Ipso Facto

ISSUE 33 FEBRUARY, 1983

INDEX PAGE

A PUBLICATION OF THE ASSOCIATION OF THE COMPUTER-CHIP EXPERIMENTERS (ACE) 1981

Executive Corner	2
Editor's Corner	3
Members' Corner	4
1861 Line Drawing Program	8
A "Craps" Program for Quest Basic	11
Cross-Reference Chart - 1802 OP Codes	12
The EIA RS-232C Standard	13
1802 Tiny Pilot	20
Infestation II	29
A Scanning Hex Keyboard Encoder	33
Notes on Netronics Tiny Basic	35
Catalogue Sheet - ACE CPU Board	36
Club Communique	39

1982/1983 EXECUTIVE OF THE ASSOCIATION OF COMPUTER-CHIP EXPERIMENTERS

President:	Tony Hill	416-689-0175
Treasurer:	Ken Bevis	416-277-2495
Directors:	Bernie Murphy Fred Pluthero John Norris Mike Franklin	
Newsletter:		
Production Manager:	Mike Franklin	416-878-0740
Editors:	Fred Feaver Tony Hill	
Advertizing:	Fred Plutero	416-389-4070
Publication:	Dennis Mildon John Hanson	
Hardware & R. and D.:	Don McKenzie Fred Pluthero Ken Bevis Mike Franklin	416-423-7600

Vice-President:	John Norris	416-239-8567
Secretary:	Fred Feaver	416-637-2513
Membership:	Bob Silcox Earle Laycock	416-681-2B48
Program Convener:		
Tutorial/Seminars:	Ken Bevis Fred Feaver	
Software:	Wayne Bowdish	416-388-7116
Product Mailing:	Ed Leslie (Publication)	416-528-3222
	Fred Feaver (Boards)	416-637-2513

CLUB MAILING ADDRESS:

A.C.E.
c/o Mike Franklin
650 Laurier Avenue
Milton, Ontario
Canada
L7T 4R5
416-878-0740

CLUB MEETINGS

Meetings are held on the second Tuesday of each month, September through June at 7:30 in Room B123, Sheridan College, 1430 Trafalgar Road, Oakville, Ontario. A one hour tutorial proceeds each meeting. The college is located approximately 1.0 km north of QEW, on the west side. All members and interested visitors are welcome.

ARTICLE SUBMISSIONS

The majority of the content of Ipso Facto is voluntarily submitted by club members. While we assume no responsibility for errors nor for infringement upon copyright, the Editorial staff verify article content as much as possible. We can always use articles both hardware and software of any level or type relating directly to the 1802 or to micro computer components periferals, products etc. Please specify the equipment or support software upon which the article content applies. Articles which are typed are prefered, and usually printed first, while handwritten articles require some work. Please, please send originals, not photocopy material. We will return photocopies of original material if requested. Photocopies usually will not reproduce clearly.

ADVERTISING POLICY

ACE will accept advertising for commercial products for publication in Ipso Facto at the rate of $25 per quarter page per issue with the advertiser submitting camera-ready copy. All advertisments must be pre-paid.

PUBLICATION POLICY:

The newsletter staff assume no responsibility for article errors nor for infringement upon copyright. The content of all articles will be verified, as much as possible and limitations listed (i.e. Netronics Basic only, Quest Monitor required, require 16K at 0000-3FFF etc.). The newsletter staff will attempt to publish Ipso Facto by the first week of: Issue 31 - Oct. 82, 32 - Dec. 82, 33 - Feb. 83, 34 - Apr. 83, 35 - June 83, and 36 - Aug. 83. Delays may be incurred as a result of loss of staff, postal disruptions, lack of articles, etc. We apologize for such inconvenience, however, they are generally caused by factors beyond the control of the club.

MEMBERSHIP POLICY:

A membership is contracted on the basis of a club year - September through the following August. Each member is entitled to, among other privileges of membership, all 6 issues of Ipso Facto published during the club year.

Editor's Corner

I would like to thank the one person who cared enough about ACE to write the Editor, and the four persons who sent programs for publication. All are reproduced in this Newsletter.

CLUB BOARDS

We had a significant run on ACE boards over the last four months, with many boards being sold out and even re-orders sold out. For those of you who had to wait - our apologies; we hope you soon receive the board you ordered.

For those of you who ordered the new VDU Board (ver.2) you should receive it shortly after receiving this Newsletter.

Board projects currently underway: a modem using the new LSI chip AM7910 - the board will be available within four months.

CLUB CONFERENCE

RCA have agreed to assist ACE in holding another conference this year - probably in late August or early September. The conference will be held on a Saturday, either in Oakville or Welland.

More information will be given in the next issue. Plan to attend (RCA gives out lots of goodies)!

Members' Corner

FOR SALE:

B. Willem, R.R.#2, Fisherville, Ontario. NOA 1GO (416) 779-3057.

1 Aluminum Card Rack with Plastic Guides with ACE Backplane ver. 1 mounted on it, c/w sockets.

1 ACE 64K Dynamic Memory Board

1 ACE 8K Eprom Board

1 ACE Kluge Board

1 ACE VDU Board, rev.1, MC6847 & 1372

1 TEC CPU Board with IC's

1 TEC IF1 Board with IC's

1 TEC 6K Memory Board

1 Keyboard

Complete Set of Instructions. COMPLETE LOT: $300.00 or BEST OFFER

W. Steiner, 1204 2725 Melfa Rd., Vancouver, B.C. V6T 1N4 (688) 228-1733

ACE NAB (Netronics - Ace Adapter Board) BOARD
Never been used. Partially socketed with 2 86 pin connectors. Complete with schematics and documentation. $20.00.

O. Hoheisel, Hermann-Bossdorf-Str.33, 2190 Cuxhaven, West Germany

a) Netronics Video Display Board (300 Baud max.), assembled, all ICs in sockets, includes dip headers and switches for quick characters/line change US$ 70 (Can$ 84)
b) Netronics 4k static RAM Board assembled by Netronics (i.e. without sockets), with DIP switch US$ 45 (Can$ 55)
c) Same as b), except nedds troubleshooting (probably 74C902 chip) US$ 30 (Can$ 36)
d) Quest Super Monitor Ver. 1.1 in 2708 EPROM, runs at address C000 with stack page D800 (not 8000 and 9800 as in Quest system). No manual or other info available US$ 12 (Can$ 15)
e) Integrated circuit MC 3480 L US$ 8 (Can$ 10)
f) Integrated circuit MC 3242 AP US$ 7 (can$ 9)

All prices include shipping (to US or Canada) and (except d)) all original documentation. Payment by bank draft, order cheque, etc. or money order (preferred).

Thomas E. Jones
Berlinerstr. 20
6944 Hemsbach (FRG)

Even though the ACE COSDOS is being released, I suspect we can still use sophisticated cassette based O.S.'s as well. this paper describes an expanded version of Steve Nies's MONITOR and TEXT EDITOR, reviewing them a bit, and introduces UPII which adds value to both of them. Each section will run in a 2716.

The 6K package offers a good full screen text editor for all terminals, and now includes for the 6847 an upper/lower case emulation (like the TRS80 Color) and auto-scrolls a "window" accross 80 columns of text. Cassette motor controls are now included.

The major advance is the use of device independant I/O. Existing programs can now be linked to "Logical Channels" in the Monitor, and then your device drivers assigned to that channel by using the command: ASsign N=dv. Compatible drivers can be swapped at will,(video and printer, for example.) There are selected devices assigned to the channels, also, on a cold start of monitor. Judicial selection of channels when installing a program will allow "load and run" ability without use of AS commands. The AS command can still over-ride any of these initial defaults. In fact, programs can be written to change output (input) devices directly, and since Monitor also uses I/O channels for it's services, even it's console devices can be re-assigned.

The basic monitor provides a means of: running programs, continuing after a break, examining/modifying both memory and registers R2-RF, stepping through memory, setting breakpoints in both RAM and ROM, tracing programs, filling blocks of memory with any hex code, searching blocks for hex strings, copying blocks of memory to other areas, and file handling on cassettes. Named files can be saved, loaded and verified either to the original memory it was recorded from, or the memory block specified.

The Text-Editor command TE 'NAME' 1000 will start the cassette motor, find the file labeled 'NAME' and load it, stop the cassette, reserve 1000 (hex) bytes for the text buffer, and display the first page of text ready to edit. The text-ed Save, Load, Move, Copy, etc. commands use a pointer on screen to define the start and end of blocks of text they operate upon....not hex addresses.

UPII has a BOot command to execute non-SCRT programs with X=0, P=0 on entry. It also features Netronics ElfII format Save and Load commands. If the load does not finish, the last location of memory loaded to will be displayed on the screen.

A disassembler command will generate full mneumonic lists, with addresses and codes, and also has options for formatted hex listings and legal-ascii only lists. This command is based on the work of Harley Shanko, Van Nuys, Calif.

The HEX command is a data entry service to allow quicker entry of hex programs from dumps, using the hex keypad while displaying in the hex-dump format on screen as you enter. The TYPE command uses the printer and ascii keyboard as a typewriter to add footnotes, corrections, etc. to hardcopy.

Device drivers presently included in the UPII eprom include a hex-pad driver, uart driver, printer driver, and paper-tape Rdr driver. The Monitor contains the 6847 driver and ascii keyboard I/O. The initialization sets up the ACE Vid-Bd. The PTR DRVR will read Intel hex-ascii tapes and output hex only.

Since the listings for UPII alone run 30 pages, I think it will be best for anyone interested to contact me direct and we can arrange for listings, tapes, eproms, etc. I also have patches for installing other programs to the monitor. For ElfII serial VID-BD users I have some patches and serial drivers for Monitor that were developed thanks to work by Fred Hannan, of East Lyme, Ct.

This operating system does have some hardware associated with it, of course. It is possible that I could arrange to produce a System Monitor Board containing the Cmos Uart, memory mapped decoders, special break/trace interrupt circuit, cassette relays, and a parallel I/O port. ACE or ElfII buss. Let me know if you want it.

Residents of the USA can write to me easier to this address;

Tom Jones
c/o 295th AVN CO., SFTS DET.
APO N.Y., 09028

FRED HANNAN
10 Filosi Road
East Lyme, Conn. 06333

December 20, 1982

Dear Mike:

I think Wes Steiners comment in IF #32 is appropiate.

The small cadre of authors submitting articles are those most deeply interested in "High Tech" modifications, disk operating systems, etc. which, I believe, are beyond the desires and understanding of the majority of the membership. Since the "High Tech" trend has been going on for quite some time, it may account for the decline in general interest articles and, possibly, membership.

Speaking only for myself, I found nothing in IF #32 that I wanted to read twice. The last articles I really dug into were yours on your EPROM BURNER and SYMON. Again, echoing Wes, if I wanted or needed a disk system, I wouldn't bother trying to kludge up an 1802 but would buy an Apple or Lobo or Morrow or some such. I wouldn't be too much surprised if most members keep their 1802s because they are easy to program and simplistic in design. Asking the average member to hang disks on his system or replace his motherboard is just driving him away. Not too many of us have really realized the full potential of the original system and have been left behind by the "High Tech" group.

I don't know what the membership count is at present but selling ONLY 70 copies of Forth to a membership that used to number in the neighborhood of 500 seems to illustrate my point. Those of the members that are fortunate enough to be able to attend the meetings in person can iron out their problems together, patch up Forth together, etc., but where does that leave us who cannot attend?

I believe you once stated that you did not wish to put large code listings into IF. This was a good idea when the trend of the articles was still fairly elementary. Now, it might be the only way to get usable, working, programs out to the members and start raising our programming abilities, as well as raising the interest level of members. As far as I can recall, the various monitors have been the only large code articles in the past year or so that might be called "General Interest". When was the last time a game article was printed? Any utility program?

In memory serves me right, the last membership survey showed that the vast majority of members owned ELFs - either Netronics or Quest. Most of these members are probably happy with their system as it is and are not "High Tech" hardware tinkerers, in spite of the fact that they joined a club named "Association Of Computer-Chip Experimenters". I guess what I am trying to say is that the FEW "High Tech" members are leaving the MANY "Low Tech" members behind and may be the only ones left in the club in time.

I hope I have not offended you or anyone else but this is how I feel.

Fred

Dear Fred:

Thank you for your comments. I appreciate the time you took to write - the only letter I received in response to my editorial.

