IPSO FACTO

ASSOCIATION OF COMPUTER EXPERIMENTERS

Issue #13
September, 1979

TABLE OF CONTENTS PAGE

>

Information furnished by IPSO FACTO is believed to be accurate and reliable. However, no responsibility is assumed by IPSO FACTO or the Association of Computer Experimenters for its use; nor for any infringements of patents or other rights of third parties which may result from its use.

Send Newsletter correspondence to: Bernie Murphy
102 McCrany Street,
Oakville, Ontario
Canada L6H 1H6

1979-1980 ACE EXECUTIVE COMMITTEE

PRESIDENT	Edward Fleet	5 Wells St. Toronto, ONT., M5R 1N8 416-925-5740
PAST PRESIDENT	Ken Bevis	220 Cherry Post Rd. Toronto, ONT., L5A 1H9 416-277-2495
TREASURER	Mike Franklin	24 Duby Rd, Acton, ONT., L7J 2P1 519-853-3421
ASSOCIATE EDITOR	Earle Laycock	2772 Hollington Cres., Mississauga, ONT, L5K 1E7 416-823-1345
ASSOCIATE EDITOR	Vic Kushnir	2640 Barnstone Cr., Mississauga, ONT., L5K 2C1 416-822-6505
ASSOCIATE EDITOR	Bob Silcox	562 Forestwood Cr., Burlington, ONT., L7L 4K3 416-681-2848
CONSULTING EDITOR	Bernie Murphy	102 McCraney St., Oakville, ONT., L6H 1H6 416-845-1630
DRAFTING	J. Myszkowski	99 Augusta St., Hamilton, ONT. 416-529-0250
MEMBERSHIP CO-ORDINATOR	Blair Gerrish	2110 Prospect St., Apt. 3, Burlington, Ont., L7R 1Y9 416-634-0268
TRAINING CO-ORDINATOR	Rod Dore	660 Oxford Rd., Unit 32, Burlington, ONT., L7N 1Y8 416-681-2456
HARDWARE CO-ORDINATOR	Fred Feaver	105 Townsend Ave., Burlington, ONT., L7T 1Y8 416-637-2513
PROGRAM CO-ORDINATOR	Fred Pluthero	1013 Upper Wellington St., Hamilton, ONT., L9A 3S4 416-389-4070
PUBLISHING COMMITTEE	Dennis Mildon	44 Wildewood Ave., Hamilton, ONT., L8T 1X3 416-385-0798
	John Hanson	955 Harvey Place, Burlington, ONT., L7T 3E9 416-637-1076

EDITORIAL COMMENTS

Since this is the first issue of the new season, there are a number of points to cover.

First, as you may have noticed on the list of the new executive, the position of 'editor' has changed to consist of a group of 'associate editors', it being hoped that in this way we will be able to maintain the high quality of work provided by our predecessors, while at the same time dividing up the fairly large amount of work involved in this undertaking.

A second point relates to the physical appearance and format of the newsletter. We have been able to make use of a word-processing system in the preparation of some of the articles in this issue (some of issue #12 was also prepared this way), and this should not only contribute to a more professional appearance, but more importantly, the newsletter will be easier to read.

One aspect of this appearance change is the use of a new logo on the cover page of the newletter. We would like to thank Allan Jackson for sending it in. Although too late for the T-shirt contest, it seemed a shame to waste such an excellent effort. To quote the designer: "The logo is stacked like an indigenous Canadian totem pole, and reads almost the same upside down".

ARTICLE SUBMISSIONS

We can always use lots of articles, whether software or hardware oriented, and whether they are on a highly technical or fairly elementary level. If anything, it is the elementary-level articles which are in the shortest supply, and it is often these which are of the most use to the largest number of readers. Even though a lot of our members are highly skilled technically, such skills tend to be fairly specialized, and so, for instance, a real hardware 'whiz' may find good instructional articles on machine language, or assembly language to be quite useful. Similarly, those with good programming backgrounds will appreciate good explanatory articles about hardware subject-matter.

We appreciate getting whatever you can send in, in whatever form you can manage to write it up. However, you can make our job much easier (and are likely to see your article in print at an earlier point) if you could manage to send articles to us in camera-ready form. By 'camera-ready' we mean typed, single-space, with a reasonably dark ribbon. Diagrams should be as large as possible --- we can always photo-reduce them. Again, if you do not have the facilities to type up submissions, then by all means send us what you have in whatever form is easiest for you.

Enough for now --- we look forward, as you do, to another good year for A.C.E., and hope to keep providing the best 1802 newsletter in the universe.

>

NOTE: This issue is being sent only to those who have renewed their membership for the coming year. Therefore, receipt of this issue will constitute acknowledgment of your renewal.

>

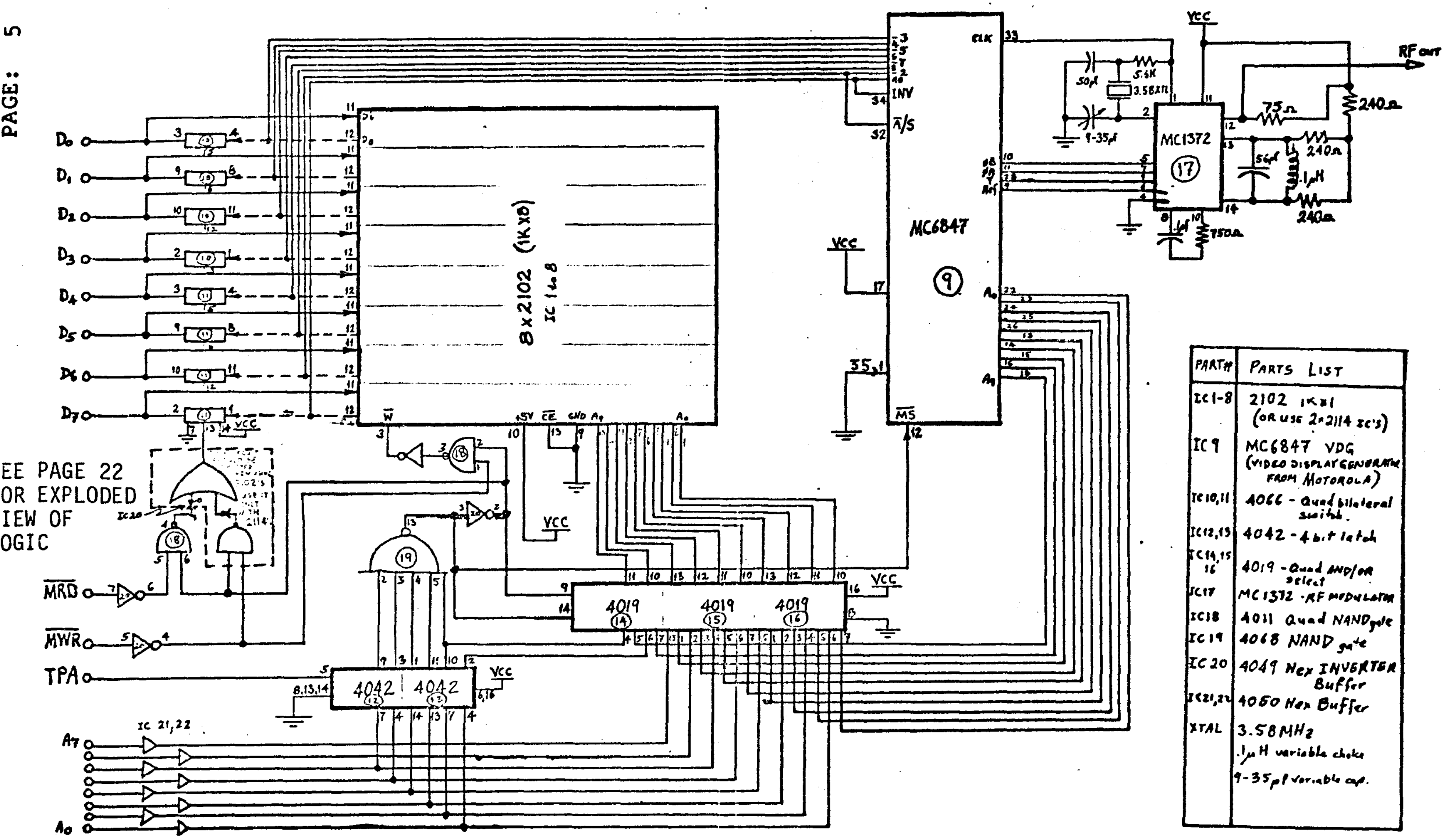
1802 FULL COLOUR DISPLAY

by: John Myszkowski
99 Augusta St
HAMILTON, Ont., Canada

Are you one of the unfortunate people that built himself (or even herself) an affordable microcomputer (namely the ELF, SuperELF, ELF II, the TEC-1802 or even the VIP) and then realized that you cannot afford a decent alphanumeric display for it? Well, now you can rejoice, because finally everybody is making ASCII Video displays for the ELFs and VIPs, and even TECs will soon have one. That is just great and fantastic--but expensive. "So I'll make my own" you say, but can't really decide which circuit to use (there are quite a few of them). There were a couple of good ones in IPSO FACTO; but you prefer something even simpler than those two. Well again, you're in luck. Thanks to growing (or is it shrinking?) technology, we finally have an IC that does practically everything that is expected of most video terminals. This chip is called the MCM6847 VDG ("VIDEO DISPLAY GENERATOR" for short) and is made by Motorola Semiconductor Division. It can display 32 characters in 16 lines in the alphanumeric mode, or up to 256 (horizontal) by 192 (vertical) dots in full graphic mode (using up to 6K bytes of refresh memory), or it can mix ASCII characters and graphic dots at the same time and on the same screen. It can do all that, plus it can display up to eight colours. The 6847 chip can be obtained through any distributor that handles Motorola parts, for about $20. Motorola is also selling an RF modulator chip which is made especially for the 6847 (but can be used with other video display generators) and contains an on chip oscillator. This chip is the MC1372 COLOUR TV VIDEO MODULATOR, it costs about $3. The VDG(6847) was designed with the MC6800 microprocessor in mind, but it is actually very easy to interface to other systems.

To make it work with the 1802 microprocessor only a handfull of IC's are needed. The "VDU for the 1802" is a complete video display board for the 1802 based systems and it uses the 6847 chip. The complete minimum system (32x16 lines ASCII and 128H x 64V dots graphics) uses only sixteen chips and that includes all the buffers and refresh memory (which is like an extra 1K bytes of fully accessible memory). This refresh memory can be expanded up to 6K, to give the board full graphics capability of 256x192 dots. This board can do graphics that will beat the 1861 chip graphics by a mile because of the 8 colours and 4 luminance levels of each pixel not to mention the ASCII display.

