CORTEX USERS GROUP

CORTEX USER GROUP NEWSLETTER (AUG 1987)

Issue Number 12

CONTENTS

ERRORS

RAMDISC ISSUE 9 PAGE 9.5

Faulty disk access can occur when this programme is installed sometimes resulting in files being erazed. The fault as caused because the routine corrupts R12 which cdos uses for something else. To correct the problem re-write the routine changing all referances to R12 to R2 instead.

SHORT TIP DMA MOD ISSUE 7 PAGE 7.12

Pin 4 of IC24 should be isolated and taken to +5V not pin 5

NEWS

Ameeting of Cortex and TI994A users will take place at Sneyd school Bloxwich on September the 5th. More details enclosed seperately.

Chris Young has sent in a disk full of Cdos utilities including the following:-

PDIR	Print directory utility programme
SORTDIR	Sorts directory entries into alphabetical order
DV	Verifies disk sectors and flags any errors
DISKNAME	Puts a name on the disk for use by PDIR etc.

These programmes will be printed in this and future issues but if users would prefer to obtain the programmes on disk free of charge just send in a Formatted disk plus £2.00 to cover copying and postage.

Chris has also written a small machine code routine called SMALL which allows printing of 64 columns to the screen in graph mode. Details in this issue.

Also from Chris Young is a CDOS compatible editor assembler package written in machine code. The full screen editor can handle very large source files and is simple to use. A help screen is provided in case you forget some of the commands. The very fast assembler supports all the 9995 opcodes and also the Cortex Mid opcodes MSG, WRIT etc. Two special opcodes have been added TEXT and GRAPH for changing the screen mode fromwithin a machine code programme. Routines for printing source and label listing are also included.

The complete package is available from the group for £15.00
Please specify exact disk format required.

```
            PRINT DIRECTORY PROGRAMME BY C.J.YOUNG
20  REM *************
50  REM * PRINT DIR *
80  REM *************
110  REM *** Display title ***
130  TEXT
140  ? " CDOS Print directory 1.0 1987"
150  ? " Input Drive ? ";
160  IK=KEY[0]
170  IF IK=0: GOTO 160
180  IF IK<48: GOTO 160
190  IF IK>51: GOTO 160
200  DRV=IK-48
210  ? DRV
220  REM
230  REM *** Set up Arrays ***
240  REM
250  DIM B[100]  !Buffer
260  DIM $NM[9]  !Name of disc
270  DIM MC[20]  !Machine code
280  REM
290  REM *** Set up variables ***
300  REM
310  AMC=ADR[MC[0]]  !Addr of MC
320  AB=ADR[B[0]]  !  Addr of buffer
330  ERR=0  !Error
340  D2=DRV*2
350  USD=0
360  AR=0A5A5H
370  NA=05A5AH
380  SD=0FFFFH
390  NF=0
400  REM
410  REM * Machine code *
420  REM
430  DATA 0420H,06180H,0D000H,01601H
440  DATA 0380H,0460H,06550H,04F2H
450  DATA 04D2H,0C0F1H,0704H,0A13H
460  DATA 01701H,0592H,0924H,016FBH
470  DATA 0600H,016F7H,0380H,0H
480  FOR I=AMC TO AMC+38 STEP 2
490   READ X
500   MWD[I]=X
510  NEXT I
520  REM
530  REM * Get disk name *
540  REM
550  $NM[0]="SYSTEM DISK"
560  CALL AMC,0,DRV*256,0,AB,128
570  IF MWD[AB]: GOTO 610
580  FOR X=1 TO 20
590   $NM[0;X]=%MEM[AB+X+1]
600  NEXT X
610  UNIT 2: UNIT -1
620  ? " <1B>-1CDOS Print directory Utility  1.0   ";
```

```
630  ? "Time: ";: TIME
640  ? "<1B>-0"
650  ? " Disk Name: "$NM[0]
660  UNIT -2: UNIT 1
670  REM
680  REM * Get disk parameters *
690  REM
700  DP=MWD[06382H+D2]
710  SPT=MWD[DP]  ! Sectors/track
720  NS=MWD[DP+2]  !no of sectors
730  DS=MWD[DP+4]  !Directory start
740  MF=MWD[DP+6]  !Max Files
750  BPS=MWD[06362H+D2]  !Bytes/sector
760  BMA=SPT*BPS  ! Bit map Addr
770  DDA=DS*BPS  !  Disk Dir Addr
780  BML=DDA-BMA  ! Bit Map Length
790  REM
800  REM * Calc used *
810  REM
820  CALL AMC,0,DRV*256,BMA,AB,BML
830  CALL AMC+14,BML,AB,ADR[USD]
840  UNIT 2: UNIT -1
850  ? " Total number of sectors = "£'9999'NS
860  ? " Number of Used  Sectors = "£'9999'USD
870  ? " Number of Free  Sectors = "£'9999'NS-USD
880  ? " Number of Bytes per Sector = "BPS
890  ? " Maximum number of files ="MF
900  ? "============================================================"
910  ? " Name      Type             Length Load    Sector   Length"
920  ? "------------------------------------------------------------"
930  FOR I=0 TO MF-1
940   CALL AMC,0,DRV*256,DDA+I*64,AB,64
950   IF MWD[AB]=0: GOTO 1420
960   REM
970   REM * Print name *
980   REM
990   FOR X=1 TO 8
1000    Q=MEM[AB+X+1]
1010    IF Q=0: Q=32
1020    $NM[0;X]=%Q%0
1030   NEXT X
1040   ? " ";$NM[0];
1050   REM
1060   REM * Check type *
1070   REM
1080   IF MWD[AB]=AR: GOTO 1210
1090   IF MWD[AB]=NA: GOTO 1250
1100   IF MWD[AB]=SD: GOTO 1150
1110   ? " Random Data ";
1120   ? £'999'MWD[AB];
1130   DL=MWD[AB]
1140   GOTO 1170
1150   ? " Sequential data";
1160   DL=6
1170   X=MWD[AB+18]
```

```
1180   ? " R=";
1190   ? £'9999'X/DL;
1200   GOTO 1340
1210   ? " Auto-Run";
1220   IF MWD[AB+10]: ? " Basic ";
1230     ELSE ? " Code  ";
1240   GOTO 1290
1250   IF MWD[AB+10]: GOTO 1320
1260   ? " Code ";
1270   IF MWD[AB+14]: ? "program    ";
1280     ELSE ? "           ";
1290   IF MWD[AB+10]: GOTO 1330
1300   ? "  ";£,MWD[AB+16];"H ";£,MWD[AB+12];"H  ";
1310   GOTO 1350
1320   ? " Basic program  ";
1330   ? " ";£'99999'MWD[AB+16];" ";
1340   ? "         ";
1350   FOR ZZ=32 TO 60 STEP 4
1360    ? " ";£,MWD[AB+ZZ];"H    ";
1370    ? £,MWD[AB+ZZ+2];"H"
1380    IF MWD[AB+ZZ+4]=0: ZZ=64: GOTO 1400
1390    ? "                                          ";
1400   NEXT ZZ
1410   NF=NF+1
1420  NEXT I
1430  ?
1440  ? "-------------------------------------------------------"
1450  UNIT 1
1460  ? " End Of Directory"
1470  ? " Number of files Used ="NF
1480  UNIT -1
1490  ? "<C>";
1500  UNIT 1: UNIT -2
1510  END
```

```
EXAMPLE  RUN
 CDOS Print directory 1.0 1987
 Input Drive ?  1
 -1CDOS Print directory Utility  1.0   Time:  00:12:55-0
 Disk Name: SYSTEM DISK
 Total number of sectors =  640
 Number of Used  Sectors =  285
 Number of Free  Sectors =  355
 Number of Bytes per Sector =  128
 Maximum number of files = 30
=============================================================
 Name      Type              Length Load     Sector   Length
-------------------------------------------------------------
 SWING     Basic program     4036            0020H    0020H
 LETTER    Auto-Run Basic    3750            0040H    0018H
                                             017FH    0001H
                                             0226H    0005H
 SCROL     Code              0400H E600H     0184H    0008H
 BASIC     Basic program     1160            01ECH    0009H
                                             0180H    0001H
```

