

The logo features the word "AMSTRAD" in a bold, white, sans-serif font. The letters are set against a dark grey rectangular background. A prominent red diagonal stripe cuts across the background from the top right to the bottom left, passing behind the text.

**AMSTRAD**

**CPC6128+**

**MANUAL**

The product described in this manual and products for use with it are subject to continuous development and improvement.

This manual is provided to you free of charge and is intended only to assist the reader in the use the product. The information contained in this manual and literature provided with product is given by AMSTRAD in good faith.

AMSTRAD MAKES NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND AMSTRAD ACCEPTS NO LIABILITY FOR ANY LOSS OR DAMAGE ARISING FROM THE USE OF ANY INFORMATION PROVIDED OR OMITTED.

This manual may include technical inaccuracies or typographical errors. Changes are periodically made to the information contained herein; these changes will be incorporated in new editions of the publications. AMSTRAD may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time but does not undertake to notify customers of these changes.

The products referred to herein are not designed and should not be used for, or in connection with life critical functions or any activity in which an error or a fault may result in physical damage or injury to person(s) and AMSTRAD does not authorise such use.

You must carefully read these instructions and all other literature provided with the product, including the Guarantee Card, prior to operation.

## **IMPORTANT**

The terms and instructions, contained in the Guarantee Card provided with the product, must be strictly complied with.

## **EXCLUSIONS OF CONSEQUENTIAL LOSS**

IN ANY EVENT, AMSTRAD ACCEPTS NO LIABILITY FOR ANY CONSEQUENTIAL LOSS OR DAMAGE ARISING FROM THE USE OF THIS MANUAL OR PRODUCTS USED WITH IT OR ANY INFORMATION PROVIDED IN OR OMITTED FROM THIS MANUAL INCLUDING, BUT NOT LIMITED TO, ECONOMIC OR FINANCIAL LOSS, DAMAGE TO PERIPHERAL EQUIPMENT OR PRODUCTS, LOSS OF USE, PRODUCTIVITY OR TIME.

**Nothing in this document is intended to exclude or restrict any of the consumer's rights where to do so would be illegal.**

All correspondence relating to the product or this manual should be addressed to:

AMSTRAD PLC  
BRENTWOOD HOUSE  
169 KINGS ROAD  
BRENTWOOD  
ESSEX, CM14 4EF  
ENGLAND

©1990 AMSTRAD plc ENGLAND.

---

Neither the whole nor any part of the information contained herein, nor the products described in this manual may be adapted or reproduced in any material form except with the prior written approval of AMSTRAD plc.

**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 guide is intended only to assist the reader in the use of the product, and therefore AMSTRAD shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in this guide or any incorrect use of the product.**

CP/M and CP/M Plus are trademarks of Digital Research Inc.

Z80 is a trademark of Zilog Inc.

The name and logo AMSTRAD is the registered trademark of AMSTRAD PLC.

AMSDOS, CPC6128, CPC664, CPC464, 6128, 464, 6128 Plus and 464 Plus are trademarks of Amstrad plc.

Unauthorised use of the name or logo AMSTRAD is strictly forbidden.

First Published 1990.

Compiled by Roland Perry

Published by AMSTRAD PLC

PRINTED IN ENGLAND

---

## **You must read this....**

### **Installation Notes**

1. Always connect the Mains Lead to a 3-pin plug following the instructions contained in part 1 of the Foundation course.
2. Never attempt to connect the system to any Mains Supply other than 220-240V ~ 50Hz.
3. There are no user serviceable parts inside the system. Do not attempt to gain access into the equipment. Refer all servicing to qualified service personnel.
4. To avoid eye-strain, the monitor should be placed as far away as possible from the keyboard and operated in an adequately lit room. The monitor BRIGHTNESS control should be kept to as low a setting as possible.
5. Do not block or cover any ventilation holes.
6. Do not use or store the equipment in excessively hot, cold, damp, or dusty areas.
7. Do not attempt to insert or remove the cartridge while the power is switched on.
8. Do not attempt to operate the computer without a cartridge installed.

### **Compatibility**

The 464 Plus and 6128 Plus may be used with most software and peripherals intended for the earlier CPC464, CPC664 and CPC6128 models. While every effort has been made to retain compatibility it is inevitable that some software written for the CPC464 and CPC6128 models may not operate correctly with the 464 Plus and 6128 Plus models. Before buying software or peripherals intended for the earlier models you should check with your dealer that they are suitable for use with the 464 Plus or 6128 Plus.

NB. The connectors for Expansion peripherals, Printers and Second Disc drives have the same function as those on the earlier models, but may have a different connector.

---

## Special Disc Operation Notes (6128)

(Don't worry if you are a little baffled by the technical jargon in this section; the importance of these warnings will become clearer as you work through this manual.)

1. Never switch the 6128 on or off with a disc in the drive. Doing so will corrupt your disc, losing valuable programs or data.
  2. Always make back-up (duplicate) copies of discs which contain valuable programs. It is especially important to make back-up copies of the master CP/M system disc provided with the 6128. Should you otherwise accidentally lose or corrupt your disc, replacing it could prove expensive.
  3. Make sure that you do not accidentally overwrite your master CP/M system disc, by ensuring that the Write Protect holes on the disc are always open.
  4. For maximum data reliability, the disc drive section of the computer should NOT be placed directly in front of the monitor, but to the right of it. Do not place the computer close to any source of electrical interference.
  5. Never touch the floppy disc surface itself, inside its protective casing.
  6. Do not eject a disc while it is being read from or written to.
  7. Always remember that formatting a disc will erase any previous contents.
  8. Always keep disc drives and discs away from magnetic fields.
  9. The licence agreement for your CP/M system disc, (which is electronically serial-number encoded) permits its use on a single computer system only. In particular this means that you are prohibited from giving any other person a disc with YOUR serial-numbered copy of CP/M on it. Carefully read the Software License Agreement (Appendix 1), towards the end of this manual.
-

## **Chapter 1**

### **Foundation Course**

Setting up  
Connecting peripherals  
Floppy discs and cassettes  
Keyboard familiarisation  
Loading software  
Introduction to BASIC keywords  
Saving programs  
Modes colours and graphics  
Sound  
Introduction to AMSDOS and CP/M  
Introduction to the Bank Manager

## **Chapter 2**

### **Beyond Foundations**

Writing a simple program  
Evolution and afterthoughts  
Using an array  
Introducing a menu  
Loading and saving variables to disc or cassette  
Editing and line numbering  
Tidying-up a program

## **Chapter 3**

### **Complete List of AMSTRAD 464/6128 BASIC Keywords**

Description of notation used  
Alphabetical listing of keywords comprising:  
    Keyword  
    Formal syntax  
    Example  
    Description  
    Special notes (where applicable)  
    Associated keywords

---

## **Chapter 4**

### **Using Discs (6128 only)**

Backup master disc  
Getting started with CP/M Plus  
Single and multiple drive operation  
Copying files  
BASIC disc applications

## **Chapter 5**

### **AMSDOS and CP/M**

AMSDOS:

- Introduction to AMSDOS
- Disc directory
- Filenames filetypes and headers
- Wildcards
- Summary of AMSDOS commands
- Manipulating and copying files
- Reference guide to error messages

CP/M:

- Introduction to CP/M
- Booting CP/M Plus
- Direct Console Mode
- Transient programs
- Managing peripherals

## **Chapter 6**

### **For Your Reference....**

Cursor locations and control code extensions  
Interrupts  
ASCII and graphics characters  
Key references  
Sound  
Error messages  
BASIC keywords  
Planners  
Connections  
Printers

---

## **Chapter 6 continued....**

- Joysticks
- Disc organisation
- Resident System eXtensions (RSX's)
- Memory
- CP/M Plus Terminal Emulator
- CP/M Plus character set

## **Chapter 7**

### **More about the Bank Manager....**

Using the second 64K of memory

Storing screen images:

- Hardware bank switching
- Swapping screens
- Copying screens

Pseudo-file operation:

- Creating a RAMdisc
- Current record
- Reading, writing, and searching for strings
- RAMfile example program

## **Chapter 8**

### **At Your Leisure....**

General:

- The world of microcomputers
- Hardware and software
- Comparing computers
- Some popular misconceptions
- How a computer deals with your instructions
- The digital world
- Bits and bytes
- The BINARY number system

---



## **Chapter 8 continued....**

464/6128 Specific functions:

- Character set
- Variables
- Logic
- User defined characters
- Print formatting
- Windows
- Interrupts
- Data
- Sound
- Graphics
- Graphics using the second 64K of memory
- Screen Designer program

## **Appendices**

Appendix 1 End User Program Licence Agreement

Appendix 2 Glossary of Terms

Appendix 3 Some Programs For You....

Bustout

Bomber

Telly tennis

Electric fencing

Amthello

Raffles

Appendix 4 Index

---

# Chapter 1

## Foundation Course

### Part 1: Setting Up....

The 464/6128 can be set up with either:

1. The Amstrad MM12 Monochrome Monitor
2. The Amstrad CM14 Colour Monitor

### Fitting a Mains Plug

The 464/6128 operates from 220-240 Volt AC 50Hz Mains supply.

Fit a proper mains plug to the mains lead of either the MM12 or CM14. If a 13 Amp (BS1363) plug is used, a 5 Amp fuse must be fitted. The 13 Amp fuse supplied in a new plug must NOT be used. If any other type of plug is used, a 5 Amp fuse must be fitted either in the plug or adaptor or at the distribution board.

**WARNING: THIS APPARATUS MUST BE EARTHED**

### Important

The wires in the mains lead are coloured in accordance with the following code:

Green-and-yellow	:Earth
Blue	:Neutral
Brown	:Live

As the colours of the wires in the Mains Lead of this apparatus may not correspond with the coloured markings identifying the terminals in your plug, proceed as follows:

The wire which is coloured GREEN-AND-YELLOW must be connected to the terminal which is marked by the letter 'E' or by the safety earth symbol or coloured green or green-and-yellow.

The wire which is coloured BLUE must be connected to the terminal which is marked with the letter 'N' or coloured Black.

The wire which is coloured BROWN must be connected to the terminal which is marked with the letter 'L' or coloured Red.

Disconnect the Mains Plug from the Supply Socket when not in use.

Do not attempt to remove any screws, nor open the casing of the computer or monitor. Always obey the warning on the rating label which is located underneath the case of the computer and on the rear cabinet of the monitor.

**WARNING - LIVE PARTS INSIDE. DO NOT REMOVE ANY SCREWS**

# Setting up the 464/6128

Place the computer on a suitable flat surface.

**IMPORTANT:** When choosing a position for the computer:

- make sure that it is installed close to, and within easy access of, the electrical mains supply socket.
- make sure that it is **NOT** situated near an artificial heat source, such as a radiator, near a water supply, nor in direct sunlight. These elements can damage your computer.

1. Make sure that the monitor is **NOT** plugged into the mains supply socket.
  2. Connect the lead from the front of the monitor, which is fitted with the larger (8-pin DIN) plug, into the 8-pin DIN socket on the rear of the computer.
  3. Connect the lead from the front of the of the monitor, which is fitted with the smaller (5V DC) plug, into the power socket on the rear of the computer.
-

4. Insert the supplied cartridge into the socket on the left-hand side of the computer.  
**IMPORTANT:** The cartridge can only be inserted or removed when the sliding power switch is in the OFF position.
5. Insert the mains plug into the supply socket and switch on.
6. Press the Power button on the front of the monitor, so that it is set to the IN position.  
When the button is set to the OUT position, the Mains supply to the system is OFF.
7. Switch on the computer using the Power switch on the left hand side. Sliding it towards you turns ON the power and sliding it away from you turns OFF the power.

When operating correctly the computer and monitor will display the following picture:

To avoid unnecessary eye-strain, adjust the controls for Brightness and Contrast on the lower front panel of the monitor. There is also a volume control to adjust the volume of the built-in loudspeakers.

---

## Selecting BASIC or the Demonstration Game

The cartridge supplied with your 464/6128 includes both AMSTRAD BASIC and a Demonstration Game. To operate the AMSTRAD BASIC press the key with [f1] written on it (which is found towards the right-hand side of the keyboard) within 30 seconds of switching on the computer. If you press the key with [f2] written on it, or do not press the key with [f1] written on it within 30 seconds, then the Demonstration Game will be played.

Complete instructions on how to operate the AMSTRAD BASIC are contained in later sections of this manual.

NB. To play the demonstration game, after you have operated AMSTRAD BASIC, type in the command |GAME and then press the **[ENTER]** key. The | symbol is typed by holding down either of the **[SHIFT]** keys and then pressing the key with the @ and | symbols on it.

---

# Part 2: Connecting your peripherals....

This section explains how the various peripherals, or add-ons, are connected to the 464/6128 system. Details concerning the use of these devices will be found in the appropriate sections of this manual. A diagram identifying the various sockets is printed in Chapter 6, Part 9.

**IMPORTANT: DO NOT CONNECT OR DISCONNECT ANY PERIPHERALS WHILE THE POWER TO YOUR COMPUTER IS SWITCHED ON.**

## Joystick/Paddle Controller (PD-1)

The 464/6128 is supplied with one Paddle Controller. In addition a second Joystick or Paddle Controller is an item that you may wish to purchase if you are using the 464/6128 computer with games software which incorporates the facility for using more than one Joystick/Paddle Controller.

Connect the plug fitted to the supplied Paddle Controller lead into the frontmost Joystick Socket on the side of the computer. The 464/6128 can be used with two Joysticks or Paddle Controllers. The second one is fitted to the rearmost Joystick Socket on the computer.

Further information on writing your own software to use Joysticks will be found later in this manual.

## Light Pen/Light Gun

A Light Pen or Light Gun is an additional item that may be available if you are using the 464/6128 computer with games which incorporates the facility for a Light Gun or other software that incorporates the facility for a Light Pen.

Connect the plug fitted to the Light Gun or Light Pen lead into the AUXiliary Socket on the side of the computer.

These peripherals can only be used with specially written software supplied on tape, disc or cartridge.

---

## **Analog joystick**

An Analog Joystick is an additional item that you may wish to purchase if you are using the 464/6128 computer with games software which incorporates the facility for analog joystick control and "firing". An Analog Joystick provides more sophisticated control of the game than a normal Joystick or Games Controller.

Connect the plug fitted to the Analog Joystick lead into the Analog Joystick Socket on the side of the computer.

An Analog Joystick can only be used with specially written software, which will normally only be available on cartridge.

## **Printer**

The 464/6128 can be used with any Centronics compatible parallel printer. You will require a suitable lead, which may be included with the printer.

Connect the end of the lead which is fitted with a 25 pin connector plug into the Printer Socket on the rear of the computer. Connect the other end of the lead which is fitted with a Centronics Style 36 pin plug into the socket on the rear of the printer. If the printer is equipped with security clips at each side of the socket, these may be clipped into the cut-outs at the side of the printer plug.

**DO NOT** plug the lead into the 36 pin Centronics Connector on the rear of the 6128. That is intended only for attaching a second disk drive.

Details of printer operation will be found later in this manual.

---

## **External Amplifier/Speakers**

The 464/6128 may be connected to an external Stereo Amplifier. The input lead to your Stereo Amplifier should be terminated with a 3.5mm stereo jack plug which should be inserted into the Stereo Socket on the rear of the computer.

The computer will provide a fixed voltage signal out of the Stereo Socket and you should therefore use the controls on the amplifier itself to regulate the volume.

## **Expansion Devices**

There is a general purpose 50-way expansion socket that is provided for add-ons such as serial interfaces, modems etc. Full connection and operation details should be included with the relevant add-on.

## **2<sup>nd</sup> Disc Drive (6128 only)**

There is a socket provided for the attachment of a 2nd Disc Drive. Full instructions for attaching and operating the second drive should be included with the second drive.

---



## **Part 3: About Discs and Cassettes....**

The 464/6128 memory is only able to store data as long as the power is connected to the computer - and the unit itself is switched on - in computer terminology it is a 'volatile' storage medium. If you want to store programs or data when the power is switched off - then this must be stored onto the cassette or disc.

### **Cassette controls (464 only)**

At the right hand rear of the keyboard unit you will find the cassette recorder. The mechanics of this cassette mechanism are essentially the same as those of an audio cassette system - except that the electronics that control the signals are specifically optimised for use with computer data storage systems.

---

The right way to insert a cassette into the recorder.



Similarly, operation of the auxiliary keyboard for the cassette recorder is the same as for most types of audio cassette recorder:

### **The 464 cassette controls**

Note that these control keys require to be pressed considerably harder than the keys on the main 'typewriter-style' keyboard.

**[REC]** = **[REC]**ord (operates simultaneously with **[PLAY]** to record data when instructed by the program to do so). It can not be activated unless a cassette with the 'record protect' tab is placed in the machine, and the door shut.

To activate the function, you must press **[REC]** - and then whilst holding down the **[REC]** key, press the **[PLAY]** key. The computer will then write data onto the cassette when instructed to do so by the program currently in the memory - or by a direct **SAVE** command entered from the keyboard.

**[PLAY]** = **[PLAY]** the tape to either **LOAD** or **RUN** a program from the cassette system. The computer will then read data from the cassette when instructed to do so by the program currently in the memory - or by a direct command entered from the keyboard. Both the **[REC]** and **[PLAY]** keys are reset by a mechanical trip in the mechanism when the cassette tape reaches its end.

**[REW]** = **[REW]**ind the tape from the right hand spool to the left hand spool. There is no mechanical cancellation of this function at the tape end, so you should not leave the machine running in rewind mode or the tape drive motor may overheat when stalled.

**[F.F.]** = Fast Forward will advance the tape rapidly, winding from the left hand spool to the right hand spool. There is no mechanical cancellation of this function at the tape end, so you should not leave the machine running in fast forward mode or the tape drive motor may overheat when stalled.

**[STOP/EJECT]** = Stop any current cassette operation, and reset all the cassette operation keys to the original state. If the key is released and then pressed again, the cassette lid will open, allowing a cassette to be either removed or inserted as required. The cassette cannot be taken out until the operation of the drive has ceased.

**[PAUSE]** = A mechanical pause operating in conjunction with the **[PLAY]** or **[REC]** and **[PLAY]** keys. It should not be used to pause during either a data read or data write operation, or an error will occur. All pauses during play and record operations are handled by internal instructions in the 464 software, leaving the mechanical pause facility largely unused.

---

## Write Protection

In order to prevent accidental erasure of data that you wish to keep on cassette, a mechanical interlock tab is provided on all standard compact cassettes. This tab may be removed by snapping it off using a small pair of pliers etc., whereupon you will be unable to depress the **[REC]** key with the 'write protected' cassette in the drive.

