4M22 DISK EMULATOR MANUAL

Version 1.0

Mesa Electronics
4175 Lakeside Drive, Suite #100
Richmond, CA 94806-1950
Tel (510) 223-9272 - Fax (510) 223-9585
E-Mail: tech@mesanet.com - Website: www.mesanet.com

TABLE OF CONTENTS

HANDLING PRECAUTIONS

STATIC ELECTRICITY

The CMOS integrated circuits on the 4M22 can be damaged by exposure to electrostatic discharges. The following precautions should be taken when handling the 4M22 to prevent possible damage.

A. Leave the 4M22 in its antistatic bag until needed.
B. All work should be performed at an antistatic workstation.
C. Ground equipment into which 4M22 will be installed.
D. Ground handling personnel with conductive bracelet through 1 megohm resistor to ground.
E. Avoid wearing synthetic fabrics, particularly Nylon.

INTRODUCTION

GENERAL

The 4M22 is a nonvolatile memory module designed as a replacement for mechanical disk drives. The 4M22 will operate reliably in environments that are unsuitable for standard disk drives. Because the 4M22 requires no maintenance, it is ideal for applications with limited accessibility. On board firmware emulates a standard fixed disk controller, and allows booting from the disk emulator.

The 4M22 can use battery backed-up CMOS RAM for general read-write use, flash EEPROM for read-mostly use, or EPROM for low cost read-only file use. The 4M22 can use EPROM, CMOS RAM, and three types of flash EEPROMs: 12V Intel, 5V AMD and 5V Atmel/SST.

Using flash, the 4M22 is not merely a PROMDISK, but a normal read/write drive. The 4M22 firmware includes a built-in, bootable flash filing system that allows continuous file updates without running out of space, or requiring special packing utilities.

The 4M22 uses six 32 pin JEDEC standard memory devices and has a capacity of 3M byte using RAM, EEPROM, or EPROM. The six sockets are divided into banks of 2 and 4 chips. Each bank can use a different memory type, for example, a 4M22 could be configured with flash EEPROM for updateable program storage and battery backed RAM for frequently rewritten files.

Write inhibit logic and precision power fail sense circuits protect data from inadvertent damage during power-up, power-down, or program malfunction. Disk emulator data integrity can optionally be verified by CRC checking. Utility programs are supplied with the 4M22 for testing flash EEPROM, and RAM disks, and converting the disk emulator memory image to a binary files for EPROM programming.

HARDWARE CONFIGURATION

GENERAL

The 4M22 has some jumper configurable options that must be properly set to match the memory type(s) used and system requirements. Each jumper block will be discussed separately by function. In the following discussions, when the words "up", and "down" are used it is assumed that the 4M22 disk emulator is oriented with its bus connector at the bottom edge of the card.

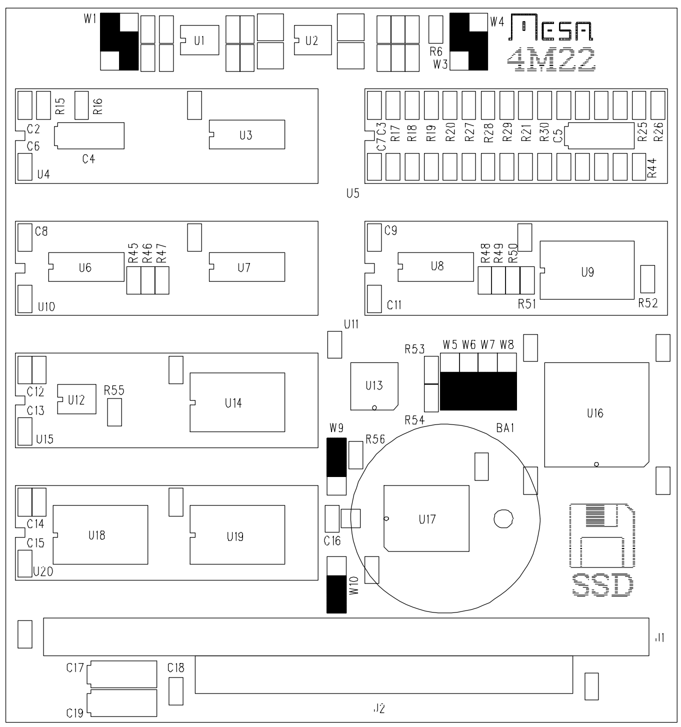
DEFAULT CONFIGURATION

The default configuration of 4M22's, as shipped from the factory, is as follows:

