FPXX USERS MANUAL

Version 1.0

Copyright 1997 by MESA ELECTRONICS Emeryville, CA. Printed in the United States of America. All rights reserved. This document and the data disclosed herein is not to be reproduced, used, disclosed in whole or in part to anyone without the written permission of MESA ELECTRONICS.

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

TABLE OF CONTENTS

CPU OPERATION

DISK EMULATOR OPERATION

LCD OPERATION

KEYPAD OPERATION

REFERENCE INFORMATION

HANDLING PRECAUTIONS

LITHIUM CELL

The FPXX card contains a lithium cell which can create a fire or explosion hazard if improperly handled.

Do not expose battery to temperatures in excess of 100 degrees Celsius or dispose of in fire.

Do not attempt to charge battery or modify battery related circuitry on the FPXX.

Do not short circuit battery (take care not to set the FPXX on conductive

STATIC ELECTRICITY

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

A. Leave the FPXX in its antistatic bag until needed.
B. All work should be performed at an antistatic workstation.
C. Ground equipment into which FPXX will be installed.
D. Ground handling personnel with conductive bracelet through 1 Meg resistor to ground.

INTRODUCTION

GENERAL

The FPXX is a tiny PC compatible CPU with integrated LCD graphic display and a thin, flat form factor (2.6"H x 4.5"W x 1.0"D). The FPXX is a complete OEM embedded system user interface CPU with display, keyboard, serial and parallel I/O, flash disk, and power supply.

Since the FPXX is PC compatible, most standard PC development tools and languages can be used for application programming. The FPXX has ROM-DOS pre-installed in its BIOS EPROM, and needs only your application program to make a complete user interface.

The FPXX display is a 128 by 64 resolution monochrome, graphic, backlit or reflective LCD. Display dot pitch is .52 mm. Text display modes include a 16 character by 4 line mode (8X14 character cell) and a 21 character by 8 line mode (6X8 character cell). Graphics are handled directly by the FPXX BIOS (drawdot, drawline, and bitblt.) A BGI graphics driver is supplied to support Borland compilers. FPXX display contrast can be adjusted via built-in software commands. The optional EL backlight can be turned on and off under software control. The keyswitch array surrounds the display area so that the keys can be labeled in the display. The FPXX BIOS provides support for soft-key graphic and text labels.

The FPXX requires only +5V power for operation, since display and RS-232 interface power are generated on card. The low operating power and small size of the FPXX make it well suited to portable applications. Maximum operating current is 250 mA. with backlight off. FPXX Power can be reduced to approximately 100 mA. by halting the CPU.

The FPXX CPU is a 14.75 MHz 386SX compatible processor with 2MBytes of system RAM, a 128K or 256K BIOS EPROM, and a 2 or 4MByte flash disk. The on card flash disk is supported by the FPXX BIOS, and appears to the system as a standard hard disk. A MESA provided file download utility is included in the FPXX BIOS. This allows you to down load your application program to the FPXX's disk emulator from any PC with a serial port. All utilities for using the flash disk are provided with the FPXX.

On card I/O includes a 16C550 compatible RS-232 (optionally RS-485) serial port capable of up to 115K baud, 16 bits of parallel I/O (2/3 82C55), an eight input, 12 bit A-D converter, and a battery backed clock calendar.

A 64 pin XT bus pin-out compatible header is provided for user expansion. Adaptors to ISA bus cards and PC104 cards are available from MESA. A form factor compatible system support card is available as an option. This card, the FPXX-SSB, has a 1.44M floppy controller, 2 RS-232 serial ports, and a parallel printer port.

HARDWARE CONFIGURATION

GENERAL

The FPXX has 2 hardware setup jumpers and I/O connectors accessible from the back side of the FPXX unit. When changing these option jumpers or installing I/O connectors. the FPXX should be set display side down on a soft pad. In the following discussions, when the words "up", "down", "right", and "left" are used it is assumed that the FPXX card has been set display side down with the PC/104 expansion connector at the bottom edge of the card (nearest the person doing the configuration). Note that these jumpers will not be accessible if a PC/104 expansion card is installed on the FPXX.

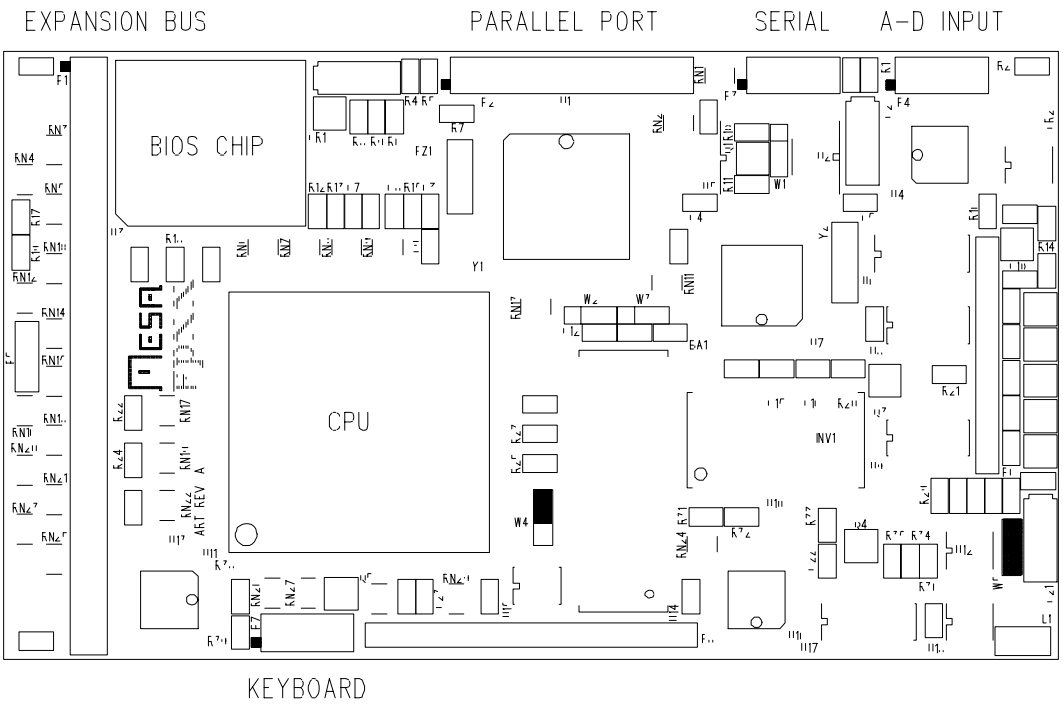
DEFAULT JUMPER SETTINGS

Factory default FPXX jumpering is as follows:

