

Einstein Magazine

& ALL MICRO NEWS

Number 104



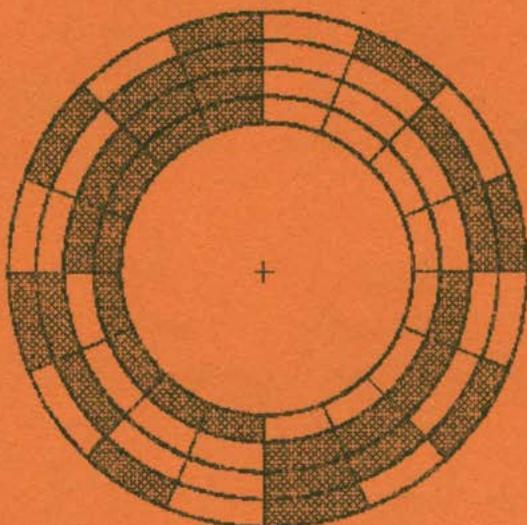
POLICE DEPARTMENT

NEW YORK, N. Y. 10013

INTERNAL AFFAIRS DIVISION

This is a confidential report prepared by the Internal Affairs Division. Any individual not specifically authorized in writing by the Commissioner who examines this report or who removes it for any purpose from the office of the Internal Affairs Division will be subject to departmental trial and-or criminal prosecution. Do not proceed beyond this cover sheet unless confident of your authorization.

Digitiser Revisited part 1 John Marriott



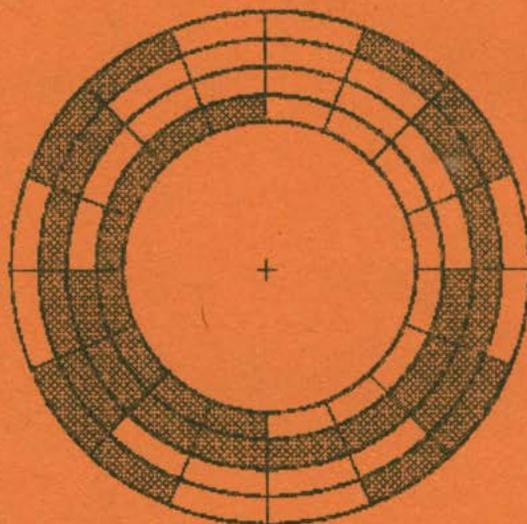
4-bit BINARY encoding disk

JM

In both cases, the outermost ring is the Least Significant Byte with the innermost ring being the Most Significant Byte.

The 'clear' sector(s) being a "0" with the shaded sector(s) being a "1".

From the BINARY to GRAY conversion program a linear list could be printed which'll provide a "length" which can be translated into a circumference & radius, draw the circle, nearly cut right across the list from LSB to MSB & glue that 'splayed' list round that circumference - rather like a daisywheel petal!



4-bit GRAY encoding disk

JM

You now have the basics of a GRAY-encoding wheel.



Einstein Magazine

& ALL MICRO NEWS

Number 104

Published for users of Einstein (and other) computers
by RPM Society.

Publisher and Secretary:
A E Adams, Ivy Cottage, Church road, New Romney,
Kent TN28 8TY

Editor: Bob Deeley
33 Britain St., Dunstable, Beds. LU5 4JA

Software Library, Shows & Useful Bits:
Steve Potts 85 Thorold Ave, Cranwell Village, Lincs.
NG34 8DS s.p.potts@talk21.co.uk

Einstein Web Site:
http://members.tripod.co.uk/~tatung_einstien
ftp://tatung_einstein:gwcdrw@ftp.tripod.co.uk/
Ken Ross can be contacted by his email...
cbm8032@bigfoot.com

Contents

Graphic Digitiser Revisited	2
CPMHELP.DOC.....	8

Graphic Digitiser Revisited

Part 1 John Marriott March 2001

In my article, (EM99/5-10) I referred to a system known as the 'GRAY SCALE' (except I spelt it with an 'E!') as a means of deriving an angular measurement - then pleaded 'complexity' as a way of chickening out. But, I started wondering about it (...usually about 10 minutes after draining the dregs of a bottle of plonk, shared bottle I might add - equally!), and came up with the following observations...

...if a true 360 degree circular GRAY scale (the same goes for a BINARY one) was aimed for, then a simple 8-bit "IN" command was out of the question as, say, DEC-360 is BIN-101101000 or GRAY-111011100 with the need of a 9-bit "IN" port (NOTE the leading 'minus signs'). However, in the Digitiser Project there was the probability that only a maximum 120 degree arc(s) would be involved - which in a way is good news as this brings the "IN" port requirement down to a 7-bit with the 'program' preloading an angular start/datum point which the disk reading modifies +/-.

To those who feel that they have a project which 'must' have a full 360 degree capability but staying within an 8-bit "IN" port - okay, but it lands you up with $360/255=1.41$ degrees per disc section (how's your maths?) and the headache of dividing a circumference by 255 (see how to do it the easy way - later)!

Another point to ponder on - if each 'sensor read' area is 5mm x 5mm minimum - then the 'innermost (MSB)' circumference would be $5\text{mm} \times 360=1800\text{mm.}$, about a 286mm radius. Mind you, as the Digitiser 'arms' are somewhere in that dimension region and we're only interested in parts of a circle i.e. arc(s) - reshape them as arcs with the GRAY scale included on the arc.

