

July 86

9



UK EINSTEIN USER GROUP  
NEWSLETTER

Issue Number Nine July 1986

EDITORIAL

Number nine already, and somebody said it would not last (I think it was Robby on his way to the airport).

If you all read these editorials (and if you don't then let me know as it takes me at least 3 weeks to think of some thing to write) then you will know about our future exhibition. We will print full details about who will be there in the September issue, in the meantime on with the rest of the news.

EINSTEIN 256

Yes it is official the new Einstein will be announced at the PCW show. The full spec is still secret but here goes:

- 256k Video Ram
- 1 3" internal drive
- at least 256 colours
- High res Graphics
- ( 80 col in COLOUR )
- Upward Compatible
- ( anything you have now will run )
- 2 Joystick ports
- ( one doubles as a printer port )
- RS232 Port ?

We saw one in May but there was no software written specifically for it then so we could not really test it but it did run standard Einstein software ok.

Hopefully the casing has been tarted up a bit as it looked rather bland. We will review it as soon as possible.

Having mentioned the PCW show reminds me that someone will be on the TATUNG stand representing us (my tour of duty will be on sunday) so pop along and say hello.

~~~~~

EINSTEIN USER

As you know the above magazine has a bad track record for being on time ( I know we are always late )but the latest (get it )issue even has an April fool in it.

Admit it, YOU fell for it too.

The article in question is the one about increasing drive 1: to 80 tracks.

What the article is really doing is explaining how to configure Sytem5 to enable it to handle drive 1: as an 80 track drive IF you have a drive which can access 80 tracks. The standard 3" drive CANNOT handle 80 tracks and makes funny noises if you try to make it.

So if you have not already tried it then DON'T.

Leaving that aside there have been some changes for the better!

Karl Dallas has left ( the above was from his last effort) and the magazine is now being produced inhouse, which I have been promised means that it will appear on time. The next issue will be out just before the PCW show with surprise surprise a review on the Einstein 256. There will also be a competition giving away such things as T.V.s ,Holidays and tours of TATUNG'S factory (in Telford not Korea).

~~~~~

IT'S A HARD LIFE

Not so long ago I received a telephone call from A.C.C Computer Services asking me if I would like to review their hard disk system for the Einstein. Being no fool I jumped at the chance thinking they will have to send the Boys round to get back, (they sent Securicor which was good enough). When it arrived I decided to use it on the Einstein at work thereby testing it under real conditions rather than just for a few hours of an evening.

The first surprise was that it was already formatted, this is done at the factory to test the drive, the second was that formatting only takes about two minutes (the early hard drives could take up to an hour).

The unit is roughly 12.5 x 12.5 x 4 inches and has a mains lead and a ribbon cable which plugs into the pipe. The ribbon cable could do with being an inch or two longer as the only place the drive could go was between the computer and the monitor which made the monitor rather high (now that the drive has been returned I find that the monitor is too low) but it would be nice to have the option of putting beside the computer.

The drive is 10MB, that's a 10 with an awful lot of noughts after it, and comes configured as two 5MB drives, the idea being that if you do crash it the chances are you still have one drive full of info which is still accessible. Talking of crashing a certain editor of another magazine did crash this drive, maybe he was testing one of his articles out?

OK having plugged it in we now load the system disk in drive 0: and boot just as we usually do. The usual System Five appears with an additional ACC Hard Disk BIOS message, typing 2: or 3: then logs onto either drive just like any other but BIGGER.

Now being lazy when I use Dbase II I always call the starting .CMD file MENU.CMD that way when I make a new master disk I only have to backup an old one and the autoboot is still valid. So trying to be clever I stored all the relevant files in different user areas and then found that System five is thick and does not support user areas. Anyway having moved all the files about again we were up and running and does it run. The only thing which is faster than a hard drive is a Silicon or Mem disk but these are nowhere near as flexible. When you switch on the hard drive all of your data is still there and with Wordstar you could get over halfway to it's maximum 8meg file size.

After two weeks when it was time to put it back in it's box I came across my first problem, how to arrange all the command, index and data files so that it would run on a twin drive system again, it's true when they say the more space you have the more you fill.

As a home user I can't see any justification for a hard drive unless you want to run a large bulletin board, but as a business user and having used one, although only for a short time I can see how usefull they can be and shall start saving up for one.

PRICE £895.00 plus VAT

A.C.C Computer Services  
8 Water Street  
Abergele  
Clwyd  
LL22 7SH

~~~~~

ADVENTURE

=====

Adventure is known by many other names to adventurers, such as Colossal Cave, Dungeon and Colossal Adventure. These games are all essentially the same one. The game was developed many years ago by Willie Crowther with features added by Don Woods. It has since been adapted for many machines - micros, minis and mainframes - this particular version has been adapted by Jay R. Jaeger and is available from the CP/M USER GROUP.

This group will supply Public Domain Software to it's members and will only charge you the cost of copying your choice of software and postage. I must stress that the group are not distributors of pirated software as many people seem to think, they will only supply software that has been placed in the Public Domain. However, there are hundreds of programs in their library, so they are well worth getting in touch with. But this isn't a CP/M USER GROUP review, so on to the business in hand.....

You commence the game in a well house along with those standard adventurers items - food, water, keys and a lamp. After picking up these items you exit the building to find yourself in a forest. Exploring the forest will eventually reveal a grating through which you can enter the dungeon complex below. The dungeon complex is very big indeed, and there are many puzzles to occupy your mind well into the wee hours.

Many of the problems and locations in this game have been copied and/or adapted in other adventure games, so if you come across anything which seems familiar remember that it probably originated in this game. For example, I have seen a situation involving a wand and a crevice in at least one other game. Another feature of the game which has been borrowed is the use of a magic word to transport yourself from certain locations to the well house and back again. You have to do this periodically in order to cache the treasure you collect in your explorations. Other commands include a HELP command which gives you information about communicating with the game and other bits and pieces, INVENTORY which is self explanatory, and BUILDING which takes you through the dungeons to the well house. This is not the same as the magic word, as it means you must find your way back into the dungeons and through them to your previous location. You can travel to many of the locations in the dungeon in this way. The SCORE command presents you with your score in the following format : TREASURE, SURVIVAL, GETTING WELL IN and SCORE. Personally, I am not too bothered about getting a score for collecting treasure, I much prefer exploring the dungeon.

Although this is a very old game, and therefore has it's limitations (such as a two word input parser), this is a very enjoyable game - a true classic. I recommend it to any adventure game player, it is worth joining the CP/M USER GROUP just to get your hands on this.

I hope to be printing a number of hints and tips about various games in future issues. In order to prevent those of you who don't wish to see these from doing so, I will be giving the hints and tips in code. The letter 'A' will be represented by a 'Z', 'B' by 'X' and so on (I know this isn't very original, but it works). To make it easier to translate these tips you will find a simple translation program somewhere in the pages of this issue. If you have any hints, tips or comments you wish to pass on to other Einstein adventurers, or if you are having problems with any adventure game please send them to me at the following address :

TONY STANSFIELD  
 FLAT 1  
 53 WOOD ROAD  
 WHALLEY RANGE  
 MANCHESTER  
 M16 8BJ

TRANSLATION PROGRAM