The write protect feature applies individually to EACH SIDE of the cassette - so in order to protect both sides of the cassette, both of the tabs must be removed. If you subsequently wish to re-enable the facility to record on a cassette that has been protected in this way, then you can cover the depression in the cassette housing created by removal of the tab using a piece of self adhesive tape pulled taut across the opening.

There are various techniques available using instructions in the 464 machine operating system to protect programs using software.

## Loading the cassette

The cassette should be fully rewound (from the right hand to the left hand spool) - so if it is not, press the **[REW]** key until the motor stalls at the tape end. If the tape is accidentally looped out of the aperture on the front of the cassette, then you should rewind the tape into the body of the cassette before inserting into the cassette machine or the cassette tape and the information stored on it may be lost.

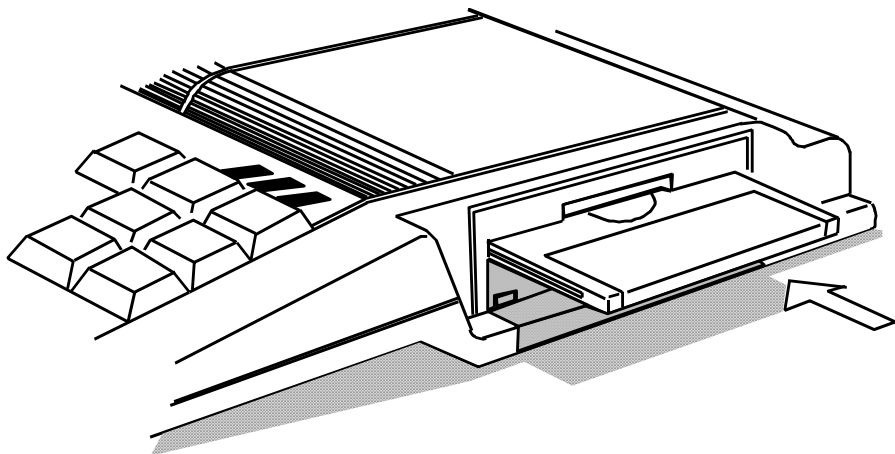
Please note that whilst it is possible to use cassette tapes on audio systems that have suffered various forms of abuse leading to creases and other damage to the tape surface - it is not possible to maltreat a computer data recording in the same way and still expect it to continue to work reliably.

For example, if you damage a tape by trapping a stray loop in the cassette door etc., but find that you are still able to load or run the program from it, you should immediately re-SAVE the program on a fresh undamaged tape (while the program is safe in the 464 memory), and throw the damaged tape away before you are tempted to try and re-use it.

---

## Insertion of Discs (6128 only)

Each side of a disc may be used separately. A disc should be inserted with its label facing outward from the drive, and with the side that you wish to use face up:



## **Write Protection**

In the left hand corner of each side of a blank disc, you will see an arrow pointing to a small shuttered hole. This is called the Write Protect hole, and facilitates protection against erasure or 'overwriting':

When the hole is closed, data can be 'written' onto the disc by the computer. When the hole is open however, the disc will not allow data to be written onto it, thus enabling you to avoid accidental erasure of valuable programs.

Various compact floppy disc manufacturers employ different mechanisms for opening and closing the Write Protect hole. The operation may be carried out on a typical compact floppy disc as follows:

To open the Write Protect hole, slide the small shutter located at the left hand corner of the disc, and the hole will be opened:

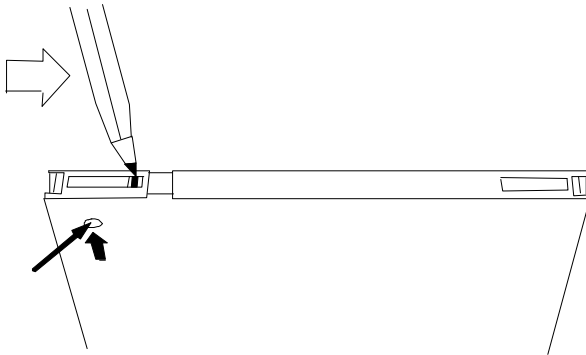
---

To close the Write protect hole, simple slide the shutter to its closed position.

---

Alternatively, other compact floppy discs employ a small plastic lever located in a slot at the left hand corner:

To open the Write Protect hole on this type of disc, slide the lever towards the middle of the disc, using the tip of a ball-point pen or similar object:



Note that regardless of the method employed to open and close the Write Protect hole, opening the hole in all cases facilitates protection against overwriting.

**IMPORTANT**

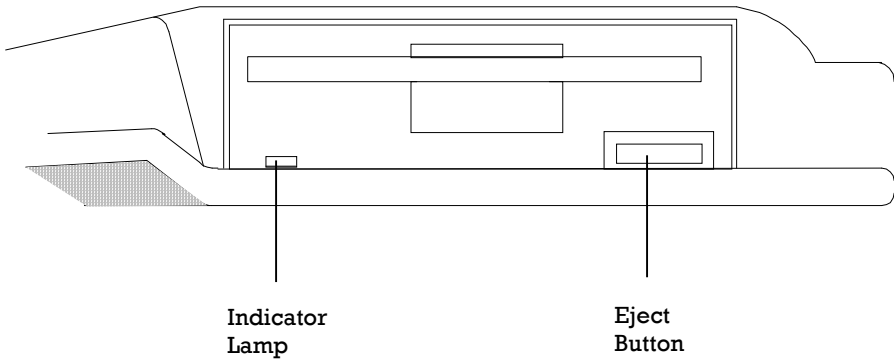
Always ensure that the Write Protect holes on your master CP/M system disc are open.

---



## When Your Disc is in

At the front of the computer's disc drive which you will find in the righthand side of the 6128, you will see a red indicator lamp, and a push button for Eject:



## Indicator Lamp

This indicates that data is being read from, or written to the disc.

If a 2nd disc drive is connected, the red indicator on the 2nd disc drive (Drive B) will illuminate constantly. It will extinguish when the main disc drive within the computer (Drive A) is reading or writing to disc.

## Eject Button

Pressing in the Eject button allows you to remove your disc from the drive.

Before we start loading software and saving programs to disc or cassette, let's get familiar with some of the keys on the computer. Those of you who are experienced in using computers may skip this section.

With the computer switched on and the opening message on the screen, we're going to find out what the various keys do....

The cursor keys (at the bottom right hand corner of the keyboard) move the position of the cursor (the small solid block) on the screen.

Press each of the cursor keys in turn, and you will see the cursor move about the screen.

The [RETURN] key enters the information that you have typed into the computer. After the [RETURN] key is pressed, a new line is started on the screen. Each instruction that you type into the computer should be followed by pressing the [RETURN] key.

From now on, we will show [RETURN] as meaning press the [RETURN] key after each instruction or program line.

---

Under normal circumstances (by default), this key has the same effect as [RETURN] and can be used as such. However, like the function keys on the numeric keypad, the [ENTER] key may be re-defined for other uses. This will be explained later in the manual.

This key is used to delete a character to the left of the cursor on the screen (for example a letter or a number) which is not required.

Type in abcd and you will see that the letter d is positioned to the left of the cursor. If you decide that you do not want the letter d, press [DEL] once and you will see the d removed. If you press [DEL] and continue to hold it down, the letters abc will also be removed.

There are two [SHIFT] keys. If you press either of these and hold it down whilst typing a character, a capital letter or upper case symbol will appear on the screen.

Type in the letter e then hold down the [SHIFT] key and type in the letter e again. On the screen you will see:

eE

Now type in a few spaces by holding down the space bar. Try the following using the number keys which are on the top line of the keyboard, above the letter keys. Type in the num 2, then hold down the [SHIFT] key and type in the number 2 again. On the screen you will see:

2"

You can now see what happens when the [SHIFT] key is held down whilst pressing a character key. Experiment by pressing any of the character keys, either on their own, or together with the [SHIFT] key.

---

This has a similar operation to [SHIFT] except that you only have to press it once. From then on each letter that you type in will be in capitals, although the number keys will not be shifted.

Press [CAPS LOCK] once, then type in:

abedefl23456

On the screen you will

ABCDEF123456

You will notice that although all the letters are shifted to capitals, the numbers have not been shifted to symbols. If you wish to type in a shifted symbol while [CAPS LOCK] is in operation, simply hold down the [SHIFT] key before pressing a number key. Type in the following while holding down the [SHIFT] key:

abcdefl23456

On the screen you will see:

ABCDEF123456

If you wish to return to small (lower case) characters again, press the [CAPS LOCK] key once again.

If you wish to type in capital letters and shifted upper case symbols without having to constantly hold down the [SHIFT] key, this can be carried out as follows:

Holding down the [CONTROL] key, then press the [CAPS LOCK] key once. This performs the function of a 'SHIFT LOCK'. Now type in:

abcdefl23456

On the screen you will see:

ABCDEF!"#\$%&

---

Note that it is still possible to type in numbers while [CONTROL] and [CAPS LOCK] are in operation, by using the number keys (f0 to f9) at the right of the keyboard.

---

Holding down the [CONTROL] key and pressing [CAPS LOCK] once, will return you to the mode that you were previously in, (i.e. either lower case or capital letters-only). If you have returned to the capital letters-only mode, simply press [CAPS LOCK] once again to return to the lower case mode.

This key is used to clear a character within the cursor.

Type in ABCDEFGH. The cursor will be positioned to the right of the last letter typed (H). Now press the cursor left key 0 four times. The cursor will have moved four places to the left, and will be superimposed over the top of the letter E.

Notice how the letter E is still visible within the cursor. Press the [CLR] key once and you will see that the letter E has been cleared and the letters F G H have each moved one space to the left with the letter F now appearing within the cursor. Now press the [CLR] key again and hold it down. You will see how the letter F is cleared followed by the letters G and H.

This key is used to [ESC]lape from a function that the computer is in the process of carrying out. Pressing the [ESC] key once will cause the computer to temporarily pause in its function, and will continue again if any other key is pressed.

Pressing the [ESC] key twice will cause the computer to completely [ESC]lape from the function which it is carrying out. The computer is then ready for you to type in some more instructions.

### Important

When you reach the right hand edge of the screen by entering 40 characters on a line, the next character will automatically appear on the following line at the left edge of the screen. This means that you should NOT press [RETURN] as those of you accustomed to

---

typewriters might press a carriage return towards the right edge of a page.

The computer does this automatically for you, and will react to an unwanted [RETURN] by printing an error message - usually a `S y n t a x e r r o r`, either there and then, or when the program is run.

SyntaxError

If the message: `S y n t a x e r r o r` appears on the screen, the computer is telling you that it does not understand an instruction that you have entered. For example type in:

`p r i n t t` [RETURN] On the screen you will see the message:

Syntax error The message appears because the computer has not understood the instruction: `printt` If you type a mistake in the line of a program, such as:

`10 printt "abc"` [RETURN] The `S y n t a x e r r o r` message will not appear until the instruction is processed by the computer when the program is run. Type in:

`run`[RETURN] (This command tells the computer to carry out the program that you have just typed into the memory.) On the screen you will see:

Syntax error in 10

`10 Erinnt "abC` This message tells you in which line the error has occurred, and displays the program line, together with the editing cursor so that you can correct the mistake. Press the cursor-right key `Q` until the cursor is over the `t` in `p r i n t t`. Now press the [CLR] key to remove the unwanted `t`, then press the [RETURN] key to enter the corrected line into the computer. Now type in:

`run`[RETURN] .... and you will see that the computer has accepted the instruction, and has printed:

---

abc

Part 5: Loading Software and games....

Welcome to those of you who skipped here from the beginning of the previous section!

Cartridge Games.

As we saw in Part 1, running the Demonstration Game supplied with your 46416128 is as easy as pressing [f2]. If you purchase additional cartridges then these will not even need that simple step. The game will start automatically as soon as the computer is switched on. ,

Loading Software from Disc (6128 only)

Each software disc should come with its own instructions, but you will probably find that you are asked to type something like:

```
run "di sc" [RETURN]
```

After a few seconds, your game will be loaded and ready to play.

If the program hasn't loaded, study any error message on the screen to see where you went wrong:

Drive A: disc missing Retry, Ignore or Cancel?

.... means that you have either not inserted your disc correctly, or, if you have a 2-drive system, that you have inserted it into Drive B.

DISC            not found

.... means that you have either inserted the wrong disc, (or the wrong side, of the disc), or you have not carefully and precisely typed in the name, D I S C

Bad command

---



.... means that you have probably mis-typed D I S C by introducing an unwanted space character or punctuation mark.

Type mismatch

\_nwans that you have omitted the quotation marks

Syntax error

\_means that you mistyped the word r u n

Drive A: read fail

Retry, Ignore or Cancel?

means that the computer has failed to read data from your disc. Check that you have inserted the correct disc and press R to Retry. This message will appear if ever ym corrupt your disc by switching the system on or offwhile the disc is in the drive.

Once you have learned how to make back-up copies of discs, always do so for valuable grams, and especially for your master CP/M system disc.

adi g Software from Tape (464 only)

Cach. software tape should come with its own instructions, but you will probably find thatyou are asked to type something like:

r u n " [RETURN]

There is a shorthand way oftyping this, which only requires two keystrokes.

1) Press the [CONTROL] key, and while holding that key down, press the [ENTER] key.

2) Release those keys and then press the [RETURN] Key.

---

You will then be prompted by the message:

Press PLAY then any key:

11w [PLAY] key you are being prompted to press is the key on the cassette drive whechinism. Press it now until it 'latches' and stays down. The part of the prompt ummage that asks youpress any key' is a commonly used phrase in computing that can mislead the unwary. It is used to simplify sequences that instruct a program to get on with the next operation by avoiding the need for further specific instruction keywords.

R would be more correct to say 'press any key - other than the [SHIFT][CAPS LOCKI[CTRLI[ES01 keys (or any of the keys on the cassette sub-panel)' - but all computers inevitably have~to make certain assumptions in the interests of simplicity. Wherever the term 'press any key'occurs in this user guide or in programs you buy, assume the exceptions outlined above still apply.

The key you press will not result in a character appearing on the screen, it will simply cause the tape motor to start playing the tape. Ifthe tape does not start -press another key (it's a good habit to use the large [ENTER] key) and check that the [PAUSE] key has not been accidentally pressed down.

Ifyou type more than one key, the computer will discard any other keystrokes once it has started to load the program.

Note that you have not specified any particular program name. As long as nothing else occurs on the aame line immediately after the

RUN "

instruction, the computer will search for the first program it can find on the tape, and start to load it. When the computer has found the first correctly recorded program on the tape, the following message will appear:

---

## Loading GAMEPLAY 1 block 1

This advises you that the computer has located the first of a series of blocks of the program entitled 'Gameplay'. Each program is saved on the tape in the form of blocks of data (up to 2kbytes long) that are read into the computer - each data block is separately identified on the tape, and the message on the screen advises you of the current block being read. After each block of data, the tape pauses momentarily, and then starts again - shortly after restarting, the message updates to include the block number presently being read.

If at any time the computer detects bad data, it will display an error message that advises the nature of the problem. It is not generally possible to do anything other than attempt to run the program again until it loads without an error.

Assuming that your tape is loading, read the instructions on the screen, and your 464 will do the rest .....

## Read errors

If you get the message displayed that a Read error has occurred while the 464 has been trying to load a program or data from the cassette, then the tape will continue to play, and the computer will continue to read the blocks that it finds after the error - except that it will not attempt to LOAD them unless they are identified as being block 1 of the program it first tried to load (unsuccessfully).

This means that after a read error, you can stop the tape using the cassette [STOP/EJECT] key, [REW]ind the tape to the beginning, and press [PLAY] again. The computer will then have another try at loading the program in which it found the read error - and with luck, this time you will be successful.

---

Read errors arise from a number of causes - the most common of which is accidental damage to the cassette tape through creasing, stretching or other afflictions of the recording surface. They can also arise from the apparently innocent practise of switching off the computer whilst the cassette [PLAY] key or the [REC] and [PLAY] keys are still pressed down.

This is because when the key is pressed down, the tape is held up against the tape head, and a brief pulse of electricity can pass through this record/replay head as the power supply discharges itself. Even though the tape itself has remained static -this 'switch off pulse' (and also the switch on pulse) can effectively scramble the information in that part of the tape, and make it unreadable. Furthermore, the stationary tape is held tightly between the capstan and pinch roller, which, if greatly prolonged, can lead to tape creasing.

Read errors can also occur if the tape was [PAUSE]d during the record or play process, or if it has been originally recorded on another 464 where the tape heads are incorrectly aligned.

Read errors will also arise from time to time for purely arbitrary reasons. The compact cassette was not originally designed as a medium for data storage, and as such it has several shortcomings that set it apart from the more complex and considerably more costly 'professional' tape storage systems.

Notwithstanding this, the cassette has performed an excellent job of providing a standard medium for low cost computer data storage and retrieval. The limitations of the physical characteristics of the size of the magnetic particles on the tape surface, coupled with the speed at which the tape passes the head places a finite limitation on the rate at which data will transfer between tape and computer. To attempt to push speeds beyond the upper tolerance of the system will cause

---

=reliable operation - particularly with mass  
cassette duplication methods used for  
manufacturing low cost software.

NOTE that cassettes containing programs from other  
types of computer cannot be read or loaded on the 464.  
They may look the same - they may even make similar  
sounds' if played through an audio cassette playback  
system - but they will not load and run. If you do find  
any that appear to load and run - then we would be  
pleased to hear from you giving details of the computer  
type and program concerned.

---

## **Part 7: Save It....**

Now that you've exercised your fingers by typing in a few instructions, you probably want to know how to save the program. If you have a 464, then you will save it to cassette. On the other hand, if you have a 6128, you will save it to disc. Later on you will then be able to re-load the program any time you want.

Apart from the obvious physical differences, there are a number of operational differences between using a cassette and a disc. A new blank disc cannot just be taken out of its wrapper and used, like a tape can. First it has to be 'formatted'. This process will be described in a moment. And while the disc stores and retrieves information automatically on various portions of the disc surface a cassette has to be positioned and rewound by hand. The computer does automatically start and stop the tape, however.

Another point worth mentioning here is the importance of correctly naming the stored files. Cassette file names generally conform to very loose standards, varying greatly in length, and can sometimes be omitted altogether. Disc file names, however, must conform strictly to certain standards, which are described later in this section.

If you have a 464 please turn to the section entitled "Saving a Program in Memory onto cassette".

