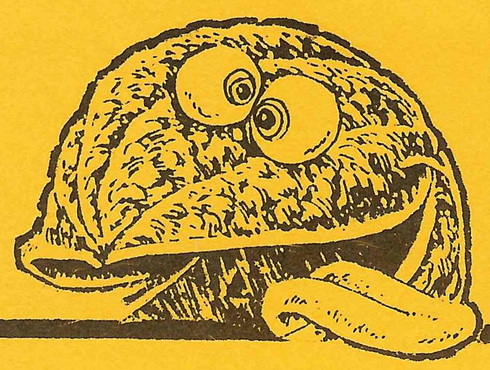
MEMOPAD

VOL 010

ISSUE 8



In This Issue

EDITORIAL ..... PAGE 218
DISC MANIA ..... PAGE 219
PROMZAP PART 2 ..... PAGE 222
FORTH WITH STRINGS ..... PAGE 226
FILE HANDLING ROUTINES ..... PAGE 229
VIEWPOINT ..... PAGE 239
MTX SCREEN DUMP ..... PAGE 241
SUBSCRIPTIONS ..... PAGE 244
SOFTWARE LIST ..... PAGE 245
HARDWARE LIST ..... PAGE 246
LATE XTRA! ..... PAGE 247

£1·35

memopad

Memotech Computer User Club Magazine

Edited by Tim Marstian

Artwork Anthony "Joe90"

Exective Editor Keith Hook

MEMOPAD IS PUBLISHED BY SYNTAXsoft FOR THE MEMOTECH USER GROUP
UNIT 109,GLENFIELD PARK, GLENFIELD ROAD, NELSON BB9 8AR
TELEPHONE: 0282 698849

COVER PRICE £1.25. MEMOPAD IS COPYRIGHT SYNTAXsoft 1985

AVAILABLE BY SUBSCRIPTION ONLY.

Editorial

Before I start rambling I must apologise for the fact that you are receiving this edition along with the previous month's. Believe it or not, it is not our fault ! The March edition of Memopad would have been delivered early but our printer has had a complete refit of his equipment and it appears that the gremlins have had a feast - who is responsible for spilling the water is a question for Sherlock ! Seriously, his new equipment has not functioned correctly since its installation and apart from our account, his other clients are in a worse state - at least we persuaded him to go straight into the next edition, and this just about brings us up to date.

We are now ready to release VORTEX SOFTWARE'S " HIGHWAY ENCOUNTER ". This is a fabulous 3D game and is definitely the best piece of software available on the MTX to press. Another good simulation is "CHAMPIONSHIP SNOOKER" this is now ready for despatch and costs #7.95p. It has a demonstration mode that shows a game based on two top class snooker champions. These two games courtesy of KH's begs and pleas to the software houses.

There is no further news from the bowels of Memotech and we are still awaiting clearance from the Receiver so that production can recommence. However, this lunch time I heard Keith in deep conversation with one of the directors, and from the amount I did hear, I gather that Memotech are just about ready to start processing all the back orders. Anyway, we will keep you informed, and if necessary, as soon as we hear any sort of concrete news we will do a special mail shot to let you know what the new company is to be called - we have heard, but not confirmed, that it is to be MEMOTECH COMPUTERS LIMITED.

We are aware that it is a most frustrating time for all of you. We also know, from the letters we have received, that the majority of you are remaining loyal to 'Black Beauty'. However,we do realise that some of our members are getting a little impatient, and may even be thinking of moving on to another machine. In all honesty, what is on the market - forgetting the support that certain other machines get - that is technically better than the MTX ? The simple answer is: nothing ! This is where the shame appears .... we, at Genpat, are hoping that Geoff Boyd manages to pull all the leading players into line in time to take on the market, and before the curtain rises on the boom - which is surely going to happen later this year.

Some of you may have read in the trade papers, and in the national papers of the latest happenings at Syntaxsoft - the signing of exclusive contracts with Mastertronics, the fact that Syntaxsoft have been signed to work on the Rom and development of a new product from another company .. so much has taken place recently that it is hard to keep track. It is important, however, to realise that Syntaxsoft is devoted to the MTX and whenever a deal is struck with another company, Keith always makes sure that it includes the MTX.

It is now 9-30pm and Keith has just spoken with Geoff Boyd. The up to minute news is that Geoff is about to go into production with a new computer, based on the MTX 512. It will be downward compatable so that existing software will run on the new machine. The computer will use a 3 1/2" drive (the industry standard) with a capacity of 1megobyte. It will be offered with a printer etc at around #399.00. Geoff also hinted that he had designed a new computer around the current Hitachi chip but would not go into further details as everything is still under wraps. A direct quote from Geoff's phone call to Keith:

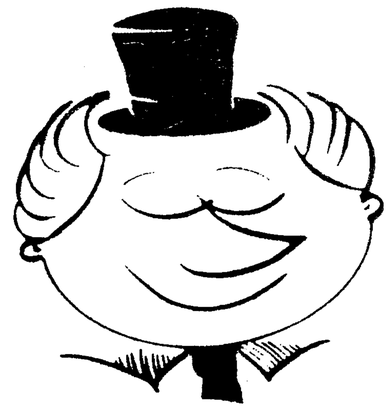
KEITH " What I am concerned with is the fact that all these users have remained loyal to you ... and I don't want them letting down by polictical back biting with the you know who ...."

GEOFF " Yes, I agree, I have had many letters ... it would have been a far easier course to have forgotten the MTX and started from scratch ...but loyalty does deserve to be backed up ... that is why, as you know, I am still having this hassle with the existing company ..... I want to support those people who have shown faith from the very start ... they made us money ... they should be supported..."

Nuff said !

TM

Disc Mania by *Eric Roy*



BDOS FUNCTION CALLS

	CODE	FUNCTION OF CALL TO BDOS	ADDRESS
**	0	System Reset	#E503
**	1	Read Console Byte	#D9C8
**	2	Write Console Byte	#D890
**	3	Read "Reader" Byte	#D9CE
**	4	Write "Punch" Byte	#E512
**	5	Write Printer Byte	#E50F
**	6	Direct Console Input/Output	#D9D4
**	7	Get IOBYTE	#D9ED
**	8	Set IOBYTE	#D9F3
**	9	Print Console String	#D9F8
**	A	Read Console String	#D8E1
**	B	Read Console Status	#D9FE
	C	Get CP/M Version Number	#E37E
	D	Disc System Reset	#E383
	E	Select Disc	#E345
	F	Open File	#E39C
	10	Close File	#E3A5
	11	Search for First Name Match	#E3AB
	12	Search for Next Name Match	#E3C8
	13	Erase (delete) File	#E3D7
	14	Read Sequential	#E3E0
	15	Write Sequential	#E3E6
	16	Create File	#E3EC
	17	Rename File	#E3F5
	18	Get Active (logged-in) discs	#E3FE
	19	Get Current Default Disc	#E404
	1A	Set DMA (Read/Write) Address	#E40A
	1B	Get Allocation Vector Address	#E411
	1C	Set Disc to Read Only	#DC2C
	1D	Get Read Only Discs	#E417
	1E	Set File Attributes	#E41D
	1F	Get Disc Parameter Block Adr.	#E426
	20	Set/Get User Number	#E42D
	21	Read Random	#E441
	22	Write Random	#E447
	23	Get File Size	#E44D
	24	Set Random Record Number	#E30E
	25	Reset Drive	#E453
**	26	Functions 26/27 not supported	#DA04
**	27	#DA04 = RET instruction	#DA04
	28	Write Random with Zero Fill	#E49B

FUNCTIONS MARKED ** ARE NOT AVAILABE ON SDX

As can be seen from the above table most of the BDOS functions found on a CP/M system are available on the SDX system. The ones marked with ** are mainly connected with the CCP input/output and are not used by the SDX. Having said that if you type ROM 3 then use the Panel to examine location #E503 (System Reset) you will find it is a JP #E685 which resets a true CP/M system's 0 page. So the code is already there if you decide to upgrade to CP/M.

It is the above functions which make CP/M such a popular operating system, in that a call to any of the BDOS functions will produce the same results regardless of the machine it is running in. The only difference being the address from which to call BDOS. In a CP/M system it is via #0005 while on the SDX it is via #F5BO.

I don't intend to go into detail of the functions and the parameters required by each. I suggest you get hold of a book on CP/M if you are interested in finding out more about it as it will explain the workings of CP/M better than I can.

Before making a call to BDOS the register C must contain the function code number. If a simple byte is to be passed to BDOS then it should be in reg. E (mainly used by CCP). In some functions BDOS expects a 16 bit value/address in reg. pair DE.

On return regs. A and L will contain the same value, as will regs. B and H. If a 16 bit value/address is returned it will be in reg. pair HL.

A typical calling sequence could be...

```
LD C,#0C            ; Return version number
CALL #F5BO          ; BDOS
On return HL =      version number.

LD C,#1A            ; Set DMA address
LD DE,ADDRESS       ; DE = address of DMA
CALL #F5BO          ; BDOS
On return the DMA will be at address specified.
```

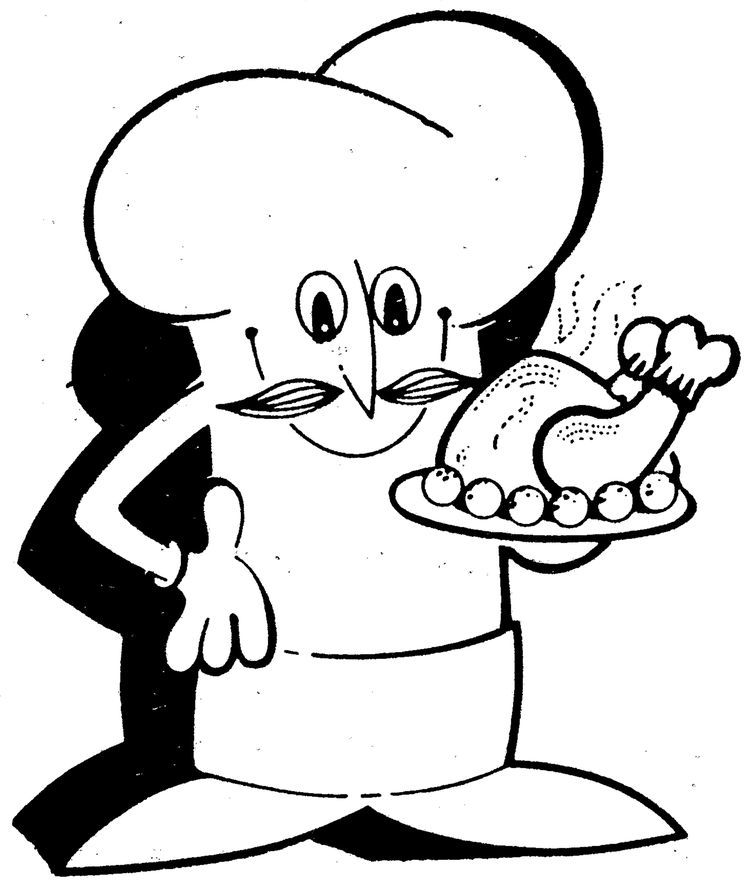
A call to BDOS at #F5BO jumps to #D706 which contains a jump to #D711. The code at #D711 does the following.....

1. Saves entry value of DE in #DA43,4
2. Saves entry value of E in #E4D6
3. Sets #DA45,6 to zero
   If a value is returned by BDOS it is stored here as well as being returned in A or HL.
4. Saves stack pointer in #DA0F,10 and sets BDOS stack to #DA00.
5. Pushes return address on BDOS stack so that return is via #E474.
6. Checks that reg. C contains a valid function code.
   Returns via #E474 if C is not in range.
7. Get address of BDOS function from table at #D747. (HL = address)
8. Restores entry value of DE.
9. Jumps to BDOS function (JP (HL))

Item 4 caused me a lot of problems when I started writing assembly programs for the SDX. For example the following will not work.....

```
MAIN PROGRAM
....
CALL SETUP          ; Reset disc / DMA
....
....
SETUP: LD C,#0D     ; Reset disc
       CALL BDOS    ; Stack at #DA00
       CALL SDMA
       RET
SDMA:  LD C, #1A
       LD DE, ADROMA
       CALL BDOS    ; Stack reset to #DA00 so return
       RET          ; Address to SETUP is lost.
```

The correct calling sequence should be.....



```
        MAIN PROGRAM
        ....
        CALL SETUP
        CALL SDMA
        ....
        ....

SETUP:LD C,#0D
      CALL BDOS
      RET
SDMA: LD C,#1A
      LD DE,ADRDMA
      CALL BDOS
      RET
```

To finish this month is a short program LPDIR which dumps the directory to a printer. It uses function #0D to reset the disc system, and shows how discs can be changed under program control without resorting to the ROM 3 command. Just keep feeding the drive with discs when prompted and the directory will be dumped to the printer, cut round the dotted line (get your mum to help you here) and store printout with disc for future reference. ★

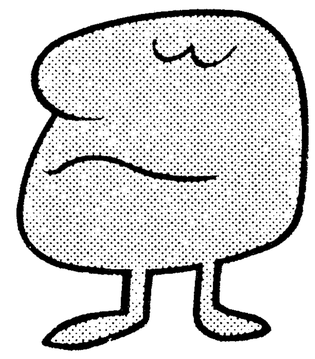
```
100  REM ********************************
110  REM ***DIRECTORY TO PRINTER*******
120  REM *** by E. Roy March 86 *******
130  REM ********************************
140  REM  Demo listing showing how SDX
142  REM  discs can be changed under
143  REM  program control, by using
144  REM  BDOS function #0D - Reset.
145  REM -------------------------------
150  GOTO 200
160  CODE

417C            LD  C,#0D    ;  Reset disc system
417E            CALL #F5B0
4181            RET

Symbols:
```

```
170 RETURN
200 VS 5: CLS
210 CSR 6,0: PRINT "---print disc directory---"
220 CSR 4,3
230 INPUT "INSERT DISC THEN PRESS <RET>";A$
240 GOSUB 160
250 CLS : USER DIR
260 GOSUB 400: CSR0,0
270 FOR Y=0 TO 23
280 LET A$=""
290 FOR X=0 TO 38
300 LET A$=A$+SPK$
310 NEXT X
320 LPRINT CHR$(124)+CHR$(32)+A$+CHR$(124)
330 NEXT Y
340 GOSUB 400
350 CLS
360 CSR 4,8
370 INPUT "Print another directory Y/N ";A$
380 IF A$="Y" OR A$="Y" THEN GOTO  150
390 CLS : STOP
400 FOR X=1 TO 42
410 LET A$(X)=CHR$(45)
420 NEXT X
430 LPRINT A$
440 RETURN
```



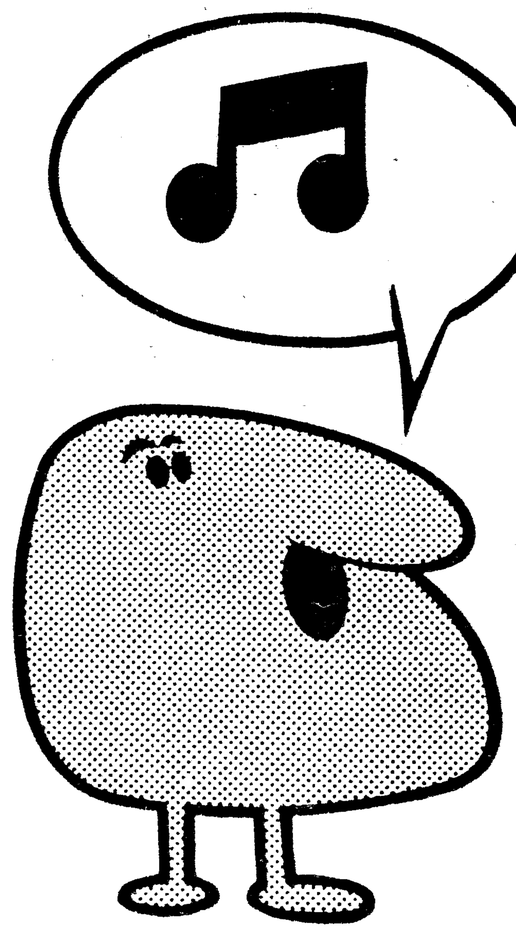
```
-----------------------------------------------
:  BACKUP   .HEX BACKUP    .MAC DCD      .MAC  :
:  DCD      .HEX SGATT     .MAC SGATT    .HEX  :
:  TITLE    .BAS LPDIR     .BAS                :
:                                              :
:                                              :
```

