AMSTRAD PERSONAL COMPUTER

PC1512

AMSTRAD PC TECHNICAL REFERENCE MANUAL

SOFT 50011

Preface

This Technical Reference Manual is intended primarily to assist writers of software for the Amstrad PC1512, although in conjunction with the PC1512 Service Manual it will be of interest to designers of add-on hardware.

It is assumed that the reader has a working knowledge of the Industry Standard architecture comprising of an 8086 (or 8088) with DMA, PIT, RTC and Interrupt Controller support chips; plus Colour Graphics Adapter with Floppy Disk, Serial and Parallel Adapters.

The information contained herein is largely unique to this document, with the exception of parts of Appendices 1, 2 & 3 which expand on the information contained in the PC1512 User Instructions and the Microsoft MSDOS Reference Manual.

Whilst the PC1512 implements a superset of the Industry Standard, this manual makes no attempt to identify those areas of the PC1512 specification which exceed the Industry Standard. Users should, therefore, exercise caution when writing software for a range of manufacturers' PCs and only use the "Lowest Common Denominator" facilities if simple portability is required.

All information of a technical nature and particulars of the product are given by AMSTRAD in good faith. However, it is acknowledged that there may be errors or omissions in this manual.

All correspondence should be addressed to:

AMSTRAD Consumer Electronics Plc
Brentwood House
169 Kings Road
Brentwood
ESSEX CM14 4EF

All maintenance and service on the product must be carried out by AMSTRAD authorised dealers. AMSTRAD cannot accept any liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This manual is intended to assist the reader in the use of the product, and therefore AMSTRAD shall not be liable for any damage or loss whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual or any incorrect use of the product.

Written by Bill Weidenauer, AMSTRAD Consumer Electronics Plc
Published by AMSTRAD

First Published 1986

Cover Index Contents

Table of Contents:

Section 1 - Hardware

Section 2 - Firmware

Section 3 - Tables

Appendices

Intro Index Section 1

SECTION 1 - HARDWARE

1.0 Introduction

This manual provides a comprehensive description of the AMSTRAD PC1512 hardware and firmware. General information about the PC1512, GEM Desktop and the delivered operating system software is contained in the AMSTRAD PC USER MANUAL. This manual is intended to satisfy the needs of advanced developers who must have access to the various resources available within the PC1512.

1.1 Central Processing Unit (CPU)

The CPU is an 8086-2 microprocessor with 1 Megabyte memory addressing capability (See Figure 1.1) running at a clock frequency of 8MHz. The CPU is connected to an on-board 16-bit system memory bus requiring four 125nS timing cycles (T-States) per access resulting in a 500nS memory cycle for 16-bit memory. The CPU is also connected on an on-board 8 bit I/O and memory peripheral bus with a 4 MHz clock, which in turn connects to an external (off-board) expansion bus. Operations on the 8-bit bus automatically incur 125nS wait states as follows:

Operation	**Wait States**	**Bus Cycle**
8-bit (Memory)	4	1.0 µS
16 to 8-bit convert (Memory)	12	2.0 µS
8-bit (I/O)	6	1.25 µS
16 to 8-bit convert (I/O)	16	2.5 µS

The CPU is configured to run in maximum mode and the instruction set may be optionally extended by the addition of an 8087-2 Numeric Data Coprocessor. The 8087 BUSY output is connected directly to the 8086 NOT TEST input.

1.2 MEMORY LAYOUT

The main board memory consists of 512K bytes of system RAM with parity checking and 16K bytes of system ROM without parity checking. Note that all address constants in this document are in hexadecimal form except as noted.

The 512K byte user RAM starts at CPU memory address 00000 and extends to 7FFFF. An additional 128K bytes may be added externally in 32K byte blocks in the address space from 80000 to 9FFFF to extend the total system RAM to 640K bytes, provided the 640K byte option is not already installed on-board.

The 128K byte address space from A0000 to BFFFF is reserved for the 8-bit memory associated with certain controllers, and is not used by CPU programs. The on-board Alpha/Graphics VDU controller uses the 16K byte address range from B8000 to BBFFF for screen refresh memory. Also an optional external monochrome alphanumeric VDU controller uses the 4K memory address range from B0000 to B0FFF.

The 192K byte address space from C0000 to EFFFF is reserved for external expansion ROM address space. The optional Hard Disk controller uses the 20K byte address range from C8000 to CBFFF.

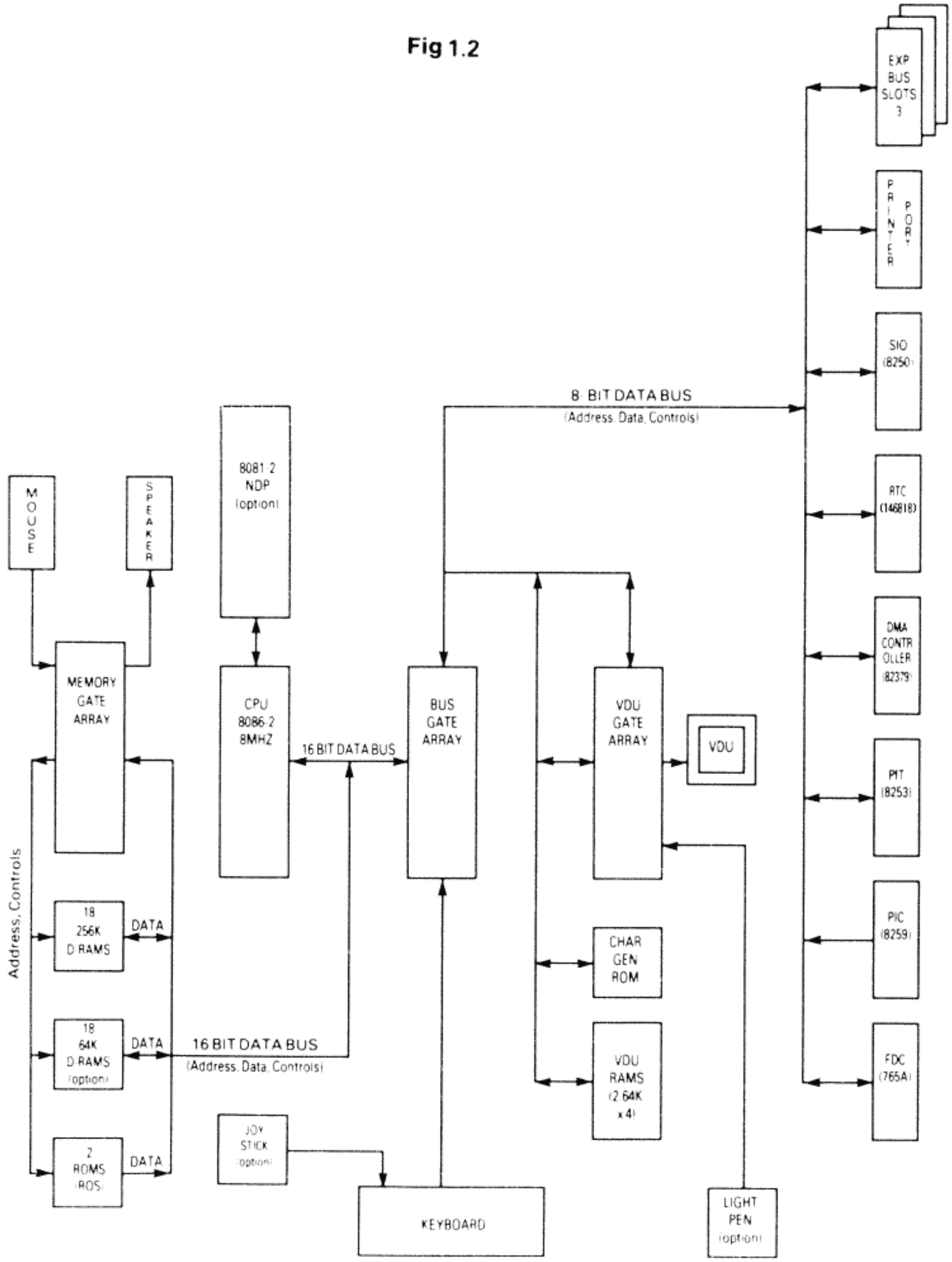
The 16K byte system ROM is at FC000 to FFFFF and contains the Resident Operating System (ROS) firmware. The 48K byte address range from F0000 to FBFFF is reserved for ROM space expansion.

FIGURE 1.1 - MEMORY LAYOUT

FFFFF	FC000	16K BYTES (ROS) RESIDENT OPERATING SYSTEM ROM	64K BYTE SYSTEM ROM AREA
	FBFFF F0000	48K BYTES RESERVED	
	EFFFF	192K BYTES EXPANSION ROMS	
		C8000 - CBFFF: HD ROM	
	C0000		
1M BYTE ADDRESS RANGE	BFFFF	128K BYTES 8-BIT CONTROLLER MEMORY	
		B8000 - BBFFF: VDU SCREEN	
	A0000	A0000 - BFFFF: EGA SCREEN	
	9FFFF 80000	128K BYTES ADDITIONAL USER MEMORY (EXTERNAL 32K BYTE BLOCKS)	640K BYTE USER MEMORY RANGE
	7FFFF	512K BYTES	

00000	ON-BOARD USER RAM	

Fig 1.2



1.3 MAIN BOARD I/O CHANNELS

The interfaces on the main board occupy the 8086 I/O addresses as follows:

ADDRESS(hex)	OUTPUT USE	INPUT USE
000 - 00F	8237 DMA Controller	8237 DMA Controller
010 - 01F	Do Not Use	Do Not Use
020 - 021	8259 Interrupt control	8259 Interrupt control
022 - 03F	Do Not Use	Do Not Use
040 - 042	8253 PIT Load Count (0-2)	8253 PIT Read Count (0-2)
043	8253 PIT Load Mode	Undefined
044 - 05F	Do Not Use	Do Not Use

ADDRESS(hex)	OUTPUT USE	INPUT USE
060	No Effect	Port A - Keyboard Code or System Status 1
061	Port B - System Control	Port B - (Readback)
062	No Effect	Port C - System Status-2
063	No Effect	Do Not Use
064	Write System Status-1	Do Not Use
065	Write System Status-2	Do Not Use
066	System Reset	Do Not Use
067 - 06F	Do Not Use	Do Not Use
070	146818 RTC Address	Do Not Use
071	146818 RTC Data	146818 RTC Data
072 - 077	Do Not Use	Do Not Use
078	Clear Mouse X-Coordinate	Mouse X-Coordinate
079	Do Not Use	Do Not Use
07A	Clear Mouse Y-Coordinate	Mouse Y-Coordinate
07B - 07F	Do Not Use	Do Not Use
080	Do Not Use	Do Not Use
081	DMA Page Register Ch 2	Do Not Use
082	DMA Page Register Ch 3	Do Not Use
083	DMA Page Register Ch 0,1	Do Not Use
084 - 09F	Do Not Use	Do Not Use
0A0	NMI Mask Control	Do Not Use
0A1 - 0BF	Do Not Use	Do Not Use
378	Printer Data Latch	Printer Data Latch
379	Do Not Use	Printer Status
37A	Printer Control Latch	Printer Control Latch
37B - 37F	Do Not Use	Do Not Use
3D0	CRTC Address Register	Do Not Use
3D1	CRTC Data	CRTC Data
3D2 - 3D7	Do Not Use	Do Not Use
3D8	VDU Mode Control	Do Not Use
3D9	VDU Colour Select	Do Not Use
3DA	Do Not Use	VDU Status
3DB	Clear Light Pen Latch	Do Not Use
3DC	Set Light Pen Latch	Do Not Use
3DD	VDU Colour Plane Write	Do Not Use
3DE	VDU Colour Plane Read	Do Not Use
3DF	VDU Graphics Mode 2 Border	Reserved
3F0 - 3F1	Do Not Use	Do Not Use
3F2	Drive Selection	Do Not Use
3F3	Do Not Use	Do Not Use
3F4	Do Not Use	765 FDC Status
3F5	765 FDC Data	765 FDC Data
3F6 - 3F7	Do Not Use	Do Not Use
3F8 - 3FF	8250 UART Tx Data/Control	8250 UART Rx Data/Control

Note: I/O addresses above 3FF, if accessed, wrap around and are mapped onto the range 000-3FF. Any of the unlisted addresses may be used in the future.

1.4 EXPANSION BUS I/O CHANNELS

The 8086 CPU I/O addresses on the expansion bus are as follows:

ADDRESS(hex)	USE
200 - 20F	External Game Control Interface
210 - 217	External Bus Expansion Unit
278 - 27F	External Printer Port
2B0 - 2DF	External Secondary Enhanced Graphics Controller
2F8 - 2FF	External Asynchronous Serial RS232C Port
300 - 31F	External Prototyping Card
320 - 32F	External Hard Disk Controller
380 - 38C	External SDLC Serial RS232C Port or Extended BSC Controller
390 - 393	External Cluster Controller
3B0 - 3BB	External Monochrome VDU Controller
3BC - 3BF	Printer Port
3C0 - 3CF	External Graphics Controller

I/O Addresses wrap around above 3FF occurs as described on page 10.

External cluster controllers at 790-793, B90-B93, 1390-1398 and 2390-2393 wrap around to I/O addresses 390-393

1.5 Direct Memory Access (DMA)

The AMSTRAD PC supports four DMA channels on the system board, using an 8237-4 DMA controller and programmable page registers to extend its addressing range from 64k bytes to 1M bytes. Each channel is able to transfer data in blocks of up to 64K bytes within a page. The DMA channels are for 8-bit data transfers between (8-bit) I/O devices and 8-bit or 16-bit memory.

In peripheral (slave) mode, CPU I/O address lines A0 - A3 are connected conventionally so that 16 command codes appear in the order described in the 8237 data sheets (See section 3.5).

The DMA controller CLK is driven at 4MHz (± 0.1%). In master mode during DMA transfers on channels 1,2 and 3, one wait state is added resulting in a five-clock DMA bus cycle of 1.25uS. Channel 0 transfers have a four-clock bus cycle of 1uS.

The DMA channel request signals are as follows:

DMA Channel	USE
0	8253 Timer/Counter OUT1 output - for memory refresh.
1	Spare for use by expansion bus. Used by external SDLC Serial Port.
2	765 Floppy Disk Controller DRQ output. Available on the expansion bus.
3	Spare for use by expansion bus. Used by external Hard Disk Controller.

1.5.1 DMA Page Registers

DMA channels 1, 2 and 3 can address the entire 1M byte addressing range of the 8086 CPU through the use of their associated DMA page registers. There are three DMA registers, one each for channels 1 through 3. Each page register defines for its channel which one of sixteen 64K byte pages in the 1M byte address range DMA transfers are to occur. The page registers are static so that modulo 64K byte addressing occurs at page boundaries. They are located in the I/O address space in the range 081-083.

The DMA page register bit assignments are as follows:

Bit	Output Use
7	Not Connected
6	Not Connected
5	Not Connected
4	Not Connected
3	Address bit A19
2	Address bit A18
1	Address bit A17
0	Address bit A16

1.5.1 DMA Initialisation

Following a reset, system initialisation firmware (in the ROS) sets up the 8237 for channel 0 (dynamic refresh) operation as follows:

Function	Initialised State
Word Count	64K Transfers
Mode Register	Read
	Autoinitialise
	Increment
	Single Mode
Command Register	Disable Memory to Memory
	Enable Controller
	Normal Timing
	Fixed Priority
	Late Write
	DREQ Active High
	DACK Active Low
Mask Register	Clear Channel 0 Mask Bit

After power-up or system reset the DMA page registers are undefined and are initialised to zero by the ROS firmware and all 8237 internal locations for channels 1-3 are initialised to a state comparable to the channel zero initialisation above.

Following industry compatibility, memory to memory DMA is not supported on the PC1512. It is prohibited due to timing considerations.

1.6 System Interrupts

Nine levels of hardware interrupt are provided for in the system by the CPU Non Maskable Interrupt (NMI) and by an 8259A-2 Interrupt Controller. All levels including NMI, are maskable under software control.

CPU I/O address line A0 is connected conventionally so that the command codes appear in the order described in the 8259 data sheets. The SP/EN pin is tied high signifying that the device is to be hardware un-buffered and designated as a master, not a slave.

1.6.1 Interrupt Levels

The interrupt levels are assigned as follows:

Level	Assigned Function
NMI	Memory Parity Error and 8087 NDP INT output.
0	8253 Timer/Counter Out0 output.
1	Keyboard Scan Code Receiver.
2	146818 Real Time Clock IRQ output. Available on expansion bus. May be used by Enhanced Graphics Adapter
3	Spare for use by expansion bus. Used by external (secondary) Asynchronous Serial Port and external SDLC Serial Port.
4	8250 UART INTRPT output. Available on expansion bus. Used by external SDLC Serial Port.
5	Spare for use by expansion bus. Used by external Hard Disk Controller.
6	765 Floppy Disk Controller INT output. Available on the expansion bus.
7	Parallel Printer Port. Available on the expansion bus. Used by external Printer Port (secondary) Printer Port (tertiary), external Monochrome VDU Controller, and external cluster controllers.

1.6.2 Interrupt Controller Initialisation

Following a reset, the initialisation firmware in the ROS sets the 8259 Interrupt Controller to operate as follows:

8086 system,
Single (not cascaded),
Normal fully nested (not special),
Edge-triggered,
Buffered mode - slave,
Normal EOI (not auto),
Fixed priority - level 0 highest, level 7 lowest.

1.6.3 NMI Mask Control

The NMI Mask Control is a write only register at I/O address 0A0 and allows the CPU non-maskable interrupt (NMI) input to be enabled or disabled by software. The Bit assignments are as follows:

Bit	Output Use
7	Enable NMI.
6 - 0	Not Connected

Following a reset NMI is disabled.

NMI can be connected to the 8087 NDP, the on-board memory parity check circuit, and the expansion bus I/OCHCK (I/O Channel Check).

1.7 Programmable Interval Timers

Three programmable timer/counters are provided at I/O Addresses 040 - 043 by an 8253 Programmable Interval Timer (PIT) device. They are defined as follows:

Counter	Use
0	General Purpose Timer.
1	Used by DMA channel 0 (for dynamic ram refresh).
2	Tone Generation for Speaker.

1.7.1 Timer Configuration

The 8253 timers are configured as follows:

Function	Configuration

Function	Configuration
CLK 0,1,2	1.193 MHz (± 0.1%)
GATE 0,1	Always 'ON'.
GATE 2	Controlled via Port B (System Control Channel) Speaker Modulate output.
OUT 0	Interrupts on 8259 PIC IR0 input.
OUT 1	Requests on 8237 DMA DREQ0 input.
OUT 2	Logical 'AND' with Port B (System Control Channel) Speaker Drive output. Also goes to Port C (System Status-2 Channel) as an input.

1.7.2 Counter 1 initialisation

Following a reset, the system initialisation firmware in the ROS programs the 8253 PIT for counter 1 (dynamic ram refresh) operation as a rate generator producing a signal with a period of 15.13 uS. There are no restrictions requiring the initialisation and programming of counters 0 and 2.

1.8 System Status and Control

Two system status input channels and four output channels are provided on-board. Ports A, B and C emulate a pre-programmed 8255 PPI device. They are located in the I/O address space in the range 060 - 06F. Port B is programmed for control output, Port A is programmed either for Status-1 input or for receiving data from the keyboard, and Port C is programmed for Status-2 input.

Ports A, B and C emulate an 8255 PPI that has been set up as follows:

Group A Mode 0,	Group B Mode 0,
Port A = input,	Port B = output,
Port C(U) = input.	Port C(L) = input.

Unlike an 8255, power-up and reset do not affect this configuration.

1.8.1 Port B - System Control

The System Control channel is located at I/O Address 061. Its bit assignment is as follows:

Bit (PBn)	Output Use
7	Enable Status-1/Disable Keyboard Code on Port A.
6	Enable incoming Keyboard Clock.
5	Prevent external parity errors from causing NMI.
4	Disable parity checking of on-board system Ram.
3	Undefined (Not Connected).
2	Enable Port C LSB / Disable MSB. (See 1.8.3)
1	Speaker Drive.
0	8253 GATE 2 (Speaker Modulate).

When bit 7 is set high (1), Status-1 data is enabled on Port A, the keyboard data path and keyboard interrupts are disabled. When bit 7 is set low (0), keyboard input data is enabled on port A, the keyboard data path and keyboard interrupts are enabled.

The keyboard interface operates as follows: Each incoming keycode is latched on-board, causing a keyboard interrupt (on level 1). While the interrupt remains pending, the incoming keyboard data signal is forced low as an acknowledgement to the keyboard that the keycode has been received. As soon as the interrupt has been cleared, the keyboard may use the Data signal to transmit the next keycode.

PB1 may be toggled to drive the speaker with a corresponding pulse train. The speaker may also be driven by a wave form from the 8253 PIT OUT2 output (simultaneously with PB1, if required).

PB0 may be toggled to drive the 8253 gate input, hence modulate counter 2 operations and therefore the speaker with a corresponding waveform. Thus there are three different methods of driving the speaker which may all be performed simultaneously to create various audio effects.

1.8.2 Port A - Status-1 Input/Keyboard Code

Port A is a read only location located at I/O Address 060. When PB7 is set to high (1) reading Port A loads Status-1. When PB7 is set low (0) reading Port A loads keyboard data. The bit assignments for port A are as follows:

Bit (PAn)	Status-1	Keyboard Input
7	Always 0.	KBD7
6	Second Floppy disk drive installed.	KBD6
5	DDM1 - Default Display Mode bit 1.	KBD5
4	DDM0 - Default Display Mode bit 0.	KBD4
3	Always 1.	KBD3
2	Always 1.	KBD2

Bit (PAn)	Status-1	Keyboard Input
1	8087 NDP installed.	KBD1
0	Always 1.	KBD0

The Default Display Mode bits (DDM1, DDM0) are set up by the ROS during system initialisation based on Non-Volatile Ram (NVR) selections as follows:

01 (1)
Colour, alpha, 40 X 25 chars, bright white on black.
10 (2)
Colour, alpha, 80 X 25 chars, bright white on black.
11 (3)
External Monochrome controller, 80 X 25 chars.

Following a reset, the hardware selects VDU mode 2. The ROS then sets the initial VDU state based on the DDM value.

1.8.3 Port C - Status-2 Input

Port C is a read only location located at I/O Address 062. Its bit assignments are as follows:

Bit (PCn)	Input Use	
7	On-board system RAM parity error.	
6	External parity error (I/OCHCK from expansion bus).	
5	8253 PIT OUT2 output.	
4	Undefined (Not Connected).	
	LSB or	MSB (depends on PB2)
3	RAM3	Undefined
2	RAM2	Undefined
1	RAM1	Undefined
0	RAM0	RAM4

PC7 is forced to the zero state when on-board system RAM parity checking is disabled by PB4.

When the I/OCHCK condition (external parity error) from the expansion bus is disabled from causing NMI (by PB5 set high), PC6 reflects the state of the I/OCHCK input else it reflects the latched state of I/OCHCK.

The value of RAM4-RAM0 denotes the amount of system RAM fitted to the system as follows:

RAM4	RAM3	RAM2	RAM1	RAM0	
0	1	1	1	0	**512K bytes on-board.**
0	1	1	1	1	**544K bytes (32K external).**
1	0	0	0	0	**576K bytes (64K external).**
1	0	0	0	1	**608K bytes (96K external).**
1	0	0	1	0	**640K bytes (128K external or fitted on-board).**

1.8.4 Write System Status-1

The Write System Status-1 register is a write only register at I/O Address 064 and is initialised by the Resident Operating System (ROS) firmware based on values obtained from the Non-Volatile Ram (NVR). It is used in conjunction with the 8255 PPI Port A emulation. The bit assignments are as follows:

Bit	Output Use
7	No effect.
6	PA6 - Second Floppy disk drive installed.
5	PA5 - DDM1.
4	PA4 - DDM0.
3	No effect.
2	No effect.
1	PA1 - 8087 NDP installed.
0	No effect.

1.8.5 Write System Status-2

The Write System Status-2 register is a write only register at I/O Address 065 and is initialised by the Resident Operating System (ROS) firmware based on values obtained from the Non-Volatile Ram (NVR). It is used in conjunction with the 8255 PPI Port C emulation. The bit assignments are as follows:

Bit	Output Use
7	PC2 (MSB) - Undefined.

Bit	Output Use
6	PC1 (MSB) - Undefined.
5	PC0 (MSB) - Undefined.
4	PC3 (MSB) - RAM4.
3	PC3 (LSB) - RAM3.
2	PC2 (LSB) - RAM2.
1	PC1 (LSB) - RAM1.
0	PC0 (LSB) - RAM0.

1.8.6 System Reset

Any write access to I/O Address 066 regardless of the value written will cause the hardware to generate an immediate 512uS system reset and pulse the reset line on the expansion bus. The contents of the on-board system RAM is preserved following a system reset.

1.9 Real Time Clock

HD146818 Real Time Clock plus RAM device is installed and backed up by a set of four non-rechargable size AA batteries. The clock device provides a time of day clock with alarm, a one hundred year calendar, a programmable periodic interrupt, and 50 bytes of static RAM. The static RAM is called the Non-Volatile RAM (NVR) and used to store system configuration data such as number of disk drives, memory size, serial I/O parameters, and default VDU screen mode. The ROS firmware maintains a checksum of the NVR and will reset the configuration data to "sensible values" during startup whenever the checksum value is incorrect (thus destroying your actual configuration). Even though direct hardware access to the NVR is possible it is recommended that the programs make use of the ROS Enhanced Function Interrupt (Interrupt 21) to access the NVR because these properly maintain the NVR checksum value.

When system power is off and the 146818 is on battery backup power, the functions which remain active are the clock and the retention of RAM data. No battery power is used while the system power is on.

The input crystal oscillator runs at 32.768 KHz.

The 146818 interrupt request is connected to the 8259 system interrupt controller on level 2 (which is also available on the expansion bus). The 146818 power-sense input PS is connected to a battery condition sensor. When the backup battery voltage is sufficiently low, the VRT bit in register D becomes set indicating that the time, the calendar and the NVR data are no longer valid. When this condition is noted during startup, the firmware outputs the message "Please fit new batteries" and resets the NVR to default values (See section 2.4).

All the features described in the 146818 data sheet are available with the exceptions that the CKOUT (clock output) and SQW (square wave output) pins are not connected on the main board.

Writing or reading the NVR involves a two step sequence for each byte that is accessed. The RTC Address channel (I/O Address 070) is first loaded with the NVR location to be accessed. Then the RTC Data channel (I/O Address 071) is either written or read to complete the I/O operation. This facility should be used with caution in order to avoid disturbing the system configuration data.

1.10 Parallel Printer Port

The printer port provides an interface for driving 8-bit and 7-bit Centronics compatible printers. The timing of the signals to the printer is under direct software control. There is a read/write control latch for sending control signals to the printer, an unlatched read-only printer status channel, and a read/write data latch for sending printer data.

1.10.1 Printer Data Latch

The printer data latch is a read/write record at I/O address 378 and its layout is as follows:

Bit (Dn)	Output/Input Use	Cable Polarity
7	Data 7	Hi
6	Data 6	Hi
5	Data 5	Hi
4	Data 4	Hi
3	Data 3	Hi
2	Data 2	Hi
1	Data 1	Hi
0	Data 0	Hi

The contents of the data latch are undefined following a power-up or system reset.

1.10.2 Printer Control Latch

The printer control latch is a read/write record at I/O address 37A and its layout is as follows:

Bit	Output/Input Use	Reset State	Cable Polarity
7-5	No effect		

Bit	Output/Input Use	Reset State	Cable Polarity
4	Enable Int on ACK	False	
3	Select Printer	False	
2	Reset Printer	True	
1	Select Auto Feed	False	
0	Data Strobe	False	

When Interrupt on ACK is enabled an incoming Printer Acknowledge condition will cause a system interrupt on level 7 (which is also available on the expansion bus).

If the printer control lines normally driven via latched bits D0 - D3 are driven externally, the data read on input to this channel will be the logical OR of the latched bits and the externally driven bits, e.g. If a data bit is false and the corresponding cable bit is driven true by the external driver, the bit input will be true.

Following power-up or system reset, the control latch contents assume reset conditions as shown.

Note that this is a general purpose printer interface and that not all printers require all the control signals, hence the provision for non-standard printers to be able to drive some of the control signals as inputs to the main board. The timing requirements on Centronics compatible printers generally specify that data must be present at 1uS before the strobe is made active, and must remain valid for at least 1uS after strobe goes inactive. The strobe duration must be between 1uS and 500 uS. Printer Busy status can be inspected as soon as the strobe is inactive in order to determine when more data can be sent.

1.10.3 Printer Status Channel

The Printer Status Channel is a read only register at I/O Address 379 (hex). Its layout is as follows:

Bit	Input Use	Cable Polarity
7	Printer Busy	Hi
6	Printer Acknowledge	Lo
5	Paper Out	Hi
4	Printer Selected	Hi
3	Printer Error	Lo
2	LK3 fitted	
1	LK2 fitted	
0	LK1 fitted	

LK1 - LK3 are general purpose factory installed option links on the main board which are used by the firmware to distinguish different machine configurations. The first seven states (0 - 6) are used for language variants and the all ones (7) state is used for Power-Up Self Test Maintenance Mode (See section section 2.2).

Note that this is a general purpose printer interface and that not all printers implement all the status lines, nor do they all attach the same meanings to the error conditions.

Printer Busy normally indicates that a printer cannot receive data, for example during data entry, printing, when offline, or during a printer error condition.

Printer Acknowledge, if implemented is generally asserted by a printer to indicate that data has been received and the printer is ready to receive the next data. Note that Printer Acknowledge (ACK) can also be set to cause interrupts (See 1.10.2).

Section 1.14 contains the printer connector pin assignments.

1.11 Alpha/Graphics Colour VDU Controller

The VDU controller is implemented as a partially emulated MC6845 CRTC and provides either a colour alphanumeric display (Alpha) or a colour pixel display (Graphics). The frame rate is 60 Hz non-interlaced and 200 scan lines are displayed between the top and the bottom borders. The highest resolution on a scan line between the left and the right borders is 640 pixels displayed.

The 16K Byte VDU screen memory area (from B8000 to BBFFF) is overlaid (in Graphics modes) in four planes giving a resultant 64K bytes of display RAM and enabling sixteen colours to be available.

The sixteen colours available on the display are as follows:

Red	Green	Blue	Intensity	Luminance	Colour
0	0	0	0	0	Black
0	0	1	0	1	Blue
0	1	0	0	4	Green
0	1	1	0	5	Cyan
1	0	0	0	2	Red
1	0	1	0	3	Magenta
1	1	0	0	6	Brown

Red	Green	Blue	Intensity	Luminance	Colour
1	1	1	0	7	White
0	0	0	1	8	Grey
0	0	1	1	9	Light Blue
0	1	0	1	12	Light Green
0	1	1	1	13	Light Cyan
1	0	0	1	10	Light Red
1	0	1	1	11	Light Magenta
1	1	0	1	14	Yellow
1	1	1	1	15	Intense White

The "Light" colours imply higher intensity video.

On the monochrome monitor the R,G,B and I signals are summed and weighted 2, 4, 1 and 8 respectively to form the 16 level grey scale luminance column in the table.

1.11.1 Alpha Display

Two Alpha modes are available: either (medium resolution) 40 characters by 25 rows or (high resolution) 80 characters by 25 rows. The modes require 2000 bytes or 4000 bytes of display RAM respectively. A low resolution 160 by 100 pixels 16 colour graphics mode may be obtained in high resolution Alpha mode by programming the CRTC for two scan lines per character, and by using certain half-block characters in the character set as pixels. This mode requires 16000 bytes of display RAM.

The character set is formed by a ROM character generator and each of the 256 characters is made up of a 8 by 8 pixel matrix. Each character displayed takes up two bytes of display RAM consisting of a character code byte and an associated attribute byte. The attribute byte allows a choice of either 16 foreground and 8 background colours per character, plus blinking, or a choice of 16 colours for both foreground and background without blinking. The display border may be any one of 16 colours.

The display starting address in the display RAM is programmed via the CRTC. The starting address must be on an even address boundary and it addresses the first (leftmost) character position in the top row of the display. In each pair of display RAM bytes, the even address is for the character code and the odd address is for the attribute byte. Subsequent characters are displayed along the row from left to right. When the end of a row is reached the next pair in the display RAM appears in the first character position of the next row down.

The attribute byte for each is as follows:

Bit (ATn)	Definition
7	Intensity (Background) or Enable Blink (depends on Mode Control Bit 5)
6	Red (Background)
5	Green (Background)
4	Blue (Background)
3	Intensity (Foreground)
2	Red (Foreground)
1	Green (Foreground)
0	Blue (Foreground)

1.11.2 Graphics Display

There are two Graphics modes available, either Mode 1 (medium resolution) 320 pixels per scan line with a choice of four colours per pixel or Mode 2 (high resolution) 640 pixels per scan line with a choice of sixteen colours per pixel. The modes require 16000 and 64000 bytes of display RAM respectively.

The display RAM is divided into four 16K byte planes, one each for all the Red, Green, Blue and Intensity bits. Each plane may be individually written or read by the CPU, and two or more planes may be selected by the CPU for writing simultaneously with the same data.

1.11.2.1 Graphics Mode 1

The colour planes are not used in Mode 1, since only 16K bytes of display RAM (i.e. one plane) is ever required. For this reason system reset or entry into any other display mode other than graphics mode 2 forces selection of all planes for simultaneous writing by the CPU, which appears (to the software) as though only one 16K byte display RAM is installed in the address range from B8000 to BBFFF.

In Mode 1, the display memory for one scan line (320 pixels) consists of 80 bytes. Each pixel requires two bits such that four pixels are specified by each byte. The leftmost pixel is contained in the two MS bits of the byte and the two bit pairs for the remaining pixels follow on logically in left to right fashion. The two bit field for each pixel specifies one of four colours in one of three palettes (See 1.11.3 - VDU Control Registers). The three palettes are as follows:

B1	B0	Palette 0	Palette 1	Palette 2
0	0	Background	Background	Background
0	1	Green	Cyan	Cyan
1	0	Red	Magenta	Red
1	1	Yellow	White	White

In Mode 1 the 100 even scan lines (0, 2, 4, ... 198) are contained in the graphics memory space from B8000 to B9F3F and the 100 odd scan lines (1, 3, 5, ... 199) are contained in the memory address range from BA000 to BBF3F. The graphics memory appears as follows:

	<-- 320 Pixels (2 Bits Per) -->	
B8000	Scan Line 0 (80 Bytes)	B804F
B8050	Scan Line 2	B809F
B80A0	Scan Line 4	B80EF
.	.	
.	.	
B9EF0	Scan Line 198	B9F3F
BA000	Scan Line 1	BA04F
BA050	Scan Line 3	BA09F
BA0A0	Scan Line 5	BA0EF
.	.	
.	.	
BBEF0	Scan Line 198	BBF3F

The layout of a byte of graphics RAM in Mode 1 is as follows:

RAM Bit:	7:6	5:4	3:2	1:0
Pixel:	0	1	2	3
PIXEL Bit:	B1 : B0	B1 : B0	B1 : B0	B1 : B0

1.11.2.2 Graphics Mode 2

In Mode 2 the four memory planes become active and the colour of each pixel is controlled by the setting of the VDU Colour Plane Write Register at the time a pixel is written. The display memory for one scan line consists of 80 bytes, each containing 8 pixels. Since there are four planes, each displayed pixel is actually made up from 4 bits of information (R,G,B,I) specifying one of 16 colours. The border is programmable to any one of the 16 colours.