To begin with, the VDG IC addresses up to 6K of refresh memory, which also has to be accessed by the microprocessor to deposit or retrieve data. To accomplish this we have to do a few things. First, decode the address buss, so that the CPU will be able to access at least 512 bytes (for the 32x16 ASCII display) of memory, and possibly more if it is desirable. Second, connect the VDG address lines and the CPU address bus to the memory address, in such a way that neither one will interfere with the other when both are working at the same time. This is done by "multiplexing" the two address busses. To allow the CPU to access the refresh memory, the $\overline{\text{MS}}$ (memory select) line on the VDG is pulled low, which in turn forces



SEE PAGE 22 FOR EXPLODED VIEW OF LOGIC

PART#	PARTS LIST
IC1-8	2102 1Kx1 (OR USE 2x2114 IC's)
IC 9	MC6847 VDG (VIDEO DISPLAY GENERATOR FROM MOTOROLA)
IC 10,11	4066 - Quad bilateral switch.
IC12,13	4042 - 4 bit latch
IC 14,15 16	4019 - Quad AND/OR select
IC17	MC1372 - RF MODULATOR
IC18	4011 Quad NAND gate
IC 19	4068 NAND gate
IC 20	4049 Hex INVERTER Buffer
IC21,22	4050 Hex Buffer
XTAL	3.58MHz
	.1μH variable choke
	9-35pf variable cap.

SCHOOL	DR BY - JOHN MYSZKOWSKI	FORM-
TITLE VDU- FOR THE 1802	DATE - 4 MAY 1979	CHK-
	SCALE- Reduce to 8"x12"	DWG NO- [illegible]

the VDG address lines into a high impedance state. This task is taken care of by a couple of gates that decode the valid memory address and switch the memory address from the VDG buss to the CPU buss. The CPU address buss is buffered or turned off by three 4Ø19 IC's.

The data handling is just as simple as the address switching; logic gates detect valid address and switch the data buss through a pair of 4Ø 4Ø66's (4Ø66's are bi-directional switches). The direction of the data flow is controlled by the CPU R/W lines ($\overline{\text{MRD}}$ & $\overline{\text{MWR}}$) and the memory block address decoding.

The "VDU for the 18Ø2" uses 2114's for the display refresh memory. "Why not use 21Ø2's, they are cheaper and easier to get?" you might ask. The reason is simple. The 2114 memory has bi-directional data lines, this makes interfacing much easier. It also is equivalent to four 21Ø2's and therefore saves power and space. To the CPU, the board looks just like ordinary RAM, therefore it can be addressed just like an other memory location. With an average good TV receiver (preferably transistorized) the display looks very good and sharp. The characters are the standard 5x7 matrix type. The graphics are colour squares, usually in a 64x32 matrix when used simultaneously with the alpha characters. In the graphic, semigraphic, or alphanumeric modes, each pixel on the screen can be accessed without being worried about timing and DMA cycles like with the 1861 video chip. Therefore, all the games that can be displayed by this VDU chip will be very easy to make up and I'm sure just as colourful.

The schematic diagram shows 21Ø2's used as memory refresh. In order to help you build your own "VDU for the 18Ø2" as fast as you can, lay your hands on all the necessary parts. This article contains hints on implementing either kind of memory chips. When using 2114's, you will have to include the OR-ed ($\overline{\text{MRD}}$ + $\overline{\text{MWR}}$) memory select circuit which is enclosed within the dotted lines. Since the 2114 IC's have four bi-directional data lines, it is only necessary to connect dotted data lines to the respective data I/O pins on the 2114's. As you might have already noticed, the 21Ø2 IC's are much more readily available and you might even have enough of them in your parts cabinet ("junk box"), so you probably will want to use them up (might as well, right?). If this is what you want to do, then just hook up the circuit as shown on the diagram, leaving out only the OR-ed memory block selector IC (#19) as shown, will decode the display at ØCØØ; but, the address can be changed at will just by choosing the Q or $\overline{Q}$ outputs of the 4Ø42's.

This is a rather simple project if you have the p.c. board already made up; so, if I get enough interested people, then the boards will be made available at cost to all interested parties. A whole kit might even be put together if necessary. If enough interested people write in quickly then I will continue the discussion on the 6847 project backed with some ASCII graphics-display input routines and maybe some interesting games for use with the "VDU for the 18Ø2" project. Write to either the editor or to me directly (preferably me) ((YES YOU-ED.)). If you have any ideas about changes or improvements, then by all means share them with others. This project is just the bare bones of what can be accomplished with the VDG chip; but, it can be expanded very easily

to its full potential. If you require more information on this chip then either watch for information in future newsletters or write Motorola for a spec sheet.
>

CASSETTE 'FILE COUNTER' SYSTEM

Tom Crawford

One of the more aggravating aspects of using my home computer system is looking for a specific block of data on a cassette tape, in order to load it into my system, since neither of my cassette recorders has a tape counter. One way to resolve this problem, of course, is to put only one block, or file, on a cassette. You can very quickly find the right cassette by referring to a written directory which keeps track of cassettes by number. Unfortunately, this can become very expensive, with good quality cassettes costing several dollars each. Also, it makes very inefficient use of the tape, since only a small percentage of the tape length is used to record one file.

The only way to make efficient use of a tape is to record more than one file on a tape, but now we are back to the problem of finding a particular block of data part way through a cassette, with no tape counter on the recorder. This means that you, the user, must listen to the tape, starting at the beginning, and count blocks of data until you arrive at the one you want. Then, you can finally request your computer to read the block.

It seemed to me that I ought to be able to get my computer to do some of this work for me, so I set about to find a way.

The first problem was to enable the computer to count blocks of data, just as I was doing by ear. In my system, each block of data is preceeded and followed by ten seconds of all-ones -- Leader and Trailer. In between leaders and trailers there is an unknown amount of 'dead tape', tape which contains random noise. The problem is to distinguish between leader/trailer, noise, and data. If this can be done satisfactorily, then the computer can count data blocks for me.

One way to distinguish between these three states involves the following signals, which are available on my cassette interface (see IPSO FACTO, Issue #3, pp. 23-38):

Carrier Detect - indicates whether the tape contains a tone
representing a '1' (2400 Hz) or a '0' (1200 Hz).
Data Out - indicates whether the data is a '1' or a
'0' when Carrier Detect is true.

These signals are used in the following ways:

Noise - .NOT. Carrier Detect
Data - Carrier Detect .AND..NOT. Leader/Trailer
Leader/Trailer
- [Carrier Detect .AND. Data Out = 1] for 5 seconds
continuouosly.

A sequence state diagram for this system would look like this

```
                           --------
      +---------H--------->| DATA |------------A---------+
      |                    --------<-----------B----+    |
      |                      |   ^                  |    |
 ---------                   |   |                  |    V
 | START |                   E   F                 -----------
 ---------                   |   |                 | LEADER/  |
   |    |                    V   |                 |  TRAILER |
   |    |                  ---------               -----------
   |    +----------G------>| NOISE |                    ^
   |                       ---------                    |
   +--------------I-------------------------------------+
```

The transitions are defined as follows:

A: Data Out stays at 1 and Carrier Detect is true, for Ø.5 seconds or more.
B: Data Out goes to Ø; Carrier Detect stays true.
C: Carrier Detect goes false.
D: Same as A.
E: Carrier Detect goes true; Data Out is Ø.
F: Same as C.
G: Carrier Detect is false.
H: Carrier Detect is true and Data Out = Ø.
I: Same as A.

Now it is simply necessary to design some code to check for and and implement the transitions defined above, to record the state the system is in, and to count the data blocks. A block will be counted when transition A is made (Data to Leader/Trailer). The flowchart to implement this routine is shown in Figure 1, and is coded as a subroutine in Listing 1. Note the inclusion of a call to BRKCHK in the routine. This allows a block count procedure to be aborted by simply pressing the Break key on the console device keyboard.

In the subroutine code, there are no references to the Data Out signal. This signal feeds a hardware UART in my system, and hence is not available to the micro-processor. Instead, the Data Available signal is latched when it is true, and so code is included to reset it when it is tested and found true.

Listing 2 shows a routine which can be used to specify a block, which will be found by the FILCNT subroutine.

FILCNT Subroutine

This routine is used to count blocks of data on a cassette tape. The number of blocks to be counted (1 to 255 inclusive) is in D upon entry to this subroutine. The routine returns when it detects the end of the last block to be counted, or when a Break occurs on the console device keyboard. If a return is caused by a Break, DF=1, otherwise DF=Ø. It is assumed that the tape device is already running when this routine is entered.

```
;                                  ;assume 1.7895 MHz Clock freq.
T5SECS: .EQL   1Ø36                ;Ø.5 seconds delay for timing loop
        .ORG   #Ø9ØØ               ;start in free RAM
FILCNT: .EQL   @                   ;start here. Block count in D and  RF.1
        PLO    R7                  ;put block count into R7.Ø
        LDI    #ØØ                 ; and state code <START> into R7.1
FIL5:   PHI    R7
       +DLDI   T5SECS,R1Ø          ;initialize lead/trail counter
FIL8:   LDI    C1ØCTL              ;point RMMIO to UART status word
        PLO    RMMIO
        LDN    RMMIO               ; read the status word

        ANI    #2Ø                 ;isolate carrier detect bit
        BNZ    FIL1Ø               ;branch if carrier detected
        LDI    #Ø1                 ;no carrier - set state=noise
        BR     FIL5Ø               ;and go try again
FIL15:  DEC    R1Ø                 ;we are checking on leader/trailer
        GLO    R1Ø                 ;decrement timer and test for Ø
        BNZ    FIL3Ø               ;no, keep trying
        GHI    R1Ø                 ;maybe
        BNZ    FIL3Ø               ;no, go look for break
        GHI    R7                  ;yes!
        SDI    #Ø2                 ;is state = DATA?
        BNZ    FIL2Ø               ;no
        DEC    R7                  ;yes - count 1 block
FIL2Ø:  LDI    #Ø3                 ;update state = LEADER/TRAILER
        PHI    R7
        GLO    R7                  ;have we counted needed blocks?
        BZ     FIL4Ø               ;yes! go exit
FIL3Ø: +CALL   BRKCHK              ;no, go check for breaks
        BDF    FIL45               ;break? yes.
        BR     FIL8                ;no, keep looking for blocks
FIL4Ø:  ADI    #ØØ                 ;ensure DF=Ø
FIL45: +RETRN                      ;and return
FIL5Ø: +CALL   BRKCHK              ;go check for breaks
        BNF    FIL5                ;any? no.
        BR     FIL45               ;yes
```

GETBLK Subroutine

This routine requests a block number via the console terminal. This number, input as 2 hex digits, sill be used in a call to FILCNT routine, after starting the tape drive. After the return from FILCNT the tape drive is stopped, and the Exec loop is re-entered at RING. If an error is made in entering the 2 hex digits, this routine is aborted and the Exec loop is re-entered at QUEST.

```
GETBLK: +CALL  ALMPRT          ;ask for block#
        .DBYTE MESG1
        +CALL  BYTE            ;input 2 hex digits
         LBDF  QUEST           ;error? go print "?"
         STXD                  ;save number on stack
         LDI   #11
        +CALL  STRTPE          ;start tape
         IRX                   ;recover number
         LDX
        +CALL  FILCNT          ;count files
         LDI   #ØØ
        +CALL  STRTPE          ;stop tape
         LBR   RING            ;return to executive
MESG1   .ASCIZ "BLK # "
```

NOTE: This subroutine makes reference to several routines contained in RCABUG (IPSO FACTO #1Ø, pg. 55).