FUNCTION	JUMPER(S)	DEFAULT SETTING
MEMORY TYPE BANK 0:	W1,W2	5V FLASH (up,down)
MEMORY TYPE BANK 1:	W3,W4	5V FLASH (up,down)
FIRMWARE ADDRESS:	W5,W6	C8000H (down,down)
I/O ADDRESS	W7,W8	220H (down,down)
LITHIUM CELL POWER	W9	DISABLED (up)
VPP ENABLE	W10	DISABLED (down)

HARDWARE CONFIGURATION

DEFAULT JUMPER LOCATIONS



HARDWARE CONFIGURATION

MEMORY SOCKETS

The 4M22 has six 32 pin sockets that accommodate JEDEC standard memory devices of various types. These sockets are arranged as two independently configurable banks, Bank 0 and Bank 1. Bank 0 consists of the four sockets on the left hand side of the card, while bank 1 consists of the two sockets on the right hand side of the card.

When installing memory chips, the pin one end (notched end) points to the left. Memory chips that are installed backwards can damage the memory chip, the 4M22 or both.

Some of the utility programs refer to "chip numbers". The following table shows the correspondence between chip numbers and U numbers:

BANK 0:		**BANK 1:**	
U4	CHIP 0	U5	CHIP 4
U10	CHIP 1	U11	CHIP 5
U15	CHIP 2		
U20	CHIP 3		

MEMORY TYPE JUMPERS

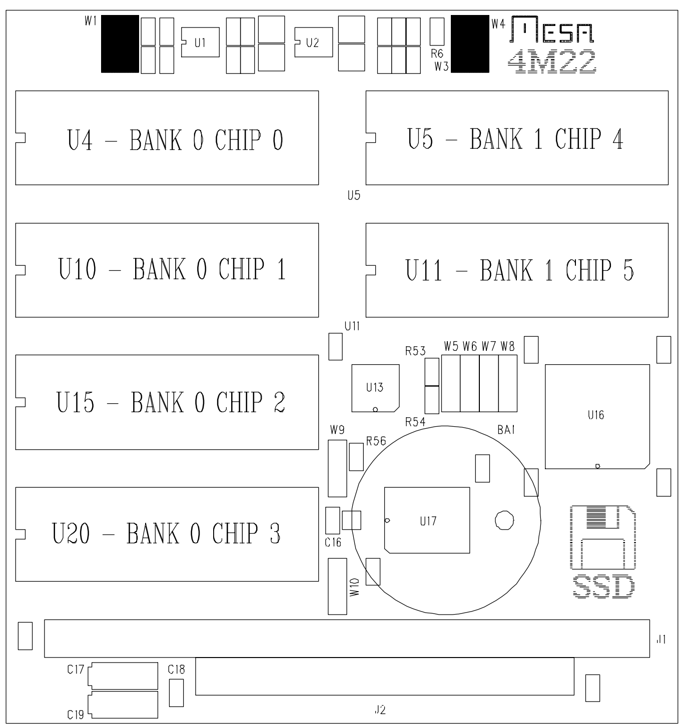
These jumpers are used to configure the Bank 0 and Bank 1 memory arrays for the type of memory device used. Note that bank 0 and bank 1 can be configured independently to allow mixing of memory types on one 4M22 card.

The following table gives correct jumper settings for applicable memory types. The jumper blocks have 3 pins and 2 valid jumper locations, up and down. Jumper blocks W1 and W2 configure Bank 0, while Jumper Blocks W4 and W5 configure Bank 1. ***Make sure that W1 and W3 are up if 5V flash EEPROMs or EPROMs are used or you may damage the memory chips, and/or the 4M22 card.***

BANK 0	**W1**	**W2**
BANK 1	**W3**	**W4**
EPROMS	up	down
STATIC RAMS:	up	up
5V FLASH EEPROMS:up	down	
12V FLASH EEPROMS:	down	down

HARDWARE CONFIGURATION

MEMORY SOCKET LOCATIONS AND TYPE JUMPERS



HARDWARE CONFIGURATION

FIRMWARE START ADDRESS

The firmware on the 4M22 occupies 32K bytes of address space and can be located at 3 jumper selectable locations in system memory. These locations are C8000H and D0000H and D8000. The suggested location is C8000H. You should make sure that the selected firmware start address does not overlap other memory in the system.

