

1. ABOUT THIS MANUAL

This manual gives all the necessary information you will need to install and run the CP/MULATOR. It also describes the differences between the emulator and the original operating system.

Section 1: The CP/MULATOR User Manual  
However, it is not a full replacement for the original CP/M. It is a simulator which runs on a variety of hardware and software environments. This manual describes how to use the emulator.

Section 2: Getting Started  
This section describes how to install and run the emulator.

Section 3: The Command Interpreter  
Section 4: Compatibility with CP/M

Copyright (c) 1987 Peter Szymanski

All rights reserved. No part of this manual or the CP/MULATOR software may be reproduced in any form or by any means or used to make a derivative work including translation, transformation or adaptation without the express permission in writing of the worldwide licensees:

Sandy (UK) Personal Computer Products Ltd.  
Unit 33, Murdock Road, Manton Lane  
Bedford MK41 7PQ. England

Although great care has been taken in the preparation of the CP/MULATOR program and this manual, neither the author nor distributor accept any liability for loss or damage whether direct, indirect, incidental or consequential including but not limited to loss of stored data, profit or contracts which may arise from any error, defect, omission or failure of any part of this program.

Trademarks

CP/M is a registered trademark of Digital Research Inc.  
Z-80 is a registered trademark of Zilog Inc.  
QL & QDOS are registered trademarks of Sinclair Research Ltd.

### 3. System Requirements

To run CP/Mulator you need at least one floppy disk and at least 80 KBytes of free memory. The package will thus run without memory expansion, provided you do not have anything else in memory at the time.

The program is always run in 'monitor mode' with a screen of 80 characters and 24 lines. A TV is of little use, you should have a monitor or RGB input TV/Monitor for CP/M.

An optional printer is supported as well as devices such as a modem connected to the second serial RS232C port.

Up to 16 directory devices such as floppy disk drives, winchesters or RAM disks may be used. There is no limit on the maximum capacity of these devices. All input/output is carried out via QDOS, it is therefore not necessary to partition directory devices for the emulator.

### 4. Getting Started

Make a backup copy of your distribution disk first. To do this LRUN the program CLONE\_BAS. This automatically copies all the required files to the destination device of your choice. If you have only one disk drive, use a RAM disk or microdrive. Safely store the distribution disk, use only the backup copy to run the CP/Mulator.

As supplied, CP/Mulator is configured to operate with two floppy disk drives (FLP1\_ & FLP2\_), a printer connected to SER1, and SER2 as an auxiliary channel (CP/M devices RDR: & PUN:). To change these default settings use INSTALL\_BAS. (See section 'System Installation')

The package includes a ROM cartridge. This must be carefully inserted in the ROM slot at the back of your QL. It can only be inserted one way up and is located by a slot in the circuit board tongue. \*\*\* DISCONNECT THE POWER FROM THE QL BEFORE INSERTING THE ROM CARTRIDGE \*\*\*. Inserting or removing a ROM cartridge while the QL is powered up may cause severe damage both to the cartridge and the QL.

CP/Mulator will not run without the ROM cartridge in place, there are two reasons for using this system :

Firstly: the QL is very slow when accessing its internal RAM. Using the ROM improves performance by approximately 30%.

Secondly: it enables CP/Mulator to run without memory expansion.

Do not attempt to run CP/Mulator without the ROM installed as the QL may function unpredictably.

To run CP/Mulator type the command line :

```
EXEC_W FLP1_CPMU_EXE <ENTER>
```

The program will be loaded into memory and signs on with a copyright message, serial number and version numbers.

As soon as the system prompt 'A' appears the command interpreter is waiting for your commands.

CP/Mulator may also be run concurrently with other jobs using the command line :

```
EXEC FLP1_CPMU_EXE <ENTER>
```

although this is not recommended. CP/Mulator is a very "well-behaved" program - it does not use hidden features of your QL but in order to gain the maximum speed the section of code which interprets the Z-80 code runs in supervisor mode with all interrupts disabled. This may cause the response time of other jobs running concurrently to be extended.