You may be right, that most of our members don't appreciate "High Tech" and need "Lo Tech" articles - but where do those articles come from? How do people learn to be "High Tech"? I print what I get, within an issue or two of receiving it, and after I have verified that it works. I write what interests me and at my current level - as do most of our members. The sad truth is that few of our members care enough about the Club to contribute anything to it, and quite possibly this will be the last year we have a publication to turn to.

One of the great joys of being part of something is contributing to the organization, and being recognized by others as contributing. I will print anything and just about everything from the Membership. ACE came into existence in response to a need for an information exchange on the 1802. We have been quite possibly the most prolific and diverse club anywhere - we had to be, for there was no one else to support the 1802.

Perhaps the time has come for the club to fold and for you to go your own way .... I will be saddened by its passing.

Mike Franklin

1861 Line Drawing Program

- by J. Munch, 20228 Clark Street, Woodland Hills, Ca. USA 91367

This program demonstrates a LINE DRAWING program on the ubiquitous 1861.

A subroutine (0016-0024) generates random values for a zig-zag display. Inputs X0, X1, Y0 and Y1 are used for the current line. X1 and Y1 are saved in RB to be used the second time around as beginning points for the next line.

Many variations for inputting the lines are possible, such as loading via a register and advancing (4N). Then by storing each line (X0, X1, Y0, Y1) in a reg. as many lines as memory available/4 can be drawn. For the 1861 (in the 1K mode) the range for X=00 to 3F, and for Y the range is 00 to 7F.

The program has also been used with the 6847 chip in the Resolution Graphics Two mode, with but slight adjustment in the plot routine to account for the grid. This mode has 128 pixels horizontally by 96 vertical where each dot is the size of the character"period" in the standard alpha-numeric mode.

On a standard TV set the pixels are square, and the available dots are six times that of the 1861 operating with a 64 X 32 display.

Some background:

A search for a plot routine led me to an article in EDN, May 27, 1981, where the RS Color Computer, also using the 6847, was discussed. Some 6809 machine code showed how the plotting of points were performed. This code was then laboriously converted to "1802". Being able to plot dots on the screen, with Tinys USR calls led to a search for a line routine. Another article in the August 81 Byte, Programming Quickies, by Mike Higgins, showed how to use his form of the DDA, (Digital Differential Analyzer). A sample Basic program, where lines of any slope between -1 and 1 could be drawn was given. Converting to Tiny was easy.

Many evenings were spent playing around with variations of this Basic line drawing method. The line draw routine, now a proven program, was then written in 1802 code, and then made part of the code that the USR called.

From Basic it was then possible to generate lines by providing four inputs: beginning points X0, Y1 and end points X1, Y1.

(If X0=X1 and Y0=Y1 a single point is plotted)

******** THE 1861 IN THE HI-RESOLUTION MODE, 64 HOR 128 VERT. **********

The program uses one page for the code and four pages for the display.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	90	B1	B2	B3	B4	B7	B8	F8	FF	A2	F8	27	A3	F8	17	A7
0010	F8	E1	A1	B9	D3	00	D3	99	FE	52	FE	F3	FE	89	7E	A9
0020	99	7E	B9	30	16	00	00	EA	F8	00	A4	F8	04	BA	F8	FF
0030	AA	F8	00	73	9A	3A	31	1A	E2	69	D7	37	3B	84	FB	1F
0040	32	27	14	F8	00	BE	AF	F8	01	AE	BF	9B	52	A8	99	FA
0050	3F	BB	F7	BD	33	5C	FD	00	BD	F8	FF	BF	8B	52	AA	89
0060	FA	7F	AB	F7	AD	33	6D	FD	00	AD	F8	FF	AE	9D	52	8D
0070	F5	33	81	9D	AC	8D	BD	8C	AD	9F	BE	8E	AF	F8	00	AE
0080	BF	9D	F6	BC	F8	01	AC	8A	73	88	73	F6	F6	F6	F9	F8
0090	52	88	FA	07	FC	F0	A8	F8	20	BA	8A	FE	AA	9A	7E	BA
00A0	3B	9A	8A	F5	AA	F8	04	52	9A	75	BA	08	52	0A	F1	5A
00B0	60	72	A8	F0	AA	8C	52	9D	F5	3B	BD	3A	38	88	52	9F
00C0	F4	A8	8A	52	8F	F4	AA	9C	52	8D	F4	BC	1C	52	9D	F5
00D0	3B	87	BC	88	52	9E	F4	A8	8A	52	8E	F4	AA	30	87	72
00E0	70	C4	22	78	22	52	E2	E2	F8	01	B0	F8	00	A0	30	DF
00F0	80	40	20	10	08	04	02	01	00	00	00	00	00	00	00	00

Execute @ 0000, INP 1(69) @ 0039, Pressing INPUT KEY halts motion.

REGISTER:	VALUE:	DESCRIPTION:
R0	01 00	1861-DMA
R1	00 E1	INTERRUPT PTR
R2	00 FF	TOP OF STACK
R3	00 27	START OF MAIN
R4	00 00	R4.1 NOT USED, R4.0=LINE CTR,(User can adjust @ 003F)
R5	-- --	SPARE REGISTER
R6	-- --	SPARE REGISTER
R7	00 17	RANDOM SUBROUTINE PTR.
R8	00 (X)	R8.1=PIXEL PAGE, R8.0=X COORDINATE
R9	E1 --	RANDOM REG, SEEDED WITH NON-ZERO (E1)
RA	01 (Y)	RA.1=SCREEN PAGE(01 TO 04), RA.0=Y COORDINATE
RB	XX YY	RB.1=TEMPORARY X, RB.0=TEMPORARY Y, NEW BEGINNING PTS
RC	(D3 N1)	LINE DRAWING REGISTERS.
RD	(DX DY)	" " "
RE	(S1 S2)	" " "
RF	(A1 A2)	LINE DRAWING REGISTERS.

Please refer to flow chart.

If this routine is to be modified for some other application it should noted that HOME is at the lower left corner of the screen, 1st quadrant.

```
7F ..............................................
 ^                                              1
 1                                              1
 1                                              1
 1                 Center:                      1
 1.................20,40 H.....................1
 1                   X   Y                      1
 1                                              1
 1                                              1
00..............................................>3F
```

==========

new subject...

T.Hills WINDOW is a program that should have been delivered with any new system. Thank you.

When bringing up Window, a usually flawless tape recorder,gave me a lot of trouble . The problem was traced to a faulty lever arm inside the recorder, The end result was to put WINDOW in EPROM. Some spare space ?? at 03F3 -03FF and 04F2-04FF was used for a specific routine to down load Window to RAM @ (in my case)2000 H. Even if this space is used in Window , having it in RAM should present no problems. Now that WINDOW is resident it is being put to good use,probably more so now, than if it had been on tape.

Even if it takes a 6847, or similar display generator, to use WINDOW , do it! Comments on the line drawing program would be appreciated.

LINE DRAWING → POINT PLOT – USING 8 BIT VALUES D7 = SIGN BIT

BASIC LINE #. P. 416.

6020,30,40,50 — ENTRY: SAVE REGISTERS IF REQUIRED

S1=0, S2=1, A2=0, A1=1

50,70 6000 — INPUT X0, INPUT X1: DX=X1−X0

6060 — IS DX >= 0 ? YES / NO

6080 6070 — DX = −DX; A1 = −1

50,70 6010 — INPUT Y0, INPUT Y1: DY=Y1−Y0

6090 — IS DY >= 0 ? YES / NO

6100 6110 — DY = −DY; S2 = −1

6120 — IS DX >= DY ? YES / NO

6130,40 6150,60 6181,70,90 — N = DX, DX = DY; DY = N, S1 = A1; A2 = S2, S2=0, A1=0

6200 6210 — D3 = DX/2; N1 = 1

90 — SAVE X0,Y0; PLOT POINT *; RECOVER X0,Y0

100 — IS N1 > DX ? YES → EXIT (RESTORE REGISTERS) / NO

6300 6310 6320 6330 — X0=X0+A1; Y0=Y0+A2; D3=D3+DY; N1=N1+1

6340 — IS D3 <= DX ? YES / NO

6350 6360 6370 — D3=D3−DX; X0=X0+S1; Y0=Y0+S2

(THE LINE # ARE FROM P. 416.)

NOTE: THIS PROGRAM IS BASED ON AN ARTICLE IN BYTE, AUG 81 PAGE 414 — 416, PROG. QUICKIES. MIKE HIGGINS DESCRIBED IN DETAIL HOW A LINE COULD BE DRAWN BETWEEN POINTS, ANY DIRECTION AND ANY SLOPE WHEN USING HIS VERSION OF THE DDA, (DIG. DIFFERENTIAL ANALY)

```
F8#ØØ, BE AF , F8#Ø1 AE BF
F8 | XØ | 52, A8
F8 | X1 | F7, BD

33 [ ]
FD ØØ  BD
F8#FF  BF

→ F8 | YØ | 52, AA
  F8 | Y1 | F7, AD

33 [ ]
FD ØØ , AD
F8#FF , AE

→ 9D 52
  8D F5
  33 [ ]
  9D AC , 8D BD,
  8C AD , 9F BE , 8E AF
  F8 ØØ , AE BF

→ 9D, F6, BC
  F8#Ø1 AC

→ 8A 73, 88 73
  * NOT SHOWN. DISPLAY PECULIAR
  6Ø, 72 A8, FØ AA
  8C 52
  9D F5
  3B [ ]
  3A [ ] → EXIT

→ 88 52, 9F F4 A8
  8A 52, 8F F4 AA
  9C 52, 8D F4 BC,
  1C
  52
  9D F5
  3B [ ] →
  BC,
  88 52, 9E F4 A8
  8A 52, 8E F4 AA
  3Ø [ ]
```

RØ		
R1		
R2		
R3		
R4		
R5		
R6		
R7		
R8	PIX PAGE	X COOR
R9		
RA	SCREEN	Y COOR
RB		
RC	D3	N(1)
RD	DX	DY
RE	S1	S2
RF	A1	A2

A "Craps" Program for Quest Basic 3.0

- by P. Muir, Scarborough, Ontario.

```
 10  CLS:  PRINT   "         CRAPS" : PRINT
 20  B=100
 30  PRINT "  A TOTAL OF 7 OR 11 ON THE "
 40  PRINT "FIRST THROW WINS.  YOU CAN ALSO"
 50  PRINT "WIN BY THROWING A 4, 5, 6, 8, 9, 10"
 60  PRINT "AND MATCHING IT BEFORE THROWING"
 70  PRINT "A 7.  IF ON THE FIRST THROW A 2"
 80  PRINT "3 OR 12 COMES UP, YOU LOSE."
 90  PRINT "HIT LINE FEED TO CONTINUE";
100  X=INP(1)
110  IF X <>#0A GOTO 100
120  C=0:CLS
130  PRINT "ACCOUNT=$";B," ",
140  INPUT "BET"B1
150  IF B1>B GOTO 120
160  D1=INT (6*RND)+1
170  D2=INT (6*RND)+1
180  T=D1+D2
190  FOR N=1 TO 10:  PRINT  :  NEXT N
200  GOSUB 480
210  C=C+1
220  IF C>1 GOTO 310
230  T0=T
235  WAIT (200)
240  IF T=7 GOTO 360
250  IFT=11 GOTO 360
260  IF T<4 GOTO 400
270  IF T=12 GOTO 400
280  CLS
290  PRINT "ACCOUNT0=$" ;B," ","BET=";B1
300  GOTO 160
310  IF T=7 GOTO 400
320  IF T=T0 GOTO 360
330  PRINT "TRYING TO MATCH" ;T0,C; "ROLLS"
340  WAIT (250)
350  GOTO 280
360  B=B+B1
370  PRINT " **********  YOU WIN  **********"
380  PRINT"YOUR TOTAL=$";B:IF B>1000 PRINT"LET'S SEE THOSE DICE,YOU!"
390  GOTO 440
400  B=B-B1
410  PRINT " --------- YOU LOSE --------- YOU LOSE ---------"
420  PRINT "YOUR TOTAL =$";B
430  IF B<=0 PRINT "THROW THE BUM OUT!"
440  INPUT "TYPE E'E' TO EXIT "E$
450  IF E$="E" END
460  GOTO 120
470  END
480  V=@E065:D=D1:  GOSUB 510
490  V=@E072:D=D2:  GOSUB 510
500  RETURN
510  FOR N=0 TO 8
520  POKE (V+N,#43)
530  NEXT N
540  FOR N=1 TO 4
550  Q=V+32*N
560  POKE (Q,#4F):  POKE (Q+8,#4F)
570  NEXT N
580  Q=V+128
590  FOR N=1 TO 7
600  POKE (Q+N,#43)
610  NEXT N
620  IF D>2*INT(D/2) POKE (V+68,#2E)
630  IF D=1 RETURN
640  POKE (V+34,#2E): POKE (V+102,#2E)
650  IF D<4 RETURN
660  POKE (V+38,#2E):  POKE (V+98,#2E)
670  IF D<6 RETURN
680  POKE (V+66,#2E): POKE (V+70,#2E)
690  RETURN
```

Cross Reference Chart - 1802 OP Codes to User Manual Page Number (MPC201C)

- by Derek Claridge, Nelson B.C.

