AND THE WINNER IS.....
MICKEY SCHMITT !

MICKEY'S ENTRY WAS DONE PRIMARILY TO SET AN EXAMPLE. WELL, SHE WON AND IT WAS A LANDSLIDE. THANKS TO ALL WHO ENTERED. NEXT MONTH I'LL HAVE THE LOGO PROPERLY PRESENTED.

ESTABLISHED 1985
WEST PENN 99'ERS
SUPPORTING THE TI-99/4A COMPUTER

ISSUE #6 June 1989

FOR THE RECORD

by Frank N. Zic
(Acting Secretary)

The June meeting was called to order by President Mickey Schmitt at 7:15 PM. She mentioned that 20 more MacPaint picture disks were put in the library along with some other nice Utility programs. A list of these new additions was issued to all present. Gene gave a brief Corresponding Secretary's report and added that the plans for the fall TI Carlisle Computer Faire are progressing satisfacturally(My that's a long word that doesn't say a lot). Scott Coleman passed around a map of the new location for our future meetings. We will meet at the St. Stevens Byzantine Catholic Church. It is located between the Norwin Shopping Center and Spitz Auto on Bethal road. We thank our present host for the nice meeting place we have been using. New members who join our club at this time will pay only 1/2 the normal yearly dues for the remaining portion of this year.

Old business-Eric Zeno showed us the finished Print File needed to start manufacture of his new Zeno board, shouldn't be too long now folks. The board will allow components to be added on the board that will support Speech,XBasic and a Clock.This can then be placed within the TI console for a neat package. Isn't it ironic that Mickey herself should win the contest to develop a new club symbol for our news letter? She submitted her design just to show support for the project and none of the 3 submitting authors names were known prior to the voting. Each applicant won a nice gift for their entry. Look for the new club symbol on our future newsletters.

New business-Mickey's new Adventure Reference Guide is now being sold by Asgard for $14.95 + S&H. She also mentioned that Lynn Gardner's new adventure game Zoom Flume is also complete and will also be offered soon. Mickey gave a very favorable report on the Lima Faire. A VCR tape of the proceedings is available, ask for it and check things out - it's the next best thing to being there. A huge raffle was held with about 16 prizes distributed. It should be mentioned that a good part of the success for the raffle was the friendly threat, all who did not purchase a ticket would have to go home with a large number of old computer magazines under their arm. Of course part of the prizes were from odds and ends in our clubs equipment cupboard that we were trying to lighten for our move to the new location. Many thanks go to many busy hands that pitched in to remove our supplies. Tonight was the last night to take advantage of the 50 per cent discount on Great Lakes software. Our cassette members were happy to hear that a Cassette Reference Booklet, using Mickey's previouslly written articles, will soon be developed by Mike Wright of the Boston User's group. When it is ready, Barry Traver of Computer Shopper will print it up. Keep in mind that any member wishing to check out the clubs Oscilloscope need only ask for the equipment. May the good 4's be with you.

Respectfully submitted,
Frank N. Stein, I mean Frank N. Zic

WEST PENN 99'ERS CLUB INFORMATION

NEXT MEETING DATE: JULY 18, 1989

MEETING LOCATION: ST. STEPHEN'S BYZANTINE CATHOLIC CHURCH

JUST OFF ROUTE 30
BETHEL ROAD, NORWIN

TIME OF MEETING: 7:00 P.M.

LIST OF WEST PENN OFFICERS FOR 1989

PRESIDENT:	MICKEY	335-0163
VICE PRESIDENT:	SCOTT	523-3754
TREASURER:	JAN	863-1575
RECORDING SEC:	ED	864-4924
CORRESPONDING SEC:	GENE	829-0469
LIBRARIAN:	ROB	864-1233
NEWSLETTER EDITOR:	JOHN	527-6656

GENERAL ITINERARY OF THE CLUB'S MEETING

6:45 P.M.	DOORS OPEN
7:00 P.M.	GENERAL MEETING
7:45 P.M.	DEMOS AND NEW INFO
8:45 P.M.	HARDWARE CLASS
8:45 P.M.	INTRO TO FORTH
8:45 P.M.	TIPS FOR BEGINNERS
8:45 P.M.	USING YOUR CASSETTE
11:00 P.M.	DOORS CLOSE

MEETING HIGHLIGHTS FOR THIS MONTH

JIFFY FLYER 3.0, DEMO BY MICKEY SCHMITT

NEW ZOOM FLUME, WRITTEN BY LYNN GARDNER

ASGARD'S NEW BATCH IT!, DEMO BY ROB EKL

LIBRARY "DEMO OF THE MONTH" BY ROB EKL

LATEST SOFTWARE DEMOS BY JOHN WILLFORTH

RENEW YOUR MEMBERSHIP DUES!