FUNCTION	JUMPER(S)	SETTING
WatchDog enable	W4	UP (enabled)
LCD Refresh	W5	UP for backlit LCDs Down for non-backlit

HARDWARE CONFIGURATION

JUMPERS AND CONNECTOR LOCATIONS



HARDWARE CONFIGURATION

WATCHDOG ENABLE

The FPXX has a hardware watchdog that can reset the CPU should a system malfunction occur. This watchdog can be disabled if desired. Jumper W4 enables or disables the watchdog. If W4 is in the UP position,(the default) the watchdog is enabled. If W4 is in the down position, the watchdog is disabled.

LCD REFRESH RATE

The LCD CL2 rate can be changed to generate the correct refresh rate for the backlit and non-backlit displays. The jumper W5 controls this function. It should be set to the up position for backlit displays and the down position for non-backlit displays. It however will work in both positions with either display and can be changed for best display appearance.

CONNECTORS

GENERAL

The FPXX has 4 user I/O connectors, and an expansion bus connector. All connectors are 2 mm male headers. Pin one of all I/O connectors is marked with a square pad on the connector. The following lists the I/O connectors, their sizes, and functions:

CONNECTOR	FUNCTION	PINS
P1	XT EXPANSION BUS	64
P2	PARALLEL PORT	26
P3	SERIAL PORT + POWER	10
P4	ANALOG-IN	10
P5	EL BACK LIGHT POWER	2
P6	MEMBRANE SWITCH SCAN	10
P7	KEYBOARD AND RESET	10
P8	LCD INTERFACE	14

P5, P6, and P8 are internal connections between the FPXX CPU card and the display.

KEYBOARD CONNECTOR

Connector P7 is the AT keyboard, reset-in and speaker connector. P8 is a 10 pin dual row 2mm header. An external reset switch input and speaker output are also available on P7. The reset circuit works by grounding the /RESIN signal. The speaker output is intended to drive high impedance speakers (40 ohms or more) . Eight Ohm speakers will be too quiet for most applications. The speaker output idles at +5V so the speaker common is +5V.

CONNECTORS

KEYBOARD CONNECTOR

Keyboard connector P7 pin-out is as follows:

PIN	SIGNAL	FUNCTION
1	SPKOUT	Speaker output
2	SPKVCC	Speaker common (+5V)
3	/RESIN	Reset-in
4	RGND	Reset-in common (ground)
5	NC	No Connect
6	KBCLK	Clock from keyboard
7	KDATA	Data from keyboard
8	KEY	(Pin missing - key)
9	KGND	Keyboard power return
10	KVCC	Keyboard +5V power

A keyboard adapter cable is available from MESA (The KBADPT).

SERIAL CONNECTORS

Connector P3 is the serial port connector. The serial port connector is a 10 pin, 2 mm male header. The suggested mating connector for P3 is Suyin20043-10G2 or 3M 152210-100-GG. These are both cable mount female headers for 1 mm pitch IDC flat cable. Individual wire type connectors are available for 2 mm connectors but are not suggested due to their fragility.

P3 has the same pinout as a standard 10 pin .1" serial header. Power to the FPXX can be supplied through this connector, if the FPXXSSB is not used.

CONNECTORS

SERIAL PORT CONNECTORS

Serial port connector s P3 pinout is as follows:

HDR PIN	DSUB PIN	SIGNAL	FUNCTION
1	1	CD	Handshake in
2	6	DSR	Handshake in
3	2	XD	Data in
4	7	RTS	Handshake out
5	3	TXD	Data out
6	8	CTS/RS-485A	Handshake out
7	4	DTR	Handshake out
8	9	RI/RS-485B	Handshake in
9	5	GND	Signal ground
10	NC	+5V	+5V user power

NOTE: The serial port can optionally be configured to be RS-485 compatible (data leads only). In this case only pins 6,8, 9, and 10 are used. This is a factory assembly option. When the RS-485 option is installed make sure that you do not connect any RS-232 signals to the RS-485 pins (6 and 8) or you may damage the RS-485 transceiver chip(s). A serial adapter cable is available from MESA (The FPXXSERADPT)

PARALLEL PORT CONNECTOR

Connector P2 provides access to 16 bits of user I/O. P2 is a 26 pin, 2 mm male header. The suggested mating connector for P2 is Suyin 20043-26G2 or 3M 152226-100-GG. These are both cable mount female headers for 1 mm pitch IDC flat cable. Individual wire type connectors are available for 2 mm connectors but are not suggested due to their fragility

The parallel I/O port is not a standard printer port but an 82C55 programmable peripheral interface. Ports A and B of the PPI are available for any use, but port C is used by the FPXX for membrane keyswitch scanning. PPI port B has pullup resistors, and if inputs are needed, it is suggested that port B be used. *Default FPXX BIOS programming sets port A to output mode, and port B to input mode.*

CONNECTORS

PARALLEL PORT CONNECTOR

Port B bits 0 and 1 have a special purpose if the FPWX compatibility option is enabled. If an external AT type keyboard is installed, port B bit 0 is used as keyboard clock, and bit 1 is used as keyboard data.

PPI port C is brought out to connector P2 to allow external membrane switches to be used, or in those cases where the built in membrane switch is not used. For normal FPXX operation, you should make no connections to the port C bits.