0X	1X	2X	3X	4X	5X	6X	7X	8X	9X	AX	BX	CX	DX	EX	FX	
38	16	16	31	18	20	16	41	17	17	17	18	35	39	39	19	X0
18			33			42	41					36			21	X1
			32				19					35			22	X2
			33				20					36			22	X3
			34				26					38			25	X4
			34				28					38			27	X5
			34				24					37			23	X6
			34				30					38			30	X7
			32/37				40					35/37			19	X8
			34				41					36			21	X9
			33				40					36			23	XA
			33				39					36			22	XB
			34				27					38			26	XC
			34				29					38			28	XD
			34				25					37			24	XE
			34				31					37			30	XF

THE EIA RS-232-C STANDARD

AN INTERFACE BETWEEN DATA TERMINAL EQUIPMENT

AND

DATA COMMUNICATION EQUIPMENT

EMPLOYING SERIAL BINARY DATA INTERCHANGE

Published by

ELECTRONIC INDUSTRIES ASSOCIATION
Engineering Department
2001 Eye Street, N.W., Washington, D.C. 20006
August 1969

The following excerpts from the above Standard are provided to assist you in applying RS-232-C communications on your micro.

Signal

During the transmission of data, the marking condition shall be used to denote the binary state ONE and the spacing condition shall be used to denote the binary state ZERO.

For timing and control interchange circuits, the function shall be considered ON when the voltage (V_1) is more positive than plus three volts with respect to signal ground. The function is not uniquely defined for voltages in the transition region between plus three volts and minus three volts.

Signal Voltage	Negative (-12v)	Positive (+12v)
Binary State	1	0
Signal Condition	Marking	Spacing
Function	OFF	ON

Interchange Circuits (defined by pin number of a Db 25 connector). DB25 pins 1 to 7 and 20 form the "typical micro-computer circuit".

1. Pin 1 - Protective Ground
   Direction: Not applicable

   This conductor shall be electrically bonded to the machine or equipment frame. It may be further connected to external grounds as required by applicable regulations.

2. Pin 2 - Transmitted Data
   Direction: TO data communication equipment (computer to device).

   Signals on this circuit are generated by the data terminal equipment and are transferred to the local transmitting signal converter for transmission of data to remote data terminal equipment.

   The data terminal equipment shall hold transmitted data in marking condition during intervals between characters or words, and at all times when no data are being transmitted.

   In all systems, the data terminal equipment shall not transmit data unless an ON condition is present on all of the following four circuits, where implemented:

   1. Request to Send
   2. Clear to Send
   3. Data Set Ready
   4. Data Terminal Ready

   All data signals that are transmitted across the interface Transmitted Data during the time an ON condition is maintained on all of the above four circuits, where implemented, shall be transmitted to the communication channel.

3. Pin 3 - Received Data
   Direction: FROM data communication equipment (device to computer)

   Signals on this circuit are generated by the receiving signal converter in response to data signals received from remote data terminal equipment via the remote transmitting signal converter. Received Data shall be held in the binary ONE (Marking) condition at all times when Received Line Signal Detector is in the OFF condition.

   On a half duplex channel, Received Data shall be held in the Binary One (Marking) condition when Request to Send is in the ON condition and for a brief interval following the ON to OFF transition of Request to Send, to allow for the completion of transmission.

4. Pin 4 - Request to Send
Direction: TO data communication equipment (computer to device)

This circuit is used to condition the local data communication equipment for data transmission and on a half duplex channel, to control the direction of data transmission of the local data communication equipment.

On one way only channels or duplex channels, the ON condition maintains the data communication equipment in the transmit mode. The OFF condition maintains the data communication equipment in a non-transmit mode.

On a half duplex channel, the ON condition maintains the data communication equipment in the transmit mode and inhibits the receive mode. The OFF condition maintains the data communication equipment in the receive mode.

5. Pin 5 - Clear to Send
Direction: FROM data communication equipment (device to computer)

Signals on this circuit are generated by the data communication equipment to indicate whether or not the data set is ready to transmit data.

The ON condition, together with the ON condition on interchange circuits Request to Send, Data Set Ready and, where implemented, Data Terminal Ready, is an indication to the data terminal equipment that signals presented on Transmitted Data will be transmitted to the communication channel.

The OFF condition is an indication to the data terminal equipment that it should not transfer data across the interface on interchange Transmitted Data.

The ON condition of Clear to Send is a response to the occurrence of a simultaneous ON condition on Data Set Ready and Request to Send, delayed as may be appropriate to the data communication equipment for establishing a data communication channel (including the removal of the MARK HOLD clamp from the Received Data interchange circuit of the remote data set) to a remote data terminal equipment.

6. Pin 6 - Data Set Ready
Direction: FROM data communication equipment (device to computer)

Signals on this circuit are used to indicate the status of the local data set.

The ON condition on this circuit is presented to indicate that -

a) the local data communication equipment is connected to a communication channel ("OFF HOOK" in switched service),

AND b) the local data communication equipment is not in test (local or remote), talk (alternate voice) or dial* mode.

AND c) the local data communication equipment has completed, where applicable:

1. any timing functions required by the switching system to complete call establishment, and

2. the transmission of any discreet answer tone, the duration of which is controlled solely by the local data set.

7. Pin 7 - Signal Ground or Common Return
Direction: Not applicable.

8. Pin 20 - Data Terminal Ready
Direction: TO data communication equipment (computer to device)

Signals on this circuit are used to control switching of the data communication equipment to the communication channel. The ON condition prepares the data communication equipment to be connected to the communication channel and maintains the connection established by external means (e.g., manual call origination, manual answering or automatic call origination).

When the station is equipped for automatic answering of received calls and is in the automatic answering mode, connection to the line occurs only in response to a combination of a ringing signal and the ON condition of Data Terminal Ready. However, the data terminal equipment is normally permitted to present the ON condition on Data Terminal Ready whenever it is ready to transmit or receive data, except as indicated below.

The OFF condition causes the data communication equipment to be removed from the communication channel following the completion of any "in process" transmission. The OFF condition shall not disable the operation of Ring Indicator.

---

* The data communication equipment is considered to be in the dial mode when circuitry directly associated with the call origination function is connected to the communication channel. These functions include signalling to the central office (dialing) and monitoring the communication channel for call progress or answer back signals.

EIA RS-232-C STANDARD

The following signals are typically implemented in a modem installation:

1. Pin 8 - Received Line Signal Detector
   Direction: FROM data communication equipment (device to computer)

   The ON condition on this circuit is presented when the data communication equipment is receiving a signal which meets its suitability criteria. These criteria are established by the data communication equipment manufacturer.

   The OFF condition indicates that no signal is being received or that the received signal is unsuitable for demodulation.

   The OFF condition of Received Line Signal Detector shall cause Received Data to be clamped to the Binary One (Marking) condition.

   The indications on this circuit shall follow the actual onset or loss of signal by appropriate guard delays.

   On half duplex channels, Received Line Signal Detector is held in the OFF condition whenever Request to Send is in the ON condition and for a brief interval of time following the ON to OFF transition of Request to Send.

2. Pin 15 - Transmitter Signal Element Timing (DCE Source)
   Direction: FROM data communication equipment (device to computer)

   Signals on this circuit are used to provide the data terminal equipment with signal element timing information. The data terminal equipment shall provide a data signal on Transmitted Data in which the transitions between signal elements nominally occur at the time of the transitions from OFF to ON condition of the signal on the circuit. When this circuit is implemented in the DCE, the DCE shall normally provide timing information on this circuit whenever the DCE is in a POWER ON condition. It is permissible for the DCE to withhold timing information on this circuit for short periods, provided Data Set Ready is in the OFF condition. (For example, the withholding of timing information may be necessary in performing maintenance tests within the DCE).

3. Pin 17 - Receiver Signal Element Timing (DCE Source)
   Direction: FROM data communication equipment.

   Signals on this circuit are used to provide the data terminal equipment with received signal element timing information. The transition from ON to OFF Condition shall nominally indicate the center of each signal element on Received Data. Timing information on the circuit shall be provided at all times when Received Line Signal Detector is in the ON condition.

4. Pin 21 - Signal Quality Detector
   Direction: FROM data communication equipment (device to computer)

Signals on this circuit are used to indicate whether or not there is a high probability of an error in the received data.

An ON condition is maintained whenever there is no reason to believe that an error has occurred.

An OFF condition indicates that there is a high probability of an error. It may, in some instances, be used to call automatically for the retransmission of the previously transmitted data signal. Preferably the response of this circuit shall be such as to permit identification of individual questionable signal elements on Received Data.

5. Pin 22 - Ring Indicator
Direction: FROM data communication equipment (device to computer)

The ON condition of this circuit indicates that a ringing signal is being received on the communications channel.

The ON condition shall appear approximately coincident with the ON segment of the ringing cycle (during rings) on the communication channel.

The OFF condition shall be maintained during the OFF segment of the ringing cycle (between "rings") and at all other times when ringing is not being received. The operation of this circuit shall not be disabled by the OFF condition on Data Terminal Ready.

6. Pin 23 - Data Signal Rate Selector (DTE Source)
Direction: TO data communication equipment (computer to device)

Signals on this circuit are used to select between the two data signalling rates in the case of dual rate synchronous data sets or the two ranges of data signalling rates in the case of dual range non-synchronous data sets.

An ON condition shall select the higher data signalling rate or range of rates.

The rate of timing signals, if included in the interface, shall be controlled by this circuit as may be appropriate.

7. Pin 23 - Data Signal Rate Selector (DCE Source)
Direction: FROM data communication equipment (device to computer)

Signals on this circuit are used to select between the two data signalling rates in the case of dual rate synchronous data sets or the two ranges of data signalling rates in the case of dual range non-synchronous data sets.

An ON condition shall select the higher data signalling rate or range of rates.

The rate of timing signals, if included in the interface, shall be controlled by this circuit as may be appropriate.

8. Pin 24 - Transmitter Signal Element Timing (DTE Source)
   Direction: TO data communication equipment (computer to device)

   Signals on this circuit are used to provide the transmitting signal converter with signal element timing information.

   The ON to OFF transition shall nominally indicate the center of each signal element on the Transmitted Data. When the circuit is implemented in the DTE, the DTE shall normally provide timing information on this circuit whenever the DTE is in a POWER ON condition. It is permissible for the DTE to withhold timing information on this circuit for short periods, provided Request to Send is in the OFF condition. (For example, the temporary withholding of timing information may be necessary in performing maintenance tests within the DTE).

The following signals are typically a duplicate or secondary circuit.

1. Pin 12 - Secondary Received Line Signal Detector
   Direction: FROM data communication equipment (device to computer)

   This circuit is equivalent to Received Line Signal Detector except that it indicates the proper reception of the secondary channel line signal instead of indicating the proper reception of primary channel received line signal.

2. Pin 13 - Secondary Clear to Send
   Direction: FROM data communication equipment (device to computer)

   This circuit is equivalent to Clear to Send except that it indicates the availability of the secondary channel instead of indicating the availability of the primary channel. This circuit is not provided where the secondary channel is useable only as a circuit assurance or an interrupt channel.

3. Pin 14 - Secondary Transmitted Data
   Direction: TO data communication equipment (computer to device)

   This circuit is equivalent to Transmitted Data except that it is used to transmit data via the secondary channel.

4. Pin 16 - Secondary Received Data
   Direction: FROM data communication equipment (device to computer)

   This circuit is equivalent to Received Data except that it is used to receive data on the secondary channel.