```

10 REM Alpha reversal program.
20 REM Free to all members of the UK EINSTEIN USER GROUP.
30 DIM QS(100)
40 CLS
50 PRINT "ALPHA REVERSAL TRANSLATION"
60 PRINT
70 PRINT "WILL NOT TRANSLATE LOWER CASE."
80 PRINT "*****"
90 PRINT
100 REM To allow translation of lower case simply extend
110 REM the variables A$ and Z$ to repeat their lower case values.
120 REM e.g. A$="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
130 REM And change the value 26 on lines 240 and 250 to 52
140 REM NOTE : The speed of this already slow program will be reduced
150 REM still further if it has to search thru the lower case values.
160 A$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
170 Z$ = "ZYXWVUTSRQPONMLKJIHGFEDCBA"
180 INPUT "PLEASE ENTER STRING (OR X TO END)          ";W$
190 IF W$ = "X" THEN GOTO 370
200 L = LEN(W$)
210 FOR I = 1 TO L
220 LET FL = 0
230 OLD$ = MID$(W$,I,1)
240 FOR F = 1 TO 26
250 IF MID$(A$,F,1) = OLD$ THEN QS(I) = MID$(Z$,F,1):LET FL = 1:
    LET F= 26
260 NEXT F
270 IF FL = 0 THEN QS(I) = OLD$
280 NEXT I
290 CLS
300 PRINT "THE TRANSLATION OF "
310 PRINT W$;" IS "
320 FOR M = 1 TO L
330 PRINT QS(M);
340 NEXT M
350 PRINT:PRINT "PRESS ANY KEY";:P$=INCH$
360 GOTO 40
370 CLS
380 END

```

~~~~~

MACHINE CODE PROGRAMMING FOR THE BEGINNER  
 ( or assembly language for the initiated )  
 ( using the BBCBASIC assembler )  
 By CHRIS GILES

This months routine is for all you budding games writers. I expect you have noticed that if you try pressing two or more keys at once then the machine either freezes until they are released or simply picks up on one and ignores the others.

This routine demonstrates multiple key detection and the short routine at the end demonstrates specific key detection. The first routine is put in to illustrate the effect but the second is the one that will probably be the most usefull.

Describing the first one first.

CLEAR SCREEN you should be familiar with by now.

TURN CURSOR OFF is just a CONTROL function initiated by putting the appropriate code to the screen (like PRINT CHR\$(20) in basic). Next,

using the LDIR instruction from last month, we set all our flags to zero so that each pass is made with a clean slate. Next we load the 'D' and 'E' registers with 1. These are the registers used to indicate the row and column numbers we wish to scan. From now on the explanation of the program is in the program notes.

```

120 VCOL=64330
320 LD A,12      ;CLEAR SCREEN
330 RST 8
340 DEFB 9EH
350 LD A,20      ;TURN CURSER OFF
360 RST 8
370 DEFB 9EH
380 CLEAR:LD BC,7 ;8 FLAGS TO CLEAR
390 LD HL,FLAGS  ;POINT TO FIRST
400 LD (HL),0    ;SET TO ZERO
410 LD DE,FLAGS+1 ;POINT TO NEXT POSITION
420 LDIR        ;EXPLANATION LAST MONTH
430 LD D,1      ;SEE TEXT FOR EXPLANATION
440 LD E,1      ;OF THESE TWO LINES
450 CKS:CALL KS;
460 JR Z,ONE    ;RETURNS Z IF KEY PRESSED
470 ZERO:LD A,30H ;NO KEY PRESSED SO LOAD 'A' WITH CHR '0'
480 JR PRIN    ;SKIP THE NEXT LINE
490 ONE:LD A,31H ;KEY PRESSED SO LOAD 'A' WITH CHR '1'
500 PRIN:PUSH AF ;SAVE CHR TO PRINT
510 LD B,8     ;COUNTER
520 LD A,1     ;CALCULATOR
530 LOOP:CP D
540 JR Z,GPH   ;HORIZONTAL POSITION
550 DEC B     ;DECREASE COUNTER
560 ADD A,A   ;MULTIPLY BY 2
570 JR NC,LOOP ;IF NOT DONE 8 TIMES
580 GPH:LD L,B ;SAVE POSITION
590 LD B,8    ;ROWS NOW
600 LD A,1
610 LOOP1:CP E
620 JR Z,GPV
630 DEC B
640 ADD A,A
650 JR NC,LOOP1
660 GPV:LD H,B ;SAVE POSITION
670 POP AF    ;GET 'A' TO TEST TO SEE IF WE WANT TO SET FLAGS
680 PUSH AF   ;SAVE IT AGAIN TO PRINT LATER
690 CP 30H    ;DO WE WANT TO SET THE FLAGS?
700 JR Z,OK   ;IF NOT JUMP TO OK
710 LD BC,FLAGS ;GET THE FIRST FLAG
720 LD A,C    ;GET LOW BYTE IN 'A'
730 ADD A,L   ;ADD ROW NUMBER
740 LD C,A   ;PUT IT BACK (NOW WE HAVE A FLAG FOR EACH ROW)
750 LD A,(BC) ;GET THE FLAG
760 OR H     ;TURN THE 'BIT' ON
770 LD (BC),A ;PUT THE FLAG BACK
780 OK:LD A,H ;GET ROW NUMBER
790 ADD A,A   ;DOUBLE IT
800 LD H,4   ;START AT ROW 4
810 ADD A,H   ;ADD TOGETHER
820 LD H,A   ;STORE IT BACK
830 LD A,L   ;DO THE
840 ADD A,A   ;SAME THING
850 LD L,10  ;WITH
860 ADD A,L   ;THE