FILCNT

```
ADDR      CODE
Ø9ØØ      D4 38 1A                 Ø934      87
  Ø3      A7                         35      32 3E
  Ø4      F8 ØØ                      37      D4 38 2Ø
  Ø6      B7                         3A      33 4Ø
  Ø7      F8 ØØ AA                   3C      3Ø ØD
  ØA      F8 2Ø BA                   3E      FC ØØ
  ØD      F8 FD                      4Ø      D4 38 1Ø
  ØF      AE                         43      D5
  1Ø      ØE                         44      D4 38 2Ø
  11      FA 2Ø                      47      3B Ø6
  13      3A 19                      49      97
  15      F8 Ø1                      4A      D4 3A 1F
  17      3Ø 44                      4D      3Ø 4Ø
  19      ØE                       Ø95Ø      D4 3A 4A
  1A      FA Ø8                      53      Ø9 6E
  1C      3A 24                      55      D4 39 CC
  1E      2E                         58      C3 39 96
  1F      ØE                         5B      73
  2Ø      F8 Ø2                      5C      F8 11
  22      3Ø 44                      5E      D4 3A E2
  24      2A                         61      6Ø
  25      8A                         62      FØ
  26      3A                         63      D4 Ø9 ØØ
  28      9A                         66      F8 ØØ
  29      3A 37                      68      D4 3A E2
  2B      97                         6B      CØ 39 49
  2C      FD Ø2                      6E      42 4C 4B
  2E      3A 31                      71      2Ø 23 2Ø
  3Ø      27                         74      ØØ
  31      F8 Ø3 B7
```

Dear Editor:

Steve Nies,
251Ø Deas St.,
Bossier City, LA.,
USA 71111

After reading Issue #12 of IPSO FACTO, I was both excited and dismayed. The excitement was caused by the editor and assembler programs in the issue. I have been doing a lot of machine language programming and was delighted to see that one was available in IPSO FACTO. Now I can use the money that I was going to use to buy a commercial assembler to pay for my next years dues to ACE (A wise decision-ED!). I do have one question, though, is it possible to have these programs reassembled somewhere else in memory? I would like to put these programs in ROM for future use.

Even though I liked most of Issue 12, I was dismayed to hear that dues are going up to $18. My question is, except for the superb newsletter, what am I entitled to for my $18? That is a lot of money for a bi-monthly subscription. Also, since I live down in Louisiana, I can't come to any activities that the club sponsors. Although I believe the club should have a firm financial backing, I feel that the majority of the members are supporting the activities of a few.

However, since I don't like to complain without trying to help, I have some suggestions that might be of some value: (1) . Have a two-fee system. Those that are able to come to the activities could be charged the full rate, while the remaining members could be charged a lower rate to cover the cost of the newsletter. I feel that this method would allow those who can't afford the full rate (one member being myself) to contribute something to the club. (2) Sell advertisements (computer related only) in the newsletter to help cover the cost of the newsletter. I realize that the club would become commercialized but look at the possibilities. How would you like a magazine comparable to Kilobaud or Interface Age dedicated entirely to 18Ø2 fans?

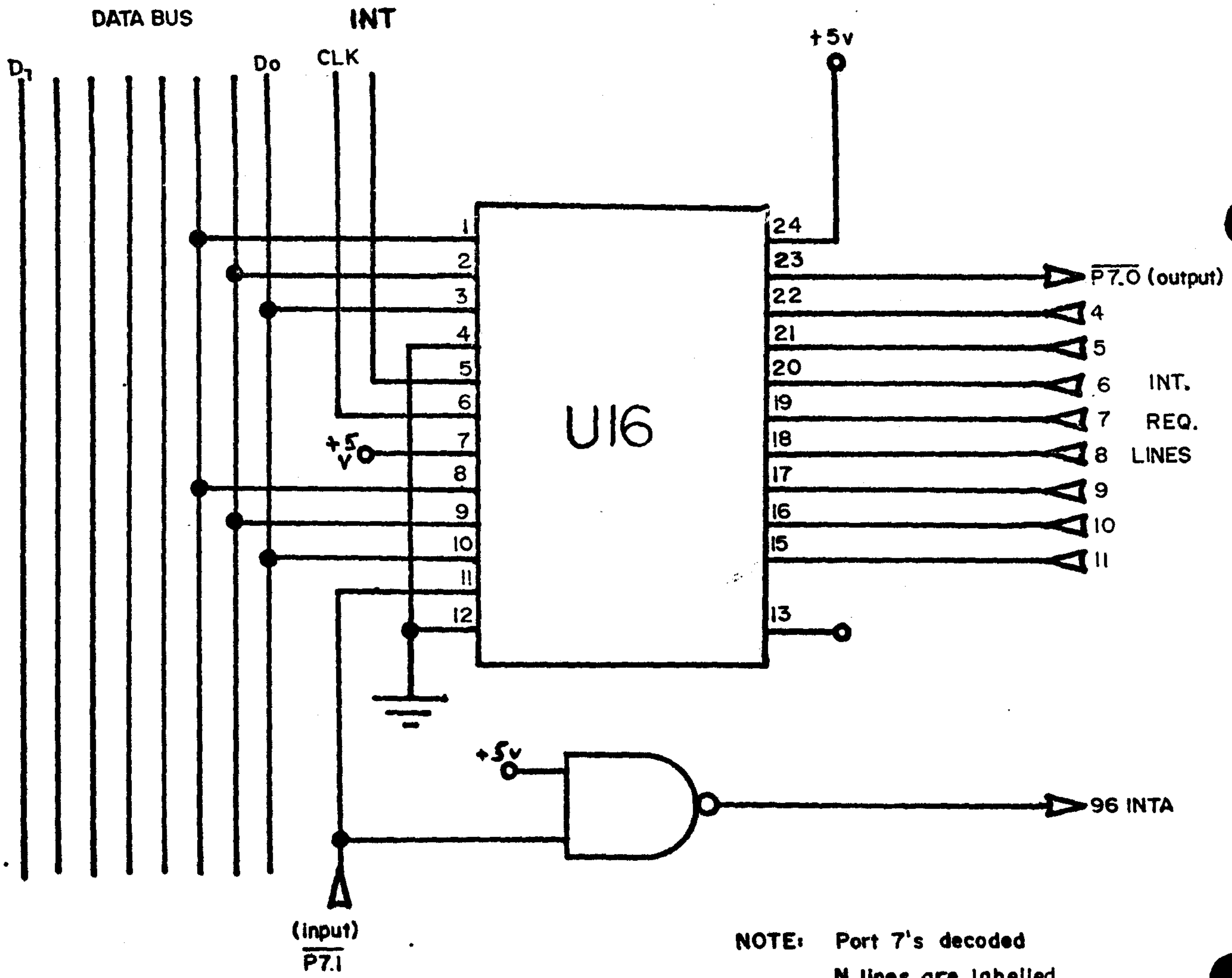
Even if my suggestions are not used, I hope some conclusion can be made about the higher rates. I would hate to see such a fine club as ACE die out due to lack of members.

(Dear Steve: Thanks for the good words about Issue 12. The club did a lot of soul-searching before deciding to increase our annual fees to cover the cost of first class postage for IPSO FACTO. We finally took that step because of all the irate letters complaining about not receiving newsletters on time. Each letter had to be answered, records checked and mail traced. The problem was that third class postage travels very slowly. My own copies sometimes took 3 weeks to to travel less than 3Ø miles. Maybe you didn't mind the wait or perhaps you are blessed with good postal service but many others are not.

We considered a two fee system for first and second class postage but this was dropped as being too difficult to administer. We also thought of seeking advertising but this too was rejected for three reasons: (1) we would lose our independence from commercial concerns (2) volunteers would have to be found to solicit advertisements and (3) there are legal implications to be investigated since we are not an incorporated club.

One point you made concerns us in that it is not true. The club fees cover the cost of producing IPSO FACTO. Local club activities are not paid for out of these fees. Our meeting place is provided free of charge thanks to the Steel Company of Canada and all activities have been of a self-supporting nature. For instance, our guest speakers are not paid (cheapskates-eh?) although usually a local member will spring for coffee. The only exception to this has been the development of the cassete interface/wire-wrap board currently being financed by the club. Once the bugs are out, however, the boards will be sold at cost to all interested club members and the money recouped. Also, if it were not for the hundreds of hours of volunteer work by local members (and distant contributers like yourself) there would not be a newsletter. Hopefully you understand our position a little better now. Sure the $18 is a lot but hopefully you will find it worthwhile. For other members, please let us know your feelings on the new rates and the postal service you are currently receiving. We only make these decisions with one thing in mind--the ultimate satisfaction of club members--YOU!! Ed.)

>



NOTE: Port 7's decoded
N lines are labelled
P7.0 and P7.1

SCHEMATIC A

Steve Nies
2510 Deas Street
Bossier City, LA.
71111 USA

1802 8-LEVEL INTERUPT

Here is my idea for giving an 1802 system an 8 level interrupt ability. This design assumes that a 8214 Priority Interrupt IC is connected to port 7. See schmatic A for details.

