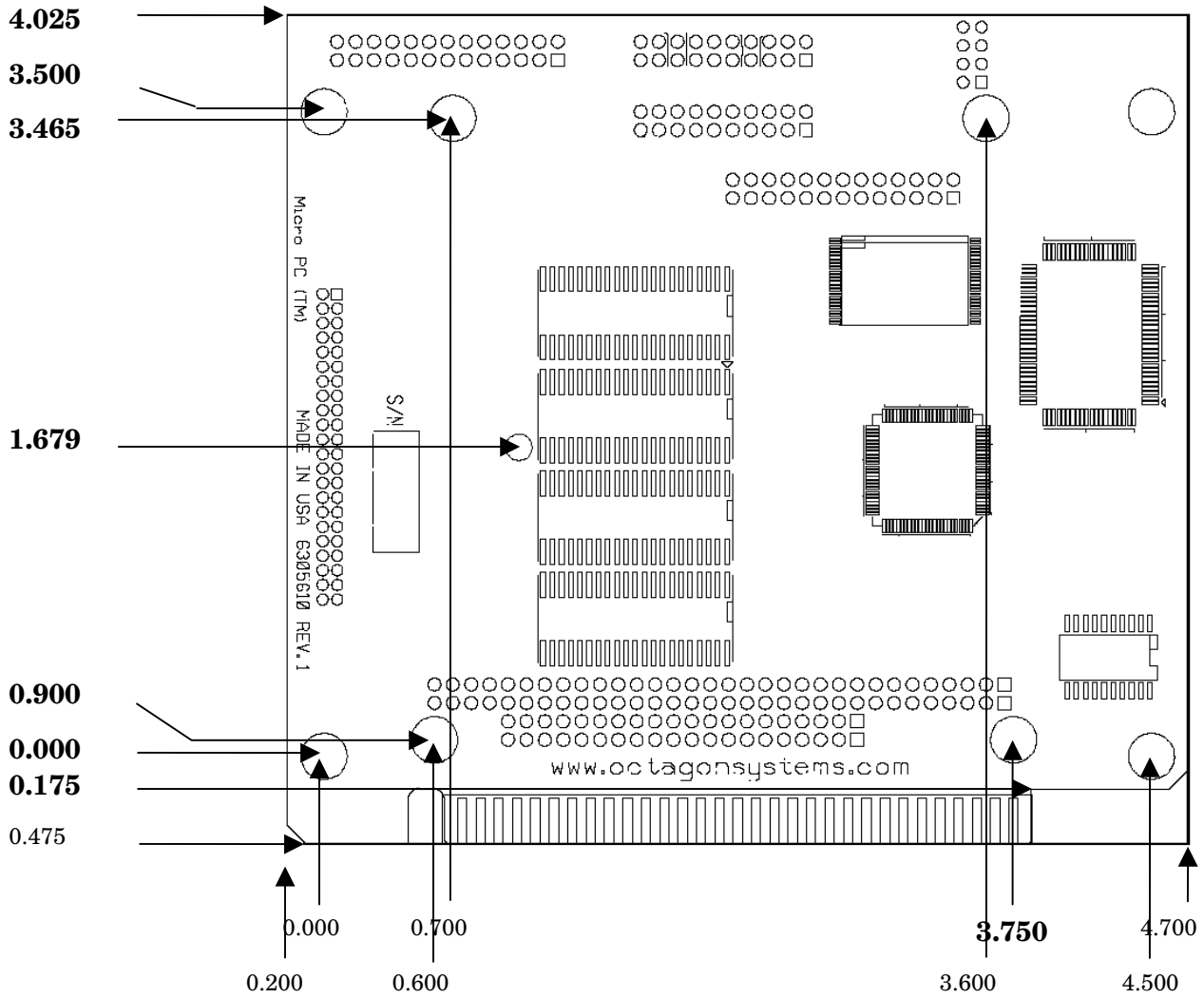


Figure 2-2 6225 CPU Card center-to-center hole dimensions



Your 6225 CPU Card can be installed in one of several ways:

- Plugging it directly into an 8-bit Micro PC card cage
- Using the optional PC mounting bracket and plugging it into any 8-bit passive ISA backplane
- Panel mounting it using the four mounting holes
- Stacking it with other Micro PC cards.

Using a Micro PC card cage

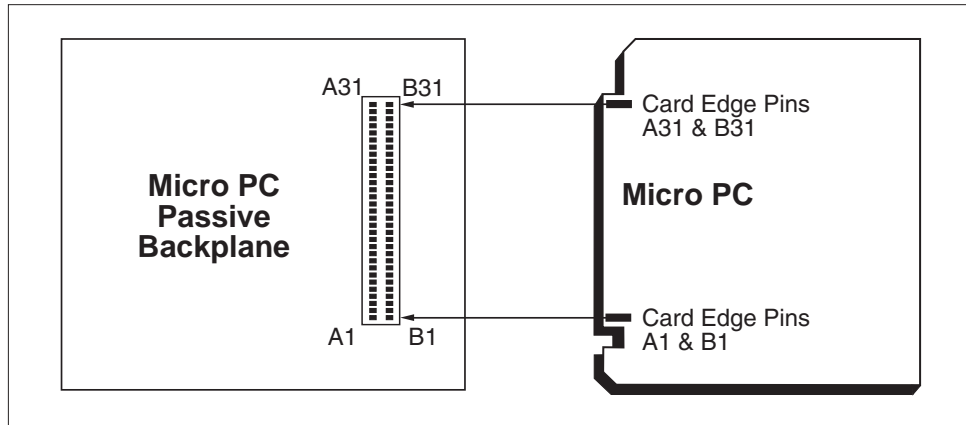
To install the 6225 CPU Card in a Micro PC card cage, you will need the following equipment (or equivalent):

- 6225 CPU Card
- Micro PC card cage (5xxx Card Cage)
- Power module (510x or 71xx Power Module)
- VTC-20F Cable
- Standard DB9 Serial Cable
- Null modem adapter
- 6225 CPU Card CD ROM
- PC SmartLINK with manual
- Your PC

To install the 6225 CPU Card:

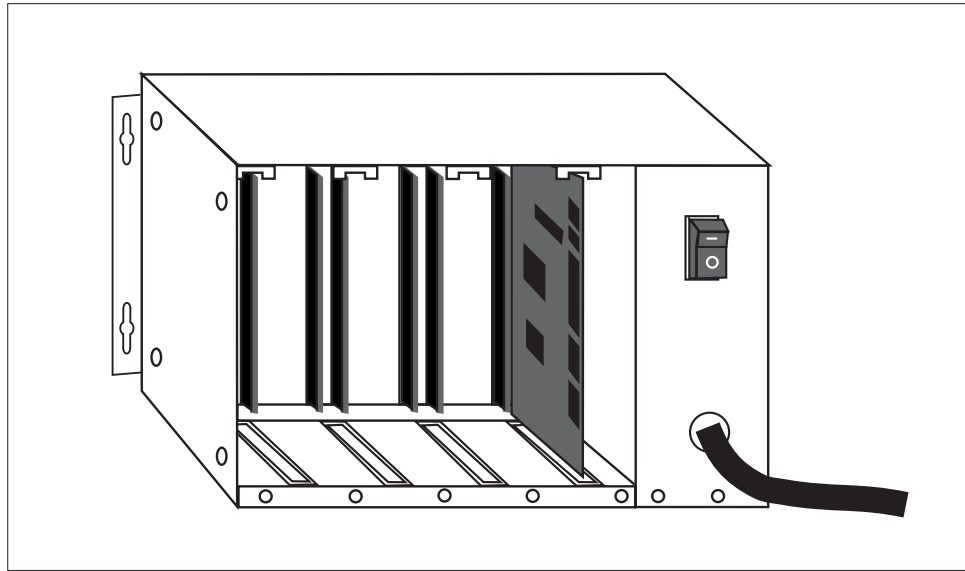
1. Refer to the component diagram on page 23 for the location of various connectors before installing the 6225 CPU Card.

Figure 2-3 Edge connector orientation



2. Attach the Octagon power module to the card cage following the instructions supplied with the power module.
3. Make sure power to the card cage is OFF.
4. Slide the 6225 CPU Card into the card cage. The ROM-BIOS label on the card should face away from the power supply. See Figure 2-4 for an illustration of a 6225 CPU Card in a Micro PC card cage.

Figure 2-4 Populated Micro PC card cage



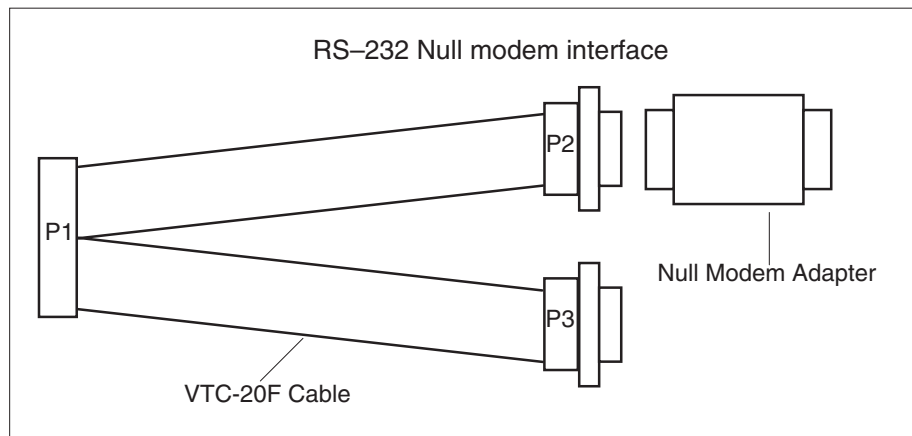
WARNING!

Plugging in the card incorrectly will destroy the card!

5. Connect the VTC-20F cable to J6 (COM1/COM2) on the 6225.

Note You must use COM1 on the 6225 CPU Card in order to establish a serial communications console I/O link with your PC.

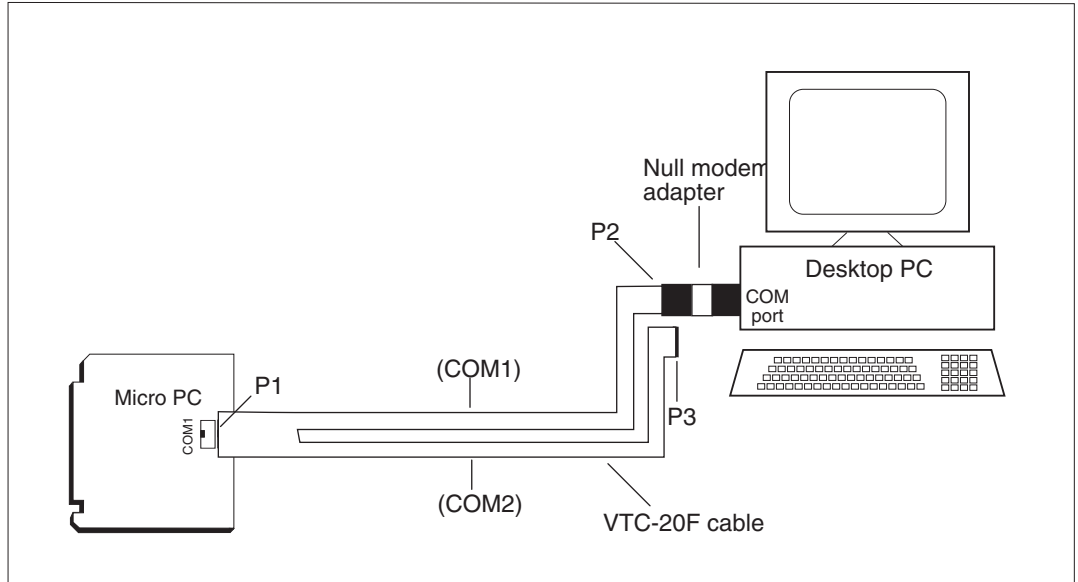
Figure 2-5 VTC-20F cable



6. Connect a standard serial cable to the connector on the VTC-20F cable which has a colored stripe on it. Connect a null modem cable to the other end of the serial cable.
7. If your PC has a 9-pin serial connector, connect the null modem adapter to any serial port (COM1 through COM4) on your PC. If your

PC has a 25-pin serial connector, attach a 9–25 pin adapter to your null modem adapter, then insert the matching end of the 9–25 pin adapter into the serial port. See Figure 2–6.

Figure 2–6 Serial communications setup



Note Refer to the PC SmartLINK manual for more information on using a desktop PC COM port other than COM1.

You are now ready to transfer files between your PC and the 6225 CPU Card. Continue with the section, *Establishing communications with the 6225 CPU Card*, in this chapter.

Panel mounting or stacking the 6225 CPU Card

To panel mount or stack the 6225 CPU Card, you will need the following equipment (or equivalent):

- 6225 CPU Card
- 5V power supply
- VTC–20F Cable
- Standard DB9 Serial Cable
- Null modem adapter
- 6225 CPU Card CD ROM
- PC SmartLINK with manual

- Your PC
- 5252MB stacking kit (required for stacking only) (P/N 3590)

If you are panel mounting the 6225 CPU Card, a connector is provided to supply the power. Refer to Figure 2-7 for an illustration of panel mounting the 6225 CPU Card.

WARNING!

Miswiring the voltage at P2 of the 6225 CPU Card or at the power connector of the 5252MB stacking kit (reversing +5V and ground, or applying a voltage greater than +5V), will destroy the card and void the warranty!

Figure 2-7 Panel mounting the 6225 CPU Card

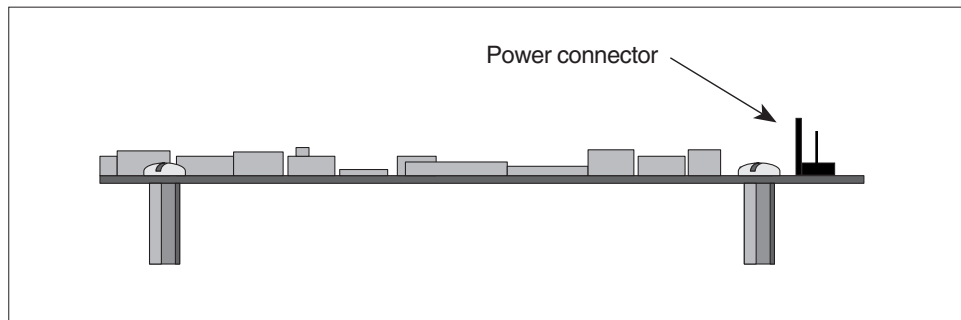
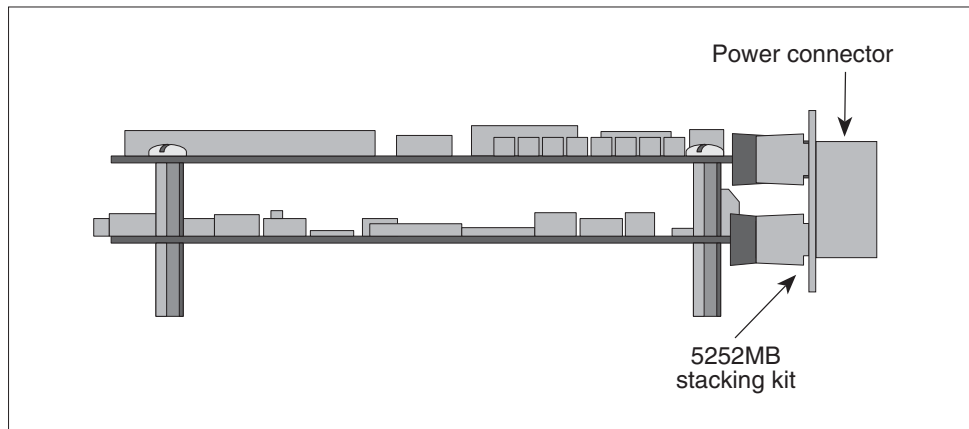


Figure 2-8 Stacking the 6225 CPU Card



1. To panel mount the 6225 CPU Card, use #4—40 standoffs and screws to secure the card. Figure 2-2 on page 24 shows the center-to-center mounting hole dimensions.

To stack the 6225 CPU Card, refer to the *5252MB stacking kit product sheet* enclosed with the kit. Then proceed with Step 2 in this section.

2. Connect the ground and 5V wires to the power connector of the 6225 CPU Card or P2 of the stacking kit.
3. Connect the VTC-20F cable to J6 (COM1/COM2) on the 6225.

Note You must use COM1 on the 6225 CPU Card in order to establish a serial communications console I/O link with your PC.

4. Connect a standard serial cable to the connector on the VTC-20F cable which has a colored stripe on it. Connect a null modem cable to the other end of the serial cable.
5. If your PC has a 9-pin serial connector, connect the null modem adapter to any serial port (COM1 through COM4) on your PC. If your PC has a 25-pin serial connector, attach a 9-25 pin adapter to your null modem adapter, then insert the matching end of the 9-25 pin adapter into the serial port. See Figure 2-5.

Note Refer to the PC SmartLINK manual for more information on using a desktop COM port other than COM1.

You are now ready to transfer files between your PC and the 6225 CPU Card. Continue with the section, *Establishing communications with the 6225 CPU Card* in this chapter.

Using the 6225 CPU Card in a passive ISA backplane

To plug the 6225 CPU Card into a passive ISA backplane, you will need the following equipment (or equivalent):

- 6225 CPU Card
- Unterminated backplane
- Mounting bracket (optional)
- Power module
- VTC-20F Cable
- Standard DB9 Serial Cable
- Null modem adapter
- 6225 CPU Card CD ROM
- PC SmartLINK with manual
- Your PC

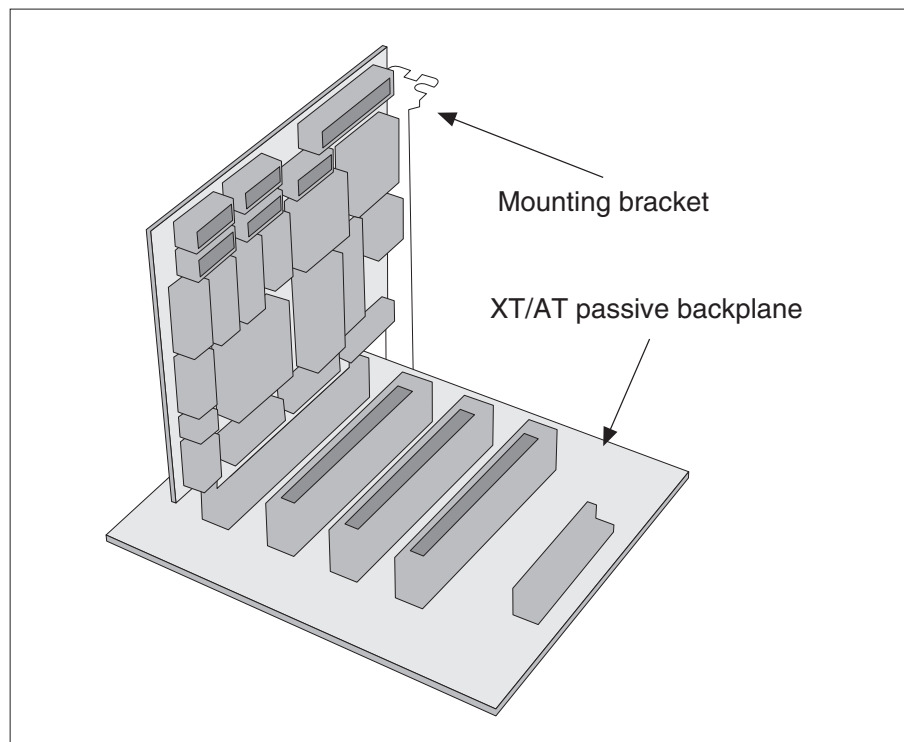
To install the 6225 CPU Card

1. Make sure power to the backplane is OFF.
2. Insert the 6225 CPU Card into a connector on the backplane (see Figure 2–9). Take care to correctly position the card's edge with the connector of the backplane. Figure 2–4 shows the relative positions of the 6225 CPU Card as it is installed into a backplane.

WARNING!

Incorrectly plugging the card into the backplane will destroy the card and void the warranty!

Figure 2–9 Using a passive ISA backplane



3. Connect the VTC–20F cable to J6 (COM1/COM2) on the 6225.

Note You must use COM1 on the 6225 CPU Card in order to establish a serial communications console I/O link with your PC.

4. Connect a standard serial cable to the connector on the VTC–20F cable which has a colored stripe on it. Connect a null modem cable to the other end of the serial cable.
5. If your PC has a 9–pin serial connector, connect the null modem adapter to any serial port (COM1 through COM4) on your PC. If your

PC has a 25-pin serial connector, attach a 9-25 pin adapter to your null modem adapter, then insert the matching end of the 9-25 pin adapter into the serial port. See Figure 2-6.

Note Refer to the PC SmartLINK manual for more information on using a desktop PC COM port other than COM1.

You are now ready to transfer files between your PC and the 6225 CPU Card. Continue with the section, *Establishing communications with the 6225 CPU Card* in this chapter.

Establishing communications with the 6225 CPU Card

1. Install PC SmartLINK (or other communications software) on your PC if you have not already done so. Refer to the PC SmartLINK manual for installation instructions.
2. Copy the 6225 CPU Card files from the supplied CD ROM to a subdirectory on your PC hard drive.

```
C:
MD C:\MPC
XCOPY D:\6225\*.* C:\MPC /S
```

3. Start PC SmartLINK. You are now ready to establish communications between your PC and the 6225 CPU Card.
4. Power on the 6225 CPU Card.
5. A logon message similar to the one below will appear on your PC monitor:

```
PhoenixBIOS(TM) A386 Version 1.03
Copyright (C) 1985-1992 Phoenix Technologies Ltd.
All Rights Reserved
Octagon Systems Corp. 40MHz 6225 CPU
Release x.xx-mm/dd/yy
```

```
640K Base Memory, 03072K Extended
```

```
6225 INT 17h BIOS extension v1.15 Copyright (c) 1995-2000,
Octagon Systems
```

```
Datalight FlashFx
V4.04.292 386 DOS
Copyright (c) 93-99
Patent US#5860082
Octagon Systems Vx.xx - 6225
SETSSD SSD1 /BEFORE
```

```

1MB AMD Flash detected in SSD1

Starting ROM-DOS...

HIMEM v7.10 (Revision 3.00.42)
Copyright (c) 1989-2000 Datalight, Inc.
Using PTL A20 Control (ON)
32 XMS handles available.
Minimum HMA usage is 0K.

VDISK v6.22 (Revision 3.00.42)
Copyright (c) 1989-2000 Datalight, Inc.

Installed 960KB XMS RAM disk as drive E:

6225 C:\>_

```

If you do not get the proper logon message:

- Check the PC SmartLINK serial parameters of your PC to make sure they are set correctly. Parameters should be 9600 baud, 8 data bits, no parity, and 1 stop bit.
 - Make sure a video card is not installed in the card cage
 - Make sure all jumpers are set to factory defaults
 - If the system still does not respond, refer to the *Troubleshooting* chapter.
6. Use the directory command to make sure your equipment and software are working properly. Enter:

```
6225 C:\> DIR
```

A directory listing of ROM-DOS files stored in the BIOS socket should appear:

```

Volume in drive C is SSD1
Volume Serial Number is 281F-9D7D
Directory of C:\

COMMAND  COM           34,565 01-21-2000  6:22a
CONFIG   SYS            78 04-26-2000  1:51p
AUTOEXEC BAT          43 09-13-1999  2:14p
DOS      <DIR>          04-28-2000  12:09a
UTILS    <DIR>          04-28-2000  12:09a
CAMBASIC <DIR>          04-28-2000  12:09a
        6 file(s)          34,686 bytes
                               469,472 bytes free

```

7. You are now ready to transfer files between your PC and the 6225 CPU Card.

Transferring files between the 6225 CPU Card and your PC

Once you have established communications between your PC and the 6225 CPU Card, you can serially download files to any read/write drive used by the 6225 CPU Card. You can then test and debug your application files. You can also upload files from the 6225 CPU Card to your desktop PC for editing and debugging.