```

```

870 LD L,A          ;COLUMNS
880 LD (VCOL),HL    ;POSITION CHARACTER ON SCREEN
890 POP AF          ;GET CHR TO PRINT
900 RST 8           ;PRINT IT
910 DEFB 9EH        ;(AN OLD ROUTINE NOW)
920 LD A,E          ;GET ROW JUST SCANNED
930 ADD A,A         ;INC SCAN ROW
940 LD E,A          ;PUT IT BACK
950 JR NC,CKS       ;IF LESS THAN 8 THEN CONTINUE TO LOOP
960 LD E,1          ;RESET ROW
970 LD A,D          ;GET COLUMN JUST SCANNED
980 ADD A,A         ;INCREASE IT
990 LD D,A          ;PUT IT BACK
1000 JR NC,CKS      ;IF LESS THAN 8 THEN CONTINUE TO LOOP
1010 LD HL,FLAGS+4 ;POINT TO A PREDETERMINED FLAG
1020 LD A,(HL)      ;GET IT TO SEE IF IT EQUATES TO 00001111 BINARY
1030 CP 0FH         ;IF SO FOUR SET KEYS PRESSED AT THE SAME TIME
1040 RET Z          ;SO RETURN TO BASIC
1050 JP CLEAR       ;OTHERWISE DO IT ALL OVER AGAIN
1060 KS:LD A,14     ;SET UP THE PSG
1070 OUT(2),A       ;TO READ THE KEYBOARD
1080 LD A,E         ;SEND THE
1090 CPL            ;COLUMN NUMBER
1100 OUT(3),A       ;TO THE PSG
1110 LD A,15        ;THIS IS THE COMMAND
1120 OUT(2),A       ;TO TELL IT TO SCAN THE COLUMN
1130 IN A,(2)       ;THIS IS THE RESULT
1140 AND D          ;SET ZERO FLAG IF ROW IS THE SAME AS 'D'
1150 RET
1160 FLAGS:DEFS 8

```

The second routine illustrates specific keyboard detection. By fixing the values in D & E before calling the keyboard scan we can look for one key, for example 'Q'. It will not matter if the Alpha Lock is on or off, neither will it matter if any other key is pressed. If the 'Q' key is pressed it will return from the routine with the zero flag set. To use this in a program of your own look for the 'Q' key and if it is pressed then on return from the routine put in a line such as

```
CALL Z,QROUTINE
```

Then set the D & E registers with the coordinates of (say) the 'P' key and scan for that followed by

```
CALL Z,PROUTINE
```

In this way your program will respond to any combination of keys.

```

120 VCOL=64330
320 LD A,12
330 RST 8
340 DEFB 158
350 LD D,1
360 LD E,1
370 .CKS V KS
380 JR NZ,CKS
390 RET
400 .KS LD A,14
410 OUT(2),A
420 LD A,E
430 CPL
440 OUT(3),A
450 LD A,15
460 OUT(2),A
470 IN A,(2)
480 AND D
490 RET

```

Now look at the table at the end of the article. The numbers in the left hand column represent the value in the 'E' register and the numbers along the bottom represent the values in the 'D' register. Where the column and row cross you will find the key that those particular values will detect. So try changing them and check to see if all the grid is correct????

128	F6	Z	X	C	V	B	N	M
64	F1	A	S	D	F	G	H	J
32	F2	Q	W	E	R	T	Y	U
16	F3	1	2	3	4	5	6	7
8	F4	^	=		8	/	.	,
4	F5	9		]	:	;	L	K
2	0			[	P	O	I	
1	ESC	SP	ENT	ALP	F7	F0		
	1	2	4	8	16	32	64	128

- 1,1 IS THE ESC KEY
- 1,2 IS THE SPACE BAR
- 1,4 IS THE ENTER KEY
- 1,8 IS THE ALPHA LOCK KEY
- 1,64 HAS NO KEY ASSOCIATED WITH IT.

~~~~~

JAMES COYLE,  
 7 COCHRANE STREET,  
 FALKIRK,  
 STIRLINGSHIRE,  
 SCOTLAND.  
 FK1 7QB.

Dear Keith,

Firstly let me thank you for producing such an interesting newsletter. I've been a reader since the first issue and have found it to be of great value. You and your contributors have helped me with various problems on more than one occasion.

So I've decided to become a contributor myself and have enclosed my offering for your perusal.

The original program from which I got the basic idea, was printed in a magazine called Personal Software in 1982 and written by someone called Kit Brown. However I've had to change the program so much that I doubt if it will incur any copyright problems. The original was 130 Bytes in length whereas this is 139 Bytes. Also the philosophy of the routine is somewhat different as the original was intended to run on a NascomII machine whose organisation of Basic is also quite different.

The Assembly listing is fully documented and will give little problems (I hope).

The idea behind the routine is to remove all REM statements and unwanted spaces in a Basic program. In effect, it will COMPACT it - hence the name. I have tried the routine on some lengthy Basic programs and have seen some quite considerable savings in space and thus an increase in execution times.

For example, on the Master disc with programs such as "KEYBOARD" AND "MASTERMIND" I have had space savings of up to 28% which obviously means a dramatic increase in speed of execution. Unfortunately, these programs will not RUN after using the compacter as the authors of said programs have some of the GOSUBS and GOTOS referring to lines with a REM at the head. So it will be necessary to change the problem lines to the correct place - the line after the REM! It is supposed to be done like this anyway! Also in the KEYBOARD program there is an error at line 2640 (the inverted comma is missing from the end of the line) and errors such as this will corrupt the end result of the routine I'm afraid. Effectively, the end of a string is not detected by the routine, causing all kinds of problems and gnashing of teeth. I'm down to the gums now!

There are another couple of points of interest to those of you wishing to modify the routine.

The only machine dependant instructions are these:-

&DE00 LD HL,&44A1 This refers to the start of Basic with the System 5 Basic installed. Check if this is different with PTR(0) from Basic and replace.

&DE7E LD(ENDPROG),DE This refers to the pointer in Basic that holds the End of Program address. In the System 5 Basic, this is at &0150. It will be necessary for you to find this if you are using something different. See the TRACE routine.

&DE86 LD(END1),DE These refer to addresses &0152/3 and &0154/5.  
&DE87 LD(END2),DE which hold the pointers PTR(15) and PTR(16) respectively. There should be a differential of 16 bytes between these and PTR(14).

I've been using the XED - XSM facility from System 5 and have assembled the routine using this. If you have no assembler then go into MOS and use the (M)odify command to load this into &DE00 onwards. Now into Basic and use SAVE "COMPACT.OBJ",&DE00,&DE8C. You should now program a small routine in Basic to use this.

e.g. 10 CLEAR &DE00:LOAD "COMPACT.OBJ":NEW

This will load the compact routine and "NEW" itself. If your using an assembler then you will know how to make up an OBJECT file for Basic to use. See page 177 of the introductory manual. I have no good reason for using &DE00 as the start of the routine, it was the first 'big' number I thought of to keep clear of any Basic I might use. Anyway you should now have this routine in memory ready to be called (CALL &DE00). I should say that it is imperative that you keep a good copy of your Basic program complete with REM's in case anything goes wrong. You do always make backups don't you!

I hope this will be of use to other Einstein users and I would appreciate their comments and news of modifications.

The next routine is much simpler but just as useful. Its a simple trace where a two byte number is supplied and the whole of memory is checked for any occurrence of it. I used it to find all and any occurrence of the end of Basic address in the COMPACT routine so it could be used by users not having System 5 Basic.

This routine is also documented on the assembler listing and should cause no problems. To find the occurrence of a number other than PTR(14) then change the value of X in line 40 of the Basic prog.

Now to other matters.

I have an interest in Adventure programs and have bought the INFOCOM game "PLANETFALL". I have amassed some 56 points, about 80% but I'm now well and truly stuck! Can anyone tell me how to get past the MURAL or where is the RADIATION SUIT or the REACTOR ELEVATOR ACCESS CARD?

In the first issue of the BRAIN the listing for the TENNIS game is in error. Line 328 has the hex code &84 missing from the mnemonic code 1.st. Also the list given for the DATA codes has a few bytes missing. It may be better to check through the hex codes along with the assembly list and add where necessary. See lines 386,388,400, 401,403 and 405. It looks like it has been a printer error.

BASIC TRACE LISTING.

