FPCX USERS MANUAL

Version 1.3

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

INTRODUCTION

HARDWARE CONFIGURATION

CONNECTORS

INSTALLATION

CPU OPERATION

TABLE OF CONTENTS

CPU OPERATION

DISK EMULATOR OPERATION

LCD OPERATION

KEYPAD OPERATION

REFERENCE INFORMATION

HANDLING PRECAUTIONS

INTRODUCTION

LITHIUM CELL

The FPCX 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 FPCX.

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

GENERAL

STATIC ELECTRICITY

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

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

The FPCX is a small, low cost panel mount display computer for user interface applications. The overall dimensions of the FPCX are 4.6" H x 9.5"W x 1.5"D with the numeric keypad, and 4.6" H x 5.7"W x 1.5"D without. The FPCX is a complete OEM embedded system user interface CPU with display, keyboard, serial and parallel I/O, solid state disk, and LCD power supply.

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

The FPCX display is a 320 by 240 resolution monochrome, graphic, CCFL backlit LCD. Display dot pitch is .3 mm. Text display modes include a 20 character by 8 line mode (16x28 font), a 40 character by 17 line mode (8X14 font) and a 53 character by 30 line mode (5X7 font). Graphics are handled directly by the FPCX BIOS (drawdot, drawline, and bitblt.) A BGI graphics driver is supplied to support Borland compilers. Blinking graphics with no processor overhead are possible by using multiple frame buffers. Extra RAM can be used for additional screen buffers, allowing nearly instantaneous changing between screen layouts. FPCX display contrast and backlight intensity can be adjusted via built-in software commands. The keyswitch array surrounds the display area so that the keys can be labeled in the display.

The FPCX requires only +5V @ 850 mA max. for operation, since all display and RS-232 interface power are generated on card. The FPCX CPU is a 14.7 MHz PC compatible processor (NEC V40) with 512K to 832K of system RAM and a 192K BIOS EPROM.

The disk emulator has three 32 pin sockets for EPROM, 5V flash, or battery backed RAM disk. The on card disk emulator is supported by the FPCX BIOS, and appears to the system as a standard hard disk. Multiple drives are supported. All utilities for using the disk are provided with the FPCX.

On card I/O includes a battery backed clock/calendar, floppy interface, three serial ports, two of which are 16C550 compatible, a bi-directional parallel printer port, 8 user I/O bits, a standard PC keyboard port, and an 8 input, 10 or 12 bit A-D converter. Additional I/O can be added via the PC/104 expansion site on the back of the FPCX.

HARDWARE CONFIGURATION

GENERAL

The FPCX has hardware setup jumpers and I/O connectors accessible from the back side of the FPCX unit. When changing these option jumpers or installing I/O connectors. the FPCX 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 FPCX 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 FPCX.

