

PC-Four

A TRS-80 Model 4 Emulator Program
for
MS-DOS and PC-DOS Compatible Computers

Version 2

(C) Copyright 1988 Hypersoft

PO Box 51155, Raleigh, NC 27609

READ THIS FIRST

Notice: This program is capable of simulating many programs originally written for the TRS-80 Model 4. Some manufacturers of software attempt to restrict your rights as to copying, transferring to other machines or use on more than one machine. It is your responsibility to ensure that you have the legal right to use these programs, on a machine running PC-Four.

Please read, and make sure you thoroughly understand this manual. No responsibility can be accepted for damaged or lost data through misuse or mistakes on the part of the user. You should always retain backups of any material you consider valuable so that you will not lose in any way if something goes wrong.

Every effort has been made to ensure this program operates as described in this manual. If you find any problem, please report it directly to Hypersoft, and we will exert our best efforts to resolve it and return you a revised copy. We would appreciate any constructive comments that would help us improve the program in any way.

This program and manual is the original work of The Author and as such is protected by Federal and International Copyright Law. Please feel free to make copies of the program for archival and work purposes however you must treat this manual and program like a book - i.e. while you can copy it you can only use one copy at a time.

Trade Marks:

IBM PC and PC-DOS are registered trade marks of International Business Machines Inc.

TRS-80 is a registered trade mark of Tandy Corporation.

MS-DOS is a registered trade mark of Microsoft Inc.

Other Registered Names:

From time to time this manual mentions names of software packages such as TRSDOS, SCRIPSIT, etc. All these are registered names and or the copyright of their respective owners.

PC-Four

(C) Copyright 1987, Hypersoft
PO Box 51155, Raleigh, NC 27609

<i>Page</i>	<i>Contents</i>
1	1.0 Introduction
2	2.0 Getting Started
3	2.1 Making a back up of your distribution disk.
4	Contents of your PC-Four Disk
4	2.2 Making your working PC-Four System.
9	2.3 Transferring files to MS-DOS
10	Using Hypercross
11	Using PC Cross Zap (PCXZ)
12	Using a Modem
12	Using TRSCROSS
13	2.4 What you should know about MS-DOS
13	Drives, Directories and Sub-Directories
14	Paths, Current Drive and Directory
15	Changing Defaults
15	Making and Deleting Sub-Directories
15	Copying files
16	3.0 Running PC-Four
19	3.1 Configuration Options.
20	3.2 The Function Keys
21	3.3 Other Keyboard Keys
22	3.4 JCL Files.
23	3.5 MS-DOS Command Line Arguments
24	3.6 Use of the DISK command
25	4.0 PC-Four DOS Commands
25	AUTO, CAT
26	CLS, COPY, CR
27	DEBUG
28	DIR
29	DISK, DO, EXIT
30	HELP, KEY
31	KILL, LIST, LOAD, MEMORY
32	MOD1000, PRINT, REBOOT, RENAME
33	SETUP, SETUP COLOR, SETUP DRIVE
34	SETUP KEY, SETUP SAVE, QUIT
35	SHELL, STATUS
36	ZERO, 64K, 128K

37	5.0 RUNNING TRS-80 Programs
38	5.1 Special Keyboard Keys
38	5.2 Specific TRS-80 Program Information
39	Allwrite
40	BASIC (6.2), BASIC (DOS+)
41	BASIC (6.3), BASCOM, BEEP, C
42	CAU MZAL, Comm Programs, DISnDATA
43	EDAS, HartFORTH, LESCRIPT
43	Little Brother, MultiPlan
44	PATCH, PASCAL (Alcor/Tandy), PAYROLL
45	PowerMail, PFS File, Profile 4+
46	Superscripts
48	Visicalc, Zeus
49	6.0 TECHNICAL INFORMATION
49	6.1 DOS RST 28 ServiceCalls
58	Video control codes
59	6.2 Special PC Keyboard keys
60	6.3 TRS-80 Keyboard Memory Map
61	6.4 The File Control Block
62	6.5 Port I/O Simulation
63	6.6 Simulator Memory Map
66	6.7 Undocumented Z80 Op-Codes
68	6.8 Error Messages
69	Problem Report Form

PC-Four - An Introduction

Problem: You have a TRS-80 Model 4 which you wish to dispose of, replacing it with something more modern such as an IBM PC or compatible. However you have accumulated valuable Model 4 software and data which you cannot afford to immediately replace. What you would really like to do is to be able to continue to use your old Model 4 software and data on your new PC. With some work, you could copy over and modify your BASIC programs to run. Unfortunately most of the more important software packages written for the TRS-80 are written in machine language for the Z80 microprocessor and your new computer uses an 8088 or 80286 which does not understand Z80 code. Even if you copied your old programs over to your PC they would not work.

The Solution is PC-Four. PC-Four makes your PC emulate a TRS-80 Model 4 (in true Model 4 mode only) complete with a pseudo operating system. Once it is running the screen and keyboard of your PC behave as if you were using a TRS-80. You have four disk drives which may be physically set up as subdirectories on your floppy or hard disks although your virtual TRS-80 sees them as logical drives 0 - 3. Any printer connected to your PC's printer port (LPT1:) appears as if it is on the printer port of your TRS-80.

To use PC-Four you must first copy the programs and files you want to run to the sub-directories you will use as TRS-80 drives 0 to 3. You must also copy PC-Four itself and any other relevant files (see the section on getting started). Now you run PC-Four by typing PC4<enter> and you have a TRS-80 on your screen. You can run many of your programs such as editors, assemblers, word processors and spreadsheets, unmodified. You can even run BASIC/CMD from your old Model 4 operating system and hence your original BASIC programs.

PC-Four provides a complete implementation of the Z80 microprocessor including the undocumented op-codes and will run most machine language programs. Frequently, such programs make extensive use of calls to the operating system for screen, keyboard, printer and file I/O. PC-Four emulates these calls as faithfully as possible, converting them to MD-DOS functions. Some calls are not implemented, see the technical reference section for more details.

PC-Four does not emulate the floppy disk controller of the TRS-80 so you cannot run programs that access this directly. Examples of such programs are Copycat, Hypercross, Hyperzap, Traccess and SuperUtility.

It should be noted that PC-Four is continuously evolving and improving. So many programs used features of the TRS-80 operating system in different ways that it is impossible to test for every eventuality. As time progresses we will make PC-Four run more and more programs. If it does not run your favorite program, or if it gives problems, send a sample disk for evaluation and a letter explaining the problem. We will attempt to diagnose what is wrong and will return your original disk. Please use the problem report form at the end of this manual and return your master PC-Four disk. We will return it with the latest version including a fix for your problem if possible.

2 - Getting Started.

The proper sequence for getting started with PC-Four is as follows:

- 0) **READ THIS MANUAL !!! (Especially section 2.4)**
- 1) Make a back up of your PC-Four distribution disk.
- 2) Put the original away in a safe place.
- 3) Make your actual working disks from the copy.
- 4) Transfer your TRS-80 Files to MS-DOS disks.
- 5) Run PC-Four.
- 6) Finally - Run your TRS-80 program.

However

Just to whet your appetite your PC-Four distribution disk is set up so that it can be run immediately out of drive A. This way you can run it and see that it works and get an idea of some of the commands. To try it out, insert your master disk (or a backup copy) in drive A: and type:

PC4 <Enter>

PC-Four will run and you will see a greeting screen displayed. Press any key to continue and you will see the PC-Four DOS prompt. This is just like TRS-DOS. Type a DOS command like DIR and you will see the contents of "the TRS-80 disk in drive :0". Actually what you see are files in a subdirectory called 0 on your PC-Four disk. We have included a demo TRS-80 Z80 machine language program called RUNME/CMD on your disk so that you can see that, just like a TRS-80, if you type the name of a /CMD file the operating system will load and run it.

RUNME is a simple program that clears the screen, prints some text and returns to DOS. To see it go type:

RUNME <Enter>

When you type the name of a program PC-Four appends /CMD to it and then searches all its "Drives" for a file called RUNME/CMD. If it finds one, it loads it into memory and starts Z80 emulation mode. Your "Disk Drives" 0, 1, 2 and 3 of the TRS-80 are actually sub-directories on your MS-DOS disk.

To see a list of available PC-Four DOS commands type HELP. To see more information on any command type:

Getting Started

HELP [command] <Enter>

Try some of the commands.

To leave PC-Four and return to DOS type:

EXIT <Enter>

Before we get serious, let's see two more features of PC-Four. On "Drive 0" there is a file called TRYTHIS/JCL. PC-Four will execute /JCL files which are text files as if the contents had been typed directly on the keyboard. Also, when you run PC-Four you can include a TRS-80 command on the line after the PC4. At the MS-DOS prompt type:

PC4 TRYTHIS <Enter>

PC-Four will run and search for a file TRYTHIS/CMD or TRYTHIS/JCL. When it finds TRYTHIS/JCL it will execute the contents.

2.1 Making a back up of your PC-Four distribution disk.

For systems with only one diskette drive, insert your DOS system disk in drive A: and type

DISKCOPY A: A: <Enter>

and follow the prompts. When asked to insert the source disk in drive A:, use your master PC-Four disk, and for the target disk use a new, blank disk.

For two diskette drive systems, insert your DOS system disk in drive A: and a blank disk in B: and type

DISKCOPY A: B: <Enter>

When asked to insert the source disk in drive A:, replace the DOS System disk with your master PC-Four disk.

For hard disk systems there should be no need to insert a DOS system floppy disk. With the system booted up and the prompt on C:> (assuming C: is your hard drive and has a DOS System on it) simply use the procedure for one or two floppy drives as given above.

Contents of the PC-Four distribution disk.

The distribution disk contains the following files:

PC4.EXE	The main executable PC-Four program.
PC4CFG.DAT	PC-Four's configuration file.
PC4BOOT.DAT	A special PC4 data file.
HELP.DAT	Help file for PC-Four HELP command.
MOD4.PIC	A file containing the TRS-80 picture.
CONFIG.SYS	A file to configure DOS properly.
HINSTALL.BAT	To install PC-Four on a hard disk.
INSTALL1.BAT	To install on a single drive system.
INSTALL2.BAT	To install on a two drive system.
READ.ME	May contain updates to the manual.
0\	A subdirectory representing TRS-80 drive :0
1\	A subdirectory representing TRS-80 drive :1
2\	A subdirectory representing TRS-80 drive :2
3\	A subdirectory representing TRS-80 drive :3

2.2 Making your working PC-Four System.

Before you start you should understand some of the commands in MS-DOS that you are going to need to use. If you are a long time user of MS-DOS then go right ahead. However if you are new to PCs and don't know about root directories, subdirectories, paths, or the commands 'mkdir', 'rmdir' and 'cd' then read section 2.4 "What you should know about MS-DOS" before you go any further.

For proper operation PC-Four requires that most of the above files be present on the disk from which you start. What you have to do is to create a working disk or disks which include the PC-Four files and up to four MS-DOS file path specifications (drive specs or subdirectories) which will represent four possible TRS-80 disk drives 0 - 3. Note that if you wish you can restrict yourself to less than four "drives" by only making the necessary subdirectories.

To be sure that you can run any program your DOS CONFIG.SYS file should contain the line

FILES=15

The CONFIG.SYS file is a file on your MS-DOS boot disk. It causes your MS-DOS operating system to set itself up automatically. It consists of a series of statements like the one above. Files=15 ensure that DOS can have up to 15 files open at once. If you already have this statement present with a number larger than 15 then leave it unchanged.

Follow one of the procedures below to make a working setup. Three cases are presented for one drive, two drive and hard-disk systems. If you have a hard disk it is highly recommended that you install PC-Four on your hard disk as this will accelerate the performance substantially. For each of the three cases you can use the appropriate batch file listed above to automatically install PC-Four

or, if you want to customize it in some way follow the detailed instructions and adapt as needed.

1) Single Drive Systems.

You should create two disks:

Disk One, the startup disk should contain a minimum MS or PC-DOS system including COMMAND.COM plus the following files: PC4.EXE, PC4CFG.DAT, MOD4.PIC and CONFIG.SYS. The file CONFIG.SYS should have a line in it with FILES=15 to ensure that at least 15 files can be opened at once. COMMAND.COM is a DOS file.

Disk Two, the running disk should contain the files COMMAND.COM, HELP.DAT, PC4CFG.DAT, MOD4.PIC and PC4BOOT.DAT. It should also contain up to four subdirectories called 0 1 2 and 3.

To make Disk One, insert your DOS disk in drive A, and type:

FORMAT A:/S

You will see the following message:

**Insert new diskette for drive A:
and strike any key when ready**

Now put a blank disk in drive A and strike any key. You should see a sequence of messages indicating the formatting is proceeding, formatting the disk double sided with 40 tracks numbered from 0 to 39. Label this disk Disk One.

Next, insert your PC-Four working disk in drive A: and type:

COPY *.* B: <Enter>

You will then be prompted to change disks as needed. Each time you are asked for the disk for drive A: insert your PC-Four disk and for drive B use the Disk One disk you just formatted. Label this "Disk One" when you have finished.

To make Disk Two, format a disk as above but without a system on it using:

FORMAT A: <Enter>

With your DOS disk in drive A: add the file COMMAND.COM to Disk Two with:

COPY COMMAND.COM B: <Enter>

Label this disk Disk Two, insert your PC-Four disk in Drive A:. You can then either use the batch file INSTALL1.BAT to finish making Disk Two or you can do it manually as follows:

Getting Started

Copy the needed files using the commands:

```
COPY  HELP.DAT  B: <Enter>
COPY  PC4BOOT.DAT  B: <Enter>
COPY  MOD4.PIC  B: <Enter>
COPY  PC4CFG.DAT  B: <Enter>
```

Each time you are prompted to swap disks, use your PC-Four disk for A and Disk Two for B. Consult your DOS manual if you have trouble copying files.

Finally, insert Disk Two in your drive and type:

```
MKDIR  0 <Enter>
MKDIR  1 <Enter>
MKDIR  2 <Enter>
MKDIR  3 <Enter>
```

You have created four subdirectories representing the four TRS-80 disk drives :0, :1, :2 and :3. You will have to transfer your working TRS-80 programs and data files to these subdirectories before you can do much more. Assuming that you have already transferred them to an MSDOS disk you should insert that disk in drive A: and type

```
COPY  *.* B:\0 <Enter>
```

Each time you are prompted for disk for drive A insert the source disk and each time you are prompted for the diskette for drive B: insert your Disk Two. This example will copy all files (*.*) on your diskette to the subdirectory named 0 on the target disk.

Label this disk "Disk Two". You can make as many of these disks as you like, putting different TRS80 files in the subdirectories.

2) Two Disk Drive Systems.

You should create three disks:

Disk One, the startup disk which contains a minimum MS or PC-DOS system including COMMAND.COM plus the following files: PC4.EXE, PC4CFG.DAT, MOD4.PIC and CONFIG.SYS.