When booting from the 6225 CPU Card BIOS drive, the default C: drive is SSD1 and D: is SSD0. Drive E: is a virtual drive. All drives assigned, can be accessed as read/write drives and files can be serially transferred to and stored on any of these drives.

Note The virtual drive is optional. If you do not need a virtual drive, do not use VDISK.SYS.

There are two methods to download files through the serial port to the 6225 CPU Card:

- The TRANSFER utility is used to download files, one at a time, to the 6225 CPU Card using the XMODEM protocol. TRANSFER.EXE resides on the 6225 CPU Card BIOS drive and on the 6225 CPU Card CD ROM and 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 In Windows 95 when the TRANSFER utility is used to download files, set the idle time sensitivity of PC SmartLINK on your desktop PC to “low” for TRANSFER to run quickly. To change your settings, follow the steps below:

1. Open Windows Explorer.
2. Select SL.EXE with the right mouse button.
3. Select the Properties menu item.
4. Select the Miscellaneous tab in the Properties window.
5. Move the Idle Sensitivity slide bar to low.
6. Select the Apply button.
7. Exit the Properties window.

- REMDISK/REMSERV utilities allow access to all of the files on a remote disk drive. REMDISK.EXE and REMSERV.EXE are located on the 6225 CPU Card BIOS drive and the 6225 CPU Card CD ROM. Once these programs are executed, single or multiple

files can then be transferred to and from the 6225 CPU Card using DOS COPY or XCOPY commands.

Note REMDISK/REMSERV will not work with Windows 95 or on a network. Use REMDISK/REMSERV with ROM-DOS or MS-DOS.

TRANSFER.EXE, REMDISK.EXE, and REMSERV.EXE are located on the 6225 CPU Card \DOS directory, and on the 6225 CPU Card CD ROM in the \ROMDOS\v7.1 directory. Refer to the *Software utilities* chapter for more information on these programs.

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.

The following information on downloading files between the 6225 CPU Card and your PC uses the example program DEMO.EXE. This file is on the 6225 CPU Card CD ROM in the \6225\DEMO directory.

Downloading files to the 6225 CPU Card using TRANSFER.EXE

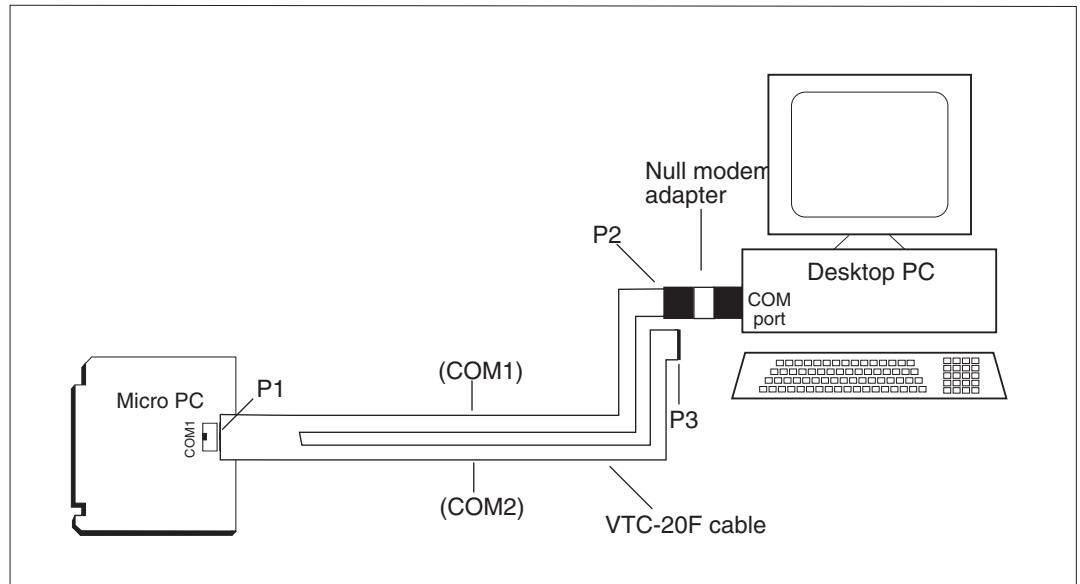
The following procedures assume you are using PC SmartLINK and that it is included in your directory path. For other communication programs, refer to their instructions on sending a file from your PC to a target system. Refer to the *Software utilities* chapter for specific information on using TRANSFER.EXE.

Hardware and software requirements:

- Desktop PC, running PC SmartLINK, connected by a VTC-20F cable, DB9 serial cable, and a null modem adapter to COM1 of the 6225 CPU Card
- A 6225 CPU Card running TRANSFER.EXE out of COM1.

1. Connect the equipment as per the following diagram:

Figure 2–10 Serial console connections



2. On the desktop PC, log into the directory which contains the file(s) you will download to the 6225 CPU Card, for example:

```
C:\MPC\6225\DEMO
```

3. Start PC SmartLINK and power on the 6225 CPU Card.
4. Execute the TRANSFER.EXE program from the 6225 CPU Card by entering:

```
6225 C:\> TRANSFER E:DEMO.EXE
```

Note In this case, E: is the virtual drive assigned in CONFIG.SYS. Any 6225 CPU Card read/write drive could be substituted.

Note When sending a file, enter the following:

```
6225 C:\> TRANSFER /S
```

The following message is displayed from the 6225 CPU Card:

```
Receiving E:DEMO.EXE . . .
```

5. Execute the following steps using PC SmartLINK:
 - a. Press <ALT><D> to enter the download screen.
 - b. Type in the name of the file to transfer, e.g. **DEMO.EXE** (if PC SmartLINK was not started in the DEMO directory as instructed in Step 2, then the entire path may have to be entered **C:\MPC\DEMO\DEMO.EXE**)

- c. To begin the transfer, do one of the following:
 - press **ENTER** (default download START)
 - tab to **START** and press **ENTER**
 - mouse click on the **START** button in the download screen.
- d. 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 6225 CPU Card is displayed:

```
Failed to receive E:DEMO.EXE!
Deleting E:DEMO.EXE
```

6. When the file transfer is complete, type the following DOS command to view the E: drive directory and confirm that your file has been transferred to the 6225 CPU Card:

```
6225 C:\> DIR E:
```

The system will display the contents of drive E:

```
Volume in drive E is VDISK vX.XX
Directory of E:\
DEMO          EXE                27264          06-07-96      2:57p
                                     1 file(s)      27264 bytes
```

7. To execute the program you have just downloaded, type:

```
6225 C:\> E:DEMO
```

The DEMO program displays a message on your PC.

Downloading files to the 6225 CPU Card using REMDISK/REMSERV

There are three methods of using REMDISK/REMSERV with a 6225 CPU Card:

- 6225 CPU Card with no video card and one serial cable
- 6225 CPU Card with no video card, two PCs, and two serial cables
- 6225 CPU Card with a 5420 video card and one serial cable.

Refer to the *Software utilities* chapter for specific information on using REMDISK.EXE and REMSERV.EXE.

Note REMDISK/REMSERV will not work with Windows 95 or on a

network. Use REMDISK/REMSERV with ROM-DOS or MS-DOS.

6225 CPU Card with no video card and one serial cable

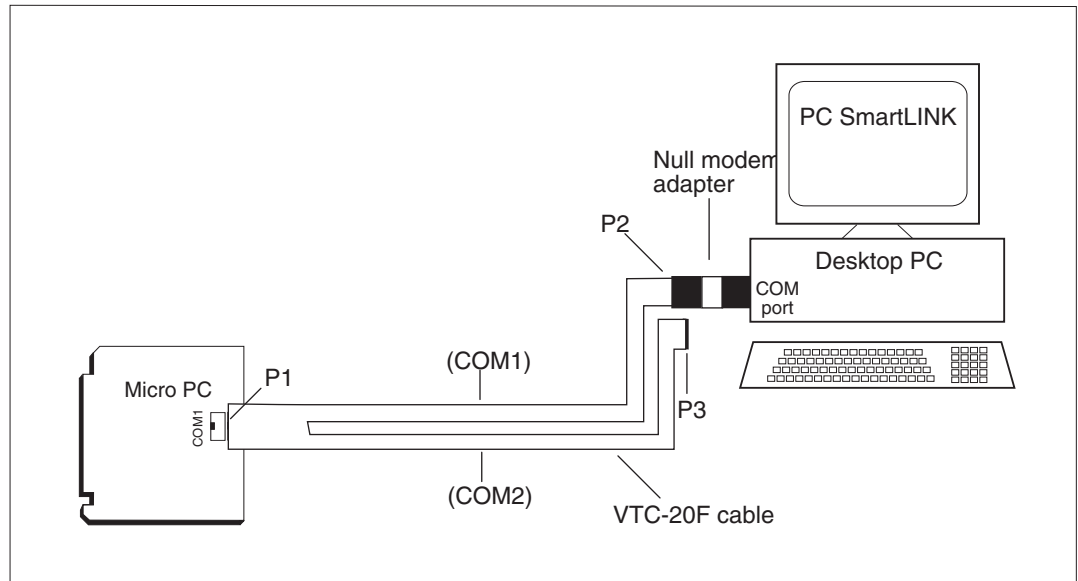
Hardware and software requirements:

- Desktop PC, running C:\MPC\DOS\REMDISK, connected by a VTC-20F cable, DB9 serial cable, and a null modem adapter to COM1 of the 6225 CPU Card

- A 6225 CPU Card running C:\DOS\REMSERV out of COM1

1. Connect the equipment and load appropriate software on each system as per the following diagram:

Figure 2-11 Downloading files to the 6225 CPU Card with no video card using REMDISK/REMSERV



2. On the desktop PC, start PC SmartLINK from the C:\MPC\ROMDOS\v7.1 directory and power on the 6225 CPU Card.
3. Execute REMSERV.EXE on the 6225 CPU Card. Read/write SSD flash drive C: is the shared drive and COM1 is the default port.
Enter:

```
6225 C:\DOS> REMSERV C:
```

The following message is displayed from the 6225 CPU Card:

```
REMSERV v1.0  
Copyright (c) 1990-1994 Datalight, Inc.  
All rights reserved.  
Using COM1 at 115K+ baud. Accessing Drive C:
```

Time-out is 9 seconds
Press <Esc> to Exit.(There may be a delay before exit occurs)

4. Exit PC SmartLINK by pressing <ALT><X>.
5. Execute REMDISK.EXE on the PC, by entering:

```
6225 C:\> REMDISK
```

The following message is displayed on the PC:

```
Remote Disk v1.0  
Copyright (c) 1990-1994 Datalight, Inc.  
All rights reserved.  
Installed as Drive E: /COM1 /B115+ /T10
```

Note REMDISK.EXE is located in the \ROMDOS\v7.1 directory on the 6225 CPU Card CD ROM. REMDISK assigns the remote drive as the last drive in the system. In this case, drive E: was assigned.

6. Files are transferred to the 6225 CPU Card's read/write drives by using the DOS COPY or XCOPY commands. Enter:

```
6225 C:\> COPY C:\MPC\6225\DEMO.EXE E:  
6225 C:\> DIR E:  
6225 C:\> E:DEMO.EXE
```

The DEMO program displays a message on your PC.

In this case, drive E: is the remote read/write SSD flash disk drive of the 6225 CPU Card. Files are easily copied between the drives.

7. When finished, execute:

```
6225 C:\> REMDISK /U
```

This unloads REMDISK from the desktop PC.

8. Restart PC SmartLINK and reset the 6225 CPU Card.

6225 CPU Card with no video card, two PCs, and two serial cables

The first desktop PC is used as the terminal for the 6225 CPU Card, and the second desktop PC's hard drive is accessed as a remote drive, containing the files to be downloaded to the 6225 CPU Card.

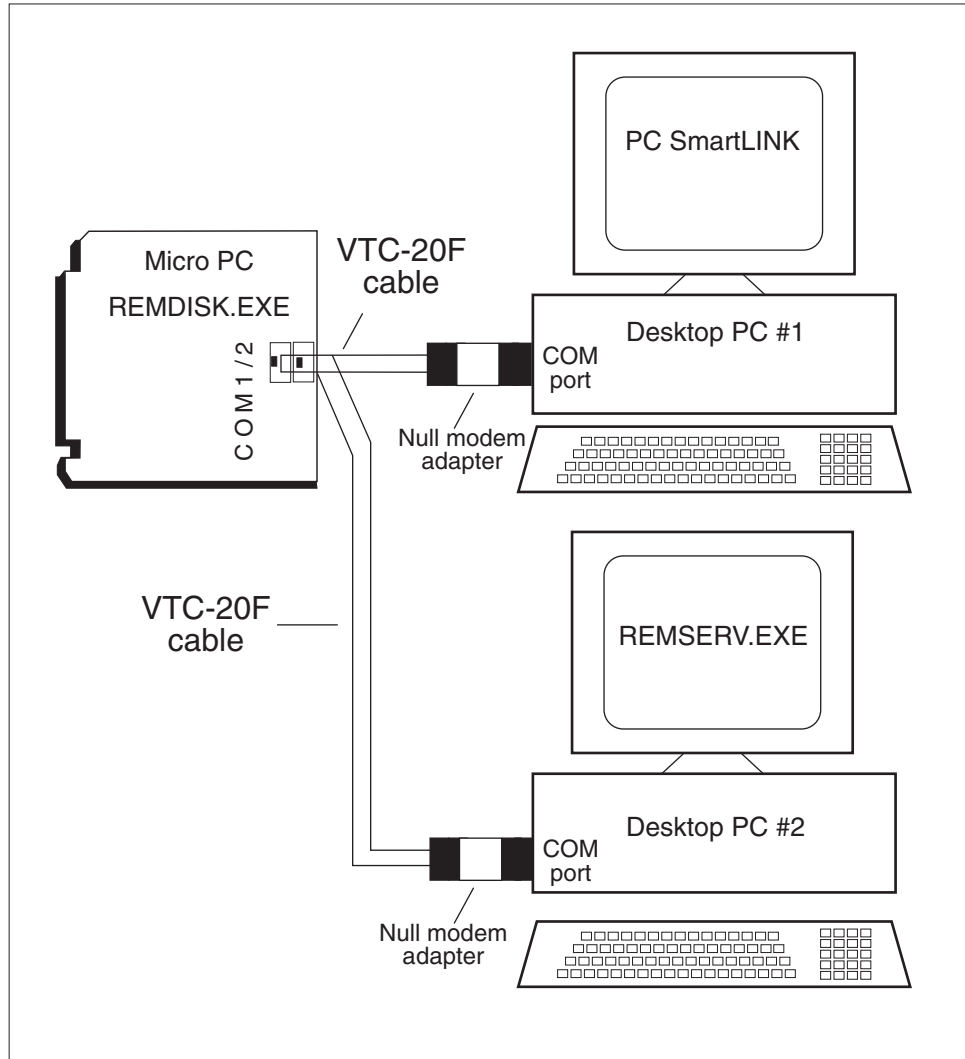
Hardware and software requirements:

- Desktop PC, running PC SmartLINK, connected by a VTC-20F cable, DB9 serial cable, and a null modem adapter to COM1 of the 6225 CPU Card
- Desktop PC, running REMSERV.EXE, connected by a VTC-20F

cable, DB9 serial cable, and a null modem adapter to COM2 of the 6225 CPU Card

- A 6225 CPU Card running REMDISK.EXE from COM2.
1. Connect the equipment and load the appropriate software on each system as per the following diagram:

Figure 2-12 Downloading files to the 6225 CPU Card with no video card and two PCs



2. On PC #1 (i.e., the terminal PC), start PC SmartLINK and power on the 6225 CPU Card.
3. Execute REMDISK.EXE from COM2 on the 6225 CPU Card by entering:

```
6225 C:\> REMDISK /COM2
```

The following message is displayed from the 6225 CPU Card:

```
Remote Disk v1.0
Copyright (c) 1990-1994 Datalight, Inc.
All rights reserved.
Installed as Drive F: /COM2 /B115+ /T10
6225 C:\>
```

4. On PC #2 (i.e., the remote disk drive PC), execute REMSERV.EXE by entering:

```
6225 C:\> REMSERV C:
```

The following message is displayed on PC #2:

```
REMSERV v1.0
Copyright (c) 1990-1994 Datalight, Inc.
All rights reserved.
Using COM1 at 115K+ baud. Accessing Drive C:
Time-out is 9 seconds
Press <Esc> to Exit.(There may be a delay before exit occurs)
```

5. At PC #1, access the remote disk drive by entering:

```
6225 C:\> F:
6225 G:\> CD F:\MPC\PC 6225\DEMO
```

6. Files are transferred to the 6225 CPU Card read/write drives by using the DOS COPY or XCOPY commands. Enter:

```
6225 F:\MPC\6225\DEMO> COPY DEMO.EXE C:
6225 F:\MPC\6225\DEMO> DIR C:
6225 F:\MPC\6225\DEMO> C:DEMO.EXE
```

The DEMO program displays a message on your PC.

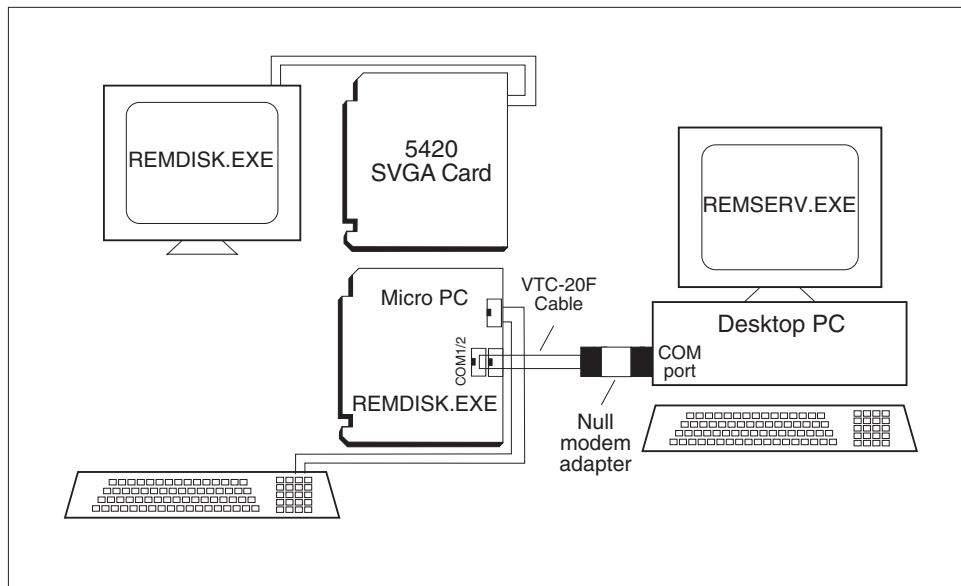
In this case, drive F: is the remote disk drive of PC #2, and drive C: is the read/write SSD flash disk drive of the 6225 CPU Card. Files are easily copied between the drives.

6225 CPU Card with a video card and one serial cable

Hardware and software requirements:

- Desktop PC, running REMSERV, connected by a VTC-20F cable, DB9 serial cable, and a null modem adapter to COM1 or COM2 of the 6225 CPU Card.
 - A 6225 CPU Card system, including a keyboard, a 5420 SVGA video card and VGA monitor, running REMDISK from COM1.
1. Connect the equipment and load the appropriate software on each system as per the following diagram:

Figure 2-11 Downloading files to the 6225 CPU Card with a video card



2. On the 6225 CPU Card system, execute REMDISK.EXE by entering:

```
6225 C:\> REMDISK
```

The following message is displayed on the 6225 CPU Card monitor:

```
Remote Disk v1.0
Copyright (c) 1990-1994 Datalight, Inc.
All rights reserved.
Installed as Drive F: /COM1 /B115+ /T10
```

Note REMDISK assigns the remote drive as the last drive in the system. In this case, drive F: was assigned.

3. Execute REMSERV.EXE on the desktop PC:

```
6225 C:\> REMSERV C:
```

The following message is displayed on the PC:

```
REMSERV v1.0
Copyright (c) 1990-1994 Datalight, Inc.
All rights reserved.
Using COM1 at 115K+ baud. Accessing Drive C:
Time-outs is 9 seconds
Press <Esc> to Exit.(There may be a delay before exit occurs)
```

Note REMSERV.EXE is located in the 6225 CPU Card CD ROM \ROMDOS\v7.1 directory.

4. Files are transferred to the 6225 CPU Card's read/write drives by using the DOS COPY and XCOPY commands. From the 6225 CPU Card system, enter:

```
6225 C:\> COPY F:\MPC\6225\DEMO.EXE C:
6225 C:\> DIR C:
6225 C:\> C:DEMO.EXE
```

The DEMO program displays a message on the 6225 CPU Card monitor.

In this case, drive F: is the remote PC disk drive, and C: is the read/write SSD flash drive on the 6225 CPU Card. Files are easily copied between the drives.

5. When finished, on the 6225 CPU Card system, execute:

```
6225 C:\> REMDISK /U
```

This unloads REMDISK from the 6225 CPU Card.

6. On the desktop PC press <ESC> to exit REMSERV.

Chapter 3: **Setup programs**

This chapter discusses running the SETUP configuration program, the SETSSD program, and the PMISSETUP program on the 6225 CPU Card.

- SETUP — Configures devices set up by the BIOS such as serial ports, floppy drives, etc.
- SETSSD — Configures FlashFx device order.
- PMISSETUP — Configures power management options at a more detailed level than SETUP.

SETUP

SETUP can be entered in one of two ways:

- Running SETUP.COM
- Pressing the “backspace” key followed by the “S” key during BIOS POST sequence (this occurs between the memory test and bootup).

Also, by removing the USESETUP jumper from the “S” position at W6[1-2], you may force the setup to temporarily revert to the defaults shown in the following table, which allows the user to reconfigure the setup.

The SETUP program defines the 6225 CPU Card system parameters. It 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 SSD0 drive and on the 6225 CPU Card CD ROM.