The ROS emulates two colour 640 by 200 graphics mode (software mode 6) by writing to all four colour planes (Intense White). Then the VDU Colour Plane Select register is used as the foreground palette selector. See section 2.3.7.

The layout of a byte of graphics RAM in Mode 2 is as follows:

RAM Bit:	7	6	5	4	3	2	1	0
Pixel:	0	1	2	3	4	5	6	7

The address mapping of the scan lines in display RAM for mode 2 is identical to that depicted for display Mode 1 - all even scan lines (from B8000 to B9F3F) followed by all odd scan lines (from BA000 to BBF3F). The major difference is that data written to one location may actually be stored in up to four planes simultaneously.

Note that even though the same physical address range is used between graphics and text modes, the internal data storage within VDU RAM changes between the different modes. This means that data patterns stored in successive locations in text modes will be found in the IRGB planes when graphics mode 2 is selected. Also data patterns stored in graphics mode 1 will be found on all planes when graphics mode 2 is selected. It is therefore not generally an acceptable practice to store information in one VDU mode and then switch to another mode else unexpected results may be encountered.

1.11.3 VDU Control Registers

There are five programmable registers for VDU mode and colour selection. These consist of the VDU Mode Control Register, the VDU Colour Select Register, the VDU Colour Plane Write Register, the VDU Colour Plane Read Register, and the VDU Graphics Mode 2 Border Register.

On power-up or following a system reset, the control registers are preset as described below.

1.11.3.1 VDU Control Registers

The VDU Mode Control Register is a write only register located at I/O address 3D8. It is used to control the state of the VDU circuitry, selecting Alpha or Graphics mode and the various sub options available within Alpha and Graphics modes.

The layout of the VDU Mode control register is as follows:

Bit	**Output Use**
7	No effect
6	No effect
5	Enable Blinking Chars (disable intensified backgrounds)
4	Select Graphics Mode 2 (de-select graphics mode 1)
3	Enable Video Display

Bit	Output Use
2	Select Palette 2 (de-select palettes 0,1)
1	Select Graphics modes (de-select Alpha modes)
0	Select Alpha 80 Char mode (de-select 40 Char mode)

When bit 5 is set in Alpha modes, the foreground of all displayed characters with attribute bit 7 (AT7) set will blink at 1.875 Hz (1/32 frame rate) in synchronism with frame flyback. Bit 5 has no effect in Graphics modes.

Bit 4 (Select Graphics Mode 2) has no effect in Alpha modes.

The Select Palette 2 bit (bit 2) has no effect in Alpha modes or in Graphics mode 2. It is used in conjunction with bit 5 of the VDU colour select register to control graphics mode 1 palette. To select palette 2, bit 5 of the VDU colour select register must be reset and bit 2 of the VDU mode control register must be set.

Bit 0 (Select Alpha 80 Char mode) has no effect in Graphics modes.

To avoid unsightly effects on the screen, this register should be updated during frame flyback time. Any kind of mode changing should preferably be done with video disabled. Mode changing involves the use of bits 1 and 0 and usually some re-programming of the CRTC.

On power-up and following a system reset, all bits of this register are cleared to zero.

1.11.3.2 VDU Colour Select Register

The VDU Colour Select Register is a write only register located at I/O address 3D9 and is used for controlling border colour in alpha modes and for selecting palette, border and pixel colour options in the graphics modes. The layout of the VDU Colour select register is as follows:

Bit	Alpha Modes	Graphics Mode 1	Graphics Mode 2
7,6	No Effect	No Effect	No effect
5	No Effect	Select Palette 1 (Deselect palette 0)	No effect
4	No Effect	Foreground Intensity for palettes 0, 1 & 2	No effect
3	Intensity (Border)	Intensity (Backgnd and Border)	Intensity (Pixel)
2	Red (Border)	Red (Background and Border)	Red (Pixel)
1	Green (Border)	Green (Background and Border)	Green (Pixel)
0	Blue (Border)	Blue (Background and Border)	Blue (Pixel)

In Graphics mode 2, the display border colour is programmed via the VDU Graphics Mode 2 Border register. The ROS makes use of the VDU Colour Select register in VDU I/O software mode 6 as a palette selector by writing all graphics in bright white. The overall screen palette is then controlled by setting the VDU Select register (which is initialised to 07h by the ROS on selection of mode 6). This mode effectively emulates 2 colour 640 by 200 graphics mode using 16k bytes of VDU memory. (See section 2.3.7 - VDU I/O.)

When using the full 16 colour capability in Graphics mode 2, programmers must set the VDU Colour select register to 0Fh and vary the plane selection with the Colour Plane Write register.

To avoid unsightly effects on the screen this register should only be updated during frame flyback time.

On power-up or following a system reset, all bits in this register are cleared.

1.11.3.3 VDU Colour Plane Write Register

The VDU Colour Plane Write Register is a write only register located at I/O address 3DD. It is used in Graphics Mode 2 for controlling which colour plane or combination of colour planes will be written to when addressing the display RAM (B8000 to BBFFF hex).

The bit assignments for the VDU Colour Plane Write register are as follows:

Bit	Output Use
7 - 4	No effect
3	Allow CPU write to Intensity Plane
2	Allow CPU write to Red Plane
1	Allow CPU write to Green Plane
0	Allow CPU write to Blue Plane

The CPU simultaneously writes the memory planes that are enabled for writing.

Writing to this register has no effect except in Graphics Mode 2.

Following the selection of Graphics Mode 2, (other than when already in Graphics Mode 2) bits 0 through 3 are set to one, but may subsequently changed by writing to this channel.

1.11.3.4 VDU Colour Plane Read Register

The VDU Colour Plane Read Register is a write only register located at I/O address 3DE. It is used in Graphics Mode 2 for controlling the selection of which one of the four colour planes in the display RAM is to be read by the CPU when memory in the 16K byte address range from B8000 to BBFFF (hex) is read. The bit assignments for the VDU Colour Plane Read register are as follows:

Bit	Output Use
7 - 2	No effect
1	Read Select bit 1 (RDSEL1)
0	Read Select bit 0 (RDSEL0)

The CPU reads from whichever one of the four colour planes in display RAM is selected according to the RDSEL value as follows:

RDSEL	Colour Plane Selected for CPU read
0	Blue Plane
1	Green Plane
2	Red Plane
3	Intensity Plane

Writing to this register has no effect except in Graphics Mode 2.

On power-up or following a system reset or selection of any other mode other than Graphics Mode 2 the register is cleared and fixed so that the CPU will only read from the Blue Plane.

1.11.3.5 VDU Graphics Mode 2 Border Register

The VDU Graphics Mode 2 Border Register is a write only register located at I/O address 3DF. It is used in Graphics Mode 2 for specifying the border colour.

The bit assignments for the VDU Graphics Mode 2 Border register are as follows:

Bit	Output Use
7 - 4	No effect
3	Border Intensity
2	Border Red
1	Border Green
0	Border Blue

Bits 0 through 3 specify one of sixteen colours.

To avoid unsightly effects on the screen this register should only be updated during frame flyback time.

On power-up or following a system reset, or following selection of Graphics Mode 2 (other than when already in Graphics Mode 2) the register is cleared (Graphics Mode 2 border black).

1.11.4 VDU Status Register

The VDU Status Register is a read only register located at I/O address 3DA. It may be read at any time to determine the following:

Bit	Input Use
7 - 4	Undefined
3	Frame Flyback Time
2	Light-pen switch off
1	Light-pen latch set
0	Toggle bit

Frame flyback time starts at the same time as the bottom border and lasts for 46 horizontal scan periods, ending 16 scans before the end of the subsequent top border.

Bit 2 reflects the state of the light-pen push button switch, which is neither latched nor debounced.

Bit 1 when set (1) indicates that the light pen latch is set, triggered either by a pulse from the light pen or by writing data to the light pen channel. Writing any data to the Clear Light Pen channel clears the latch, which is undefined following power-up and unaffected by system reset.

When the VDU Status Register is read, bit 0 toggles to the opposite state. Bit 0 is cleared (0) following power-up or following system reset.

1.11.5 MC6845 CRTC Emulation

The VDU controller is a partial emulation of a MC6845 CRT Controller device. Some of the registers which are programmable in an actual MC6845 are not programmable in the AMSTRAD PC1512. These differences are noted in the table which follows.

The remaining registers must be programmed according to the VDU mode of operation required in conjunction with the VDU Mode and Colour Select Registers previously described. A mode changing operation should be performed in the following sequence: Disable video, reprogram the CRTC as required, reprogram the Mode and Colour select registers as required, (maintaining video disabled), initialise display RAM as required, enable video.

The CRTC is controlled by way of two I/O addresses, the CRTC Address register and the CRTC Data I/O location. The CRTC Address register is a write only register located at I/O address 3D0 (and all even addresses to 3D6). The address register is a 5 bit register used to select one of eighteen internal control registers (R0 - R17). Addresses greater than 17 produce no results. Once the CRTC Address register has been loaded, the CRTC Data I/O location which is located at I/O 3D1 (and all odd addresses to 3D7) allows access to the selected internal CRTC register. Depending on the particular register selected the location may be either Read/Write (RW), Read Only (RO) or Write Only (WO).

The emulated MC6845 CRTC internal register layout and their initialised values are as follows:

Register Number	Register Name	Alpha 40 Char Mode	Alpha 80 Char Mode	Graphics Modes	R/W Type
R0	* Horizontal Total	--	--	--	-
R1	Horizontal Displayed	[40]	[80]	[40]	WO
R2	* Horizontal Sync Posn.	--	--	--	-
R3	* Horizontal Sync Width	--	--	--	-
R4	* Vertical Total	--	--	--	-
R5	* Vertical Total Adj.	--	--	--	-
R6	Vertical Displayed	[25]	[25]	[100]	WO
R7	* Vertical Sync Posn.	--	--	--	-
R8	* Interlace	--	--	--	-
R9	Max. Raster Address	07	07	01	WO
R10	Cursor Start Raster	06	06	06	WO
R11	Cursor End Raster	07	07	07	WO
R12	Start Address (MS)	00	00	00	WO
R13	Start Address (LS)	00	00	00	WO
R14	Cursor Location (MS)	00	00	00	RW
R15	Cursor Location (LS)	00	00	00	RW
R16	Light Pen Posn. (MS)	--	--	--	RO
R17	Light Pen Posn. (LS)	--	--	--	RO

Note that all values above are in decimal format.

Registers marked '*' are not software programmable in the emulated MC6845 CRTC implementation, unlike a real MC6845. They are fixed in hardware according to the VDU mode currently selected.

The two sets of registers marked by '[ ]' should normally be programmed nonzero. The magnitude of the non-zero value will have no effect on the display, but a zero value in either register will cause the whole screen to display the border colour.

The cursor function of a 6845 CRTC is supported only in Alpha modes. The cursor blinking function is performed by a circuit external to the CRTC and blinks the cursor at 3.75Hz (1/16 frame rate) in synchronism with frame flyback. The cursor non-display feature, variable blink rate feature and variable blink period feature of the 6845 CRTC are not supported.

The valid range of values for Cursor Start Raster (R10) is from 0 to 30 (decimal) and the value of 31 (which will turn the cursor off). Values greater than 31 are not recommended.

The light pen function of a 6845 CRTC is supported:

	Horizontal Resolution	Vertical Resolution
Alpha Modes:	1 Char column	1 Char row
Graphics Modes:	1 Pixel Word	2 Scans (ie. 1 row)

If "M" = Current Display Address then in...

Graphics, LPEN value = M + 1 or M + 2.
Alpha 80, LPEN value = M + 3 or M + 4.
Alpha 40, LPEN value = M + 1 or M + 2.

1.11.6 CRTC Display Addressing

In Alpha modes, the CRTC register values for start address and light pen position are in the 8K range 0000h to 1FFFFh, and wraparound occurs above this range. The register value corresponding to a character position in display RAM must be derived from the even byte address in the 16K range B8000h to BBFFFh by subtracting B8000h and halving.

In Graphics modes, the CRTC register values for start address and light pen position are in the 4K range 000h to FFFh, and wraparound occurs above this range. A register value corresponds to two pairs of pixel bytes in display RAM on word boundaries, one pair displayed on an even scan and the other pair displayed on the following odd scan in the same horizontal position.

The register corresponding to the pixel byte pair position in display RAM must be derived from the even byte address in the 8K range B8000h to B9FFFh (for an even scan) by subtracting the address offset B8000h and halving. Similarly for the odd scan line the offset BA000h is subtracted from an even byte address in the range BA000h to BBFFFh and halved.

__1.11.6.1 Mode Mapping Relationships__

There are important differences in the way the VDU Controller maps and uses the display RAM for different VDU display modes.

The Alpha Mode to Graphics Mode memory mapping follows the scheme depicted in fig 1.3.

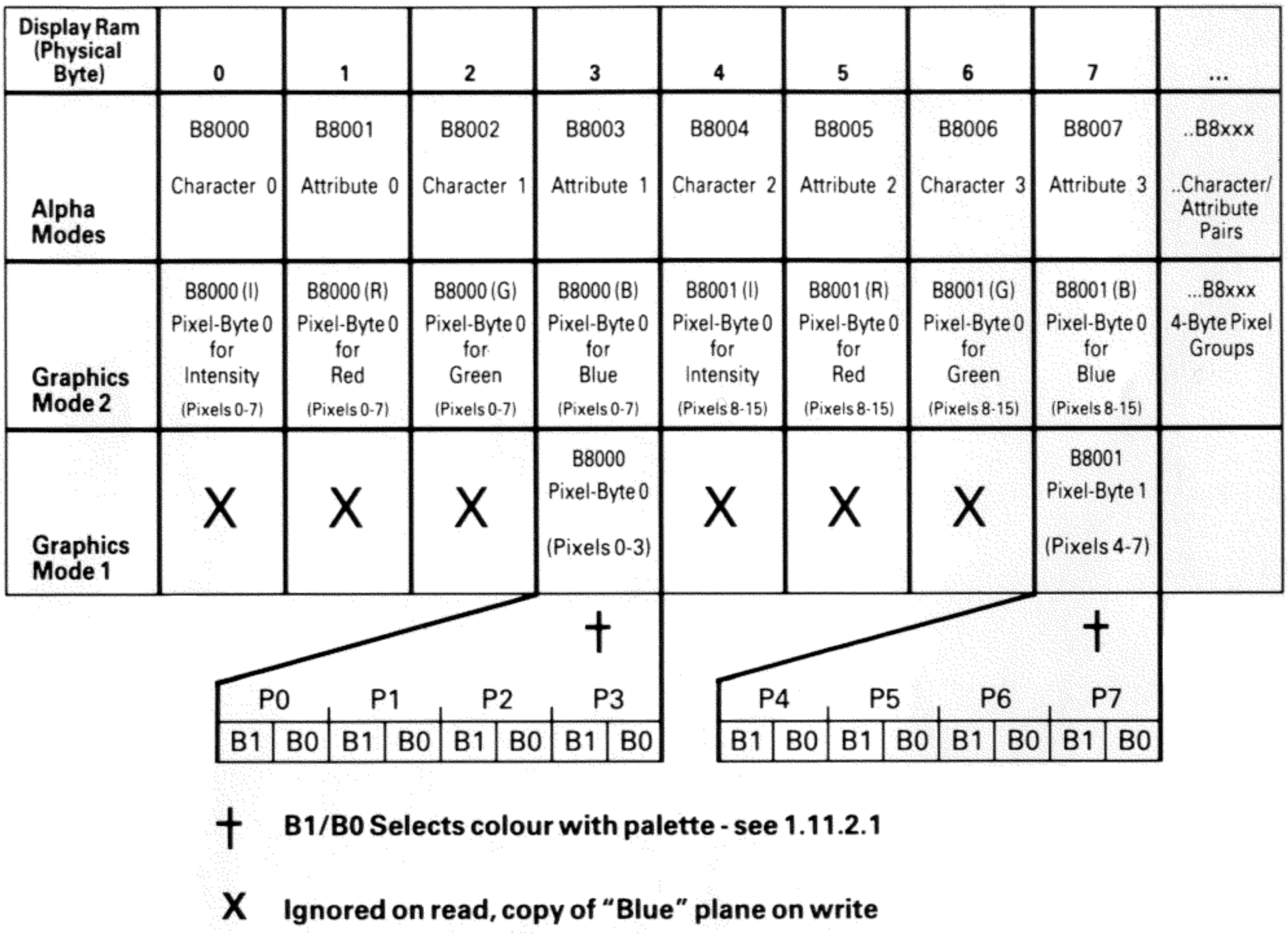
The concept of planes does not exist in character modes and character data is stored sequentially in display RAM. In Graphics modes, the display RAM is viewed logically as colour planes which are grouped in four byte groups. This means that there is an effective "gear change" between alpha and graphics modes and storing in an address such as B8000h in graphics modes spans the address range from B8000 to B8003 in alpha modes.

In Graphics Mode 1, the display RAM mapping is the same as for Graphics Mode 2, but only the data contained on the Blue plane is used to form pixels on the screen. When the CPU writes to display RAM in Graphics Mode 1, the same data is stored in all planes.

It is very important to note that it is impossible to store character and attribute pairs in consecutive display RAM in graphics mode. Attempting to clear the text screen with text data while in graphics mode and then switching to an alpha display mode will produce a "Stars and Bars" effect on the screen.

Display RAM Mapping

Fig 1.3



Display Ram (Physical Byte)	0	1	2	3	4	5	6	7	...
Alpha Modes	B8000 Character 0	B8001 Attribute 0	B8002 Character 1	B8003 Attribute 1	B8004 Character 2	B8005 Attribute 2	B8006 Character 3	B8007 Attribute 3	..B8xxx ..Character/ Attribute Pairs
Graphics Mode 2	B8000 (I) Pixel-Byte 0 for Intensity (Pixels 0-7)	B8000 (R) Pixel-Byte 0 for Red (Pixels 0-7)	B8000 (G) Pixel-Byte 0 for Green (Pixels 0-7)	B8000 (B) Pixel-Byte 0 for Blue (Pixels 0-7)	B8001 (I) Pixel-Byte 0 for Intensity (Pixels 8-15)	B8001 (R) Pixel-Byte 0 for Red (Pixels 8-15)	B8001 (G) Pixel-Byte 0 for Green (Pixels 8-15)	B8001 (B) Pixel-Byte 0 for Blue (Pixels 8-15)	...B8xxx 4-Byte Pixel Groups
Graphics Mode 1	X	X	X	B8000 Pixel-Byte 0 (Pixels 0-3)	X	X	X	B8001 Pixel-Byte 1 (Pixels 4-7)	

† **B1/B0 Selects colour with palette - see 1.11.2.1**

X **Ignored on read, copy of "Blue" plane on write**

__1.11.6.2 Display RAM Access Overhead__

The VDU display timing and system CPU/DMA timing are derived from different, unrelated reference frequencies. For this reason CPU accesses to the display RAM must be synchronised to the display timing by the VDU controller, and this is done by inserting CPU wait states as appropriate.

Whenever the CPU accesses the display RAM, the total number of 125nS wait states incurred, including those already added automatically by the 8bit bus conversion process, will range from a minimum of 12 to a maximum of 46. At most, a situation in which 46 wait states are necessary could occur once every 63.7uS, in all modes. Similarly, the need for 38 wait states could occur at most every 63.7uS in Graphics mode 2 or in 80 x 25 Alpha mode.

1.11.6.3 Alpha Mode Character Generator ROM

The 8K character generator ROM can contain a maximum of four different character sets, each in a different 2K byte quadrant of the ROM. Only one character set can be used at a time and it is selected by means of two option links, LK6 and LK7, located on the right hand side main board.

The character set selection is as follows:

LK7	LK6	Character Set Selected
Out	Out	ROM Fourth Quadrant - Default factory setting.
In	Out	ROM Third Quadrant - (optional Danish).
Out	In	ROM Second Quadrant - (optional Danish).
In	In	ROM First Quadrant - (optional Greek - late models).

Floppy Disk Controller

The floppy disk controller is based on the NEC uPD765A single chip controller, and supports one or two 5.25 inch single or double sided, MFM double density floppy disk drives with a data rate of 250 kilobits per second.

The FDC is controlled by way of the Drive Selection register (at I/O Address 3F2) and it is defined as follows:

Bit (Dn)	Output Use
7 - 6	No effect
5	Switch motor(s) on and enable drive 1 selection
4	Switch motor(s) on and enable drive 0 selection
3	Allow 765A FDC to interrupt and request DMA
2	$\overline{\text{765A reset}}$
1	Drive Select Bit 1 (DS1)
0	Drive Select Bit 0 (DS0)

The Drive Select bits (DS1, DS0) are only valid for values of 00 and 01 for drives 0 and 1 respectively. The drive selection qualification is only completed when either bit 4 (for drive 0) or bit 5 (for drive 1) is set. In addition setting bits 4 or 5 will have no effect until the value of DS1, DS0 is correspondingly set.

Bit 2 when cleared (0) holds the 765A reset until bit 2 is again set (1). Note that a reset must last for at least 3.5 uS.

On power-up or following a system reset, all bits in this register are cleared to zero.

1.12.1 FDC Hardware Conditions

The hardware imposes the following conditions on the use of the 765A controller and disk drives:

1. The clock frequency of the 765A FDC is fixed at 4.0 MHz.
2. Disk data transfers are done by DMA using the on-board DMA controller. The 765A DRQ output may connected to or disconnected from the DMA controller DRQ2 input by software using Drive Selection Register bit 3.
3. An interrupt level is available for use by the 765A to signal command completion and attention status to the CPU. The 765A INT output may be connected to or disconnected from the interrupt controller IRQ6 input by software using Drive Selection Register bit 3.
4. Drive 0 is always present. Drive 1 is optional. Drives 2 and 3 are not implemented and can never be accessed. Drive Ready output signal from the currently selected drive is connected to the 765A RDY input. For drives which do not a drive ready output the 765A RDY input may be optionally fixed to the true condition.
5. The 765A Drive Select outputs US1 and US0 are not used to select the drives. This function together with motor control is done via the Drive Selection Register which is external to the FDC 765A.
6. The FLT (Fault) input 765A is forced permanently false.
7. A Two-Sided status signal from the drive(s) is not provided but interface to the drives allows the use of double sided drives.
8. Write precompensation of 250 nS is provided.
9. The 765A may be individually reset by software using Drive Selection Register bit 2.

1.13 RS232C Asynchronous Serial Port

The asynchronous serial port is based on the National INS8250 ACE (or UART), single channel device.

The clock frequency input of the 8250 is 1.8432 MHz (± 0.1%).

The 8250 BAUD OUT output is connected to the RCLK input.

An interrupt level is available for use by the 8250. When the 8250 OUT2 output is driven low (i.e. a 1 is written to bit 2 of the 8250 MODEM Control Register) then the INTRPT signal is connected to the interrupt control IRQ4 input.

1.13.1 Serial Channel Interface

The serial interface uses a 25-way subminiature D type plug (male) connector emulating a DTE (Data Terminal Equipment).

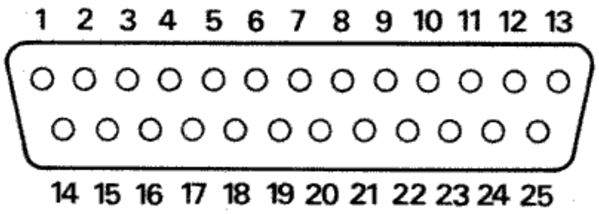
The electrical levels of signal lines on this interface conform with EIA (Electronics Industry Association) standard RS-232C (and the equivalent V.24 interface standard).

The RS232C drivers and receivers between the 8250 and the serial channel connector are all inverting.

1.13.2 Serial Channel Pin Arrangement

Pin	EIA	CCITT	Description
1	AA	101	Frame Ground
2	BA	103	TxD - Serial Data Output
3	BB	104	RxD - Serial Data Input
*4	CA	105	RTS - Request to Send Output
5	CB	106	CTS - Clear to Send Output
6	CC	107	DSR - Data Set Ready Input
7	AB	102	Signal Ground (Common Return)
8	CF	109	DCD - Data Carrier Detect Input
*20	CD	108.2	DTR - Data Terminal Ready Output
22	DE	125	RI - Ring Indicator Input

* These interchange circuits, where implemented, shall be used to detect either a power off condition in the equipment across the interface, or the disconnection of the interconnecting cable. The terminator for these circuits shall interpret the power off condition or the disconnection of the interconnecting cable as an OFF condition.



Viewed from rear of machine

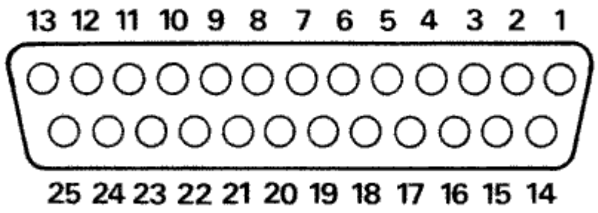
See Appendix 3 for additional details of serial signals and cable connections.

1.14 Parallel Printer Interface

The parallel printer port is described in Section 1.10 and is a general purpose 'centronics' style 8-bit interface. The printer interface uses a 25-way subminiature 'D' socket (female) connector located at the back of the PC1512.

The Pin assignments for the printer connector is as follows:

Pin	Assignment
1	Data Strobe
2	Data Bit 0
3	Data Bit 1
4	Data Bit 2
5	Data Bit 3
6	Data Bit 4
7	Data Bit 5
8	Data Bit 6
9	Data Bit 7
10	$\overline{\text{Printer Acknowledge}}$
11	Printer Busy
12	Paper Out
13	Select Printer
14	$\overline{\text{Select Auto Feed}}$
15	$\overline{\text{Printer Error}}$
16	$\overline{\text{Reset Printer}}$
17	$\overline{\text{Printer Selected}}$
18	GND
19	GND
20	GND
21	GND



Viewed from rear of machine

Pin	Assignment
22	GND
23	GND
24	GND
25	GND

Appendix 4 contains the Amstrad PL-2 printer lead specification for the DMP3000 printer.

1.15 Keyboard Interface

Keyboard data input to the CPU is via the 8255 PPI Port A, and the keyboard interrupt (level 1) of the 8259A PIC. Both of these have been previously described in sections 1.6 and 1.8.

1.15.1 Serial Clock and Serial Data

The Serial Clock and Serial Data signals are used for keyboard interface. These two bidirectional signals are used by the keyboard microcontroller to send keycodes to the main electronics board. The main electronics board also uses the same two signals to indicate readiness to receive another keycode back to the microcontroller. In addition these two signals are used to reset the microcontroller under hardware or software control.

1.15.2 Keyboard to Main Board Interface

The quiescent state for both Serial Clock and Serial Data is high. A minimum of 5 uS must separate a transition of one signal from another transition of the same signal, or any transition of the other signal.

Keycodes are sent from the keyboard microcontroller to the main board in 8-bit serial form MS bit first. Keycode data received by the main board is clocked into a shift register as either a "1" bit sequence or as a "0" bit sequence. To be interpreted as a "1" bit, the Serial Data signal must remain high during the time period when Serial Clock goes low and returns to the high state. To be interpreted as a "0" bit, Serial Data must be low prior to the Serial Clock transition from high to low, Serial Data will then go high followed by Serial Clock. The "1" bit or the "0" bit is clocked into the shift register on the falling edge of Serial Clock.

1.15.3 Main Board to Keyboard Interface

Upon receiving a keystroke from the microcontroller, within 5 uS of the last clock falling edge, the main board electronics drives the Serial Data line low and maintains it low until it is ready to receive a new keystroke. When the main board returns the Serial Data signal to the high state the microcontroller is free to send another keystroke. This response to the reception of a keycode is termed the ACKNOWLEDGE sequence.

The mainboard electronics causes a RESET to the keyboard microcontroller by driving the Serial Clock line low for 10 milliseconds or more. The state of the Serial Data signal does not affect the reset sequence.

1.15.4 Keycodes

The 8-bit keyboard data is capable of 128 make codes correspondingly 128 break codes. For any key which is pressed, the (make) keycode produced is in the range of 0 - 127 decimal. When a key is released, the (break) keycode produced is the same as the make keycode except that the top bit is set so that the value is in the range of 128 - 511 (decimal). The keycodes and their corresponding token values are covered in the ROS firmware (Section 2.3.5).

After a key is pressed and the keycode has been sent to the main board electronics, if no new keys are pressed and the key has remained pressed for more than one second, then the keyboard microcontroller re-sends the keycode every 83 milliseconds provided that the main board indicates by an Acknowledge sequence that is ready to accept a new keycode.

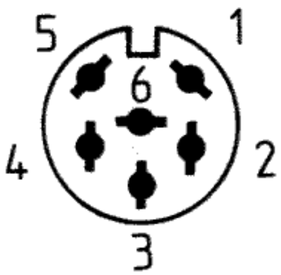
The keycode AA hexadecimal is sent after a reset to indicate successful completion of power-up tests.

1.15.5 Keyboard Connector

The Keyboard connector is a 6-way Din socket. The pin assignment is as follows:

Pin	Assignment
1	KBCLK
2	KBDATA
3	M1
4	GND
5	+5 Volts DC

Viewed from left hand side of machine

Pin	Assignment	
6	M2	

The M1 and M2 signals are connected directly to the keyboard controller in order to produce keycodes.

1.16 Mouse Interface

The mouse interface consists of two switch inputs from push buttons and two 8-bit X & Y coordinate counters. The two mouse switches (M1 & M2) are arranged to form part of the keyboard matrix and are handled as keyboard data (producing low level keycodes 7E and 7D respectively).

The Mouse X-Coordinate at I/O Address 078 is an 8-bit counter which can be read by the CPU. Any write access regardless of the value written to the X-Coordinate location clears the counter. Similarly the Mouse Y-Coordinate at I/O Address 07A can be read by the CPU or cleared by any write access to its I/O address.

The counters are incremented or decremented according to the direction of movement of the mouse, and their values indicate the mouse movement since last read or cleared. The X-Coordinate counter increments for "Right" motion and decrements for "Left" motion. The Y-Coordinate counter increments for "UP" motion and decrements for "Down" motion. In order to properly track mouse motion, software should read and clear the coordinate counters at a rate high enough to prevent overflow from positive values to negative values or negfative values to positive values for a fairly high rate of mouse movement. The scaling of mouse movement is such that one increment of the counter represents 1/8 mm of physical mouse motion.

The delivered operating systems have AMSTRAD specific mouse drivers which actively perform the Read-and-Clear operation (every 18 ms) using the ticker interrupt. This can cause the appearance of no mouse motion to the casual observer sampling the mouse coordinate counters. See Appendix 1 for additional details concerning Mouse Software Interfaces.

1.16.1 Mouse Connector

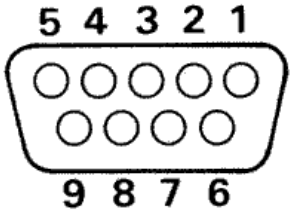
The mouse connector is a 9 way D type (female) connector located on the left hand side of the case and it has an AMSTRAD specific pinout. Attaching any other manufacturer's hardware (even though the connector may be similar) to the PC1512 mouse connector may cause serious damage to either the main board electronics or to the alternative (mouse) hardware.

The mouse connector pin assignments are as follows:

Pin	Assignment
1	XA
2	XB
3	YA
4	YB
5	Spare
6	M1
7	+5 Volts DC
8	GROUND
9	M2

The first four pins contain optically encoded phase XA, XB, YA and YB square waves. For positive motion the square wave on the A phase leads the B phase by 90 degrees with the reverse being true for negative motion.

The remaining pins carry Mouse Button 1 (M1), 5V power, Ground and Mouse Button 2 (M2) signals.



Viewed from left hand side of machine

1.17 Joystick Interface

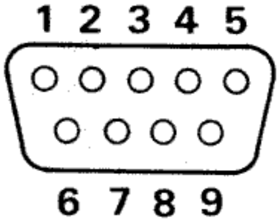
The AMSTRAD PC supports an industry standard joystick interface. The joystick inputs are handled as keycodes from the keyboard interface. The low level keycodes are in the range of 7C down to 77 (hexadecimal) corresponding to Up, Down, Left, Right, Fire1 and Fire2 respectively. The ROS firmware (See section 2) translates the directional codes to cursor key tokens and the Fire buttons can be assigned variable tokens depending on the NVR settings. Please note that this is not an analog interface. In order to use an analog joystick an analog card for the expansion slots is required.

1.17.1 Joystick Connector

The joystick connector is a 9 way D type (male) connector with an industry standard pinout. Attaching an incorrect device (even though the connector may be similar) to the PC1512 joystick connector may cause serious damage to either the main board electronics or to the incorrect (joystick) hardware.

The Joystick Socket is located on the rear left corner of the keyboard. Its pinout is as follows:

Pin	Assignment
1	Up
2	Down
3	Left
4	Right
5	Spare
6	Fire 2
7	Fire 1
8	Common
9	Not Connected



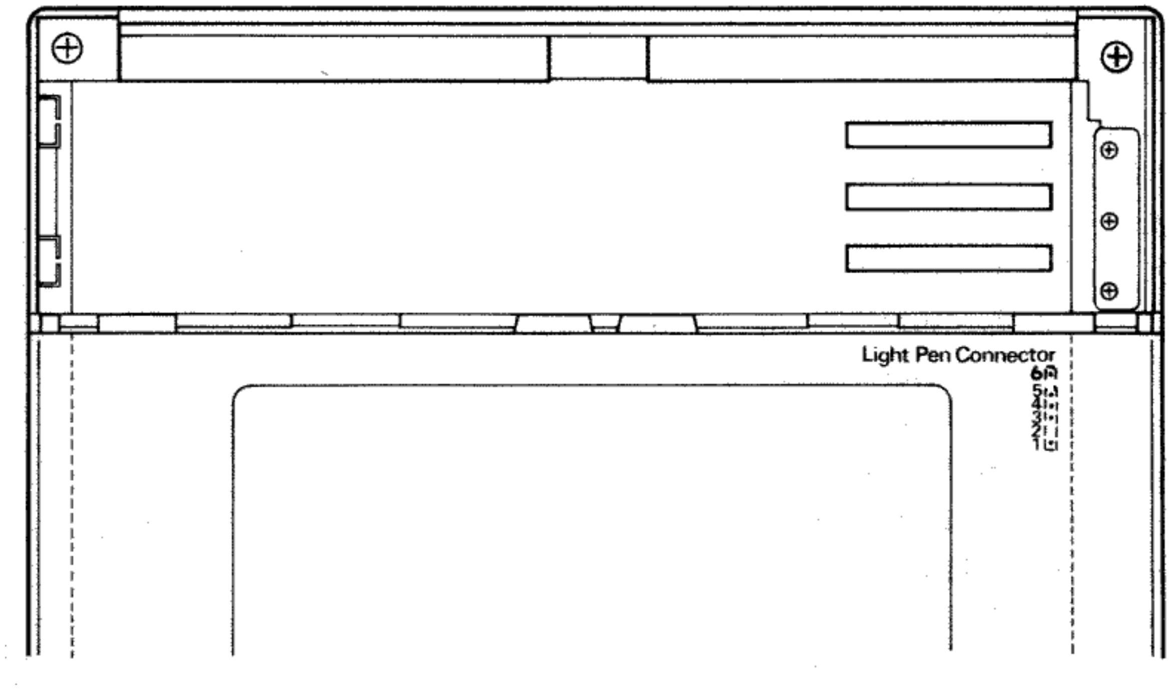
Viewed from rear of keyboard

1.18 Light Pen Connector

The AMSTRAD PC1512 Supports a standard light pen interface via the emulated MC6845. The Light Pen connector is located by removing the expansion slot cover at the rear of the machine. The connector is located inside the PC case on the right hand edge of the main board just forward of the expansion card connectors. It consists of a 6-way berg strip and is labeled PL8 (LIGHT PEN) in large letters. Pin 1 is the forward most pin viewed from in front of the machine (the disk drive end).

