CMF8680 cpuModule™ & CM102 utilityModule™ User's Manual

Real Time Devices, Inc.

"Accessing the Analog World"®

Publication No. 8680-9524

CMF8680 & CM102
User's Manual

REAL TIME DEVICES, INC.
Post Office Box 906
State College, Pennsylvania 16804
Phone: (814) 234-8087
FAX: (814) 234-5218

Published by
Real Time Devices, Inc.
P.O. Box 906
State College, PA 16804

Printed in U.S.A.

Rev. E 9524

Table of Contents

INTRODUCTION

CMF8680 OVERVIEW

The CMF8680 cpuModule™ is a low power, IBM PC/XT compatible single board computer for embedded microcomputer applications. With its small form factor and self-stacking PC/104 compatible bus, this versatile module can be used as a low cost, highly integrated drop-in package, or it can be stacked with other PC/104 modules to form a cubical MS-DOS system of less than 4 inches per side. Add a keyboard or X-Y matrix keypad, CRT or LCD display, hard or floppy disk drive, and +5 volt power supply to create a complete, XT compatible system which functions like a desktop PC. The thousands of programs designed for the IBM PC/XT can be run on your CMF8680 cpuModule.

Key features of the CMF8680 include:

- Chips & Technologies F8680 16-bit, 14 MHz processor
- Ultra compact 3.6 x 3.8 x 0.6 inch PC/104 form factor
- Consumes 1.6W typical @ 14.3 MHz, 1W @ 7.2 MHz from single +5V supply
- Programmable CPU clock rates: 14.3, 7.2, 4.77 & 3.58 MHz
- Sleep modes, suspend mode with hardware or software wakeup, 0.1W typical
- Datalight® ROM-DOS with up to 1 MB bootable Solid State Disk and support software
- RTD enhanced BIOS ensures PC/XT compatibility
- Compatible with MS-DOS & real-time operating systems such as RTXC, AMX & Nucleus
- 2 MB on-board DRAM (16-bit data bus)
- High density floppy controller & 16-bit IDE hard drive interface; optional 1.8 inch IDE drive with up to 85 MB storage capacity on CM104 hard disk utilityModule
- On-board CGA CRT/LCD video interface; VGA CRT/LCD supported by CM106 Super VGA controller utilityModule
- 2 RS-232 serial ports, 1 RS-485 serial port
- AT enhanced bidirectional parallel, XT keyboard & speaker ports
- Optional PCMCIA memory card interface and keypad scanning/with CM102 utilityModule which supports up to 32 MB memory cards and up to 16 x 8 X-Y matrix keypads
- Battery backed real time clock
- Watchdog timer & power monitor
- Self-stacking PC/104 compatible bus for easy expansion & PC/104 mezzanine bus support

Appendix A provides a complete listing of CMF8680 specifications. Figure i-1 shows the interfaces and major components of the CMF8680.

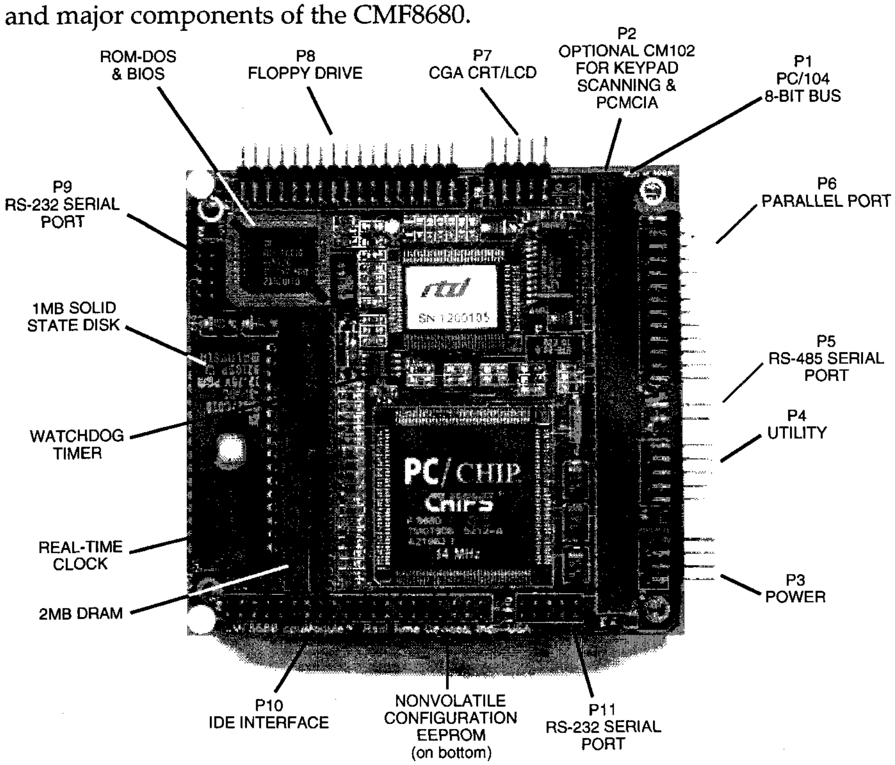


Figure i-1 — CMF8680 cpuModule

C&T F8680 PC/CHIP™

A true single chip PC, the Chips & Technologies' F8680 PC/CHIP™ features 16-bit, 14 MHz XT compatible performance, low power consumption, the SuperState R management system, direct PCMCIA card support, power management, and flexible memory support. Its on-chip CGA controller provides direct support of CRT or LCD panel displays.

The 82C721 super peripheral controller chip adds two serial ports, one parallel port, and up to two high density floppy drives (A: and B:).

Figure i-2 shows a basic block diagram for the F8680. Complete specifications are provided in Appendix A.

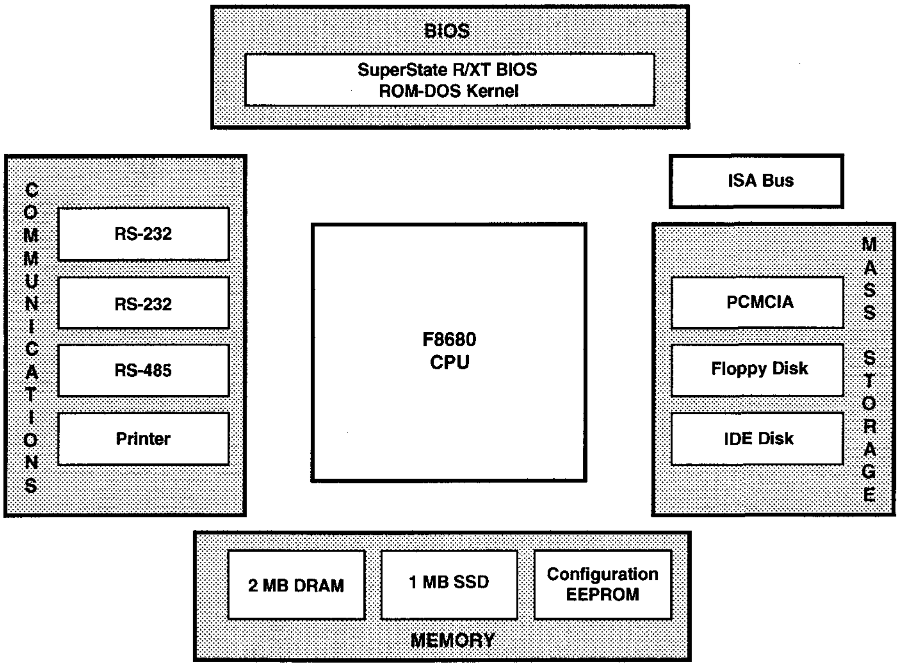


Figure i-2 — F8680 Block Diagram

ABOUT THIS MANUAL

This manual is written for CMF8680 users and system integrators. Here's what you'll find in this manual:

Chapter 1, "Hardware," describes the different types of memory included on your CMF8680, interrupts, DMA, and the interfaces you'll use to attach devices to your CMF8680.

Chapter 2, "Installation," explains how to stack your CMF8680 with other modules and prepare your system for power-up.

Chapter 3, "Start Up," explains the setup software configuration and how you can modify it; system boot options; and system BIOS and SuperState™ R.

Chapter 4, "Power Management," explains the power saving modes of the CMF8680.

Chapter 5, "Using the Solid State Disk," describes how to build a ROMDISK.EXE file for use the 32-bit byte-wide SSD socket.

Chapter 6, "CM102 utilityModule: PCMCIA Memory and Keypad Scanning" describes the PCMCIA interface and X-Y matrix keypad scanning.

Appendix A, "Specifications," summarizes the technical specifications of the CMF8680.

Appendix B, "Memory and I/O Maps," presents the CMF8680 memory and I/O maps.

Appendix C, "Connector Pin Assignments," provides the pinouts for each CMF8680 I/O connector.

Appendix D, "Application Notes," contains a current set of CMF8680 application notes.

Appendix E, "Warranty," details Real Time Devices' warranty.

REFERENCES

These reference books are available for the F8680 PC/CHIP by Chips and Technologies:

F8680 PC/CHIP Data Sheet, Publication Number DS133
F8680 PC/CHIP Programmer's Reference Manual, Publication Number UG75
F8680 PC/CHIP Application Notes, Publication Number AN75

IMPORTANT

If you have trouble using your CMF8680 cpuModule, or if you need more in-depth information for your application, call us Monday through Friday between 8 a.m. and 6 p.m. eastern time at (814) 234-8087 or fax us 24 hours a day at (814) 234-5218 for a prompt reply.

CHAPTER 1

Hardware

This chapter describes the different types of memory supported by the CMF8680, interrupt and DMA channels, and interface connectors. This information is useful for configuring the hardware before installation into a system.

MEMORY DEVICES

2 MB Dynamic RAM — The CMF8680 includes 2 MB of 16-bit wide DRAM installed as shown in Figure 1-1. Four 1M x 4 20-pin ZIP memory devices provide 2 MB of memory to be used as system RAM, shadow RAM for ROM, and 1MB of extended memory which can be configured as expanded memory or a RAM disk.

32 KB Static RAM — This on-board SRAM is used for the CGA graphics display (U3, bottom side of module).

256 KB ROM — This device, shown in Figure 1-1, contains the XT BIOS, IDE hard drive BIOS, Datalight® ROM-DOS, SuperState™ R code, and RTD BIOS enhancements.

4 K bit Configuration EEPROM — This non-voltatile memory stores the system configuration with 2 K bits available for user and keypad layout (U22, bottom side of module).

Solid State Disk EPROM (1 MB Solid State Disk) — The CMF8680 supports up to 1 MB of memory installed in this standard 32-pin, byte-wide JEDEC socket shown in Figure 1-1. Up to 1M x 8, 32-pin memory devices such as non-volatile RAM, PROMs, and flash EPROMs can be installed.

When using the byte-wide socket, you must configure the jumpers on header connectors A through E for the type of device you have installed. Figure 1-2 shows the jumpers and lists the signal carried and pin number on the byte-wide socket. By installing jumpers on A and D, 27C010, 27C020, 27C040, and 27C080 EPROMs are supported.

NOTE: To access a ROM disk, you must enable the ROM-DOS BIOS extension in the RTDMOD setup program, or boot from a floppy or hard disk using the provided Datalight ROM-DOS.

CAUTION: Make sure that jumpers A through E are properly configured for your device. Failure to do so can cause damage to the device.

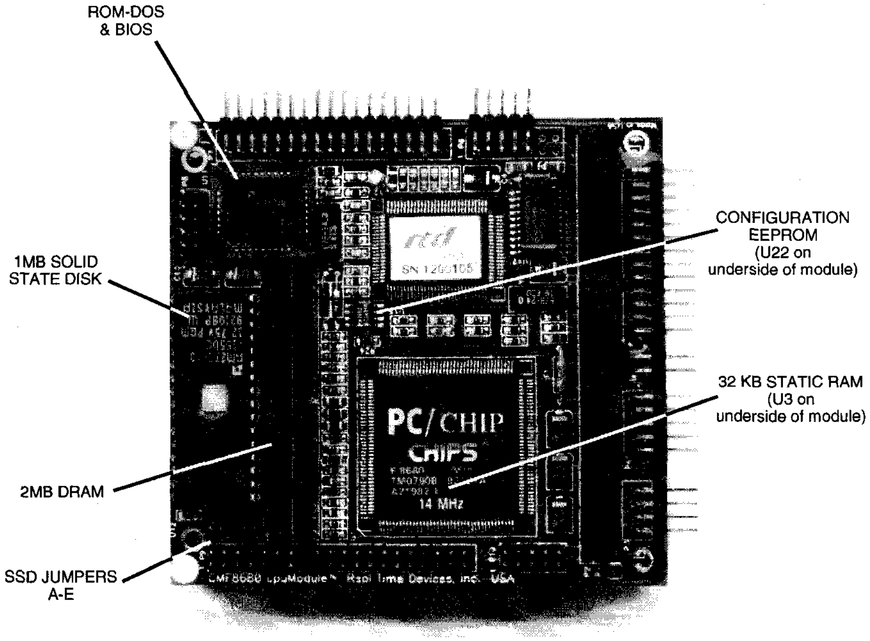
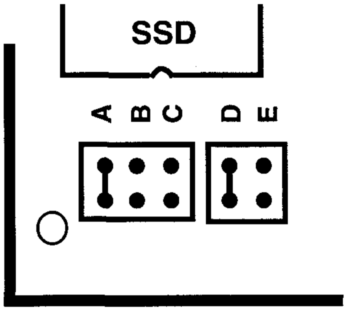

ROM-DOS & BIOS
1MB SOLID STATE DISK
2MB DRAM
SSD JUMPERS A-E
CONFIGURATION EEPROM (U22 on underside of module)
32 KB STATIC RAM (U3 on underside of module)
PC/CHIP
CHIPS
14 MHz
CMF8680 cpuModule™ Real Time Devices, Inc. USA

Figure 1-1 — CMF8680 Memory Devices



Jumper	Signal	SSD Pin
A	+5 volts	1
B	ADR19	1
C	ADR18	1
D	ADR18	31
E	MEMW-	31

NOTE: Installing jumpers on A and D as shown in this diagram supports 27C010, 27C020, 27C040 & 27C080 EPROMs. These jumpers are located in the lower left corner of the board, near the SSD socket.

Figure 1-2 — Byte-Wide Socket Device Jumpers

INTERRUPTS

Eight interrupt channels, IRQ0 through IRQ7, are provided by the PC/104 bus. IRQ0 and IRQ1 are reserved by system functions. IRQ2 through IRQ7 are available to the user for peripheral devices. The list below shows the interrupt channel assignments:

IRQ0: Timer Channel 0 (not available to the user)
IRQ1: Keyboard Interface (not available to the user)

These peripheral devices can be disabled and their interrupts made available for user applications:

IRQ2: Video
IRQ3: COM2, COM4
IRQ4: COM1, COM3
IRQ5: IDE Hard Disk
IRQ6: Floppy Drive Controller
IRQ7: Parallel Port

DMA

Four DMA channels, DMA0 through DMA3, are provided by the PC/104 bus. DMA0 and DMA2 are reserved for system use. DMA1 and DMA3 are available to the user for DMA transfer operations.

INTERFACE CONNECTORS

The CMF8680's versatility is enhanced by the many on-board interfaces provided. These interfaces include a 16-bit IDE hard drive interface, high density floppy controller, PCMCIA interface, CGA CRT/LCD graphics controller, two RS-232 serial ports, one RS-485 serial port, an AT enhanced bidirectional parallel port, XT keyboard and speaker ports (utility connector), and power connector. Figure 1-3 shows these interfaces. The following paragraphs describe each interface. Pin assignments for all connectors are included in Appendix C.

NOTE: The CMF8680 is available with (CMF8680-2) or without (CMF8680-1) the PCMCIA connector, P2, and P1 and/or P2 are available as stackthrough or soldertail connectors.

Connectors P3 through P8, P10, and P11 are normally supplied as right angle connectors, and P9 is normally supplied as a vertical connector. OEMs can specify any combination of right angle and vertical connectors for P3 through P11.

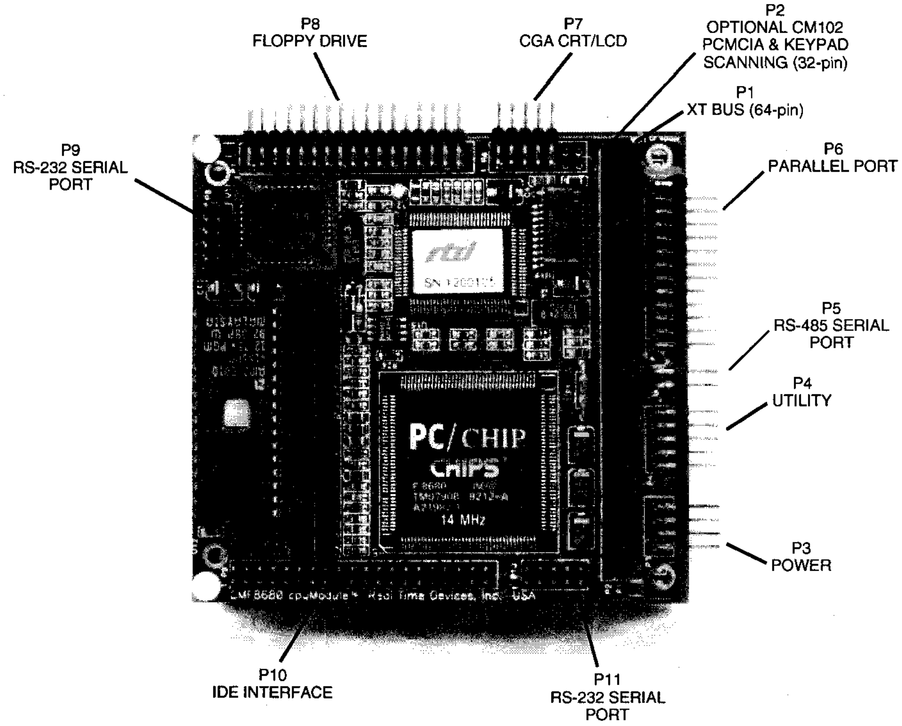


Figure 1-3 — CMF8680 cpuModule Interface Connectors

IDE Hard Drive Interface

The 16-bit IDE hard drive interface, P10, is a standard 40-pin connector. This interface supports 16-bit data transfers in a single cycle. P10 can be connected to the CM104 1.8 inch hard drive utilityModule™ or to any other 16-bit IDE hard drive.

Floppy Drive Interface

The high density floppy drive interface, P8, is a 34-pin connector which supports two high density floppy drives (360K, 720K, 1.2MB, or 1.44MB) configured as disk drives A: and B:. Note that each of the two PCMCIA cards supported by the system are also accessed as disk drives A: and B:. Therefore, the system can support a combined total of two floppy drives or PCMCIA cards.

PCMCIA Interface

The P2 PCMCIA interface, included on the CMF8680-2 module, is a 32-pin connector which mates to the CM102 PCMCIA utilityModule. This 32-pin auxiliary bus stacks directly with the CM102 utilityModule, requiring no additional external cabling. SRAM cards, ROM cards, and EPROM cards which adhere to the PC Memory Card International Association Release 1.0 standard can use this interface. Memory cards of up to 32 MB can be read and written (SRAM only) through this interface.

Keypad Scanning

The CM102 utilityModule adds X-Y matrix keypad scanning which is supported in the CMF8680 BIOS. The keypad layout is user programmable, and the keypad can be used with or without a standard XT keyboard.

CGA CRT/LCD Graphics Controller

The CGA CRT/LCD graphics controller, P7, is a 10-pin connector which interfaces to most popular CGA and LCD flat panel displays. This interface supports 80 x 25 and 40 x 25 text modes, as well as 640 x 200 pixel 2-color, 320 x 200 pixel 4-color, and 640 x 400 double scan graphics modes. When driving a CRT or LCD panel, the CGA graphics controller displays colors or up to 16 shades of grey.

RS-232 Ports

Two 16450 compatible RS-232 serial ports, P11 and P9, provide support for user peripherals such as a mouse or modem. P11 is from the F8680 CPU chip and can be programmed as COM1, COM2, or disabled. P9 is from the 82C721 controller chip and can be programmed as COM1, COM2, COM3, COM4, or disabled. Both serial ports operate at up to 115K baud.

RS-485 Port

The RS-485 serial port, P5, supports multidrop system configurations where several devices are controlled through a single port. It can be programmed as COM1, COM2, COM3, or COM4. When using P5, you must configure the associated jumper on header connector P14 so that the line is terminated or not terminated. The RS-485 path must be terminated at both ends for proper operation. If the CMF8680 is at the end of the path, then install the jumper at P14 to terminate the line. If the CMF8680 is located somewhere in the middle of the RS-485 path, remove the jumper at P14 so that the line is not terminated. Asserting the RTS signal on the serial port enables the RS-485 driver. The RS-485 receiver is always enabled. The RS-485 uses a 16450 compatible UART.

Parallel Port

This port, P6, can be programmed as a PC/AT enhanced bidirectional parallel port or as a standard printer port and can be programmed at address 3BCH, 378H, or 278H, or disabled. The parallel port cable should be less than 10 feet long to ensure reliable data transfer.