Getting Started

Disk Two, the running disk should contain the files COMMAND.COM, HELP.DAT, MOD4.PIC and PC4BOOT.DAT. It should also contain two subdirectories called 0 and 1.

Disk 3 should contain 2 subdirectories called 2 and 3.

Proceed as for a single drive system to make Disk One except that you will start with a blank disk in drive B: and, with an MS-DOS system disk in drive A:, type:

FORMAT B:/S <Enter>

Then, insert your PC-Four disk in drive A and type:

COPY A:*. * B: <Enter>

To make the working disks with subdirectories for TRS-80 drives 0 to 3, you have the choice of putting them all on Disk Two, or making a supplementary disk, Disk Three.

To make Disk Two or Disk Three, insert a blank disk in drive B: and type:

FORMAT B: <Enter>

With your DOS disk in drive A: again, add the file COMMAND.COM to Disk Two in drive B: with

COPY COMMAND.COM B: <Enter>

Then, if you are making Disk Two copy the necessary files (HELP.DAT, MOD4.PIC and PC4BOOT.DAT) to it and make the subdirectories using your PC4 disk in drive A: and the command

INSTALL2

You might prefer to restrict yourself to just two TRS-80 equivalent disk drives, in which case you do not need to make subdirectories: you can simply use drive A: as :0 and B: as :1.

You will have to transfer your working TRS-80 programs and data files to these subdirectories before you can do much more. Assuming that you have already transferred them to an MSDOS disk you should insert that disk in drive A: and and with Disk Two in drive B: type

COPY *. * B:\0 <Enter>

This example will copy all files (*.*) on the first disk in drive A to the subdirectory named 0 on "Disk Two" in drive B:

3) Hard disk systems.

If you have a hard disk you should use install PC-Four on it for optimum performance. A batch file **HINSTALL.BAT** is provided on your PC-Four disk for this. To use, insert your PC-Four disk in drive A: and type:

A:HINSTALL C: <Enter>

Note: If your Hard Disk is not C:, substitute the appropriate letter. Also, HINSTALL will add the line FILES=15 to your CONFIG.SYS and will rename your old CONFIG.SYS file as CONFIG.OLD.

If you want to customize your installation you should do this by creating a subdirectory called say, TRS80 by typing:

C> MKDIR TRS80 <Enter>

and then change directories to it and copy all the PC-Four files over:

C> CD TRS80 <Enter>

C> COPY A:*.*

Finally, make four sub-directories thus:

MKDIR 0 <Enter>

MKDIR 1 <Enter>

MKDIR 2 <Enter>

MKDIR 3 <Enter>

You will have to transfer your working TRS-80 programs and data files to these subdirectories before you can do much more. Assuming that you have already transferred them to an MSDOS floppy disk here is an example of how you should get them in the right subdirectory. Insert the disk in drive A: and type

COPY A:*. * 0 <Enter>

This example will copy all files (*.*) on the disk in drive A to the subdirectory named 0 in the current directory on the default disk drive.

2.3 Transferring files to MS-DOS.

To actually use PC-Four you will have to have some real TRS-80 files on your PC. PC-Four and all its support files are strictly MS-DOS and function as an interpreter of your TRS-80 programs. So, somehow you must move your old TRS-80 programs, data files etc., from their original Model 4 TRS-DOS 6 disks. You must move them to MS-DOS formatted disks in such a way that you have an identical file with no changes whatsoever. We call this an "Image Copy" in that you have an identical image of the file in MS-DOS exactly as it was in TRS-DOS.

If you are using a hard disk or sub-directories on floppy disks, don't forget that your files must be copied into those subdirectories and not into the root directory. If you don't do this properly you will not see any files when you run PC-Four. The easiest way to do this is to change directory (cd) to the subdirectory you intend to use before running the transfer program (or copy if you have used Hypercross).

If you have not got any means of transferring files already, Hypersoft sells two programs which can do the transfer for you. One, **HyperCross**, runs on the TRS-80 Model 4 and the other, **PC Cross-Zap** runs on the PC. Use of these programs is described below.

2.3.1 Format of files on PC-DOS vs TRS-DOS.

TRS-80 files differ from PC files in several ways. Firstly, there are no passwords associated with PC files. If you use Hypercross to transfer password protected files you will have to specify the password at transfer time. Or, you can remove the password before you make any transfers. In any case it will be lost when it reaches the PC disk.

Secondly, TRS-80 files have a "Logical Record Length" (or LRL) associated with them. This is usually set to 256, being the same as the size of one sector on a TRS-80 disk. For some files, particularly data files, the LRL is set to some other size. This is used as an indicator to DOS and user programs that each "Record" is only LRL bytes long. For instance, if you see on your TRS-80 directory a file which has an LRL of 80 this means it has records (for example names and addresses of customers) which are each exactly 80 bytes long and are stacked end to end through the file. In practice this information is not needed and MS-DOS does not provide this information in its directory structure.

Third, file names will appear in MS-DOS with the familiar / replaced by a period. For example a file such as TASMOM/CMD on the TRS-80 will become TASMOM.CMD when copied to the PC if you look at it using the MS-DOS DIR command. However, if you look at it with PC-Four's DIR command you will see it as TASMOM/CMD. This is essential as TRS-80 application programs expect to see a '/' and not a '.'.

TRS-80 file names should consist, just as before, of a main part of up to 8 alphanumeric characters (A-Z,0-9) with the first character being a letter. A second part of up to 3 alphanumeric characters can be appended after the period.

Getting Started

Lastly, note that TRS-80 text files use only an ODH (carriage return) character at the end of each line whereas PC files use an ODH OAH (carriage return, line feed) pair of characters. Files for PC-Four must be an exact image of the TRS-80 files - that is with an ODH only. **Do not use ASCII transfer mode** with HyperCross or PC Cross-Zap but treat these files as Binary. To view an ASCII file use the PC-Four LIST command. To print use the PRINT command. Note that each ODH will be padded with an OAH so your PC printer will see the correct PC format and advance to the next line properly.

2.3.2 Transferring files using HyperCross on the Model 4

Follow your Hypercross manual. With a Model 4 system disk in drive :0 and an MS-DOS disk in drive :1 copy your files from drive :0 to drive :1 using the -I image copy option.

A special point to note: To transfer some files such as BASIC, you will have to remove the passwords and/or make BASIC visible. For access, BASIC is unprotected. The update password for BASIC/CMD and BASIC/OV1 is BASIC. So, in this example use:

ATTRIB BASIC/CMD.BASIC (VIS) to make the file visible

ATTRIB BASIC/CMD.BASIC (USER=,OWNER=) to remove passwords.

You can then transfer it freely using Hypercross.

Once the files are on MS-DOS disks you can take them to your PC and copy them onto the appropriate disks or subdirectories as needed. For example, if you have a hard disk and have created subdirectories 0, 1, 2 and 3 in your TRS80 subdirectory do this:

```
C> CD C:\TRS80\0 <enter>
C> COPY A:*. * <enter>
```

Or you can type this:

```
C> COPY A:*. * C:\TRS80\0
```

If you have a floppy system put the PC disk that you made with Hypercross in drive A: and the PC-Four Disk Two in drive B and type:

```
A> COPY A:*. * B:\0
```

See section 2.4 for more information on MS-DOS commands.

For your information the following TRS-80 Model 4 files are password protected and you must remove the passwords before they can be accessed:

DOS	File	Update Password HEX codeword
TRS-DOS 6.2	BASIC	782F BASIC

TRS-DOS 6.2	PATCH/CMD	BFAE	QJ60
LS-DOS 6.3	BASIC/CMD	BFAE	BASIC
LS-DOS 6.3	BASIC/OV1	782F	BASIC
LS-DOS 6.3	BASIC/OV2	782F	BASIC
LS-DOS 6.3	PATCH/CMD	BFAE	QJ60
LS-DOS 6.3	TED/CMD	BFAE	QJ60
DOS+ 4.0	BASIC/CMD	A7CB	CMD
DOS+ 4.0	BASIC/OV1	782F	BASIC
DOS+ 4.0	REF/CMD	A7CB	CMD
DOS+ 4.0	SORT/CMD	A7CB	CMD

2.3.3 Transferring files using PC Cross Zap

Follow your PCXZ manual. Select the alien format corresponding to TRSDOS 6.x single or double sided as appropriate. Go to the file transfer menu (#2 from the main menu) then change the defaults so that the program will normally copy in Image mode. You might want to save these settings in a configuration file so that PCXZ will come up automatically in this mode whenever you run it. Then, from the file transfer menu select the Copy Alien to MS-DOS option and copy your files from TRS-DOS to MS-DOS.

If you have a floppy only system, insert your PCXZ disk in drive 0, run PCXZ and remove the disk. Then insert a pre-formatted MS-DOS disk in drive A and a TRS-80 Model 4 disk in drive B. Change to the file transfer menu and change defaults if necessary to ensure that the files will be copied in image format and that they will go to the disk in drive A and the appropriate sub-directory on drive A if necessary.

If you are using a hard disk you must copy the files into the correct subdirectory. To do this you could change directory (e.g. CD \TRS80\0) to the appropriate sub-directory before starting PCXZ. Alternatively, at the file transfer menu, select the change defaults option and enter the appropriate sub-directory name when asked.

Note that if you want to transfer BASIC/CMD and BASIC/OV1 you will have to first make them visible. PCXZ does not care about passwords, however files must be visible. To make invisible files visible you can use either the ATTRIB command of TRS-DOS as above (in section 2.3.2) or, you can use the Disk Navigator feature of PCXZ to zap (modify) the byte in the directory entry which sets the files visibility.

The PCXZ manual contains sufficient information in the disk formats section to allow you to work out how to change the protection on any disk using the Disk Navigator feature but here, in detail, is how you do it for BASIC/CMD and BASIC/OV1.

Select the Disk Navigator Menu (#6 from the main menu). Then choose #2, View Directory Sectors, with of course the TRS-DOS disk containing BASIC in the default drive. You will see the first sector of the directory. Press the PgUp (Page Up) key until you see BASIC/CMD or BASIC/OV1 at the right hand side of the

sector display. Move the cursor using the arrow keys until you are on the first byte of the directory entry for this file - it should be a 1D. Press the "M" key which will put you in modify mode. Change the 1D to 16 and below it the 782F password to 9642 by overtyping the correct characters. Then press F2 to exit modify mode and Z to zap (write) the sector back to disk. You have now made the file visible and removed the password and can copy the file just like any other.

2.3.4 Transferring using RS232 null-modem cable or modem

Another way to transfer files from a TRS-80 to your PC is to connect the computers together via the serial RS232 ports using a null modem cable. You will need programs that run on each machine that will talk to one another and allow you to transfer binary files so that you end up on the PC with exact copies of the originals. Programs that use Kermit or XMODEM protocol will do this for you.

Instead of a cable you can of course use a modem on each computer and transfer files via a phone line. Many public domain Model 4 programs are available on Bulletin Boards and public access computer services such as CompuServe and GENie. You will have to download your files using Kermit or XMODEM protocol to ensure they are exact and error free.

There may be some limitations to transferring files using xmodem protocol. The problem is that XMODEM does not preserve the exact end of file mark and it will not work with files with a Logical Record Length that is not 256. For some programs it is essential that the exact file size on the PC matches that on the TRS-80. If you are having problems check the original file size, number of records and end of file (EOF) in the TRS-80 directory and the equivalent in your PC-Four directory.

XMODEM transmits files in multiples of 128 bytes and does not preserve the exact file size. A way round this is to chop the end off after transfer using a cleanup program. Xmodem will also not send files which do not have an LRL of 256. Your only way round this (apart from buying Hypercross or PC Cross-Zap) is to pre-convert these files to an LRL of 256. This can be done by simply zapping the LRL byte in the directory to 00 or by copying it to a new file using a simple BASIC program.

2.3.5 Transferring files using TRSCROSS.

With caution you CAN use Powersoft's TRSCROSS to copy files to MS-DOS for use with our PC-Four but you should only do this using TRSDOS transport disks previously formatted using TRSCROSS. You must tag ALL files to be copied as BINARY.

From dealing with customers who have used TRSCROSS and tests we have run we are of the opinion that TRSCROSS does not report sector read errors when copying BINARY files. If it finds a sector it cannot read, it simply substitutes the last one it read. This may occur as frequently as once on every track i.e. about one in 18 sectors will be lost. This is due to the infamous "gap format problem" and can only be overcome if you first format a disk on TRSCROSS, carry it back

Getting Started

to the Model 4, copy your files to that, then return to the PC and copy the files off using TRSCROSS. Even then, if an error occurs, you may have no indication of it until you try and run the files with PC-Four.

These comments relate to TRSCROSS 2.0 and may of course change with future releases or versions of the program.

2.4 What you should know about MS-DOS.

If you are just upgrading from a TRS-80 to a PC type machine running MS or PC DOS the you should read this section. There are some important commands and concepts you should understand before using PC4.

2.4.1 Disk Drives and Directories.

Each of your disk drives is known by a letter instead of a number. For example the first drive in the system is A: (instead of :0 in a TRS80). Normally the system boots off drive A (or the hard disk if you have one and no disk is in drive A).

Each disk has a main directory known as the Root Directory. This is just like the directory on a TRS-80 disk. Typing DIR will display the directory. Typing DIR B: will show the files on the disk in drive B:.

2.4.2 Sub Directories.

Within your main directory you can have one or more subdirectories. These are just like the main directory except the contents are not immediately visible if you type DIR. All you will see is an entry with the subdirectory name followed by a <DIR> indicator to show that it is a directory and not a file.

Subdirectories are a convenient way of dividing up your main directory into subject areas so that it is not cluttered with too many entries. For example you could have a subdirectory called BAS and keep all your BASIC files in that.

In PC Four we use subdirectories to separate TRS80 files from one another and from PC-DOS files. For example you could have a directory called PF with all your PROFILE files, or a directory called SS with SCRIPSIT files in. Using the DISK command in PC-Four will define any one of these directories to be a "TRS-80 disk drive".

To see the contents of a subdirectory type DIR subdirectory name.

You can copy files from one disk or subdirectory to another but you must be careful to specify the disk names and subdirectories properly. The full specification for where a file is on a disk is called a PATH.

2.4.3 PATHS.

The PATH is the name of a Disk Drive followed by a \ followed by any subdirectory name followed by successive sub-subdirectory names separated by \ symbols.

For example suppose you have a Hard Disk C: and in the main directory of your hard disk you have a subdirectory called TRS80 containing your PC-Four files and some sub-subdirectories. If one of those directories is called "0" and in that is a file called BASIC.CMD then the path for that subdirectory is

C:\TRS80\0

The full path description for the file is C:\TRS80\0\BASIC.CMD

You can imagine the directory structure to be a tree with the main directory at the root and each of the subdirectories a branch. The PATH gives us a way of specifying how to get to any file from the root.