$15.00 PER YEAR FOR INDIVIDUAL / FAMILY
$10.00 PER YEAR FOR JUST THE NEWSLETTER

TREASURER'S REPORT FOR JUNE 20,'89

FROM JAN TRAYERS

6/20	CASH ON HAND	$ 50.00
"	LIBRARY SALES	20.00
"	MICROPENDIUMS	27.25
	DISKS & CASES	33.00
	DUES	60.00
	RAFFLE	49.00
	GR. LAKES SFT WR	252.50
	TOTAL	$501.75
7/7	DEPOSIT	- 451.75
7/7	CASH ON HAND	50.00
6/20	BANK BALANCE	$1303.77
6/20	DISKS (SCOTT)	-215.00
		1088.77
6/20	POSTAGE (JOHN)	- 70.00
		1018.77
7/1	MICROPENDIUM	- 30.00
		988.77
7/1	GR.LAKES SFT WR	- 252.50
		736.27
7/7	DEPOSIT	+ 451.75
	BALANCE	1188.02
	TOTAL CASH BALANCE	$1238.02

**** ATTENTION ALL CLUB MEMBERS ****

IF YOU ARE NOT SURE HOW TO FIND OUR NEW MEETING LOCATION OF ST. STEPHENS BYZANTINE CATHOLIC CHURCH, PLEASE CALL MICKEY SCHMITT AT 335-0163 BEFORE JULY 17, AND ASK TO BE MET AT THE UNITED PRESBYTERIAN CHURCH OF THE COVENANT, (OUR OLD LOCATION), SO THAT YOU WILL BE ABLE TO FOLLOW SOMEONE WHO ALREADY KNOWS HOW TO GET THERE.

DOCUMENTATION FOR CASSETTE INDEX UTILITY VERSION 1.0 (XB)
CREATED BY HARRY BRASHEAR OF THE WESTERN NY 99'ERS
MODIFIED (V.2.0) BY MICKEY SCHMITT OF THE WEST PENN 99'ERS

PROGRAM NAME: CS1*INDEX (SEE LIBRARIAN)
PROGRAM DOCS: CS1*DOCS (DOCUMENTATION)
PROGRAM FILES: MOCKFILE1 AND MOCKFILE2 (BASIC SET-UPS)

THIS EXTENDED BASIC PROGRAM WILL ALLOW YOU TO PRINT OUT A NEAT LITTLE INSERT FOR YOUR CASSETTE TAPES, JUST LIKE THE ONES THAT COME WITH THEM.

ALL PRINTER CODES ARE EPSON COMPATIBLE. YOU MAY WISH TO MODIFY PROGRAM LINE 570 TO OPEN YOUR PARTICULAR PRINTER. CURRENTLY THIS PROGRAM OPENS THE FOLLOWING PRINTER DEVICE: (RS232/2.DA=8.BA=9600).

WHEN YOU FIRST BOOT UP THE PROGRAM, YOU'LL BE ASKED IF YOU WANT TO USE FILES OR INPUT THE INFORMATION DIRECTLY. (MORE ON THIS LATER). YOU WILL THEN BE ASKED FOR YOUR FULL NAME, FOR GOOD REASON. THAT LITTLE FOLD OVER THE OTHER SIDE OF YOUR TAPE WILL BE PUT TO GOOD USE BY STATING WHO THE TAPE BELONGS TO. (THE DEFAULT SPACE LIMIT IS 28 CHARACTERS).

NEXT YOU WILL BE ASKED IF YOU WANT ONE OR BOTH SIDES OF THE TAPE INDEXED. (THE DEFAULT IS 2 FOR BOTH SIDES).

YOU WILL THEN BE ASKED FOR THE CASSETTE TITLE (THE DEFAULT SPACE LIMIT IS 25 CHARACTERS), AND ANY SPECIAL INFORMATION THAT YOU MAY WISH TO RECORD. (AGAIN, THE DEFAULT SPACE LIMIT IS 25 CHARACTERS).