Utility Port

The 10-pin utility port, P4, provides XT keyboard signals, piezoelectric speaker signals, system reset, suspend/resume switch signal, sleep mode monitor, and battery backup connections. The pin assignments are given below. The speaker signal can drive an external piezoelectric speaker. When the RESET line is taken low at pin 3, a system reset occurs. Pin 4, PWRUP, is used to place the system in and out of the suspend mode for power conservation. When connected to a slide switch, a high keeps the system in the run mode and a low places it in the suspend mode. When connected to a pushbutton switch, the mode toggles between run and suspend each time the switch is depressed. The switch type is user configurable using RTDMOD.EXE (see Chapter 3, Power Management Setup discussion, for more details). Note that the PWRUP signal must be properly debounced. An XT keyboard with a standard 5-pin DIN connector is connected to pins 5 through 8. A normally open reset switch can be connected between pins 3 and GND. A battery to back up the real-time clock can be connected between pins 9 and GND. Pin 10 is low when the cpuModule is in the suspend mode.

PIN	SIGNAL
1	+SPKR
2	GND
3	RESET SW-
4	PWRUP
5	–KBDATA

PIN	SIGNAL
6	–KBCLK
7	GND
8	KBDPWR
9	+3.6 VDC battery
10	˜SLEEPY

Power Connector

Power is provided to the CMF8680 through the 8-pin power connector, P3. This connector is not used if power is provided through the XT bus. The table below shows the pin assignments of this connector. The CMF8680 requires only +5V and GND. The remaining voltages are supplied to the PC/104 bus.

PIN	SIGNAL
1	GND
2	+5 V
3	N/C
4	+12 V

PIN	SIGNAL
5	–5 V
6	–12 V
7	GND
8	+5 V

PC/104 Expansion Bus

The 64-pin PC/104 XT compatible ISA expansion bus, P1, allows you to directly stack PC/104 compatible modules.

CHAPTER 2

Installation

This chapter describes the installation procedures for integrating the CMF8680 into a PC/104 system.

STACKING MODULES

The CMF8680 can be stacked with other PC/104 form factor modules to form a cubical system, as shown Figure 2-1. When stacking the CMF8680, first make all of your peripheral connections. Be sure to connect pin 1 of the cable to pin 1 of the connector. The connectors are labeled on the board. If you cannot read the label, pin 1 can be identified by its square solder pad, visible on the top and bottom sides of the module.

After all peripheral connections are made, insert the pins of the PC/104 expansion bus of one module into the sockets on the second module, making sure the pins are properly aligned before seating them into the sockets. The order of the modules in the stack is not important as long as all modules in the middle have a stackthrough PC/104 bus connector. After the modules are stacked, insert the spacers provided with each module to secure the stack. Figure 2-2 shows a typical stack. The stack is now ready to install in your system.

NOTE: Since most dataModules do not have the extra 32-pin auxiliary bus connector, we recommend that, if you have the -2 version of the CMF8680 with the optional P@ PCMCIA connector, you place dataModules on top of the CMF8680 and utilityModules under the CMF8680, as shown in Figure 2-2.

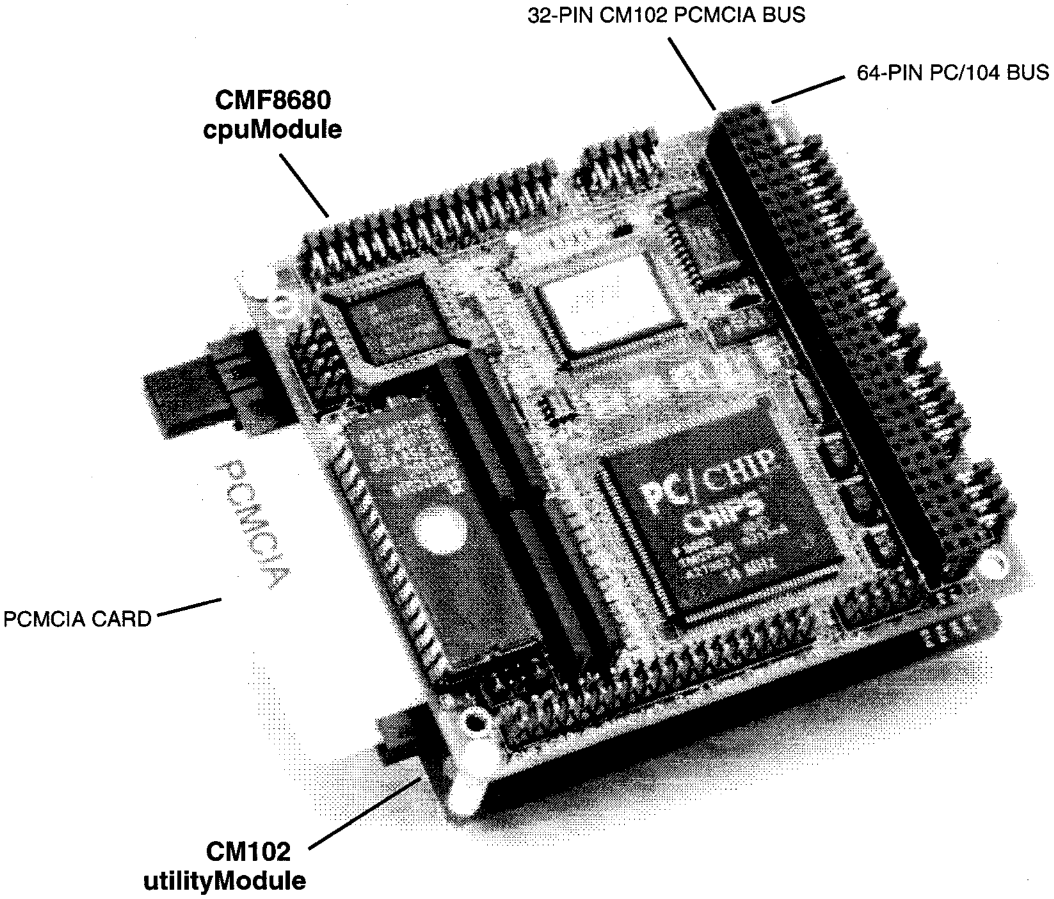
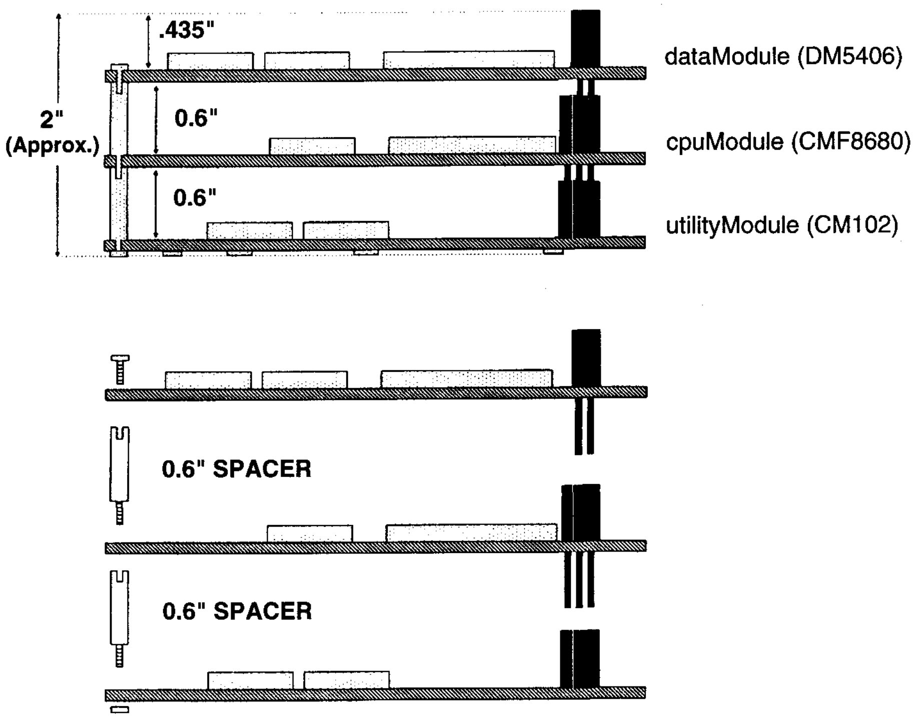


Figure 2-1 — PC/104 Stack



NOTE: When installing a dataModule with only a P1 connector (2 rows of pins), make sure that the 2 rows of pins engage with the 2 rows of pins closest to the edge of the mating module, as shown in the diagram.

Figure 2-2 — Typical Stack Assembly

OTHER OPTIONS

In addition to stacking, modules can be connected to backplanes, other boards with a PC/104 bus connector, or in a number of custom configurations, depending on your application. If you need help with hardware installation, please consult the factory.

CHAPTER 3

Start Up

This chapter tells you how to use the configuration setup program, RTDMOD.EXE, and how to start up your CMF8680. System boot options, the BIOS and SupterState R, and the watchdog timer are also discussed in the chapter. The settings shown in the manual are the default settings.

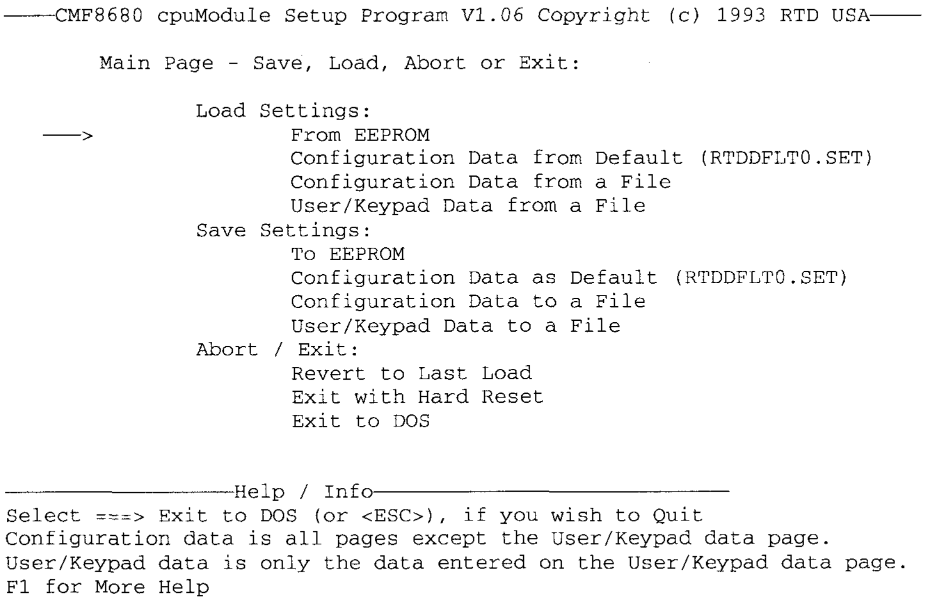
RTDMOD SETUP PROGRAM

The system configuration program, RTDMOD.EXE, is used to set up the various parameters of the CMF8680 before you boot the system. The setup program is on a utility disk shipped with your CMF8680. To run the setup program, you will need a CRT, keyboard, and floppy drive. The setup program can be placed in a ROM disk, or on the hard drive, if desired.

The first screen in the RTDMOD setup program is shown in Figure 3-1. When the program starts, it loads the settings from the configuration EEPROM to ensure that you are working with the actual system configuration. A default setting, RTDDFLT0.SET, is provided. If you should get into trouble when configuring your system, such as turning off the CRT display, you can hard reset the system and simultaneously hold down both <Shift> keys and the system will boot using the default settings in the BIOS. The RTD utility ROM disk must be installed in the user EPROM socket. The default settings are:

Drive A: 5-1/4 inch 1.2M floppy
Drive B: PCMCIA
IDE disabled
ROM-DOS enabled
CGA graphics on a CRT

The main page lets you load the current settings, save the new settings, and then exit the program or abort the new setup. The current settings can be loaded from the EEPROM, the default file, or from a user file. Settings can be saved to any one of these three locations. The abort/exit menu lets you revert to the last saved configuration, exit with a hard reset (a hard reset is necessary to implement most configuration changes), or exit to DOS without making any changes. The HELP menu is also shown in Figure 3-1 at the bottom of the screen. In addition to the keystrokes described in HELP, ESC will send you back to the Main Page.



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       Main Page - Save, Load, Abort or Exit:

              Load Settings:
   ——>               From EEPROM
                     Configuration Data from Default (RTDDFLT0.SET)
                     Configuration Data from a File
                     User/Keypad Data from a File
              Save Settings:
                     To EEPROM
                     Configuration Data as Default (RTDDFLT0.SET)
                     Configuration Data to a File
                     User/Keypad Data to a File
              Abort / Exit:
                     Revert to Last Load
                     Exit with Hard Reset
                     Exit to DOS


————————————————Help / Info————————————————————
Select ===> Exit to DOS (or <ESC>), if you wish to Quit
Configuration data is all pages except the User/Keypad data page.
User/Keypad data is only the data entered on the User/Keypad data page.
F1 for More Help
```

Fig. 3-1 — Setup Program Main Page

The F1 more help screen for Figure 3-1 looks like this:

```
----Help / Info--------------------------------------------------------
Space / Enter - Selects | TAB or Down Arrow or Left Arrow Move forward
PageUp Previous Screen  | PageDn  Next Screen
ALT-C  Toggle Colors    | ALT-D  Goto Display Screen
ALT-X  Goto Main Screen | SHIFT TAB or Up Arrow or Right Arrow Move backwards
```

The next screen, shown in Figure 3-2, lets you set the CPU clock frequency.

```
----CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA----

       System CPU Speed:

              System Clock Speed
   --->       >      14.318 MHz
                     7.159 Mhz
                     4.773 Mhz
                     3.580 Mhz








----Help / Info--------------------------------------------------------
The Floppy and IDE controllers will not work reliably, if the
system clock speed is set below 4.773 Mhz.

F1 for More Help
```

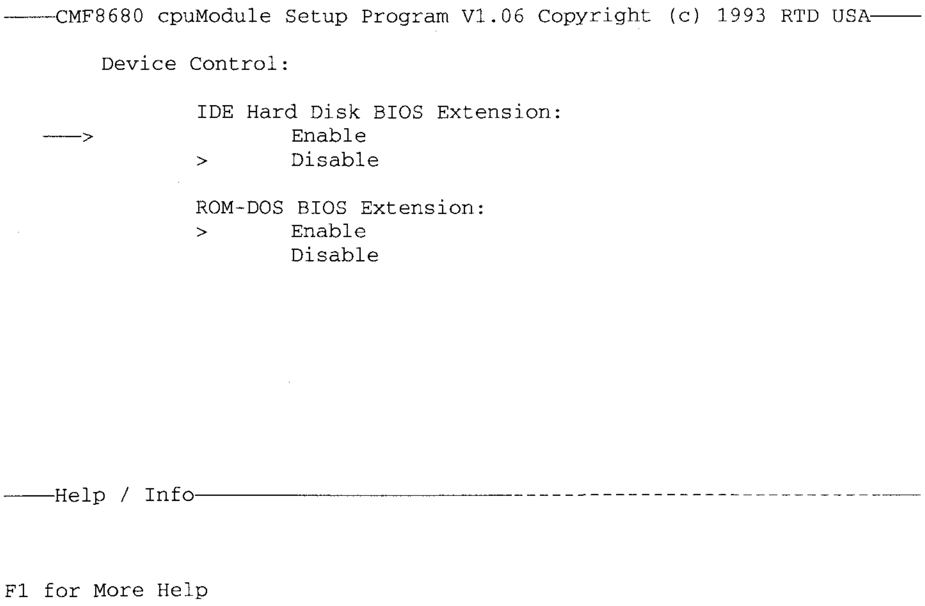
Fig. 3-2 — CPU Speed Screen

The PC104 Bus and 721 Access Speed screen, shown in Figure 3-3, sets the number of CPU clocks per bus access and enables/disables the DRAM refresh cycle.

```
----CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA----

       PC104 Bus and 721 Access Speed:

              CPU Clocks per Bus and 721 Clock
   --->       >      One
                     Two
                     Three
                     Four

              Refresh Cycles on PC104 Bus
              >      Enabled - Slows System
                     Disabled





----Help / Info--------------------------------------------------------
Refresh Cycles are only required for certain PC104 boards.


F1 for More Help
```

Fig. 3-3 — CPU Clocks/Refresh

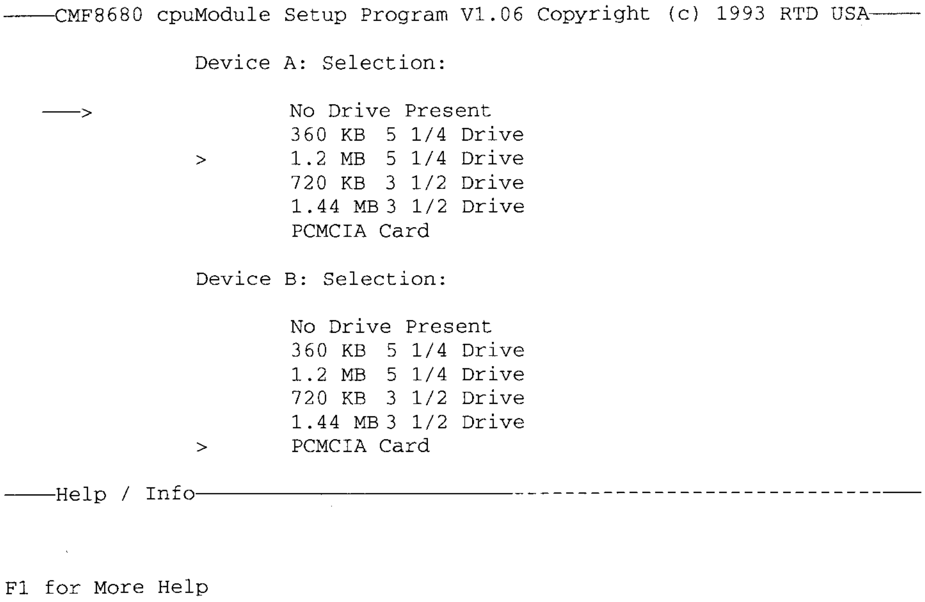
Figure 3-4 shows the enable/disable for the IDE hard drive and ROM-DOS. System boot options are defined in Table 3-2 near the end of this chapter.



```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

            Device Control:

                    IDE Hard Disk BIOS Extension:
   ───>                    Enable
                    >      Disable

                    ROM-DOS BIOS Extension:
                    >      Enable
                           Disable







────Help / Info────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-4 — IDE Hard Drive, ROM-DOS Enable/Disable

The next screen, shown in Figure 3-5, configures drive A: and drive B:. The system can be booted from drive A:.