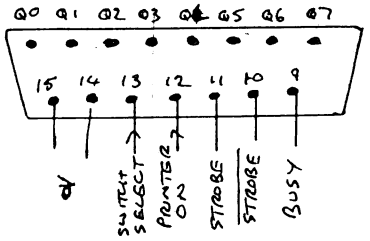
W.D. EAVES CAITHNESS

Ther are two unused pins on the Centronics socket that can be used to test whether a printer is connected to the interface and if it is on line. Most printers do provide the necessary signals and users will have to refer to their printer manual. I use a Centronics 739 which provides a +5V printer on signal and a Select signal set by the On Line/Local switch.

On the underside of the centronics board connect wires from SK1 pins 13&12 to IC4 pins 1&2. Connect a resistor of about 3.3k between pin 2 of IC4 and ground. This is to ensure a zero volt siganl when no printer is connected.

Rewire the printer lead so that the On signal goes to pin 13 on the connecting plug and the select (if available) goes to pin 12.

In a program checks can be made by setting BASE 800H and interrogating CRB[10] to check the printer is available and switched on. CRB[11] if applicaable can check the state of the on line switch.



See Cortex II P.20

Some users may have noticed that when using my KEYS program in Newsletter 9 that the cassette command CLOAD will not work(or LOAD if not using a disk based system). One of the characters is used in the header checksum and if the key routine is enabled then a TAPE READ ERROR will occur as soon as the program is found. This problem can be avoided by disabling the KEYS routine.

To disable the routine, memory locations 48EH and 490H need to be set to 4DDH and 0D778H respectively. To re-enable set the locations to 0460H and A0 (the start address of the keys code). I have added a section to the program to create two new commands KON (Keys ON) and KOF (Keys OFf) to do these operations simply.

The listing is the same as that in Newsletter 9 with the following lines. Line 51 and lines 140-260. Lines 200-230 with the data at 140-150 set up a small m/c routine to set/unset the keys routine. Line 240 with the data at 160 to 170 put the command names into the lookup table to run the new coding. The code can be relocated by altering the variable SA which holds the start address of the added code.

One point omitted from the previous writeup for KEYS is that 'a' and 'b' (shown in the new listing) are dummy characters to allow quotes and carriage returns to be included in the labels.