PIN	SIGNAL	PIN	SIGNAL
1	PA0	2	PA1
3	PA2	4	PA3
5	PA4	6	PA5
7	PA6	8	PA7
9	PB0 (KBCLK)	10	PB1 (KBDATA)
11	PB2	12	PB3
13	PB4	14	PB5
15	PB6	16	PB7
17	PC0	18	PC1
19	PC2	20	PC3
21	PC4	22	PC5
23	PC6	24	PC7
25	GND (KBGND)	26	VCC (KBVCC)

CONNECTORS

ANALOG INPUT CONNECTOR

Connector P4 is the analog input connector. The FPXX has an 12 bit, 3.75VV full scale A-D converter with 8 user inputs. The analog input connector is a 10 pin, 2 mm male header. The suggested mating connector for P4 is Suyin20043-10G2 or 3M 152210-100-GG. These are both cable mount female headers for 1 mm pitch IDC flat cable. Individual wire type connectors are available for 2 mm connectors but are not suggested due to their fragility.

Analog input connector pin-out is as follows:

PIN	SIGNAL
1	AIN 0
2	AIN 1
3	AIN 2
4	AIN 3
5	AIN 4
6	AIN 5
7	AIN 6
8	AIN 7
9	GND
10	VCC

EXPANSION CONNECTOR

The dual row 64 pin 2 mm male header on the FPXX is for PC bus expansion. This connector is pin-out compatible with 8 bit PC/104 I/O cards, and 8 bit ISA cards. A PC-104 and a ISA bus adapter are available from MESA to allow one ISA card or up to two PC-104 expansion cards to be added to the FPXX. The adapters (PN FPWXISA for the ISA adapter and FPWX104 for the PC-104 adapter) are intended mainly for development use, for example, to use a floppy or network to transfer data to the FPXX.

When adding I/O cards to the FPXX, make sure that the total +5V supply current does not exceed 1A. (FPWX power connector limitation).

CPU OPERATION

GENERAL

The FPXX CPU is an ALI M6117 80386SX compatible processor running at 14.75 MHz. The 6117 integrates the 80386SX CPU and AT compatible chipset in a single package. All standard AT integrated peripherals are available on the M6117, including the battery backed clock. FPXX system memory is 2M bytes. The FPXX BIOS chip can be either 128K bytes or 256K bytes. The 256K BIOS allows for storage of more graphic objects, fonts etc. The FPXX can write to 5V flash memory BIOS chips to allow BIOS upgrades and adding new graphic resources.

STARTUP ERRORS

The BIOS performs a variety of system tests at startup. Serious problems are reported by beep codes. TThe BIOS beep codes are as follows:

BEEPS	ERROR
2	Bad external ROM checksum
3	External ROM initialization error
4	No system memory found
5	Can't boot - no resident language
6	BIOS ROM checksum error
7	Bad local RAM
8	VGA ROM initialization faulure
9	Invalid system configuration data (or forced default)
10	No ROM BIOS image found
11	Corrupted BIOS module found

FPWX COMPATIBILITY

The FPXX is intended to be a form and function compatible upgrade to the FPWX. There are some differences related to the CPU and DMA changes from the FPWX.

The serial port on the FPXX is a standard 16C550 type COM port, instead of the 8251 type port built into the V40. The FPXX's serial connector in not pinned out the same as the FPWX. This is because we wished to maintain full COMX compatibility including all the handshake lines on the interface. This also means the RS-485 compatibility is an assembly time option (you can have RS-485 or RS-232 but not both)

The CPU on the FPXX is faster than the V40 on the FPWX and fully AT compatible. Since the CPU is 80386 compatible, protected mode software can run on the FPXX. One disadvantage of the faster CPU is that the FPXX draws about twice as much power as the FPWX. The FPXX uses an AT compatible keyboard instead of the XT compatible keyboard that the FPWX uses.

The 8237 compatible DMA on the FPXX is not capable of having the starting DMA address changed without simultaneously changing the current DMA address. What this ultimately means is that the FPXX does not support the blinking graphics that the FPWX does.

CPU OPERATION

PC COMPATIBILITY

The FPXX CPU is compatible with most programs developed with standard PC software tools, including protected mode programs. The display however is only compatible if the character output goes through the BIOS. Programs that access video directly (that dont go though the BIOS or DOS) will not work on the FPXX. (They may work but the output will not be seen). Programs that use the hardware tic interrupt instead of the user tic interrupt (INT1C) may not operate properly.

SERIAL PORT

The serial port on the FPXX is a standard PC COMX type port with a FIFOed (16C550A compatible) UART. The port interface and is located at the standard COM4 location (2E8H). It can use IRQ3, the normal interrupt for COM4 or IRQ10, in case the FPXXSSB is used and needs IRQ3 for COM2.

RS-485 OPERATION

The FPXX has assembly time options to the serial port to be built with a RS-485 interface instead of RS-232. Note that this is an either-or option, you can not have RS-232 and RS-485 on the same port. When RS-485 levels are used , the RTS bit in the modem control register is used to control transmit enable.

RS-485 communication on the FPXX is always half-duplex, because of the fact that the receiver is disabled when the transmitter is enabled and vice-versa. When RS-485 is used with asynchronous serial ports, it is important that the idle (non-driven) line voltages be held in the marking state. This can be done by providing a 1K pull-up resistor (to +5V) on the RS-485A line and a 1K pull down resistor (to ground) on the RS-485B line somewhere on the RS-485 bus. When using RS-485, it is the responsibility of the character or packet output routine to enable and disable the RS-485 transmitter.

The Pascal include file SERIAL.PAS in the source directory of the distribution disk has some low level serial port and RS-485 enable-disable procedures that can be used as examples for writing your own code.

CPU OPERATION

PARALLEL PORT

16 parallel I/O bits are available for user applications. These bits are the A and B ports of the on card 82C55. The 82C55 is located at 0208H.

This 82C55 is also used to scan the membrane keypad. Because the chip is shared by the keypad scan routine, you must only set the 82C55 mode register (at 020BH) to certain values. The legal values are 081H (port A and port B out), 083H (port A out and port B in), 091H (port A in and port B out), and 093H (port A and port B in). If you set any other modes, the keypad will stop working.