Jumpers W5 and W6 select the firmware start address range. The ROM can be disabled by setting both jumpers to the up position. When more than one 4M22 is used in a system, the ROMs should be disabled on all but one 4M22. Disabling the ROM can be also used for debugging purposes.

The following table shows the jumper positions for setting the firmware address.

W5	**W6**	**FIRMWARE ADDRESS**
down	down	C8000H
down	up	D0000H
up	down	D8000H
up	up	ROM DISABLED

I/O PORT ADDRESS

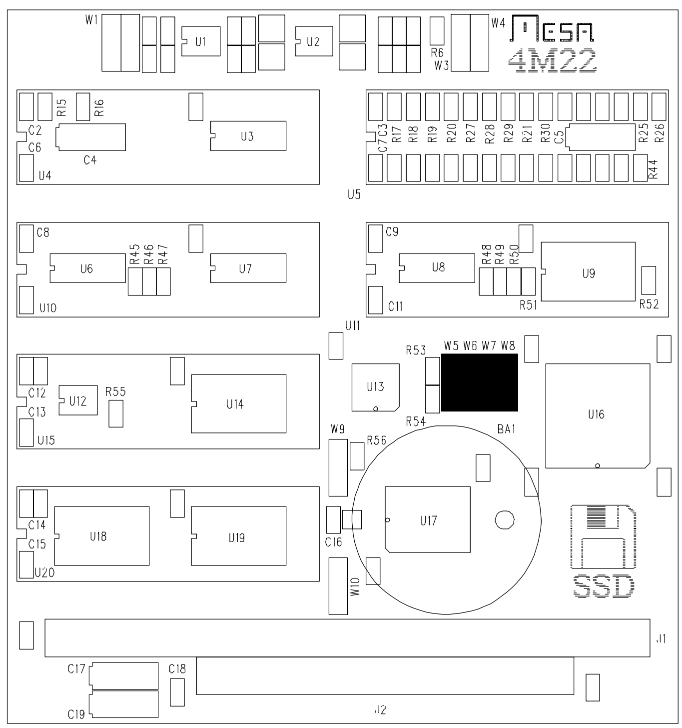
Up to four 4M22 cards can reside in a PC/104 computer system. Each card in a system must have a different I/O address. When a single 4M22 is used, the I/O addresses can be moved to avoid conflicts with existing I/O ports.

Port addresses are set with jumpers W7 and W8. The following table shows the jumper positions for setting the port addresses.

W7	**W8**	**I/O ADDRESS**
down	down	220H
down	up	230H
up	down	240H
up	up	250H

HARDWARE CONFIGURATION

FIRMWARE LOCATION AND I/O PORT LOCATION JUMPERS



HARDWARE CONFIGURATION

VPP ENABLE

When 12V flash EEPROMs are used, VPP power must be enabled. Jumper W10 connects VPP power on the 4M22. When W10 is in the down position, VPP power is disabled. This is the suggested setting when fussing around, as the VPP +12V can destroy 5V only memory chips if the other jumpers are set incorrectly. For use with 12V flash EEPROMS, W10 must be set to the up position. Jumper W10 can be used to write protect 12V flash EEPROMS by setting it in the down position.

The 4M22 firmware BIOS uses a 12V flash part to allow firmware upgrades in place. The firmware BIOS gets it's 12V from the disk emulators switch, therefore, you need to enable VPP power on the card if you need to upgrade the 4M22's BIOS.

VPP REQUIREMENTS

The 4M22 does not generate 12V on card, and therefore depends on the PC/104 bus +12 supply for programming 12V flash parts. These parts typically require a voltage tolerance of +- 5% maximum.

LITHIUM CELL ENABLE

The 4M22 has a Lithium cell for support of battery backed SRAM. The Litium cell is normally disconnected during shipping. If you are using SRAMs you need to enable the backup power by moving W9 to the down position.