Promzap by John Hudson — Part 2



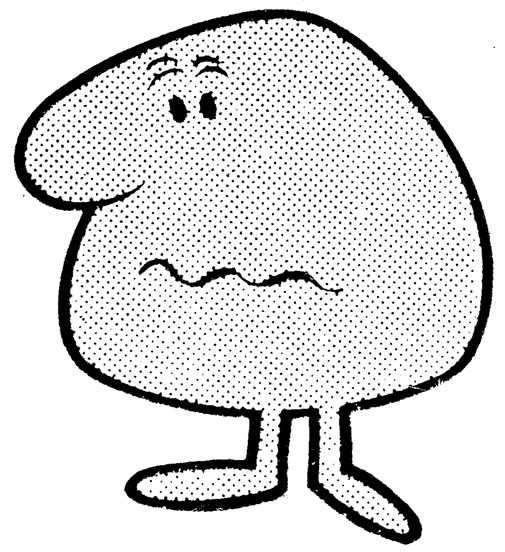
Here is the Eprom listing which goes with the previous month's article.★

```
4007          LD A,0              3E00
4009          LD (BLAN),A         32D843
400C          LD (PRIN),A         32D743
400F          LD A,4              3E04
4011          OUT (115),A         D373
4013          LD HL,0             210000
4016          LD (SAVE),HL        22D443
4019          LD (ADD),HL         22D043
401C          LD (SIZE),HL        22D243
401F PROM:    RST 10              D7
4020          DB £6D,£9F,£1C,"  PROM PROG. (c) J.Hudson 1986"    6D9F1C20205052
4F4D2050524F472E20286329204A2E48756475F6E2031393836
4041 RES:     RST 10              D7
4042          DB £B7,10,10,10,13,"  (1)  Program Prom"          B70A0A0A0D2020
2831292020507261726D2050726F6D
405A          DB £B6,10,10,13,"  (2)  Blank Check "             B60A0A0D202028
3229202042616E6B20436865636B20
4071          DB £B6,10,10,13,"  (3)  Verify  Prom"            B60A0A0D202028
3329202056657269667920250726F6D
40B8          DB £B4,10,10,13,"  (4)  Read Prom "              B40A0A0D202028
3429202052656164205072F6D20
40D9          DB £B9,10,10,13,"  (5)  Select Printer "         B90A0A0D202028
3529202053656C656374205072696E74657220
40B7          DB £99,10,10;13,"  (6)  Return to Basic"         990A0A0D202028
3629202052657475726E20746F204261736963
40D1          RST 10              D7
40D2          DB £6F              6F
40D3          DB £93,".....enter option >"      932E2E2E2E2E656E7465722064F7074
696F6E203E
40E7 IN2:     CALL £79            CD7900
40EA          JR Z,IN2            28FB
40EC          LD B,A              47
40ED          CALL £BC            CDBC00
40F0          LD A,B              78
40F1          SUB £30             D630
40F3          CP 1                FE01
40F5          JP Z,BLOW           CAA041
40F8          CP 2                FE02
40FA          JP Z,BLANK          CA5244
40FD          CP 3                FE03
40FF          JP Z,VERIFY         CAD943
4102          CP 4                FE04
4104          JR Z,READ           280A
4106          CP 5                FE05
4108          JR Z,PRINTER        282A
410A          CP 6                FE06
410C          RET Z               C8
410D          JP PROM             C31F40
4110 READ:    LD A,138            3E8A
4112          CALL TYPE           CDD041
4115          LD DE,(ADD)         ED5BD043
4119          LD BC,(SIZE)        ED4BD243
411D          LD HL,0             210000
4120 RD2:     CALL ZAPA           CD7C43
4123          LD A,138            3E8A
4125          OUT (115),A         D373
4127          IN A,(113)          DB71
4129          LD (DE),A           12
412A          INC DE              13
412B          INC HL              23
```

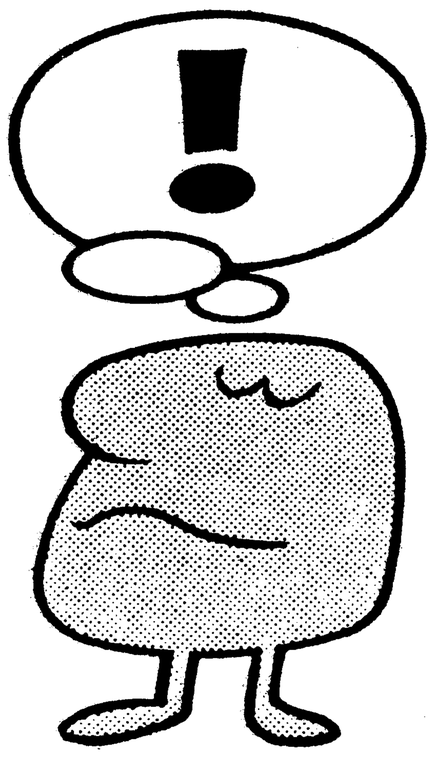


```
        412C        DEC BC          0B
        412D        LD A,B          78
        412E        OR C            B1
        412F        JR NZ,RD2       20EF
        4131        JP PROM         C31F40
        4134 PRINTER:RST 10         D7
4135        DB £6D,£BB,"Fail messages to printer or"        6DBB4661696C20
6D6573736167657320746F207072696E746572206F72
4152        DB £9F,"            screen type P or S"         9F2020202020202
0202020202020207363726565E20747970652050206F722053
        4172 SC:    CALL £79        CD7900
        4175        JR Z,SC         28FB
        4177        OR £20          F620
        4179        CP "p"          FE70
        417B        JR Z,PTR        2806
        417D        CP "s"          FE73
        417F        JR Z,SCR        280A
        4181        JR PRINTER      18B1
        4183 PTR:   LD A,1          3E01
        4185        LD (PRIN),A     32D743
        4188        JP PROM         C31F40
        418B SCR:   LD A,0          3E00
        418D        LD (PRIN),A     32D743
        4190        JP PROM         C31F40
        4193 TEST:  LD BC,9         010900
        4196 LOOP:  DEC C           0D
        4197        RET Z           C8
        4198 ON:    AND A           A7
        4199        RLC A           CB07
        419B        JR C,LOOP       38F9
        419D        INC B           04
        419E        JR LOOP         18F6
        41A0 BLOW:  LD A,136        3E88
        41A2        CALL TYPE       CDD041
        41A5        RST 10          D7
41A6        DB £98,3,10,10,"Prom now being Zapped"          980306A0A50726F
6D206E6F772062656696E67205A6170706564
        41BF        LD HL,£3030     213030
        41C2        LD (£FD59),HL   2259FD
        41C5        LD (£FD5B),HL   225BFD
        41C8        LD A,0          3E00
        41CA        LD (£FD7D),A    327DFD
        41CD        JP COMMON       C31643
        41D0 TYPE:  RST 10          D7
41D1        DB £6D,£B7,"     SELECTION MENU",10,10,10,13        6DB720202020
53454C454354494F4E204D454E550A0A0A0D
41EA        DB £B5,"     (1)     2764 ",10,10,13                B520202020202028
312920202020203237363420A0A0D
4200        DB £B8,"     (2)     27128",10,10,10,10,10,13       B8202020202028
322920202020203237313238A0A0A0A0A0D
4219        DB £9B,"   Enter type  ( 1 or 2 )  ",10,10,13       9B2020204E74
6572207479706520202820312067220322029202A0A0D
        423B TYP:   CALL £79        CD7900
        423B        JR Z,TYP        28FB
        423D        LD B,A          47
        423E        CALL £BC        CDBC00
        4241        LD A,B          78
        4242        CP "1"          FE31
        4244        JR Z,TWO764     2B4D
        4246        CP "2"          FE32
```

```
        4249        JR Z,TWO7128    2802
        424A        JR TYPE         1884
        424C TWO7128:RST 10         D7
        424D        DB £6D,£BF,"  Selected  Prom type = 27128",13,10   6DBF20225365
6563746564202050726F6D207479706520532032373132380D0A
        426E        DB £98,13,10,"   Is this O.K. (Y or N) "         980D0A20202049
7320746869732047E4B2E202859206F72204E2920
        428A        LD DE,16384     110040
        428D        LD (SIZE),DE    ED53D243
        4291        JR IN1          1844
        4293 TWO764:RST 10          D7
        4294        DB £6D,£BE,"   Selected Prom type = 2764",13,10    6DBE2220225365
6C6563746564205072F6D207479706520532032373640D0A
        42B4        DB £9B,13,10,"   Is this O.K. (Y or N) "         9B0D0A20202049
7320746869732047E4B2E202859206F72204E2920
        42D0        LD DE,8192      110020
        42D3        LD (SIZE),DE    ED5320243
        42D7 IN1:   CALL £79        CD7900
        42DA        JR Z,IN1        28FB
        42DC        AND EDF         E6DF
        42DE        CP "Y"          FE59
        42E0        JP Z,OKTYPE     CAE742
        42E3        POP HL          E1
        42E4        JP ERROR        C38944
        42E7 OKTYPE:LD A,(BLAN)     3AD843
        42EA        OR A            B7
        42EB        CP 0            FE00
        42ED        JR Z,OKTYP      2805
        42EF        LD HL,BLAN      21D843
        42F2        JR RETS         181B
        42F4 OKTYP: CALL £CB        CDCB00
        42F7        RST 28          EF
        42F8        DB £AB,"Enter start address  ",£A0              AB456E74657220
7374617274206164647265737320A0
        430F RETS:  LD (ADD),HL     22D043
        4312        LD (ADD),HL     22D043
        4315        RET             C9
        4316 COMMON:LD HL,(ADD)     2AD043
        4319        LD A,(HL)       7E
        431A        CALL TEST       CD7341
        431D        LD (SAVA),A     32D643
        4320        LD A,0          3E00
        4322        OR B            B0
        4323        LD A,(SAVA)     3AD643
        4326        JR NZ,ZAP       2009
        4328        LD HL,(SAVE)    2AD443
        432B        INC HL          23
        432C        LD (SAVE),HL    22D443
        432F        JR FF           1806
        4331 ZAP:   CALL ZAP0       CD7943
        4334        CALL SETBYTE    CD9E43
        4337 FF:    LD DE,(SIZE)    ED5BD243
        433B        DEC DE          1B
        433C        LD (SIZE),DE    ED53D243
```



```
4340          LD A,D              7A
4341          OR E                B3
4342          JR NZ,LAB           200A
4344          LD A,4              3E04
4346          OUT (115),A         D373
4348          CALL £953           CD5309
434B          JP PROM             C31F40
434E LAB:     LD HL,(ADD)         2AD043
4351          INC HL              23
4352          LD (ADD),HL         22D043
4355 TIME:    PUSH HL             E5
4356          PUSH DE             D5
4357          PUSH BC             C5
4358          RST 10              D7
4359          DB £60,£8B,3,1,1,"  Time  "     60BB03010120205469D652020
4366          LD DE,£FD59         1159FD
4369          LD BC,4             010400
436C TI:      LD A,(DE)           1A
436D          CALL £BC            CDBC00
4370          INC DE              13
4371          DEC C               0D
4372          JR NZ,TI            20F8
4374          POP BC              C1
4375          POP DE              D1
4376          POP HL              E1
4377          JR COMMON           189D
4379 ZAP0:    LD (SAVA),A         32D643
437C ZAPA:    LD A,136            3E88
437E          OUT (115),A         D373
4380          LD A,3              3E03
4382          OUT (115),A         D373
4384          LD HL,(SAVE)        2AD443
4387          LD A,L              7D
4388          OUT (112),A         D370
438A          LD A,H              7C
438B          OUT (113),A         D371
438D          INC HL              23
438E          LD (SAVE),HL        22D443
4391          LD A,1              3E01
4393          OUT (115),A         D373
4395          LD A,0              3E00
4397          OUT (115),A         D373
4399          LD A,2              3E02
439B          OUT (115),A         D373
439D          RET                 C9
439E SETBYTE: LD A,5              3E05
43A0          OUT (115),A         D373
43A2          LD A,(SAVA)         3AD643
43A5          OUT (113),A         D371
43A7 ZAP7:    LD A,6              3E06
43A9          OUT (115),A         D373
43AB          CALL PAUSE          CDC543
43AE ZAP1:    LD A,7              3E07
43B0          OUT (115),A         D373
43B2 ZAP2:    IN A,(114)          DB72
43B4          AND 16              E610
43B6          JR Z,ZAP2           28FA
43B8 DEL:     IN A,(114)          DB72
43BA          AND 16              E610
43BC          JR NZ,DEL           20FA
43BE          DEC B               05
43BF          RET Z               C8
43C0          LD A,(SAVA)         3AD643
```



```
43C3          JR ZAP7             18E2
43C5 PAUSE:   PUSH BC             C5
43C6          LD BC,£40           014000
43C9 DEC2:    DEC BC              0B
43CA          LD A,B              78
43CB          OR C                B1
43CC          JR NZ,DEC2          20FB
43CE          POP BC              C1
43CF          RET                 C9
43D0 ADD:     DW 0000             0000
43D2 SIZE:    DW 0000             0000
43D4 SAVE:    DW 0000             0000
43D6 SAVA:    DB 00               00
43D7 PRIN:    DB 0                00
43D8 BLAN:    DB 0                00
43D9 VERIFY:  LD A,138            3E8A
43DB          CALL TYPE           CDD041
43DE          LD BC,(SIZE)        ED4BD243
43E2          LD DE,(ADD)         ED5BD043
43E6          LD HL,0             210000
43E9 VER1:    CALL ZAPA           CD7C43
43EC          LD A,138            3E8A
43EE          OUT (115),A         D373
43F0          IN A,(113)          DB71
43F2          EX DE,HL            EB
43F3          CP (HL)             BE
43F4          CALL NZ,FAIL        C40144
43F7          EX DE,HL            EB
43F8          INC DE              13
43F9          DEC BC              0B
43FA          LD A,B              78
43FB          OR C                B1
43FC          JP Z,PROM           CA1F40
43FF          JR VER1             18E8
4401 FAIL:    PUSH BC             C5
4402          PUSH DE             D5
4403          PUSH HL             E5
4404          PUSH AF             F5
4405          LD A,(PRIN)         3AD743
4408          LD (£FD75),A        3275FD
440B          RST 10              D7
440C          DB £65,£8A,13,10,"Fail at "     658A0D0A4661696C20617420
4418          LD B,D              42
4419          LD C,E              4B
441A          DEC BC              0B
441B          CALL £1B50          CD501B
441E          RST 10              D7
441F          DB £8B,"  Expected "            BB2020457870656374656420
442B          LD A,(HL)           7E
442C          LD C,A              4F
442D          CALL £1B55          CD551B
4430          RST 10              D7
4431          DB £89,"  Actual "              892020416374756161C20
443B          POP AF              F1
443C          LD C,A              4F
443D          CALL £1B55          CD551B
4440          LD A,0              3E00
4442          LD (£FD75),A        3275FD
4445          POP HL              E1
4446          POP DE              D1
4447          POP BC              C1
4448 IN4:     CALL £79            CD7900
444B          CP 131              FE83
```

```
444D          JR Z,RET          2801
444F          RET               C9
4450 RET:     POP HL            E1
4451          RET               C9
4452 BLANK:   LD A,138          3E8A
4454          LD A,255          3EFF
4456          LD (BLAN),A       32D843
4459          CALL TYPE         CDD041
445C          LD (HL),A         77
445D          EX DE,HL          EB
445E          LD BC,(SIZE)      ED4BD243
4462          LD DE,(ADD)       ED5BD043
4466          LD HL,0           210000
4469 BLK:     CALL ZAPA         CD7C43
446C          LD A,138          3E8A
446E          OUT (115),A       D373
4470          IN A,(113)        DB71
4472          CP 255            FEFF
4474          JR Z,BLKG         2805
4476          EX DE,HL          EB
4477          CALL FAIL         CD0144
447A          EX DE,HL          EB
447B BLKG:    DEC BC            0B
447C          LD A,B            78
447D          OR C              B1
447E          JP NZ,BLK         C26944
4481          LD A,0            3E00
4483          LD (BLAN),A       32D843
4486          JP PROM           C31F40
4489 ERROR:   RST 10            D7
448A          DB EBC,10,10,10,13,"  Please install correct"      20206C6561736520696E7374616C6C20636F727265637
44A7          DB E9E,10,13,"  Prom type and start again."       6F5020747970652061E642073746172742061676169E2E
44C6          RET               C9
```


VIEWDATA FOR YOUR MEMOTECH
FULL COLOUR CP/M VERSION NOW AVAILABLE !