Port B is more suited to input use as it has 10K pullup resistors, while port A has none. This means that if you want 8 inputs and 8 outputs, mode 083H is probably a better choice than mode 091H.

CPU OPERATION

SERIAL FILE DOWNLOAD

To allow transferring of application programs to the FPXX, which may not have a floppy drive or other means of transferring programs, a set of utility programs are provided. The are called WSEND and WRECVCOM.

WSEND and WRECVCOM comprise a very simple file download utility set. When the FPXX is supplied with ROM-DOS, WRECVCOM is normally supplied built into the BIOS ROM as part of the ROM drive (C:).

The first requirement for WSEND and WRECVCOM to work is the proper cable. This cable has only three wires, and is a 'data only null modem' cable. Assuming that your host machines serial port is a 9 pin male (AT pinout) type, and that the FPXX has its serial port adapter cable installed, the cable would have 9 pin female connectors at both ends and the following connections:

```
HOST 9 pin   FPXX 9 pin

    5 --------- 5   ( ground )
    2 --------- 3   ( data <- )
    3 --------- 2   ( data -> )
```

If your host machine has a 25 pin serial connector, the cable needs a female 25 pin connector on the host end and a female 9 pin connector on the FPXX end. This cable must have the following connections:

```
HOST 25 pin  FPXX 9 pin

    7 --------- 5   ( ground )
    2 --------- 2   ( data -> )
    3 --------- 3   ( data <- )
```

Another option is to use the MESA DWNLDADPT cable. The DWNLDADPT cable is a five foot long flat cable that connects directly between the FPXX's 10 pin header and a 9 pin DB type connector on the host. WSEND runs on a host machine. This host machine must be a PC with a standard COMX RS-232 serial port available.

SEND is invoked this way:

```
SEND PPP [BR]
```

PPP is the hexadecimal port address of the serial port on the host machine (3F8 = COM1, 2F8 = COM2, 3E8 = COM3, and 2E8 = COM4). BR is an optional baud rate parameter. if BR is not supplied, send uses 9600 baud. For example SEND 2F8 38400 would send files through COM2 at 38400 baud.

CPU OPERATION

SERIAL FILE DOWNLOAD

Once WSEND is running on the host machine, WRECVCOM is run on the client CPU card to download files.

WRECVCOM is invoked this way:

```
WRECVCOM RFN LFN [BR] [Q]
```

RFN is the remote file name (the source file) which is relative to the path where send was launched. LFN is the local file name (the target file).The Q parameter causes WSEND to be aborted when the file transfer is complete. BR is an optional baud rate parameter. If BR is not supplied, WRECVCOM uses 9600 baud. For example:

```
WRECVCOM FOO GOO 38400 Q
```

Would get the remote file FOO, write it to the local file GOO, and abort WSEND when done. All data transfers would be done at 38400 baud.

If you set the baud rates on the command line, the SEND baud rate must match the WRECVCOM baud rate. You may not be able to use the maximum baud rate, depending on your host CPU speed, serial port characteristics, interconnect cable etc. (Your mileage may vary)

The standard ROM drive supplied with ROM-DOS versions of the FPXX has a batch file R.BAT that simplifies receiving files. R.BAT assumes that send was invoked with 38400 as the baud rate. To use R.BAT you type:

```
R FILENAME
```

R.BAT consists of one line: **`D:WRECVCOM %1 %1 38400`**

CPU OPERATION

CONSOLE SWITCHING

The FPXX can use an AT keyboard, the scanned keypad or the serial port for console input. Console out can be directed to a video card (yes, the FPXX _will_ work with a video card if the proper bus adapter is available!), the LCD module or the serial port. To determine which console option is used, the FPXX supports extended INT 1A functions that allow dynamic switching of console input and output.

If no video card is detected in the system, output defaults to the LCD screen and the default console in comes from the AT keyboard port. Console output defaults to a video card if a video card is detected in the system. If a video card is detected, the console input comes from the AT keyboard port.

The FPXX distribution disk has some batch files in the DEMO directory that dynamically reroute the console in and out. Consult the BATREAD.ME file in that directory for more information on those files.

The key pad console in option must be explicitly enabled by calling the appropriate extended INT 1A function.

USING THE A-D CONVERTER

The FPXX has a built-in 12 bit A-D converter. There are 8 available inputs for user applications. The A-D converter is read with a BIOS call. The BIOS A-D read function F_SYSATODRAWREAD returns an unsigned 16 bit number (0 to 65535 full scale). The reference voltage is 3.75V, so a full scale reading of 65535 represents an input of 3.75V.

The HADDEMO and VADDEMO programs are an simple examples of using the A-D. The source code for HADDEMO and VADDEMO are in the \SOURCE\PAS subdirectory of the distribution disk. ADDEMO relies on the WIDGET.PAS include library, as do all PASCAL demonstration programs.

CPU OPERATION

POWER CONSUMPTION

Typical FPXX power consumption is less than 1.5W (300 mA). This can be reduced to less than 100 mA by turning off the backlight and halting the CPU. It is the responsibility of the application program to execute the halt instruction when idle.

HARDWARE TIC CLOCK

To simplify application timing tasks, and allow responsive keypad scanning, the hardware tic clock on the FPXX runs at 100 CPS instead of the normal 18 CPS. This should have no serious side effects with well behaved programs, as the user tic clock still runs at 18 CPS. The 100 CPS to 18 CPS conversion is done with a rate multiplier type algorithm that does not detract from the long term accuracy of the system tic clock.

Application programs that intercept the user tic interrupt (INT 1C) will be still be interrupted at an average rate of 18 CPS, but instead of the interrupts occurring regularly, the will occur at either 50 or 60 mS intervals. If an application program requires interrupts with a constant interval , it should use the hardware tic interrupt instead of the software tic.