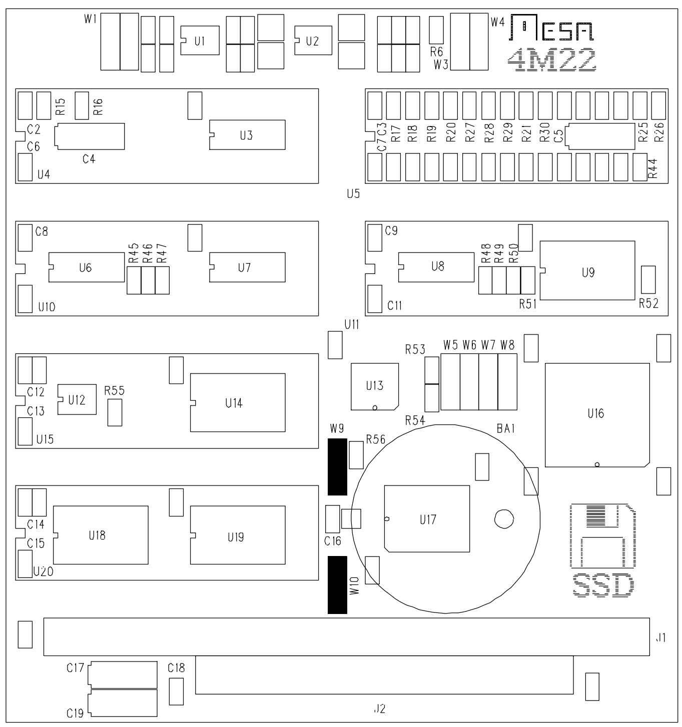
Total Lithium cell current can be measured by removing W9 and placing a current meter across W9's pins 2 and 3. Measuring the backup current is suggested when installing memory chips for the first time. Approximate Lithium cell life (in hours) can be calculated by dividing the mAH capacity of the Lithium cell (280) by the current drain.

Typical Lithium cell current for a 4M22 with two 128K X 8 static RAMS modules should be less than 2 uA.

Removing W9 on a working disk will of course destroy any information stored in SRAMs.

HARDWARE CONFIGURATION

VPP AND LITHIUM CELL JUMPERS



HARDWARE CONFIGURATION

COMPATIBLE CMOS MEMORY DEVICES

5V FLASH EEPROM 128K BYTE:

ATMEL	AT29C010-20PC
SST	PH29EE010-150

5V FLASH EEPROM 512K BYTE:

ATMEL	AT29C040-20PC
SST	PH28SF040-150

12V FLASH EEPROM 256K BYTE: (Requires special BIOS ROM)

INTEL	P28F020-150P1C4
AMD	P28F020-150C4PC

EPROM 128K -512K BYTE:

Any generic JEDEC pinout EPROM 200 nS or faster

STATIC RAM 128K

Any generic JEDEC pinout 128KX8 static RAM 150 nS or faster, low power versions preferred for maximum Lithium cell life.

STATIC RAM 512K

Any generic JEDEC pinout 512KX8 static RAM 150 nS or faster, low power versions preferred for maximum Lithium cell life. Modular RAMs are fine but check backup power requirements.

SOFTWARE CONFIGURATION

GENERAL

When using the disk emulator, the disk type configuration jumpers must be first set to match the type of disk emulator chip(s) used. (See the HARDWARE CONFIGURATION section of this manual). After this is done, the disk emulator need to be initialized before it can be used as a disk by the system.

The 4C22 disk emulator is viewed as a hard disk by system software. This means that in most systems, the first emulated drive will be drive C: , and the next emulated drive will be drive D:

RELIABILITY

In an embedded system environment where a system that won't boot is basically a failed system, it is important to understand some charateristics of the DOS operating system that applies to disk access.

When DOS writes a file, it writes to the FAT and directory areas of the drive (emulated or real). If there is any chance that a system can be reset or power can fail when writing to this disk, all information on the disk could become inaccessable, ***not*** just the file that was being written. The reason is that when DOS writes to a directory or FAT area it always writes a full sector, not just the directory or FAT entry required. If the sector write is not completed, the sector with the directory or FAT entry that was being written will have an invalid CRC. This can affect any file on the drive!