JARO COMPUTER SERVICES

ELECTRONIC MAIL · HOME BANKING · TELEX

(INCOMING · OUTGOING U.K. · INTERNATIONAL)

TELESHOPPING · DATABASES · INFORMATION

FULL COLOUR MTX CP/M VIEWDATA

Our full colour CP/M software is now available. This offers:
* Full colour text and graphics; flash; graphics hold; etc.
* 'Split-screen' user menu with easy-use prompts, offering:
* Ten pages of internal page store for fast comparisons;
* Sixteen fully programmable keys (up to 32 characters each);
* Store programmable keys to disk or make temporary changes;
* Store current page to disk ('file exists' warning prompts);
* Load page from disk (page directory list before loading);
* Fast bit-graphics page print to Epson-compatible printers;
* x4 magnification page print with border, as above;
* Fully buffered page printing reduces waiting times.

Supplied on 5.25 inch DSDD disk: (copy protected) £24.95

FREE CP/M XMODEM SOFTWARE (MTX)

We have configured a program from the CP/M User Group Library, written by David Back, for use with the Memotech CP/M systems. The program is a powerful terminal and 'file transfer' utility which will allow you, using a SpeedSplitter Board, to transfer files using the Ward Christiansen XMODEM protocol via the B.T. telephone system at 1200 baud. The protocol has built-in error correction facilities. All support software to control the SSI is provided. This is available free, but we have to charge the sum of £5 to cover the cost of the disk, postage, etc.

JARO VIEWDATA SOFTWARE CASSETTE

Available for all MTX machines (state MTX500,512,RS128): £9.9
Turn your Memotech computer into a monochrome (black on cyan) 1200/75 (V23) viewdata terminal:
* Full viewdata character set including contiguous/seperated graphics, cursor on/off and addressing from host, & flash
* Viewdata keypad with 'asterisk' and 'hash' keys
* Seven page internal frame store with page number in cursor
* Page print via parallel (Centronics) port (Epson codes)
* 16 programmable keys for frequently used frames or user-id

If the 'Walkbury Consultants Ltd' modem is used, in addition:
* Up to 8 autodial telephone numbers and keypad dialling

(Requires Jaro Speed-splitter Board, RS232 board and modem)
Note: The following viewdata features cannot be implemented o the cassette version of JaroViewdata due to limitations of th MTX Video Display Processor: colour attributes, double height and concealed text.

SOFTWARE AVAILABILITY

	Cassette	SDX(250k)	SDX(500k)	CPM(AnyK)
Autodialling:				
Walkbury (Demon)	Yes	Yes	Yes	No
Pace Nightingale	No	Yes	Yes	No
Non-autodial:				
Any V.23 1200/75	Yes	Yes	Yes	Yes

Make cheques etc. payable to Jaro Computer Services. Remember: list disc drive size (250/500K), MTX type (512,500,RS128), and if applicable, the type of modem which you will be using.

JARO SPEED-SPLITTER BOARD

* Allows 'split-speed' operation of MTX channel A (ie 1200/75 'user', or 75/1200 'reverse Prestel' or 'host')
* Completely transparent to normal MTX operation, requires no alteration to any existing software ('off' at power-up)
* Software control via output port 7 (uncommitted PIO)
* Fits inside lid of MTX casing - does not prevent fitting of any MTX add-on boards, e.g. extra memory, etc.

BOARD ONLY - 1 solder connection to RS232 required: £14.75
BOARD FITTED TO RS232 BOARD - no soldering required: £44.75
Both options include lead and easy-to-follow instructions. The speed-splitter can be easily installed by the user without any technical knowledge at all in under 30 minutes.
NEW: AdaptorBox - add £15 to above prices. If you already have a Jaro SpeedSplitter and you wish to add an AdaptorBox, return the grey multiway lead with £15 and we will upgrade it.

JARO VIEWDATA DISC SOFTWARE

SDX VERSION - 250K OR 500K DRIVES (SPECIFY WHICH) - £14.95 !!!
The SDX version is similar to the cassette version except for:
* Select ink and paper colours; stored to disk if required;
* Programmed key values stored to disk if required;
* Autodial directory (autodial modems only) stored to disk;
* Easy entry and display of control codes in key programming.

There are two versions of this program as SDX systems type 1.7 and later are slightly different from early versions. Both are supplied on the disk, but you MUST specify 250K or 500K drive.

NEW PRODUCT!! - Jaro AdaptorBox

Connect a Pace Nightingale Autodial card to your MTX using our Jaro AdaptorBox unit. A Pace autodialler can then be used with our Viewdata software in a similar fashion to a Demon/Walkbury type modem, BUT - the Pace modem is BT approved. Customers who already have a JaroBoard fitted can upgrade it to include this new product. Please note: we can fit a 25-way 'D' plug to your Pace modem lead for use with an MTX computer - send £5 and the original lead to us; or send £10 for a new 1m lead (2m - £12). SpeedSplitter with AdaptorBox -- add £15 to appropriate price.

MICROVITEC MONITOR FOR MEMOTECH

When you upgrade your system to CP/M, you will require a good quality medium resolution colour monitor. The Microvitec 1451 is such a monitor, and you can buy it from us, complete with a lead for the Memotech (and composite video lead - PAL version) and ready for use. Prices include insured delivery charges.

1451MS or DS (metal or plastic case) RGB 14in. monitor: ~~£280~~ 220
As above plus PAL comp. video (MTX monitor) and audio: £325 275

JARO COMPUTER SERVICES
MICRO COMPUTERS ACCESSORIES PRINTERS SOFTWARE

4 Finnart Close Weybridge Surrey KT13 8QE
Tel: Weybridge (0932) 57398
Prestel Mailbox 019995085

Forth With Strings by Brian Houghton

FORTH, like PASCAL, is not exactly famous for its string-handling ability, and a number of algorithms have been published to deal with this problem.

In the case of FORTH, the trouble is that too many algorithms have been published. They are mutually incompatible, and tend to lead to everyone writing an individual FORTH.

I have based my string package on the conventions of FORTH-83, as the algorithm is fast, produces words which need little or no low-level code to use them, and which leave on the stack parameters which merely need TYPE or some equivalent to display the contents of the strings.

The words STRING, $CONSTANT and $ARRAY are all based on the following data-structure:

```
SIZE OF ARRAY = Maximum length + 2;
BYTE 0        = Maximum length;
BYTE 1        = Actual length;
BYTE 2        = Start of string data.
```

Calling any of these words returns the address of BYTE 2 and the contents of BYTE 1.

The following glossary is printed by the FORTH-83 word GLOSS from the RAM-disk.

```
GLOSSARY

TOOLS CONTINUED    (   " )
SCAN> ( ---addr Looks ahead )
"   ( ---$addr,count if not compiling )
TOOLS   ( $CONSTANT )
$CONSTANT ( " string"--- DOES> ---addr,count )
$ARRAY ( n,length--- )
STRING FUNCTIONS
STRING ( n---name;---addr+2,ct )
CPACK ( addr1,addr2,ct--- Moves a string )
S! ( string1,string2--- )
String2 must be left by a string buffer.
S+ ( string1,string2---- )
concatenate string1 to string2
COMPARE ( string1,string2----1/01 )
/STRING ( string1,n--- drop the first n chars )
(LOCATE) ( addr,ct,c---addr+1 )
return addr+1 of located char
LEX ( string,char---string1,string2 )
```

LOCATE (string, ct, c --- addr+1, f)

NOTES

The two non-standard words in the source code are [: NOT 0=;] and [ASCII]. The latter is very useful for those of us who don't find it easy to remember ASCII codes, as it returns the decimal equivalent of the next character in the input stream. If you don't want to use it, then simply type the appropriate number into the definition. Some FORTH systems require rather elaborate definitions of ASCII, but in fig-FORTH the following simple one works:

```
: ASCII  BL WORD  HERE C@;
```

and has been used in these definitions.

The basic user word is ["], which behaves like [."] except that instead of executing a TYPE, it leaves on the stack the start address and count of the string literal. If it is used in a definition, the string is compiled into the dictionary and its parameters are left when the definition is invoked. Used interactively the string is stored at PAD and the corresponding parameters are left on the stack ready for TYPE.

[$CONSTANT] is useful for repetitious run-time messages, e.g.:

```
"PRESS ANY KEY WHEN READY" $CONSTANT PROMPT
```

and behaves like a numeric constant except for leaving the TYPE parameters:you should not try to change its contents.

A STRING VARIABLE is declared by a statement as follows:

```
40 STRING NAME
```

where '40' may be any number between 1 and 255 and is the maximum length of the string. An uninitialised variable returns a count of 0, giving you some protection against trying to TYPE it. It may be initialised either from a string literal or from another string variable. The word [S!] replaces the existing contents of a string variable with a preceding string buffer, while [S+] concatenates two strings (in postfix order I'm afraid!).

Examples:

```
30 STRING ALPHA  30 STRING BETA  30 STRING GAMMA
" THIS ONE" ALPHA S!
"  AND THAT ONE."BETA S!
ALPHA GAMMA S!
BETA GAMMA S+
ALPHA TYPE (RET)
      THIS ONEok
BETA TYPE (RET)
      AND THAT ONE.ok
GAMMA TYPE (RET)
      THIS ONE AND THAT ONE.ok
```

[$ARRAY] has been designed to behave precisely as an array of strings, and to exit with an error-message if you try to address a non-existent element. The statement:

```
50 20 $ARRAY NAMES
```

defines an array of 50 20-character strings; elements of NAMES may now be addressed by declaring 'n NAMES', where 'n' is any number between 0 and 49.

The words [COMPARE],[/STRING],{LOCATE] & [LEX] are reasonably self-explanatory, but the following short dialog with FORTH will help to illustrate their use:

```
ok                                                  1
ALPHA TYPE THIS ONEok                               30151
BETA TYPE  AND THAT ONE.ok                          22
GAMMA TYPE THIS ONE AND THAT ONE.ok                 30141 ok
ALPHA BETA COMPARE . 52 ok                          DROP OVER - .S
BETA ALPHA COMPARE . -52 ok                         30129
ALPHA GAMMA COMPARE . 0 ok                          22
ALPHA ALPHA COMPARE . 0 ok                          30141 ok
" AAA" ALPHA COMPARE . -19 ok                       DROP ok
" ZZZ" ALPHA COMPARE . 6 ok                         9 /STRING .S
GAMMA 2DUP ASCII A LOCATE .S                        13
```

```
30150 ok
2DUP TYPE AND THAT ONE.ok
ALPHA S! ok
ALPHA TYPE AND THAT ONE.ok
" THIS ONE :AND THAT ONE." GAMMA S! ok
GAMMA TYPE THIS ONE :AND THAT ONE.ok
GAMMA ASCII : LEX .S
13
```

```
30151
9
30141 ok
TYPE  AND THAT ONE.ok
TYPE  THIS ONE ok
" AAAAA" ALPHA S! " AAAAB" BETA S! ok
ALPHA BETA COMPARE . -1 ok
BETA ALPHA COMPARE . 1 ok
```

Full source-code is appended.★

```
SCR # 2
  0 \ TOOLS CONTINUED    (   " )
  1  : NOT 0= ; : (") R COUNT DUP 1+ R> + >R ;
  2 : SCAN> ( ---addr Looks ahead )
  3     BLK @ -DUP IF BLOCK ELSE TIB @ THEN IN @ + ;
  4 : "   ( ----$addr,count if not compiling )
  5  SCAN>  C@ ASCII " = NOT
  6  IF ASCII " STATE @
  7      IF  COMPILE (") WORD HERE
  8        HERE OVER C@ 1+ CMOVE
  9        HERE C@ 1+ ALLOT
 10       ELSE TEXT PAD COUNT
 11      THEN
 12  ELSE 1 IN +!
 13 THEN ; IMMEDIATE
 14
 15 ;S
```

```
SCR # 3
  0 \ TOOLS   ( $CONSTANT )
  1 : $CONSTANT ( " string"--- DOES> ---addr,count )
  2    <BUILDS DROP 1 - , DOES> @ COUNT ;
  3 : $ARRAY ( n,length--- )
  4    <BUILDS 2DUP SWAP C, C, 2+ * DUP ALLOT
  5    DUP HERE SWAP - SWAP 0 FILL DOES>
  6    2DUP C@ 1 - OVER < SWAP 0< OR IF ." Out of range." EXIT THEN
  7    DUP ROT ROT DUP 1+ C@ ROT * + 2+
  8    SWAP 1+ C@ OVER 1+ C! 2+ COUNT ;
  9
 10
 11
 12
 13
 14
 15 ;S
```

```
SCR # 4
  0 \ STRING FUNCTIONS
  1 : STRING ( n---name;---addr+2,ct )
  2    <BUILDS DUP C, 0 C, ALLOT
  3    DOES> 1+ COUNT ;
  4 : CPACK ( addr1,addr2,ct---- Moves a string )
  5  SWAP 2DUP C! 1+ SWAP CMOVE ;
  6 : S! ( string1,string2--- )
  7 \ String2 must be left by a string buffer.
  8  DROP 1 - DUP 1 - C@
  9  ROT MIN CPACK ;
 10 : S+ ( string1,string2----- )
 11 \ concatenate string1 to string2
 12  ROT >R OVER 1 - 2DUP 1 - C@
 13  SWAP - R> MIN
 14  SWAP OVER >R +! + R> CMOVE ;
 15 ;S
```

```
SCR # 5
  0 ( STRINGS CONTINUED )
  1 : COMPARE ( string1,string2----1/01 )
  2   DROP -TEXT ;
  3 : /STRING ( string1,n--- drop the first n chars )
  4  ROT OVER + ROT ROT - ;
  5 : (LOCATE) ( addr,ct,c---addr+1 )
  6 \ return addr+1 of located char
  7  SWAP 0 DO >R 1+ DUP C@
  8  R> DUP ROT = IF  1+ LEAVE  THEN LOOP DROP 1+ ;
  9 : LEX ( string,char---string1,string2 )
 10  ROT ROT 2DUP 2DUP + ROT ROT DROP
 11  2>R ROT (LOCATE) DUP DUP 2R> >R SWAP
 12  - ROT R> SWAP OVER 1+ - 2SWAP ;
 13 : LOCATE ( string,c---addr+1,f )
 14  3 PICK 3 PICK + >R (LOCATE) DUP R> > 0= ;
 15 ;S
```