The pin assignment is as follows:

Pin	Assignment
1	Light Pen Input.
2	(Keyway)
3	Light Pen Switch.
4	Ground.
5	+5 Volts DC.
6	+12 Volts DC.

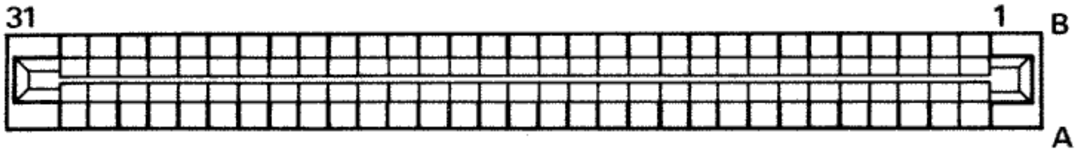


1.19 Expansion Card Interface

The AMSTRAD PC1512 has three slots for additional peripheral cards. These consist to a set of connectors in the right rear of the main board. The Pin numbering of the each connector is the same and is such that the left (ground plane) side is numbered B1 - B31 top to bottom and the right (component) side is numbered A1 - A31 top to bottom. The following table defines the pin assignments of the expansion interface:

Pin	Signal	In/Out
A01	Not I/O CHCK	In
A02	I/O Data Bit D7	In/Out
A03	I/O Data Bit D6	In/Out
A04	I/O Data Bit D5	In/Out
A05	I/O Data Bit D4	In/Out
A06	I/O Data Bit D3	In/Out
A07	I/O Data Bit D2	In/Out
A08	I/O Data Bit D1	In/Out
A09	I/O Data Bit D0	In/Out
A10	I/O RDY	In
A11	AEN - Address Enable	Out
A12	I/O + Mem/Address Bit A19	Out
A13	I/O + Mem/Address Bit A18	Out
A14	I/O + Mem/Address Bit A17	Out
A15	I/O + Mem/Address Bit A16	Out
A16	I/O + Mem/Address Bit A15	Out
A17	I/O + Mem/Address Bit A14	Out
A18	I/O + Mem/Address Bit A13	Out
A19	I/O + Mem/Address Bit A12	Out
A20	I/O + Mem/Address Bit A11	Out
A21	I/O + Mem/Address Bit A10	Out
A22	I/O + Mem/Address Bit A09	Out
A23	I/O + Mem/Address Bit A08	Out
A24	I/O + Mem/Address Bit A07	Out
A25	I/O + Mem/Address Bit A06	Out
A26	I/O + Mem/Address Bit A05	Out
A27	I/O + Mem/Address Bit A04	Out
A28	I/O + Mem/Address Bit A03	Out
A29	I/O + Mem/Address Bit A02	Out
A30	I/O + Mem/Address Bit A01	Out
A31	I/O + Mem/Address Bit A00	Out
B01	Ground	--
B02	RESET	Out
B03	+ 5 Volts DC	--

Pin	Signal	In/Out
B04	IRQ2	In
B05	- 5 Volts DC	--
B06	DREQ2	In
B07	- 12 Volts DC	--
B08	Not Connected (Reserved)	In
B09	+ 12 Volts DC	--
B10	Ground	--
B11	MEW (Memory Write)	Out
B12	MRD (Memory Read)	Out
B13	IOW (I/O Write)	Out
B14	IOR (I/O Read)	Out
B15	DACK3	Out
B16	DREQ3	In
B17	DACK1	Out
B18	DREQ1	In
B19	DACK0	Out
B20	CLK	Out
B21	IRQ7	In
B22	IRQ6	In
B23	IRQ5	In
B24	IRQ4	In
B25	IRQ3	In
B26	DACK2	Out
B27	T/C	Out
B28	ALE	Out
B29	+ 5 Volts DC	--
B30	CK14	Out
B31	Ground	--



Expansion Bus Connector
Viewed from above while standing in front of machine

The I/O expansion slots are laid out the same as the industry standard 16-bit Personal Computer bus. The translation from the internal 16-bit 8086 bus to the 8-bit I/O bus layout is done by main board circuitry. Any 16-bit CPU I/O transfers will be broken down into two 8-bit cycles (with wait states) by this circuitry.

All signals are TTL compatible and can support a maximum of two low-power schottky (LSTTL) loads per slot. Power supply loading per slot should be limited to a maximum of 900 milliamperes on the + 5 Volt supply, to 33 milliamperes on the - 5 Volt supply, to 50 milliamperes on the + 12 Volt supply and to 50 milliamperes on the - 12 Volt supply. See Appendix 6.

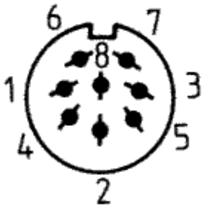
Note that direct access to the on-board 16-bit fast memory bus is not available via the I/O expansion slots.

Additional engineering details for prototyping adapter boards should be supplied as part of the documentation for that particular hardware.

1.20 Video Connector

The video connector is a 8-way Din socket located in the rear of the computer. Its pinout is as follows:

Pin	Assignment
1	Composite H & V sync
2	Intensity
3	GND
4	Black
5	Green
6	Blue
7	GND
8	Red



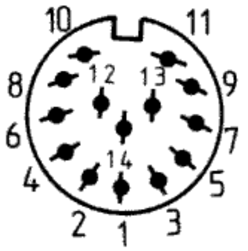
Video Plug (from monitor)

1.21 Power Connector

The power connector is a 14-way Din socket located in the rear of the computer. Power is routed from the power supply located in the monitor to the main board electronics through the power connector. Its pinout is as follows:

Pin	Assignment	MAX CURRENT
1	Not Connected	
2	0 Volts DC	
3	+ 5 Volts DC	- (SEE NOTE 1)
4	0 Volts DC	
5	+ 5 Volts DC	- (SEE NOTE 1)
6	Not Connected	
7	Not Connected	
8	0 Volts DC	
9	- 12 Volts DC	0.24 AMP
10	0 Volts DC	
11	+ 12 Volts DC	4.9 AMP
12	0 Volts DC	
13	- 5 Volts DC	1 AMP
14	Not Connected	

Power Plug (from monitor)



Note 1: Pins 3 & 5 are connected together to the +5 V bus for a total of 7.00 Amp maximum rating.

Note 2: See Appendix 6 for Power consumption data.

Contents Index Section 2

SECTION 2 - FIRMWARE

This section describes the AMSTRAD PC1512 Resident Operating System (ROS). It defines the interfaces to all the interrupt service routines provided by the AMSTRAD PC1512 ROS firmware (ROM) and all RAM locations used by the ROS.

The following copyright message is stored at the beginning of the ROS starting at location 0003 (relative to the origin):

(C) Copyright 1986 Amstrad Consumer Electronics plc

The ROS physically occupies the highest 16K bytes (in the address range FC000 to FFFFF) in the 1 Mega Byte addressing range of the 8086-2 CPU (See Figure 1.1). The total 64K byte address range from hexadecimal F0000 to FFFFF is reserved for system ROM and this contains the reset and initialisation address FFFF0. For this reason the system ROM always spans the highest locations and extends downward. Also address wraparound (block repeat) occurs in the 64K byte range. Note that all address constants in this document are in hexadecimal form unless otherwise noted.

All calls to the ROS firmware should be made through the software interrupts disclosed in this manual. Application programs should not attempt to access the locations within the ROM area directly. AMSTRAD reserves the right to modify the coding within the Resident Operating System ROM as it sees fit.

The ROS provides a set of service routines which perfrom various I/O functions:

1. Power-Up initialisation and Self Test.
2. Keyboard input.
3. VDU display of characters and pixels.
4. Character I/O to the printer and serial ports.
5. System clock and real time clock support.
6. Disk I/O including format, read and write.

To ensure hardware independence of application programs all I/O processes should be done using the ROS. This avoids possible problems due to any hardware modifications and/or enhancements.

Note that all ROS messages will be displayed in the language selected by the language links (see section 3.1).

2.1 Power-Up initialisation and Self Test

The Power-Up initialisation and Self Test function is entered at location FFFF0, the CPU reset entry point. This routine performs all necessary hardware initialisation and self tests, sets up the BIOS RAM variable area, initialises all the interrupt locations used by the ROS, initialises external ROMs then loads and runs the disk bootstrap.

The ROS does not use the SYSTEM RAM (User RAM Area) for stack or program variables until it has been successfully tested. If a RAM error is found an error message should be displayed correctly, assuming there is no other fault that may result in incorrect operation of the CPU or VDU.

The Power-Up initialisation and Self Test process is as follows:

1. Disable maskable and non-maskable interrupts.
2. Run the ROS self test which tests the following:
    - Checksum of the ROS.
    - All system RAM fitted.
    - 64k of VDU RAM.
    - VDU controller status bits.
    - 8237 DMA controller.
    - 8253 Programmable Interval Timer.
    - 8255 Programmable Peripheral Interface.
    - RTC counting.
    - The system serial interface.
    - The system printer interface.
    - 8259 programmable interrupt controller.
    - Mouse X and Y counter registers.

    After a system reset all the self tests except the RAM tests are rerun. If the three option links in the least significant part of the system printer status register are set to all ones then only the keyboard and disk tests are run.

    Refer to section 2.3 for the individual power-up self test details.

3. Checksum the NVR.

    If the checksum of the NVR is incorrect then it is loaded with its default values (see section 2.5).

4. Initialise the 8253 Programmable Interval Timer.

    Set up counter 0 to interrupt every 54.9337 milliseconds. Set up counter 1 to generate an output signal with a period of 15.13 microseconds. Disable counter 2.

5. Initialise the 8237 DMA controller.

    Set up DMA channel 0 for memory refresh. Disable channel 1, 2 and 3.

6. Initialise the 8259 Programmable Interrupt Controller.

    Disable (mask) all interrupt levels. Note that levels 0, 1 and 6 are enabled (unmasked) later.

7. Initialise the Write Status Registers.

    Write Status-1 is initialised from a byte in the NVR defining the number of drives fitted and the default VDU mode. The ROS also sets or resets bit 1 in the status register depending on whether or not an 8087 NDP is installed. See section 1.8 for further System Status-1 information.

    Write Status-2 is initialised according to the amount of memory installed. The ROS assumes a minimum of 512K bytes and that additional RAM may be added in contiguous 32k byte increments up to the maximum of 640K bytes. The additional memory is sized according to the following procedure. The segment address of each of the four 32K byte RAM blocks is written to the first two bytes of each respective block. The segments are then verified from low to high until a non matching segment address or the last block is encountered. The setting of the Write Status-2 register is according to the RAM0-RAM4 table in section 1.8.3.

8. Initialise the ROS variable area in system RAM.

The ROS uses variables in the address range of 00300 to 00500. Refer to section 2.4 (RAM Variables) for a complete description of these variables and their respective initialised values.

9. Initialise the first 32 Interrupt Vectors.

   The first 32 interrupt vectors are set up to reference the ROS routines as listed below. Software interrupt routines which do not perform any function reference a dummy routine that simply does a return from interrupt (IRET) instruction. Hardware service interrupts which do not perform any function reference a dummy (HWIRET) routine which issues a nonspecific end-of-interrupt to the 8259 interrupt controller and then executes an IRET instruction.

Interrupt	Purpose	Type
0	Divide by Zero	Hardware (HWIRET)
1	Single Step	Hardware (HWIRET)
2	Parity error routine (NMI)	Hardware
3	Break	Hardware (HWIRET)
4	Overflow	Hardware (HWIRET)
5	Print Screen	Software
6	Mouse button control	Software
7	Reserved	Software
8	System Clock interrupt	Hardware
9	Keyboard interrupt	Hardware
10	RTC interrupt	Hardware (HWIRET)
11	COMMS	Hardware (HWIRET)
12	COMMS	Hardware (HWIRET)
13	Hard Disk	Hardware (HWIRET)
14	Floppy Disk interrupt routine	Hardware
15	Printer interrupt	Hardware (HWIRET)
16	VDU I/O	Software
17	System Configuration	Software
18	Memory Size	Software
19	Disk I/O	Software
20	Serial I/O	Software
21	Enhanced Function	Software
22	Keyboard I/O	Software
23	Printer I/O	Software
24	System Restart	Software
25	Disk Bootstrap	Software
26	System Clock and RTC I/O	Software
27	Keyboard Break	Software (IRET)
28	External Ticker interrupt	Software (IRET)
29	VDU initialisation parameter table	Software
30	Disk Parameter table	Software
31	External VDU matrix table	Software

The interfaces to the above routines are detailed in section 2.3.

10. Initialise and Test the Disk interface.

    The initialise function of interrupt 19 in invoked followed by the disk test (see 2.2.14).

11. Keyboard Self Test.

    The Keyboard microcontroller returns 0AAh upon successful completion of its power-up self test (See 2.2.15).

12. Initialise the VDU.

    The initial VDI setting is done according to the Port-A value as described in section 1.8.2

13. Initialise the 8259 Interrupt controller.

    Enable 8259 interrupt controller on levels 0 (8253 counter 0), 1 (keyboard scan code receiver) and 6 (765 floppy disk controller). All other 8259 interrupt levels are masked.

14. Display the ROS sign-on message.

    During power-up the ROS checksums the NVR. After the sign-on message has been displayed, the ROS outputs a warning message if the NVR sum was incorrect. In the case that the NVR is OK (and last startup data are valid) the time and date of last switch-on are displayed.

15. Enable the NMI.

    If a NMI occurs the default ROS interrupt handler displays a RAM parity error message and hangs the system. This condition can only be rectified by switching the machine off.

16. Initialise all external ROMs.

    The ROS checks for external ROMs between addresses C0000 and F4000 in 800h (2k) byte increments. An external ROM which conforms to the following specification will be initialised by the ROS:

    1. The first two bytes contain the hexadecimal value 55AA.
    2. The next two bytes contain the size in 512 (1/2K) byte increments.
    3. The next byte is the initialisation routine entry point.
    4. The LS byte of the byte sum of the ROM is zero.

    When a ROM conforming to this specification is located then the initialisation entry is called. If the sum test fails then an error message is displayed and initialisation is not called.

17. Floppy Disk Bootstrap.

    The ROS attempts to load the bootstrap sector (from drive A, side 0, track 0, sector 1) into memory at 07C00h. If the bootstrap sector loads successfully it is given control (far jump to segment 0000 offset 7C00). If after 10 retrys the bootstrap sector cannot be loaded then the ROS displays a message prompting the user to insert a system disk into drive A and press a key. The ROS then waits for the key press and repeats the bootstrap procedure.

2.2 Power-UP Self Tests

On Power-Up or following a system reset, the ROS performs a series of self tests on the hardware to verify proper operation. When a test failure occurs, the ROS displays an error message on the VDU and the system is locked up. The keyboard interface is treated differently in that the ROS repeats keyboard self test until it is successful.

The ROS executes all self tests except when the option links (LK1 - LK3) are all set (See section 1.10.3), the ROS will only run the keyboard interface test and the disk test. If either of these two tests fail an error message is displayed but the error is ignored. This allows the system to be brought up for diagnostic testing.

When a soft reset (Control, Alt and Del) is issued the ROS performs all the self tests except the system RAM (User Area RAM) test and the VDU RAM test.

2.2.1 Test Procedure.

Upon completion of the VDU RAM and 6845 VDU Controller test the ROS displays a message ("Please wait") on the first line of the screen to indicate that self testing is in progress and as each successive self test is started a dot is displayed on the screen.

The tests are run in the following order:

1. ROS checksum test.
2. VDU RAM and VDU (6845) Controller test.
3. Direct Memory Access (8237) Controller test.
4. Programmable Interval timer (8253) test.
5. Programmable Peripheral Interface (8255) test.
6. Real Time Clock (HD146818) test.
7. Asynchronous Communications Element (8250) test.
8. Parallel Printer Port test.
9. Mouse X and Y count register test.
10. System RAM test.
11. Programmable Interrupt Controller (8259) test.
12. Disk test.
13. Keyboard Interface test.

The ROS uses the stack during the Disk test, the Keyboard interface test and the Programmable Interrupt Controller test. All other self tests are executed without using either the stack or any RAM variables.

2.2.2 Test Methods.

Most of the device diagnostic tests consist of a Data Path test and a Waveform test as described below:

Data Path test.

> The data path test checks the read/write path between the CPU and a particular device. A pattern is written to a device and then read back to verify the integrity of the data path. The patterns are as follows:
>
> All zeros.
> All ones.
> Sliding single bit and complement across 8 bits.

Waveform test.

> The waveform test detects address decoding errors in a hardware device. The waveform test consists of selecting a specific address in a device, writing a test pattern (usually 0FFFFh) and verifying that the same pattern can be read back. The waveform test is done in both ascending sequential order (upwards) and descending sequential order (downwards) in order to check that the address decoding logic works correctly.

2.2.3 ROS Checksum Test.

All bytes in the Resident Operating System ROM are summed and then checked that the least significant byte of the sum is zero. If the check fails then an error message indicating faulty ROM checksum is displayed.

2.2.4 VDU RAM and VDU Controller test.

The following tests are performed to test the VDU RAM and the VDU controller:

> Data path test on the first byte of the four planes in VDU RAM.
> Upwards/Downwards waveform test on every location in the VDU RAM.
> 6845 VDU status register toggle bit test. This test confirms that the toggle bit in the VDU status register reverses states after each read of the status register.
> 6845 VDU status register frame flyback test. This test checks that the duration of the frame flyback signal conforms to the specification.
> If any of the tests fail the faulty VDU error message is displayed.

2.2.5 Direct Memory Access Controller test.

The upwards/downwards waveform test is used to confirm that the registers in the DMA controller chip can be addressed. Any failure will cause the faulty DMA error message to be displayed.

2.2.6 Programmable Interval Timer test.

The fiest 8253 test is a read/write data path test to counter 2 followed by a check that counter 1 counts at the correct rate. If either test fails a interval timer error message is displayed.

2.2.7 Programmable Peripheral Interface test.

The 8255 PPI tests consists of a data path test on each of the two system status channels (Status-1 and Status-2). The 8253 PIT OUT2 (Status-2) bit is also checked for proper operation. If either test fails the faulty real time clock error message is displayed.

2.2.8 Real Time Clock test.

The RTC seconds counter is tested to be counting at the correct rate. Next a data path test on the checksum byte of the NVR is run (and the checksum byte is restored). If either test fails the faulty real time clock error message is displayed.

2.2.9 Asyncronous Communications Element test.

This test confirms that the transmitter and receiver of the 8250 (i. e. the system serial port) are functioning correctly (at least in diagnostic mode).

The 8250 is configured in loop mode, 9600 baud, 8 data bits, 1 stop bit and no parity. Two test patterns are transmitted and the received patterns are checked. The status register is monitored for no parity, framing or overrun errors. If either received pattern does not equal the sent pattern or an error is set in the status register the faulty system serial port error message is displayed.

2.2.10 Printer Parallel Port test.

A data path test is performed on the printer data latch. If any incorrect test pattern is returned, the faulty printer port error message is displayed.

2.2.11 Mouse X and Y Count Register test.

The X and Y registers are cleared and then read to verify that they both contain zero. If the test fails the faulty mouse coordinate register error message is displayed.

2.2.12 System RAM test.

The amount of System (User Area) RAM is determined using the procedure described in section 2.1. The data path test is run on all available RAM followed by an upwards/downwards waveform test. If either test fails the faulty RAM error message is displayed.

2.2.13 Programmable Interrupt Controller test.

The 8259 tests consist of a data path test on the interrupt mask register and an interrupt acknowledge test to confirm that interrupts can occur and be serviced. If the test tails the faulty interrupt controller message is displayed.

2.2.14 Disk test.

The disk test attempts to establish whether the drives fitted to the system seek correctly. The test moves the read/write heads to track 10 on each drive. The ROS does not verify that the correct track was attained. If any errors are reported then a floppy disk controller error message is displayed.

2.2.15 Keyboard Interface test.

Upon power-up or reset, the keyboard self test is performed by the keyboard controller firmware. The keyboard returns keycode 0AAh to signify the successful completion of its testing. If any key code other than 0AAh is returned the keyboard error message is displayed amd keyboard reset is issued (which reruns the keyboard self test). The Keyboard test is repeated until the keyboard test passes. When test pass is received, the error message is removed from the screen and the test is exited as normal. During the keyboard test a short beep is sounded every five seconds to indicate that the test is in progress.

2.3 ROS Interrupts.

The first 32 interrupt vectors are initialised by Power-Up initialisation. The software IRET and hardware HWIRET entries are dummy routines which require no entry or exit conditions and are not detailed here.

Any application program which replaces a default interrupt vector with its own entry point must not invoke any ROS interrupts from within its own interrupt service routine.

2.3.1 INTERRUPT 2: Parity Error (NMI).

The Interrupt 2 routine deals with system RAM parity error. The screen is switched to the default display mode, cleared and a RAM parity error message is displayed. The machine cannot be used until the power switch is cycled off and on again.

This routine does not use RAM for stack or program variables.

An application program which makes use of the 8087 NDP must supply an interrupt 2 service routine for the 8087 NDP.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    Doesn't exit.

2.3.2 INTERRUPT 5: Print Screen.

The Interrupt 5 routine dumps the screen in character mode to the primary printer port. Since the screen dump is character based, attempting to dump graphic pictures to the printer may produce incorrect results. Characters that cannot be read back from the screen in graphics mode (using the VDU interrupt read character sub-function) are printed as spaces.

If a screen print is already in progress the interrupt takes no action.

The Print Screen Status variable (at address 00500) is set to 1 while the screen dump is in progress. When complete the variable is set to zero. If the screen dump is abandoned due to printer port timeout, the variable is set to 255.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    All flags and registers preserved.

2.3.3 INTERRUPT 6: Mouse Button Control.

The ROS interrupt 6 routine provides default mouse button services. The keyboard firmware generates a set of make/break keycodes when either of the two mouse buttons is pressed and released. When the keyboard interrupt routine recognizes a mosue button keycode it invokes interrupt 6. The default ROS routine will either obtain the appropriate keycode from the NVR and return with the carry flag set in the case of a make code or return with the carry flag clear in response to a mouse button break code.

CPU registers are used as follows:

Entry:
    Register AL = Mouse Key Code
    Bit 0 specifies which mouse button:
        0 = Mouse Button M1.
        1 = Mouse Button M2.
    Bit 7 specifies whether make or break:
        0 = Mouse Button make.
        1 = Mouse Button break.
Exit:
    Carry flag & Register Ax specify action:
    Carry SET:
        Insert a key token into the keyboard buffer.
        AX = Key Token to be inserted.

Carry CLEAR:

No action to be taken.

AX is corrupt.

All other flags corrupt. (also BX, CX & DX may be corrupt.)

All other registers preserved.

Note: A key token value of FFFFh is ignored and is not put in the keyboard buffer.

2.3.4 INTERRUPT 8: System Clock Interrupt.

The interrupt 8 routine is invoked by the system clock (counter 0 of the 8253). The default ROS routine does the following:

1. Increment the 32-bit system clock count held in RAM (location 0046C). If the clock reaches the 24 hour time (0001855000h) then the count is reset to zero and the 24 hour flag (location 00470) is set to 0FFh.
2. If the least significant byte of the system clock count is zero then the current time and date in the real time clock (RTC) is copied to the NVR. The time that is last copied from the RTC before the machine is switched off is displayed when the machine is next switched on.
3. If the disk motor timeout count is not zero then it is decremented by one. If the count reaches zero all the drive motors are turned off.
4. Invoke interrupt 28. Application programs that want to be interrupted by the system clock should use interrupt 28.

CPU registers are used as follows:

Entry:

No conditions.

Exit:

All flags and registers preserved.

2.3.5 INTERRUPT 9: Keyboard Interrupt.

The ROS Keyboard hardware interrupt reads a key code from the keyboard interface, translates the key code into a 16-bit key token using an internal translation table and the key token is put into the key token buffer. If the buffer is full the key token is discarded and a bleep is output on the speaker.

Entry:

No conditions.

Exit:

All flags and registers preserved.

The ROS Keycode translation table is as follows:

Key Code	(UK) Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
01	ESC	011B	Ignored	011B	011B	N/A
02	1 and !	0231	7800	Ignored	0221	N/A
03	2 and "	0332	7900	0300	0340	N/A
04	3 and £	0433	7A00	Ignored	0423	N/A
05	4 and $	0534	7B00	Ignored	0524	N/A
06	5 and %	0635	7C00	Ignored	0625	N/A
07	6 and ^	0736	7D00	071E	075E	N/A
08	7 and &	0837	7E00	Ignored	0826	N/A

Key Code	(UK) Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
09	8 and *	0938	7F00	Ignored	092A	N/A
OA	9 and (	0A39	8000	Ignored	0A28	N/A
0B	0 and)	0B30	8100	Ignored	0B29	N/A
0C	- and _	0C2D	8200	0C1F	0C5F	N/A
0D	= and +	0D3D	8300	Ignored	0D2B	N/A
0E	<-DEL	0E08	Ignored	0E7F	0E08	N/A
0F	TAB	0F09	Ignored	Ignored	0F00	N/A
10	Q	1071	1000	1011	1051	N/A
11	W	1177	1100	1117	1157	N/A
12	E	1265	1200	1205	1245	N/A
13	R	1372	1300	1312	1352	N/A
14	T	1474	1400	1414	1454	N/A
15	Y	1579	1500	1519	1559	N/A
16	U	1675	1600	1615	1655	N/A
17	I	1769	1700	1709	1749	N/A
18	O	186F	1800	180F	184F	N/A
19	P	1970	1900	1910	1950	N/A
1A	[and {	1A5B	Ignored	1A1B	1A7B	N/A
1B	] and }	1B5D	Ignored	1B1D	1B7D	N/A
1C	CR ←	1C0D	Ignored	1C0A	1C0D	N/A
1D	CTRL	Ignored	Ignored	----	Ignored	N/A
1E	A	1E61	1E00	1E01	1E41	N/A
1F	S	1F73	1F00	1F13	1F53	N/A
20	D	2064	2000	2004	2044	N/A
21	F	2166	2100	2106	2146	N/A
22	G	2267	2200	2207	2247	N/A
23	H	2368	2300	2308	2348	N/A
24	J	246A	2400	240A	244A	N/A
25	K	256B	2500	250B	254B	N/A
26	L	266C	2600	260C	264C	N/A
27	; and :	273B	Ignored	Ignored	273A	N/A
28	' and @	2827	Ignored	Ignored	2822	N/A
29	# and ~	2960	Ignored	Ignored	297E	N/A
2A	LEFT SHIFT	Ignored	Ignored	Ignored	----	N/A
2B	\ and |	2B5C	Ignored	2B1C	2B7C	N/A
2C	Z	2C7A	2C00	2C1A	2C5A	N/A
2D	X	2D78	2D00	2D18	2D58	N/A
2E	C	2E63	2E00	2E03	2E43	N/A
2F	V	2F76	2F00	2F16	2F56	N/A
30	B	3062	3000	3002	3042	N/A
31	N	316E	3100	310E	314E	N/A
32	M	326D	3200	320D	324D	N/A
33	, and <	332C	Ignored	Ignored	333C	N/A
34	. and >	342E	Ignored	Ignored	343E	N/A
35	/ and ?	352F	Ignored	Ignored	353F	N/A
36	RIGHT SHIFT	Ignored	Ignored	Ignored	----	N/A
* 37	* and PRTSC	372A	Ignored	7200	Prnt Scrn	N/A

Key Code	(UK) Key Name	Normal	ALT	CTRL	SHIFT	Num Lock
38	ALT	Ignored	----	Ignored	Ignored	N/A
39	SPACE	3920	3920	3920	3920	N/A
* 3A	CAPS LOCK	Ignored	Ignored	Ignored	Ignored	N/A
3B	F1	3B00	6800	5E00	5400	N/A
3C	F2	3C00	6900	5F00	5500	N/A
3D	F3	3D00	6A00	6000	5600	N/A
3E	F4	3E00	6B00	6100	5700	N/A
3F	F5	3F00	6C00	6200	5800	N/A
40	F6	4000	6D00	6300	5900	N/A
41	F7	4100	6E00	6400	5A00	N/A
42	F8	4200	6F00	6500	5B00	N/A
43	F9	4300	7000	6600	5C00	N/A
44	F10	4400	7100	6700	5D00	N/A
* 45	NUM LOCK	Ignored	Ignored	PAUSE	Ignored	N/A
* 46	SCROLL LOCK	Ignored	Ignored	BREAK	Ignored	N/A
47	KEY PAD 7	4700	Ignored	7700	N/A	4737
48	KEY PAD 8	4800	Ignored	Ignored	N/A	4838
49	KEY PAD 9	4900	Ignored	8400	N/A	4939
4A	KEY PAD -	4A2D	Ignored	Ignored	N/A	4A2D
4B	KEY PAD 4	4B00	Ignored	7300	N/A	4B34
4C	KEY PAD 5	Ignored	Ignored	Ignored	N/A	4C35
4D	KEY PAD 6	4D00	Ignored	7400	N/A	4D36
4E	KEY PAD +	4E2B	Ignored	Ignored	N/A	4E2B
4F	KEY PAD 1	4F00	Ignored	7500	N/A	4F31
50	KEY PAD 2	5000	Ignored	Ignored	N/A	5032
51	KEY PAD 3	5100	Ignored	7600	N/A	5133
* 52	KEY PAD 0 (INS)	5200	Ignored	Ignored	N/A	5230
53	KEY PAD .	5300	Ignored	Ignored	N/A	532E
54 - 6F	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored
* 70	DEL ->	N/A	N/A	N/A	N/A	N/A
71 - 73	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored
* 74	ENTER	N/A	N/A	N/A	N/A	N/A
75 - 76	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored
* 77	JOY FIRE2	N/A	N/A	N/A	N/A	N/A
* 78	JOY FIRE1	N/A	N/A	N/A	N/A	N/A
* 79	JOY RIGHT	4D00	4D00	4D00	4D00	4D00
* 7A	JOY LEFT	4B00	4B00	4B00	4B00	4B00
* 7B	JOY DOWN	5000	5000	5000	5000	5000
* 7C	JOY UP	4800	4800	4800	4800	4800
* 7D	MOUSE M2	N/A	N/A	N/A	N/A	N/A
* 7E	MOUSE M1	N/A	N/A	N/A	N/A	N/A
7F	UNDEFINED	Ignored	Ignored	Ignored	Ignored	Ignored

Joystick keys produce their respective cursor keys.

Key codes marked with '*' cause special actions as explained on the next page.

2.3.5.1 Special Key Actions.

Some keys or set of keys invoke a special action as detailed below. Unless otherwise stated they do not result in any key tokens being inserted into the buffer.

1. [CTRL]+[ALT]+[DEL]: Reset.
   When reset is detected, a system hardware reset is issued. The power-up initialisation process is entered but System RAM and VDU RAM tests are not run.
2. [CTRL]+[NUM LOCK]: Pause.
   The ROS waits for another key to be pressed (except [CTRL]+[NUM LOCK]), thus suspending any application that is running.
3. [CTRL]+[SCROLL LOCK]: Break.
   When break is detected, interrupt 27 is invoked and the keyboard buffer is cleared. Key token 0000h is then inserted into the buffer.
4. [SHIFT]+[PRTSC]: Print Screen.
   When print screeen is detected interrupt 5 is invoked, the ROS print screen function.
5. [INS]: Insert Toggle.
   Each time the INS key code (52) is received, except in NUM LOCK mode, the INS key toggle bit (bit 7 of RAM location 00417) is inverted.
6. [SCROLL LOCK]: Scroll Toggle.
   Each time the SCROLL LOCK key is pressed the scroll key toggle bit (bit 4 of RAM location 00417) is inverted. Note that CTRL - SCROLL LOCK (break) does not flip the scroll toggle.
7. [CAPS LOCK]: Caps Lock Toggle.
   Each time the CAPS LOCK key is pressed the Caps Lock toggle bit (bit 6 of RAM location 00417) is inverted.
8. [NUM LOCK]: Num Lock Toggle.
   Each time the NUM LOCK key is pressed the Num Lock toggle bit (bit 5 of RAM location 00417) is inverted.
9. [ALT]+[NUMERIC KEY PAD 0 to 9]: Absolute Key Token.
   When the ALT key is held down, an absolute key token may be entered via the numeric keypad. Pressing any other key resets the absolute key token to zero (and inserts the associated ALT-key token for the key pressed). When ALT is released the absolute key token modulo 256 is placed into the keyboard buffer, unless the token is zero, in which case it is discarded.
10. [ENTER] and FIRE buttons.
    When the ENTER key code (74) or one of the two Joystick FIRE button key codes (78 or 79) is received an associated key token is obtained from the NVR and inserted into the key token buffer.
11. MOUSE buttons.
    The keyboard firmware generates four key codes to indicate when the two mouse buttons are pressed or released. When the ROS receives one of these codes from the keyboard it does a far call to the address held in the Mouse Button interrupt vector (Interrupt 6). A default ROS routine is loaded into this vector upon power-up or system reset. This routine requests the ROS to insert a key token held in the NVR into the keyboard buffer, the token used depends on which mouse button (M1 or M2) is received. The mouse button release codes are ignored.
12. [DEL ->]: Forward Delete.
    When the forward delete key code (code 70) is received a key token is obtained from the NVR and placed in the key token buffer.

13 [ALT], [CTRL], [SHIFT], [CAPS LOCK] & [NUM LOCK].

The translation of various key codes into their respective tokens is affected by the current states of these keys (which is stored in location 00417). The SHIFT key, while pressed, reverses the current state of the CAPS LOCK and NUM LOCK. If more than one of ALT,CTRL, SHIFT,NUM LOCK or CAPS LOCK is active at one time then the order of precedence for key code translation is ALT, then CTRL, then SHIFT, then CAPS LOCK or NUM LOCK.

CAPS LOCK, when active, converts the key tokens for the lower case alphabetic keys (a - z) to their upper case counterparts.

Note that some operating systems (such as DOS Plus) install their own entry points into the interrupt vectors and these interrupt routines may exhibit different characteristics than those of the ROS routines described here.

2.3.6 INTERRUPT 14: Floppy Disk Controller.

The ROS service routine for interrupt 14 sets bit 7 of the RAM DRIVE RESTORE FLAG, to indicate that the Floppy Disk Controller interrupt has occurred.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    All flags and registers preserved.

2.3.7 INTERRUPT 16: VDU I/O.

The ROS interrupt 16 service routine provides a set of routines for reading and writing characters in alpha and graphics mode. In graphics mode the characters are constructed using a character matrix table (see section 2.3.20). It also provides facilities for scrolling the screen up or down, reading and writing pixels (graphics only) and reading the light pen.