W.D. EAVES

```
1  REM ******** KEYS WITH ENABLE/DISABLE COMMANDS ********
10  DIM $C[2]
15  TEXT : COLOUR 1,13: ? @"C14R";"KEY LABELS": ? @"13R";"**************"
20  ? : ? "   Key          Label": ?
25  A0=05FB0H: KD=05F00H: RESTOR 105
30  FOR F=0 TO 47: READ A: MWD[A0+F*2]=A: NEXT F
35  MWD[048EH]=0460H: MWD[0490H]=A0
40  FOR N=176 TO 185: X=N-176: GOSUB 65: NEXT N
45  N=173: X=N-163: GOSUB 65
50  FOR N=219 TO 223 STEP 2: X=(N-197)/2: GOSUB 65: NEXT N
51  GOSUB 200
53  ? : ? "To disable 'CALL";£SA+018H;"' or type 'KOF'"
55  ? "Re-enable 'CALL";£SA+8;"' or type 'KON'": ? : WAIT 100
56  ? "Loading CAT Command": LOAD 0,"CAT"
60  ? "Loading FIND Command": LOAD 0,"FIND"
65  F=KD+12*(X): MEM[F]=N: READ $C[0]: GOSUB 90
70  FOR G=1 TO 10: MEM[F+G]=ASC[$C[0;G]]: NEXT G: MEM[F+G]=0
75  $A=%13: P=POS[$A,$C[0]]: IF P THEN $C[0;P]="[Return]"
80  $A=%(N-128): ; "[GRAPH]-";$A;"      ";$C[0]
85  RETURN
90  P=POS["a",$C[0]]: IF P THEN $C[0;P]=%13%0
95  P=POS["b",$C[0]]: IF P THEN $C[0;P]=%34
100  RETURN
105  DATA 0420H,A0+0CH,04DDH,0D778H,0460H,0492H,A0+010H,A0+030H
110  DATA KD,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
115  DATA 0C22DH,010H,0C040H,0202H,11,09631H,01304H,0A042H
120  DATA 0D451H,016FBH,0380H,0DE31H,0D451H,01601H,0380H,0DE31H
125  DATA 05A0H,0EDA8H,010F9H,0DE03H,05A0H,0EDA8H,0380H,0
130  DATA "RUNa","LIST","LOAD 0,b","LDIRba","DATA ","GOSUB","RETURN"
135  DATA "SAVE 0,b","b,REP,EXa","CONTa","COLOUR","GOTO","GRAPH","TEXT"
140  DATA 0460H,A0,04DDH,0D778H
145  DATA MOV,SA,A1,MOV,SA+2,A1+2,0460H,021CH
150  DATA MOV,SA+4,A1,MOV,SA+6,A1+2,0460H,021CH
160  DATA 03A9AH,073D6H,04038H,SA+8
170  DATA 03A9CH,033D6H,0403AH,SA+018H
199  REM ******** CODE TO ENABLE/DISABLE KEYS ********
200  MOV=0C820H: SA=06FD0H: A1=048EH
210  FOR I=SA TO SA+38 STEP 2
220   READ A: MWD[I]=A
230  NEXT I
239  REM ******* CODE TO PUT AS COMMANDS KON & KOF *******
240  FOR I=1 TO 4: READ A,B: MWD[A]=B: NEXT I
260  RETURN
```

£ = #

MDEX UTILITIES DISK FROM ANTHONY ROWELL

Anthony has put together a few MDEX utilities for user group members The disc is in 40 track single sided double density MDEX format and contains the following :

1.	MOD256K.TXT	A description of how to upgrade the CORTEX memory to 256K Bytes onboard.
2.	WPRINT	A utility to print out a file with multiple columns per page (like a newspaper) on an MDEX system. Very useful for disassembly etc.
3.	SWAP	A utility to exchange drives 1/ and 2/ under MDEX (and back again if called again). Useful for systems having mixed drive types. NB. the FORMAT program access the disc controller dirctly and so bypasses any SWAP.
4.	DISASM	A utility to disassemble either program files or relocatable files (better). Various options including initial symbol table input from a file for iterative disassembly. Still under development so a few bugs but still quite useable (feedback welcome!).
5.	MENU.OBJ	A library routine for menu input, used by WPRINT etc. for linking to user programs.
6.	README	This document.
7.	SYMBOLS	An example symbol table input file for the disassembler.
8.	MENU.DOC	Brief documentation for using MENU.OBJ
9.	ADVENTURE	This is a copy of Colin Hinsons version of Colosol Cave.

These items are free to CORTEX users for non-comercialuse only.

Anthony is currently writing new device drivers etc. for MDEX as a result he now has a MDEX system that allows upto 4 disc drives and a RAMDISK. These are currently working but need some final tidying up. To use them you need the MDEX system generation kit but the result is worth it (eg. using 5" drives the assembly of the 1200 lines of floppy driver source takes 2min. 13sec. but in ramdisk takes 15sec.) and the programable function keys make life easier too.

If anyone wants a copy of the disk please send a formatted and prepped disk in plus £2 to cover copying and postage.

We hope Anthony will send in more details of his new device drivers when they are finished.

256K Memory upgrade for the Cortex. Anthony Rowell

The onboard memory of the cortex can increased to 256K bytes at low cost and with minimal hardware modification. I operate a cortex modified in this way and have no resulting problems.