File Handling Routines

Here are some programs showing what discs are for.★

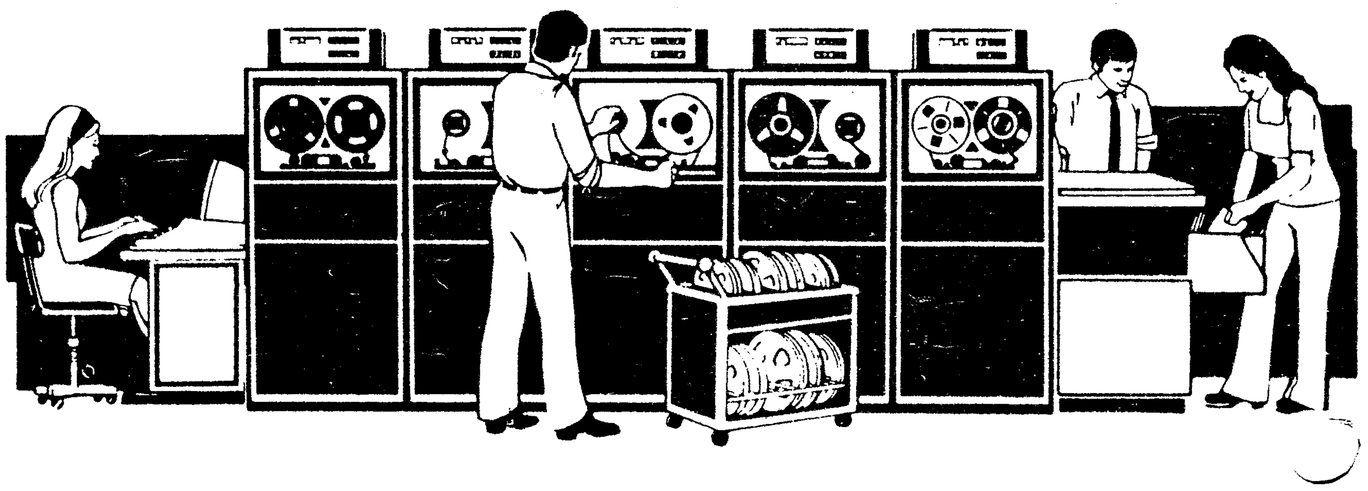
```
10 REM ************************************
20 REM MAIN MENU FOR
30 REM ADDRESS FILE
40 REM MIKE HURLEY
50 REM JAN 86
60 REM ************************************
70 REM FILE NAME  "ADDRES.BAS"
80 REM ************************************
90 DIM F$(5,8)
100 LET F$(1)="F1......": LET F$(2)="F2......": LET F$(3)="F3......": LET F$(4)=
"F4......": LET F$(5)="F5......"
110 LET B$="Add-Update-Delete": LET C$="Address Book By M.A.Hurley"
120 LET N=1
130 VS 4
140 COLOUR 0,12: COLOUR 1,1: COLOUR 2,12: COLOUR 4,12: CLS
150 CSR 3,0: PRINT C$
160 CSR 14,3: PRINT "MENU": CSR 13,4: PRINT "======"
170 FOR I=7 TO 19 STEP 3
180 CSR 2,I: PRINT F$(N)
190 LET N=N+1
200 NEXT
210 COLOUR 1,14: CSR 10,7: PRINT "Update Address File": CSR 10,10: PRINT "Search
 Address File": CSR 10,13: PRINT "Sort Address File"
220 CSR 10,16: PRINT "Print Address File": CSR 10,19: PRINT "End Session"
230 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>132 THEN  GOTO 230
240 IF A=128 THEN  GOTO 290
250 IF A=129 THEN  GOTO 330
260 IF A=130 THEN  GOTO 370
270 IF A=131 THEN  GOTO 410
280 IF A=132 THEN  NEW
290 VS 4: CLS : VS 5: PAPER 12: INK 1: CLS
300 CSR 9,10: PRINT "Update Module Loading"
310 CSR 12,12: PRINT "Please Wait...."
320 USER  LOAD "UPDATE.MEN"
330 VS 4: CLS : VS 5: PAPER 12: INK 1: CLS
340 CSR 9,10: PRINT "Search Module Loading"
350 CSR 12,12: PRINT "Please Wait...."
360 USER  LOAD "SEARCH.BAS"
370 VS 4: CLS : VS 5: PAPER 12: INK 1: CLS
380 CSR 9,10: PRINT "Sort Module Loading"
390 CSR 12,12: PRINT "Please Wait...."
```

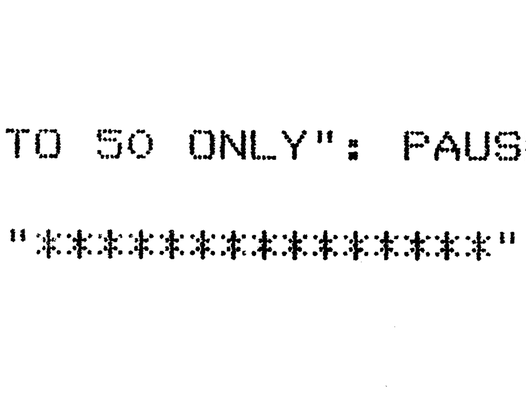
```
400 USER  LOAD "ADDSORT.BAS"
410 VS 4: CLS : VS 5: PAPER 12: INK 1: CLS
420 CSR 9,10: PRINT "Print Module Loading"
430 CSR 12,12: PRINT "Please Wait...."
440 USER  LOAD "ADDPRINT.BAS"
450 USER  SAVE "ADDRES.BAS"
460 RUN
```

```
10 REM ********************************
20 REM UPDATE MENU FOR
30 REM ADDRESS FILE
40 REM MIKE HURLEY
50 REM ********************************
60 REM FILE NAME "UPDATE.MEN"
70 REM ********************************
80 CLEAR
90 DIM F$(5,8)
100 DIM N$(50,9,31)
110 DIM EMP$(5,31)
120 LET EMP$(1)="                          ": LET EMP$(2)="    ": LET EMP$(3)="
                    ": LET EMP$(4)="           ": LET EMP$(5)="                    "
130 LET Y=0
140 LET Z=0
150 LET R=0
160 LET F$(1)="F1......": LET F$(2)="F2......": LET F$(3)="F3......": LET F$(4)=
"F4......"
170 LET B$="Add-Update-Delete"
180 LET C$="Address File By M.A.H"
190 VS 4: PAPER 12: INK 1: COLOUR 4,12: CLS
200 CSR 7,0: PRINT B$
210 FOR I=0 TO 31: PRINT "_";: NEXT
220 LET N=0
230 FOR I=7 TO 16 STEP 3
240 LET N=N+1
250 CSR 2,I: PRINT F$(N)
260 NEXT
270 COLOUR 1,14
280 CSR 10,7: PRINT "Add An Address": CSR 10,10: PRINT "Modify An Address": CSR
10,13: PRINT "Delete An Address": CSR 10,16: PRINT "Return to Main Menu"
290 COLOUR 1,1
300 CSR 0,21: FOR I=0 TO 31: PRINT "-";: NEXT
310 CSR 5,22: PRINT C$
320 LET A$=INKEY$: LET A=ASC(A$): IF A<128 OR A>131 THEN  GOTO 320
330 IF A=128 THEN  GOTO 420
340 IF A=129 THEN  GOTO 1070
350 IF A=130 THEN  GOTO 1070
360 IF A=131 THEN  GOTO 370
370 VS 4: CLS : VS 5: PAPER 12: INK 1: CLS
380 CSR 8,10: PRINT "Returning to Main Menu": CSR 12,12: PRINT "please wait..."
390 USER  LOAD "ADDRES.BAS"
400 USER  SAVE "UPDATE.MEN"
410 RUN
420 USER OPEN£1,"ADDRESS.DAT","O"
430 VS 5: PAPER 12: INK 14: CLS
440 CSR 9,0: PRINT C$
450 CSR 8,1: PRINT "——————————————————————————————"
460 CSR 0,3: PRINT "Surname."
470 CSR 8,4: PRINT " ————————————————————"
480 CSR 0,5: PRINT "Forename."
490 CSR 9,6: PRINT " ——————————————————————————————"
500 CSR 25,3: PRINT "Initials."
510 CSR 34,4: PRINT " ———"
520 CSR 0,7: PRINT "Address."
```

```
530 CSR 8,8: PRINT " ——————————————————————————————————————"
540 REM CSR 0,9: PRINT "Road."
550 CSR 5,10: PRINT " ———————————————————————————————————————"
560 REM CSR 0,11: PRINT "Town."
570 CSR 5,12: PRINT " ———————————————————————————————————————"
580 REM CSR 0,13: PRINT "County."
590 CSR 5,14: PRINT " ———————————————————————————————————————"
600 CSR 0,15: PRINT "Post Code."
610 CSR 10,16: PRINT " ——————————"
620 CSR 20,15: PRINT "Tel."
630 CSR 24,16: PRINT " ————————————————"
640 CSR 0,18: FOR I=0 TO 38: PRINT "-";: NEXT
650 CSR 0,22: FOR I=0 TO 38: PRINT "-";: NEXT
660 CSR 0,20: PRINT "F1. To Enter Data.": CSR 20,20: PRINT "F4. Return To Menu."
670 GOTO 970
680 LET R=R+1
690 CRVS 0,0,10,3,16,1,40
700 PAPER 12: INK 14
710 EDITOR N$(R,1)
720 CRVS 0,0,36,3,4,1,40
730 PAPER 12: INK 14
740 EDITOR N$(R,2)
750 CRVS 0,0,11,5,27,1,40
760 PAPER 12: INK 14
770 EDITOR N$(R,3)
780 CRVS 0,0,10,7,28,1,40
790 PAPER 12: INK 14
800 EDITOR N$(R,4)
810 CRVS 0,0,7,9,31,1,40
820 PAPER 12: INK 14
830 EDITOR N$(R,5)
840 CRVS 0,0,7,11,31,1,40
850 PAPER 12: INK 14
860 EDITOR N$(R,6)
870 CRVS 0,0,7,13,31,1,40
880 PAPER 12: INK 14
890 EDITOR N$(R,7)
900 CRVS 0,0,12,15,9,1,40
910 PAPER 12: INK 14
920 EDITOR N$(R,8)
930 CRVS 0,0,26,15,14,1,40
940 PAPER 12: INK 14
950 EDITOR N$(R,9)
960 GOTO 1030
970 LET A$=INKEY$: LET A=ASC(A$)
980 IF A=128 THEN  GOTO 680
990 IF A=131 THEN  GOTO 1010
1000 GOTO 970
1010 USER CLOSE£1
1020 GOTO 190
1030 FOR I=1 TO 9
1040 USER  PRINT £1,N$(R,I)
1050 NEXT I
1060 GOTO 430
1070 VS 4: CLS
1080 CSR 2,9: PRINT "F1... To Modify/Delete Data": CSR 2,12: PRINT "F4... Return
 To Menu"
1090 LET A$=INKEY$: LET A=ASC(A$)
1100 IF A=128 THEN  GOTO 1130
1110 IF A=131 THEN  GOTO 190
1120 GOTO 1090
1130 USER OPEN£1,"ADDRESS.DAT","I"
1140 CSR 4,20: PRINT "PLEASE WAIT LOADING DATA"
1150 FOR I=1 TO 50
1160 USER EOF£1,1210
1170 LET Y=Y+1
1180 FOR N=1 TO 9
1190 USER  INPUT £1,N$(I,N)
1200 NEXT N: NEXT I
1210 LET Z=Y
1220 USER CLOSE£1
1230 VS 5: PAPER 4: INK 10: CLS
1240 CSR 11,1: PRINT "ADDRESS DIRECTORY"
1250 FOR I=0 TO 38
1260 CSR I,2: PRINT "-"
1270 NEXT
1280 FOR I=1 TO 26
1290 IF I>=14 THEN  GOTO 1330
1300 CSR 0,2+I: PRINT I
1310 CSR 4,2+I: PRINT N$(I,1)
1320 GOTO 1350
1330 CSR 20,2+I-13: PRINT I
1340 CSR 24,2+I-13: PRINT N$(I,1)
1350 NEXT
1360 CSR 0,16
1370 FOR I=1 TO 38
1380 PRINT "-";
1390 NEXT
```