Anyway - back to the GRAY scale problem ... which in fact turned out to be quite simple, and fig.1 displays how a 4-bit BINARY number is converted to a 4-bit GRAY number. Using DEC-15/BINARY-1111 as an example and using the fig.2 XOR Truth Table, we arrive at GRAY-1000.

BINARY to GRAY electronic encoding

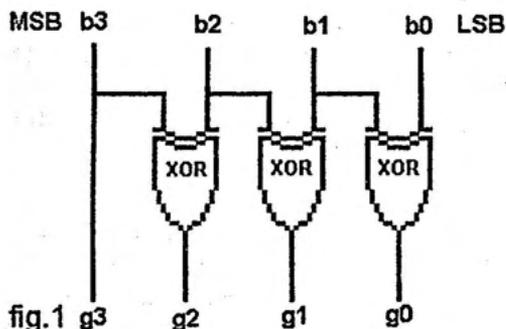


fig.1

XOR TRUTH TABLE

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

fig.2

GRAY to BINARY electronic encoding

Using my reasonably simple CASIO fx-107 Scientific calculator I came to the conclusion that the formula ' $\text{XOR INT}(\text{no.}/2)$ ' would mimic the above, resulting in the following program...

```

                                :Comments, don't type in!
10   FOR T=0 TO 360           :For 360 degree disk values
20   PRINT T,BIN$(T,9),BIN$(T :
      XOR(INT(T/2))),9)       :Watch brackets!
30   NEXT                     :Round the loop & end

```

Before going further - a reminder 'Why GRAY Scale?'. With BINARY and at a 'junction' of changing from 'one value disc sector to another' it is obvious that this value could 'hunt' between those sectors, but not so obvious that it could 'hunt' diagonally - so giving rise to 4 possible values! Not so obvious is the fact that the 'diagonal values' can provide completely incorrect information ... not such a problem at the LSB end, but at the MSB end I wouldn't like to rely on navigating across my shower tray, let alone to the Moon or Mars with such 'inconsistencies'!

Of course, the GRAY Scale has the same problem, except as it only changes ONE sector bit at a time removes the 'diagonal' effect, where the problem lies. Further on in this article are 'disks' which you can compare and the best way to 'look at' a disk is to rotate the sector you want to look at/evaluate to the '3 o'clock' position - the same if you're constructing one as sooner or later you'll 'cross-over' the LSB/MSB bits - and 'colouring in' a 360 sector disk ain't a bundle of laughs after the first half-dozen cock-ups!

Sadly, that old saying comes to mind - "...you don't get 'owt for n'owt..." for the simple reason the GRAY Scale has no directly usable logic and has to be converted back to a 'value' both we and our computer can 'understand', i.e. BINARY - which means either a 'formula translation' or a 'look up table' to get to that. Unfortunately, I've not found a simple conversion formula for GRAY to BINARY, fig.3 and fig.4 showing why.

GRAY to BINARY electronic encoding

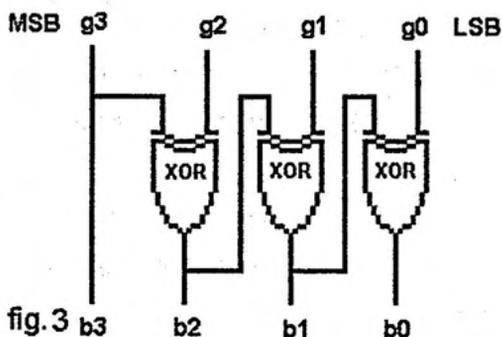


fig. 3

XOR TRUTH TABLE

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

fig. 4

The MID\$ Command appears to be the key i.e. A\$="1000" (dummy GRAY value):LET MID\$(A\$,1,1)=MID\$(B\$,1,1):LET (MID\$(A\$,1,1)) XOR (MID\$(A\$,2,1))=MID\$(B\$,2,1):LET(MID\$(A\$,3,1)) XOR (MID\$(B\$,2,1))= MID\$(B\$,3,1) & etc.

The following program demonstrates what I'm aiming for.

```

10 B$="" :Define null string
20 X$="ABCDEFGH" :Define string
30 FOR T= 8 TO 1 STEP -1 :Declining loop
40 PRINT MID$(X$,T,1) :Print a character from X$
50 B$=B$+MID$(X$,T,1) :Copy that character to B$
60 NEXT :T=T-1
70 PRINT B$ :B$="HGFEDCBA"

```

Seemed easy enough, but after some trial and error (mainly frustrating error!) the following program resulted.

```

10 REM GRAY to BINARY :Comments, don't type in
   converter. John Marriott : colon and after !
   March 2001 :
20 A$="1000" :Dummy GRAY value
30 L=LEN(A$) :String length check/counter
40 IF L<2 THEN 20 :Value to small!
50 R=L :Loop value store
60 IF MID$(A$,1,1)<>"1" :
   THEN 20 :Check for incorrect A$ input
70 C$=MID$(A$,1,1) :1st. GRAY to BIN transfer
80 F$=C$:PRINT F$ :Now store & check print it
90 L=L-1 :Loop value reduce
100 D$=MID$(A$,2,1) :Next GRAY character
110 E=VAL(C$) XOR VAL(D$) :And convert
120 B$=STR$(E) :Convert a number to a letter
130 G$=RIGHT$(B$,1) :Strip off leading <space>
140 F$=F$+G$:PRINT F$ :Now store & check print it

```

```

150 L=L-1:IF L=0 THEN 230      :Check for end of loop
160 FOR T=3 TO R              :Rest of GRAY string
170 C$=MID$(A$,T,1)          :Next GRAY string character
180 E=VAL(C$) XOR VAL(G$)    :Lines 110 to 140 refer
190 B$=STR$(E)                :
200 G$=RIGHT$(B$,1)          :
210 F$=F$+G$:PRINT F$       :
220 NEXT T                    :
230 PRINT "GRAY ";A$;" equals :
      BINARY ";F$            :And that's-it!