FINALLY, YOU WILL BE ABLE TO INPUT YOUR CASSETTE PROGRAMS, USING ANY CODING SYSTEM YOU MAY PREFER. (THE DEFAULT SPACE LIMIT IS 27 CHARACTERS).

SHOULD YOU MAKE A MISTAKE WHILE ENTERING YOUR PROGRAM DATA JUST HIT THE (ENTER KEY) AT A BLANK LINE AND YOU WILL THEN BE PROMPTED BACK TO YOUR PREVIOUS ENTRY.

WHEN YOU HAVE COMPLETED ALL OF YOUR PROGRAM DATA INPUTS, JUST TYPE (Q) AND HIT THE (ENTER KEY). (NOTE: THE (Q) WILL NOT APPEAR ON YOUR PRINTED INSERT).

LAST, YOU WILL BE ASKED IF YOU WOULD LIKE TO FILE OR PRINT YOUR INPUTS. (THIS SPECIAL FEATURE WAS INCLUDED BY HARRY SO THAT PEOPLE WITHOUT PRINTERS COULD FILE THE INFORMATION TILL THEY COULD BORROW A PRINTER OR CON A FRIEND INTO PRINTING THE INSERTS).

NOTE: IF YOU HAVE FILED INFORMATION STORED ON YOUR DISK YOU COULD JUST TYPE (F) AT THE FIRST PROMPT, WHICH WOULD PROMPT YOU FOR THE FILENAME YOU HAVE YOUR DATA STORED UNDER. ONCE YOU HAVE ENTERED YOUR FILENAME, JUST SIT BACK AND WATCH YOUR PRINTER GO INTO ACTION. (PROVIDED YOU DID REMEMBER TO TURN YOUR PRINTER ON).

YOU MAY ALSO MAKE MULTIPLE COPIES BY SIMPLY REPRINTING. THE "BUTTONS" ARE JUST FOR FUN, AND YOU WILL HAVE TO HOLD YOUR KEY PRESSES LONGER BECAUSE OF THEM.

SIDE 1	SIDE 2
123456789012345678901234567	123456789012345678901234567
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0
1	1
2	2
3	3
4	4
523456789012345678901234567	523456789012345678901234567

SIDE 1: 123456789012345678901234 5 123456789012345678901234 5

SIDE 2: 123456789012345678901234 5 123456789012345678901234 5

THIS TAPE IS FROM THE LIBRARY OF 1234567890123456789012345678

MOCKFILE1

SIDE 1	SIDE 2
PROGRAM 001 - COUNTER 000 B	PROGRAM 016 - COUNTER 000 B
PROGRAM 002 - COUNTER 010 B	PROGRAM 017 - COUNTER 010 B
PROGRAM 003 - COUNTER 020 B	PROGRAM 018 - COUNTER 020 B
PROGRAM 004 - COUNTER 030 B	PROGRAM 019 - COUNTER 030 B
PROGRAM 005 - COUNTER 040 B	PROGRAM 020 - COUNTER 040 B
PROGRAM 006 - COUNTER 050 B	PROGRAM 021 - COUNTER 050 B
PROGRAM 007 - COUNTER 060 B	PROGRAM 022 - COUNTER 060 B
PROGRAM 008 - COUNTER 070 B	PROGRAM 023 - COUNTER 070 B
PROGRAM 009 - COUNTER 080 B	PROGRAM 024 - COUNTER 080 B
PROGRAM 010 - COUNTER 090 B	PROGRAM 025 - COUNTER 090 B
PROGRAM 011 - COUNTER 100 B	PROGRAM 026 - COUNTER 100 B
PROGRAM 012 - COUNTER 110 B	PROGRAM 027 - COUNTER 110 B
PROGRAM 013 - COUNTER 120 B	PROGRAM 028 - COUNTER 120 B
PROGRAM 014 - COUNTER 130 B	PROGRAM 029 - COUNTER 130 B
PROGRAM 015 - COUNTER 140 B	PROGRAM 030 - COUNTER 140 B