This manual is being written through the CP/Mulator running under QRAM multi-tasking manager from Tony Tebby on a Sandy SuperMouse/SuperQBoard 640K system and actually has two copies of the CP/Mulator running concurrently with different programs in each - multi-tasking CP/M!

Please note that as CP/M application programs use CTRL/C the 'Change Job Queue' character is CTRL/SHIFT/= while the emulator is running.

## 5. File Naming Conventions

In CP/M a file name consists of the drive specifier, a colon, up to eight characters for the name, a dot and up to three characters extension.

To make the files accessible CP/Mulator generates filenames for QDOS by preceding them with the character (c) - SHIFT/ESC. The dot is replaced by an underscore.

ie. The CP/Mulator file 'A:MYFILE.COM' will appear in the directory of FLP1\_ as '(c)MYFILE\_COM' if FLP1\_ is installed as logical drive 'A:'

You may access or generate CP/Mulator files with a QDOS program if you follow these conventions :

Lines in a text file are separated by a CR/LF sequence (0DH,0AH) unlike QDOS which uses only LF.

The end of a text file is marked by one or more EOF characters (1AH = CTRL/Z).

The length of the file should always be a multiple of 128 bytes (one logical sector in CP/M)

## 6. The Command Interpreter

As soon as the system has signed on, the command interpreter is invoked. It displays the system prompt in the form :

X> Where X is the currently selected drive (A ... P)

You can then type in a command line. As the QDOS line editor is used the cursor keys function as normal for editing. When <ENTER> is pressed the command line is analysed and executed.

The command interpreter fully supports the usual wildcard characters '\*' and '?' in filenames.

If no disk drive is specified, all commands work on the currently selected logical drive. To change the current drive type :

X: <ENTER>

Where X is one of the 16 logical devices (A: ... P)

Example: B: <ENTER>  
Selects drive B as the default disk drive.

Resident commands :

DIR

Displays a directory of all or selected CP/M files on a drive in a CP/M format.

**Example:**

DIR display all CP/M files on current drive  
DIR B: display all CP/M files on drive B:  
DIR D:\*.COM display files on drive D with extension COM

**DIRS**

Display a directory of all or selected CP/M files on a drive together with their file length and update date.

**Example:**

DIRS display all files on current drive  
DIRS C:T\*.\* display all files on drive C: which start with the character 'T'

**REN**

Renames a single file

**Example:**

REN A:NEWNAME.DTA=A:OLDNAME.DTA  
assigns the new name 'NEWNAME.DTA' to the file 'OLDNAME.DTA' on drive A:  
(if you think this is the wrong way round - well you are right, but thats CP/M)

**ERA**

Erase the specified file(s)

**Example:**

ERA A:\*.BAS Erase all files with extension '.BAS' on drive A:

ERA MYFILE Erase the file 'MYFILE' on the current drive.

Use this command with great care as there is usually no way to recover a deleted file unless you have a backup - another good reason for making regular backups!

**ERQ**

Erase the specified file(s) but ask for confirmation

**Example:**

ERQ B:\*. \* Erase all files on drive B:

The system will display the name of the file to be erased. If you type 'Y' the file will be erased.

**TYPE**

Display the contents of a text file

**Example:**

**TYPE MYTEXT.DTA** Display file 'MYTEXT.DTA' of the current drive.

You may pause the command at any time by pressing any key. CTRL/C aborts the command.

**SAVE**

Save a given number of pages of memory in the specified file

**Example:**

**SAVE 12 MYFILE.DTA** Save 12 pages of memory of 256 bytes each starting at address 0100H of the Z-80 program section in the file 'MYFILE.DTA'

If the file already exists, it will be overwritten.

**CLS**

Clear the screen

**TIME**

Display the day of the week, date and time until a key is pressed.

**QDOS**

Stop running CP/Mulator and return to QDOS

CP/Mulator does not allocate any memory other than 64 Kbytes in the Transient Program Area for the Z-80 program section. QDOS will be able to fully reclaim the memory without problems.

On exit, the 'Change Job Queue' character is changed back to the normal CTRL/C.