Table 3-1 6225 setup parameters and defaults

SETUP parameters	Description	Default
Serial console for COM1	Specifies that COM1 is to be used for console if video card is not present	Enabled
COM1 console baud rate	Specifies communications rate between PC & 60xx when no video card is in use	9600
Power-on memory test	Extensive memory testing performed on bootup	Enabled
Boot sequence	Specifies whether the floppy drive will be ignored as a boot device	C: Only
Serial port A	Specifies COM1 enable/disable	Enabled
Serial port B	Specifies COM2 enable/disable	Enabled
Serial port C	Specifies COM3 enable/disable	Enabled
Serial port D	Specifies COM4 enable/disable	Enabled
Parallel (LPT) port	Specifies LPT port enable/disable	Enabled
Parallel port mode	Specifies mode to use with parallel port	Bidirectional printer port
Parallel port address	Specifies LPT address	378h
Number of floppy drives	Specifies number of floppy drives attached	0
Number of hard drives	Specifies number of hard drives attached	0
SETUP entry via hotkey	Specifies <backspace><S> hotkey enable/disable	Enabled
Power management for DOS	Specifies power management enable/disable	Enabled
Time update after suspend	Specifies to allow update of time after suspend mode	Enabled
Shadow video BIOS area	Specifies video BIOS shadow enable/disable	Disabled
Shadow C8000h-CFFFFh	Shadow enable/disable	Disabled
Shadow D0000h-D7FFFh	Shadow enable/disable	Disabled
Shadow D8000h-DFFFFh	Shadow enable/disable	Disabled
Enable SSD0 Scan	Searches for devices installed in SSD0, such as a DOC	Disabled

Running SETUP

1. Make sure you have established a serial communications console I/O link between the 6225 CPU Card and your PC. Refer to the *Quick start* chapter for more information on establishing communications with your 6225 CPU Card.
2. Enter:

```
6225 C:\> SETUP
```

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 6225 CPU Card 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.

Note Options having an * are default settings.

- Serial Console on COM1:
Enabled*
Disabled

WARNING!

Disabling the serial console when there is no video card present will stop further serial console communication with the system after the system resets. Once disabled, you may re-enable the serial console by running SETUP. To run SETUP, choose one of the following methods:

- **Remove the USESETUP jumper, reboot, and run SETUP**
 - **Install a video card/monitor, reboot, and run SETUP. (This method disables the serial console.)**
- 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 When using a DOC, the “C: Only” option is ignored

- Serial Port A:
Enabled*
Disabled
- Serial Port B:
Enabled*
Disabled
- Serial Port C:
Enabled*
Disabled
- Serial Port D:
Enabled*
Disabled
- 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

Note Do not use 278H or a conflict with digital I/O will occur.

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 at these modes for enhanced performance.

- Number of floppy drives:
0*, 1, 2
- Onboard floppy controller:
Enabled
Disabled*
- 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

Note The 6225 CPU Card does not support floppy drives on the 5800A without first making some modifications to the 5800A. Call Technical Support for assistance.

- Onboard IDE interface:
Enabled
Disabled*

Note The 6225 has an on-board floppy controller and IDE controller. If an external controller is desirable, the on-board controllers can be disabled through SETUP.

- Auto Drive Configuration
Enabled*
Disabled
- Drive 0 parameters
Cylinders (xxx):
Heads (x):
Sectors (xx):
- Setup entry via hotkey
Enabled*
Disabled
- Power management
Enabled*
Disabled
- Time update after suspend
Enabled*
Disabled
- Hardware watchdog timeout
Disabled*
Enabled
8 seconds
16 seconds
32 seconds
64 seconds
- Shadow video BIOS area
Disabled*

- Enabled
- Shadow C8000H - CFFFFH
Disabled*
Enabled
- Shadow D0000H - D7FFFH
Disabled*
Enabled
- Shadow D8000H - DFFFFH
Disabled*
Enabled
- Shadow D8000H - DFFFFH
Disabled*
Enabled
- Enable SSD0 scan
Disabled*
Enabled

Note When using an M-Systems DOC, this option must be 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.

SETUP example

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

```
OCTAGON SYSTEMS CORPORATION
6225 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
Parallel (LPT) Port:       ENABLED
Parallel Port Mode:        Bidirectional Printer Port
Number of floppy drives:    1
Floppy drive 1 size:       3.5", 1.44 MB
Number of hard drives:     1
Auto Drive Configuration:  ENABLED
SETUP Entry via Hotkey:    ENABLED
```

```
Power Management:          DISABLED
Shadow Video BIOS Area:    DISABLED
Shadow C8000h-CFFFFh:     DISABLED
Shadow D0000h-D7FFFh:     DISABLED
Shadow D8000h-DFFFFh:     DISABLED
Enable SSD0 scan:         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.
6225 C:\>
```

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

Note Power management should be disabled when using CAMBASIC.

SETSSD

SETSSD allows the user to set or change the FlashFx drive (SSD) order. FlashFx drives are “simulated” hard drives. They can exist before or after any IDE drives and can appear in any order. By setting the order, the SSDs may be accessed as C:, D:, etc. For example,

- To set SSD0 first and SSD1 second, enter the following command:

```
6225 C:\> SETSSD SSD0 SSD1
```

If there are other hard drives on the system, add the **/before** option to place the order of the SSDs before the hard drives, or add the **/after** option to place the SSDs after the hard drives. For example,

- To set SSD0 as the first drive, SSD1 as the second drive, and an IDE drive as the third drive, enter the following command:

```
6225 C:\> SETSSD SSD0 SSD1 /before
```

- To set the IDE drive as first in order and SSD0 as second, enter the following command:

```
6225 C:\> SETSSD SSD0 /after
```

In the last example, the IDE drive is C: and SSD0 is D:. Other drive letter designations may be added by device drivers (such as VDISK.SYS), which are in the CONFIG.SYS file on the boot drive.

The boot drive is based upon the drive order set by the SETSSD command and by SETUP’s “boot sequence” option. If the boot sequence is set to “A: THEN C:,” the system will look for a floppy disk in drive A:. If a diskette is not installed, or a floppy is not defined, the boot drive will be the first drive specified in the SETSSD command. If the boot sequence is set to “C: ONLY,” the check for a disk is bypassed.

Note The SETSSD parameters may also be overwritten by removing the USESETUP jumper from the “S” position at W6[1:2] and resetting the system. If the parameters specified at the FlashFx first/second drive prompt are different from the previous SETSSD command, and you answered “No” to the “Save” prompt, the SETSSD output will not be accurate. Therefore, we recommend that you answer “Yes” to the save option to prevent confusion.

Note After you run SETSSD and the drive order has changed, the new parameters will take effect after a reset.

Note The drive order affects the number entered at the FFXFMT 8n command.

PMISETUP

PMISETUP allows the user to customize the power management features of the 6225 CPU Card. Refer to the *CPU power management* chapter. See also the *Software utilities* chapter for details.

Chapter 4: **Save and run programs**

Save and run your programs on the 6225 CPU Card

Once you have written, tested and debugged your application, you can then save it to flash memory in SSD1 or SSD0 (if flash, DOC or SRAM). When you reboot the 6225 CPU Card, your program can automatically load into DOS memory and execute. As shipped from the factory, SSD1 already contains a bootable ROM-DOS.

This chapter describes the following:

- Saving an application program to SSD1
- Autoexecuting the program from the 6225 CPU Card
- Overriding autoexecution of your program.

The information in this chapter assumes you are using ROM-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. We recommend booting from SSD1, using your own DOS, when using programs with undocumented DOS calls. Refer to the section *Adding operating system startup files* in the *Managing SSDs* chapter for more information on saving and autoexecuting programs.

Saving programs and support files

By default, the drive in SSD1 comes preformatted from the factory, loaded with ROM-DOS startup files and an example demo program. To replace the demo program on SSD1 with your own, see the section *Adding your application*, in this chapter. To reformat an SSD or to add your own operating system, please refer to the *Managing SSDs* chapter.

WARNING!

Reformatting SSD1 requires the use of a floppy or a hard disk to restore system files.

Adding your application

To add your application to your SSD, do the following:

1. Three methods of copying your application to the SSD are available. Do one of the following:
 - From a local drive to the 6225 CPU Card, issue the COPY command.
 - From a host drive, download your application by issuing the TRANSFER command when using PC SmartLINK. Refer to the section, *Transferring files between the 6225 CPU Card and your PC* in the *Transferring files / remote disks* chapter.
 - To establish a remote drive and copy from it, issue the REMDISK and REMSERV commands. Refer to the section, *Transferring files between the 6225 CPU Card and your PC* in the *Transferring files / remote disks* chapter.
2. Add or remove any device drivers for your application. You may want to do the same for the CONFIG.SYS file on SSD1. Remember to add these drivers to your drive as well.
3. To autoexecute your application, add your application name to the AUTOEXEC.BAT file. To replace our example program (DEMO.EXE) with your application, substitute DEMO in the AUTOEXEC.BAT file with your applications' filename.

Autoexecuting your application

This section describes how to autoexecute your application.

1. To autoexecute your application in SSD1, use the SETSSD command to define your SSD as the boot device. Since you need to define the order of SSD1 as the first of the SSDs (and before any IDE drives), enter the following command:

```
6225 C:\> SETSSD SSD1 SSD0 /before
```

2. Reset the system. SSD1 is now drive C: and your application should begin execution.

Note If the SETUP option "Boot Sequence" is set to "A: THEN C:", remove any floppy in drive A: before resetting the system.

Note The SETSSD options are not used when the SETUP jumper W6[1-2] is not jumpered.

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 your local keyboard. For more information, see your ROM-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:.

Option 4

1. Remove the SETUP jumper W6[1-2].
2. Boot the system. When prompted for FlashFx drives enter them. When prompted to Save, select “Y”. When prompted for Send Abort Sequence select “Y”.

Overview: **Section 2 – Hardware**

Section 2 discusses usage, functions, and system configurations of the 6225'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: Digital I/O port
- Chapter 11: Ethernet
- Chapter 12: PC/104 expansion

Chapter 5: **Serial ports**

Description

The 6225 has four serial ports, COM1 through COM4. These serial ports interface to a printer, terminal, or other serial device. All 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 and COM3 are dedicated 8-wire RS-232. COM2 can be configured with jumpers to provide 8-wire RS-232C, RS-422 or RS-485. COM4 can be configured with jumpers to provide 8-wire RS-232C or RS-485.

All serial ports have the following specifications:

- 16550 compatible
- 16-byte FIFO buffers
- IEC 1000, level 3, ESD protection
 - Contact discharge ± 4 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

Table 5–1 shows the configurations available for the COM ports.

Table 5–2 shows the jumper settings for the COM port configurations.

Table 5–1 COM port configurations

COM Port	Communication Type	Address / Interrupts
COM1	RS-232C, eight wire	Address selectable in SETUP, default 3F8h. Interrupt IRQ4
COM2	Jumper-selectable between eight wire RS-232C and RS-422/485. Additional jumper selection for Normal or Optomux 422/485 protocols.	Address selectable in SETUP, default 2F8h. Interrupt IRQ3
COM3	RS-232C, eight wire	Address selectable in SETUP, default 3E8h. Interrupt IRQ12
COM4	Jumper-selectable between eight wire RS-232C and RS-485	Address selectable in SETUP, default 2E8h. Interrupt IRQ11

Table 5-2 COM port jumper settings

COM Port	Communication Type	Jumper Settings
COM2	RS-232C*	W1[7-8] Off W5[2-3] On*
	RS-422	W1[1-2] On W1[3-4] On** W1[5-6] Off W1[7-8] On W5[1-2] On W6[5-6] On *** Optomux Mode W6[5-6] Off
	RS-485	W1[1-2] Off W1[3-4] On** W1[5-6] On W1[7-8] On W5 Off W6[5-6] On *** Optomux Mode W6[5-6] Off
	Normal Mode	W6[5-6] On ***
COM4	RS-232C*	W4[2-3] On*
	RS-485	W4 Off
*Default ** This jumper terminates the network. If the 6225 is not at an end of the network, leave this jumper off. *** This jumper should remain On unless you specifically want Optomux mode.		

COM1/COM2

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. For example, when COM1 is the console, COM1 uses IRQ4.

COM2 is jumper-configurable for full 8-wire RS-232, RS-422, or RS-485. RS-232 uses pins 11-20 on the J6 connector. RS-422 and RS-285 use the 5-pin header at J3.

Note Although COM2 uses different headers for RS-232 and RS-422/485, you cannot use both functions at the same time.

COM2 uses the RTS pin to enable or disable the RS-422/485 transmit/receive function. That is, a logic '1' in bit position 1, register offset 4, will enable the transmitter function and disable the receiver function. A logic '0' in bit position 1, register offset 4, will disable the

transmitter function and enable the receiver function.

RI on COM2 can be used as a method of “waking up” the 6225 from a power management SUSPEND operation. Refer to the CPU power management chapter for more information. The default I/O address for COM2 is 2F8h.

COM2 has one additional feature. Jumper W6[5–6] selects between normal RS–422/485 protocol mode and Optomux–compatible RS–422/485 protocol mode. For additional information on Optomux–compatible mode, refer to Optomux mode in this chapter.

IRQ3 is dedicated to COM2. If COM2 is not used under interrupt control, IRQ3 may be used by other resources.

COM1 and COM2 share the J6, 20–pin header for RS–232. Table 5–3 shows the pinout for COM1 and COM2.

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

Table 5–3 COM1/2 and COM3/4 pinout for RS–232 (J6 and J7 connectors)

Port	Pin	RS–232C	
COM2/4 (J6/J7)	20	nc	
	19	GND	
	18	RI	
	17	DTR	
	16	CTS	
	15	TXD	
	14	RTS	
	13	RXD	
	12	DSR	
	11	DCD	
COM1/3 (J6/J7)	10	nc	
	9	GND	
	8	RI	
	7	DTR	
	6	CTS	
	5	TXD	
	4	RTS	
	3	RXD	
	2	DSR	
	1	DCD	

COM3/COM4

COM3 is a full 8-wire RS-232. The I/O address for COM1 is 3E8h. IRQ12 is dedicated to COM3. If COM3 is not used under interrupt control, IRQ12 may be used by other resources.

COM4 is jumper-selectable for full 8-wire RS-232 interface or 2-wire RS-485. The I/O address for COM4 is 2E8h. IRQ11 is dedicated to COM4. If COM4 is not used under interrupt control, IRQ11 may be used by other resources.

COM4 RS-232 uses pins 11–20 on the J7 connector. RS-485 uses the 5-pin header at J8. Table 5-4 shows the pinout for J8.

Note Although COM4 uses different headers for RS-232 and RS-485, you cannot use both functions at the same time.

COM4 uses the RTS pin to enable or disable the RS-485 transmit/receive function. That is, a logic '1' in bit position 1, register offset 4, will enable the transmitter function and disable the receiver function. A logic '0' in bit position 1, register offset 4, will disable the transmitter function and enable the receiver function.

COM3 and COM4 share the J7, 20-pin header for RS-232.

Note See the Accessories appendix for mating information on COM3 and COM4.

Table 5-4 COM2 (RS-422/485) and COM4 (RS-485) pinout (J3 and J8 connectors)

Port	Pin	RS-422	RS-485
COM2 J3 Header	5	RX-	DATA-
	4	RX+	DATA+
	3	GND	GND
	2	TX-	
	1	TX+	
COM4 J8 Header	5	Not available	DATA-
	4	on COM4	DATA+
	3		GND
	2		
	1		

Function and use of serial ports

COM1 as serial console device

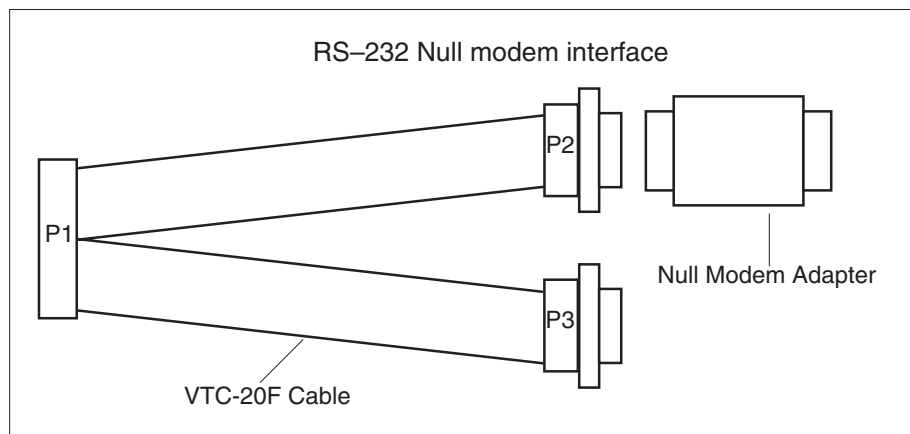
You can use COM1 as a console device. See the *Console devices* chapter for more information.

Mating receptacle

Use a VTC-20F or VTC-20M cable to connect the COM ports to external serial equipment. The P2 and P3 connectors are DB-9 female (VTC-20F) or DB-9 male (VTC-20M) connectors which plug directly into a 9-pin PC serial cable.

Note When interfacing the 6225 to your desktop PC, you must use a null modem adapter.

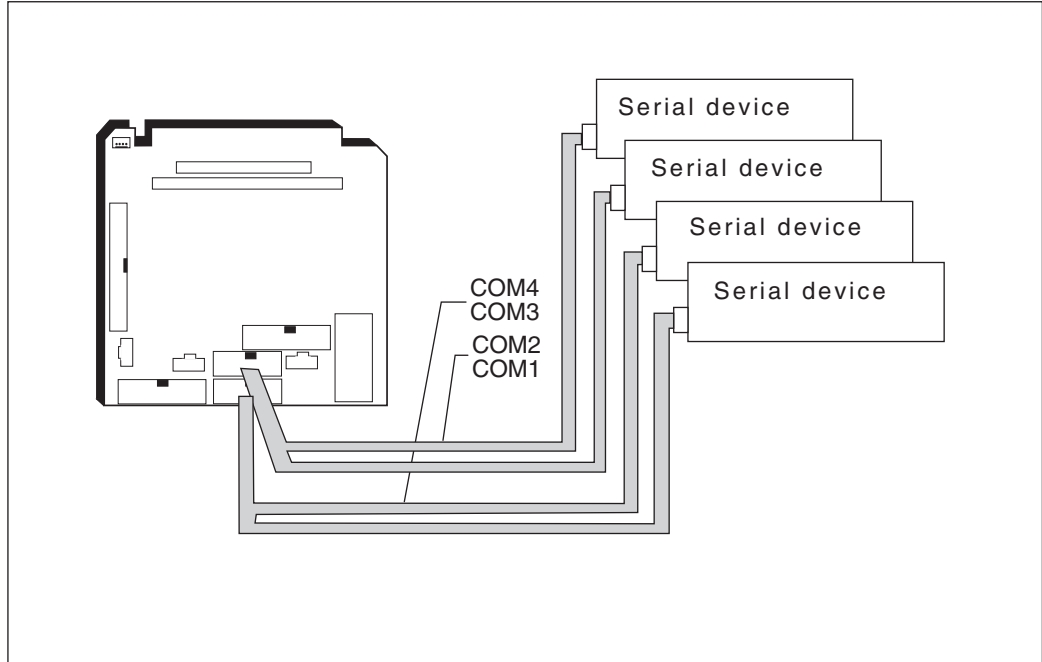
Figure 5-1 Null modem adapter and VTC-20F/VTC-20M cable



COM1 through COM4 as RS-232 I/O

COM1 through COM4 can be used as RS-232 serial ports in 8-wire configurations. Use a VTC-20F or VTC-20M cable to interface from the 6225 to the serial device.

Figure 5–2 6225 serial devices



COM2 and COM4 as RS–485 network

RS–485 uses 2–wire differential signaling to communicate between the devices on a network. Differential signal reduces the effect of environmental noise, allowing communication over distances up to 1200 meters.

An application may implement a node as either the “host” node or as a “remote” node in an RS–485 network. There can be as many as 32 nodes without any bus repeaters in the network. A “host” is referred to as the node that initiates communication; while a “remote” is referred to as a node that is addressed by the host.

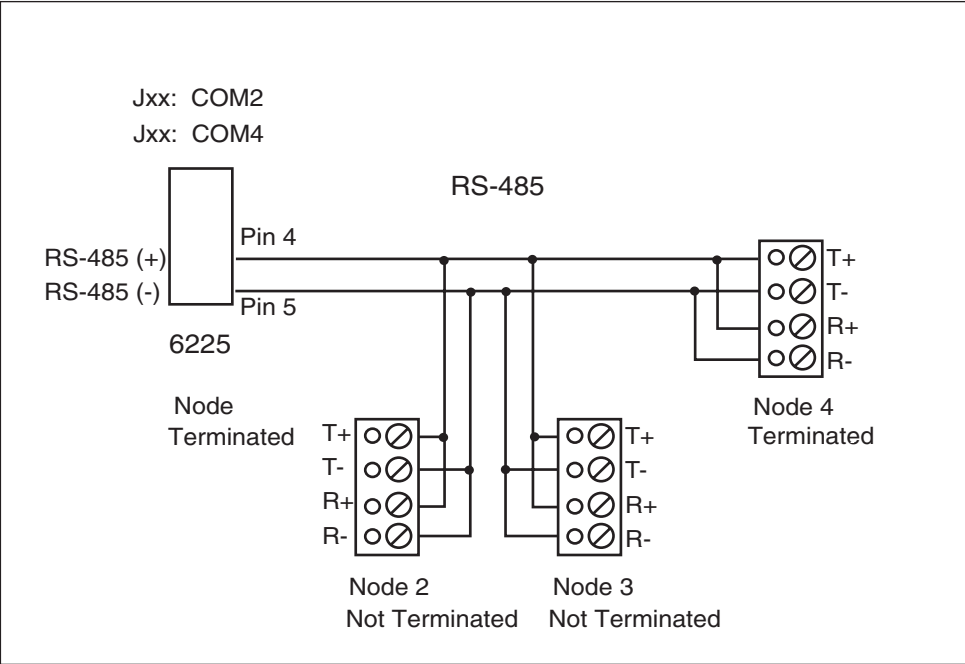
In any given communication sequence in an RS–485 network, there can only be one host. The host is responsible for initiating communication, maintaining network registration, and providing housekeeping tasks with other nodes. Remotes, however, cannot initiate a communication. They can only respond to messages that are addressed to them from the host.

The devices at each end of an RS–485 network must be terminated.

Any node located between the end points should not be terminated. If the 6225 is at one end of the network, use jumper W1[3-4] to make sure that both ends of the network are terminated.

Figure 5-3 shows a typical RS-485 network.

Figure 5-3 RS-485 network



COM2 as RS-422 network

RS-422 uses two pairs of 2-wire differential signaling to communicate between the devices on a network. Differential signal reduces the effect of environmental noise, allowing communication over distances up to 1200 meters.

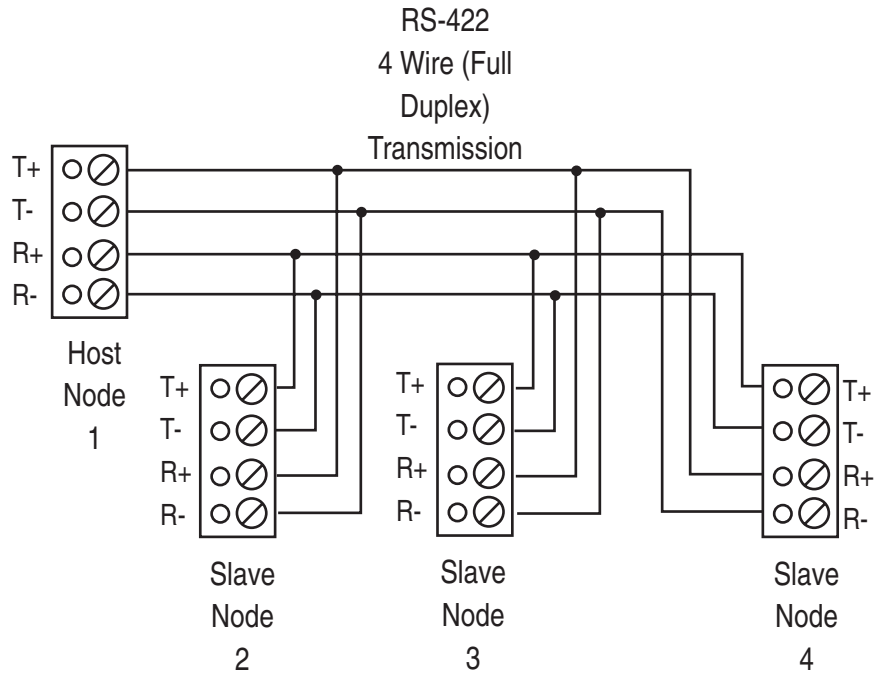
Since RS-422 uses two pairs, it can operate in full duplex (transmit and receive at the same time). For a two-node system, there is no distinction between host and remote. If more than two nodes are connected to a network, one application is designated as the “host” node and the rest are “remote” nodes. There can be as many as 32 nodes without any bus repeaters in the network. A “host” is referred to as the node that initiates communication; while a “remote” is referred to as a node that is addressed by the host.