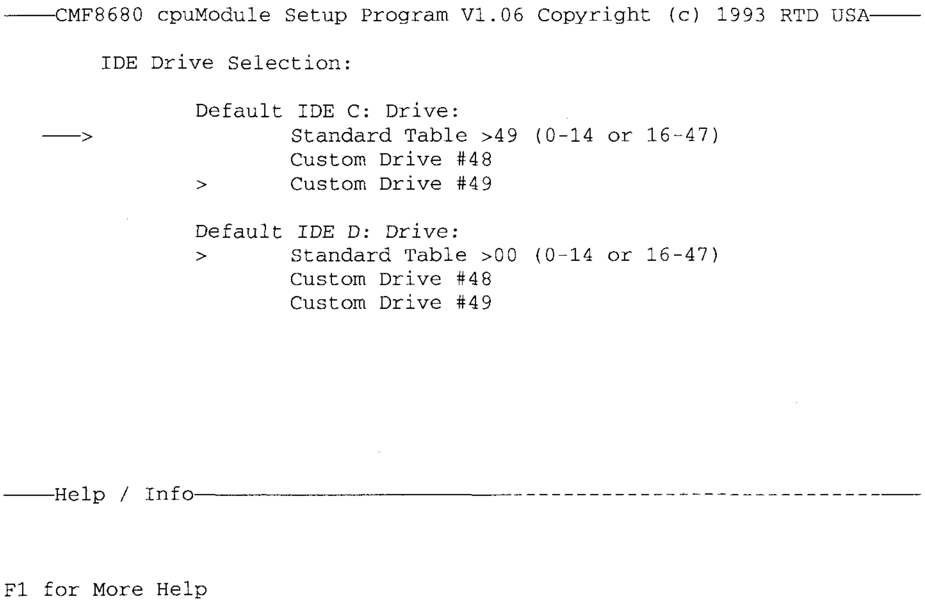


```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

                    Device A: Selection:

   ───>                    No Drive Present
                           360 KB  5 1/4 Drive
                    >      1.2 MB  5 1/4 Drive
                           720 KB  3 1/2 Drive
                           1.44 MB 3 1/2 Drive
                           PCMCIA Card

                    Device B: Selection:

                           No Drive Present
                           360 KB  5 1/4 Drive
                           1.2 MB  5 1/4 Drive
                           720 KB  3 1/2 Drive
                           1.44 MB 3 1/2 Drive
                    >      PCMCIA Card
────Help / Info────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-5 — Drive A: and Drive B: Setup

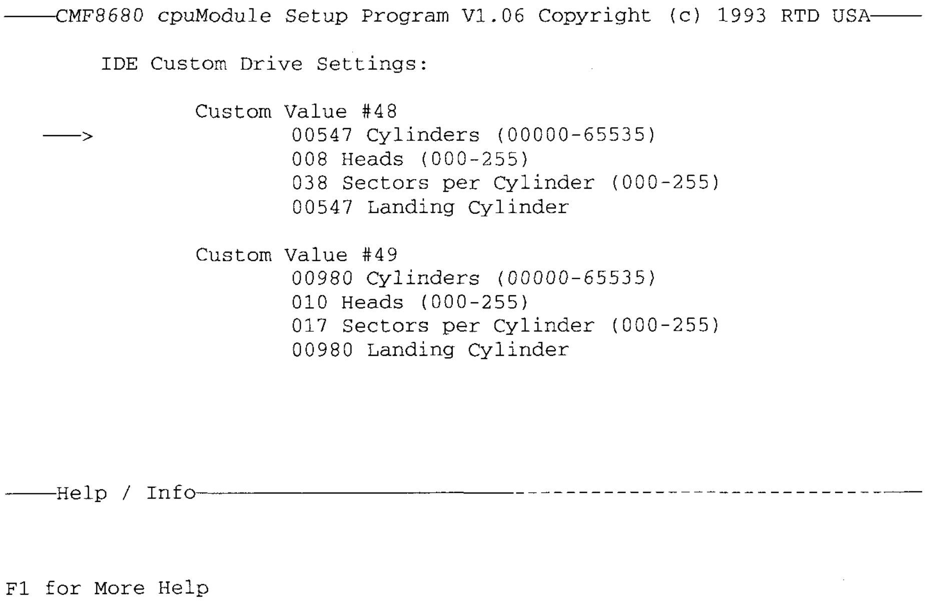
The IDE hard drive type is specified on the screen shown in Figure 3-6. The standard table for fixed disk drive types 1 through 47 is listed in Table 3-1.

It is recommended that you select user programmable drive 48 or 49 so that you can be sure the drive is configured correctly. If you choose the custom values (48 or 49), then you must fill in the next screen, shown in Figure 3-7. Note that you must enter all of the places in each number: for example, 00650 cannot be entered as 650 on the first line.



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       IDE Drive Selection:

              Default IDE C: Drive:
   ——>               Standard Table >49 (0-14 or 16-47)
                     Custom Drive #48
              >      Custom Drive #49

              Default IDE D: Drive:
              >      Standard Table >00 (0-14 or 16-47)
                     Custom Drive #48
                     Custom Drive #49




——Help / Info——————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-6 — IDE Hard Drive Setup



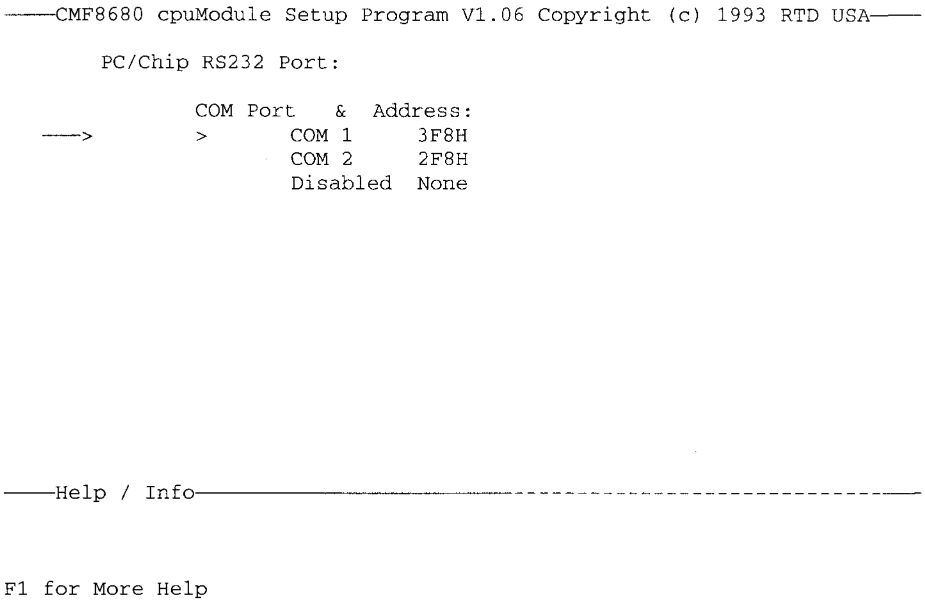
```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       IDE Custom Drive Settings:

              Custom Value #48
   ——>               00547 Cylinders (00000-65535)
                     008 Heads (000-255)
                     038 Sectors per Cylinder (000-255)
                     00547 Landing Cylinder

              Custom Value #49
                     00980 Cylinders (00000-65535)
                     010 Heads (000-255)
                     017 Sectors per Cylinder (000-255)
                     00980 Landing Cylinder



——Help / Info——————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-7 — IDE Custom Drive Settings

Table 3-1 — Standard IDE Hard Drive Table

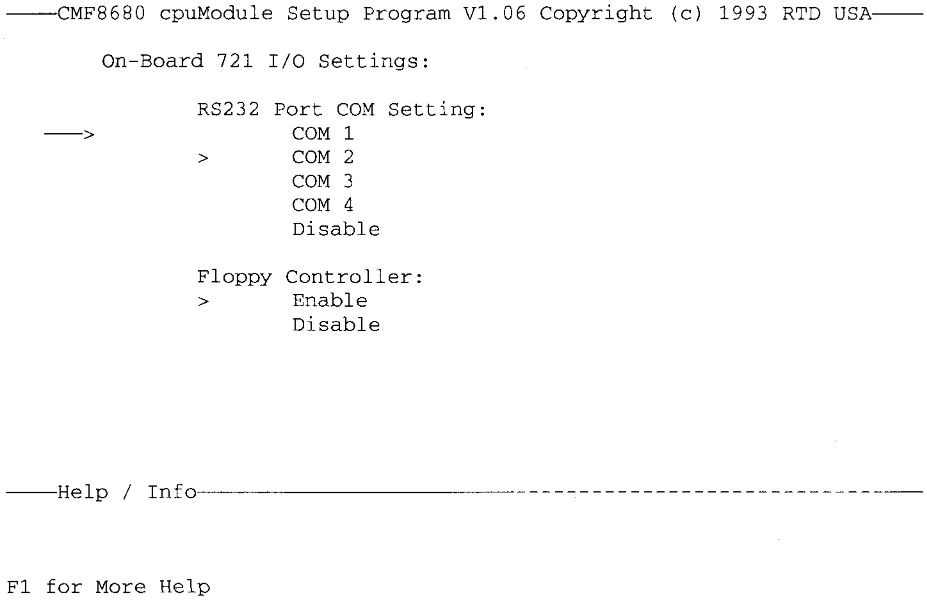
	Cyl	Heads	WPC	Control	Land	Sec
Fixed disk type 1	306	4	128	0	305	17
Fixed disk type 2	615	4	300	0	615	17
Fixed disk type 3	615	6	300	0	615	17
Fixed disk type 4	940	8	512	0	940	17
Fixed disk type 5	940	6	512	0	940	17
Fixed disk type 6	615	4	NONE	0	615	17
Fixed disk type 7	462	8	256	0	511	17
Fixed disk type 8	733	5	NONE	0	733	17
Fixed disk type 9	900	15	NONE	8	901	17
Fixed disk type 10	820	3	NONE	0	820	17
Fixed disk type 11	855	5	NONE	0	855	17
Fixed disk type 12	855	7	NONE	0	855	17
Fixed disk type 13	306	8	128	0	319	17
Fixed disk type 14	733	7	NONE	0	733	17
Fixed disk type 15	0	0	0	0	0	0
Fixed disk type 16	612	4	0	0	663	17
Fixed disk type 17	977	5	300	0	977	17
Fixed disk type 18	977	7	NONE	0	977	17
Fixed disk type 19	1024	7	512	0	1023	17
Fixed disk type 20	733	5	300	0	732	17
Fixed disk type 21	733	7	300	0	732	17
Fixed disk type 22	733	5	300	0	733	17
Fixed disk type 23	989	12	NONE	8	989	35
Fixed disk type 24	612	4	305	0	663	17
Fixed disk type 25	612	2	300	0	612	17
Fixed disk type 26	614	4	NONE	0	614	17
Fixed disk type 27	820	6	NONE	0	820	17
Fixed disk type 28	977	5	NONE	0	977	17
Fixed disk type 29	1023	9	NONE	8	1023	17
Fixed disk type 30	1024	5	NONE	0	1024	17
Fixed disk type 31	1024	8	NONE	0	1024	17
Fixed disk type 32	809	6	128	0	809	17
Fixed disk type 33	830	7	NONE	0	830	17
Fixed disk type 34	830	10	NONE	8	830	17
Fixed disk type 35	776	8	0	0	775	33
Fixed disk type 36	1024	8	NONE	0	1024	17
Fixed disk type 37	615	8	128	0	615	17
Fixed disk type 38	615	8	NONE	0	615	17
Fixed disk type 39	925	9	NONE	8	925	17
Fixed disk type 40	1024	9	NONE	8	1023	17
Fixed disk type 41	80	2	NONE	0	80	32
Fixed disk type 42	160	2	NONE	0	160	32
Fixed disk type 43	672	8	NONE	0	671	39
Fixed disk type 44	522	4	NONE	0	522	40
Fixed disk type 45	980	5	NONE	0	980	17
Fixed disk type 46	980	10	NONE	8	980	17
Fixed disk type 47	862	4	NONE	0	862	38

The next three screens, Figures 3-8 through 3-10, configure the serial ports and the I/O address for COM3 and COM4. The three serial ports are 16450 compatible serial ports that operate at up to 115.2K baud.



```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

       PC/Chip RS232 Port:

              COM Port   &  Address:
  ──>         >      COM 1     3F8H
                     COM 2     2F8H
                     Disabled  None









────Help / Info──────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-8 — F8680 RS-232 Serial Port Setup (P11)



```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

       On-Board 721 I/O Settings:

              RS232 Port COM Setting:
  ──>                COM 1
              >      COM 2
                     COM 3
                     COM 4
                     Disable

              Floppy Controller:
              >      Enable
                     Disable




────Help / Info──────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-9 — 82C721 RS-232 Serial Port Setup (P9)

```
——-CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       On-Board 721 I/O Settings:

              RS485 Port COM Setting:
   ——>               COM 1
                     COM 2
              >      COM 3
                     COM 4


              Select COM 3 and COM 4 I/O Address:
                     COM3    COM4
                     338H    238H
              >      3E8H    2E8H
                     2E8H    2E0H
                     220H    228H


——Help / Info——————————————————————————————————————————————————————————


F1 for More Help
```

NOTE: The RS-485 port cannot be disabled because some of the pins on this port are shared with the configuration EEPROM. This does not interfere with proper RS-485 operation.

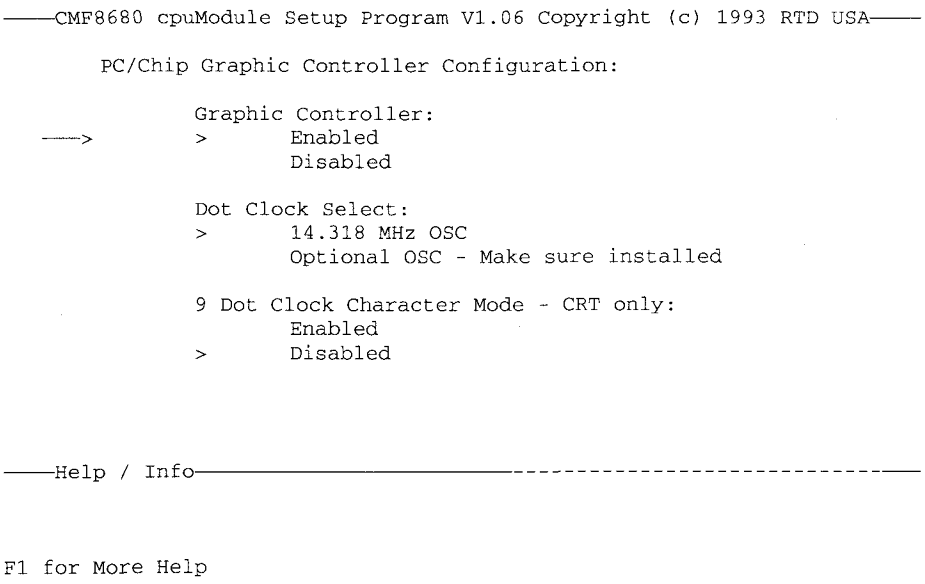
Fig. 3-10 — 82C721 RS-485 Serial Port Setup (P5)

Figure 3-11 shows the parallel port mode and address screen. The extended mode configures the port as an AT enhanced bidirectional port, and the standard mode configures the port as a standard printer port. Either mode can be used when you are interfacing to a printer.

```
——-CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       On-Board 721 I/O Settings:

              Parallel Port Mode::
   ——>               Extended Parallel Port Mode
              >      Standard Printer Mode

              Parallel Port Address:
                     3BCH
              >      378H - PC Standard
                     278H
                     Disable




——Help / Info——————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-11 — Parallel Printer Port Setup (P6)

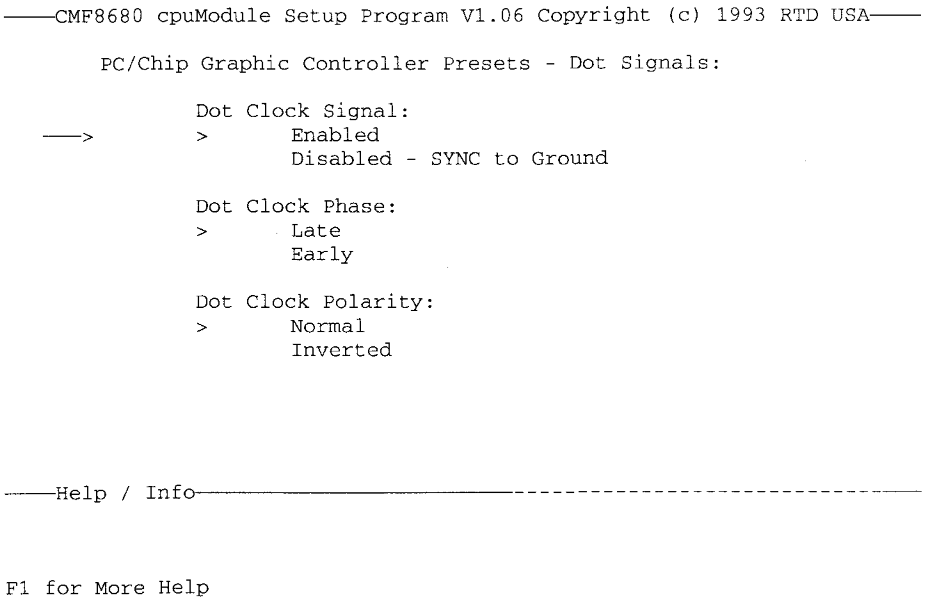
Figure 3-12 configures the internal graphics controller. Remember that if you disable the controller, you will not have CGA graphics and you must make sure that you have another display device (e.g., VGA card and monitor). When using a standard VGA board with the CMF8680, it is not necessary to disable the CGA graphics; however, most VGA boards will try to emulate the CGA modes, and this could cause problems.



```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

       PC/Chip Graphic Controller Configuration:

              Graphic Controller:
   ───>       >      Enabled
                     Disabled

              Dot Clock Select:
              >      14.318 MHz OSC
                     Optional OSC - Make sure installed

              9 Dot Clock Character Mode - CRT only:
                     Enabled
              >      Disabled



────Help / Info──────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-12 — Graphics Controller Setup

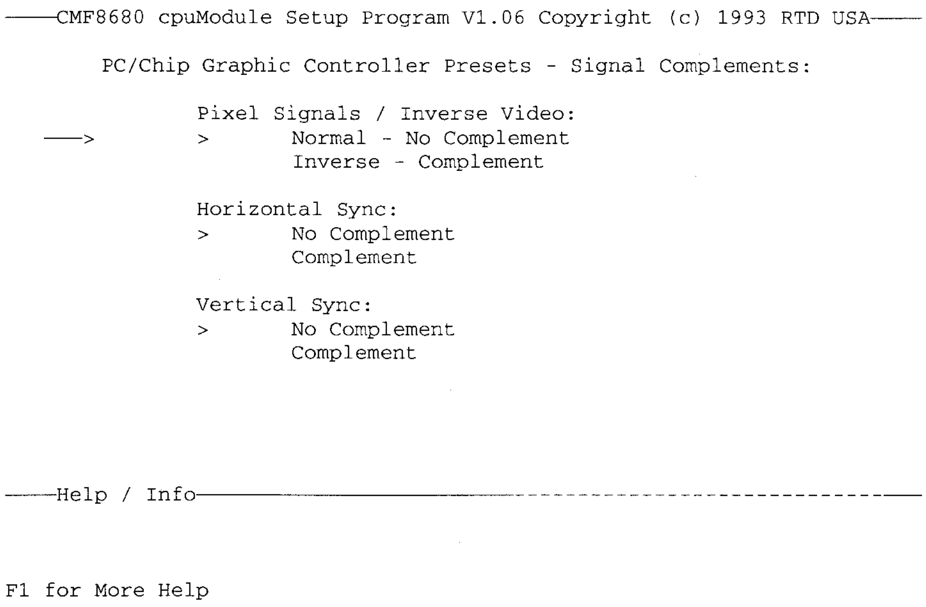
Figures 3-13 through 3-15 show the screens for the CGA CRT/LCD graphics setup. Figure 3-13 selects one of the four positions for the Dot Clock.



```
────CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA────

       PC/Chip Graphic Controller Presets - Dot Signals:

              Dot Clock Signal:
   ───>       >      Enabled
                     Disabled - SYNC to Ground

              Dot Clock Phase:
              >      Late
                     Early

              Dot Clock Polarity:
              >      Normal
                     Inverted




────Help / Info──────────────────────────────────────────────────────────


F1 for More Help
```

Fig. 3-13 — Graphics Controller, Dot Signals

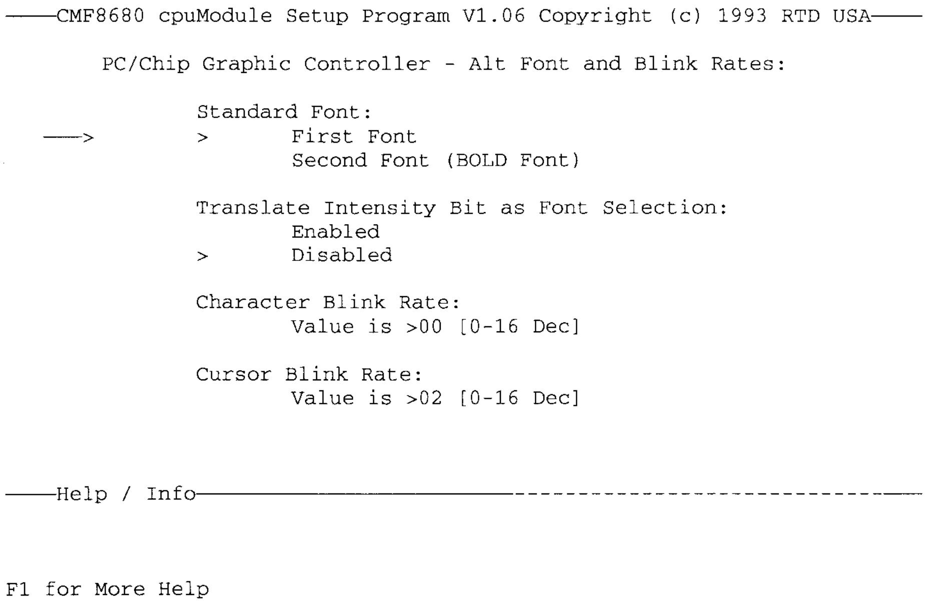
Figure 3-14 allows inverting of various video signals.



```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

      PC/Chip Graphic Controller Presets - Signal Complements:

            Pixel Signals / Inverse Video:
  ——>       >      Normal - No Complement
                   Inverse - Complement

            Horizontal Sync:
            >      No Complement
                   Complement

            Vertical Sync:
            >      No Complement
                   Complement



———Help / Info————————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-14 — Graphics Controller, Signal Complements

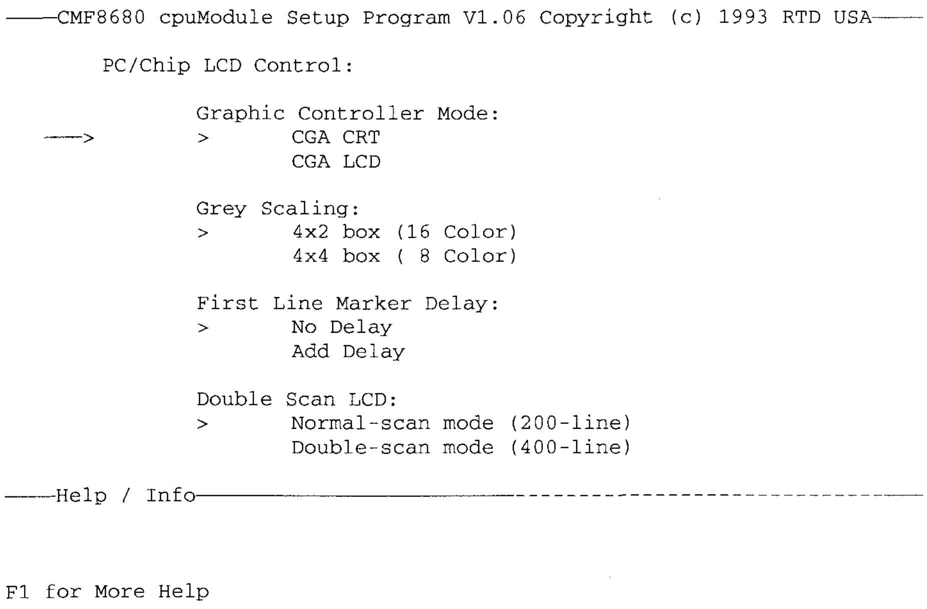
Figure 3-15 selects the font which is displayed and the blink rate options.