5. Pin 17 - Secondary Request to Send
   Direction: TO data communication equipment (computer to device)

   This circuit is equivalent to Request to Send except that it requests the establishment of the secondary channel instead of requesting the establishment of the primary data channel.

```
1       =  00 00          .ORG    #0000
2                         .TITLE  TINY PILOT V1.0
3                   ;
4                   ;             1 8 0 2   T I N Y   P I L O T
5                   ;
6                   ;             ORIGINAL AUTHOR:  R. W. PETTY
7                   ;
8                   ;    REVISION HISTORY:
9                   ;
10                  ;    SEP./82 W. BOWDISH - REARRANGED CODE TO CONSOLIDATE FREE
11                  ;            T. HILL      SPACE AND MADE MINOR CODING CHANGES
12                  ;                         - ADDED EDIT MODE REPLACE LINE COMMAND
13                  ;                         - ADDED EDIT MODE COMMAND TO PRINT THE
14                  ;                           ADDRESS OF THE LAST USED BYTE IN THE
15                  ;                           TEXT BUFFER
16                  ;                         - ADDED PILOT CALL TO SCRT CALLABLE
17                  ;                           MACHINE LANGUAGE SUBROUTINES
18                  ;
19                  ;    TINY PILOT HAS 2 ENTRY POINTS ( COLDST AT #0000 AND WARMST AT
20                  ;    #0002 ). IF THE PROGRAM IS ENTERED AT 'COLDST', THEN THE SCRT
21                  ;    REGISTERS AND STACK POINTER ARE INITIALIZED AND THE DATA PAGE
22                  ;    AND TEXT BUFFER ARE INITIALIZED. IF THE PROGRAM IS ENTERED AT
23                  ;    'WARMST'  THE SCRT REGISTERS AND STACK POINTER ARE ASSUMED TO
24                  ;    BE SET UP AND THE CONTENTS OF THE TEXT BUFFER ARE PRESERVED.
25                  ;
26                  ;    *NOTE* TINY PILOT USES A MODIFIED SCRT CALL/RETURN TECHNIQUE
27                  ;           TO SAVES/RESTORES REGISTER-7 ACROSS A CALL AND RETURN
28                  ;
29 0000  30 16      COLDST: BR      PILINZ  ; COLD START ENTRY POINT
30 0002  30 33      WARMST: BR      PILWRM  ; WARM START ENTRY POINT
31                  ;
32                  ;         BRANCH VECTORS FOR EXTERNALLY DEFINED SUBROUTINES
33                  ;
34                  ;    THE FOLLOWING  TABLE CONTAINS LONG BRANCHES TO ROUTINES WHICH
35                  ;    MUST BE SUPPLIED BY THE USER. THESE ROUTINES ARE CALLED USING
36                  ;    THE SCRT CALL/RETURN TECHNIQUE. BEFORE USING TINY PILOT THESE
37                  ;    BRANCH VECTORS MUST BE MODIFIED TO POINT TO YOUR SUBROUTINES.
38                  ;
39 0004  C0 07 70   TTYOUT: LBR     OUTCHR  ; TERMINAL CHARACTER OUTPUT ROUTINE
40                                          ; ( CHARACTER PASSED IN D-REG. )
41 0007  C0 07 69   TTYINP: LBR     INPCHR  ; TERMINAL CHARACTER INPUT ROUTINE
42                                          ; (CHARACTER RETURNED IN D-REG. AND RF.HI)
43 000A  C0 00 F6   CRLF:   LBR     $CRLF   ; <CR><LF> OUTPUT ROUTINE
44 000D  C0 81 A4   CASOUT: LBR     #81A4   ; CASSETTE CHARACTER OUTPUT ROUTINE
45                                          ; ( CHARACTER PASSED IN D-REG. )
46 0010  C0 81 40   CASINP: LBR     #8140   ; CASSETTE CHARACTER INPUT ROUTINE
47                                          ; (CHARACTER RETURNED IN D-REG. AND RF.HI)
48 0013  C0 10 00   MONXIT: LBR     #1000   ; ENTRY POINT TO SYSTEM MONITOR ( EDIT
49                                          ; MODE "M" COMMAND EXITS VIA THIS LBR)
50                        .SLW
```

```
                    ;
                    ;              R E G I S T E R   U S E A G E
                    ;
                    ;       R0-R6   STANDARD
                    ;       R7      SAVE/RESTORE REGISTER
                    ;       R8      SCRATCH
                    ;       R9      GENERAL PURPOSE COUNTER
                    ;       RA      TEXT POINTER
                    ;       RB      LINE POINTER
                    ;       RC-RE   RESERVED FOR I/O AND PASSING PARAMETERS TO
                    ;               MACHINE LANGUAGE ROUTINES ( VIA S: COMMAND )
                    ;       RF      RF.HI HOLDS I/O ASCII CHAR. AND D-REG FOR SCRT
                    ;
                    ;    NOTE: NUMERIC VARIABLES ARE SINGLE BYTE ENTRIES IN THE VARIABLE
                    ;          PAGE WITH OFFSET EQUAL TO THEIR ASCII CODES. IE. VARIABLE
                    ;          A (#41) IS AT ADDRESS #XX41 AND VARIABLE Z (#5A) IS AT
                    ;          ADDRESS #XX5A
                    ;
                    ;
                    ;    FOR ASSEMBLY, THIS SYMBOL SHOULD BE REDEFINED
                    ;
=   00 20           TOPMEM: .EQL    #20     ; HIGHEST RAM PAGE AVAILABLE FOR TEXT BUFFER
                    ;
                    ;    ERROR CODES
                    ;
=   00 01           .ERDZR: .EQL    1       ; DIVISION BY ZERO
=   00 02           .ERCMD: .EQL    2       ; INVALID COMMAND
=   00 03           .ERFUL: .EQL    3       ; MEMORY FULL
=   00 04           .ERNNV: .EQL    4       ; BAD NUM. VAR. SYNTAX IN TYPE STATEMENT
=   00 05           .ERNTV: .EQL    5       ; CAN'T FIND (OR BAD SYNTAX) TEXT VAR. IN TYPE
=   00 06           .ERBNV: .EQL    6       ; BAD NUM. VAR. SYNTAX IN ASK STATEMENT
=   00 07           .ERBTV: .EQL    7       ; BAD TEXT VAR. SYNTAX IN ASK STATEMENT
=   00 08           .ERNRT: .EQL    8       ; UNDEFINED RETURN STATEMENT
=   00 09           .EREXP: .EQL    9       ; BAD EXPRESSION SYNTAX
=   00 0A           .ERNVR: .EQL    10      ; NO VAR. NAME IN COMPUTE STATEMENT
=   00 0B           .ERNEQ: .EQL    11      ; NO "=" SIGN IN COMPUTE STATEMENT
=   00 0C           .ERNLB: .EQL    12      ; CAN'T FIND LABEL OF JUMP OR USE
=   00 0D           .ERBLB: .EQL    13      ; BAD STATEMENT LABEL SYNTAX
=   00 0E           .ERUSR: .EQL    14      ; NO ADDR. OR SYNTAX ERROR IN "S" COMMAND
=   00 0F           .ERMXM: .EQL    15      ; ERROR IN MAX. MEMORY SPECIFICATION
                            .SLW
```

PILOT EDIT MODE COMMANDS

COMMAND	SYNTAX	FUNCTION
BEGIN	B	move text pointer to start of text buffer and display current (first) line
CLEAR	C	clear text buffer (ie. delete all program lines currently in the text buffer)
DOWN	Dnnn	move text pointer down 'nnn' lines and display new current line. If there were less than 'nnn' lines of text after the text pointer then the text pointer is positioned immediately after the last character of the last line.
END	E	move text pointer to end of text. The text pointer is left pointing to the byte past the last entered program byte.
HIGH	H	display the address (in hex) of the last byte used in the text buffer. This is useful if you are going to use a monitor cassette dump routine to write out the contents of the text buffer. Note that the text buffer starts at address #0900.
INSERT	I<text>	insert one line of <text> in front of the curre line.
KILL	Knnn	kill (delete) 'nnn' lines of text, starting at the current line, and display the new current line.
LOAD	L	load text from cassette. The text lines are appended to the text currentlyy in the text buffer.
MONITOR	M	return to the system monitor
PILOT	P	execute the pilot program
REPLACE	R<text>	replace the current line with the line <text>
STORE	S	store entire contents of the text buffer on cassette.
UP	Unnn	move text pointer up 'nnn' lines and display new current line. If there were less than 'nnn' lines of text above the text pointer, then the pointer is positioned at the start of the first line.
WRITE	Wnnn	write 'nnn' lines to the terminal starting with the current line. The position of the text pointer is not changed.
note:	nnnn	represents a number in the range 0 to 255
	<text>	represents a line of text up to 60 characters long

PILOT INSTRUCTIONS

COMMAND	SYNTAX	FUNCTION
ASK	A:	Request input from the terminal. The input is not saved in a variable but is available for matching (see the M: command).
	A:#v	request numeric input from terminal and store in numeric variable 'v'
	A:$v	request string data from terminal and store in string variable 'v'
COMPUTE	C:n=exp	compute the value of the expression 'exp' and assign the value to numeric variable 'n'.
END	E:	stop execution of program. This command may be placed anywhere in the program where execution should logically stop.
JUMP	J:exp	jump (branch) to statement with the same label number as the value of the expression 'exp'.
CONTROL	K:exp	output the ASCII equivalent of the value of 'exp' to the terminal. The control command may be used to output any ASCII character to the terminal. ex. K:13 outputs a carriage return to the terminal
MATCH	M:s1(,s2...)	compares the strings 's1' , 's2' ... to the response from the previous ask command. If a match is found the match flag is set 'yes' otherwise the match flag is set 'no'.
NO	cN:xxx	if the match flag is set to 'no' then the command c:xxx is executed. 'c' represents any pilot command and 'xxx' represents any text required by the command. for example: JN:123 will jump to the statement labeled %123 if the match flag is set to 'no'. If the match flag is set to 'yes' the jump command will not be executed and the next sequential statement will be executed.
RETURN	R:	Return from subroutine (see USE command)
SCRT CALL	S:!addr(,a1,a2,a3)	calls a machine language subroutine using the S.C.R.T. technique. 'addr' is the address of the machine language subroutine in hex. 'an' are optional arguments of the form #v or $v (ie. either numeric or string variables). The addresses of the arguments are passed in registers RC, RD, and RE.

TYPE	T:(text)(#v)($v)(;)	type text and/or variables on the terminal. Text and variable may be present on the command line in any order. A carriage return and line feed are output at the end of the line unless a semi-colon ';' is placed at the end of the line in which case the carriage-return/line-feed are suppressed.
USE	U:exp	call a pilot subroutine with a label number equal to the value of the expression 'exp'. Subroutines can be nested to a depth of 24.
EXAMINE	X:exp	sets match flag to 'yes' if 'exp' evaluates to 0 otherwise the match flag is set to 'no'
	X:v(<>=exp)	sets the match flag to 'yes' if the condition is true, otherwise the match flag is set to 'no'.
YES	cY:xxx	if the match flag is set to 'yes' then the command c:xxx is executed. 'c' represents any pilot command and 'xxx' represents any text required by the command. for example: JY:123 will jump to the statement labeled %123 if the match flag is set to 'yes'. If the match flag is set to 'no' the jump command will not be executed and the next sequential statement will be executed.
RANDOM Z	Z:exp	assigns a random number in the range 0 to 'exp'-1 to the numeric variable Z.

note:	exp	represents an expresion consisting of constants and/or numeric variables with operators + - * and /. Expressions are evaluated from left to right and must evalute to a value in the range 0 to 255.
	(xxx)	represents an optial part of a statement
	#v	represents a numeric variable
	$v	represents a string variable
	v	variable names are the upper case letters A to Z
	%nnn	line labels consist of a number in the range 0 to 255 preceeded by a percent sign. Statements need not be labeled but if they are the label must preceed the command. For example: %123 T:HI