For hard disk owners DOS has a very useful command called PATH. Using this you can specify what subdirectories are to be searched each time you type a command. For instance if you put all your utilities including PC Cross-Zap (XZ.EXE) in a main subdirectory called UTILITY then you can run this from anywhere (any subdirectory) by typing XZ if you have previously typed:

PATH=C:\;C:\UTILITY;

This causes DOS to search the current directory, the root directory on C: and subdirectory C:\UTILITY for XZ.EXE or XZ.COM or XZ.BAT. Put this line in your hard disk autoexec.bat file so it will be automatically set every time you start your computer.

2.4.4 Current Drive and Directory.

When you type DIR or do any other kind of disk operation (COPY, DELETE etc.) your DOS assumes you mean the current or default disk and directory unless you specify otherwise. The default disk is shown by the DOS prompt. For example if you see A> that means that A: is your default drive. Type DIR and you will see the directory on A:. If C: was the default then the prompt would be C> and DIR would show the contents of C.

But wait, there is more. Not only do you have a default drive you have a default directory. Unless you have changed it with the CD command it will be the main directory of the default drive. If you have changed it then any command will act on files in the current directory and not the root or main directory.

This is important. If you have a hard disk then your PC-Four files will be in the subdirectory called TRS-80 and you must be in that directory to run PC-Four.

2.4.5 Changing defaults.

To change default drives you simply type the name of the drive followed by a colon (:). Example to change from A: to C: you type C: <enter>.

To change subdirectories downwards you type CD subdirectory name or CD path. For example if you are on hard drive C: and want to go down one level to TRS80 type CD TRS80 <enter>. To go down again type CD 0 <enter>. To go directly from any directory to 0 type CD \TRS80\0 <enter>.

To go back up one level of directory type CD .. <enter>. To go right back to the root directory type CD \ <enter>.

To change the default on another drive, not the one that is currently your default include the drive name in the path specifier. For example if your default is A: and you want to change the one for C: you might type

CD C:\TRS80\0

2.4.6 Making and Deleting Subdirectories.

To make a new directory use the mkdir command:

MKDIR directory name.

The directory name can have up to 8 characters and the new directory will be placed in the current drive and subdirectory unless you specify otherwise.

To delete a directory use the rmdir (Remove Directory) command:

RMDIR directory name.

Note: the directory must be empty before removing it.

2.4.7 Copying files.

To copy files from one place to another you must give the source and destination paths as well as the file names. For example to copy a file from a disk in drive A to a subdirectory on drive C: you might type

COPY A:BASIC.COMD C:\TRS80\0

3 - Running PC-Four.

If you are running PC-Four on a PC with floppy drives only you should make working disks up as described in Chapter 2. You can also use a copy of the PC-Four distribution disk.

To start PC-Four you must insert a floppy in drive A: containing the following files: PC4.EXE, HELP.DAT, PC4CFG.DAT, PC4BOOT.DAT and MOD4.PIC.

Once PC-Four is running you can remove the starting disk and replace it with a working disk containing subdirectories representing TRS-80 floppy disks.

Using the DISK command you can define any of your floppy drives as a TRS80 disk drive directly without using subdirectories. We have provided an alternative PC4CFG.DAT file named PC4CFG.ALT for this purpose. To use this delete the standard PC4CFG.DAT and rename PC4CFG.ALT to PC4CFG.DAT. This file will set up drive A: as :0 and drive B: as :1.

Disk One, the startup disk on a floppy system should have these files:

DOS system + CONFIG.SYS + PC4.EXE, PC4CFG.DAT, MOD4.PIC

Disk Two, the running disk on a floppy system should have these files:

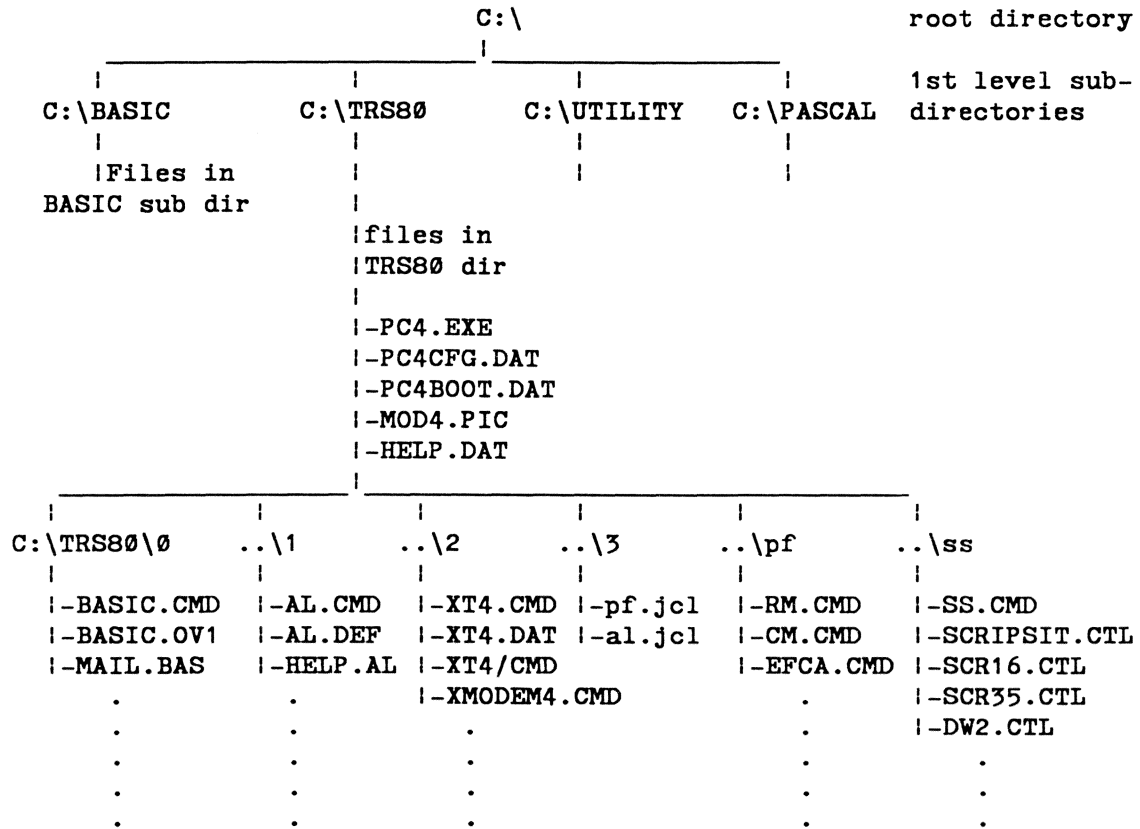
Main PC-Four files	Subdirectories	Equivalent TRS-80 drives
A:\	A:\0	= :0 - user programs
COMMAND.COM	A:\1	= :1 "
PC4CFG.DAT		
PC4BOOT.DAT	A:\2	= :2 "
MOD4.PIC		
HELP.DAT	A:\3	= :3 "

Here is an example of how to arrange PC-Four on a hard disk:

On the next page is a suggested directory structure that you might set up to run PC-Four on a hard disk drive C: Note that in the TRS80 subdirectory are all the files that are needed to run PC-Four. Below this are the subdirectories representing various TRS80 "disks". You can have as many of these as you like. Each one has a unique name and can be selected using the PC-Four DISK command. For example, to do the equivalent of inserting your Profile disk in drive 0 type DISK 0 PF <enter>.

Running PC-Four

Hard Disk Directory Structure for PC-Four



You should copy your working TRS-80 programs to one or more of the subdirectories equivalent to one of the TRS-80 drives. Major programs such as SCRIPSIT or PROFILE should have all their working files put on drive :0.

To run PC-Four from a **one drive** system insert Disk One and type:

PC4 <Enter>

When running, insert Disk Two in drive A:.

To run PC-Four from a **two drive** system insert Disk One and type:

PC4 <Enter>

When running, insert Disk Two in drive A: and Disk Three in drive B:.

To run PC-Four from a **hard disk** change to the PC4 sub-directory and type:

Running PC-Four

PC4 <Enter>

When PC-Four runs it displays the DOS level screen and you are in TRS-80 emulation mode. You can type any valid PC-Four DOS level command at this point and it will be executed. See the section in this manual on DOS commands or type HELP in PC-Four for details of the available commands and syntax. You can also type the name of any /CMD or /JCL file on any drive :0 to :3 and PC-Four will search and if found execute it. Programs that call supplementary overlay files should have those files on drive :0.

3.1 Configuration options.

PC-Four allows you to specify various set-up options so that you can customize it to your own preferences. This information is contained in a text file called **PC4CFG.DAT**. If this file is present when you start PC-Four it will be loaded and used to set the configuration. If not then a default setting for each parameter will be used.

Once PC-Four is running you can change some of the options using the **SETUP** command and once set to your liking you can save the new configuration using the **SETUP SAVE** command. Since it is a regular MS-DOS text file you can also edit it with a PC editor and change it that way if you prefer. This lets you change items that cannot be altered directly from within PC-Four (those items marked with a *)

The Drive Specifications provide PC-Four with the translation information so that when you, or a program, attempt to access a TRS-80 drive, your request is converted from - say: drive :0 to A:\PROFILE (the subdirectory called PROFILE on the disk in drive A:). You can change this from within PC-Four using the **SETUP DRIVE PATH** or **DISK** commands.

Foreground and background colors to be displayed on normal and inverse TRS-80 video are also specified within **PC4CFG.DAT** and can be changed from PC-Four using the **SETUP COLOR** command.

When you first run PC-Four it displays a picture of a TRS-80 on the screen. You can bypass this by changing the Y on line 10 of **PC4CFG.DAT** to a N.

This is the format of the configuration file:

Line Number	Use	Example
1	Drive :0 specification	0\
2	Drive :1 specification	1\
3	Drive :2 specification	2\
4	Drive :3 specification	A:\
5	Normal Foreground Color	10
6	Normal Background Color	0
7	Inverse Foreground Color	14
8	Inverse Background Color	4
9	PC-Four AUTO command	CAT :0
10 *	Start with TRS-80 picture	Y
11 *	DOS level Function Key F1	Help
12 *	DOS level Function Key F2	Exit
13 *	DOS level Function Key F3	List
14 *	DOS level Function Key F4	Print
15 *	DOS level Function Key F5	Copy
16 *	DOS level Function Key F6	Ren
17 *	DOS level Function Key F7	Debug
18 *	DOS level Function Key F8	Kill
19 *	DOS level Function Key F9	Cat
20 *	DOS level Function Key F10	Dir

3.2 The Function Keys:

The PC function keys F1 - F10 have a dual purpose. At the PC-Four DOS level they give you a quick way of typing a command. You can substitute any other commands you prefer, including program names by editing the PC4CFG.DAT file using an MS-DOS editor. Whenever you press a function key at the PC-Four DOS level, the equivalent text is substituted as if you had typed it directly. Display of the function key information at the bottom of the screen can be controlled using the KEY ON and KEY OFF commands.

Once you are running a TRS-80 program the above assigned values for the function keys are no longer active. Instead each 'F' key generates a special TRS80 keyboard code. The codes are held in a programmable table. Using the command SETUP KEY you can define any special code you might need for a TRS-80 program. The table below shows the settings of the function keys as initially supplied in PC-Four:

Function Key	Use	Code Generated
F1	TRS-80 Function Key F1	80H
F2	TRS-80 Function Key F2	81H
F3	TRS-80 Function Key F3	82H
F4	Shift-@ = Pause	60H
F5	Shift-Clear Key	1FH
F6	Shift-Clear-Equals	BDH
F7	Clear-Shift-1	A1H
F8	Clear-Shift-2	A2H
F9	Clear-B	C2H
F10	Break	80H
Ins	Clear-I	C9H
Del	Clear-D	C4H

You should probably not change keys F1-F3 as they are used by a number of TRS-80 programs. Shift-@ key F4 will cause a pause in listing a file in a BASIC program. The Shift-Clear key is used by some programs as a function select key and by others to clear the screen.

To change any F key assignment type

SETUP KEY n 0xxH where n is key #, xx is Hex code.

See the section on DOS commands for more details.

3.3 Other Keyboard Keys

Most keys on your keyboard correspond exactly to keys on the Model 4 with the exception of some special keys and key combinations. PC-Four maintains a keyboard map just like the memory mapped TRS-80. Each time a key is pressed a bit is set in the map corresponding to the TRS-80 row-column intersection for that key. When the key is released the bit is reset. Programs that use this map through the DOS RST 28H call can use almost all keyboard combinations. Use the **ALT** key for the TRS-80 **CLEAR** key and for combinations of Clear-key and other keys. Thus for Clear-C use Alt-C.

NOTE: For some programs, you must release the Alt key before pressing the second key of a two key combination.

The TRS-80 **BREAK** key is simulated using the **Esc** (Escape Key) or the **F10** key. Both return the code 80H however there is one difference between these two codes: The Esc key also sets the break bit in the system flag table. Some programs (BASIC for example) check this bit periodically to see if a Break has been requested recently. In this case only the Esc key will work.

The PC keyboard has a keypad with arrow keys and the Home, End, PgUp and PgDn keys. The arrow keys correspond to the TRS80 arrow keys while the others correspond to pressing shift and arrow at the same time as shown below:

Keyboard Key	Use	Code Generated
Up-Arrow	Up-Arrow	0BH
Page Up (PgUp)	Shift Up-Arrow	1BH
Down-Arrow	Down-Arrow	0AH
Page Down (PgDn)	Shift Down-Arrow	1DH
Right-Arrow	Right-Arrow	09H
End	Shift Right-Arrow	19H
Left-Arrow	Left-Arrow	08H
Home	Shift Left-Arrow	18H

3.4 JCL Files

PC-Four provides limited support for JCL batch files in this version. A JCL file is a text file that provides a sequence of instructions that are executed automatically. The instructions are used as if they had been typed at the keyboard. If the NAME of a JCL file is type PC-Four DOS will search the available drives 0..3 for any file called NAME/JCL. If it is found then it will be read in line by line and executed. For compatibility with TRSDOS you can also type DO NAME.

PC-Four supports the // SUSPEND pseudo command only at present. If this occurs, interpretation of the JCL file stops and a message is printed. Operation resumes when any key is pressed.

Users of hard disks can put /JCL files in say drive :3. Then they can be used like alias commands. For instance if you have a subdirectory called AW for ALLWRITE and another for documents called DOC you can have a simple JCL file called AW/JCL containing the following:

```
DISK 0 AW
DISK 1 DOC
CR OFF
CAT :1
```

Place this in drive :3 and by typing AL <Enter> PC-Four will find AL/JCL, run it which will reallocate drives 0 and 1, turn automatic printer padding off and display a catalog of your DOC directory. See the DISK command for more information on reallocating disk drives.