After an interrupt occurs, execution is transferred to XX04, where XX can be any location in memory. The interrupt routine then saves T, D, DF, R(3), and R(6). Then the routine reads port 7 to distinguish which interrupt occured. This data is added to a pointer to point to an interrupt vector. Note that location XX1D should be changed so that it plus the data from the port will point to the correct vector. After the vector pointer is loaded into R(3), the return pointer (R(6)) is set to the return-from-interrupt routine (location XX2B). The interrupt subroutine then begins execution at the address pointed to by the vector. These interrupt subroutines are treated as though they were subroutines that were called by the SCRT method. Note that this process allows for the interrupt subroutines to be nested to any level (memory permitting, of course). To exit from the interrupt subroutine, a D5 instruction should be executed. Control then passes back to the return-from-interrupt routine. In this routine, T, D, DF, and R(3) are restored. R(6) was restored by the return (D5) instruction. Execution is then restored back to the interrupted program. Note that with this approach, a low-level subroutine can be interrupted by a higher priority interrupt. When the higher level subroutine is finished, execution then resumes with the interrupted subroutine.

Note that the interrupt vectors must be located in RAM in 16 successive memory locations. These locations can be anywhere in RAM. However, the bytes in program locations XX14 and XX20 must be changed to point to the beginning of the vector table. The vectors are stored in the table high order address first, then low order address second. The two byte vectors themselves are stored in descending order, with vector 7 being at the beginning of the table and vector 0 being at the end of the table.

There are three basic requirements for this program. First, R(1) must be initilized to XX04 before interrupts are permitted. Second, the memory pointed to by R(2) must be a free area of RAM to serve as the stack. Third, R(4) and R(5) must point to the call and return portions of the SCRT routines, respectively.

One advantage not mentioned yet is that with this program, a two byte software interrupt instruction is available . By changing vector 0 to point to the SWI routine, the user can insert a SWI (bytes 79 D1) into his program. When these bytes are executed, an interrupt is faked.

Although this program has been tested, a 8214 was not available for testing. I have completely tested the SWI instruction and partially tested the remaining interrupts. Everything should work fine.

```
                          /* INTERRUPT */

XX00  70 C8                                Go Execute Program
XX02  70 23                                Go Execute Subroutine
XX04  E2 22 78 22 73 7E 73                 Save T, D, DF
XX0B  93 73 83 73 96 73 86 73              Save R(3), R(6)
XX13  F8 S B6                              Set Up Interrupt Pointer
XX16  6F FB FF 52 67 22 FA 07
xx1E  FE FC 70 A6                          S 70 is the beginning of the
XX22  46 B3 46 A3                            interrupt table in my system.
XX26  91 B6 F8 2E A6                       Set Up Return Pointer
XX2B  E1 30 02
XX2E  E3 71 23                             Disable Interrupts
XX31  93 B1 F8 37 A1 D1                    Set PC to R(1)
XX37  F8 FF 52 67
XX3B  42 A3 42 B3                          Restore R(3)
XX3F  42 F6 42                             Restore DF, D
XX42  30 00

                            /* SWI */

XX44  93 B0 F8 4A A0 D0                    On Entry, Set PC to R(0)
XX4A  12 42 A6 42 B6                       Restore R(6)
XX4F  42 A3 42 B3                          Restore R(3)
XX53  42 F6 42                             Restore DF, D
XX56  12 12                                Restore R(2)
XX58  C0 ZZ ZZ                             Jump to user's Program
                                             (in my case, the monitor)
```

TIC-TAC-TOE PROGRAM FOR TWO PLAYERS

by Guy R. Gilbert
3Ø4 Vassal
Drummondville, Que.

Here is a TIC-TAC-TOE program for two players, requiring 1K of memory. It's similar to Ed McCormick's program (Pop. Elec. Nov. 1978, p. 98) except that one player plays on an ASCII keyboard and the other on a hex keyboard. Also, the grid is drawn by software instead of being entered through the keyboard.

To run, INPUT is first depressed to draw the grid. Then ØØ(hex) or Ø1(hex) is entered, depending on who's going to play first -- 'O' or 'X'. 'O' plays on the hex keyboard and 'X' plays on the ASCII keyboard. Who plays next is shown on the upper right hand corner of the screen. If an occupied location is chosen, an 'E' is printed. A new game can be started at any time by entering Ø(zero) on either keyboard.

TIC-TAC-TOE FOR TWO

```
ADDRESS     INSTRUCTIONS        COMMENTS

0003        F800BFF830AF        Initialization.
  09        F801B1B2B4B5
  0F        B8B9BBBDBE
  14        F89BA1F8F2A2
  1A        F84BA4F82DA8
  20        F8FFA9F88EAB
  26        F815ADF82DAE
  2C        F802B6DF
  30        E969                Start TV.
  32        3F323734            Press "INPUT" to
  36        F800A6AC            clear screen
  3A        8C561686
  3E        3A3A26              and
  41        F802A6D8
  45        F804A6D8            print
  49        F850A6DB
  4D        F8A8A6DB            grid.
  51        3F513753            Enter '00' or '01'(hex) for first player.
  55        E96C32B7            If '00', go to '0' plays.
  59        F85CA5              'X' plays: ptint 'X' in upper right-hand
  5C        F80FA6D4            corner of screen.
  60        E93E616F            Input ASCII choice.
  64        FF30
  66        3236                If zero, start new game.
  68        AA596429            Display 'X' choice.
  6C        FB013A74            Check
  70        F809A6DD
  74        8AFB023A7D
  79        F80BA6DD
  7D        8AFB033A86
                                if
  82        F80DA6DD
  86        8AFB043A8F
  8B        F861A6DD
                                 location
  8F        8AFB053A98
  94        F863A6DD
  98        8AFB063AA1
  9D        F865A6DD
                                chosen
  A1        8AFB073AAA
  A6        F8B9A6DD
  AA        8AFB083AB3          is
  AF        F8BBA6DD
  B3        F8BDA6DD            occupied.
  B7        F80FA6              'O' plays: print 'O' in upper right-hand
  BA        F864A5D4            corner of screen.
  BE        E93FBF              Input
  C1        37C16C              HEX choice.
  C4        3236                If zero, start new game.
  C6        AA6429              Display 'O' choice.
  C9        FB013AD1            Check
  CD        F809A6DE
  D1        8AFB023ADA
  D6        F80BA6DE
  DA        8AFB033AE3          if
```

```
ADDRESS    INSTRUCTIONS        COMMENTS

OODF       F80DA6DE
  E3       8AFB043AEC          location
  E8       F861A6DE
  EC       8AFB053AF5
  F1       F863A6DE
  F5       8AFB063AFE          chosen
  FA       F865A6DE
  FE       8AFB073A07
0103       F8B9A6DE            is
  07       8AFB083A10
  OC       F8BBA6DE
  10       F8BDA6DE            occupied.
  14       D4E6F03A21          Subroutine DD: if location chosen
  19       F85CA5              for 'X' is not occupied, print 'X'
  1C       F8B7AF3014          at chosen location; if occupied,
  21       F80FA6              print 'E' in upper right-hand
  24       F86CA5              corner of
  27       F860AF3014          screen.
  2C       D4E6F03A3C          Subroutine DE: if location chosen
  31       F864A5F800BF        for 'O' is not occupied, print 'O'
  37       F859AF302C          at chosen location; if occupied,
  3C       F80FA6F86CA5
  42       F8BEAFF800BF        print 'E' in upper right-hand
  48       302C                corner of screen.
  4A       DFE5F808AC          Subroutine D4:
  4F       F0562C8C            for
  53       324A1586            printing
  57       FC08A6304F          O,X or E.
  5C       8142241818244281    Characters:  X
  64       3C4281818181423C                 O
  6C       001F10101F10101F                 E
  74       07FFFFFFFFFFE000                 grid horizontal segment
  7C       DF86FAF0            Subroutine D8:
  80       FBF0327C            for printing
  84       F81856              vertical part of grid.
  87       86FC08A6307D
  8D       DFF874A5E5          Subroutine DB:
  92       45328D              for printing
  95       56163092            horizontal part of grid.
  99       7270C42278          Interrupt subroutine.
  9E       2252F802B0
  A3       F800A0C4C4
  A8       E280E220A0
  AD       E220A0E220A0
  B3       3CA83099
```

Mike Franklin
24 Duby Road
Acton, Ontario
L7J 2P1
July 17, 1979

More Power to the Elf!

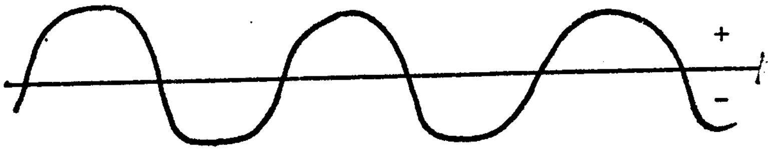
It will happen one day. You will plug in your latest super kluge, flip the power switch, and - nothing! Amps required exceeds Amps available. Time to build a bigger power supply. Scenario two - your super kluge requires a variety of voltages which are not available on your power supply. Time to build a second supply. Better yet, put them both together, and keep your 1802 humming along happily. The following brief discussion on power supply theory and design may help you design and build a unit for your own particular requirements.

The Power Supply

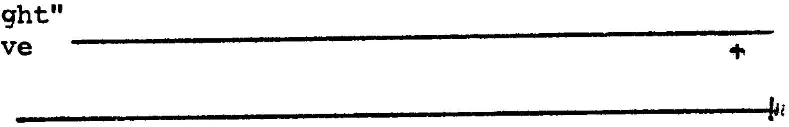
The unit basically consists of a transformer to reduce available primary Alternating Current (AC) voltage (110-120, 220-240) to a lower A.C. voltage which can be effectively converted to Direct Current (D.C.) via a diode or bridge rectifier, and then capacitively filtered to produce a continuous, low ripple un-regulated D.C. supply. Finally the D.C. voltage is regulated to a lower voltage to ensure continuous, ripple-free output required for electronic components.

A Little Theory

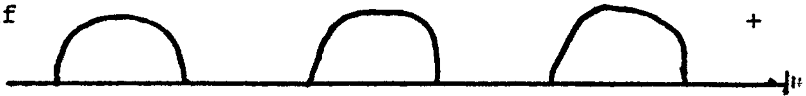
Alternating Current (A.C.) flows as a sine wave, with equal waves alternating above and below a common ground plane.