The host is responsible for initiating communication, maintaining network registration, and providing housekeeping tasks with other nodes. Remotes, however, cannot initiate a communication. They can only respond to messages that are addressed to them from the host.

The devices at each end of an RS-422 network must be terminated. Any node located between the end points should not be terminated. If the 6225 is at one end of the network, use jumper W1[3-4] to make sure that both ends of the network are terminated.

Figure 5-4 shows a typical RS-422 network.

Figure 5-4 RS-422 network



Optomux mode

When COM2 is configured for RS-422/485, jumper W6[5-6] selects between normal RS-422/485 protocol mode and Networked Optomux-compatible RS-422/485 protocol mode. The jumper distinguishes the COM2 mode on powerup. Normal mode ("N" position at W6 jumpered, default) configures COM2 as a standard RS-422/485 port. Optomux-compatible mode ("N" position at W6 not jumpered) configures COM2 to communicate at 38.4 KB, respond to Optomux type commands, respond to Octagon commands, and automatically perform network housekeeping functions. If you are not using Optomux-compatible mode the jumper should be installed, since it uses additional system resources.

Table 5-5 COM2/4 mode select

"N" position at W6[5-6]	Mode	Description
Jumpered	Normal mode	Use default configuration for COM2 of 9600 baud, 8 data bits, 1 stop bit, and no parity.
Not jumpered	Optomux-compatible mode, RS-422/485 INT 17 enabled	Install IRQ3 vector. Change default baud rate to 38.4 KB. Read serial EEPROM bytes 120h,121h to determine if an ID has been previously established. If it has, use the existing ID, if not, use FFh. Discard messages addressed to other nodes. Selectively respond to messages addressed to node. Await INT 17H BIOS calls to collect messages.

Optomux-compatible mode is an easy-to-use software interface for using COM2/4 as an RS-422/485 port. This software interface supports Optomux-type message passing as well as additional Octagon messages. Up to 32 nodes are supported at a default baud rate of 38.4 KB. This built-in feature provides a simple, low cost, and effective method of rapidly implementing an RS-422/485 network.

This mode configures COM2 to communicate at 38.4 KB, respond to Optomux type commands, respond to Octagon commands and automatically perform network housekeeping functions.

Up to 32 nodes with valid ID range from 0 to FF are supported. ID 0 is reserved for the host. ID FF is reserved for a new node that connects to the network and has not yet been assigned an ID.

Both input and output ring buffer size is 2 KB.

While there can be many remotes, all remotes initially respond to FFh as the ID before the ID is assigned or recognized by the host. To avoid conflict, only one new node at a time shall be added to the network.

Other unregistered nodes must not be powered up while the new node is being registered. This allows assigning each node a unique node ID. Once a node has been added to the network and its ID stored in serial EEPROM, any node can thereafter attach to the network in any powerup sequence. Periodically, the host shall try to communicate to all the existing nodes in the network, plus the potential new node with the ID FFh.

Refer to Octagon application notes *AN-0047*, *AN-0048*, and *AN-0049* for additional information in setting up an RS-485 network. Contact Octagon Systems Technical Support, Customer Service, or Octagon's web site at www.octagonsystems.com for this information.

RS-422/485 support functions

This section provides definitions for the following INT 17h BIOS routine functions pertinent to RS-422/485 support functions.

- Initialization
- Send message
- Receive message
- Get receiver status
- Get transmitter status
- Get ID
- Set ID
- Get incoming message buffer pointer
- Get outgoing message buffer pointer
- Set incoming message buffer pointer
- Set outgoing message buffer pointer
- Set roll call response
- Set state wish response
- Set internal state response

Function 00 – Initialization

On entry:	AH = 0FAh	= RS-485 function signature
	AL = 00h	= RS-485 sub-function
	DX = 0ffffh	
On exit:	AL = status	= 0 -> ok
		= Not 0 -> error

Function 01 – Send message

On entry: AH = 0FAh = RS-485 function signature
 AL = 01h = RS-485 sub-function
 DX = 0ffffh
 ES:BX = Transmit buffer pointer
On exit: AL = status = 0 -> ok
 = Not 0 -> error

Function 02 – Receive message

On entry: AH = 0FAh = RS-485 function signature
 AL = 02h = RS-485 sub-function
 DX = 0ffffh
 ES:BX = Receive buffer pointer
On exit: AL = status = 0 -> message available
 = Not 0 -> message not collected yet

Function 05 – Get receiver status

On entry: AH = 0FAh = RS-485 function signature
 AL = 05h = RS-485 sub-function
 DX = 0ffffh
On exit: AX = incoming message status
 BX = incoming message count

Function 06 – Get transmitter status

On entry: AH = 0FAh = RS-485 function signature
 AL = 06h = RS-485 sub-function
 DX = 0ffffh
On exit: AX = outgoing message status
 BX = outgoing message count

Function 07 – Get ID

On entry: AH = 0FAh = RS-485 function signature
 AL = 07h = RS-485 sub-function
 DX = 0ffffh
On exit: AX = our current ID

Function 08 – Set ID

On entry: AH = 0FAh = RS-485 function signature
 AL = 08h = RS-485 sub-function
 BX = desirable ID
 DX = 0ffffh
On exit: AL = status = 0 -> ok
 =Not 0 -> error

Function 09 – Get incoming message buffer pointer

On entry: AH = 0FAh = RS-485 function signature
 AL = 09h = RS-485 sub-function
 DX = 0ffffh
On exit: ES:BX = incoming message buffer pointer

Function 0a – Get outgoing message buffer pointer

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Ah = RS-485 sub-function
 DX = 0ffffh
On exit: ES:BX = outgoing message buffer pointer

Function 0b – Set incoming message buffer pointer

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Bh = RS-485 sub-function
 DX = 0ffffh
 ES:BX = incoming message buffer pointer
On exit: AL = status = 0 -> ok
 = Not -> error

Function 0c – Set outgoing message buffer pointer

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Ch = RS-485 sub-function
 DX = 0ffffh
 ES:BX = outgoing message buffer pointer
On exit: AL = status = 0 -> ok
 = Not -> error

Function 0d – Set roll call response

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Dh = RS-485 sub-function
 DX = 0ffffh
 BX = roll call response code
 (See roll call reply message format)
On exit: AL = status = 0 -> ok
 = Not -> error

Function 0e – Set state wish response

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Eh = RS-485 sub-function
 DX = 0ffffh
 BX = state wish response code
 (See state wish reply message format)
On exit: AL = status = 0 -> ok
 = Not -> error

Function 0f – Set internal state response

On entry: AH = 0FAh = RS-485 function signature
 AL = 0Fh = RS-485 sub-function
 DX = 0ffffh
 BX = internal state response code
 (See internal state reply message format)

On exit: AL = status = 0 -> ok
 = Not -> error

Octagon's command set

This section provides definitions for the following Octagon RS-485 network commands:

Table 5-6 Definitions list for Octagon RS-485 network commands

From host	From remote
Roll Call (Are you there?) >zzA**.	Reply to Roll Call >00Zxxyy**. xx=my ID yy=status=00->doing fine, no request =01->ask for ID request =02->tell me to disconnect =03->ask for internal state
State Wish (What ID do you want?) >zzB**.	Reply to State Wish >00Yxxzz**. xx=my existing ID zz=new ID I want to be
Assign ID (Your new ID is xx.) >zzCnn**.	Reply to Assign ID >00Xxxzz**. xx=my existing ID zz=my new ID
Report State (Report internal state.) >zzD**.	Reply to Report State >00Wxxyy**. xx=my ID yy=user defined state response
> = start character zz = intended listener's ID field ** = checksum field . = end message character	

Checksum field

The checksum field of Octagon's command set is computed by adding the ID field to the original Optomux type checksum. Note that by providing the checksum in this way, Optomux type equipment treats Octagon's add-on commands as invalid messages while Octagon's equipment gains a unique set of commands.

Examples

Full C code examples are included in the \6225\EXAMPLES directory on the 6225 CPU Card CD ROM. The following examples are for concept only.

Example 1:

The following is a description of how Roll Call is implemented:

Host sends: (Dear node #1, are you there?)

```
>01AA3.
; >           ;message start character
; 01           ;intended listener is ID 01
;  A           ;command code A (roll call)
;   A3         ;checksum = uchar ('0'+ '1'+ 'A'+0x01) = 0xA3
;             ;message end character
```

Remote replies: (Dear 00: 01 is here. I have no special request.)

```
>00Z01007B.
; >           ;message start character
; 00           ;intended listener is ID 00 (host)
;  Z           ;command code Z (reply to roll call)
;   01         ;01 = my ID
;    00        ;00 = status = no special request
;     7B       ;checksum = uchar ('0'+ '0'+ 'Z' + '0'+ '1' + '0'+ '0')
;              = 0x7B
;             ;message end character
```

Host sends: (Dear node #2, are you there?)

```
>02AA5.
; >           ;message start character
; 02           ;intended listener is ID 02
;  A           ;command code A (roll call)
;   A5         ;checksum = uchar ('0'+ '2'+ 'A'+0x02) = 0xA5
;             ;message end character
```

Remote replies: (Dear 00: 02 is here. I have no special request.)

```
>00Z02007C.
; >           ;message start character
; 00           ;intended listener is ID 00 (host)
;  Z           ;command code Z (reply to roll call)
;   02         ;02 = my ID
;    00        ;00 = status = no special request
;     7C       ;checksum = uchar ('0'+ '0'+ 'Z' + '0'+ '2' + '0'+ '0')
;              = 0x7C
;             ;message end character
```

The host continues with the roll call until all nodes have been queried. Not all of the 32 possible nodes need to be queried, only the known nodes. When the last known node has been queried, then the host queries for an unknown node.

Host sends: (Dear FF, are you there?)

```
>FFACC.
; >          ;message start character
; FF         ;intended listener is ID FF
;  A        ;command code A (roll call)
;   CC      ;checksum = uchar ('F'+ 'F'+ 'A'+0xff) = 0xCC
;           ;message end character
```

Remote replies: (Dear 00: FF is here. I have no special request.)

```
>00ZFF00A6.
; >          ;message start character
; 00         ;intended listener is ID 00 (host)
;  Z        ;command code Z (reply to roll call)
;   FF      ;FF = my ID
;   00      ;00 = status = no special request
;   A6      ;checksum = uchar ('0'+ '0'+ 'Z'+ 'F'+ 'F'+ '0'+ '0')
;           ;           = 0xA6
;           ;message end character
```

Example 2:

The following is a description of how an ID can be assigned to a new remote node:

Host sends: (Dear FF: are you there?)

```
>FFACC.
; >          ;message start character
; FF         ;intended listener is ID FF
;  A        ;command code A (roll call)
;   CC      ;checksum = uchar ('F'+ 'F'+ 'A'+0xff) = 0xCC
;           ;message end character
```

Remote replies: (Dear 00: FF is here. I have no special request.)

```
>00ZFF00A6.
; >          ;message start character
; 00         ;intended listener is ID 00 (host)
;  Z        ;command code Z (reply to roll call)
;   FF      ;FF = my ID
;   00      ;00 = status = no special request
;   A6      ;checksum = uchar ('0'+ '0'+ 'Z'+ 'F'+ 'F'+ '0'+ '0')
;           ;           = 0xA6
;           ;message end character
```

Host sends: (Dear FF: your new ID shall be 05.)

```
>FFC0533.  
; >           ;message start character  
; FF          ;intended listener is ID FF  
; C           ;command code C (assign ID)  
; 05          ;new ID  
; 33          ;checksum = uchar ('F'+ 'F'+ 'C'+ '0'+ '5'+ 0xff)  
;             ; = 0x33  
; .           ;message end character
```

Remote replies: (Dear 00: FF acknowledges my new ID to be 05.)

```
>00XFF05A9.  
; >           ;message start character  
; 00          ;intended listener is ID 00 (host)  
; X           ;command code X (reply to assign ID)  
; FF          ;FF = my ID  
; 05          ;05 = my new ID  
; A9          ;checksum = uchar ('0'+ '0'+ 'X'+ 'F'+ 'F'+ '0'+ '5')  
;             ; = 0xA9  
; .           ;message end character
```

Example 3:

The following is a description of how a remote (of ID 02) can notify the host of a state change:

1. The application program sets internal state using INT 17h function 0fh.
2. The application program sets roll call reply status to 03 (ask for Report).
3. Since a remote cannot initiate a communication, it must wait until it is spoken to at this point.

Host sends: (Dear 02: are you there?)

```
>02AA5.  
; >           ;message start character  
; 02          ;intended listener is ID 02  
; A           ;command code A (roll call)  
; A5          ;checksum = uchar ('0'+ '2'+ 'A'+ 0x02) = 0xA5  
; .           ;message end character
```

Remote replies: (Dear 00: 02 is here. Ask for my internal state.)

```
>00Z02031F.  
; >           ;message start character  
; 00           ;intended listener is ID 00 (host)  
; Z           ;command code Z (reply to roll call)  
; 02          ;02 = my ID  
; 03          ;03 = status = Ask for a report of internal state  
; 1F          ;checksum = uchar ('0'+0+'Z'+0+'2'+0+'3')  
              = 0x1F  
; .           ;message end character
```

Host sends: (Dear 02: What is your internal state?)

```
>02DA8.  
; >           ;message start character  
; 02           ;intended listener is ID 02  
; D           ;command code D (Internal state query)  
; A8          ;checksum = uchar ('0'+2+'D'+0x02) = 0xA8  
; .           ;message end character
```

Remote replies: (Dear 00: 02 internal state = 47.)

```
>00W024784.  
; >           ;message start character  
; 00           ;intended listener is ID 00 (host)  
; W           ;command code W (reply to internal state query)  
; 02          ;02 = my ID  
; 47          ;47 = my internal state  
; 84          ;checksum = uchar ('0'+0+'W'+0+'2'+4+'7')  
              = 0x84  
; .           ;message end character
```

The application program interprets the message and then responds accordingly. In this case, state 47 would have been defined in the host application as a specific response from a Remote with a user-defined meaning.

Chapter 6: **LPT1 parallel port, LCD and Keypad**

LPT1 parallel port

LPT1 uses a standard 26-pin connector. It supports the unidirectional standard mode, bidirectional mode, enhanced parallel port (EPP) mode, extended capabilities port (ECP) mode, and floppy drive mode. The default I/O address for LPT1 is 378h, with the default interrupt is IRQ7. You can choose the addresses 278h to 27Bh, or interrupt IRQ5, in the 6225 SETUP utility.

The LPT1 port supports a number of devices including a PC compatible printer, a floppy drive, an LCD display, or a keypad.

Note The digital I/O port uses address 278h and IRQ5. If you are using the digital I/O port, do not select either of those options for LPT1. Also, make sure that the interrupts used for other devices do **not** conflict with the IRQ selected for LPT1 in SETUP.

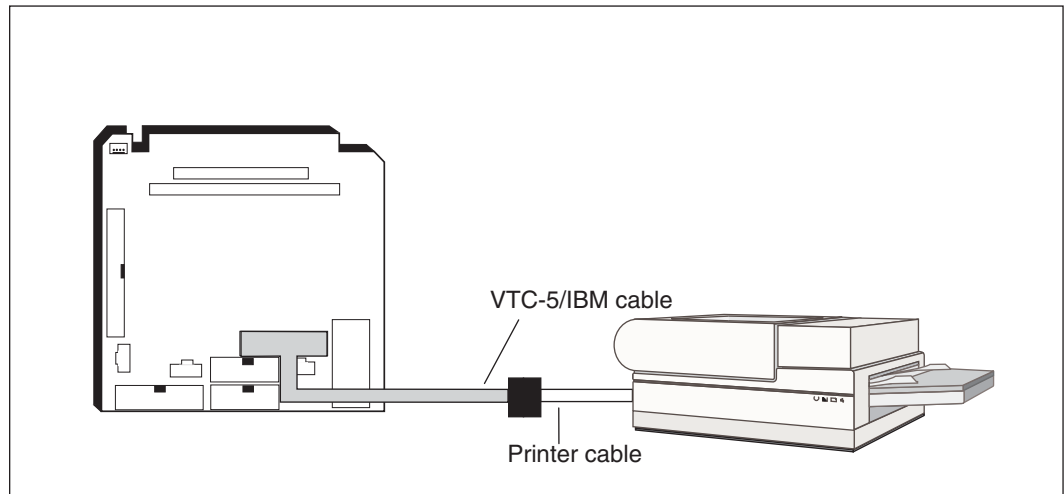
Printer

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

Installing 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 (J4) to the 25-pin connector on your printer cable.
3. Connect the cable to your printer.

Figure 6-1 LPT1 as a printer port



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 6225, use the Octagon 2010 interface board. A CMA-26 cable is required to connect the interface board to the 6225. The program DISPLAY.EXE (found on the 6225 CD ROM) provides an easy method to use the display. Refer to the file DISPLAY.DOC on the 6225 CD ROM for information on initializing and using the display. Also, refer to the *2010 product sheet* for more information on the interface board.

Installing a display

1. Connect a CMA-26 cable from the LPT1 port on the 6225 (J4) to J1 on the 2010. See Figure 6-2.
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.

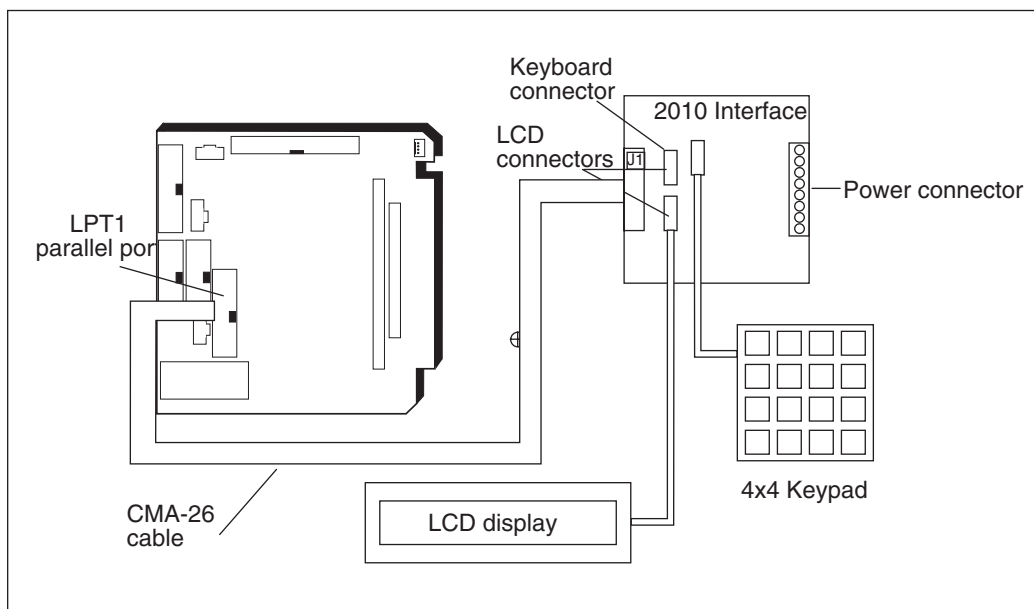
Keypad

LPT1 also supports 4 x 4 matrix keypads. To interface the keypad to the 6225, use the Octagon 2010 interface board. A CMA-26 cable is required to connect the interface board to the 6225. The program DISPLAY.EXE (found on the 6225 CD ROM) provides an easy method to use the keypad. Refer to the file DISPLAY.DOC on the 6225 CD ROM for information on initializing and using the keypad. Also, refer to the *2010 product sheet* for information on the interface board.

Installing a keypad

1. Connect a CMA-26 cable from the LPT1 port on the 6225 (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



LPT1 as a floppy disk drive

The LPT1 parallel port can also be used as a floppy disk drive port. Octagon has a custom cable that connects directly to LPT1 and a floppy disk drive (part# 4809, FCA-12, floppy, 12-inch). The pinout for the cable is shown in table 6-1.

Table 6-1 LPT1/floppy drive cable

#4809, FCA-12, floppy, 12-inch		
LPT1 port pin #	Function	Floppy pin #
1	DS0*	12
2	DenSel	nc
3	Index*	8
4	HDSel*	32
5	Trk0*	26
6	Dir*	18
7	WP*	28
8	Step*	20
9	Rdata*	30
10	Gnd	29
11	DskChg*	34
12	Gnd	31
13	Msen0	nc
14	Gnd	17
15	Mtr0*	16
16	Gnd	19
17	Msen1	nc
18	Gnd	27
18	DS1*	14
20	Gnd	33
21	Mtr1*	10
22	Gnd	21
23	Wdata*	22
24	Gnd	23
25	Wgate*	24
26	+5V	nc
* active low		

To install a floppy disk drive:

1. Make certain that in Setup the LPT1 port is in floppy disk mode.
2. Connect a properly configured cable from the LPT1 port (J4) to the connector on your floppy disk drive cable.
3. Connect the cable to your floppy disk drive.
4. Connect power to the floppy drive using an external cable.

Chapter 7: **Console devices**

Description

The 6225 has four options for console devices. You can use a PC/104 video card with a monitor and a keyboard as your console. You can use an ISA–backplane video card with a monitor and a keyboard as your console (only when the 6225 is mounted on an ISA backplane or in a card cage). 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 6225's four options for console devices:

- A standard PC/104 video card, such as an Octagon 2430 SVGA card, and a keyboard.
- A standard ISA–backplane video card, such as an Octagon 5420 SVGA card, and a keyboard. This option is available only when the 6225 is mounted on an ISA backplane or in a card cage.
- 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.

If you are using a PC/104 or an ISA video card, refer to the instructions included with the card for installation procedures.