WATCHDOG TIMER

The FPXX is intended mainly for embedded system applications where there is no one to hit the reset switch should something go awry. To prevent a crashed or otherwise hung system from remaining so indefinitely, the FPXX is provided with a built in watchdog timer that will reset the FPXX if not 'fed' regularly. The time-out period of this counter is about 1 second . The default INT 1C (user tic clock) task 'feeds' the watchdog. User software must be careful not to disable interrupts for more than the timeout period or the watchdog may bite! Any program that intercepts INT 1C must either chain through the old vector, or be responsible for 'feeding' the watchdog itself.

If the watchdog timer causes a problem with software that must disable interrupt for long periods of time, it can be disabled by moving jumper W4 to the down position.

CPU OPERATION

EXTENDED INT 1A FUNCTIONS

Console I/O redirection, and some other miscellaneous control functions on the FPXX are accessed via extended INT 1A calls. The calling convention used in all these calls is as follows: register AH = 87H, register BX is the offset part of the structure pointer, and register CX is the segment part of the structure pointer. CX:BX points to a structure, the first byte of which is the function number, and the second byte is the returned status byte. After these first bytes, a variable number of byte or word parameters follow.

For a full description of the INT 1A extended functions, you should refer to the SINT1A.H (for C users), and SINT1A.INC (for assembly programmers) files in the source directory of the FPXX distribution disk. The following is a brief list of extended INT 1A functions for quick reference: Subfunctions are indented and listed under main functions.

F_SYSCNTRLINFOQ = 0
Get system control code revision level, etc.

F_SYSKBSOURCEQ = 1
Inquire about keyboard source.

F_SYSKBREROUTE = 2
Select keyboard source.

- KBSRC_KEYBOARD = 0
  Take input from PC keyboard.
- KBSRC_SERIAL = 1
  Take input from local serial channel.
- KBSRC_OFF = 2
  Turn off serial and PC keyboard input.

F_SYSVIDEOSOURCEQ = 3
Inquire about video destination.

F_SYSVIDEOREROUTE = 4
Select video destination.

- VIDDEST_VIDEO = 0
  Send video output to standard video.
- VIDDEST_SERIAL = 1
  Send video output to local serial channel.
- VIDDEST_LCD = 2
  Send video output to local LCD display.
- VIDDEST_STUB = 3
  Send video output to black hole.

CPU OPERATION

EXTENDED INT 1A FUNCTIONS

F_SYSSCUBAUDSEL = 5
Set Console baud rate.

BAUDSEL_110	=	0
BAUDSEL_150	=	1
BAUDSEL_300	=	2
BAUDSEL_600	=	3
BAUDSEL_1200	=	4
BAUDSEL_1800	=	5
BAUDSEL_2000	=	6
BAUDSEL_2400	=	7
BAUDSEL_3600	=	8
BAUDSEL_4800	=	9
BAUDSEL_7200	=	10
BAUDSEL_9600	=	11
BAUDSEL_19200	=	12
BAUDSEL_38400	=	13
BAUDSEL_57600	=	14
BAUDSEL_115200	=	15

The various baud rate selection constants.

F_SYSSCUBAUDSELQ = 6
Get current Console baud rate

F_SYSVIDEOHOOK = 7
Get video code entry point

F_SYSCPUREV = 8
Get CPU card revision number

F_SYSSCANINFO = 10
Set membrane keyboard event handler address

F_SYSSCANINFOQ = 11
Get current membrane keyboard event handler address

CPU OPERATION

EXTENDED INT 1A FUNCTIONS

F_SYSATODTYPE = 12
Get A-D converter type

F_SYSTEMPSENSE = 13
Read card temperature (averaged)

F_SYSATODRAWREAD = 14
Read A-D converter

SETUP STORAGE

Many FPXX options can be saved in the serial EEPROM on the FPXX card. These options include: initial baud rate, LCD parameters, contrast setting, etc. These parameters can be set with the INT 1A functions or the provided utility SETFPXX.EXE

SETFPXX.EXE reads a text file of setup options, and programs these into the FPXX's EEPROM. These setup files have a default extension of .CF. SETFPXX and a number of configuration files are located in the UTILS directory of the FPXX distribution floppy. SETFPXX is invoked with the configuration file name as a parameter: For example:

```
SETFPXX FPXX.CF
```

Would configure the FPXX with the EEPROM settings in the FPXX.CF configuration file.

SETFPXX has three command line switches: /D, /N and /Q. These command line switches follow the file name. The /D option causes the FPXX EEPROM to be initialized to it's default configuration. When the /D option is used, no file name is needed. The /N option causes the configuration file to modify the default configuration, and store the result into the EEPROM. If /N is not specified, all options not specifically changed in the configuration file will remain at their previous settings.

As long as the /N or /D switches are not used, configuration files loaded with SETFPXX only affect the options specified in the file. This makes it possible to separate the configuration files into pieces that only affect a certain aspect of FPXX operation.

Note that EEPROM settings do not take effect until the FPXX is reset.

DISK EMULATOR OPERATION

GENERAL

The FPXX has a built in nonvolatile disk emulator with a capacity of up to 4M bytes using FLASH EEPROM.

The FPXX disk emulator is viewed as a hard disk by system software. This means that the first emulated drive will be drive **C**: , and the next emulated drive will be drive **D**: etc.

RELIABILITY

In an embedded system environment where a system that won't boot is basically a failed system, it is important to understand some characteristics 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 inaccessible, not just the file that was being written.

The reason is that when DOS writes a directory entry 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.

As a further precaution the data drive can be split into two logical drives with FDISK. If the data drive was physical drive D, the two logical drives would be drive D: and drive E:. When this is done, data corruption on one logical drive will not effect the other drive, allowing a dual write scheme to be used to protect valuable data.

DISK EMULATOR OPERATION

DISK EMULATOR INITIALIZATION

The disk emulator chip on thr FPXX has been initialized at MESA and should not need any further initialization. However it is possible to corrupt the file system image by turning off power in the middle of a disk write or some other calamity. To recover from this, you will need to re-initialize the disk emulator.

This initialization is done with INITRMDB.EXE. INITRMDB.EXE is supplied in the UTILS subdirectory of the FPXX distribution disk. If INITRMDB is run with a /L parameter, it will list the types of disk emulator chip on the FPXX card.