```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

      PC/Chip Graphic Controller - Alt Font and Blink Rates:

            Standard Font:
  ——>       >      First Font
                   Second Font (BOLD Font)

            Translate Intensity Bit as Font Selection:
                   Enabled
            >      Disabled

            Character Blink Rate:
                   Value is >00 [0-16 Dec]

            Cursor Blink Rate:
                   Value is >02 [0-16 Dec]


———Help / Info————————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-15 — Graphics Controller, Alternate Font and Blink Rates

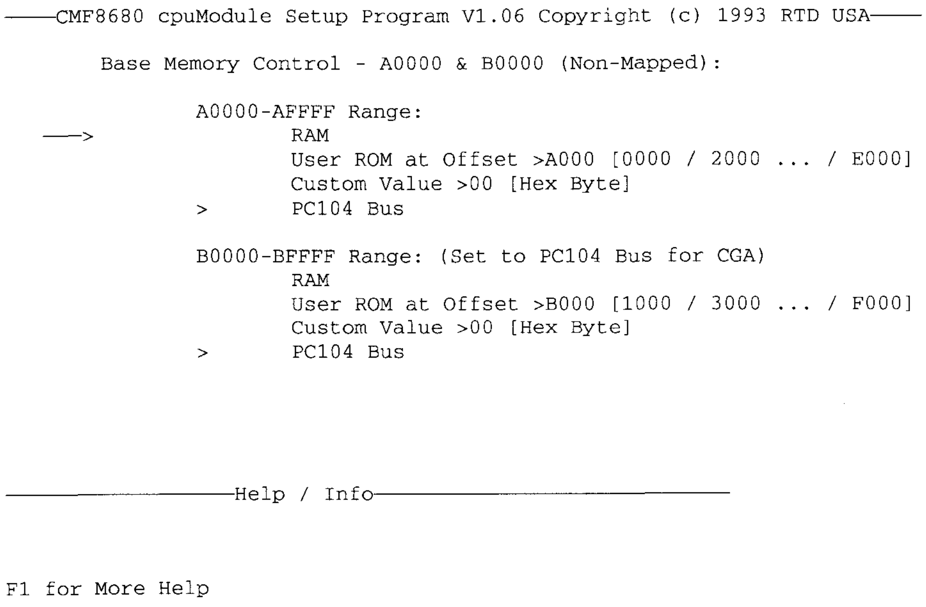
Figure 3-16 controls the CRT or LCD selection and the LCD parameters for grey scaling, first line marker delay, and double scan mode.



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

        PC/Chip LCD Control:

                Graphic Controller Mode:
   ——>          >       CGA CRT
                        CGA LCD

                Grey Scaling:
                >       4x2 box (16 Color)
                        4x4 box ( 8 Color)

                First Line Marker Delay:
                >       No Delay
                        Add Delay

                Double Scan LCD:
                >       Normal-scan mode (200-line)
                        Double-scan mode (400-line)

——Help / Info——————————————————————————————————————————————————————


F1 for More Help
```

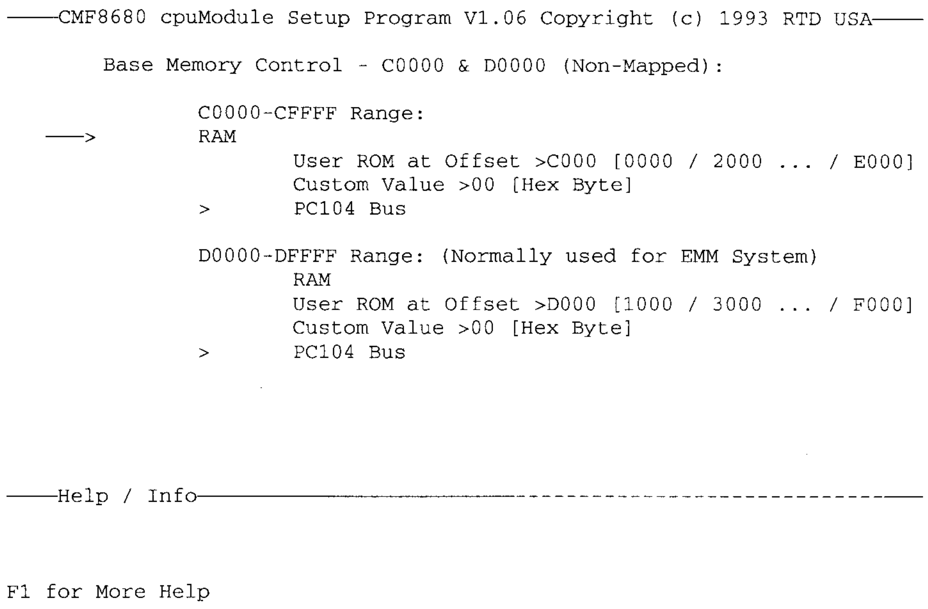
Fig. 3-16 — LCD Control Setup

Figures 3-17 and 3-18 show the Base Memory addressing. It is recommended that the factory settings (PC/104 BUS) or RAM be used unless you have complete information about the PC/CHIP memory management in order to avoid contention.



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

        Base Memory Control - A0000 & B0000 (Non-Mapped):

                A0000-AFFFF Range:
   ——>                  RAM
                        User ROM at Offset >A000 [0000 / 2000 ... / E000]
                        Custom Value >00 [Hex Byte]
                >       PC104 Bus

                B0000-BFFFF Range: (Set to PC104 Bus for CGA)
                        RAM
                        User ROM at Offset >B000 [1000 / 3000 ... / F000]
                        Custom Value >00 [Hex Byte]
                >       PC104 Bus



——————————————————Help / Info——————————————————————————


F1 for More Help
```

Fig. 3-17 — Base Memory, Screen 1



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

     Base Memory Control - C0000 & D0000 (Non-Mapped):

          C0000-CFFFF Range:
  ——>     RAM
               User ROM at Offset >C000 [0000 / 2000 ... / E000]
               Custom Value >00 [Hex Byte]
          >    PC104 Bus

          D0000-DFFFF Range: (Normally used for EMM System)
               RAM
               User ROM at Offset >D000 [1000 / 3000 ... / F000]
               Custom Value >00 [Hex Byte]
          >    PC104 Bus




——Help / Info—————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-18 — Base Memory, Screen 2

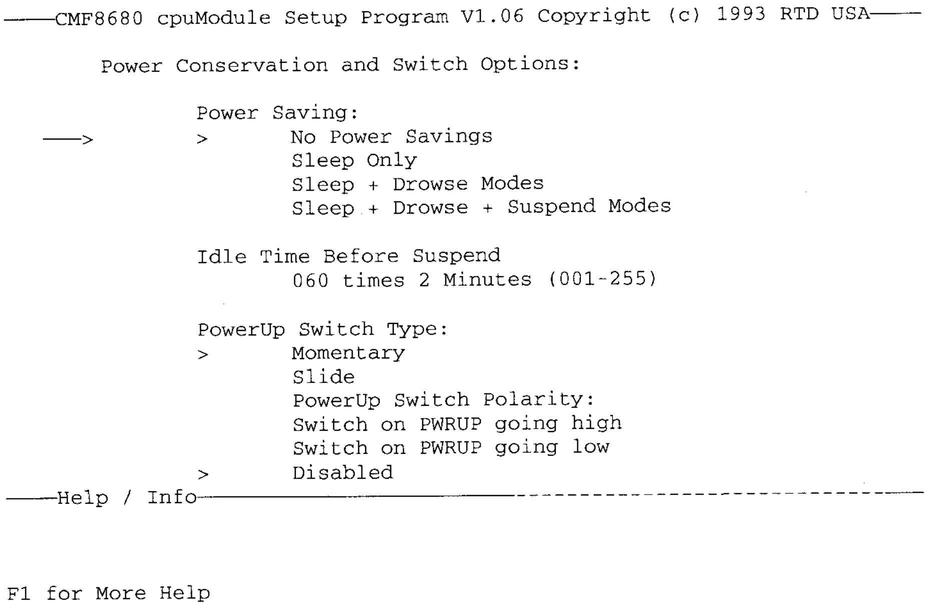
Power management options are shown in Figure 3-19. The PWRUP switch type and polarity are set here according to the switch connected to utility connector P4, pin 4. Refer to Chapter 4 for more information about power management. While there are many possible combinations of switch type and mode of operation of the PWRUP signal, the following paragraphs describe some ways you can implement this feature. We will look at slide and momentary switch operation to describe the basic function of the PWRUP signal.

> **NOTE:** It is very important that any switch connected to the PWRUP signal be properly debounced.

Slide Switch Operation: When the PWRUP signal is high, the CMF8680 is running, and when PWRUP is low, it is in the suspend mode. To implement this configuration, use RTDMOD.EXE to set the PowerUp Switch Type to *Slide* and the PowerUp Switch Polarity to *Switch on PWRUP going High.* The switch you connect to the PWRUP signal can be a slide switch that is open in one position and grounded in the other position (PWRUP is internally pulled up on the CMF8680). When the switch is open, the CMF8680 will run, and when it is grounded, the CMF8680 will be in the suspend mode. A normally open momentary pushbutton switch can also be used in this configuration. When the button is released, the CMF8680 is running, and when it is pressed, the CMF8680 enters the suspend mode.

Momentary Switch Operation: This mode requires that you use a normally closed pushbutton switch connected to the PWRUP signal. PWRUP should be low for normal operation. When PWRUP is brought high and then back to low, the CMF8680 enters suspend mode. When PWRUP goes high and then low a second time, the CMF8680 resumes running. To implement this configuration, use RTDMOD.EXE to set the PowerUp Switch Type to *Momentary* and the PowerUp Switch Polarity to *Switch on PWRUP going Low.*

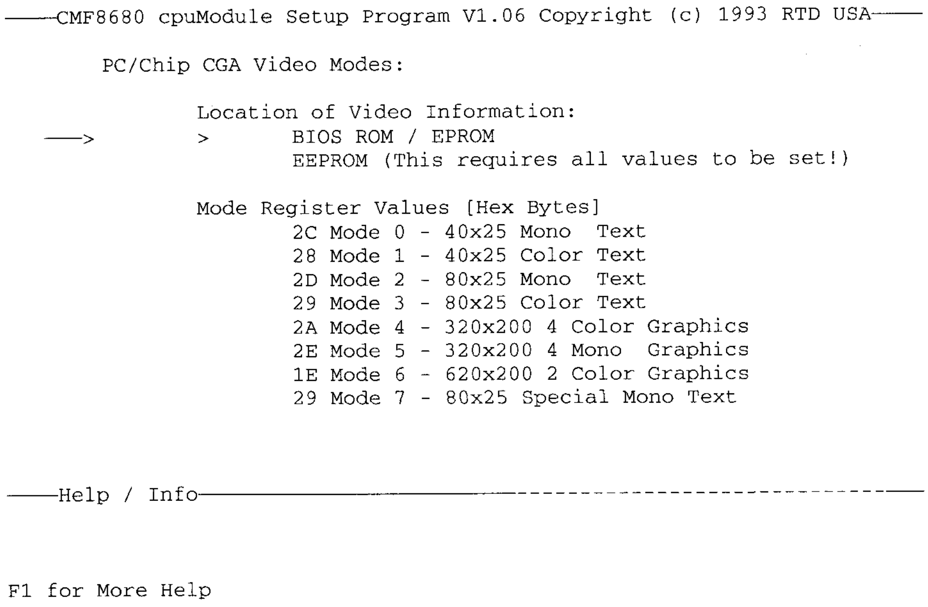
If you do not use the PWRUP signal, set the PowerUp Switch Polarity to *Disabled.*



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       Power Conservation and Switch Options:

              Power Saving:
   ——>        >      No Power Savings
                     Sleep Only
                     Sleep + Drowse Modes
                     Sleep + Drowse + Suspend Modes

              Idle Time Before Suspend
                     060 times 2 Minutes (001-255)

              PowerUp Switch Type:
              >      Momentary
                     Slide
                     PowerUp Switch Polarity:
                     Switch on PWRUP going high
                     Switch on PWRUP going low
              >      Disabled
——Help / Info——————————————————————————————————————————————————————————


F1 for More Help
```

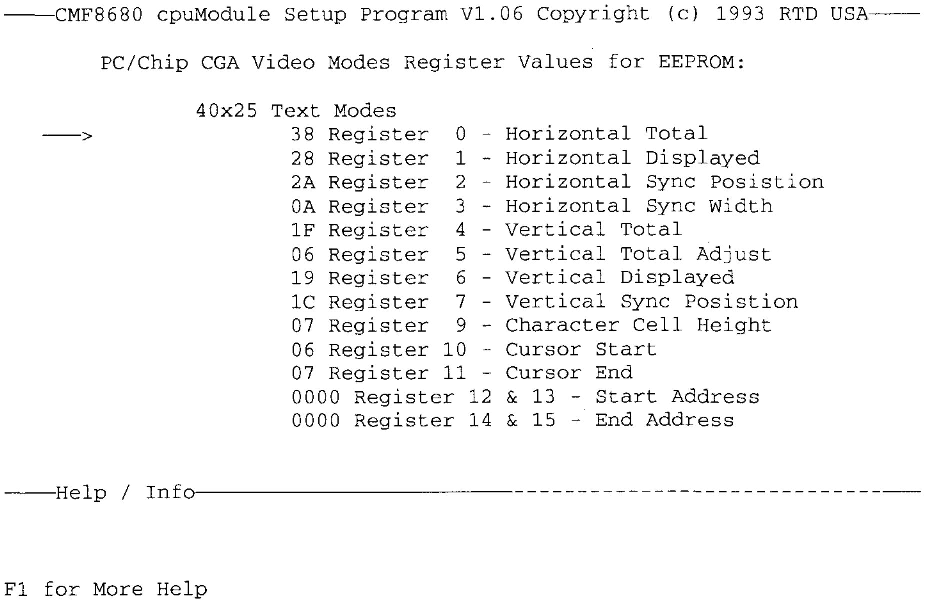
Fig. 3-19 — Power Management Setup

Figures 3-20 through 3-24 let you select whether the CGA video information is stored in the BIOS or in the EEPROM. If you select the EEPROM, you must set each value on the screens shown in Figures 3-20 through 3-24. The CRT controller is 6845 compatible.



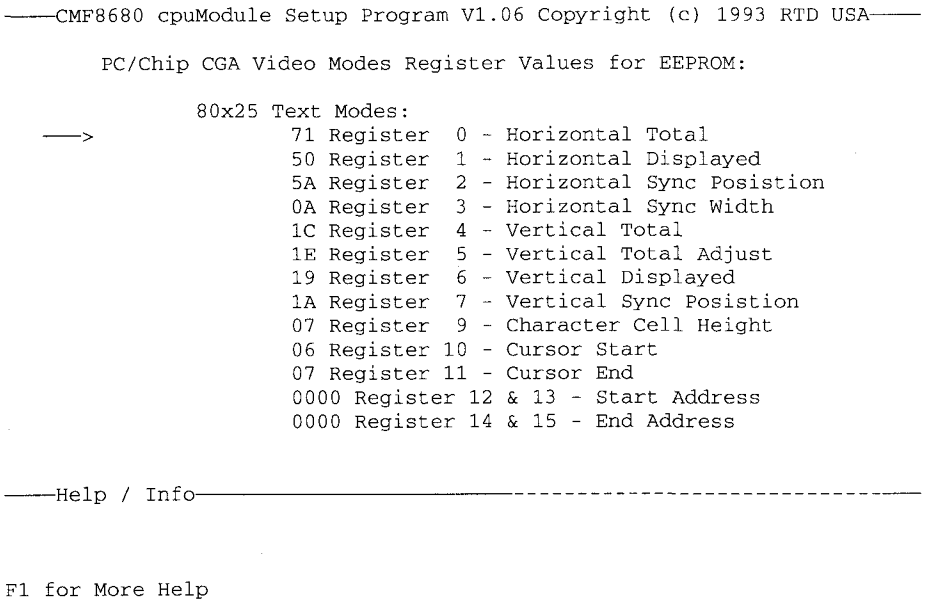
```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       PC/Chip CGA Video Modes:

              Location of Video Information:
   ——>        >      BIOS ROM / EPROM
                     EEPROM (This requires all values to be set!)

              Mode Register Values [Hex Bytes]
                     2C Mode 0 - 40x25 Mono  Text
                     28 Mode 1 - 40x25 Color Text
                     2D Mode 2 - 80x25 Mono  Text
                     29 Mode 3 - 80x25 Color Text
                     2A Mode 4 - 320x200 4 Color Graphics
                     2E Mode 5 - 320x200 4 Mono  Graphics
                     1E Mode 6 - 620x200 2 Color Graphics
                     29 Mode 7 - 80x25 Special Mono Text


——Help / Info——————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-20 — CGA Video Modes



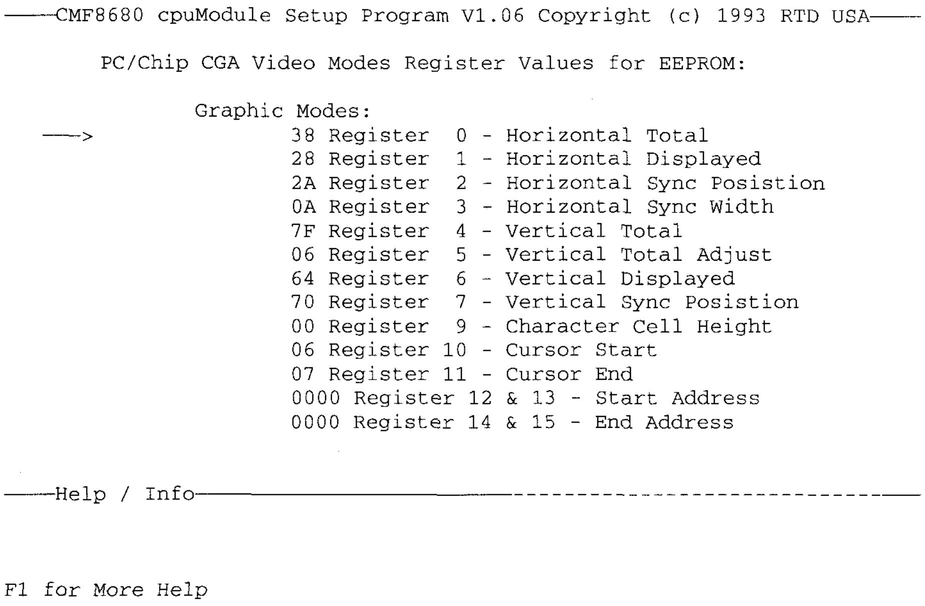
```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       PC/Chip CGA Video Modes Register Values for EEPROM:

              40x25 Text Modes
 ——>                 38 Register  0 - Horizontal Total
                     28 Register  1 - Horizontal Displayed
                     2A Register  2 - Horizontal Sync Posistion
                     0A Register  3 - Horizontal Sync Width
                     1F Register  4 - Vertical Total
                     06 Register  5 - Vertical Total Adjust
                     19 Register  6 - Vertical Displayed
                     1C Register  7 - Vertical Sync Posistion
                     07 Register  9 - Character Cell Height
                     06 Register 10 - Cursor Start
                     07 Register 11 - Cursor End
                     0000 Register 12 & 13 - Start Address
                     0000 Register 14 & 15 - End Address

——Help / Info————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-21 — CGA Video Modes Register Values for EEPROM, 40x25 Mode



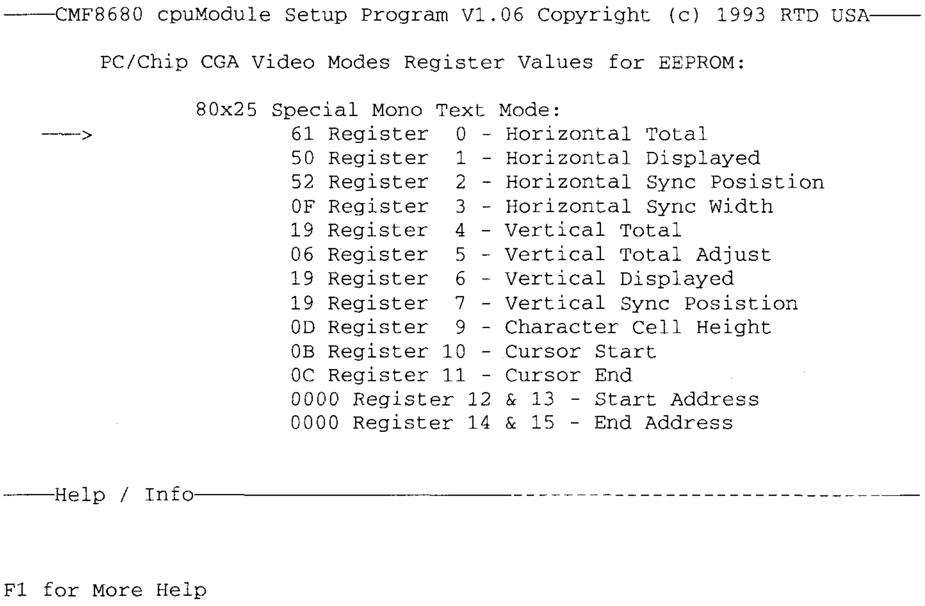
```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       PC/Chip CGA Video Modes Register Values for EEPROM:

              80x25 Text Modes:
 ——>                 71 Register  0 - Horizontal Total
                     50 Register  1 - Horizontal Displayed
                     5A Register  2 - Horizontal Sync Posistion
                     0A Register  3 - Horizontal Sync Width
                     1C Register  4 - Vertical Total
                     1E Register  5 - Vertical Total Adjust
                     19 Register  6 - Vertical Displayed
                     1A Register  7 - Vertical Sync Posistion
                     07 Register  9 - Character Cell Height
                     06 Register 10 - Cursor Start
                     07 Register 11 - Cursor End
                     0000 Register 12 & 13 - Start Address
                     0000 Register 14 & 15 - End Address

——Help / Info————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-22 — CGA Video Modes Register Values for EEPROM, 80x25 Mode



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       PC/Chip CGA Video Modes Register Values for EEPROM:

              Graphic Modes:
  ——>                38 Register  0 - Horizontal Total
                     28 Register  1 - Horizontal Displayed
                     2A Register  2 - Horizontal Sync Posistion
                     0A Register  3 - Horizontal Sync Width
                     7F Register  4 - Vertical Total
                     06 Register  5 - Vertical Total Adjust
                     64 Register  6 - Vertical Displayed
                     70 Register  7 - Vertical Sync Posistion
                     00 Register  9 - Character Cell Height
                     06 Register 10 - Cursor Start
                     07 Register 11 - Cursor End
                     0000 Register 12 & 13 - Start Address
                     0000 Register 14 & 15 - End Address

——Help / Info————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-23 — CGA Video Modes Register Values for EEPROM, Graphic Mode

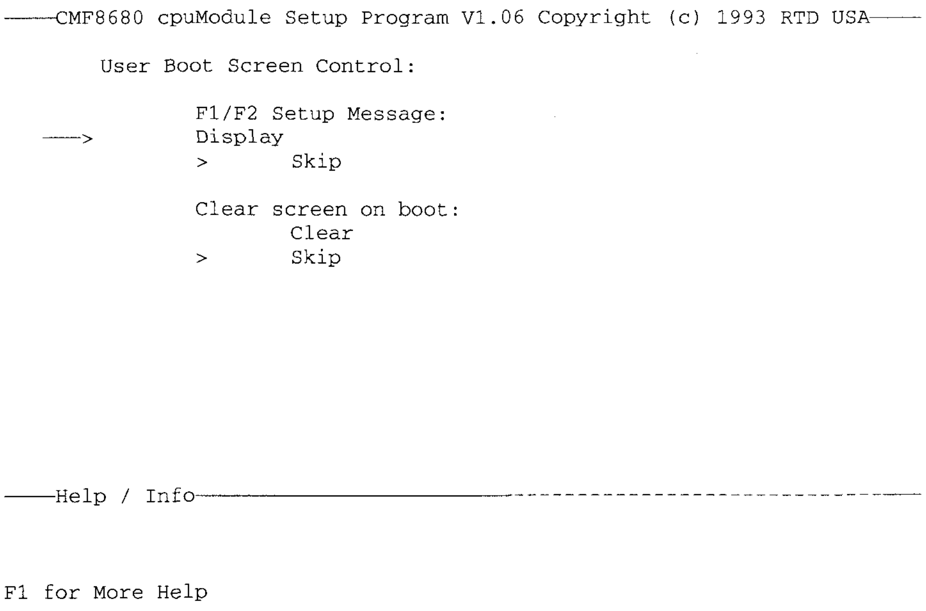


```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       PC/Chip CGA Video Modes Register Values for EEPROM:

              80x25 Special Mono Text Mode:
  ——>                61 Register  0 - Horizontal Total
                     50 Register  1 - Horizontal Displayed
                     52 Register  2 - Horizontal Sync Posistion
                     0F Register  3 - Horizontal Sync Width
                     19 Register  4 - Vertical Total
                     06 Register  5 - Vertical Total Adjust
                     19 Register  6 - Vertical Displayed
                     19 Register  7 - Vertical Sync Posistion
                     0D Register  9 - Character Cell Height
                     0B Register 10 - Cursor Start
                     0C Register 11 - Cursor End
                     0000 Register 12 & 13 - Start Address
                     0000 Register 14 & 15 - End Address

——Help / Info————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-24 — CGA Video Modes Register Values for EEPROM, 80x25 Special Mono (640x400 Double Scan Mode)

The screen control is set up as shown in Figure 3-25. The F1/F2 setup message allows the user to change system parameters using the setup provided in the Chips & Technologies BIOS.

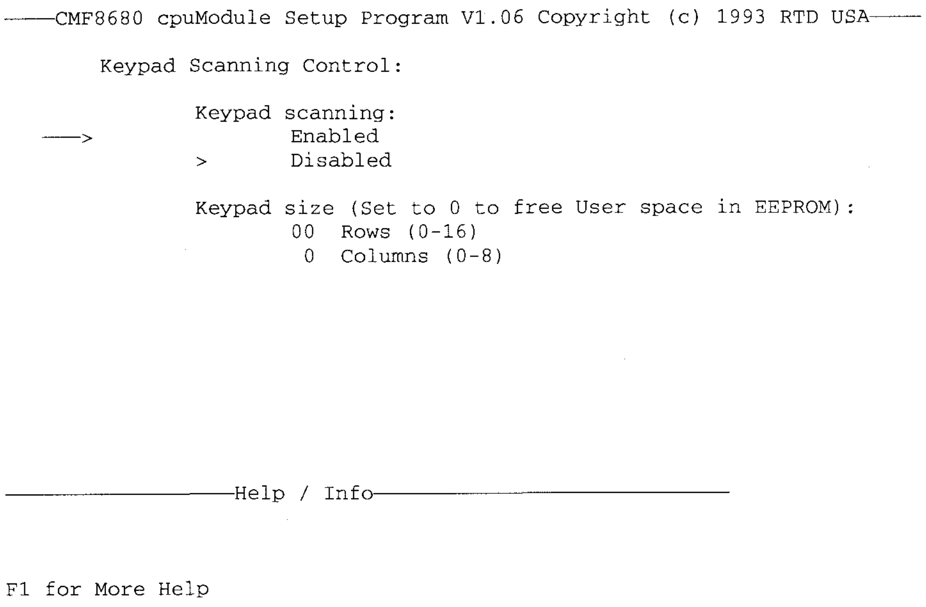
NOTE: Any changes made in this setup program will take effect as soon as the program is exited; however, the settings made here will not be saved. Also, the data values displayed for the hard disk types are incorrect. The values used are shown in Table 3-1.



```
————CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA————

       User Boot Screen Control:

              F1/F2 Setup Message:
   ——>        Display
              >      Skip

              Clear screen on boot:
                     Clear
              >      Skip







————Help / Info——————————————————————————————————————————————————————————


F1 for More Help
```

Fig. 3-25 — User Boot Screen Control

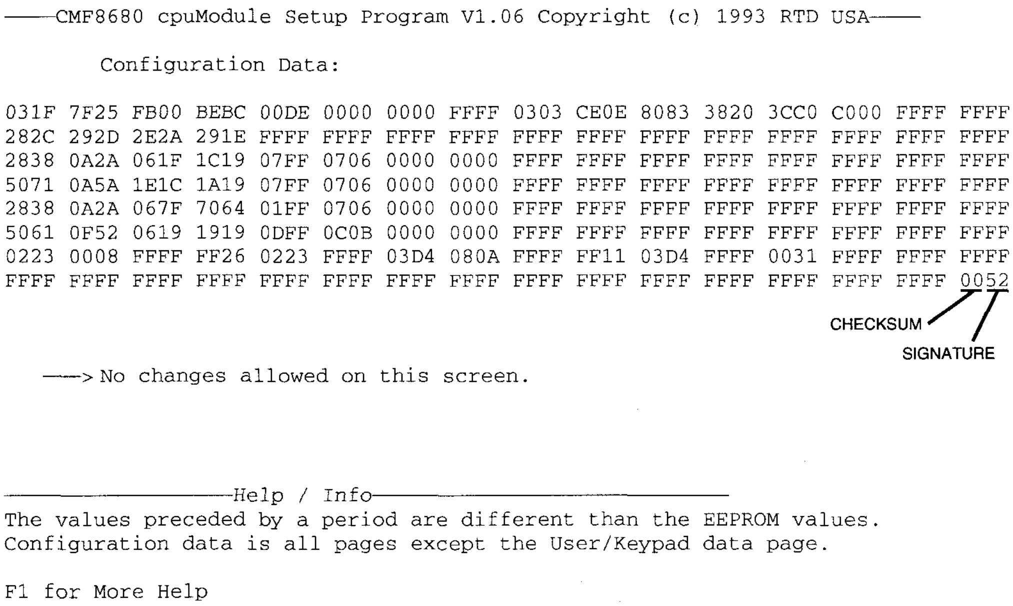
Keypad scanning is enabled and the size of the matrix is set up in the screen shown in Figure 3-26. You can program up to 256 keys, as described in Chapter 6. Here, you set up the number of rows by the number of columns of your keypad. Then go to the user/keypad data screen (see Figure 3-28 and text) to program the values for the keys.



```
————CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA————

       Keypad Scanning Control:

              Keypad scanning:
   ——>               Enabled
              >      Disabled

              Keypad size (Set to 0 to free User space in EEPROM):
                     00  Rows (0-16)
                      0  Columns (0-8)







————————————————Help / Info—————————————————————————


F1 for More Help
```

Fig. 3-26 — Keypad Scanning Setup

Each value of the serial EEPROM is shown in Figure 3-27. Note that the last word contains the signature byte "52" in the low byte and the checksum in the high byte.

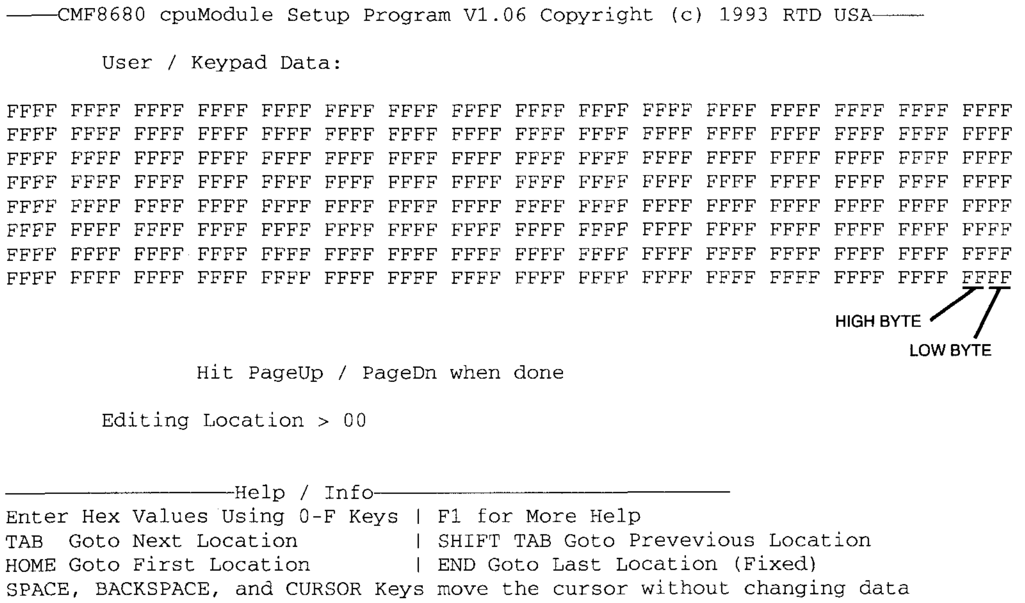


```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

      Configuration Data:

031F 7F25 FB00 BEBC 00DE 0000 0000 FFFF 0303 CE0E 8083 3820 3CC0 C000 FFFF FFFF
282C 292D 2E2A 291E FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
2838 0A2A 061F 1C19 07FF 0706 0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
5071 0A5A 1E1C 1A19 07FF 0706 0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
2838 0A2A 067F 7064 01FF 0706 0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
5061 0F52 0619 1919 0DFF 0C0B 0000 0000 FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0223 0008 FFFF FF26 0223 FFFF 03D4 080A FFFF FF11 03D4 FFFF 0031 FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0052

   ——> No changes allowed on this screen.


——————————Help / Info———————————————
The values preceded by a period are different than the EEPROM values.
Configuration data is all pages except the User/Keypad data page.

F1 for More Help
```

Fig. 3-27 — EEPROM Values Screen

Figure 3-28 shows the screen where you program the values for your keypad matrix when you are using the keypad scanning feature of the CM102 utilityModule with the CMF8680. Depending on the number of rows by columns you configured on the screen shown in Figure 3-26, all or part of this table will be used to enter the scan codes. The portion of the table where the scan codes are to be entered is highlighted. For example, suppose you select a matrix size smaller than 16 rows by 8 columns, such as 4 rows by 2 columns. The table in Figure 3-28 will have the last eight words highlighted to enter your scan codes. When your matrix is smaller than the full table, the data words used to enter scan codes are always highlighted from some point within the table to the end. This allows unused memory to be free for other functions.

When you enter data, the first word (going from top to bottom and left to right) corresponds to row 1, column 1 on the keypad; the second word corresponds to row 1, column 2; and so on. Each word in the table programs two keys: the low byte programs the scan code for the keystroke when the Fn (or Ctrl + Alt) key is NOT depressed; the high byte programs the scan code for the keystroke when the Fn key IS depressed. Note that in order to activate the Fn key feature, at some location in the table, you must define both the low and high bytes of the SAME WORD with the Fn key scan code. The scan codes are defined in Chapter 6. Refer to this chapter for complete information on programming scan codes.



```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

       User / Keypad Data:

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF


              Hit PageUp / PageDn when done

       Editing Location > 00


—————————————————Help / Info——————————————————————————
Enter Hex Values Using 0-F Keys | F1 for More Help
TAB  Goto Next Location         | SHIFT TAB Goto Prevevious Location
HOME Goto First Location        | END Goto Last Location (Fixed)
SPACE, BACKSPACE, and CURSOR Keys move the cursor without changing data
```

Fig. 3-28 — User/Keypad Data Screen

SYSTEM BOOT OPTIONS

The CMF8680 can be configured to boot from any one of several sources. They are:

- 360K, 1.2M, 720K, 1.44M floppy configured as drive A:
- PCMCIA memory card configured as drive A:
- IDE hard drive
- ROM-DOS ROM disk

If both the IDE hard drive and ROM-DOS are enabled, you can access both; however, the system will boot from the ROM-DOS disk. If ROM-DOS is disabled but you boot with ROM-DOS on a floppy or hard drive, then the ROMDISK will be available. If you do not boot the system with ROM-DOS, then the ROMDISK will not be available. Holding both <Shift> keys down while the CMF8680 is booting will abort the EEPROM load, and the system will boot to the default settings of: ROM-DOS enabled, IDE disabled, Drive A: 5-1/4 inch floppy, and CGA CRT enabled. A bootable floppy disk with ROM-DOS is provided with your CMF8680. Table 3-2 shows the system boot options.

Table 3-2 — System Boot Options

IDE Drive	ROM DOS	Drive A:	Boot From
Disable	Disable	Not Ready	Will not boot
Disable	Enable	Not Ready	ROMDISK
Enable	Disable	Not Ready	IDE drive with ROMDISK available if using ROM-DOS
Enable	Enable	Not Ready	ROMDISK with IDE available
Disable	Disable	Ready	Floppy drive with ROMDISK available if using ROM-DOS
Disable	Enable	Ready	Floppy drive with ROMDISK available if using ROM-DOS
Enable	Disable	Ready	Floppy drive with IDE available & ROMDISK available if using ROM-DOS
Enable	Enable	Ready	Floppy drive with IDE available & ROMDISK available if using ROM-DOS

BIOS AND SUPERSTATE™ R

SuperState R is a combination of special software instructions and internal PC/CHIP™ hardware that extend the processor core, allowing additional features to be supported transparently to normal DOS and BIOS operation, without the need for complex hardware.

SuperState provides:

– Control over hardware extensions within the PC/CHIP™

Five programmable pins that are used to:
provide the PCMCIA card R/W detect.
provide LCD signals.
allow watchdog timer control.

Power control logic which allows the chip to be powered-up by internal timer activity or hardware switch and powered-down by software.

Memory access controller registers which generate the appropriate cycles for ROM, dynamic RAM, PCMCIA cards, and the PC/104 bus.

Performance control registers which allow CPU waits between instruction execution to be adjusted to reduce system power consumption.

– Memory Management, which allows up to 64 MB of memory to be addressed through a 64 KB mapping system, which in this design can reach the combination of ROM (1.25 MB), RAM (2 MB), and two PCMCIA devices (32 MB each). This control also allows the following 'user' features without additional hardware:

EMM driver which provides support for the LIM EMS 4.0 standard, that allows the DOS programmer to reach memory beyond the 1 MB addressing space of the 8086 class processor.

HIDOS device driver which allows DOS to be loaded in HMA memory space (memory above the 1 MB address space).

ROMDISK access for ROM-DOS and user EPROM based disk emulation.

- Virtual I/O™ and Virtual Interrupts™, which allow for the emulation of floppy disk hardware to support PCMCIA disks that act like floppy diskettes to software that does direct hardware access and control. These features are also used to access IDE hard disks.

WATCHDOG TIMER

The CMF8680 has a hardware watchdog timer that is user controllable. When enabled, the watchdog timer must be reset every 1.6 seconds or it will perform a hardware reset on the CMF8680. See Application Note CMF8680-1, *Using the CMF8680 Watchdog Timer,* in Appendix D for detailed operation and example programs.

BATTERY BACKUP

The system switches to external battery backup from utility connector P4-9 to maintain the real-time clock when power is switched off. If maintaining the date and time while the system is off is not required, the battery can be omitted. All system configuration parameters are saved in the non-volatile serial EEPROM.

LOADING DOS HIGH

HIDOS.SYS is a device driver for the PC/CHIP that allows MS-DOS 5.00 to be loaded in HMA (memory above 1 megabyte). The following two lines should be added to the CONFIG.SYS file:

```
DEVICE = HIDOS.SYS
DOS = HIGH
```

EXPANDED MEMORY

This section describes the MS-DOS device driver PCCEMM.SYS, which, in conjunction with a Chips and Technologies F8680 PC/CHIP, provides support for LIM EMS 4.0.

This EMS driver is supplied as a device driver that can be loaded at DOS boot time. The user is required to edit the CONFIG.SYS file so that it contains a line in the form of:

```
DEVICE = d:[path]\PCCEMM.SYS [options]
```

where d: and [path] represent the drive and path where the driver is located.

The driver is designed so that, under normal circumstances, it does not need any command line options in order to give satisfactory performance. However, command line options are provided so that the driver may be tailored to a user's specific needs.

Before installing the driver, you should run your system SETUP and make sure to set aside sufficient memory for your needs. This memory should be designated as expanded memory.

Command Line Parameters

The following command line parameters are available for knowledgeable users who wish to tailor the driver for their own requirements. Option names may be shortened to their first two characters. Options should be separated by one or more spaces. Spaces are optional before and after the '-' and '=' symbols.

HANDLES = hhh

Specifies the number of handles and names available. "hhh" must be in the range of 16 to 255, with the default being 64.

IOADDR = aaa

Overrides the default base I/O address at which the PC/CHIP should be configured. aaa can be 208, 218, 258, 268, 2A8, 2B8, or 2E8.

DIAGS = xxx

Specifies whether EMS memory diagnostics are performed during initialization of the driver. "xxx" must be ON or OFF. The default is OFF.

SIZE = ssss

Specifies how much of the RAM above 1M should be used for EMS memory. If there is no XMS driver in the system, then the default for the EMS driver is to use as much memory as is available. If there is an XMS driver in the system, then the EMS driver will default to using 256K for EMS. These defaults may be overridden by using this option. ssss is specified in k's and must be a multiple of 64k.

Error Messages

"Initialization Error. Memory Manager not Installed"

This is a generic error message that is accompanied by another error message. This means that for some reason (given by the other error message) the EMM driver could not be installed correctly.

"Cannot detect mapping hardware"

The EMM driver could not find a F8680 PC/CHIP in the system. Make sure that your EMS is enabled. This error may indicate that you have the wrong driver for your system.

"Problems encountered with memory initialization"

The EMM driver could not set up the page frame correctly. Make sure that there is a contiguous 64K segment of unused memory space at D000.

"Interrupt 67 Vector already allocated"

Some other driver has already allocated the EMS interrupt vector. Remove all other EMS drivers from your CONFIG.SYS file.

"I/O address specified is not one of the permitted values"

The EMS mapper I/O address you indicated using the IOADDR command line option is an invalid address. Make sure that it is 208.

"An equals (=) sign was expected"

One of your PCCEMM.SYS command line options was not immediately followed by an equals '=' sign. Check your CONFIG.SYS file to make sure the PCCEMM.SYS command line is formatted correctly.

"Number of handles invalid or out range"

The value you specified for the HANDLES command line option is invalid. Make sure it is a decimal number between 16 and 255.

"No working memory can be found. EMM aborted"

The driver cannot find any memory set aside as expanded memory. Make sure to run your system SETUP, and set aside some memory for expanded memory.

"Unrecognized command/option in command line"

The string displayed after this error message was found to be invalid. Edit your CONFIG.SYS to make sure this invalid option is corrected.

"The command line is incomplete"

Options are incomplete on the PCCEMM.SYS command line. Make sure that all command line options are correctly formatted.

"Problems encountered in setting default page frame address"

The default page frame address (D000) is invalid. Make sure that no other driver is using this area of memory.

"Decimal number input overflow error"

The number of handles specified is an invalid number. Edit your CONFIG.SYS and make sure that the parameter for HANDLES is correctly formatted.

"Error EMS I/O not enabled"

EMS has not been enabled on the Chips F8680 PC/CHIP. Run your system SETUP and make sure EMS has been enabled.

"The EMS hardware is disabled or not functional"

This is a generic error message that is accompanied by another error message. This means that for some reason (given by the other error message) the EMS hardware is not operating properly.

TRANSFER PROGRAM

TRANSFER is a file exchange utility that allows embedded systems to upload and download files over a serial link. TRANSFER uses BIOS calls to send and receive bytes. This allows TRANSFER to operate on any system with a BIOS.

TRANSFER executes on a target system. The program running on the host system may be either COMM, the serial communications utility, or another copy of TRANSFER.

TRANSFER may be used to transfer files via the console, assuming that the console is implemented via a serial port. In this case, TRANSFER uses BIOS interrupts 10H function E, and 16H function 0.

The TRANSFER protocol used by both COMM and the TRANSFER program is referred to as Xmodem by the communications community.

In order to move a file between systems, run TRANSFER on the target system. Either TRANSFER or COMM may be run on the host PC. If COMM is running on the host PC, press the PgUp key on the PC to tell COMM to send a file to the target system. COMM will prompt for the file name and the protocol for TRANSFER. Specify the Xmodem protocol. If you are using TRANSFER on the host, select the COM port, the baud rate, and specify either send or receive.

TRANSFER is run as follows:

```
A:>TRANSFER {Options} FileName
```

The options to TRANSFER start with a slash (/) and are followed with a letter. These options tailor the operations of TRANSFER. The *FileName* argument specifies the file to be uploaded or downloaded. A path and a drive may precede the actual file name. Wildcards are not allowed in the *FileName* argument.

The /B*rate* option allows the user to set the baud rate. The *rate* number may be 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200. This causes the TRANSFER to use a COM1, whether or not the /COM is specified. The default baud rate is 9600.

The /COM# option causes TRANSFER not to use the console and allows the user to set the COM port. The COM number (#) may be either 1 or 2. This option causes TRANSFER to use the serial port BIOS interrupt 14.

The /R option causes TRANSFER to receive a file.

The /S option causes TRANSFER to send a file.

Examples:

```
A:>transfer /r B:file.dat
```

This example causes TRANSFER to receive a file via the console. The data of the file will be placed on drive B: in a file named *file.dat*.

