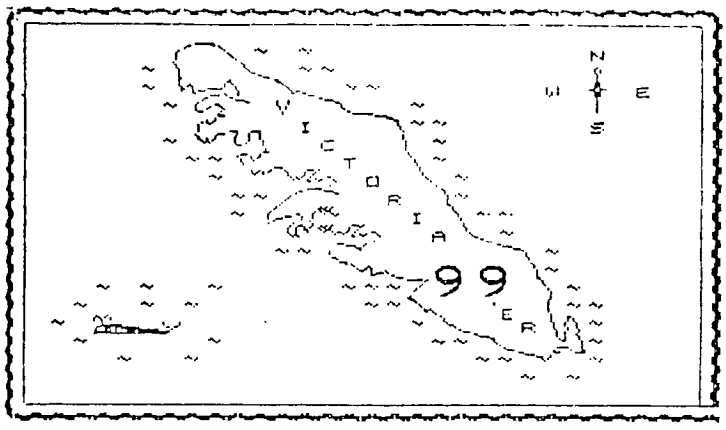

JULY 1985

THE CLUB HAS RECENTLY RECEIVED MANY BACK ISSUES OF MICROPENDIUM MAGAZINE AS WELL AS ITS CURRENT SUBSCRIPTION. THERE IS LOTS OF INTEREST IN THESE ISSUES AND MEMBERS SHOULD CHECK WITH THE LIBRARIAN AND CHECK THESE EXCELLENT MAGAZINES OUT.

IN THE LAST MONTH A COUPLE OF IN CONSOLE MEMORY EXPANSION SYSTEMS HAVE COME TO LIGHT FOR THE TI. THE MATCHBOX EXPANSION FROM AUSTRALIA USES 4 8K CMOS CHIPS AND THE EXISTING DECODING TO DO A 32K EXPANSION USING THE 8 BIT WIDE DATA BUS. TOP RADIO SUPPLY IN DETROIT HAS A $100 U.S. KIT OR $ 160 U.S. INSTALLED PRICE FOR EXPANSION ONTO THE 16 BIT DATA BUS. THIS SYSTEM ALSO APPEARS TO USE 8K CMOS CHIPS. NEITHER SYSTEM OFFERS BATTERY BACKUP. I WAS THINKING I MIGHT DO SOMETHING ALONG SIMILAR LINES, EXCEPT WITH BATTERY BACKUP AND A TOTAL OF 64K ALL ON THE 16 BIT DATA BUS. THIS WOULD ALLOW ME TO RUN FORTH IN THE WHOLE ADDRESS SPACE. THE SCHEME WOULD INVOLVE A NON PAGED 32K EXPANSION IN THE USUAL LOCATION WHICH WOULD ALLOW NORMAL OPPERATION OF THE CONSOLE AND ALL MEMORY MAPPED DEVICES AS WELL AS PERPHERIALS AND GROM. CRU INSTRUCTIONS WOULD PAGE OUT THE OTHER 32K BANK WHICH HAS ALL THE MEMORY MAPPED DEVICES. THIS 32 WOULD REMAIN ACTIVE UNTIL A CALL WAS MADE TO AN INTERFACE ROUTINE IN THE HIGH MEMORY. I INTEND TO FLOAT THIS DEVICE ON A SISTER BOARD WITH THE CPU ON BOARD AND DRIVE THE MOTHER BOARD FROM THIS BOARD. A FEW EXTRA SOCKETS AND SOME EXTRA CRU DECODING, WILL ALLOW EPROM PAGING AND EXTRA MEMORY IN THE PERPHERIAL ADDRESS BANK AT >4000 TO >5FFF .

I THINK A 16 BIT BUS UPGRADE WITH EXTRA MEMORY AND BATTERY BACKUP WOULD BE MUCH MORE FUN AND USEFULL THAN JUST STUFFING THE 32K CARD IN THE CONSOLE. A FEW OTHER CLUB MEMBERS HAVE EXPRESSED INTEREST IN SOME SORT OF SPARE CONSOLE HARDWARE HACKING AND THIS MAY BE A GOOD WAY TO GO.