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

```
INITRMDB /CStartChip /NNumberOfChips /T /Y
```

On a FPXX , StartChip must be 0 and NumberOfChips must be 1.

```
INITRMDB /C0 /N1 /T /Y
```

(Would be the standard command to re-initialize the disk emulator)

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

After initalizing the drive, you must reboot the FPXX and then run FDISK followed by FORMAT as you would a normal DOS drive.

LCD OPERATION

GENERAL

The FPXX uses a 128 by 64 pixel graphic LCD display. A display with EL backlight is available as an option. The FPXX display is not hardware compatible with any standard PC display, but hardware compatibility is of little value for such a small display. Instead, the FPXX BIOS supports the display with standard INT 10 BIOS calls plus some extended function calls for graphics, etc. This means that normal text output will work on the FPXX screen, but programs that access video memory directly will not work on the FPXX.

SYSTEM RESOURCES USED

The FPXX display is DMA driven, and uses DMA channel 0. IRQ 11 is used for a frame rate interrupt. The frame rate interrupt is used for cursor blinking

The display buffer memory resides in system RAM. The standard configuration uses 2560 bytes of system RAM.

TEXT MODE

When the FPXX LCD panel is used for console output, the display is compatible at the BIOS or INT 10 level with a standard PC text display. The display is *not* hardware compatible with the standard monochrome or color IBM type displays.

When the display panel is used, there are no equivalents to the video modes used on a normal PC. Changing video mode (INT 10 function 00) will only clear the screen. The display is always in a graphics mode, allowing freedom to mix text and graphics.

The FPXX BIOS does support most of the INT 10 video display calls, so most software that accesses the display through the BIOS will work. (Within the limits of the smaller display size).

The FPXX BIOS does *not* maintain a copy of the characters and attributes in the current display. This means that INT 10 functions that depend on the previous display contents will not work. INT 10 functions that may cause trouble are function 08h (read character and attribute) and function 0Ah (write character at cursor). Function 0Ah may cause trouble because it assumes that the character attribute to use is the one at the current character position in video memory. The FPXX's BIO's' function 0Ah simply writes characters with the default attributes.

Most compilers have an option to use the BIOS instead of direct memory access when writing to the display. You will need to enable this option when compiling applications using the LCD display.

LCD OPERATIONDISPLAY

TEXT FONTS

The standard FPXX BIOS has two available fonts for text display, a large and a small font. The small font is suggested for most applications. This font uses a 5X7 sized character in a 6X8 cell. This allows a 8 line by 21 character text output. The large font is useful where the display must be visible from a distance. The large font uses 7X9 characters in a 8X14 cell. The large font allows a 4 line by 16 character display.

The font type is chosen via an extended INT 10 function. See the section EXTENDED INT 10 FUNCTIONS for more information.

TEXT ATTRIBUTES

The FPXX BIOS supports reversed video attributes. This character attribute is selected with the standard INT 10 function. Underline, blink, and intensity attributes are ignored.

TEXT WINDOW

FPXX text output can be confined to a user defined rectangular region within the borders of the display. All standard text operations will be confined to this region, including scrolling, screen clearing, and characters that wrap at the end of the line.

The purpose of this text window allow graphic areas around the periphery of the display for labeling softkeys, status indicators etc. The text window function can also be used with the region save and restore functions to pop-up notifiers on a graphics screen, and then restore the screen when the notifier is dismissed.

The text window dimensions are defined with an extended INT 10 function. Window dimensions are specified in pixels. See the EXTENDED INT 10 FUNCTIONS section for more information.

The text window is set to the physical display limits when the BIOS initializes the LCD panel. *Text windows should normally be set so that the window height is a multiple of the current font height. If this is not done, scrolling may leave partial characters visible on the display.*

TEXT FONTS

The standard FPXX BIOS has two available fonts for text display, a large and a small font. The large font is suggested for most applications. This font uses a 7x9 sized character in an 8x14 cell. The small font can be used where it is necessary to get the maximum amount of text on the screen or for labeling small buttons or graphic objects. The small font uses a 5x7 sized character in a 6x8 cell. An auxiliary 14x18 font in a 16x28 cell is provided as a file for applications requiring visibility at some distance.

The BIOS actually maintains pointers to two fonts at any time, the TTY font and the NON-TTY font. The standard BIOS text output routines use the TTY font, while graphic text drawing uses the NON-TTY font. Both the TTY font and the NON-TTY font are initially set to the large (7X9) character font.

The font type for the TTY font and NON-TTY font are chosen via an extended INT 10 functions.

LCD OPERATION

DISPLAY CONTRAST

The FPXX display contrast can be read or set via extended INT 1A system calls. There are calls for setting the current contrast voltage and for permanently saving the current value. The initial contrast value is set by the FPXX.CF file.

A supplied utility program: SC.EXE can be used to dynamically adjust the contrast and save the adjusted value. The source code for SC.EXE is in the SOURCE directory of the distribution disk. This program can be used as an example if you want to include contrast adjustment in your application program.

BACKLIGHT CONTROL

The EL backlight inverter on the FPXX can be turned off after a period of inactivity if desired. The backlight timeout value and backlight turn on events are specified in the FPXX.CF file.

LCD OPERATION

GRAPHICS

The FPXX BIOS has support for several graphics operations. These operations include BitBLT, single pixel width line drawing (Bresenham algorithm), screen region save and restore, and individual dot drawing.

EXTENDED INT 10 FUNCTIONS

The LCD specific video control functions are accessed via extended INT 10 calls. The calling convention used in all these calls is as follows: register AH = 80H (control functions) or 81H (graphic functions), register BX is the offset part of the structure pointer, and register CX is the segment part of the structure pointer. CX:BX points to a structure, the first byte of which is the function number, and the second byte is the returned status byte. After these first bytes, a variable number of byte or word parameters follow.

For a full description of the INT 10 extended functions, you should refer to the SINT10.H (for C users), and SINT10.INC (for assembly programmers) files in the source directory of the FPXX distribution disk. Some example programs that use the extended INT 10 functions are provided in the source directory.