3.5 MS-DOS Command Line Arguments

Instead of just typing PC4 to run the emulator you can add an argument to the command such as

```
PC4 AL MYFILE/DOC
```

This will cause MS-DOS to run PC4 as usual however if any command exists on the line immediately following the PC4 then this will be interpreted as if it had been typed as a DOS level command from within PC4. This example would run ALLWRITE and load the textfile MYFILE/DOC to start editing.

A command line argument or AUTO command forces PC4 to skip the display of the Model 4 picture and to begin immediate execution of the command.

The command line argument can of course call a JCL file to initiate a sequence of DOS commands.

You can create MS-DOS BAT files to auto run specific TRS-80 programs. For instance you could have a file in MS-DOS called AW.BAT containing the following:

```
CD \TRS80
PC4 AW
```

The first line forces change to the PC-Four subdirectory.

The 2nd runs PC-Four which executes the AW/JCL file.

3.6 Use of the DISK Command

For Hard Disk users PC-Four has a very useful command which lets you keep your programs in separate subdirectories. Using the DISK command you can select which ones are in use. You can think of it as just like inserting disks in a disk drive.

For example you could have your Profile files in a subdirectory called PF and your data files in a directory called PFDAT. Using the DISK command you type

```
DISK 0 PF <Enter>
DISK 1 PFDAT <Enter>
```

and the DISK command sets up PF as drive 0 and PFDAT as drive 1.

A good way to automate this is to create a number of /JCL files and put them in drive :3 so that whatever drives you have selected for 0 and 1 they can always be found. Then, if you have put the above two lines in a file called PROFILE/JCL you can switch to running PROFILE by typing PROFILE <Enter>. Remember that PC-Four does not require you to type D0 to execute a /JCL file.

You might also want to keep all your utilities on drive :3 so that they can always be found regardless of the 'disks' in drives 0-2.

4 - PC-Four DOS Commands.

This section describes in detail each of the available DOS commands available in PC-Four. In each case the command is executed by typing it, together with any arguments, and terminating the line with an Enter (or Return). Note that commands at the DOS level are not case sensitive so you can type commands in either upper or lower case. In the detailed descriptions of commands given below the use of square brackets [] indicates an optional parameter.

AUTO - create an AUTO PC-Four startup command.

Syntax: AUTO [autocommand]

Function: Sets up a new AUTO command 'autocommand' in the configuration file PC4CFG.DAT so that when you run PC-Four the command is automatically executed. AUTO without any argument removes the current AUTO command. If you are running from a floppy based system, make sure the disk in the default drive is not write protected as the AUTO command forces a SETUP SAVE, saving a fresh copy of PC4CFG.DAT.

Example: AUTO myfile (runs MYFILE/JCL at startup).

CAT - Display shortform disk directory.

Syntax: CAT [:d] where d is a drive # from 0 - 3

Function: Displays a simplified directory listing on the screen showing only file names sorted alphabetically.

Example: CAT :0 displays directory of drive 0

DOS Directory Drive 0

bank/cmd	dodir0/cmd	dos31a/pat	ed/asm	ed/cmd
lrl/asm	lrl/bas	lrl/cmd	ltest/cmd	lu4150/doc
lu4150/dqc	pf/jcl	qb/inf	qb1/inf	r43/asm
r43/cmd	rand0/cmd	rand1/cmd	ramtest/cmd	return/cmd
squsq4/doc	tas/jcl	tas/jql	tas1/jcl	tasmon1/cmd
tasmon4/cmd	tasmonc/cmd	tasmond/cmd	tasx/jcl	tasx/jql
ted/cmd	test/asm	test/cmd	testc/asm	testc/cmd
testd/asm	testfile/doc	vc/cmd	vidtest/cmd	yyy/dup
ztest/asm	ztest/cmd			

>

CLS - Clear Screen.

Syntax: CLS

Function: Clears Display Screen and returns cursor to top left hand corner.

COPY - Copy a file from one disk to another.

Syntax: COPY source/ext[:s] [destin/ext][:d]

Function: Copies the file called "source/ext" on drive number s to drive number d where it is called "destin/ext"

Example: COPY COSTS86/VC:0 COSTS86/BAK:2 - copies the file called COSTS86/VC on drive 0 to drive 2 where the name is COSTS86/BAK.

Example: COPY MYFILE:3 :1 - copies the file called MYFILE on drive 3 to drive 1.

CR - Enable/Disable Printer Linefeeds

Syntax: CR ON or CR OFF

Function: enable/disable automatic adding of linefeeds to carriage returns on printer output. TRS80 type programs normally put out only a carriage return (0DH) character at the end of each printed line. If you have a printer on your PC that expects a carriage return, line feed at the end of each line PC-Four will automatically add a line feed (0AH) to each carriage return sent to the printer. However some programs like ALLWRITE put out their own line feeds when printing in proportional mode. For these you must disable the auto line feed with the AUTO OFF command.

Example: CR OFF disable auto line feed.

Example: CR ON reenable auto line feed.

DEBUG - Inspect, modify and disassemble memory.

Syntax: DEBUG 0nnnnh
 or: DEBUG BANKn (n=1 or n=2)

Function: Runs a built in program which allows you to view Z80 code space, including the extended banks. The screen editing features allow you to modify memory. The argument 0nnnnh is the hex address of the start of the area you want to see. This is not the same as TRSDOS debug.

Example: debug 6000h

```
|Main Memory at Segment: 23C5
|6000  CD 54 6A 06 04 3E 0F EF 22 DD 79 31 52 7A 3E 65   .Tj..>..".y1Rz>e
|6010  EF FD CB 03 0E FD CB 0A EE 0E 0F 3E 02 EF CD B9   .....>....
|6020  66 21 9F 78 3E 0A EF 3E 01 32 C2 79 CD 03 6B CD   f!x>...>2.y..k.
|6030  BF 63 CD EB 69 CD 43 60 CD AC 60 CD 43 60 31 52   .c..i.C'...'C'1R
|6040  7A 18 F5 D5 C5 E5 2A DF 79 24 7C FE 18 CC 5C 60   z....*.y$|...'
|6050  22 DF 79 06 03 3E 0F EF E1 C1 D1 C9 F3 3E 86 D3   ".y.>.....>..
|6060  84 3E 0F 21 B0 FA 22 6A 60 11 00 00 62 6B 01 50   >.!.."j'...bk.P
|6070  00 09 22 6A 60 01 20 00 ED B0 3D 20 EC 3E 87 D3   .."j'. ...= >..
|6080  84 FB 21 30 17 E5 06 03 3E 0F EF 0E 1E 3E 02 EF   ..!0...>....>..
|6090  E1 C9 06 05 3E 0F 2A EF 79 EF F3 3E 86 D3 84 3A   ....>*.y.>....:
|60A0  40 F4 E6 01 20 F9 3E 87 D3 84 FB C9 0E 5E 3E 02   0... >.....^>.
|60B0  EF 3E 01 32 C2 79 CD 5A 61 FE 1F 20 09 CD 54 6A   >2.y.Za...Tj
|60C0  CD EB 69 C3 BF 63 FE 49 CA A3 67 FE 47 CA C4 66   ..i..c.I..g.G..f
|60D0  FE 44 CA 52 67 FE 42 CA 31 66 FE 43 CA 93 66 FE   .D.Rg.B.1f.C..f
|60E0  4D CA 13 65 FE 52 CA 69 64 FE 0A CA AE 67 FE 46   M..e.R.id....g.F
|60F0  CA BF 67 FE 41 CA 48 68 FE 48 CA 3E 68 FE 45 CA   ..g.A.Hh.H.>h.E.
|
|  FA=4016 BC=0303 DE=79B0 HL=0730 FA'=0000 BC'=0000 DE'=0000 HL'=0000
|  IX=0000 IY=006A SP=7A50 PC=68C7 IF1=00 IF2=00 R=00 I=00 IM=00
|
|                                     Flags S Z - H - PO N C
| 6000 CD546A          CALL    6A54          0 1 0 0 0 0 0 0
|
|*Modify Mode OFF
```

Notes: This shows 256 bytes of memory starting at 6000 (Hex). While in this mode you can move the cursors about using the arrow and page up/down keys. Two cursors show the position in the HEX and ASCII displays simultaneously while a numerical value showing the actual address of the cursor appears in the lower left hand corner of the screen. Immediately to the right of this is a disassembly of the code starting at the cursor location. To step forward in memory one instruction at a time press the END key. To view a different part of memory press the N for New Address.

To modify memory press M for modify. Then you can overtype any location in HEX or ASCII. Use F5 to switch to HEX modify and F6 to switch to ASCII modify mode. Use Esc to exit Modify Mode.

PC-Four DOS Commands

To set a new value in any of the Z80 registers press:

a for AF	b for BC	d for DE	h for HL
A for AF'	B for BC'	D for DE'	H for HL'
x for IX	y for IY	p for PC	s for SP

To jump into memory and start executing code type GO, execution will start with whatever states are indicated in the Z80 registers and at the address indicated by PC.

To exit DEBUG press the Esc or Q key.

DIR - Display longform directory.

Syntax: DIR [:d]

Function: Display directory with file size, date etc.

Example: DIR :0

Directory of Logical Drive :0		Physical path 0\			
Filespec	LRL	#Recs	EOF	File Size	Date
vc/cmd	256	105	3	26.25k	22-Jan-1987
tasmon4/cmd	256	30	243	7.74k	16-Jan-1987
bank/cmd	256	1	20	0.27k	30-Jan-1987
ed/cmd	256	12	255	3.00k	10-Jan-1987
ramtest/cmd	256	2	127	0.63k	10-Jan-1987
tasmonc/cmd	256	32	2	8.00k	30-Jan-1987
tasmonl/cmd	256	32	2	8.00k	30-Jan-1987
vidtest/cmd	256	1	39	0.29k	31-Jan-1987
squsq4/doc	256	23	128	5.88k	27-Jan-1987
testfile/doc	256	1	133	0.38k	26-Jan-1987
dos31a/pat	256	94	31	23.53k	31-Aug-1986
qb/inf	256	16	5	4.01k	22-Dec-1986
rand0/cmd	256	1	21	0.27k	3-Feb-1987
rand1/cmd	256	1	21	0.27k	3-Feb-1987
dodir0/cmd	256	1	21	0.27k	4-Feb-1987
ed/asm	256	117	255	29.25k	4-Feb-1987
lu4150/doc	256	158	255	39.50k	2-Jun-1987
return/cmd	256	1	9	0.26k	6-Feb-1987
ted/cmd	256	22	73	5.57k	8-Feb-1987
test/asm	256	1	255	0.25k	10-Feb-1987

Any Key --->

Notes:

- 1 In the version of PC-Four the LRL always indicates 256.
- 2 **Recs** is the total number of 256 byte records
- 3 **EOF** is points to last byte in last record, EOF is 255 if last record is completely full.
- 4 File size in k (1024 bytes) is rounded to nearest .01

DISK - redefine a TRS80 drive.

Syntax: DISK n path

Function: Equivalent to putting a floppy disk in one of your TRS80 disk drives. Redefines a TRS80 drive with a new MSDOS path (equivalent to SETUP DRIVE :n path).

Example: DISK 0 AW redefines TRS80 drive 0 to be the subdirectory AW in the current directory of the default drive.

DISK 1 B: redefines TRS80 drive 1 to be the floppy disk in drive B:.

DO - Execute a /JCL file.

Syntax: DO filename.

Function: Causes the file 'filename/JCL' to be opened and the text therein takes control and is used as if it had been type from the keyboard. Programs run from a /JCL file may accept lines from the file if they call the DOS keyboard line input service routine. Note that the DO keyword is actually redundant in this implementation and you can simply type "myfile" and DOS will search for myfile/JCL after having first searched for myfile/CMD.

Example: DO MYFILE (causes MYFILE/JCL to be run).

EXIT - Terminate PC-Four and return to PC/MS-DOS.

Syntax: EXIT

Function: Return to DOS.

Example: EXIT

HELP - get online help on DOS commands.

Syntax: HELP [command]

Function: Provides an online help function for DOS commands. If information about the command 'command' is present in the file HELP.DAT then PC-Four opens a window and displays a short form explanation of the function in question. HELP without an argument lists the available DOS commands. Press any key to remove the window and restore the screen to how it was before.

Example: HELP

```
|Current Setup:
|
|Logical drive equivalents:
|      Drive :0 --> 0\
|      Drive :1 --> 1\
|      Drive :2 --> 2\
|      Drive :3 --> 3\
|
|Color settings :      Foreground  Background
|No
|Re
|Available DOS level commands are:
|Auto
|  auto  cat   cls   copy  debug  dir   do    exit
|Show|  help  key   kill  list   load  memory print quit
|    |  reboot remove rename setup  shell  status
|> he
|
|Type: 'help command' for help on any command
|
|----- Any Key --> -----
|
|1 Help 2 Exit 3 List 4 Print 5 Copy 6 Ren 7 Debug 8 Kill 9 Cat 10 Dir
```

KEY - Turn On/Off function key display.

Syntax: KEY ON or KEY OFF

Function: Enables/Disables display of the function key prompt line at the bottom of the DOS command screen.

Example: KEY ON

KILL - Delete or Remove a file.

Syntax: KILL filename[:d]

Function: Deletes file called filename from drive d where d is 0..3.
Defaults to drive 0 if drive number is omitted.

Example: KILL EDITOR/CMD:3

LIST - Print a file to the screen.

Syntax: LIST filename[:d]

Function: Displays the designated file, line by line on the screen. You may temporarily suspend listing by pressing the space bar. Use Enter to resume listing.

Example: LIST MAILLIST/DOC:2

LOAD - Load a /CMD file to memory.

Syntax: LOAD filename[:d]

Function: Loads the file called filename on drive number d into the Z80 memory. The file must be in standard TRS-80 load module type format.

Example: LOAD TASMON4/CMD:1

MEMORY - Set/Display Memory Limits.

Syntax: MEMORY [(high=x'hhhh')]

Function: Set High Memory Limit. This value will be respected by most Model 4 Z80 programs which will not try and use any memory above this limit. Useful if you want to load utility programs to be resident at the same time as your main application program. MEMORY without an argument will return the current memory settings.

Example: MEMORY (high=x'f000') - reserves 1000 hex (4096 decimal) bytes from F000-FFFF.

Syntax: MEMORY (low=x'hhhh')

Function: Set Low Memory Limit. This value will be respected by most Model 4 Z80 programs which will not try and use any

PC-Four DOS Commands

memory below this limit. Useful if you want to load utility programs to be resident at the same time as your main application program. Note that PC-Four's DOS simulator uses memory from 0000 to 0FFFH leaving 1000H to FFFFH free. Most application programs load at 3000H or higher although some, like TRSDOS BASIC load an overlay in starting at 2600H.