### Transient Commands

If the first word of your command line is not any of the above commands or is preceded by a drive specifier, CP/Mulator loads the file with the given name and the extension '.COM' into the Z-80 memory section at location 100H, and transfers control to it. As in conventional CP/M additional parameters may be passed to the program. If the file cannot be found, an error message will be given.

#### Examples:

##### PROGNAME

Start the program contained in the file 'PROGNAME.COM' from the current drive.

##### C:PROGNAME

Start the program from drive C:

##### PASCAL MYFILE \$Z

Compile the file 'MYFILE.PAS' with Pascal and pass the option '\$Z' to the compiler.

##### MENU

Start the utility program supplied with your master disk

## 7. Compatibility with CP/M

One of the most often asked questions on CP/Mulator is 'Does it run the program XYZ?' and the answer is usually 'Yes!'

Of course this does not mean it runs any Z-80 program - you cannot run your old ZX Spectrum software with it; and it does not run programs that use hardware features of a specific system, ie. if a CP/M program tries to read a real-time clock which is connected to some I/O line in a Z-80 system, it will not work.

As outlined later on, there are also some problems in the Z-80 emulator due to hardware specific features of the Z-80 processor (interrupts, I/O).

So far we have found that any 'Well behaved' CP/M program will run. It runs in either 8080 code or Z-80 code. As your QL system does not have a real Z-80 processor, everything has to be done in software. Every single byte of the CP/M program has to be interpreted; and the function calls to the CP/M operating system (disk I/O, character I/O etc) have to be translated into something QDOS can understand.

Programs thus run slower, but it is impossible to predict how much slower. Any programs which makes excessive use of disk I/O may even run faster on your system as QDOS moves data to and from disks quite briskly compared to most CP/M machines.

We have carried out extensive testing. Most of the programs worked perfectly and detail testing of the following very popular titles showed no problems.

TURBO PASCAL (Borland) - works exceptionally well and is highly recommended. It is even faster than a popular interpreting Pascal written directly for QDOS.

PASCAL MT+ (Digital Research) - no known problems. The choice for the professional programmer.

WORDSTAR (MicroPro) - no known problems other than speed. QL users, you can add another slow word processor to your software library, but this one is at least reliable!

BASIC-80 INTERPRETER & COMPILER (Microsoft) - The interpreter is a bit slow but works fine.

MACRO-80, LINK-80, LIB-80 (Microsoft) - This is your assembler package if you want to develop Z-80 or 8080 assembly language programs. No known problems and highly recommended.

DTZ (CP/M Users Group) - works as well as ZSID (Digital Research) or GEMDEBUG (Gemini Computer Systems). One of these debuggers could complement the above assembler package.

Note that many CP/M programs need to be installed to your video terminal. You don't have one - that's right, but you do have a terminal emulator, see 'The Screen Driver' section.

Some programs such as disk utilities access the disk through the BIOS functions READ and WRITE. These programs do not work with CP/Mulator.

CP/M provides two data structures called the 'disk parameter block' and the 'allocation table'. A program may use the tables to calculate the capacity of a disk drive. CP/Mulator supports the appropriate BDOS functions but always supplies tables for a standard 8" disk drive with no used sectors. If a program tries to calculate the storage capacity of a disk (like Turbo Pascal does in its directory display) it will always find 243 KBytes free. So don't be confused by this phenomenon. On the other hand the BDOS function 'Compute File Size' works well.

As the directory structure of CP/M is different from that of QDOS, the codes in the file control block which are reserved for internal system use are different. Application programs thus fail if they attempt to refer to these bytes. Of course, such a program could hardly be described as 'Well behaved'. We only know of one such program 'STAT' that comes with the original operating system.

Listed below is a complete list of differences in the emulator BDOS and BIOS compared to the CP/M operating system:

**Functions 7/8 Get/Set I/O byte**  
The I/O byte is not supported

**Function 10 Read Console Buffer**  
This function uses the QDOS line editor, you may edit the line using the cursor keys. As you cannot type CTRL/C you can insert the '(c)' character (SHIFT/ESC) This will abort the CP/M program and return you to the Command Interpreter.