CPU registers are used as follows:

Entry:

AH = Sub-function number:

- 0 - Set VDU Mode.
- 1 - Set Cursor Size.
- 2 - Set Cursor Address.
- 3 - Get Cursor Address.
- 4 - Get Light Pen Address.
- 5 - Set Display Page.
- 6 - Scroll Screen Up.
- 7 - Scroll Screen Down.
- 8 - Read Character and Attributes.
- 9 - Write Character and Attributes.
- 10 - Write Character only.
- 11 - Write Colour Select Register.
- 12 - Write Pixel.
- 13 - Read Pixel.
- 14 - Write Character in Teletype Emulation mode.
- 15 - Get VDU Parameters.

All other registers as required by the sub-function.

Exit:

If selector is greater than 15 then carry is set, else carry is clear.

All other flags and registers as specified by the sub-function.

Alpha modes 0 and 1 require 2000 bytes of VDU RAM while alpha modes 2 and 3 require 4000 bytes of the VDU RAM. The ROS takes advantage of all the (16K bytes) VDU RAM available in alpha modes by supporting multiple display pages. This means that application programs can set up a number of display pages and switch them as required.

In general parameters passed to ROS routines are not checked and care should be taken when choosing unusual parameters as unexpected results may occur. In particular be careful of boundary conditions such as setting the top of the display window equal to the bottom of the display window (for sub-functions 6 and 7) effectively creating a one line display. In this instance the scroll screen routines may not perform as expected.

VDU Sub-Function 0: Set VDU Mode.

CPU registers are used as follows:

Entry:

AH = 0
AL = VDU Mode:

0 - Alpha 25 Rows by 40 Columns.
1 - Same as Mode 0.
2 - Alpha 28 Rows by 80 Columns.
3 - Same as mode 2.
4 - Graphics 200 pixels by 320 pixels using palette 1.
5 - Graphics 200 pixels by 320 pixels using palette 2.
6 - Graphics 200 pixels by 640 pixels.
7 - Alpha 25 Rows by 80 Columns using Monochrome Adapter.

Exit:

All flags and registers preserved.

In mode 4 palette 0 may be selected by writing the colour select register using VDU sub-function 11. The definition of the palettes is contained in Section 1.11.2.1, Graphics Mode 1.

When mode 5 is selected it must be followed by a selection of palette zero (VDU Colour select register - See 1.11.3) in order to enable palette 2.

If the Status-1 default display mode bits (see 1.8.2: DDM1-DDM0) are both set, indicating an external monochrome adapter, then mode 7 is selected regardless of the mode in AL.

To select the VDU mode the ROS does the following:

1. Disable video output.
2. Reset the Cursor Addresses for all pages to row 0 column 0.
3. Output the mode to the VDU mode select register.
4. Reload the VDU controller 6845 emulation registers from the VDU parameter table (which is supplied by interrupt 31).
5. Clear the (16K bytes of) VDU RAM or the 4k bytes of the monochrome adapter if mode 7 is selected. If an alpha mode is selected, the VDU RAM is filled with white space, i.e. ASCII space (020h) and the default attribute byte held in the NVR. The graphics mode fill is zeroes.

6. Set up the VDU colour select register:

   Set the border colour to the default background colour.
   In graphics modes except mode 6 set intensified foreground colours.
   In mode 6 (Graphics 640 Mode) set white foreground colour.

NOTE: The mode 6 setting of white foreground colour is done by setting the VDU Colour Select Register (I/O Address 3D9) to 7. This means that in order to subsequently make use of the full 16 colour display capability the VDU Colour Select Register must be set to 0F.

7. For modes 0 to 3 select page zero.
8. Set the cursor size to start cursor display on scan 6 and end on scan 7.
9. Enable VDU output.

VDU Sub-Function 1: Set Cursor Size.

This function is only relevant in alpha modes as the hardware cursor is not supported in graphics modes. It sets the start and end scan numbers of the cursor.

CPU registers are used as follows:

Entry:
    AH = 1
    CH = Starting scan of cursor in range 0 to 31d.
    CL = Ending scan of cursor in range 0 to 31d.
Exit:
    All flags and registers preserved.

To hide the cursor specify a starting scan value of 31. Also Values greater than 31 will be interpetered as 'hide cursor' by the ROS.

VDU Sub-Function 2: Set Cursor Address.

This function sets the current row and column addresses of the cursor in the specified page.

CPU registers are used as follows:

Entry:
    AH = 2
    BH = Page number for modes 0 to 3.
        (Must be zero for all other modes.)
    DH = Cursor Row Address.
    DL = Cursor Column Address.
Exit:
    All flags and registers preserved.

In modes 0 and 1 (25 X 40 alpha mode) eight pages (0-7) are supported.

In modes 2 and 3 (25 X 80 alpha mode) four pages (0-3) are supported.

VDU Sub-Function 3: Get Cursor Address.

This function returns the current row and column address of the cursor in the specified page.

CPU registers are used as follows:

Entry:
    AH = 3
    BH = Page number for modes 0 to 3.
        (Must be zero for all other modes.)

Exit:
    DH = Cursor Row Address.
    DL = Cursor Column Address.
    CH = Starting scan of cursor.
    CL = Ending scan of cursor.
    All flags and other registers preserved.

In modes 0 and 1 (25 X 40 alpha mode) eight pages (0-7) are supported.

In modes 2 and 3 (25 X 80 alpha mode) four pages (0-3) are supported.

VDU Sub-Function 4: Get Light Pen Address.

This function returns the address of the light pen.

CPU registers are used as follows:

Entry:
    AH = 4.
Exit:
    If Light Pen switch set then
        AH = 1.
        DH = Character Row address (0 to 24).
        DL = Character Column address (0 to 79).
        CH = Pixel Row address (0 to 199).
        BX = Pixel Column address (0 to 639).
    If Light Pen switch clear then
        AH = 1.
        BX, CH & DX preserved.
    Always
        All flags and other registers preserved.

VDU Sub-Function 5: Set Display Page.

This function sets the active display page.

CPU registers are used as follows:

Entry:
    AH = 5.
    BH = Page number to be displayed.
Exit:
    All flags and registers preserved.

In modes 0 and 1 (25 X 40 alpha mode) eight pages (0-7) are supported.

In modes 2 and 3 (25 X 80 alpha mode) four pages (0-3) are supported.

VDU Sub-Function 6: Scroll Screen UP.

This function scrolls the active display page, or part of the active display page up a specified number of lines.

CPU registers are used as follows:

Entry:

AH = 6.
DH = Bottom Row of area to scroll.
DL = Right most Column of area to scroll.
CH = Top Row of area to scroll.
CL = Left most Column of area to scroll
BH = Attributes for blank lines scrolled onto the bottom of the scroll area.
AL = Number of lines to roll up.
If AL = 0 then blank the specified area.
If AL not zero then roll specified area up by the number of lines in AL.
If DH = CH then AL must be zero.

Exit:

All registers preserved.
Carry is clear and all other flags corrupt.

Scrolling always takes effect on the current active display page.

Hardware scrolling is not supported. Scrolling is achieved by copying areas of VDU RAM.

In graphics modes blank lines are filled with attribute byte specified in BH to display the current background colour.

Note this function will fail to operate properly if on entry CH equals DH and AL is not zero. This is also true for all other compatible ROM environments.

VDU Sub-Function 7: Scroll Screen down.

This function scrolls the active display page, or part of the active display page down a specified number of lines.

CPU registers are used as follows:

Entry:

AH = 7.
DH = Bottom Row of area to scroll.
DL = Right most Column of area to scroll.
CH = Top Row of area to scroll.
CL = Left most Column of area to scroll
BH = Attributes for blank lines scrolled onto the top of the scroll area.
AL = Number of lines to roll down.
If AL = 0 then blank the specified area.
If AL not zero then roll specified area down by the number of lines in AL.
If DH = CH then AL must be zero.

Exit:

All flags and registers preserved.

Scrolling always takes effect on the current active display page.

Hardware scrolling is not supported. Scrolling is achieved by copying areas of VDU RAM.

Note this function will fail to operate properly if on entry CH equals DH and AL is not zero.

VDU Sub-Function 8: Read Character and Attributes.

This function reads the character and its associated attribute byte at the current cursor address in a specified display page.

In graphics modes, the character pixel data is generated either from an internal character matrix table for characters 0 to 127 or from an external character matrix table for characters 128 to 255, the address of which is held in interrupt vector 31. See 2.3.22 for additional details.

CPU registers are used as follows:

Entry:
    AH = 8.
    BH = Page to read for alpha modes 0 to 3.
        (Must be zero for all other modes.)
Exit:
    AL = Character. (0 if no match found in Graphics Modes).
    AH = Attributes byte. (Unchanged in graphics modes).
    All flags and registers preserved.

Refer to 1.11.1 for the definition of the character attributes byte.

VDU Sub-Function 9: Write Character and Attributes.

This function writes a character (or a block of the same character) and its associated attribute byte to the current cursor position in a specified display page.

In graphics modes, the character pixel data is generated either from an internal character matrix table for characters 0 to 127 or from an external character matrix table for characters 128 to 255, the address of which is held in interrupt vector 31. See 2.3.22 for additional details.

CPU registers are used as follows:

Entry:
    AH = 9.
    AL = Character to write.
    BH = Page to write for alpha modes 0 to 3.
        (Must be zero for all other modes.)
    BL = In alpha modes
        Attributes of character.
    In graphic modes
        Write mode as follows:
            Bit 7 = 0 for character overwrite mode.
            Bit 7 = 1 for character XOR mode.
            Bits 0 and 1 = required character colour (for modes 4 & 5 only).
    CX = Repeat Count.
Exit:
    All flags and registers preserved.

The repeat count specifies the number of consecutive locations to which the character and attributes are written. In graphics modes all characters must fit on the current line.

In graphics mode if bit 7 of BL is set then the data for the specified character is exclusive ORed with the data already in the VDU RAM at the cursor address.

VDU Sub-Function 10: Write Character Only.

This function writes a character (or a block of the same character) to the current cursor position in a specified display page. In alpha modes the attribute bytes for all characters written remains unchanged.

In graphics modes, the character pixel data is generated either from an internal character matrix table for characters 0 to 127 or from an external character matrix table for characters 128 to 255, the address of which is held in interrupt vector 31. See 2.3.22 for additional details.

CPU registers are used as follows:

Entry:
    AH = 10.
    AL = Character to write.
    BH = Page to write for alpha modes 0 to 3.
        (Must be zero for all other modes.)
    BL = In alpha modes
        BL is not used.
    In graphic modes
        Write mode as follows:
            Bit 7 = 0 for character overwrite mode.
            Bit 7 = 1 for character XOR mode.
            Bits 0 and 1 = required character colour (for modes 4 & 5 only).
    CX = Repeat Count.
Exit:
    All flags and registers preserved.

The repeat count specifies the number of consecutive locations to which the character is written.

In graphics mode if bit 7 of BL is set then the data for the specified character is exclusive ORed with the data already in the VDU RAM at the cursor address.

VDU Sub-Function 11: Write Colour Select Register.

This function writes the VDU Colour Select Register IRGB bits or the Palette select bits.

CPU registers are used as follows:

Entry:
    AH = 11.
    BH = Function select:
        Zero - Set the IRGB bits (3-0) as specified by BL.
        Non Zero - Set the Palette (0 or 1) as specified by BL.
Exit:
    All flags and registers preserved.

Changing the palette number (BH non-zero) only has effect in modes 4 and 5 (320 pixel graphics mode). Refer to section 1.11.3 for further details.

VDU Sub-Function 12: Write a Pixel.

This function writes an individual pixel (only valid in graphics modes).

CPU registers are used as follows:

Entry:
    AH = 12.
    DX = Pixel Row (0 to 199)
    CX = Pixel Column (0 to 639)
    AL = Write Mode:
        Bit 7 = 0 for character overwrite mode.
        Bit 7 = 1 for character XOR mode.
        Bits 0 & 1 = Character Colour for 320 graphics modes.
Exit:
    All flags and registers preserved.

The pixel colour specified in AL should be in the range 0 to 3 in modes 4 and 5 (graphics 320 pixel mode) and in the range 0 to 1 for mode 6 (graphics 640 pixel mode).

VDU Sub-Function 13: Read a Pixel.

This function is used for reading an individual pixel (only in graphics modes).

CPU registers are used as follows:

Entry:
    AH = 13.
    DX = Pixel Row (0 to 199)
    CX = Pixel Column (0 to 639)
Exit:
    AL = Colour of the specified pixel.
    All flags and other registers preserved.

VDU Sub-Function 14: Write in TTY Emulation Mode.

This function writes the specified character in Teletype emulation mode at the current cursor address in the active display page.

CPU registers are used as follows:

Entry:
    AH = 14.
    AL = Character to write.
    BL = In alpha modes
        BL is not used.
    In graphic modes
        Write mode as follows:
            Bit 7 = 0 for character overwrite mode.
            Bit 7 = 1 for character XOR mode.
            Bits 0 and 1 = required character colour (for modes 4 & 5 only).
Exit:
    All flags and registers preserved.

Upon completion of the write the cursor column is incremented by one. If the column address is greater than the line length then the column address is set to zero and the cursor row address is incremented by one.

If the incremented row address is greater than the last visible line then it is decremented to its original value and the entire page is scrolled up one line. In alpha modes the line added to the bottom of the page is cleared to spaces with the attributes the same as the first character in previous line. In graphic modes the bottom line is cleared to zeroes.

The following display characters are executed rather than displayed symbolically:

BEL (07h)
 Sounds a short (bleep) tone on the speaker.
BS (08h)
 Decrements the cursor column one character position unless the column is already zero in which case it is ignored.
CR (0Dh)
 Sets the cursor column address to zero.
LF (0Ah)
 Increments the cursor row address by one and follows the scroll up procedure as detailed in the paragraph above.

All other control characters are displayed.

VDU Sub-Function 15: Get Current VDU Parameters.

This function returns the current VDU mode, the current display page and number of visible columns.

CPU registers are used as follows:

Entry:
 AH = 15.
Exit:
 BH = Current active display Page (or zero if in graphics modes or alpha mode 7).
 AH = Number of visible columns (40 or 80).
 AL = Current VDU mode (0 to 7).

 All flags and other registers preserved.

2.3.8 INTERRUPT 17: System Configuration.

This software interrupt returns the current system configuration status ad defined in RAM locations 00410 and 00411 hex (see section 2.4).

CPU registers are used as follows:

Entry:
 No conditions.
Exit:
 AX = System Configuration status:

Bit(s)	**Function**
14 & 15	Number of printers (1-3).
13	Not used.
12	Set if an optional games adapter is fitted.
11	Always zero.
9 & 10	Number of serial interfaces (1 or 2).
8	Not used.
7	Always zero.

Bit(s)	Function
6	Set if second floppy disk drive is fitted.
4 & 5	Default VDU mode.
2 & 3	Always set.
1	Set if 8087 NDP is installed.
0	Always set.

All flags and other registers preserved.

Section 1.8.2 (Port A - Status-1 Input) contains the default mode states as defined in the DDM1 and DDM0 bits.

2.3.9 INTERRUPT 18: Memory Size.

This software interrupt returns the system RAM size as held in system locations 00413 and 00414 hex.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    AX = Number of 1K memory blocks fitted.
    All flags and other registers preserved.

2.3.10 INTERRUPT 19: Disk I/O.

This software interrupt provides disk read, write, verify, and format functions for the drives fitted to the standard floppy disk controller.

CPU registers are used as follows:

Entry:
    AH = Disk I/O sub-function selector:
        0 - Initialise the disk sub-system.
        1 - Return the status of the last operation.
        2 - Read a number of consecutive sectors.
        3 - Write a number of consecutive sectors.
        4 - Verify a number of consecutive sectors.
        5 - Format a track.
Exit:
    AH = Status Byte:
        0 - Operation completed successfully.
        1 - Incorrect sub-function (or drive) specifier.
        2 - Missing address mark error.
        3 - Disk write protected (Write or Format commands only).
        4 - Record not found.
        8 - DMA overrun error.
        9 - Attempted DMA over a 64K segment boundary.
        16 - CRC error.
        32 - Floppy disk controller error.
        64 - Seek error.
        128 - Floppy disk controller timeout (Drive Not Ready).

All other registers as specified by the selected sub-function.

For all disk sub-functions the Carry Flag (CF) will be clear if no error else it is set if an error (and AH = error number). All other flags are corrupt.

Disk Sub-Function 0: Initalise Disk Sub-System.

This sub-function performs a total initialisation of the disk interface as follows:

1. Reset the FDC (Floppy Disk Controller).
2. Re-configure the FDC parameters to those specified in the disk parameter table (see interrupt 30).

CPU registers are used as follows:

Entry:
    AH = 0.
Exit:
    AH/Flags = Status as specified above.
    All registers preserved.

When an error is returned by any other disk I/O sub-function, the Initialise Disk sub-function should be called prior to the next disk I/O operation.

Disk Sub-Function 1: Return Last Status.

This sub-function returns the status byte and Carry Bit of the last disk I/O operation.

CPU registers are used as follows:

Entry:
    AH = 1.
Exit:
    AH/Flags = Status of last disk I/O as specified above. (also AL = AH).
    All registers preserved.
    Interrupts enabled.

Disk Sub-Function 2: Read Sector.

This sub-function reads a number of consecutive sectors. All sectors to be read must be on the same track.

CPU registers are used as follows:

Entry:
    AH = 2.
    DH = Head Number (0 or 1).
    DL = Drive Number (0 or 1).
    CH = Track Number.
    CL = Starting Sector Number.
    BX = Offset Address of Read Data Buffer.
    ES = Segment Address of Read Data Buffer.
    AL = Number of Sectors to Read.
Exit:

AH/Flags = Status as specified above.
AL = Number of Sectors successfully read.
(Corrupt if Timeout error.)
All other registers preserved.
Interrupts enabled.

Disk Sub-Function 3: Write Sector.

This sub-function writes a number of consecutive sectors. All sectors to be written must be on the same track.

CPU registers are used as follows:

Entry:
AH = 3.
DH = Head Number (0 or 1).
DL = Drive Number (0 or 1).
CH = Track Number.
CL = Starting Sector Number.
BX = Offset Address of Write Data Buffer.
ES = Segment Address of Write Data Buffer.
AL = Number of Sectors to Write.
Exit:
AH/Flags = Status as specified in 2.3.10.
AL = Number of Sectors successfully written.
(Corrupt if Timeout error.)
All other registers preserved.
Interrupts enabled.

Disk Sub-Function 4: Verify Sector.

This sub-function verifies a number of consecutive sectors. All sectors to be verified must be on the same track.

CPU registers are used as follows:

Entry:
AH = 4.
DH = Head Number (0 or 1).
DL = Drive Number (0 or 1).
CH = Track Number.
CL = Starting Sector Number.
AL = Number of Sectors to Verify.
Exit:
AH/Flags = Status as specified above.
AL = Number of Sectors successfully verified.
(Corrupt if Timeout error.)
All other registers preserved.
Interrupts enabled.

Since the verification process is halted upon the first occurrence of an error, AL represents the number of sectors successfully verified prior to the occurrence of an error or total sectors verified if no error.

Disk Sub-Function 5: Format Track.

This sub-function formats an entire track.

CPU registers are used as follows:

Entry:
AH = 5.
DH = Head Number (0 or 1).
DL = Drive Number (0 or 1).
CH = Track Number.
BX = Offset Address of Format Buffer.
ES = Segment Address of Format Buffer.
Exit:
AH/Flags = Status as specified above.
All other registers preserved.
Interrupts enabled.

The format buffer contains four bytes of information for each sector on the track:

1. Track Number.
2. Side Number.
3. Sector Number.
4. Sector Size Code:
   0 - 128 bytes / Sector.
   1 - 256 bytes / Sector.
   2 - 512 bytes / Sector.
   3 - 1024 bytes / Sector.

The gap length, filler byte and sectors per track required by the FDC Format command are obtained from the DPT (See Disk Parameter Table - Section 2.3.21).

2.3.11 Interrupt 20: Serial I/O.

This software interrupt provides functions for character I/O to one of the two serial channels and functions for configuring the serial parameters.

Two channels are supported, logical serial device 0 (COM1:) which is always configured and logical serial device 1 (COM2:) which is optional. Power-up initialisation determines whether serial device 1 is installed.

CPU registers are used as follows:

Entry:
AH = Sub-function selector:
0 - Initialise Serial Port.
1 - Write Character to Serial Port.
2 - Read Character from Serial Port.
3 - Return status of Serial Port.
DX = Logical Channel Number (0 or 1).
All other registers as required by the specified sub-function.
Exit:
AX = Returned Status/Character as defined by the sub-function.
All flags and other registers preserved.

If logical channel number is out of range (greater than 1) or is not fitted then the function is abandoned and the timeout error status (bit 7 of AH) is returned and all other bits in AX are undefined.

The Logical Serial Device Timeout Count RAM variables (locations 0047C & 0047D) specify the time out delay (in half seconds) used for channel timeout. See section 2.4.

Serial Sub-Function 0: Initalise Port.

This sub-function performs a complete reinitialisation of a serial channel. Setting the Baud Rate, Data Bits, Stop Bits and Parity.

CPU registers are used as follows:

Entry:

AH = 0.
DX = Logical Channel Number (0 or 1).
AL = Hardware configuration:

Bit(s)	Function
5 - 7	Baud Rate Code (0 - 7).
4	Set for Even Parity / Clear for Odd Parity.
3	Set Parity Enable.
2	Set for 2 Stop Bits / Clear for 1 Stop Bit.
1	Always set.
0	Set for 8 Data Bits/Clear for 7 Data Bits.

Exit:

AH = 8250 Line Status register (See section 3.4).
AL = 8250 Modem Status register.
All flags and other registers preserved.

The Baud Rate code (bits 5 thru 7) is one of the following:

0 -	110 Baud.
1 -	150 Baud.
2 -	300 Baud.
3 -	600 Baud.
4 -	1200 Baud.
5 -	2400 Baud.
6 -	4800 Baud.
7 -	9600 Baud.

If the hardware flow control bit in the NVR default VDU mode byte is set then RTS is raised true and DTR is set false. Otherwise the current state of the control lines is preserved.

Serial Sub-Function 1: Send Character.

This sub-function performs a character out sequence to the selected port. The character is output when CTS and the 8250 Tx Holding Register Empty status is also true. If the character cannot be sent within the time specified in the logical serial device timeout count RAM variable then the command is abandoned and AH is returned with bit 7 set.

CPU registers are used as follows:

Entry:

AH = 1.

AL = Character to be sent.
DX = Logical Channel Number (0 or 1).

Exit:

AH = 8250 Line Status register bits 0 to 6. Bit 7 is set if the channel timed out else bit 7 is clear and the character was sent.

All flags and other registers preserved.

When this sub-function is called, RTS is raised true and if the hardware flow bit is disabled then DTR is set as well.

Upon exit both the RTS and DTR control lines are left in their current state.

The Logical Serial Device Timeout Count RAM variables (locations 0047C and 0047D) specify the time out delay (in half seconds) used for channel timeout.

Serial Sub-Function 2: Read Character.

This sub-function attempts to read a character from the specified serial port. The character is not read until both Data Ready (DR) and Data Set Ready (DSR) status bits are both true. If a character is not received within the time specified by the logical device timeout count then the command is abandoned and timeout status is flagged.

CPU registers are used as follows:

Entry:

AH = 2.
DX = Logical Channel Number (0 or 1).

Exit:

If character received from 8250 then
AL = Character received.
AH = Character status:

Bit(s)	Meaning
7 - 5	Always '0'.
4	Break status.
3	Set if framing error.
2	Set if parity error.
1	Set if overrun error.
0	Always '0'.

If logical channel timed out then
AH = 080h (bit 7 = 1).
Always
All flags and other registers preserved.

If the character is received with no errors then AH = 0 on exit.

Upon entry, if no character is available at the serial port DTR is set in the Modem Control Register. When a character is read or timeout occurs DTR is set false only if hardware flow control is enabled.

If logical channel number is out of range or is not fitted then the function is abandoned and the timeout error status (bit 7 of AH) is returned and all other bits in AX are undefined.

The Logical Serial Device Timeout Count RAM variables (locations 0047C and 0047D) specify the time out delay (in half seconds) used for channel timeout.

Serial Sub-Function 3: Get Channel Status.

This function returns the status of the specified logical channel.

CPU registers are used as follows:

Entry:
    AH = 3.
    DX = Logical Channel Number (0 or 1).
Exit:
    AH = 8250 Line Status register (See section 3.4).
    AL = 8250 Modem Status register.
    All flags and other registers preserved.

All flags and other registers preserved.

If logical channel number is out of range or is not fitted then the function is abandoned and the timeout error status (bit 7 of AH) is returned and all other bits in AX are undefined.

2.3.12 Interrupt 21: Enhanced Function Interrupt.

This software interrupt provides access to the enhanced hardware features of the AMSTRAD PC1512.

CPU registers are used as follows:

Entry:
    AH = Enhanced function Selector:
        0 - Read and Reset Mouse
        1 - Write NVR Location.
        2 - Read NVR Location.
        3 - Write the VDU Colour Plane Write register.
        4 - Write the VDU Colour Plane Read register.
        5 - Write the VDU Graphics Border registers.
        6 - Return ROS Version Number.
    All other registers as required by specified sub-function.
Exit:
    If sub-function number out of range then
        Carry True.
        All other flags corrupt.
        All registers preserved.
    If sub-function within range then
        All flags and registers as specified by the sub-function.

Enhanced Sub-Function 0: Read/Reset Mouse X/Y Counts.

Read and reset the mouse X and Y count registers. Each register is read twice. If the data from two consecutive reads differs then the process is repeated until two consecutive reads produce the same data. Upon completion of the read procedure the registers are cleared to zero.

CPU registers are used as follows:

Entry:
    AH = 0.
Exit:
    CX = Signed X count.
    DX = Signed Y count.
    Carry False.
    Other flags corrupt.
    All other registers preserved.

Enhanced Sub-Function 1: Write NVR Location.

This sub-function writes a specified location in the Real Time Clock Non-Volatile RAM (NVR), re-computes and stores the new checksum value. The location written is then read back and compared with the new value and if different an error code is returned.

CPU registers are used as follows:

Entry:
    AH = 1.
    AL = NVR Address to be written (0 to 63).
    BL = NVR Data to be written.
Exit:
    AH = Return Code:
        0 - NVR written successfully.
        1 - NVR Address out of range.
        2 - NVR Data write error.
    Carry false.
    All other flags corrupt.
    All other registers preserved.

Although locations 0-13 may be accessed using this function, they are used by the RTC hardware and should not be modified with this function.

Section 2.5 (Non Volatile RAM) contains the NVR information layout.

Enhanced Sub-Function 2: Read NVR Location.

This sub-function reads a specified location in the Real Time Clock Non-Volatile RAM (NVR). The checksum is computed and compared with the actual value and if the NVR checksum is incorrect an error code is returned.

CPU registers are used as follows:

Entry:
    AH = 2.
    AL = NVR Address to be read (0 to 63).
Exit:
    AH = Return Code:
        0 - NVR read successfully.
        1 - NVR Address out of range.
        2 - NVR checksum error.
    AL = Byte read from NVR.
    Carry false.
    All other flags corrupt.

All other registers preserved.

Section 2.5 (Non Volatile RAM) contains the NVR information layout.

Enhanced Sub-Function 3: Write VDU Colour Plane Write Register.

This sub-function writes an 8-bit value to the VDU Colour Plane Write register. Section 1.11.3 contains the details of this register.

CPU registers are used as follows:

Entry:
AH = 3.
AL = Value (I, R, G, B bits).
Exit:
Carry false.
All other flags corrupt.
All other registers preserved.

Enhanced Sub-Function 4: Write VDU Colour Plane Read Register.

This sub-function writes an 8-bit value to the VDU Colour Plane Read register. Section 1.11.3 contains the details of this register.

CPU registers are used as follows:

Entry:
AH = 4.
AL = Value (RDSEL1 and RDSEL0).
Exit:
Carry false.
All other flags corrupt.
All other registers preserved.

Enhanced Sub-Function 5: Write VDU Graphics Border Register.

This sub-function writes an 8-bit value to the VDU Graphics Mode 2 Border register. Section 1.11.3 contains the details of this register.

CPU registers are used as follows:

Entry:
AH = 5.
AL = Value to be written.
Exit:
Carry false.
All other flags corrupt.
All other registers preserved.

Enhanced Sub-Function 6: Return ROS Version Number.

This sub-function returns the two part ROS version number.

CPU registers are used as follows:

Entry:
    AH = 6.
Exit:
    BH = Release number.
    BL = Issue Number.

    Carry false.
    All other flags corrupt.
    All other registers preserved.

The Release Number is incremented only when the interface to the ROS is changed. The Issue Number is incremented for each version of a particular release. A new release always starts with issue number zero.

Note that this function call can be used to detect whether a program is running on an Amstrad PC1512. Prior to entry clear the carry flag and set BX to zero. Upon return if carry is set or BX is zero then the program is not running in an Amstrad PC1512.

2.3.13 Interrupt 22: Keyboard I/O.

This software interrupt provides access to the keyboard buffer and the current toggle status.

CPU registers are used as follows:

Entry:
    AH = Keyboard I/O sub-function selector.
        0 - Get a key token from the keyboard buffer.
        1 - Return Keyboard Buffer status.
        2 - Return current Key Toggle and Key States.
Exit:
    If sub-function selector out of range then
        AH = AH - 2.
    If sub-function within range then
        All flags and registers as specified by sub-function.

Keyboard I/O Sub-Function 0: Get Key Token.

Return the next token from the key token buffer. If no key token is available then wait until a key token is available.

CPU registers are used as follows:

Entry:
    AH = 0.
Exit:
    AX = Key Token.
    All flags and other registers preserved.

Keyboard I/O Sub-Function 1: Return Keyboard Buffer Status.

Test whether the key token buffer is empty. If it is not empty return the next key token to be taken out of the buffer without removing it from the buffer.

CPU registers are used as follows:

Entry:
    AH = 1.
Exit:
    If key token buffer is empty then
        Zero flag true.
        AX corrupt.
    If one or more tokens in buffer then
        Zero flag false.
        AX = next key token to be removed from buffer.
    Always
        Interrupts enabled.
        All other flags corrupt.
        All other registers preserved.

Keyboard I/O Sub-Function 2: Return Shift States.

Return the current value of the shift states (from 00417h).

CPU registers are used as follows:

Entry:
    AH = 2.
Exit:
    AL = Current shift states:

Bit(s)	Function (Set if key active)
7	INS
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK
3	ALT
2	CTRL
1	LEFT SHIFT
0	RIGHT SHIFT

All flags and other registers preserved.

2.3.14 Interrupt 23: Printer I/O.

This software interrupt provides access to the three printer channels.

CPU registers are used as follows:

Entry:
    AH = Printer I/O sub-function selector:
        0 - Send character to printer port.
        1 - Initialise printer port.
        2 - Return printer port status.
    DX = Logical Channel Number (0 - 2).
    Other registers as specified by sub-function.
Exit:
    AH = Printer Port Status (Bits 1 - 7):

Bit(s)	Function (Bit Set True)
7	Printer Idle.
6	Printer Acknowledge
5	Paper Out.
4	Printer Selected.
3	I/O Error.
1 & 2	Always Zero.
0	Zero if I/O successful or set if Timeout. (see sub-functions).

All flags and other registers preserved.

Three logical channels are supported. Logical printer device 0 is the system port and is standard to all machines. The power-up initialisation sequence determines if additional external printer ports are present. When both additional printer interfaces are present, device 1 is the external printer port and device 2 is the printer port on the external monochrome VDU controller. If only one additional printer interface is present it is always logical device 1.

Locations 0478h - 047Ah contain the Logical Printer Device timeout counts (see section 2.4).

Printer Sub-Function 0: Print Character.

This Sub-function attempts to output a character to the specified printer port. If the character cannot be sent within the time specified by the logical printer timout count RAM variable then the command is abandoned and AH is returned with bit 0 set.

CPU registers are used as follows:

Entry:
    AH = 0.
    AL = Character to be printed.
    DX = Logical Channel Number (0 - 2).
Exit:
    AH = Printer Port status (as given above) or Timeout (Bit 0) set.
    All flags and other registers preserved.

Printer Sub-Function 1: Initialise Printer Channel.

This Sub-function performs a complete reinitialisation of a specified printer channel (if present). The printer INIT signal is held low for approximately 4 milliseconds. Printer interrupts and auto linefeed are disabled.

CPU registers are used as follows:

Entry:
    AH = 1.
    DX = Logical Channel Number (0 - 2).
Exit:
    AH = Printer Port status (as given above) or Invalid Channel (Bit 0) set.
    All flags and other registers preserved.

Printer Sub-Function 2: Return Channel Status.

This Sub-function returns the status register of the specified logical printer channel (if present).

CPU registers are used as follows:

Entry:
    AH = 2.
    DX = Logical Channel Number (0 - 2).
Exit:
    AH = Printer Port status (as given above) or Invalid Channel (Bit 0) set.
    All flags and other registers preserved.

2.3.15 Interrupt 24: System Restart.

This software interrupt is intended to provide an orderly system restart capability. A message is displayed on the active VDU requesting that the user "Insert a SYSTEM disk into Drive A" and "Then press any key." When the keypress is received, the Disk Bootstrap process (Interrupt 25) is invoked.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    Disk Bootstrap.

2.3.16 Interrupt 25: Disk Bootstrap.

This software interrupt to provide access to the disk bootstrap process which is normally executed after power-up initialisation tests.

The ROS attempts to load the bootstrap sector (from drive A, side 0, track 0, sector 1) into memory at 07C00. If the bootstrap sector is loaded successfully it is given control (far jump to segment 0000 offset 7C00). If the bootstrap sector cannot be loaded after 10 retries, the ROS will display a message prompting the user to "Insert a SYSTEM disk into drive A" and "Then press any key." The ROS then waits for the keypress and repeats the bootstrap procedure.

CPU registers are used as follows:

Entry:
    No conditions.
Exit:
    To program loaded by Disk Bootstrap.

2.3.17 Interrupt 26: System Clock & Real Time Clock.

This software interrupt routine provides access to both the system (software maintained) clock location as well the Real Time Clock (RTC) hardware.

CPU registers are used as follows:

Entry:
    AH = Clock sub-function selector:
        0 - Get System Clock.
        1 - Set System Clock.
        2 - Get RTC time.
        3 - Set RTC time.
        4 - Get RTC date.

5 - Set RTC date.
6 - Set RTC alarm.
7 - Reset RTC alarm.

All other registers as required by sub-function.

Exit:

All registers as specified by sub-function.

Clock Sub-Function 0: Get System Clock.

This sub-function returns the current value of the 32 bit system clock value.

CPU registers are used as follows:

Entry:

AH = 0.

Exit:

DX = Least Significant Word of the clock count.
CX = Most Significant Word of the clock count.
AL = 24 Hour Flag:
0 if not past 24 hours.
1 if past 24 hours.

All flags and other registers preserved.

The 32 but system clock is incremented every 54 milliseconds by the ticker hardware interrupt routine. When the count reaches the 24 hour value (000185000h) the 24 Hour flag is set and the system clock count is reset to zero.

Note that the 24 hour flag is reset to zero after it has been read.

Clock Sub-Function 1: Set System Clock.