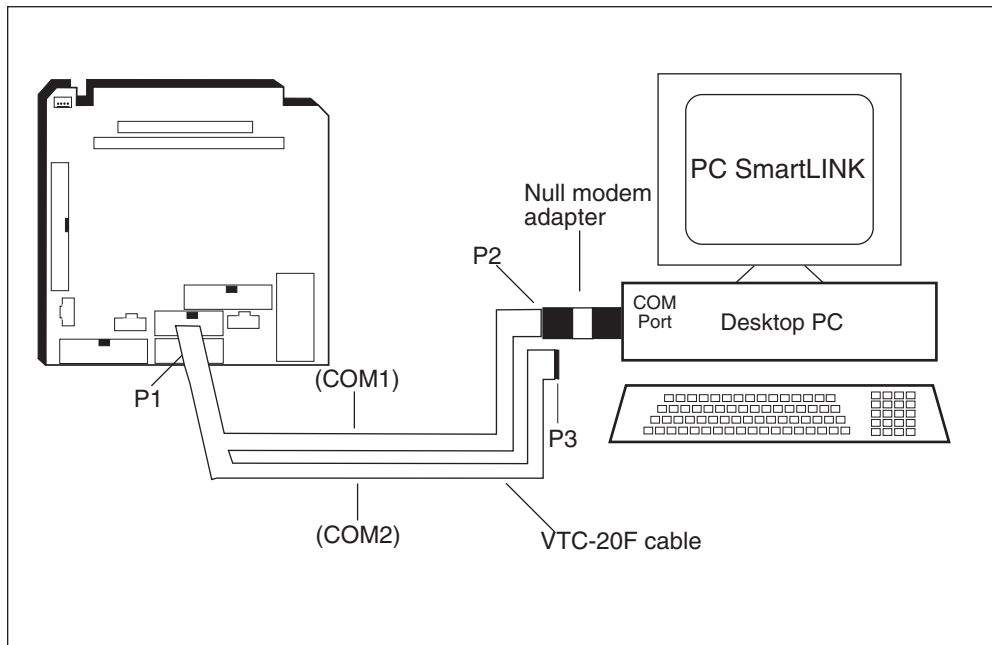
Serial console

If the serial console is enabled COM1 is used as the console device.

Follow these steps to use the serial console:

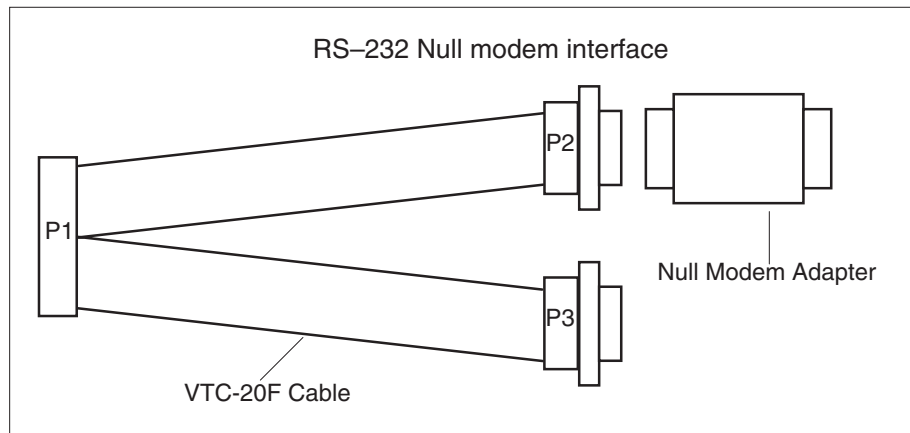
1. Power off the 6225.
2. Connect a VTC-20F to J6 of the 6225. Refer to Figure 7-1.
3. Connect P2 (COM1 side of the VTC-20F cable) to the 9-pin null modem adapter. Refer to Figure 7-2.

Figure 7-1 The 6225 and a serial console



4. If your PC has a 9-pin serial connector, connect the 9-pin null modem adapter to any serial port (COM1–COM4) on your PC.
5. If your PC has a 25-pin serial connector, attach a 9- to 25-pin adapter to the 9-pin null modem adapter, and then insert the matching end of the 9- to 25-pin adapter into the serial port.
6. Start PC SmartLINK.
7. Power on the 6225.

Figure 7-2 VTC-20F cable and null modem adapter



Keyboard

You can use any of the console options mentioned in the *Selecting console devices* section with a local keyboard.

Mouse

If your application requires a mouse, you can connect a serial mouse to any of the COM ports, and load a mouse driver. The 6225 does not require a mouse for operation.

Note See the *Accessories* appendix for mating information on the keyboard, speaker, and mouse connectors.

Chapter 8: **SSDs, DRAM, and battery backup**

Description

The 6225 supports up to two solid state disks (SSDs). SSD0 is a 32-pin DIP socket. It accepts either an M-Systems DOC, 5V flash, 5V EPROM, or 512 KB SRAM.

SSD1 is a 2 MB SMT boot flash which contains the BIOS drive, ROM-DOS 7.1, and a suite of embedded software.

Flash file system software is included which allows the SSDs to emulate hard disk operation.

SSD0 (optional)

The 6225 contains a socket at position U6 for an SSD0 device. SSD0 is an optional feature and is not installed on the standard version of the 6225. You can install a flash, SRAM, M-Systems DOC or EPROM device in the SSD0 socket. Jumper blocks W2 and W3 must be set appropriately. See Table 8-1 for settings.

WARNING!

Incorrect installation of SSD0 will destroy the chip. To install SSD0, be sure to match the notch in the SSD0 chip with the notch in the SSD0 marked silkscreen on the 6225.

Selecting SSD0 device type

Table 8-1 *SSD0 device selection jumpers*

Jumper	SRAM *	Flash	DiskOnChip	EPROM
W2	1-2 ON *	1-3 ON	5-6 ON	1-3 ON
	3-5 ON *	2-4 ON		2-4 ON
	4-6 ON *	5-6 ON		6-8 ON
W3	2-3 ON *	1-2 ON	1-2 ON	1-2 ON

* Default

Use the SETSSD.EXE command to notify the BIOS that a new drive has been installed. SETSSD sets the boot order for the drives, and assigns a drive letter to the drives. After you run SETSSD, reset the system for the changes to take effect.

Flash and SRAM devices must be formatted using the FXFMT.EXE command. After the devices are formatted, reset the system. You are now ready to copy files to the device in SSD0.

For further information on installing and programming SSD0 devices refer to the *Managing SSDs* chapter.

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 SSD0 SRAM and real time calendar clock” in this chapter).

Figure 8-1 SSD socket locations

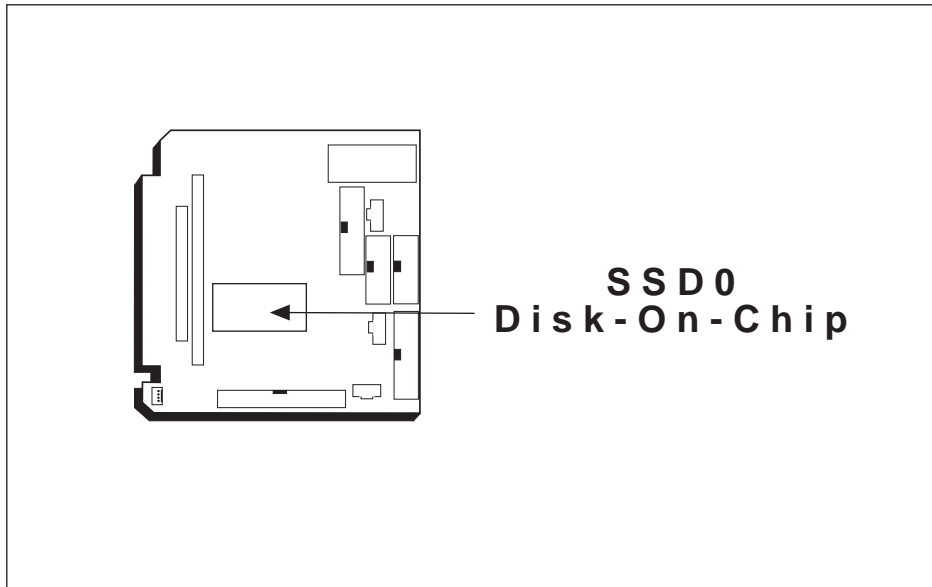
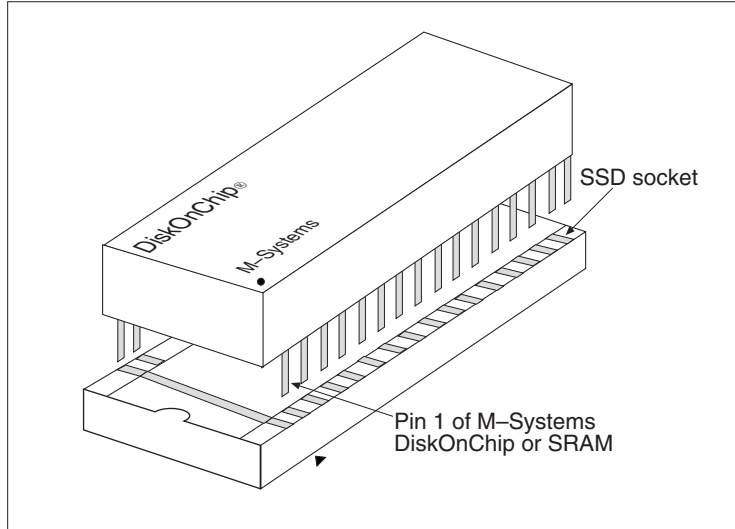


Figure 8-2 M-Systems pin alignment with the 6225 SSD0 socket



WARNING!

Incorrect installation of an SSD0 device will destroy the chip. To install a device in SSD0, be sure to match the notch in the device with the notch in the SSD0 marked silkscreen on the 6225.

Selecting an SSD BIOS

Each SSD may contain a system. To select the SSD BIOS in SSD0, add jumper W6[7-8]. To use SSD1 BIOS, remove jumper W6[7-8]. See Table 8-2 for the proper settings. To program the system BIOS, refer to the *Managing SSDs* chapter.

Table 8-2 Selecting an SSD BIOS: W6[7-8]

State	Result
On*	Use SSD0 BIOS
Off	Use SSD1 BIOS

*Default

SSD1

Generally, SSD1 is meant for storing the application programs to be executed on power-up. The programs are automatically loaded into DOS memory and executed. The 6225 comes with a 2 MB flash soldered on-board. Your application programs can be saved to flash

using the FlashFx driver which makes the flash device a read/write disk on your 6225. Saving your programs onto the read/write disk allows you to do this up to 100,000 times, allowing updates to your application programs. These devices are erased automatically during the programming process.

SSD1 can be accessed directly as a read/write DOS drive with the FlashFx driver. Also, it can be accessed directly as a read/write DOS drive when the FXDOS.SYS driver is loaded. While this is convenient for product development, the flash, however, has a limited number of writes allowed. Therefore, Octagon does not recommend SSD1 be used as a data logging device. Refer to the *Software utilities* appendix for information on supported flash devices and a description of FlashFx.

Battery backup for SSD0 SRAM and real time calendar clock

SSD0 can use 512 KB SRAM by changing W5. Refer to the table below for the correct pinout listing for the SRAM with battery backup configuration. The SSD0 requires an AT battery for battery backup of the SRAM files.

An AT battery also backs up the CMOS real time clock.

Installing an AT battery

1. Power off the 6225.
2. Install the 3.6V AT clock battery at the J10 connector. Refer to the component diagram in the *Quick start* chapter for the location of J10.

Table 8-3 Battery connector: J10

Pin	Function
1	Battery
2	Key
3	Gnd
4	Gnd

Note See the *Accessories* appendix for mating information on the battery connector.

Managing SSDs

FlashFx flash file system

By default the FlashFx extended BIOS is loaded at system boot. A message similar to the following is displayed on your screen as the system boots each FlashFx drive found:

```
PhoenixBIOS(TM) A486 Version x.xx  
Copyright (C) 1985-1994  
Phoenix Technologies, Ltd.
```

The FlashFx extended BIOS allows the 6225 to boot from the flash file system on-board the 6225. These drives are “simulated” hard drives that look like a standard hard drive on a PC.

DataLight’s FlashFx™ includes an extended BIOS (FXBIOS.IMG), a device driver (FXDOS.SYS), a format utility (FXFMT.EXE), and a test utility (EOM.EXE).

Use the device driver FXDOS.SYS to access SSDs when booting from a floppy or hard drive, when the extended BIOS (FXBIOS.IMG) is disabled.

Defining SSDs using SETSSD

The following examples demonstrate how to define SSD drive order:

- To set SSD0 first and SSD1 second, enter the following command:

```
6225 C:\> SETSSD SSD0 SSD1
```

Note SSD0 is C: and SSD1 is D:. Drive designators will change depending on the SETSSD settings.

- To set SSD1 first and SSD0 second, enter the following command:

```
6225 C:\> SETSSD SSD1 SSD0
```

Note In the above example SSD1 is C: and SSD0 is D:. Drive designators will change depending on the SETSSD settings.

- To use only SSD1, enter the following command:

```
6225 C:\> SETSSD SSD1
```

If there are other hard drives on the system, add the **/before** option to place the order of the SSDs before the hard drives, or add the **/after** option to place the SSDs after the hard drives. For example,

- To set SSD0 as the first drive, SSD1 as the second drive, and an IDE drive as the third drive, enter the following command:

```
6225 C:\> SETSSD SSD0 SSD1 /before
```

- To set the IDE drive as first in order, SSD1 as second, and SSD0 as third, enter the following command:

```
6225 C:\> SETSSD SSD1 SSD0 /after
```

In the last example, the IDE drive is C:, SSD1 is D: and SSD0 is E:. Other drive letter designations may be added by device drivers (such as VDISK.SYS), which are in the CONFIG.SYS file on the boot drive.

The boot drive is based upon the drive order set by the SETSSD command and by SETUP's "boot sequence" option. If the boot sequence is set to "A: THEN C:," the system will look for a floppy diskette in drive A:. If a diskette is not installed, or a floppy is not defined, the boot drive will be the first drive specified in the SETSSD command. If the boot sequence is set to "C: ONLY," the check for a diskette is bypassed.

Note The SETSSD parameters may also be overwritten by removing the USESETUP jumper (W6[1-2]) and resetting the system. If the parameters specified at the FlashFx first/second drive prompt are different from the previous SETSSD command, and you answered "No" to the "Save" prompt, the SETSSD output will not be accurate. Therefore, we recommend that you answer "Yes" to the save option to prevent confusion.

Note After you run SETSSD and the drive order has changed, the new parameters will take effect after a reset.

The drive order affects the number entered at the FFXFMT 8n command. See the *Using FFXFMT to format an SSD* section in this chapter.

Using FXFMT to format an SSD

Formatting SSD1

This section describes how to format SSD1.

WARNING!

Reformatting SSD1 requires the use of a floppy or a hard disk to restore system files.

1. Define the SSD order with the SETSSD command. Since the command input varies depending upon the parameters you need to enter, see the SETSSD command in the *Software utilities* appendix.
2. To begin formatting SSD1, execute FXFMT as follows:

```
6225 C:\> FXFMT 8n
```

where n is the hard drive sequence number. This number includes IDE drives and SSDs.

For example, if you have 0 IDE drives and SETSSD shows:

```
[HDDs] SSD0 SSD1
```

then enter:

```
6225 C:\> FXFMT 81
```

On the other hand, if you have 1 IDE drive, enter:

```
6225 C:\> FXFMT 82
```

Note If the drive had not been formatted previously, reset the system before accessing the drive. This allows DOS to recognize the drive and add a letter designation to it.

Note FXFMT.EXE must be downloaded from the 6225 CD ROM. This file is located in the \6225\UTILS directory.

Note FXFMT.EXE will not format IDE/ATA drives.

Using SYS to make an SSD bootable

Adding operating system startup files (using SYS)

To add the system files, issue the following operating systems command:

```
C:\> sys d:
```

where *d*: specifies the drive letter.

For example, if your system has 1 IDE drive, and SETSSD shows “[HDDs] SSD0 SSD1,” then SSD1 should be drive E:. To SYS this drive, use the “SYS *d*.” command.

Note SYS.COM must be downloaded from the 6225 CD ROM. This file is located in the \ROMDOS\v7.1 directory.

Note If you are adding the MS-DOS operating system to SSD1, you must first boot from an MS-DOS bootable device (floppy or hard drive).

Note If you are not booting from ROM-DOS, and wish to SYS ROM-DOS back to the drive, the SYS command requires you to issue the following ROM-DOS commands: COMMAND.COM, ROM-DOS.SYS and SYS.COM.

Changing boot SSD

Autoexecuting your applicatio

This section describes how to autoexecute your application.

1. To autoexecute your application in SSD1, use the SETSSD command to define your SSD as the boot device. Since you need to define the order of SSD1 as the first of the SSDs (and before any IDE drives), enter the following command:

```
6225 C:\>SETSSD SSD1 SSD0 /before
```

2. Add your application to SSD1, and modify the AUTOEXEC.BAT program to include your application.
3. Reset the system. SSD1 is now drive C: and your application should begin execution.

Note If the SETUP option “Boot Sequence” is set to “A: THEN C:”, remove any floppy in drive A: before resetting the system.

Note The SETSSD options are not used when W6[1–2] is not jumpered.

Using SETSSD and TESTOEM to test an SSD

To test SSD0 and SSD1 (i.e., destroying all drive data on the SSD), follow the steps below:

1. Run SETSSD.EXE to determine which socket to use with TESTOEM. The socket (either 0 or 1) depends on the order of the SSDs. The first SETSSD' SSD is socket 0 and the second SSD is socket 1. To configure the order of the SSDs, see the *Defining SSDs using SETSSD* section in this chapter.

For example, running SETSSD with no parameters may give the following output:

```
6225 C:> SETSSD
[FDDs] SSD1 SSD0 [HDDs]
```

In the above example, SSD1 is the first SSD which means TESTOEM is socket 0, and SSD0 is the second SSD which means TESTOEM is socket 1.

2. Execute TESTOEM /Sn where n is the socket determined in step 1. For SSD0 in the above example, enter:

```
TESTOEM /D1 /E
```

For SSD1 in the above example, enter:

```
TESTOEM /D0 /E
```

Making copies of the 6225 SSD

GETIMG.EXE is used to capture an image from an existing FlashFx drive. This is used to program SSD drives on other target systems from one “development” system.

1. To make a copy of the 6225 to a local file, enter:

```
GETIMG SSD1 <filename>
```

2. To make a copy of the 6225 SSD to a remote file:

- a. Establish serial communications between the 6225 and the desktop PC using SmartLINK.

- b. On the 6225 enter:

```
GETIMG SSD1 <filename>/COM1
```

- c. On the desktop PC:, exit SmartLINK by pressing <ALT><X> and then enter:

```
GETIMGH <filename>/COM1
```

- d. Write the image, <filename>, to the host drive.

Programming copies of the 6225 SSD

PGMIMG.EXE is used to program an image from an existing FlashFx drive to other SSDs on target systems.

1. Run GETIMG.EXE to make a copy of the 6225 SSD to a local file or GETIMGH.EXE to make a copy of the 6225 SSD to a remote file. Refer to the above section, *Making copies of the 6225 SSD*.

2. To program a copy of the 6225 image from a local drive, enter

```
d: PGMIMG <filename> SSD1
```

3. To program a copy of the 6225 image to a target system from a host system:

- a. Establish serial communications between the 6225 and the desktop PC using SmartLINK.

- b. On the 6225, enter:

```
PGMIMG SSD1 <filename>/COM1
```

- c. On the desktop PC:, exit SmartLINK by pressing <ALT><X> and then enter:

DRAM

The standard version of the 6225 is shipped with 4 MB of DRAM on-card. The 6225 can also be configured with 8 MB of surface mounted DRAM.

Chapter 9: **External drives**

Description

The 6225 is compatible with all common floppy disk drives used on desktop PCs. The 6225 also drives any standard IDE hard drives that have 16-bit IDE interfaces. The BIOS extension ROM for the hard drive is supplied on the card so that no additional software is needed. The floppy drives use DMA channel 2.

Floppy disk controller

You can add one 1.44 MB floppy drive with the 5815 Disk Drive Card (the 5815 also supports a 2.5" IDE hard drive) or use the oncard multifunctional parallel port at J4. Refer to the *LPT1 parallel port, LCD and Keypad* chapter for information on using LPT1 as a floppy port.

If adding an off-card floppy disk drive, install the 5815 Disk Drive Card. Follow the instructions included with that product.

If adding an on-card floppy disk drive, install the disk drive via the LPT port at J4 on the 5066. In addition, use SETUP to set the LPT mode to "Floppy disk", the number of floppy drives to "1", and set the type (i.e., size) of floppy. You must also route external power for the floppy drive.

Run SETUP to set the number of floppy drives and their size.

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

The 6225 requires +5V for operation. You must also 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

1. Disconnect power to the 6225.
2. Insert one end of your 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 6225.
4. Connect power to the floppy drive.
5. Run SETUP in the \6225\UTILS subdirectory to set up the AT BIOS. You can execute this program either by pressing <BACKSPACE><S> during system bootup or by executing the file SETUP.COM. This file is found on the 6225 BIOS drive and also on the 6225 CD ROM. The system steps you through the configuration. Also, refer to the *SETUP programs* chapter for more information on the AT BIOS SETUP program.

Note If floppy is twisted on the connector, then answer “no” to Swap Floppy A and B.

Hard disk controller

The 6225 will drive up to 4 gigabytes hard drives that have 16-bit IDE interfaces via a 44-pin connector at J12.

Note See the *Accessories* appendix for mating information on the IDE hard connector.

Installing a hard drive:

1. Disconnect power to the 6225.
2. Insert one end of the Octagon hard drive adapter cable into the rear of the hard drive. Make sure pin 1 on the cable is connected to pin 1 on the drive.
3. Insert the other end of the IDE cable into J12 on the 6225. Make sure pin 1 on the cable is connected to pin 1 on the 6225.
4. Execute the BIOS SETUP program to configure your system for a hard drive. You can execute this program either by pressing <BACKSPACE><S> during system bootup or by executing the file SETUP.COM. This file is found on the 6225 BIOS drive and the 6225 CD ROM. The system steps you through the configuration. Also, refer to the *SETUP programs* chapter for more information on the BIOS SETUP program.
5. If you want to boot the system from the hard drive, you need to format

the drive accordingly.

Booting with ROM-DOS

If you want to boot from a floppy/hard drive with ROM-DOS, do the following steps:

1. On the host PC install the CD ROM.
2. Change to \ROMDOS\v7.1 directory.
3. Install a new floppy in drive A:.
4. Execute the following command:

```
sys a:
```

The system copies COMMAND.COM as well as a hidden file to the floppy disk.

5. Insert the floppy disk into a floppy disk drive connected to the 6225.
6. Execute the following command:

```
sys c:
```

The system copies COMMAND.COM as well as a hidden file to the hard disk.

Chapter 10: *Digital I/O*

Description

The 6225 CPU card provides a second LPT port which is configured as a digital I/O port, using the standard printer port register functionality. The I/O address is 278h. There are no options for the digital I/O in SETUP.

The digital I/O port provides 17 lines of digital I/O: eight lines are configurable as a group for input or output, five lines are input only, and four lines are output only.

The digital I/O lines can be connected to an STB-26 Termination Board for easy access for field wiring.

The following table shows the QuickBASIC commands for reading and writing each individual I/O line. For CAMBASIC, delete the "H" after the &, i.e., QuickBASIC &H278 is equal to CAMBASIC &278.