The following is a brief listing of the extended INT 10 functions:

CONTROL FUNCTIONS (AH= 80H)

F_DISPCNTRLINFOQ = 0
Get display control code revision level

F_DISPCHECKSETUP = 1
Validate setup parameters.

F_DISPINIT = 3
Activate the LCD display.

F_DISPMODEGET = 4
Get display operating modes information.

LCD OPERATION

EXTENDED INT 10 FUNCTIONS

CONTROL FUNCTIONS (AH= 80H)

F_DISPMODESET = 5
Set display operating mode.

F_DISPGETBUFPTR = 10
Get segment of display buffer.

F_DISPSETBUFPTR = 11
Set segment of display buffer.

F_DISPSTATEGET = 12
Get display on/off state information.

F_DISPSTATESET = 13
Set display on/off state.

F_DISPBKLTSTATEGET = 14
Get display backlight on/off state.

F_DISPBKLTSTATESET = 15
Set display backlight on/off state.

F_DISPCONTRASTGET = 16
Get current display contrast setting.

F_DISPCONTRASTSET = 17
Set display contrast.

LCD OPERATION

EXTENDED INT 10 FUNCTIONS

GRAPHIC FUNCTIONS (AH= 81H)

F_GRFXDISPINFOQ = 0
Get graphics code revision level

F_GRFXDISPDIMQ = 1
Get display dimensions.

F_GRFXTTYINFOGET = 2
Get TTY region information.

F_GRFXTTYINFOSET = 3
Set TTY region information.

F_GRFXCRSRXABLQ = 4
Get cursor xable state information.

F_GRFXCRSRXABL = 5
Xable cursor display.

F_GRFXTTYPATGET = 6
Get TTY region fill pattern.

F_GRFXTTYPATSET = 7
Set TTY region fill pattern.

F_GRFXPATGET = 8
Get non-TTY region fill pattern.

F_GRFXPATSET = 9
Set non-TTY region fill pattern.

LCD OPERATION

EXTENDED INT 10 FUNCTIONS

GRAPHIC FUNCTIONS (AH= 81H)

F_GRFXERASEREGN = 10
Erase a rectangular region.

F_GRFXDRAWDOT = 11
Draw a dot.

F_GRFXDRAWLINE = 12
Draw a single-pixel line.

F_GRFXBITBLT = 13
BitBLT to a rectangular screen region.

F_GRFXUNBITBLT = 14
BitBLT from rectangular screen region.

F_GRFXERASETTY = 15
Erase the TTY region.

F_GRFXERASENONTTY = 16
Erase the non-TTY region.

F_GRFXSETTTYDIM = 17
Set TTY region location/dimensions by point and size.

F_GRFXFONTINFOGET = 18
Get information about ROMBIOS-resident TTY fonts.

F_GRFXATTACHFONT = 19
Specify the font for the TTY font.

LCD OPERATION

EXTENDED INT 10 FUNCTIONS

GRAPHIC FUNCTIONS (AH= 81H)

F_GRFXSETTTYLOC = 21
Set TTY region location/dimensions by rectangular region.

F_GRFXSELFONT = 20
Select TTY font by number.

F_GRFXTFONTINFOGET = 22
Get information about the current TTY font.

F_GRFXNTFONTINFOGET = 23
Get information about the current non-TTY font.

F_GRFXATTACHNTFONT = 24
Specify the font for the non-TTY font.

F_GRFXDRAWCHAR = 25
Draw a character from the non-TTY font at the specified pixel coordinate.

F_GRFXSAVEREGNSIZEQ = 26
Get size of buffer required by the specified LCD display region image.

F_GRFXSAVEREGN = 27
Save specified LCD display region to image buffer.

F_GRFXUNSAVEREGN = 28
Restore specified LCD display region from image buffer.

F_GRFXSHOWPLANE = 29
Display a specific plane.

F_GRFXUNBITBLTB = 30
Alternate access to F_GRFXUNBITBLT

F_GRFXTTYCHARLOCQ = 31
Translate from TTY region to absolute pixel coordinates

LCD OPERATION

DEMO PROGRAMS

A number of demonstration programs for the LCD display are included on the FPXX distribution disk.

BLT.EXE is a demonstration program that shows some of the FPXX BIOS's graphic capabilities.

SC.EXE is a program for setting the FPXX display contrast and backlight timeout. SC.EXE uses the keypad for adjusting the contrast and timeout settings. The settings can be permanently saved by pressing the SAVE button. HADDEMO.EXE and VADDEMO read the A-D converter channels and display the output on bargraph type meters.

GRAPHIC LIBRARIES

The FPXX is supplied with C and a PASCAL function libraries that access the built in BIOS graphic functions. In addition the libraries contain some simple graphic widgets for display control. These include pop up text windows, labeled buttons, bargraph displays, edgemeter displays and general display utilities. The Pascal library is called WIDGET.PAS .

LCD OPERATION

MINT

MINT.EXE is a simple program for calling MESA's extended interrupt functions from the command line. MINT allows fairly complex demonstrations of the extended interrupt functions to be executed from batch files. MINT is invoked with a series of parameters on the command line. The command- parameter sequence is:

```
MINT IN AH FN EC SF PAR0 PAR1 PAR2...
```

Where the first five parameters are always 2 digit hexadecimal values as follows: IN is the interrupt number, AH is the register AH value, FN is the main function number, SF is the sub function a number, and EC is the function status return variable.

MINT loads register AH with the AH value, loads CX:BX with a pointer to a structure consisting of the sequence FN,RV,SF,PAR0,PAR1,PAR2..., and then calls interrupt IN.

The PARX values that follow these can be byte or word hexadecimal parameters. MINT expects word parameters to be 4 characters long and byte parameters to be 2 characters long. For example 0010 is a word parameter and FF is a byte parameter.