```
10 CLEAR &7550:LOAD "TRACE.OBJ"
20 CLS
30 X=PTR(14):REM FOR END OF PROGRAM
40 DOKE &7550,X
50 CALL &7530
```

TRACE ASSEMBLY LISTING

```
*****
*
*                               *
*          TRACE JUNE 1986      *
*                               *
*          J.COYLE              *
*                               *
*****
```

```
7530 =          ORG 7530H

7550 =          HIBYTE:EQU 7550H
7551 =          LOBYTE:EQU 7551H

7530 01 FF FF          START:LD BC,OFFFHH ; SET COUNT
7533 21 00 01          LD HL,0100H ; SET START POINT
7536 3A 50 75          CMPR: LD A,(HIBYTE); GET HIGH BYTE
7539 BE              CP (HL) ; IS IT SAME AS HL?
753A 20 09          JR NZ,NEXT ; NO THEN JUMP THIS BIT
753C 23              INC HL ; YES POINT TO NEXT BYTE
753D 0B              DEC BC ; DECREASE COUNTER
753E 3A 51 75          LD A,(LOBYTE); GET LOW BYTE
7541 BE              CP (HL) ; IS IT SAME AS HL?
7542 CC 49 75          CALL Z,PRINT ; YES PRINT LOCATION
7545 23              NEXT: INC HL ; INCREASE TAKE POINTER
7546 10 EE          DJNZ CMPR ; DECREASE COUNT AND
                          ; REPEAT UNTIL BC=ZERO
7548 C9              STOP: RET ; RETURN TO BASIC
```

```

7549 2B          PRINT:DEC HL          ; GO BACK TO HIGH BYTE
754A CF          RST 08                ; SYSTEM RESTART FROM MOS
754B A9          DEFB 0A9H            ; PRINT 4 HEX DIGITS HELD IN HL
754C CF          RST 08                ; SYSTEM RESTART FROM MOS
754D A8          DEFB 0A8H            ; PRINT A SPACE
754E 23          INC HL                ; SET HL TO NEXT LOCATION
754F C9          RET                  ; RETURN TO CALLING ROUTINE
7550             END

```

COMPACT ASSEMBLY LISTING

```

*****
*
*          COMPACT JUNE 1986
*
*          J.COYLE FROM AN ORIGINAL
*          BY KIT BROWN 1982.
*
*****

```

```

DE00             ORG 0DE00H

0150             ENDPORG:EQU 0150H ;END OF PROGRAM POINTER ADDRESS
0152             END1  :EQU 0152H ;END OF VARIABLE SPACE
0154             END2  :EQU 0154H ;END OF ARRAY SPACE

DE00 21 A1 44    START:LD HL,44A1H ; START OF PROG WITH SYSTEM 5 BASIC
DE03 E5          PUSH HL           ; SAVE THE START IN
DE04 C1          POP BC            ; REGISTER BC
DE05 23          INC HL            ; JUMP OVER THE NEXT FOUR BYTES
DE06 23          INC HL            ; THESE HOLD THE LENGTH OF LINE
DE07 23          INC HL            ; AND THE LINE NUMBER.
DE08 23          INC HL
DE09 E5          PUSH HL           ; LOAD THE FIRST PROGRAM LOCATION
DE0A D1          POP DE            ; INTO DE
DE0B 7E          AGAIN:LD A,(HL)   ; GET THE NEXT BYTE
DE0C FE 20       CP 20H            ; IS IT A SPACE?
DE0E CC 70 DE    CALL Z,L 3TH;YES THEN REDUCE THE LENGTH OF LINE
DE11 28 0E       JR Z,SPA          ; AND JUMP TO SPACE TO DELETE IT.
DE13 FE 22       CP 22H            ; IS IT A STRING?
DE15 28 0D       JR Z,STRING       ; YES THEN JUMP TO STRING TO KEEP IT
DE17 FE A4       CP 0A4H           ; IS IT A REM?
DE19 28 13       JR Z,REM          ; YES THEN JUMP TO REM TO DELETE IT.
DE1B FE 00       REMADDED:CP 00H   ; IS IT THE END OF THE LINE?
DE1D 28 3B       JR Z,ENDLINE; YES THEN JUMP TO ENDLINE TO
                ; RELOCATE.
DE1F 12          PUTIT:LD (DE),A    ; LOAD BYTE INTO NEXT PUT LOCATION.
DE20 13          INC DE            ; POINT TO NEXT PUT LOCATION.
DE21 23          SPACE:INC HL       ; POINT TO NEXT TAKE LOCATION.
DE22 18 E7       JR AGAIN          ; GO BACK AND DO IT AGAIN.
DE24 12          STRING:LD (DE),A  ; PART OF A STRING SO KEEP IT.
DE25 13          INC DE            ; POINT TO NEXT PUT LOCATION.
DE26 23          INC HL            ; POINT TO NEXT TAKE LOCATION.
DE27 7E          LD A,(HL)         ; TAKE THE NEXT BYTE.
DE28 FE 22       CP 22H            ; IS IT END OF THE STRING YET?
DE2A 28 F3       JR Z,PUTIT        ; YES THEN PUT IT BACK.
DE2C 18 F6       JR STRING         ; NO THEN CONTINUE WITH STRING.
DE2E C5          REM:PUSH BC        ; SAVE ALL THE REGISTERS
DE2F D5          PUSH DE
DE30 E5          PUSH HL

```