The sequence of events is as follows :

1. Link pin 1 of all the 4164 Ram's together (IC86 to IC93).
2. Isolate pin 13 of IC52 (74LS02).
3. Link Pins 1 and 2 of IC55 (74LS20) together.
*4. Remove links LK1 to LK4 located next to IC66 (74LS612).
*5. Cut the four "dots" alongside IC80 (74LS244).
*6. Install IC66 (74LS612).
7. Mount a 74LS157 on the PCB by a suitable means and wire it as follows:-

   Link Pin 1 to Pin 4 of any DRAM (RAS-)
   Link Pins 2,3,5,6,8,15,14,13 to GND
   Link Pin 16 to +5v
   Link Pin 9 to Pin 1 of any DRAM
   Link Pin 10 to Pin 23 of IC66 (74LS612 MO7 XA3)
   Link Pin 11 to Pin 22 of IC66 (74LS612 MO6 XA2)

8. Connect a 1K Ohm resistor between pins 9 and 16 of the new 74LS157
9. Replace all eight 4164 DRAMS with 256K x 1 DRAMS (I used NEC 41256-15)

The IC numbers given are as marked on the PCB and not as given in the original ETI articles. The upgrade works by using the existing 4500A DRAM controller to control larger DRAMS than it is intended for and the extra DRAM address line (A0) being provided via a multiplexor from XA3 & XA2. The onboard memory decode is modified such that it extends from 00000 to 3FFFF.

When using the memory mapper the 4K of ram at 0Fxxx can be used by loading any mapping register (oher than 15) with the value 0F.

The cost of the upgrade is 25..30 Pounds and the resulting memory operates faster than an E-BUS based memory expansion.

For anyone who would like to increase this to 512K bytes I suggest that the use XA1 to feed REN1, isolate XA1 from the decode logic and use a second bank of DRAMS in parallel with the main set but with a seperate RAS- line driven frim RAS1-.
As a result the select line of the 74LS157 should be driven by (RAS0- anded with RAS1-). I have not tried the 512K version but it should be ok.

* - Steps not required if memory mapper already fitted.

SMALL CHARACTERS IN GRAPH MODE "SMALL.S" C.J.YOUNG

Listing Of Assembler Source File "SMALL.S"

```
01  ; SMALL CHARACTERS IN GRAPH MODE
02  ;
03  ; TO USE CALL ORG,Y,X,CHAR
04  ;
05  ; Y= ROW 0 TO 24  X=COLUMN 0 TO 63
06  ; CHAR = CHAR CODE
07          ORG   >F200
08  ;
09  XSVDC   EQU   >05F2
10  XVDC    EQU   >F120
11  ;
12  START:  SLA   R0,8
13          SLA   R1,2
14          A     R0,R1
15          MOV   R1,R9
16          ANDI  R9,>FFF8
17          CLR   R12
18          LI    R10,>0F0F
19          ANDI  R1,4
20          JNE   ODD
21          SETO  R12
22          SRC   R10,4
23  ODD:    MPY   @>1C80,R2
24          LI    R1,>5AEE
25          A     R3,R1
26          SETO  R2
27          LI    R7,>0101
28          SETO  R6
29  MORE:   CLR   R5
30          CLR   R0
31          LI    R4,>0101
32          LI    R3,6
33  CHAR1:  INV   R0
34          JEQ   EVEN
35          SRC   R4,1
36  EVEN:   SRC   R7,1
37          JNC   NOLOAD
38          MOVB  *R1+,R6
39  NOLOAD: COC   R7,R6
40          JNE   NOBIT
41          SOCB  R4,R5
42  NOBIT:  DEC   R3
43          JGT   CHAR1
44          MOV   R9,R8
45          BL    @XSVDC
46          MOVB  @XVDC,R8
47          SZCB  R10,R8
48          MOV   R12,R12
49          JNE   NOSHT
50          SRL   R5,4
```

Listing Of Assembler Source File "SMALL.S" Page 2

```
01  NOSHT:  SOCB R8,R5
02           LI   R8,>4000
03           A    R9,R8
04           BL   @XSVDC
05           INC  R9
06           MOVB R5,@XVDC
07           SRL  R2,2
08           JNE  MORE
09           RTWP
10  ;
11  ;END OF FILE
12  ;
```

Labels From Assembler File "SMALL.Z"