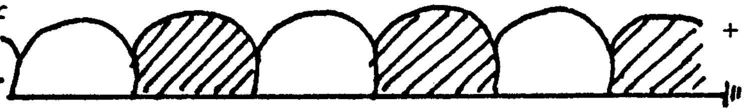
Direct Current, (D.C.), flows in a "straight" line, either positive or negative, relative to a ground plane.



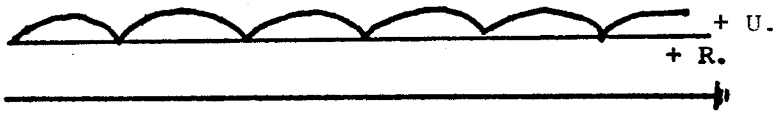
A.C. voltage is converted to D.C. voltage by rectification, or eliminating one half of the sine wave.



The "gap" between two waves is referred to as ripple, and is filled in by a capacitor filter, which discharges a voltage equal to, or slightly less than the A.C. output.



The D.C. output will still not be a "straight" line, however, a reduced voltage maybe produced which is relatively ripple free, by use of a voltage regulator (a regulated output).



Power Requirements

Obviously the first step in designing your unit is to determine which voltage levels you are going to require and how much current (Amperage) at each level. Let us assume you will require +5v regulated (R) for your 1802 and RAM, +5vR, -5vR and -12vR for Eprom, +5vR and -12vR for a keyboard and video board, and +15v and -15v unregulated (U) for an RS 232 circuit.

Amperage requirements vary tremendously with component specification and circuit design. My advice is to be conservative. Transformer prices do not increase significantly for larger Amperage units and larger units provide better regulation. Therefore, sum all the maximum power requirements of the components you are using.

My own requirements turned out to be 5 Amps for +5vR,and 1 Amp for each -5vR, +12vR and -12vR, rated as continuous serv+ce. Continuous service refers to long term continuous output, rather than peak output, and is rather akin to the reduced D.C. regulated voltage output. It is a level which can always be delivered with all components operating.

The above power requirements, for ±5vR, ±12vR and ±15vU are not uncommon if one is using 3 voltage EPROMS. The negative voltage is always available when the positive voltage is generated by a rectifier or diode bridge.

Voltage Regulation

To achieve a regulated voltage output, the unregulated voltage is passed through a voltage regulator. Manufacturers specify that the input voltage must exceed the output voltage by at least 2 volts. A 5 volt regulator therefore, requires a 7 to 8 volt input, and a 12 volt regulator requires a 14 to 15 volt input. A few extra volts wills allow for line fluctuation. The common TO-220 and TO-3 package units will sink up to 35 volts each, however, excess voltage produces heat, and should be avoided.

Negative regulators have a different pin out than the positive units, as follows:

	TO-220			TO-3		
regulator	pin 1	pin 2 & tab	pin 3	pin 1	pin 2	case
positive	input	common	output	input	output	common
negative	common	input	output	common	output	input

Obviously, negative regulators have to be insulated from their mounting hardware and heat sinks if these are common with other regulators or components. Mica washers and nylon inserts are available for both case types for this purpose.

Voltage Levels and Polarity

There are a variety of ways to achieve a positive and a negative voltage from a transformer.

A centre tap transformer may be employed, where the centre is considered common, or ground, and the two end leads connected to a rectifier to produce a positive and a negative voltage, relative to the centre tap.

Voltage Levels and Polarity (Cont'd)

Alternatively, one end lead would be considered common, and the centre tap and the remaining end lead connected to two diode bridges to produce two different voltage levels, each with a positive and a negative output. This method however, reduces the amperage current available significantly.

A third method of obtaining different voltage levels employs a voltage doubler to increase a level of voltage output. This method has the undesirable characteristic of changing the voltage level of the ground plane, which may affect the operation of circuits connected to it. This method should be avoided for this reason.

I chose the first method and used two transformers to deliver the two voltage levels required. This method reduces the amount of voltage which the 5 volt regulators must sink, and therefore reduces the amount of heat generated, which in turn, might contribute to component failure.

Transformer Specification

Transformer manufacturers refer to centre tapped transformers by a single voltage value. This value represents the voltage obtained between the two end leads only, and, if the centre tap is used as common ground, the voltage available at each lead will be half that of the given value. Also, the Amperage given refers to the total available from the transformer,but this amount may be drawn unequally by the positive and negative voltage circuits. However, the calculations don't end there. The ratings are given in A.C. and not D.C., and the final voltage and amperage delivered by the transformer depends upon the rectification and filtering circuits used.

Transformer Calculations

A bridge capacitive circuit, such as I used, will produce D.C. voltage output 1.25 times the A.C. voltage input, half on each side of common. It will also produce 0.56 times the Amperage input, continuous service, shared between the two outputs.

To determine the transformer size required, the following formula is applied: (conservative rating)

A.C. secondary output = D.C. output (totalled) x 1.25 (totalled refers to both +ve and -ve).

For 9 volts per polarity, sec. output = 18 ÷ 1.25 = 14.4v A.C. centre tap transformer.
For 15 volts per polarity, sec. output = 30 ÷ 1.25 = 24.0v A.C. centre tap transformer.

Nine volts will yield a regulated 5 volt supply and 15 volts will yield a regulated 12 volt supply, as well as provide the ±15vU for a RS232 or 20ma loop circuit, quite sufficient for my 1802's circuits.

The bridge capacitive circuitry will yield only 0.56 times the input Amperage. To provide 6 Amps continuous service for my +5 and -5v supply, a transformer with a 10 Amp capacity is required. The 2 Amp requirement of my +12 and - 12 volt circuits is provided by a transformer with a 4 Amp rated transformer.

Transformer Calculations (Cont'd)

To provide the required ripple filtering, the following formula is applied to determine the size of the computer grade electrolytic capacitors used:
(5% ripple)

$$\text{Capacitor size, micro farads} - \frac{\dfrac{2 \times 10^{-7} \times \text{max. current drawn}}{377}}{\text{output voltage}}$$

That translates to: $\frac{\dfrac{2 \times 10^{7} \quad \times 6}{377}}{9}$ or 35,367 micro farads for my -9v D.C. circuit, which was met by an industry standard 37,500 micro farads, 16 volt capacitor.

I rated this circuit to handle all 6 Amps, while the other voltages and polarities were rated for 1 Amp each.

For my remaining three voltages, I used the same capacitor value, derived as follows:

$\frac{\dfrac{2 \times 10{,}000{,}000 \times 1}{377}}{9}$ or 5894 micro farads, which was met by an industry standard value of 7000 micro farads. 16 and 25v capacitor.

When dealing with the negative voltage circuits, remember to connect the capacitor's positive terminal to the common ground circuit.

Finally, the common leads of the transformers are connected to assure a common voltage level for the ground plane, and heavy wire, 14 guage, should be used for all the internal power supply connections to minimize line resistance, which could reduce voltage levels.

The enclosure you provide should allow good ventilation,,particularly if your voltage regulators are located within the cabinet.

Make sure that the connectors and output lines are well insulated to prevent unwanted short circuits.

The following circuit diagram of my power supply may assist you in designing your own unit.

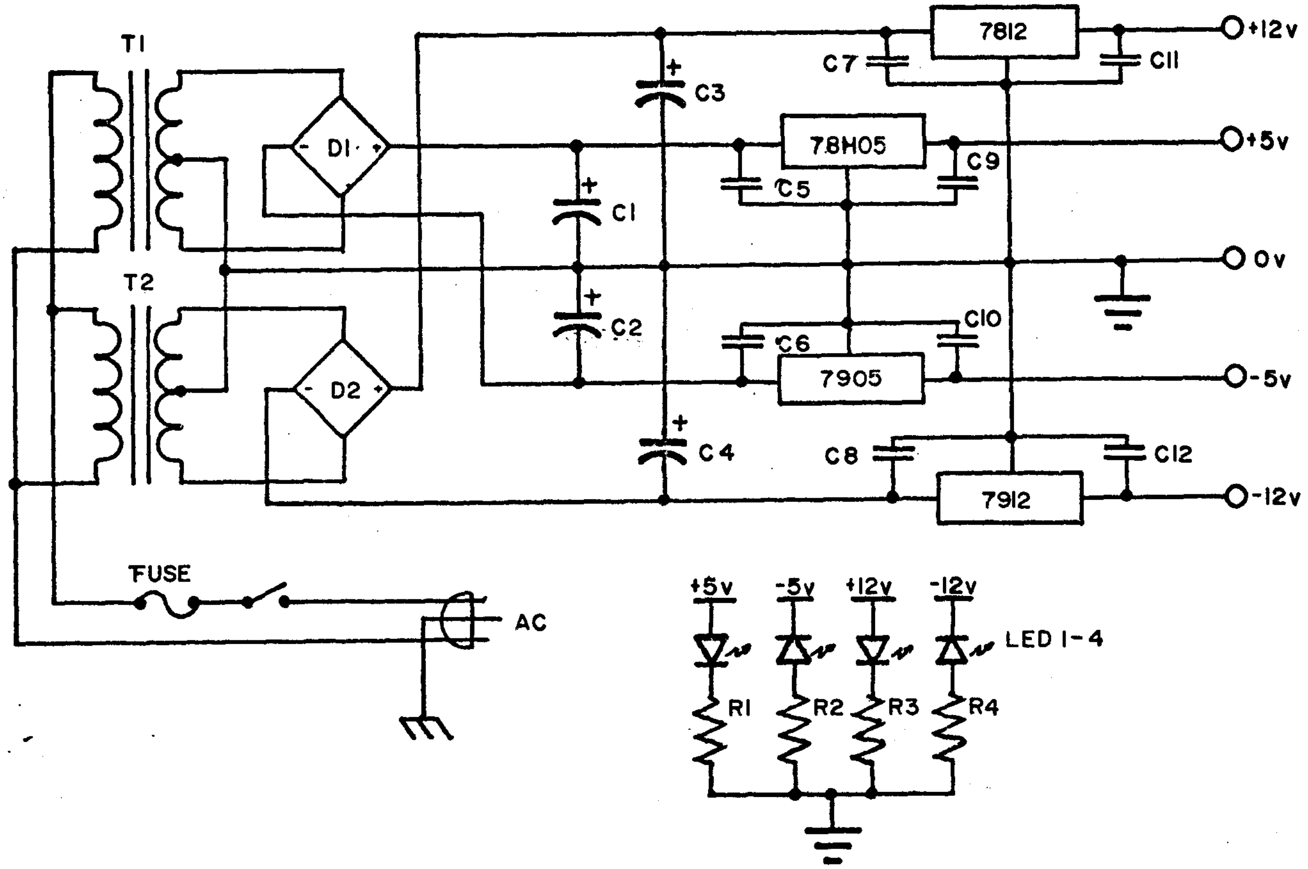
Bypass capacitors C5-12 have to be mounted right on the REGULATOR IC's to prevent internal oscillations.