RUMORS CONTINE TO FLY ABOUT THE ELUSIVE TI 99/8 BUTTERFLY. LAST REPORTED TO HAVE BEEN SPOTTED IN MISSOURI. HOWEVER TI OWNERS FROM THAT AREA ARE SAID TO BE TAKING A WAIT AND SEE ATTITUDE. AN IEEE PUBLICATION FROM MID 1984 STATED THAT THE 9900 HAD THE LARGEST 16 BIT SOFTWARE BASE BAR NONE. APPARANTLY THE 9995 PROCESSOR HAS (HAD?) THROUGH PUT SIMILAR TO A 68000 . IF THE 9900 WERE PUT ON THE 16 BIT BUS AS IN THE PREVIOUS HARDWARE SCHEME, SPEED WITH EXISTING SOFTWARE WOULD PROBABLY PICK UP 25% OR SO.

A LITTLE NOTE TO ALL MEMBERS ABOUT UPCOMING EXECUTIVE ELECTIONS IN OCTOBER. HERE IS YOUR OPPORTUNITY TO GET INVOLVED WITH YOUR CLUB. IF YOU WOULD LIKE TO DROP IN ON THE EXECUTIVE MEETINGS JUST CONTACT ONE OF THE EXECUTIVE FOR FURTHER DETAILS. THE EXECUTIVE MEET THE SECOND THURSDAY OF EACH MONTH AND WE WOULD WELCOME MEMBERS AT THE EXECUTIVE MEETINGS.

THE CLUB RAN AN AD IN THE PAPER PRIOR TO THE LAST MEETING AND SEVERAL TI ENTHUSISTS TURNED OUT IN RESPONSE TO THE AD. WE PLAN TO RUN ADDITIONAL NEWSPAPER ADS PERIODICALY TO LET TI USERS KNOW THAT THERE IS AN OFFICIAL TI USERS GROUP IN VICTORIA TO HELP THEM OUT.

THE FOLLOWING IS A REPRINT FROM ABOUT 1 1/2 YEARS AGO. STILL A FUN LITTLE PROGRAM. PERHAPS SOMEONE WOULD LIKE TO MODIFY IT SO THE PLAYER SCRAMBLES THE PUZZLE AND THE COMPUTER SORTS IT OUT ?