```

DE31 D9                EXX                ; EXCHANGE THEM
DE32 E1                POP HL
DE33 D1                POP DE
DE34 C1                POP BC                ; LOAD ALTERNATE REGISTERS.
DE35 23                CPOINT:INC HL        ; NEXT TAKE LOCATION.
DE36 7E                LD A,(HL)          ; TAKE THE NEXT BYTE.
DE37 FE 00            CP 00H              ; IS IT THE END OF LINE?
DE39 28 05            JR Z,STEP          ; YES SO JUMP TO STEP.
DE3B CD 70 DE        CALL LENGTH        ; NO THEN REDUCE LINE LENGTH.
DE3E 18 F5            JR CPOINT          ; GO AND DO IT AGAIN.
DE40 D9                STEP: EXX           ; RECALL ORIGINAL REGISTERS.
DE41 CD 70 DE        CALL LENGTH        ; REDUCE LINE LENGTH.
DE44 2B                DEC HL              ; POINT TO LAST BYTE BEFORE REM.
DE45 7E                LD A,(HL)          ; TAKE IT.
DE46 FE 3A            CP 3AH              ; IS IT A COLON (:)?
DE48 20 08            JR NZ,DLINE        ; NO THEN JUMP OVER NEXT BIT.
DE4A AF                XOR A              ; MAKE A=ZERO.
DE4B D9                EXX                ; RESTORE REGISTERS.
DE4C CD 70 DE        CALL LENGTH        ; REDUCE LINE LENGTH.
DE4F 1B                DEC DE              ; POINT DE AT (:) TO DELETE IT.
DE50 18 C9            JR REMADED        ; REM APPENDED TO PROGRAM LINE.
DE52 D9                DLINE:EXX          ; RESTORE REGISTERS.
DE53 23                INC HL              ; THIS SETS UP HL AND DE TO THE
DE54 1B                DEC DE              ; CORRECT PLACE FOR DELETION OF
DE55 1B                DEC DE              ; A COMPLETE LINE INCLUDING LINE
DE56 1B                DEC DE              ; NUMBER AND LINE LENGTH DATA.
DE57 1B                DEC DE              ;
DE58 18 03            JR MOVE            ; JUMP TO MOVE
DE5A 12                ENDLINE:LD (DE),A    ; PLACE THE LAST BYTE.
DE5B 13                INC DE              ; POINT TO NEXT PUT LOCATION.
DE5C 23                INC HL              ; POINT TO NEXT TAKE LOCATION.
DE5D D5                MOVE:PUSH DE        ; SAVE DE LOCATION FOR NEXT LINE.
DE5E 01 04 00        LD BC,04H          ; SET UP A COUNTER.
DE61 ED B0            LDIR              ; TRANSFER HL TO DE AND INCREMENT.
                        ; BOTH TO GET TO NEXT PROGRAM BYTE.
DE63 C1                POP BC              ; SAVE LINE LENGTH LOCATION IN BC.
DE64 3A 50 01        LD A,(ENDPROG)    ; LOW BYTE OF END OF PROGRAM.
DE67 C6 02            ADD A,02H          ; ADD 2 FOR DISPLACEMENT IN HL
DE69 BD                CP L              ; ARE WE AT END OF PROGRAM?
DE6A 28 0A            JR Z,HBIN          ; YES THEN CHECK HIGH BYTE.
DE6C 7E                LBIN:LD A,(HL)     ; NO GET NEXT TAKE BYTE.
DE6D 12                LD (DE),A          ; LOAD IT TO NEXT PUT LOCATION.
DE6E 18 9B            JR AGAIN          ; GO AND DO IT ALL AGAIN.
DE70 F5                LENGTH:PUSH AF     ; SAVE REGISTER A AND FLAGS.
DE71 0A                LD A,(BC)          ; LOAD A WITH LENGTH OF LINE
DE72 3D                DEC A              ; REDUCE IT BY ONE.
DE73 02                LD (BC),A          ; LOAD NEW LINE LENGTH.
DE74 F1                POP AF            ; RESTORE REGISTER
DE75 C9                RET              ; GO BACK TO CALLING ROUTINE.
DE76 3A 51 01        HBIN:LD A,(ENDPROG+1); GET HIGH BYTE
DE79 BC                CP H              ; ARE WE AT END OF PROGRAM?
DE7A 20 8F            JR NZ,AGAIN        ; NO THEN BACK AGAIN.
DE7C ED 53 50 01    LD (ENDPROG);DE; ; LOAD NEW END OF PROGRAM.
DE80 21 10 00        LD HL,10H          ; ADD DISPLACEMENT TO HL
DE83 19                ADD HL,DE          ; HL=DE+10
DE84 22 52 01        LD (END1),HL      ; LOAD POINTERS
DE87 22 54 01        LD (END2),HL      ; LOAD POINTERS
DE8A C9                RET              ; ALL DONE BACK TO BASIC.
                        END

```

I hope this has been of interest to someone and even useful perhaps.