This sub-function sets the current value of the 32 bit system clock value.

CPU registers are used as follows:

Entry:

AH = 1.
DX = Least Significant Word of the clock count.
CX = Most Significant Word of the clock count.

Exit:

All flags and other registers preserved.

Clock Sub-Function 2: Get RTC Time.

This sub-function gets the current time from the Real Time Clock.

CPU registers are used as follows:

Entry:

AH = 2.

Exit:

If RTC not operating then
Carry True.

```
        CX DX preserved.
    If RTC operating then
        Carry False.
        CH = Hour (BCD).
        CL = Minute (BCD).
        DH = Second (BCD).
    Always
        All other flags corrupt.
        All other registers preserved.
```

Clock Sub-Function 3: Set RTC Time.

This sub-function sets the Real Time Clock time.

CPU registers are used as follows:

```
Entry:
    AH = 3.
    CH = Hour (BCD).
    CL = Minute (BCD).
    DH = Second (BCD).
    DL = 1 to enable daylight savings option (otherwise 0).
Exit:
    If RTC not operating then
        Carry True.
    If RTC operating then
        Carry False.
    Always
        All other flags corrupt.
        All other registers preserved.
```

When the daylight savings option is set it enables two special updates of the current time. On the last Sunday in April, the time increments from 1:59:59 AM to 3:00:00 AM. Also on the last Sunday in October the time increments from 1:59:59 AM to 1:00:00 AM.

Note that this option also disables the alarm function.

Clock Sub-Function 4: Get RTC Date.

This sub-function gets the current date from the Real Time Clock.

CPU registers are used as follows:

```
Entry:
    AH = 4.
Exit:
    If RTC not operating then
        Carry True.
        CX DX preserved.
    If RTC operating then
        Carry False.
        CH = Century (BCD).
        CL = Year (BCD).
        DH = Month (BCD).
```

        DL = Day Of Month (BCD).
    Always
        All other flags corrupt.
        All other registers preserved.

The century byte is set to 19 (BCD) if the year is 80 (BCD) or above otherwise it is set to 20 (BCD).

Clock Sub-Function 5: Set RTC Date.

This sub-function sets the Real Time Clock time.

CPU registers are used as follows:

```
Entry:
    AH = 5.
    CH = Century (BCD) [Ignored].
    CL = Year (BCD).
    DH = Month (BCD).
    DL = Day of Month (BCD).
Exit:
    If RTC not operating then
        Carry True.
    If RTC operating then
        Carry False.
    Always
        All other flags corrupt.
        All other registers preserved.
```

Century is ignored and is computed as described in clock sub-function 4.

Clock Sub-Function 6: Set RTC Alarm.

This sub-function sets the alarm time and arms the Real Time Clock alarm interrupt. The alarm interrupt will occur then the current time matches the alarm time. An application program which uses this function must first write the address of its alarm interrupt routine into interrupt vector 10.

CPU registers are used as follows:

```
Entry:
    AH = 6.
    CH = Hour (BCD).
    CL = Minute (BCD).
    DH = Second (BCD).
Exit:
    If RTC alarm already set then
        Carry True.
    If RTC alarm not already set then
        Carry False.
    Always
        All other flags corrupt.
        All other registers preserved.
```

Clock Sub-Function 7: Reset RTC Alarm.

This sub-function disarms the Real Time Clock alarm function.

CPU registers are used as follows:

Entry:
    AH = 7.
Exit:
    All flags and registers preserved.

2.3.18 Interrupt 27: Keyboard Break Interrupt.

This software interrupt is invoked by the keyboard hardware interrupt routine when a keyboard break ([CTRL] + [NUM LOCK]) is detected.

The power-up initialisation process loads the address of a dummy break handler routine which does an interrupt return (IRET) instruction.

Application programs which supply a keyboard break interrupt must conform to the following register conventions:

Entry:
    DS = 0040h (Spanning the ROS data).
Exit:
    All registers must be preserved except AX, BX, CX, DX, DS and Flags which may be corrupt.

The supplied interrupt routine must not invoke any other ROS interrupts from within itself but may modify any of the system RAM locations used by the ROS.

2.3.19 Interrupt 28: External Ticker Interrupt.

This software interrupt is called from within the System Clock hardware interrupt routine. It is initialised by power-up with a dummy handler which returns from interrupt by doing an IRET instruction. It can be used by application programs which require a process to be run at a regular interval.

Application programs which supply an external ticker interrupt must conform to the following register conventions:

Entry:
    DS = 0040h (Spanning the ROS data).
Exit:
    All registers must be preserved except AX, DX, DS and Flags which may be corrupt.

The supplied interrupt routine must not invoke any other ROS interrupts from within itself but may modify any of the system RAM locations used by the ROS.

2.3.20 Interrupt 29: VDU Parameter Table.

This interrupt vector location contains the 32-bit address of the VDU parameter table used in setting up the video hardware when changing VDU mode. Upon power-up or after a reset, the initialisation process loads the ROM table address into this vector location (0074-0077 hex).

The VDU parameter table consists of four consecutive 16 byte entries. Each entry contains an initialisation quantity for each of the emulated MC6845 CRTC registers (See section 1.11.5). When a new VDU mode is selected the table entry used to initialise the VDU is as follows:

Table Entry	VDU Mode
0	0 - Alpha 25 by 40 Chars.
0	1 - Alpha 25 by 40 Chars.
1	2 - Alpha 25 by 80 Chars.
1	3 - Alpha 25 by 80 Chars.
2	4 - Graphics 200 by 320 Pixels, palettes 0 or 1.
2	5 - Graphics 200 by 320 Pixels, palette 2.
2	6 - Graphics 200 by 640 Pixels.
3	3 - Alpha 25 by 80 chars using monochrom adapter.

The table contains the following initialisation data:

Register Number	Function	Entry 0	Entry 1	Entry 2	Entry 3
R0	* Horizontal Total	56	113	56	97
R1	Horizontal Displayed	[40]	[80]	[40]	80
R2	* Horizontal Sync Posn.	45	90	45	82
R3	* Horizontal Sync Width	10	10	10	15
R4	* Vertical Total	31	31	127	25
R5	* Vertical Total Adj.	06	06	06	06
R6	Vertical Displayed	[25]	[25]	[100]	25
R7	* Vertical Sync Posn.	28	28	112	25
R8	* Interlace	02	02	02	02
R9	Max. Raster Address	07	07	01	13
R10	Cursor Start Raster	06	06	06	11
R11	Cursor End Raster	07	07	07	12
R12	Start Address (H)	00	00	00	00
R13	Start Address (L)	00	00	00	00
R14	Cursor Location (H)	00	00	00	00
R15	Cursor Location (L)	00	00	00	00

Registers marked '*' are not software programmable in the emulated MC6845 CRTC implementation and their corresponding values are place holders in the table.

The two sets of values marked by '[ ]' have no effect on the display, but a zero value in either register will cause the whole screen to display the border colour.

2.3.21 Interrupt 30: Disk Parameter Table.

This interrupt vector location contains the 32-bit address of the parameter table of configuration parameters for the disk interface. Upon power-up or after a reset, the initialisation process loads the ROM table address into this vector location (0078 - 007B hex).

The Disk Parameter Table consists of 11 bytes as follows:

Byte	Function	Value
0	2nd byte of the disk controller specify command. (6 Ms Step Rate, Head Unload delay disabled.)	208
1	3rd byte of the disk controller specify command. (Head Load delay disabled.)	0
2	Motor off timeout (approx 5 seconds).	100

Byte	Function	Value
3	Sector size selector (512 bytes)	2
4	End of Track (sector 9)	9
5	Gap length for Read/Write commands.	42
6	DTL - Data Length	255
7	Gap Length for format command.	80
8	Filler byte for format command.	246
9	Head Settling Delay (15 Ms)	15
10	Motor on Delay (500 Ms)	4

2.3.22 Interrupt 31: VDU Matrix Table.

This interrupt vector location contains the 32-bit address of the VDU matrix table used in graphics modes for generating pixel data for characters 128 to 255.

Upon power-up or after a reset, the initialisation process loads this vector (007C-007F) with all zeros to indicate that no external VDU matrix table is loaded.

Each of the 128 character table entries consists of eight bytes, one for each character scan. The first byte is the top scan value and the last byte is the button scan value. The MSB, bit 7, is the left most pixel and the LSB, bit 0, is the right most pixel of the scan. A set bit displays the foreground colour and a reset bit displays the background colour.

2.4 RAM Variables.

The System RAM address space from 00300 to 00500 is used by the ROS for variable storage. The following table lists the variables and their usage. They are either classified as Byte (8-bit), Word (16-bit), Long Word (32-bit) or Buffer (greater than 32-bit) storage locations.

Location(s)	Usage
00300-003FF	Initialisation Stack (Buffer). Used as stack area only during initialisation.
00400	Logical Serial Device 0 Base I/O Address (Word). Contains the base address of logical serial device 0. Initally the System Asynchronous Serial port address.
00402	Logical Serial Device 1 Base I/O Address (Word). Contains the base address of logical serial device 1. Initally the external asynchronous serial port or zero if it is not present at initialisation.
0404 - 0407	Reserved.
00408	Logical Printer Device 0 Base I/O Address (Word). The base address of logical printer device 0. Initally the System Parallel Printer port.
0040A	Logical Printer Device 1 Base I/O Address (Word). The base address of logical printer device 0. Initially the external parallel printer port if it is present else it points to the external monochrome VDU controller if it is present. If neither is present it is initialised to zero.
0040C	Logical Printer Device 2 Base I/O Address (Word). Initially points to the external monochrome VDU controller if both the external parallel printer port and the external monochrome VDU controller are present. If either is not installed initialised to zero.
0040E	Reserved (Word).

<table>
<tr><th>Location(s)</th><th>Usage</th></tr>
<tr><td>00410</td><td>System Configuration Status (Word).
Contains the System Configuration as follows:
<table>
<tr><th>Bit(s)</th><th>Function</th></tr>
<tr><td>14 & 15</td><td>Number of printers (1-3).</td></tr>
<tr><td>13</td><td>Not used.</td></tr>
<tr><td>12</td><td>Set if an optional games adapter is fitted.</td></tr>
<tr><td>11</td><td>Always zero.</td></tr>
<tr><td>9 & 10</td><td>Number of serial interfaces (1 or 2).</td></tr>
<tr><td>8</td><td>Not used.</td></tr>
<tr><td>7</td><td>Always zero.</td></tr>
<tr><td>6</td><td>Set if second floppy disk drive is fitted.</td></tr>
<tr><td>4 & 5</td><td>Default VDU mode.</td></tr>
<tr><td>2 & 3</td><td>Always set.</td></tr>
<tr><td>1</td><td>Set if 8087 NDP is installed.</td></tr>
<tr><td>0</td><td>Always set.</td></tr>
</table></td></tr>
<tr><td>00412</td><td>Reserved (Byte).</td></tr>
<tr><td>00413</td><td>Total RAM Size (Word).
Initially set to the number of 1K User (System) RAM Blocks installed.</td></tr>
<tr><td>00415</td><td>Extra RAM Size (Word).
Initially set to the number of 1K User (System) RAM Blocks installed minus 64</td></tr>
<tr><td>00417</td><td>Key Toggles and Key States (Byte).
This byte is used to record the state of the Key Toggles (bits 4-7) and Key States (bits 0-3) as follows:
<table>
<tr><th>Bit</th><th>Key (Bit set if active)</th></tr>
<tr><td>7</td><td>INS</td></tr>
<tr><td>6</td><td>CAPS LOCK</td></tr>
<tr><td>5</td><td>NUM LOCK</td></tr>
<tr><td>4</td><td>SCROLL LOCK</td></tr>
<tr><td>3</td><td>ALT</td></tr>
<tr><td>2</td><td>CTRL</td></tr>
<tr><td>1</td><td>LEFT SHIFT</td></tr>
<tr><td>0</td><td>RIGHT SHIFT</td></tr>
</table></td></tr>
<tr><td>00418</td><td>Keys down (Byte).
This byte is used to record the state of the toggle keys so that they do not repeat when the key is held down.
<table>
<tr><th>Bit</th><th>Key (Set if down)</th></tr>
<tr><td>7</td><td>INS</td></tr>
<tr><td>6</td><td>CAPS LOCK</td></tr>
<tr><td>5</td><td>NUM LOCK</td></tr>
<tr><td>4</td><td>SCROLL LOCK</td></tr>
</table></td></tr>
<tr><td>00419</td><td>Absolute Key Token Number (Byte).
When an absolute key token numbered is entered via ALT and the numeric key pad, this variable holds the current state of the token.</td></tr>
<tr><td>0041A</td><td>Key Token Buffer Out Pointer (Word).
This variable holds the absolute offset to the next key token to be removed from the key token buffer.
Note that the ROS assumes that the buffer has a segment paragraph address of 0040h.</td></tr>
</table>

<table>
<tr><th>Location(s)</th><th>Usage</th></tr>
<tr><td>0041C</td><td>Key Token Buffer In Pointer (Word).
This variable holds the absolute offset to the next empty position in the key token buffer. The buffer is empty when this location is the same as the Out Pointer.</td></tr>
<tr><td>0041E</td><td>Key Token Buffer (Buffer).
The Key Token Buffer is a 16 word circular buffer used to store up to 16 key tokens.</td></tr>
<tr><td>0043E</td><td>Drive Restore Flag (Byte).
Each floppy disk drive has a restore flag associated with it (bit 0 for drive 0 and bit 1 for drive 1).
If the restore flag for the specified drive is reset prior to any disk access (read/write /verify/format), then the restore command is issued to the FDC for that drive. If successful then the associated flag bit is set. When the initialise sub-function of the disk interrupt is called the restore flag is cleared.
Bit 7 is used for handling FDC hardware interrupts.</td></tr>
<tr><td>0043F</td><td>Drive Motor Flag (Byte).
When a disk drive motor is running then either bit 0 or bit 1 will be set to which drive (0 or 1 respectively) is selected.</td></tr>
<tr><td>00440</td><td>Drive Motor Timeout Counter (Byte).
After each disk operation the the motor off timeout count is copied from the Disk Parameter table (See interrupt 30) into this variable. Each time the system clock interrupt is executed, the count is decremented. When it reaches zero the Drive Motor Flag is reset.</td></tr>
<tr><td>00441</td><td>Disk Status (Byte).
This byte holds the status returned by the last disk operation. (See section 2.3.19 Disk I/O Interrupt - Sub-Function 1.)</td></tr>
<tr><td>00442</td><td>FDC Results Buffer (Buffer).
This seven byte buffer is used for storage of the FDC status information returned upon the completion of a disk I/O operation.</td></tr>
<tr><td>00449</td><td>Current VDU Mode (Byte).
The current VDU mode in the range of 0 - 7 is stored here.</td></tr>
<tr><td>0044A</td><td>Visible VDU Columns (Word).
The number of visible character columns currently being displayed is stored here.</td></tr>
<tr><td>0044C</td><td>VDU Display Buffer Size (Word).
This word holds the amount of VDU RAM used by the ROS to display one page as defined below:
<table>
<tr><th>Mode(s)</th><th>Size</th></tr>
<tr><td>0 &amp; 1</td><td>2048</td></tr>
<tr><td>2 &amp; 3</td><td>4096</td></tr>
<tr><td>4 - 6</td><td>16384</td></tr>
<tr><td>7</td><td>4096</td></tr>
</table></td></tr>
<tr><td>0044E</td><td>VDU Display Start Address (Word).
Contains the origin of the currently active VDU display page.</td></tr>
<tr><td>00450</td><td>Cursor Address Buffer (Buffer)
This 16 byte buffer contains the row and column addresses for up to eight display pages.</td></tr>
<tr><td>00460</td><td>Cursor End Scan (Byte).
This byte contains the current end scan number that was programmed into the VDU controller.</td></tr>
<tr><td>00461</td><td>Cursor Start Scan (Byte).
This byte contains the current start scan number that was programmed into the VDU controller.</td></tr>
</table>

Location(s)	Usage
00462	VDU Active Display Page (Byte). This byte contains the selected display page number.
00463	VDU I/O Address (Word). This word contains the I/O address of the VDU interface currently in use. For all modes except mode 7 this is the internal emulated MC6845 CRTC device. For mode 7 it is the external Monochrome VDU controller.
00465	Current VDU Mode Control Register (Byte). This byte contains the current contents of the VDU Mode Control Register (See 1.11.3.1).
00466	Current VDU Colour Select Register (Byte). This byte contains a copy of the data loaded into VDU colour select register.
0467-046B	Reserved
0046C	System Clock (Long Word). The 32 bit system clock count
00470	24 Hour Flag (Byte). When the system clock reaches 000185000h then it is cleared and this flag byte is set to 0FFh. Note that reading the clock via interrupt 26 clears this flag.
00471	Break (Byte). This byte is initially set to zero. Each time Break ([CTRL]+[NUM LOCK]) is detected, bit 7 is set. An application program using this bit to detect break must reset bit 7 when it detects the break event.
00472	System Reset Flag (Word). When system reset is ([CTRL]+[ALT]+[DEL]) detected this location is set to 01234h prior to issuing a system reset. The power-up self test routine then recognizes this pattern and does not repeat the RAM tests.
0474-0477	Reserved for Hard Disk BIOS ROM.
00478	Logical Printer Device 0 Timeout Count (Byte). (See box below)
00479	Logical Printer Device 1 Timeout Count (Byte). (See box below)
0047A	Logical Printer Device 2 Timeout Count (Byte). (See box below)
	Locations 00478 - 0047A specify how long the ROS should wait in half second mulitples, while trying to output a character to a logical printer channel. The are iniitally set to 20 (10 Second timeout).
0047B	Reserved.
0047C	Logical Serial Device 0 Timeout Count (Byte). (See box below)
0047D	Logical Serial Device 1 Timeout Count (Byte). (See box below)
	Locations 0047C & 0047D specify the length of the wait time in half second intervals for character I/O to a particular logical serial channel. All counts are set to 1 (for a $^{1}/_{2}$ second timeout).
0047E	Reserved.
00480	Key Token Buffer Start Address (Word). Offset pointer to the start of the key token buffer. Note that the assumed buffer segment paragraph address is 0040h.
00482	Key Token Buffer End Address (Word). Offset pointer to the start of the key token buffer.
00500	Print Screen Status (Byte).

Value	Meaning
0	Print Screen completed OK.
1	Print Screen in progress.
255	Print Screen abandoned due to timeout.

2.4 Non-Volatile RAM (NVR)

The first 40 bytes of the battery backed RAM within the RTC hardware are for system parameter storage as follows:

Byte(s)	Usage	Default
0-9	Time and Date parameters.	--
1	RTC Control Register A.	070
11	RTC Control Register B.	002
12	RTC Control Register C.	--
13	RTC Control Register D.	--
14 - 19	Time and Date when machine last used.	--
20	User RAM Checksum.	
21 - 22	Enter Key translation token.	1C0D
23 - 24	Forward Delete Key translation token.	2207
25 - 26	Joystick Fire Button 1 translation token.	FFFF
27 - 28	Joystick Fire Button 2 translation token.	FFFF
29 - 30	Mouse Button 1 translation token.	FFFF
31 - 32	Mouse Button 2 translation token.	FFFF
33	Mouse X direction scaling factor.	00A
34	Mouse Y direction scaling factor.	00A
35	Initial VDU mode and drive count	020
36	Initial VDU Character attributes.	007
37	Size of RAM disk in 2K blocks.	000
38	Initial system UART hardware setup byte.	0E3
39	Initial external UART hardware setup byte.	0E3
40-63	Unused	--

After power-up or upon system reset the NVR is checksummed as part of initialisation. If the lower byte of the sum is not 0AAh or if the battery voltage low bit is set in the RTC status register, then the values in the default column are loaded into their respective locations and a warning message is displayed on the VDU. Those locations without defaults (marked with '--') are not changed.

The default key token value in bytes 25 to 32 is a special value (FFFF) which signals the keyboard hardware interrupt to ignore the key press rather than to insert the key token into the buffer.

The initial VDU mode (byte 35) is used to set up the system status-1 channel. (DDM - bits 4 & 5. See 1.8.2 for the valid combinations.) Bits 4 and 5 of byte 35 are set up correspondingly. Bit 6 is set if two drives are fitted else it is cleared. The default version of byte 35 has bit 6 set (two drives) and bits 4 & 5 set to 1 & 0 (Colour, alpha, 80 x 25 chars).

Bit 7 of byte 35 is used to enable or disable the serial I/O flow control option. Refer to section 2.3.11 for serial I/O and flow control details.

The initial VDU character attribute (byte 36) is written to all the attribute bytes of the VDU buffer when one of the alpha modes is selected. The default value selects a white foreground on a black background.

The RAM disk size (byte 37) is used by the MS-DOS and DOS Plus operating systems to specify their RAM disk setup size.

The initial UART parameters (bytes 38 and 39) specifies 9600 baud, 8 data bits, 1 stop bit and no parity.

These values are loaded to their respective serial channel by the Serial I/O Initialise sub-function (See Section 2.3.11: Interrupt 20).

2.6 ROS Messages

The ROS outputs a number of messages during initialisation and self test as detailed below. The language in which these messages are displayed is dependent of the three option links connected to the three least significant bits of the system printer port status. (See Table 3.1 for the interpretation of the three link bits.)

2.6.1 Non-Fatal ROS Messages

The following messages are displayed on the VDU (in the default display mode as specified by the NVR) in the situations as described. The initialisation process is allowed to complete even though some of them may represent self test failures.

Please wait

This message is displayed on the top line of the screen after Power-Up or after a System Reset ([CTRL]+[ALT]+[DEL]) from the keyboard. A dot is displayed after it for each major hardware self test segment completed successfully.

Amstrad PC nnnK (Vv.i) Last used at hh:mm on dd mn yy

This message is displayed after the successful completion of all self tests, where:

nnn = the RAM size in kilobytes.
v.i = the ROS Version (v) and Issue (i) number.
hh:mm = the hours (hh) and the minutes (mm) of last on time.
dd mn yy = the day (dd), the calendar month (mn) and the year (yy) of the last date used.

Please fit new batteries

This message id displayed below the AMSTRAD PC message when it is noted that the RTC battery voltage low bit (VRT) is set (indicating that there is either no battery installed or that the battery is very nearly flat).

Check keyboard and mouse

This message is displayed when the keyboard self test firmware does not respond with the test pass (0AAh).

Insert a SYSTEM disk into drive A
Then press any key

This message set is displayed when the floppy disk bootstrap is unable to successfully read the bootstrap sector from drive A after 10 retries.

Error: External ROM checksum incorrect: ROM address = nnnnnh

This message is displayed when the checksum on an external ROM is not zero (See section 2.1 - 16). The physical address of the ROM is displayed in five (nnnnn) hexadecimal digits.

2.6.2 Fatal ROS Messages

The following messages indicate that a self test segment has failed and that initialisation cannot continue. In this situation the machine must be switched off an on again in order to reinitiate operations. The VDI us switched to 80 column alpha mode and cleared prior to displaying any of these messages.

Error: Faulty SYSTEM RAM
Error: Faulty VDU RAM
Error: Faulty interrupt controller
Error: Faulty direct memory access controller

Error: Faulty floppy disk controller or disk drive
Error: Faulty interval timer
Error: Faulty system status register
Error: Faulty real time clock
Error: Faulty VDU controller
Error: Faulty system printer port
Error: Faulty system serial port
Error: Faulty mouse coordinate register
Error: Faulty ROS ROM checksum
Error: Faulty memory (parity error)

As is evident from the message content, the faulty system component is named. When one of these failures occurs, no other testing is run since further testing may require use of the failing component. For this reason the system is placed in a non-interruptible loop. Failures of this sort are not expected to occur even intermittently. When any self test failure does occur it should be referred to a qualified AMSTRAD service facility for further diagnostic testing.

Section 1 Index Section 3

SECTION 3 - TABLES

3 Reference Information

The following tables document a number of the hardware and software features of the AMSTRAD PC1512 some of which may have been already mentioned in earlier sections but are repeated here for easy reference.

3.1 Language Links.

The lower three bits of the Printer Status Channel (I/O address 379) are wired to reflect the (one's complement) state of a set of option links (LK1 - LK3) located on the left side of the main board about 2 inches below the printer connector. They are used by the ROS firmware to define the language option or diagnostic mode option as detailed below.

Link Value	ROS Usage
0	English Language.
1	German Language.
2	French Language.
3	Spanish Language.
4	Danish Language.
5	Swedish Language.
6	Italian Language.
7	Diagnostic Mode.

The ROS messages are displayed in the selected languge. In diagnostic mode, the messages revert to English, and the normal testing is skipped. Any self test failures are reported but are ignored and upon completion a disk bootstrap is attempted. This enables loading of an extended set of diagnostic software.

3.2 Processor Memory Usage.

The following is a repeat of the processor's physical memory layout in tabular form with interrupts and ROS areas included.

Location(s)	Usage
00000 - 003FF	Processor interrupt vectors 0 to 255. To derive an individual interrupt vector's starting address multiply the vector number by four.
00400 - 00500	ROS Variables. (See section 2.4)
00501 - 7FFFF	System (or User) RAM artea. The 512K byte area (inclusive of the previous entries) is the normal complement of system memory installed.
80000 - BFFFF	128K Byte area where additional memory may be installed in external 32K byte increments.
A0000 - BFFFF	128K byte area reserved for 8-bit controller memory. The standard VDU screen memory resides in the address range from B8000 to BBFFF. Individual peripheral

Location(s)	Usage
	cards may make use of the other addresses as they see fit. For additional details, consult the manual supplied the card or its manufacturer's agent.
C0000 - EFFFF	192K byte area reserved for the expansion ROMs of the various peripheral cards. The optional Hard Disk controller uses the address range from C8000 to CBFFF.
E0000 - FFFFF	64K byte area reserved for System ROM. The ROS resides in the 16K byte area from FC000 to FFFFF. The remaining 48K bytes is reserved for future expansion. Address block occurs repeat from 16K byte ROS area occurs in this range.

3.3 Keyboard and Key Codes.

Key Code	Hex	(UK) Key Cap
1	01	ESC
2	02	1 and !
3	03	2 and "
4	04	3 and £
5	05	4 and $
6	06	5 and %
7	07	6 and ^
8	08	7 and &
9	09	8 and *
10	0A	9 and (
11	0B	0 and)
12	0C	- and _
13	0D	= and +
14	0E	<-DEL
15	0F	TAB
16	10	Q
17	11	W
18	12	E
19	13	R
20	14	T
21	15	Y
22	16	U
23	17	I
24	18	O
25	19	P
26	1A	[and {
27	1B	] and }
28	1C	CR (↵)
29	1D	CTRL
30	1E	A
31	1F	S
32	20	D
33	21	F
34	22	G
35	23	H
36	24	J

Key Code	Hex	(UK) Key Cap
37	25	K
38	26	L
39	27	; and :
40	28	' and @
41	29	# and ~
42	2A	LEFT SHIFT
43	2B	\ and |
44	2C	Z
45	2D	X
46	2E	C
47	2F	V
48	30	B
49	31	N
50	32	M
51	33	, and <
52	34	. and >
53	35	/ and ?
54	36	RIGHT SHIFT
55	37	* and PRTSC
56	38	ALT
57	39	SPACE
58	3A	CAPS LOCK
59	3B	F1
60	3C	F2
61	3D	F3
62	3E	F4
63	3F	F5
64	40	F6
65	41	F7
66	42	F8
67	43	F9
68	44	F10
69	45	NUM LOCK
70	46	SCROLL LOCK
71	47	KEY PAD 7
72	48	KEY PAD 8
73	49	KEY PAD 9
74	4A	KEY PAD -
75	4B	KEY PAD 4
76	4C	KEY PAD 5
77	4D	KEY PAD 6
78	4E	KEY PAD +
79	4F	KEY PAD 1
80	50	KEY PAD 2
81	51	KEY PAD 3
82	52	KEY PAD 0
83	53	KEY PAD .

Key Code	Hex	(UK) Key Cap
84 - 111	54 - 6F	UNDEFINED
112	70	DEL ->
113 - 115	71 - 73	UNDEFINED
116	74	ENTER
117 - 118	75 - 76	UNDEFINED
119	77	JOY FIRE2
120	78	JOY FIRE1
121	79	JOY RIGHT
122	7A	JOY LEFT
123	7B	JOY DOWN
124	7C	JOY UP
125	7D	MOUSE M2
126	7E	MOUSE M1
127	7F	UNDEFINED

3.4 Asynchronous Communications Element (8250) Registers.

For serious design purposes, it is recommended that the designer obtain the standard INS8250 data sheets. The following excerpt are the major software accessible registers.

Modem Status Register (MSR) [R6] - I/O Address 3FEh.

Bit(s)	Function
7	Data Carrier Detect (DCD).
6	Ring Indicator (RI).
5	Data Set Ready (DSR).
4	Clear To Send (CTS).
3	Delta Data Carrier Detect (DDCD).
2	Trailing Edge Ring Indicator (TREI).
1	Delta Data Set Ready (DDSR).
0	Delta Clear To Send (DCTS).

Line Status Register (LSR) [R5] - I/O Address 3FDh.

Bit(s)	Function
7	Always Clear (0).
6	Transmitter Shift Register Empty (TSRE).
5	Transmitter Holding Register Empty (THRE).
4	Break Interrupt (BI).
3	Framing Error (FE).
2	Parity Error (PE).
1	Overrun Error (OE).
0	Data Ready (DR).

Modem Control Register (MCR) [R4] - I/O Address 3FCh.

Bit(s)	Function
7	Always Clear (0).

Bit(s)	Function
6	Always Clear (0).
5	Always Clear (0).
4	Loop (Diagnostic Mode).
3	Out2 (Looped to RI).
2	Out1 (Looped to DCD).
1	Request to Send (RTS) (Looped to DSR).
0	Data Terminal Ready (DTR) (Looped to CTS).

Line Control Register (MCR) [R3] - I/O Address 3FBh.

Bit(s)	Function
7	Divisor Latch Access (DLAB) (Selects Regs 0 & 1).
6	Set Break.
5	Stick Parity (Holds parity as EPS not if PEN set).
4	Even parity Select (EPS).
3	Parity Enable (PEN).
2	Number of Stop Bits (STB) (0=1 Stop Bit, 1= >1).
1	Word Length Select Bit 1 (WLS1). (0-3 = 5-8 Bits)
0	Word Length Select Bit 0 (WLS0).

Interrupt Identification Register (IIR) [R2] - I/O Address 3FAh.

Bit(s)	Function
7	Always Clear (0).
6	Always Clear (0).
5	Always Clear (0).
4	Always Clear (0).
3	Always Clear (0).
2	Interrupt ID Bit 1 (IID1).
1	Interrupt ID Bit 0 (IID0).
0	Not Interrupt Pending.

IID	Int Type
3	Rx Line Status
2	Rx Data Avail.
1	Tx H.Reg Empty
0	MODEM Status

Interrupt Enable Register (IER) [DLAB = 0:R1] - I/O Address 3F9h.

When the Divisor Access Latch Bit (Line Control Register bit 7: DLAB) is clear, inputting I/O address 3F9 reads the IER.

Bit(s)	Function
7	Always Clear (0).
6	Always Clear (0).
5	Always Clear (0).
4	Always Clear (0).
3	Modem Status (EDSSI).
2	Receiver Line Status (ELSI).
1	Transmitter Holding Register Empty (ETBEI).
0	Received Data Available (ERBAI).

Receive Buffer Register (RBR)
Transmit Holding Register (THR) [DLAB = 0:R0] - I/O Address 3F8h.

When the Divisor Access Latch Bit (Line Control Register bit 7: DLAB) is clear, reading and writing I/O location 3F8 accesses the RBR/THR registers. An input from I/O address 3F8 reads the Receiver buffer Register (bits 0 to 7). Outputting to I/O address 3F8 writes the Transmitter holding Register.

Divisor Latches MS & LS (DLL & DLM) [R0 & R1 when DLAB Set].

When the Divisor Access Latch Bit (Line Control Register bit 7: DLAB) is set, then registers 0 & 1 are the (16-bit) Divisor Register. The least significant bits are written to by outputting to address 3F8 and the most significant bits are written to by an output to location 3F9. The divisors and their respective baud rates are as follows.

Baud Rate	Divisor	R1 & R0 (hex)
75	1536	06 - 00
300	384	01 - 80
600	192	00 - C0
1200	96	00 - 60
2400	48	00 - 30
4800	24	00 - 18
9600	12	00 - 0C

3.5 High Performance Programmable DMA Controller (8237A-4) Registers.

The following are the major software accessible 8237A registers.

Command Register - Write I/O Address 008.

Bit(s)	Function (Action ... { 1 / 0 })
7	DACK sense active { hi / lo }.
6	DREQ sense active { hi / lo }.
5	{Extended/Late} write selection.
4	{Rotating/Fixed} priority.
3	{Compressed/Normal} timing.
2	{Disable/Enable} Controller.
1	{Enable/Disable} Channel 0 address hold.
0	{Enable/Disable} Memory-to-memory (not supported).

Status Register - Read I/O Address 008.

Bit(s)	Function
7	Channel 3 Request.
6	Channel 2 Request.
5	Channel 1 Request.
4	Channel 0 Request.
3	Channel 3 has reached TC.
2	Channel 2 has reached TC.
1	Channel 1 has reached TC.
0	Channel 0 has reached TC.

Mode Register - I/O Address 00B [WO].

Bit(s)	Function
7	Mode Select Bit 1. (Modes: 0 = Demand, 1 = Single,
6	Mode Select Bit 0. 2 = Block, 3 = Cascade)
5	Address {decrement/increment} select.
4	Autoinitialisation {enable/disable}.
3	Transfer Type Bit 1. (Modes: 0 = Verify, 1 = Write,
2	Transfer Type Bit 0. 2 = Read, 3 = Illegal)
1	Channel Select Bit 1. (Channels: 0-3 respectively)
0	Channel Select Bit 0.

Request Register - I/O Address 009h [WO].

Bit(s)	Function
7	Don't Care.
6	Don't Care.
5	Don't Care.
4	Don't Care.
3	Don't Care.
2	Request Bit {Set/Reset}.
1	Channel Select Bit 1. (Channels: 0-3 respectively)
0	Channel Select Bit 0.

Mask Set/Reset Register - I/O Address 00Ah [WO].

Bit(s)	Function
7	Don't Care.
6	Don't Care.
5	Don't Care.
4	Don't Care.
3	Don't Care.
2	{Set/Reset} Mask Bit.
1	Channel Select Bit 1. (Channels: 0-3 respectively)
0	Channel Select Bit 0.

Mask Write Register - I/O Address 00F [WO].

Bit(s)	Function
7	Don't Care.
6	Don't Care.
5	Don't Care.
4	Don't Care.
3	{Set/Clear} Channel 3 Mask Bit.
2	{Set/Clear} Channel 2 Mask Bit.
1	{Set/Clear} Channel 1 Mask Bit.
0	{Set/Clear} Channel 0 Mask Bit.

3.6 Programmable Interrupt Controller (8259A-2) Command Words.

The Initialisation Command Word (ICW) sequence is as follows:

Initialisation Command Word 1 (ICW1) - Write I/O Address 020.

Bit(s)	Function (Action ... { 1/0 })
7	N/A.
6	N/A.
5	N/A.
4	Always Set (1)
3	{Level/Edge} Trigger Mode.
2	Call Address Interval of {4/8}.
1	{Single/Cascade} Mode (Need ICW3 if Single Mode).
0	ICW4 {Needed/Not Needed}.