In applications that do frequent disk writes, there are two possible solutions to this problem. The first solution is to disable emulated disk CRC checking. This will make a partially re-written sector readable by the operating system. This will only improve the odds of surviving a power off or reset during a file write, not totally eliminate the problem. Turning off CRC's will also mask possible hardware problems, so is not generally suggested.

The second solution is to configure a two drive system, with a drive (usually C:) used as the software drive, and the other drive (usually D:) used as the data drive. Any files writes during normal operation would be done to the D: drive. If any problem occurs on the D: drive, software on the C: drive can attempt to recover the data, and then re-initialize the D: drive. By adding one more data drive (E:), data files can be alternately written, and verified on separate drives (D: and E:), so stored data can be recovered inalmost all situations.

FLASH WRITE CYCLES

Flash memory chips have a limited write cycle life, and are better suited to program storage than data aquisition and logging files. This is not an absolute rule, as flash chip write cycle lifetimes are improving as the technology develops. Each time a file is changed, the directory and FAT area of the disk is re-written. This is the area of the flash disk that will tend to fail.

When purchasing flash chips, be sure to specify the write or erase cycle life desired. The number of file writes before failure is approximately the write cycle life of the chip divided by four.

The 4M22 firmware can improve the average number of writes till failure figure with certain flash chip types by pre-allocating a few spare sectors. The firmware will then replace failed sectors with these spares until all the spares have been used.

SOFTWARE CONFIGURATION

DISK EMULATOR INITIALIZATION

Before using the disk emulator, it needs to be initialized so that the 4M22 firmware knows the size, chip type, and organization of the disk emulator.

This initialization is done with INITRAMD.EXE. INITRAMD.EXE is supplied on the 4M22 distribution disk. If INITRAMD is run with a /L parameter, it will list the types of disk emulator chips supported by the 4M22 firmware. Each type of disk emulator chip has a corresponding device type number

INITRAMD 'looks' at 4M22 cards simply as an array of memory chips starting with 0. Chip 0 is the first chip in bank 0 of the first 4M22. When multiple 4M22 cards are used, the chip numbering sequence follows the I/O address order of the 4M22 cards. For example if two 4M22s were used in a system, the first card (the card with the lower I/O address) would have chips 0 through 5, and the next card would have chips 6 through 11.

Each time INITRAMD is run, it creates a single emulated drive. Multiple drives can be created by multiple invocations of INITRAMD.

To initialize a disk emulator, you invoke INITRAMD as follows:

```
INITRAMD /C StartChip /N NumberOfChips /D DeviceType [/FD] | [/FO]
```

Where StartChip is the chip number of the first chip in the drive. StartChip would commonly be 0 or 4, for an emulated drive starting in bank 0 or bank 1 of the first card, but can be any number that points to a chip on a 4M22 card in the system.

NumberOfChips is the number of chips that comprise a drive. NumberOfChips can range from 1 to 24 (all chips used on four 4M22 cards). For example if StartChip is 0 and NumberOfChips is 6, both banks on a single card would be used to create a single drive. If StartChip is 4 and NumberOfChips is 2, all of bank 1 would be used for a single drive. If a drive spans banks on a 4M22 card, both banks must be jumpered identically. It is possible to create muttple drives in a single bank as long as the major device types (as selected by the per bank jumpering) are the same.

DeviceType is a number listed by the INITRAMD /L command. Device type numbers are determined by the firmware ROM on the 4M22. These may change with 4M22 ROM revisions, so you should not depend on the type number - memory type correspondence to remain unchanged.

The /F parameter invokes a built-in FDISK and FORMAT option. There are two options when formatting drives, The /Fd option and the /Fo option. The /Fd option does a standard DOS format. This is compatible with most MS-DOS versions. The MS-DOS compatible format has the disadvantage of using large (4K) cluster sizes, and wastes considerable space when many small files are present.

The /Fo option is for use with Datalight ROM-DOS. This format uses 512 byte clusters and so is more efficient with small files. This format only works with ROM-DOS.