Date/Time = 01:04:32

XSVDC	05F2	XVDC	F120	START	F200	ODD	F21C
MORE	F22E	CHAR1	F23A	EVEN	F240	NOLOAD	F246
NOBIT	F24C	NOSHT	F262				

Basic programme to demonstrate the use of SMALL

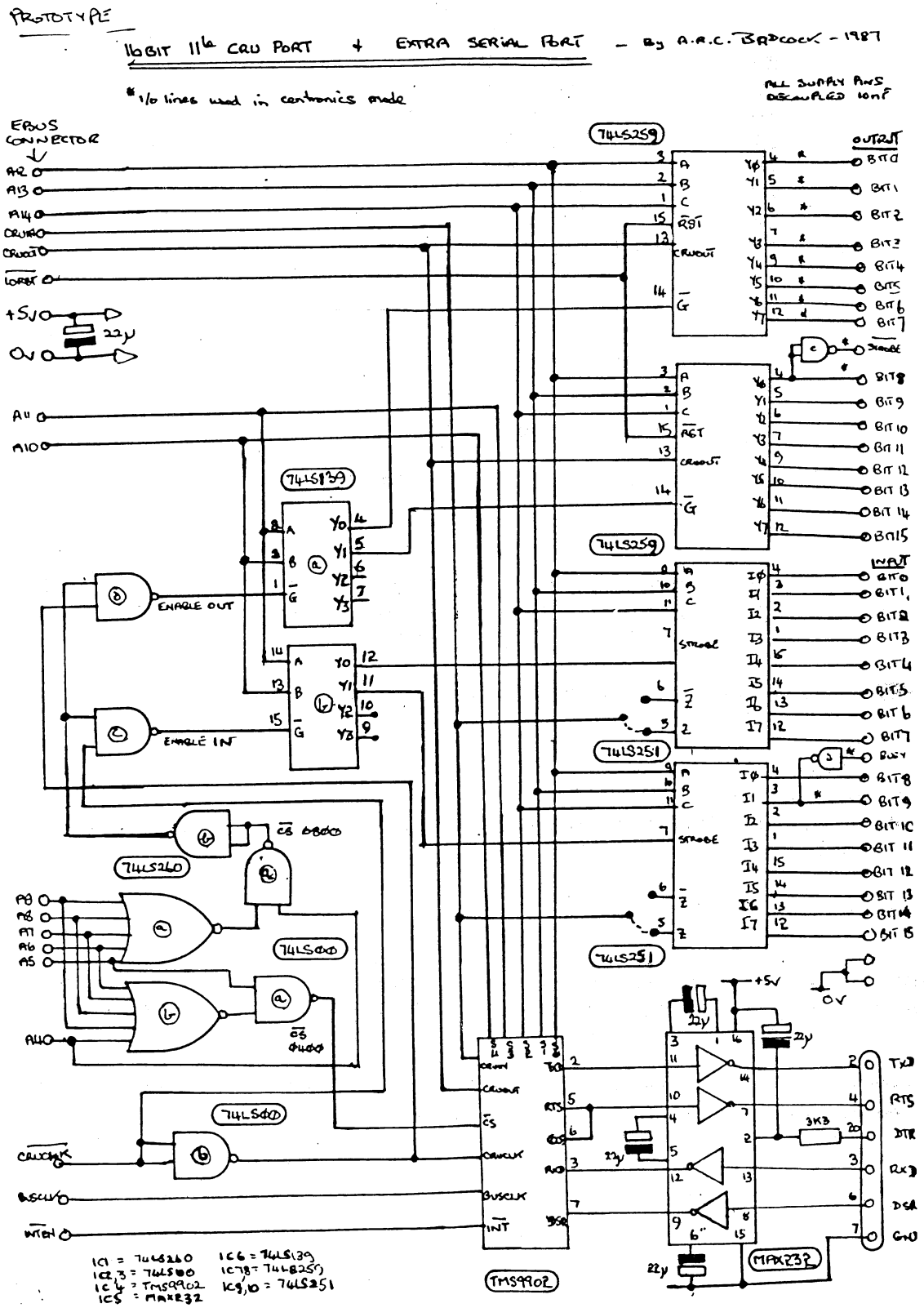
```
10  GRAPH :'B=2
20  FOR A=32 TO 64
30   CALL 0F200H,B,A-30,A+32
40  NEXT A
50  B=B+2
60  IF B<24 THEN GOTO 20
70  END
```

SMALL uses the existing character set to form the small characters and uses no memory other than the space taken by the code itself.

The programme offers a very low cost method of obtaining a 64 column screen for text layout work etc.

This circuit for a combined paralell / serial I/O card has been sent in by Alen Badcock. He says he will develope a P.C.B. for it soon.

The inputs to the 74LS260 can be shuffled for other base addresses ie, 0200H.



KEYBOARD MODIFICATIONS

by John Mackenzie

AUTO REPEAT

Have you ever wanted your keys to repeat with out reaching for that key on the far right. Well here is a simple Cct that will do just that.

On switch on the auto repeat is selected. Press a key and hold it down then after a short delay it will automatically repeat. The delay is to allow you to input just one character and release the key, with out multiple characters being input. If you press repeat then Auto repeat is switched off. Press repeat again and it is reselected.

HOW IT WORKS

The second monostable of IC5 removes any repeat key switch bounce. The repeat key then toggles the first J K flip flop of IC6. Thus selecting `repeat on´` repeat off´.

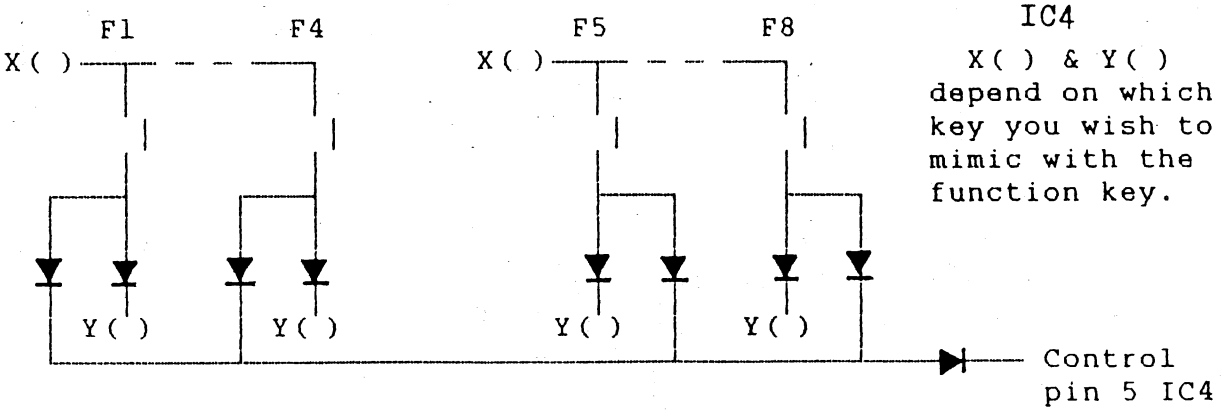
Repeat off inhibits the oscilator IC1 via 1CLR & 2CLR.