```

Line 20 can be changed to INPUT A\$ so that you can input a GRAY number, but it must start with a "1" and be at least 2 characters long, i.e. 10 or 11 and the PRINT F\$ littering the program are just there for checking purposes. Line 230 is the OUTPUT for use as, say, angular measurement within your own digitiser program - what have you. Being in BASIC it is more suited to 'slow sampling', i.e. wind vane direction reading - but my intention was to design a working model for evaluation, working on my ethos if I couldn't do it in BASIC then I'd no chance in M.Coding!

EM has run projects on 'Optical shaft encoders/circuitry (interrupted LED light sources)' and I've no intention of deliberately 're-inventing the wheel'. Hopefully, in the second part of this article I'll cover 'Look up/comparison tables' with a M.Code 'search & find' for those requiring a faster response.

Going back to 'those' who want to divide a circumference up into 255 (or any other number for that) - mark off the edge of a long strip off paper, say, every 5mm, measure its length, that is the circle's circumference, so divide that by 2Pi (18.86 is close enough) to derive the circle's radius and proscribe (draw!) that on a piece of stiffish white card. Make that marked strip of paper into a tube (marks outward) and lightly PVA glue it to the proscribed circle. When dry, use the point of the compass to mark the card where the

tube marks fall and then gently ease/remove the tube from the card. Dressmaker pin through centre so that the card can swivel and the pin stay stable/fixed - temporary pin to first point mark on circumference and lightly lay a ruler against the pins, draw radius line - move temporary pin to next mark and continue & etc with the rest, BUT move card NOT ruler. Yes, the pencil line won't line up with the outside mark, but relatively the lines will with each other - hence LIGHTLY laying the ruler against the pins.

Right, for those who've got this far - Crystal BASIC can XOR 'normal' mathematical numbers (so much for my Casio calculator being Scientific!) which means a core program for the GRAY-BINARY conversion can be based on the following.

```

10   G=15:R=G           :Dummy GRAY value G-1000
20   FOR L=1 TO 3       :Loop counter for G-1000
30   B=G XOR INT(G/2)   :Learnt something there!
40   PRINT "B=";B       :Check print it
50   G=B                :New value into G
60   NEXT               :Round the loop
70   PRINT "GRAY(15)=";BIN$(R,      :As it says...
      8);"=DECIMAL-";BIN$(B,8)

```

What was that I was saying, about "...no easy formula..." - mulling over fig.3 on a 2 hour walk in the wind and rain resulted in a "...try it, don't assume..." and of course it worked. Line 10 could be a 'derived/pick-up' value with the need to obtain a 'binary string' value so that its 'length' could be placed in the loop counter of Line 20 after any leading zeroes are stripped from its 'front end', but it would be the 'normal' mathematical number which would be 'operated' on.

Anyway - I've learnt something, hopefully you too?

John Marriott - March 2001 - Graphic digitiser re-visited - part 1

Ed. Coding wheel diagrams pertain, printed on flyleaf.

----@@@----

CPMHELP.DOC

by Tom Kell and John Ibberson

CPMHELP.DOC is not meant to be comprehensive. It does not go into depth on the more advanced CP/M commands, nor does it attempt to compete with the MicroTechnology 'Supplements' which cover all the special features that they added to their Sharp versions of CP/M. Its main purpose is to explain the most common CP/M commands, many of which are pretty incomprehensible to the average beginner.

THE CP/M SYSTEM PROMPT

When you load CP/M, the system prompt A\> is displayed. This indicates that A: is the current 'default' drive i.e. the drive that will be assumed if no drive identifier is given. You may set any available drive to be the default drive by typing in the desired drive letter, followed by a colon and a carriage return. For example, if the current default drive is A:, the command:

A\>B:(cr)

will change the default drive from A: to B:, and cause the system to respond with a new prompt:

B\> (showing that the default drive is now B:)

IT IS IMPORTANT TO CHECK THE DEFAULT DRIVE BEFORE EXECUTING ANY COMMAND, AS THE RESULT OF THAT COMMAND MAY WELL DEPEND ON IT.

CP/M FILE NAMES

A disk drive may be considered as a cabinet holding a number of files holding information such as accounts, letters, or data, etc. Each file has a label which may be, and usually is, in two parts. To identify a file, it is necessary to specify the drive and both parts of the label. Under CP/M, the disk drive is referred to by its 'drive identifier', and the two parts of the file label are called the 'filename', and the 'filetype' (or 'extension').

FILE NAME FORMAT

A complete file name thus consists of three parts, and these must be given in the correct format i.e:

d:filename.typ

The drive identifier d: is an alphabetic character followed by a COLON; the letter is usually in the range A-D, but other letters are possible. This identifier may be omitted altogether, in which case the default drive, as shown by the system prompt, is assumed.

The 'filename' must be given, and be in the range 1-8 characters.

The 'typ' is optional; if used, it MUST be preceded by a dot, and can thereafter consist of 1-3 characters.