Table 10-1 Digital I/O programming examples

I/O Line	Addr.	Bit	Active		QuickBASIC Input
0 (I/O)	278	0	Low	A=INP(&H278) AND 1	OUT &H278, &HFE
1 (I/O)	278	1	Low	A=INP(&H278) AND 2	OUT &H278, &HFD
2 (I/O)	278	2	Low	A=INP(&H278) AND 4	OUT &H278, &HFB
3 (I/O)	278	3	Low	A=INP(&H278) AND 8	OUT &H278, &HF7
4 (I/O)	278	4	Low	A=INP(&H278) AND 16	OUT &H278, &HEF
5 (I/O)	278	5	Low	A=INP(&H278) AND 32	OUT &H278, &HDF
6 (I/O)	278	6	Low	A=INP(&H278) AND 64	OUT &H278, &HBF
7 (I/O)	278	7	Low	A=INP(&H278) AND 128	OUT &H278, &H7F
8 (O)	27A	0	High	NA	OUT &H27A, 5
9 (O)	27A	1	High	NA	OUT &H27A, 6
10 (O)	27A	2	Low	NA	OUT &H27A, &0
11 (O)	27A	3	High	NA	OUT &H27A, &HC
12 (I)	279	4	Low	A=INP(&H279) AND 16	NA
13 (I)	279	5	Low	A=INP(&H279) AND 32	NA
14 (I)	279	6	Low	A=INP(&H279) AND 64	NA
15 (I)	279	7	High	A=INP(&H279) AND 128	NA
16 (I)	279	3	High	A=INP(&H279) AND 8	NA

Note On powerup, the first eight modules (0-7) default to outputs. If you want to use any of them for inputs, you must write the following commands out to the control register:

```
OUT &H27A, 0
OUT &H279, &HAA
```

If you want to switch the port back to the output mode, you must write the following commands out to the control register:

```
OUT &H27A, 0
OUT &H279, &H0B
```

```
' Demonstration program for Octagon's 6225 CPU Card
' Digital I/O
' Written in QuickBASIC V4.5
' Date: 11-25-92
'
' Note: If you want to run this demo without a video
' card, you can use the PRINTS function which is
' included and explained on the utility disk that comes
' with each Micro PC Control Card. The PRINT
' statement would need to be replaced with PRINTS.
```

' first let's turn on two of the I/O lines

```
OUT &H278, &HFE      'turn on I/O line #0
OUT &H27A, &HC      'turn on output line #11
```

' now let's switch to input mode so we can read an
' input line. Note that the printer must be in bidirectional mode.

```
OUT &27A, &H20      'Sets data lines as input
```

or

```
OUT &27A, &H00      'Sets data lines as output
```

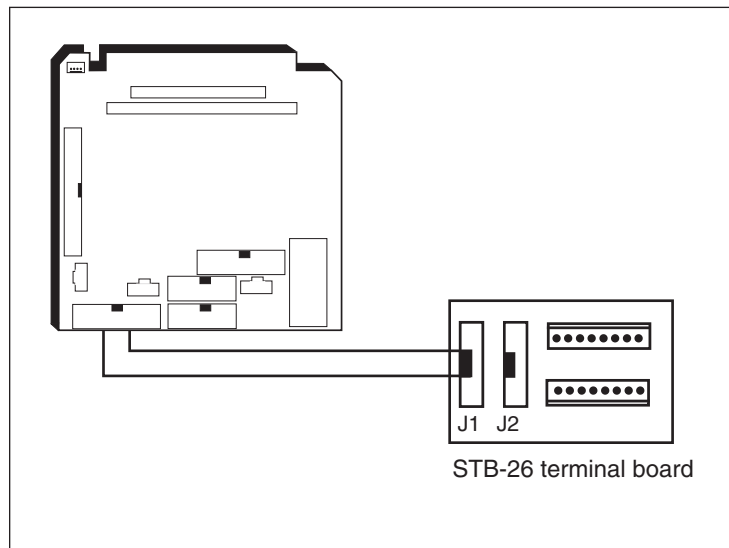
```
DO
    A=INP(&H278) AND 2      'read module #1
    IF A=2 THEN
        PRINT "I/O line #1 is OFF" 'line #1
                                         'active low
    ELSE PRINT "line #1 is ON"
    END IF
    FOR X=0 TO 1000
    NEXT X                    'delay a bit
LOOP
```

The following table shows the pinout for the digital I/O port. The illustration after shows how to connect an STB-26 Terminal Board.

Table 10-2 Digital I/O pinout

I/O Line	Pin	Pin	I/O Line
OUT0	1	2	OUT3
I/O0	3	4	I4
I/O1	5	6	OUT2
I/O2	7	8	OUT1
I/O3	9	10	
I/O4	11	12	
I/O5	13	14	
I/O6	15	16	
I/O7	17	18	
I0	19	20	
I1	21	22	
12	23	24	
13	25	26	VCC5

Figure 10-1 STB-26 Terminal Board



Chapter 11: **Ethernet**

Description

The 6225 provides a 10Base-T Ethernet port and supports the IEEE 802.3 Ethernet standard. The Ethernet controller IC chip provides the following:

- 8K x 16 SRAM buffer
- Integrated 10 Base-T transceiver interface
- Two LEDs for link and traffic status integrated into connector

The 6225 Ethernet uses twisted-pair wiring cable, which is built in a star configuration. The interface terminates at the standard, 8-position, RJ-45 latching phone jack which can be vertically accessed.

For more information on programming the Ethernet port, see the README.DOC on the Ethernet utility disk. By default the Ethernet port connects to IRQ10, but can be reconfigured to IRQ11 via the SETIRQ.DOC program found on the 6225 CD ROM.

Table 11-1 Ethernet LEDs

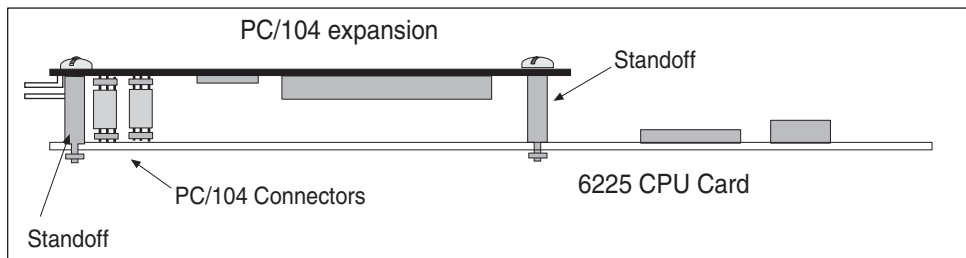
Color	CR7
Amber	Activity LED: Activated by access to I/O space
Green	Link LED: Activated by network link

Chapter 12: *PC/104 expansion*

Description

This connector allows you to interface to one or two PC/104 modules including A/D converters, digital I/O, serial ports, etc. The 6225 supports 8- and 16-bit cards and can provide $\pm 12V$ from the power supply at J11. These modules can be stacked on top of the 6225 to form a highly integrated control system.

Figure 13-1 Typical PC/104 module stack



WARNING!

When installing any PC/104 module, avoid excessively flexing the 6225 card. 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 6225 in the areas of internal control and troubleshooting. The following chapters are included:

- Chapter 13: Watchdog timer and hardware reset
- Chapter 14: Serial EEPROM
- Chapter 15: CAMBASIC
- Chapter 16: CPU power management
- Chapter 17: Troubleshooting

Chapter 13: Watchdog timer and hardware reset

Description

The watchdog timer is a fail-safe against program crashes or processor lockups. It has a programmable timeout period, ranging from 0.5 seconds to 64 seconds. The initial timeout period can be selected in BIOS setup and automatically initiated on power-up. INT17 software calls, a built-in function on the 6225, are then used to enable and set the timeout, strobe, and disable the watchdog timer from your application. If the timer expires, it performs a hardware reset.

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. It is called out by the 6225 BIOS. Once executed it is active, but it must be executed each time the system is rebooted. If you use a different BIOS 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 BX timeout (0=0.5 seconds, 1=2, 2=4, ..., 7=64) 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 the timeout period 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 the timeout period 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
}
```

Hardware reset

The 6225 has a button which allows you to reset the system without turning off the power. This provides a more complete reset than the <CTRL><ALT> method. The RESET command also accomplishes the same thing as the reset button. Refer to the component diagram in the *Quick start* chapter for the location of the reset button.

WARNING!

When using COM1 as the console, the <CTRL><ALT> commands on the host system keyboard only reset the host system. Use the RESET command to issue a hardware reset on the 6225.

Chapter 14: **Serial EEPROM**

Description

Up to 512 bytes 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 calendar/clock provides the user with 242 bytes of user–defined CMOS RAM. This RAM requires battery backup to maintain data. If a battery is not desirable, this data can be stored in serial EEPROM, written to CMOS RAM on powerup, changed and written back to serial EEPROM.

Enhanced INT 17h function definitions

This section provides definitions for both serial EEPROM and CMOS RAM functions. 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.

The CMOS RAM definitions include the following functions: Read extended CMOS RAM, Write extended CMOS RAM, Check CMOS battery, Copy contents of serial EEPROM to extended CMOS RAM, and Copy contents of extended CMOS RAM to serial EEPROM.

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
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 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
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
}

```

Chapter 15: **CAMBASIC**

Note Power management should be disabled when using CAMBASIC.

Introduction

All PC/104 CPUs and 6225 CPU Cards are programmable in CAMBASIC—Octagon’s version of an industrial programming language specifically designed for embedded applications. CAMBASIC is a data acquisition and an industrial control language which is easy-to-use, fast, and multitasking. CAMBASIC has the ability to communicate directly to all on-card I/O including digital, analog, timing, and interrupt. This eliminates the need to write hardware drivers which means you can spend your time writing the applications software.

CAMBASIC is optimized for a 32-bit processor and supports 133 QuickBASIC compatible commands. CAMBASIC’s industrial power comes from the 93 additional commands contained in this language.

CAMBASIC’s major strengths include an extensive vocabulary of industrial BASIC commands along with a list of built-in help messages which makes the language self-documenting. Since the language is syntax compatible with languages like Microsoft QuickBASIC, there is no need for professional programmers. If you have any experience in writing BASIC programs, you are already considered an expert in the CAMBASIC language. Even if you have not had any previous programming experience, there are dozens of books that can teach you to program in BASIC.

Major features

Below are the descriptions of the major features found in the CAMBASIC language.

Event Multitasking™

CAMBASIC uses Event Multitasking™—an efficient method that compiles tasks to machine code for fast assembly operation. During the execution of the main program, all tasks are sampled in the background at 160 times per second. Depending upon the command, the CPU type and speed, the foreground program speed ranges from

5000 to 50,000 commands per second. For example, with 32 tasks activated, the background task checking rate translates to approximately 10,000 tasks per second.

CAMBASIC task types

Event Multitasking lets you do a number of system tasks in the background while you execute your program. The following tasks are available in CAMBASIC:

- Calling subroutines every 0.01 to 655 seconds

Note Use ON TICKA and ON TICKB statements to call subroutines.

- 8 counters with an interrupt on a present count
- 8 programmable timed outputs
- 8 interrupts on changes in digital inputs
- Keypad scanning and debounce
- Capturing serial input without slowing or stopping the program
- Printing data without slowing the program.

Easy-to-write programs

CAMBASIC is termed as the “no hassle” embedded language. It only takes four simple steps to write the program and much less time than C.

1. At the C:> prompt, type CAMBASIC.
2. Write your program. CAMBASIC comes with several programming examples.
3. Debug your program.
4. Type SAVE *name*.

Keyboard mode to debug hardware

The immediate keyboard mode used to examine system components and wiring hardware can quicken system debugging. The BIT command can be used to turn on and off motors, relays, etc. Even if C or another language is used for the applications program, CAMBASIC is always in flash and remains a fast and useful tool to validate correct system operation.

Built-in help and error messages

CAMBASIC has built-in help messages for pin-pointing problems. For example, when an error occurs in a user program, the syntax of the command that caused the error is displayed.

Industrial commands

CAMBASIC has more than 93 commands, many of which are tailored to the industrial environment. CAMBASIC can do the following:

- Read switch inputs individually or in groups
- Write to lamps, relays, and opto-isolator modules one at a time or in groups
- Write analog data to motor controllers, linear actuators, and linear indicators
- Read analog data from pressure transducers, RTDs, thermocouples, and strain gauges
- Send position and velocity profile information to smart motion control cards
- Measure elapsed time and frequency and generate frequency outputs.

Downloading programs remotely

Programs in CAMBASIC can be written and changed via the communication serial ports. Even the built-in text editor is designed to operate over a serial link. A system can be reprogrammed thousands of miles away with the addition of a modem or radio link. You can also perform immediate mode commands to exercise equipment and verify operation from any distance.

Network support

CAMBASIC supports RS-485 networking with direct access to the BIOS network kernel in which 32 systems may be networked at the same time. CAMBASIC has several commands to link directly to the BIOS kernel to receive and send operation codes and packets of data.

Compatibility with other programs

Using the CALL or CALL ABSOLUTE commands to access other

programs, CAMBASIC lets you combine it with C, assembly, or any compiled language. Since entry and exit conditions are well defined, few limitations are placed on the called programs.

Industrial extensions of CAMBASIC

In addition to the list of industrial commands, there is also a list of industrial extensions supported by CAMBASIC:

AUTORUN	Autoruns a CAMBASIC program
BCCF	Returns the binary check code (BCC)
BCD	Converts binary to 4–digit BCD
BIN	Converts BCD to binary
BIN\$	Returns the binary representation to string form
BIT	Reads or writes specific bits at I/O addresses
CLEAR DISPLAY	Clears display
CLEAR TIMER	Clears software timer
CONFIG AIN	Selects drivers for specified A/D
CONFIG AOT	Selects drivers for specified D/A
CONFIG BAUD	Configures COM port parameters
CONFIG COM\$	Configures COM port
CONFIG COUNT	Configures software counter
CONFIG DISPLAY	Configures display
CONFIG KEYPAD\$	Configures keypad
CONFIG PIO	Configures 82C55 parallel port
CONFIG TIMER	Configures software timer
COUNT	Returns the count from software counter
CRC	Returns the cyclic redundancy (CRC) of a memory block
DEC	Fast decrement of a numeric variable
DELAY	Introduces delay to program flow
DEV	Returns standard deviation of an array
DINP	Inputs a 16–bit value to I/O address
DISPLAY	Prints to a display
DOSINT	Executes DOS software interrupt
DOUT	Writes to a 16–bit value to I/O address
DPEEK	Read a 16–bit value from memory
DPOKE	Writes a 16–bit value to memory
ERROR	Returns error code or simulates run time error
EXIT DO	Exits DO loop unconditionally
EXIT FOR	Exits FOR loop unconditionally
INC	Fast increment of variable
HALT	Halts processor for lowest power mode
INDENT	Declares the network identification code
KEYPAD\$	Reads the keypad

MAXVAL	Returns maximum value of elements in an array
MEAN	Returns mean value of the elements in an array
NET\$	Returns a string from the RS-485 network
NLIST	Lists without a line number
NLLIST	Lists to the line printer without a line number
ON COUNT	Specifies condition for branching by the state of the software counters
ON ERR	Specifies location for branching when errors occur
ON INP	Specifies condition for branching by the state of an I/O address
ON KEYPAD\$	Specifies location for branching when a key is pressed at the keypad
ON TICK	Specifies condition for branching by the state of the tick timer
POLY	Returns value from a power series in an array
PRINT\$	Writes a string of characters
QUIT	Exits CAMBASIC to DOS
RMS	Returns Root-Mean-Square value of an array
SUM	Sums elements of array
UNNEW	Undoes the NEW command if possible
VARSEG	Returns segment of variable

Easy-to-use programming examples

CAMBASIC comes with a number of programming examples. Below are programming examples present in CAMBASIC.

Example 1

The following exemplifies the program to write display information to Octagon DP Series and LCD Series display:

```

10 CONFIG PIO &100,0,0,0,0,0: `Configure an 8255 at &100 to be
output ports
20 CONFIG DISPLAY &100,0,0: `Use 2x20 fluorescent display with a
hidden cursor at I/O address &100
30 CLEAR DISPLAY
40 DISPLAY (2,0) "OCTAGON SYSTEMS";
50 DISPLAY (2,1) "CORPORATION";

```

Example 2

In the following example, a 16-key keypad is connected to the printer port at I/O address &378. This program returns inputs from the keypad in use.

```

20 CONFIG KEYPAD$ &378,0,8: `Keypad type=0(16-key), port

```

```
address=&378,debounce=80ms
30 ON KEYPAD$ GOSUB..getkey
40 ...idle
50 GOTO..idle
100 ..getkey
110 A$ = KEYPAD$ (0) : `Get key
120 POSITION = KEYPAD$(1) : `Get position of key
130 PRINT "Key pressed is ";A$," Key position is ";
    POSITION
140 IF A$ = "0" THEN ON KEYPAD$ GOSUB : `Disable key
    pad if the key pressed is "0'
150 RETURN KEYPAD$
```

Chapter 16: **CPU power management**

Description

The power management in the 6225 only functions in DOS. 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.

Power management can be enabled in the 6225 SETUP program and fine tuned with the PMISSETUP program. DOS-supplied advanced power management (APM) programs, such as POWER.EXE are also supported. See the 6225 CD ROM for a list of example programs located in the \6225\EXAMPLES directory. For more information on using the SETUP utility, refer to the *Setup programs* chapter. For more information on using the PMISSETUP utility, see the *PMISSETUP* 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. For example, once the IDLE timer is configured by the firmware, the IDLE timer monitors specific activities and enters the halt state after periods of inactivity. These activities are monitored constantly to determine if power management is suitable. If the specific events do not occur, the IDLE timer will eventually expire, which places the system into the

SUSPEND state where devices can be optionally powered down. At the same time, the CPU is in the HALT state to conserve power. While the CPU is in the HALT state, specific events can be monitored to RESUME the system to full power.

In a stand-alone environment (no APM software active), the firmware works in conjunction with the hardware doze timer 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. Whenever this device is not accessed for a specified period, it is powered down to reduce system power consumption. Whenever none of the monitored system devices has been accessed for a specified period of time, the performance of the system is reduced or the CPU is halted altogether to further reduce power consumption.

In a cooperative environment, devices are still controlled by the firmware, but the CPU is never halted 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 IDLE timer is configured to reset by specific events which include the keyboard, video, hard disk, line printer, serial port, floppy disk, and selected IRQ or DRQ activities. When these specific events do not occur, the system enters the SUSPEND mode. In this mode, devices such as the hard disk, floppy disk, parallel port, serial ports, video, super I/O, and peripheral chip are powered down to conserve power.

In addition to powering down these specified devices, the CPU then enters the halt state. The hard disk, parallel port, video, real time clock, IN access, DRQs, GP0 (memory access at E8000h–EFFFFh) and GP1 (I/O access at 3xxh) are specified events that are configured to resume the system from the SUSPEND state.

Device power management

The hard disk is power managed on an individual basis. The firmware configures a hardware timer that is reset each time the device is accessed. If there is no access to the hard disk, the timer will eventually expire and the hard disk will then power down to conserve power. Later, when the hard disk is restored by a triggering event, such as a keystroke, the access SMI is disabled and the timer is restarted.

System power management

At the system level, power management is very similar to the device level management, with a few 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.

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 SUSPEND mode directly. These modes are entered, but that occurs through the APM interface (INT15h) at the request of the APM driver.

SUSPEND mode is the lowest power state that the system can attain while still powered. The system enters the SUSPEND state when the RESUME switch is turned off. All controllable devices are powered down and the CPU halts. In the SUSPEND state, an IRQ event including a timer tick will cause the CPU to resume from halt. If the CPU determines that the only cause of resume was the IRQ event from the timer tick, then the CPU halts again. Otherwise, the CPU resumes from the SUSPEND state. 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. The BIOS can read the actual time from the real time clock and restore the operating system's timer from that value. The time update can be enabled or disabled using the SETUP program. In SETUP, the option initiating the SUSPEND/RESUME with system activities, is available.

Initiating the SUSPEND/RESUME option with system activities

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

SETUP:

Power management:	ENABLED
Time updated after suspend:	ENABLED

2. Create a .PMI file such as TEST.PMI. Include the following commands in the TEST.PMI file:

TEST.PMI file:

pmi enable=Y	Enables the power management
suspend delay=5	SUSPEND after 5 minutes of idling

Note The qualified activities to prevent the system from entering the SUSPEND mode are keyboard, video, hard disk, printer, COM ports, floppy drive, IRQs, and DRQs.

3. Specify the devices to be powered down in SUSPEND mode by entering the commands below:

suspend hdd=Y	Powers down the hard disk
suspend fdd=Y	Powers down the floppy disk
suspend COM2=Y	Powers down COM2

4. Specify the events to resume the system from SUSPEND mode.

keyboard resume=Y	Resumes the system from SUSPEND mode
com1 resume=Y	Resumes the system from SUSPEND mode

Note The qualified activities to resume the system from SUSPEND mode are keyboard, video, hard disk, printer, real time clock, DRQs, COM ports, and IN accesses (interrupts related to input devices).

5. When the real time clock is used to initiate RESUME from SUSPEND, enter:

```
rtc resume=Y                               Resumes the real time clock
                                             from the SUSPEND mode
```

An example program named WAKEIRQ8 is included in the \6225\EXAMPLES directory on the 6225 CD ROM. This program powers the CMOS clock to generate an IRQ8 after a 30-second delay. During the delay, the system halts. After 30 seconds, the system resumes from SUSPEND.

6. Load the .PMI file changes by including the .PMI file on the PMISSETUP command line. PMISSETUP is located in the \6225\UTILS directory:

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

7. Hardware reset the system for the PMISSETUP options to take effect. The system is now under power management and is ready for SUSPEND/RESUME.
8. Once the system enters the SUSPEND mode, specified devices are powered down. Triggering the specified events will cause the system to resume.

Power management configuration

The power management functions can be globally enabled or disabled in CMOS. The 6225 SETUP.COM and PMISSETUP.EXE utilities provide an option for enabling or disabling power management.

Enabling power management

Whichever utility, SETUP or PMISSETUP, 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 PMISSETUP, the power management option in SETUP will now show power management as being enabled.

In 6225 SETUP, the power management options can be enabled or disabled. The fine-tuning of the power management is done through a PMISSETUP with a .PMI file. All supported options are described in the subsequent sections and are represented by the | symbol. For example, “pmi enable = YES | NO” can either mean “pmi enable = YES” or “pmi enable = NO” as the valid settings. In abbreviated

terms, “Y” is also accepted for “YES” and “N” for “NO”.