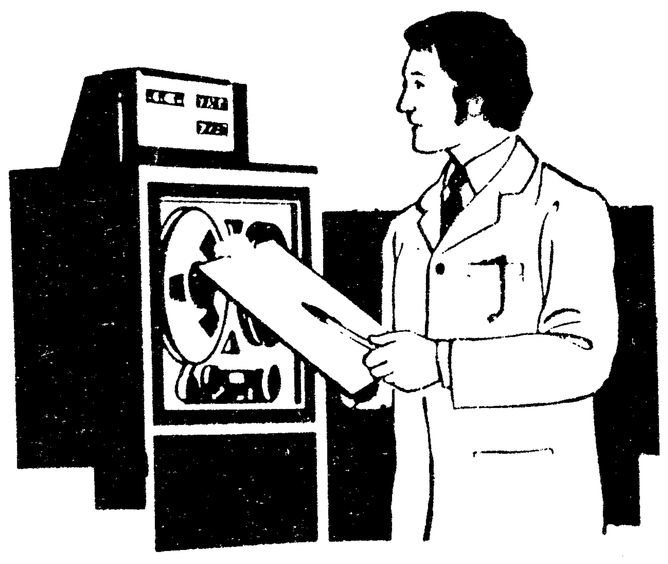
```
1400 CSR 16,17: PRINT "OPTIONS"
1410 CSR 0,19: PRINT "F1. Next Screen": CSR 0,20: PRINT "F2. Modify/Delete Addre
ss"
1420 CSR 19,19: PRINT "F4. Return To Menu"
1430 LET A$=INKEY$: LET A=ASC(A$)
1440 IF A=128 THEN  GOTO 1480
1450 IF A=129 THEN  GOTO 1710
1460 IF A=131 THEN  GOTO 190
1470 GOTO 1430
1480 CLS : CSR 11,1: PRINT "ADDRESS.DIRECTORY"
1490 FOR I=0 TO 38
1500 CSR I,2: PRINT "-"
1510 NEXT
1520 FOR I=27 TO 50
1530 IF I>39 THEN  GOTO 1570
1540 CSR 0,2+I-26: PRINT I
1550 CSR 4,2+I-26: PRINT N$(I,1)
1560 GOTO 1590
1570 CSR 20,2+I-39: PRINT I
1580 CSR 24,2+I-39: PRINT N$(I,1)
1590 NEXT
1600 FOR I=0 TO 38
1610 CSR I,16: PRINT "-"
1620 NEXT
1630 CSR 16,17: PRINT "OPTIONS"
1640 CSR 0,19: PRINT "F1. Prev Screen": CSR 0,20: PRINT "F2. Modify/Delete Addre
ss"
1650 CSR 19,19: PRINT "F4. Return To Menu"
1660 LET A$=INKEY$: LET A=ASC(A$)
1670 IF A=128 THEN  GOTO 1230
1680 IF A=129 THEN  GOTO 1710
1690 IF A=131 THEN  GOTO 190
1700 GOTO 1660
1710 CSR 0,21: INPUT "Enter Directory Number > ";N
1720 IF N<1 OR N>50 THEN  CSR 10,22: PRINT CHR$(7);"ERROR 1 TO 50 ONLY": PAUSE 3
000: CSR 0,22: PRINT CHR$(5): GOTO 1710 ELSE  GOTO 1730
1730 CLS : CSR 13,2: PRINT "MODIFY/DELETE": CSR 12,3: PRINT "**************"
1740 FOR I=1 TO 9
1750 CSR 0,5+I: PRINT N$(N,I)
1760 NEXT
1770 PRINT : PRINT
1780 INPUT "Is this the correct address   Y/N > ";E$
1790 IF E$="y" OR E$="Y" THEN  GOTO 1810
1800 IF E$="n" OR E$="N" THEN  GOTO 1230
1810 PRINT : PRINT "Enter 'M' To Modify Address": PRINT "Enter 'D' To Delete Add
ress"
1820 PRINT : INPUT "Enter M or D  --> ";E$
1830 IF E$="M" OR E$="m" THEN  GOTO 1850
1840 IF E$="d" OR E$="D" THEN  GOTO 2400 ELSE  GOTO 1820
1850 VS 5: CLS : CSR 16,0: PRINT "MODIFY"
1860 CSR 15,1: PRINT "----------"
1870 CSR 0,3: PRINT "Surname."
1880 CSR 8,4: PRINT " ------------------------"
1890 CSR 0,5: PRINT "Forename."
1900 CSR 9,6: PRINT " ------------------------------------------"
1910 CSR 25,3: PRINT "Initials."
1920 CSR 34,4: PRINT " ---"
1930 CSR 0,7: PRINT "Address."
1940 CSR 8,8: PRINT " ------------------------------------------"
1950 CSR 5,10: PRINT " ------------------------------------------"
1960 CSR 5,12: PRINT " ------------------------------------------"
1970 CSR 5,14: PRINT " ------------------------------------------"
1980 CSR 0,15: PRINT "Post Code."
1990 CSR 10,16: PRINT " ----------"
2000 CSR 20,15: PRINT "Tel."
```





```
2010 CSR 24,16: PRINT " ------------------"
2020 CSR 0,18: FOR I=0 TO 38: PRINT "-";: NEXT
2030 CSR 0,22: FOR I=0 TO 38: PRINT "-";: NEXT
2040 CSR 1,20: PRINT "Press <RET> to leave field unchanged"
2050 CSR 9,3: PRINT N$(N,1)
2060 CSR 35,3: PRINT N$(N,2)
2070 CSR 10,5: PRINT N$(N,3)
2080 CSR 9,7: PRINT N$(N,4)
2090 CSR 6,9: PRINT N$(N,5)
2100 CSR 6,11: PRINT N$(N,6)
2110 CSR 6,13: PRINT N$(N,7)
2120 CSR 11,15: PRINT N$(N,8)
2130 CSR 25,15: PRINT N$(N,9)
2140 LET N$(N,1)=EMP$(1): LET N$(N,2)=EMP$(2): LET N$(N,8)=EMP$(4): LET N$(N,9)=
EMP$(5)
2150 FOR I=3 TO 7: LET N$(N,I)=EMP$(3): NEXT
2160 CRVS 0,0,10,3,16,1,40: PAPER 4: INK 10: EDITOR N$(N,1)
2170 CRVS 0,0,36,3,4,1,40: PAPER 4: INK 10: EDITOR N$(N,2)
2180 CRVS 0,0,11,5,27,1,40: PAPER 4: INK 10: EDITOR N$(N,3)
2190 CRVS 0,0,10,7,28,1,40: PAPER 4: INK 10: EDITOR N$(N,4)
2200 CRVS 0,0,7,9,31,1,40: PAPER 4: INK 10: EDITOR N$(N,5)
22  CRVS 0,0,7,11,31,1,40: PAPER 4: INK 10: EDITOR N$(N,6)
2220 CRVS 0,0,7,13,31,1,40: PAPER 4: INK 10: EDITOR N$(N,7)
2230 CRVS 0,0,12,15,9,1,40: PAPER 4: INK 10: EDITOR N$(N,8)
2240 CRVS 0,0,26,15,14,1,40: PAPER 4: INK 10: EDITOR N$(N,9)
2250 PRINT CHR$(7);CHR$(7)
2260 VS 5: CLS : CSR 9,6: PRINT "Do you want to modify": CSR 9,8: INPUT "another
 record Y/N  ";E$
2270 IF E$="N" OR E$="n" THEN  GOTO 2280 ELSE  IF E$="Y" OR E$="y" THEN  GOTO 12
30 ELSE  GOTO 2260
2280 PRINT : PRINT "   SAVING CHANGED DATA PLEASE WAIT..."
2290 USER OPEN£1,"BACKUP.DAT","O"
2300 FOR I=1 TO Z
2310 IF LEFT$(N$(I,1),5)="DELXY" AND LEFT$(N$(I,3),4)="XYDE" AND LEFT$(N$(I,4),3
)="BYE" THEN  GOTO 2350
2320 FOR N=1 TO 9
2330 USER  PRINT £1,N$(I,N)
2340 NEXT N
2350 NEXT I
2360 USER CLOSE£1
23  USER ERA"ADDRESS.DAT"
2380 USER REN"ADDRESS.DAT"="BACKUP.DAT"
2390 GOTO 80
2400 VS 5: PAPER 6: INK 1: CLS
2410 CSR 6,3: PRINT "ARE YOU SURE YOU WANT TO": CSR 10,5: PRINT "**** DELETE ***
*": CSR 12,7: PRINT "THIS RECORD  ? "
2420 FOR I=1 TO 9
2430 CSR 0,10+I: PRINT N$(N,I)
2440 NEXT
2450 INPUT "ENTER 'Y' (Yes)  'N' (no) -->";ANS$
2460 IF ANS$="Y" OR ANS$="y" THEN  GOTO 2480
2470 IF ANS$="N" OR ANS$="n" THEN  GOTO 190 ELSE  GOTO 2450
2480 LET N$(N,1)="DELXY": LET N$(N,3)="XYDE": LET N$(N,4)="BYE"
2490 GOTO 2280


10 REM ********************************
20 REM SEARCH FILE FOR
30 REM ADDRESS FILE
40 REM MIKE HURLEY
50 REM JAN 86
60 REM ********************************
70 REM FILE NAME "SEARCH.BAS"
80 REM ********************************
```

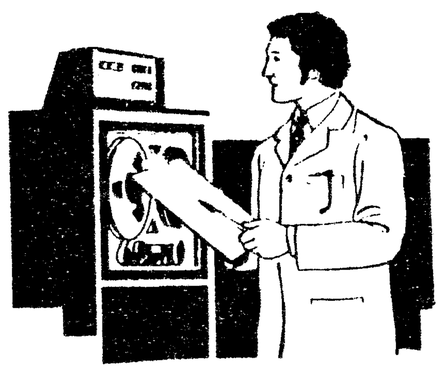
```
90 DIM N$(50,9,31)
100 VS 5: PAPER 4: INK 7: CLS
110 CSR 10,2: PRINT "SEARCH ADDRESS FILE"
120 CSR 9,3: PRINT "************************"
130 CSR 0,6: PRINT "Search Address File By Field."
140 PRINT : PRINT "Which Field do you want to do a search on ?"
150 PRINT
160 PRINT "0. Return To Main Menu"
170 PRINT "1. Surname.": PRINT "2. Initials.": PRINT "3. First Name.": PRINT "
 Address Fields 4,5,6 or 7.": PRINT "8. Post code.": PRINT "9. Telephone No."
180 CSR 0,20: INPUT "Enter Number --> ";S
190 IF S<0 OR S>9 THEN  CSR 0,21: PRINT CHR$(7);"ERROR 1 TO 9 ONLY": PAUSE 3000:
 CSR 0,21: PRINT CHR$(5): GOTO 180
200 IF S=0 THEN  GOTO 210 ELSE  GOTO 220
210 USER  LOAD "ADDRES.BAS"
220 USER OPEN£1,"ADDRESS.DAT","I"
230 CSR 4,22: PRINT "PLEASE WAIT LOADING DATA"
240 FOR I=1 TO 50
250 USER EOF£1,290
260 FOR N=1 TO 9
270 USER  INPUT £1,N$(I,N)
280 NEXT N: NEXT I
290 USER CLOSE£1
300 VS 5: PAPER 4: INK 7: CLS
310 CSR 11,1: PRINT "ADDRESS DIRECTORY"
320 FOR I=0 TO 38
330 CSR I,2: PRINT "-"
340 NEXT
350 FOR I=1 TO 26
360 IF I>=14 THEN  GOTO 400
370 CSR 0,2+I: PRINT I
380 CSR 4,2+I: PRINT N$(I,S)
390 GOTO 420
400 CSR 20,2+I-13: PRINT I
410 CSR 24,2+I-13: PRINT N$(I,S)
420 NEXT
430 CSR 0,16
440 FOR I=1 TO 38
450 PRINT "-";
460 NEXT
470 CSR 16,17: PRINT "OPTIONS"
480 CSR 0,19: PRINT "F1. Next Screen"
490 CSR 19,19: PRINT "F4. Return To Menu"
500 CSR 0,20: PRINT "F2. List Address"
510 LET A$=INKEY$: LET A=ASC(A$)
520 IF A=128 THEN  GOTO 560
530 IF A=129 THEN  GOTO 800
540 IF A=131 THEN  GOTO 100
550 GOTO 510
560 CLS : CSR 11,1: PRINT "ADDRESS.DIRECTORY"
570 FOR I=0 TO 38
580 CSR I,2: PRINT "-"
590 NEXT
600 FOR I=27 TO 50
610 IF I>39 THEN  GOTO 650
620 CSR 0,2+I-26: PRINT I
630 CSR 4,2+I-26: PRINT N$(I,S)
640 GOTO 670
650 CSR 20,2+I-39: PRINT I
660 CSR 24,2+I-39: PRINT N$(I,S)
670 NEXT
680 FOR I=0 TO 38
690 CSR I,16: PRINT "-"
```



```
700 NEXT
710 CSR 16,17: PRINT "OPTIONS"
720 CSR 0,19: PRINT "F1. Prev Screen"
730 CSR 19,19: PRINT "F4. Return To Menu"
740 CSR 0,20: PRINT "F2. List Address"
750 LET A$=INKEY$: LET A=ASC(A$)
760 IF A=128 THEN  GOTO 300
770 IF A=129 THEN  GOTO 800
780 IF A=131 THEN  GOTO 100
790 GOTO 750
800 CSR 0,21: INPUT "Enter Directory Number > ";N
810 IF N<1 OR N>50 THEN  CSR 10,22: PRINT CHR$(7);"ERROR 1 TO  50 ONLY": PAUSE 3
000: CSR 0,2: PRINT CHR$(5): GOTO 800 ELSE  GOTO 820
820 CLS
830 FOR I=1 TO 9
840 CSR 4,5+I: PRINT N$(N,I)
850 NEXT
860 CSR 6,21: PRINT "F4. To Return To Directory": CSR 6,22: PRINT "_____________
_____________"
870 LET A$=INKEY$: LET A=ASC(A$)
880 IF A=131 THEN  GOTO 300
890 GOTO 870
900 STOP
910 USER  SAVE "SEARCH.BAS"
920 RUN
```

```
 10 REM  ************************
 20 REM SORT PROGRAM FOR
 30 REM ADDRESS FILE
 40 REM MIKE HURLEY
 50 REM JAN 86
 60 REM ************************
 70 REM FILE NAME "ADDSORT.BAS"
 80 REM ************************
 90 DIM N$(50,9,31),TEMP$(50,9,31)
100 LET Y=0
110 LET L=0
120 VS 5: PAPER 13: INK 14: CLS
130 CSR 17,2: PRINT "SORT": CSR 16,3: PRINT "******"
140 PRINT : PRINT : PRINT "This procedure will  load Data from     the Address fi
le and automaticaly do   an APLHABETIC SORT."
150 CSR 0,12: INPUT "Do You Want To Continue Y/N  > ";A$
160 IF A$="Y" OR A$="y" THEN  GOTO 210
170 IF A$="N" OR A$="n" THEN  GOTO 190
180 GOTO 150
190 PRINT : PRINT "Returning To Main Menu"
200 USER  LOAD "ADDRES.BAS"
210 USER OPEN£1,"ADDRESS.DAT","I"
220 PRINT : PRINT "PLEASE WAIT LOADING DATA"
230 FOR I=1 TO 50
240 USER EOF£1,290
250 LET Y=Y+1
260 FOR N=1 TO 9
270 USER  INPUT £1,N$(I,N)
280 NEXT N: NEXT I
290 USER CLOSE£1
300 PRINT : PRINT "SORT IN PROGRESS PLEASE WAIT."
310 FOR K=1 TO Y-1
320 FOR I=1 TO Y-1
330 IF N$(I,1)<=N$(I+1,1) THEN  GOTO 430
340 FOR M=1 TO 9
350 LET TEMP$(I,M)=N$(I,M)
360 NEXT M
```

```
370 FOR M=1 TO 9
380 LET N$(I,M)=N$(I+1,M)
390 NEXT M
400 FOR M=1 TO 9
410 LET N$(I+1,M)=TEMP$(I,M)
420 NEXT M
430 NEXT I
440 NEXT K
450 PRINT "SAVING SORTED DATA PLEASE WAIT..."
460 USER OPEN£1,"SORT.BAK","O"
470 FOR I=1 TO Y
480 FOR N=1 TO 9
490 USER  PRINT £1,N$(I,N)
500 NEXT
510 NEXT
520 USER CLOSE£1
530 USER ERA"ADDRESS.DAT"
540 USER REN"ADDRESS.DAT"="SORT.BAK"
550 PAUSE 3000: GOTO 190
560 STOP
570 USER  SAVE "ADDSORT.BAS"
580 RUN
```



```
10 REM ***************************
20 REM PRINT PROGRAM FOR
30 REM ADDRESS FILE
40 REM MIKE HURLEY
50 REM JAN 86
60 REM ***************************
70 REM FILE NAME "ADDPRINT.BAS"
80 REM ***************************
90 DIM N$(50,9,31)
100 LET Y=0
110 LPRINT CHR$(27);"@";
120 VS 5: PAPER 4: INK 7: CLS
130 CSR 13,0: PRINT "PRINT OPTIONS"
140 FOR I=0 TO 38: PRINT "-";: NEXT
150 PRINT
160 PRINT
170 PRINT "1. ELITE PITCH      2. PICA PITCH"
180 CSR 0,6: INPUT "Enter 1 or 2 > ";T
190 IF T<0 OR T>2 THEN  CSR 0,7: PRINT "ERROR 1 OR 2 ONLY": CSR 0,7: PAUSE 3000:
 PRINT CHR$(5): GOTO 180
200 IF T=1 THEN  LPRINT CHR$(27);"P";CHR$(1);
210 IF T=2 THEN  LPRINT CHR$(27);"P";CHR$(0);
220 CSR 0,8: PRINT "TYPE OF CHARACTER ?"
225 CSR 0,10: PRINT "0. NORMAL  "
230 CSR 0,11: PRINT "1. DOUBLE WIDTH ELONGATED ": PRINT "2. EMPHASIZED": PRINT "
3. DOUBLE PRINTED": PRINT "4. UNDERLINE": PRINT "5. ITALIC"
240 PRINT "6. EMPHASIZED AND DOUBLE WIDTH": PRINT "7. EMPHASIZED AND DOUBLE PRIN
TED"
250 CSR 0,19: INPUT "Enter Option 0 to 7 > ";O
260 IF O<0 OR O>7 THEN  CSR 0,20: PRINT "ERROR 1 OR 2 ONLY": CSR 0,20: PAUSE 300
0: PRINT CHR$(5): GOTO 250
270 ON O GOSUB 430,290,310,330,350,370,390,410
280 GOTO 430
290 LPRINT CHR$(27);"W";CHR$(1);
300 RETURN
310 LPRINT CHR$(27);"E";
320 RETURN
330 LPRINT CHR$(27);"G";
340 RETURN
```

```
350 LPRINT CHR$(27);"-";CHR$(1);
360 RETURN
370 LPRINT CHR$(27);"4";
380 RETURN
390 LPRINT CHR$(27);"E";: LPRINT CHR$(27);"W";CHR$(1);
400 RETURN
410 LPRINT CHR$(27);"E";: LPRINT CHR$(27);"G";
420 RETURN
430 PRINT "PLEASE WAIT LOADING DATA"
440 USER OPEN£1,"ADDRESS.DAT","I"
450 FOR I=1 TO 50
460 USER EOF£1,510
470 LET Y=Y+1
480 FOR N=1 TO 9
490 USER  INPUT £1,N$(I,N)
500 NEXT N: NEXT I
510 USER CLOSE£1
520 CSR 0,21: INPUT "DO YOU WANT TO PRINT THE WHOLE FILE Y/N";A$
530 IF A$="Y" OR A$="y" THEN  GOTO 560
540 IF A$="N" OR A$="n" THEN  GOTO 650
550 GOTO 520
560 FOR I=1 TO Y
565 LPRINT N$(I,3)+" "+N$(I,2)+" "+N$(I,1)
570 FOR N=4 TO 9
580 LPRINT N$(I,N)
590 NEXT N
600 FOR L=1 TO 5: LPRINT CHR$(10);
610 NEXT L
620 NEXT I
630 PRINT "Returning To Main Menu"
640 USER  LOAD "ADDRES.BAS"
650 VS 5: CLS
660 CSR 11,1: PRINT "ADDRESS DIRECTORY"
670 FOR I=0 TO 38
680 CSR I,2: PRINT "-"
690 NEXT
700 FOR I=1 TO 26
710 IF I>=14 THEN  GOTO 750
720 CSR 0,2+I: PRINT I
730 CSR 4,2+I: PRINT N$(I,1)
740 GOTO 770
750 CSR 20,2+I-13: PRINT I
760 CSR 24,2+I-13: PRINT N$(I,1)
770 NEXT
780 CSR 0,16
790 FOR I=0 TO 38
800 PRINT "-";
810 NEXT
820 CSR 16,17: PRINT "OPTIONS"
830 CSR 0,19: PRINT "F1. Next Screen": CSR 19,19: PRINT "F2. Print Address"
840 CSR 0,20: PRINT "F4. Return To Main Menu"
850 LET A$=INKEY$: LET A=ASC(A$)
860 IF A=128 THEN  GOTO 900
870 IF A=129 THEN  GOTO 1150
880 IF A=131 THEN  GOTO 630
890 GOTO 850
900 REM DIRECTORY SCREEN TWO
910 REM **********************
920 CLS : CSR 11,1: PRINT "ADDRESS DIRECTORY"
930 FOR I=0 TO 38
940 CSR I,2: PRINT "-"
950 NEXT
960 FOR I=27 TO 50
970 IF I>39 THEN  GOTO 1010
```

```
980 CSR 0,2+I-26: PRINT I
990 CSR 4,2+I-26: PRINT N$(I,1)
1000 GOTO 1030
1010 CSR 20,2+I-39: PRINT I
1020 CSR 24,2+I-39: PRINT N$(I,1)
1030 NEXT
1040 FOR I=0 TO 38
1050 CSR I,16: PRINT "-"
1060 NEXT
1070 CSR 16,17: PRINT "OPTIONS"
1080 CSR 0,19: PRINT "F1. Prev Screen": CSR 19,19: PRINT "F2. Print Address"
1090 CSR 0,20: PRINT "F4. Return To Main Menu"
1100 LET A$=INKEY$: LET A=ASC(A$)
1110 IF A=128 THEN  GOTO 650
1120 IF A=129 THEN  GOTO 1150
1130 IF A=131 THEN  GOTO 630
1140 GOTO 1100
1150 CSR 0,21: INPUT "Enter Directory Number > ";N
1160 IF N<1 OR N>50 THEN  CSR 10,22: PRINT CHR$(7);"ERROR 1 TO 50 ONLY": PAUSE 3
000: CSR 0,22: PRINT CHR$(5): GOTO 1150 ELSE  GOTO 1170
1170 CLS
1180 FOR I=1 TO 9
1190 CSR 0,5+I: PRINT N$(N,I)
1200 NEXT
1210 PRINT : PRINT
1220 INPUT "Is this the  correct address Y/N >";E$
1230 IF E$="Y" OR E$="y" THEN  GOTO 1252
1240 IF E$="n" OR E$="N" THEN  GOTO 650
1250 GOTO 1220
1252 LPRINT N$(N,3)+" "+N$(N,2)+" "+N$(N,1)
1260 FOR I=4 TO 8
1270 LPRINT N$(N,I)
1280 NEXT
1290 GOTO 650
1300 USER  SAVE "ADDPRINT.BAS"
1310 RUN
```