**Function 27 Get ADDR of Allocation Table**  
See above.

**Function 28 Write Protect Disk**  
The function is not supported.

**Function 29 Get Read Only Vector**  
All drives are in a Read/Write state.

**Function 30 Set File Attributes**  
The function is not supported.

**Function 31 Get ADDR of Disk Parameters**  
See Above.

**Function 32 Set/Get User Code**  
User areas are not supported.

A CP/M program may still use the above functions without crashing as CP/Mulator always returns values that make sense.

**BIOS functions READ and WRITE**  
The functions are not supported.

## 8. Changing Disk

Unlike QDOS, CP/M has to be told when a disk has been changed in an online disk drive - that's the famous CTRL/C. QDOS is a bit more strict on that subject - you may not change a disk as long as any file is left open. This usually presents no problem - but it does in CP/Mulator. If a file is read in CP/M it is not necessary to explicitly close it. Application programs thus never close files that were read into memory.

We have found a very tricky way round this that allows you to change a disk if you follow these rules carefully :

a) You may change the disk whenever the command interpreter displays the system prompt.

b) You may change the disk if all write operations are complete.

c) If read - files are left open, you may change the disk, but an error message like 'files still open' may appear somewhere on the screen and the next access to the directory may give wrong results. What really happens depends on your floppy disk driver software. Some controllers return the proper error codes to the calling program, some do not.

You can easily find out what happens if you proceed as follows :

In Superbasic type

```
OPEN_IN #3,FLP1_TESTFILE
```

where TESTFILE is the name of a file that exists on the disk in FLP1\_

Remove the disk in FLP1\_ and insert another one, then

Type

```
DIR FLP1_
```

If you now get the error message 'Bad or changed medium' there will be no problem for CP/Mulator, but if you simply get 'not found' you should take care when changing disks in the CP/Mulator.

QDOS operating system writers please note: 'not found' is the wrong error message in the above situation.

## 9. The Z-80 Emulator

Roughly 70% of CP/Mulator consists of the Z-80 emulator. Its purpose is to interpret the Z-80 or 8080 code as fast as possible.

As the Z-80 structure differs greatly from that of the 68000 processor, there are numerous instructions that cannot be successfully interpreted. Fortunately, CP/M programs do not generally use these instructions because CP/M is based on the 8080 processor and the instructions deal with special hardware features of a system.

If you write your own Z-80 programs with CP/Mulator, you should note the following points :

All I/O instructions are treated as NOP.

All instructions that deal with interrupts are treated as NOP

The Z-80 instruction HALT (076H) is used to switch to the emulators BDOS. Do not use the HALT instruction in your program as this may lock the system.

## 10. The Screen Driver

Many CP/M programs expect to work with a terminal that caters for special features like cursor addressing, attributes etc. CP/Mulator thus contains a screen driver which allows screen handling in a terminal-like manner.

All the features are invoked by sending one or more control characters to the screen.

- 07H CTRL/G BELL  
Emit a short beep.
- 08H CTRL/H BACKSPACE  
Move the cursor to the left by one character.
- 09H CTRL/I TAB  
Move the cursor to the next tab stop. Tab stops are at multiples of 8 character positions.
- 0AH CTRL/J LF  
Move the cursor down one line. If at last line, scroll up.

- 0DH CTRL/M CR  
Move the cursor to the first position in the current line.
- 1AH CTRL/Z CLS  
Clear screen and move cursor to the top left.
- 1BH 3DH y+20H x+20H CURSOR ADDRESSING  
This sequence of four characters is used to move the cursor to a specific position on the screen where y is the line and x is the column.

The above codes are compatible with the Lear Siegler ADM 3A terminal. This terminal type can be found in most installation programs for CP/M packages. To install Wordstar for example simply select the ADM 3A terminal from the installation menu.

To gain maximum performance, you may also use the following codes :

- 10H NORMVIDEO  
Display white characters from now on.
- 11H LOWVIDEO  
Display green characters from now on.
- 12H UNDERON  
Start underline.
- 13H UNDEROFF  
Stop underline.
- 1CH ERAEOL  
Erase to end of line.
- 1DH DELLN  
Delete line and scroll up.
- 1EH INSLN  
Insert line.