### **Formatting discs for use**

Before writing any data onto a new blank disc, the disc itself must first be formatted. Formatting can be likened to building a series of shelves and dividers onto a disc prior to the storage of information on those shelves; in other words, laying down an organised framework around which data can be put in or taken out.

Formatting divides the disc into 360 distinctly separate areas:

---

There are 40 tracks from the outside of the disc (Track 0), to the inside (Track 39), and the circumference of the disc is divided into 9 sectors.

Each track in a sector can store up to 512 bytes of data; hence the total available space on each side of a disc is 180Kbytes.

## **First Steps Using the Master CP/M System Discs Package**

To prepare a new blank disc for reading and writing your own programs onto, you will need to format the new disc. This is carried out using Side 1 of your master CP/M system disc (supplied with the computer).

Switch the system on, and insert Side 1 of the master system disc into the drive.

Type in:

```
|cpm [RETURN]
```

(You will find the bar symbol | by holding down [SHIFT] and pressing the @ key.)

After a few seconds you will see the following message at the top of the screen:

```
CP/M PLUS           for the 6128. (c)1985 Amstrad plc
```

This is a 'Sign on' message indicating that the computer is now under the control of the CP/M Plus operating system. You will also see the letter A> together with the cursor displayed on the screen. This is a prompt, (similar to Ready during normal BASIC operation) indicating that the computer is awaiting your instructions.

Once you are operating CP/M you cannot enter BASIC commands into the computer, as these will not be understood.

If for example, you type in the BASIC command:

```
cls [RETURN]
```

The Computer will return your entry, together with a question mark:

```
CLS?
```

...indicating that it does not understand your command.

---

To look briefly at some of the CP/M commands, type in:

```
dir [RETURN]
```

On the screen you will see a DIRectory of CP/M and utility COMmands, one of which is DISCKIT. Type in:

```
diskit [RETURN]
```

After a few moments, you will see the DISC KIT opening message at the top of the screen, followed by:

```
One drive found
```

This message tells you that you are running the DISC KIT utility program, and the computer has detected that you are operating only one disc drive (the one within the computer).

If you have connected an additional disc drive to the system, the message will say:

```
Two drives found
```

At the bottom of the screen, you will see the following:

```
Copy      7
```

```
Format    4
```

```
Verify    1
```

```
Exit from program 0
```

This is known as the main DISC KIT menu. The numbers in boxes refer to the function keys at the right hand end of the keyboard (marked f0, f1, f4 and f7), and pressing one of these keys takes you to your selected menu choice.

Note that pressing function key number 0 at this point will exit from the DISC KIT program back to CP/M Direct Console Mode (the A > prompt).

We now want to format a disc, so press function key number 4(f4).

```
BEWARE
```

```
FORMATTING A PREVIOUSLY RECORDED DISC WILL ERASE ITS CONTENTS
```

---



You will now see a new menu offering you a choice off different formats, namely:

```
System format  9
Data format    6
Vendor format  3
```

Exit menu .

As before, we can now press one of the function keys (f3, f6, or f9) to select the type of format we want. Each of these different formats will be explained later in the manual, but for now, select Data format by pressing function key number 6.

Note that pressing the . key (below f3, f6, and f9) exits from the Format mode back to the main DISC KIT menu.

Having pressed function key number 6, (and assuming that you have not connected an additional disc drive to the computer) you will see the message:

```
Y Format as Data
```

Any other key to exit menu

At this point, you should remove your master CP/M system disc, and insert the disc that you wish to format. The side of the disc to be formatted should be placed face-up in the drive.

Now press the Y key, (Y for Yes, meaning 'go-ahead and format the disc').

The disc will be formatted, tracks 0 to 39; the current track number being indicated at the top left of the screen.

You will not be able to format a disc which has its write protect hole open. Attempting to do so will result in the message:

```
Disc write-protected
Insert disc to format
R-etry or C-ancel
```

...whereupon you should type C to Cancel, remove the disc, and insert the correct disc for formatting, with its write protect hole closed.

---

Make sure that you don't close the write protect hole on a disc that contains programs you want to keep, and NEVER close the write protect holes on your master CP/M system disc package.

When formatting is completed, you will be asked to remove your newly formatted disc from the drive, then to press any key to continue.

Having done so, you will be able to format another disc by inserting it and pressing Y again. This may be repeated any number of times until you have formatted all the discs you require in that particular format.

When you have finished formatting, press any key (other than Y) to return to the main DISC KIT menu once again.

The menu options C o p y and V e r i f y will be dealt with later in this manual, but for now, having learnt to format with CP/M, reset the computer using [CONTROL] [SHIFT] [ESCI].

Always keep the master copy of your CP/M system disc in a safe place, as it is literally the key to your system. Later in this manual, you will be shown how to make 'working copies' of your system disc, so that you can keep your master copy safely locked away.

#### Formatting on a 2-drive system

Follow the instructions above, selecting the F o r m a t option from the main DISC KIT menu by pressing key ^ then select D a t a f o r m a t by pressing key W

At this point you will be shown a third menu offering you the choice of drive on which to format:

Format A:        8  
Format 8:        5

Ch

Exit menu        2

Choosing the option F o r m a t B : (key f5) will enable you to leave your system disc                      am  
(Side 1) in Drive A, while you place the disc to be formatted in Drive B.

After selecting F o r m a t B : , you may then either press Y to go ahead with

formatting, or any other key to exit to the main DISC KIT menu.

If you have selected the F o r m a t A : option (key f8), you MUST then remember to remove your system disc from Drive A, and instead insert the disc to be formatted.

Don't forget - NEVER RISK OVERWRITING YOUR MASTER CP/M SYSTEM DISC.

---

Now that we have a formatted blank disc (or two), we can start to manipulate BASIC programs to and from disc.

#### Saving a Program in Memory onto Disc

Having typed a program into the computer's memory, save it onto disc by typing in:

```
save "filename" [RETURN]
```

Note that the naming of the program is obligatory.

A filename on disc consists of 2 parts (fields). The first part is obligatory and can contain up to 8 characters. Letters and numbers may be used but NO spaces or punctuation marks. This first field usually contains the name of the program.

The second field is optional. You can use up to 3 characters, but again no spaces or punctuation. The 2 fields are separated by a dot .

If you do not specify a second field, the system will automatically label it with a token of its own, such as . B A S for BASIC files or . B I N for binary (machine code) files.

As an example of saving to disc, write a short program into the memory, insert a formatted disc, then type in:

```
save "example" [RETURN]
```

After a few seconds, the prompt R e a d y will appear on the screen, and the program will have been saved onto disc. (If not, check any error message on the screen to establish whether you either forgot to insert your disc into the correct drive, forgot to close the write protect hole, or mistyped the command.)

#### Catalog

After saving the above program, type in:

```
c a t [RETURN]
```

On the screen you will see:

```
Drive A: user 0 EXAMPLE.BAS IK .177K free
```

The filename will be displayed, including any specified or token second field, together with the file length (to the nearest higher Kbyte). The amount of free space on the disc will also be displayed.

#### Lo~g fIrom

Programs may be loaded from disc then run, using the commands:

```
load "filename" [RETURN]
```

```
r u n [RETURN]
```

---

.... or they may be run directly using the command:

```
run "f i Lename" [RETURN]
```

Note that protected programs maybe run directly only.  
Saving a Program in Memory onto Cassette

Having typed a program into the computer's memory, save it onto cassette by typing:

```
SAVE "FILENAMC [RETURN
```

The file name, can be any combination of 16 keyboard characters (including spaces). If you try to enter a longer name, the 17th and subsequent characters will be discarded.

The computer responds with the prompt message:

Press REC and PLAY then any key:

Remember the qualification of the term 'any key', the tape will start and the computer will save the program with the specified filename.

#### IMPORTANT

Note that the computer cannot detect whether or not you have actually pressed the correct cassette keys - so if you only press [PLAY], the tap., will start and the program will appear to be saved when it is not.

BEWARE: If you accidentally press [REC] and ~YI when you want to read/load in a program from tape, then you will erase whatever programs exist on the cassette~ If uninterrupted by pressing the [EX, ] -key, the tape will wind through to the end and be completely erased since the tape will not have been halted by the computer finding the program it was searching for. If you are anxious to avoid losing data in this way, get in to the habit of using the snap-offwrite protect tabs on the cassette casing before you remind yourself the hard way.

#### Unnamed files and CAT

If you SAVE a file without giving it a file name:

```
SAVE ""
```

then BASIC will save it as an U n n a m e d f i l e. The cassette can save as many files of the same name (including unnamed files) as can be fitted onto a tape one after another, unlike a disc system which requires each entry to have a unique name.

You will quickly lose track of your programs if you don't give them names that will, remind you of their content -, and you are advised

---

to add some date code to the name so that you can tell which are the most recent updates of your programs and data files.

You can C A T catalogue the contents of a cassette by entering the command C A T, and following the instructions:

Press PLAY then any key:

BASIC will now list the contents of the tape, returning all file names, as UPPER CASE, followed by the number of blocks present, and then a single character that tells you what sort of file it is:

```
S           is a standard BASIC program
%           is a protected BASIC program
*           is an ASCII text file
&           is a binary file
```

An O k at the end of the line indicates that file is readable, and would have loaded had the computer been requested to do so. Performing the C A T function will not affect the program currently in the computer's memory..

#### Supersafe and Speedload

The 464 offers two speeds to the user: a Supermfe speed of 1000 baud (bits of data per second), and a Speedload rate of 2000 baud. The Speedload rate thus writes, and reads data at twice the Supersafe rate, although it sacrifices the safety margin sometimes required to account for inconsistencies in low cost tapes and the errors that can arise from the different alignment of tape heads in different recorders.

For programs saved and loaded on the same machine, the Speedload rate should prove to be acceptably reliable using reasonable quality tapes. Speedload will also be capable of loading most commercial software tapes directly without errors -although AMSTRAD advise that all such software be recorded using the option of both speeds.

The computer automatically sets itself to read the rate at which the tape has been recorded. When S A V E ing a program, you need to instruct the computer if you wish to use the Speedload rate, or it will default to the Supersafe speed.

To select the higher rate to save your programs and data, check you are in direct mode (the R e a d y prompt displayed), and type:

```
SPEED WRITE1
```

to return to the default speed, you can either reset the computer (in which case all data will be lost, or you can type (at the R e a d y prompt):

```
SPEED WRITE0
```

Cassette considerations

---

Although the Datacorder will happily accept all types of cassette up to C90, -you should only use C12 (6 minutes per side) and C30 at the most. Programs stored at the end of long cassettes are hard to locate unless you are prepared to wait for them to be found (and you can remember the filename you gave them), or you are meticulous about using the tape counter in Your indexing system (ie in the file name). If you want to overwrite a program stored on a long cassette, then you must be careful to locate the start point, and take care that you do not overwrite any other program that you may wish to keep.

Chapter I Page 48            Foundation Course

Overall, use each separate cassette for as few programs as possible. C12 cassettes are relatively cheap - and should you damage the tape for any reason, it's considerably less tempting to want to save a cassette which is good in parts'.

Finally, please remember that commercial software is nearly always supplied under strict conditions of copyright. You should not attempt to copy or otherwise duplicate software supplied on cassette other than in accordance with the terms of sale of that software (some programs actually encourage you to make backup copies) - even if it's 'only for a friend'. The laws of copyright are being revised to deal with all forms of unauthorised software duplication, and although there have been few prosecutions to date, this situation will change substantially over the next few years, and may be retrospectively enforceable.

Advanced SAVEing techniques

~ are four ways in which files may be Saved by the 46416128. In addition to ordinary BASIC file saving, by:

```
save "filename" [RETURN]
```

-there are three alternative methods, for more specialised purposes:

```
~rues save "filename",a [RETURN]
```

Adding the suffix , a instructs the computer to save the program or data in the form of an ASCII text file. This method of saving data applies to files created by ~processors and other applications programs, and its use will be further discussed in applications are encountered.

Protected Files

```
save "filename",p [RETURN]
```

Adding the suffix , p tells the computer to protect the data so that the program can be Listed after Loading it, or Running it then stopping its execution ~ the [ESC] key function.

---

PS~ saved in this way can only be run directly, using the commands:  
run "filename" [RETURN]

\_\_W ....  
    chain "filename" [RETURN]

If you anticipate wanting to edit or alter the program, you should also keep a copy for ~elfin unprotected form, i.e. without the .p suffix.

COMW Chapter 1 Page 49  
save "filename", starting address, , <length in bytes>  
optional entry point, l [RETURN]

This option allows you to perform a binary save where a complete block of data in the computer's RAM is stored onto disc or cassette exactly as it occurs in the memory. It is necessary to instruct the computer where the section of memory you need to save starts, how long it is, and if required, the memory address at which to start execution should the file be run as a program.

#### Screen Dump

This binary save feature allows data from the screen memory to be stored directly onto disc or cassette in the form of a screen dump. The contents of the screen will be saved exactly as it is seen, using the command:

save "scrndump",b,49152,16384 [RETURN]

... where 49152 is the starting address of the screen memory, and 16384 is the length of the screen memory that you wish to save.

To call it back onto the screen, type in:

load "scrndump" [RETURN]

#### Copying Programs from Disc to D\*

Using the commands already learnt in this section, it can be seen that disc to disc or tape to tape program copying is performed simply by loading the program into the memory from the original (source) disc or tape, removing the source disc or tape, and saving the program onto the new (destination) disc or tape.

More information on using the system to manipulate program files between discs (and cassette), will be found later in this manual.  
Part 10: Introducing AMSDOS and CP/M (6128 only)....

What is AMSDOS?....

AMSDOS is an abbreviation of AMStrad Disc Operating System, and offers the following file handling commands and functions; modified

---

to use the disc drive of the 6128, instead of the cassette drive of the 464.

```
LOAD "filename"  
RUN "filename"  
SAVE "filename"  
CHAIN "filename"  
MERGE "filename"  
CHAIN MERGE "filename"  
OPENIN "filename"  
OPENOUT "filename"  
CLOSEIN  
CLOSEOUT  
CAT  
EOF  
INPUT #9  
LINE INPUT #9  
LIST #9  
PRINT #9  
WRITE #9
```

#### Copying a Whole Disc

The entire contents of a disc can be copied from one to another using the D I S C K I T program on Side 1 of the CP/M system disc. You may use this method to make back-up copies of the master system disc itself. Insert Side 1 of the system disc into the computer's disc drive, and type: |cpm After the A > prompt, type: diskkit After a few seconds, you will see the DISC KIT opening message at the top of the screen, followed by:

One drive found

This message tells you that you are running the DISC KIT utility program, and the computer has detected that you are operating only one disc drive (the one within the computer).

If you have connected an additional disc drive to the system, the message will say:

Two drives found

At the bottom of the screen, you will see the following:

```
Copy      7  
Format   4  
Verify   1  
Exit from program      0
```

This is known as the main DISC KIT menu. The numbers in boxes refer to the function keys at the right hand end of the keyboard (marked

---



f0, f1, f4, and f7), and pressing one of these keys takes you to your selected menu choice.

Note that pressing function key number 0 at this point will exit from the DISC KIT program back to CP/M Direct Console Mode (the A > prompt).

We now want to copy a disc, so press function key number 7 (f7).

BEWARE

COPYING ONTO A PREVIOUSLY RECORDED DISC WILL ERASE ITS CONTENTS

Copying on a 1-drive system

Assuming that you are using a 1-drive system (i.e. you have NOT connected an additional drive), you will then see the message:

Copy

Any other key to exit menu

At this point, you should remove your master CP/M system disc, and insert the disc that you wish to copy from. If you wish to copy the CP/M system disc itself, simply leave it in the drive.

When the disc that you wish to copy from is in the drive, press the Y key, (Y is for Yes, meaning 'go-ahead and copy the disc').

The format of the disc will be examined by the computer, and displayed at the top of the screen for your information.

After a while, you will see the message:

Insert disc to WRITE

Press any key to continue

At this point, you should take the disc that you are copying out of the drive. Then insert the disc for copying onto, and press any key.

Information about the format of the disc to be copied onto (even if it is a new unformatted disc) will be indicated at the top of the screen.

If the disc that you wish to copy onto is not correctly formatted (or not formatted at all), this will be taken care of during copying, and you will see a message such as:

Disc isn't formatted (or faulty)

Going to format while copying

Disc will be system format

... or some other similar message, depending upon the discs that you are copying between.

---

When the computer is again ready for the disc to copy from, it will display the message:

```
Insert disc to READ
Press any key to continue
```

.... and you should re-insert the disc to copy from.

After repeating this process a few times, the contents of the first disc will be copied onto the second disc, and the message:

```
Copy completed
Remove disc
Press any key to continue
```

.... will appear, whereupon you should follow the instructions on the screen, which give you the option to either copy another disc (by typing Y) or 'exit'ing to the main DISC KIT menu.  
Write protection

Note that you will not be able to copy onto a disc which has its write protect hole open. Attempting to do so will result in the message:

```
Disc write-protected
Insert disc to WRITE
R-etry or C-ancel
```

.... whereupon you should type C to Cancel, remove the disc, and insert the correct disc for copying onto, with its write protect hole closed.

Make sure that you don't close the write protect hole on a disc that contains programs you want to keep, and NEVER close the write protect holes on your master CP/M system disc.

#### Verifying discs

The D I S C K I T program also provides you with the facility to verify (check) a disc. Information about the format of the disc is displayed, and all files on the disc are read. Any errors present in the files are reported on the screen. To verify a disc, insert Side 1 of your system disc, and type:

```
1 cpm
```

At the A > prompt, type:

```
diskit
```

Select the V e r i f y option from the main DISC KIT menu (key f1), and then follow the instructions on the screen.

---

When the disc that you wish to verify is inserted in the drive, type Y to commence verifying.

When verifying is completed, the following message will be displayed:

```
Verify completed
Remove disc
Press any key to continue
```

You may then repeat the verifying process by typing Y, or exit to the main DISC KIT menu by pressing any other key.  
Part 11: Introducing the Bank Manager (6128 only)....

Using the second 64K of memory

The 6128 contains 128K of RAM (Random Access Memory) in two lots of 64K. CP/M Plus uses the full 128K all the time, but BASIC does not normally use the second 64K only the available memory in the first 64K. It would be a pity to leave the extra 64K completely unused when programming in BASIC, so a program has been provided to make special use of this extra memory. The program offers some extra commands that make it possible to use the second 64K of RAM as either a storage space for screen images, or as a storage space for strings.

The program providing these extra commands is known as the 'BANK MANAGER' where 'bank' is a technical term used to describe a section of memory.