MINT is best used in short batch files when testing FPXX functions. For example the following batch file (VIDOUT.BAT) switches the video output stream to the video monitor.

```
@ECHO OFF
MINT 1A  87 04       00      00
rem  INT  AH MAINFUNC ERRCODE STDVID
rem route video out to standard video
```

The following batch file (SETCON.BAT) expects a 4 digit hexadecimal contrast value on the command line, and sets the contrast value (VEE voltage) to that value.

```
@ECHO OFF
MINT 10  80 11       00      %1
rem  INT  AH MAINFUNC ERRCODE CONTRAST VALUE
rem LCD contrast value (VEE) must be 4 (hex) digits!
```

The DEMO directory of the distribution disk has a series of these batch files for demonstration purposes.

LCD OPERATION

PCX2HEX

PCX2HEX is a MESA supplied utility that makes it simpler to embed images into application programs. PCX2HEX converts a monochrome PCX picture file to a hexadecimal ASCII array of bytes suitable for inclusion in C or Pascal source code.

To make a picture for embedding in an application program, you first create the PCX file with one of the many available 'Paint' type programs. Most of these paint programs allow you to specify the image size when you create a new file. You need to specify this size to match the size of your desired image. You should make the picture width a multiple of 16 pixels wide for compatibility with PCX editors. The PCX file must be a 1 plane (monochrome) file type.

Once you have created a suitable image file, you run PCX2HEX to convert the PCX file to a hex constant. PCX2HEX can make files suitable for C, Pascal, or assembler programs. To make a C type file from GOOBER.PCX, you would type:

```
PCX2HEX GOOBER.PCX C
```

This would create a GOOBER.H file for inclusion in C source code. To make a Pascal type file from GOOBER.PCX, you would type:

```
PCX2HEX GOOBER.PCX PASCAL
```

This would create a GOOBER.PAS file for inclusion in Pascal source code The constant array will have the same name as the PCX file that it was created from. In addition to the array of bytes that make up the picture, two 16 bit constants are defined in the hex file. These are the X and Y dimensions of the picture. These dimensions have the same name as the PCX file that they were created from except that the X dimension has XDIM appended to the name and the Y dimension has YDIM appended. These dimensions are needed by the BITBLT function.

After the desired hex constant file(s) have been created, they can be included in your application program source code. They can then be displayed on the LCD screen using the INT 10 BITBLT function. For an example of using the hex constant files with BITBLT, you can refer to the SC.PAS file in the SOURCE directory of the FPXX distribution floppy.

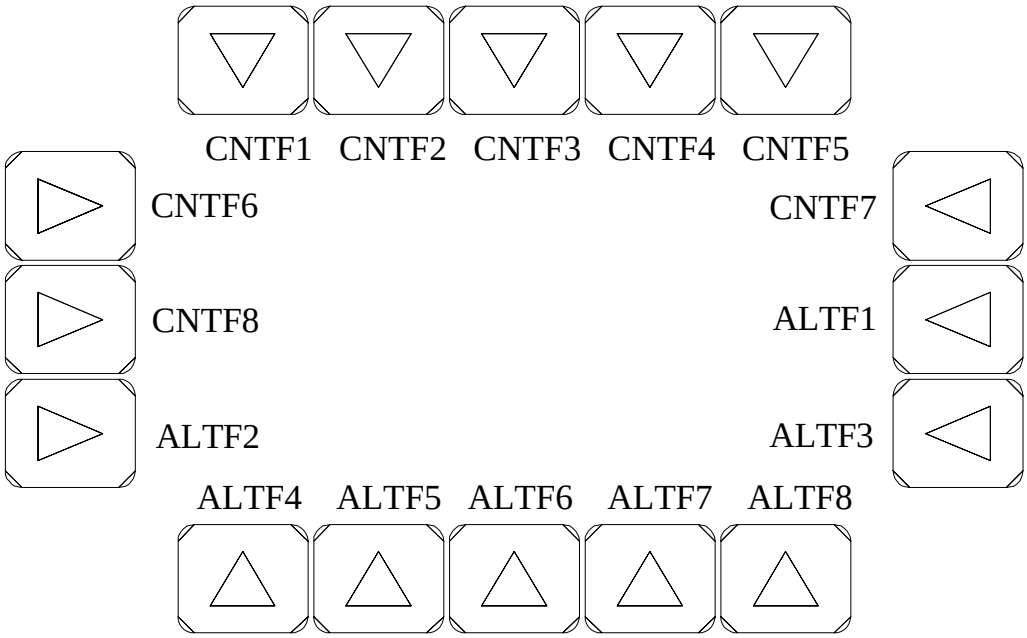
KEYPAD OPERATION

GENERAL

The FPXX has a built in 16 key membrane switch that surrounds it's LCD panel. This keypad array is intended for use as soft labeled keys (keys labeled by the display). The default BIOS setup enables keypad scanning, and converts keypad switch operations into standard PC keycodes, (as read via int 16). Key up and down events can also call a user supplied event handler if desired.

KEYPAD CODES

The key pad switch returns control and ALT function key keycodes:



DISPKEYS

DISPKEYS is a simple utility provided with the FPXX for the purpose of displaying keyboard scancodes and keynames. You exit DISPKEYS by pressing the same key 5 times in a row.

REFERENCE INFORMATION

SPECIFICATIONS

	MIN	MAX	UNIT	
POWER SUPPLY:				
Voltage	4.75	5.25	V	
Supply current	---	250	mA	(Backlight off)
Supply current	---	350	mA	(Backlight on)
EXPANSION BUS LOADING AND DRIVE:				
Input capacitance	---	15	pF	
Input leakage current	---	5	uA	
Output drive capability	100	---	pF	
Output sink current	---	6	mA	
ENVIRONMENTAL:				
Temperature range (display opr.)	0	45	°C	
Temperature range (display non-opr.)	0	70	°C	
Relative humidity	0	90	Percent Non-condensing	

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 DAMAGES, WHETHER IN CONTRACT, TORT, OR OTHERWISE.

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.

REFERENCE INFORMATION

REFERENCE INFORMATION

SCHEMATICS

REFERENCE INFORMATION

MECHANICAL DRAWINGS