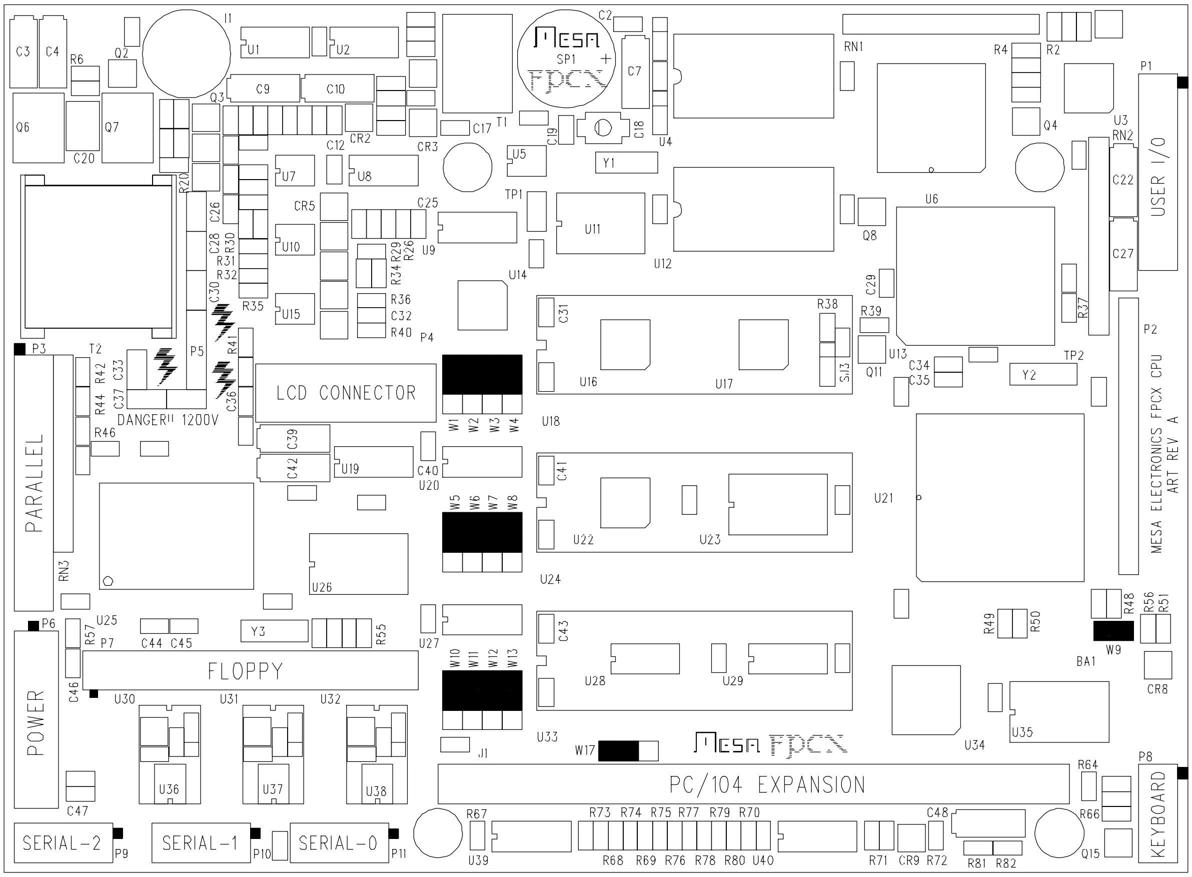
DEFAULT JUMPER SETTINGS

Factory default FPCX jumpering is as follows:

FUNCTION	**JUMPER(S)**	**SETTING**
Disk emulator U18	W1,W2,W3,W4	Flash EEPROM
Disk emulator U24	W5,W6,W7,W8	Flash EEPROM
Disk emulator U33	W10,W11,W12,W13	Flash EEPROM
Lithium cell connect	W9	Enabled
AEN source	W17	DACKS

HARDWARE CONFIGURATION

DEFAULT JUMPERING



HARDWARE CONFIGURATION

DISK EMULATOR TYPE SELECTION

The FPCX has three 32 pin sockets available disk emulator use. These sockets are above the PC/104 connector on the FPCX (U18 and U24 and U33). The disk emulator can use battery backed RAM, EPROM, or 5V flash EEPROM. Each socket can be configured independently. If 2 or more sockets use the same chip type, they can be combined into a single, larger disk emulator. For example. A system could be configured with a 512K flash EEPROM disk for program storage, and a 1M byte RAM disk for data collection.

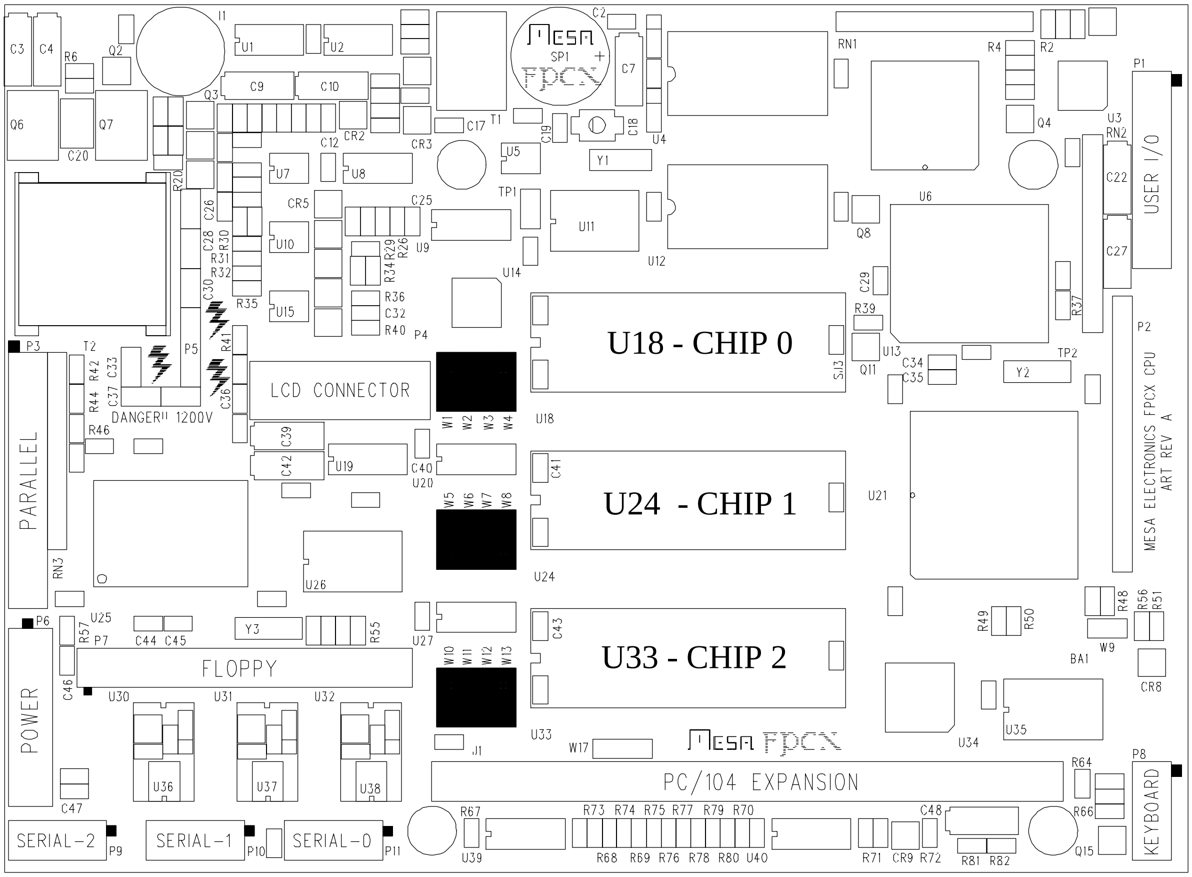
The FPCX disk emulator hardware needs to be configured to match the memory type used. Three sets of four jumper blocks configure the memory type. Jumpers W1,W2,W3, and W4 select U18's type, jumpers W5,W6,W7, and W8 select U24's type and finally jumpers and W10,W11,W12, and W13 select U33's type. These jumper blocks all have three pins and 2 valid shorting jumper positions, up and down. The following table shows the jumper setting for different disk emulator memory types:

MEMORY TYPE	**JUMPER POSITIONS**			
U18 type jumpers	W1	W2	W3	W4
U24 type jumpers	W5	W6	W7	W8
U33 type jumpers	W10	W11	W12	W13
Flash EEPROM (Default)	up	up	up	up
EPROM	up	up	up	up
RAM	down	down	down	down

For more information on disk emulator usage, see the CPU operation section of the manual

HARDWARE CONFIGURATION

DISK EMULATOR SOCKETS AND TYPE JUMPERS



HARDWARE CONFIGURATION

LITHIUM CELL CONNECT

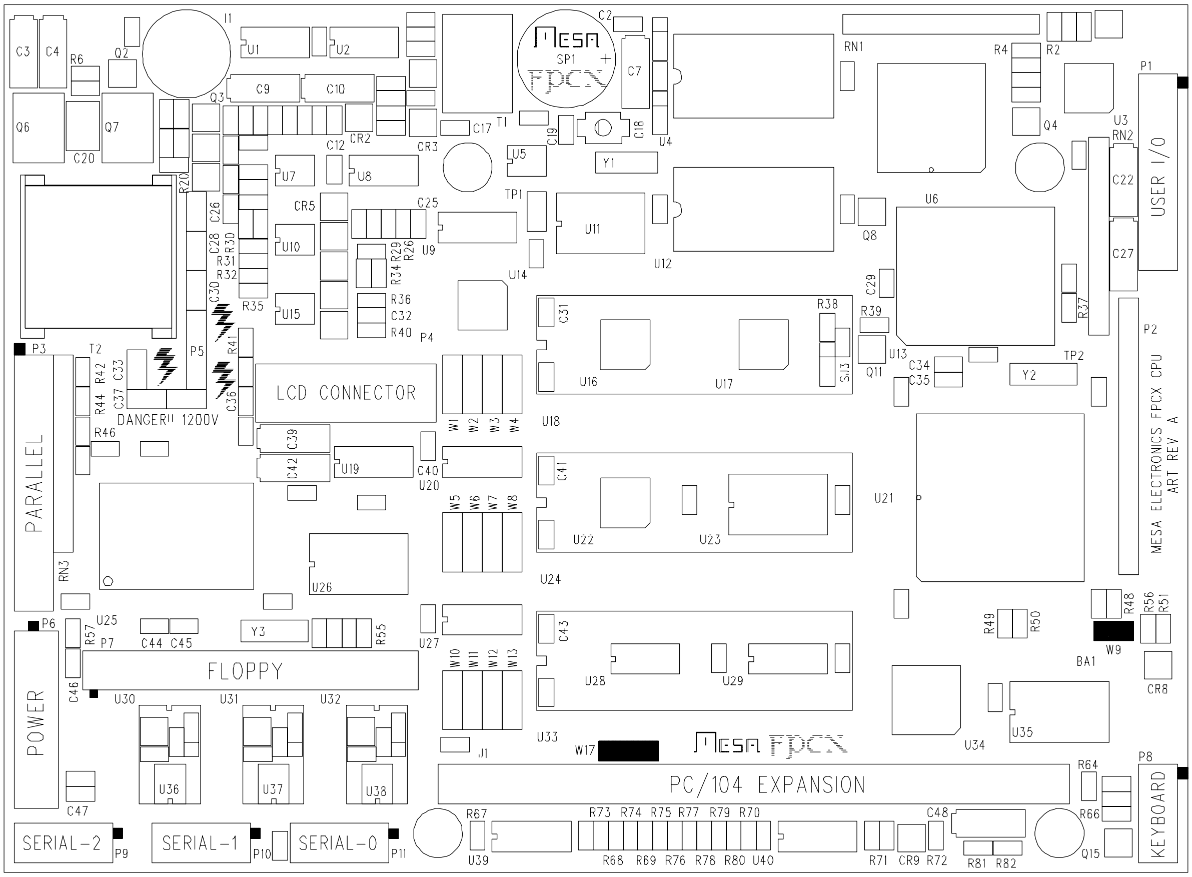
The FPCX's Lithium cell is disconnected for shipping and can be disconnected if long term storage is planned. The cell should also be disconnected when battery backed disk emulator RAMs are installed or removed. To disconnect the Lithium cell from all FPCX circuitry, remove the shorting jumper from jumper block W9. The shorting jumper can be left on one pin so that it doesn't become lost. ***When W9 is removed, the TOD clock calendar forgets the time and date, and any data in RAM disk emulators will be lost!***