Using the BANK MANAGER for screen images

The 6128 is displaying a screen image all the time. To do this, it requires 16K of memory in which to store the information about the colour and brightness of every pixel (dot) on the screen. The 6128's memory allows up to six screen images (each in a 16K block) to be present in the computer's memory at any one time. The BANK MANAGER provides the facilities for you to juggle and display up to five of the six possible screens from BASIC.

When you first switch on, the screen is displayed from a 16K block of memory (which we will call 'Block 1') out of the first 64K. The other four screens are held in the second 64K of memory and are called Block 2, Block 3, Block 4, and Block 5.

Only Block 1 (from the first 64K) can be used to actually display a screen. Therefore, to see a screen stored in the second 64K (Blocks 2 to 5), it is necessary to move the required screen into Block 1. The BANK MANAGER provides all the commands needed to move screens around, such as I S C R E E N C O P Y which simply transfers one screen to another, overwriting the contents of the existing screen; and I S C R E E N S W A P which exchanges the contents of two screens.

The BANK MANAGER uses 'external commands' which start with the bar symbol I (obtained by typing (SHIFT]@).  
How to use the BANK MANAGER

---

Reset the computer [CONTROL] [SHIFT] [ESC] then insert Side I of your system disc package and type:

```
RUN "BANKMAN"
```

The loading procedure is detailed in Chapter 6 part 13 on RSX's (Resident System eXtensions), and it is advisable to have some understanding about RSX's and also about reserving memory space, before using these routines in your programs. However, to try the following examples, you will not need to understand the loading procedure.

Type:

```
MODE 1
```

```
PRINT "THIS IS THE DEFAULT SCREEN" ISCREENSWAP,1,2
```

The text should have disappeared. You are now looking at what used to be stored as Screen 2 (in Block 2). If the machine has just been switched on, this will probably be a random pattern. To clear this pattern, type:

```
MODE 1
```

.... then type:

```
PRINT "THIS IS SCREEN 2" ISCREENSWAP,1,2
```

Your original text reappears. If you now repeat the I S C R E E N S W A P , I , 2 command, you can see that the contents of the two screens are being exchanged. You can swap the contents of any of the five screens with each other, but bear in mind that only when you do a swap involving Screen 1 will the display be altered.

The other command available is I S C R E E N C O P Y. This allows you to copy one screen onto another, overwriting it with a new image.

Type:

```
MODE 1
```

```
PRINT "THIS IS A SCREEN TO COPY" ISCREENCOPY,2,1
```

The contents of Screen 1 are copied into Screen 2. If you reverse the parameters by typing:

```
MODE 1
```

```
ISCREENCOPY,1,2
```

.... the contents of the currently displayed screen are overwritten from screen 2.

---

So the first parameter is the screen to copy to; the second parameter is the screen to copy from.

When copying screens, note that you will produce a disjointed display if the screens' M O D E s are different, or if the viewing screen has been 'scrolled' since the last M O D E command. The BANK MANAGER's screen commands are intended for use with pictorial screens rather than text screens, as scrolling is less likely to have occurred.

Using the BANK MANAGER for String Storage

There are four more commands provided by the BANK MANAGER, which allow the use of the additional 64K of memory as a filing system for string variables.

Most programs can be divided into two parts: firstly, the actual program instructions, and secondly, the data that the program uses. A good example of this is a database program such as an address book. Such a program would use an array of strings to store the names and addresses of the people listed in the book.

Strings can be stored in the second 64K of memory, one after another, end to end. The memory that the strings are stored in can be divided into compartments, which are called records. A record can be of any fixed length between 2 and 255 characters, whereas the length of a string in BASIC varies according to its contents. The purpose of a record is to provide standard sized compartments, like a set of pigeon holes, in which to store the untidy string information. Each operation to store or retrieve data from a record is automatically followed by a step to the next record, ready for the next operation. The record to be used for the next operation is called the 'current record' and will be used automatically unless a different record is specified.

We call this system of memory management a 'RAMdisc'. This is because it operates in a similar way to random access disc systems, but using RAM instead.

Read the following descriptions of the various commands so that you understand what each command is for, if not how to use it, then go on to try out the examples.

The first of the RAMdisc commands is I B A N K O P E N. This command specifies how many characters of string information each record can contain. Its syntax is:

```
IBANKOPEN,n
```

Where n is a number specifying the amount of characters in the record. The value n can be in the range 0 to 255, but values of 0 and 1 will have strange effects.

18 A N K W R 1 T E stores a string into the current record. The current record is then incremented to point at the record after the

---

one that has just been written, ready for the next operation. Its syntax is:

```
IBANKWRITE,@r%,a$
```

.... or....

```
1BANKWRITE,@r%Oa$,n
```

.... where r % is an integer variable in which a code is returned giving information about the operation. a \$ is a string variable containing the characters to be written into the record. In the first of the two examples, the record is the current record. In the second example, the optional parameter n specifies the record to be written to.

1 B A N K R E A D examines the current record, and returns its contents in a string. The current record is then incremented to point at the record after the one that has just been read, ready for the next operation. Reading the contents of a record does not change it, so the record can be read over and over again. The syntax for IBANKREAD is:

```
IBANKREAD,@r%,a$
```

.... or....

```
IBANKREAD,@r%,a$,n
```

.... where r % is an integer variable in which a code is returned giving information about the operation. a \$ is a string variable into which the contents of the record is read. In the first of the two examples, the record is the current record. In the second example, the optional parameter n specifies which record is to be read from.

The last command is 1 B A N K F I N D. This command searches through the records trying to find a particular string. If the string is found, then the number of the record in which it is found, is returned. The syntax for 1 B A N K F I N D is:

```
1BANKFIND,@r%,a$
```

.... or....

```
1BANKFIND,@r%Oa$,n
```

.... or....

```
IBANKFIND,@r%oa$,n,m
```

.... where r % is an integer variable in which either the record number (where the string was found) or a code indicating that it was not found, is returned. a \$ is the string to search for. The optional parameter n, specifies which record to begin searching from. If n is unspecified, then the search will start at the current record. The

---

second optional parameter m specifies at which record to end the search even if a match is not yet found. If m is omitted, the search continues through the whole 64K, which may overrun the end of the data you have written.

Now try this out. If you have already run the BANK MANAGER program to look at the screen swapping commands, and have not reset the computer, then the extra commands will currently be resident in memory. If not, you will need to insert Side 1 of your system disc package, and type:

```
RUN "BANKMAC
```

Now type:

```
1BANKOPEN,20
```

This sets the record length to 20 characters and sets the current record to be 0. Now type:

```
a$="FIRST ENTRY11+SPACE$(9)
```

... which sets up a \$ to be exactly 20 characters long. Now type:

```
r%=0
```

... to initialise r %. Now type:

```
1BANKWRITE,@r%,a$
```

This will write a \$ into the first record (record 0). Now type:

```
d$=SPACE$(20) 1BANKREAD,@r%,d$.0 PRINT d$
```

The first command sets up d \$ to contain 20 spaces. This is enough room to contain the contents of the record when it is read. The second command reads record number 0 and places the result in d \$. As the current record would now be stepped to record 1 (by the previous 1 B A N K W R I T E operation) it is necessary to specify record 0, as the record to read. Remember - if no record number is specified, then the current record is read. Finally the result of the reading operation is printed out. Therefore d \$ ought to contain 'F I R S T E N T R Y' and 9 spaces.

Now type this:

```
b$="TWO"+SPACE$(17) c$="THREE"+SPACE$(15) 1BANKWRITE,@r%,b$,1  
1BANKWRITE,@r%,c$
```

This puts b \$ and c \$ into records 1 and 2. Again the optional parameter in the first 1 B A N K W R I T E command is used, allowing record 1 to be specified as the record in which to place b\$. The current record number is thereafter set to the following record, so it is unnecessary to specify in which record to put c \$ - it will be placed in record 2 automatically.

---

Type:

```
PRINT r%
```

The result of `PRINT r %` after the above example should be 2. This number can either be considered as the last record operated on, or as the current record minus one. In the above example the last record operated on was 2 and the next record will be 3.

The purpose of this 'return code' is to provide information about the operation just carried out. A successful operation will return a positive number indicating a record number; an unsuccessful operation will return a negative number indicating an error code. There are two possible errors that can be returned by `IBANKWRITE` and `IBANKREAD`. These are:

- 1        Indicates that the end of the file has been reached. This happens when all the records have been used up or a record has been specified that does not exist.

-3                Indicates a bank switching failure. This should never happen.

Try some further examples:

```
d$=STRING$(20,"X") IBANKOPEN,20 FOR n=1 to 3:IBANKWRITE,@r%,d$:NEXT
```

This will set up `d $` to contain 20 X's. `IBANKOPEN` resets the current record to be 0, so the `IBANKWRITE` command will overwrite the contents of records 0, 1, and 2 with `d $`.

Now type:

```
a$="FIRST" IBANKWRITE,@r%,a$.0
```

This puts the word " F I R S T " into record 0, overwriting some of the X's. Now reset `d $` to spaces by:

```
d$=SPACE$(20)
```

Now type:

```
IBANKREAD,@r%,d$,0
```

This reads back record 0 into `d`

To recap the steps taken so far:

All three records contain X's. Into record 0 has been added the word " F I R S T ". `d $` has then been filled with spaces. Finally, record 0 has been read back into `d $`. Now type:

```
PRINT d$
```

---



The result should be " F I R S TXXXXXXXXXXXXX X ". This illustrates an important consideration when using these commands. If the string being placed into a record does not completely fill that record, the old characters that have not been overwritten will remain in the record. Therefore, it is advisable to put a string of spaces or C H R \$ ( 3 2 Ys into the record before storing fresh information in it. This avoids reading back a string and finding that it has characters which were, not intended to still be in it. The same consideration should also be made to the string into which the record will be read. If the string is longer than the length of the record, then there will be characters left unchanged at the end of the string. That is why d \$ was cleared (filled with spaces) before reading record 0 into it.

It is possible to write a string into a record which is too short to accept the complete string. If this happens, the excess characters will be ignored.

Likewise, it is also possible to read the contents of a long record into a short string. and once again, excess characters (read from the record) will be ignored. In normal string operations, BASIC would automatically extend the string to accept the additional characters but this does not happen within an external command.

Finally the I B A N K F I N D command. This allows the records to be searched for a particular entry. For example, if record number 24 starts with the word " F R E D

this could be found with the command:

```
IBANKFIND,@r%,"FRED"
```

This is extremely useful for database programs where for example, a name, or address, is being searched for.

I B A N K F I N D will start at the current record, and will work its way through the records until it reaches the searched-for string, or the end of the second 64K in which the records are stored.

It is possible to specify the record number at which to start from, by including a record number at the end of the command.

A further record number, after the starting record number, can be included to specify the record at which to end the search.

I B A N K F I N D can be used to look for a string which does not lie at the start of a record. To do this, C H R \$ ( 0 )'s are used in the character positions before the significant characters to be searched for. The C H R \$ ( 0 )'s will be treated as 'wildcards' (like Vs in CP/M filenames) which means that they count as 'any character'. An example of this would be:

```
a$=STRING$(10,0)+"FRED" IBANKFIND,@r%,a$,0
```

This would find the first occurrence of the word " F R E D " (lying between the eleventh and fourteenth character positions). The first

---

ten characters could contain a phone number or some other information that 1 B A N K F I N D would ignore.

The number of the record in which the string was found is placed in the integer variable r % used for the return code (if the word " F R E D " was found), otherwise the return code will contain - 2.

Further details....

Further information about the BANK MANAGER will be found in Chapter 7, and you should study parts 13 and 14 of Chapter 6 on the subject of using RSX's.

Well, that concludes this 11 part Foundation course on the 464/6128. By now you should know what most of the keys on the keyboard do, how to use some of the simpler BASIC commands, how to format a brand new disc so that it's ready to use, and how to perform some of the most elementary disc functions, i.e. LOADING, SAVEing, CATALOGuing etc, together with a few AMSDOS, CP/M, and BANK MANAGER commands.

The chapters that follow, look at the more specialised aspects of computing and AMSTRAD BASIC.

Good luck, and happy reading!  
Chapter 4

### Using Discs (6128 only)

Part 1: Discs

#### **Making working discs**

*This section discusses how to make discs to use from day to day, and introduces some facilities of CP/M and its Utility programs.*

Subjects covered:

Making a backup of the Master Disc. Getting started with CP/M Plus. Use of Help files. Single and Multiple drive operation. Copying files with PIP. Operating with a BASIC only disc. Turnkey AMSTRAD BASIC application. Installing a Turnkey CP/M Plus application.

Part 7 of the Foundation course described how to format a blank system disc, which you can use for BASIC and games, as well as CP/M.

Part 10 of the Foundation course showed you how to make exact copies of discs with the D I S C K I T program (on Side 1 of your system disc package).

This section considers how to use discs with the programs that you want on them.

---



## **Backup Master Discs**

It is most important to make a copy of the Master System/Utility disc provided with your computer, and keep the original safe - it will be very costly to replace if damaged! Remember that the disc supplied has two sides. Every disc, in fact, has two sides and you are free to use either side for any purposes.

Side 1 is the most important - it contains the master copy of CP/M Plus and a set of utility programs for handling discs. Side 2 has files for assembler programmers.

You should regard your copy of the master disc as a 'library' of programs. Normally, you will select the program that you require by inserting the 'library disc' on which the program is located, rather than by copying the program onto a blank disc and running it from there.

Once again, it has to be emphasised that the 'library discs' that you use, **MUST BE COPIES**, made from the master disc package supplied with the computer.

Remember that if you are using a new blank disc to copy onto, the **D I S C K I T** program (on Side 1) will format for you as well as doing the copying.

## **Getting started with CP/M Plus**

You will be used to AMSTRAD BASIC appearing when your 6128 is switched on. The BASIC will remain in charge until superseded by either a BINary program run from AMSDOS, or by loading CP/M Plus with the command **1 C P M**.

Once CP/M Plus is loaded, the 6128 will not need to refer to Side 1 again - unless of course, you want to run any of the utility programs contained on it. Only the start-up disc need be a System disc; all others can be Data-only discs, which have a greater storage capacity.

Running a program is simply a case of inserting the disc containing the required program and typing its name. Data used by the program may be on the same, or other, disc as the program. CP/M Plus allows the user to swap discs in the same way that AMSDOS does. If a number of programs, with perhaps a few utilities, are required on one disc for convenience, then use the program called **P I P** on Side 1, as described later in this chapter and in Chapter 5.

---

### **Maintaing a Hi profile!**

The system disc provides a special file called `PROFILE.SUB`, which contains a list of commands that are executed automatically when CP/M Plus is started., You may then (if you have not already done so) insert a COPY of Side 1 of the System disc, and at the `A >` prompt, type:

```
REN PROFILE.SUB=PROFILE.ENG
```

.... which creates the file `PROFILE.SUB` from `PROFILE.ENG`. This profile, which will be acted on next time CP/M Plus is started, contains the commands:

```
SETKEYS KEYS.CCP
```

```
LANGUAGE 3
```

.... to set the cursor keys to be suitable for typing CP/M commands, and to convert the screen output to UK (from USA), effectively making **[SHIFT]3** display as a '#' symbol.

When the keyboard has been set up with the `SETKEYS.KEYS.CCP` command, then CP/M command lines can be edited in much the same way as a line of BASIC. Full details of `SETKEYS` are given in Chapter 5 part 2.

### **One Drive or Two?**

When CP/M Plus is first loaded, it detects the number of disc drives attached. This number is displayed as part of the sign-on message. Note that the process can be fooled if the second drive has a disc partially inserted.

All error messages relating to the disc mechanisms are displayed by default, as a banner on the 25th line of the screen. The programs themselves use only the first 24 lines of the screen.

When you have only one built-in drive, the bottom line will also display the message 'Drive is A:' or 'Drive is B:'. This is CP/M Plus allowing you to work with one physical mechanism as if it was two. You will have two discs to alternate between, and the bottom line of the screen will prompt you to insert the correct disc as the program requires it. This mode of operation avoids the need to purchase a second disc drive, but often requires considerable swapping of discs, which is time-consuming and introduces the possibility of human error.

---

### **Copying files from disc to disc**

A standard utility program called P I P (Peripheral Interchange Program) is provided to copy files from one disc to another.

First load P I P from Side 1, by typing at the A > prompt:

PIP

.... then a new prompt \* will show that P I P has loaded correctly. Normally, you will copy files from a Source disc (in Drive A:) to a Destination disc (in DriveB:). We have already seen that in a single drive system, Drives A: and B: are the same mechanism.

To copy one file, for example S U B M I T . C O M, type after the \* prompt:

**B:=A:SUBMIT.COM**

To copy all the files from the Source disc to the Destination disc use the command:

B:=\*.\*

To exit from P I P, press **[RETURN]** at the \* prompt.

P I P is a very sophisticated program, and further details of its operation appear in Chapter 5.

### **A BASIC only disc**

As already described, a System disc is normally only used as the disc for starting up CP/M Plus. Discs used for BASIC can therefore be Data-Only discs, which have a slightly greater capacity.

The disc must be formatted using the D I S C K I T program. To copy programs onto this type of disc, you must use P I P (loaded from Side 1), or L O A D and S A V E them from BASIC.

---

### **Turnkey AMSTRAD BASIC discs**

If you buy an application program written in AMSTRAD BASIC for the 6128, it should be ready to operate when you switch on. (The expression 'turnkey' comes from the days when all small computers had a key-operated power switch). As with the Master system disc package supplied with the 6128, it is strongly recommended that you keep the original safe, and work from a copy.

### **Turnkey CP/M disc**

The CP/M operating system allows you to load and run an immense library of software which has already been written for personal computers that support CP/M. The fundamental 'logic' of these programs has already been devised; all that is required to use them on your 6128 is to establish them on a suitable disc, and maybe to inform them of the particular method that the 6128 uses to operate the screen.

A set of programs on one disc designed to fulfil a specific application is called a 'package'. These packages are normally designed to work on a large range of different computers, each of which has its own size of screen and way of moving the cursor around.

The 6128 is provided with a built-in 'terminal emulator' when running CP/M Plus programs, and the characteristics are different from the Control Codes supported by BASIC.

Sometimes the package that you buy will have already been 'installed' for the **AMSTRAD** system, or cater for it by offering an installation compatible with the 6128. If available, simply follow the instructions provided with the software for a Zenith Z19/Z29 protocol. If the package does not have this, nor a specific AMSTRAD variant built in, then the section ahead entitled 'Configuring a CP/M Program' indicates some of the commands that can be sent to the 6128 screen to produce the sorts of effects that packages require. Normally, the installation, or customisation, procedure will involve typing in the relevant codes when requested to. Again, follow the instructions provided with the package.