Repeat on allows IC1 via 1CLR & 2CLR, But the oscilator is inhibited for a short time after the key is pressed by the first monostable of IC5 which has been triggered by the keydown signal from IC4. The delay time can be adjusted by the 500k pot.

The repeat state is indicated by the LED.

FUNCTION KEYS

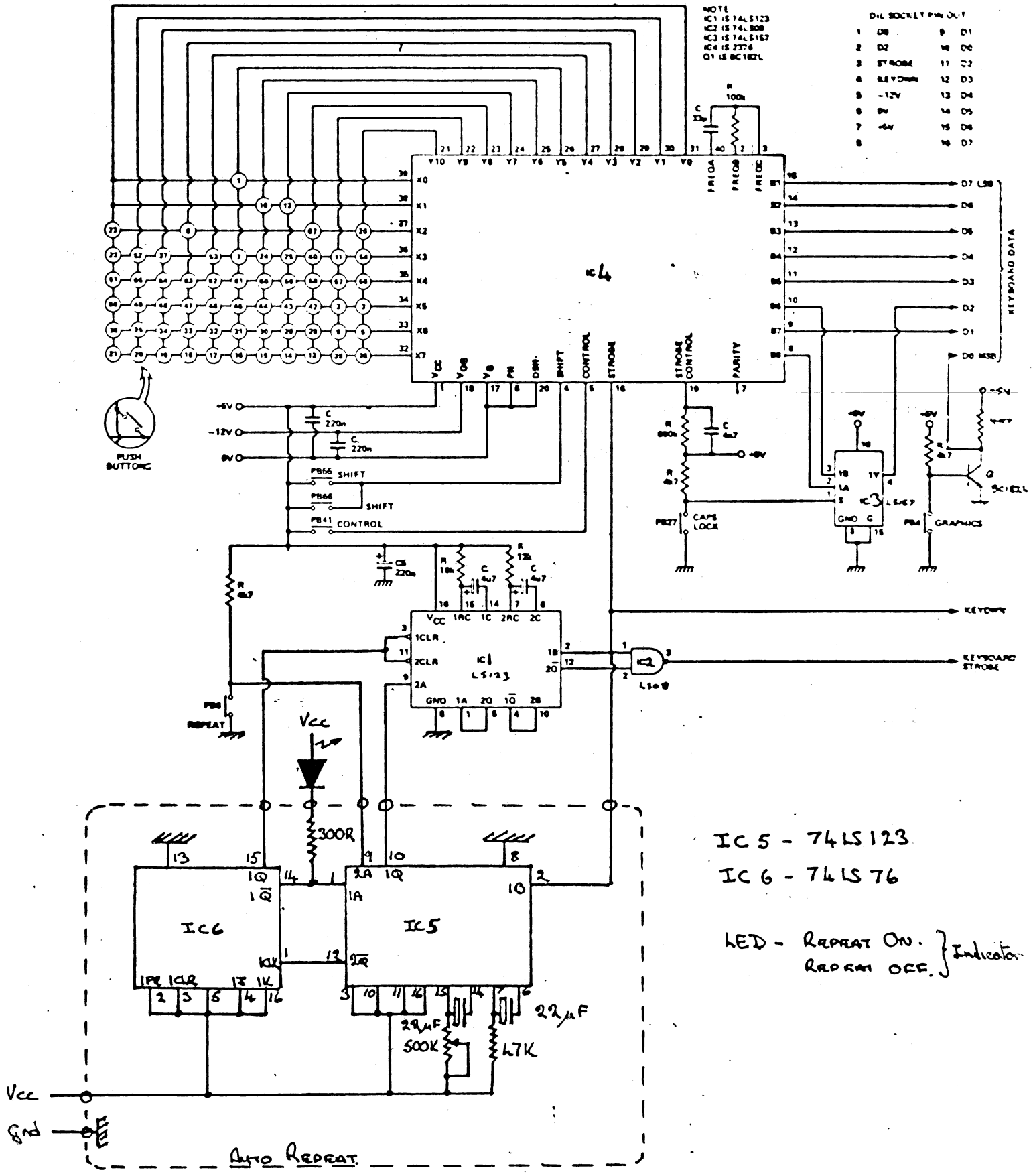
I have created a set of single pole single through Function Keys on my Cortex above the top row of keys using this simple Cct.



The Cortex basic is very good for this.

```
10 INPUT ?100,$A
20 GOTO 10
100 POP
110 A=SYS[0]
120 PRINT A
130 GOTO 10
```

Will print out the system number for the CTRL + key you operate or in the case of the above Cct the function key you press.