0000	30	16	30	33	C0	07	70	C0	07	69	C0	00	F6	C0	81	A4
0010	C0	81	40	C0	10	00	F8	1C	A3	90	B3	D3	F8	08	B2	F8
0020	FF	A2	93	B4	B5	F8	9A	A4	F8	B0	A5	F8	0C	D4	00	04
0030	D4	00	CF	D4	00	C1	D4	00	D6	D4	00	E9	D4	01	19	00
0040	32	51	D4	01	21	F8	3F	D4	00	04	D4	00	0A	D4	01	5A
0050	00	F8	3E	D4	00	04	D4	01	6E	D4	01	01	93	B7	F8	67
0060	A7	4B	D4	01	D8	30	3C	42	02	2D	43	00	CF	44	02	47
0070	45	01	13	48	03	3D	49	02	6E	4B	03	4A	4C	02	B3	4D
0080	00	13	50	03	FB	52	02	68	53	02	94	55	02	E1	57	02
0090	CB	03	03	03	03	03	03	03	9F	D3	BF	E2	96	73	86	73
00A0	97	73	87	73	93	B6	83	A6	46	B3	46	A3	30	98	9F	D3
00B0	BF	E2	12	96	B3	86	A3	72	A7	72	B7	72	A6	F0	B6	30
00C0	AE	F8	08	BF	F8	5F	AF	F8	00	2F	5F	8F	3A	C7	D5	D4
00D0	00	D6	F8	03	5A	D5	F8	09	BA	F8	00	AA	F8	02	5A	1A
00E0	D5	46	32	E0	D4	00	04	30	E1	D4	00	0A	D4	00	E1	45
00F0	44	49	54	00	30	0A	F8	0D	D4	00	04	F8	0A	D4	00	04
0100	D5	D4	01	0C	38	1B	0B	FB	20	32	05	D5	F8	08	BB	F8
0110	61	AB	D5	D4	01	C0	03	03	D5	46	AF	F8	08	BF	0F	A9
0120	D5	F8	00	A8	89	B9	F8	64	D4	01	33	F8	0A	D4	01	33
0130	99	30	41	A9	99	B7	D4	01	47	89	3A	3F	88	32	46	18
0140	89	F9	30	D4	00	04	D5	89	52	32	57	F8	00	A9	97	38
0150	19	F7	33	50	F4	B9	D5	F8	01	38	46	52	F8	00	30	67
0160	AF	89	52	30	68	52	46	AF	F8	08	BF	F0	5F	D5	D4	01
0170	0C	F8	00	A7	D4	00	07	5B	FB	08	32	A8	FB	10	32	B4
0180	FB	15	32	B9	FB	04	32	94	17	87	FB	3E	32	B9	4B	D4
0190	00	04	30	74	F8	20	5B	17	1B	D4	00	04	87	FB	3E	32
01A0	B9	87	FA	03	32	74	30	94	87	32	74	27	2B	F8	08	D4
01B0	00	04	30	74	D4	00	0A	30	6E	F8	0D	5B	D4	00	0A	D5
01C0	46	B7	46	A7	EA	F8	03	F3	32	D5	87	F3	32	D5	97	F3
01D0	32	D7	1A	30	C5	F8	01	D5	52	07	FB	03	32	F1	47	F3
01E0	32	E6	17	17	30	D9	93	B8	F8	EC	A8	D8	47	B3	47	A3
01F0	D3	D4	01	5A	02	D5	D4	01	19	02	FE	52	F8	0F	F4	A8
0200	F8	08	B8	D5	D4	01	F6	9A	58	18	8A	58	D5	D4	01	F6
0210	48	BA	08	AA	D5	0A	FF	41	3B	22	0A	FF	5B	33	22	F8
0220	00	D5	F8	01	D5	4A	57	17	FB	03	3A	25	D5	D4	00	D6
0230	9A	B8	8A	A8	08	FB	03	32	43	FB	0E	32	43	48	D4	00
0240	04	30	34	D4	00	0A	D5	D4	03	23	32	30	D4	02	51	30
0250	30	4A	FB	03	32	5F	FB	0E	3A	51	29	89	3A	51	38	2A
0260	D5	D4	01	C0	03	03	30	04	F8	01	A9	D4	03	56	EA	9A
0270	B7	8A	A7	4A	FB	03	3A	73	9A	FB	20	32	FB	07	52	F8
0280	02	57	2A	72	73	FB	02	3A	82	17	02	57	4B	5A	1A	FB
0290	0D	3A	73	D5	D4	00	D6	4A	B7	F8	FF	A7	27	87	3A	9C
02A0	97	D4	00	0D	97	FB	03	32	B1	FB	0E	3A	97	F8	0A	30
02B0	98	2A	D5	D4	01	13	38	1A	9A	FB	20	32	FB	D4	00	10
02C0	FB	0A	32	B8	9F	5A	FB	03	3A	B7	D5	D4	03	23	32	E0
02D0	9A	B8	8A	A8	D4	02	34	48	FB	03	32	E0	29	89	3A	D4
02E0	D5	D4	03	23	32	F7	2A	0A	FB	02	32	F6	2A	0A	FB	0D
02F0	3A	E7	29	89	3A	E6	1A	D4	02	30	D5	2A	F8	03	5A	D4

0300	01	5A	03	D5	B9	FF	30	3B	20	99	FF	3A	33	20	99	FA
0310	0F	B9	89	FE	FE	52	89	F4	FE	52	99	F4	A9	F8	00	D5
0320	F8	01	D5	D4	01	0C	F8	00	A9	A7	38	17	4B	D4	03	04
0330	32	2B	87	3A	3B	99	FB	0D	3A	2C	19	89	D5	D4	01	13
0340	9A	D4	05	EC	8A	D4	05	EC	30	7E	D4	03	23	32	52	D4
0350	03	56	D4	02	30	D5	9A	B7	8A	A7	D4	02	51	D4	02	25
0360	97	BA	87	AA	D5	38	1A	0A	FB	0D	32	7E	FB	29	32	92
0370	FB	07	32	82	FB	18	32	81	0A	D4	00	04	30	66	D4	00
0380	0A	D5	1A	0A	D4	02	15	3A	C5	0A	D4	01	1A	D4	01	21
0390	30	66	1A	9A	B7	8A	A7	D4	03	B2	3A	CB	1A	0A	FB	0D
03A0	32	AC	FB	29	32	AC	0A	D4	00	04	30	9C	97	BA	87	AA
03B0	30	66	D4	01	13	1A	D4	01	C0	24	03	3A	C4	07	52	1A
03C0	0A	F3	3A	B5	D5	D4	01	5A	04	30	7E	D4	01	5A	05	30
03D0	7E	38	1A	0A	FB	20	32	D2	D4	02	15	3A	F1	4A	B7	D4
03E0	01	C0	3D	0D	3A	F6	1A	D4	06	05	3A	F0	97	D4	01	60
03F0	D5	D4	01	5A	0A	D5	D4	01	5A	0B	D5	D4	00	D6	D4	00
0400	C1	D4	02	04	D4	01	13	1A	F8	03	5A	D4	02	0D	1A	D4
0410	01	C0	3A	03	3A	51	D4	02	04	2A	0A	FB	20	32	19	0A
0420	FB	4E	32	5D	0A	FB	59	32	55	0A	B9	D4	02	0D	1A	93
0430	B7	F8	65	A7	99	D4	01	D8	D4	01	19	00	32	0B	D4	02
0440	0D	38	2A	0A	FB	02	32	4D	0A	FB	0D	3A	42	1A	D4	02
0450	30	D4	00	E9	D5	D4	01	19	01	32	0B	30	19	D4	01	19
0460	01	32	19	30	0B	41	04	90	43	03	D1	45	02	61	4A	05
0470	00	4B	05	5E	4D	05	68	52	05	4C	53	06	C5	54	03	65
0480	55	05	3D	58	06	84	5A	05	AB	03	03	03	03	03	03	03
0490	F8	3F	D4	00	04	D4	01	6E	D4	01	C0	24	0D	32	C2	D4
04A0	02	0D	D4	01	C0	23	0D	3A	B7	1A	D4	03	23	0A	D4	02
04B0	15	3A	B8	0A	D4	01	60	D5	D4	01	5A	06	D5	D4	01	5A
04C0	07	D5	1A	9A	B7	8A	A7	07	D4	02	15	3A	BD	07	B8	D4
04D0	03	B2	3A	EC	9A	B7	8A	A7	27	0A	FB	03	32	E5	FB	27
04E0	32	E5	1A	30	D9	D4	02	25	97	BA	87	AA	D4	01	13	F8
04F0	24	5A	1A	98	5A	1A	F8	03	5A	D4	01	01	D4	02	6E	D5
0500	D4	06	05	3A	26	89	B7	D4	00	D6	D4	01	C0	25	03	3A
0510	27	1A	0A	D4	03	04	3A	2C	D4	05	31	97	52	89	F3	3A
0520	0A	D4	02	04	F8	00	D5	D4	01	5A	0C	D5	D4	01	5A	0D
0530	D5	F8	00	A9	4A	D4	03	04	32	34	2A	2A	D5	D4	01	19
0540	02	FC	01	D4	01	65	02	D4	05	00	32	58	D4	01	19	02
0550	32	59	FF	01	D4	01	65	02	D5	D4	01	5A	08	D5	D4	06
0560	05	3A	67	89	D4	00	04	D5	F8	00	D4	01	65	01	38	1A
0570	0A	FB	0D	32	8F	D4	01	0C	9A	B7	8A	A7	D4	05	96	32
0580	8F	1B	0B	FB	0D	3A	7C	D4	01	C0	2C	0D	32	6F	D5	F8
0590	01	D4	01	65	01	D5	9B	B8	8B	A8	07	FB	0D	32	AA	07
05A0	FB	2C	32	AA	47	52	48	F3	32	9A	D5	D4	01	19	5F	3A
05B0	B4	4B	32	B1	B9	FA	01	52	99	FE	3B	BE	FC	01	F3	D4
05C0	01	65	5F	B7	D4	06	05	3A	D1	D4	01	47	99	D4	01	65
05D0	5A	D5	F8	00	A7	97	32	E9	FA	01	32	E1	87	52	89	F4
05E0	A7	89	FE	A9	97	F6	B7	30	D5	87	A9	D5	73	F6	F6	F6
05F0	F6	D4	05	F6	60	F0	FA	0F	FC	30	52	FD	39	02	CF	FC

0600	07	D4	00	04	D5	F8	00	B7	A7	38	1A	0A	FB	20	32	0A
0610	FB	2D	32	43	FB	31	32	43	FB	01	32	43	FB	03	32	43
0620	FB	15	32	0A	FB	06	32	48	FB	07	32	4B	FB	05	32	4E
0630	0A	D4	02	15	32	53	0A	D4	03	04	32	5A	D4	01	5A	09
0640	F8	01	D5	97	A9	F8	00	D5	F8	01	C8	F8	02	C8	F8	03
0650	A7	30	0A	0A	D4	01	1A	A9	30	5D	D4	05	31	87	32	6B
0660	27	87	32	71	27	87	32	77	27	30	7D	97	52	89	F4	30
0670	81	97	52	89	F5	30	81	D4	05	D2	89	30	81	D4	01	47
0680	89	B7	30	0A	D4	06	05	3A	B3	F8	00	A7	0A	FB	0D	32
0690	C0	FB	31	32	B8	FB	02	32	BA	1A	89	B7	D4	06	05	3A
06A0	B3	97	52	89	F5	32	AA	3B	BD	27	87	32	B1	F8	00	30
06B0	B7	F8	01	D4	01	65	01	D5	27	38	17	30	99	17	30	AA
06C0	89	32	B1	30	AD	D4	01	C0	21	0D	3A	FC	1A	D4	07	01
06D0	33	FC	B8	D4	07	01	33	FC	A8	D4	07	29	33	FC	32	F7
06E0	BC	8F	AC	D4	07	29	33	FC	32	F7	BD	8F	AD	D4	07	29
06F0	33	FC	32	F7	BE	8F	AE	98	B6	88	A6	D5	D4	01	5A	0E
0700	D5	D4	07	06	33	26	4A	FF	30	3B	26	FF	0A	3B	19	FF
0710	07	3B	26	FF	06	33	26	FC	06	FC	0A	52	8F	FE	FE	FE
0720	FE	F1	AF	FC	00	D5	FF	00	D5	D4	01	C0	2C	0D	3A	64
0730	1A	0A	FB	24	32	49	0A	FB	23	3A	26	1A	0A	D4	02	15
0740	3A	26	4A	AF	F8	08	BF	30	62	1A	0A	D4	02	15	3A	26
0750	9A	B7	8A	A7	D4	03	B2	3A	26	1A	9A	BF	8A	AF	97	BA
0760	87	AA	9F	C8	F8	00	FC	00	D5	3C	69	69	FA	7F	D5	C4
0770	F8	9F	AD	F8	08	BD	4D	BC	0D	AC	9F	FB	0C	32	A2	F8
0780	20	5C	9F	FB	08	32	BB	FB	05	32	C4	FB	07	32	99	9F
0790	FA	3F	5C	1C	9C	FB	E2	32	DE	F8	5F	5C	8C	5D	2D	9C
07A0	5D	D5	F8	FF	BF	F8	00	AF	AC	F8	02	5F	F8	E0	BF	BC
07B0	F8	20	5F	1F	9F	FB	E2	3A	B0	30	99	2C	9C	FB	DF	3A
07C0	99	1C	30	99	8C	FA	E0	AC	8C	FC	20	3B	D9	52	9C	7C
07D0	00	FB	E2	32	DC	FB	E2	BC	F0	AC	30	99	8C	C8	F8	E0
07E0	5D	2D	F8	E1	5D	1D	F8	E0	BC	BF	F8	20	AC	F8	00	AF
07F0	4C	5F	1F	9C	FB	E2	3A	F0	8F	AC	9F	BC	30	B0	00	00

```
OUTPUT <FF> K:12/
%10 T:PPPP     III   L        000     TTTTT
    T:PFP       I    L       0 0       T
    T:P   P     I    L      0   0      T
    U:20/
    T:P   P     I    L      0   0      T
    U:20/
    T:PPPP      I    L      0   0      T
    U:20/
    T:P         I    L      0   0      T
    U:20/
    T:P         I    L      0   0      T
    U:20/
    T:P        III  LLLLL    000       T
    U:20
    T:
    U:20
    J:10
```

```
%20 C:T=150        DELAY  SUBROUTINE
%30 C:T=T-1
    X:T=0
    JN:30
    R:
```

```
LDN RC #0C     C:A=12
STR R2 #52     C:B=82
OUT 4  #64     C:C=100
DEC R2 #22     C:D=34
SEP R5 #05     C:E=213

%10 A:#X
    X:X=0
    EY:
    S:!0841,#X
    J:10
```

```
%10 T:WHAT DAY IS IT ;
    A:$D
    M:MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY,SUNDAY,TODAY
    TN: $D IS NOT A DAY - TRY THURSDAY
    E:
```

```
    T:HI. WHAT'S YOUR NAME ;
    A:$N
    T:HI $N, LETS PLAY AN ADDING GAME
    T:I'LL PRINT 2 NUMBERS AND YOU TYPE THE SUM
%10 Z:10
    C:A=Z
    Z:11
    C:B=Z+1
    C:A+B
    T:WHAT IS #A + #B ;
    A:#D
    X:C=D
    JN:30
%20 T:THAT'S RIGHT $N. LETS TRY ANOTHER ONE.
    J:10
%30 T:OOPS, THATS NOT RIGHT. TRY AGAIN ;
    A:#D
    X:C=D
    JY:20
    T:THATS NOT RIGHT EITHER.
    T:THE CORRECT ANSWER TO #A + #B IS #C
    J:10
```