The software you have purchased must be on a disc suitable for use in this system. Almost every different computer uses a different form of disc. Although many have the same size of disc, this does not necessarily mean that there is any compatibility between one and another in the information contained on them. Ask your supplier for an AMSTRAD 3 inch version.

---

### **Creating a Turnkey CP/M disc**

As well as the application program itself, it is often useful to have the utilities `SETKEYS.COM` and perhaps `SUBMIT.COM` (with their associated instruction files) on the Turnkey disc.

`PIP` can be used to transfer the `.COM` files and also to make the instruction file for `SUBMIT`. In this latter mode, `PIP` is effectively a simple one-line-at-a-time editor. For example, the file `PROFILE.ENG` on Side 1 could have been created with the following commands:

(Insert System disc, Side 1 in Drive A Type:

`PIP`

(Remove System disc, insert destination disc). Type:

```
PROFILE.ENG=CON:  
SETKEYS KEYS.CCP  
[CONTROL]J LANGUAGE 3  
[CONTROL]Z
```

### **Configuring a CP/M Program**

The 6128 supports a wide range of control codes suitable for customising a software package to run with CP/M. Most data-processing and many other packages require to be able to print messages at any part of the screen, to accept input from any part of the screen and to generally understand cursor controls.

If your package has already been customised for the AMSTRAD system, then you need not concern yourself further.

### **Configuring the Output from the package**

The installation procedure for a package will normally consist of running a special program (often called `INSTALL`) which, if it does not support either a `Z19/Z29` type of terminal or the 6128 specifically, will ask a number of questions about the parameters of the 6128 screen. The answers should be derived from the table overleaf, which is an extract from Chapter 6 part 15.

---



Control Codes		Hex	Decimal	Operation
[BEL]		&07	7	Sound Bleeper.
[BSI]		&08	8	Move cursor back one position.
[LF]		&0A	10	Move cursor down one line.
[CR]		&0D	13	Move cursor to left edge of window on current line.
[ESC]A		&1B &41	27 65	Move cursor up one line.
[ESC]C		&1B &43	27 67	Move cursor forward one position.
[ESC]E		&1B &45	27 69	Clear screen.
[ESC]H		&1B &48	27 72	Home cursor.
[ESC]J		&1B &4A	27 74	Clear from and including the current cursor position to end of screen.
[ESC]K		&1B &4B	27 75	Clear from and including the current cursor position to right edge of screen.
[ESC]L		&1B W	27 76	Insert Line.
[ESC]M		&1B &4D	27 77	Delete line.
[ESC]N		&1B &4E	27 78	Delete character at cursor position.
[ESC]Y	,C>	&1B &59 r>	27 89 <c> r>	Move cursor to given position on screen. <c> is column + 32,
[ESC]d	<r> is row + 32.	&1B &64	27 100	Clear from start of screen to and including the current cursor position.
[ESC]o		&1B &6F	27 111	Clear from left edge of screen to and including the cursor position.
[ESC]p		&1B &70	27 112	Enter inverse video.
[ESC]q		&1B &71	27 113	Exit inverse video.

### Configuring the Input to the package

The programs in the package will expect to be able to interrogate the keyboard. Most of the keys on the 6128 keyboard return standard values except for the cursor keys. It is possible to use the S E T K E Y S utility to re-define the codes produced by the keyboard, although where possible, it is preferable for each different package to be configured to accept standard values.

It is an unfortunate fact that there is not a general understanding between different items of software as to which keystroke to use for control functions. Printable characters, and 'space', [TAB] and [RETURN] are fairly universal, but disagreement sets in with 'backspace', and from there-on, matters get worse! Compare for example, the different codes expected for the operation 'move cursor to start of line':

CP/M Commands require: [CONTROL]B

.... and a typical wordprocessor might require: [CONTROL]Q S

Three useful sets of keyboard codes are provided as standard. Each configuration can be called up from files contained on Side 1 of your system discs package:

SETKEYS KEYS.CCP

.... already described as one of the commands automatically sent by P R O F I L E . S U B, will set up the keyboard suitably for CP/M commands.

#### **Starting a Turnkey CP/M Package**

Normally, all that is required is to type the package's main program name at the A > prompt. For example to run a wages program called P A Y R O L L . C O M, simply type:

PAYROLL

If any configurations require to be set up then perhaps a SO Bmit file will be provided.

#### **Autostarting a Turnkey CP/M Package**

It is possible to arrange for the CP/M Plus operating system to automatically run a particular program from the start-up system disc. This is performed by including the program's name at the end of the P R O F I L E . S U B file on that disc.



**Chapter 5**  
**AMSDOS and CP/M (6128 only)**

**Part 1: AMSDOS**

Subjects covered:

Introduction to AMSDOS  
Disc directory  
Changing discs  
Filenames and filetypes  
AMSDOS headers  
Filenames on 2 drives  
Wild cards  
Example program using AMSDOS commands  
Summary of AMSDOS commands  
Manipulating and copying files  
Reference guide to error messages

**Introduction**

AMSDOS extends the AMSTRAD BASIC supplied with your computer by providing a number of external commands, which are identified by the preceding ! (bar) symbol.

AMSDOS allows the user to change discs freely, as long as no files are in use - in which case an error message will be displayed and there could be a loss of data if the open file was being written to.

---

## **Disc &rectory**

Every disc has two sections, the directory and the data area. The directory contains a list of all the filenames and a 'map' of whereabouts on the disc each file is to be found. AMSDOS or CP/M can calculate the size of a particular file by inspecting its directory entry. Calculation of the amount of space left on a disc is made by adding up all the files in the directory and seeing how much remains unused.

Whenever a file is read, its directory entry is examined, giving the disc location. When a new file is created, free space is allocated to it, and when a file is erased the space is relinquished. The directory works in units of 1K and can have up to 64 different entries. Large files will have one entry for every 16K although normally this fact is hidden from the user.

## **Changing discs**

Under **AMSDOS (and CP/M Plus)** a disc may be changed, or removed, whenever the drive is not being accessed and neither the input nor output files are open on that drive.

Changing a disc while it is still being written to, may corrupt the data on the disc. If a disc is changed while there are still files open on it, then as soon as AMSDOS detects this, all the open files on the drive will be abandoned and an error message produced. Any data yet to be written will be lost and the latest directory entry will not be written to disc. However, AMSDOS can only detect this change when it reads the directory, which it does every 16K of the file (and whenever a file is opened or closed). Thus, potentially 16K of data could be corrupted by changing a disc while there are still files open on it.

## **AMSDOS filenames and filetypes**

It is standard practice to name disc files in such a way that there is an indication of which 'type' they are. This naming convention DOES NOT 'force' the computer to use the file in any particular way, however some programs will only accept a file when it has the correct type of name. AMSDOS will accept any type of name, but will search in preference for certain file types if not otherwise specified. (See 'AMSDOS headers' ahead.)

---

## Construction of filenames

The filename is constructed from two parts with a . (dot) separating them. The first part can be up to 8 characters long, and the second up to 3 characters long. Thus for example: "ROINTIME.DEM" , "DISCKIT3.COM" , and "DISC.BAS" are all legal filenames.

The second part of the filename is called the filetype. Filenames and filetypes can be composed of a mixture of letters and numbers, but cannot have embedded spaces or punctuation marks. Some common conventional filetypes are:

<space> Unspecified type. May be a data file created by an O P E N O U T  
" <filename>" or BASIC program saved by AMSDOS using SAVE  
" <filename>" , A style.  
.BAS BASIC program saved by AMSDOS using S A V E " <filename>" or  
S A V E " <filename> , P or S A V E " , <filename> . B A S " , A  
styles.  
.BIN Program or area of memory saved by AMSDOS using S A V E  
" <filename>" B ,binary parameters> style.  
.BAK Old version of a file, where AMSDOS or a utility program has saved a  
newer version of a file using an existing name. This allows the user to  
back-track to the previous (B AcK-up) version if required.  
COM Command file. CP/M utility programs are all of this filetype.  
SUB Instruction file for the CP/M S U B M I T program.

## AMSDOS headers

AMSDOS automatically S A V E s files with a suitable type identifier, so it is not normally necessary to specify one unless you wish to override the defaults described previously. BASIC program files, protected BASIC program files and binary files are saved to the disc with a header record, so that the AMSDOS command:

LOAD "<filename>"

.... can recognise them and take the appropriate action. If the AMSDOS command L O A D cannot find a header, it assumes that the file is a program in ASCII, i.e. plain text.

---

ding the contents of the header, when AMSDOS is asked to LO A D a file

Notwithstand 8 specif type:  
where no filetype fied, then it first looks for a file oft

..space>

if that does not exist, it looks for a file of type:

.BAS

.... then finally, one of type:

.BIN iate the filename, i.e. not needing to specify the

This allows the user to abbrev filetype, in most instances. O P E H O U T and subsequently written to,

A disc data file started with the command, will have no header, and the contents will be in ASCII, i.e. plain text, from the BASIC

O P E N I N will search for WR I T E, PR I N T or L I S T commands. The disc command files in tile same order as L O r k D, if no file type is specified.

4

### Filenames on two drives

an additional drive has been connected to the computers

On a 2-drive system, i.e. if a file on

flies can exist on either drive. The computer will not automatically look for both drivels, so the user must specify which drive to use. You can either employ the 1 A or 1 B or 1 D R I V E commands (full description ahead) to select one or other drive, and then use a normal filename, or alternatively you can override the default drive assignment by specifying the drive as an A . or B : prefix to the filename. Thus, for example:

1 B  
SAVE "PROG.BAS"  
|A

.... and....

|A  
SAVE "B:PROG.BAS"

.... both save the program to the second drive, Drive B.

---

Similarly, you can override the default U S E R number assignment (U S E R numbers allow you to partition-off the directory) by specifying the U S E R number (in the range 0 to 15) as a prefix to the filename. Thus, for example:

**LOAD "15:PROG.BAS"**

.... and....

**SAVE "15:PROG.BAS"**

.... would load and save the program to the U S E R number 15 section of the disc, whatever the default U S E R number setting. (See the I U S E R command, ahead.)

Finally, it is possible to override both default U S E R and D R I V E settings (in that order) by specifying them together in the prefix to the filename, for example:

**RUN "15B:PROG.BAS"**

### Wildcards

It is often required to perform some disc operation (copying, erasing, etc.) on more than one disc file. When a filename is specified for a particular operation, AMSDOS scans the disc directory looking for a name which exactly matches. However, it is possible (where the command allows) to perform the operation on a group of files where some of the characters in the filenames can be 'don't care'. This is shown by using the character ? in the 'don't care' position. If the whole block (or remainder of the whole block) of any part of the filename is 'don't care', then the block of ?'s can be abbreviated to the symbol \*. Thus, for example, F R E D . \* is shorthand for **FRED. ???** and **F\* . BAS** is shorthand for **F? ????? . BAS**

Finally, the expression means 'all files'.

Examples:

DIRECTORY

Match\*. BA S

Match FRE D?. BAS Match F\*. BA?

**BERT.BAS**  
**FRED1.BAS**  
**FRED2.BAS**  
**FRED3.BAK**  
**FRED3.BAS**

**BERT.BAS**

**FRED1.BAS**  
**FRED2.BAS**  
**FRED3.BAK**  
**FRED3.BAS**

**FRED1.BAS**  
**FRED2.BAS**  
**FRED3.BAS**

**FRED1.BAS**  
**FRED2.BAS**  
**FRED3.BAS**

**1 FINISH.BAS1 FINISH.BAS1**

**FINISH.BAS**

---

## Using AMSDOS commands in a program

To give you a good understanding of the AMSDOS commands, we recommend that you work through the examples, referring to the relevant sections in the rest of this chapter as you go. DO NOT type in, or run, these programs with one of your original Master CP/M system discs installed.

### Saving variables and performing a screen dump

The following example program writes to the disc, and you will therefore need a blank (formatted) disc or working disc inserted in the drive to run the program. The program draws a Union Jack flag, and then saves the whole screen to disc.

```
10 dumpfile$="flagdump.srn"
20 MODE !:BORDER 0
30 DIM colour(2)
40 FOR i=0 TO 2
50             READ colour(i): REM get colours from DATA statement
60             INK i,colour(i)
70 NEXT
80 ON ERROR GOTO 430
90 OPENIN "param.dat" ' test if file exists
100             CLOSEIN:ON ERROR GOTO 0
110             IF errnum=32 AND DERR=146 THEN CLS:
             GOTO 160 ' file doesnt exist
120             CURSOR I:PRINT "Do you want to overwrite
             old file? Y/N ";
130             a$=INKEY$:ON INSTR(" YN",UPPERVa$))
             GOTO 130,150,140:GOTO 130
140             PRINT a$:PRINT "Program abandoned":END
150             PRINT a$:CURSOR 0
160             OPENOUT "param.dat"
170             WRITE #9,dumpfile$,1: REN save filename and mode
180             FOR i=0 TO 2
190             WRITE #9,colour(i): REM save colours
200             NEXT i
210             CLOSEOUT
220             CLS
230             gp=l:GRAPHICS PEN gp:w=125
240             x=-65:a=240:y=400:b=-150:GOSUB 400
```



```
250 Y=O:b=150'GOSUB 400 260 x.=,575:a=-240:y=400:b=-150:GOSUB 400 270 yeO:b=150:GOSUB 400 280 gpFR2:GRAPHICS PEN gp:w=40 290
a=2,QO:x=40:y=400:b=-150:GOSUB 400 300 x=O:y=O:b=150:GOSUB 400 310-a=240:x=640;Y::~'-0.1b=150.:GOSUB 400
.,320xxr-600:y=400!b=-150:GOSUB 400 330 ORIGIN 0,0,256,380,0,400:CLG 1 340 ORIGIN 0,0,0,640,150,250:CLG 1 350 ORIGIN
0,0,280,352,0,400:CLG 2 360 ORIGIN.,0,0,0,640,168,230:CLG 2 370 SAVE'danofite$,b,&COOO,&4000 380 DATA 2026,6 390 END 400 MOVE
x,y:DRAWR a,b:DRAWR w,O:DRAWR -a,-b 410 MOW(x+a12+w12,y+b12:FILL gp 420 RETURN 430 errnum=ERR:RESUME NEXT run
```

Note the use of - D A T and . S R N filetypes. These filetypes are used to remind us of what is in the file, rather than because they have any inherent significance. The file **PARAM.DAT** will be an **ASCII** data file without a header, whilst **F L A G D U M P . S R N** is an **AMSDOS** binary file with a header.

Note how the program deliberately tries to read from the file **P A R A M . D A T** before writing to it, in order to establish if the file already exists. If the file does NOT exist, then an error is reported by **BASIC**; the error is trapped by the program, and execution proceeds without interruption. If the file **DOES** already exist, then no error is reported, and the program automatically asks if you wish to overwrite the existing file.

The particulars of the screen dump, namely the screen mode, palette colours and name of file containing the actual information, are saved into a parameter file. This illustrates the use of a data file to **W R I T E** program variables (d u m p f i l e \$) and constants ( 1 ), saving them for use by another program.

### Loading the screen back

The following example is a general purpose screen dump displaying program, using a parameter file to control its action. Note how variables are I N P U T from the data file, with the E O F function allowing automatic variation in the size of the file. It is important that the screen dump displayed by this program was saved with the screen in a known position in memory, otherwise the result will be 'skewed'. This is ensured by the saving program executing a M O O E command and thereafter being careful not to cause the screen to scroll.

```
10 DIM colour(15): REM Provision for 16 colours 20 OPENIN "param.dat" 30 INPUT #9, filename$,screenmode 40 i=0 50 WHILE NOT EOF 60 INPUT #9,colour(i) 70 INK i,colour(i) 80 i=i+1 90 WEND 100 CLOSEIN 110 MODE screenmode:BORDER 0 120 LOAD filename$ run
```

### Summary of AMDOS external commands

|A

|A

**COMMAND:** Set default drive to Drive A. Equivalent to 1 D R I V E with parameter A. (The main drive within the computer is Drive A.)

|B

|B

**COMMAND:** Set default drive to Drive B. Equivalent to 1 D R I V E with parameter B. (The main drive within the computer is Drive A.)

---

## 1 |CPM

1 |CPM

COMMAND: Switch to alternative disc environment by loading the operating system from a system disc. The operating system supplied with the computer is CP/M Plus.

This command will fail if the computer's drive does not contain a system disc with CP/M.

## 1 |DIR

1 D I R [,<string expression,]

**IDIR,"\*.BAS"**

COMMAND: Display the disc directory (In CP/M style), and free space. If the string expression is omitted, the wild-card is assumed.

1 |DRIVE

1 D R I V E ,string expression>

**IDRIVE,"A"**

COMMAND: Set the default drive. This command will fail if AMSDOS is unable to read the disc in the requested drive.

1 =a

1 E R A ,string expression>

**|ERA,"\*.BAK"**

COMMAND: Erases all files which match the filename and which are not Read/Only. Wild cards are permitted.

---

|REN

1 R E N , <string expression> ,string expression,

**IREN,"NEWNAME.BAS","OLDNAME.BAS"**

**COMMAND:** Give a file a new name. Any other file with the chosen new name must not already exist. Wild cards are not permitted.

The U S E R (see 1 U S E R ahead) parameter may be specified within the string expression>s to override any default settings. For example, the command **IREN,"0:NEW.BAS","15:OLD.BAS"** will rename the file in **USER 15** called " O L D . B A S %" to a file called " N E W . 8 A S " in U S E R 0, regardless of any default or previously issued settings of 1 U S E R.

|USER

1 USER ,integer expression>

**|USER,3**

**COMMAND:** Determines which of up to 16 individual sections of the directory (in the range 0 to 15), disc functions (e.g. C A T, L O A D, 1 D I R etc.) are to be performed on.

A file on one U S E R number may be transferred to another, by a 1 R E N command. For example, **IREN,"15:EXAMPLE.BAS","0:EXAMPLE.BAS"** transfers a file from U S E R number 0 to U S E R number 15, although the name of the file itself (E X A M P L E . B A S) is not changed.

---

## **Copying files from disc to disc**

### **AMSDOS files with headers**

It is possible to copy this type of file in the CP/M environment using P I P (see part 2 of this chapter). Any file created by AMSDOS which has a header record (see 'AMSDOS headers', previously described) will be copyable as a whole - disc to disc - but in general the contents of the file will not be understood by any CP/M programs.