IC 5 - 74LS123

IC 6 - 74LS 76

LED - Repeat On. } Indicator
Repeat OFF. }

R.M. LEE

MODS TO 2 PASS ASSEMBLER TO CORRECT ERRORS IN THE @LABLE ROUTINE

```
LIST 11400 TO 11510
 11400  REM @LABEL
 11410  P=0
 11420  GOSUB 11800
 11460  FOR XX=0 TO L1-1
 11470   IF $LAB[0]=$LBL[XX,0]: GOTO 11500
 11480  NEXT XX
 11485  IF PAS=1: PRINT "NO LABEL FOUND IN LINE ";R: AE=AE+1
 11490  GOSUB 10700: GOTO 11510  !NO LABEL ON PASS 1
 11500  NUM=LBL[XX,1]
 11505  GOSUB 10050
 11510  RETURN


LIST 11800 TO 11999
 11800  REM PULL LABEL OUT OF LINE
 11810  REM DEPENDENT ON 'Z'
 11815  IF Z>0: GOTO 11900
 11820  FOR Q=1 TO LEN[$LIN[2,0]]: IF ASC[$LIN[2,0;Q]]<030H: GOTO 11840
 11830  NEXT Q
 11840  $LAB[0]=$LIN[2,0;2],Q-2
 11850  RETURN
 11900  LL=POS[",",$LIN[2,0]]+2
 11910  FOR Q=LL TO LEN[$LIN[2,0]]: IF ASC[$LIN[2,0;Q]]<030H: GOTO 11940
 11920  NEXT Q
 11940  $LAB[0]=$LIN[2,0;LL],Q-LL
 11970  RETURN
 11999  STOP


LIST 10500 TO 10670
 10500  REM FIND TD
 10510  Z=POS[",",$LIN[2,0]]+1
 10530  IF ASC[$LIN[2,0;Z]]=052H: REG=$LIN[2,0;Z+1],D: REG=REG*64: GOSUB 10230:
TD=0H: RETURN
 10540  $C[0]=$LIN[2,0;Z],2: IF $C[0]<>"*R": GOTO 10570
 10545  REG=$LIN[2,0;Z+2],D
 10546  REG=REG*64
 10547  GOSUB 10230
 10550  TD=0400H  ! REG INDIRECT
 10555  IF $D="+": TD=0C00H  !REG INDIRECT AUTO INCREMENT
 10560  RETURN
 10570  IF ASC[$LIN[2,0;Z]]<>040H: PRINT "NO '@' IN LINE ";R: AE=AE+1
 10580  IF ASC[$LIN[2,0;Z+1]]<>03EH: GOSUB 11400: GOTO 10595
 10590  $C[0]=$LIN[2,0;Z+2],4: $C[0]=$C[0]+"H": $C[0]=/"0": NUM=$C[0],D: GOSUB 1
0050
 10595  $C[0]=$HEX
 10600  TD=0800H  !SYMBOLIC
 10610  P1=POS["(R",$LIN[2,0;Z]]
 10620  IF P1=0: RETURN
 10625  P1=P1+Z
 10630  REG=$LIN[2,0;P1+1],D
 10635  REG=REG*64
 10640  IF $D<>")": PRINT "NO ')' IN LINE ";R: AE=AE+1
 10650  OP=OP LOR REG
 10660  TD=0800H  ! INDEXED
 10670  RETURN
```

CORTEX USERS GROUP

WELCOME TO THE NEW CORTEX USERS GROUP . THE FOLLOWING ARE DIRECTIONS FOR SNEYD SCHOOL

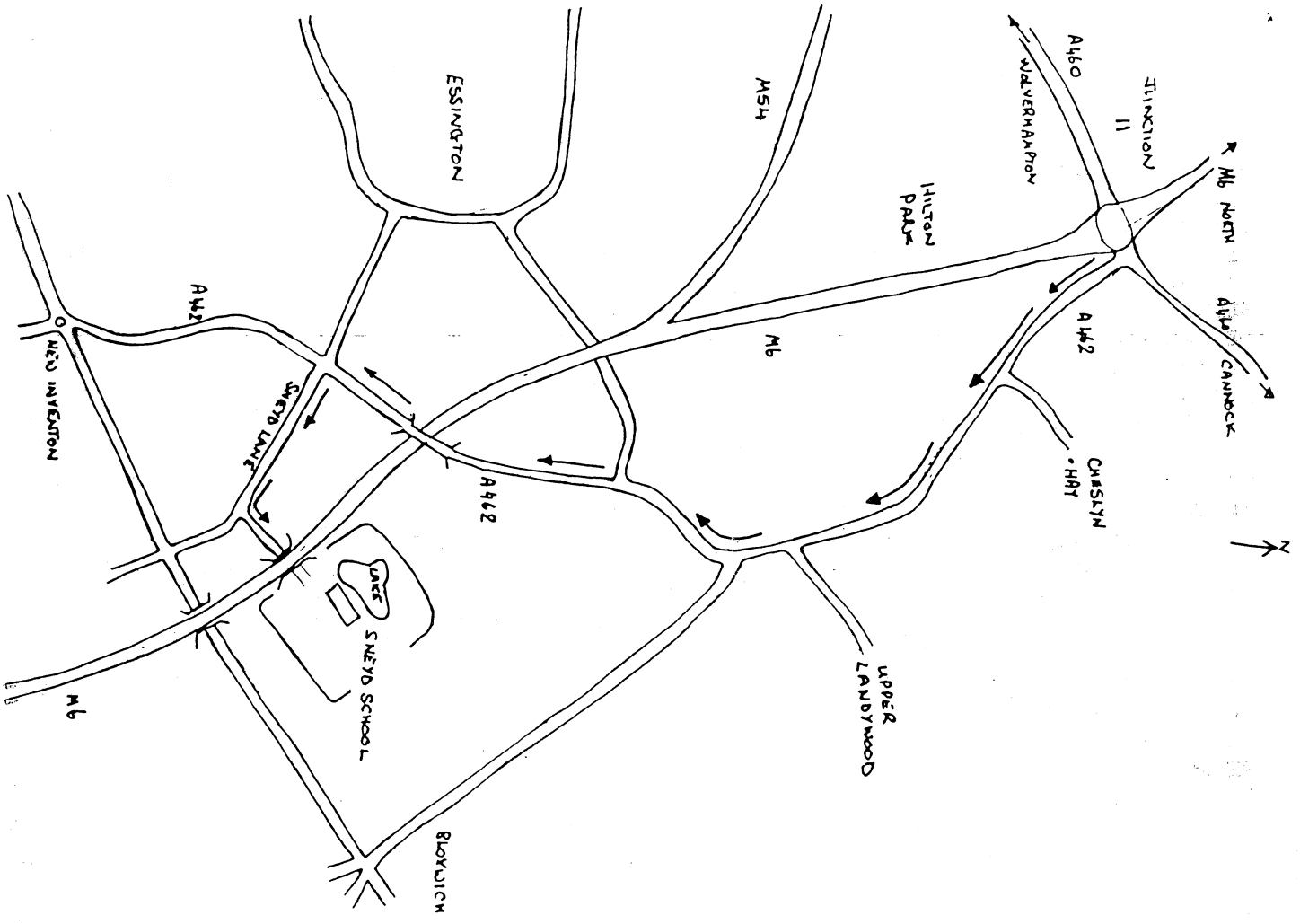
EXIT JUNCTION 11 OF MOTORWAY AND FOLLOW SIGN FOR A462