Infestation II - A Simulation Game in Quest Basic

- by P.B. Liescheski III, Dept. of Chem., U. of Texas, Austin, Texas 78712

INFESTATION II is a computer modeling algorithm which simulates the propagation of disease in a forest. It is similar to the LIFE algorithm except that it models disease propagation instead of growth. This program was inspired by an article written by Frank C. Hoppensteadt (1). The basic idea and rules were obtained from this article.

This program simulates disease propagation in a forest according to certain rules. Each tree which is loosely defined as a unit of vegetation, can exist in one of three states. These three states are green, infested and defoliated. The state of a tree for the next season or generation is determined by applying the four basic rules:

1) A green tree remains green as long as it is not infested.
2) An infested tree becomes defoliated in the next season.
3) A defoliated tree becomes green in the next season.
4) An infested tree can infest its eight nearest neighbors with a seventy percent chance.

These rules determine the manner in which the disease travels in the forest. It should be noted that Rule #4 introduces chance into the algorithm, so the rules are not deterministic in nature. As a result of this, the same initial forest pattern will produce different results on every run.

In order to give the program some color, it has been written with the Quest Gremlin color video board in mind. The graphics generated by the program is not at all sophisticated. A healthy tree is represented merely by a green square. An infested tree is displayed as a red square while a defoliated tree is a yellow square. Since the graphics is quite simple, it can be adapted for other video boards including black-and-white boards. This adaptation can be implemented by modifying the video memory parameters (line #13) and the graphics color codes (lines #22-24). For example, INFESTATION II can be adapted to the Solid State Music VB1B video board by the following modifications:

```
13 B=@F000: W=64
```

and:

```
22 G$="*": G=ASC(G$)
23 I$="O": I=ASC(I$)
24 D$="+": D=ASC(D$)
```

In this case, the VB1B video memory begins at address F000, so B=@F000. In the VB1B, 64 bytes of memory are alotted to a line; eventhough, only the first 32 are displayed on a conventional television screen, so W=64. Also a green tree is represented by an asterisk. An infested tree is displayed as an O, while a defoliated tree is a plus sign. If one is interested in a forest of different size, the value for N can be changed (line # 9). Using this as an example, the modification of INFESTATION II should pose no problems.

INFESTATION II has been written in Quest Super BASIC V3.0. To use this program, one must boot-up the 1802 system with the Quest Super BASIC interpreter. For proper operation, the Gremlin video board must be set for the Alphanumeric/Semigraphics-4 mode. After the BASIC program has been typed and executed, it will display the initial forest pattern as a N by N color grid (N=9). This grid is updated about every half-minute for each season (generation). The program will continue until the pestilence has past (if it does). With this, the message: "DISEASE HAS PAST" will be displayed on the screen.

The initial forest pattern is stored in DATA statements (lines #61-69). The initial pattern can be changed by modifying these DATA statements or by exchanging the READ with an INPUT statement (line #85). This will allow the operator to enter the initial pattern interactively. It should be stated once again that one need not change the initial pattern to get different results, since the rules of the game contain some element of chance. As the symbol table indicates, the letter G represents a healthy tree in the input string of the initial forest pattern. The letter I indicates an infested tree, while D is a defoliated tree.

The operation of INFESTATION II is quite simple. Basically, the initial forest pattern is read into matrix F. The program checks for errors in the initial forest pattern. Next the video screen is cleared (CLS) and flag T is checked to see whether the infestation has past. If not, then the contents of F are transferred to matrix L and matrix F is refreshed. The contents of L are displayed on the Gremlin video memory via the POKE command. With this, the four basic rules are applied to determine the state of the forest for the next season. After this, the process is repeated until the pestilence has past or until the 1802 system is unplugged. Finally to simplify the logic, the boundaries of the forest are avoided. This is the reason for dimensioning the matrices L and F as N+2 by N+2 (where N=9).

INFESTATION II is based upon a simple modeling algorithm. Its major flaw is its speed. Since the algorithm is simple, the task of writing it in a faster, low-level language, such as machine code, should not be too difficult. In a faster language, the program can operate faster with larger forests. INFESTATION II is meant to demonstrate the basic principles behind population simulation. Like LIFE, it will also produce very interesting and colorful patterns.

Reference

[1]Hoppensteadt, Frank C. "Mathematical Aspects of Population Biology." In Mathematics Today: Twelve Informal Essays, pp. 297-320. Edited by Lynn Arthur Steen. New York: Springer-Verlag, 1978.

```
1    REM **
2    REM ** INFESTATION - VERSION 2
3    REM **
4    REM ** PHILLIP B. LIESCHESKI III ** 12/17/82
5    REM **    GREMLIN VIDEO BOARD IS USED FOR COLOR GRAPHICS
6    DEFINT Z
7    REM *
8    REM * FOREST GRID SIZE AND INITIAL PARAMETRERS
9    N=9: T=1
10   DIM L(N+2,N+2),F(N+2,N+2)
11   REM *
12   REM * VIDEO MEMORY PARAMETERS - STARTING ADDRESS, SCREEN WIDTH
13   B=@E000: W=32
14   V=B+((16-N)/2+1)*W
20   REM *
21   REM * GREMLIN GRAPHICS COLOR CODE
22   G=#8F
23   I=#BF
24   D=#9F
30   REM **
35   REM ** ENTER INITIAL FOREST PATTERN
38   REM *
40   REM * SYMBOL TABLE:
45   REM *   G-GREEN=LIVING TREE
50   REM *   I-RED=INFESTED TREE
55   REM *   D-YELLOW=DEFOLIATED TREE
60   REM *
61   DATA ""GGGGGGGGG""
62   DATA ""GGGGGGGGG""
63   DATA ""GGGGGGGGG""
64   DATA ""GGGGGGGGG""
65   DATA ""GGGGIGGGG""
66   DATA ""GGGGGGGGG""
67   DATA ""GGGGGGGGG""
68   DATA ""GGGGGGGGG""
69   DATA ""GGGGGGGGG""
70   REM *
80   FOR R=2 TO N+1
85        READ L$
90        FOR C=2 TO N+1
99             REM *
100            REM * REQUEST EACH TREE STATE
110            A$=MID$(L$,C-1,1)
120            IF A$=""I"" THEN F(C,R)=I: GO TO 160
130            IF A$=""D"" THEN F(C,R)=D: GO TO 160
140            IF A$=""G"" THEN F(C,R)=G: GO TO 160
150            PRINT ""ERROR IN FOREST PATTERN"": END
160       NEXT C
170  NEXT R
180  REM **
181  REM ** CLEAR THE SCREEN AND BEGIN
190  CLS
208  REM **
209  REM ** CHECK IF INFESTATION IS GONE
210  IF T<>0 GO TO 217
211  REM **
212  REM ** DISEASE HAS PAST
213  PRINT ""DISEASE HAS PAST"": END
217  T=0
220  REM **
221  REM ** DISPLAY FOREST PATTERN
```

```
230  FOR R=2 TO N+1
240       FOR C=2 TO N+1
241            REM *
242            REM * TRANSFER NEW GENERATION PATTERN
245            L(C,R)=F(C,R)
248            REM *
249            REM * REFRESH TEMPORARY FOREST MATRIX
250            F(C,R)=G
258            REM *
259            REM * CALCULATE VIDEO MEMORY ADDRESS
260            X=C-1+(32-N)/2
270            Y=V + (R-2)*W
278            REM *
279            REM * PUSH COLOR CODE INTO VIDEO MEMORY
280            POKE(X+Y,L(C,R))
290       NEXT C
300  NEXT R
310  REM **
311  REM ** DETERMINE PATTERN FOR NEXT GENERATION
312  REM ** ...BOUNDARIES OF FOREST ARE AVIODED TO
313  REM **   SIMPLIFY THE LOGIC...
320  FOR R=2 TO N+1
330       FOR C=2 TO N+1
338            REM *
339            REM * IF DEFOLIATED, MAKE GREEN
340            IF L(C,R)=D THEN F(C,R)=G: GO TO 460
348            REM *
349            REM * IF NOT INFESTED, LEAVE IT ALONE
350            IF L(C,R)<>I GO TO 460
355            T=1
358            REM *
359            REM * IF INFESTED, DEFOLIATE...
360            F(C,R)=D
369            REM *
370            REM * ... AND INFEST NEIGHBORS WITH 70% CHANCE
375            P=70: H=100
380            IF L(C+1,R+1)=G IF RND(H)<P THEN F(C+1,R+1)=I
390            IF L(C+1,R-1)=G IF RND(H)<P THEN F(C+1,R-1)=I
400            IF L(C-1,R+1)=G IF RND(H)<P THEN F(C-1,R+1)=I
410            IF L(C-1,R-1)=G IF RND(H)<P THEN F(C-1,R-1)=I
420            IF L(C,R+1)=G IF RND(H)<P THEN F(C,R+1)=I
430            IF L(C,R-1)=G IF RND(H)<P THEN F(C,R-1)=I
440            IF L(C+1,R)=G IF RND(H)<P THEN F(C+1,R)=I
450            IF L(C-1,R)=G IF RND(H)<P THEN F(C-1,R)=I
460       NEXT C
470  NEXT R
480  GO TO 210
999  END
```