The LEDS have a dual purpose: to indicate that all supplies are ON, and when off they drain the residual charge that is left in the electrolytic capacitors.

R1,2 should be 500 ohms; R3,4 should be about 1K. Values can be picked to get 20 ma through each LED:

$$I = \frac{E}{R} \quad \text{therefore:} \quad R = \frac{E}{I}$$

"ELF POWER" SUPPLY



ERRATA -- 27Ø8 EPROM PROGRAMMER

The errors are as follows: both 74LS373 and 74LS374 are wrong for the data decoders. They should be 74LS273. 74LS374 may be used if pin 1 is grounded (and not tied to +5). The same is true for a 74LS373, but it is not recommended. One other error exists...The '63' instruction going into the 4Ø42 should be a '65' instruction.

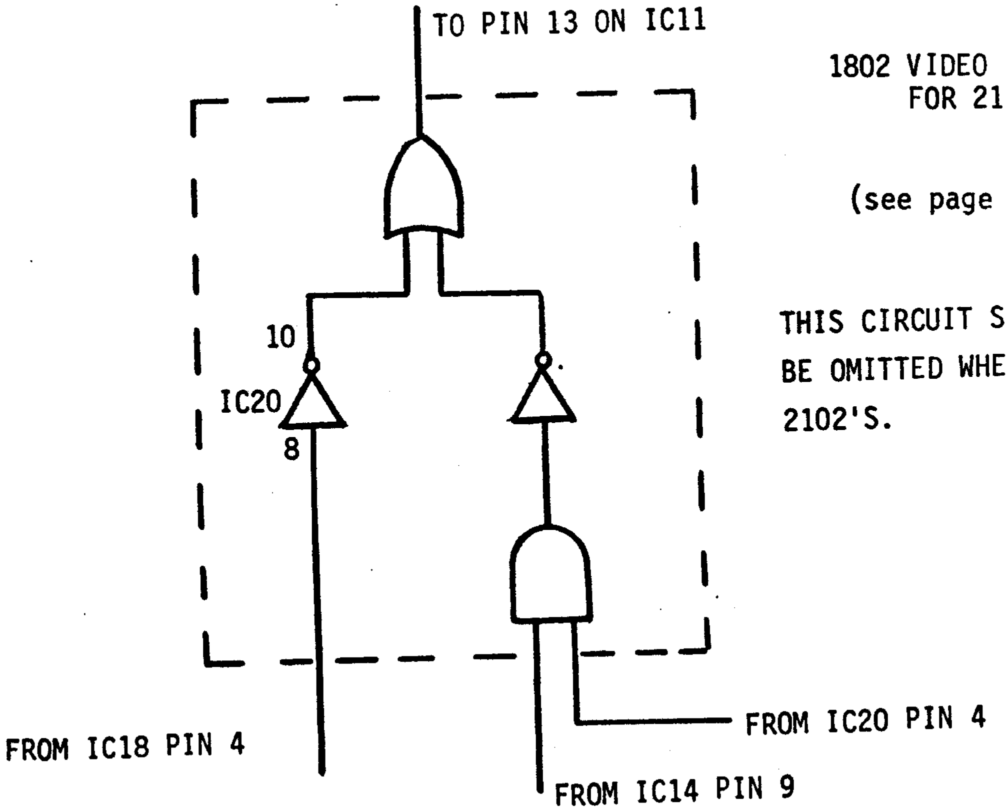
One additional note on the 27Ø8 Programmer. The instruction decoder is not necessary if the Giant Board is used. Simply use its I62, I63 and I65 lines. On Quests Super Elf Device 2 is 'pseudo' N1, and Device 3 is available at U3-15, and Device 5 is available at U-3 pin 6. They need the decoder shown, driven by MRD (which was unreadable in my IPSO).

12" Monitors - For Sale
Brand new Electrohome - in box
Designed for Westinghouse Terminals.
Uses Column Scan $90.00
Modification information
supplied for standard scan.
Modified Units $125.00
Contact: John Polkinghorne
639-4528
(after 7:00 P.M.)

PARTS LIST:

T1	1 - 18 V ct transformer, 1Ø Amp.
T2	1 - 25 V ct transformer, 4 Amp.
D1,2	2 - 35 Amp. 5Ø V bridge rectifiers
C1	1 - 37,5ØØ microfarad computer grade capacitor, 16 V.
C2	1 - 7,5ØØ microfarad computer grade capacitor, 16 V.
C3,4	2 - 7,5ØØ microfarad computer grade capacitor, 25 V.
Misc:	
	1 - 1 Amp fuse
	1 - 3 wire cord
	1 - suitable vented enclosure
	- suitable insulated lead connectors
	- heat shrink tubing and insulated connectors
	1 - switch
Optional Onboard Power Regulation:	
	1 - 78Ø5, 5 Amp.
	1 - 7812, 1 Amp.
	1 - 79Ø5, 1 Amp.
	1 - 7912, 1 Amp.
	2 sets mica washers, insulators and inserts
	1 - 4 x 6" heat sink
C9,11	2 - Ø.1Ø microfarad tantalum capacitors
C5,7	2 - Ø.33 microfarad tantalum capacitors
C1Ø,12	2 - 1.Ø microfarad tantalum capacitors
C6,8	2 - 2.Ø microfarad tantalum capacitors
	1 4 x 6" heat sink
LED1-4	4 RED LEDS (green or yellow will do)

>



1802 VIDEO DISPLAY MODIFICATION FOR 2114 MEMORY

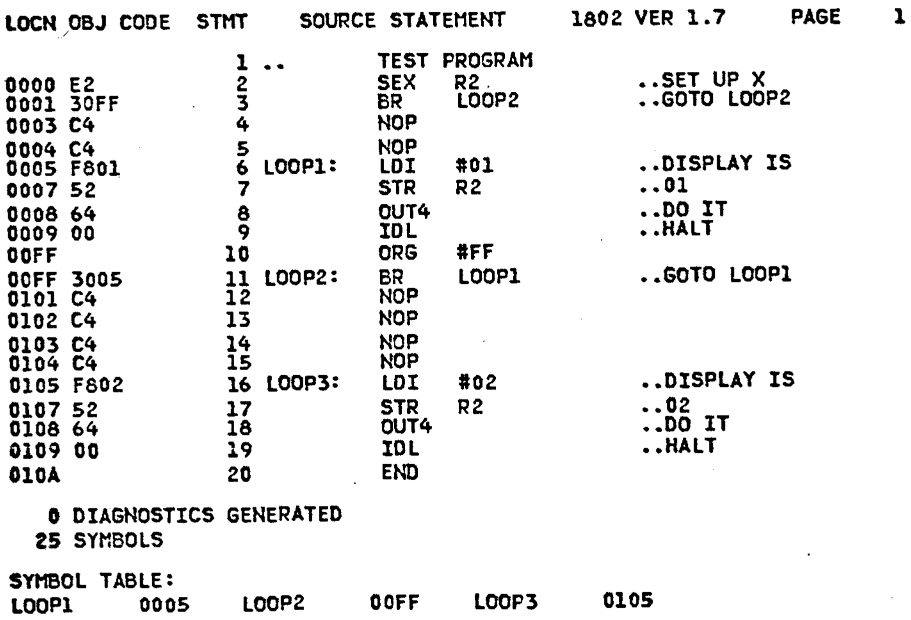
(see page 5)

THIS CIRCUIT SHOULD BE OMITTED WHEN USING 2102'S.

MACHINE LANGUAGE PUZZLER

Bernie Murphy
102 McCrany Street
Oakville, Ont.

Read this piece of code very carefully, and try to figure exactly what will be displayed if the program executes (and why). Be careful -- there is a tricky point involved.
Answer next month.



```
LOCN OBJ CODE  STMT    SOURCE STATEMENT      1802 VER 1.7     PAGE    1

                  1 ..         TEST PROGRAM
0000 E2           2            SEX    R2               ..SET UP X
0001 30FF         3            BR     LOOP2            ..GOTO LOOP2
0003 C4           4            NOP
0004 C4           5            NOP
0005 F801         6 LOOP1:     LDI    #01              ..DISPLAY IS
0007 52           7            STR    R2               ..01
0008 64           8            OUT4                    ..DO IT
0009 00           9            IDL                     ..HALT
00FF             10            ORG    #FF
00FF 3005        11 LOOP2:     BR     LOOP1            ..GOTO LOOP1
0101 C4          12            NOP
0102 C4          13            NOP
0103 C4          14            NOP
0104 C4          15            NOP
0105 F802        16 LOOP3:     LDI    #02              ..DISPLAY IS
0107 52          17            STR    R2               ..02
0108 64          18            OUT4                    ..DO IT
0109 00          19            IDL                     ..HALT
010A             20            END

   0 DIAGNOSTICS GENERATED
  25 SYMBOLS

SYMBOL TABLE:
LOOP1      0005     LOOP2      00FF     LOOP3      0105
```

Future A.C.E. Meetings

October 9 - the guest speaker will be from Hewlett-Packard and the subject will be trouble-shooting hardware problems.

November 13 - Eugene Tekatch will be the guest speaker.

Location - Stelco Auditorium.

We hope to resume tutorials soon, but there are none scheduled at the present time.

Dear Bernie:

First of all, let me mention that the crew up there at IPSO FACTO is doing one heck of a good job. I look forward to getting it each time.

I completed my S1ØØ interface a few weeks ago and up to now it seems to be working quite well. I do not know whether by now you have any article on this yet. (Please send us an article--sounds interesting. What specifically are you using the interface for? ed.) I have maintained the standard 18Ø2 signals. The only exception being the latched 16 bit address lines.

I wrote Osborne to find out about the 18Ø2 book which they were supposed to be preparing, and got the following answer "we have no plans to publish such a book in the near future".

Next I will be working on hooking up tripple Data General cassette drives. I would appreciate any help from any one who could help me with info on ratio recording. I have the theory worked out but the hardware has me down. C.W VLAUN, P.O. BOX 624, SEROE COLORADO, ARUBA NETH. ANT.