```

0 TCOL 15,10: BCOL 10
10 CLS
20 PRINT " *****"
30 PRINT " * *"
40 PRINT " * P L A N E T S *"
50 PRINT " * *"
60 PRINT " * A programme for the U.K. *"
70 PRINT " * *"
80 PRINT " * Einstein User Group *"
90 PRINT " * *"
100 PRINT " * by *"
110 PRINT " * *"
120 PRINT " * P.W.H. Moon *"
130 PRINT " * *"
140 PRINT " * Version 1.1 28 May 1986 *"
150 PRINT " * *"
160 PRINT " * *"
170 PRINT " * (Xtal BASIC) *"
180 PRINT " * *"
190 PRINT " *****"
200 :
210 FOR I=1 TO 2500:NEXT I
220 CLS
230 REM SCREEN MESSAGE
240 TCOL 3,6: BCOL 6
250 CLS
260 PRINT " PLANETARY MOTION"
270 PRINT " "
280 PRINT
290 PRINT " To a reasonably close approximation,"
300 PRINT "the planets all revolve in circles with"
310 PRINT "the Sun as their centre. This makes"
320 PRINT "them appear to move slowly in the same"
330 PRINT "direction at different speeds against"
340 PRINT "the background of fixed stars in the"
350 PRINT "night sky as seen from Earth."
360 PRINT
370 PRINT " Because of the Earth's own movement"
380 PRINT "round the Sun, the apparent motions of"
390 PRINT "the planets are not steady; at times"
400 PRINT "they reverse their apparent direction"
410 PRINT "of movement relative to the fixed stars"
420 PRINT "as seen from Earth."
430 PRINT
440 PRINT " This programme plots the apparent"
450 PRINT "position of any planet as seen from"
460 PRINT "Earth over a period of time."
470 PRINT
480 PRINT " Press space bar to go on."
490 A$=INCH$
500 CLS
510 FOR I=1 TO 3:PRINT:NEXT I
520 PRINT "The scale of the display is shown by"
530 PRINT "a bar 1 Astronomical Unit (A.U.) in"
540 PRINT "length, which is the mean size of the"
550 PRINT "Earth's orbital radius."
560 PRINT
570 PRINT "If you include on the system tracks"
580 PRINT "the screen dump routine published in"
590 PRINT "Newsletter No.1 (page 24), you can"
600 PRINT "print the result."
610 PRINT

```

```

620 PRINT "WARNING:- The programme will crash if"
630 PRINT "you ask for a dump but don't have the"
640 PRINT "machine code for the screen dump"
650 PRINT "on your system tracks."
660 PRINT:PRINT:PRINT
670 PRINT "      Press space bar to go on."
680 AS=INCH$
690 CLS
700 TCOL 1,3: BCOL 3
710 CLS
720 REM EARTH ORBIT RADIUS, ANGULAR VELOCITY,
      AND PHASE ANGLE FOR STARTING PLOT.
730 RE=149.6: OMEGE=0.0172024: EPS=1.5
740 DATA "Mercury",0.071408,57.95,2.2,2
750 DATA "Venus",0.027963,108.11,2.7,4
760 DATA "Mars",0.0091461,227.84,4.0,6
770 DATA "Jupiter",0.0014503,778.14,10,10
780 DATA "Saturn",0.00058940,1427.0,17,20
790 DATA "Uranus",0.00020467,2870.3,32,30
800 DATA "Neptune",0.0001043920,4999.9,64,45
810 DATA "Pluto",0.0000972667,5909,64,70
820 PRINT "      M E N U "
830 PRINT "      _____"
840 PRINT
850 PRINT "(1) Mercury"
860 PRINT "(2) Venus"
870 PRINT "(3) Mars"
880 PRINT "(4) Jupiter"
890 PRINT "(5) Saturn"
900 PRINT "(6) Uranus"
910 PRINT "(7) Neptune"
920 PRINT "(8) Pluto"
930 PRINT
940 PRINT
950 PRINT "      or exit from the programme (9)"
960 PRINT:PRINT:PRINT
970 INPUT "Input a number for your choice ";CHOICE
980 IF CHOICE = 9 THEN 1400
990 IF CHOICE > 0 AND CHOICE <9 THEN 1000ELSE 970
1000 PRINT:PRINT
1010 PRINT "FOR HOW MANY DAYS WOULD YOU LIKE THE"
1020 INPUT "SIMULATION TO RUN (DEFAULT 1500)";NDAYSS$
1030 IF NDAYSS$ <> "" THEN 1060
1040 NDAYSS=1500
1050 GOTO 1070
1060 NDAYSS=VAL(NDAYSS$)
1070 PRINT
1080 PRINT "DO YOU WANT A SCREEN DUMP AFTER THE DISPLAY ?"
1090 INPUT "ENTER Y OR N AND READY PRINTER IF Y ";YNS$
1100 GCOL 15,4 : BCOL 4
1110 TCOL 15,4
1120 CLS
1130 ORIGIN 120,96
1140 DRAW -5,0 TO 5,0
1150 DRAW 0,-5 TO 0,5
1160 PRINT@ 22,12,"E"
1170 FOR I= 1 TO CHOICE
1180 READ PLANETS$,OMEGP,RP,SCALEF,S
1190 NEXT I
1200 RESTORE
1210 AU=RE/SCALEF
1220 PRINT@ 30,1;PLANETS$

```

```

1230 DRAW 120-AU,-93 TO 120,-93
1240 DRAW 120-AU,-96 TO 120-AU,-90
1250 DRAW 120,-96 TO 120,-90
1260 PRINT@32,22;"1 A.U."
1270 RE=RE/SCALEF: RP=RP/SCALEF
1280 FOR DAY=1 TO NDAYS STEP S
1290 X=RP*COS(OMEGP*DAY+EPS) -RE*COS(OMECE*DAY)
1300 Y=RP*SIN(OMEGP*DAY+EPS) - RE*SIN(OMECE*DAY)
1310 REM CORRECT FOR SCREEN ASPECT RATIO
1320 X=0.75*X
1330 PLOT X,Y
1340 NEXT DAY
1350 IF YN$<>"Y" THEN 1370
1360 CALL &E270
1370 PRINT @ 1,22;"<SPACE> TO RETURN TO MENU"
1380 A$=INCH$
1390 GOTO 690
1400 TCOL 15,4
1410 END

```

~~~~~  
DRIVE PATCH

WARNING - ONLY TO BE USED WITH DOS 1.31

-----  
(If you still have DOS 1.11 it's time you complained to Tatung).