Alice, Mission Alphatrom, Tachyon Fighter, Star Command, Memosketch, Nemo, Crystal, Kilopede at £2.50 each.

Adventure Quest, Emerald Isle and Snowball at £4.00 each.

From: Gordon Hurd, 10 Ardmore, Vicarage Road,
Leigh Woods, Bristol BS8 3PH.
Telephone 0272 734769.

View Letters TO THE EDITOR point

Following letter received from I C Price, 15 Briscoe Close Hoddesdon Herts

This is a plea for 'Noddy' since there seems to be some resistance to the use of this facility on the MTX. Looking through the last half dozen or so Memopads I notice that menus and the like are usually written out in Basic together with those cumbersome CSR commands.

Why not use Noddy? I find it quicker to type in and its easier to arrange the format rather than working out the CSR instructions.

Can I suggest the following alternative to lines 30 to 120 inclusive in the Disc Database program in Memopad No. 6?

```
30   CLS: PAPER P : INK I
40   PLOD "PROG"
50   PRINT CHR$(7)
60   IF INKEY$<>" "THEN GOTO 60
70   LET IN$  = INKEY$: IF IN$ =" "THEN GOTO 70
80   IF IN$ <"1" OR IN$> "g" THEN GOTO 80
```

Noddy pages

PROG *D MENU. * R

MENU MAIN MENU
etc.

The Noddy title for the page MENU would be deleted for neatness on the finished page.

I've also added lines 50,60 & 70 which are not all strictly necessary. Line 50 sounds the bell to let me know the program is waiting, line 60 prevents a run on from a previous key depression and line 70 allows a quicker check to ensure a key has actually been depressed.

I'm not setting myself up as any sort of expert on Noddy but I do find it very convenient for Introduction, Instruction and Menu pages.

FORTH PRINTER AMENDMENT

To alter your version of the printer routine, first load FORTH and then exit to Basic (i.e. hold ESC key down).

Then enter panel and display from 5667H, alter the bytes from 5667H as below.

5667H	00H	NOP
	00H	NOP
	00H	NOP
	00H	NOP
	00H	NOP
	00H	NOP

```
            00H   NOP
566EH       00H   NOP
566FH Wait: DB,04 IN A,(04)  ;  Get printer status
            E6,0F AN1 0FH    ;  Discard MSN
            FE,0A CP 0AH     ;  Is printer busy
            20,F8 JR NZ,WAIT ;  Wait if yes
            78    LD A,B     ;  Get byte to print
            D3,04 OUT(04),A  ;  Send it
            DB,00 IN A,(00)  ;  Send data strobe
            DB,04 IN A,(04)
            00    NOP
567FH       C9    RET        ;  Return
```

Go back into FORTH and re-save your new version : this should run on any centronics printer.

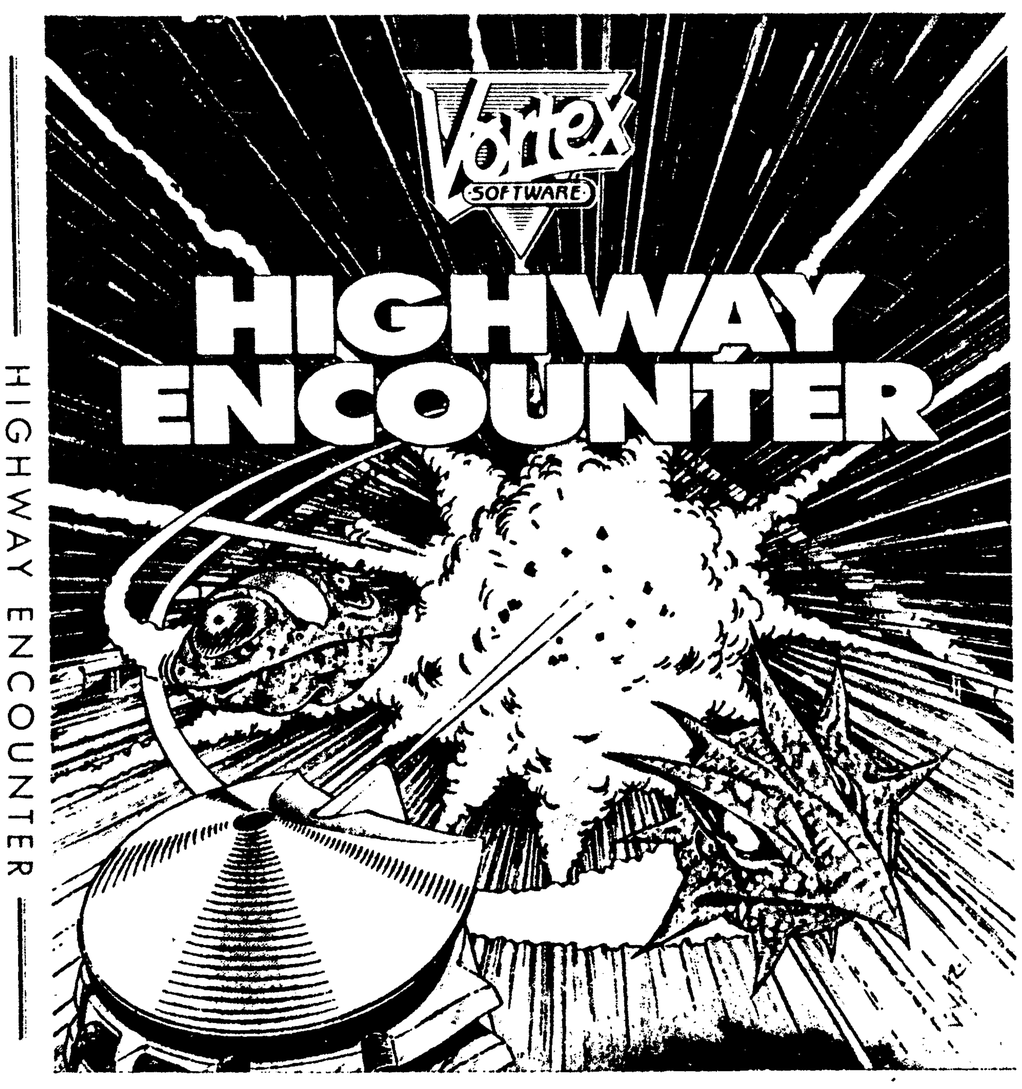
Query from Leslie Gornall, The Old School, Binchester, Bishop Auckland, Co. Durham, DL14 8AU

I would be grateful if a fellow reader could send me a program (in assembly language) to take input from any key on the keyboard and store the key pressed in a memory location. I tried to write such a program myself - but fell on the problem that many keys seem to have the same code. ✱

Special Offer

BUY ANY 2 PROGRAMS
&
GET 'MINER DICK' FREE!

Offer lasts for one month only



Number 1 game on the Amstrad. Now available on the Memotech at only £7.95

Syntaxsoft

MTX Screen Dump by A.F Wilson

Before discussing how to use simple software to dump a graphics or text screen to a centronics printer, let us first take a brief look at the hardware side.

A centronics interface, unlike its rival serial interface, sends a 8-bit number (or 1 byte) along 8 parallel wires. Each of these wires represents the '0's & '1's of the binary number as 0 or 5 volts respectively, in otherwords as electrical signals. The centronics interface sends 1 pulse down each of the eight wires simultaneously, i.e. in parallel, whereas the serial interface sends 8 pulses down just the one wire one after each other, i.e. in series. If you turn to page 253 of the MTX manual you will see that there is more to the centronics interface than the 8 data lines.

The ACKNLG signal or acknowledge signal tells the computer that the printer is ready.

The BUSY signal tells the computer that the line is full or busy at the moment.

The PE signal tells the computer to stop sending data until the printer receives more paper.

The ERROR line checks for other errors that may occur.

The GND signal indicates what the ground voltage level of the computer is.

The SLCT signal is used to indicate that the computer has selected to send the data to a printer device.

The last and most important signal is the STROBE. This is the timing signal which tells the computer when to send more data.

The MTX uses the commands IN (0),d ;IN (4),d ; and OUT (4),d to talk to the printer. (See pages 248-249 of MTX manual). Briefly:-

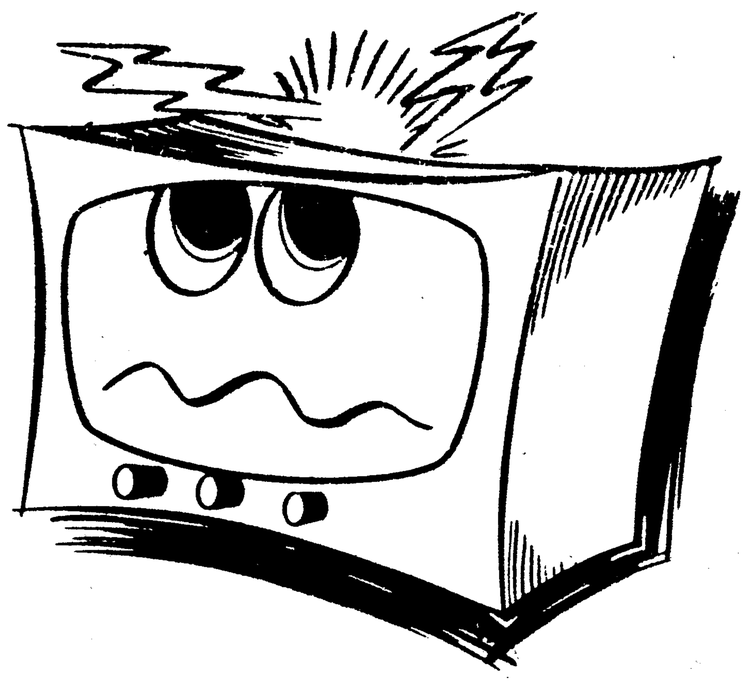
IN (0),d is used to set the strobe low for sending data.
IN (4),d is used to monitor the status of the printer. Only the lower 4 bits (or nybble) is used:

D0=busy, when high the line is busy
D1=error, when low there is an error
D2=paper empty, when high then empty
D3=slct, when high the printer is selected.

OUT (4),d is used to send the byte of data, only when the strobe is forced low and the status line is ten (#0A). Note after strobing low use IN (4),d to force it high again after 1 microsecond has elapsed.

If you browse through the MTX rom from #0CE3 then you will see how the MTX sends data to the printer, i.e.

```
#0CE0     CALL #0CF3
#0CE3     IN A,(4)       ;READ STATUS
          AND #0F        ;SELECT LOWER NYBBLE & CLEAR FLAGS
          CP #0A         ;CHECK IF CONDITIONS RIGHT TO SEND DATA
          JR NZ,#0CE0    ;ELSE GOTO #0CE3
          LD A,B         ;LET A=B,A IS USED TO OUTPUT DATA
          OUT (4),A      ;SEND DATA TO DEVICE
          IN (0),A       ;STROBE THE DATA TO THE PRINTER
          IN A,(4)       ;RESET STROBE
          RET            ;RETURN FROM CALLER
#0CF3     CALL #09F2
          JP NZ,#0259    ;IF BRK PRESSED THEN Z FLAG NOT ZERO.
```



```
        RET             ;BACK TO #0CE3 & SEE IF THE LINE IS READY.
#09F2   PUSH HL         ;SAVE HL
        LD HL,#FD81     ;POINT HL BREAK FLAG
        BIT 0,(HL)      ;HAS BRK BEEN PRESSED.
        POP HL          ;RESTORE HL
        RET
```

As you can see the MTX uses register B to store the data which is to be passed to routine #0CE3, which sends the data to the printer. On the MTX we have three main screens:-graphics, text and panel. Fortuntely both the panel and text screens are just the VS 5 . They both have the same text graphic generator and window. The text mode in the MTX is the most favourable print dump because it only needs to send 960 bytes (24 x 40) of information, i.e. one screen to the printer. Both it and its mirror-panel send only the ascii codes. The printer takes these codes and prints out its equivalent ascii coded data from its own character set.

Unfortunately we cannot use any redefined characters in this mode as the printer character set only uses the ascii code number and not the computers data for the code. But this mode has the advantage of all the functions as shown in table 1. This means we have complete control of the character font, length of page etc. The text mode is very easy to program and is shown in listing 2. Note this routine can be used for the front panel except that the lines in listing 3 should be inserted at position shown. This code is explained in the listing. To show how flexible this mode is we can very easily adjust the width of the text from 40 to 80 by changing the number in the DB 27,"Q",40 line. This will mean we will have 12 lines of 80 columns instead of 24 lines of 40 columns.