AEN SOURCE

The AEN signal on the PC/104 expansion bus is normally generated as the nand function of the DACK signals. In rare cases this may cause timing problems with peripheral cards on the expansion bus. In this case, jumper W17 can be moved from the default left hand position (AEN from DACKS) to the right hand position (AEN from MIO).

HARDWARE CONFIGURATION

AEN SOURCE AND LITHIUM CELL CONNECT JUMPERS



CONNECTORS

POWER CONNECTOR

The FPCX power connector (P6) is a 9 pin, single row, .1" header. The suggested mating connector is an AMP MTA type connector 641191-9 (non-feedthrough) or 641199-9 (feedthrough). These are both gold plated type connectors for 24 AWG wire size.

Note that the +12 and -12 volt power pins on the power connector only supply the PC/104 expansion connector, and are not necessary for normal FPCX operation. Because only 5V is needed to operate the FPCX, A 4 pin power connector can be used that connects to the top 4 pins of the power connector. If a 4 pin connector is used, the suggested mating connector is an AMP MTA type connector 641191-4 (non-feedthrough) or 641199-4 (feedthrough).

Since the power connector on the FPCX powers the PC/104 expansion bus, it is suggested that only gold plated connectors be used. Tin plated connectors have a pronounced tendency to fail over time via increased contact resistance when anywhere near their rated current is drawn.

Power connector pinout is as follows:

PIN	SIGNAL	CURRENT RATING
1	+5V	1A
2	GND	1A
3	GND	1A
4	+5V	1A
5	GND	1A
6	-12	200 MA
7	+12	200 MA
8	GND	1A
9	KEY	XX

KEYBOARD CONNECTOR

P8 is the XT keyboard, reset-in and speaker connector. P8 is a 10 pin dual row .1" header. The suggested mating connector is AMP PN 499934-1. This is an IDC (flat cable) type connector.

The XT keyboard interface on the FPCX is intended mainly for development applications. The keyboard interface will not work with AT type keyboards.

An external reset switch input and speaker output are also available on P8. The reset circuit works by grounding the EXTPF 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. An external PNP transistor can be used to drive an eight Ohm speaker to obnoxious volume levels if required.

CONNECTORS

KEYBOARD CONNECTOR

Keyboard connector P8 pin-out is as follows:

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

Notice that the first 5 signals match the signal order on the XT keyboard. If a flat cable is used, the first 5 wires can be split off for connection to the keyboard. If you make your own keyboard adapter cable, make sure you get the connections to the DIN connector correct (the pins on the DIN connector are not in ascending sequence) A keyboard adapter cable is available from MESA.

SERIAL PORT CONNECTORS

P9,P10, and P11 are serial port connectors. P11 is the alternate console port, and uses the V40 CPUs built in serial port. P10 is referred to as SERIAL-1 and is a 16C550 compatible serial port with a default port assignment of COM1. P9 is referred to as SERIAL-2 and is a 16C550 compatible serial port with a default port assignment of COM2. All serial ports use a 10 pin dual row .1" header. The suggested mating connector is AMP PN 499934-1. This is an IDC (flat cable) type connector.

When the flat cable from a 10 pin serial port connector is terminated with a male 9 pin D type connector (suggested connector AMP 747306-4), the 9 pin connector will have a similar pin-out to the AT type 9 pin serial port. The pin 10 wire must be stripped from the cable before installing the D connector. A foot long serial port adapter cable, and a five foot long download cable are available from MESA.