TAKE THE FIRST TURNING TO THE LEFT AFTER THE ROAD HAS PASSED OVER THE MOTORWAY

YOU SHOULD NOW BE IN SNEYD LANE

THIS ROAD IS STRAIT WITH A TIGHT BEND AT THE END ,THE TURNING FOR THE SCHOOL IS JUST BEFORE THE BEND ON THE LEFT HAND SIDE

IF YOU ARE TRAVELLING ALONG THE A5 TAKE THE A460 TURNING (CANNOCK TO WOLVERHAMPTON ROAD) THIS WILL BRING YOU ONTO JUNCTION 11



CORTEX USERS CLUB SALE

RGB INTERFACE	BARE BOARD £8.00	KIT £20.00
CENTRONICS INTERFACE	BARE BOARD £7.00	KIT £15.00
E BUS -ALL IC'S		KIT £30.00
E BUS BACK PLANE		£15.00

SEMICONDUCTORS

TMS9901	£3.00
TMS9902	£3.00
74LS612 (3 AVALABLE)	£25.00
74LS611/74LS611 (NEED PUL UP RESISTORS)	£15.00

E BUS EPANSION

E BUS (4K RAM,8K EPROM SCKT,16 IN/OUT LINES)	£15.00
NOTE-THESE CARDS ARE EX EQUIPMENT TESTED AND WORKING	
E BUS (8*8K EPROM SCKT CARD BUILT NO EPROMS FITTED)	£28.00

NEW E BUS 512K DRAM BARE BOARD £40.00

CORTEX EXPANSION

EXTERNAL VIDIO INTERFACE	BARE BOARD £15.00	KIT £80.00
DISK CONTROLER (WD 2797+BOARD)		£40.00

CORTEX SOFTWARE

DISK OPERATING SYSTEM CDOS 1.20 AND 2.00 £45.00

CDOS 1.20 FOR 9909 SYSTEM

CDOS 2.00 FOR 2797 SYSTEM

(FORMAT AVAILABLE 80T.SD.DS,80T.SS.DD,80T.SS.SD,40T.SS.SD)

(3"40T.SD)

3" HITACHI 3055/SX DISC DRIVES AVAILABLE FROM

MATMOS LTD
1 CHURCH STREET
CUCKFIELD
W SUSSEX RH17 5JZ

PRICE £24.95 + £3.00 (CARRIAGE)+VAT

MEMBERS SOFTWARE

WORTEX-WORD PROCESSING £15.00

INCLUDES SPELLING CHECKER

SEND TO J S MACKENZIE
4 WERSTAN CLOSE
MALVERN WR14 3NH

(INCLUDE TWO 5"" DD DISKS)

DRAWTECH-GRAPHICS DRAWING PACKAGE £20.00

SEND TO T GRAY
1 LARKSPUR DRIVE
FEATHERSTONE
WOLVERHAMPTON
WEST MIDLAND WV10 7TN

TWO PASS ASSEMBLER £10.00

SEND TO R M LEE
3 CHURCH AVENUE
WESTHAM
PEVENSEY
EAST SUSSEX
BN24 5LN

CORTEX USERS GROUP SOFTWARE

ALL GAMES £2.50 EACH

BURGLAR	MUNCHER
FROGGER	G DESIGN
INVADERS&ASTEROIDS	WALL
HUNCHBACK	ARCHIE
MAZE	NIGHT ATTACK
OLIMPICS NEW	CORTELLO
FIRE BIRD	RESCUE
PENGO	MOONBASE II
LABYRINTH OF TAG	CENTIPEDE