In 6225 SETUP, the following options are available:

- Power management ENABLED | DISABLED
- Time updated after suspend: ENABLED | DISABLED

In PMISSETUP, the following options are available:

- pmi enable=Y | N Enables/disables the power management
- Time updated after suspend=YES | NO Enables or disables the time updated after suspend

System timers

The system timers define the delays associated with power state transitions in the system. In addition to full on, only one managed system power mode is supported, the SUSPEND mode. It is possible to configure the suspend delay timer so that when the timer expires, the system will enter the SUSPEND mode. There is also an individual timer for hard disk power management. These timer settings are configured in the .PMI file of the PMISSETUP program as follows:

suspend delay=xx	Sets delay time before system SUSPENDs (xx=0–31 minutes)
hdd delay=xx	Sets delay time before hard drive suspends, xx=0–31 minutes

IDLE timer resets

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

keyboard reset idle=Y N	Enables reset of IDLE timer if Keyboard access occurs
video reset idle=Y N	Enables reset of IDLE timer if Video access occurs
HDD reset idle=Y N	Enables reset of IDLE timer if Hard Disk Drive access occurs

IRQ reset idle=Y N	Enables reset of IDLE timer if IRQ access occurs
LPT reset idle=Y N	Enables reset of IDLE timer if LPT access occurs
COM reset idle=Y N	Enables reset of IDLE timer if COM access occurs
FDD reset idle=Y N	Enables reset of IDLE timer if Floppy Disk Drive access occurs
DRQ reset idle=Y N	Enables reset of IDLE timer if DRQ 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
irqnmi reset idle=Y N	Enables reset of IDLE clock if IRQNMI 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

If DRQ reset idle is set to YES, then the following DRQ options are specified.

drq0 reset idle=Y N	Enables reset of IDLE clock if DRQ0 occurs
drq1 reset idle=Y N	Enables reset of IDLE clock if DRQ1 occurs
drq2 reset idle=Y N	Enables reset of IDLE clock if DRQ2 occurs
drq3 reset idle=Y N	Enables reset of IDLE clock if DRQ3 occurs
drq5 reset idle=Y N	Enables reset of IDLE clock if DRQ5 occurs
drq6 reset idle=Y N	Enables reset of IDLE clock if DRQ6 occurs
drq7 reset idle=Y N	Enables reset of IDLE clock if DRQ7 occurs

Suspended devices in SUSPEND mode

Certain devices need to be powered down while the system is in the SUSPEND mode. These devices are configured in the .PMI file of the PMISETUP program.

suspend HDD=Y N	Suspends hard disk drive
suspend FDD=Y N	Suspends floppy disk drive
suspend LPT=Y N	Suspends LPT
suspend VIDEO=Y N	Suspends video
suspend COM1=Y N	Suspends COM1
suspend COM2=Y N	Suspends COM2
suspend COM3=Y N	Suspends COM3
suspend COM4=Y N	Suspends COM4
suspend SIO=Y N	Suspends SIO

Resume conditions

Once the system has entered the SUSPEND mode, certain peripheral activities can be specified to return the system to the full power mode. The access activities are configured in the .PMI file of the PMISETUP program.

VIDEO resume=Y N	Enables resume activities if Video suspend occurs
HDD resume=Y N	Enables resume activities if access to Hard Disk suspend occurs
LPT resume=Y N	Enables resume activities if access to LPT suspend occurs
GP0 resume=Y N	Enables resume activities if access to GP0 suspend occurs
GP1 resume=Y N	Enables resume activities if access to GP1 suspend occurs
RTC resume=Y N	Enables resume activities if access to RTC suspend occurs
DRQ resume=Y N	Enables resume activities if access to DRQ suspend occurs
IN resume=Y N	Enables resume activities if access to IN suspend occurs
COM1 resume=Y N	Enables resume activities if access to COM1 suspend occurs
COM2 resume=Y N	Enables resume activities if access to COM2 suspend occurs
keyboard resume=Y N	Enables resume activities if access to Key board Display suspend occurs

If DRQ access resume is set to YES, then the following options are available:

drq0 event=Y N	Enables resume from SUSPEND mode if DRQ0 occurs
drq1 event=Y N	Enables resume from SUSPEND mode if DRQ1 occurs
drq2 event=Y N	Enables resume from SUSPEND mode if DRQ2 occurs
drq3 event=Y N	Enables resume from SUSPEND mode if DRQ3 occurs
drq5 event=Y N	Enables resume from SUSPEND mode if DRQ5 occurs
drq6 event=Y N	Enables resume from SUSPEND mode if DRQ6 occurs
drq7 event=Y N	Enables resume from SUSPEND mode if DRQ7 occurs
IN includes irq3=Y N	Resume from SUSPEND mode if IRQ3 occurs and IN resume is enabled
IN includes irq4=Y N	Resume from SUSPEND mode if IRQ4 occurs and IN resume is enabled
IN includes irq12=Y N	Resume from SUSPEND mode if IRQ12 occurs and IN resume is enabled

Chapter 17: **Troubleshooting**

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

No screen activity—checking console serial communications

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

- Make sure all PC/104 expansion cards are removed from the 6225 CPU Card. This ensures that other cards are not interacting with the 6225 CPU Card.
- Remove the jumper from the “S” position at W6[1–2].
- The VTC–20F serial cable turns the 6225 CPU Card serial port into a 9-pin AT serial port. Make sure a null modem adapter is installed on the other end next to the colored stripe, and that the assembly is inserted into the proper serial port on the PC. Make sure the VTC–20F serial cable is connected to COM1 on the 6225 CPU Card.
- Make sure your power module provides +5V (+/–0.20V) and at least 2.5A of current.
- After verifying the above conditions, you can monitor voltage levels by connecting an oscilloscope between the TxD* line on COM1 and ground. After powerup, you should see a burst of activity on the oscilloscope screen. The voltage level should switch between +/- 8V.

Garbled console screen activity

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

- Remove the jumper from the “S” position at W6[1–2] to force 9600, N, 8, 1 for COM1.
- 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.
- 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 6225 CPU Card: 9600 baud, 8 bits, 1 stop bit, and no parity.

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

1. Remove the jumper from the “S” position at W6[1–2] and reboot. When FlashFx prompts you, select SSD1 as the first drive and SSD0 or floppy drive (if installed) as the second drive.
2. Display the directory of SSD1 and verify that all the necessary boot files exist. Copy any missing files to SSD1.
3. If no files are missing, remake SSD1 to overwrite any files which may have become corrupted. In addition, you may want to do a **FXFMT** and **SYS** to SSD1.

Note FXFMT requires a floppy disk drive to restore system files.

FlashFx reports a drive, but issuing a DIR generates an error message

1. The SSD may not be formatted. Run either of the following:

```
FXFMT 8n
```

or

```
FXFMT 8n /m
```

where *n* represents the hard drive number.

For more information, see the *Save and run programs* chapter.

FlashFx does not report the drive

1. Run SETSSD and make sure it is correct.
2. Make sure that the “X” position at W6[3–4] is jumpered or that FXDOS.SYS is in your CONFIG.SYS file of your floppy hard drive.
3. Install a jumper on the “S” position at W6[1–2].
4. Reboot your system.

System locks up on powerup; may or may not respond to reset switch

A common cause is using a non-Octagon power supply such as a PC desktop supply. Most of these PC supplies are rated at 5V at 20A or more. Switching supplies usually requires a 20% load to operate properly, that is, 4A or more. Since a typical Micro PC system takes less than 2A, the supply does not regulate properly. Output drift up to 6–7V and/or 7–8 voltage spikes have been reported. If the power supply comes up slowly (that is, longer than 50 ms), the sequencing of ICs on the board may be out of sync, thus, causing the system to lock up.

Octagon supplies are designed to ramp up fast (less than 50 ms), discharge fast on powerdown and to regulate properly under a no load condition.

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.

Octagon supplies are designed to ramp up fast (less than 50 ms), discharge fast on powerdown and to regulate properly under a no load condition.

LED signaling of “beep” codes

Description

The 6225 has a bicolor LED that is used by the BIOS to signal system status.

Immediately after the 6225 powers on, both LEDs are lit and display an amber color. Upon completion of the boot sequence, the amber LED turns off and the green LED remains on.

If a failure occurs during the boot sequence, visual beep codes are displayed to the LEDs. The visual beep codes are defined in the following table.

The bicolor LED also indicates memory suspend status. Upon entering

memory suspension, the green LED turns off and the amber LED turns on. On a resume condition, the amber LED turns off and the green LED turns on.

When the system enters the cool down clocking state, the green LED and the amber LED turn on. When the system exits the cool down clocking state, the amber LED turns off and the green LED remains on.

Table 17-1 Phoenix BIOS beep odes

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 17-1 Phoenix BIOS beep (cont.)

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 17–2 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

Technical assistance

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

Technical Support telephone: 303-4264521

E-mail Technical Support: Support@octagonsystems.com

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

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

Overview: Section 4 – Appendices

Section 4 contains a series of appendices which provides additional information about the 6225.

- Appendix A: Technical data
- Appendix B: Software utilities
- Appendix C: Third party support
- Appendix D: Accessories

Appendix A: 6225 technical data

Technical specifications

CPU

ALi M6117 386SX Embedded Microprocessor

Bus clock

25 MHz, 40 MHz

BIOS

AT compatible with industrial extensions

DRAM

4 MB DRAM soldered on-card, 8MB OEM option.

Floppy drive

Floppy drive support via the LPT1 parallel port at J4 or a 5815 card.

Hard drive

Hard drive BIOS supported using external hard drive controller which allows extended IDE drives larger than 528 MB. The 6225 has on-board hard drive interface at J12.

Solid-state disk 0

Supports an M-Systems DOC, 5V flash, 512 KB SRAM or EPROM

Solid-state disk 1

Supports a 2048 KB flash

ROM-DOS

ROM-DOS 7.1 compatible

Serial I/O

COM1 through COM4 are 16C550 compatible

Digital I/O

The digital I/O port provides 17 lines of digital I/O: eight lines are configurable as a group for input or output, five lines are input only, and four lines are output only.

Parallel port

LPT1 is PC compatible with multifunctional capability

Watchdog timer

Time-out is from 0.5 seconds to 64 seconds, software enabled and strobed. Disabled on powerup and reset. Controls are through built-in, enhanced INT 17h function calls.

Bus mastering

Bus mastering is not supported

Power requirements

5V \pm 0.25V @ 1.0 Amp. maximum

Full 40MHz operation: 480 mA typical

Suspend: 167 mA typical

Environmental specifications

-40° to 85° C when operating at 25 MHz

0° to 60° C when operating at 40 MHz

-55° to 90° C, nonoperating

RH 5% to 95%, noncondensing

Size

4.5 in. x 4.9 in.

Mating connectors

J2 keyboard port, 8-pin header:

Female housing: AMP #104482-3

Crimp terminal: AMP #1-141708-1

J3 and J8 RS-485 ports, 5-pin header:

Female housing: Leoco #2500S050000

Crimp terminal: Leoco #2403TCB00V1 or V2

J4 and J9 LPT, digital I/O ports, 26-pin shrouded header:

Receptacle: AMP #746288-6

Strain relief: AMP #499252-3

J5 PC/104 interface, PC/104 8/16-bit receptacle:

For 8-bit: Samtec #ESQ-132-14-G-D

For 16-bit: Samtec #ESQ-120-14-G-D

J6 and J7 serial ports, 20-pin shrouded header:

Receptacle: AMP #746288-1

Strain relief: AMP #499252-5
 J10 battery, 4-pin in-line connector:
 Housing: DuPont BERG #746288-1
 Crimp to wire pins: DuPont BERG #499252-5
 J11 power, 4-pin in-line connector:
 Female housing: Molex #22-01-3047
 Crimp terminal: Molex #08-52-0123
 J12 IDE hard drive port, 44-position 2 mm x 2 header:
 Receptacle: AMP #111626-0

Maps

Table A-1 6225 DMA map

Channel	Description
Channel 0	Reserved for bus memory refresh
Channel 1	Available/reserved for ECP parallel port
Channel 2	Floppy disk drive
Channel 3	Available
Channel 4	Slave
Channel 5	Available for PC/104
Channel 6	Available for PC/104
Channel 7	Available for PC/104

Table A-2 6225 I/O map

Hex range	Function
000H-0A7H	System I/O functions
0A8H-0AFH	General purpose status registers
0B0H-0FFH	System I/O functions
100H-207H	Off-card I/O space
208H-20AH	System control registers
20BH	Watchdog control
20C	System
2F8H-2FFH	COM2
378H-37BH	Bidirectional parallel port (LPT1)
3F8H-3FFH	COM1

Table A-3 6225 interrupt map

Interrupt	Description
IRQ0	System timer
IRQ1	Keyboard
IRQ2	Unavailable
IRQ3	COM2 serial port
IRQ4	COM1 serial port
IRQ5	LPT A
IRQ6	Floppy disk controller (available and connected to BIRQ6 or FDC)
IRQ7	Available and connected to BIRQ7
IRQ8	RTC alarm
IRQ9	User-configurable (connected to OPTO B)
IRQ10	Available and connected to BIRQ3
IRQ11	BIRQ4
IRQ12	PC/104 connector
IRQ13	Floating point unit
IRQ14	Available and connected to HDC
IRQ15	Power management interrupt

Table A-4 6225 memory map

Address	Description
00000H-9FFFFH	System memory (640 KB base RAM)
A0000H-BFFFFH	Off-card memory (usually reserved for video memory)
C0000H-C7FFFH	Off-card memory (usually reserved for video BIOS)
C8000H-CFFFFH	Shadow enable/disable option in SETUP Off-card memory
D0000H-DFFFFH	Shadow enable/disable option in SETUP Off-card memory
E0000H-E7FFFH	32 KB BIOS extension area, enabled by "x" jumper. Shadow always enabled
E8000H-EFFFFH	32 KB SSD memory paging window Shadow always disabled in this region
F0000H-FFFFFH	64 KB BIOS area Shadow always enabled
10000H-FFFFFH	16 MB addressable extended memory

Jumper settings

Table A-5 6225 jumper settings: W6 System Configuration

Jumper position	Pins	Description
"S"	W6[1-2]*	USESETUP
"X"	W6[3-4]*	BIOS extension enable
"N"	W6[5-6]*	Network mode
"B"	W6[7-8]*	BIOS device
"U"	W6[9-10]*	User jumper

*Default

Table A-6 6225 jumper settings: W1, W4, W5, W6 COM port jumper settings

COM Port	Communication Type	Jumper Settings
COM2	RS-232C*	W1[7-8] Off W5[2-3] On*
	RS-422	W1[1-2] On W1[3-4] On** W1[5-6] Off W1[7-8] On W5[1-2] On W6[5-6] On***
	Normal Mode	W6[5-6] On***
	Optomux Mode	W6[5-6] Off
COM4	RS-485	W1[1-2] Off W1[3-4] On** W1[5-6] On W1[7-8] On W5 Off W6[5-6] On***
	Normal Mode	W6[5-6] On***
	Optomux Mode	W6[5-6] Off
	RS-232C*	W4[2-3] On
	RS-485	W4 Off
*Default ** This jumper terminates the network. If the 6225 is not at an end of the network, leave this jumper off . *** This jumper should remain On unless you specifically want Optomux mode.		

Table A-7 6225 jumper settings: W2, W3 SSD0 device selection jumpers

Jumper	SRAM	Flash	DiskOnChip	EPROM
W2	1-2 ON	1-3 ON	5-6 ON	1-3 ON
	3-5 ON	2-4 ON		2-4 ON
	4-6 ON	5-6 ON		6-8 ON
W3	2-3 ON	1-2 ON	1-2 ON	1-2 ON

Table A-8 6225 jumper settings: Selecting an SSD BIOS: W6[7-8]

State	Result
On	Use SSD0 BIOS
Off	Use SSD1 BIOS

Connector pinouts

Table A-9 COM1/2 and COM3/4 pinout for RS-232 (J6 and J7 connectors)

Port	Pin	RS-232C
COM2/4	20	nc
	19	GND
	18	RI
	17	DTR
	16	CTS
	15	TXD
	14	RTS
	13	RXD
	12	DSR
	11	DCD
COM1/3	10	nc
	9	GND
	8	RI
	7	DTR
	6	CTS
	5	TXD
	4	RTS
	3	RXD
	2	DSR
	1	DCD

Table A-10 COM2 (RS-422/485) and COM4 (RS-485) pinout (J3 and J8 connectors)

Port	Pin	RS-422	RS-485
COM2 J3 Header	5	RX-	DATA-
	4	RX+	DATA+
	3	GND	GND
	2	TX-	
	1	TX+	
COM4 J8 Header	5	Not available	DATA-
	4	available	DATA+
	3	on	GND
	2	COM4	
	1		

Table A-11 6225 keyboard connector pinout: J2

Pin	Function	Pin	Function
7	+5V (fused)	8	NC
5	NC	6	NC
3	KBDAT	4	NC
1	Gnd	2	KBCLK

Table A-12 6225 PC/104 connector pinout: J5

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	IRQ2/9	LA21	IRQ11
5	SD4	NC (-5V)	LA20	IRQ12
6	SD3	DRQ2	LA19	IRQ15
7	SD2	NC (-12V)	LA18	IRQ14
8	SD1	0 WS**	LA17	DACK0*
9	SD0	NC (+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	BUSCLK	—	—
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 Safe	—	—
30	SA1	14 MHz	—	—
31	SA0	Gnd	—	—
32	Gnd	Gnd	—	—

* = active low

** = wait state

Table A-13 6225 power connector pinout: J11

Pin	Function
1	+5 VDC
2	Gnd
3	+12 VDC
4	-12 VDC

Table A-14 Battery connector: J10

Pin	Function
1	Battery
2	Key
3	Gnd
4	Gnd

Table A-15 6225 hard drive connector pinout: J7

Pin	Function	Pin	Function
1	HRST*	23	HLOW*
2	Gnd	24	Gnd
3	HD07	25	HIOR*
4	HD08	26	Gnd
5	HD06	27	HDCHRDY
6	HD09	28	HDALE
7	HD05	29	NC
8	HD10	30	Gnd
9	HD04	31	IRQ14
10	HD11	32	HDCS16*
11	HD03	33	HDA1
12	HD12	34	NC
13	HD02	35	HDA0
14	HD13	36	HDA2
15	HD01	37	HDCS0*
16	HD14	38	HDCS1*
17	HD00	39	HDACT*
18	HD15	40	Gnd
19	Gnd	41	+5V
20	Key	42	+5V
21	NC	43	Gnd
22	Gnd	44	NC

* = active low

Appendix B: *Software utilities*

Introduction

The 6225 CD ROM comes with the utilities listed below. This chapter describes these utilities and their use. Refer to your ROM-DOS manual for information on other utilities that are included from ROM-DOS but are not mentioned in this section

Support commands

- COM1CON.EXE
- FXCHK.EXE
- FXFMT.EXE
- FXINFO.EXE
- FXRDIMG.EXE
- FXREMNT.EXE
- FXWRIMG.EXE
- GETBIOS.EXE
- GETIMG.EXE
- I17HNDLR.EXE
- LPT1CON.COM
- PGMBIOS.EXE
- PGMIMG.EXE
- PMISSETUP.EXE
- REMDISK.EXE
- REMQUIT.COM
- REMSERV.EXE
- RESET.COM
- SCONSOLE.EXE
- SETSSD.EXE
- SETUP.COM
- TESTOEM.EXE
- TRANSFER.EXE

Support device drivers

- HIMEM.SYS
- FXDOS.SYS

- VDISK.SYS

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.

FXCHK.EXE

Purpose

This support command checks the FlashFX flash media for bad spare units.

Syntax

```
FXCHK drive
```

Parameter

drive specifies the drive letter to check.

Remarks

FlashFX usually keeps at least one spare unit so that if a flash device

develops a bad block, the spare unit takes it's place.

FXFMT

Purpose

This support command formats a FlashFX flash disk.

Syntax

```
FXFMT <Drive> { Options }
```

Parameter

<Drive> Specifies the drive to format. This can be either a drive letter ie. C: (if the drive has already been formatted) or drive descriptor ie. 80 if the drive is not yet formatted.

Drive descriptors are of the format 8x where x is the number of the hard drive beginning with 0. For example 80 is the first hard drive, 81 is the second hard drive.

/C Don't confirm before formatting.
/B Force a system reboot and do NOT confirm.
/M Do NOT write an MBR to the media.
/Px[K] Preserves (x) bytes from start of flash.

Default is current format, Max is 960.

/Sx Format with x number of spare units.

Default is current format.

/Tx[KM] Formats (x) bytes of flash.

Default is current format, Max is media size.

/Qx Reserves (x) percent of the media size for cushion.

Default is 2%. Min is 1, Max is 25.

/Vx Place volume label "x" on disk.
/Dx Where (x) is the number of root directory entries.

Default is 240. Min is 16, Max is 512.

Remarks

The previous the contents of the drive are lost.

After reformatting, the drive can be SYSed to become bootable which requires another boot device with an installed O/S (such as a floppy).

FXFMT will not affect IDE hard drives.

FXFMT uses the current SETSSD information. This defines the order and the amount of space reserved for the BIOS. If the amount of space reserved for the BIOS changes, FXFMT must be re-executed.

See also

SETSSD.EXE.

FXINFO.EXE

Purpose

This support command displays information about a FlashFx disk.

Syntax

```
FXINFO drive
```

Parameter

drive specifies the drive letter to show.

Remarks

FlashFX keeps a host of information about each drive.

FXRDIMG.EXE

Purpose

This support command reads an image from a FlashFX disk.

Syntax

```
FXRDIMG <drive> <file[,offset[K|M][,size[K|M]]> ...
```

Parameters

drive specifies the drive letter of the disk to read from.

file specifies the file name of the image to write.

offset specifies the offset into the flash disk to read.

size specifies the size of the file.

Remarks

This command is similar to GETIMG.EXE. This command is included for completeness.

FXREMNT.EXE

Purpose

This support command forces a remount of the FlashFX flash media.

Syntax

```
FXREMNT drive
```

Parameter

drive specifies the drive letter to remount.

Remarks

FlashFX will remount the drive specified. This may be helpful if the drive was just re-programmed using PGMIMG and when a reboot of the system is not wanted.

FXWRIMG.EXE

Purpose

This support command writes an image to a FlashFX disk.

Syntax

```
FXWRIMG <drive> <file[,offset[K|M][,size[K|M]]> ...
```

Parameters

drive specifies the drive letter of disk to write to.

file specifies the file name of the image to read.

offset specifies the offset into the flash disk to start the write.

size specifies the size of the file.

Remarks

This command is generally compatible with PGMIMG.EXE. It is included for completeness.

GETBIOS.EXE

Purpose

This support command stores the BIOS information in a specific file.

Syntax

```
GETBIOS SSD $x$  filename
```

Parameters

- **SSD x** specifies the target SSD that stores the BIOS. The variable x represents a value from 0 to 1.
- *filename* specifies the output file for saving or programming.

GETIMG.EXE

Purpose 1

This support command captures an image of an SSD and places it into

a local file.

Syntax 1

```
GETIMG SSDx filename
```

Purpose 2

This support command captures an image of an SSD and transfers it to a host PC running **GETIMGH**.

Syntax 2

```
GETIMG SSDx /COMx [/Bxx] [/Ixx]
```

Purpose 3

This support command captures an image of an SSD and transfers it to a host PC running **GETIMGH**. A nonstandard serial port I/O address is used and the **IRQ** value must be specified.

Syntax 3

```
GETIMG SSDx /Uxxxx [/Bxx] [/Ixx]
```

Parameters

- **SSDx** specifies the target SSD for file saving or programming. The variable *x* represents a value from 0 to 1.
- *filename* specifies the output file for saving or programming.
- **/COMx** specifies the PC COM port for serial transfer where *x* represents a value from 1 to 4.
- **/Uxxxx** specifies the UART base address to use for serial transfer. The base address, 100–3FF, is in hexadecimal format.
- **/Bxx** specifies baud rate of transfer where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200). The default is 38400 when using a **/COMx** switch.
- **/Ixx** specifies the interrupt to which the UART is connected. The value *x* represents a variable from 3 to 15.

Remarks

GETIMG compatibility:

The following devices share the same image and are interchangeable: Atmel's AT29C040 and AT29C040A; 512K EPROM; and 512K SRAM. Use an EPROM to make the drive read-only.

AMD, Intel, and Sharp flash memory are interchangeable. They are not interchangeable with Atmel devices.

HIMEM.SYS

Purpose

This device driver manages extended memory and the High Memory Area (HMA) in a 286, 386, or greater PS/2 system. HIMEM prevents programs from simultaneously using the same area of memory for two different purposes. HIMEM supports the eXtended Memory Specification (XMS) 2.0. HIMEM is installed as a device driver in CONFIG.SYS.