CONNECTORS

SERIAL PORT CONNECTORS

Serial port connectors P9 and P10 pin-out 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

Serial port connector P11 pin-out is as follows:

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

NOTE: All serial ports 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).

CONNECTORS

PRINTER PORT CONNECTOR

The FPCX has a single printer port. The printer port connector is P3. P3 is a 26 pin, .1" header. Suggested mating connector is AMP PN 746285-6 When the flat cable from the 26 pin printer port connector is terminated with a female 25 pin D type connector (pin 26 unconnected) the 25 pin connector will have the standard PC printer port pinout.

HDR PIN	DSUB PIN	SIGNAL	FUNCTION
1	1	/PSTB	Strobe (out)
2	14	/PAFD	Auto LF (out)
3	2	PD0	Data 0
4	15	/PERROR	Printer error (in)
5	3	PD1	Data 1
6	16	/PINIT	Reset printer (out)
7	4	PD2	Data 2
8	17	/PSLIN	Select printer (out)
9	5	PD3	Data 3
10	18	GND	Ground
11	6	PD4	Data 4
12	19	GND	Ground
13	7	PD5	Data 5
14	20	GND	Ground
15	8	PD6	Data 6
16	21	GND	Ground
17	9	PD7	Data 7
18	22	GND	Ground
19	10	/PACK	Printer Ack (in)
20	23	GND	Ground
21	11	PBUSY	Data in (in)
22	24	GND	Ground
23	12	PPE	Paper out (in)
24	25	GND	Ground
25	13	PSLCT	Printer selected (in)
26	NC	+5V	Key

CONNECTORS

USER I/O

The FPCX has a 20 pin connector for USER I/O. This connector is P1. There are 8 analog inputs and 8 I/O bits available on P1. P1 pinout is as follows:

HDR PIN	SIGNAL	FUNCTION
1	AIN0	Analog input 0
2	AIN1	Analog input 1
3	AIN2	Analog input 2
4	AIN3	Analog input 3
5	AIN4	Analog input 4
6	AIN5	Analog input 5
7	AIN6	Analog input 6
8	AIN7	Analog input 7
9	GND	Signal ground
10	VCC	Power
11	PC0	Parallel bit 0
12	PC1	Parallel bit 1
13	PC2	Parallel bit 2
14	PC3	Parallel bit 3
15	PC4	Parallel bit 4
16	PC5	Parallel bit 5
17	PC6	Parallel bit 6
18	PC7	Parallel bit 7
19	GND	Signal ground
20	VCC	Power

FLOPPY CONNECTOR

Connector P7 is the floppy disk interface. The FPCX hardware currently supports 720K and 1.44M floppy drives. The FPCX supports two floppy drives, using the standard twisted floppy disk cable. P7 is a 34 pin connector with pin 4 missing.

PC/104 EXPANSION

The FPCX provides an 8 bit only PC/104 expansion connector for user supplied I/O cards. Note that +12 and - 12V power for the expansion connector comes from the FPCX's power connector P6.

INSTALLATION

GENERAL

When the FPCX has been properly configured for its application, expansion PC/104 cards can be installed on the back of the card. These need to be installed last as they cover the configuration jumpers and the serial-0 and floppy connectors. The PC/104 standoffs should then be tightened to secure the expansion card(s) in place.

MOUNTING

The FPCX is intended for panel mounting from behind a user supplied panel or on the front of a user supplied panel using the MESA supplied bezel. The FPCX bezel is a frame that surrounds the FPCX front panel , and clamps it to the user panel via screws inserted from the rear of the panel.

I/O CONNECTOR ORIENTATION

The serial port and keyboard connectors on the FPCX are 10 pin, .1" headers. The keyboard connector has pin 3 missing to prevent plugging the keyboard adapter cable on the serial port connector. Pin 10 of the serial port connector can be cut, and a keying plug installed in the cable mount header if desired. This will prevent plugging the serial cable on the keyboard port connector.

All connectors on the FPCX have their pin one ends marked with a white square on the circuit card. This corresponds with the red stripe on typical flat cable assemblies.

The parallel port on the FPCX is a 26 pin .1" header . If not used, the 5V output pin (pin 26) can be cut. If this is done, a key may be installed in the pin 26 location of the IDC cable mount receptacle to prevent reverse installation.

The power connector normally has pin 9 removed in order to act as a key. If the power or ground pins on the user I/O connector are not needed, one of them can be cut to serve as a key.

The PC/104 connector is not keyed, so be very careful not to install a PC/104 card off by one pin in either direction. This is especially important if +12V and - 12V power are supplied through P6.

CPU OPERATION

POWER CONSUMPTION

The FPCX is an all CMOS CPU, so overall power consumption of the is typically less than 4 watts (about 800 mA) running and about 280 mA with the backlight off. If the absolute lowest power consumption is required, the CPU can be halted when not active. 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 FPCX 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 FPCX 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 FPCX is provided with a built in watchdog timer that will reset the FPCX 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.