Sharp Users Club - General Section - CPMHELP.DOC p.2

Most keyboard characters can be used in 'filename.typ'; but the following are reserved for special uses and may NOT be used:

\ < / > . , : ; = ? * []

N.B. In CP/M, SPACES are NOT allowed in 'filename.typ' !!

FILE TYPES

Normally the filetype is used to indicate what sort of information is held in the file. Examples of common file types:

- .BAS BASIC file i.e. a program written in a BASIC language
- .COM COMMAND file i.e. a ready-to run program in machine-code
- .DAT DATA file e.g. figures to be used by a spreadsheet
- .DOC DOCUMENT or text file e.g. a letter.

However, a filetype can impart other information e.g. README.1ST

AMBIGUOUS FILENAME CHARACTERS (otherwise known as 'WILDCARDS')

If you wish to specify a range of files rather than one file, you may use '?' or '*' as Ambiguous File Characters or 'Wildcards', in a filename or filetype. '?' gives an unqualified 'match' for any one

If you try to erase ALL files from a disk by e.g. A\>ERA *.*\<cr\> the program responds with 'ALL (Y/N)'. You may then type Y\<cr\> to execute the command, or N\<cr\> to change your mind.

Some files are protected so that they cannot be erased, in which case the following message will be displayed:

Bdos error on d: file R/O (N.B. d: is the disk drive identifier).

The only way to escape from this situation is to re-boot the CP/M system with a CTRL-C.

(N.B. If the message is accompanied by a continuous clunk, clunk from the drive, you may find that this is because the relevant disk is fitted with a write-protect tab!)

Sharp Users Club - General Section - CPMHELP.DOC p.4

REN - Rename a file

As the filename is simply a label which enables identification, we can change its name without changing its contents. To rename a file, both its NEW NAME and its OLD NAME must be specified in the following general syntax:

A\>REN d:NEWNAME.typ=d:OLDNAME.typ(cr)

Where d is the drive identifier, which MUST be the same for both OLD and NEW names, as the REN command will not copy a file from one drive to another. NOTE that if the new filename already exists an error message will be displayed. Examples:

A\>REN THATFILE.DAT=THISFILE.DAT\<cr\> (Rename THISFILE.DAT to THATFILE.DAT, on the default drive)

A\>REN B:NEW.TXT=B:OLD.DOC\<cr\> (Rename OLD.DOC to NEW.TXT, on B drive)

TYPE - List the contents of a file

The TYPE command allows the contents of a file to be viewed on the screen. The general syntax for this command is:

A\>TYPE d:PEEPFILE.DOC\<cr\>

You will find that some files (e.g. all .COM files) do not make sensible reading, and may even upset the screen completely. This is because the information is intelligible only to the computer. Files of types .DOC, .TXT, or .ASM will usually 'TYPE' O.K.

If the file is too long to be displayed on one screen, it will be scrolled up at a speed which will be much too fast to read. To temporarily stop the listing type Control-S, any key pressed will restart the listing. To stop the listing altogether hit any key during the listing.

SAVE - Save a file

When you load a CP/M '.COM' program, it is put into the Transient Program Area (TPA), starting at 0100H. This is the start of 'page 1' of memory (a 'page' is a block of 256 or 0100H bytes; 'page 0', which is used by the CP/M system, is \$0000-\$00FF, and 'page 1' is 0100H-01FFH). Thus, for example, a program which uses 0100H-03FFH actually fills 3 'pages' of memory; you can always tell the number of pages occupied from the first two hexadecimal digits of the end address of the program - in the above case, it is 03H pages.

Using the 'DDT' utility (see p.8 of this guide) you can modify an existing CP/M program, or even write a new program of your own. If you do this you will, upon exiting from DDT with CTRL-C, wish to SAVE the program before you do anything else. This is done with the SAVE command, which requires not only an unambiguous filename, but also the number of 'pages' of memory to be saved, specified in DECIMAL notation. The general syntax is as follows:

SAVE n filename.typ\<cr\> ('n' is the DECIMAL NO. of pages)

Sharp Users Club - General Section - CPMHELP.DOC p.5

Examples of using the SAVE command:

A\>SAVE 3 TEST.COM\<cr\> saves from 0100H to 03FFH (3 pages), as TEST.COM, on the default drive A.

A\>SAVE 20 BOAT.COM\<cr\> saves from 0100H to 14FFH (20 pages) as BOAT.COM, on the default drive A.

A\>SAVE 30 B:CANOE.COM\<cr\> saves from 0100H to 1EFFH (30 pages) as CANOE.COM, on drive B.

THE CP/M SYSTEM TRACK(S)

The built-in files are part of the 'kernel' of the CP/M system. Physically, the system is always stored on the early tracks (00 and upwards) of a CP/M disk; on the MZ-80A/B/700, the system is small and fits onto track 00; on later machines such as the MZ-3500/MZ-800 the system is larger and uses more than one track.

The first part of track 00 is always formatted to the machine's native SHARP disk format (commonly 256-byte sectors) to allow the disk to boot via the Sharp FD card; the rest of track 00, and all other tracks, are in CP/M format, which can be very different. For example, CP/M on the MZ-80B/A/700 uses 512-byte sectors - 10 per physical track (which amounts to 20 per logical CP/M track because a logical CP/M track covers BOTH sides of the disk).