This note is for users of CP/M software who can never remember whether a particular program uses its own f.c.b and requires an alphabetic drive specification or whether it uses the default f.c.b and requires numerical drive. The change suggested allows DOS to accept either numerical or alphabetic drives. There is no change in the display: for example the prompt is still numerical. The command DRIVE is not altered and still accepts only numbers - however it is possible to change the default drive by entering 1: or B: or A: as appropriate.

When the patch is installed you can run all CP/M programs by specifying only alphabetic drives. Einstein programs which process file names are not affected and if you want CP/M programs to accept numerical drives you will have to patch them individually.

The original code uses SUB 47: I have removed this and inserted

```

E553 D6 40          SUB      64          ;adjust alpha input
E555 30 02         JR       NC,E559H    ;test for number
E557 C6 11         ADD      17          ;adjust numeric input

```

Instructions  
-----

The patch is in file ALPHCODE.HEX. If you don't know what a HEX file is enter MOS and type:

```

R 0500 06FF 0200 0
M 0535
>type in the modified code in file ALPHCODE.ASM (between the
lines)
W 0500 06FF 0200 0
T 0530 0577
<CNTRL-A>

```

The last instruction will allow you to check the printout to ensure that the code has been entered correctly (or hang the machine if you haven't got a printer).

C.P. WALLIS

Original code	Modified code
E52D AF	E52D AF
E52E 21 1C EB	E52E 21 1C EB
E531 CD 13 E5	E531 CD 13 E5
E534 E5	E534 E5

E535 AF	
E536 32 3E EB	
E539 ED 5B 8A E3	E535 ED 5B 8A E3
E53D CD 01 E5	E539 CD 01 E5
E540 ED 53 8C E3	E53C ED 53 8C E3
E544 28 0A	E540 28 08
E546 D6 2F	
E548 47	E542 47
E549 13	E543 13
E54A 1A	E544 1A
E54B FE 3A	E545 FE 3A
E54D 28 08	E547 28 09
E54F 1B	E549 1B
E550 3A 3D EB	E54A 3A 3D EB
E553 3C	E54D 3C
E554 77	E54E 77
	E54F AF
E555 18 06	E550 18 09
E557 78	E552 78
	E553 D6 40
	E555 30 02
	E557 C6 11
E558 32 3E EB	
E55B 70	E559 77
E55C 13	E55A 13
	E55B 32 3E EB
E55D 06 08	E55E 06 08
E55F CD 84 E5	E560 CD 84 E5
E562 CD 7F E5	E563 CD 7F E5
E565 23	E566 23
E566 36 00	E567 36 00
E568 10 FB	E569 10 FB
E56A ED 53 8A E3	E56B ED 53 8A E3
E56E E1	E56F E1
E56F 01 0B 00	E570 0E 0B

E572 23	E572 23
E573 7E	E573 7E
E574 FE 3F	E574 FE 3F
E576 20 01	E576 20 01
E578 04	E578 04
E579 0D	E579 0D
E57A 20 F6	E57A 20 F6
E57C 78	E57C 78
E57D B7	E57D B7
E57E C9	E57E C9

~~~~~

BUGWATCH REPORT

Program: DISCMOD

Distributor: Kuma Computers Ltd.

Machine: Tatung Einstein

Problem: Control characters are echoed to the screen following entry of hexadecimal values less than 20H. With some values there is no problem, but others are more tiresome. For example to complete the writing of 01H (screen dump code) one needs to borrow a printer.

Action: I wrote to Kuma asking for assistance, and after two months I received an offer to help me "get used to it".

Elimination: A patch is enclosed which gets round the problem and may be helpful to your readers in the meantime until Kuma pulls up its socks. The programming is somewhat 'dirty' but it does avoid the need for relocating code.

Background: For the benefit of anyone not familiar with it, DISCMOD is a direct disc editor. It provides both a character display and a hexadecimal display and data can be entered in either format except for control characters which must be entered in hexadecimal. I use it mainly for altering the lengths of files, because the Einstein DOS has a habit of rounding up the number of sectors.

C. P. Wallis

Modification of DISCMOD

All addresses are hexadecimal bytes when loaded at 0100H

| Original |    | Modified |    |
|----------|----|----------|----|
| 03E9     | 4F | 03E9     | 2F |
| 03EA     | 3E | 03EA     | C6 |
| 03EB     | 18 | 03EB     | 19 |
| 03EC     | 91 | 03EC     | 47 |
| 03ED     | 47 | 03ED     | 4F |
| 03EE     | 4F | 03EE     | 3E |
| 03EF     | 3E | 03EF     | 04 |
| 03F0     | 04 | 03F0     | CD |
| 03F1     | CD | 03F1     | 03 |
| 03F2     | 03 | 03F2     | 04 |
| 03F8     |    | 03F8     | 7B |
| 0579     | E5 | 0579     | 06 |
| 057A     | 06 | 057A     | 08 |
| 057B     | 08 | 057B     | E5 |