Initialisation Command Word 2 (ICW2) - Write I/O Address 021h.

Bit(s)	Function (Action ... { 1/0 })
7	Interrupt Type Bit 7 (T7).
6	Interrupt Type Bit 6 (T6).
5	Interrupt Type Bit 5 (T5).
4	Interrupt Type Bit 4 (T4).
3	Interrupt Type Bit 3 (T3).
2	Not used.
1	Not used.
0	Not used.

This byte selects one of the interrupt service vector locations (in absolute locations 0 through 3FF) to be used when interrupting. Type bits 3 - 7 (asserted on the data bus during the INTA cycle) map to address bits 5 - 9 for interrupt vector selection. The lower three type bits are derived from the interrupt level.

Initialisation Command Word 3 (ICW3) - Write I/O Address 021.

This command word is not used since Single (ICW1 bit 1) is always true in the PC1512. When used, this command word specifies which IR has a slave in Master mode, or it a slave then bits 0 through 3 specify the slave ID number (0 to 7).

Bit(s)	Function (Action ... { 1/0 })
7	Always clear (0).
6	Always clear (0).
5	Always clear (0).
4	{Enable/Disable} Special Fully Nested Mode.
3	Buffered Mode {On/Off}.
2	{Master/Slave} Mode (Only valid in Buffered Mode).
1	{Auto/Normal} EOI.
0	Always set (1) - (8086/8088 Mode).

Operation Control Words

The operation control words select various 8259A modes of operation.

Operation Control Word 1 (OCW1) - Write I/O Address 021.

Bit(s)	Function (Action ... { 1/0 })
7	Interrupt Mask 7 {Set/Reset}.
6	Interrupt Mask 6 {Set/Reset}.
5	Interrupt Mask 5 {Set/Reset}.
4	Interrupt Mask 4 {Set/Reset}.
3	Interrupt Mask 3 {Set/Reset}.
2	Interrupt Mask 2 {Set/Reset}.
1	Interrupt Mask 1 {Set/Reset}.
0	Interrupt Mask 0 {Set/Reset}.

The eight mask bits either mask (i.e. inhibit when M=1) or enable their respective channels.

Operation Control Word 2 (OCW2) - Write I/O Address 020.

Bit(s)	Function
7	Rotate (R) Bit.
6	Specific (SL) Bit.
5	End of Interrupt (EOI) bit.
4	Always zero.
3	Always zero.
2	Level bit 2 (L2).
1	Level bit 1 (L1).
0	Level bit 0 (L0).

The level bits are required when specific (SL) is set.

Operation Control Word 2 (OCW2) - Write I/O Address 020.

Bit(s)	Function (Action ... { 1/0 })
7	Always zero.
6	Enable Special Mask Mode (ESMM) bit.
5	Special Mask Mode (SMM) {Set/Reset}.
4	Always zero.
3	Always set.
2	{Enable/Disable} Poll Command.
1	Read Register (RR) enable bit.
0	Read {IS/IR} register on next -RD pulse (RIS).

The ESMM bit must be set for the SMM bit to have any effect. Similarly the RR bit must be set for the RIS bit to have an effect.

3.7 Programmable Interval Timer (8253) Registers.

The 8253 PIT has four addressable elements, the three counters (0 - 2) which are read or written 8 bits at a time (on I/O addresses 40 - 42) and the Control Word register (write I/O address 043).

Bit(s)	Function (Action ... { 1/0 })
7	Select Counter bit 1 (SC1).

Bit(s)	Function (Action ... { 1/0 })
6	Select Counter bit 0 (SC0).
5	Read/Load bit 1 (RL1).
4	Read/Load bit 0 (RL0).
3	Mode bit 2 (M2).
2	Mode bit 1 (M1).
1	Mode bit 0 (M0).
0	{Enable/Disable} Binary Coded Decimal (BCD) counter.

The SC bits select counters 0-2 and the 3 (both bits set) state is illegal.

The RL bits enable the counter's Read/Load operation as follows:

0:
Counter Latching - Snapshot current counter (to a holding register) for next read operation.

1:
Read/Load MS byte only.

2:
Read/Load LS byte only.

3:
Read/Load LS byte first then the MS byte.

The Mode bits select one of five valid modes (six & seven wrap around to modes two and three). The modes are as follows:

0:
Interrupt on Terminal count.

1:
Programmable One-Shot.

2:
Rate Generator.

3:
Square Wave generator..

4:
Software Triggered Strobe.

5:
Hardware triggered Strobe.

3.8 Real Time Clock (HD146818) Registers.

The HD146818 is a CMOS peripheral device which combines three unique features: a complete time-of-day clock with an alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM.

The figure below shows the address map of the HD146818. The memory consists of 50 bytes of general purpose RAM, 10 RAM bytes which normally contain the time, calendar, and alarm data, and four control and status bytes. All bytes are directly readable readable and writable by the processor except Registers C and D which are read only. Bit 7 of Register A and the seconds byte are also read only.

0	Seconds	00
1	Sec Alarm	01
2	Minutes	02

3	Min Alarm	03
4	Hours	04
5	Hr Alarm	05
6	Day of Wk	06
7	Day of Mo	07
8	Month	08
9	Year	09
10	Register A	0A
11	Register B	0B
12	Register C	0C
13	Register D	0D
14		0E
	50 Bytes User RAM	
63		3F

Time, Calendar and Alarm Locations

The processor obtains time and calendar information by reading the appropriate locations. The program may initialise the time, calendar and alarm by writing these locations. The contents of the 10 time, calendar and alarm bytes may either be binary or binary-coded decimal (BCD).

Before initialising the internal registers the SET bit in register B should be set to a "1" to prevent time/calendar updates from occurring. The program initialises the 10 locations in the selected format (binary or BCD), then indicates the format in the data mode (DM) bit of register B. All 10 locations must use the fame data mode, either binary or BCD. The SET bit may now be cleared to allow updates. Once initialised the real-time clock makes all updates in the selected data mode. The data mode cannot be changed without reinitialising the 10 data bytes.

The table below shows the binary and BCD formats of the time, calendar and alarm locations.

Address	Function	Range	Binary Data Mode	BCD Data Mode
0	Seconds	0-59	00h-3Bh	00h-59h
1	Sec Alarm	0-59	00h-3Bh	00h-59h
2	Minutes	0-59	00h-3Bh	00h-59h
3	Min Alarm	0-59	00h-3Bh	00h-59h
4	Hours 12-Hr Mode	1-12	01h-0Ch (AM) 81h-8Ch (PM)	01h-12h (AM) 81h-92h (PM)
	24-Hr Mode	0-23	00h-17h	00h-23h
5	Hrs Alarm 12-Hr Mode	1-12	01h-0Ch (AM) 81h-8Ch (PM)	01h-12h (AM) 81h-92h (PM)
	24-Hr Mode	0-23	00h-17h	00h-23h
6	Day of Wk	1-7	01h-07h	01h-07h
7	Day of Mon	1-31	01h-1Fh	01h-31h
8	Month	1-12	01h-0Ch	01h-12h
9	Year	0-99	00h-63h	00h-99h

For the Day of the Week, Sunday = 1.

The 24/12 bit in register B establishes whether the hour locations represent 1-to-12 or 0-to-23. The 24/12 bit cannot be changed without reinitialising the hour locations. When the 12-hour format is selected the high-order bit of the hours represents PM when it is a "1". The time, calendar and alarm bytes are not always accessible by the processor. Once per second the 10 bytes are switched to the update logic to be advanced by one second and to check for an alarm condition. If any of the 10 locations are read at this time, the data outputs are undefined. The update-in-progress (UIP) bit in Register A may be used to determine if the update cycle is in progress or not. The UIP bit goes high once a second and the update cycle begins 244 uS later. Therefore, if a "0" is read on the UIP bit, the user has at least 244 uS before the time/calendar data will be changed.

RTC Register Locations

The HD 146818 has four registers which are accessible to the processor. The four registers are fully accessible during the update cycle.

The bit assignments for Register A (address 0Ah) are as follows:

Bit	Assignment
7	Update In Progress (UIP)
6	Divider Bit 2 (DV2)
5	Divider Bit 1 (DV1)
4	Divider Bit 0 (DV0)
3	Rate Selection Bit 3 (RS3)
2	Rate Selection Bit 2 (RS2)
1	Rate Selection Bit 1 (RS1)
0	Rate Selection Bit 0 (RS0)

The UIP bit indicates whether the 10 time, calendar and alarm bytes are being updated or not as explained above.

The three Divider bits (DV2-DV0) are used to identify which of the three time base frequencies is in use or to reset the divider chain.

The four rate selection bits (RS3-RS0) select one of 15 taps on the 22-stage divider chain, or disable the divider output. The tap selected may be used to generate an output on the square (SQW) pin and/or a periodic interrupt.

The bit assignments for Register B (address 0Bh) are as follows:

Bit	Assignment
7	SET Bit
6	Periodic Interrupt Enable (PIE) Bit
5	Alarm Interrupt Enable (AIE) Bit
4	Update-ended Interrupt Enable (UIE) Bit
3	Square-Wave Enable (SQWE) Bit
2	Data Mode (DM) Bit
1	12/12 hour format Bit
0	Daylight Savings Enable (DSE) Bit

When the SET bit is a "0" the update cycle functions normally by advancing the counts once per second. When the SET bit is written to a "1", any update cycle in progress is aborted and the processor may initialise the time and calendar locations without updates occurring. SET is a read/write bit which is not modified by $\overline{\text{RES}}$ or internal functions of the HD146818.

The PIE bit is a read/write bit which allows the periodic-interrupt (PF) bit to cause the $\overline{\text{IRQ}}$ pin to be driven low. The program writes a "1" to the PIE bit in order to receive periodic interrupts at the rate specified by the RS3 RS0 bits in Register A. A "0" in PIE blocks 'IRQ from being generated, bu the periodic flag (PF) bit still goes high at the periodic rate.

The AIE bit is a read/write bit which when set to "1" permits the alarm flag (AF) to assert $\overline{\text{IRQ}}$. An alarm interrupt occurs for each second that the three time bytes equal the three alarm bytes. When AIE is a "0" the AF bit does not initiate an $\overline{\text{IRQ}}$. The $\overline{\text{RES}}$ pin clears AIE to "0". The internal functions do not affect the AIE bit.

The UIE bit is a read/write bit which enables the update-end flag (UF) bit to assert $\overline{\text{IRQ}}$. The $\overline{\text{RES}}$ pin going low or the SET bit going high clears the UIE bit.

When the SQWE bit is set to a "1" by the processor, a square-wave signal at the frequency specified by the rate selection bits (RS3 to RS0) appears on the SQW pin. When the SQWE bit is set to "0" the SQW pin is held low. The SQWE bit is cleared by the $\overline{\text{RES}}$ pin. SQWE is a read/write bit.

The DM bit indicates whether time and calendar updates are to use binary or BCD format. DM is a read/write bit and is not modified by $\overline{\text{RES}}$ or internal functions of the HD146818. A "1" in DM signifies binary data and a "0" specifies BCD data mode.

The 24/12 control bit specifies the format of the hour bytes. A "1" specifies 24-hour mode and a "0" specifies 12-hour mode. It is a read/write bit and is not affected by $\overline{\text{RES}}$ or any HD146818 internal functions.

The DSE bit is a read/write bit which when set to "1" enables daylight savings mode. When enabled, two special updates take place. On the last sunday in April the time increments from 1:59:59 to 3:00:00 AM. On the last sunday in October when the time first reaches 1:59:59 AM is decremented to 1:00:00 AM. DSE is not changed by $\overline{\text{RES}}$ or any internal operations.

The bit assignments for Register C (address 0Ch) are as follows:

Bit	Assignment
7	Interrupt Request Flag (IRQF) Bit
6	Periodic Interrupt Flag (PF) Bit
5	Alarm Interrupt Flag (AF) Bit
4	Update-Ended Interrupt Flag (UF) Bit
3	0
2	0
1	0
0	0

The C register is a read-only register and a program write has no effect any of the bits.

The IRQF bit is set by the logical equation: IRQF = PF·PIE + AF·AIE + UF·UIE. Any time the IRQF bit is a "1", the $\overline{\text{IRQ}}$ pin is driven low. All flag bits in the C register are cleared after a program read or when the $\overline{\text{RES}}$ pin is low.

The PF bit is set to "1" when a particular edge is detected in the selected tap of the divider chain as selected by the RS3 to RS0 bits. The PF bit is set to a "1" independent of the state the PIE bit.

The AF bit is set to a "1" when the current time matches the alarm time.

The UF bit is set after each update cycle.

The remaining bits (3 to 0) are always low.

The bit assignments for Register D (address 0Dh) are as follows:

Bit	Assignment
7	Valid RAM Time (VRT) Bit
6	0
5	0
4	0
3	0
2	0
1	0
0	0

The VRT bit indicates that the contents of the RAM and time are valid. A "0" appears in the VRT bit when the power sense (PS) pin is low. The processor can set the VRT bit when the time and calendar are initialised to indicate that they are valid. The VRT bit is a read-only bit and is not modified by the $\overline{\text{RES}}$ pin. The VRT bit can only be set by reading the D register.

Bits 6 to 0 are unused and are always read as zeroes.

3.9 Floppy Disk Controller (uPD765A).

The uPD765A Floppy Disk Controller (FDC) contains two registers which are accessible to the CPU; the Main Status Register (at I/O address 03F4) and the Data Register (at I/O address 03F5) both of which are 8 bits wide. The Status register contains the status of the FDC and may be accessed at any time. The Data Register is actually made up of several registers in a stack and stores data, commands and Floppy Disk Drive (FDD) status information. Data is written into the data register in order to program a particular command. The data address is read in order to obtain the result after an operation. The Main Status register (I/O address 3F4h) may only be read and is used to facilitate the transfer of data between the CPU and the uPD765A FDC.

There are 15 separate commands which the uPD765A FDC can execute. Each of these commands require multiple bytes to fully specify the operation. The result after execution of the command may also be a multi-byte transfer back to the processor. Because of this multi-byte interchange of information between the processor and the FDC, it is convenient to consider each command as consisting of three phases:

Command Phase:
    The FDC accepts all information to perform a particular operation from the CPU.
Execution Phase:
    The FDC performs the operation.
Result Phase:
    After completion of the operation, status and housekeeping information are made available to the CPU.

The uPD765A contains five status registers. The main status register mentioned earlier which may be read at any time and four result phase status registers (ST0, ST1, ST2 and ST3) which are only made available during the Result Phase after completion of a command. The particular command which has been executed determines which status registers will be returned.

The Command bytes which are sent to the uPD765A during the Command Phase must occur in the order shown in the command table. That is, the command code must be sent first followed by the other bytes in the prescribed sequence. No foreshortening of the Command Phase or the Result Phase is allowed. After the last byte of data in the Command Phase is sent the Execution Phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result Phase, the command is automatically ended and the uPD765A is ready for a new command.

It is important to note that during the Result phase all bytes shown in the Command table msut be read. The Read Data command, for example has seven bytes listed in the result phase. All seven bytes must be read out else a new command will not be accepted.

The status registers as follows:

Main Status Register

Bit(s)	**Function**
7	Request for Master (RQM).
6	Data Input/Output (DIO).
5	Execution Mode (EXM).
4	FDC Busy (CB).
3	FDD 3 Busy (D3B).
2	FDD 2 Busy (D2B).
1	FDD 1 Busy (D1B).
0	FDD 0 Busy (D0B).

Status Register 0 (ST0)

Bit(s)	**Function**
7	Interrupt Code bit 1 (IC1).
6	Interrupt Code bit 2 (IC2).
5	Seek End (SE).
4	Equipment Check (EC).
3	Not Ready (NR).
2	Head Address (HD).
1	Unit Select 1 (US1).
0	Unit Select 2 (US2).

Status Register 1 (ST1)

Bit(s)	**Function**
7	End of Cylinder (EN).
6	Always zero.
5	Data Error (DE).
4	Over Run (OR).
3	Always zero.
2	No Data (ND).
1	Not Writable (NW).
0	Missing Address Mark (MA).

Status Register 2 (ST2)

Bit(s)	Function
7	Always zero.
6	Control Mark (CM).
5	Data Error in Data Field (DD).
4	Wrong Cylinder (WC).
3	Scan Equal Hit (SH).
2	Scan Not Satisfied (SN).
1	Bad Cylinder (BC).
0	Missing Address Mark in Data Field (MD).

Status Register 3 (ST3)

Bit(s)	Function
7	Fault (FT).
6	Write Protect (WP).
5	Ready (RY).
4	Track 0 (T0).
3	Two Side (TS).
2	Head Address (HD).
1	Unit Select 1 (US1).
0	Unit Select 0 (US0).

The Commands are as follows:

Read Data

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	0.
3	0.
2	1.
1	1.
0	0.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector to be read.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Gap 3 Length.

Byte 9: DTL - Data Length to be read.

During execution data is transferred between the FDD and the CPU memory.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head read.

Byte 6: Final Sector read.

Byte 7: Number of bytes read.

Read Track

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	0.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	0.
3	0.
2	0.
1	1.
0	0.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.

Bit(s)	Function
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector to be read.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Gap 3 Length.

Byte 9: DTL - Data Length to be read.

During execution data is transferred between the FDD and the CPU memory. The FDC reads all data fields from index hole to EOT.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head read.

Byte 6: Final Sector read.

Byte 7: Number of bytes read.

Read Deleted Data

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	0.
3	1.
2	1.
1	0.
0	0.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector to be read.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Gap 3 Length.

Byte 9: DTL - Data Length to be read.

During execution data is transferred between the FDD and the CPU memory.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table). Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head read.

Byte 6: Final Sector read.

Byte 7: Number of bytes read.

Read ID

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	0.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	0.
4	0.
3	1.
2	0.
1	1.
0	0.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

During execution the first correct ID information on the cylinder is stored in the Data Register.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Cylinder.

Byte 5: head.

Byte 6: Sector.

Byte 7: Number of bytes per sector.

Write Data

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	0.
4	0.
3	0.
2	1.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

During execution data is transferred between the cpu memory and the FDD.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head written.

Byte 6: Final Sector written.

Byte 7: Number of bytes written.

Write Deleted Data

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	0.
4	0.
3	1.
2	0.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Gap 3 Length.

Byte 9: DTL - Data Length to be written.

During execution data is transferred between the CPU memory and the FDD.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head written.

Byte 6: Final Sector written.

Byte 7: Number of bytes written.

Format Track

Command Phase: 6 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	0.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	0.
4	0.
3	1.
2	1.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Number of bytes per sector.

Byte 4: Number of sectors per track.

Byte 5: GPL - Gap 3 Length.

Byte 6: D - Filler Byte.

During execution the FDC writes address headers to the entire track.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Cylinder number.

Byte 5: Head.

Byte 6: Sector.

Byte 7: Number of bytes per sector.

Scan Equal

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	1.
3	0.
2	0.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Gap 3 Length.

Byte 9: STP - Step Factor: 1 = Contiguous: 2 = Alternate Sectors.

During execution data is transferred from the CPU memory and compared with data from the FDD.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head compared.

Byte 6: Final Sector compared.

Byte 7: Number of bytes compared.

Scan Low or Equal

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	1.
3	1.
2	0.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector to be compared.

Byte 6: Number of bytes per sector.

Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Length of Gap 3.

Byte 9: STP - Step Factor: 1 = Contiguous: 2 = Alternate Sectors.

During execution data from the CPU memory is compared with data from the FDD.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head compared.

Byte 6: Final Sector compared.

Byte 7: Number of bytes compared.

Scan High or Equal

Command Phase: 9 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	(MT) Multi-Track {Enable/Disable}.
6	(FM) Select {MFM/FM} (Single/Dobule density) Mode.
5	(SK) Enable Skip deleted data address mark.
4	1.
3	1.
2	1.
1	0.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Select (0 or 1).
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: Cylinder Number (0-76).

Byte 4: Head Number (as specified in the ID field).

Byte 5: Sector to be compared.

Byte 6: Number of bytes per sector.Byte 7: EOT - Final sector number on track.

Byte 8: GPL - Length of Gap 3.

Byte 9: STP - Step Factor: 1 = Contiguous: 2 = Alternate Sectors.

During execution data from the CPU memory is compared with data from the FDD.

The Result phase returns 7 bytes:

Byte 1: ST0 - Status register 0 (See ST0 table).

Byte 2: ST1 - Status register 1 (See ST1 table).

Byte 3: ST2 - Status register 2 (See ST2 table).

Byte 4: Final Cylinder number.

Byte 5: Final head compared.

Byte 6: Final Sector compared.

Byte 7: Number of bytes compared.

Recalibrate

Command Phase: 2 bytes.

Byte 1: Command Code.

Bit(s)	Function
7	0.
6	0.
5	0.
4	0.
3	0.
2	1.
1	1.
0	1.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	0.
1	US1.
0	US0 - Unit Select (0 or 1).

During execution phase, the Head is retracted to Track zero.

No status information is returned during the result phase.

Sense Interrupt Status

Command Phase: 1 byte.

Byte 1: Command Code = 08h.

The Result phase returns two bytes:

Byte 1: ST0 - Status Register 0.

Byte 2: PCN - Present Cylinder Number.

Specify

Command Phase: 3 bytes.

Byte 1: Command Code = 03h.

Byte 2: SRT/HUT - Step Rate Time (4 MS bits - in 1 ms increments)/Head Unload Time (4 LS bits - in 16 ms increments).

Byte 3: HLT/ND - Head Load Time (Bits 1 to 7 - in 2 ms increments)/Non-DMA Mode (Bit 0).

Seek

Command Phase: 3 bytes.

Byte 1: Command Code = 0Fh.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	HD - Head Number.
1	US1.
0	US0 - Unit Select (0 or 1).

Byte 3: New Cylinder Number.

During execution phase, the Head is positioned to the specified Cylinder.

No status information is returned during the result phase.

Sense Drive Status

Command Phase: 2 bytes.

Byte 1: Command Code = 04h.

Byte 2: Head and Unit select.

Bit(s)	Function
7-3	Don't Care.
2	0.
1	US1.
0	US0 - Unit Select (0 or 1).

Result phase: 1 byte.

Byte 1: ST3 - Status Register 3.

Invalid Opcodes

All command codes not listed above are considered invalid. When an invalid code is encountered the FDC returns the ST0 register with the MS bit (Invalid Opcode bit) set.

Section 2 Index Appendices

Appendix 1: Mouse Software Interfaces

The two operating systems supplied with your PC1512, MS-DOS 3.2 and DOS Plus, both support a set of Mouse functions which allow a program to access the mouse and control the cursor. These functions can be called from within your application programs by using the software interfaces described below.

The mouse functions described in this appendix are as follows.

Number	Function
0**	Mouse Initialisation
1	Show Cursor
2	Hide Cursor
3	Get Mouse Position and Button Status
4**	Set Mouse Cursor Position
5	Get Button Press Information
6	Get Button Release Information
7**	Set Minimum and Maximum X-Cursor Position
8**	Set Minimum and Maximum Y-Cursor Position
9	Set Graphics Cursor Block
10	Set Text Cursor
11	Read Mouse Motion Counters
12**	Set User Defined Subroutine Input Mask
13	Light Pen Emulation Mode On
14	Light Pen Emulation Mode Off
15**	Set Mickey/Pixel Ratio
16	Conditional Off
19	Set Double Speed Threshold

The DOS Plus operating system supports only the subset of mouse functions marked with the double asterisks built into the DOS Plus input/output system.

For the MS-DOS operating system, the program MOUSE.COM must be loaded by either typing "MOUSE" at the keyboard or by having a "MOUSE" line in your AUTOEXEC.BAT.

When MOUSE.COM is loaded, it performs an initialisation process and installs the mouse driver software into the system. Once installed, the mouse driver remains permanently resident until the next time you bootstrap your computer. After successfully completing initialisation, the following message is output to the display:

```
"--- Installing Mouse Device Driver V5.00 ---"
```

If the mouse driver fails to load you will get one of the two following messages:

1. "MOUSE: Mouse Driver already installed."
   Because either you've previously installed MOUSE.COM or you've attempted to install it under DOS Plus which has its own built in mouse driver.
2. "MOUSE: Amstrad Mouse not found."
   Because either there's a hardware fault or your hardware isn't an Amstrad PC.

During initialisation, the following actions take place:

1. The hardware ticker routine residing at software interrupt 8 which is invoked every 54ms is replaced by a Mouse Ticker routine that is invoked every 18ms.
2. Counter 0 of the 8253 interrupt controller is re-programmed so that it produces an interrupt every 18ms rather than every 54ms.
3. The Mouse Buttons Interrupt routine is inserted into the Mouse Buttons Interrupt vector (software interrupt 6).
4. The Amstrad PC Mouse X and Y movement registers are initialised to zero. The Mouse Buttons are both marked as being released.
5. The Non Volatile RAM is read to determine the X & Y Scaling factors, which are to be used during cursor key generation for mouse movement in text mode.
6. The Mouse driver is initialised to be in Text Mode.

The above initialisation applies only to MOUSE.COM in the MS-DOS environment. For DOS Plus which has a built in mouse driver, a similar initialisaiton takes place during system bootstrap. The basic ticker rate for DOS Plus is also 18ms (or about a 54Hz rate).

The general procedure for making an assembly language program call to the mouse function driver program is:

1. Load the specified register parameters.
2. Execute software interrupt 51 (033h).

The cursor coordinates required for the various function calls are in the form of X-Cursor (horizontal) and Y-Cursor (vertical) values. The range of the X-Cursor is always the full 0 to 639 points of the high resolution graphics screen and the Y-Cursor ranges from 0 to 199. This coordinate system defines the "virtual" screen and when in modes with less resolution than 640 points then the least significant bits of the X-Cursor are ignored. In 4-Colour (320 x 200) graphics only even values are significant while in 80 column text mode only every eighth position is valid and in 40 column text modes only every 16th position is valid. Supplied values are rounded to the nearest values permitted for the current screen mode.

The standard unit of mouse motion is called the "mickey" and is equal to approximately 1/200 of an inch. See Mouse function 15 which sets the mickey to pixel ratios.

In Text Mode, mouse movement will cause cursor key tokens to be inserted into the keyboard buffer. The scaling factors read from the NVR during initialisation are used to determine how many units of mouse movement are to be sensed before a single cursor key token is inserted into the keyboard buffer. Invoking any Mouse Function except Function 0 or Function 2 will disable this extra mode (i.e. cursor movement tokens are not generated). Invoking Function 0 (Initialisation) enables this extra mode (i.e. cursor tokens are generated) and invoking Function 2 (Hide Cursor) does not change the current mode.

In Text Mode, the mouse buttons interrupt routine (interrupt 6) translates the Left and Right mouse buttons into the appropriate scan codes which are held in NVR bytes 29-30 (for Left) and 31-32 (for Right). The default NVR value for these scan codes is the ignore code (all F's). The NVRPATCH programs can be used to set the mouse button codes to handy values such as CR and ESC.

In Graphics Mode, the mouse buttons interrupt routine translates the Right mouse button to the shift key token, and the Left button is passed through as a mouse event to the user defined subroutine. (See mouse function 12.)

Function 0: Mouse Initialisation.

This function initialises the mouse driver and returns the current status of the mouse hardware and software.

CPU registers are used as follows:

Entry:
    AX = 0
Exit:
    AX = Mouse Status
    BX = Number of Buttons.

    All Flags and other registers preserved.

Since the mouse hardware is verified by power-up testing the driver always returns a mouse status of true (-1). If the mouse driver is not resident then AX is returned as false (0).

Both MS-DOS and DOS Plus return the number of buttons as 1.

The mouse driver parameters are reset to the following values:

Parameter	Value
Cursor Flag	Hidden (-1)
Cursor Position	Center Screen
Graphics Cursor	Arrow
Hot Spot	-1, -1
Text Cursor	Inverting box
User Defined Call Mask	Zeros
Light Pen Emulation Mode	Enabled
Mickey to X-Pixel Ratio	8
Mickey to Y-Pixel Ratio	16 (8 for DOS Plus)
Min/Max X-Cursor Position	0/639
Min/Max Y-Cursor Position	0/199

The mouse X and Y hardware counters are reset and a number of internal software counters are zeroed.

The mouse driver is initialised to be in Text Mode (and cursor tokens are generated in response to mouse motion).

Function 1: Show Cursor.

This function increments the Cursor Flag and, if the flag is zero, the cursor display is enabled.

CPU registers are used as follows:

Entry:
    AX = 1.
Exit:
    All flags and registers preserved.

Function 2: Hide Cursor.

This function decrements the Cursor Flag.

CPU registers are used as follows:

Entry:
    AX = 2.
Exit:
    All flags and registers preserved.

Function 3: Get Mouse Position and Button Status.

This function returns the state of the Left and Right buttons and the current cursor position.

CPU registers are used as follows:

Entry:
    AX = 3.
Exit:
    BX = Button status.
    CX = X-Cursor Position.
    DX = Y-Cursor Position.
    All flags and other registers preserved.

The Button Status word returned in BX is a single integer value. Bits 0 and 1 represent the Left and Right buttons, respectively. A bit is set if a button is down and clear if it is up.

Function 4: Set Mouse Cursor.

This function sets the cursor to the specified X-Cursor and Y-Cursor positions. The values must in range of the virtual screen. If the screen is not in high resolution mode, the values are rounded to the nearest values permitted for the current screen mode.

CPU registers are used as follows:

Entry:
    AX = 4.
    CX = X-Cursor position.
    DX = Y-Cursor position.
Exit:
    All flags and registers preserved.

Function 5: Get Button Press Information.

This function returns the current button status, a count of button presses since last call to this function, and the X-Cursor and Y-Cursor positions at the last button press.

CPU registers are used as follows:

Entry:
    AX = 5
    BX = Button Number (0=Left/1=Right).
Exit:
    AX = Button status.
    BX = Count of Presses since last call (0-32k).
    CX = X-Cursor at last press.
    DX = Y-Cursor at last press.
    All flags and other registers preserved.

Function 6: Get Button Release Information.

This function returns the current button status, a count of button releases since last call to this function, and the X-Cursor and Y-Cursor positions at the last button release.

CPU registers are used as follows:

Entry:
    AX = 5.
    BX = Button Number (0=Left/1=Right).
Exit:
    AX = Button status.
    BX = Count of releases since last call (0-32k).
    CX = X-Cursor at last release.
    DX = Y-Cursor at last release.

All flags and other registers preserved.

Function 7: Set Minimum and Maximum X-Cursor Position.

This function sets the minimum and maximum X-Cursor position. Subsequent cursor motion is restricted to the specified range.

CPU registers are used as follows:

Entry:
AX = 7.
CX = Minimum X-Cursor Position.
DX = Maximum X-Cursor Position.
Exit:
All flags and registers preserved.

If the cursor is outside the area when the call is made, it is set to just inside the area. If Minimum is greater than Maximum, the two values are exchanged.

Function 8: Set Minimum and Maximum Y-Cursor Position.

This function sets the minimum and maximum Y-Cursor position. Subsequent cursor motion is restricted to the specified range.

CPU registers are used as follows:

Entry:
AX = 8.
CX = Minimum Y-Cursor Position.
DX = Maximum Y-Cursor Position.
Exit:
All flags and registers preserved.

If the cursor is outside the area when the call is made, it is set to just inside the area. if Minimum is greater than Maximum, the two values are exchanged.

Function 9: Set Graphics Cursor Block.

This function defines the shape, colour, and center of the cursor for graphics mode.

CPU registers are used as follows:

Entry:
AX = 9.
BX = X-Cursor Hot Spot.
CX = Y-Cursor Hot Spot.
DX = Pointer to Screen and Cursor Masks.
ES = Segment Address of Screen and Cursor Masks.
Exit:
All flags and registers preserved.

The Hot Spot is a point relative to the upper left corner of the cursor block used to determine the cursor coordinates. Both coordinates must be in the range of -16 to +16.

The values in the screen mask and cursor mask are used to build the cursor shape and colour. The ES register contains the segment address of the screen and cursor mask array and DX is the offset to be applied to the ES register.

The screen and cursor masks are two 16- by 16-bit arrays arranged contiguously in memory. The screen mask determines whether the cursor pixel is part of the shape or part of the background. The cursor mask determines how the pixel under the cursor contributes to the colour of the cursor. To create the cursor, the mouse software first logically ANDs the screen mask with the 256 bits of data that define the pixels under the cursor. Then, it logically XORs the cursor mask with the result of the AND operation. The following truth table shows the relationship between the screen mask, the cursor mask, and the resultant screen memory.

The values in the screen mask and cursor mask are used to build the cursor shape and colour. The ES register contains the segment address of the screen and cursor mask array and DX is the offset to be applied to the ES register.

The screen and cursor masks are two 16- by 16-bit arrays arranged contiguously in memory. The screen mask determines whether the cursor pixel is part of the shape or part of the background. The cursor mask determines how the pixel under the cursor contributes to the colour of the cursor. To create the cursor, the mouse software first logically ANDs the screen mask with the 256 bits of data that define the pixels under the cursor. Then, it logically XORs the cursor mask with the result of the AND operation. The following truth table shows the relationship between the screen mask, the cursor mask, and the resultant screen memory.

The values in the screen mask and cursor mask are used to build the cursor shape and colour. The ES register contains the segment address of the screen and cursor mask array and DX is the offset to be applied to the ES register.

The screen and cursor masks are two 16- by 16-bit arrays arranged contiguously in memory. The screen mask determines whether the cursor pixel is part of the shape or part of the background. The cursor mask determines how the pixel under the cursor contributes to the colour of the cursor. To create the cursor, the mouse software first logically ANDs the screen mask with the 256 bits of data that define the pixels under the cursor. Then, it logically XORs the cursor mask with the result of the AND operation. The following truth table shows the relationship between the screen mask, the cursor mask, and the resultant screen memory.

The values in the screen mask and cursor mask are used to build the cursor shape and colour. The ES register contains the segment address of the screen and cursor mask array and DX is the offset to be applied to the ES register.

The screen and cursor masks are two 16- by 16-bit arrays arranged contiguously in memory. The screen mask determines whether the cursor pixel is part of the shape or part of the background. The cursor mask determines how the pixel under the cursor contributes to the colour of the cursor. To create the cursor, the mouse software first logically ANDs the screen mask with the 256 bits of data that define the pixels under the cursor. Then, it logically XORs the cursor mask with the result of the AND operation. The following truth table shows the relationship between the screen mask, the cursor mask, and the resultant screen memory.

Screen Mask Bit	**Cursor Mask Bit**	**Resultant Screen Bit**
0	0	0
0	1	1
1	0	Unchanged
1	1	Inverted

In high resolution (640 by 200) graphics mode each bit in the screen and cursor masks logically maps to one bit on the screen. In four colour (320 by 200) graphics mode each pair of bits correspond to one pixel.

Function 10: Set Text Cursor.

This function selects the software or hardware text cursor and defines the attributes of the selected cursor.

CPU registers are used as follows:

Entry:
    AX = 10.
    BX = Cursor Type (0=Software/1=Hardware).
    CX = Screen Mask/Scan Line Start.
    DX = Cursor Mask/Scan Line Stop.
Exit:
    All flags and registers preserved.