On the MZ-80B/A/700, track 00 is reserved for the CP/M system, and if a disk does not carry the system, this track is left empty; so on those machines there is no point in not putting the CP/M system on all disks, using SYSGEN as described below. In any case, it is always more convenient if the disk in drive A: is a system disk.

CP/M TRANSIENT COMMANDS

Transient Commands are not part of the 'kernel' of CP/M, and are not kept in memory as part of the CP/M system; they are stored on disk as separate '.COM' files, loaded into memory as required, and cleared from memory on exit. So, if you wish to use a Transient Command, the appropriate '.COM' file MUST be on one of your disk drives. If it is not on the default drive, the drive identifier must be entered before the command. Note that, when executing a transient command, it is NOT necessary to type in the '.COM' extension - and indeed if you do, the command is not recognised !

PIP - FILE COPYING UTILITY

PIP has many uses, but the most common are copying a file onto the same disk but with a different name, copying one file or several files to another disk, or merging several files into one file.

COPYING FILES

The general syntax is as follows:-

A\>PIP d:DESTN.typ=d:SOURCE.typ\<cr\>

NOTE that the source file is not affected by any PIP operation.

Sharp Users Club - General Section - CPMHELP.DOC p.6

Examples of using the PIP command:

A\>PIP A:NEWCOPY.DOC=A:NEWFILE.DOC\<cr\> Copies NEWFILE.DOC on the A drive, to a file named NEWCOPY.DOC, also on the A drive.

A\>PIP B:NEWFILE.DOC=A:OLDFILE.TXT\<cr\> Copies OLDFILE.TXT on the A drive, to a file named NEWFILE.DOC on the B drive.

If the drive identifier is omitted, the default is assumed e.g:

A\>PIP B:NEWFILE.DOC=OLDFILE.TXT\<cr\> Copies OLDFILE.TXT on A to NEWFILE.DOC on B i.e. the same as the previous example.

A\>PIP PAPER.DOC=SHEET.TXT\<cr\> Copies SHEET.TXT to a file named PAPER.DOC, both on the A drive.

If destination and source filenames are the same, the destination filename need not be given, as it will be assumed e.g.:

A\>PIP B:=LATE.DOC\<cr\> Copies LATE.DOC on the default (A) drive, to LATE.DOC on the B drive.

If the destination and source drives are the same, the filenames must be different, otherwise an error will occur. Multiple files can be copied by using 'Wildcards' in the source filenames; but the destination must be another drive and destination filenames cannot be specified, as all the new files are exact copies e.g.:

A\>PIP B:=*.COM\<cr\> Copies all files on the default drive which are of filetype '.COM', to the B drive.

A\>PIP B:=CH*.*\<cr\> Copies all files on the default drive which have filenames starting 'CH', to the B drive.

If the destination file already exists it will be overwritten, unless it is READ ONLY. In that case 'DESTINATION IS R/O DELETE (Y/N)' will

be displayed; to continue, enter 'Y'; to cancel, enter 'N'. If 'N' is entered, the destination file will not be affected.

MERGING FILES WITH PIP

Several files can be merged into one large file using the PIP command. The general syntax is:-

```
A\>PIP
d:DESTN.typ=d:SOURCE.typ1,d:SOURCE.typ2,...,d:SOURCE.typN
\<cr\>
```

The SOURCE files are merged into the DESTN file in the order 1,2,...,N as specified. The original SOURCE files are not affected. Do not use 'Wildcards' as they give unreliable results.

STAT - File Status Utility

Each directory entry includes the filename, disk location, size, and status of the relevant file. The status shows, amongst other things, whether the file is Read and Write (R/W), or Read Only (R/O); R/W files may be altered or deleted, but R/O files may not. For this reason, all essential CP/M system files are made R/O. To protect such a file even further, it is possible to suppress its appearance on a DIR listing, by making it a SYS file, see below.

Sharp Users Club - General Section - CPMHELP.DOC p.7

(STAT - File Status Utility, continued)

One use of STAT is used to list a file's status (or attributes). Using STAT, you can find the status or attributes of a single file, or of an ambiguous group of files. The general syntax is:

```
A\>STAT d:filename.typ\<cr\>
```

Examples:

```
A\>STAT A:CHECK.COM\<cr\>
```

```
RECS  BYTES  EXT  ACC
  30   4K    1   R/W  A:CHECK.COM
```

Bytes remaining on A: 290K

```
A\>STAT A:*.DAT\<cr\>
```

RECS	BYTES	EXT	ACC
25	4K	1	R/W A:FINE.DAT
180	24K	2	R/W A:COLDER.DAT
1	2K	1	R/O A:SUCH.DAT
30	4K	1	R/W A:(SPEND.DAT) (Suppressed)

Bytes remaining on A: 282K.

RECS = number of 128-byte records occupied. The real size of the file may therefore be calculated to the nearest 128 bytes.

BYTES = disk space occupied, Kbytes. CP/M uses 2Kb blocks so this is always a multiple of 2Kb, and may be more than the real size.

EXT = how many 'extents' are used (1 extent holds 16kb max.)

ACC = status (R/W, R/O etc.) If the filename is in brackets () its DIR listing is suppressed (but the file is still available).

Another use of STAT is to change the status to R/W, R/O, or SYS. In this case a \$ or # 'operand' is required after the filename. 'Wildcards' are permitted, and the general syntax is:

```
A\>STAT d:filename.typ $operand\<cr\>.
```

Possible '\$' or '#' operands are:-

\$R/W sets file to Read/Write.