### **ASCII files**

Files created by AMSDOS without headers are generally in ASCII, and are both copyable and understood by CP/M programs. In particular it should be possible to exchange ASCII program files, ASCII data files and ASCII text files freely between AMSDOS and CP/M programs.

### **Read/Only files**

It is possible, using CP/M, to set any file to be Read/Only, and/or set to a special System status in the directory. Such attributes can only be set or reset in the CP/M environment, but are honoured by AMSDOS. For further details, see part; 2 of this chapter (SET utility).

## Reference guide to AMSDOS error messages

When AMSDOS cannot carry out a command for some reason, it will display an error message. If there is a problem with the hardware, the error message is followed by the question:

! Retry, Ignore or Cancel?

R causes the operation to be repeated, possibly after the user has taken some preventative action.

I causes the computer to continue as if the problem had not occurred, which will often lead to unexpected and possibly inconvenient results.

C causes the operation to be cancelled, which will often lead to a further error message.

### Error message meanings

Unknown command

.... The command is not spelt correctly.

Bad command

.... The command cannot be carried out for some reason. Syntax error or inappropriate hardware configuration.

<filename> already exists

.... User is trying to rename a file with a name that's already in use.

<filename> n o t f o u n d

.... File does not exist.

Drive A: directory full

.... No more room in the disc directory for a new entry.

Drive <drive>: disc full

.... No more room on the disc for new data.

---



Drive drive,: disc changed, closing filename,

.... Disc has been changed with files still open on it.

<filename> i s read only

.... File cannot be operated on because it is Read/Only. Files can only be set Read/Only or Read/Write in the CP/M environment.

Drive drive,: disc missing

.... No disc in drive, or disc is not seated and spinning properly. Recommended action is to eject and re-insert the disc, then type R.

Drive drive>: disc is write protected

.... Attempt has been made to write on a disc with the Write Protect hole open. To use the disc, eject, close the write protect hole, re-insert the disc, and then type R.

D r i v e drive>: read f a i L

.... Hardware error reading disc. Recommended action is to eject and re-insert the disc, then type R.

Drive drive>: write fail

.... Hardware error writing disc. Recommended action is to eject and re-insert the disc, then type R.

Failed to load CP/M

.... Read error loading CP/M during the 1 C PM command, or you are not using a valid system disc containing CP/M Note that trying to load CP/M from a Data format disc will produce 'read fail' error.

---





## Part 2: CP/M

### CP/M PLUS

#### Subjects covered:

- Introduction to CP/M
- Booting CP/M Plus
- Direct Console Mode
- Transient programs
- Managing peripherals

CP/M Plus is a disc operating system. It is a special program which gives you access to the full power of your 6128. The 128K of RAM is used to the full, with over 61K available to user programs. CP/M incorporates random access to data files and the 6128 implementation includes a sophisticated VDU emulator.

Because CP/M is available for so many different computers, it means that there are thousands of applications packages available for you to choose from, and a whole wealth of knowledge and experience for you to draw upon.

#### Introduction

The CP/M operating system provides a way for you to communi with the computer, and manipulate files and peripherals. Special commands (and programs on the disc called utilities) are there to help you get on with the main task - which is running your applications programs with your data.

Needless to say, it is possible to become quite an expert at how CP/M and all the various utilities work, and at times, such expertise can be very useful in helping out when we get into trouble. Most of us, however, only need to know enough to get us started, and the rest of this chapter is designed to introduce all the features and facilities without obscuring the vital fads with too many frills.

Whereas BASIC has its Direct Mode and the 'R e a d y' prompt, CP/M has a Direct Console Mode and is identified by the A > or B > prompt. Certain built-in commands are available but the majority of the 'housekeeping' work is done by loading and running 'transient programs'. They are called transient because they are only in the computer (loaded from the disc) while you are using them, as opposed to being built-in.

As well as standard CP/M error messages, the system also generates a number of specialised hardware error messages which can be distinguished by the fad that they normally appear on the bottom line of the screen in a banner form.

CP/M Plus on the disc

---

The major part of CP/M Plus resides in a special file which has the filetype'. EMS', and is found on Side 1 of the system discs package. The computer loads CP1M from this file into the memory using a two stage process.

Initially, the AMSDOS command 1 C PM loads the first sector of track 0. On a system disc this sector has been arranged to be a program which then loads the . E M S file into memory. The remainder of the system tracks are unused.

Early monding start profile

During the loading process, when CWM Plus is first activated, if the file P R O F I L E . S U B is present on the disc, then the instructions in that file are S U B M I Tted. This facility can be used to re-arrange the keyboard, customise the screen output, initialise a printer and even auto-start an application program. In chapter 4 we saw how to rename the profile file supplied on Side 1 in order to activate it.

While the profile file is working, a small temporary file is opened on the disc, which must therefore be write-enabled. This is why the master disc itself cannot include a recognisable profile file.

---Ar-biks d21

Profile files can be constructed using a word processor, text editor (such as E D . C O M), or even from BASIC. The small BASIC program below could have been used to generate the file P R O F I L E . S U 8:

```
10 OPENOUT "PROFILE.SUB"
20 PRINT #9,"SETKEYS KEYS.CCP"
30 PRINT #9,"LANGUAGE 3"
40 CLOSEOUT
```

Console control codes

In the CP1M environment, a variety of special key operations are used. These keystrokes replace the action of the [Escape and cursor keys used in AMSTRAD BASIC. The control codes below are assigned after running the command:

SETKEYS KEYS.CCP

.... where both the transient program S E T K E Y S . C O M and the command file K E Y S . C C P are found on Side 1 of the system discs package.

Control

Code	Key	Action
[CONTROL]A		Moves the cursor one character to the left.
[CONTROL]B	[CONTROL]D	Moves the cursor to the beginning of the

or line. If the cursor is already at the

beginning, moves to the end.  
[CONTROL] [ESC] Abandon.

[CONTROL] [RETURN] Physical carriage return.

[CONTROL] 0 Moves cursor one character to the right.

[CONTROL] JCLRJ Deletes character under cursor.

[CONTROL] [DEL] Backspace delete.

[CONTROL] [TAB] Moves cursor to the next tab stop.

[CONTROL] Send command line.

Chapter 5 Page 16 ANSDOS and CP/M(6128only)

[CONTROL] [CLR] Delete to end of line.

[CONTROL] [RETURN] Send command line.

.... or....

[ENTER]

[CONTROL] Hardcopy toggle. Turn on/off of all screen output to printer.

[CONTROL] Resume screen output.

[CONTROL] [ENTER] Retype command line.

[CONTROL] [ESC] Halts the screen output from CP/X Use [CONTROL] to resume.

[CONTROL] Discard line.

[CONTROL] [COPY] Recall last-typed command line.

[CONTROL] [DEL] Delete from beginning of line to cursor.

[CONTROL] End of text.

## File names

Many of the commands take file names as a parameter, and where specified, the file name may contain wild-cards (see the section entitled 'Wild cards' in part 1 of this chapter). All file names will be forced to upper case.

Direct Console Commands and most utility programs do NOT require that file names are contained in double quotes ". Remember that file names can have an A : or B prefix to force CP/M to use the appropriate drive.

Therefore a typical CP/M command is:

TYPE KEYS..UP

---

.... where T Y P E is the function required, meaning 'display on the screen', and K E Y S . C C P is a filename specifying which file we wish to see.

AMSDOS and MIM(6128 only)          cmpter 5 paw 17

### Switching default drives

If you have an additional disc drive connected, then it is possible to switch the default drive selection between Drive A and Drive B by typing A : or B : at the B > or A > prompt. That prompt, of course, tells you the current default drive. Adding the A: and B : prefix to filenames overrides, but does not reset, the default drive setting.

### Direct Console Comands

There are a number of Direct Console Commands which can be typed at the A > or B > prompt. Each command can be abbreviated, and 'although the simple functions described below are genuinely built-in, there are also more sophisticated transient commands with the same name.

#### DIRcommand

D I R lists the D I Rectory of the disc. The filenames are not sorted into any particular order, but the position of the filename in the D I R display indicates the position of that file's entry in the disc directory. Wild cards are permitted. Files set with the 'S Y V attribute will not be listed.

DIR        will list files on the default drive.  
DIR B:    will list files on Drive B :  
DIR \*.BAS                will list files of type . B A S  
DIR B:\*.BAS              will list files of type . B A S on Drive B  
DIR PIP.,COM             will list only the file P I P . C O M (if it exists).

#### DUtMorDlItSconmmd

D I R S V S or D I R S lists only those directory entries with the 'S Y S' attribute set. Otherwise it operates as D I R. The S Y S attribute is described later.

#### ERMorEP.Aconunand

E R A is used to E R A se files from the directory. Only the directory entry is erased, so the data is still in the data section of the disc until the space is re-used by another file; however, the information is nevertheless not recoverable. Wild card filenames are permitted, and if used, E R A will ask for confirmation. E R A does not list the filenames that are deleted. If any file about to be, ~ is

---

found to be Read/Only then the command will abort. The Read/Only attribute is described later.

ERA PIP.COM                    will erase the file P I P . C O M  
ERA B:PIP.COM                 will erase the file P I P . C O M on Drive B  
ERA \*.BAS                     will erase all . B A S files

~AM or UN conunmd

R E N allows you to R E Name an existing file. The new filename is specified first, followed by = then the existing filename. If the new filename already exists, an error message will be displayed.

Wild cards in the filenames are outside the scope of R E N as a built-in command, and require R E N A M E . C O M, the transient program.

REN NEWNAME.BAS=OLDNAME.BAS    will rename the file O L D N A M E . B A S toNEWNAME.BAS

REN 8:NEWNAME.BAS=OLDNAME.BAS will rename the file O L D N A M E . B A S to N E W N A M E . 8 A S, on Drive B.

llMorT"commia.nd

T Y P E asks for the specified file to be T Y P Ed onto the screen. If the file is not an ASCII textfile, unpredictable and possibly undesirable side-effects may occur.

TYPE KEYS.CCP

.... will display the file K E Y S . C C P

U= or USE conunmd

USER changes the current user number. CP1M Plus starts with the current user number set to 0. Normally you can only access files identified with the current user number, thus providing a way ofpartitioning-offthe directory.

A file in User 0, with the 'S Y S' attribute set can be accessed from all other user numbers. This is a very powerful facility for making utility and applications programs available to all user numbers, without having an actual copy of them in each user area.

USER 3

.... will set the current user number to 3.  
Transient c                    --M

To perform more sophisticated file management than permitted by the Direct Console Mode, you must employ one of the various utility programs provided. These are invoked merely by typing the program name, possibly followed by a filename and/or some parameters. You have probably already used D I S C K I T.

---

The commands fall into a number of categories as indicated below.

The commands DISCKIT, SETKEYS, SETLST, SETSIO, PALETTE, LANGUAGE and AMSDOS, are designed by AMSTUD, and work exclusively on the AMS~ system. They will not work on any other CP/M system.

It is possible to enter multiple commands on a single line, where the commands are separated by an exclamation mark. For example:

```
LANGUAGE 3!SETKEYS KEYS.WP
```

Peripheral Management  
np

DISCKIT is a complete disc formatter, copier and checker. It is quicker to format whilst copying than to format and then copy. Comprehensive menus indicate which keystrokes (mainly from the function key area of the keyboard) should be made. Vendor format discs are a special form of System disc intended for software distribution, although Data format discs are perhaps more suitable for this activity in the CPIM Plus environment.

WARMG

The licence agreement for your CP/11 (which is electronically serial-number encoded) permits its use on a single computer system only. In particular this means that you are prohibited from giving any other person a disc with YOUR serial-numbered copy of CRM on it. Because copies of Side 1 of your master package will include your CRM (in the .EMS file) on it, you must be careful not to sell, exchange, or in any other way part with, any disc with that file on it.  
Language characters

The 6128 has a full set of international characters. The LANGUAGE command exchanges certain of these characters so that simple software can display alternative and accented characters on the screen. Further details are contained in part 16 of the chapter entitled 'For your reference .....

The command:

```
LANGUAGE 3
```

.... will set the 6128 to the UK displayed character set, which swaps the # and £ signs

(compared to the default USA set).

Colours

The default colours of CP/11 Plus on the 6128 (with colour monitor) are bright-white characters on a blue background. These colours can be changed by the PALETTE

---

command, which takes a number of parameters, one for each ink - ink 0 includes the background and border area; ink 1 is for the text. Each colour is represented by a number in the range 0 to 63, which can be used to gauge the colour's intensity

(brightness) on a white monitor.

It is possible to specify any number of inks, from one to sixteen, although only the first two will be visible in 80 column mode.

The command:

```
PALETTE 63,1
```

.... will reverse the normal settings of ink 0 and ink 1, giving a background of bright-white (63) with text in blue (1).

Use the values in the following table to select the colours (or intensities) that you require. You can use either the hex or decimal representation as you prefer.

Colour	Hex	Decimal	Colour	Hex	Decimal
Black	&00	0	Pastel blue	&2B	43
Blue	&02	2	Orange	&2C	44
Brightblue	&03	3	Pink	&2E	46
Red	&08	8	Pastel magenta	&2F	47
Magenta	&0A	10	Bright green	W0	48
Mauve	&0B	11	Sea green	02	50
Bright red	&0C	12	Bright cyan	03	51
Purple	&0E	14	Lime green	&38	56
Bright magenta	&0F	15	Pastel green	WA	58
Green	&20	32	Pastel cyan	0B	59
Cyan	&22	34	Brightyellow	WC	60
Skyblue	&23	35	Pastel yellow	WE	62
Yellow	&28	40	Brightwhite	&W	63
White	&2A	42			

## Keyboard

The codes generated by the keyboard can be altered by the SETKEYS command. This allows suitable codes to be assigned to keys and to expansion tokens. The actual codes must be written into a file, whose name is then presented to the SETKEYS command. The command file can be created by a text editor, by PIP, or even from BASIC. For example:

```
SETKEYS KEYS.TST
```

.... where the file KEYS.TST contains:

```
E &8C "DIRM" expansion token 12
8 N S C "TH backspace = [CONTROL]H, ASCII 08
```



.... will firstly redefine the [CONTROL] [ENTER] expansion token (represented by &SC) to be the string D I R [RETURN], and then turn the cursor left key 0 (key number 8) into a backspace.

Standard files provided with the 6128 are K E Y S . C C P for CP/M command editing and K E Y S . W P which is suitable for many word processors.  
Printers

Initialising printers can be performed by the command:

SETLST filename,

.... where filename, contains the string, or strings to send to the printer. As with the command file for S E T K E Y S, control codes can be represented by:

T character>

.... or by....

T ',character value,'

.... or by....

l ',control code name,'

.... where control code names are ESC, FF, etc., as shown in the table of ASCII characters in the chapter entitledFor your reference  
.....

A useful initialising code for many printers is the value 15, setting the printer into condensed printing.

The command:

PRINT #8,CHR\$05)

.... would set this in BASIC. In CP1M, issue the command:

SETLST CONDENSE

.... where the file C O N D E N S E contains any one of the following as a single line of text:

TISI9 TO T 1W

T1151

.... which are all interpreted as the decimal value 15.

Some applications programs require a screen that is 24x80. The command S E T 2 4 X 8 0 may be used to set the screen size.

The commands:

---

SET24X80

.... or....

SET24X80 ON

.... turns 24 x 80 mode on, and:

SET24X80 OFF

.... turns 24 x 80 mode off.

The normal 6128 full screen size is 24 x 80, with the bottom line reserved for status messages. Turning off 24 x 80 mode will only be noticeable if the status line is also disabled. Refer to part 15 of Chapter 6 for details of how to turn the status line on and Off.

### SerWinterface

Built into CPIM Plus is support for a single channel serial Input/Output interface (RS232). Its vital statistics can be examined by typing the command S E T S 10 (with no parameters):

SETSIO

.... or can be set using a command which may include any (or all) of the selections:

SETSIO, RX 12000 TX 750 PARITY NONE0 STOP 10 BITS 8, HANDSHAKE ON, XOFF OFF

.... which would set a new configuration.

Baud rates and X 0 N/X OFF status are also affected by one of the assignments possible with the D E V I C E command. D E V I C E deals with logical and physical devices. Logical devices are indicated by a : colon. To examine all the current device attributes, type:

DEVICE

.... and the attributes can be altered by commands such as:

DEVICE SIO[12001 .... sets S 10 to 1200 baud. DEVICE SIOEXONI .... toturnonSIO XON/XOFF protocol. DEVICE SIOCNOXON] .... toturnoffSIO XON/XOFF protocol.

The connections between logical and physical devices can be altered. Normally C 0 N : is set to C R T (keyboard/screen), A U X : is set to S 10 (the optional serial interface), and L S T : is set to L P T (the Centronics printer interface). The command:

DEVICE LST:=SIO

.... will send the printer output to the serial interface (iffitted).

---

Note how this is a channel re-direction, not to be confused with the file copying facilities provided by P I P. The two commands G E T ~filename, and P U T ~filename, re-direct console input or output, and printer output, instructing them to use a file, rather than the device channel.

PIP

The P I P utility (Peripheral Interchange Program) allows you to transfer information between the computer and its peripherals.

In general, the form of the command is:

PIP destination~==~source,

The ~source, and destination, can be either a filename, with wild-cards allowed in the source, or a logical device. The following logical devices may be used:

As Source	As destination		
CON:	console input	CON:	console output
AUX:	auxiliary input	AUX:	auxiliary output
EOF:	an end-of-file mark	LST:	printer
	PRN:		a printer with added tab expansion, line numbers and page breaks

Examples of P I P:

PIP B:=A:\*.COM

.... copy all C O M file from Drive A : to Drive B  
.Each disc can be assigned a label, or name, and also a password.  
That password will protect the directory itself, rather than the files named in the directory. The individual files can Also be assigned a password.

```
SET ENAME=ROLAND3
SET EPASSWORD=SALLY3
SET EPROTECT=ON3
```

.... act on the default disc.

```
SET *.*LPASSWORD=SALLY1
SET *.*EPROTECT=READ3
```

.... act on files on the default disc (the wild cards used here indicate 'all files').