A SCANNING HEX KEYBOARD ENCODER

-by Grant Thomson, 3793 The Boulevard, Westmount, Quebec H3Y 1T3

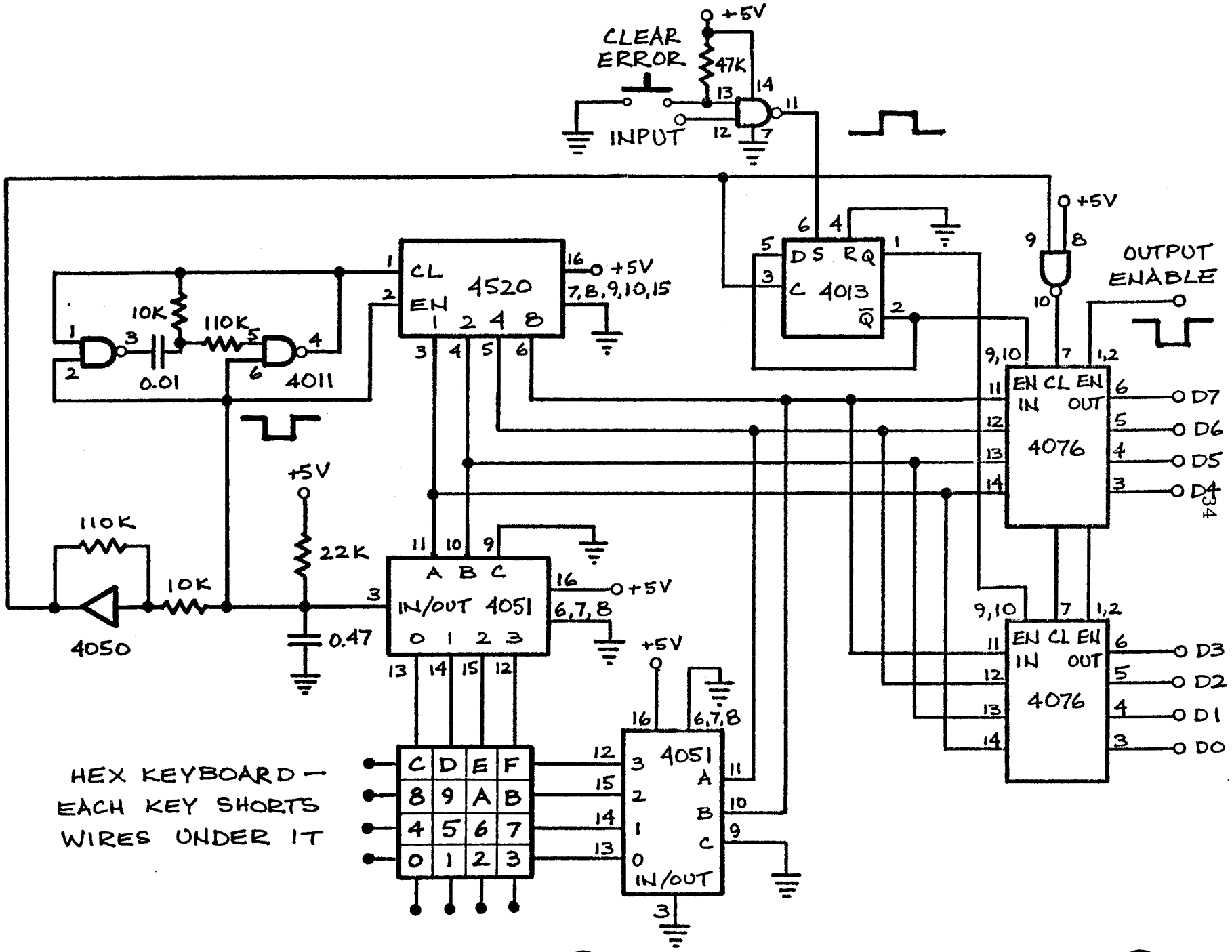
Here is a scanning hex keyboard circuit that I think is elegant and economical. It is adapted from 'A Scanning ASCII Keyboard Encoder' from Don Lancaster's CMOS Cookbook. Only four IC's make up the scanning and encoding circuitry. Two four-bit latches, a flip-flop, and a Schmitt trigger (here I used a spare 4050 with feedback) are also needed. Here's how it works.

The two 4011 gates form a 4 kHz clock that is gated. If the clock is allowed to run, the 4520 binary counter continuously cycles through its counts.

The top two bits of the counter are routed to a 4051 which is used as a one-of-four selector that sequentially connects rows of the keyboard to ground. Meanwhile, the lower two bits of the counter, which are cycling faster, are routed to a second 4051 which is monitoring sequential columns of characters. When a key is pressed, the output through both selectors goes to ground and stops the gated oscillator and holds the count. The output of the counter is the binary representation of the key that is pressed. A Schmitt trigger provides a keypressed output which is used to latch the binary data and to clock a flip-flop that alternately selects high and low four-bit latches.(The flip-flop is reset when either the INPUT or CLEAR ERROR key is pressed.)

When the key is released, scanning resumes until a new key is pressed. If a second key is pressed while one is already down, nothing happens right away. But, when the first key is released, scanning resumes and then stops at the second key location. This is called "two-key rollover" and helps to minimize errors.

The keyboard is fully debounced by the 0.47 μF capacitor and the Schmitt trigger. I used a spare 4050 gate and two resistors to make a Schmitt trigger. I found the values of the resistors and the capacitor were quite critical for proper gating of the oscillator. It may be necessary to experiment with the values in your circuit.


CLEAR ERROR
+5V
47K
INPUT
4520
CL
EN
+5V
7,8,9,10,15
10K
110K
0.01
4011
4013
DS RQ
C
Q
OUTPUT ENABLE
EN CL EN
IN OUT
4076
D7
D6
D5
D4
D3
D2
D1
D0
110K
10K
4050
22K
0.47
A B C
IN/OUT 4051
6,7,8
HEX KEYBOARD — EACH KEY SHORTS WIRES UNDER IT
C D E F
8 9 A B
4 5 6 7
0 1 2 3

Notes on Netronics Tiny Basic

- by O. Hoheisel, Herm. - Bossd. - Str. 33, 2190 Cuxhaven 1, W. Germany

While experimenting with Netronics Tiny Basic, I found the following very interesting capability. Type or load a program that at some time stops and enters the command mode. RUN it to test it and then execute a POKE 1897,167 command. The program will run as if you entered a RUN command, and when it reaches an END statement (or an error occurs), it will automatically start at the beginning again. This can be very useful if only a small amount of memory is available, but you want to write large programs, because now you can store several parts of a program on cassette, and at the end of a part the following lines will load and start the next part.

```
30000 PRINT "PRESS 'PLAY' ON TAPE DECK, PLEASE"
30010 REM TURN ON AUTO-RUN
30020 POKE 1897,167
30030 REM NOW LOAD NEXT PART
30040 LOAD
```

With cassette motor control, this could be done completely automatically. By the way, if you want to turn auto-run off again, use a POKE 1897,39 command or use your monitor to write a (hex) 27 into (hex) address 0769. This is also a great tool to make your programs fail-safe for all users since error stops and even a break will just start the program over.

While looking at Tiny's command table, I found that Netronics' version has an additional function not mentioned in any of their manuals. The syntax is FLG(X), and it will return the logic value (0 or 1) of the EF-flag determined by X. This X can be any valid expression from 0 to 4, but FLG(0) will always return -256, because there is no such flag. The following example will wait for the user to press the 'I' switch to continue.

```
30000 PRINT "PRESS 'I' TO CONTINUE"
30010 IF FLG(4)=0 GOTO 30010
30020 REM CONTINUE HERE
```

ACE CPU Board

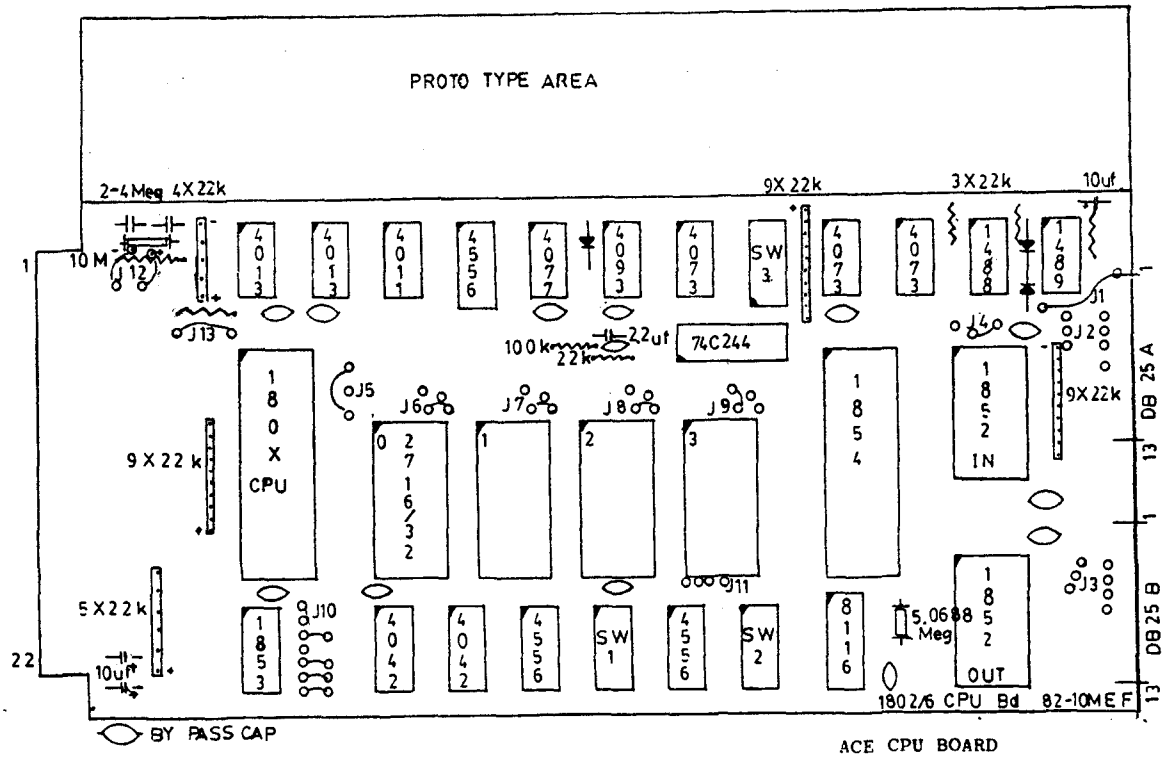
Size: 6" x 9.5"

Function: to provide
- a system micro computer (1802 -04 -05 -06)
- control logic, power on reset, fully decoded INTERUPT, DMAIN and DMAOUT.
- selectable BOOT to any PAGE ADDRESS
- 4 JEDEC EPROM/RAM sockets, with DUAL ADDRESS decoding for 2 locations or sizes of memory.
- INPORT and OUTPORT
- UART with selectable baud rate
- RS 232 C with 2 Db 25 connectors
- extensive prototype area (1.5" x 8.5")

CPU Board is designed to be a system or standalone micro controller board.

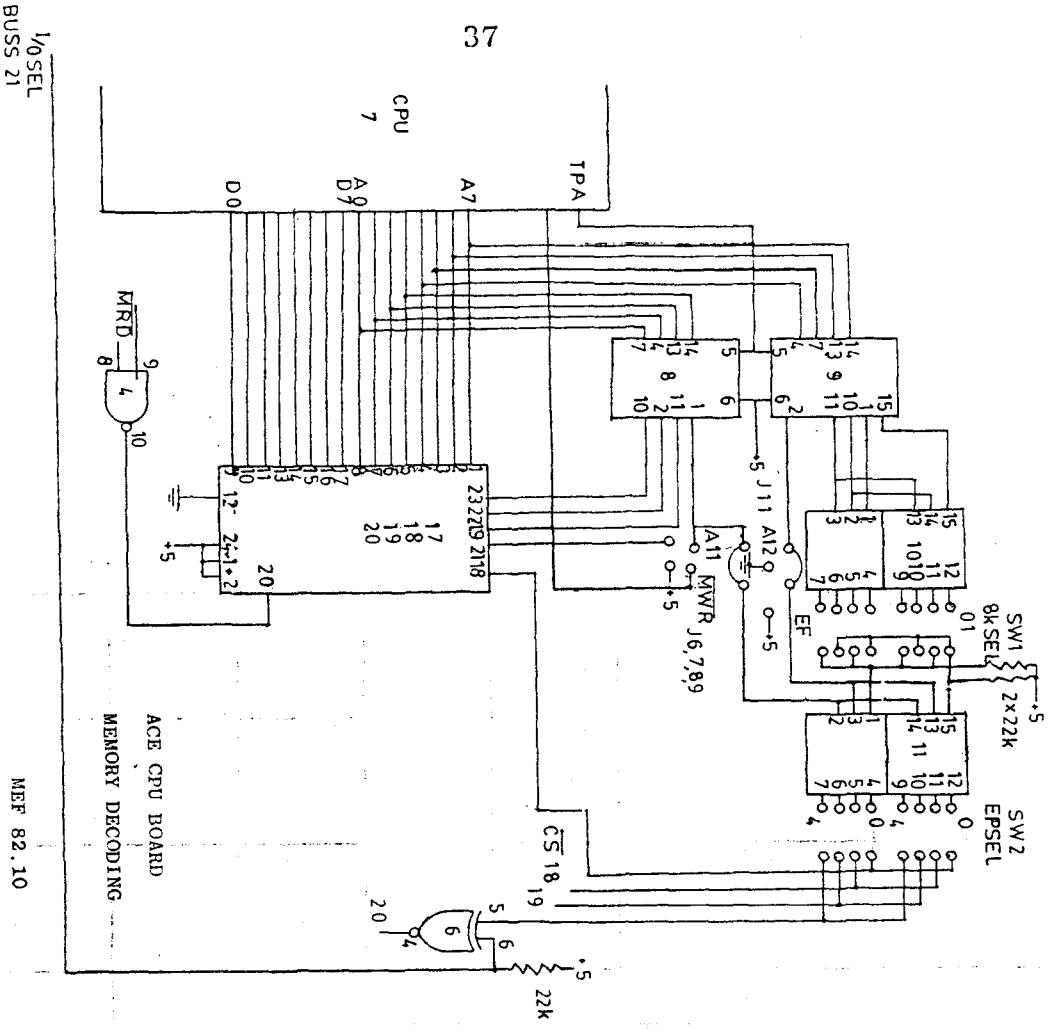
Power: +5 v. Gnd. ±12v for RS 232C circuit.