SERIAL PORTS

The FPCX has three serial ports, serial-0, serial-1 and serial-2. Serial-0 uses the V40's built in 8251 compatible serial port hardware. The FPCX BIOS has some INT 1A functions that support the V40 serial port, including baud rate setting.

The FPCX BIOS also supports using serial-0 as the alternate console port. An extended INT 1A function is used for console switching. The SERIAL.BAT file in the DEMO directory of the distribution disk will switch the console I/O from the LCD to the serial-0 port.

The V40SER.PAS file in the source directory of the distribution disk has some example V40 serial port interface routines. Since the V40's serial port is not 8250 (COMX) compatible, application programs that directly access the serial port will not work with the FPCX serial-0.

CPU OPERATION

SERIAL PORTS

The serial-1 and serial-2 ports are 16C550A (FIFOed) compatible serial ports. They are fully compatible with standard PC serial ports. The default port assignment sets serial-1 to be COM1 (Port 3F8H, IRQ4) and serial-2 to be COM2 (Port 2F8H, IRQ3). These default port locations can be changed if necessary with the supplied utility SETSUIO.EXE.

RS-485 OPERATION

The FPCX has assembly time options to allow any of the serial ports to be built with RS-485 interfaces 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, Serial-0 uses its DTR bit to control the transmit enable. Serial-0's DTR bit is accessed by I/O writes to bit 0 of location 34H. Low data enables transmit. Serial-1 and serial 2 use the RTS bits in their respective modem control registers to control transmit enable.

RS-485 communication on the FPCX 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 files V40SER.PAS (for serial-0) and SERIAL.PAS (for serial-1 and serial-2) in the source directory of the distribution disk have some low level serial port and RS-485 enable-disable procedures that can be used as examples for writing your own code.

PRINTER PORT

The FPCX printer port is a PC compatible port with bidirectional PS/2, EPP and ECP capabilities. The default port address is 278H. This can be changed with the supplied utility SETSUIO.EXE.

USER PARALLEL I/O

8 general purpose I/O bits are available on the USER I/O connector. These bits are port C of the 82C55 that is used for membrane keypad scanning. The base address of this 82C55 is 038H. The I/O bits are accessed at 03AH. Because port A and B of this chip must remain in the A=out, B= in mode, the only valid 82C55 mode register settings are: 082H (C= all out), 083H (C0-C3 = in C4-C7 =out), 08AH (C0-C3 = out C4-C7 = in), and 08BH (C=all in). For more detailed information you should consult an 82C55 data sheet.

CPU OPERATION

SERIAL FILE DOWNLOAD

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

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

The first requirement for SEND and RECV 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 FPCX 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   FPCX 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 FPCX end. This cable must have the following connections:

```
HOST 25 pin  FPCX 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 FPCX's 10 pin header and a 9 pin DB type connector on the host. SEND 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 SEND is running on the host machine, RECV is run on the client CPU card to download files. Note that RECV only works on MESA V40 based CPU's!

RECV is invoked this way:

```
RECV 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 SEND to be aborted when the file transfer is complete. BR is an optional baud rate parameter. If BR is not supplied, RECV uses 9600 baud. For example:

```
RECV FOO GOO 38400 Q
```

Would get the remote file FOO, write it to the local file GOO, and abort SEND 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 RECV baud rate. Maximum practical baud rate is 57600 baud. 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 optional tool chip supplied with ROM-DOS versions of the FPCX 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:RECV %1 %1 38400**

CPU OPERATION

CONSOLE SWITCHING

The FPCX can use an XT keyboard, the scanned keypad or the serial port for console input. Console out can be directed to a video card, the LCD module or the serial port. To determine which console option is used, the FPCX 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 XT 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 XT keyboard port.

The FPCX 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. There are some restrictions about combinations of input devices. The serial-0 port and the XT keyboard port use the same interrupt (IRQ 1), so they can not be enabled simultaneously.

USING THE A-D CONVERTER

The FPCX has a built-in 10 or 12 bit A-D converter. The FPCX BIOS uses the converter for automatic LCD contrast adjustment versus temperature, but there are 8 available inputs for user applications. The A-D converter is read with a BIOS call. This is the only safe way to access the A-D due to the fact that it is shared with BIOS background tasks. The BIOS A-D read function F_SYSATODRAWREAD always returns an unsigned 16 bit number (0 to 65535 full scale) regardless of A-D resolution. The reference voltage is 3.75V, so a full scale reading of 65535 represents an input of 3.75V.

The 10 bit converter is fairly noisy, so averaging may be necessary to obtain a clean signal. The 12 bit converter is much quieter, so averaging is not needed unless the input signal requires it.