MOD1000 - set keyboard in Model 1000 mode.

Syntax: MOD1000

Function: Some of the keys on the keypad of the Model 1000 generate different scan codes to a regular PC keyboard. If you are using Scripsit use the Model 1000 command to switch the keyboard translation so that the arrow keys work properly.

Example: MOD1000 enable Model 1000 keyboard.

Example: MOD1000 disable Model 1000 keyboard.

PRINT - Send a file to the Printer port.

Syntax: PRINT filespec

Function: Sends the designated file to the printer on port LPT1:. For compatibility with PC type printers carriage returns are padded with a line feed (0DH --> 0DH 0AH). The file is sent through the standard DOS printer handler so if you have a printer spool buffer then the file will go through that automatically.

Example: PRINT RECIPES/DOC:2 - prints the file called RECIPES/DOC on logical drive number 2 and directs the output to LPT1:.

REBOOT - Reinitialize PC-Four's DOS.

Syntax: REBOOT

Function: Reinitializes DOS and low memory space. The BOOT file PC4BOOT/DAT must be present on you default drive for this.

RENAME - Rename a file.

Syntax: RENAME oldname/ext[:d] newname/ext

Function: Renames file named "oldname/ext" on drive d (0 if drive number is omitted) to newname/ext.

Example: RENAME MYFILE/DOC:2 YOURFILE/TXT

renames the file called MYFILE/DOC on drive number 2 to have the name YOURFILE/TXT.

SETUP - Show current Setup parameters.

Syntax: SETUP

Function: Display current SETUP parameters.

Example:

> SETUP

Current Setup:

Logical drive equivalents:

Drive :0 --> 0\
Drive :1 --> 1\
Drive :2 --> 2\
Drive :3 --> 3\

Color settings :	Foreground	Background
Normal video :	7	0
Reverse video:	0	7

Auto command: 'CAT :0'

Show greeting: No

SETUP COLOR - Set up Normal and Inverse video colors.

Syntax: SETUP COLOR

Function: Allows you define your own foreground and background colors for both normal and inverse video. Thus four different colors can be displayed simultaneously in a regular TRS-80 type program. COLOR will give you a menu of colors to choose from for each attribute. The colors can also be predefined in the auto start up file PC4CFG.DAT

SETUP DRIVE - reassign drive logical path.

Syntax: SETUP DRIVE :D dos-filepath

Function: Reassigns the filepath associated with a TRS-80 logical drive number.

Example: SETUP DRIVE :3

SETUP KEY - redefine meaning of runtime function keys.

Syntax: SETUP KEY n 0xxH

Function: Redefines the hex code generated by function key Fn to be the value in the argument 0xxH where xx is a hexadecimal number from 00 to FF (0 to 255 decimal). Useful for programs that require a special code generated by a combination of TRS80 keyboard keys (e.g. Clear Shift =). Keys F1-F3 are preprogrammed to generate the codes 81, 82 and 83H but can be reprogrammed if you wish however any TRS80 program that uses keys F1-F3 will need the standard values.

Example: SETUP KEY 7 01DH redefines key F7 to generate 1DH so that when you are running MSCRIPT you can generate the code equivalent to SHIFT-ENTER.

Example: SETUP KEY 6 0BDH redefines key F6 to generate BDH needed as the exit key (clear shift =) for the programs TED from in LSDOS 6.3.

Example: SETUP KEY Displays a table of the current settings for the function keys. Note that two columns are displayed. One is the function of the keys in DOS mode while the other is the hex code generated when running a TRS80 program.

SETUP SAVE - save the current setup.

Syntax: SETUP SAVE

Function: Saves the current drive path settings, color assignments and other information in the auto start up file PC4CFG.DAT

Example: SETUP SAVE

QUIT - Exit to DOS. (same as EXIT)

Syntax: QUIT

Function: Return to DOS.

SHELL - Execute an MS/PC-DOS command.

Syntax: SHELL [MS-DOS command]

Function: Allows you to execute a DOS command from within PC-Four. For instance you could do a DOS directory, format a floppy disk or copy a file. This function generally requires extra memory as PC-Four remains resident while DOS loads an additional copy of COMMAND.COM and any programs that need to be run to execute the command you typed. Note: if you execute SHELL without an argument you will enter DOS and can run any sequence of commands and programs before returning. What you can do is limited only by your free memory. To return to PC-Four type EXIT and press the Return Key.

Note: Calling this from versions of MS or PC-DOS earlier than 3.0 may result in you being unable to return to DOS when you finally leave PC-Four. You will have to reboot your computer using Ctrl-Alt-Del.

Example: SHELL dir c:\utilities*.bas

STATUS - Display status information on PC-Four.

Syntax: STATUS

Function: Causes PC4 to print out the status of some PC-Four parameters and the current settings of the simulators Z80 registers.

Example:

```
> STATUS
```

```
V1.1 06/12/87
```

```
Emulator code space starts at 23D6:0000
```

```
Z80 registers at 1B0A:5AC8
```

```
Bank 1 at 33D6:0000
```

```
Bank 2 at 3BD6:0000
```

```
Z80 register status:
```

```
FA=0000 BC=0000 DE=0000 HL=0000 FA'=0000 BC'=0000 DE'=0000 HL'=0000
```

```
IX=0000 IY=0000 SP=0000 PC=0000 IF1=00 IF2=00 R=00 I=00 IM=00
```

```
Flags S Z - H - PO N C
```

```
0 0 0 0 0 0 0 0
```

```
>
```

The above information can be useful to you if you wish to inspect memory direct from MS-DOS's DEBUG. Addresses give are the absolute segment:code start of the items named.

ZERO - create a zero length file

Syntax: ZERO filename

Function: create a new file which is 0 bytes long. Useful for programs like PROFILE which must have all their files present even if they are empty. Note that the MSDOS COPY command will not copy a zero sized file.

Example: ZERO MAIL000/DAT:1 creates a file MAIL000/DAT on drive 1 with a length of 0 bytes.

64K - mark extra 64K bank as in use (disable it).

Syntax: 64K

Function: sets flags in DOS to indicate that the additional 64K memory bank is already in use by some other program. That means that when you run a program like Multiplan 1.7 that uses the extra 64K you can lock the extra memory out to force Multiplan to use only 64K and hence run much faster. Use 128K command to re-enable the extra 64K bank.

Example: 64K

128K - mark extra 64K bank as free (enable it).

Syntax: 128K

Function: sets flags in DOS to indicate that the additional 64K memory bank is free and can be used by any new program. That means that when you run a program like Multiplan 1.7 that uses the extra 64K it will test the flags and, finding the memory free will use it thus running very very slowly indeed. Use 64K command to disable memory.

Example: 128K

5 - Running TRS-80 Programs.

PC-Four will run many programs originally designed for the TRS-80 running TRS-DOS 6.x. They must be copied exactly from the TRS-80 disk format to MS-DOS. Any errors in the file or the file size may cause a failure. If you are having trouble, check that all the files necessary are present and that they are exactly the same size as they were when they were on a TRS-DOS disk.

An important point: Some TRS-80 programs create zero length files which MUST be present for the program to run properly. You cannot use the MS-DOS COPY command to copy a zero length file, so to move it from one disk to another, you must recreate it on the new disk rather than copy it. A new ZERO command is now in PC-Four to allow you to create zero length files. Note that the PC-Four COPY command CAN copy zero sized files from one "TRS-80 drive" to another.

An important difference between TRS-80 text files and MS-DOS files is that the end of each line on a TRS-80 is marked with an ODH (carriage return) and on a PC a ODH and an OAH (carriage return, line feed) is used. A printer attached to a PC expects to see a ODH and OAH when a normal text file is printed. PC-Four pads the output of normal TRS-80 type programs with an OA every time it sees an ODH. This works for most situations but if you are using a word processor to print proportionally or to send escape codes then you must use the CR OFF command in PC-Four to suppress automatic addition of line feeds. If, unlike Allwrite, your word processor can not be programmed to automatically send line feeds when needed then you will have to set your printer to work in TRS-80 mode, that is to expect only an ODH at the end of each line. Use the CR ON command to restore line feed addition. See the example in Allwrite below.

Don't forget that you can use /JCL files to make your life easier, especially if you have a hard disk. For example, you can put all your Allwrite files in a subdirectory called "AW" and, if you create a AW/JCL file such as:

```
DISK 0 AW
CR OFF
CAT
```

you can, simply by typing "AW" cause the execution of the above file which will substitute your existing "drive 0" with the Allwrite subdirectory, it will turn off the automatic line feed addition, and it will show you a directory. You might create another /JCL file called RESTORE/JCL which will turn the CR ON and return your original drive 0.

A suggestion: put all your /JCL and utility files in drive :3 so PC-Four will always find them no matter what drives you have temporarily selected for :0 an :1.

The current version of PC-Four uses only text display mode on your PC and, for the graphics characters, makes the nearest approximation using the more limited selection of block graphic characters available on the PC. This is not normally important as graphics is generally only used on opening screens of a few programs.

5.1 Special Keyboard Keys:

Some programs, for example Allwrite, ask you to press key combinations to achieve certain special functions. A problem can arise where the manual for the TRS-80 program asks you to press say 'Shift-2', that is the 'Shift' key and the '2' key together. On the TRS-80 this would give the " character whereas on a PC it gives the @ character. What your program really wants is the " character so you must find it wherever it is on the PC keyboard. In this example you will find the " on the PC together with the ' on the key on right of the keyboard next to the Return key.

The main part of the TRS-80 keyboard looks like this:

!	"	#	\$	%	&	'	()	*	=	
1	2	3	4	5	6	7	8	9	0	:	- Break
Q	W	E	R	T	Y	U	I	O	P		
q	w	e	r	t	y	u	i	o	p	@	
A	S	D	F	G	H	J	K	L	+		
a	s	d	f	g	h	j	k	l	;	Enter	Clear
	Z	X	C	V	B	N	M	<	>	?	
Shift	z	x	c	v	b	n	m	,	.	/	Shift
Ctrl					Space Bar					Caps	

Note the special symbols that are not on the same keys or in the same location on your PC keyboard. Particularly note the " & * (and) keys on the top row. If something in your program does not respond to a particular key combination check the manual and see what they are really asking you to do. It may be that they are, like Allwrite, telling you to press Shift and a key to get some character which is not in the same place on the PC keyboard.

5.2 Specific TRS-80 Program Information.

The next section gives details of some programs that will run on your PC under PC-Four and what you must do to ensure success.

ALDS - Assembly Language Development System (Tandy #26-2012).

ALDS can be used on the PC under PC-Four to develop, test and debug TRS-80 Model 4 and other Z80 assembly language programs. You need the following files:

alasm/cmd	albug/cmd	aledit/cmd	allink/cmd	altran/cmd
-----------	-----------	------------	------------	------------

main/rel main/src mod4/rel mod4/src

and any other /src files you may have from your TRS-80 days. There may be some limitations in debug operations because the TRS-DOS 6.2 debug does not exist in PC-Four. We suggest TASMOM as one of the best debuggers available for the TRS-80.

Allwrite (PROSOFT):

Copy the following files to your MSDOS Allwrite drive:

al/cmd	al/def	alf/cmd	alf/def
alindex/cmd	alinstal/cmd	help/al	prtables/prt

Once you have run the install program you will also have /def and /tab printer definition files such as:

ml92/def	ml92/tab	rx80/def	rx80/tab
----------	----------	----------	----------

Installation: run ALINSTAL and answer questions as below. Select the default for questions not listed.

Q: Selection of printer

A: Enter number for your printer

Q: How is this printer connected to the computer:

A: 3. User or DOS supplied printer driver.

Q: Must a Line Feed be added to each Carriage Return while printing?

A: YES

Q: CURSOR SHAPE: specify as a number:

A: 95 (this is underscore character _)

Q: ENTER SYMBOL SHAPE: specify as a number:

A: 140

Q: BLINKING CURSOR "ON" time (1-255):

A: 1

Q: BLINKING CURSOR "OFF" time: (1-255):

A: 1

Q: Do you want to use our keyboard driver (YES or NO):

A: NO

Q: Select your DOS by number. No default:

A: 5 (5 is TRSDOS)

Running TRS-80 Programs

Running Allwrite: First, from PC-Four DOS type CR OFF to suppress automatic addition of line feeds on carriage returns. This is important since control codes for proportional printing, font selection and general printer control may contain the ODH code which would otherwise get converted to ODH 0AH. Allwrite will add carriage returns where necessary because you said YES in the install program.

Special Allwrite keys: Use Home for Shift-Left, End for Shift-Right, Pg Up for Shift-Up and Pg Dn for Shift-Down. Use the Alt key in place of the TRS-80 Clear key. For two key combinations press the Alt key first, then release it before pressing the 2nd key. Note that some Allwrite control functions use keys such as CLR-Shift-2 and CLR-Shift-6 which are not the same on the PC keyboard.

For CLR-Shift-2 use ALT shift-'
 CLR-Shift-6 use ALT shift-7
 CLR-Shift-7 use ALT '
 CLR-Shift-8 use ALT shift-9
 CLR-Shift-9 use ALT shift-0

You must release the ALT key before pressing the 2nd key.

Use the Esc key to bring in the command line at the bottom of the screen.

Note that the DIR command in Allwrite does not work because they are looking for the DIR/SYS file which does not exist in PC-Four.

BASIC (Model 4 TRSDOS 6.2).

Model 4 Basic from TRSDOS 6.2 runs well on the PC under PC-Four. To use BASIC you will have to copy these files:

basic/cmd basic/ov1

Notes: basic/ov1 must be on drive :0

Both basic files are invisible and protected on the original TRS-DOS disk. You must remove the password and make the file visible to copy them. The update password for BASIC/CMD and BASIC/OV1 is BASIC. In TRS-DOS 6.2 type:

ATTRIB BASIC/CMD.BASIC (VIS) to make the file visible

ATTRIB BASIC/CMD.BASIC (USER=,OWNER=) to remove passwords.

Programs that use graphics will not look exactly the same as the PC graphic set does not match the TRS-80 graphics. This version of PC-Four makes the nearest approximation.

BASIC (Model 4 DOS-Plus 4.0).

Running TRS-80 Programs

You can use the Basic from DOS-Plus 4.0 on PC-Four. The enhanced version requires the following files:

basic/cmd basic/ov1 ref/cmd resolve/cmd sort/cmd sr/cmd

Note: BASIC/OV1 must be on drive :0 You should also have the file DPFIX/CMD (you will find this in subdirectory 0 on your original PC-Four disk).

The enhancements BE1/CMD and BE2/CMD must be installed on DOS-Plus. The APPEND (CMD) command is need for installation and this is not available in PC-Four. To run DOS+ Basic with the enhancements installed you must first run the program DPFIX/CMD. This temporarily patches PC-Four. If you do not run this first, enhanced DOS+ BASIC will hang.

