13 The Machine Pack.

The Machine Pack deals with the low level driving of the hardware. It also talks to the Centronics port (and hence the printer) and is in charge of running 'load and go' programs.

13.1 Hardware Interfaces.

The routines provided for driving the hardware are only to be used by those who understand the hardware and how the firmware drives the hardware. The user should not access the hardware directly when a Machine Pack routine is provided for this purpose.

Often there are higher level routines that accomplish the same effects but also keep the firmware informed of the current settings. Where possible these higher level routines should be used and the Machine Pack routines avoided. Using the Machine Pack routines may cause the firmware to make erroneous assumptions about the current settings and may cause it to go wrong.

The Machine Pack makes certain assumptions about the state of the hardware when it accesses it. In particular, PPI port A is assumed to be in output mode and the sound chip, ULA, CRTC and Centronics port are assumed to be inactive; that is, not halfway through setting a value into a chip register. It is usually essential that interrupts be disabled when accessing the hardware directly.

There are four main areas of the hardware that the Machine Pack deals with:

a. The screen.

There are three aspects of the screen display that can be set using the Machine Pack. These are the screen mode (set by calling MC SET MODE) and the screen base and offset (set by calling (MC SET OFFSET).

The screen mode sets how many pixels are displayed on the screen and how many inks may be used as follows:

Mode	Resolution	Inks
0	160x200	16
1	320x200	4
2	640x200	2

The screen base sets which 16K block of memory is used for the screen memory. Theoretically, any of #0000, #4000, #8000 or #C000 could be used but, in practice, other considerations mean that only #4000 and #C000 are useful.

The screen offset sets which byte in the screen memory is to be displayed first. Changing the screen offset will move the contents of the screen in one go. This is used for rolling the screen.

A fuller description of the screen layout and its relationship to these aspects can be found in section 6 on the Screen Pack.

If addresses are to be read back from the CRT controller chip, when using a light pen for instance, then careful inspection of the way the screen memory is addressed will be needed to translate the screen address read from the chip to the actual position on the screen.

The Machine Pack also provides a routine (MC WAIT FLYBACK) to wait until frame flyback occurs (the start of the vertical retrace period). This may be used to ensure that operations on the screen are performed with as little disruption as is possible to the picture on the monitor since no picture is generated during this period. As an alternative to waiting for frame flyback explicitly the user should consider setting up a frame flyback event as described in section 11.5.

The vertical retrace period is not very long. Furthermore, approximately 100 microseconds from its start, a time interrupt occurs that will cause the frame flyback events to be processed (see section 11). These may take a significant length of time out of the retrace period.

b. The inks.

The Machine Pack deals with setting the colours of inks. There is a fuller explanation of the relationship between inks and colours in section 6.2. Briefly, the colour for each ink and the border can be specified independently and changed at will. Note, however, that the Machine Pack deals with the hardware representations of colours and not the grey scale colours that the Screen Pack uses and also that an ink may only be set to one colour, the flashing inks are made by the Screen Pack setting two colours alternately.

Two routines are provided for setting the colours of inks. MC SET INKS allows the colours of all 16 inks and the border to be set (although not all of the inks may be visible on the screen in the current mode). MC CLEAR INKS sets the colour of the border ink and sets all 16 inks to the same colour. The latter is used when clearing the screen to make the operation appear instantaneous.

c. The sound chip.

A routine, MC SOUND REGISTER, is provided to write to a register of the sound chip. This is used by the Sound Manager for hardware access.

d. The Centronics port.

Two routines are provided to access the Centronics port. MC BUSY PRINTER tests if it is busy. MC SEND PRINTER strobes data out of it. Data should not be sent while the port is busy.

The Centronics port is used by the printer routines provided in the Machine Pack and described below.

13.2 The Printer.

There is a routine, MC PRINT CHAR, which calls an indirection, MC WAIT PRINTER, for sending characters to the printer, or rather, to the Centronics port. In V1.1 firmware, before sending the characters MC PRINT CHAR translates them using the printer translation table. The user can set the translations by calling MC PRINT TRANSLATION. The default translation table (see Appendix XIV) only affects characters in the range #A0..#AF and is designed to make these characters print more reasonably on the DMP-1 printer for various language options.

MC WAIT PRINTER waits until the Centronics port is not busy and then sends the given characters to it. If the port remains busy for a long time then the routine times out and returns indicating that it has failed to send the character. This time out can be used to prevent programs 'hanging' because they are waiting for a (possibly non-existent) printer to become ready.

MC WAIT PRINTER allows the user to intercept characters to be sent to the printer. This could allow special escape sequences to be inserted if needed, or it could allow the printer to be disabled or the length of the time out to be changed.

13.3 Loading and Running Programs.

The Machine Pack provides two routines for running programs, MC START PROGRAM and MC BOOT PROGRAM.

MC START PROGRAM is the simpler of the two routines. It completely re-initializes all the firmware and then enters the given program.

MC BOOT PROGRAM is more complex. It is for loading a program into RAM and running it. The user supplies a routine to MC BOOT PROGRAM that will load the program and return its entry point. Before this load routine is called as much of the firmware as is possible is reset so that the are of memory between #0040 and the base of the firmware RAM at #B100 is available for use. If the system were not reset then an active indirection, event or interrupt routine might be overwritten with disastrous consequences.

If the program is loaded successfully by MC BOOT PROGRAM then the firmware is completely initialized and the program is entered. However, if the loading fails then an appropriate message is printed and the previous foreground program is restarted. If the previous program was itself a RAM program then the default ROM is entered instead because it is likely that the previous program was corrupted when the attempt to load the new one was made.