\$R/O sets the file to Read Only.

\$SYS suppresses directory listing of the specified file.

\$DIR restores directory listing of the specified file.

EXAMPLES:

```
A\>STAT A:TREAT.DAT $R/O\<cr\>
```

changes the status of TREAT.DAT on the A drive, to R/O.

```
A\>STAT A:*.COM $SYS\<cr\>
```

changes the DIR attribute of all COM files on A: to SYS

'#' is permitted instead of '\$'. If the desired setting is already in force (e.g. when using 'Wildcards'), no change is made.

Sharp Users Club - General Section - CPMHELP.DOC p.8

(Further examples of the STAT command):

STAT (on its own) returns the amount of unused space on all drives logged on the system.

STAT d: returns the amount of unused space on drive d:

STAT VAL: returns the POSSIBLE logical assignments for the four I/O devices.

STAT DEV: returns the CURRENT logical assignments for the four I/O devices, in 1978 jargon e.g.

CON: is CRT: (Console is a CRT)

RDR: is TTY: (Paper tape reader is RS-232 OUT)

PUN: is TTY: (Paper tape punch is RS-232 IN)

LST: is LPT: (List device is a 'line printer')

STAT DSK: returns the current logged-on disk drives

STAT d:=R/O Turns the drive d: to READ ONLY.

OTHER STANDARD CP/M FILES ON YOUR SYSTEM DISK

(NOTE: All Microtechnology versions of CP/M for Sharp computers contain files which are not found in standard CP/M systems; these files are dealt with in the next section of this guide.)

ASM.COM and LOAD.COM

ASM.COM assembles an 8080 source code file into a .HEX file. LOAD.COM converts a .HEX file into an executable .COM file. These programs are for experts, and are not covered further here.

DDT.COM

This is the standard CP/M 'dynamic debugging tool'. Once mastered, it allows you alter programs, or even write new ones. The general syntax is DDT filename.typ - this loads DDT.COM into high memory

and then loads filename.typ at \$0100, where it may be altered, tested, or disassembled, in its normal CP/M location.

However, DDT.COM is a very ancient program, and uses 8080 Assembly mnemonics, instead of Zilog Z80 mnemonics. It is therefore tricky to use in many respects, and I shall not discuss all its commands. If you are interested, read the DDT section of any CP/M Manual.

The most useful command is the 's' command, which allows you to alter memory locations within a program stored at \$0100 in RAM. You can then exit from DDT with CTRL/C and then IMMEDIATELY save the altered program with the SAVE command as mentioned earlier.

Alternatively, you can just type the command DDT without a program filename. This will load DDT high in memory, and you can then use the 's' command to write your own program, starting at \$0100, and then exit from DDT and SAVE it as a brand new program.

DUMP.COM

This dumps a file to the screen, in HEX bytes; unfortunately there is no ASCII interpretation alongside it, and you need CTRL-S to halt scrolling. The syntax is DUMP filename.type

ED.COM

A VERY primitive text editor. Forget it - almost any wordprocessor which can generate plain ASCII text can be used much more easily !

Sharp Users Club - General Section - CPMHELP.DOC p.9

(Other Standard CP/M files, continued):

MOVCPM.COM

Relocates the CP/M system. Most users never use this command, but it is needed to install extra CP/M routines for a hard disk drive.

SUBMIT.COM

Allows you to 'SUBMIT' a 'batch file' i.e. a plain ASCII file listing the commands to be executed, in order. The file may have any name,

but it **MUST** have the extension .SUB (you can prepare it using any W.P. which creates plain ASCII files). For example, if the batch file is SETUP.SUB, the command SUBMIT SETUP will execute the commands in SETUP.SUB, in order. Two typical .SUB files, with explanatory comments which are NOT part of them, are shown below:

SETUP.SUB (to set up MZ-80B)	PT.SUB (to run PT on MZ-80A)
SPIN8 ; set spin time	SPIN8 ; set spin time
CONSOLE R ; reverse video	ACPMKEYS ; set 'CP/M' keyboard
TIME ; display time	SOFTCENA ; set Soft Centronics
FILES ; do sorted DIR	CONSOLE C80 ; set 80 columns
	PT ; run Peachtext

A further very useful refinement is possible. If you rename (or copy) SUBMIT.COM to a file called INIT.COM, and call the batch file SYS.SUB, the batch file is run automatically at boot-up.

SYSGEN.COM

Copies the CP/M system on to another CP/M disk. The command SYSGEN begins a dialogue with the program asking for a SOURCE DRIVE NAME, to which you supply (usually) A and get the message:

SOURCE DRIVE on A: THEN TYPE RETURN
 FUNCTION COMPLETE
 DESTINATION DRIVE NAME

to which you type (usually) B and get the message:

DESTINATION DRIVE on B: THEN TYPE RETURN
 FUNCTION COMPLETE

After this procedure has been completed, the disk in drive B: will contain the CP/M system on its first track (or tracks).

SPECIAL SHARP COMMANDS ADDED TO CP/M BY
 MICROTECHNOLOGY LTD.

NOTE: Some commands exist on only one machine e.g. BOOT.COM on the MZ-80B, or MZ80B.COM, PC3201.COM, etc. on the MZ-3500:

BOOT - re-boots the MZ-80B. In theory, you can do this by pressing the the IPL button, but BOOT is safer under some circumstances.