Please note :

CP/Mulator always uses a screen of 80 characters by 24 lines. The 'paper' is set to black and the 'ink' to green. The cursor is only enabled when the system is waiting for a keypress.

To use the above functions in your programs, the codes for the function must be sent to the CP/M device 'CON:'.

**EXAMPLES**

To clear the screen from a BASIC program, you would use the line: PRINT CHR\$(26);

To move the cursor to the top left of the screen you would write in Pascal: WRITE(CHR(27),'=',CHR(32),CHR(32));

or in a more general form:

```
PROCEDURE GOTOXY(X,Y : INTEGER);
  {move the cursor to column x [0 .. 79]
   line y [0 .. 23]}
BEGIN
  WRITE(CHR(27),'=',CHR(Y+32),CHR(X+32));
END;
```

**11. The Keyboard Driver**

To make life a bit easier for the user, CP/Mulator translates a number of keyboard codes into often used codes for CP/M programs.

ENTER is translated into CR (13H)

CURSOR LEFT is translated into BACKSPACE (08H)

CURSOR RIGHT is translated into CTRL/D (04H)

CURSOR DOWN is translated into CTRL/X (05H)

CURSOR UP is translated into CTRL/E (05H)

SHIFT/CURSOR LEFT is translated into CTRL/A (01H)

SHIFT/CURSOR RIGHT is translated into CTRL/F (06H)

SHIFT/CURSOR DOWN is translated into CTRL/C (03H)

SHIFT/CURSOR UP is translated into CTRL/R

CTRL/CURSOR LEFT is translated into DELETE (7FH)

CTRL/CURSOR RIGHT is translated into CTRL/G (07H)

F1 is translated into CTRL/J (0AH)

The translations for the cursor keys enable their use by on-screen editors like Wordstar and Turbo Pascal.

## 12. A Sample BOOT Program

Once you have installed CP/Mulator, you may start it at any time by the simple command line :

```
EXEC_W FLP1_CPMU_EXE
```

but you may prefer to have an autostart BOOT program for it that can also perform other functions. A simple example is listed below and it can easily be tailored to your exact requirements :

```
10 REMark ** Sample boot loader for CP/Mulator
20 REMark ** Set baud rate
30   BAUD 2400
40 REMark ** Install a RAM disk if you have one and
50 REMark ** sufficient memory capacity
60   a=RESPR(2048):LBYTES FLP1_RAMDISK_CDE,a:CALL a
70   FORMAT RAM1_600
80 REMark ** Now start the CP/Mulator
90   EXEC_W FLP1_CPMU_EXE
```

## 13. Installing CP/Mulator

The program `INSTALL_BAS` from your distribution disk may be used to install CP/Mulator to your system configuration.

It will ask you a number of questions which you simply have to answer :

Please do not run `INSTALL_BAS` on your master disk but only on your working backup copy.

It allows you to :

- assign a QDOS directory device to any of the 16 logical CP/Mulator disk drives.
- assign a QDOS character device to the CP/Mulator printer LST:
- assign a QDOS character device to the CP/Mulator auxiliary devices RDR: and PUN:

As supplied, CP/Mulator is set to the following defaults :

```
Drive A:      ---  FLP1_
Drive B:      ---  FLP2_
Drives C: to P: ---  FLP1_
```

The length of the directory device name is limited to four characters.

LST: --- SER1HR  
RDR: --- Input from SER2IR  
PUN: --- Output to SER2IR

The length of the device names is limited to 22 characters. Note that baud rates have to be set in the BOOT program.

#### 14. Special Notes

You can install any directory device of your QL to be a logical CP/Mulator drive. This includes microdrives. The problem here is that the QDOS trap for renaming files on microdrives does not work.

This can be overcome if the QL has expanded RAM and the RAM version of SuperToolkit II or any other program which enables Trap #3 with D0=4A to rename a microdrive file. CP/Mulator can be installed to run only on microdrives - an unusual environment for CP/M and rather slow - but it works!

A RAM disk and expanded memory is strongly recommended if you intend to run CP/M programs which make a lot of use of disk I/O.

You have probably already discovered that there are some problems with the QL serial ports when using both ports at the same time. This should be avoided.