BASIC (Model 4 LS-DOS 6.3).

You can use the enhanced Model 4 Basic from LS-DOS 6.3 if you copy these files:

basic/cmd basic/ov1 basic/ov2

BASCOM (Microsoft Model 4 Basic Compiler)

BASCOM runs under PC-Four and gives a factor of 5-10 speedup over interpreted BASIC. You need the following files:

BASCOM/CMD	BASCOM/REL	BASRUN/CMD	BASRUN/REL
BCLOAD/L80	L80/CMD		

BEEP (TRS-DOS 6.2 BASIC Enhancement Package).

You can use the enhanced 6.2 BASIC if it had been previously enhanced on the Model 4 using BEEP. You can also do the enhancement on PC-Four if you have the files INSTALLB/CMD and BE/LMF. However the install program expects BASIC to have a password and must be modified before you can use it. To patch INSTALLB change:

Record 0 byte 86H from 4C to 46
Record 1 byte 55H from 2E 42 41 to 3A 20 03

C Compiler (Alcor).

Running TRS-80 Programs

Limited tests we have run indicate that Alcor's C Compiler (The same one was also sold by Radio Shack, Cat # 26-2230) runs fine. You need to transfer most of the following files:

cc/cmd	cc/ov1	cc/ov2	cc/ov3	cc/ov4
clib/obj	ccb/cmd	cerrors/dat	clib/c	clib/obj
cmd/hlp	codegen/cmd	codeinit/dat	csupport/bin	edit/cmd
help/hlp	hex2bin/cmd	key/hlp	linkload/cmd	optimize/cmd
printf/c	printf/obj	random/obj	runc/cmd	runc/obj
scanf/c	scanf/obj	setedit/cmd	setup/edt	stdio/
strings/obj	system/obj	trslib/obj		

CAU Macro Assembler (MZAL).

The Macro Assembler and Editor supplied with DOS-Plus 4 runs well under PC-Four. You need the following files:

TED/CMD	the text editor.
MASM/CMD	the macro assembler.
DISASM/CMD	the disassembler.

Communication Programs

PC-Four supports the RS232 communication port of the TRS-80 and programs that use it. Some public domain communication programs we have tested and successfully transferred files between a true TRS-80 and a PC running PC-Four include:

fastterm/doc xmodem4/cmd xt4/cmd

PC-Four uses an interrupt driven RS232 receiver in MS-DOS which passes bytes received to a buffer and sets an interrupt in the emulator. Most TRS-80 communications programs set up a receive buffer and enable the receive interrupt. There may be some speed limitations depending on the performance of the PC that you are using.

DISnDATA.

DISnDATA, an excellent Z80 disassembler from Pro-Am software runs on PC-Four. All you need is the file DISNDATA/CMD.

EDAS (Misosys).

EDAS and its editor SAID can be used to generate Model 4 and other Z80 assembly language programs on the PC under PC-Four. You will need most of the following files:

edas/cmd	mas/cmd	med/cmd	said/cmd	xref/cmd
----------	---------	---------	----------	----------

HARTFORTH (Molimerx/Hypersoft).

HARTFORTH, an excellent implementation of the Forth-79 standard for the TRS-80 Model 4 is available from Hypersoft under license from Molimerx UK. This consists of one file FORTH/CMD. Once this is transferred to MS-DOS you can run it under PC-Four with very few restrictions. The disk sector editor cannot be used because you do not have a TRS-80 disk structure.

LESCRIPT (Anitek)

Does not run yet. V 1.7 runs through the greeting screen to the editor screen and then hangs. We are working on this.

LITTLE BROTHER (Logical Systems)

You can run the Little Brother Data Base program from Logical Systems Inc. if you transfer over the following files:

lb/cmd	lbstart/cmd	lbhelp/hlb	lbstart/cmd	
lb0/ovr	lb3/ovr	lb4/ovr	lb5/ovr	lb6/ovr
lb6a/ovr	lb6b/ovr	lb7/ovr	lb8/ovr	lb9/ovr

Treat Little Brother as if you were running on a hard disk system, even if you are not. Do not try and use the install program.

MULTIPLAN

For Multiplan you will need

mp/cmd	mp/ovr	mp/l	mp/s
--------	--------	------	------

Running TRS-80 Programs

and any /mp user files that you may have from using Multiplan on your TRS-80.

Note that only the 64K version of Multiplan 1.7 will run satisfactorily on PC-Four. This is because the 128K version does too many memory swaps and runs too slowly. To make sure you use the 64K version use the PC4 "64K" command before running Multiplan.

PATCH (TRS-DOS 6.2).

PATCH/CMD is a useful DOS utility that allows you to make small fixes to TRS-80 files. PATCH is password protected, see the section on transferring files for details on this. Once transferred to MS-DOS you can use commands of the following forms:

- 1) PATCH filename (Drr,bb=hh hh hh:Frr,bb=hh hh hh)
- 2) PATCH filename (X'aaaa'=hh hh hh hh)
- 3) PATCH filename (Y)

rr is a record #, bb a byte #, hh is a hex number 00-FF, aaaa a hex address

Y is to yank a patch of type 2 above.

You can combine a number of separate PATCH commands into a /JCL file.

PASCAL (Alcor or TANDY #26-2212).

Pascal will run on the PC under PC-Four. You need to copy all the files from the original Pascal distribution disks to MS-DOS. This includes at least

pascal/cmd	pascal/ov1	pascal/ov2	pascal/ov3	pascal/ov4
codegen/cmd	codeinit/dat	edit/cmd	linkload/cmd	optimize/cmd
setedit/cmd	strings/obj	trslib/obj		

PAYROLL (Tandy).

PC-Four will run COBOL generated business application programs such as the Payroll program from Tandy. To use this you need to transfer all the /cob and /dta files together with:

input/cmd	pr4/cmd	runcobol/cmd	upgrade/cmd
-----------	---------	--------------	-------------

PowerMAIL Plus (PowerSOFT).

POWERMAIL is a very powerful mailing system and you can run the Model 4 version on your PC if you have the following files transferred:

PMAIL/CMD	PMAILA/M4	PMAILB/M4	PMAILC/M4	PMAILD/M4
PMAILE/M4	PMAILM/M4	PMAILS/M4	TMERGE/CMD	

TMERGE/CMD is the Text Merge utility that comes with Powermail.

PFS FILE (Software Publishing Corp).

For PFS file you need the files PFS/CMD and FILE/CMD plus any user data files you may have generated on the TRS-80. Use the PC F1-F3 keys as needed to run the program just as you used the TRS-80 F1-F3 keys.

PROFILE 4 Plus (Radio Shack).

To run Model 4 Profile you will need the following files:

cm/cmd	efc1/cmd	efc2/cmd	efc3/cmd
efc4/cmd	efc5/cmd	efc6/cmd	efc7/cmd
efc8/cmd	efc9/cmd	efca/cmd	efcb/cmd
efcc/cmd	efcd/cmd	efce/cmd	efcf/cmd
efcm/cmd	rm/cmd		

You will also need your data files. For example:

mail0000/da2	mail0000/da3	mail0000/dat	mail0000/key
mail0000/map	mail0000/mth	mail0000/olm	mail0000/pml
mail0000/pr1	mail0000/sl1	mail0000/vc1	

These files may be on a different drive number. Note that Profile creates some zero length files for unused data segments. If any of these are missing Profile will not run properly. Most file transfer programs will not copy a zero length file so you must artificially re-create it on the PC.

To create a missing zero length file you can use the special PC-Four file creation command 'ZERO'. This is new with version 1.5 of PC-Four and allows you to create a zero length file by typing:

ZERO filename[:D]

For example:

ZERO MAIL0000/DA2:1

creates a file MAIL0000/DA2 zero bytes long on drive :1.

SuperScripsit (Radio Shack)

Scripsit needs the following files:

scripsit/cmd	scripsit/ctl	errors/ctl	help/ctl
scr16/ctl	scr17/ctl	scr18/ctl	scr19/ctl
scr32/ctl	scr33/ctl	scr35/ctl	scr38/ctl
scr50/ctl	system/ctl		

You will also need one or more printer driver files:

dmp2100/ctl	dmp400/ctl	dw2/ctl	dwp210/ctl
dwp410/ctl	lp4/ctl	lp8/ctl	

If you have trouble with proportional printing you may have to turn off the auto line feed with CR OFF and set your printer in TRS-80 mode, that is so that it expects only an ODH at the end of each text line.

An Error in the /CTL files EOF.

SuperScripsit has appeared in many versions since it was first introduced. So far we have seen versions from 1.00.00 through 1.02.00. All have one problem in common: every printer /CTL file has an incorrect EOF byte. If you type DIR you might see the entry for say DMP2100/CTL which indicates that it takes 4 records (a record is 256 bytes) and that the EOF is 89 - i.e. the last byte is the 89th byte of the 4th record. THIS IS WRONG. The true end of file is 135 (for this particular driver).

Normally this does not matter; by a stroke of luck TRSDOS and Scripsit work, even if you copy the file from one disk to another. HOWEVER - if you copy the above file from TRSDOS to MS-DOS using Hypercross, PC Cross-Zap, TRSCross or even a modem you will only get up to the 89th byte of the last record and the rest will be lost. When Scripsit sees a bad printer file, it doesn't tell you it's bad it says instead (for example): "Cannot find DMP2100/CTL".

To fix this problem you need to edit the EOF byte in the directory before you copy the file from TRSDOS to MS-DOS. On TRSDOS a disk sector editor such as Hyperzap, LSFED or Superutility can be used. On MS-DOS you can use PC Cross-Zap. We give here specific instructions for PC Cross-Zap.

Run PC Cross-Zap and select the appropriate TRS-DOS format (P for single sided, Q for double). Then insert a backup copy of your TRS-80 Scripsit disk in drive A and press 6 to select the Disk Navigator menu. Press 2 to view/edit the directory sectors and you will see the first one displayed. On the right you will see names of files. Press the Page Up key (PgUp) to step through the directory

Running TRS-80 Programs

until you see the name of the entry you want to correct. You will see something like:

```
0040  10 07 DB 5A 00 44 4D 50 32 31 30 30 20 43 54 4C  ...Z.DMP2100 CTL
0050  96 42 96 42 04 00 24 40 FF FF FF FF FF FF FF FF  .B.B..$@.....
```

The underlined entry (4th in) is the EOF byte. Using the arrow keys move the cursor until it is on the first character (5). Press M to enter the modify mode and then type the replacement EOF - in this case 88 - you should then see:

```
0040  10 07 DB 88 00 44 4D 50 32 31 30 30 20 43 54 4C  ...Z.DMP2100 CTL
0050  96 42 96 42 04 00 24 40 FF FF FF FF FF FF FF FF  .B.B..$@.....
```

Press F2 to exit the Modify Mode and then press Z (make sure there is no write protect tab on the disk) to rewrite the modified directory sector. Change any other /CTL files as appropriate and then select the copy menu and transfer the files. Below is a table of the correct EOFs for each /CTL file.

Printer Driver	Incorrect EOF (HEX)	Change to (HEX)
DMP400/CTL	49	90
DMP2100/CTL	5A	88
DW2/CTL	66	54
DWP210/CTL	11	43
DWP410/CTL	11	43
LP4	FA	2B
LP8	21	53

Scipsit Versions.

Of the versions we have tested, not all work as well using PC-Four. This is mainly due to different keyboard routines used in each version. Versions 1.00.00 - 1.00.09 work reasonably well but are a little slow. Versions 1.01.00 - 1.01.03 have a driver which is very slow and may lock up your PC when you try to quit Scipsit. Version 1.02.00 is the latest and seems to perform the best. We suggest you either use this or go back to one before 1.01.00.

Sources of Printer Drivers for Scipsit.

Since you may be using your PC without the original Radio Shack printer, here are some possibilities for sources of printer drivers.

1) Compuserve and GENie. These public access systems have large libraries of public domain software for the TRS-80 including some Scipsit printer drivers.

2) Powersoft Products, 17060 Dallas Parkway, Suite 114, Dallas, TX 72548. Powersoft has a driver called PowerDriver Plus which allows EPSON and PC printers to be used.

3) ALPS, 23 Angus Road, Warren NJ 07060, (201) 647-7230 advertised as recently as Jan 1987 that they had drivers to support over 200 different printers.

Visicalc. (Software Arts)

All you need is vc/cmd and your own, previously created /vc user files. Model 4 Visicalc is inherently slow and it is even slower under PC-Four. Turn off the automatic recalculation mode using the /GRM (Global Recalc Manual) command. This will avoid a time consuming delay each time you make a new cell entry. Type ! to force a recalculation only when you are ready.

ZEUS Editor/Assembler. (Cosmopolitan Electronics)

With the Model 4 version of ZEUS you can edit and assemble your own Z80 machine language programs. Zeus supports all the undocumented Z80 op-codes as does PC-Four Version 2. To use ZEUS all you need is the ZEUS/CMD file. You may also wish to copy over Z80E/ASM to verify that ZEUS assembles to the entire Z80 instruction set.

6 - Technical Information

This section provides sufficient technical information for you to write your own programs to interface with PC-Four's DOS simulator. PC-Four emulates all the officially documented instructions of the Z80 so any program written in true Z80 code should run. The only practical problem is interfacing with the outside world. This is done by using DOS function calls. On a real TRS-80 it is possible to do all your I/O through hardware Port Input and Output commands and using memory mapped output to the screen memory. This is implemented only to a limited extent here. See the section on Port I/O.

6.1 DOS Function Calls.

On the Model 4, application programs interface with the real world via DOS calls which use the Z80 RST 28 instruction. This instruction is like a subroutine call except that it jumps to a fixed address at 0028H. The TRS-80 Model 4 DOS is set up so that a call to this address causes one of a number of special system functions to be executed. The function executed depends on the value in the A register when the call is made.

PC-Four emulates the TRS-80 DOS by trapping the RST-28 call and jumping to special 8086 type routines which perform the same function on the PC but make it look to the Z80 as if they had been executed in a TRS-80 environment. To use, load the Z80 accumulator A register with the number for the operation desired, load other registers as needed and issue a RST 28 command.

Example: print a "*" character on the screen

```
LD      C,"*"   ; put * value in C
LD      A,02    ; set up call # 02
RST     28H     ; do it
```

Detailed functional description of implemented calls

Note: all unimplemented calls will cause a simulation abort and return to DOS.

Call # 00 Reboot System

Call # 01 Wait for keyboard character, put in A
Enter with A=01H, Returns: A = key typed, Z set

Call # 02 Display the character in C
Enter with A=02H and C = character to be displayed.
The function will display the character at the current cursor location and then advance the cursor.