****

THE 9-PUZZLE
extended
basic

```
*******
*1 2 3*
*4 5 6*
*7 8  *
*******
```

by
TONY BIGRAS

```
100 DIM TILE(9)      * make array for tiles.
110 RANDOMIZE
120 CALL MAGNIFY(2) * set sprite size for dblsize i used sprites for the tiles.

130 RANDOM$="973815642137945962387194625491836275416235798"
140 RANDOM$=RANDOM$&"263751849361459278356921874631598724184935672"
150 RANDOM$=RANDOM$&"351764892345628719748961352896745321987654321"
160 RANDOM$=RANDOM$&"741852963238495617314796258248135697891743265"

    * make one big string with the scrambled patterns for the puzzles.

170 MOVES=-1         * this gets incremented before it is first displayed.
```

```
180 X(1),X(2),X(3)=73   *
190 X(4),X(5),X(6)=89   * sets the X and Y coordinates for the sprites
200 X(7),X(8),X(9)=105  * that are used to display the tiles.
210 Y(1),Y(4),Y(7)=105  *
220 Y(2),Y(5),Y(8)=121  *
230 Y(3),Y(6),Y(9)=137  *

240 CALL CLEAR
250 DISPLAY AT(24,5)BEEP:"release alpha-lock"

    * both the arrow keys and the joysticks are usable if alpha-lock is up!


260 CALL COLOR(14,16,7)
270 CALL CHAR(142,"00000000000000FFFFFFFFFFFFFFFF")

    * sets game color and initializes character patterns.
    * all values have been initialized *


280 FOR T=72 TO 112 STEP 8      *
290 CALL HCHAR(1+T/8,14,143,6) * draws a 6x6 white rectangle on the screen.
300 NEXT T                      *

310 CALL HCHAR(9,14,142,6)      *
320 CALL HCHAR(16,14,142,6)     * draws a red border around the rectangle.
330 CALL VCHAR(9,13,142,8)      *
340 CALL VCHAR(9,20,142,8)      *

350 FOR T=1 TO 8 :: CALL SPRITE(#T,T+48,2,240,100):: NEXT T
360 CALL SPRITE(#9,143,9,240,100)

    * sets sprite patterns to 1 thru 8 and a solid and holds them off screen.


370 JUMBLE$=SEG$(RANDOM$,(INT(RND*20)+1)*9-8,9)
380 FOR T=1 TO 9
390 TILE(T)=VAL(SEG$(JUMBLE$,T,1))
400 IF TILE(T)=9 THEN BLANK=T
410 NEXT T

    * randomly selects pseudo-random patern from RANDOM$ and loads the pattern
    * into the TILE() array.


420 FOR T=1 TO 9 :: CALL LOCATE(#TILE(T),X(T),Y(T)):: NEXT T

    * puts the sprites on the screen in scrambled pattern.


430 GOSUB 690 * start game enter play loop at end of loop to put MOVES on screen

440 CALL JOYSTK(1,K,S) * call JOYSTK to scan input from keys and from joysticks.

    * JOYSTK converts joystick input to call key(1,k,s) output.
```

```
450 IF S=0 THEN 440 * if no input then ask for input again.

460 IF K=18 THEN MOVES=-1 :: GOTO 370 * if fire button is hit or key Q is hit

    * then the player wants to end game or pick a different pattern, or both.


470 IF K<>0 AND K<>2 AND K<>3 AND K<>5 THEN 440 * if invalid key then ask again.

480 IF K<>5 THEN 520     * if input is not down then next check.
490 IF BLANK>6 THEN 440  * if down move not legal at this time then ask again.
500 TEMP=BLANK+3         * TEMP = new position of sprite #9 solid.
510 GOSUB 660            * go move the tile.

520 IF K<>0 THEN 560     *
530 IF BLANK<4 THEN 440  * same as above except in up direction.
540 TEMP=BLANK-3         *
550 GOSUB 660            *

560 IF K<>2 THEN 600                                  *
570 IF BLANK=3 OR BLANK=6 OR BLANK=9 THEN 440         * same but to the right.
580 TEMP=BLANK+1                                      *
590 GOSUB 660                                         *

600 IF K<>3 THEN 640                                  *
610 IF BLANK=1 OR BLANK=4 OR BLANK=7 THEN 440         * same but to the left.
620 TEMP=BLANK-1                                      *
630 GOSUB 660                                         *

640 GOTO 440             * this is the end of the play loop start over again.

650 GOTO 650             * this is left over from program development!

660 TILE(BLANK)=TILE(TEMP) * this is where the tile get moved around.
670 TILE(TEMP)=9           * have to keep track of what was and what will be.

680 CALL LOCATE(#TILE(BLANK),X(BLANK),Y(BLANK)):: CALL LOCATE(#TILE(TEMP),X(TEMP
),Y(TEMP)):: BLANK=TEMP

    * moving sprites around and keeping track of wher the blank tile is now.

690 MOVES=MOVES+1 :: DISPLAY AT(4,11):"MOVES";MOVES   * this is where we came to
700 CALL SOUND(40,1400,1,4000,5,3000,15)              * enter the play loop from
710 RETURN                                            * line 430

    * this shows moves makes noise and return from whence it came.


720 SUB JOYSTK(SIDE,KEY,STATUS)                       *
730 CALL JOYST(SIDE,X,Y)                              * this is where key and
740 IF X=0 AND Y=0 THEN STATUS=0 :: GOTO 840          * joystick input come in
750 STATUS=1                                          * and become output that
760 IF X=0 AND Y=4 THEN KEY=5 :: GOTO 840             * works like a
770 IF X=4 AND Y=4 THEN KEY=6 :: GOTO 840             * call key(1,k,s) statment
780 IF X=4 AND Y=0 THEN KEY=3 :: GOTO 840             *
790 IF X=4 AND Y=-4 THEN KEY=14 :: GOTO 840           * i already know how it
800 IF X=0 AND Y=-4 THEN KEY=0 :: GOTO 840            * works but you can figure
810 IF X=-4 AND Y=-4 THEN KEY=15 :: GOTO 840          * it out for yourself or
820 IF X=-4 AND Y=0 THEN KEY=2 :: GOTO 840            * give me a call if you
830 IF X=-4 AND Y=4 THEN KEY=4 :: GOTO 840            * have any questions about
840 CALL KEY(SIDE,K,S):: IF S=0 THEN 870              * this program 383-3946
850 IF SIDE=1 THEN KEY=K :: STATUS=S
860 IF SIDE=2 THEN KEY=K :: STATUS=S :: IF KEY=18 THEN KEY=11
870 SUBEND
```

THE GPL (GRAPHICS PROGRAMING LANGUAGE) SYSTEM IS DESIGNED SO THAT WITH THIS HARDWARE THE BUILT IN SOFTWARE ALLOWS ONE CARTRIDGE TO ACCESS THE DEVICES AND CALLS IN ANOTHER. THIS ALLOWS FOR EXAMPLE CONSOL BASIC TO ACCESS ALL OF THE PLUGGED IN MODULES CALL ROUTINES AND DEVICE NAMES AT ONE TIME. "MINIMEM1" , "MINIMEM2" AND "SPEECH" AND CALL PEEKV,POKEV,LOAD, ETC ARE ALL AVALIBEL FROM BASIC.