Printer output from CP/Mulator can be sent to any file but the system will crash if the file cannot be opened. Such files remain open until CP/Mulator is terminated.

QL screen graphics cannot be used with CP/Mulator. We have seriously considered it but there is no simple solution. Most CP/M machines don't have graphics and the GSX (Graphics System Extension) would be a bit too much for your QL to support.

Please help to make this program even better. A program as complex as this one (more than 240 KBytes of assembler source code) must have bugs somewhere. If you find a bug, please let us know. You will find a form for doing this in the manual and we will make an update available at a nominal charge after each major revision, we would like to incorporate any extra features which can be added. So, if you have any good ideas, let us know.

If you experience any difficulty in running the program our service department will try to help by answering written questions, but make sure that you have thoroughly read and understood the manual, most of the answers are already there. The manual is not a primer on the CP/M operating environment. We can recommend:

'Using CP/M' by Peter Gosling published by Macmillan  
'CP/M Programmers Encyclopaedia' by Bruce Brigham  
published by Que Corporation.

### 15. The Utility Program MENU.COM

Included with the CP/Mulator package is the program MENU, this provides housekeeping utilities such as file copying, file printing, directory control, renaming etc.

MENU is fully menu driven and you will be able to use it immediately even if you have never worked with CP/M before.

You can run the program after starting CP/Mulator, when the command interpreter displays the system prompt just type :

MENU <ENTER>

provided you have a disk with MENU.COM in the currently selected drive.

MENU was written in Turbo Pascal and is a good example to demonstrate the excellent performance of this compiler when used with CP/Mulator.

Have fun with CP/Mulator!

Index of Contents :

1. **Getting started**
2. **A word about Disk Formats**
3. **Using the predefined Formats**
4. **Creating your own Formats**
5. **Notes to Users with a single floppy drive**
6. **Note to QL Users**
7. **A Word of Warning**

\*\*\*\*\*

## 1. Getting Started

RDCPM is a program to allow reading of CP/M disks in many different formats by a Sinclair QL computer system.

The program is supplied in the single file 'RDCPM\_EXE' on your distribution disk. To run it you need at least one disk drive (double sided, double density, 80 tracks per side, in 3.5" or 5.25" size). QL users see section 6. If the program is run with a single floppy disk drive, a further directory device such as a microdrive or RAM disk will be required. The program occupies less than 60 KBytes of memory.

To run the program insert the disk containing 'RDCPM\_EXE' into FLP1\_ and type the command line :

```
EXEC_W FLP1_RDCPM_EXE
or EW RDCPM_EXE for SuperToolkit II users
```

Note that although the program can be run with other jobs in memory, it should not be run together with jobs that use files on a floppy disk.

On startup the program will ask you which disk drive you want to use for the CP/M disk. Answer '1' if you want to use FLP1\_; '2' for FLP2\_ etc. The selected drive remains the same throughout the session and cannot be used for anything else.

## 2. A Word about Disk Formats

Although there are a number of widely accepted standards for floppy disks in the computer industry (connectors, disk sizes, density, low-level format etc.), most operating systems have their own unique format. Format refers to the conventions used for exactly how data is stored on the medium and where the index to that data is located. Not surprisingly the QL has a unique format which is incompatible with any other system. This means that it is not possible to directly read a CP/M disk on the QL. Fortunately, the low-level format of most CP/M systems is the same as that used by QDOS, so it is possible to read individual sectors (small blocks of data) from the disk.

In the early days of micro computing, most small computers only had disk drives with a diameter of 8 inches. When CP/M appeared and gained wide acceptance with its version 1.4, it created a single standard for 8 inch disks. (Sorry, but most floppy disk interfaces cannot attach 8" drives to the QL). When the first CP/M computers with mini disk drives (that is 5.25") appeared on the market, most manufacturers tried to make the most of the new technology - to be positive, and created their own new format. The result was the strange situation where a user of this very portable operating system had to face the problem that disks from a different make of micro computer could not be read even though both were CP/M machines.