```
A:>transfer /s /B1200 /COM1 junk.abc
```

This example causes TRANSFER to send the file *junk.abc* over COM1 at 1200 baud.

Example:

```
          Target System Command
A:>TRANSFER /r B:VI.EXE
          Host PC System Command
A:>TRANSFER /s C:\BIN\ED.EXE
```

This example shows using TRANSFER on both the host PC and the target system. The file ED.EXE is being sent from the host PC to the target system. The file received on the target system will be called VI.EXE.

COMM PROGRAM

The COMM communications program provides the ability to communicate with the remote ROM-DOS system. Using COMM, you may also communicate with the Datalight BBS. COMM supports Xmodem file transfer, autodialing, and simple terminal emulation.

Command Line Options

All command line options must be separated by a space.

/B#	Set the baud rate to # on startup. The available baud rates are 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200.
/COM#	Set the communications port to #. COM1 and COM2 are supported.
/8N1	Set the serial port to 8 data bits, no parity, 1 stop bit.
/7E1	Set the serial port to 7 data bits, even parity, 1 stop bit.

No other parameters for data bits besides 8N1 and 7E1 are currently supported.

Environment Variable

An environment variable COMM is supported, which may set certain communications parameters. The switches are identical to the command line options.

Example:

```
SET COMM= /COM2 /B2400 /7E1
```

This sets COMM to start up using COM2, at 2400 baud, with 7 data bits, even parity, and 1 stop bit. If new options are specified on the command line, they override the environment variable settings. Invalid options are ignored.

Commands

Most commands are entered by pressing an Alt-letter combination. Some commands take effect immediately (such as changing the baud rate), while others require further information (such as a file name) before continuing.

Esc is the general abort key. If you do not wish to execute a command, or want to stop a command while it is going (such as a file transfer), simply press the Esc key and you will be back in terminal mode.

Alt-B	Set the baud rate. This command toggles between all the available baud rates. Continue to press Alt-B until the baud rate you wish appears on the status line.
Alt-C	Clear the screen.
Alt-D	Autodial. This command allows you to type in a number to autodial; you may press return to redial the previous one. Esc aborts the command.
Alt-E	Toggle echo (duplex). Pressing Alt-E will toggle the duplex between full (echo off) and half (echo on).
Alt-H	Hang up. If the modem is capable of hanging up with an ATH0 command, the line will be disconnected.
Alt-P	Set parameters. This command toggles through the available parameters. Esc will abort the command.
Alt-T	Toggle CR/LF. When this is enabled, pressing the Enter key will generate a CR/LF instead of just a CR.
Alt-X	Exit the program. This command does not drop the carrier, so use this command if you need to do MS-DOS operations while on-line. You can run COMM again, without losing the carrier, and continue with telecommunications.
PgUp	Upload a file. This command sends a file to a remote computer, giving you the option of either Xmodem or ASCII file transfer protocols. Esc will abort at any time during the transfer.
PgDn	Download a file. This command receives a file using the Xmodem or ASCII file transfer protocol. Esc will abort at any time during the transfer.

Terminal Emulation

Currently, COMM automatically supports a subset of the ANSI Escape codes. This emulation cannot be turned off and is the only terminal emulation available. It should, however, suffice most needs since the emulation includes such features as cursor positioning and erase to the end of the line and/or page. We will not attempt to explain here how the ANSI Escape codes work, but if you understand them, the codes supported are A, B, C, D, H, J, and K.

CHAPTER 4

Power Management

The CMF8680 has sophisticated power management which features three power saving modes: drowse, sleep, and suspend. You can program which modes are active in the configuration setup program, RTDMOD. Suspend can also be entered using the PWRUP pin on utility connector P4 (see Chapter 1 for connector description and Chapter 3, Power Management Setup discussion for operation using the PWRUP pin). This chapter describes how these modes are executed. Table 4-1 summarizes the power saving features and Figure 4-1 shows how the power management is configured.

DROWSE

Drowse uses the Performance Control feature of the F8680 CPU to reduce power consumption. Performance Control adds delays of 1 to 127 CPUCLK cycles (dummy cycles) between each CPU instruction cycle. The CPU and the system continue to run at normal clock frequency, but the CPU cycle takes place at a slower rate. The power saving arises from less memory and peripheral accesses over time.

SLEEP

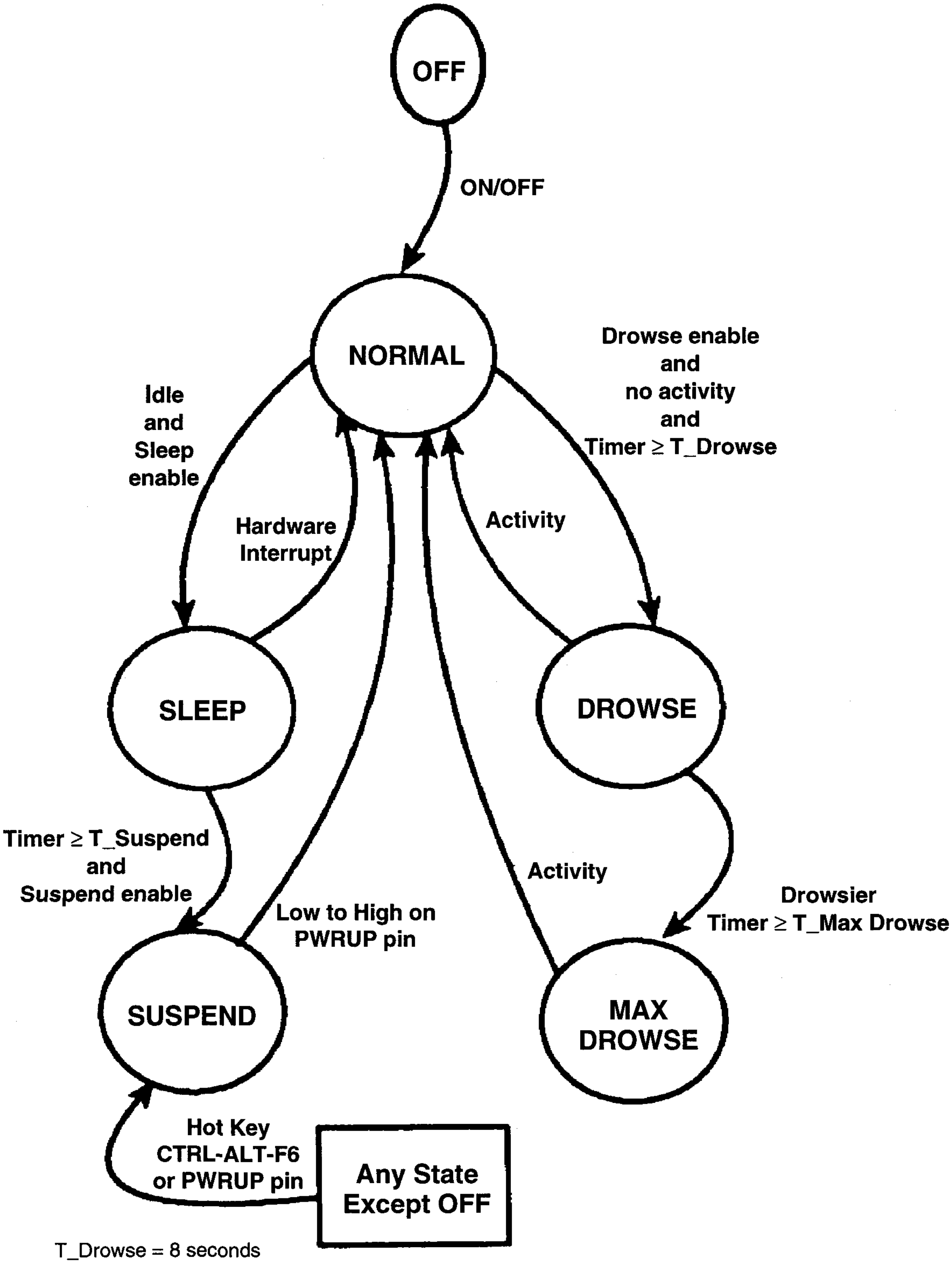
The processor enters the sleep mode by executing a HLT instruction in non-SuperState R mode with interrupts enabled. The sleep mode is transparent to the BIOS, DOS, and any application programs because transitions out of sleep mode occur on any enabled hardware interrupt or Superstate hardware interrupt.

SUSPEND

The suspend mode occurs when the system is turned off except for the portions that need to keep the critical information. The 32 kHz oscillator continues to provide the clock to the CPU's internal 32-bit counter and provides the timing clock for memory refresh. Upon resume, the OSCPWR signal is automatically turned on and the system begins executing instructions where it left off (not a system reset). The ~SLEEPY signal that is available at the utility connector P4-10 is buffered OSC PWR.

Table 4-1 — Power Saving Features

Power Savings Mode	F8680 Features Used
Drowse Mode	Performance reduction technique CPU clock divide SuperState interrupt trap SuperState I/O trap SuperState I/O count
Sleep Mode	Halt of CPU clock SuperState interrupt trap Superstate timer tic
Suspend Mode	Dedicated suspend state machine Dedicated resume input signal Separate power planes for CPU core and PAD Programmable on/off of F8680 internal modules Programmable tristate/active of PC/104 bus and PCMCIA Very low standby current


OFF
ON/OFF
NORMAL
Idle and Sleep enable
Hardware Interrupt
Activity
Drowse enable and no activity and Timer ≥ T_Drowse
SLEEP
DROWSE
Timer ≥ T_Suspend and Suspend enable
Low to High on PWRUP pin
Activity
Drowsier Timer ≥ T_Max Drowse
SUSPEND
MAX DROWSE
Hot Key CTRL-ALT-F6 or PWRUP pin
Any State Except OFF

T_Drowse = 8 seconds
T_Suspend = 2 minutes
T_Max Drowse = 4 * T_Drowse
No activity: no INT, no I/O, no video
Idle: no KB activity
Max Drowse: Max performance control and CLK/4

Fig. 4-1 — Power Management States

CHAPTER 5

Using the Solid State Disk

BUILDING A ROM DISK

ROMDISK is a utility for creating a disk file that represents a ROM disk. A ROM disk contains all the standard parts of a disk that reside on a floppy or hard disk. Each of these disks contains a boot sector, a File Allocation Table and a root directory, and the files selected to be included on the disk. The ROMDISK utility supports ROM disks up to 32 Megabytes in size, sub-directories, and the creation of RXE programs on the ROM disk.

A ROM disk can be used by **ROM-DOS** for accessing the user application in a diskless system. This *disk* is much the same as a RAM disk used under DOS, except that it is *read-only* and always resides in ROM. The ROMDISK image is built using the ROMDISK.EXE utility included in your CMF8680 utility software package.

Running ROMDISK

You can run the ROMDISK utility at the DOS command line by typing ROMDISK with or without parameters.

When it is run without parameters, ROMDISK displays a summary of parameters and options that looks like this:

```
C:\ROMDOS>ROMDISK

ROMDISK v5.0
Copyright (c) 1989-1992 by Datalight

Usage = ROMDISK <filespec> [outfile] {options}

Options
   /E          Do not place extended address in Intel HEX files
   /F#         Set fill byte (default=FFH)
   /H[#]       Output a hex file (default ROMDISK.HEX)
   /I[#]       Output an image file (default ROMDISK.IMG)
   /R#         Chose an interrupt # for RXEs (default=90H)
   /S          Recurse into subdirectories
   /T          Test mode only - don't build ROM disk
   /V "str"    Volume label (default "ROM DISK")
   /Z#         Set Sector Size (default 128 bytes)

Example
        C>ROMDISK ..\romdir disk2 /s/iE000
```

The syntax of ROMDISK is ..

```
ROMDISK <filespec> [outfile] {options}
```

The ROMDISK.EXE program allows you to produce a ROM disk binary image or Intel HEX file. This image file is later burned into ROM to create a "ROM disk". The size of the ROM disk is only limited by the number and size of files placed on the ROM disk and the 1 Megabyte user EPROM.

The ROM disk will have the contents of a standard disk including a boot record, FAT, file directory and data. The sector size, which defaults to 128, may be set by entering the sector size on the ROMDISK command line. There is no limit to the number of files that may be placed on a ROM disk.

Place all of the files to be included on the ROM disk in a directory. Your directory may contain subdirectories. The main directory that you create will become the root directory on the ROM disk. All subdirectories will remain at their original level. You must use the "/S" option if you wish to transfer subdirectories to the ROM disk. The ROM disk syntax for *<filespec>* may include a path.

ROMDISK.EXE is run by typing the program name at the DOS prompt.

This example places the contents of the directory "\tmp", including subdirectories, in the image file "\ROM-DOS\disk.img".

Example:

```
C:\ROM-DOS> ROMDISK \tmp \ROMDOS\disk.img /s
ROM Disk built from C:\TMP\*.*
Placed in file c:\ROMDOS\disk.IMG

 102784          bytes total ROM disk size
   128           bytes in boot sector
  1280           bytes in 10 FAT sectors
   256           bytes in root directory
   256           bytes in 2 directories
 100864          bytes in 6 user files
     0           bytes available on disk