Call # 03 Get one byte from file or device
Enter with A=03H and DE --> DCB or FCB
Returns: Success - Z set, A=byte, Fail: NZ set A=error #

Call # 04 Put one byte to file or device
Enter with A=04H and DE --> FCB, C = byte to output
Returns: Success - Z set, Fail: NZ set A=error #

Call # 05 Device control request
Enter with A=05H and DE --> DCB, vector in DCB at DCB+1
C = 0 get status, 1, send break, 2 init driver
3 reset all buffers, 4 spec wakeup vector, 8 next char preview

Implemented in this version of PC4:

KI 03 clear typeahead, FF copy keyboard map to (IY)

DO none

PR if C=0 get status else Returns: NZ, A=0

Status returned according to the following

Printer Status	PC LPT1:	TRS80 Port	J1 Pin #	Printer Pin #
not busy	bit7=1	bit7=0	21	11=0
acknowledge	bit6=1	-	-	10=0
out of paper	bit5=1	bit6=1	23	12=1
ptr selected	bit4=1	bit5=1	25	-
i/o error	bit3=1	bit4=0	28	32=0
timed out	bit0=1	-	-	-

Call # 06 Out to printer the character in C
Enter with A=06H and C = character to be printed

Call # 07 Determine current Program Counter state.
Enter with A=07H, returns HL=location after RST 28H.

Call # 08 Scan keyboard and return
Enter with A=08H, Returns: A=key depressed, Z set

Technical Information

If no key pressed, A=0, NZ.

Call # 09 Get a line of input from keyboard

Enter with A=H and HL --> user buffer of length B+1

Returns: HL --> start, B=chars, Z if OK, Carry set if Break. DE and C may be altered

Call # 10 Display a message line at <HL>

Enter with A=0AH and HL --> message string terminated in 0D or 03. AF, DE altered

Call # 11 Issue log message - Does nothing

Call # 12 Display & log message - Displays only

Call # 13 Message line handler - Not Implemented

Call # 14 Print line of text

Enter with A=0EH and HL --> string ending in 03 or 0dh

Call # 15 Video control

Enter with A=0FH and

- B=1 get char in A at row H, column L preserving cursor
- B=2 display character C at row H, column L
- B=3 move cursor to row H, column L
- B=4 get cursor position into HL, H=row L=column
- B=5 move 2048 bytes of data from (HL) to video memory
- B=6 move 2048 byte block from video mem to (HL)
- B=7 scroll protect top part of screen (max 7 lines)
- B=8 change cursor character

Call # 17 Parse parameter string

Enter with A=11H, DE -> parameter table, HL -> string

Returns: NZ - if bad parameters in string.

Call # 18 Get date at HL > string MM/DD/YY

Enter with A=12H, Returns: current date indicated by PC real time clock

Call # 19 Get time at HL > string HH:MM:SS

Enter with A=13H, Returns: current time indicated by PC real time clock

Call # 20 CHNIO - pass control to next device - Not Implemented

Call # 21 Abort to DOS - put FFFF in HL

Call # 22 Return to DOS

Enter with A=16H, Exits Emulator and returns to PC-Four's DOS.

Call # 24 Return to DOS Command Interpreter.

Enter with A=18H, HL --> string to be executed by DOS command interpreter.

Technical Information

Executes command, does not return to calling program. String must end with 0DH and must be a maximum of 80 characters long.

Call # 25 Execute DOS command, return to user.

Enter with A=19H, HL --> string to be executed by DOS command interpreter.
Like call # 24 except that it returns to calling program.

Call # 26 Post an error message

Enter with A=1AH and C = error #

if bit 6=1 for long message

if bit 7=0 then abort to DOS

if bit 6 of CFlag =1, copy message to buffer at DE "

Call # 27 Enter DEBUG - Not Implemented

Call # 28 Test if task slot available

Enter with A=1CH, C=task # (8 or 11 only)

Returns Z if slot available, else NZ

Task slots other than 8 and 11 will always return NZ

Call # 29 Add a task

Enter with A=1DH, DE -> Task Control Block, C=Task # (8 or 11 only)

Returns nothing.

Call # 30 Remove a task.

Enter with A=1EH, C=Task # (8 or 11 only)

Returns nothing.

Call # 31 Replace current task.

Enter with A=1FH

Replaces current task vector with address following the call and then returns to foreground.

Call # 32 Kill current task

Enter with A=20H

Kills task and returns to foreground.

Call # 33 Check if Drive available

On the TRS-80 this tests if a given disk drive is available. On PC-Four, this function always returns drive ready (Z=1). Enter with A=21H.

Call # 34 Show directory on display or put in RAM buffer

Enter with A=22H, HL -> buffer to receive info, C=drive #

if B=0 dir display to screen 5 across

if B=1 dir to buffer at HL

if B=2 dir to screen, files matching partspec at HL

Not implemented

if B=3 dir to buffer1, files matching partspec at HL

Not implemented

if B=4 get disk name, capacity and free space at HL

Not implemented

Call # 35 get directory record or free space to a ram buffer

Enter with A=23H, HL --> buffer to receive info, B=drive #

Technical Information

if C=0 get dir records of all files
if C=1-254 get specific record.
if C=255 get free space.

Calls 36-39 undefined.

Call # 40 DCSTAT test if drive assigned

Enter with C=drive # 0-3

Returns Z if C=0..3, else NZ

Call # 41	SLCT	select a new drive	- Not Implemented
Call # 42	DCINIT	init FDC	- Not Implemented
Call # 43	DCRES	reset fdc	- Not Implemented
Call # 44	RSTOR	fdc restore	- Not Implemented
Call # 45	STEPI	fdc step in	- Not Implemented
Call # 46	SEEK	fdc seek track	- Not Implemented
Call # 47	RSLCT	is drive busy	- Not Implemented
Call # 48	RDHDR	read sector header	- Not Implemented
Call # 49	RDSEC	read sector	- Not Implemented
Call # 50	VRSEC	verify sector	- Not Implemented
Call # 51	RDTRK	read track	- Not Implemented
Call # 52	HDFMT	hard disk format	- Not Implemented
Call # 53	WRSEC	write sector	- Not Implemented
Call # 54	WRSSC	write system sector	- Not Implemented
Call # 55	WRTRK	write track	- Not Implemented

Call # 56 Rename a File

Enter with A = 38H,

DE --> old name in FCB buffer, must be closed

HL --> new name terminated with 03 or 0D

Returns Z OK, NZ with error # in A

Call # 57 Delete file or remove device

Enter with A=39H

fcf must be open DE > fcf, Returns: Z if OK

Call # 58 Open or initialize file or device

Enter with A=3AH, HL --> 256 byte I/O buffer, DE --> FCB,
B=LRL

Returns: Success - Z set, C set if new file,

Fail: NZ set A=error #

Only AF altered by this call

Call # 59 Open existing file or device

Enter with A=3BH, HL --> 256 byte I/O buffer, DE --> FCB,
B=LRL

Returns: Success - Z set, Fail: NZ set A=error #

AF altered by this call

Call # 60 Close file or device

Enter with A=3CH, DE --> FCB of file/dev to close

Returns: Success Z set, filespec restored to FCB

Fail NZ set, A=error #

Technical Information

Call # 61 (fbksp) backspace file ptr 1 logical record
Enter with A=3DH, DE --> FCB
Returns Z if OK, NZ else.

Call # 62 Check for end of file
Enter with A=3EH and DE --> FCB,
Returns: NZ and A=1C if at end, A=1D if past end

Call # 63 Return pointer to current logical record #
Enter with A=3FH and DE --> FCB. returns BC=record #

Call # 64 Get last logical record # of a file
Enter with A=40H and DE --> FCB. returns BC=record #

Call # 65 Position file pointer to end of file
Enter with A=41H and DE --> FCB.

Call # 66 Position file pointer to a specific record #
Enter with A=42H and DE --> FCB, BC=logical rec #,
Returns: Z or A=1C or A=1D if OK

Call # 67 Read record from file
Enter with A=43H and DE --> FCB for file,
HL --> UREC if LRL<>256

Call # 68 Rewind pointer to start of file.
Enter with A=44H and DE --> FCB.

Call # 69 Reread current sector
Enter with A=45H, DE --> FCB
Refreshes current 256 byte sector buffer.
Exit Z if OK.

Call # 70 Rewrite current sector
Enter with A=46H, DE --> FCB
Rewrites current 256 byte sector buffer.
Exit Z if OK

Call # 71 Seek cylinder,sector - Not Implemented

Call # 72 Skip next record. - Not Implemented.

Call # 73 Write record to file with verify.
Enter with A=49H
PC-Four treats this the same as # 75. If you want
to insure write verification put a VERIFY ON
statement in your PC/MS DOS AUTOEXEC.BAT

Call # 74 Write End Of File - Not Implemented yet ***

Call # 75 Write record to file
Enter with A=4BH and DE --> to FCB for file
HL --> UREC if LRL<>256

Technical Information

Call # 76 Load file to memory
Enter with A=4CH and DE --> FCB of file to load
Returns: Success - Z set, Fail: NZ set A=error #

Call # 77 Load and execute file
Enter with A=4DH and DE --> FCB of file to run
Returns: Success - Z set, Fail: NZ set A=error #

Call # 78 Moves File spec from buffer to FCB
Enter with A=4EH and HL --> buffer, DE --> FCB
Returns: if Z, HL --> last char, DE --> start of FCB
if NZ, HL --> invalid char, DE --> start FCB,
A=invalid char

Call # 79 Set default extension into FCB if none present
Enter with A=4FH and DE > FCB, HL > default
AF and BC altered

Call # 80 (fname) Fetch file/ext from dir - Not Implemented

Call # 81 GTDCT get drive code table address
Enter with A=51H, C=logical drive #
Returns IY=pointer to entry for drive

Call # 82 Find Device Control Block
Enter with A=52H and DE=DCB. If DE=0 find 1st free DCB

Call # 83 GTMOD find memory module address
Enter with A=53H
Returns NZ, A=08, device not available error

Call # 84 undefined

Call # 85 RDSSC read a system sector
- Not Implemented

Call # 86 GATRD - not implemented

Call # 87 DIRRD read directory record
- Not Implemented

Call # 88 DIRWR write directory record
- Not Implemented

Call # 89 undefined

Call # 90 8 bit unsigned multiply.
Enter with A=5AH
Multiplies C by E, result in A, DE altered

Call # 91 16*8 bit unsigned multiply
Enter with A=5BH
Multiplies HL by C, result in H:L:A, DE altered

Call # 92 undefined

Call # 93 8 bit unsigned divide.

Enter with A=5DH

Divides E by C result A, E=remainder

Call # 94 Divide 16 bit by 8 bit unsigned.

Enter with A=5EH

Divides HL by C, result HL, A=remainder

Call # 95 undefined

Call # 96 Convert Decimal Ascii to Binary

Enter with A=60H and HL points to string

Returns: result in BC

Call # 97 Convert hex to decimal

Enter with A=61H and HL=value, DE=buffer ptr

Converts HL to a decimal text string at DE

Call # 98 Convert C to ASCII at HL

Enter with A=62H

Converts C to a text string at HL

Call # 99 Convert DE to ASCII at HL

Enter with A=63H

Converts DE to a hexadecimal text string at HL

Call # 100 Get/set high/low memory limits.

Enter with A=64H, B=0 high, 1 low memory limit.

If HL=0 return current high/low setting else set to HL.

Call # 101 get system flag table

Enter with A=65H

Returns IY pointing at system flag table.

Call # 102 Memory Bank control.

Enter with A=66H and

B=0 select bank C = 0..2 or 80..82H. If bit 7 set
then control is transferred to address in HL
after bank changed. If Z80 Stack Pointer > 7FFE
then call aborts with error # 2BH.

B=1 Reset bank reserved flag, C=Bank # 0..2
Always OK. Returns Z flag set.

B=2 Test bank reserved flag, C=Bank #
Z if OK, else if A=2Bh then C out of range

B=3 Set flag to reserve bank, C=Bank #
Z if OK, else if A=2Bh then C out of range.

B=4 Return the No of bank in use at 8000-FFFFH
Result in A

Call # 103 Set Z80 break vector to HL

Enter with A=67H, HL=break vector

Technical Information

Exchanges HL with current Break vector.

Call # 104 Make beep via PC speaker
Enter with A=68H

Call # 105 Clear Screen.
Enter with A=69H, Causes home up and clear screen.

Call # 106 CKBRKC
Enter with A=6AH
Check break bit to see if BREAK KEY has been pressed
If set, reset bit, wait 1/30 sec and clear typeahead buffer.

Special video codes acted on by video output calls:

00-07h	not printed	
08h	backspace cursor	
09h	printed as dos 09	
0ah	move to start of next line, scroll if needed	
0dh	start new line, scroll if needed	
0eh	turn cursor on	
0fh	turn cursor off	
10h	reverse video and highbit routine ON	
11h	turn highbit routine OFF	
15h	swap tab/alternate	
16h	special/alternate	Not Implemented
17h	40 column	Not Implemented
18h	backspace cursor	
19h	cursor forward	
1Ah	move cursor down	
1Bh	move cursor up	
1Ch	move cursor to 0,0 (home up)	
1Dh	move cursor back to start of current line	
1Eh	clear from cursor to end of line	
1Fh	clear from cursor to end of screen	

6.2 Special Keys on the PC Keyboard.

This table gives a summary of the uses of the special keys on the PC Keyboard. Note that keys F1-F10 are programmable using the SETUP KEY command.

Function Key	Use	Code Generated
Escape (Esc)	Break	80H
Alt	Clear Key	None
Up-Arrow	Up-Arrow	0BH
Page Up (PgUp)	Shift Up-Arrow	1BH
Down-Arrow	Down-Arrow	0AH
Page Down (PgDn)	Shift Down-Arrow	1DH
Right-Arrow	Right-Arrow	09H
End	Shift Right-Arrow	19H
Left-Arrow	Left-Arrow	08H
Home	Shift Left-Arrow	18H
F1	TRS-80 Function Key F1	80H
F2	TRS-80 Function Key F2	81H
F3	TRS-80 Function Key F3	82H
F4	Shift-@ = Pause	60H
F5	Shift-Clear Key	1FH
F6	Shift-Clear-Equals	BDH
F7	Clear-Shift-1	A1H
F8	Clear-Shift-2	A2H
F9	Clear-B	C2H
F10	Break	80H
Ins	Clear-I	C9H
Del	Clear-D	C4H

6.3 The Simulated TRS-80 Keyboard Memory Map.

On a real TRS-80 the keyboard is memory mapped and appears (in Model 4 mode) at F400 hex when the appropriate functions swap it in. This is done by an OUT to port 84H which also brings in the video from F800-FFFF.

While keyboard/video is switched in, RAM from F400-FFFF is not accessible. PC-Four handles this by copying this section of memory to a temporary storage area at 0800H to 0FFFFH before swapping in the video and keyboard display.