The exact CP/M format used by any particular machine is determined by a combination of many parameters covering such details as disk size, density, number of sides, tracks, sectors, sector length - to name just a few. To make the situation even more complicated, the organisation of the data varies with the total capacity of the disk. This resulted in several hundred different possible disk formats of which certain ones became more popular.

When we created our CP/Mulator for QDOS systems, there was only one way to get CP/M programs into the system and that was via the serial ports from a different machine, which is very slow when several megabytes of data must be transferred at an effective rate of only 30 bytes per second! To solve the problem we have developed a very ingenious program (Yes, we are that proud of it) - RDCPM. It enables reliable transfer of CP/M data on a file by file basis with very little effort. A library of the more popular formats is built in but you can create as many of your own as you want. During development we even managed to copy files from a disk which was in a format we did not know the name of.

RDCPM provides as much help as possible. It is able to analyse the physical (low-level) format and calculates some of the parameters from this, and it will tell you if it cannot handle the disk at all.

There are a number of formats that RDCPM cannot handle. In addition to systems which have tricky sections in the logical organisation of formats there are also hardware differences in some machines. For example, variable speed drives used to be quite common - these vary the speed of rotation with the track number being accessed. Others use the CPU to transfer data to the disk but your QL system has a dedicated disk manager chip. Hard sectored disks will not work as they are specific to one mechanical drive type; and of course you must have the correct diameter drive in the first place.

### 3. Using the Predefined Formats

To make access to CP/M software as simple as possible, we have included four different widely used formats with RDCPM.

**ATARI ST CP/M** - This is the format used with the CP/M emulation for Atari 68000 systems. Many software houses are offering disks in this format, and it is the most popular format for 3.5" disks.

**GEMQDDS** - The British company Gemini Computer Systems Ltd., manufactures a computer system which is able to handle several hundreds of formats for CP/M and other operating systems. Their major format is GEMQDDS (for 5.25") and is identical to GEM3.5 for 3.5" disks. You can use GEMQDDS for both sizes. As many software houses in Europe use the Gemini system it should not present any problem to purchase software to run your CP/Mulator. Sandy can also offer a conversion service to move your software from various formats to CP/Mulator pattern on disk or microdrive.

**JKCPM+** - A format for a CP/M plus system which is popular for industrial applications in continental Europe. It can be used for both 5.25" and 3.5" disks.

**SUPERBRAIN DD35** - The single side, double density 35 track format for the Intertec Superbrain, a machine which was quite popular a few years ago and for which much software of interest to home computer enthusiasts is available.

If you have a disk in one of the above formats and want to copy it to a QDOS disk, proceed as follows :

1. Start RDCPM
2. Select the disk drive for the CP/M disk
3. Use option 1 from the main menu to select the required format. The current format will always be shown in the top of the menu window.
4. Insert the CP/M disk into the selected drive when prompted to do so.
5. Use option 2 from the main menu to display the directory of the CP/M disk. The names of all the files on the disk will be shown together with a file number and an extension number. Write down the numbers of all the files you want to copy. Some filenames may appear several times on the list, always use the lowest number otherwise the copy option will only copy part of the file.

6. You may use option 3 to display text files on screen. It only works for files that contain plain ASCII text.
7. Use option 4 to copy file by file to a QDOS device. The program asks for the file number to copy and then for the QDOS filename. Note that you have to type the full QDOS filename here:

FLP2\_MYFILE\_TXT

8. If error messages appear at any stage, it is likely that you are using the wrong format or that the disk itself is corrupt. Other error messages such as 'disk full' or 'device not found' indicate a problem with the destination file.

You may change the disk in the CP/M drive whenever the main menu is displayed. Note that the destination file will be overwritten if it already exists.

#### 4. Creating your own Formats

RDCPM allows you to create as many different formats as you like. With the large number of parameters to adjust this can be quite a time consuming process even though the program does most of the work automatically.

After selecting option 6 from the main menu, you are prompted to insert the CP/M disk into the selected drive. The program will then analyse the physical format of the disk and display the parameters which it finds. If for some reason the disk cannot be handled the procedure will be aborted with an error message.

The program will then ask a couple of questions on the logical organisation of the disk.