The graphics mode on the otherhand is more complex due to the way the printer treats graphic data. Also we don't have any of the functions descrbed in table 1.

The MTX treats each byte of screen data as a horizontal sequence of dots, e.g. d7 d6 d5 d4 d3 d2 d1 d0 = 1 horizontal byte. But in today's computer world, the epson printer seems to be the industry standard or at least the most popular one. This is a slight problem because the epson printers bit image mode treats the data as vertical sequences of data as opposed to the MTX method. On my HR-5 printer there are two density modes-normal (480 vertical sequences) and double density (960 vertical sequences). In basic we would represent the normal mode as:

```
LPRINT CHR$(27);"K";CHR$(n);CHR$(m)
```

where n = 0 to #ff when m=0 and n=0 to #e0 when m=#1, n=LSB and m=MSB

Therefore on the MTX we have to take our 8x8 matrix (or 8 bytes of horizontal data) and rotate these into 8 vertical bytes of data. Fortunately the coding for this is relatively simple, see listing 4, section epson, for description. This technique of twisting 8 screen bytes or an alphanumeric character around is called Transposition. Note we have to perform the line feed, etc. which was done for us before automatically by the column code.★

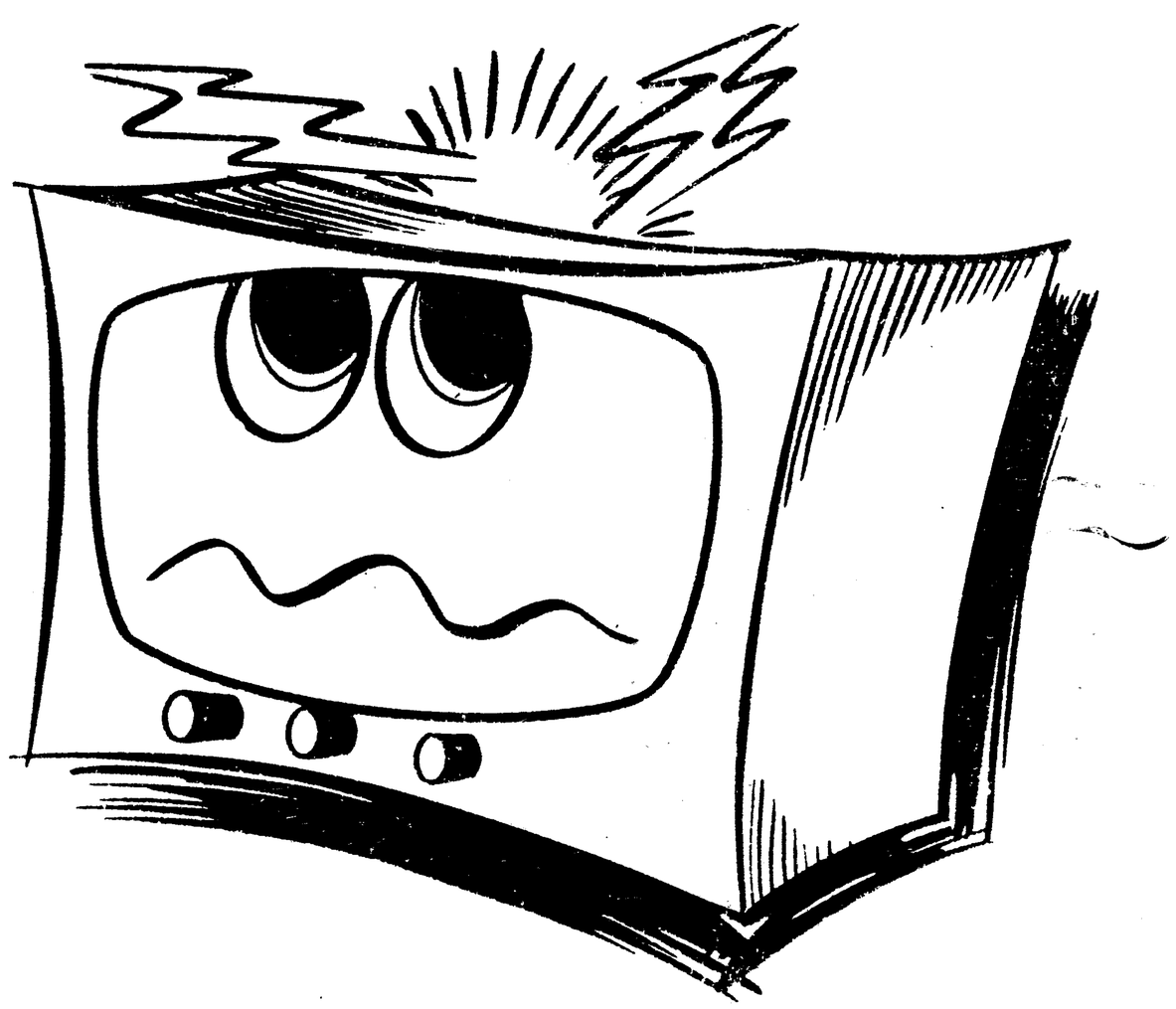
Table 1: epson printer control code data, for use in listings 2-4.

FUNCTION	Coded data for on	coded data for off
Enlarge	DB 27,"W",1	DB 27,"W",0
Reduce	DB 15,0,0	DB 18,0,0
Emphasise	DB 27,"E",0	DB 27,"F",0
Elite/Pica	DB 27,"P",0	DB 27,"P",1
Italic	DB 27,"4",0	DB 27,"5",0
Double strike	DB 27,"G",0	DB 27,"H",0
Underline	DB 27,"-",1	DB 27,"-",0
Unidirection	DB 27,"U",1	DB 27,"U",0
Subscript	DB 27,"S",1	DB 27,"T",0
Superscript	DB 27,"S",0	DB 27,"T",0
Backspace	DB 8,0,0	
Line feed	DB 27,"A",n where n=n*1/72"	
Reset printer	DB 27,64,0	

Listings 2-4: listing 2 starts at Panel;listing 3 starts at Text;listing 4 starts at graphic.

```
0 CODE

4007 JPANEL: LD A,£C3
4009         LD (£FA9E),A       ;SET FEXPAND,IE PANEL EXTENSION TO JUMP
400C         LD HL,PANEL
400F         LD (£FA9F),HL      ;POINT FEXPAND TO START OF PANEL DUMP
4012 PANEL:  CP "P"             ;WHEN IN PANEL PRESS P TO DUMP TO PRINTER
4014         RET NZ             ;ELSE BACK TO PANEL
4015 TEXT:   NOP                ;TEXT IS USED FOR BOTH VS 5/PANEL DUMPS
4016         LD HL,7168         ;START OF TEXT/PANEL SCREEN IN VRAM
4019         CALL VRAMIN        ;SET VRAM TO READ TABLE AT STARTING AT 7168
401C         LD HL,LINEFTP      ;PRINTER CODES TO BE SENT TO PRINTER
401F         LD C,6             ;6 CODES
4021         CALL P01           ;CALL SUBROUNTINE P01
4024         LD DE,960          ;LENGTH OF TEXT/PANEL SCREEN
  27 TPOUT:  IN A,(1)           ;READ IN ASCII CODES FROM VRAM
4029         LD B,A
402A         CALL £0CE3         ;SEND TO PRINTER
402D         DEC DE
402E         LD A,D
402F         OR E
4030         JR NZ,TPOUT
4032         RET
4033 LINEFTP:DB 27,"A",12       ;SET LINE FEED TO NORMAL
4036 WIDTHTP:DB 27,"Q",40       ;SET COLUMN MAX TO 40,THEN LINE FEED
4039 LINEFGM:DB 27,"A",8
403C WIDTHGM:DB 27,"K",0,1      ;SET TO NORMAL BIT MODE
4040 PBUFF:  DS 8               ;TEMPORARY BUFFER FOR 8X8 GRAPHIC MATRIX
4048 COUNTER:DS 2               ;KEEP TRACK OF POSITION ON VS 4
404A POUT:   LD C,4
404C P01:    LD B,(HL)
404D         CALL £0CE3         ;SUBROUTINE USED TO SEND THE CONTROL CODES TO THE
4050         INC HL             ;PRINTER
4051         DEC C
4052         JR NZ,P01
4054         RET
4055 VRAMIN: PUSH AF            ;SUBROUTINE SETS VRAM TO READ MODE
4056         LD A,L
4057         OUT (2),A
4059         LD A,H
405A         AND £3F
405C         OUT (2),A
405E         POP AF
405F         RET
4060 GRAPHIC:LD HL,LINEFGM      ;SET FEED TO 8/722 TO JOIN LINES UP
4063         LD C,3
4065         CALL P01
4068         LD HL,0000 ;INITIALISE COUNTER AND START OF VS 4 VRAM
406B         LD (COUNTER),HL
406E         CALL VRAMIN
4071 MLOOP:  LD HL,WIDTHGM
4074         CALL POUT
4077 READDAT:LD HL,PBUFF        ;TEMP 8 DATA LOCATION
407A         LD B,8
407C RD1:    IN A,(1)
407E         LD (HL),A  ;SAVE FISRT HORIZONTAL BYTE IN 8X8 MATRIX
407F         INC HL
4080         DJNZ RD1
```

```
4082 EPSON:   LD C,8        ;NUMBER OF DATA BYTES
4084 EP1:     LD B,0        ;INITIALISE DATA REGISTER
4086          LD D,8        ;NUMBER OF BITS IN 1 BYTE
4088 EP2:     OR £00        ;CLEAR FLAGS
408A          DEC HL
408B          RLC (HL)      ;ROTATE DATA BY 1 BIT
408D          JR NC,ZERO    ;CHECK IF CARRY BIT SET
408F          SET 7,B       ;CARRY BIT SET SO SET THE 7TH BIT IN B
4091 ZERO:    OR £00        ;CLEAR FLAGS
4093          DEC D         ;DECREASE BIT COUNT
4094          JR Z,SEND     ;IF =0 THEN 1st VERTICAL BYTE READY TO BE OUTPUTTED
4096          RR B          ;ROTATE ALL B BITS ONE PLACE RIGHT TO GET RIGHT
4098          JP EP2        ;VERTICAL ORDER THEN GET NEXT BIT
409B SEND:    CALL £0CE3
409E          LD HL,COUNTER       ;POINT HL 1 ABOVE PBUFF TO CANCEL OUT 1st DEC HL
40A1          DEC C         ;DECREASE BYTE COUNT
40A2          JR NZ,EP1     ;IF NOT=0 THEN CONTINUE ROTATING
40A4          LD DE,1
40A7          LD HL,(COUNTER)
40AA          ADC HL,DE     ;UPDATE COUNTER
40AC          LD (COUNTER),HL
40AF          LD A,L
40B0          AND £1F       ;TEST IF END OF LINE
40B2          JR NZ,READDAT
40B4          LD B,10       ;SEND LINE FEED CONTROL AT THE END OF EACH LINE
40B6          CALL £0CE3
40B9          LD A,3
40BB          CP H          ;TEST IF END OF GRAPHICS SCREEN
40BC          JR NZ,MLOOP
40BE END:     LD B,27       ;RESET PRINTER
40C0          CALL £0CE3
40C3          LD B,64
40C5          CALL £0CE3
40C8          RET
40C9          RET

Symbols:
JPANEL   4007     PANEL    4012
TEXT     4015     VRAMIN   4055
LINEFTP  4033     P01      404C
TPOUT    4027     WIDTHTP  4036
LINEFGM  4039     WIDTHGM  403C
PBUFF    4040     COUNTER  4048
POUT     404A     GRAPHIC  4060
MLOOP    4071     READDAT  4077
RD1      407C     EPSON    4082
EP1      4084     EP2      4088
ZERO     4091     SEND     409B
END      40BE
```



```
1 REM **PANEL DUMP:-ENTER PANEL THEN PRESS P**
2 REM**TEXT DUMP:-USE COMMAND RAND USR(16405) IN PROGRAM TO CALL TEXT DUMP**
3 REM**VS 4 DUMP:-USE COMMAND RAND USR(16480) IN PROGRAM TO CALL VS 4 DUMP**
```

Subscriptions

IF YOUR MEMBERSHIP NUMBER IS BETWEEN 1339 & 1529 (ALL LETTERS) YOU ARE NOW DUE TO RENEW YOUR SUBSCRIPTIONS. TO MAKE SURE OF YOUR NEXT ISSUE PLEASE SEND YOUR SUBSCRIPTION TO REACH US NO LATER THAN THE 19TH April 86