SIDE 1: WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 001--015

SIDE 2: WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 016--030

THIS TAPE IS FROM THE LIBRARY OF THE WEST PENN 99'ERS LIBRARY

MOCKFILE2

```
100 !************************
110 !                        *
120 !CASSETTE INDEX UTILITY*
130 !          BY            *
140 !    HARRY BRASHEAR      *
150 !   WESTERN NY 99ERS     *
160 !                        *
170 !************************
180 !                        *
190 !   VERSION 2.0 (XB)     *
200 ! ENHANCED AND MODIFED   *
210 !          BY            *
```

continued on page 4

```
220 !    MICKEY SCHMITT     *
230 !   WEST PENN 99'ERS    *
240 !     JUNE 13 1989      *
250 !                       *
260 !***********************
270 ON WARNING NEXT :: CALL
MAGNIFY(3)
280 GOTO 290 :: OPTION BASE
1 :: A$,A1$,A2$,AN$,F$,FN$,L
$,L1$,N$,PF$,T1$,T2$ :: I,J,
K,S,S1,S2,SS :: CALL CHAR ::
 CALL CLEAR :: CALL COLOR ::
 CALL SCREEN :: CALL SPRITE
:: CALL KEY :: DIM SIDE1$(18
),SIDE2$(18):: !@P-
290 !
300 CALL CLEAR :: CALL SCREE
N(15):: CALL CHAR(132,"3F60C
F9FBFBFBFBFBFBFBF9FCF603FF
C06F3F9FDFDFDFDFDFDFDFDF9F30
6FC")
310 CALL SPRITE(#1,132,2,160
,47):: CALL SPRITE(#2,132,2,
160,119):: CALL SPRITE(#3,13
2,2,160,190)
320 CALL COLOR(13,7,15):: CA
LL CHAR(128,"000000FFFF",129
,"0000FF00FF00FF"):: L$=RPT$
(CHR$(128),28):: L1$=RPT$(CH
R$(129),28)
330 DISPLAY AT(24,3):"RECORD
  PLAYBACK   DUPE"
340 DISPLAY AT(1,1):"CASSETT
E TAPE--INDEX UTILITY":L1$:
: :"WOULD YOU LIKE TO PRINT
FROM": :"FILES OR DIRECT INP
UT? F/D D"
350 ACCEPT AT(7,28)SIZE(-1)B
EEP VALIDATE("FD"):F$ :: IF
F$="F" THEN 730 ELSE CALL SP
RITE(#1,132,10,160,47)
360 DISPLAY AT(4,1):"PLEASE
ENTER YOUR FULL NAME:":L$ ::
 ACCEPT AT(7,1)SIZE(28)BEEP:
N$ :: DISPLAY AT(8,1):L$
370 DISPLAY AT(4,1):"ENTER N
UMBER OF TAPE SIDES:2":L$ ::
 ACCEPT AT(4,28)BEEP VALIDAT
E("12")SIZE(-1):SS
380 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 1" :: AC
CEPT AT(9,1)SIZE(25)BEEP:T1$
 :: DISPLAY AT(4,1):"SPECIAL
 INFORMATION:  SIDE 1"
390 ACCEPT AT(10,1)SIZE(25)B
EEP:A1$
400 IF SS=2 THEN DISPLAY AT(
11,1):L$ ELSE 420
410 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 2":L$ ::
 ACCEPT AT(12,1)SIZE(25)BEEP
:T2$ :: DISPLAY AT(4,1):"SPE
CIAL INFOMATION:  SIDE 2" :
: ACCEPT AT(13,1)SIZE(25)BEE
P:A2$
420 DISPLAY AT(16,1)BEEP:"AL
L INFORMATION CORRECT? Y/N"
430 CALL KEY(0,K,S):: IF S=0
 THEN 430 ELSE IF K=89 THEN
440 ELSE IF K=78 THEN 270 EL
SE 430
440 A$="SIDE ONE" :: GO SUB
670
450 FOR I=5 TO 19 :: J=I-4 :
: S1=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE1$(J):: IF SIDE
1$(J)="Q" THEN 480 ELSE IF S
IDE1$(J)="" THEN I=I-2
460 IF I<4 THEN I=5
470 NEXT I
480 IF SS=1 THEN 520 ELSE A$
="SIDE TWO" :: GO SUB 670