MZ80B b: - sets drive b: on the MZ-3500 to MZ-80B disk format (Also MZ3540.COM, PC3201.COM, SUPER.COM, IBMSD.COM, 8INCH.COM)

BACKUP.COM

BACKUP starts with a menu of backup options. The most useful option is 'B' Backup whole disk (Copy \& verify including system tracks), which by default copies from drive A: to drive B:.

Sharp Users Club - General Section - CPMHELP.DOC p.10

(Special Sharp commands added by MicroTechnology Ltd., continued):

CMT.COM

Copies Sharp tapes to a CP/M disk, or vice versa. There are many options (CMT alone displays them) but the most useful are:

A\>CMT d:filename.typ/S:XXXX:YYYY:ZZZZ

(SAVES d:filename.typ to tape, with the same name as the disk file, and SIZE/LOAD/EXEC parameters XXXX:YYYY:ZZZZ).

A\>CMT d:filename.typ/N

(LOADS first file off tape, saves it on disk as d:filename.typ)

CONSOLE.COM

On the MZ-80B, CONSOLE may be used as shown below; on other machines these may not all apply. The full options are:

CONSOLE 40... 40 column screen

CONSOLE 80 .. 80 column screen

CONSOLE N ... Normal screen (green on black)

CONSOLE R ... Reverse screen (black on green)

CONSOLE Sx,y Set the scrolling area from x to y.

(When you have ended up with a 'funny' screen after typing for instance a .COM file, then CONSOLE can put things back to normal).

COPY.COM

Copies your screen to printer (i.e. a 'screen dump').

DEL.COM

A belt-and-braces version of ERASE. The syntax is DEL filename.typ

EJECT.COM

When the console output is directed to printer (CTRL-P), EJECT will 'throw' a blank page of paper, using a FORM FEED.

FILES.COM

Displays an alphabetically-sorted directory in four columns, plus an extra line showing a summary of the used and available space on the disk. Other versions of this program exist e.g. D.COM, SD.COM

FORMAT.COM

Formats a blank disk (or reformats an already-formatted disk). You are asked which drive to format (A: B: C: D:), and then asked to press any key to continue (CTRL-C aborts at this stage). FORMAT verifies everything, and gives a verification message at the end.

INIT.COM

As supplied this is a 'dummy' or blank file. When the system boots up, INIT.COM is run automatically. You can write your own INIT.COM (in machine-code) to e.g. set up system parameters or a special printer routine. Alternatively, you can rename (or copy) any .COM file as INIT.COM, and it will run automatically at boot up.

TIME.COM

TIME on its own displays the time elapsed since you booted up. Alternatively, TIME 16:37:00\<cr\> sets the time to 16:37hrs, or whatever other time you input.

OTHER USEFUL CP/M PROGRAMS AVAILABLE AS P.D. SOFTWARE

UNERASE.COM

In CP/M, an 'active' directory entry begins with a 00H byte ; but when you ERASE a file, this byte is changed to E5H. Nothing else happens at that stage, so the file is still on disk, and you can restore it by changing the E5H back to 00H - and this is just what UNERASE does, if you ask it to. The syntax is:

UNERASE d: (list erased files on d:, ask which ones to UNERASE)

NOTE: if you ERASE a file, and then SAVE another file on the same disk, the sectors used by the erased file may be allocated to the new file i.e. all or part of the erased file may be overwritten. Therefore UNERASE only works on an erased file which has not been overwritten by a subsequent SAVE. Also, after using UNERASE, you must do CTRL-C to have the change(s) recognised by the system.

VERIFY.COM

Asks for 2 filenames, then reports whether the files are identical or not (very useful for comparing 'lookalike' system files, or for tracing 2 copies of the same user file with different names).

WASH.COM

This is a directory and file maintenance utility which starts up with a list of the available commands:

Space = next file	V = View at CRT	C = Copy a file
S = Start new drive	CR = next file	L = List file
D = Delete File	U = Show disk Space	B = Previous File
P = Punch File	F = Show file size	X = Reboot system
Z = Zip ahead	M = Tagged Copy	R = Rename File
T = Toggle Tag	Q = Tagged Delete.	

and a list of the files currently on the disk, with the 'current' file 'highlighted' by \> and \< signs e.g. \>WASH.COM\<

SPACE moves the highlight down, or you can 'ZIP' from one column to the next with 'Z'. The 'T' (tag) option places '#' beside the chosen filename; you can tag as many files as required and then block copy them with 'M', or block delete with 'Q'. To copy or delete a single file, highlight the file and press 'C' or 'D'.

SWAPPING CP/M DISKS

CP/M keeps track of disks in your drives, and if you remove a disk and replace it with another, the system will NOT let you write to that drive until you have done a CTRL-C. The reason is as follows:

At boot-up (or on CTRL-C), the CP/M system looks at the directory of every disk in use, checks it for occupied and 'free' sectors, and stores the answers in RAM, in the File Allocation Table (FAT). Thereafter, whenever you attempt to save a file, CP/M checks its filesize, examines the FAT to see if there are enough free sectors available, and then saves the file (if it can). But if you change a disk without doing a CTRL/C, the FAT of the old disk is still in memory; under such circumstances, a save may overwrite an existing file on the new disk. Hence the restriction; a CTRL-C establishes the 'credentials' of the new disk, by creating a new FAT.

----@@@----

Editors note about CPMHELP.DOC.