Syntax

```
DEVICE=[d:] [path] HIMEM.SYS [/machine:n]
```

Remarks

- The HIMEM driver can be used to allow ROM-DOS to run in High Memory.
- HIMEM supports a default of 32 handles.
- HIMEM should not be used with older versions of VDISK. Current versions of VDISK will use XMS memory if it is available.
- HIMEM recognizes PS/2 style machines A20 line control. HIMEM determines whether to use the PS/2 A20 control or the AT A20 control method automatically by calling INT15h, function C0h (get system configuration).
- The automatic detection can be overridden with the "/Machine:n" command line switch. Replacing "n" with 1 designates the PC AT A20 control method. Replacing "n" with 2 designates the PS/2 method.

Example 1

```
DEVICE=HIMEM.SYS
```

The above command installs the XMS device driver. Once this driver is installed, accessing the HMA and Extended Memory (XMS) areas are legal. The Extended Memory area can contain up to 2 GB of memory. Typical systems have 4, 8, or 16 MB XMS memory installed.

Example 2

```
DEVICE=HIMEM.SYS /machine:1
```

This example forces the use of the AT style A20 line control.

The HIMEM driver will fail to load if either the machine does not have memory above the 1 MB boundary or the BIOS does not provide support for it. It will also fail to load if another XMS manager has been previously installed .

I17HNDLR.EXE

Purpose

This support command is an alternate way to use the INT 17h functions when the extended BIOS area is disabled (i.e., the jumper at the “X” position is removed at W1. Also, use this support command to reprogram the extended BIOS area with another BIOS.

Syntax

```
I17HNDLR
```

Remarks

This command is used if the extended BIOS area (D8000–DFFFF) is not used. The I17HNDLR allows the system to use the INT 17h functions.

LPT1CON.COM

Purpose

This support command redirects the video to the LPT1 port.

Syntax

```
LPT1CON
```

Remarks

If you have a keypad and display board and an LCD display connected to the AUX I/O 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.

FXFMT.EXE

Purpose

This support command formats preformatted or unformatted solid state disk drives.

Syntax

```
FXFMT <drive> [/C] [/M] [/V]
```

Parameters

- *drive* can be specified as **C:** through **Z:** for preformatted drives. For preformatted or unformatted drives, specify *drive* as **H0** through **H9**, where 0 through 9 represents the hard drive number.
- **/C** is an optional parameter that specifies no confirmation before formatting.
- **/M** is an optional parameter that specifies FlashFx is to write an MBR. This is required for unformatted drives using non-AMD flash.
- **/V** is an optional parameter that specifies a volume label is to be placed on disk.

Example 1

To place a volume label on disk for preformatted drive D:, enter the following:

```
FXFMT d: /V
```

Example 2

To have FlashFx write an MBR to drive H0, enter the following:

```
FXFMT 80 /M
```

PGMBIOS.EXE

Purpose

This support command programs a new system BIOS into the 6225 CPU Card.

Syntax

```
PGMBIOS [filename | SSDx] SSDy [/?]
```

Parameters

- *filename* specifies the BIOS .DAT file to program into flash.
- **SSD***x* specifies the source SSD for BIOS. The variable *x* represents a value from 0 to 1.
- **SSD***y* specifies the target SSD for BIOS. The variable *y* represents a value from 0 to 1.
- **/?** requests a help menu.

Example 1

To program the BIOSFILE.BIN files into the SSD1 BIOS area, enter:

```
PGMBIOS BIOSFILE.BIN SSD1
```

Example 2

To program the AT BIOS and extended BIOS from SSD0 to SSD1, enter:

PGMIMG.EXE

Purpose 1

This support command programs a local file image to an SSD.

Syntax 1

```
PGMIMG SSDx filename
```

Purpose 2

This support command programs a local file image to an SSD and transfers it to a host PC running **PGMIMGH**.

Syntax 2

```
PGMIMG SSDx /COMx [Bxx] [Ixx]
```

Purpose 3

This support command programs a local file image to an SSD and transfers it to a host PC running **PGMIMGH**. A nonstandard serial port I/O address is used and the IRQ value must be specified.

Syntax 3

```
PGMIMG SSDx /Uxxxx [Bxx] /Ixx
```

Parameters

- *filename* specifies the input file programming.
- **SSDx** specifies the target SSD for image. The variable *x* represents a value from 0 to 1.
- **/COMx** specifies the PC COM port for serial transfer. The variable *x* represents a value from 1 to 4.
- **/Uxxx** specifies the UART base address to use for serial transfer. The base address, 100–3FF is in hexadecimal format.
- **/Bxx** specifies baud rate of transfer where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200). The default

is 38400 when using a /**COM***x* switch. If COM1 is the console, the baud rate defaults to the current console baud rate.

- /**I***xx* specifies the interrupt to which the UART base address is connected. The variable *x* represents a value from 3 to 15.

FXDOS.SYS

Purpose

This device driver is an alternate method of accessing the on-board SSD. When using FXDOS.SYS, it is possible to free up space in address area D8000H–DFFFFH by removing the jumper from the “X” position at W1.

Syntax

```
DEVICE=FXDOS.SYS
```

Remarks

Using the PICO.SYS device driver instead of the driver in the BIOS extension will prevent booting from an SSD.

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

SLOW.PMI, QUICK.PMI in the /EXAMPLES directory are provided on the 6225 CPU Card CD ROM.

Also, see the *PMISETUP* section in the *Setup programs* chapter.

REMDISK.EXE

Purpose

This support command allows access to a disk drive on a remote system via a serial cable and standard PC style (8250 UART) serial port.

Syntax

```
REMDISK [/U] [/?] [/Bxx] [+] [/COMx]
```

Remarks

In a Remote Disk setup, one system, the one that will share its drives, is termed the Server. The other system, the one that will access and use the remote drives, is called the Client. The serial ports on both systems must be connected via a null modem cable. A cabling diagram for a standard cable is shown below.

Figure 21-1 Cabling diagram for a standard cable

DB-9 to DB-9	DB-25 to DB-25	DB-9 to DB-25
2-----3	2-----3	2-----3
3-----2	3-----2	3-----2
7-----8	4-----5	7-----4
8-----7	5-----4	8-----5
5-----5	7-----7	5-----7
6-----4	6-----20	6-----6
4-----6	20-----6	4-----20

Run REMDISK.EXE on the Client system. This program creates a new drive letter for the Client. REMDISK will use the next available system drive letter. For example, if the last assigned drive was D:, REMDISK will create a drive E:. This drive acts in all ways just like any other drive, except for the fact that it requires the serial port to do its job.

REMDISK.EXE can be installed using a **DEVICE=** command in CONFIG.SYS or from the DOS prompt.

Parameters

- **/U** tells REMDISK to unload itself from memory, thereby disabling the new drive letter and freeing the memory occupied by REMDISK. The option can only be used when REMDISK is installed from the DOS command line. A Remote Disk installed via CONFIG.SYS cannot be unloaded.
- **/?** displays a short help screen for the REMDISK program. No other arguments are to be included on the command line when the **/?** is used.
- **/Bxx** specifies the baud rate for transmission where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115k. The default baud rate is 115k.
- **+** is an optional argument which specifies packet style transmission. This is recommended for any baud rates over 19200. The default for this option is to include the **+** for packet transmission.
- **COMx** is an optional argument which selects the communication port. The variable *x* represents a value from 1 to 2. COM1 is the default port.

Note To use the Remote Disk, both the REMDISK and the REMSERV programs must be running on their respective systems. Both programs must use the same baud rate and packet or non-packet

style transmission. It does not matter which program is installed first.

Example 1

To install the REMDISK program from CONFIG.SYS at 19200, on COM1, using packet style transmission, enter the following in CONFIG.SYS and then reboot the system (remember to include the full path to find REMDISK.EXE if not located in the root directory):

```
DEVICE=REMDISK.EXE /B19200 +
```

Example 2

To display a help screen for REMDISK, enter the following at the DOS prompt:

```
REMDISK /?
```

Example 3

To install REMDISK from the DOS prompt or from a Batch file (like AUTOEXEC.BAT) at 9600 baud, without packet style transmission, on COM2, enter the following:

```
REMDISK /B9600 /COM2
```

Example 4

To unload the REMDISK installed from the batch file or the DOS prompt, type:

```
REMDISK /U
```

See also

REMQUIT.COM, REMSERV.EXE

REMQUIT.COM

Purpose

This support command cancels a REMSERV session on a remote system.

Syntax

REMQUIT

Remarks

Once a REMDISK/REMSERV connection is no longer needed, the REMQUIT command is used (on the same CPU running REMDISK) to cancel the REMSERV command. You may also press the ESC key if you have access to a local keyboard to the CPU running REMSERV.

See also

REMSERV.EXE, REMDISK.EXE

REMSERV.EXE

Purpose

This support command makes a single drive at a time on the server system available to the Client. The available drive can be changed at any time by quitting the REMSERV program and then running the program again with a new drive letter.

Syntax

REMSERV.EXE *d*: [/B*xx*] [+] [/COM*x*] [/S]

Parameters

- *d*: represents the letter of the drive that the Server will make available to the Client.
- /B*xx* selects the baud rate for transmission where *xx* can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115k. The default baud rate is 115k.
- + is an optional argument which specifies packet style transmission. This is recommended for any baud rates over 19200. The default for this option is to include the + for packet transmission.
- COM*x* is an optional argument which specifies the communication port. The variable *x* represents a value from 1 to 2. COM1 is the default port.

- **/S** instructs REMSERV to run silently, that is without any screen output.
- **/?** is an unlisted option which is used to print a short help screen for the REMSERV program. If the **/?** is used, the drive letter argument is omitted, for example:

```
REMSERV /?
```

Example 1

To select drive B: as the available Server drive at 115K baud, pack style transmission, using COM1, you would enter the following:

```
REMSERV B:
```

Example 2

To set drive C: as the Server disk at 9600 baud, without packet style transmission, on COM2, you would enter the following:

```
REMSERV C: /B9600 /COM2
```

Note The Server program can be terminated at any time by pressing the <ESC> key. The Client can then no longer access the Server's drive until the REMSERV program is run again.

See also

REMDISK.EXE, REMQUIT.COM

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> reboot of the system

which only restarts the system but not the expansion cards. The RESET button on the 6225 CPU Card also accomplishes the same thing as the RESET command.

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.

SETSSD.EXE

Purpose

This support command sets the arrangement of the SSD FlashFX devices.

Syntax

```
SETSSD [SSDn] [SSDm] [/Before | /After] [/Nossd]  
[/SSDnBIOS-] [/?] [/D]
```

Parameters

SSD_n, **SSD_m** specify which SSDs are enabled and the order in which the SSD will be placed.

/After indicates the SSDs are allocated after any hard drives and the hard drive will be the boot device.

/Before indicates the SSDs are allocated before any hard drives and the first SSD specified will be the boot device.

/NoSSD specifies no SSDs are to be allocated (default if either **/A** or **/B** used and no SSDs are specified) .

/D selects default settings. (**SSD0 SSD1 /Before**)

/SSDnBIOS- reserves BIOS area on SSDn (+ enables)

Note Requires F XFMT after this option is changed.

/WPn+ write protects drive on SSDn (- enables writes)

Remarks

Defines the SSD device order used by the flash file system. If an SSD name is included on the command line, the SSD will be enabled and can therefore be detected and used by FlashFx. If more than one SSD is listed on the command line, the first SSD will be the first drive (usually C:) and the second SSD will be the second (usually D:).

When IDE hard drives are used, the **/BEFORE** and **/AFTER** switches allow the first SSD to appear before or after the hard drives. For example if two IDE drives and one flash device (SSD1) are defined by SETSSD, and the **/BEFORE** switch is used; SSD1 will be drive C: and the two IDE drives will be D: and E:. However if the **/AFTER** switch is used, the first IDE drive will be drive C:, SSD1 will be drive D: and the second IDE drive will be E:.

When other devices such as a DOC are used, the DOC extended BIOS may affect the **/AFTER** and **/BEFORE** and the "C: ONLY" setup option.

The **/SSDnBIOS** switch is used to either reserve or unreserve the BIOS area in an SSD device. When used correctly, the entire SSD area of one of the flash devices can be used for the drive. Two flash devices must be present with one device containing the system BIOS. Otherwise the system will no longer boot.

Warning

Use this option with caution!

For example if no SSD0 part exists and if the BIOS area for SSD1 is disabled (using **/SSD1BIOS-**) and F XFMT is executed against SSD1, the system will no longer boot until a flash part with a preprogrammed BIOS is inserted into SSD0 and the "B" jumper is removed.

/WPn can be used to disable writes to the part via the drivers. PGMIMG can still write to these devices.

The SETSSD device order can be overridden by removing the "S"

jumper. When removed and when the system boots up, prompts will be asked as to which device is first and which is second. Then a prompt asking whether you wish this information to be saved. If the save prompt is answered "Y" the new information overrides the current SETSSD options. If the save prompt is answered "N" the new information is used but not saved. Since the SETSSD command shows the "saved" options not the temporary "working" options, this may lead to confusion. It is therefore recommended to always "save" your options.

The device order is used by the FlashFX extended BIOS as well as the FXFMT and TESTOEM commands. Other commands such as FXCHK, FXINFO, FXRECLM etc. are indirectly affected as the device order is changed. The BIOS reserve option effects the FXFMT, PGMIMG, GETIMG commands.

See also

See the *SETSSD* section in the *Setup programs* chapter.

SETUP.COM

Purpose

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

6225 SETUP UTILITY Vx.x
(c) Phoenix Technologies, Ltd. 1985, 1995

(Press SPACE to CHANGE, ENTER to ACCEPT, ESC to EXIT)

Serial Console for COM1:	ENABLED
COM1 Console Baud Rate:	9600
Power on memory test:	ENABLED
Boot Sequence:	C: ONLY
COM1 Serial Port:	ENABLED
COM2 Serial Port:	ENABLED
Parallel (LPT) Port:	ENABLED
Parallel Port Mode:	Bidirectional Printer Port
Parallel Port Address:	378h
Number of floppy drives:	2
Floppy drive 1 size	3.5", 1.44 MB
Floppy drive 2 size	3.5", 720 KB
Number of hard drives	1
Auto Drive Configuration:	ENABLED
SETUP Entry via Hotkey:	ENABLED
Power Management:	DISABLED
Shadow Video BIOS Area:	DISABLED
Shadow C8000h-CFFFFh:	DISABLED
Shadow D0000h-D7FFFh:	DISABLED
Shadow D8000h-DFFFFh:	DISABLED
Enable SSD0 Scan:	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.
6225 C:\>

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.

TESTOEM.EXE

Purpose

This support command tests the FlashFX flash media.

Syntax

```
TESTOEM </Dx | /A> [/Cs] [/E] [/X] [/R] [/W] [P]
```

Parameter

- /Dx** Replace x with the hard drive number, zero based.
- /A** Tests all drives available in the OEM layer
- /C"s"** Replace s with command line args to send to the OEM layer during initialization
- /E** Intensive write/erase tests to the entire media
- /X** Extensive erase/write/verify tests to the hidden areas
- /R** Test removable media change detection
- /W** Test write protect status detection
- /P** Execute benchmarks on the access routines

Remarks

All data on the flash part drive area will be erased. The current FlashFX drive settings are used. See SETSSD for more details.

TRANSFER.EXE

Purpose

This support command transfers files to or from the 6225 CPU Card over a serial port.

Syntax

```
TRANSFER filepath [/S | /R] [/Bxxxx] [/V] [/COMx] [/IRQx]  
[/Axxx] [/1] [/F]
```

Parameters

- *filepath* specifies the file pathname to send or receive.
- /S specifies to send the file.
- /R specifies to receive the file. This is the default.
- /Bxxxx specifies baud rate of transfer where xxxx can be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115 (115200). The default is 9600 when using a /COMx switch. If COM1 is the console, the baud rate defaults to the current console baud rate.
- /V enables the display of "R" when a block (128 bytes) is received, or "T" when a block is transmitted.

- **/IRQ x** specifies IRQ to use. The default is IRQ4 for COM1 and COM3, and IRQ3 for COM2 and COM4.
- **/A xxx** specifies non-standard serial port address.
- **/1** specifies use 1K xmodem when sending files.
- **/F** disables fifo use.

Note Do not use **/V** when COM1 is the console device.

- **/COM x** specifies the serial port to use. The variable x represents a value from 1 to 4. The default is 1.

Example 1

To send a file named C:\MPC\DEMO\DEMO.EXE on the 6225 CPU Card when using COM1 as the console, enter the following command:

```
TRANSFER D:DEMO.EXE
```

On the remote PC executing SmartLINK, press <ALT> <D>, type C:\MPC\DEMO\DEMO.EXE, and then press <ENTER>.

Example 2

To send a file named D:DEMO.BAS to the file C:\TEST.BAS on the remote PC when using COM1 as the console, enter the following on the 6225 CPU Card:

```
TRANSFER /S D:DEMO.BAS
```

On the remote PC executing SmartLINK, press <ALT> <U>, type C:\TEST.BAS, and then press <ENTER>.

Example 3

To send a file named C:\DEMO\DEMO.EXE from a remote PC to the file named D:\DEMO.EXE on the 6225 CPU Card at 57600 baud with a serial cable from COM2 on the remote PC and COM1 on the 6225 CPU Card, enter the following command on the 6225 CPU Card:

```
TRANSFER D:\DEMO.EXE /R /V /B57600
```

Then enter the following command on the remote PC:

```
TRANSFER C:\DEMO\DEMO.EXE /S /V /COM2 /B57600
```

To receive a file named D:\MYAPP.EXE from the 6225 CPU Card and

name it C:\APPS\MYAPP2.EXE on the remote PC over a serial cable connected to COM1 on both systems at 9600 baud, enter the following command on the 6225 CPU Card:

```
TRANSFER D:\MYAPP.EXE /S
```

Then enter the following command on the remote PC:

```
TRANSFER C:\APPS\MYAPP2.EXE
```

Remarks

The TRANSFER command communicates with other XMODEM compatible file transfer programs.

The serial port on the 6225 CPU Card requires a null modem adapter when connected to a serial port on the remote PC. See the *Serial ports* chapter for more information.

The maximum baud rate is dependent on the processor speeds of the remote PC and the 6225 CPU Card.

The received file size is rounded up to the nearest 128 byte boundary.

See also

REMDISK.EXE, REMSERV.EXE

VDISK.SYS

Purpose

This device driver allows the user to use memory as a disk.

Syntax

```
DEVICE= VDISK [size [secs [dirs]]] [/E] [/NOTIFEXT]
```

Remarks

VDISK allows the user to partition some of the computer's memory as a disk. This disk is called a RAM disk or virtual disk. A RAM disk is much faster than either a floppy or hard disk. The RAM drive can use either standard DOS program memory or extended memory (above 1

MB) for the disk.

Any data on the VDISK is lost when the system power is turned off.

Parameters

- *size* specifies the size of the VDISK in kilo (K) bytes. The default is 64 KB if base memory is selected. Otherwise, if /E is entered, then the size of the VDISK is all available extended memory (size is 64 KB smaller if DOS is loaded HIGH). The memory selected will be allocated from the DOS memory pool, decreasing the amount of memory available for programs, unless the extended memory switch is used.
- *secs* specifies the sector size in bytes. The default is 512 bytes per sector. This value must be 128, 256, 512 or 1024. All other values are not valid and the default of 512 bytes will be used.
- *dirs* specifies the number of root directory entries. The default is 64 directory entries. There may be any number of root directory entries between 2 and 1024. If an odd number is given, it will be rounded up to the nearest multiple of 16, in order to fill the entire sector.
- /E causes VDISK to use extended memory (memory above the 1 MB boundary) instead of DOS program memory for the disk.

Note Interrupts are turned off during the transfer of data from extended memory to conventional memory.

The VDISK increases the resident size of DOS.

Example 1

```
DEVICE=VDISK.SYS
```

The above example builds a 64 KB RAM disk in DOS memory.

Example 2

```
DEVICE=C:\DOS\VDISK.SYS 220 /E
```

This example builds a 220 KB RAM disk in extended memory. The VDISK device driver is loaded from the C: drive and the \DOS directory. VDISK assumes the default 512 byte sector size, and 64 directory entries.

Example 3

```
DEVICE=VDISK.SYS 45 128 18
```

The above example builds a 45 KB RAM disk in DOS memory. There will be 128 byte sectors and 18 root directory entries.

Example 4

```
DEVICE=VDISK.SYS /E
```

On a 2 MB system, the above example allocates a 1024 KB drive in extended memory (960K is allocated if DOS is loaded HIGH).

Appendix C: *Third party support*

Using M-Systems DiskOnChip (DOC)

To use the M-Systems DiskOnChip (DOC) module with your 6225 CPU Card, follow these steps:

1. Confirm the jumpers for SSD0 are set to use flash.
2. Confirm the BIOS device jumper is on.
3. Boot the system, run "SETSSD SSD1 /before"
4. Run SETUP and enable the "Enable SSD0 scan"
5. Power off the system.
6. Install the DOC module. Make sure PIN 1 is installed correctly or damage will occur to the device.

WARNING!

Installing the DOC module into the wrong position on SSD1 of the 6225 CPU Card will permanently damage the DOC module.

7. Boot up the system, the DOC is usually drive D:. The drive letter depends on the extended BIOS contained in the DOC.

The next step depends on the DOC. Some DOCs come unformatted, others are formatted but do not have an O/S installed, others come preformatted and have an O/S pre-installed.

If the DOC has an O/S preloaded, to boot from the device use "SETSSD SSD1 /after", "SETSSD /NOSSDS" or remove the "X" jumper. The system should then boot from the DOC.

If the DOC has been pre-formatted but does not have an O/S installed and you wish to use ROM-DOS, boot from the SSD1 drive and run the SYS.COM command (SYS.COM is contained in the \ROMDOS\V7.1 directory) or you may create a bootable ROMDOS floppy from the \ROMDOS\V7.1 directory with the "SYS A:" command.

Note To install another O/S such as MS-DOS (not provided by Octagon Systems), you must boot from a floppy which has MS-DOS installed and run the SYS command.

If the DOC has not been formatted, use the "FORMAT D:" command

(FORMAT.EXE is in \ROMDOS\V7.1) to format the device. Then reboot the system.

Note Since the drive letter may vary, confirm the DOC is at the correct drive letter before running the FORMAT command. Otherwise you may be required to boot from a floppy to re-install the files on the SSD1 drive.

See the M-Systems documentation with the DOC and also their web site at www.m-sys.com for more instructions.

Booting from the DOC drive

2. Add an operating system to the DOC. For ROM DOS, use the SYS command.
3. Execute:
`SETSSD SSD1 /after`
3. Reset the system.

Appendix D: Accessories

Table D-1 Cables and terminal board

Product	Description	Octagon part number
KYBD	Keyboard adapter cable	5712
VTC-20F	Dual serial cable	4866
Null modem adapter	9-pin to 9-pin	2470
CMA-26-12	12" cable for EZ I/O port	2776
CMA-26-24	24" cable for EZ I/O port	1257
3.5" hard drive adapter	2mm connector to 2.5", 3.5" hard drive	4080
STB-26	Terminal board, 26-position	2905
FCA-12	LPT1/floppy cable	4809

Table D-2 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

Table D-3 Miscellaneous part numbers

Product	Description	Octagon part number
AT battery	Calendar/clock battery backup	3186
CAMBASIC	Multitasking, industrial control programming language	4059

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.