490 FOR I=5 TO 19 :: J=I-4 :
: S2=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE2$(J):: IF SIDE
2$(J)="Q" THEN 520 ELSE IF S
IDE2$(J)="" THEN I=I-2
500 IF I<4 THEN I=5
510 NEXT I
520 DISPLAY AT(24,3)BEEP:" F
ILE    REDO     PRINT" :: CA
LL SPRITE(#1,132,2,160,47)
530 DISPLAY AT(24,10)SIZE(10
):"   REDO" :: CALL KEY(5,K,
S):: CALL SPRITE(#1,132,2,16
0,47,#2,132,2,160,119,#3,132
,2,160,190)
540 IF S=0 THEN 550 ELSE IF
K=70 THEN 690 ELSE IF K=80 T
HEN 560 ELSE IF K=82 THEN 66
0 ELSE 530
550 FOR I=1 TO 80 :: NEXT I
:: CALL SPRITE(#1,132,10,160
,47,#2,132,10,160,119,#3,132
,10,160,190):: GOTO 530
560 CALL SPRITE(#1,132,2,160
,47,#2,132,2,160,119,#3,132,
10,160,190)
570 OPEN #1:"RS232/2.DA=8.BA
=9600" :: PRINT #1:CHR$(27);
CHR$(64);CHR$(27);CHR$(85);C
HR$(1);
580 PRINT #1:CHR$(27);CHR$(4
8);CHR$(15);RPT$("-",70):"¦"
;TAB(5);"SIDE 1";TAB(35);"¦"
;TAB(39);"SIDE 2";TAB(70);"¦
" :: PRINT #1:RPT$("-",70)
590 PRINT #1:"¦";TAB(35);"¦"
;TAB(70);"¦"
600 FOR I=1 TO 17 :: IF SIDE
1$(I)="Q" THEN SIDE1$(I)=""
610 IF SIDE2$(I)="Q" THEN SI
DE2$(I)=""
620 PRINT #1:"¦   ";SIDE1$(I
);TAB(35);"¦   ";SIDE2$(I);T
AB(70);"¦" :: NEXT I
630 PRINT #1:RPT$("-",70):"¦
";"   SIDE 1:    ";T1$;" ";A
1$;TAB(70);"¦":"¦";TAB(70);"
¦":"¦";"   SIDE 2:    ";T2$;
" ";A2$;TAB(70);"¦":RPT$("-"
,70)
640 PRINT #1:"¦";TAB(70);"¦"
:"¦";"   THIS TAPE IS FROM T
HE LIBRARY OF ";N$;TAB(70);"
¦":"¦";TAB(70);"¦":RPT$("-",
70):: CLOSE #1
650 GOTO 530
660 FOR I=1 TO 15 :: SIDE1$(
I),SIDE2$(I)="" :: NEXT I ::
 T1$,T2$,A1$,A2$="" :: GOTO
270
670 DISPLAY AT(4,1):"" :: DI
SPLAY AT(1,1):"ENTER PROGRAM
 NAMES: Q(QUIT)":L$:TAB(11);
A$
680 FOR I=5 TO 19 :: DISPLAY
 AT(I,1):")" :: NEXT I :: RE
TURN
690 CALL SPRITE(#1,132,10,16
0,47):: DISPLAY AT(1,1):"FIL
E NAME:   DSK1." :: ACCEPT A
T(1,19)BEEP SIZE(10):FN$
700 FN$="DSK1."&FN$ :: OPEN
#2:FN$,SEQUENTIAL,INTERNAL,O
UTPUT,VARIABLE 80
710 PRINT #2:N$ :: PRINT #2:
T1$,A1$ :: PRINT #2:T2$,A2$
720 FOR I=1 TO 15 :: PRINT #
2:SIDE1$(I),SIDE2$(I):: NEXT
 I :: CLOSE #2 :: GOTO 530
730 CALL SPRITE(#2,132,10,16
0,119):: DISPLAY AT(7,1):""
:: DISPLAY AT(4,1):"FILE NAM
E:   DSK1.":L$ :: ACCEPT AT(
4,19)SIZE(10):FN$ :: FN$="DS
K1."&FN$
740 CALL SPRITE(#2,132,10,16
0,119):: OPEN #1:FN$,SEQUENT
IAL,INTERNAL,INPUT ,VARIABLE
 80
750 INPUT #1:N$ :: INPUT #1:
T1$,A1$ :: INPUT #1:T2$,A2$
760 FOR I=1 TO 15 :: IF EOF(
1)THEN 780 ELSE INPUT #1:SID
E1$(I),SIDE2$(I)
770 NEXT I
780 CLOSE #1 :: GOTO 560
790 !@P+
800 END
```

TetrIs by Steve Karasek of the St. Louis 99'ers (June 1989)

(Ed. note) The game TETRIS was originally a game written by a young Russian, and was imported into the U.S.A. by the English on the P.C.. It is definitely a very engrossing game, if you'll just give it a chance.