If the software text cursor is selected CX & DX contain Screen and Cursor masks. The 16-bit masks are ANDed and XORed in the same manner as the graphics cursor operation and operate upon the character and attributes bytes of the character position of the cursor.

In both 40-column and 80-column text modes the 16-bits of screen data for each character take the following form:

Bits:	15	14..12	11	10..8	7........0
Contents:	B	Backgnd	I	Foregnd	Character

Where:

B
    Blink Bit.
Backgnd
    Three bit (RGB) pattern specifying background colour.
I
    Intensity bit.
Foregnd
    Three bit (RGB) pattern specifying foreground colour.

Refer to section 1.11 for the attribute byte details.

The screen and cursor masks are divided into the same fields as shown above so that the value of the masks defines the new attributes of the character when the cursor is over it. For example a screen mask of 077FFh and a cursor mask of 07700h would invert the foreground and background colours.

If the hardware cursor is selected CX & DX define the first and last scan line in the cursor shown on the screen.

Function 11: Read Mouse Motion Counters.

This function returns the horizontal and vertical mickey count since the last call to this function. CPU registers are used as follows:

Entry:
    AX = 11.
Exit:
    CX = X-Count.

DX = Y-Count.
All flags and other registers preserved.

The "mickey" is the standard unit mouse motion equal to approximately 1/200 of an inch. See Mouse function 15 which sets the mickey to pixel ratios.

Function 12: Set User Defined Subroutine Input Mask.

This function sets the call mask and subroutine address for the mouse interrupts. CPU registers are used as follows:

Entry:
AX = 12.
CX = Call Event Mask.
DX = Address Offset to Subroutine.
ES = Segment address of Subroutine.

All flags and other registers preserved.

The mouse driver uses the ticker interrupt to poll the mouse hardware (at approx. 55 times per second) and when one of the events specified in the call event mask is noted your subroutine is called. The layout of the call event mask is:

Bit	Event (1=Enabled)
15-5	Unused.
4	Right Button Released. (will never occur).
3	Right Button Pressed. (will never occur).
2	Left Button Released.
1	Left Button Pressed.
0	Cursor Position Changed.

Note that calling mouse function zero (Initialisation) disables all events so that function 12 must be called again.

When calling your subroutine the CPU registers are loaded as follows:

AX = Event bit (as per the above table).
BX = Button State (BL = Left Button / BH = Right Button - will never occur).
CX = X-Cursor.
DX = Y-Cursor.

Function 13: Light Pen Emulation Mode On.

This function enables the light pen emulation by the mouse. CPU registers are used as follows:

Entry:
AX = 13.
Exit:
All flags and registers preserved.

With the Light Pen Emulation on, the VDU I/O software interrupt (Int 16) returns mouse identification vice the normal light pen address information (See VDU Sub-Function 4).

Function 14: Light Pen Emulation Mode Off.

This function disables the light pen emulation by the mouse. CPU registers are used as follows:

Entry:
AX = 14.
Exit:
All flags and registers preserved.

Function 15: Set Mickey/Pixel Ratio.

This function sets the mickey to pixel ratio for mouse motion. CPU registers are used as follows:

Entry:
AX = 15.
CX = X-Mickey/Pixel Ratio.
DX = Y-Mickey/Pixel Ratio.
Exit:
All flags and registers preserved.

The X- and Y- ratios specify a number of mickeys per 8 pixels. The values must be in the range of 1 to 32767.

With a setting of 16 mickeys per 8 pixels horizontally it takes about 6.4 inches of mouse movement to move the cursor across the screen (640 pixels). With the same 16 mickeys per 8 pixels vertically it takes about 2 inches of travel to move the cursor the full vertical deflection (200 pixels).

Function 16: Conditional Off.

This function defines a region on the screen for updating. CPU registers are used as follows:

Entry:
    AX = 16.
    CX = Upper X-Screen Coordinate.
    DX = Upper Y-Screen Coordinate.
    SI = Lower X-Screen Coordinate.
    DI = Lower Y-Screen Coordinate.
Exit:
    All flags and registers preserved.

The mouse cursor is hidden while the screen is being updated and a call to function 1 is needed to show the cursor again.

Function 16 is similar to function 2 (Hide Cursor) bit is for advanced applications which require quicker screen updates.

Function 19: Set Double Speed Threshold.

This function defines the threshold speed for doubling the cursor's motion on the screen. CPU registers are used as follows:

Entry:
    AX = 19.
    DX = Threshold Speed in Mickeys/Second.

This function makes it easier to point at images widely separate on the screen.

A threshold value of zero sets a value of 64 mickets.second. Setting a large value (such as 32767) disables the double speed threshold.

Appendix 2 MS-DOS System Configuration

The MS-DOS operating system allows for a number of installation specific configuration options during the system startup progress through the use a file called CONFIG.SYS when it is found in the root directory of the startup disk. These configuration options include the following commands:

BREAK
    Extended BREAK checking (Ctrl-C).
BUFFERS
    Number of sector buffers.
COUNTRY
    Country Specific parameter selection.
DEVICE
    Device driver installations.
DRIVPARM
    Override the drive parameters for a logical drive.
FCBS
    Number of files open by file control blocks.
FILES
    Maximum number of file handles open concurrently.
LASTDRIVE
    Maximum drive letter allowable.
SHELL
    Top level command processor specification.

The CONFIG.SYS can be created with any text editor and the simple screen editor RPED is ideal for this purpose.

2.1 BREAK Command

This command enables the MS-DOS extended break checking to be either set or reset. Normally, MS-DOS checks to see if CTRL-C has been typed while it is reading from the keyboard, writing to the screen or a printer. Setting Break to 'on' allows CTRL-C checking to be extended to other functions such as disk reads or writes. The syntax of the BREAK command is:

```
BREAK=[ON]
    or
BREAK=[OFF]
```

If no field is specified then OFF is assumed (as the default value).

2.2 BUFFERS Command

This command allows you to specify the number of buffers that MS-DOS allocates when it starts up. A disk buffer is a block of memory where MS-DOS holds data being read from or written to a disk when the amount of data is not an exact multiple of sector size.

The syntax of the BUFFERS command is:

```
BUFFERS=n
```

Where 'n' is a number between 1 and 99. If the BUFFERS command is not used then MS-DOS defaults to 2 buffers. The number of buffers remains in effect after bootstrap until the machine is switched off or bootstrapped again.

2.3 Country Command

The country command is used to select the country dependent information as shown in appendix 3.

The syntax of the country command is:

```
COUNTRY=nnn
```

Where 'nnn' is the 3-digit country (Num) code from Appendix 3. Note that only the information in the table is affected and other country dependent factors such as the language links, N-Utility setup, KEYBxx, and national variant disks affect the total country dependent environment.

2.4 DEVICE Command

This command installs the device driver in the specified pathname to the system list.

The syntax of the DEVICE command is:

```
DEVICE=[drive:]<pathname>
```

The file specified is loaded and given control. The driver may then perform the necessary steps to configure itself and the system for its operation. See the MS-DOS Technical Reference Manual for information on how to create your own device driver.

Your MS-DOS disk (Disk 1) contains two installable device drivers, DRIVER.SYS, and RAMDRIVE.SYS which can be used for variable device configurations.

If you plan to use the ANSI escape sequences described in the PC1512 users manual, you would need to include the following command in your CONFIG.SYS file:

```
DEVICE=ANSI.SYS
```

This command causes MS-DOS to replace all keyboard input and screen output support with the ANSI escape sequences.

Note that the Amstrad disc is supplied with a version of ANSI.SYS that conforms with the NVR value (byte 36) for screen colouring and which refrains from blanking the screen unnecessarily when scrolling.

2.4.1 DRIVER.SYS

DRIVER.SYS is an installable device driver that supports external drives. To install DRIVER.SYS, include the following command in your CONFIG.SYS file:

```
DEVICE=DRIVER.SYS /D:dd [/C] [/F:ff] [/H:hh] [/N] [/S:ss] [/T:tt]
```

Where:

/D:dd is drive number (0-127: Floppy drives, 128-255 Hard drives)

and optionally:

/C indicates changeline (doorlock) support required.
/F:ff indicates the form factor where:
0 = 5.25", 320/360 KB
1 = 5.25", 1.2 MB
2 = 720 KB
3 = 8" Single Density
4 = 8" Double Density
5 = Hard Disk
6 = Tape Drive
7 = Other
/H:hh is the maximum head number (1-99).
/N indicates non-removable block device.
/S:ss is the number of sectors per track (1-99).

/T:tt is the number of tracks per side (1-999).

2.4.2 RAMDRIVE.SYS

RAMDRIVE.SYS is an installable device driver which enables the usage of a portion of the computer's memory as though it were a disk drive. This area of memory is referred to as a RAM disk or a virtual disk.

If you have extended memory installed starting at the 1MB boundary or if you have an extended memory which meets the LIM [Lotus(R)/Intel(R)/Microsoft(R)] Expanded Memory Specification, you can use this memory for one or more RAM disks. Otherwise RAMDRIVE.SYS locates RAM drives in low memory.

To install RAMDRIVE.SYS, include the following command in your CONFIG.SYS file:

```
DEVICE=RAMDRIVE.SYS [bbbb [ssss [dddd]]] [/A]
```

Where:

bbbb is disk size in kilobytes. Default is 64 and minimum is 16.
The keyword NVR is allowed and causes the RAM DISK size in the NVR to be used.
ssss is the sector size. The values 128, 256, 512, and 1024 are allowed. Default is 128.
dddd is the number of root directory entries. The default is 64, the minimum value is 2 and the maximum is 1024.
/A indicates that an extended memory board which meets the LIM Expanded Memory Specification for a RAM drive is in use.

2.5 DRIVPARM Command

The DRIVPARM command allows overriding of the device parameters for a specific logical drive.

The syntax is:

```
DRIVPARM= /D:dd [/F:ff /T:tt /S:ss /N /C /H:hh]
```

Where:

/D:dd is drive number (0-255)

and optionally:

/T:tt is the number of tracks per side (1-999).
/S:ss is the number of sectors per track (1-99).
/H:hh is the maximum head number (1-99).
/C indicates changeline (doorlock) support required.
/N indicates non-removable block device.
/F:ff indicates the form factor where:
0 = 5.25", 320/360 KB
1 = 5.25", 1.2 MB
2 = 720 KB
3 = 8" Single Density
4 = 8" Double Density
5 = Hard Disk
6 = Tape Drive
7 = Other

This command allows the overriding of default system parameters for a particular logical drive. This information would be used by the commands which create new diskettes (such as FORMAT and COPY) when writing out the directory and FAT (File Allocation Table) information. For any physical device which is read the information in the FAT ID is used when determining device characteristics.

2.6 FCBS Command

The FCBS command allows you to specify the number of file control blocks available to the system and consequently the number of files which can be opened at any one time.

The syntax of the FCBs is:

```
FCBS=x,y
```

Where <x> is the number of FCBs (in the range of 1 to 255) to allocate and <y> is the number of FCBs protected from closure when a program tries to open more than <x> files. The first <y> files opened will be protected. MS-DOS selects the least recently used (non-protected) FCB when it must automatically close a file.

If the FCBS command is not used MS-DOS defaults <x> and <y> to 4 and 0 respectively. It is an error to set <y> greater than <x>

2.7 FILES Command

The FILES command specifies the maximum number of file handles that can concurrently be opened. When a program opens a file or a device it is assigned an identifier or "handle" which can be used by that program in referring to the file.

The syntax of the FILES command is:

```
FILES=n
```

Where 'n' is the number of handles in the range of 8 to 255. When no FILES command is used MS-DOS assumes a default value of 8. Any value higher than 20 serves no useful function.

2.8 LASTDRIVE Command

The LASTDRIVE command is used to set the maximum drive letter which MS-DOS will accept.

The syntax of the LASTDRIVE command is:

```
LASTDRIVE=d
```

Where 'd' is any letter from A to Z (and is case insensitive). When the drive letter is lower than the actual physical drives then MS-DOS ignores the LASTDRIVE specification and uses the default value which is the letter 'E'.

2.9 SHELL Command

The SHELL command is used to specify an alternate top-level command processor in place of the standard COMMAND.COM file.

The syntax of the SHELL command is:

```
SHELL=[drive:]pathname [param1 [parm2 ..[paramn]]]
```

This command is used in conjunction with major software packages which furnish their own command processors. The MS-DOS technical manual contains information on developing command processors.

2.10 KEYBUK Command

When this command is invoked, a resident keyboard interrupt process is installed which replaces the ROS keyboard interrupt process. UK specific characters £, #, @, &, " are mapped to their respective keys. The system available memory will decrease by the resident size of keybuk. Pressing [CTRL] + [ALT] + [F1] restores the ROS keyboard processing and pressing [CTRL] + [ALT] + [F2] resets back to KEYBUK keyboard input.

Note that the PC1512 is supplied with a special version of keybuk which correctly handles the extra key codes generated by the [DEL->] & [ENTER] keys, the joystick port and the two mouse buttons. Using any other version of KEYBUK will render all these keys inoperative.

Appendix 3: Country Dependent Information for MS-DOS 3.2

Country	Num	DtF	DtS	TmS	TmF	CSm	CFt	CSd	ThS	DeS	DlS
Australia	061	1	-	:	1	$	0	2	,	.	,
Belgium	032	1	/	:	1	F	3	2		,	;
Canada	002	2	-	:	1	$	3	2		,	;
Denmark	045	1	/	.	1	DKR	3	2	.	,	;
Finland	358	1	-	:	1	MK	3	2		,	;
France	033	1	/	:	1	F	3	2		,	;
Germany	049	1	.	.	1	DM	0	2	.	,	;
Italy	039	1	/	:	1	Lit	1	0	.	,	;
Israel	972	1	/	:	1	ö	2	2	,	.	,
Middle East	785	1	/	:	1	$	3	3	.	,	;
Netherlands	031	1	-	:	1	*ƒ*	0	2	.	,	;
Norway	047	1	/	.	1	KR	2	2	.	,	;
Portugal	351	1	/	:	1	$	4	2	.	,	;
Spain	034	1	/	:	1	Pt	3	2	.	,	;
Sweden	046	2	-	.	1	SEK	2	2	.	,	;
Switzerland	041	1	.	.	1	Fr	2	2	,	.	,
United Kingdom	044	1	-	:	1	£	0	2	,	.	,
United States	001	0	-	:	0	$	0	2	,	.	,

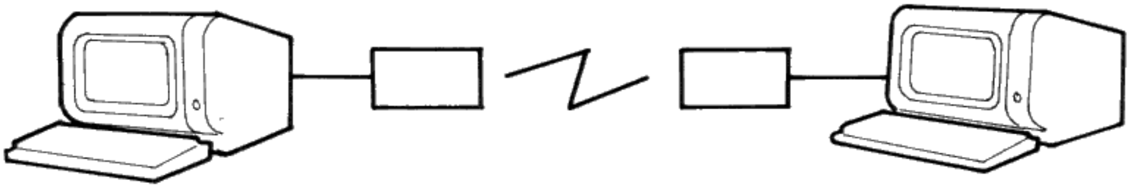
Table Columns:

Num = Country Number Code.
DtF = Date Format. (0 = U.S. M/D/Y, 1=EURO D/M/Y, 2 = JAPAN Y/M/D)
DtS = Date Separator.
TmS = Time Separator.
TmF = Time Format. (0=12-hour clock, 1=24-hour clock)
CSm = Currency Symbol.
CFt = Currency Format. (Bit 0: 0 = Currency symbol Precedes/1=Follows Field, Bits 1 & 2: Number of spaces between Value & Symbol)

CSd =Number of significant decimal digits in currency.
ThS =Thousands Separator.
DeS =Decimal Separator.
DlS =Data List Separator.

Appendix 4: RS232C Connections

For a complete understanding of the connections required between the RS232C and the outside world, it is important to realize that all devices with a serial interface can be classified as either a modem or as a terminal. Modems are merely a way of extending the length of the connection (often via a terminal wire) between two terminals. Fig 1 (below) shows a simplified, idealised terminal to terminal connection through modems.



IDEALISED TERMINAL TO TERMINAL CONNECTION

Fig 1

The standard connector used for serial interfaces has 25 pins although only up to seven are required in most cases. When connecting to a modem a 'one-to-one' cable is used, i.e. pin 1 to pin 1, pin 2 to pin 2, ... pin 25 to pin 25. Assuming such cables are in use, data is transferred as follows:

Following the signal path from left to right (in Fig 2), characters from the keyboard are sent as serial data patterns out of pin 2 of the left-hand terminal, to pin 2 of the modem (the connection marked 'transmit data'). The left-hand modem sends the characters via the telephone line, to the right hand modem. The characters are received pin 3 of the right hand modem (the connection marked 'receive data') which sends them to pin 3 of the right-hand terminal. On receipt of the characters, the right-hand terminal displays them on the screen.

Notice how the names of the connections 'transmit data' and 'receive data' are expressed from the view point of the terminals and not the modems.

The data path from left to right just described, is exactly matched by a data path from right to left which uses the same numbered connections, i.e. pin 2 from the right-hand terminal it its modem (transmitting), and then to pin 3 of the left-hand (receiving) modem to the terminal. This arrangement is perfectly symmetrical, and there is no confusion over who is using which pin number and for what direction of data transfer.

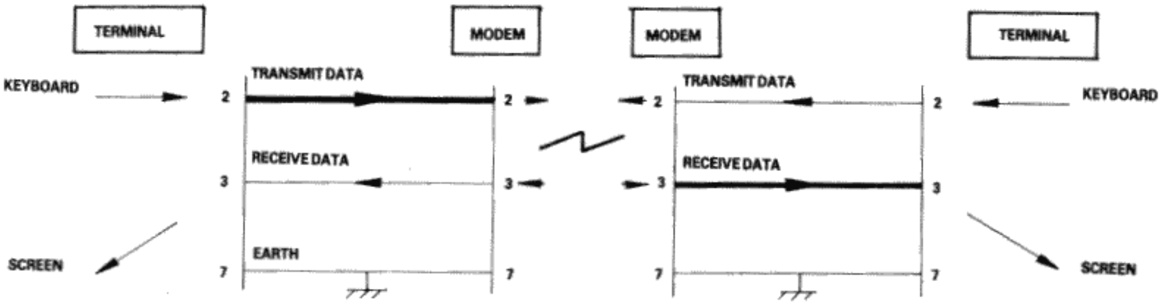


Fig 2

Problems of definition arise, howeve, when we wish to connect two terminals together locally, without the intervening pair of modems. We cannot connect pin 2 to pin 2 because both keyboard will be transmitting head-on and neither screen is connected to anyone who is sending. The obvious solution is to cross over pins 2 and 3 so that the transmit pin of each terminal is connected to the receive pin of the other. A cable containing such a cross-over connection is known as a 'Null-modem' cable because of the way in which in replaces the pair of back to back modems.

The earth pin (pin 7) is still common to both terminals using this arrangement.

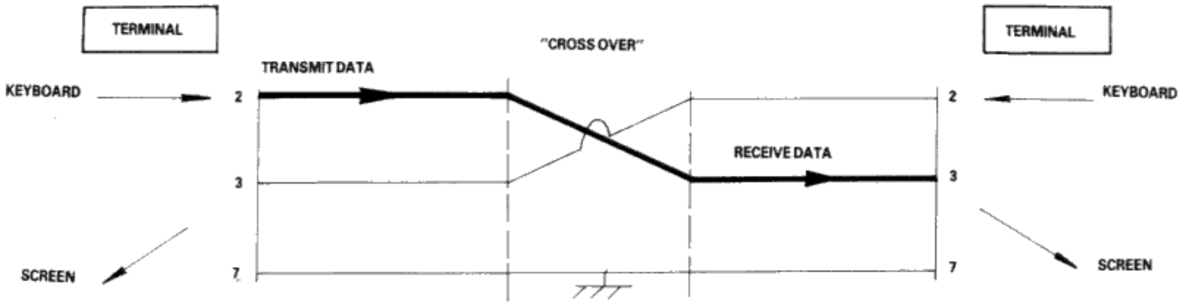


Fig 3

Naturally, the Amstrad PC1512 with its RS232C interface is considered a terminal, and therefore to connect to a modem (for example, to dial-up a database) requires a simple one-to-one cable.

The Null-modem cable is required for connecting to other terminals. The sort of equipment we mean by terminals is: a second Amstrad computer plus RS232C, a conventional Visual Display Unit (VDU), a printer with a serial interface, or any other serial interface device.

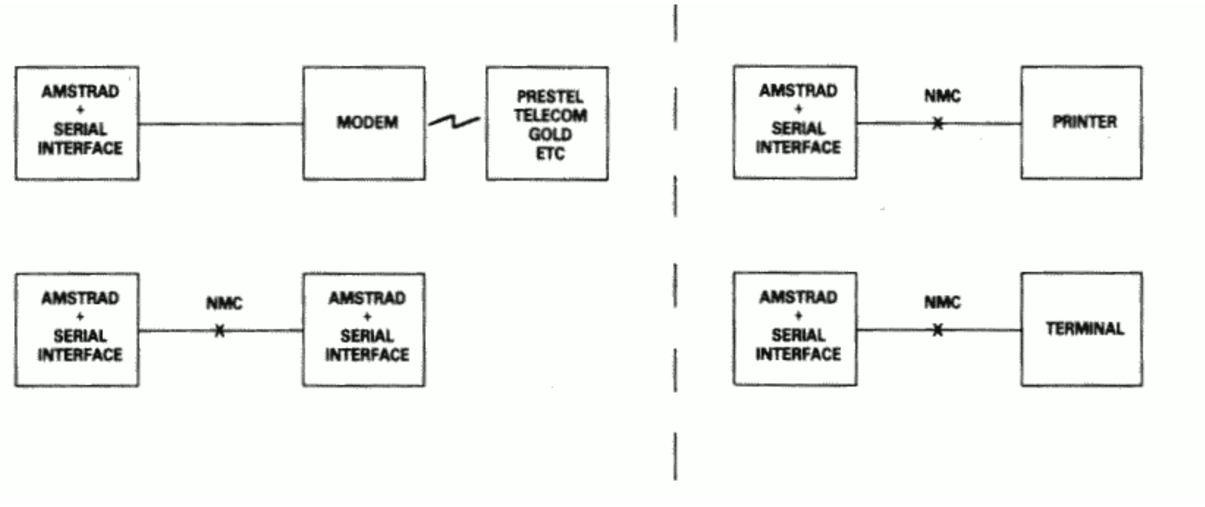

AMSTRAD + SERIAL INTERFACE
MODEM
PRESTEL TELECOM GOLD ETC
AMSTRAD + SERIAL INTERFACE
NMC
PRINTER
AMSTRAD + SERIAL INTERFACE
NMC
AMSTRAD + SERIAL INTERFACE
AMSTRAD + SERIAL INTERFACE
NMC
TERMINAL

Fig 4

There is a point to be noted here: many manufacturers of devices such as desk-top computers wire up their serial interface (for VDU or a Printer) as if it were a modem, not a terminal. This is in the belief that life will therefore be simpler because VDU's and printers can connected to that computer with one-to-one cables.

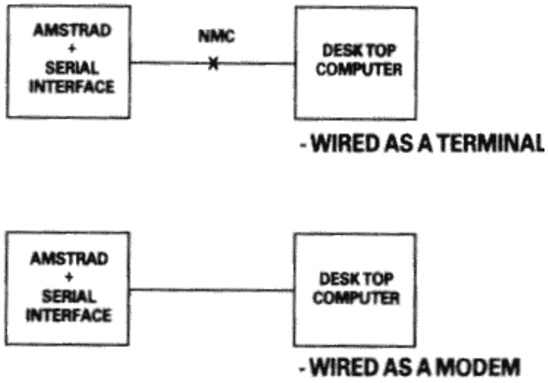


Fig 5

In a perfect world, it would be possible to identify which serial devices behave like modems and which ones behave like terminals by examining the 'sex' of the 25-way connector - terminals should have a 'male' connector, and modems a 'female' connector. This is not, unfortunately, as reliable a guide as it should be, as many manufacturers of terminals and printers equip them with 'female' connectors, mostly for reasons of electrical safety.

If in doubt, the ultimate test is to examine the user manual and determine the function of PIN 2 - if the description includes the word "TRANSMIT" then the equipment is wired as a terminal, and if it includes the word 'RECEIVE' then the equipment is wired as a modem.

Hardware Flow Control

The simplified connection described so far does not allow any control of the data flow. In practice, we often with the receiving device to have control over the transmitting device, thus preventing the receiving device from being overwhelmed (where it is slower in using the input than the rate at which the input is arriving). In addition, if the transmitting device has reason to mistrust the data which it is sending, there should be some provision for it to disable the receiving device.

In the case of modem to terminal connection; when the terminal is able to transmit it activates pin 4 - the RTS pin (Request To Send). When the modem is ready to receive input it activates pin 5 - the CTS pin (Clear To Send). The terminal will only send when CTS is activated. Thus the modem can control the flow rate using CTS.

When the modem considers that the data which it is about to send is suitable, it activates pin 8 - the DCD pin (Data Carrier Detect). When the terminal is ready to receive input it activates pin 20 - the DTR pin (Data Terminal Ready). The modem will only transmit when DTR is activated. Thus the terminal can control flow rate using DTR.

There are two further signals which must be introduced here. One is on pin 22 - the Ring indicator, which simply allows the modem to tell the terminal that the phone is ringing (at which point software in the terminal might be expected to wake up). The other signal is on pin 6 - DSR (Data Set Ready). This signal is ignored by the receiving side of the RS232C; the modem will activate this signal at much the same time that it activates DCD, and therefore no functionality is lost by ignoring DSR.

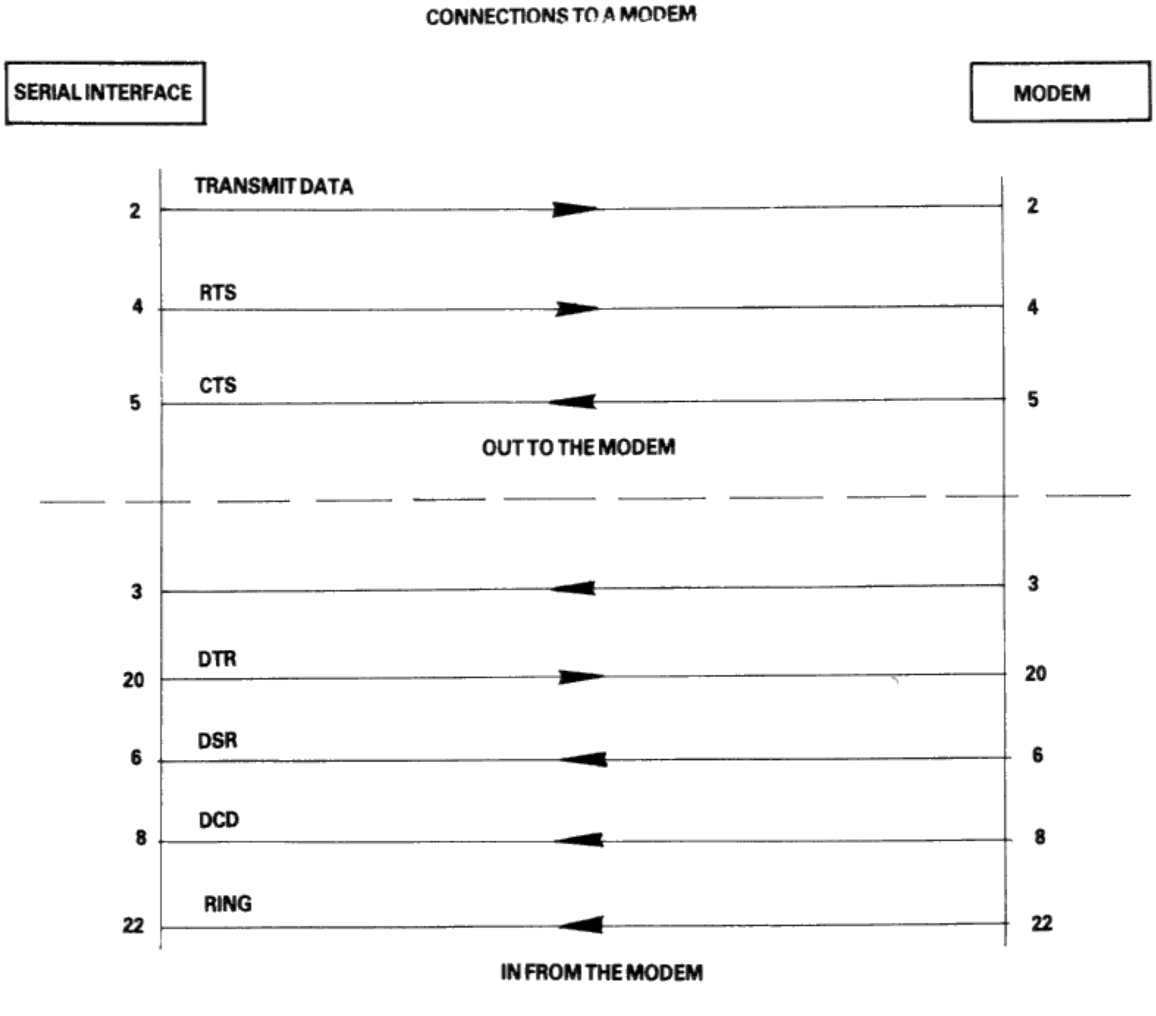


Fig 6

In the case of terminal-to-terminal conenctions, the Null-modem cable must be used with the additional connections to pins 2, 3, and 7 as already discussed. The full Null-modem cable swaps pins 4 and 8 - the RTS/DCD 'I am happy to send' signals, and pins 20 and 5 - the DTR/CTS 'busy' signals. To be on the safe side, pin 6 (DSR) is connected to pin 8 (DCD) in case that end of the cable is ever connected to a terminal which is fussy and requires DSR as well as DCD.

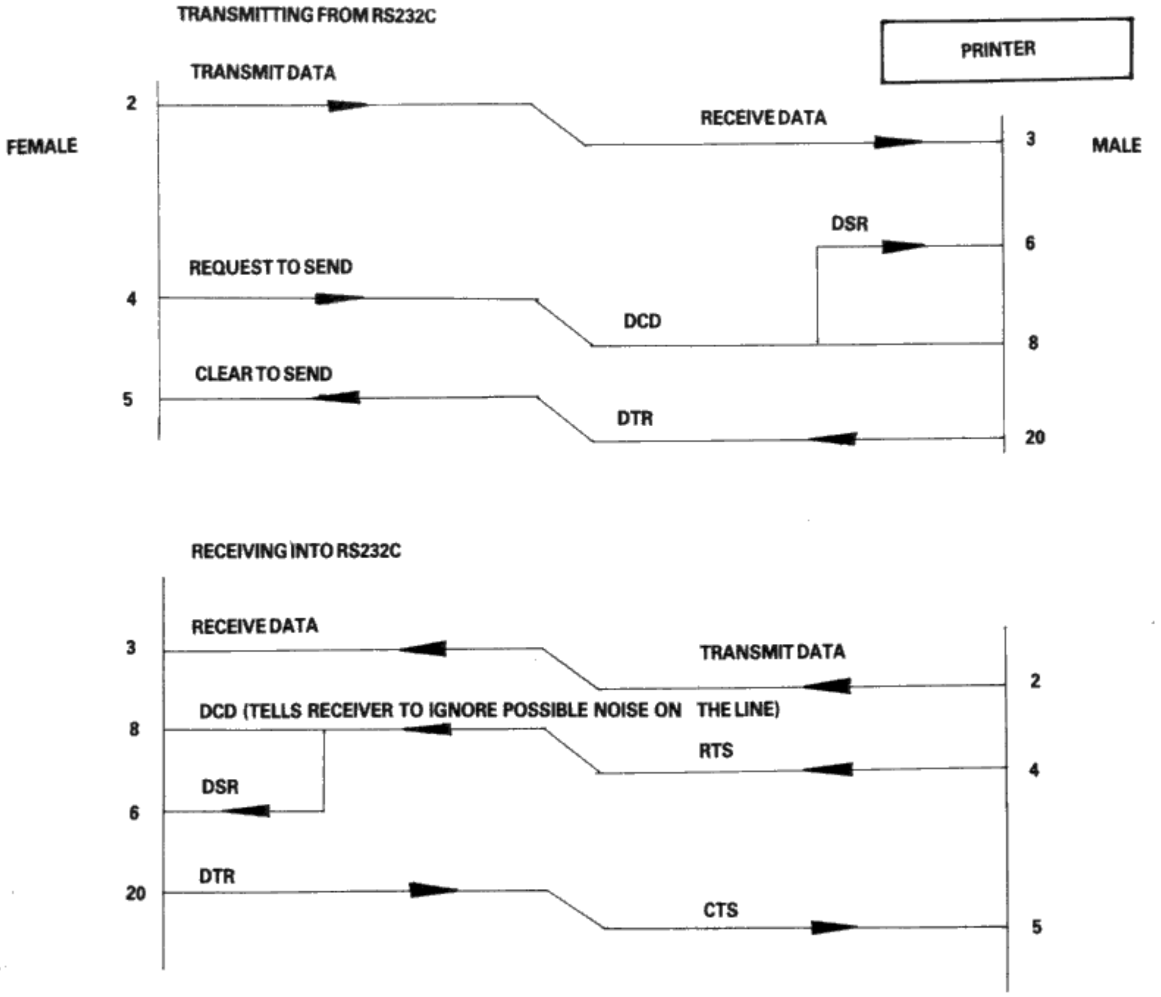


Fig 7

There is a school of thought which says that a Null-modem cable, unlike the pair of modems it replaces, is ALWAYS 'happy to send'. Therefore it is quite in order to generate DCD (and DSR) permanently. This is achieved by connecting them to the RTS at the same end of the cable, rather than to the RTS at the other end of the cable.