Date and time stamping can be activated by the I N I T D I R command (on Side 2 of your system discs package). The commands:

---

```
INITDIR
SET ECREATE=ON1 .... or.... SET LACCESS=ON1 .... and....
SET IUPDATE=ON1 .... together with....
DIR [FULL]
```

.... will initiate and display date and time stamping on the default drive. Typing in:

```
DATE SET
```

.... is then required when time CP/M Plus is started, to set the clock. Once set, the clock will keep reasonable time, updated automatically by the 6128 and inspected by:

```
DATE ... n ....
DATE,-CONTINUOUS
```

```
WI~G
```

If passwords, disc labelling, or date and time stamping have been activated, then it is recommended that the disc is NEVER subsequently written to by AMSDOS which does not support these facilities. Normally, files will be accessed on the default drive only, unless a particular drive is specified. The command:

```
SETDEF *,A:
```

.... (where \* refers to the default drive) instructs CP/M (when it searches for files) to search firstly on the default drive, then on Drive A:. In other words, if the default drive is B : then files will be found automatically even if they exist on Drive A

The commands:

```
SETDEF [PAGE] .... and....
SETDEF INOPAGE]
```

.... turn on and off the automatic pagination of console display.

Remember that most of the facilities of DEVICE, SET, and SETDEF particularly where they refer to drives (rather than files or discs) need to be set up, together with the date, each time CP/M Plus is started. This is a prime application for a suitable PROFILE .SUB file.

SUBMIT is required to execute files of individual commands automatically. The contents of the command files are text, and it is possible to include input to programs if the first character on those lines in the .SUB file is <

The drive size, amount of space, and number of directory entries left on a disc, together with the user areas containing files, and the disc label (if any) may be displayed by various combinations of the SHOW command:

---

```
SHOW B:
SHOW B:ELABEL1
SHOW B:EUSERS3
SHOW B:1DIR1
SHOW B:EDRIVE3
```

.... all specify statistics from Drive B

Exit-ingCP/M Plus

AMSDOS

This program relinquishes control from CP1M and returns to the built-in AMSTRAD BASIC, from which the AM9DOS disc commands will be available.

P0

zinter Port

13 12 11 10                    4 3 2 1

~ ' . 's .

PIN 1	STROBE	PIN 17	GND
PIN 2	DO	PIN 18	GND
PIN 3	DI	PIN 19	GND
PIN 4	D2	PIN 20	GND
PIN	D3	PIN 21	GND
PIN	D4	PIN 22	G D
PIN 7	D5	PIN 23	GND
PIN 8	D6	PIN 24	GND
PIN 9	D7	PIN 25	GND
PIN 11	BUSY		
PIN 16	+5V	All other pins	NC

Part 10: Printers

Printer interfacing

The 464/6128 allows the connection and use of an industry standard Centronics style interface printer.

The printer cable is simply constructed as a one-to-one connection between the PRINTER socket at the rear of the computer, and the connector on the parallel printer.

The actual pin interface details are illustrated in part 9 of this chapter.

The computer uses the BUSY signal (pin 11) to synchronise with the printer, and will wait if the printer is OFF-LINE.

---

There are no user set-up commands required, and the output is directed to the printer by specifying stream # 8.

Although the 464/6128 PRINTER port is envisaged for use with low cost dot-matrix printers; with a suitable interface it will support daisywheel printers, graphics plotters, and multi-colour ink-jet printers. The key to compatibility is the standard parallel interface.

#### Printer configuration

A facility is provided whereby special characters which may appear on the screen and which are supported by the printer, will be printed even though the character codes for the screen and printer may be different. The majority of these symbols will only be available when the printer is switched to one of its foreign language modes. For example:

```
PRINT CHR$(&AO)
```

```
PRINT #8,CHR$(&AO)
```

^ is printed on the printer.

This works even though the character code for a circumflex accent on the printer is & 5 E. In other words, the printer routine has recognised & A 0 as one of the codes held in a printer translation table, and has translated it to & 5 E so that the same character appearing on the screen will be printed by the printer. The code & 5 E will produce a circumflex accent on a printer no matter which language mode the printer is set to (this is not true for all the characters in the translation table). The other characters in the table are as shown in the following table:

CHR\$	Character	Printer	U.K.	U.S.A.	France	Germany	Spain
	On Screen Translation						
&AO	A	&5E	A	A	A	A	A
&A2		&7B	t	t	t	t	
&A3	L	&23	L	#	#	#	P
t							
&A6		&40		t	t	t	
&AE	z	&5D	t	t	t	t	
&AF	i	&5B			t		

t For the printed character, refer to your printer instruction manual.

The above is an extract from the default translations, which can be changed if required.

Part 11: Joysticks/Paddle Controllers

---

The built-in software in the computer supports either one or two joysticks. These are treated as part of the keyboard, and as such, may be interrogated by I N K E Y and INKEY\$.

Note that in the majority of cases, the main 'fire' button on a joystick is interpreted as 'Fire 2' by the 464/6128.

The functions J O Y ( 0 ) and J O Y ( 1 ) enable direct inspection of the first and second joysticks respectively. The function returns a bit-significant result which indicates the state of the joystick switches at the last keyboard scan.

The table below indicates the values returned by both joysticks. The J O Y values are followed by values for use in statements which take key numbers as parameters (i.e. I N K E Y and KEY DEF).

STATUS	J O Y COMMAND	KEY	VALUES		
BIT	VALUE	FIRST	SECOND	EQUIVALENT	
SET RETURNED	JOYSTICK		JOYSTICK		KEY
Up	0	1	72	48	'6'
Down	1	2	73	49	'T'
Left	2	4	74	50	W
Right	3	8	75	51	CT)
Fire 2	4	16	76	52	'G'
Fire 1	5	32	77	53	'F'

Note that when key values for the SECOND joystick are returned, the computer cannot tell whether those values have been generated by the joystick or by the equivalent keyboard key (indicated in the last column of the previous table). This means that the keyboard can be used as a substitute for the second joystick.

## Part 12: Disc organisation

The BIOS supports two different disc formats: SYSTEM format and DATA ONLY format. Under AMSDOS, the format of a disc is automatically detected each time a disc with no open files is accessed. To permit this automatic detection, each format has unique sector numbers.

3 inch discs are double sided, but only one side may be accessed at a time depending on which way round the user inserts the disc. There may be a different format on each side.

Common to both formats:

Single sided (the two sides of a 3 inch disc are treated separately).  
512 byte physical sector size. 40 tracks numbered 0 to 39. 1024 byte CP/M block size. 64 directory entries.

SYSTEM format

9 sectors per track numbered &41 to &49. 2 reserved tracks.

---

The system format is the main format supported, since CP/M can only be loaded from a system format disc. The reserved tracks are used as follows:

Track 0 sector &41 boot sector for CP/M Plus  
Track 0 sectors &42 to &49

unused

Track 1 sectors &41 to &49

Note that VENDOR format is a special version of system format which does not contain any system software on the two reserved tracks. It is intended for use in software distribution.  
DATAONLYformat

9 sectorspertrack numbered &C1 to &C9. 0 reserved tracks.

Part 13: Resident System eXtensions (RSWs)

External commands were introduced in Chapter 5 (about AMSDOS). Fundamentally, an external command is a way of extending the repertoire of the BASIC by adding new commands signalled by a 1 prefix. The machine instructions for the AMSDOS new commands are included in a ROM, and the necessary housekeeping to add the commands is carried out automatically when the 464/6128 startsBASIC.

It is also possible to add further external commands (after BASIC has started) by loading the machine instructions into RAM. Such new commands are called RSX's and operate in exactly the same way as ROM based extensions. RSX's have to be loaded from disc (or tape) each time the 464/6128 starts (or restarts) BASIC. Normally, RSX's will be used for controlling some sort of intelligent peripheral, such as a light pen or speech synthesiser.

Chapter 7 describes the use of RSX's to access the 6128's second 64K of memory.

Pad 14: Memory

The 6128 contains 128K of RAM, and uses 48K of the System Cartridge ROM. The System Cartridge ROM contains 16K for BASIC, 16K for General Firmware, 16K for Disc Firmware, 64K for a game and 16K is reserved. This is available to BASIC 1.1 as shown below. The 464 differs only by having 64K of RAM. However, the Disc Firmware is automatically ineffective. The first 64K of RAM is nominally divided into four blocks (each of 16K) numbered Block 0 to Block 3. The screen uses Block 3 and the upper section of Block 2 is filled by system variables as indicated.

FIRST 64K OF R~                      SECOND 64K OF RAM

BLC~K 3

---



zi- ~1~           FIR4~---AARE1

ko~~ 2           %

iuw

koCK

& 00,0

NOTE 1: 0~ ON EXTERNALLY FITTED ROMS-4KA6F.C WHEN NO EXTERNAL  
ROMS FITTED

464 & 6128

6128only

464 & 6128

464 & 6128

Memory Map for BASIC 1.1

The user defined characters are initially positioned immediately above HIMEM. HIMEM may be altered by a M E M O R Y command but is also automatically lowered by 4K to create a buffer whilst AMSDOS files are open. The number of user defined characters can only be altered if HIMEM is unchanged since the last time they were set (unless the last time they were set to 'no user defined characters' by a S Y M B O L A F T E R 2 5 6 command). When BASIC starts, the user defined characters are set as if a SYMBOL AFTER 2 40 command had been issued.

It is, therefore, prudent to collapse the user defined character area before permanently changing HIMEM, then re-establishing the user defined characters in the new position. This will allow subsequent programs to alter the SYMBOL AFTER assignment. The example below shows this scheme in operation when FUMEM is lowered in conjunction with loading an RSX.

100       SYMBOL AFTER 256 1 coLLapse user defined characters

110 rsxaddress=HIMEM-rsxLength

120 MEMORY rsxaddress-1

130 LOAD "rsxcode",rsxaddress

140 CALL rsxaddress

Log on rsx

150 SYMBOL AFTER 140

restore user defined

characters

FUIR

MPWAnUATA AREA

AW ~A

A

M

\_r A

-.C RS -

LA-10TMR ROMS -A AKA

---

Memory Map with RSX loaded in recommended position

AdcUtional J10

Most I/O port addresses are reserved by the computer - in particular, addresses below &7FFF should not be used at all.

It is intended that the part of the address A0 - A7 should reflect the type of external I/O device, and that address lines A8 and A9 may be decoded to select registers within the I/O device. Of the remaining address lines, only A10 must be decoded (as low) whilst lines A11 to A15 are high. Thus each device may have registers addressed as &F8??, &F9??, &FA??, and &FB??, where ?? is in the range DC to DF for communications interfaces, and E0 to FE for other user peripherals.

Note that Z80 instructions which place the B register on the upper half of the address bus (A15 - A8) must be used.

SidewaysROMs

Provision is made for additional ROMs to be selected in place of any part of the on-board ROM. The address arbitration and bank selection logic will be contained in a module connected to the expansion bus, but all the signals required are brought to the expansion bus.

Part 15: CP1M Plus Terminal Emulator

In part 1 of this chapter, a table of control characters (together with their respective actions) was illustrated. These actions are performed when text is sent to the screen by BASIC and were chosen both to be simple to use and to reflect the facilities available in the Firmware Text VDU. These facilities are unique to AMSTRAD computers, and software must therefore be adapted to their use.

In the Business and Commercial environment of CP/M Plus software, it is normal to expect a range of 'standard' Text VDU facilities in order that software is easily portable and 'installable' from one machine to another. The CP/M Plus implementation on the 6128 includes a Terminal Emulator which provides facilities very similar to a Zenith Z19/Z29 VDU. The installation procedure for CP/M Plus software will normally include, as standard, an option for this type of terminal.

The facilities offered by the CPIM Plus Terminal Emulator include many of those previously offered by the Firmware Text VDU although different control codes are required.

There are a considerable number of new and more sophisticated operations.

Characters in the range W0 to &FF are displayed at the current cursor position. If the cursor is not at the right-hand column, then it is moved right by one column. If the cursor is at the right-hand column

---

and wrapping is enabled, then it is moved to the left-hand column on the next line, scrolling up if necessary. Characters in the range &00 to &1F are interpreted as control codes as follows:

&07 BEL Soundsabeep.

&08 BS Backspace. Move left one column. If the cursor is on the left-hand column and not on the top row and wrapping is enabled, then it is moved to the right-hand column of the row above.

&0A LF Linefeed. Move cursor down one line, scrolling up if necessary.

&0D CR Carriage return. Move cursor to left-hand column of current row.

&1B ESC Introduce an escape sequence.

All other control codes are ignored.

The following escape sequences are recognised. Any other characters following an escape are displayed, and the cursor advanced. This feature may be used to display the characters corresponding to control codes &00 to &1F. (Note that many applications languages expand the control code &09 (TAB) to a number of spaces and therefore the sequence [ESC][TAB] will often not display the character for &09.)

[ESC]O Disable status line. Disc system messages will appear with CRT output. CRT may use the bottom line of the screen.

[ESC]1 Enable status line. Disc system messages will appear on the bottom line of the screen.

[ESC]2,m Change character set (see part 16 of this chapter). n, is the language parameter which is masked with &07. Certain character matrices in the range &20 to 0F are swapped with other characters in the range &80 to &FF. The action of this command is very similar to that used to control printers which have software-selectable international character sets.

(n,=0 USA  
,n,=1 France  
,n,=2 Germany  
m,=3 UK  
,n,=4 Denmark  
m,=5 Sweden  
,n,=6 Italy

m,=7 Spain

---

[ESC13,m> Change screen mode. m>=screen mode+&20. The value is masked with W to give the mode 0 to 2. Mode 3 is ignored. The screen is cleared, but the cursor position is unaffected.

[ESCIA Cursor up. If on top row, then do nothing.

[ESCIB Cursor down. If on bottom row, then do nothing.

[ESCIC Cursor forward. If on right-hand column, then do nothing.

[ESCID Cursor backwards. If on left-hand column, then do nothing.

[ESCIE Erase page. The cursor position is unaffected. This command clears the entire screen, even if set to 24x80 mode. (Other escape codes will only affect the 24x80 area when 24x80 mode is set).

[ESC1H Home cursor. Move cursor to top row, left-hand column.

[ESCII Reverse index. Move cursor up one row. Scroll screen down if necessary.

[ESCW Erase to end of page. Includes character at cursor position. The cursor position is unaffected.

[ESC1K Erase to end of line. Includes character at cursor position. The cursor position is unaffected.

[ESCIL Insert Line. All rows below and including the cursor line are scrolled down. The cursor row is cleared. The cursor position is unaffected.

[ESCIM Delete line. All rows below and including the cursor row are scrolled up. The bottom row is cleared. The cursor position is unaffected.

[ESC1M Delete character. All characters to the right of the cursor are deleted left one character position. The character at the end of the line is cleared. The cursor position is unaffected.

111 cursor to given position. If position is beyond the edge of the screen, then the cursor is moved to the edge of the screen.

Ch~ 1 ' g position+&20, ~c,=column position+&20. The top the ca. 1 1-0 corner is row 0, column 0. cursor is ' left-hand cot.

---

IESC]b,cp>                   Set foreground colour. Affects all characters  
 on the screen. cp>            is the colour parameter and is masked with &3F then  
 treated as                    three 2-bit numbers each specifying the intensity of  
 one of the                    three primary colours: bits 0,1 for Blue, bits 2,3 for  
 Red, and bits                 4,5 for Green. When running CPIM Plus, the 46416128  
 provides                      three levels of intensity, mapped onto the four  
 specifiable levels            as:  
                               464/6128                   Zero intensity HalfIntensity Full  
 intensity                      Colour bits               00 binary           01 or 10 binary 11  
 binary 1  
 (ESC]c ~cp,                   Set background colour. Affects all the  
 background and border         on the screen. The colour is specified as above.  
 [ESC]d                         Erase to beginning of page. Includes character at  
 cursor                         position. The cursor position is unaffected.  
 [ESC]e                         Enable cursor blob. To prevent unsightly flashing, the  
 cursor is                      not turned on during normal text outputting until 1/10  
 second                         after the last character was written.  
 [ESC]Y                         Disable cursor blob.  
 [ESQ                            Save the cursor position.  
 [ESC]k                         Restore cursor position as saved by [ESQ.  
 [ESC]I                         Erase line. The cursor position is unaffected.  
 [ESC]o                         Erase to beginning of line. Includes character at  
 cursor position.               The cursor position is unaffected.  
 [ESC]p                         Enter inverse video mode. Printable characters are  
 written                        with the foreground and background colours reversed.  
 [ESC]q                         Exit inverse video mode.  
 [ESC]v                         Wrap at end of line.  
 [ESC]w                         Discard at end of line.  
 [ESC]x                         Enter 24x80 mode. Some applications programs may require a  
                               'standard' 24x80 screen. This command will enable such a  
 depend                         screen regardless of the full size of the screen which may  
                               upon machine, country and whether or not the status line is  
                               enabled. The screen is cleared.  
 [ESC]y                         Exit 24x80 mode. The screen is cleared.

#### Part 16: CP/M Plus Character Set

In part 10 of this chapter, a printer translation table was described. The purpose of this table is to convert certain of the characters from the BASIC character set into a form where they can be

---

printed when a language selection is made on the printer. The facility is somewhat limited because very few printers' foreign characters actually appear in the BASIC character set.

Although this printer translation scheme still operates under CPIM Plus, the character set has been enhanced to allow an almost complete correspondence between the screen characters and the printed characters. (The one missing screen symbol is the Swedish Currency Symbol replacing the \$). The table below confirms the arrangement:

23 24 40 5B SC 5D 5E 60 713 7C 7D 7E

USA

France

pot% k

Germany

UK

Denmark le \* 110119 A \*. an \* M a

Sweden

,

Italy a A 01 RES N\_ A 1 lk a ii A 1

Spain 9 CR

CPIM Plus International Character Set

Operating the machine in a foreign environment requires two actions:

1. The printer itself should be set to the required language (often by small (DIP) switches, though some printers allow control codes to be sent).

2. The required screen character set must be invoked, either by the transient command:

L A N G U A G E <n>

.... or by sending

[ESC12,w

.... to the Terminal Emulator.

In practice, the initialisation can be performed as part of the P R O F I L E . S U B operation, using the L A N G U A G E and S E T L S T facilities. CP/M Plus is shipped with the USA environment, mainly because the keyboard displays a # over the 3 key. It is common practice to operate in the UK environment when doing word processing.

---



Charactem 0 to 127 (M0 to VF)

Charactem 128 to 255 (M0 to W)

### The Standard CP/M Plus Character Set (USA)

Software which is capable of operating with 8 bit characters may use this character set to access all of the foreign language characters at once, without having to specifically change language. In this way, all the 'bar and brackets'type characters will be available as well.

Please notp-that there is currently very little 8-bit software available, and that charactersin the ranges 0 to 31 and 128 to 255 will only ever appet in the form shown above on the screen itself, Printers, on the other hand, will have their own, and different, ideas about what these symbols will look like.