   128           bytes in each of 803 sectors
```

The file produced by ROMDISK.EXE defaults to an image file at address 0x000. This file must be burned into the user's EPROM. The ROM disk must always start at address 0000.

ROMDISK Options

The ROMDISK utility has options to configure its operation. The options are started with a slash (/) and followed with a letter and possibly a number.

The /E option causes extended records not to be placed in the Intel Hex output. These records are placed in a Hex file by default.

The /F option allows the user to set the fill bytes. The default is a fill byte of 0xFF. The number following the /F option is assumed to be in hexadecimal format.

The /H option causes a Hex file to be produced. The optional number following the /H option specifies the actual address of the start of the ROM disk. The start address is required for ROMable EXE files. The default address is 0x000. The CMF8680 requires all ROM disks to start at address 0x0000, and the user EPROM can contain only one ROM disk.

The /I option causes an image file to be produced. The optional number following the /I option specifies the actual start address of the start of the ROM disk. The start address is required for ROMable EXE files. The default address is 0x000.

The /R# specifies the RXE interrupt number. The default number is 90 Hex.

The /S option causes ROM disk to include sub-directories found in the source sub-directory selected to include on the ROM disk image.

The /T option is the test option. This option causes the ROMDISK utility to display statistics on the ROM disk but not to actually create the image file or hex. This is useful when you need to make sure that the required files fit in the available space.

The /V "str" option allows you to set the volume label to something other than "ROMDISK". The volume label string can be up to 11 characters and must be in quotes.

The /Z# option allows you to specify the sector size of the ROM disk. The default sector size is 128 bytes. Legal values for this option are 128, 256, and 512.

NOTE: /C is not supported. Use an .EXE file compression utility if a compressed disk is required. To support this, the system would not be able to boot from floppy or hard disk and would use 6K more ROM and 12K more RAM.

BURNING A ROM DISK INTO PROM

Once all options and devices have been set to your liking, it is time to burn it into ROM.

It is impossible to describe all PROM burners here, so the contents of the PROMs shall be described instead.

The file created by ROMDISK (RDISK1.HEX in this example) is:

RDISK1.HEX	The ROM Disk image made by ROMDISK

The file listed above has the .HEX extension for Intel Hex files. It could just as easily be an .IMG binary image file.

ROMDISK.HEX may be anywhere from 1K in size to just under 1 Megabyte. A practical limit is usually less than 512K. The ROM disk allows for booting on a completely diskless system. This file is burned into the EPROM starting at location 0.

CHAPTER 6

CM102 utilityModule: PCMCIA Memory and Keypad Scanning

The CMF8680-2 with the optional CM102 68-pin PCMCIA and keypad scanning connector supports PCMCIA Release 1.0, SRAM, ROM, and EPROM memory cards. These cards can be used with the CMF8680 system in the CM102 utilityModule. Up to two CM102 utilityModules are supported by the CMF8680-2. They operate as floppy drives and are configurable in the system as drive A: or drive B:. Through the PCMCIA interface, you can format, read, and write (write to SRAM cards only) PCMCIA memory cards. In addition, the CMF8680-2 and CM102 support up to 16 x 8 X-Y keypads.

This chapter describes the CM102 and its functions in the CMF8680-based system.

CM102 CONNECTION

The CM102 stacks directly onto the CMF8680 (see Chapter 1).

DIP SWITCH SETTINGS

Switch S1, a four-position DIP switch on the CM102 module, must be configured for correct operation of the CM102. Sections 1 through 3 control the PCMCIA selection. Section 4 is for keypad enable/disable. Table 4-1 shows the settings for switches 1 through 3 on S1. Make sure you use a valid setting of these switches. Note that 0 = CLOSED, and 1 = OPEN. If two CM102's are installed in the system, one must be configured as Drive A:, the other as Drive B:, and only one module can have the keypad scanning feature enabled.

Table 4-1 — CM102 Switch Settings

Configuration	S1-1	S1-2	S1-3
Not Allowed	0	0	0
Floppy A:	0	0	1
Memory	0	1	0
Not Allowed	0	1	1
Not Allowed	1	0	0
Floppy B:	1	0	1
Memory	1	1	0
Not Allowed	1	1	1

Switch S1-4 is the keypad enable/disable. When this switch is set to 0 (CLOSED), the keypad is enabled. When set to 1 (OPEN), the keypad is disabled.

PCMCIA

Like a floppy disk, a blank SRAM PCMCIA card must be formatted prior to use. The FORMAT.EXE program supplied with the CMF8680 performs this function.

After the card has been formatted, it can be read or written like a floppy. Most PCMCIA cards have a write protect switch. If the switch is in the write protect position, the system will not be able to write the card.

While the system cannot program ROM/Flash EPROM cards, it can read them.

PCMCIA cards can be removed while the system is on without damage to either the system or the card.

KEYPAD SCANNING

The F8680 chip and the BIOS of the CMF8680 provide hardware and software support for the X-Y keypad scanning feature on the CM102. You can program up to 16 rows by 8 columns in an X-Y matrix.

The keypad matrix can be any combinations of rows and columns up to 16 rows by 8 columns, such as the simple 4 x 4 matrix shown in Figure 4-1. Whenever a key is pressed, a connection is made between the column and the row to which the keyswitch is connected.

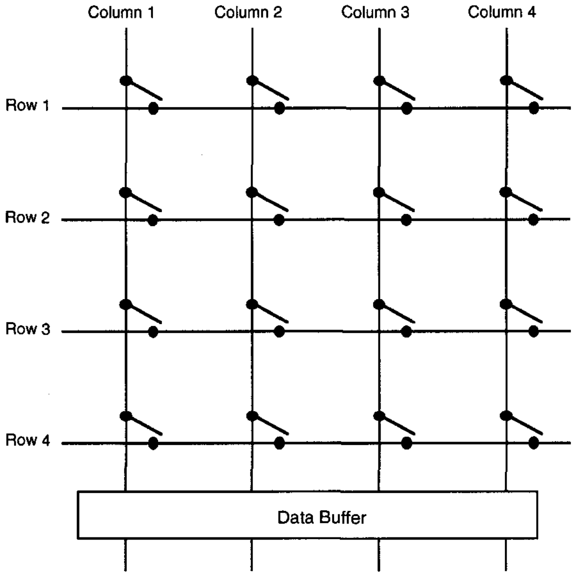


Fig. 4-1 — Simple 4 x 4 Keypad Switch Matrix

To ensure accurate translation of keystrokes, the CMF8680 BIOS scans all keys on a regular basis and transfers the scan code data to the system. The BIOS performs switch debouncing and ghost key detection. The BIOS also implements 2-key rollover, where two is the maximum number of keys that can be pressed simultaneously on the keypad before it is not possible to correctly detect other keys being pressed. Figure 4-2 shows the scanning algorithm used by the CMF8680 with the CM102's scanning feature enabled.

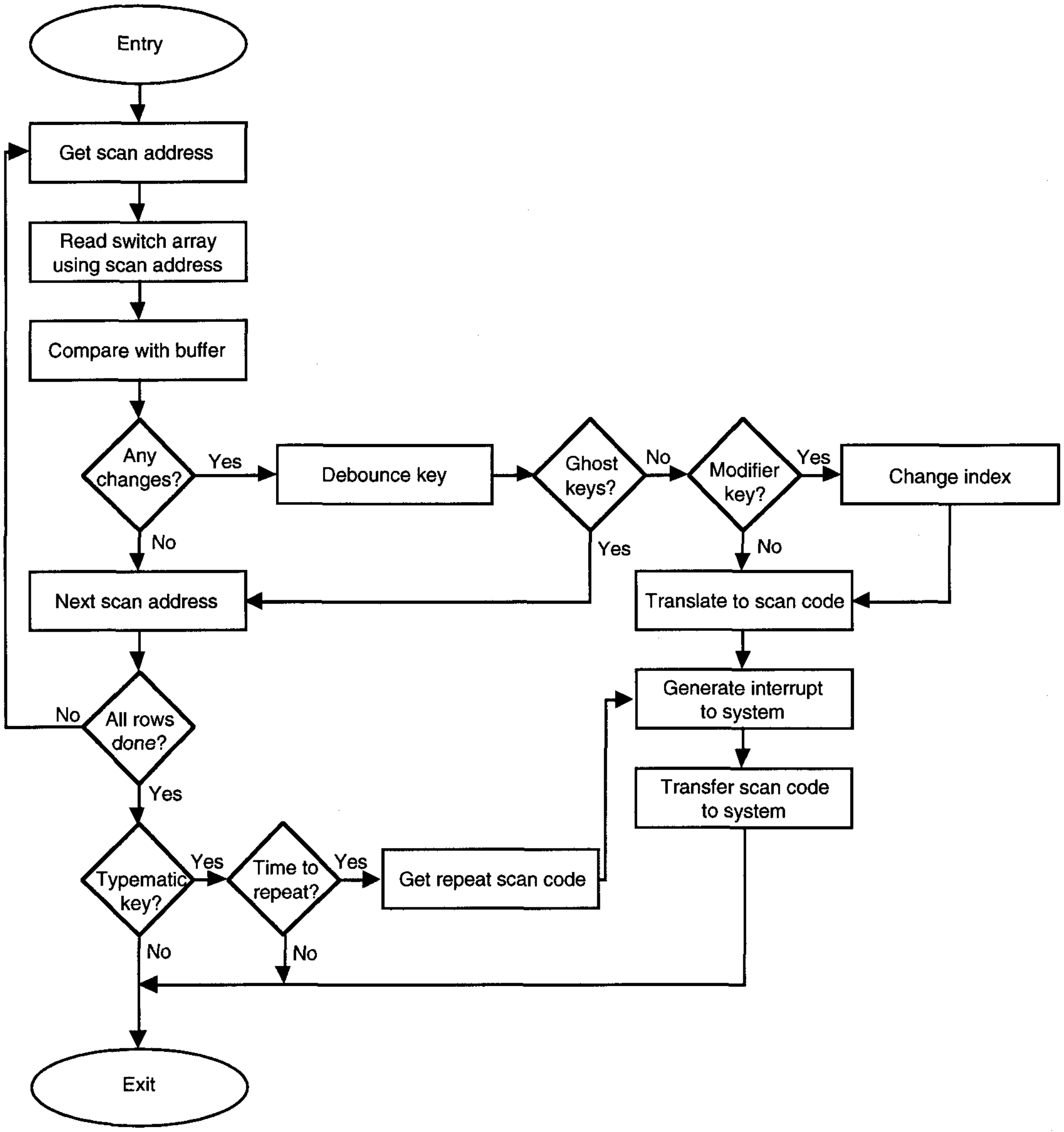
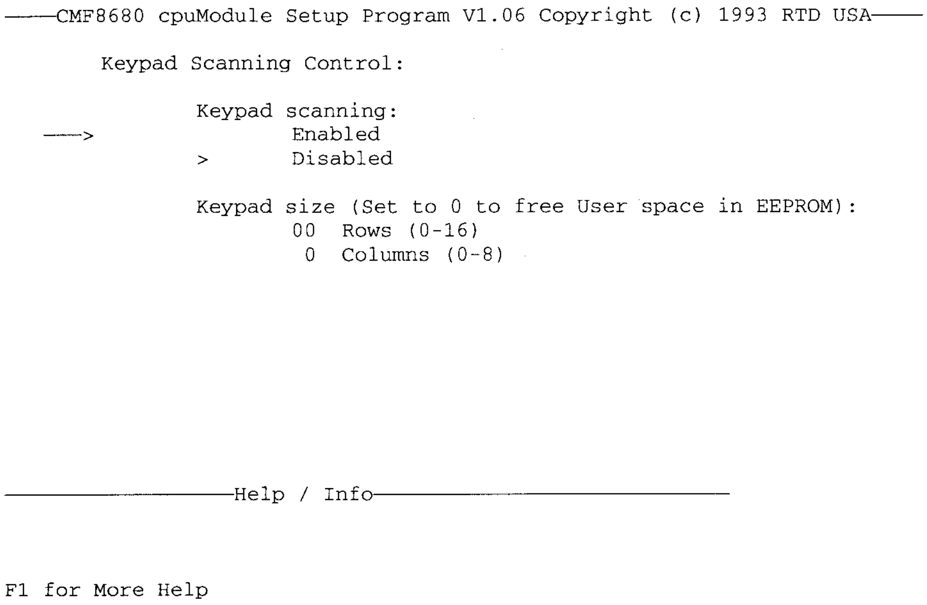


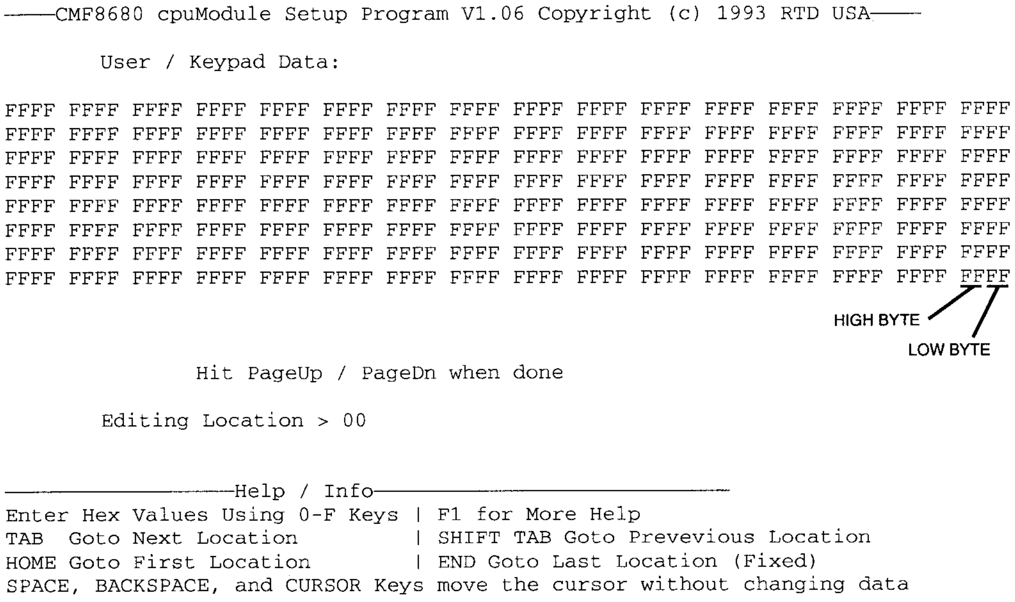
Fig. 4-2 — Scanning Algorithm

The scan codes are programmed into the 2K bit area of the EEPROM available to the user. Programming is done from the system configuration program, RTDMOD.EXE. To use the keypad scanning feature, run RTDMOD.EXE, call up the screen shown in Figure 4-3, enable keypad scanning, and set up the number of rows and columns you will need for your keypad matrix. Then, go to the screen shown in Figure 4-4 and enter the scan codes from Table 4-2 to define the keys. Note that codes 01h through 53h are standard XT keyboard codes. For a 16 x 8 matrix, 128 keystrokes can be defined. By defining an Fn key (or Ctrl-Alt) in one of these positions, an additional 128 keystrokes can be defined, for a total of 256 keys. The Fn key is a typical modifier key found on notebooks and palmtops which extends the character set when pressed along with a given key. The following example will show you how to program a matrix.



```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

       Keypad Scanning Control:

              Keypad scanning:
  ——>                Enabled
              >      Disabled

              Keypad size (Set to 0 to free User space in EEPROM):
                     00  Rows (0-16)
                      0  Columns (0-8)




—————————————————Help / Info——————————————————————————


F1 for More Help
```

Fig. 4-3 — Keypad Scanning Setup



```
———CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA———

       User / Keypad Data:

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF


              Hit PageUp / PageDn when done

       Editing Location > 00


—————————————————Help / Info——————————————————————————
Enter Hex Values Using 0-F Keys | F1 for More Help
TAB  Goto Next Location         | SHIFT TAB Goto Prevevious Location
HOME Goto First Location        | END Goto Last Location (Fixed)
SPACE, BACKSPACE, and CURSOR Keys move the cursor without changing data
```

Fig. 4-4 — Keypad Scanning Values Screen

Table 4-2 — Keypad Scan Codes (in hex)

Scan Code	Base Case	Upper Case	Scan Code	Base Case	Upper Case	Scan Code	Base Case	Upper Case
01	Esc	Esc	2C	z	Z	83	Pad 3	N/A
02	1	!	2D	x	X	84	Pad 4	N/A
03	2	@	2E	c	C	85	Pad 5	N/A
04	3	#	2F	v	V	86	Pad 6	N/A
05	4	$	30	b	B	87	Pad 7	N/A
06	5	%	31	n	N	88	Pad 8	N/A
07	6	^	32	m	M	89	Pad 9	N/A
08	7	&	33	,	<	8A	Pad Del	N/A
09	8	*	34	.	>	8B	Pad BkSlsh	N/A
0A	9	(	35	/	?	8C	Pad +	N/A
0B	0	)	36	Rt Shift	-	8D	Pad -	N/A
0C	-	_	37	*	PSc	8E	Pad *	N/A
0D	=	+	38	Alt	-	8F	Pad NumLk	N/A
0E	Bspace	Bspace	39	Space	Space	90	Pad Enter	N/A
0F	Tab	-	3A	Caps Lock	-	91	Fn + F1 (reserved)	
10	q	Q	3B	F1	N/A	92	Fn + F2 (reserved)	
11	w	W	3C	F2	N/A	93	Fn + F3 (reserved)	
12	e	E	3D	F3	N/A	94	Fn + F4 (inverse video)	
13	r	R	3E	F4	N/A	95	Fn + F5 (toggle sleep)	
14	t	T	3F	F5	N/A	96	Fn + F6 (suspend)	
15	y	Y	40	F6	N/A	97	Fn + F7 (Prt scrn)	
16	u	U	41	F7	N/A	98	Fn + F8 (Sys request)	
17	i	I	42	F8	N/A	99	Fn + F9 (Pause)	
18	o	O	43	F9	N/A	9A	Fn + F10 (Break)	
19	p	P	44	F10	N/A	9B	AT-Rt Alt	N/A
1A	[	{	45	NumLock	-	9C	AT-Ctrl	N/A
1B	]	}	46	ScrollLock	-	9D	Fn key	N/A
1C	Enter	Enter	47	Home	N/A	9E	AT-Up	N/A
1D	Ctrl	-	48	Up	N/A	9F	AT-Dn	N/A
1E	a	A	49	PgUp	N/A	A0	AT-Lt	N/A
1F	s	S	4A	Pad -	N/A	A1	AT-Rt	N/A
20	d	D	4B	Lft	N/A	A2	AT-Ins	N/A
21	f	F	4C	Pad 5	N/A	A3	AT-Home	N/A
22	g	G	4D	Rt	N/A	A4	AT-PgUp	N/A
23	h	H	4E	Pad +	N/A	A5	AT-PgDn	N/A
24	j	J	4F	End	N/A	A6	AT-End	N/A
25	k	K	50	Dn	N/A	A7	AT-Del	N/A
26	l	L	51	PgDn	N/A	A8	Fn + Up (Pan up)	
27	;	:	52	Ins	N/A	A9	Fn + Dn (Pan down)	
28	'	"	53	Del	-	AA	Fn + Lt (Pan left)	
29	`	~	80	Pad 0	N/A	AB	Fn + Rt (Pan right)	
2A	Lt Shift	-	81	Pad 1	N/A	AC	Fn + Home(Pan to cursor)	
2B	\	|	82	Pad 2	N/A	AD	Fn + R (Reset SuperState)	

NOTE: Fn, 9D, must be programmed in both the high and low bytes!

Suppose you want to program the keypad shown in Figure 4-5. First you must determine the keypad's layout. In this case, the layout is 4 rows by 3 columns. To program the scan codes, we must first define the matrix in the RTDMOD.EXE configuration program. On the screen shown in Figure 4-3, we would enable scanning and enter 4 rows and 3 columns.

After setting up the matrix size, then go to the keypad scan values screen to enter the scan codes. It will take 12 words to define the 12 keys in our 4 x 3 matrix. The 12 words into which you will enter the scan codes are already highlighted on the scan values screen. Note that when you do not use the entire table, the last code entered is always in the last position on the table. This allows the unused area in the EEPROM to be available for other functions.

As shown in Figure 4-5, we would like to set up our keypad with the numerical values 0 through 9 in the positions shown, and we want to define the Fn key so that when it is pressed, we can use the cursor position arrows, as shown in the right layout. Each word is divided into an upper byte and a lower byte. The lower byte defines the key when no Fn key is pressed with it, and the upper byte defines the key when the Fn key is pressed simultaneously. Table 4-3 shows how to define our keypad. Note that FF is entered when a switch is absent or a key combination is not used. Also note that we have defined the Fn key by entering 9Dh in both the high and low bytes at row 4, column 1. When you enter the scan codes on the RTDMOD.EXE's scan code values screen, the first word in the highlighted section is row 1, column 1, the second word is row 1, column 2, etc.

Now, look at Figures 4-6 and 4-7 to see how this example is set up on the RTDMOD.EXE configuration screens.

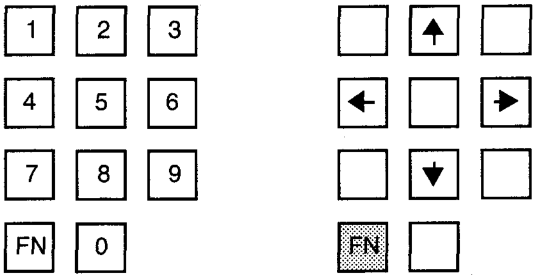
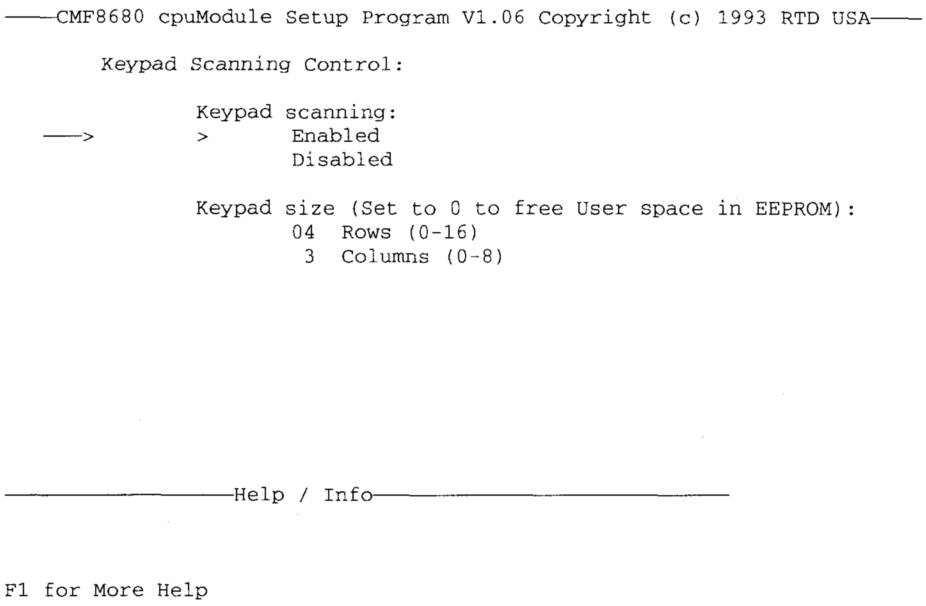


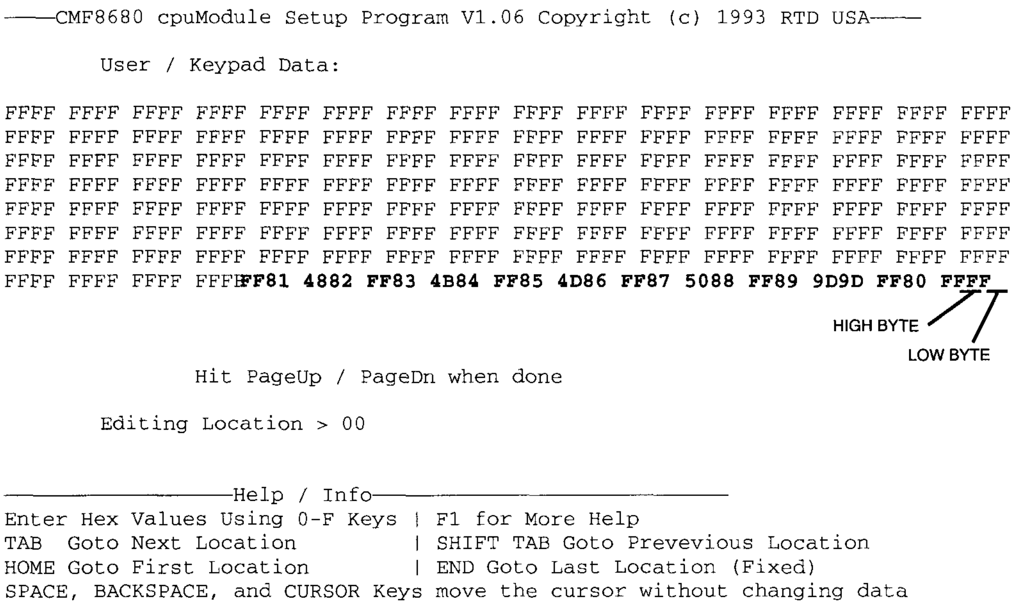
Fig. 4-5 — Simple Keypad

Table 4-3 — Keypad Scan Codes for Figure 4-5						
	Column 1		**Column 2**		**Column 3**	
	High Byte	Low Byte	High Byte	Low Byte	High Byte	Low Byte
Row 1	FF (Skip)	81 (Pad 1)	48 (Up)	82 (Pad 2)	FF (Skip)	83 (Pad 3)
Row 2	4B (Left)	84 (Pad 4)	FF (Skip)	85 (Pad 5)	4D (Right)	86 (Pad 6)
Row 3	FF (Skip)	87 (Pad 7)	50 (Down)	88 (Pad 8)	FF (Skip)	89 (Pad 9)
Row 4	9D (Fn)	9D (Fn)	FF (Skip)	80 (Pad 0)	FF (Skip)	FF (Skip)



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       Keypad Scanning Control:

              Keypad scanning:
   ——>        >      Enabled
                     Disabled

              Keypad size (Set to 0 to free User space in EEPROM):
                     04  Rows (0-16)
                      3  Columns (0-8)





——————————————————Help / Info—————————————————————————


F1 for More Help
```

Fig. 4-6 — Setting Up Keypad Scanning



```
——CMF8680 cpuModule Setup Program V1.06 Copyright (c) 1993 RTD USA——

       User / Keypad Data:

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
FFFF FFFF FFFF FFFFFF81 4882 FF83 4B84 FF85 4D86 FF87 5088 FF89 9D9D FF80 FFFF


              Hit PageUp / PageDn when done

       Editing Location > 00


——————————————————Help / Info——————————————————————————
Enter Hex Values Using 0-F Keys ¦ F1 for More Help
TAB  Goto Next Location         | SHIFT TAB Goto Prevevious Location
HOME Goto First Location        | END Goto Last Location (Fixed)
SPACE, BACKSPACE, and CURSOR Keys move the cursor without changing data
```

HIGH BYTE

LOW BYTE

Fig. 4-7 — Entering the Keypad Scan Codes

APPENDIX A

Specifications

CMF8680 Specifications

General Specifications

Size	3.6"L x 3.8"W x 0.6"H (90 x 96 x 16 mm)
Temperature range, operating	0 to +70°C
Temperature range, storage	-55 to +85°C
Humidity	5 to 95%, non-condensing