>

SUPER GRAPHICS CONTROL PROGRAM FOR THE ELF

So you bought an ELF. And here you are reading this newsletter and you may even be sorry that you bought an ELF now that you have seen that the VIP has more graphics programs. Well your troubles are over. There is a new program entitled 'SUPER GRAPHICS CONTROL PROGRAM'. This program is extremely usefull with its 31 commands. These include:

+ screen clear (to black or white)
+ 4 diagonal cursor movements, 2 vertical movements
+ blinking cursor, tone cursor or no cursor
+ audio visual error messages
+ cursor go home, cursor travel, cursor clear, cursor set
+ an automatic repeat/debounce feature which may be adjusted to repeat at almost any speed.

The program, not including video display buffer, takes about two pages (458 bytes) -- stack included. Any display resolution can be used, by making a slight program modification as described in the literature. The program may be purchased for $6.5Ø (reproduction and postage costs) from: JOHN CARTER, 36 GROUSE DRIVE, BRENTWOOD, N.Y. 11717 USA. Make payment by check or money order only. Thank you.

>

AN APPLICATION OF MEMORY-MAPPED I/O

Tom Jones
6Ø13 Whittier Dr.
Raleigh, N. C. 276Ø9

This ia an application of the mapped memory i/o circuit in IPSO FACTO. I use it for cassette motor control, but other uses are also possible. In this application the N. C. contact of the SPDT relay serves as a dynamic brake circuit. This stops the tape faster and saves tape otherwisw consumed by the motor coasting down. The two 555's can be replaced by a 556. Note the +7-+8 supply to the 555's. This comes from the unregulated supply to the +5v regulators, and although the relay works on +5, it works better and faster at +7-+8 volts.

A note on parts: The latching relay shown was purchased from Chaney Electronics, P. O. Box 27Ø38, Denver, Colorado 8Ø227. It cost $1.ØØ each, and will switch from N.O. to N.C. (actually these are my designations for this circuit only) or N.C. to N.O. with only a pulse. This works nicely for computers and saves draining your power supply. The pin numbers are those printed on the relay case (dip ic type) and are polarity sensitive. If you reverse one of the coil connections from what is shown the relay will not latch on one side.

I.

```
ØØ2Ø E2                 sex2
ØØ21 F8 EC BD           ldi 'EC', phi D
ØØ24 6C AD       LOOP:  inp4, plo D
ØØ26 64 22              out4, dec 2
ØØ28 37 2D              B4 LOAD
ØØ2A 5D                 str D
ØØ2B 3Ø 24              br LOOP
ØØ2D ØD          LOAD:  ldn D
ØØ2E 3Ø 24              br LOOP
```

1. Hex keys select mapped memory address ØØ-FF, page 'EC'.
2. 'I' key down selects 'IN', up selects 'OUT'.
3. Output of 74154 = +5, pulses low in continuous pulse stream.

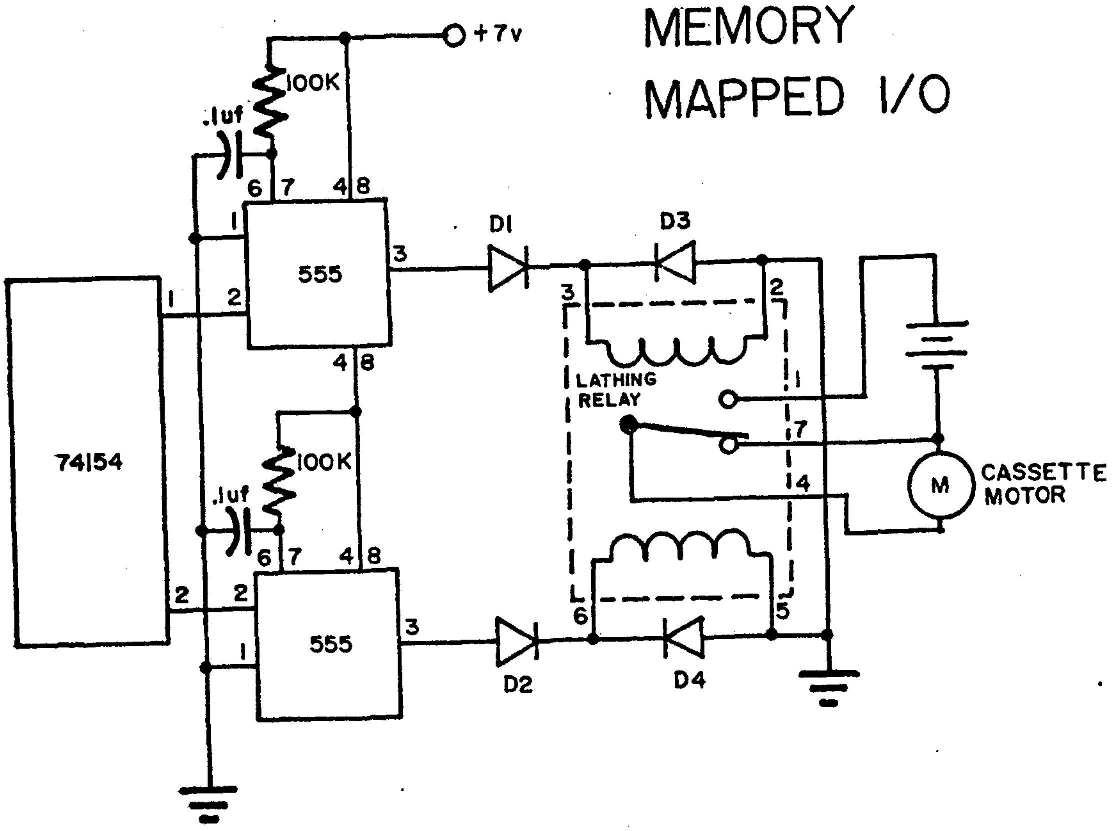
II.

```
ØØ2Ø E2                 sex 2
     F8 EC BD           ldi 'EC', phi D
     6C AD       LOOP:  inp4, plo D
     64 22              out4, dec 2
     37 28              b4 *     - loop until something input
     5D 7B              str D, seq
     3F 2C              bn4 *
     37 2E              b4 *
     ØD 7A              ldn D, req
     3F 32              bn4 *
     3Ø 24              br LOOP
```

1. Hex keys select mapped memory i/o address ØØFF, page 'EC'.
2. Push 'I' key to alternately send one pulse to ECØØ out and then to ECØØ in, and repeat. (hex keys = 'ØØ')

These are simple test routines to check out the mapped memory i/o and relay. Routine II is especially to test the latching action of the relay.

MEMORY MAPPED I/O



18Ø2 EDITOR/ASSEMBLER FIXES G. E. Millar

Please make the noted changes to the editor program (ØØE2-ØØF7). Also, note the warnings below. There are also some changes to the assembler program to correct two misspelled mnemonics, and to fix two problem areas (I/O and mnemonic error). I hope that with the following changes, everyone will be able to make good use of the programs. If there are any problems not covered here, please let me know.

In the editor note:

1. Avoid using "P" directly after delete.
2. After an "error #4" do not use I, D, or P (all others okay).

In the assembler note:

1. Do not use a semicolon directly after an opfield consisting of a label

```
e.g.  LDI LABEL;comments
               ^ a space must precede this or it won't be
                      recognized.
```

Corrections to the editor:

```
ØØE2 - 8F322230EAD400F82F93D4017DØ8
ØØFØ - 32258F32251B3ØE7
```

Corrections to the assembler:

```
Ø1CF - 3ØA4          Ø278 - F3C2
Ø239 - 6C            Ø4ØC - 4349
Ø259 - 72            Ø468 - 4F
```

IPSO FACTO
Topical Index
Issues 1-6

The following list of articles appeared in Ipso Facto in the first year of the club operation. Each article has been categorized as hardware - H; software - S; or conceptual or information - C oriented. Errors have been cross referenced.

Issue 1

Page	Article	Category	Author
5	Music and Micros	S - H	E. Tekatch
6	Morse Code Program	S	B. Fox
7	Light Pattern Display	S	B. Gerrish
9	A Subroutine for Pseudo-Random Number Generation	S	T. Crawford

Issue 2

Page	Article	Category	Author
3	16 x 16 L.E.D. Matrix	H - S	B. Waldock
9	A Cross Assembler - What's IT Mad At??	C	W. Bawdish
13	TEC 1802 Memory Expansion Board #1	H	E. Tekatch
14	Lottery Ticket Program	S	N. & D. Inkster
15	Two Simple Switch Additions for Easier Use of the TEC 1802	H	B. Gerrish
17	TEC 1802 Editor Program	S	E. Tekatch
18	Morse Code Program with ASCII Keyboard Input - errata I.F. #13 p.38	S - H	J. Kolodziej B. Fox
21	A Note on the Morse Code Program	C	W. Bowdish

Issue 3

Page	Article	Category	Author
3	The 1802 D vs the 1802 CD	C - H	K. Smith
9	A Hexadecimal Display	H	F. Feaver
11	Alternate Keyboard System for TEC 1802	H	P. Antony
11	Some Notes or a TVT-6 to TEC 1802 Interface	C - H - S	T. Crawford
19	CUTS - Computer Users Tape System	C	R. Marsh
23	Cassettes and Computers - errata I.F. 6 p. 55	C - H	T. Crawford

Issue 3 (Cont'd)

29	RCA 1802 - KC Standard Cassette Interface Test Routine - errata I.F. #4 p. 29 and I.F.#6 p. 55	S	A. Dunlop
31	A Standard Format for Cassette Data	C	W. Bowdish
39	Constitution of the Association of Computer Experimenters	C	Exec.