In this game, random shapes made up of four square blocks will drop from the top of a 10-column wide section of the screen. The object of the game is to rotate the shapes and move them from side to side so that when they drop in place among the previous shapes, they will form solid lines across the screen. When a line is formed, it will disappear from the screen, and partial lines on top of it will drop down. The game ends when the shapes have piled up to the top of the screen.

When the game starts, you are asked for a starting level from 1 to 9. At level 1, the shapes drop very slowly, giving you plenty of time to move them into position. At level 9, they move quite fast. The scoring is higher for each succeeding level, however, once you have experience, you will want to start at one of the higher levels so that you score more points.

For each game, the screen will start out empty, with the current level to the left and the score to the right. The high score for this session will be displayed above the current score. Press **ENTER** to start the game.

You may use either your right hand or your left hand to control the shapes. Make sure the **ALPHA LOCK** key is depressed. With the right hand, press **J** to move the shape to the left, **K** to rotate it 90 degrees counter-clockwise, or **L** to move it to the right. Press **;** to pause and catch your breath. Press **any** of the **other** keys to resume play. When you have the shape in position, press the **space bar** to drop it rapidly into place (but be careful!).

If you are left-handed, use **F** to move to the right, **D** to rotate, **S** to move to the left, and **A** to pause. The **space bar** still drops the shape.

Points are scored for each shape. The higher the shape lands or is dropped from, the higher the score, so it pays to move it quickly into position and then drop it with the space bar. Points are also scored for each line that is formed. The higher the level, the higher the score for each shape or line.

The current level will increase for every 5000 points scored. If you want to increase the level at any time during the game, press **U** or **R**. If you want to quit, press **Q**.

WARNING! THIS PROGRAM MAY BECOME HABIT FORMING.

I've typed in this program, and I you'll find it on page 6. Yes! It doesn't even fill the whole page. Steve is really quite a programmer. It will be worth your efforts to type the program in and try it. John Willforth

---

Unlike most of the cartoons that you see in a computer newsletter, the one below has nothing to do with computers. I don't even normally read the funnies section of the newspaper, but last week this caught my eye and it remineded me of what I see going on today. Maybe you can relate.