#### Chapter 7

More About the Bank M r....

Extensions to BASIC allowing access to the second 64K bank of RAM. This chapter applies only to the 6128.

Subjects covered:

- Storing screen images
- Pseudo-file operation

The memory map for BASIC 1.1 (shown in Chapter 6 part 14) indicates that 64K of the 128K RAM is unused. The BASIC and firmware themselves reside in a ROM which, together with the disc ROM, increase the normal memory provided from 64K to 112K (64K RAM, 48K ROM).

Each section of 16K is called a'block', and any particular selection of four blocks (to make 64K in total) is called a %ank'. The technique of selecting blocks is therefore called'bank switching'.

The Z80 microprocessor can accomodate only 64K of memory at any one time, so the operating system contains instructions to switch the fliTnware ROM into play instead of Block 0 of RAM, and to switch either the BASIC ROM or the disc ROM into play instead of Block 3. This switching takes place automatically when the BASIC or firmware is required. Bank switching of RAM merely extends this concept to include the overlaying of RAM, rather than ROM. The switching is taken care of by an assembler program.

The program B A N K M A N . B A S is provided on Side 1 of the system discs package. If the program is run after BASIC has started, it will install the standard bank management RSX code. Hence, the program is known as the'BANK MANAGER'.

One use for the memory in the second 64K is as temporary storage space for picture screens. Such applications may include, for ex~ngle, a 'Screen Designer' program which dotes a number

---



of different screens, or a vid65 game which may have a number of different screens already prepared.

Another use for the second 64K is as an extension to the variable workspace, which can be regarded as either an extension to the string array space, or as a simple 1Mdisc'.

Part 1: Storing screen images

Choose your screen....

The BANK MANAGER is able to switch out Block 1, and switch in one of the four blocks from the second 64K in its place. The diagram below illustrates this action. Note how each of the blocks from the second 64K occupies the same address space (&4000 to WFIPF). The contents of Block 1 (probably the middle of your BASIC program!) is preserved, and is restored when the BANK MANAGER has finished. There are three other bank selections possible (apart from the five shown below), but they are only useful to the implementation of CP1M Plus.

REEN 1

K 3        SCREEN 4                    REEN

SCREEN3

          SCREEN2                    ----- t~O\_C K 7  
CK 2                                BLOCK 6

BLOCK

BLOCK 4

B----,- \_

', KO

Hardware bank switching

The BANK MANAGER supports two commands for moving screen-fulls of information between one block and another. Blocks 4 to 7 are switched in and out automatically as required, and the memory map is left with Block 1 switched in.

The command:

1 S C R E E N S W A P , [~screen section> , ]<screen number> ,screen number>

.... exchanges the contents of two blocks, whereas:

1 S C R E E N C O P Y , [,screen section> , ]<destination screen number>

,source screen number>

---

.... copies the contents of one block into another block.

The optional screen section > parameter causes the software to copy only 1/64th of the block (256 bytes out of 16K). screen section> therefore takes values in the range 0 to 63. This mode of operation is useful if it is required to interleave any other processing with screen moving. A screen-swap can take around 150/300ths of a second (150 TIME-counts).

The screen number,s required are 1 (the normal screen), and then either 2,3, 4, or 5. Copy and swap operations where the source or destination is screen 1, will operate much faster. Be careful of the screen hardware-roll effect, as experienced when dealing with disc screen dumps. It should be arranged that all the screen images are constructed (and viewed) with screen 1 set to the same hardware position. The simplest (default) position is that set by a M O D E command.

Try out the screen switching commands....

First, run the BANK MANAGER program from Side 1 of your system discs package by:

```
RUN "BANKMAN"
```

Thentype:

```
MODE1
```

Screen clears. Now type:

```
          This,is the orig-inal screen
ISCREENCOPY,3,1 'Send original screen to memory 3
```

```
CLS
```

Screen clears again. Now type:

```
1 This is the intermediate screen
ISCREENCOPY,2,1      Send intermediate screen to memory 2
ISCREENSWAP,2,3      Exchange memory 2 and memory 3
ISCREENCOPY,1,3      Restore intermediate screen from memory 3
ISCREENCOPY,1,2      Restore original screen from memory 2
```

Finally on this subject, the last part of Chapter 9 includes a comprehensive 'Screen Designer' program, which incorporates the screen switching facilities provided by theBANKMANAGER.

.A 0

Part 2: Pseudo-fd\*e operation

C=esriledagain....

---

When regarded as a RAMdisc, the whole of the second 64K is divided into a T~ile' comprising a number of fixed length records. The record length can be between 0 and 255 bytes, although 2 bytes is recommended as the minimum. Once the, RAMrecord length> has been established, each record can be accessed by its RAMrecord number,. It is perfectly acceptable to write the RAMfile using one record length, and read it back using another.

NOTE - The RAMfile must contain only data; there is NO facility for it to contain program instructions.

As is common with random access disc filing schemes, there is the concept of 'current record number'. This provides, in effect, a default record number, which is particularly useful when auto-stepping through the RAMfile.

The command:

```
1 B A N K O P E N ,RAMrecord length,
```

... sets the fixed length of all records, initialises the current record number to zero, but DOES NOT clear the memory in any way.

The command:

```
1 B A N K W R I T E , @,return code, ,string expression> [ ,RAM~rd number>]
```

... writes the, string expression> to the RAWle.

The RAMrecord number, specifies which record is written. If the parameter is omitted, then the current record number is used. The current record number is then set to point at the next record.

If the ~string expression, does not completely fill the record, then old characters (which have not been overwritten) will remain at the end of the record. If the ~string expression, is longer than the record, then the excess characters are discarded to avoid spillage into the next record.

~return code, is an integer variable which returns the record number written to (if the operation was successful), or a negative error code if the write operation has failed in some way.

- 1 End ofFile error. The requested record number's address exceeds 64K.

- 3 Bank switching failure (should never happen).

Examples:

```
IBANKOPEN,10 1BANKWRITE,@r%,"123 testing",O IBANKWRITE,@r%,w$
```

The command:

---

IBANKREAD. @,return code,, @,string variable~[,,,RAMrecord numberd

.... reads a record into the,string variable) from the RAMfile.

The RAMrecord number, specifies which record is to be read. If the parameter is omitted, then the current record number is used. The current record number is then set to point at the next record.

If the record contents do not completely fill the string variable) then old characters (which have not been overwritten) will remain at the end of the ~string variable,. If the record contents are longer than the length of ~string variable,, the excess characters are discarded, as it is not possible to increase the length of a string variable during an external command.

,return code, is an integer variable which returns the record number read from (if the operation was successful), or a negative error code if the read operation has failed in some way.

- 1 End ofFile error. The requested record number's address exceeds 64K.

- 3 Bank switching failure (should never happen).

Example:

```
IBANKREAD,@r%,i$,O  
Searchdng....
```

It is possible to search through the stored records looking for a particular entry.

The command:

```
IBANKFIND , @,return code, searched for string~  
[,,,starting record number,[ , &finishing record numberd]
```

.... will scan all the RAMrecords. The starting record number, specifies from which record the search starts. If the parameter is omitted, then the current record number is used.

The search will proceed, in steps of ~RAMrecord length,, throughout the whole of the second 64K of memory until a match is found.

If a finishing record number, is specified, then the search will terminate after that record has been tested (unless a match has been found before that point).

If the search is successful, then the current record number is set to point at the record where the match was found (otherwise it remains unaffected).

---

,return code, is an integer variable which returns the record number where the match was found (if the operation was successful) or a negative error code if the search operation has failed in some way.

1 End of File. The starting record number's address exceeds 64K or exceeds the finishing record number.

2 No match found.

3 Bank switching failure (should never happen).

The ~searched for string, may contain Ald-cards, signalled by nulls - C H R \$ ( 0 and the comparison is made with reference to either the RAMrecord length, or the length of the,searched for string,, whichever is the shorter.

Examples:

```
1BANKFIND,@r%,"123 test",0
18ANKFIND,@r%,f$,100,200
Beware the mismatch ...
```

Obvious errors, such as the wrong number of parameters, are reported as a 'B a d C o i n m a n d' error. However, the mechanism of external commands does not detect errors of the 'T y p e m i s m a t c h' variety, and the user must ensure that the correct form of parameters are employed.

The program below uses the RAMdise commands to set up and interrogate a database containing anagrams of 7-letter words. It searches for matches, and you may use wild cards.

For example, anagrams of the word F I G U R E S which match with ? R U G S ? ? (the two last ?? can be left off if you wish) are FRUGSIE, FRUGSEI, IRUGSFE, I R U G S E F, E R U G S F 1, and E R U G S I F.

The database takes some time to create, but then 64K is a lot of memory to fill!

```
10 'ANAGRAMS by ROLAND PERRY
20 ' copyright (c) AMSOFT 1985
30 1
40 'Remember to RUN "BANKMAW' before running program
50
60
70 MODE 2
80 DEFINIT a-z
90 r%=0:1BANKOPEN,7
100 INPUT"What 7 letter word to scramble ";s$
110 IF LEN(M<>7 THEN 100
120 PRINT"Please wait..."
130 LOCATE 1,5:PRINT"Computing:"
140 FOR c1=1 TO 7
150 FOR c2=1 TO 7
```

---

```

160         IF c2=c1 THEN 370
170         FOR c3=1 TO 7
180         IF c3=c2 OR c3=c1 THEN 360
190         FOR c4=1 TO 7
200         IF c4=c3 OR c4=c2 OR c4=c1 THEN 350
210         FOR c5=1 TO 7
220         IF c5=c4 OR c5=c3 OR c5=c2 OR c5=c1 THEN 340
230         FOR c6=1 TO 7
240         IF c6=c5 OR c6=c4 OR c6=c3 OR c6=c2 OR c6=c1
            THEN 330
250JOR c7=1 TO 7
260         IF c7=c6 OR c7=c5 OR c7=c4 OR c7=c3 OR c7=c2
            OR c7=c1 THEN 320

```

Continued on the next page

```

270         o$=MID$(s$,c1,1)+MID$(s$,c2,1)+MID$(s$,c3,1)
+MID$(s$,c4,1)+MID$(s$,c501)+MID$(s$,c6,1)
+MIMs$,c7,1)

```

```

280 LOCATE 12.5:PRINT x;o$ 290 1BANKWRITE,@r%,o$ 300 IF r%~:0 THEN
STOP 310 x=x+1

```

```

320 NEXT c7 330 NEXT c6 340 NEXT c5 350 NEXT c4 360 NEXT c3 370 NEXT
c2 380 NEXT c1 390 lastrec=r% 400 REM now look them up 410
r%=0:9$=SPACE$(7)

```

```

420     PRINT:INPUT"What match do you require: use ? as wild card:
",m$

```

```

430 m$=LEFT$(m$,7)
440 FOR x=1 TO LEMm$)

```

```

450 IF MIMm$,x,1)="V'THEN MIMm$,x,1)=CHM0) 460 NEXT

```

```

470 1BANKFIND,@r%,m$,0,Lastrec 480 IF r%<0 THEN GOTO 420

```

```

490 1BAUREAD,@r%,g$
500 PRINT g$,

```

```

510 1BANKFIND,@r%,m$,r%+1,Lastrec 520 GOTO 480

```

Appendix I  
AMSTRAD

Software Licence Agreement

Amstrad Software licence agreement for all software described in this manual.

VERY IMPORTANT

Licence relating to the Software described in this manual.

Software licence agreement

Amstrad provides the Software and the Program described in this manual incorporated in or embedded on magnetic disc, tape, hard disc or other media and the descriptive material related to that Program in this manual or in other documents strictly subject to the term and conditions of this Licence Agreement.

By breaking the Seal on the package in which the Software is provided and/or by entering, utilising, running, listing, copying or otherwise manipulating the magnetic disc, tape, hard disc, other media, Program, manual or other documentation other than for the purpose of erasure or destruction as set out below you agree to be bound by all the terms and conditions of this Licence Agreement.

If you do not agree to be bound by any one or more of the terms and conditions of this Licence Agreement you must destroy the Software Package, manual and documents and erase the Program and Software from the disc, tape, hard disc or other media on or in which it is incorporated or embedded.

Definitions

1. "Machine" means the single microcomputer, or PC on which you use the Software. Multiple CPU systems require additional licences.
2. "Software" means the set of programs, discs, tapes, documentation and related materials described in and supplied with this manual.
3. "Program" is a part of the Software and means the instructions, codes, messages or other information contained in, or embedded on magnetic disc, tape, hard disc or any other media, or any part of it.
4. "Amstrad" means AMSTRAD PLC, 169 Kings Road, Brentwood, Essex, CM14 4EF.
5. "Licensee" means the purchaser if purchased for own use or, if purchased for company use, the purchaser, that company and its employees.

Licence

Amstrad grants a non-exclusive Software Licence to Licensee to:

---

1. Use the supplied Program on a single Machine.
2. Where copy protection is not incorporated, copy the Program into any machine-readable or printed form for backup or modification purposes in support of Licensee's use of the Program on a single Machine. Licensee may make only one (1) copy of the Program for such purposes. Copying of documentation and other printed materials is prohibited. Disassembly, reverse compilation and reverse engineering of the Program or Software is prohibited.

#### Terms

1. Software supplied by Amstrad is copyright. Licensee agrees to uphold these copyrights.
  2. This Licence Agreement enters into effect at the time you open or break the seal on the Software package, or enter, use, run, list, copy or otherwise manipulate the Program, and is effective until terminated.
  3. Licensee may terminate this Licence Agreement at any time by destroying the Software together with all copies. The benefit derived by Licensee from the Licence will terminate automatically if the terms of this Agreement are violated and Amstrad may demand the return of the Software immediately. Any such termination shall be without prejudice to any accrued rights of the parties.
  4. This Licence is personal to Licensee and may not be assigned to any other person, persons or company.
  5. The right to lend, rent, sell, or otherwise transfer the Software in part or in whole is not granted to Licensee.
  6. Licensee may not use, copy, modify, or transfer the Software or documents, or any copy, modification or merged portion, in whole or in part, except as expressly provided for in this Licence Agreement.
  7. The Program cannot be transferred via any media other than that on which it is supplied. It cannot therefore be transferred via such media as telecommunication lines.
  8. Licensee agrees to take all possible steps to protect the Software from unauthorised use, reproduction or distribution.
- LimitedWan-anty

THE SOFTWARE IS PROVIDED AND LICENSED ON AN "AS IS" BASIS. LICENSEE ASSUMES RESPONSIBILITY FOR SELECTION OF A PROGRAM TO ACHIEVE LICENSEE'S INTENDED RESULTS AND LICENSEE IS RESPONSIBLE FOR INSTALLATION USE AND RESULTS OF USE. EXCEPT AS IS HEREIN EXPRESSLY STATED, AMSTRAD MAKES NO REPRESENTATION, GIVES NO WARRANTY AND ACCEPTS NO CONDITION OF ANY KIND WHATSOEVER EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES AND

---



CONDITIONS OF MERCHANTABILITY AND FITNESS FOR PURPOSE (SPECIFIC OR GENERAL).

AMSTRAD DOES NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET LICENSEE'S REQUIREMENTS OR THAT THE OPERATION OF THE PROGRAM WILL BE UNINTERRUPTED OR ERROR FREE.

AMSTRAD DOES NOT RECOMMEND USE OF THE SOFTWARE FOR LIFE CRITICAL FUNCTIONS OR FOR ANY APPLICATIONS WHERE A SOFTWARE ERROR MAY CAUSE FINANCIAL LOSS, DAMAGE OR EXPENSE.

However, Amstrad gives a limited warranty that the discs, tapes or other non-hardware media on which the Program is embedded and supplied by Amstrad, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of receipt by, or delivery to Licensee as evidenced by a dated copy of Licensee's proof of purchase receipt. In the event that the discs, tapes or other non-hardware media are found to be so defective within the said period, Amstrad will replace them free of charge, if they are returned postage prepaid.

Amstrad will also, AT ITS DISCRETION, by modification or alteration endeavour to remedy any Program errors, attributable to Amstrad, which are provable and demonstrable, and reported to Amstrad in writing within the aforesaid period.

#### Limitations

UNDER NO CIRCUMSTANCES WILL AMSTRAD, THE AUTHOR, OR THE MANUFACTURER, DEVELOPER OR SUPPLIER OF THE LICENSED SOFTWARE BE LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR EXEMPLARY DAMAGES OR OTHER ECONOMIC OR PHYSICAL LOSS, EVEN IF AMSTRAD OR SUCH OTHER PARTY(IES) HAS/HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND LOSS. UNDER NO CIRCUMSTANCES WILL LIABILITY EXCEED THE PURCHASE PRICE OF THE SOFTWARE.

#### Indemnity

The Licensee shall indemnify Amstrad in respect of any liability incurred by Amstrad or the Software author as a result of a claim brought by Licensee against Amstrad directly or against a third party, for negligence, breach of contract or product liability.

#### Governing Law

This agreement shall be construed, interpreted and governed in accordance with the Laws of England (or in the case of Microsoft products, Washington, USA).

U.S. Government restricted rights in respect of Microsoft programs

The Program and documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the US Government is subject to restrictions as set forth in subdivision b(3)(ii) of The Rights in

---

Technical Data and Computer Software clause of 252.227-7013 of the DOD FAR.

Contractor/manufacture is Microsoft Corporation/160 11 NE 36th Way/Box 9701/Redmond, WA 98073-9717.

#### General

If any of the above provisions thereof are invalid or unlawful under any applicable Law they are to that extent only deemed omitted.

SOME STATES DO NOT ALLOW EXCLUSION OF IMPLIED WARRANTIES AND CONDITIONS OR LIMITATIONS OF LIABILITY. IN SUCH CIRCUMSTANCES THE ABOVE EXCLUSIONS AND LIMITATIONS MAY NOT APPLY TO LICENSEE. THIS WARRANTY GIVES THE LICENSEE CERTAIN LEGAL RIGHTS BUT LICENSEE MAY ALSO HAVE OTHER LEGAL RIGHTS WHICH VARY FROM STATE TO STATE.

#### F. Export of US Technology

No US technology, Program or Software may be exported or sold or transferred directly or indirectly to any person in the knowledge that export may occur without application for and obtaining necessary US Government Export or Re-Export licence issued by the US Department of Commerce, Washington DC 20230.

AMSTRAD PLC  
169 Kings Road  
Brentwood  
Essex, CM14 4EF

© 1990 Amstrad plc. All rights reserved.

---