The ADDEMO program is an simple example of using the A-D. The source code for ADDEMO is 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

EXTENDED INT 1A FUNCTIONS

Console I/O redirection, and some other miscellaneous control functions on the FPCX 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 FPCX 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 V40 SCU 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 V40 SCU 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 FPCX options can be saved in the serial EEPROM on the FPCX 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 SETFPCX.EXE

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

```
SETFPCX FPCX.CF
```

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

SETFPCX has three command line switches: /D, /N and /Q. These command line switches follow the file name. The /D option causes the FPCX 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 SETFPCX 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 FPCX operation.

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

DISK EMULATOR OPERATION

GENERAL

The FPCX has a built in nonvolatile disk emulator with a capacity of up to 1.5M bytes using battery backed RAM or FLASH EEPROM.

Three 32 pin sockets are provided on the FPCX for disk emulator memory chips. These three sockets can be used together as a single drive or each chip can be configured as an independent drive. When more than one drive is configured, the chips can be of different types. For example, a configuration with flash EEPROM for program storage and battery backed RAM for data logging is a popular arrangement.

When using the built in 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.)

The FPCX 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

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

This initialization is done with INITRAMD.EXE. INITRAMD.EXE is supplied in the DISK subdirectory of the FPCX distribution disk. If INITRAMD is run with a /L parameter, it will list the types of disk emulator chips supported by the FPCX BIOS. Each type of disk emulator chip has a corresponding Devicetype number.

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

```
INITRAMD /CStartChip /NNumberOfChips /DDeviceType [/F [O | D]]
```

Where StartChip is 0, 1, or 2, NumberOfChips is 1 to 3 and DeviceType is a number listed by the INITRAMD /L command.

The /F parameter invokes a built-in FDISK and FORMAT option. The /F option can be followed by a D or O. The D or O specifies whether a DOS format (D) or ROM-DOS format (O) is used.

The ROM-DOS format is specially optimized for small drives, and wastes a minimum of disk space. The DOS format wastes more space, but is compatible with most version of MS-DOS. ***Do not use the ROMDOS format with DOS or unpredictable file system behavior will result!***

On the FPCX, there are 3 available sockets, the upper socket (U18), the middle socket (U24), and the lower socket (U33). INITRAMD numbers these sockets such that the upper socket is socket 0 the middle socket is socket 1 and the lower socket is socket 2.

If you wanted to initialize a 2 socket disk emulator using device type 1, and starting at socket 0, the INITRAMD command would be:

```
INITRAMD /C0 /N2 /D1 /FD
```

(Initialize a disk emulator starting at socket 0, using 2 sockets and device type 1 - DOS format)

It is also possible to initialize two independent disk emulators by invoking INITRAMD twice, once per socket:

```
INITRAMD /C0 /N1 /D3 /FO
```

(Initialize disk starting at socket 0, using 1 socket and device type 3 - ROMDOS format)

```
INITRAMD /C1 /N2 /D5 /FO
```

(Initialize disk starting at socket 1, using 2 sockets and device type 5 - ROMDOS format)

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

Unlike previous MESA disk emulator software, you should not normally need to run FDISK or FORMAT as long as you specify the /F parameter when initializing a drive.

LCD OPERATION

GENERAL

The FPCX display is a 320H by 240V pixel monochrome LCD with a .3 mm pixel size. Overall active display dimensions are 3.78H by 2.83V The FPCX BIOS supports most of the standard PC character output functions in a manner suited to the smaller display.

SYSTEM RESOURCES NEEDED

The FPCX uses DMA channel 1 (DRQ1 and DACK1) and IRQ5 when driving the LCD panel. Make sure that you do not drive IRQ5 or DRQ1 when using the LCD panel or the display will malfunction.

System memory needs to be allocated for the display panel refresh buffer. This memory is allocated by the BIOS at system startup if the LCD is enabled. Approximately 20K bytes of memory are needed per plane on a FPCX display. Note that the number of planes parameter (1 or 2) is selected via the SET FPCX utility.

TEXT MODE

The FPCX BIOS supports 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 FPCX display is not hardware compatible with the standard monochrome or color IBM type displays, so software that directly accesses video memory will not work. The symptom of this problem is that the text simply disappears.

The FPCX display has no equivalent 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.

The FPCX 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 FPCX's BIOS function 0Ah simply writes characters with the default attributes.

MESA is working on a ANSI.SYS replacement that is compatible with the small LCD display environment for those text and remote terminal applications that need ANSI compatibility.

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 OPERATION

TEXT WINDOW

FPCX 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 FPCX 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. See the section EXTENDED INT 10 FUNCTIONS for more information.

TEXT SCREEN SIZES

These text sizes assume the entire display (320x240 pixels) is used for the TTY window.