SOFTWARE CONFIGURATION

DISK EMULATOR INITIALIZATION

For example, if we wished to initialize a 3 chip disk emulator in bank 0, using memory type 1 and MS-DOS format, INITRAMD would be invoked as follows:

```
INITRAMD /C0 /N3 /D1 /FD
```

It is possible to create a multiple drive system by invoking INITRAMD multiple times, once per drive. For example: to initialize a two drive disk emulator using all of bank 0 with device type 1, all of bank 1 with device type 2, and DOS format, the INITRAMD command would be:

```
INITRAMD /C0 /N4 /D1 /FD
```

(Initialize a disk emulator starting at socket 0 (bank 0), using 4 sockets and device type 1)

```
INITRAMD /C4 /N2 /D2 /FD
```

(Initialize a disk emulator starting at socket 4 (bank 1), using 2 sockets and device type 2)

Once the disk emulator has been initialized, the 4M22 needs to be reset before the new disk will be recognized by the operating system.

DOS NOTES

It is possible to create a bootable DOS drive with the 4M22. The 4M22 has been tested with DOS 3.3, DOS 5.0, DOS 6.2 and DataLight ROM-DOS. MS-DOS versions 4 and 4.01 are too buggy to be of any use.

To create a bootable drive, run the DOS SYS utility after the 4M22 has been initialized. This would typically be done from a DOS floppy:

```
A:SYS C:
```

MS-DOS versions prior to 5.0 will not work reliably with more than 2 hard drives (Each 4M22 emulated drive is viewed as a hard drive by the system). Datalight ROM-DOS does not have this limitation.

If you create a bootable drive on the 4M22, make sure that you are not violating your DOS license agreement.

INTEROPERABILITY

Initialized disk chips from one MESA product can be used on any other MESA products as long as the chip type is supported on both products. Chip order is not important as long as chips are not mixed from different drives.

INSTALLATION

GENERAL

When the 4M22 has been properly configured for its application, it can be inserted into a PC/104 stack. The standoffs should then be tightened to secure the 4M22 in its place.

Be careful not to press on Lithium cell when installing the 4M22, as its leads are fairly spindly, and will collapse if excessive pressure is applied.

Note: system power must be turned off when installing or removing the 4M22 or damage to the 4M22 or system may occur.

.

REFERENCE INFORMATION

		Min	Max	Units
POWER REQUIREMENTS:				
+5 Supply voltage		4.5	5.5	V
+5 Supply current		---	75	mA
+12 Supply voltage (12V flash only)		11.5	12.5	V
+12 Supply current (12V flash only)		---	40	mA
BUS LOADING:				
Input capacitance		---	20	pF
Input leakage current		---	15	uA
Output drive capability		150	---	pF
Output sink current		12	---	mA
LITHIUM CELL LIFE:				
280 mAh (standard)		10	---	years @ 2 uA drain
ENVIRONMENTAL:				
Operating temperature range	-I	-40	+85	°C
	-C	0	+70	°C
Relative humidity		0	90	Percent

REFERENCE INFORMATION

WARRANTY

Mesa Electronics warrants the products it manufactures to be free effects in material and workmanship under normal use and service for the period of 2 years from date of purchase. This warranty shall not apply to products which have been subject to misuse, neglect, accident, or abnormal conditions of operation.

In the event of failure of a product covered by this warranty, Mesa Electronics, will repair any product returned to Mesa Electronics within 2 years of original purchase, provided the warrantor's examination discloses to its satisfaction that the product was defective. The warrantor may at its option, replace the product in lieu of repair.

With regard to any product returned within 2 years of purchase, said repairs or replacement will be made without charge. If the failure has been caused by misuse, neglect, accident, or abnormal conditions of operation, repairs will be billed at a nominal cost.

THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED. INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS, OR ADEQUACY FOR ANY PARTICULAR PURPOSE OR USE. MESA ELECTRONICS SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL.

If any failure occurs, the following steps should be taken:

1. Notify Mesa Electronics, giving full details of the difficulty. On receipt of this information, service data, or shipping instructions will be forwarded to you.

2. On receipt of the shipping instructions, forward the product, in its original protective packaging, transportation prepaid to Mesa Electronics. Repairs will be made at Mesa Electronics and the product returned transportation prepaid.

SCHEMATIC DIAGRAMS