The keyboard map is also available by using the undocumented KI function FF of RST 28 call # 5 which copies the 8 byte keyboard map (see below) to the address specified by the Z80 IY register.

Simulated Keyboard Map at F400 (when switched in)

Address -->	F480	40	20	10	08	04	02	01

Data Bit								
01	G	F	E	D	C	B	A	@
02	O	N	M	L	K	J	I	H
04	W	V	U	T	S	R	Q	P
08						Z	Y	X
10	7'	6&	5%	4\$	3#	2"	1!	0
20	/?	.>	-=	,>	;+	:*	9(8(
40	Spc	Ra	La	Da	Ua	Brk	Clr	Ent
80	F3	F2	F1	Cap	Ctl	LSh	RSh	

Spc=Space, Ra= Right Arrow, La=Left Arrow, Da=Down Arrow, Ua=Up Arrow
Brk=Break Key, Clr=Clear Key, Ent=Enter Key, Cap=Caps Lock, Ctl=Control
LSh=Left Shift, RSh=Right Shift key.

6.4 The File Control Block

TRS-80 file I/O is accomplished through the use of the File Control Block or FCB. The FCB is a 32 byte space reserved by the calling program. Initially, when the file is closed, the FCB contains an ASCII string representing the file name. The name must be terminated with an ODh. The name must also conform to TRS-80 file name conventions, that is 8 characters maximum for the body, a '/' separator and a 3 byte maximum extension. PC-Four does not support passwords. Filenames may only contain the characters A-Z and 0-9 and both name and extension must start with a letter.

Once the file is successfully opened by PC-Four, the name is saved and the FCB is used as follows:

```
FCB+0  0 closed 80 open
FCB+1
    Bit 7  0 if LRL=256, 1 if not
    Bit 6  1 if EOF to set to ERN only if NRN > EOF
    Bit 5  0 if i/o buffer contains current sector>NRN
    Bit 4  1 if buffer contents changed since read
    Bit 3  1 if directory to be updated always
    Bits 0-2  000 = access password (not used)
FCB+2  Used by PDS files
FCB+3/4 Buffer address  low/high
FCB+5  ptr within current buffer for next I/O  byte
FCB+6  No of drive (0-3) where file is located.
FCB+7  Hash code of file name
FCB+8  EOF byte. 00=256
FCB+9  Logical record Length (LRL) 00=256.
FCB+0A/0B NRN next record #
FCB+0C/0D ERN ending record #
FCB+18/19 MS-DOS Handle for this file
FCB+1A/1B Pointer to DOS file name & other info block
```

In a TRS-80 type DOS bytes at FCB+0E to 1F would be used for the directory and granule allocation type information. However, here in PC-Four some of these locations are used for other purposes. When your application program opens a file, PC-Four converts this to a request to open an MS-DOS file. If it exists, DOS assigns a file handle to it and a space is assigned by PC-Four to save the file name while it is open and to provide work space. FCB+18 points to this.

A maximum of 8 files may be opened at one time in PC-Four. MS-DOS itself opens additional "files" for Keyboard, Screen, Printer, STDERR, etc., so to be sure you don't run out of file space, make sure that your MS-DOS CONFIG.SYS contains a line which says: FILES=15. (The maximum is normally 20 so you could set it to 20 if you have other programs that need that many.)

6.5 Port I/O Simulation

Z80 Port Input and Output instructions are intercepted by the Z80 simulator which jumps out into 8086 code space and executes the equivalent operation that would have been implemented by the Model 4 hardware. Upon execution, control returns to the Z80 instruction immediately following the port call. Only a limited number of port calls are implemented. Calls to undefined ports are ignored.

Port 84 OUT: Memory Bank and Video Control

Bit	7	6	5	4	3	2	1	0
use:	Page	Fix	Mem-Bits	Inv	80/64	Select		
	*	*	*	*	*	*	*	Video

86 ; page 1, 80 col, video at f800, keyb at f400
 87 ; full 64k ram, bank0 at 8000-ffff

Bit 0=0 toggle video/keyboard out - copy F800-FF7F to video
 copy FF80-FFFF to keyboard buffer at 0200
 restore 0800-0F80 to F800-FFFF

Bit 0=1 toggle video/keyboard in -
 save F800-FFFF at 0800-0FFF
 copy video to F800-FF7F, copy 0200 keyboard buffer to FF80

Port E0-E3 OUT: Enables Real Time Clock if bit 2 set.

IN : Reads Interrupt status. Bit 2 RTC, Bit 5 RX RS232.

Port E8 OUT: RS232 UART Master Reset

IN : Read Modem Status

D7 CTS, D6 DSR, D5 CD, D4 RI, D0 Rx input

Port E9 OUT: Modem Baud Rate Set

00 50, 01 75, 02 110, 05 300, 07 1200

0A 2400, 0C 4800, 0E 9600, 0F 19200

IN: nothing.

Port EA OUT: Uart and Modem Control

D7 1=even parity, D6-D5 word length,

D4 1=2 0=1 stop bits, D3 1=no parity

+ D2 break, D1 DTR pin 20, D0 RTS pin 4

IN : Read UART status

D7 1=data rec'd, D6 1=TX empty, D5 1=overrun

D4 1=framing error, D3 1=parity error

Port EB OUT: RS232 Transmit Data

IN : RS232 Receive Data

Port EC-EF OUT: Writes to MODOUT flag at 0076H

IN : Reset RTC interrupt flag. Returns FF

Port F8-FB OUT: Printer output port D0-D7

IN : Printer port status

D7 busy, D6 out of paper

D5 Printer Selected, D4 Printer fault

6.6 Z80 Simulator Memory Map

On a normal Model 4, using TRS-DOS 6.x, DOS occupies most of memory from 0000 to 25FFH. Space from 2600H-2FFFH is often used for overlays. User application programs load at 3000H or higher although some may load at 2600H or call in an additional overlay module at that address.

PC-Four's DOS is almost completely outside the Z80 memory space since it is implemented in 8086 assembly code. DOS calls using the RST 28H code are trapped by the emulator and execution transfers to outside the simulation. Space from 0000 to 0FFF is used as shown in the list below and space from 1000H to FFFFH is free for user programs, i.e. 60k is free.

Memory Address (HEX)	Use
0000-0002	Return to DOS 3Eh,16h,0EFh
0008	Return from RST 08 C9h (RET)
000C	Number of current RTC task
000D-000F	Reserved
0010	Return from RST 10 C9h
0013-0014	Break vector
0015	Break flag
0016-0017	Reserved
0018	Return from RST 18 C9h
0020	Return from RST 20 C9h
0028	RST 28 Intercepted by Simulator
002C-002F	Internal PC 4 use only
0030	Return from RST 30 C9h
0038-003A	RST38 Real time clock jump
003B-004F	Interrupt vector table
0050	kbdrv
0055	dodrvr ;used by the
005A	prdrv
005F	sidrvr ;not alter
0064	sodrvr
006A-0089	System flag table
	RST 28 call # 101 points IY at aflag.
	Items marked with a * not used
0069	overlay request number 0 *
006A IY -->	aflag 1 *
006B	bflag 0 *
006C	cflag 0 *
006D	device flag 0
006E	0 *
006F	port FE mask 0 *
0072	country flag 0 * (USA)
0073	jflag 0 *
0074	keyboard control bits 0
	bit 0 latched to a 1 if Break key pressed

Technical Information

	bit 2	latched to a 1 if Enter key pressed	
	bit 5	forces upper case if set to 1	
0075		31h	*
0076	port EC image	78h	
0078	port 84 image	87h	
0079	printer flag	0	*
007A	qflag	0	*
007B	fdc retry count	0	*
007C	system flag	8	
	bit 4	set if break enabled	
	bit 5	set if JCL file active	
	bit 6	error message display control	
007D	Model # type flag	4	*
007E	uflag	0	*
007F	cursor type flag	0	*
0080	wrint	4	*
0085	DOS version flag	62h	*
0086	config init vector		*
0089	keyboard task vector		*
0090-00DF	RTC task handler		
00E0-00EF	RS232 handler code		
00F0-00FF	dummy drive code table		
0100-010B	Storage for PC-4 internal use		
010C	cursor		
010E	color	07h	;normal display
010F	inverse	70h	;inverse video attribute
0110	tabflag	; 00 tabs / FF spec chars C0-FF	
0111	nofcbs	; no of files currently open	
0112	Keyboard SVC vector (Enhanced Basic only)		
0114	bureg	; shows bank now in use	
0115		; bank 0 reserved flag	
0116		; bank 1 reserved flag	
0117		; bank 2 reserved flag	
0118-0131	reserved		
0132-013D	Function key translate codes		
013E	bareg	3	;banks 1 and 2 exist
013F	highbit	;80 if highbit routine on	
0140-01BF	TRS-80 > MS-DOS disk drive path equivalents		
01C0-01FF	DCB Table		
01C0	Keyboard	DCB	(KI)
01C8	Display	DCB	(DO)
01D0	Printer	DCB	(PR)
01D8	Serial In	DCB	(SI)
01E0	Serial Out	DCB	(SO)
01E8	Spare DCBs	(3)	

Technical Information

0200-024F	Keyboard buffer is at (80 chars) 201 storage ptr, 202 retrieve ptr, 203 start
0250-027F	reserved
0280-02FF	used for ramdir search string
0300-037F	stack space, initialized at 0380 at run time return address at 380-381 to dos stak DW dos_exit dos_exit LD A,16H ;22 - ret to dos RST 28H
0382-03FF	reserved
0400-05FF	MSDOS file handle buffers, 8 buffers each 32 bytes for file handle info and 32 for FCB store
0600-07EF	reserved for PC-Four internal use
07E0-0FFF	used by PC4 for window storage space and directory operations. During runtime used as storage to save f800-ffffh for video swaps
1000-FFFF	60 K of memory for user programs initialized with HLT (76h) code.

6.7 Undocumented Z80 Op-Codes implemented in PC-Four 2.0

PC-Four implements all the documented Z80 Microprocessor codes as well as the following undocumented codes. All Z80's made to date execute these codes and some programmers have chosen to use them. You will have to generate these codes using DB statements or MACRO calls.

Shift Left and Increment (SLI) codes:

CB30	SLI	B	CB31	SLI	C
CB32	SLI	D	CB33	SLI	E
CB34	SLI	H	CB35	SLI	L
CB36	SLI	(HL)	CB37	SLI	A
DDCBnn36	SLI	(IX+nn)	FDCBnn36	SLI	(IY+nn)

Codes acting on each half of index register IX, HX & LX:

DD24	INC	HX	DD25	DEC	HX
DD26nn	LD	HX,data	DD2Enn	LD	LX,data
DD2C	INC	LX	DD2D	DEC	LX
DD44	LD	B,HX	DD45	LD	B,LX
DD4C	LD	C,HX	DD4D	LD	C,LX
DD54	LD	D,HX	DD55	LD	D,LX
DD5C	LD	E,HX	DD5D	LD	E,LX
DD60	LD	HX,B	DD61	LD	HX,C
DD62	LD	HX,D	DD63	LD	HX,E
DD64	LD	HX,HX	DD65	LD	HX,LX
DD67	LD	HX,A	DD68	LD	LX,B
DD69	LD	LX,C	DD6A	LD	LX,D
DD6B	LD	LX,E	DD6C	LD	LX,HX
DD6D	LD	LX,LX	DD6F	LD	LX,A
DD7C	LD	A,HX	DD7D	LD	A,LX
DD84	ADD	A,HX	DD85	ADD	A,LX
DD8C	ADC	A,HX	DD8D	ADC	A,LX
DD94	SUB	HX	DD95	SUB	LX
DD9C	SBC	A,HX	DD9C	SBC	A,LX
DDA4	AND	HX	DDA5	AND	LX
DDAC	XOR	HX	DDAD	XOR	LX
DDB4	OR	HX	DDB5	OR	LX
DDBC	CP	HX	DDBD	CP	LX

Technical Information

Codes acting on each half of index register IY, HY & LY:

FD24	INC	HY	FD25	DEC	HY
FD26nn	LD	HY,data	FD2Enn	LD	LY,data
FD2C	INC	LY	FD2D	DEC	LY
FD44	LD	B,HY	FD45	LD	B,LY
FD4C	LD	C,HY	FD4D	LD	C,LY
FD54	LD	D,HY	FD55	LD	D,LY
FD5C	LD	E,HY	FD5D	LD	E,LY
FD60	LD	HY,B	FD61	LD	HY,C
FD62	LD	HY,D	FD63	LD	HY,E
FD64	LD	HY,HY	FD65	LD	HY,LY
FD67	LD	HY,A	FD68	LD	LY,B
FD69	LD	LY,C	FD6A	LD	LY,D
FD6B	LD	LY,E	FD6C	LD	LY,HY
FD6D	LD	LY,LY	FD6F	LD	LY,A
FD7C	LD	A,HY	FD7D	LD	A,LY
FD84	ADD	A,HY	FD85	ADD	A,LY
FD8C	ADC	A,HY	FD8D	ADC	A,LY
FD94	SUB	HY	FD95	SUB	LY
FD9C	SBC	A,HY	FD9C	SBC	A,LY
FDA4	AND	HY	FDA5	AND	LY
FDAC	XOR	HY	FDAD	XOR	LY
FDB4	OR	HY	FDB5	OR	LY
FDBC	CP	HY	FDBD	CP	LY

6.8 Error Messages

When a program running under PC-Four encounters a problem it issues an error message number which is printed as a hex number. Wherever possible PC-Four uses the same numbers as TRSDOS. In the event that an application program uses an unimplemented DOS call or illegal Z80 OP-Code then the simulation will abort to DOS and will display the current status of the Z80 registers. Following is a list of error messages.

<u>Error Code (Hex)</u>	<u>Meaning</u>
03	File Record Read Error
05	Data Record Not Found or Close Error
08	Device (e.g. printer) not available
0B	Lost Data in file write
11	Directory Read Error
18	File not in Directory / Write Error
19	File Access Denied
1C	End of File reached.
1D	Attempted to read past End Of File.
21	Too many files.
22	Load file format error
25	Can't delete file
26	Attempted to access a file not opened.
29	Attempted to open a file already open.
2B	RST 28H call parameter error.
34	Error deleting file.

.....

We expect to be continually evolving and improving PC-Four over the next year so if you are having problems running a specific Model 4 program let us know about it, try and give us as much detail as possible, describe your setup and, best of all, send a sample disk set up so that we can duplicate the problem.

I am submitting the enclosed TRS80 [] PC [] formatted disk. Please analyze it and determine why it will not work under PC-Four.

Comments:

Name:

Company:

Address:

City: State: Zip:

Phone - Day: Night:

Mail to: HyperSoft, P0 Box 51155, Raleigh, N.C. 27609, USA

Please feel free to copy this page.