Many of us have dabbled with the CP/M operating system on Einey or elsewhere, and would like to know more - well here's material that Tony has dug out for us, once a suppliment from the Sharp Users Club as CPMHELP.DOC, eleven pages as originally printed. I could have split this over to the next EM, but thought we would all prefer to keep it together as one source of reference. Incidentally, CP/M, as it should be written, stands for control program for microcomputers, launched in 1976 by Digital Research, attributed to Gary Kildall.

----@@@----

*
|
*

MUMBLINGS FROM NEW ROMNEY

This issue of EM will reach you a month later than intended, because long after your copy ought to have been in the post, it hadn't even been printed yet. You nearly had to wait even longer though, because Bob Deeley didn't have nearly enough material to make up a complete magazine at press date, and we don't much like to send out a lot of blank pages to you.

Fortunately, some time ago the Sharp User Club issued a fairly comprehensive guide to using CP/M on Sharp Z-80 disk-based computers for users without manuals or documentation on it, and agreed that we could reproduce it too, so Bob has used this to make up a full 24-page issue this time around.

CP/M was originally developed by a couple of programmers as a monitor program for machine code programs running on Z-80 machines using reel or cassette tape storage, so they didn't have to write a new interface between the machine and the guts of the software each time they wrote a new program, but had a standard module, fully debugged, for just this task.

Many Einstein users will be familiar with XBAS -- a form of BASIC. This is a combined monitor and interpreter. An unusual feature of the Einstein is its MOS mode, which is very close to a monitor program in the original sense.

Having developed this monitor system for their own use, the programmers obtained the loan of a very scarce modular MDS Z80 computer system with 8" disk drives and "Isis" operating system which Intel had put on the market as a hardware and software development tool, and used this to add a control program for peripheral units to their monitor module. This was sold, alongside their other software, to anyone who would buy it, as their "Control Programme/Monitor" software, which quite naturally became abbreviated to "CP/M". Where the interpretation of "CP/M" as "Control Program for Microcomputers" came from isn't clear. It was often quoted later by journalists (and DR itself?), but it just ain't so.

It's amazing that we have been able to keep the user group and magazine going for nearly a decade after Graham Bettany (quite rightly) gave up on it as a hopelessly lost cause, but it looks like Einstein input alone just won't keep the magazine going. We do want to include as much Einey stuff as we can, but there isn't enough of it coming in to sustain the mag without other stuff too. We now need YOU to send in ANYTHING AT ALL that is worth sharing with the rest of us -- EVEN IF IT'S NOT COMPUTER RELATED -- so there's still a mag to print Einey stuff in, as and when we can lay hands on it.

MARKET PLACE

Don Tordoff has a pile of EINSTEIN USER mags from Vol. 1 No.1 onwards, which are looking for a new home, also some Einstein software, original manuals, two disk drives and a power supply unit. These are all free to anyone who is willing to collect or pay carriage from The Rectory, Spennithorne, Leyburn, North Yorks, DL8 5PR. (01969-624894)

EINSTEIN MAGAZINE SALE - ALL PRICES INCLUDE POSTAGE

Back numbers of the user group magazine are still available @ One 1st class stamp per mag, or 4 for £1.00, or £20.00 for a copy of every one of the back numbers that we still have.

CUPBOARD CLEAROUT - SALE OF EINSTEIN BOOKS & MANUALS
(ALL PRICES INCLUDE POSTAGE)

Bell	- InfoBase (apd utilities) user manual	£1.00
"	- Football Prediction Package instructions ..	£1.00
"	- " " " (Australian) "	£1.00
"	- Super Draw and Paradox user manual	£1.00
Crystal	- System 5 / DOS 2 user manual	£1.00
"	- Xtal BASIC Ver.5 compiler user manual	£1.00
Einstein User Group	- Einstein Compendium No.1	£1.00
"	- The Einstein Speculator (User Guide)	£1.00
"	- AEGDS (games of skill disk) instructions ..	£1.00
Emsoft	- Mouse Art user manual	£1.00
Glentop	- Einstein Assembly Language Course	£2.00
"	- Using LOGO on the Einstein	£2.00
Kuma	- Database User Manual	£1.00
"	- WPRO Reference Manual and User Guide	£1.00
Sercq	- Einstein BASIC Data File Design/Programming	£1.00
Sold	- Relatively BASIC	£1.00
JS	- GrafDraw operating instructions/user guide	£1.00
"	- POPUP desktop organiser user guide	£1.00
"	- WP80 wordprocessor instructions/user guide	£1.00
Supasoft	- EAZIDRAW pcb designer instructions	£1.00
Tatung	- Introduction to the Einstein Computer	£1.00
"	- " " LOGO	£1.00
"	- " " DOS/MOS	£1.00
"	- BASIC Reference Manual	£2.00
"	- BBC BASIC (Z80) Reference Manual	£2.00
"	- TK02 80 column card instructions	£1.00

IF YOU WANT SEVERAL OF THESE, SEND DETAILS OF WHAT YOU WANT
-- WITH A S.A.E. -- FOR A BULK-BUY DISCOUNTED SPECIAL PRICE.

Einstein User Magazine (published by Tatung from Vol.1 No.1 to Vol.3 No.3, then by B&H Computers). Issues available:-
1/1, 1/2, 1/3, 1/4, 2/1, 2/2, 2/3, 2/4; 3/1, 3/2, 3/3, 3/4,
4/1, 4/2, 4/3, 4/4, 5/2. *** ALSO UKEUG Newsletter 2/4 ***
Price (includes postage):- Single copy 50p, or 3 for £1.00.