5x7 font	-	30 lines of 53 characters
7x9 font	-	17 lines of 40 characters
14x18 font	-	8 lines of 20 characters

LCD OPERATION

TEXT ATTRIBUTES

The FPCX BIOS supports blink and reversed video attributes. These character attributes are selected with the standard INT 10 functions 9 (Write character and attribute) and 13 (Write string). Since the attribute information is not saved, INT 10 character or string output calls that do not specify the attribute byte will simply write with normal text. Underline and intensity attributes are ignored.

DISPLAY CONTRAST

The FPCX 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 FPCX.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 CCFL inverter on the FPCX can vary the backlight intensity and turn off the backlight after a period of inactivity if desired. The backlight intensity, backlight timeout value and backlight turn on events are specified in the FPCX.CF file.

LCD OPERATION

GRAPHICS

The FPCX 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.

If 2 planes are used, blinking graphic objects are supported. These are useful for calling attention to specific screen areas. Blinking graphic objects do not have any system performance overhead, as they are implemented by dynamically switching the display buffers. Using 2 planes will reduce drawing speed slightly, and doubles the size of the frame buffer.

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 FPCX 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 FPCX distribution disk.

BLT.EXE is a demonstration program that shows some of the FPCX BIOS's graphic capabilities. BLT.EXE requires only one plane but will show 2 plane operation if 2 planes are used.

SC.EXE is a program for setting the FPCX display contrast. SC.EXE uses the keypad for adjusting the contrast setting. The settings can be permanently saved by pressing the SAVE button, or restored to their previous values by pressing the OLD button.

ADDEMO.EXE reads the A-D converter channels and displays the output on bargraph type meters.

METEDEMO shows the four types of meter displays , text windows, and labeled boxes.

GRAPHIC LIBRARIES

The FPCX 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 FPCX 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 FPCX distribution floppy.

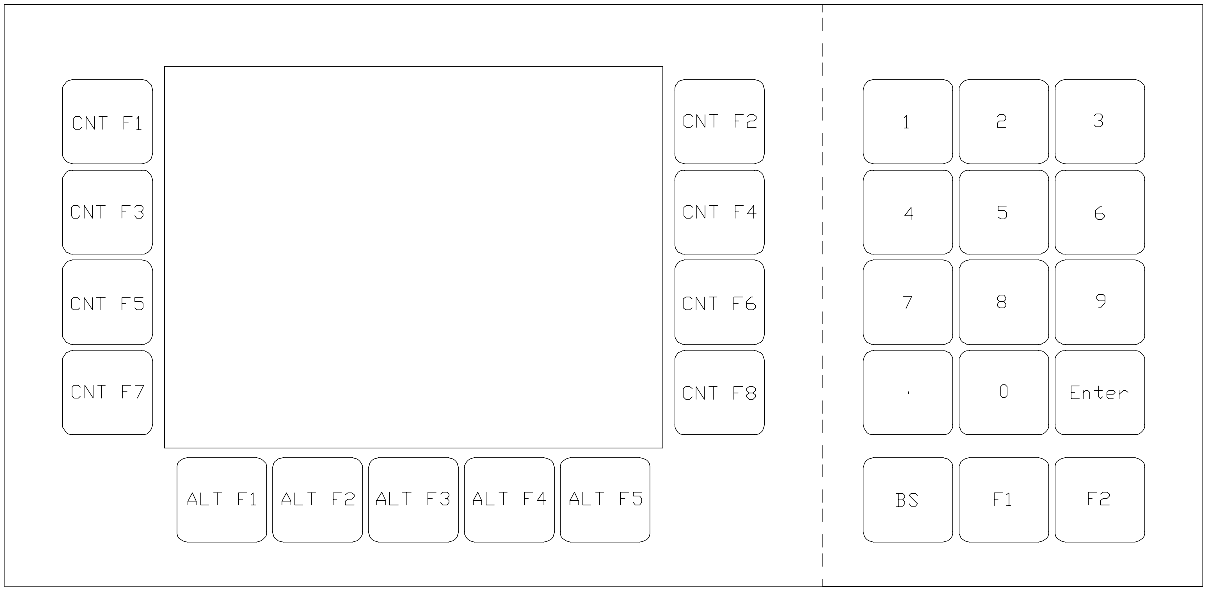
KEYPAD OPERATION

GENERAL

The FPCX has the a built-in membrane keypad with 13 display labeled keys and an optional 15 key numeric keypad. This is designed to simplify embedded instrument and controller applications where a standard keyboard is inappropriate.

The keypad scanning is enabled via an INT 1A function. Running the batch file KEYPAD.BAT gives a simple demonstration of keypad operation.

The BIOS returns a fixed set of scan codes from the keypad array. The default key codes returned by the keypad array are as follows:



DISPKEYS

DISPKEYS is a simple utility provided with the FPCX 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	---	300	mA	(Backlight off)
Supply current	---	850	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

MECHANICAL DRAWINGS