[a] Sector Skew : Imagine a program wants to read data from a number of sectors on a disk. While the contents of one sector is moved into memory, the disk continues to rotate and if the computer requests the next sector along it will already have passed under the drive head. This means that the system has to wait until the requested sector passes under the head again - a very slow series of sector accesses would result. To avoid the problem data is not stored in consecutive sectors but skips over one or more sectors before continuing the file. Later on, other files will fill in the empty sectors left by this process. The number of sectors skipped between two logically continuous sectors is called the skew factor. Some systems use a different method, this

is a special case with a skew factor of 0. Start with a skew factor of 0, increase this later on if you discover that the program reads the data from the disk in a somewhat garbled fashion.

[b] Inverted Data : Depending on the hardware used, the data on the disk can be inverted - a '1' is recorded as a '0' and vice versa. Start with 'n' - not inverted.

[c] Number of reserved tracks : The CP/M operating system information is usually stored on the first few tracks of the disk. For double density there should be one or two reserved tracks; for single density three or four.

If you have guessed the answers for 2. and 3. correctly, the program is able to show you at least part of the directory of the disk.

[d] Block size : CP/M organises the disk in blocks with a length of 1,2,4,8 or 16 KBytes. For floppy disks normally only 1,2 or 4 are used. Disks with a capacity of less than 250 KBytes can have a block size of 1; drives with a higher capacity must have 2 or 4. Try 2 for the normal double density disks.

[e] Number of Directory Blocks : this tells the program how many directory entries on the disk there may be. Each entry occupies 32 bytes, usually there is only one directory block but you may try two if you find that part of the directory is missing.

[f] Side mode : This is only relevant for double sided disks. For some formats a logical track starts on side one and ends on side two - this is side mode 1. Some systems have all tracks on side one first and then on side two - this is side mode 0.

With some patience and experimentation you should be able to adjust these parameters. The program asks you if you want save the parameters in a file for later use. You can also give a name to the format definitions. The file created here can be recalled later on with option 7 from the main menu.

### 5. Notes for users with a single Floppy Disk Drive

If you only have one disk drive, you already know that life can sometimes get a bit difficult. If you want to use RDCPM proceed as follows.

Initialise a RAM disk. You should leave enough memory free for QDOS and RDCPM, which uses roughly 60 KBytes of RAM.

If you want to load any format definitions with option 7 of the main menu, copy these definition files to your RAM disk. Start the program as usual, when the program signs on remove the disk from the drive and use this drive for your CP/M disk. You can now use the program, copying your CP/M files to RAM disk but remember to copy them back to a QDOS disk before resetting the system.

### 6. Special Notes

If you only have one disk drive, microdrives can be used as another directory device instead of the RAM disk to store the files from the CP/M disk. Of course, the operation with a RAM disk will be much faster.

Some floppy disk controllers do not support the mystical '\*D2D' file, the method to access individual sectors of a disk.

That means it is not possible to run RDCPM with the disk controller from Micro Peripherals unless you have the QFLP eprom (highly recommended) from Qjump installed. If you are in any doubt on the '\*D2D' file consult your disk controller manual or the original supplier.

### 7. A Word of Warning

The aim of RDCPM is not to make illegal copies of programs. Please do not do so. For you the user and the computer industry it is essential that as many brilliant programmers as possible spend their days creating the wonderful programs we all love. Writing high quality software is a very hard job as you probably realise - we certainly know it! So don't copy software illegally. Fortunately there are a number of excellent user groups who have produced masses of 'Public domain' CP/M software. The libraries are huge and cover everything from compilers, toolkits, utilities to dedicated application programs.

CP/Mulator Software Problem Report

Please use this form to report any errors or problems in CP/Mulator. A concise description of the problem and its circumstances is required. If a patch or interim solution is being used please describe it.

This form should also be used to describe suggestions for enhancements to the program.

Name .....

Address .....

.....

.....

.....

Telephone ..... Date .....

CP/Mulator Version ..... Serial No. ....

Dealer .....

Problem description:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Return this form to:  
Sandy (UK) Personal Computer Products Ltd.  
Unit 33, Murdock Road, Manton Lane  
Bedford MK41 7PQ, England  
Tel: (0234) 219814