Electrical @ 25°C

Power consumption, 2 MB DRAM installed:	
Operational	1.6W @ 14.3 MHz, typical 1.0W @ 7.2 MHz, typical
Drowse	370 mW typical
Sleep	350mW typical
Suspend	100mW typical
Power requirements, 2 MB DRAM installed	+5V ± 5% @ 325 mA typical

PC Functions

Chips & Technologies F8680 16-bit, 14.318 MHz CPU
Programmable CPU clock rates: 14.3, 7.2, 4.77, 3.58 MHz
RTD enhanced BIOS with Datalight® ROM-DOS kernel
2 MB DRAM
3 DMA channels
8 interrupt channels
16-bit IDE hard drive interface
High density floppy controller (360K, 720K, 1.2MB, 1.44MB)
CGA CRT/LCD video interface: 80 x 25 text, 40 x 25 text, 640 x 200 graphics 2-color, 320 x 200 graphics 4-color, 8- or 16-level gray scale for LCD
PC/XT compatible keyboard port
Optional support for X-Y keypad scanning (up to 16 x 8) and for PCMCIA card with CM102 configured as disk drive A: or B:
Speaker port
Battery backed real time clock

I/O Ports

2 RS-232 serial ports
RS-485 serial port
PC/AT compatible enhanced bidirectional parallel port
Stackable 64-pin PC/104 XT bus

Additional On-board Functions

32-pin byte-wide JEDEC memory socket for up to 1 MB non-volatile RAM, PROM, flash EPROM
4K-bit configuration EEPROM for system setup storage, with 2K bits for user and/or keypad layout
Watchdog timer

APPENDIX B

Memory and I/O Maps

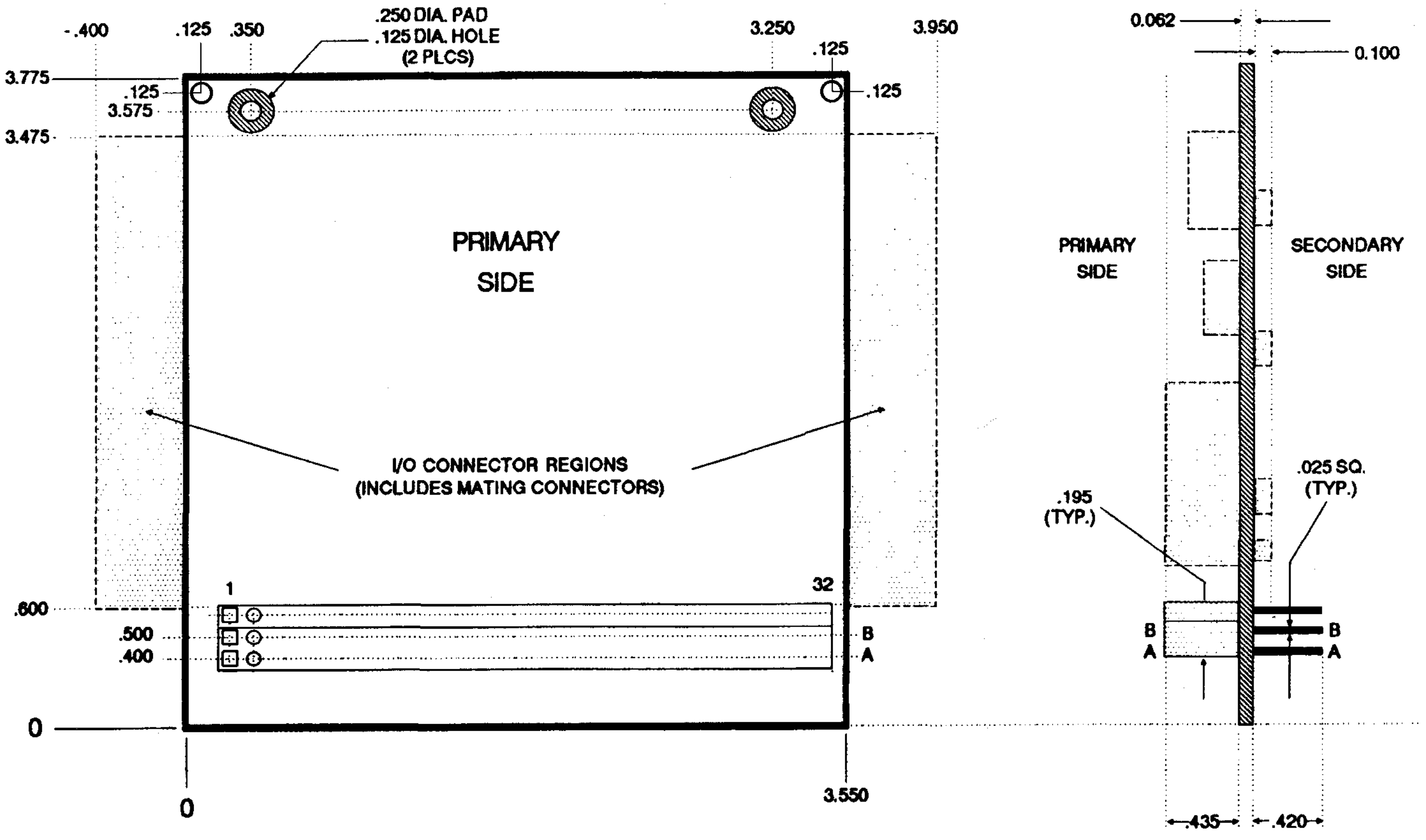
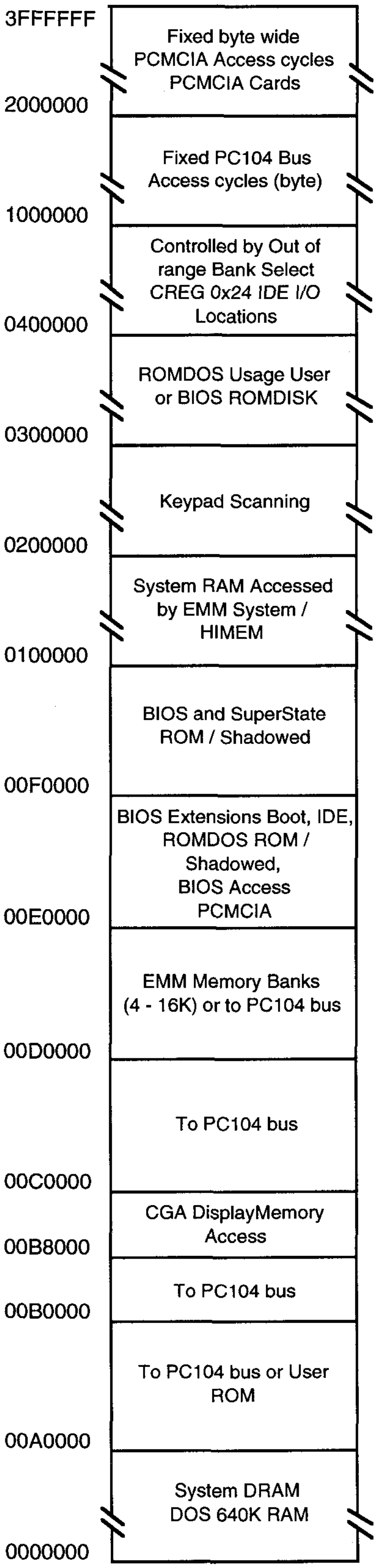


Fig. A-1 — CMF8680 cpuModule Dimensions

CMF8680 cpuModule Memory Map



3FFFFFF	Fixed byte wide PCMCIA Access cycles PCMCIA Cards
2000000	Fixed PC104 Bus Access cycles (byte)
1000000	Controlled by Out of range Bank Select *CREG 0x24 IDE I/O* Locations
0400000	ROMDOS Usage User or BIOS ROMDISK
0300000	Keypad Scanning
0200000	System RAM Accessed by EMM System / HIMEM
0100000	BIOS and SuperState ROM / Shadowed
00F0000	BIOS Extensions Boot, IDE, ROMDOS ROM / Shadowed, BIOS Access PCMCIA
00E0000	EMM Memory Banks (4 - 16K) or to PC104 bus
00D0000	To PC104 bus
00C0000	CGA DisplayMemory Access
00B8000	To PC104 bus
00B0000	To PC104 bus or User ROM
00A0000	System DRAM DOS 640K RAM
0000000	

CMF8680 cpuModule I/O Port Map

I/O Locations	Base
DMA Controller	000-00F
DMA Controller (Image) (can be remapped by VIO)	010-01F
Interrupt Controller	020-021
Interrupt Controller (Image) (can be remapped by VIO)	022-03F
Timer	040-043
Timer (Image) (can be remapped by VIO)	044-05F
Keyboard Interface	060-062
Keyboard Interface (Image) (can be remapped by VIO)	063-07F
DMA Page	080-083
DMA Page (Image) (can be remapped by VIO)	084-09F
PC104 bus	0A0-21F
PC104 bus or Serial Port COM3 721 option	220-227
PC104 bus or Serial Port COM4 721 option	228-22F
PC104 bus	230-237
PC104 bus or Serial Port COM4 721 option	238-23F
PC104 bus	240-277
PC104 bus or Printer PortC 721 option	278-27A
PC104 bus	27B-2DF
PC104 bus or Serial Port COM4 721 option	2E0-2E7
PC104 bus or **Serial Port COM3/**COM4 721 option	2E8-2EF
PC104 bus	2F0-2F7
Serial Port COM2 - (PC/Chip or **721** or PC104 bus)	2F8-2FF
PC104 bus	300-337
PC104 bus or Serial Port COM3 721 option	338-33F
PC104 bus	340-377
PC104 bus or **Printer PortB** 721 option	378-37A
PC104 bus	37B-3AF
CGA Graphics Controller - (PC/Chip or PC104 bus)	3B0-3DF
PC104 bus	3B0-3BB
PC104 bus or Printer PortA 721 option	3BC-3BE
PC104 bus	3BF-3CF
CGA Undefined (can be remapped by VIO)	3D0-3D1
CGA Index Register	3D4
CGA Data Register	3D5
CGA Undefined (can be remapped by VIO)	3D6-3D7
CGA Mode Control Register	3D8
CGA Color Select Register	3D9
CGA Status Register	3DA
CGA Clear Light Pen Strobe	3DB
CGA Set Light Pen Strobe	3DC
CGA Undefined (can be remapped by VIO)	3DD-3DF
PC104 bus	3E0-E7
PC104 bus or Serial Port COM3 721 option	3E8-3EF
Floppy Controller on 721 and 721 Configuration Regs	3F0-3F7
Serial Port COM1 - (PC/Chip or 721 or PC104 bus)	3F8-3FF

Bold - Default Settings & Default Part used to provide I/O

Notes:* Some of the Images below 0A0 do go out to the PC104 bus in the current BIOS implementation and they always can be reprogrammed by the user, if desired.

APPENDIX C

Connector Pin Assignments

CMF8680 CONNECTORS

P1 — PC/XT EXPANSION BUS (PC/104 BUS)

PIN	SIGNAL
B1	GND
B2	+RESET
B3	+5 VDC
B4	+IRQ2
B5	–5VDC
B6	+DRQ2
B7	–12 VDC
B8	RESERVED
B9	+12 VDC
B10	GND
B11	–MEMW
B12	–MEMR
B13	–IOW
B14	–IOR
B15	–DACK3
B16	+DRQ3

PIN	SIGNAL
B17	–DACK1
B18	+DRQ1
B19	–DACK0
B20	CLOCK
B21	+IRQ7
B22	+IRQ6
B23	+IRQ5
B24	+IRQ4
B25	+IRQ3
B26	–DACK2
B27	+T/C
B28	+BALE
B29	+5 VDC
B30	OSC
B31	GND
B32	GND

PIN	SIGNAL
A1	IOCHCK
A2	SD7
A3	SD6
A4	SD5
A5	SD4
A6	SD3
A7	SD2
A8	SD1
A9	SD0
A10	IOCHRDY
A11	AEN
A12	SA19
A13	SA18
A14	SA17
A15	SA16
A16	SA15

PIN	SIGNAL
A17	SA14
A18	SA13
A19	SA12
A20	SA11
A21	SA10
A22	SA09
A23	SA08
A24	SA07
A25	SA06
A26	SA05
A27	SA04
A28	SA03
A29	SA02
A30	SA01
A31	SA00
A32	GND

P2 — PCMCIA EXPANSION BUS

PIN	SIGNAL
1	GND
2	RD8
3	RD9
4	RD10
5	RD11
6	RD12
7	RD13
8	RD14

PIN	SIGNAL
9	RD15
10	ADR20
11	ADR21
12	ADR22
13	ADR23
14	ADR24
15	ADR25
16	MCBAT1

PIN	SIGNAL
17	MCBAT2
18	–MCCD1
19	–MCCD2
20	–MCCE2
21	–MCCE1
22	MCRDY
23	MEMCDWP
24	OE0-

PIN	SIGNAL
25	WE0-
26	REFRESH-
27	–REG
28	VPP CTRL
29	CARDB
30	˜SLEEPY
31	N/C
32	N/C

P3 — POWER CONNECTOR

PIN	SIGNAL
1	GND
2	+5 V
3	N/C
4	+12 V

PIN	SIGNAL
5	–5 V
6	–12 V
7	GND
8	+5 V

CMF8680 CONNECTORS (cont'd)

P4 — UTILITY CONNECTOR

PIN	SIGNAL
1	+SPKR
2	GND
3	RESET SW-
4	PWRUP
5	–KBDATA
6	–KBCLK
7	GND
8	KBDPWR
9	+3.6 VDC battery
10	˜SLEEPY

P5 — RS-485 SERIAL PORT

PIN	SIGNAL
1, 2	B
3, 4	A

P6 — AT ENHANCED BIDIRECTIONAL PARALLEL PORT

PIN	SIGNAL
1	STROBE
2	AUTOFD
3	PD0
4	ERROR
5	PD1
6	INIT
7	PD2
8	SLCTIN
9	PD3
10	GND
11	PD4
12	GND
13	PD5
14	GND
15	PD6
16	GND
17	PD7
18	GND
19	ACK
20	GND
21	BUSY
22	GND
23	PE
24	GND
25	SLCT
26	GND

P7 — CGA GRAPHICS/LCD FLAT PANEL DISPLAY INTERFACE

PIN	CRT SIGNAL	LCD SIGNAL
1	GND	Not Used
2	INTENSITY	DOT 3
3	GND	GND
4	DOTCLOCK	DOTCLOCK
5	RED	DOT 2
6	H SYNC	LATCH PULSE
7	GREEN	DOT 1
8	V SYNC	FIRST LINE MARKER
9	BLUE	DOT 0
10	Not Used	MCLK

CMF8680 CONNECTORS (cont'd)

P8 — HIGH DENSITY FLOPPY DISK DRIVE INTERFACE

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
1	GND	10	¯MTR0	19	GND	27	GND
2	MP/LO	11	GND	20	STEP	28	WRPRT
3	GND	12	¯DRV1	21	GND	29	GND
4	C	13	GND	22	WDATA	30	RDATA
5	GND	14	¯DRV0	23	GND	31	GND
6	C	15	GND	24	WGATE	32	HDSEL
7	GND	16	¯MTR1	25	GND	33	GND
8	INDEX	17	GND	26	TRK0	34	DSKCHG
9	GND	18	DIR				

P9 — RS-232 SERIAL PORT

PIN	SIGNAL	PIN	SIGNAL
1	DCD	6	CTS
2	DSR	7	DTR
3	RXD	8	RI
4	RTS	9	GND
5	TXD	10	GND

P10 — 16-BIT IDE HARD DRIVE INTERFACE

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
1	RESET	11	D3	21	N/C	31	IRQ5
2	GND	12	D12	22	GND	32	IOCS16-
3	D7	13	D2	23	WR-	33	A1
4	D8	14	D13	24	GND	34	N/C
5	D6	15	D1	25	RD-	35	A0
6	D9	16	D14	26	GND	36	A2
7	D5	17	D0	27	IOCHRDY	37	–CS0
8	D10	18	D15	28	N/C	38	–CS1
9	D4	19	GND	29	DRQ3	39	N/C
10	D11	20	N/C (KEY)	30	GND	40	GND

CMF8680 CONNECTORS (cont'd)

P11 — RS-232 SERIAL PORT

PIN	SIGNAL
1	DCD
2	DSR
3	RXD
4	RTS
5	TXD
6	CTS
7	DTR
8	RI
9	GND
10	GND

CM102 utilityModule P4 CONNECTOR

PIN	SIGNAL
1	ROW 1
2	ROW 2
3	ROW 3
4	ROW 4
5	ROW 5
6	ROW 6
7	ROW 7
8	ROW 8
9	ROW 9
10	ROW 10
11	ROW 11
12	ROW 12
13	ROW 13
14	ROW 14
15	ROW 15
16	ROW 16
17	COLUMN 1
18	COLUMN 2
19	COLUMN 3
20	COLUMN 4
21	COLUMN 5
22	COLUMN 6
23	COLUMN 7
24	COLUMN 8
25	GROUND
26	+5V

APPENDIX D

Application Notes

This appendix contains CMF8680 Application Notes which are currently available from Real Time Devices:

CMF8680-1	Using the CMF8680 Watchdog Timer
CMF8680-2	RDTNVEP.EXE: User Access to Serial EEPROM

APPLICATION NOTE CMF8680-1

Using The CMF8680 Watchdog Timer

Embedded applications often are required to run for a long time and are subject to all kinds of interference, noise, and power supply glitches. The CMF8680 has an onboard watchdog timer that will reset the processor if an application is not operating properly.

The basic principle of the watchdog timer is once the application starts the watchdog timer it must toggle the timer at least once every 1.6 seconds or the watchdog timer will perform a hardware reset. The CMF8680 implementation also allows the user to disable the timer.

Example programs in Turbo C, QuickBASIC, and Turbo Pascal demonstrate this application. The two functions that are used are Toggle_Watchdog_Timer and Disable_Watchdog_Timer. The watchdog timer is connected to an I/O pin on the F8680 CPU. When the pin is tri-stated the timer is disabled. This is the default position after a hardware reset. The pin can be driven high or low by using a Super State R interrupt function. The process is to load the AH with 14H, AL with the value to set, BH with 8CH and BL with 1H.

APPLICATION NOTE CMF8680-2

RTDNVEP.EXE: User Access to Serial EEPROM

RTDNVEP.EXE

This Terminate-Stay-Resident program enables users to access 128 16-bit locations in the configuration EEPROM. These locations are numbered 0 to 127.

To install, type:

```
RTDNVEP [/Ixx]
```

The driver will install itself on interrupt 60H by default. If the /I switch is used the supplied interrupt will be used. For example:

To install on interrupt 66 decimal (42H), type:

```
RTDNVEP /I=66
```

or

```
RTDNVEP /I=0x42
```

To use:

The EEPROM is divided into three areas. The first 2K bits are not accessible to the user and are for system configuration. The last 128 x 16 locations are divided between a user data area and a keypad area. The size of each is determined by the keypad rows and columns settings made in RTDMOD.EXE setup program. Each key defined uses one 16-bit location. The low byte is the scan code for the key without the FN key depressed, and the high byte is the scan code for the key if the FN key is depressed.

The general procedure for accessing the User/Keypad area is to load the processor registers and generate an interrupt.

There are four functions supported. They are: read a location, write user location, write any location, and read keypad rows and columns.

Function 0
To read a location:

AH = 00H
AL = address to read 0 to 127
Generate Interrupt 60H (or as set with /I)

After the interrupt returns, the results are:

AH = 0 if Read was OK, 1 if an error occurred
AL = address read 0 to 127
BX = 16-bit value read

Function 1
To write to a user location:

AH = 01H
AL = address to write 0 to 127 -(Keypad Rows * Keypad Columns)
BX = 16-bit value to write
Generate Interrupt 60H (or as set with /I)

After the interrupt returns, the results are:

AH = 0 if Write was OK, 1 if an error occurred
AL = address written 0 to 127
Note: This function will generate an error if an access to the keypad area is tried.

Function 2
To write to any location:

AH = 02H
AL = address to write 0 to 127
BX = 16-bit value to write
Generate Interrupt 60H (or as set with /I)

After the interrupt returns, the results are:

AH = 0 if Write was OK, 1 if an error occurred
AL = address written 0 to 127

Function 3
To read keypad rows and columns:

AH = 03H
Generate Interrupt 60H (or as set with /I)

After the interrupt returns, the results are:

AH = 0 if Read was OK, 1 if an error occurred
BL = Keypad Rows
BH = Keypad Columns

APPENDIX E

Warranty

LIMITED WARRANTY

Real Time Devices, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from REAL TIME DEVICES. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, REAL TIME DEVICES will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to REAL TIME DEVICES. All replaced parts and products become the property of REAL TIME DEVICES. **Before returning any product for repair, customers are required to contact the factory for an RMA number.**

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED AS A RESULT OF ACCIDENT, MISUSE, ABUSE (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by REAL TIME DEVICES, "acts of God" or other contingencies beyond the control of REAL TIME DEVICES), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN REAL TIME DEVICES. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES ARE EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND REAL TIME DEVICES EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MECHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE DURATION OF THIS WARRANTY. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, THE PURCHASER'S SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL REAL TIME DEVICES BE LIABLE TO THE PURCHASER OR ANY USER FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.