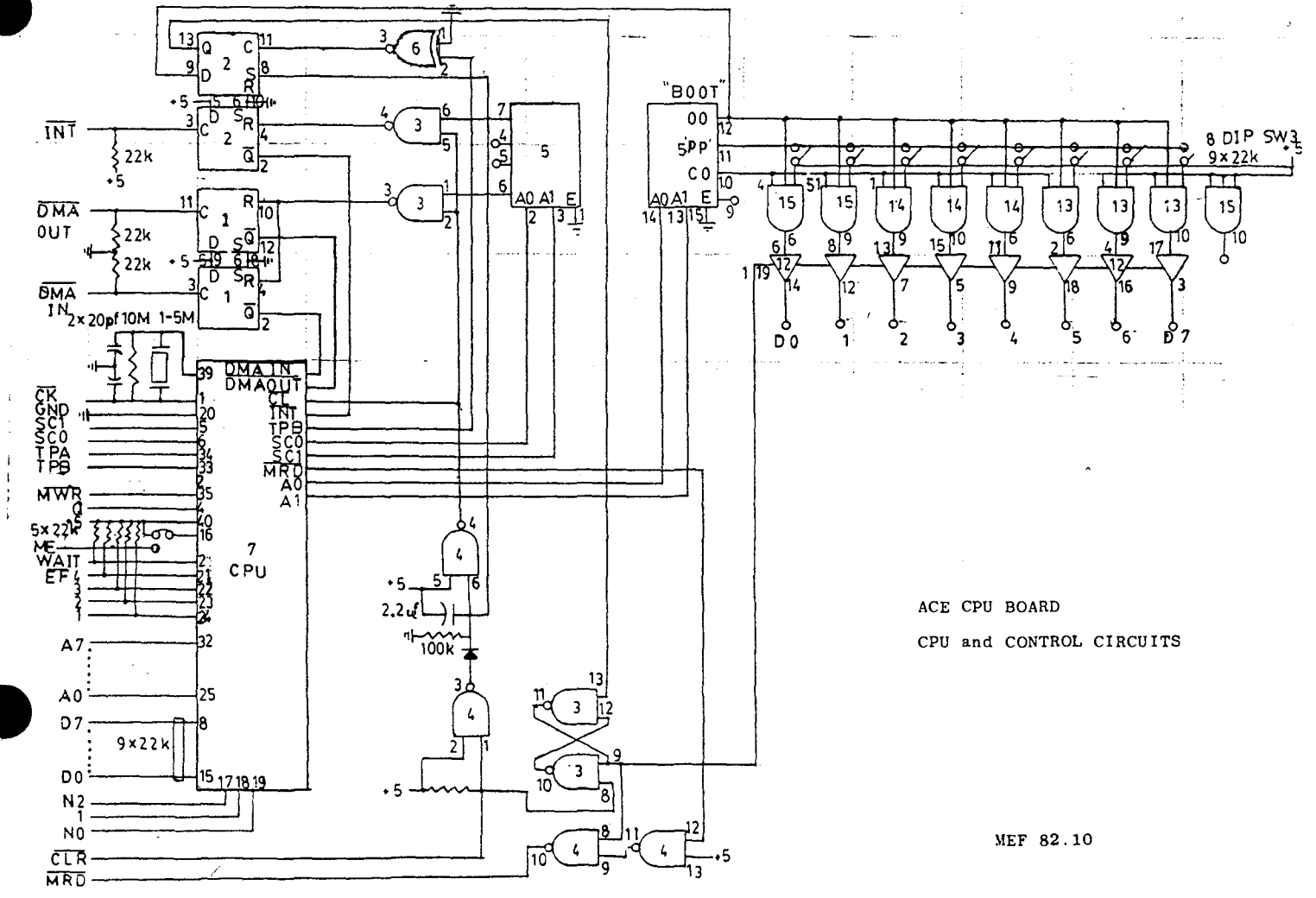
Documemtation: assembly and test instructions, software for UART.



ACE CPU BOARD

PARTS PLACEMENT





ACE CPU BOARD

CPU and CONTROL CIRCUITS

MEF 82.10

ACE CPU BOARD PARTS LIST

CPU Control & Boot

IC #	
1	4013
2	4013
3	4011
4	4093
5	4556
6	4077
7	1802/4/5/6
12	74C244
13	4073
14	4073
15	4073

Resistors

2 - 9 x 22K SIP or 18-22K 1/4 watt

10 - 22K 1/4 watt 5%

1 - 100K 1/4 watt

1 - 10 MEG 1/4 watt

Capacitors

1 - 2.2 mf tantalum

2 - 20 p.f. ceramic

3 - 10 mf tantalum (buss filters)

6 - 0.001 mf ceramic (bypass caps.)

Diode

1 - IN914

Crystal

1 - 1.0 meg to 5.0 meg

Switch

1 - 8 position dip

Memory

IC #	
8	4042
9	4042
10	4556
11	4556
17	EPROM/RAM
18	"
19	"
20	"

Resistors

2 22K 1/4 watt 5%

Switch

2 8 position dip

Port, UART, RS232C

IC #	
16	1853
21	1854
22	SMC-COM 8116 (P)
23	1852
24	1852
25	1488
26	1489

Resistors

4 22K 1/4 watt 5%

1 9 x 22K SIP or 9-22K 1/4 watt

Diodes

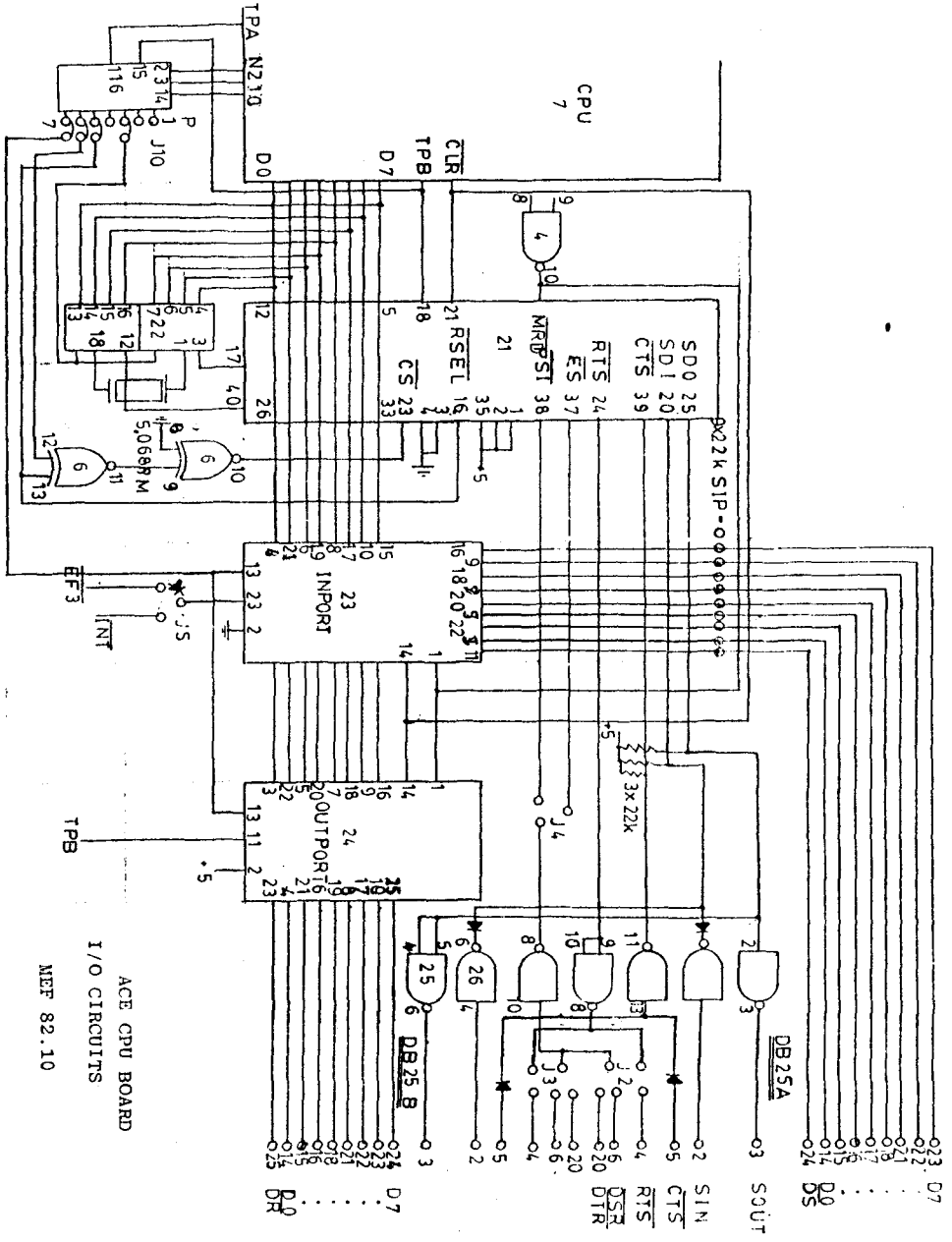
7 - IN 914

Crystal

1 5.0688 meg.

Connectors

2 Db25 Female
(wire solder type)



ACE CPU BOARD
I/O CIRCUITS
MEF 82.10

CLUB COMMUNIQUE

NAME: _____ DATE: _____

PRODUCT ORDER	QUANTITY	UNIT PRICE	TOTAL
CPU Board	_____	$40.00	_____
Backplane and I/O Board, Ver. 2	_____	40.00	_____
Front Panel (with EPROM Burner, Clock)	_____	35.00	_____
I/O Adapter for Backplane, Ver. 1	_____	20.00	_____
64K Dynamic (4116) Board	_____	50.00	_____
EPROM (2716/32) Board	_____	40.00	_____
Kluge (wire wrap) Board	_____	25.00	_____
8" Disk Controller Board	_____	40.00	_____
Netronics - Ace Adapter Board	_____	25.00	_____
Netronics - Quest Adapter Board	_____	20.00	_____
DMA Adapter Board (ELF II)	_____	3.00	_____
VDU Board	_____	40.00	_____

Software

Fig FORTH - Netronics Cassette format (6K)	_____	$10.00	_____
Tiny Pilot - Netronics Cassette format (2K)	_____	$10.00	_____

Back Issues

"Defacto" Year 1 - 3 (Edited)	_____	$20.00	_____
Year 4 Reprint	_____	10.00	_____
Year 5 Reprint	_____	10.00	_____

Membership

Current Year - Sept. '82 - Aug. '83
includes 6 issues of Ipso Facto

Canadian	_____	$20.00 Cdn.	_____
American	_____	20.00 U.S.	_____
Overseas	_____	25.00 U.S.	_____

PRICE NOTE

Prices listed are in local funds. Americans and Overseas pay in U.S. Funds, Canadians in Canadian Funds. Overseas orders: for all items add $10.00 for air mail postage. Please use money orders or bank draft for prompt shipment. Personal cheques require up to six weeks for bank clearance prior to shipping orders.

SALE POLICY

We guarantee that all our products work in an A.C.E. configuration microcomputer. We will endeavour to assist in custom applications, but assume no liability for such use. Orders will be shipped as promptly as payment is guaranteed.

NAME: _____

MAILING ADDRESS: _____

_____

_____

_____

PHONE NO.: _____

Note: Ensure mailing address is correct, complete and printed.
Please ensure payment is enclosed.

ASSOCIATION OF COMPUTER-CHIP EXPERIMENTERS
c/o M.E. FRANKLIN
690 LAURIER AVENUE,
MILTON, ONTARIO
L9T 4R5