~~~~~

Here is a label printing program written in MBASIC but should run under XBASIC with a few changes.

```

1000 '           Easy Label Written by Nigel C H Beaumont
1010 '           =====
1020 '           The Copywrite of this program is retained
1030 '           by Nigel C H Beaumont but is released for
1040 '           non-profit making distribution provided
1050 '           that this notice is not removed. March 86
1060 '
1070 '
1080 '           Taylor these next to fit the labels
1090 '           =====
1100 '
1110 LABWIDTH=30 ' Maximum label line length
1120 LABLINES=6  ' Maximum printable lines per label
1130 LABHEIGHT=9 ' The actual number of lines from label to label
1140 DIM ADDRESS$(LABHEIGHT)
1150 '
1160 '           Next set up a clear screen & banner string
1170 '           =====
1180 HME$=CHR$(27)+"*" ' Clear screen and home the cursor string
1190 HME$=HME$+"      "
1200 HME$=HME$ + "++++ E A S Y   L A B E L   +++++"
1210 UP$=CHR$(11)      ' Cursor up string
1220 WIDTH 255
1230 '
1240 '           Find format details from user
1250 '           =====
1260 PRINT HME$
1270 PRINT:PRINT
1280 INPUT"Do you want the text centred ",A$
1290 IF A$="y" OR A$="Y" THEN CENTRE = NOT 0 ELSE CENTRE = 0
1300 PREP$=STRING$(35-LABWIDTH/2," ")
1310 MASK$=CHR$(124) + STRING$(LABWIDTH,"\") + CHR$(124)
1320 MASK$= MASK$ + STRING$(LABWIDTH+1,8)
1330 '
1340 '           Get the label details from the user
1350 '           =====
1360 '
1370 PRINT HME$
1380 PRINT:PRINT
1390 PRINT "           Use $$$ to represent the label number ";
1400 PRINT "and &&& the total quantity"
1410 PRINT
1420 FOR COUNT = 1 TO LABLINES
1430 GOSUB 2540
1440 IF A$="" THEN LINES=COUNT-1 : GOTO 1470
1450 NEXT
1460 LINES = LABLINES
1470 FOR COUNT = LINES + 1 TO LABHEIGHT
1480 ADDRESS$(COUNT) = " "
1490 TAG(COUNT)=0
1500 NEXT
1510 '
1520 '           Show him what he has done
1530 '           =====
1540 '
1550 PRINT HME$
1560 PRINT:PRINT:PRINT:PRINT
1570 FOR COUNT = 1 TO LINES

```

```

1580 PRINT PREP$ COUNT " " CHR$(124) ADDRESS$ (COUNT) CHR$(124)
1590 NEXT
1600 PRINT:PRINT
1610 INPUT "Is this OK ",A$
1620 IF (A$ = "Y") OR (A$ = "y") THEN 1850
1630 '
1640 '      Sort out all the cock-up's
1650 '      =====
1660 '
1670 PRINT
1680 INPUT "Edit which line number (RETURN for all) ",A$
1690 ED = VAL(A$)
1700 IF ED = 0 THEN 1370
1710 PRINT HME$
1720 PRINT:PRINT:PRINT:PRINT
1730 PRINT "      Use $$$ to represent the label number ";
1740 PRINT "and && the total quantity"
1750 PRINT
1760 FOR COUNT = 1 TO LINES ' Display the data as is
1770 PRINT PREP$ COUNT " " CHR$(124) ADDRESS$ (COUNT) CHR$(124)
1780 NEXT
1790 FOR COUNT = LINES TO ED STEP -1 ' Then back up to error line
1800 PRINT UP$;
1810 NEXT
1820 COUNT=ED
1830 GOSUB 2540 ' Refill the error line
1840 GOTO 1510 ' & go back to see if it's ok
1850 '
1860 '      Move lines down label to balance
1870 '      =====
1880 '
1890 MOVE = INT((LABLINES-LINES)/2)
1900 IF MOVE=0 THEN 1970
1910 FOR COUNT=LINES+MOVE TO MOVE STEP -1
1920 ADDRESS$(COUNT)=ADDRESS$(COUNT-MOVE)
1930 NEXT COUNT
1940 FOR COUNT = 1 TO MOVE
1950 ADDRESS$(COUNT) = " "
1960 NEXT COUNT
1970 '
1980 '      Find quantity required
1990 '      =====
2000 '
2010 PRINT:PRINT
2020 PRINT
2030 INPUT "How many labels required ";NUMBER
2040 NUMBER = INT(NUMBER)
2050 IF NUMBER < 1 THEN END
2060 IF NUMBER < 1000 THEN 2090
2070 PRINT "Thats a lot of labels "
2080 GOTO 2030
2090 '
2100 '      Fill in any &&s and tag any $$$s in label
2110 '      =====
2120 '
2130 NUMBER$ = RIGHT$(" "+STR$(NUMBER),3)
2140 FOR COUNT = 1 TO LABLINES
2150 POSITION = INSTR(ADDRESS$(COUNT),"&&")
2160 IF POSITION <> 0 THEN MID$(ADDRESS$(COUNT),POSITION,3) = NUMBER$
2170 TAG(COUNT) = INSTR(ADDRESS$(COUNT),"$$$")
2180 NEXT COUNT
2190 '

```

```

2200 '      Set up the printer
2210 '      =====
2220 '
2230 LPRINT CHR$(27);"O";      ' Cancel Perforation skip
2240 LPRINT CHR$(18);         ' Cancel Condensed
2250 LPRINT CHR$(27);"E";     ' Set Emphasized
2260 LPRINT CHR$(27);"8";     ' Cancel NO PAPER Error
2270 LPRINT CHR$(27);"R";CHR$(3); ' Set U.K. Char Set
2280 '
2290 '      Print a sample for alignment
2300 '      =====
2310 '
2320 FOR LINENUM = 1 TO LABHEIGHT
2330 A$= ADDRESS$(LINENUM)
2340 IF TAG(LINENUM) <> 0 THEN MID$(A$,TAG(LINENUM),3)="  "
2350 LPRINT A$
2360 NEXT
2370 PRINT:PRINT:PRINT
2380 INPUT "Re-align printer as necessary and then press return ",A$
2390 '
2400 '      Print the labels
2410 '      =====
2420 '
2430 PRINT
2440 FOR LABEL = 1 TO NUMBER
2450 PRINT CHR$(13);LABEL;
2460 FOR LINENUM = 1 TO LABHEIGHT
2470 A$= ADDRESS$(LINENUM)
2480 IF TAG(LINENUM) = 0 THEN 2500
2490 MID$(A$,TAG(LINENUM),3)=RIGHT$(" "+STR$(LABEL),3)
2500 LPRINT A$
2510 NEXT LINENUM
2520 NEXT LABEL
2530 END
2540 '
2550 '      Subroutine to fill a line into A$
2560 '      =====
2570 '
2580 PRINT PREP$ COUNT " " ;MASK$;
2590 LINE INPUT "",A$
2600 IF A$="" THEN RETURN
2610 NSP = INT((30-LEN(A$))/2)
2620 IF LEN (A$) <= LABWIDTH THEN 2650
2630 PRINT "MAXIMUM "LABWIDTH" CHARACTERS"
2640 GOTO 2590
2650 IF CENTRE THEN A$ = SPACES$(NSP)+A$
2660 A$ = A$+SPACES$(30-LEN(A$))
2670 ADDRESS$(COUNT) = A$
2680 RETURN

```

~~~~~