Issue 4

3	A Text Editor	C	W. Bowdish
9	Economical Hex Display	H	C. Williams
10	A Monitor for ASCII Keyboard and Display	C	T. Crawford
15	Memory Mapped I/O for the 1802 - errata - I.F.#5 p. 43	C - H	T. Crawford
19	+/- 12 volts on the Tekatch Bus	C	T. Crawford
20	Memory Test Routine for the TEC 1802	S	A. Dunlop
23	Random Number Generation for the 1802	S	B. Murphy
25	Rom out of Ram - errata I.F.#6 p.55	H	M. Pupeza
26	Another Keyboard Approach	H	R. Kindig
28	Interupt Processing on the 1802 - errata - I.F.#5 p. 44	C - H - S	B. Murphy
31	RCA - 1802 Mini Editor	S	E. Tekatch
33	Notes on Connecting an 1861 to a T.V.	H	B. Widner

Issue 5

4	Hardware Paper Tape Loader - errata I.F. #6 - p. 55	C - H	H. Shanko J. Munck
11	Is Your Microcomputer S-100 Compatible	C	Wm. Pfefferman
12	Logic Testor	H	T. Jones
19	A Single Cycle Circuit for the 1802	H	Wm. Pfefferman
20	A Fine Resolution Audio Oscillator Program	S	R. Edwards
21	Certifying Audio Tape for Digital Use	C - H	
22	A Coin Toss Program	S	R. Taubert
24	Magnetic Tape Data Recording	C - H	K. Smith
26	A Simple 25-IC-2 Transistor Code Practice Oscillator	H - S	D. & N. Inkster D. Olenick

Issue 5 (Cont'd)

27	Hex - Decimal Conversion and ASCII	C	K. Smith
30	A Dis-Assembly of Ed McCormick's Monitor	S	R. Edwards
35	Note of Caution on Using my Programs for Cassette Interface with Clock Rates Less than 2 MHZ	C	E. McCormick
36	The 1802 Music Machine	C - H - S	C. Williams
41	RS-232C Interface	C - H - S	B. Murphy
44	TEC 1802 MB1 (0.75 K Memory Board)	C - H	D. Carrigan
49	A Low Cost 8 Digit Display	C - H	B. Gerrish
52	Using the 8 Digit Display	S	W. Bowdish

Issue 6

5	A Software Standard for Kansas City Format Tapes - errata - I.F. #7 p. 5	C	B. Murphy
10	Abstracts from Dr. Dobbs Journal	C	P. Birke
11	Machine Language Game of Life Program for Cosmac 1802	C - S	B. Hutchinson
23	A CMOS 16 x 32 Video Display	H	G. Millar & H. Shanko
29	Build a 1 K Video Ram - errata I.F. #7 p. 5	C - H	R. Parker
36	An Extra Page of ELF Memory	H	Wm. Webb
38	Instant Editor	S	S. Takahashi
38	Using a Baudot Teletype as an Output Device for an 1802 Based System	C - H - S	B. Millier
47	A Mouse Trap Game for Pixie Graphics	S	D. Rubis
50	1802 Programmer's Notebook	H - S	D. Wright
57	An 1802 Ram System	H	B. Murphy
59	Combination Lock and Door Chime	H - S	A. Tekatch
61	A Different Cassette I/O Routine	C - S	R. Edwards
65	Ticket - Winning Ticket Draw Program	S	

IPSO FACTO
Topical Index
Issues 7-12

Each article has been categorized as hardware - H; software - S; or conceptual or information - C oriented. Errors have been cross referenced.

Issue 7

Page	Article	Category	Author
3	Tapeworm Anyone?	S	J. Foster
3	Hi-Low Game	S	J. Howell
6	Is it a Keyboard?	S - H	R. Nelson
7	Electronic Metronome	H - S	R. Edwards
12	1802 Op-Code Table	C	B. Murphy
14	Hangman Program	S	J. Stephens
16	Tiny Basic on the VIP System	C	R. Blessing
18	Display Page of Memory Program	S	R. Blessing
20	Check Book Balance Program	S	R. Blessing
24	1802 State Indicator	H	D. Robarts
26	Copy Memory Routine	S	
28	Microcomputer Interfacing	C	J. Doyle
31	1802 Manual Debugger	C - S	T. Pittman
37	An 1802 Dis-assembler	S	B. Murphy
44	A Tiny Basic Square Root Routine	C - S	T. Crawford

Issue 8

4	Mouse Trap Game	S	J. Laveck
4	Binary Quiz Program	S	M. Cohen
7	VIP Software Update	C - S - H	R. Blessing
8	Add a State Display to the Cosmal ELF	H	D. Grenewetzki
10	Software for the CMOS 16X32 Video System	S - H	G. Millar
14	The Three Keyboard Problem	H	J. Smith
15.	Programmer's Zodiac	S	J. Stephens
19	An RCA CPD 1802 System	C - H	B. Freymuth
21	Tape Conversion Program ELF To VIP - VIP to ELF	S	B. Freymuth
26	Software for 1K VIDEO Ram - ref. I.F. - Pg 29		

Issue 8 (Cont'd)

Page	Article	Category	Author
28	Variations on a Theme	S	D. Hersker
30	Tiny Basic Square Root Routine Version 2	S	J. Howell
30	Extremely Fast Memory Test Program	S	K. Smith
31	An 1802 DMA Controller	H	K. Bevis
35	Memory Page Decoders	C - H	K. Bevis
36	Logic Probe	H	M. Pupeza
37	Software for the "IPSO FACTO" Standard for Kansas City Tapes	C - S	B. Murphy
44	A to D Notes	S - H	R. Nelson
45	Christmas Computer Music	S	C. Airhart
48	Hardware Basics	C	F. Feaver
52	Game of Life Update	S	B. Hutchinson Jr.

Issue 9

Page	Article	Category	Author
4	Some Thoughts on the Call and Return Technique	C	V. Raab
6	Cosmal ELF References (in Popular Electronics)	C	M. Skodny
6	Automatic Program Counter Stepper	C - H	C. Airhart
7	Notes on Netronics' Tiny Basic	C	D. Hersker
8	Some ELF II Enhancements	H - S	T. Jones
9	Some Simple 1802 Programs	S	J. Smith
10	Programming the DM8577 (32X8 Prom)	C - H	F. Feaver
13	Ping Pong	S - H	R. Delombard
18	Fix for 1861 Video Jitter	H	G. Fournier
18	Radio Shack Keyboard Enhancement	H	G. Fournier
19	New Basic for Cosmac 1802 Available	C	R. Edwards
20	Netronics Monitor Disassembled	S	K. Mantel
22	Partial Display Subroutines for 256 Byte ELF System	S	K. Mantel
22	Interesting Computer-Oriented Articles	C	C. Williams
24	Running the "IPSO FACTO Standard" Cassette Software on the ELF	H - S	B. Murphy
31	Tic Tac Toe with the 1802	H	D. Burniston
34	Key-in Loader for ELF II Format Tapes	C - S	T. Jones
36	Life for an ELF	C - S	D. Hersker

Issue 10

Page	Article	Category	Author
4	MB1 Battery Protected Ram (BPRAM)	H	F. Feaver
4	A/D Converter Experiences	C - S	J. Davis
5	Direct Video for a Netronics ELF II	H	M. Franklin
6	Increasing the Clock Frequency on the TEC-1802	C - H	F. Feaver
7	Software Tape Counter?	C - H	C. Morris
7	Replacing the TEC-1802 Keyboard	H	F. Feaver
8	The Game of "Mastermind"	S	C. Airhart
14	Netronic's Basic Now Works!--Tiny Basic Does Not Work!!!	S	M. Franklin
16	Some Thoughts on the High Level Cassette Format	C	T. Jones
18	An Open Letter to Users of Netronics Tiny Basic	S	T. Pittman
19	Some Thoughts on 1802 Pascal	C	V. Raab
20	CHIP-10 Interpreter for the Cosmac VIP	C - S	B. Hutchinson Jr.
32	A 16k Memory System	H	B. Murphy
38	Torus Life Program	S	Wm. Webb
46	Another Standard	C	T. Jones
	ref. pg 35	S	T. Jones
55	RCA Bug	S	T. Crawford

Issue 11

Page	Article	Category	Author
4	ASCII to HEX Converter	H	R. Mack
6	Packet Radio (using an 1802)	C	K. Smith
10	A 2708 Eprom Programmer Board	H - S	M. Coyne
14	Double Buffer Speedup Hardware for 64X128 Graphics with the Cosmac 1802 and 1861 Vidio-Chip	C - H	B. Hutchinson Jr.
21	Event Timer	H - S	M. Coyne
23	Chess Tutor	S	C. Rosen
26	Horse Race Program - errata I.F. #12 p 29	S	G. Gilbert
30	Stepper Motor Program for RCA 1802 Using the TEC 1802 or any M.P. with 256 Bytes of RAM	S - H	J. Rustenburg
34	ASCII Keyboad Interface	H	J. Rustenburg
35	Computer Hobbyists Mumble to Themselves A lot	C	M. Franklin

Issue 11 (Cont'd)

Page	Article	Category	Author
35	To VIP an ELF, Games and Video Manipulation from RCA	C - S	L. Clock
	- errata I.F. #12 p 29		M. Franklin
39	Kilobaud 1802 Articles	C	M. Skodny
39	Tiny Basic Notes	C - S	J. Smith
40	Tec-1802 Editor Corrections and Enhancements	C - S	F. Feaver
42	More About Hardware Basics	C	F. Feaver
42	Adding a Math Function to Your 1802	C	F. Feaver
43	Make Your Own Power Transformers	C - H	K. Bevis
47	Annotated Bibliography of Articles Pertaining to 1802 Uses	C	E. Fleming

Issue 12

4	Data Acquisition/Controller Subsystem	C - H	W. Greason.
14	Video Voice	S	D. Roberts
15	ELF Writer	S	R. Moffie
19	Simon ELF	S	R. Moffie
22	More on Subroutine Calling Conventions	C	W. Bowdish
24	2708 Eprom Programmer	S - H	B. Erick
28	On Standards	C	D. Jaeger
30	Some Thoughts on Device Independent I/O	C	W. Bowdish
49	1001 Options for the 1802	C - H	M. Franklin
55	To VIP an ELF, Part II	S	M. Frankln
60	1802 Editor/Assembler	C - S	G. Miller

MEMBERSHIP RENEWAL/CHANGE OF ADDRESS

The dues for the new club year, September 1979 to August 1980 are as follows:

Canada	$15.00
Elsewhere...............	$18.00

These new rates include provision for FIRST CLASS POSTAGE which should ensure that your copy of IPSO FACTO arrives as quickly as possible.

Please make cheque payable to "THE ASSOCIATION OF COMPUTER EXPERIMENTERS" and mail to c/o M.E. Franklin, Treasurer, A.C.E., 24 Duby Road, Acton, Ontario, CANADA L7J 2P1.

Do you have any friends? If so, they might well appreciate a gift subscription to IPSO FACTO.

New members are required to pay the full dues for the current year and will receive all the IPSO FACTO issues for the current year (September 1979 to August 1980). Back issues may be purchased for $30.00 (Issues 1-12, May 1977 - May 1979).

CHANGE OF ADDRESS

FIRST NAME

LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE

POSTAL CODE

ATTACH MAILING LABEL HERE

FOR CHANGE OF ADDRESS

NEW SUBSCRIPTION

FIRST NAME

LAST NAME

FIRST LINE OF ADDRESS

SECOND LINE OF ADDRESS

CITY & PROVINCE

POSTAL CODE