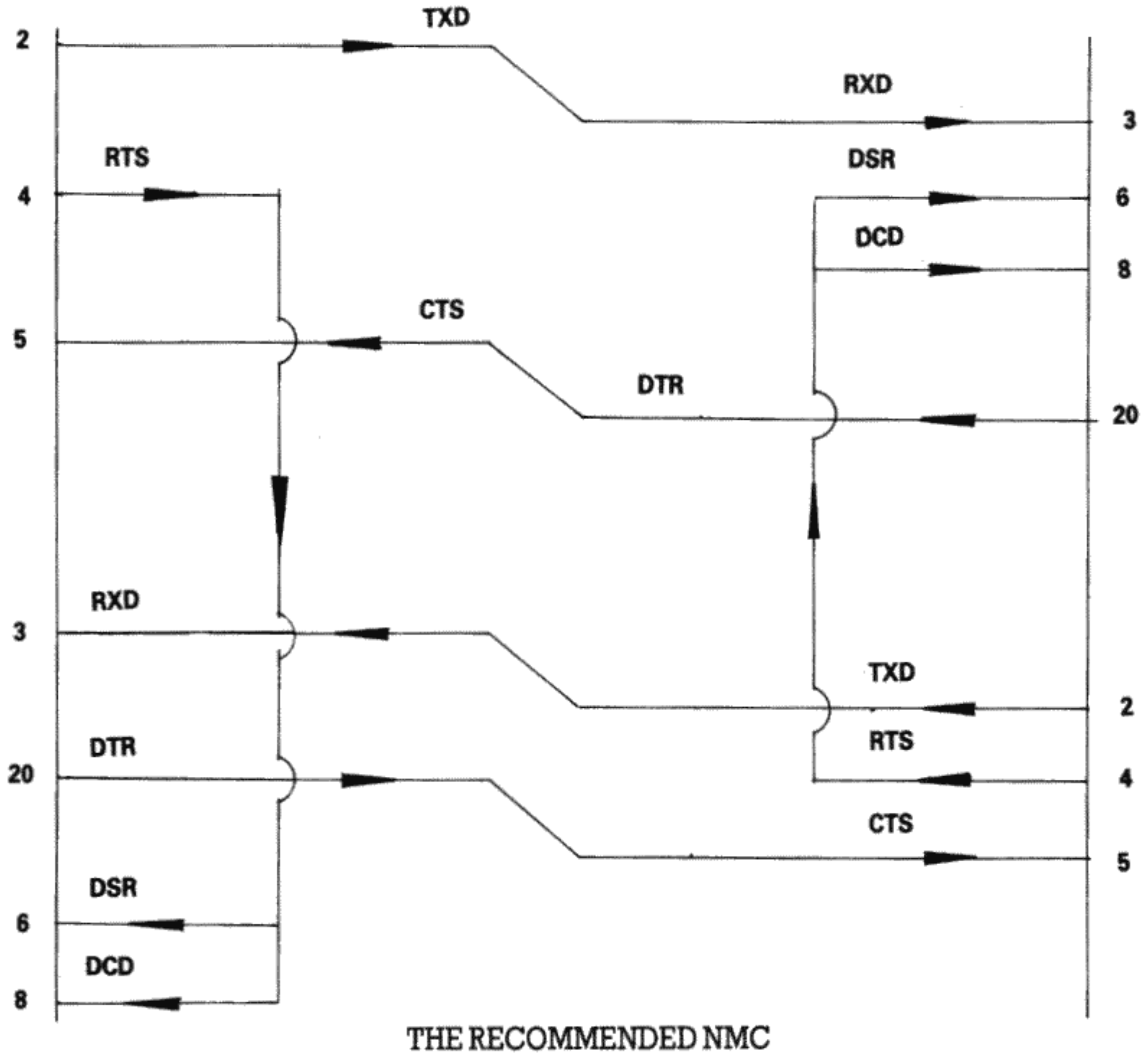
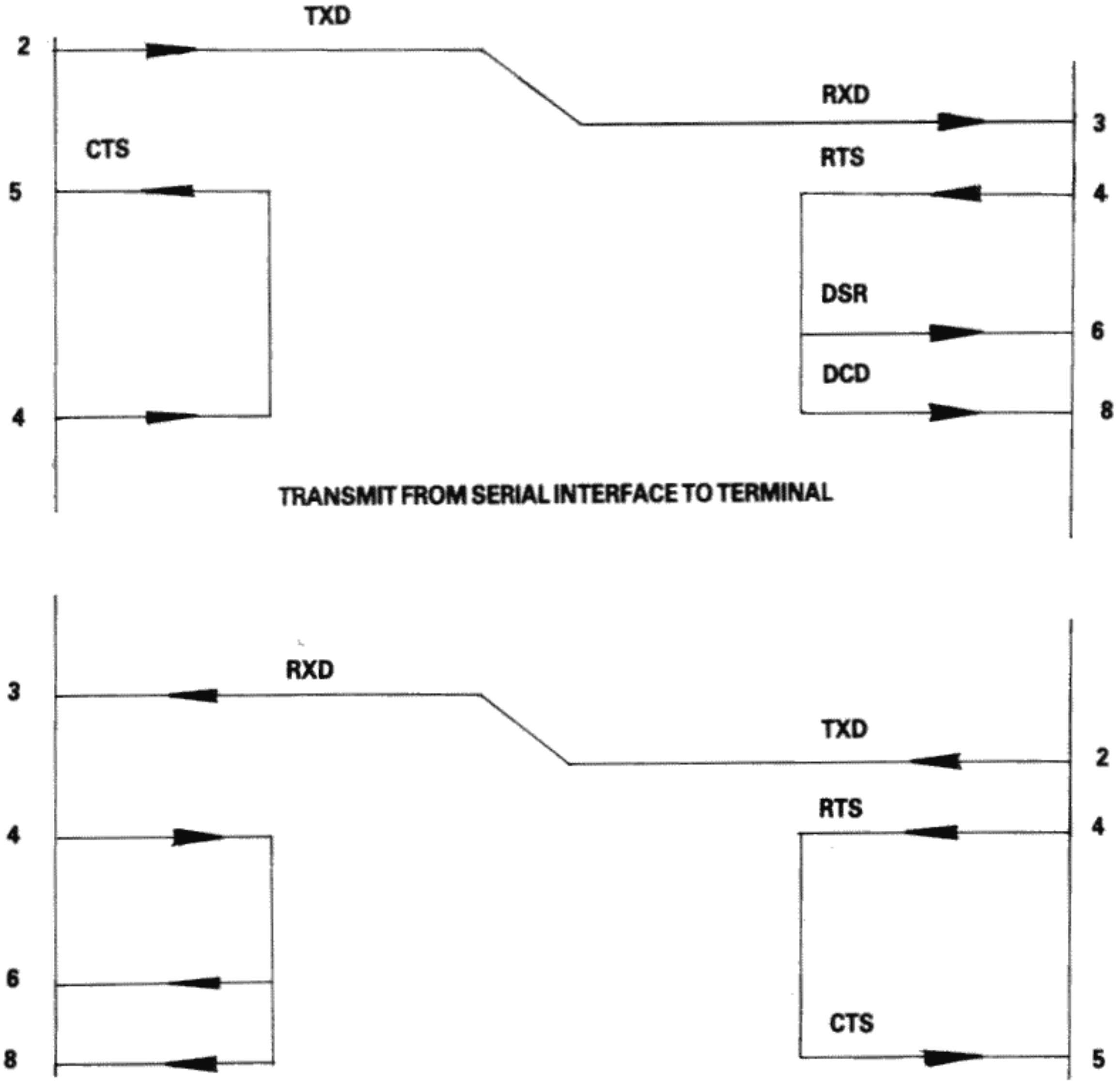


Fig 8

Finally, if the transmission rate from one of the two terminals is known to be unstoppable (e.g. a person typing at the keyboard), or is so slow and infrequent (e.g. the software handshake characters 'XON, XOFF' sent by the printer) that there is no danger of over-running the receiving end, then it is permissible to permanently enable the transmission by linking pin 5 (CTS) to pin 4 (RTS), i.e. to always send if ready (at the transmitting end of the cable). It may well be facilitated in any case, for the transmitting terminals to ignore the state of CTS under these circumstances.



THE RECOMMENDED TERMINAL CABLE

Fig 9

Appendix 5: Printer Lead (PL-2) Wiring Specification

Connectors.

1. Computer Centronics Parallel Interface connector is a 25-way, D-plug.
2. Printer Input connector is a 36-way, IEEE-488 plug.

Cable Wiring.

Line Name	**Computer Connector**	**Printer Connector**
Strobe	1	1
Data Bit 0	2	2
Data Bit 1	3	3
Data Bit 2	4	4
Data Bit 3	5	5
Data Bit 4	6	6
Data Bit 5	7	7
Data Bit 6	8	8
Data Bit 7	9	9
Ack	10	10
Busy	11	11

Line Name	Computer Connector	Printer Connector
PO	12	12
Slct Out	13	13
AutoFd	14	14
Error	15	32
Reset	16	31
Slct In	17	36
GND	18	19
GND	19	20
GND	20	21
GND	21	22
GND	22	23
GND	23	24
GND	24	25
GND	25	26
GND		27
GND		28
GND		29
GND		30
GND		33
GND		15
GND		16
GND		17
GND		18
GND		34
GND		35

Appendix 6: Power Supply Requirements

The following is a summary of the power supply requirements for the Amstrad PC 1512.

Main Electronics Board (Including Keyboard)

+5V 2.80 Amp
+12V 0.15 Amp
-12V 0.10 Amp
-5V 0.00 Amp

Expansion Slots Allowance (Three Slots)

+5V 2.70 Amp
+12V 0.15 Amp
-12V 0.15 Amp
-5V 0.10 Amp

Floppy Disc Drive (Single)

+5V 0.60 Amp
+12V 1.30 Amp
-12V 0.00 Amp
-5V 0.00 Amp

Hard Disc Drive

+5V 1.30 Amp
+12V 2.00 Amp (2.50 Amp during Motor Start-Up)
-12V 0.00 Amp
-5V 0.00 Amp

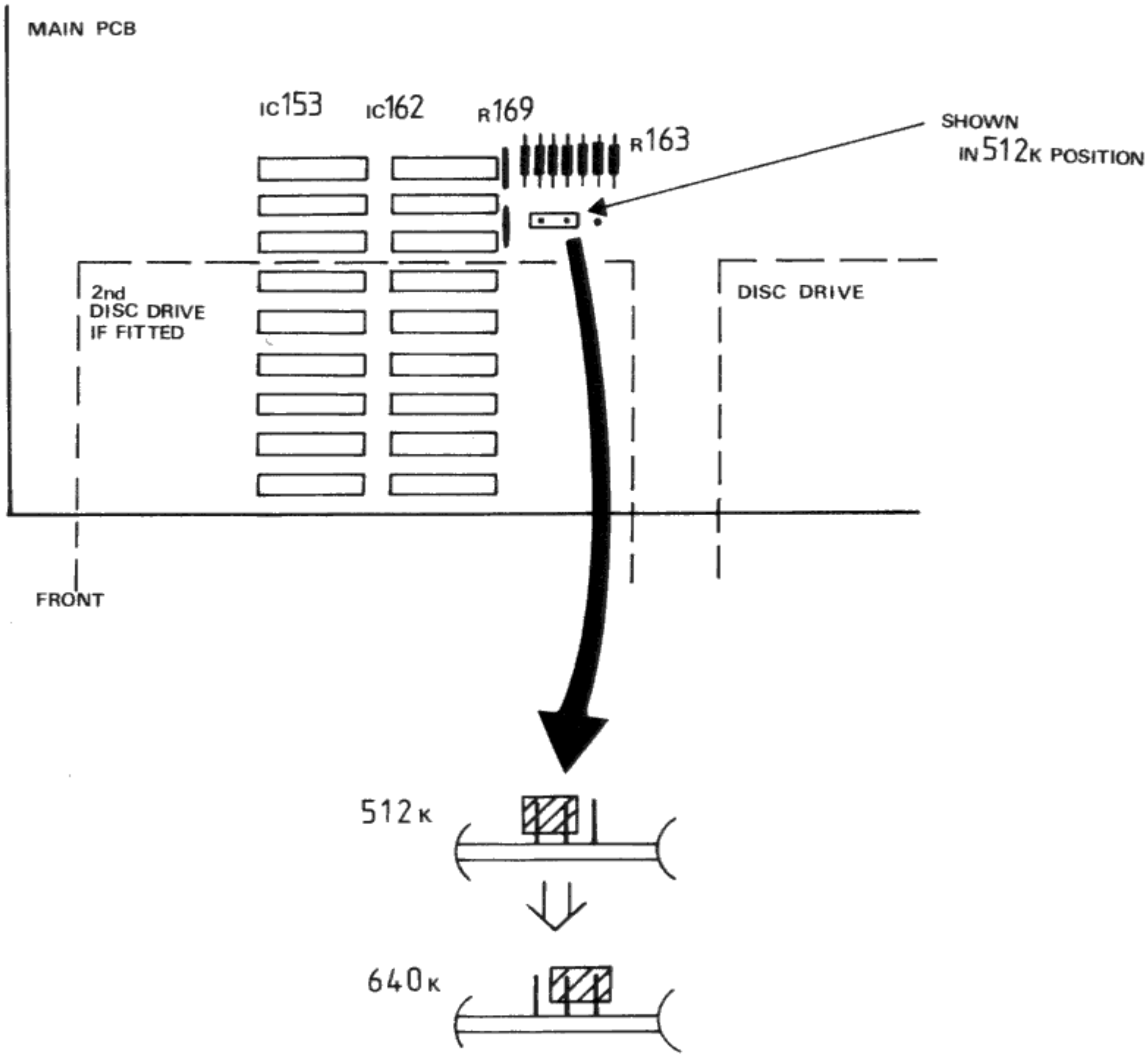
This results in the following total for a worst case configuration:

System with 2 floppy drives and Hard Disc

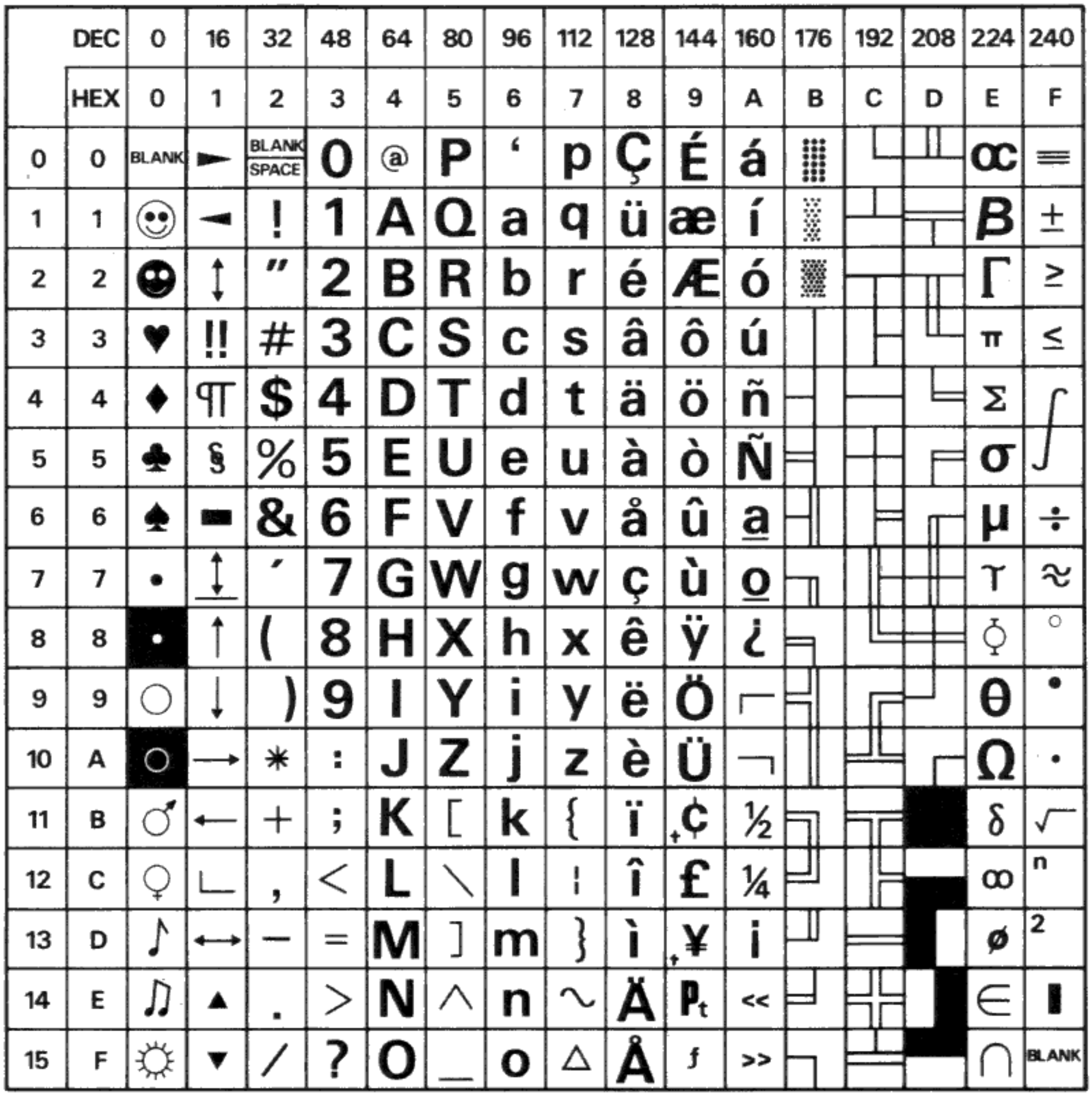
+5V 7.00 Amp
+12V 4.90 Amp
-12V 0.24 Amp
-5V 0.10 Amp

Appendix 7: 512K to 640K conversion

Use 18 pcs 64K - 150 ns Drams plugged into 18 I.C. sockets provided, numbered IC153 to IC170 on main pcb. A link switch must then be moved into the 640K position. (see link diagram). It is important that this switch position is changed before unit is switched on. Please note if 2nd drive has been fitted, it needs to be removed while adding the extra memory.



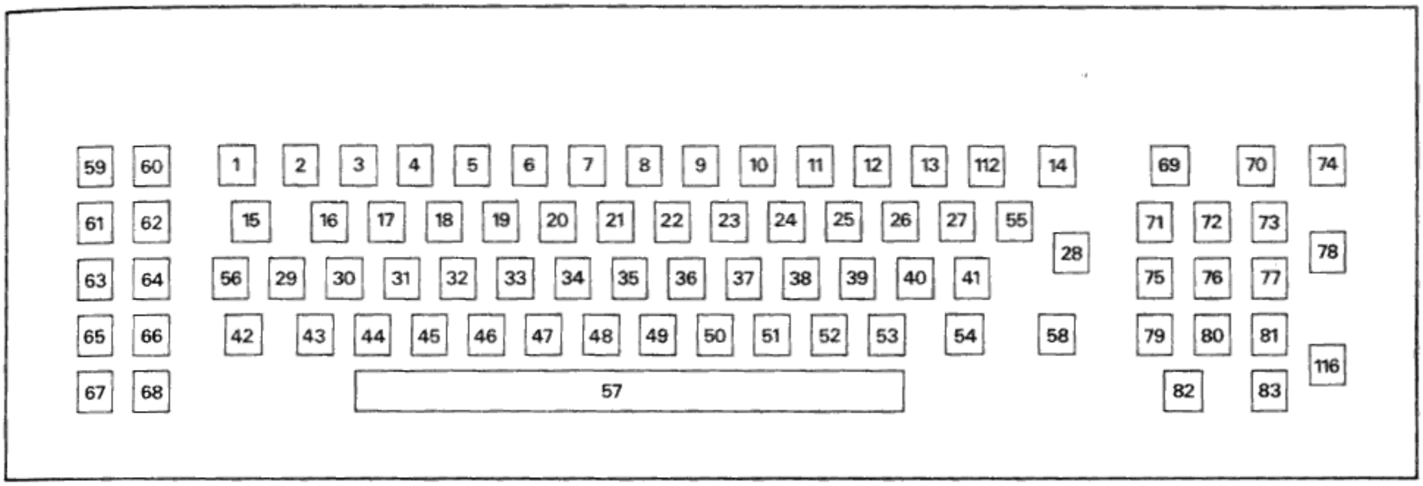
Appendix 8: ROM Character Set

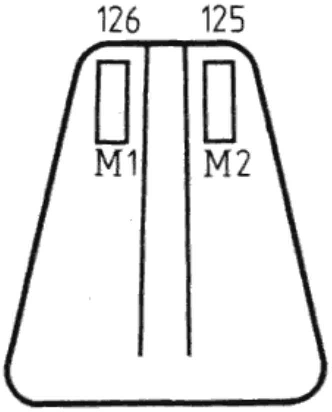


	DEC	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
	HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	BLANK	►	BLANK SPACE	0	@	P	‘	p	Ç	É	á	░	└	╨	α	≡
1	1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
2	2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7	7	•	↨	´	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
11	B	♂	←	+	;	K	[	k	{	ï	†¢	½	╗	╦	█	δ	√
12	C	♀	∟	,	<	L	\	l	¦	î	£	¼	╝	╠	▄	∞	^{n}
13	D	♪	↔	–	=	M	]	m	}	ì	†¥	¡	╜	═	▌	ø	^{2}
14	E	♫	▲	.	>	N	^	n	~	Ä	₧	<<	╛	╬	▐	∈	■
15	F	☼	▼	/	?	O	_	o	△	Å	ƒ	>>	┐	╧	▀	∩	BLANK

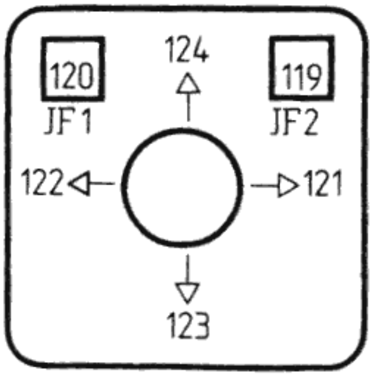
† Note that in the optional Danish character set, the character ¢ and ¥ are replaced by ø and Ø respectively.

Appendix 9: Keyboard Keycodes



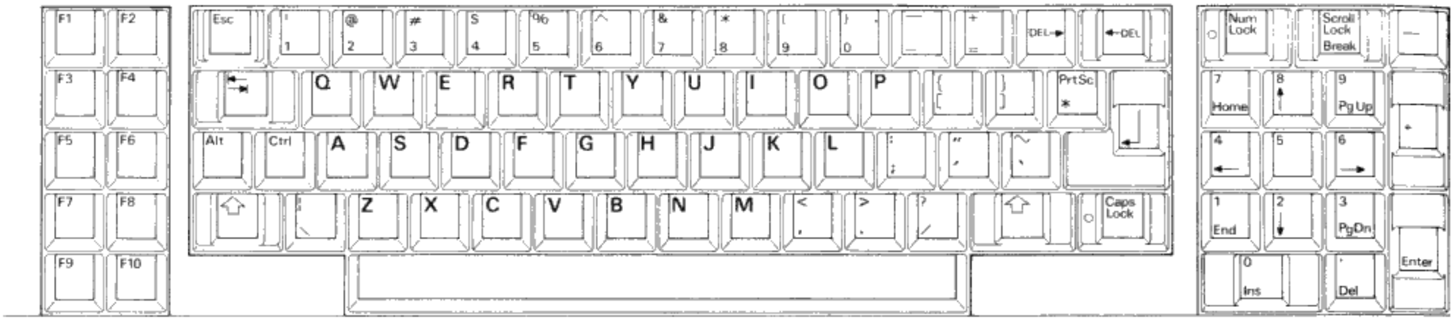


Mouse

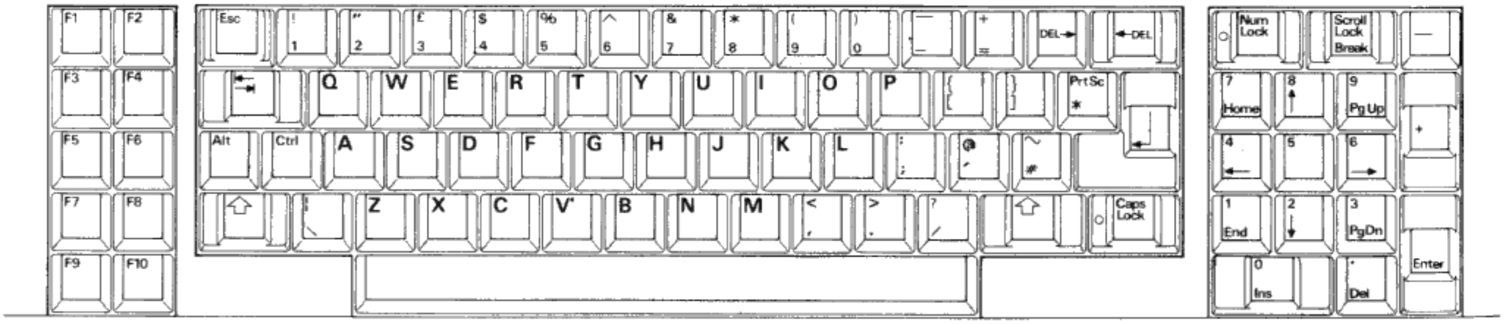


Joystick

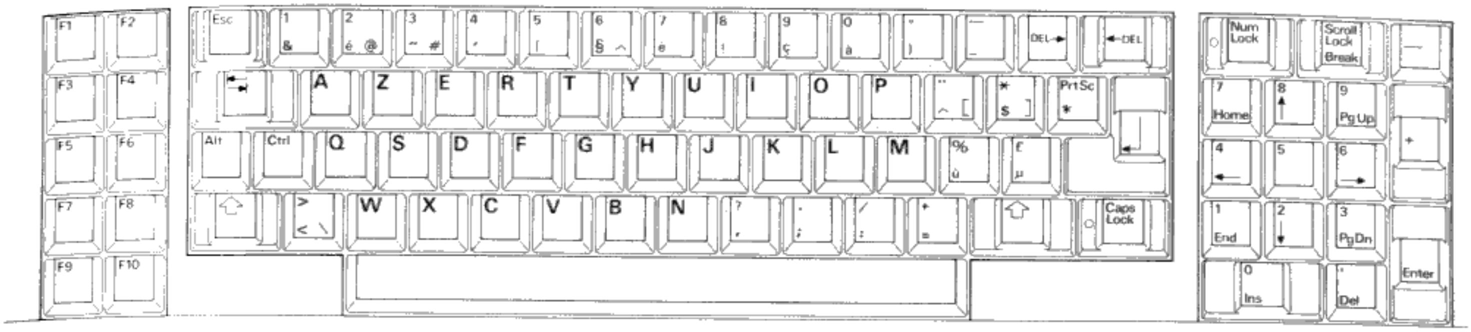
Appendix 10: Keyboard Layouts


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
Esc ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — + = DEL→ ←DEL
Q W E R T Y U I O P { [}] PrtSc *
Alt Ctrl A S D F G H J K L : ; " ' ~ `
Z X C V B N M < , > . ? / Caps Lock
Num Lock Scroll Lock Break —
7 Home 8 9 Pg Up 4 5 6 + 1 End 2 3 PgDn 0 Ins . Del Enter

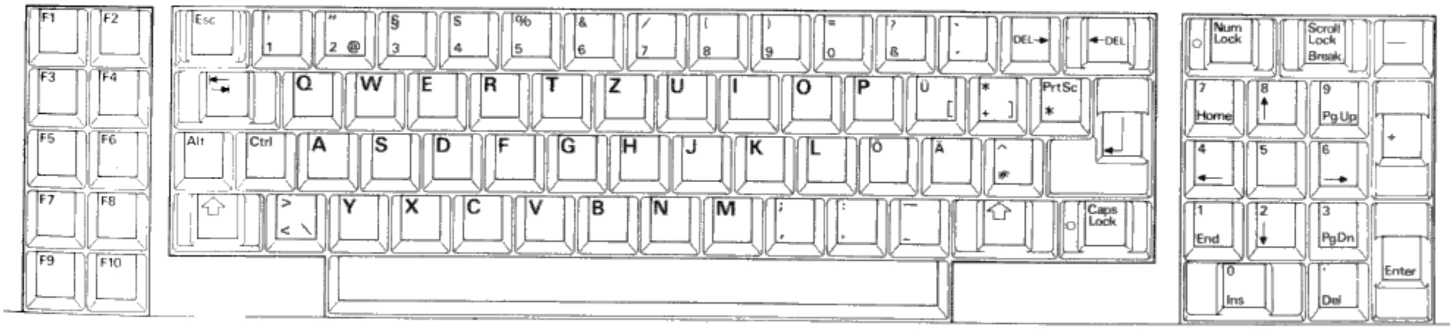
USA keyboard


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
Esc ! 1 " 2 £ 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — + = DEL→ ←DEL
Q W E R T Y U I O P { [}] PrtSc *
Alt Ctrl A S D F G H J K L : ; @ ' ~ #
Z X C V B N M < , > . ? / Caps Lock
Num Lock Scroll Lock Break —
7 Home 8 9 Pg Up 4 5 6 + 1 End 2 3 PgDn 0 Ins . Del Enter

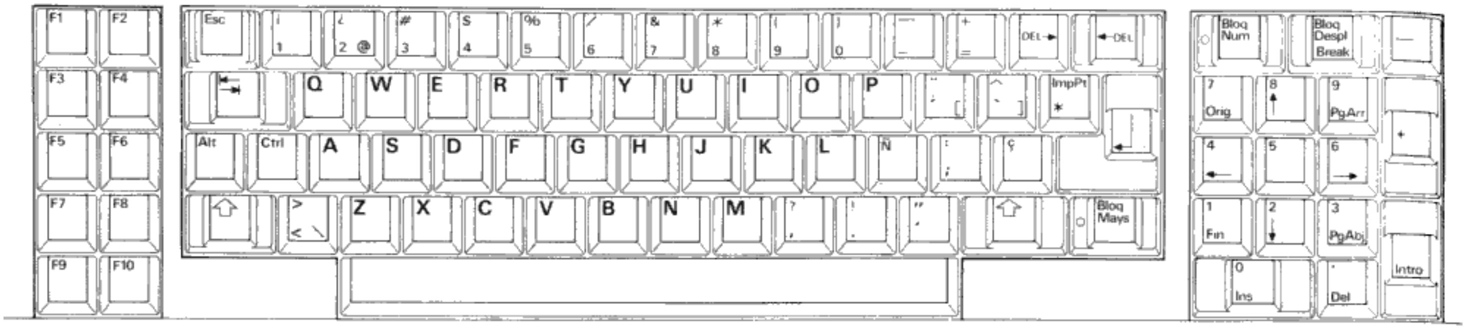
UK keyboard



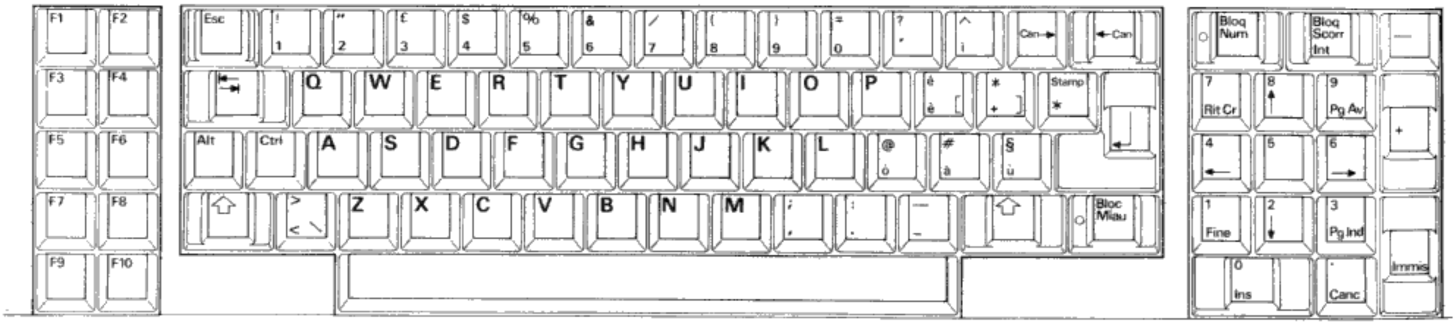
FRENCH keyboard



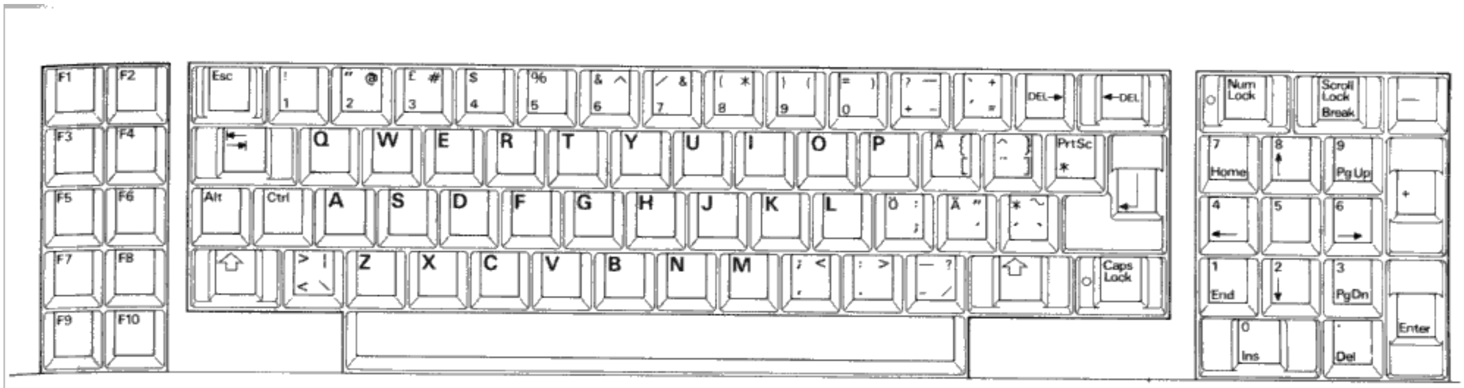
GERMAN keyboard


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
Esc ¡ 1 ¿ 2 @ # 3 $ 4 % 5 / 6 & 7 * 8 (9) 0 — _ + = DEL→ ←DEL
Q W E R T Y U I O P ¨ [^] ImpPt *
Alt Ctrl A S D F G H J K L Ñ : ; ç
> < \ Z X C V B N M ? , ! . " ' Bloq Mays
Bloq Num Bloq Despl Break —
7 Orig 8 9 PgArr +
4 5 6
1 Fin 2 3 PgAbj Intro
0 Ins . Del

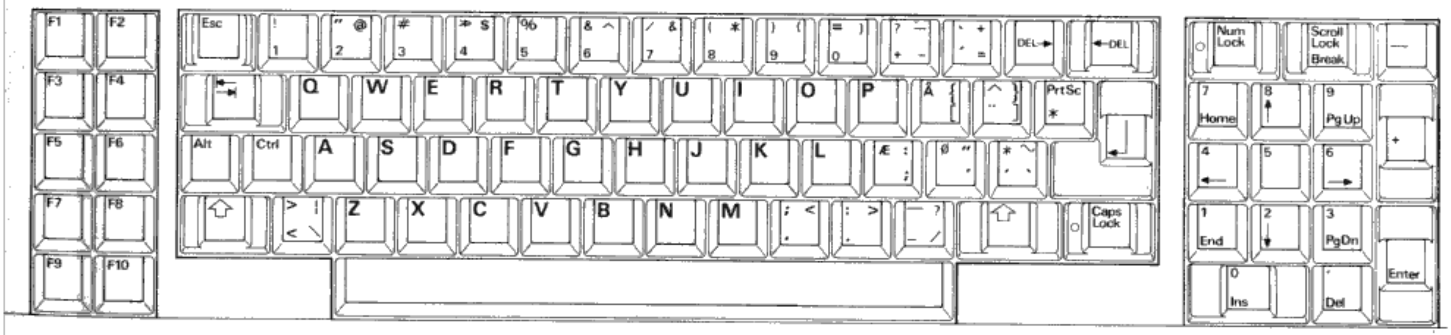
SPANISH keyboard


F1 F2 F3 F4 F5 F6 F7 F8 F9 F10
Esc ! 1 " 2 £ 3 $ 4 % 5 & 6 / 7 (8) 9 = 0 ? ' ^ ì Can→ ←Can
Q W E R T Y U I O P é è [* +] Stamp *
Alt Ctrl A S D F G H J K L @ ò # à § ù
> < \ Z X C V B N M ; , : . — _ Bloc Miau
Bloq Num Bloq Scorr Int —
7 Rit Cr 8 9 Pg Av +
4 5 6
1 Fine 2 3 Pg Ind Immis
0 Ins . Canc

ITALIAN keyboard



SWEDISH keyboard



DANISH keyboard

INDEX

Write System Status-2	20

Appendices Site Index