SOFTWARE

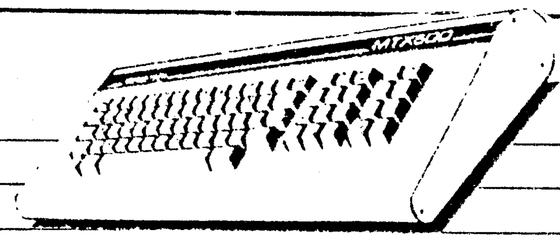
00106	26X26 SPREAD SHEET	UTIL	SYNT	7.95	I	ANY
00057	3D TACHYON FIGHTER	ARC	CONT	6.95	I	ANY
00135	9 ELECTRIC. PROGS	EDUC	SSFT	13.95	U	ANY
00062	ADVENTURE QUEST	ADV	LVL9	8.75	I	ANY
00033	AGROVATOR	ARC	SYNT	5.95	I	512
00125	AIRCRAFT MAGNETISM	FLGT	AVTN	?	I	ANY
00120	AIRCRAFT PAYLOADS	FLGT	AVTN	?	I	ANY
00122	AIRLAW 2	FLGT	AVTN	?	I	ANY
00123	AIRSPEED INDICATOR	FLGT	AVTN	?	I	ANY
00071	ALICE IN WONDER.	ADV	CONT	6.02	I	ANY
00121	ALTIMETER	FLGT	AVTN	?	I	ANY
00008	ASTROMILON	ARC	CONT	6.02	I	ANY
00047	ASTROPAC	ARC	CONT	6.02	I	ANY
00058	BACKGAMMON	BRD	CONT	7.95	I	ANY
00041	BASIC BUSINESS	BS	CONT	5.95	I	ANY
00043	BLOBBO	ARC	CONT	6.02	I	ANY
00073	BOUNCING BILL	ARC	SYNT	4.95	I	ANY
00074	BRIDGE	CARD	CONT	6.95	I	512
00130	BUSINESS GAME	BRD	SSFT	15.95	U	ANY
00077	CANVAS	UTIL	CONT	7.95	I	ANY
00085	CAVES OF ORB	ADV	SYNT	5.95	I	512
00137	CESIL INTERPRETER	LANG	SSFT	5.95	U	ANY
00094	CHAMBEROIDS	ARC	MEGA	5.95	I	ANY
00059	CHESS	BRD	CONT	8.75	I	ANY
00053	COBRA	ARC	CONT	6.02	I	ANY
00025	COLOSSAL ADVENTURE	ADV	LVL9	8.75	I	ANY
00098	COMBAT	ARC	PANS	2.95	I	512
00028	COMPOSER	UTIL	XAV	13.00	I	ANY
00046	CONT RAIDERS	ARC	CONT	6.02	I	ANY
00099	CRIBBAGE	CARD	SCRP	2.95	E	ANY
00110	CRYSTAL	ARC	MEGA	5.95	I	ANY
00050	DEN.GOES BANANAS	ARC	SCRP	2.95	I	ANY
00011	DENNIS & CHICKEN	ARC	SCRP	2.95	I	ANY
00103	DENNIS AND CIRCUS	ARC	SCRP	2.95	I	ANY
00068	DOODLEBUG	ARC	SYNT	4.95	I	ANY
00108	DOWNSTREAM DANGER	ARC	MEGA	5.95	I	ANY
00096	DR. FRANKIE	ARC	SYNT	5.95	I	512
00056	DRAUGHTS	BRD	CONT	6.95	I	ANY
00111	DRIVE THE CEE 5	ARC	MEGA	5.95	I	ANY
00063	DUNGEON ADVENTURE	ADV	LVL9	8.75	I	ANY
00067	EDASM	UTIL	SYNT	7.95	I	512
00066	EMERALD ISLE	ADV	LVL9	5.95	I	ANY
00038	ESCAPE FROM ZARKOS	ARC	MEGA	5.95	I	ANY
00081	EXTENDED BASIC	6.95	SENT	6.95	I	ANY
00082	FATHOMS DEEP	ARC	MEGA	5.95	I	ANY
00090	FIG FORTH	LANG	SYNT	15.75	I	512
00055	FIREHOUSE FREDDIE	ARC	CONT	6.02	I	ANY
00021	FIRST LETTERS 1	EDUC	CONT	8.75	I	ANY
00092	FKEY DEFINER	UTIL	MEMB	6.95	I	ANY
00037	FLUMMOX	ARC	SYNT	5.95	I	512
00132	FRACT, PERCENTAGES	EDUC	SSFT	5.95	U	ANY
00052	GAUNTLET	ARC	CONT	6.02	U	ANY
00102	GHOSTLY CASTLE	ADV	PANS	2.95	I	ANY
00031	GOLDMINE	ARC	CONT	6.02	I	ANY
00069	GRAPHICS	UTIL	CONT	5.95	I	ANY
00087	H & L DUMP	UTIL	MEM	4.95	I	ANY
00072	HAWKWARS	ARC	SYNT	4.95	I	ANY
00065	HELI-MATHS	EDUC	CONT	5.95	I	ANY
00034	HUNCHY	ARC	SYNT	4.95	I	ANY
00083	ICEBERG	ARC	SYNT	4.95	I	ANY
00105	JET SET WILLY	ARC	SPRJ	6.95	I	ANY
00015	JOHNNY REB	WAR	LOTH	6.02	I	ANY
00097	JUMPING JACK FLASH	ARC	SYNT	5.95	I	512
00115	KARATE KING	ARC	MEGA	5.95	I	ANY
00016	KEY TO TIME	ADV	LUMP	6.02	I	ANY
00042	KILOPEDE	ARC	CONT	6.02	I	ANY
00019	KNUCKLES	ARC	CONT	7.95	I	ANY
00078	LES FLICS	ARC	PSS	6.95	E	ANY
00032	LITTLE DEVILS	ARC	SYNT	4.95	I	ANY
00024	LORDS OF TIME	ADV	LVL9	8.75	I	ANY
00035	M COMMAND & ARCAD.	ARC	SYNT	4.95	I	ANY
00070	MAN FROM GRANNY	ADV	SYNT	4.95	I	512
00104	MANIC MINER	ARC	SPRJ	6.95	I	ANY
00119	MAPS AND CHARTS	FLGT	AVTN	?	I	ANY
00126	MAPS AND CHARTS 1	FLGT	AVTN	?	I	ANY
00022	MATHS 1	EDUC	CONT	8.75	I	ANY
00013	MAXIMA	ARC	CONT	6.02	E	ANY
00086	MEMOCHEQUE	UTIL	SYNT	6.95	I	ANY
00075	MEMOSKETCH	UTIL	SYNT	7.95	I	ANY
00089	MINER D1CK	ARC	XAV	6.95	I	ANY
00044	MISSION ALPHATRON	ARC	CONT	6.02	I	ANY
00030	MISSION OMEGA	ARC	SYNT	4.95	I	ANY
00054	MURDER AT MANOR	ADV	LUMP	6.02	I	ANY
00010	MUSIC PAD	UTIL	CONT	6.02	I	ANY
00003	NEMO	ARC	CONT	6.00	I	ANY
00131	NETWORK LOADER	UTIL	SSFT	8.95	U	ANY
00112	OBLITERATION ZONE	ARC	MEGA	5.95	I	ANY
00045	OBLOIDS	ARC	CONT	6.02	I	ANY
00129	PAINTBOX	UTIL	SYNT	5.95	I	ANY
00001	PAYROLL	UTIL	CONT	21.25	I	512
00005	PHAID	ARC	CONT	6.02	I	ANY
00061	PHYSICS 1	EDUC	CONT	8.75	I	ANY
00124	PILOT NAVIGATION	FLGT	AVTN	?	I	ANY
00012	PONT & BLACKJACK	CARD	CONT	6.02	I	ANY
00009	POT HOLE PETE	ARC	CONT	6.02	I	ANY
00040	PURCHASE LEDGER	BN	CONT	12.75	I	512
00048	QOGO	ARC	CONT	6.02	I	ANY
00076	QOGO 2	ARC	MEGA	5.95	I	ANY
00095	QUANTUM	ARC	SYNT	5.95	I	ANY
00109	QUAZZIA	ARC	MEGA	5.95	I	ANY
00107	RED MOON	ADV	LVL9	?	U	ANY
00127	RELATIVE VELOCITY	FLGT	AVTN	?	I	ANY
00064	RETURN TO EDEN	ADV	LVL9	8.75	I	ANY
00020	REVERSI	BRD	CONT	7.95	I	ANY
00114	ROLLA BEARING	ARC	MEGA	5.95	I	512
00100	RUTHLESS BASTARD	ARC	LSFT	2.50	I	512
00002	SALES LEDGER	UTIL	SYNT	15.75	I	512
00029	SALTY SAM	ARC	SYNT	4.95	I	ANY
00113	SEPULCRI SCELERATI	ARC	MEGA	5.95	I	512
00101	SLOOPY'S CHRISTMAS	ARC	PANS	2.95	I	ANY
00116	SMG	ARC	MEGA	5.95	I	ANY
00049	SNAPPO	ARC	CONT	6.02	I	ANY
00023	SNOWBALL	ADV	LVL9	8.75	I	ANY
00036	SON OF PETE	ARC	MEGA	5.95	I	ANY
00136	SOUND & RESISTORS	EDUC	SSFT	5.95	U	ANY
00026	SPELLI-COPTER	EDV	CONT	5.95	I	ANY
00080	SPOOLER	UTIL	MEM	4.95	I	ANY
00017	STAR COMMAND	ARC	CONT	6.95	I	ANY
00014	SUPA CODER	UTIL	SYNT	7.95	I	ANY
00084	SUPER BIKE	ARC	SYNT	4.95	I	ANY
00004	SUPER MINEFIELD	ARC	CONT	6.02	I	ANY
00093	SURFACE SCANNER	ARC	MEGA	5.95	I	ANY
00133	SYMMETRY & GLASS	EDUC	SSFT	5.95	U	ANY
00039	TAPE TO DISC	UTIL	MEM	6.95	I	ANY
00007	TAPEWORM	ARC	CONT	6.02	I	ANY
00088	TARGET ZONE	ARC	SYNT	6.95	I	ANY
00118	THE DESIGNER	UTIL	HALT	8.95	I	ANY
00128	THE WALL	ARC	SYNT	4.95	I	512
00051	THE ZOO GAME	ADV	CONT	6.02	I	512
00134	TITRATION,CHROMATO	EDUC	SSFT	5.95	U	ANY
00006	TOADO	ARC	CONT	6.02	I	ANY
00018	TURBO	ARC	CONT	6.95	I	ANY
00117	USER BASIC	UTIL	SYNT	8.95	I	ANY
00079	USER EXTEND	UTIL	MEM	7.95	I	ANY
00027	UTILITIES 1	UTIL	CONT	4.95	I	ANY
00091	VERNON & VAMPIRES	ARC	SYNT	5.95	I	ANY
00138	WOOD SIMULATION	EDUC	SSFT	5.95	U	ANY
00060	WORD & PICTURE	EDUC	CONT	8.75	I	ANY

PLEASE ONLY ORDER THOSE MARKED " I "

E=EXPECTED SOON I = IN STOCK U = UNAVAILABLE AT PRESENT

KEY:- STOCK NUMBER TITLE TYPE HOUSE MEMBER PRICE STOCK

HARDWARE



PRICES EFFECTIVE FROM 1ST DECEMBER 1985

MTX 512 ..... #119.00 MEMBERSHIP EXTRA
MTX 500 ..... # 69.95 MEMBERSHIP EXTRA

32K MEMORY EXPANSION .....#36.74
64K MEMORY EXPANSION ..... #45.43
128K MEMORY EXPANSION ..... #69.52
**** PLEASE STATE FOR WHICH MODEL ******

NEWWORD 32K ON ROM .....#36.74
PASCAL 16K ON ROM .....#36.74
SPECULATOR .....#36.95

DMX80 PRINTER .....#179.95
PRINTER CABLE .....# 8.95

SDX 500K DRIVE + INTERFACE .....#222.50
SDX 1MEG DRIVE + INTERFACE .....#265.83

2ND 250K SDX DRIVE .....# 89.00
2ND 500K SDX DRIVE .....#169.00
2ND 1MEG SDX DRIVE .....#203.00

FDX 2ND DRIVE 500K .....#141.00
FDX 2ND 1MEG DRIVE .....#163.00

DUST COVER .....#3.50
DMX80 PRINTER RIBBON .....#8.98

FLOPPY DISCS (BOX 10) GUARANTEED ..... #18.95
PACE NIGHTINGALE MODEM .....#119.00 + #5.00 P&P
250K DISC DRIVE + INTERFACE .....#199.00 + #5.00 P&P

DISC CASES HOLD 10 DISC ..... # 2.55
FLOPPICLENE DISC CLEANING KIT .....#17.20
ANTISTATIC SCREEN WIPES ...(10)....# 1.50
DISC CABINETS (LOCKABLE) 110 DISCS #36.95

CRIB CARD ..... # 2.16
ROM LISTINGS ..... #45.00
SDX CONTROLLER LISTINGS .#20.00
ROM CALLS INFO SHEET ... 50p
RST 10 CALLS INFO SHEET 50p
INTERRUPTS INFO SHEET 80p
DDT INFO BOOK .....#2.00
VDP TECHNICAL MANUAL ....#5.95
MTX SERVICE MANUAL .....#6.95

80 COL CARD + CPM +NW/SC .....#180.95
80 COL UPGRADE KIT (RS232) .....# 27.00

PACKAGE ONE
SDX 500K DRIVE + INTERFACE
+80 COL BOARD +CPM + NW + SC .....#355.00

PACKAGE TWO
AS ABOVE BUT WITH 1MEG DRIVE .....#395.00

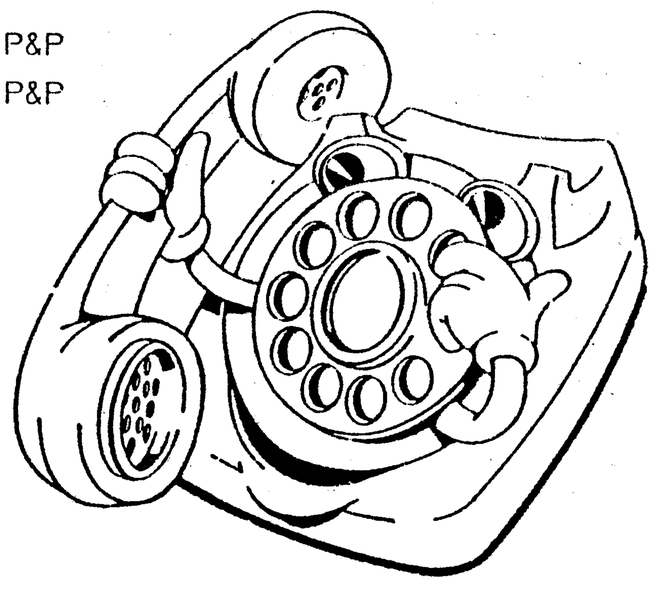
PACKAGE THREE
SDX 500K DRIVE : INTERFACE:
COMPUTER MTX512: 80 COL PCB:CPM .....#449.00

PACKAGE FOUR
AS ABOVE BUT WITH 1MEG DRIVE .....#489.00

FDX SINGLE 500K CPM SYSTEM .....#539.00
FDX SINGLE 1MEG CPM SYSTEM .....#675.00

FDX TWIN 500K CPM SYSTEM .....#569.00
FDX TWIN 1MEG CPM SYSTEM .....#740.00
(REquires RS232 comms board)

SILICON DISCS
250K ..... #145.90



THE ABOVE PRICES ONLY APPLY TO U.K SALES & BFPO SALES
BFPO SHOULD ADD AN EXTRA £30.00 TO COVER ADMINISTRATION BY MEMOTECH

UPGRADE PACKAGE 1 ..... £198.00
UPGRADE PACKAGE 2 ..... £223.39
The above require factory fitting so add an extra £25 to cover cost of this service.

DON'T FORGET IF YOU HAVE A DISC DRIVE YOU SHOULD OWN A HIGH QUALITY HEAD CLEANER see FLOPPICLENE
.... over half the problems handled by us are due to dirty disc heads.

JARO SPEED SPLITTER WITH SOFTWARE £20.70

Late Xtra!

MEMOTECH COMPUTERS LTD
TO: 635480

ATTN KEITH HOOK

FROM GEOFF BOYD, MEMOTECH COMPUTERS LTD, OXFORD

TO KEITH HOOK, SYNTAXSOFT LTD, NELSON

MEMOTECH COMPUTERS LTD WILL BE CONCLUDING NEGOTIATIONS FOR THE PURCHASE OF THE BUSINESS OF MEMOTECH LTD (IN RECEIVERSHIP) SHORTLY. WE ARE READY TO COMMENCE PRODUCTION AND WOULD APPRECIATE AN INDICATION OF THE OUTSTANDING ORDERS FROM GENPAT MEMBERS YOU WOULD LIKE US TO FULFILL.

BEST REGARDS

GEOFF BOYD

MEMOTECH COMPUTERS LTD

Coming Soon On The MTX

This file was downloaded from

www.primrosebank.net

Abridged Terms & Conditions (Downloads)

Disclaimer

www.primrosebank.net, (***the website***) is provided by Dave Stevenson as a service to the public, is provided "as is" and carries no warranties, expressed or implied, of any kind.

Dave Stevenson is not responsible for, and expressly disclaims all liability for, damages of any kind arising out of use, reference to, or reliance on any information contained within the website or made available for download. Whilst the information contained within the website site is periodically updated, no guarantee is given that the information provided on the website is correct, complete, and up-to-date.

A number of articles on the website contain technical data and practical guidance which may be of use in testing and maintaining various items of vintage computer and electronics hardware. Such articles are not intended to cover all aspects of the tasks involved and may omit essential information, including necessary safety precautions. Performance of the tasks described may risk damage to equipment and/or people. The reader is responsible for ensuring that he/she is capable of performing the tasks described and well as assessing the inherent risks involved and taking appropriate measures to mitigate such risks.

Dave Stevenson expressly disclaims all liability for, damages to equipment or injury of any kind arising out of use of such technical data and guidance.

Unless otherwise noted, all data on the website is deemed to be ***Copyright (c) Dave Stevenson, 2009-2013***

You are hereby granted permission to download data and software from the website ***for your own personal use. Redistribution of any content from the website without written authorisation from Dave Stevenson is expressly forbidden. You are also expressly forbidden from offering for sale any material obtained from the website.***

As far as possible, information included on the website from other sources has been credited to the respective author and/or publisher. The majority of content on the website is derived from material first published in the 1980s. ***This material is likely still under copyright of the original author and/or publishers.*** The authors and/or publishers may not have given express permission to copy, transmit or make this information available for download, but I believe that they would have no objection to this archive information being placed into the public domain.

However, should the author and/or publisher of the original material find any content on the website for which they wish to assert their rights, they should notify Dave Stevenson (by e-mail to: webmaster@primrosebank.net) who would be pleased to enter into a dialogue to agree a satisfactory resolution of their concerns.

If you obtained this file as part of a paid-for package, you have been scammed! I suggest that you request a refund from the seller, please also advise Dave Stevenson at the e-mail address above.