TI FORTH CAN ACCESS "MINIMEM1" AND "MINIMEM2" AND "SPEECH" AS DEVICES WITH NO NEED TO SWITCH ANYTHING OR TO MODIFY ANY CONSOLE HARDWARE. THE OPERATING SYSTEM IN THE CONSOLE HANDLES ALL ACCESS TRANSPARANTLY.

THIS WAS BUILT IN FROM DAY ONE WITH THE 99/4 AND IS ON MY PRE 1983 BLACK AND SILVER CONSOLE . I DONT KNOW FOR A FACT IF THIS IS ON THE NEWER MODELS BUT I SUSPECT THAT IT IS.

THE SOFTWARE DURING MODULE LIBRARY SELECTION FINDS ONLY THOSE PAGES THAT CONTAIN GROM OR GROM AND ROM COMBINED. THE SLOTS WITH ROM ONLY (THIRD PARTY STUFF) DO NOT COME UP ON THE MENU. MUCH LIKE THE POST 1983 CONSOLES.

THE VICTORIA COMPANY OSRAM INDUSTRIES IS CURRENTLY DEVELOPING AN INEXPENSIVE "SUPER WIDGET" TO TAKE FULL ADVANTAGE OF THIS IN CONSOLE SOFTWARE.

RADIO SHACK NOW HAS TEN FOOT EXTENSION CORDS FOR JOYSTICKS IN STOCK. PART # 276-1978 FOR $5.99 EACH.

THESE EXTEND THE FULL 9 PINS AND WILL WORK WITH TI JOYSTICKS OR WITH AN ADAPTER AND THE ATARI TYPE JOYSTICKS.

LOTS OF 128K CARDS ARE NOW AVAIBLE FOR THE TI AND MANY WITH EVEN MORE MEMORY CAPABILITY WILL PROBABLY SOON BE AVAIBLE. SOFTWARE MAY BE A LITTLE HARDER TO FIND FOR THEM ,BUT I SUSPECT A FEW RAMDISKS AND PRINT SPOOLERS WILL BECOME AVAIBLE. THER PRICES SHOULD CONTINUE TO COME DOWN TO. PRESENTLY 256 K OF DYNAMIC RAM (WITHOUT SUPPORT CIRCUTS) COSTS ABOUT $50.00 CANADIAN.

THE CURRENT ISSUE OF BYTE MAGAZINE DISCUSSES A COPROCESSOR BOARD FOR THE IBM PC. THIS IS A FULL 32 BIT PROCESSOR WITH 1 MBYTE AND WITH POWER AND SPEED EQUIVELENT TO MANY MINI-COMPUTERS . THE ITEM COSTS $1500.00 U.S. AND JUST USES THE EXPANSION SLOT IN THE PC AND RUNS ITS OWN LANGUAGES IN ITS OWN MEMORY. SOUNDS LIKE A NEAT THING TO DROP INTO A TI AIRCONDITIONER. THE PC VERSION WOULD NOT BE USABLE DIRECTLY BUT THE IDEA CERTAINLY IS PORTABLE.

THE NEW LOGO IS THE WORK OF CHRIS MACMURCHIE USING THE GRAPHX PROGRAM AND THE OLD LOGO FOR INSPIRATION. CHRIS ALSO HAS A NEAT LITTLE GAME WHICH WE WILL GET INTO THE NEXT NEWLETTER.

THE LAST PAGE IS A PROGRAMING AID CHEAT SHEET FOR ALL THOSE DETAILS THAT WE HAVE TROUBLE REMEMBERING. TOM HAS FISHED THIS OUT OF THE NEWSLETTERS THAT THE CLUB RECEIVES IN EXCHANGES WHITH OTHER TI CLUBS AROUND THE CONTINENT.

CORCOMP DISK CONTROLLER UPDATE

I recently received a letter from someone at Corcomp letting me know that not many cards had the "no disk" or "no drive" problem which I mentioned in my review two months ago. Apparently there were enough, however, that a slight design change was made so that no cards since May should have that problem. The problem itself is apparently easy to fix: a simple resistor change. Good to hear that Corcomp is still out there, still working at it...

Johan.

Edmonton 99'er UG
PO Box 11983
Edmonton, Alberta
T5J 3L1

VICTORIA 99'er GROUP
1369 Finlayson St
Victoria B.C.
V8S 2W5 canada