ANIMAL CRACKERS **By Roger Bollen**

```
1 REM * COMPUTER BRIDGE *
2 REM * JUNE 1989 ISSUE *
100 DISPLAY ERASE ALL AT(8,1
2):"Tetris" :: DISPLAY AT(10
,3):"(C) 1989 Steven Karasek
"
110 PRINT "STARTING LEVEL (1
-9)";:: INPUT E :: E=INT(E):
: IF E<1 OR E>9 THEN 110 ELS
E E=10-E
120 DIM Z$(23),Z(26),A(18,3)
,B(18,3):: RANDOMIZE :: C$="
JKL; UQSDFAR" :: Z(24)=4095
:: CALL MAGNIFY(4):: CALL CL
EAR :: FOR I=0 TO 6
130 READ N(I),C(I):: CALL CO
LOR(I+8,2,C(I)):: NEXT I ::
FOR I=0 TO 18 :: FOR J=0 TO
3 :: READ A(I,J),B(I,J):: NE
XT J :: NEXT I
140 FOR I=68 TO 143 :: READ
X$ :: CALL CHAR(I,X$):: NEXT
 I :: CALL CHAR(41,"FFFFFFFF
FFFFFFFF")
150 FOR I=0 TO 23 :: Z$(I)=R
PT$(")",10):: Z(I)=2049 :: N
EXT I :: V=E :: D,P=24 :: U=
0 :: GOSUB 450 :: CALL VCHAR
(1,12,41,240)
160 CALL KEY(0,M,W):: IF W<>
1 THEN 160
170 P=0 :: Q=4 :: J=INT(RND*
7):: S=J*2 :: J8=J*8+89 :: I
F J>3 THEN S=S-1+2*(J-4)
180 GOSUB 470 :: T=0 :: X=1
:: Y=Q*8+81 :: CALL SPRITE(#
1,K,C(J),X,Y)
190 IF Z(0)AND 2^(Q+Y1)OR Z(
X2)AND 2^(Q+Y2)OR Z(X3)AND 2
^(Q+Y3)OR Z(X4)AND 2^(Q+Y4)T
HEN 430
200 FOR I=1 TO V :: CALL KEY
(0,M,W):: IF M<0 THEN 350 EL
SE ON POS(C$,CHR$(M),1)+1 GO
TO 350,210,280,230,340,250,3
30,440,210,280,230,340,330
210 Q=Q-1 :: IF Z(P)AND 2^(Q
+Y1)OR Z(P+X2)AND 2^(Q+Y2)OR
 Z(P+X3)AND 2^(Q+Y3)OR Z(P+X
4)AND 2^(Q+Y4)THEN Q=Q+1 ELS
E Y=Y-8
220 CALL LOCATE(#1,X,Y):: GO
TO 350
230 Q=Q+1 :: IF Z(P)AND 2^(Q
+Y1)OR Z(P+X2)AND 2^(Q+Y2)OR
 Z(P+X3)AND 2^(Q+Y3)OR Z(P+X
4)AND 2^(Q+Y4)THEN Q=Q-1 ELS
E Y=Y+8
240 GOTO 220
250 Y1=2^(Q+Y1):: Y2=2^(Q+Y2
):: Y3=2^(Q+Y3):: Y4=2^(Q+Y4
):: GOSUB 450 :: P=D-X4
260 IF (Z(P)AND Y1 OR Z(P+X2
)AND Y2 OR Z(P+X3)AND Y3 OR
Z(P+X4)AND Y4)=0 THEN P=P+1
:: GOTO 260
270 P=P-1 :: CALL LOCATE(#1,
P*8+1,Y):: GOTO 360
280 S=S-1 :: T=T-1 :: IF T<0
 THEN T=N(J)-1 :: S=S+N(J)
290 GOSUB 470
300 IF (Z(P)AND 2^(Q+Y1)OR Z
(P+X2)AND 2^(Q+Y2)OR Z(P+X3)
AND 2^(Q+Y3)OR Z(P+X4)AND 2^
(Q+Y4))=0 THEN CALL PATTERN(
#1,K):: GOTO 350
310 S=S+1 :: T=T+1 :: IF T=N
(J)THEN T=0 :: S=S-N(J)
320 GOSUB 470 :: GOTO 350
330 CALL KEY(0,M,W):: IF W<>
0 THEN 330 ELSE V=V+(V<9)1)::
GOSUB 460 :: GOTO 350
340 CALL KEY(0,M,W):: IF W<>
1 THEN 340
350 NEXT I :: P=P+1 :: IF P+
X4>=D THEN 370
360 X=X+8 :: CALL LOCATE(#1,
X,Y):: GOTO 200
370 IF (Z(P)AND 2^(Q+Y1)OR Z
(P+X2)AND 2^(Q+Y2)OR Z(P+X3)
AND 2^(Q+Y3)OR Z(P+X4)AND 2^
(Q+Y4))=0 THEN 360 ELSE P=P-
1 :: GOSUB 450
380 D=MIN(D,P):: FOR I=0 TO
3 :: W=Q+B(S,I):: M=P+A(S,I)
:: Z(M)=Z(M)+2^W :: Z$(M)=SE
G$(Z$(M),1,W-1)&CHR$(J8)&SEG
$(Z$(M),W+1,10)
390 CALL HCHAR(M+1,W+11,J8):
: NEXT I :: CALL DELSPRITE(#
1):: FOR I=MIN(P+3,23)TO P S
TEP -1 :: IF Z(I)<4095 THEN
420 ELSE J=I :: M=I-1
400 Z(J)=Z(M):: Z$(J)=Z$(M):
: DISPLAY AT(J+1,10):Z$(J)::
 IF Z(J)>2049 THEN J=J-1 ::
M=M-1 :: GOTO 400
410 U=U+INT(500/V):: GOSUB 4
60 :: I=I+1 :: P=P-1 :: D=D+
1
420 NEXT I :: GOTO 170
430 H=MAX(H,U):: DISPLAY AT(
1,20):USING "##########":H ::
 CALL DELSPRITE(#1):: GOTO 1
50
440 DISPLAY ERASE ALL:"HIGH
SCORE IS";MAX(U,H):: END
450 U=U+INT((24-P)*100/V)
460 DISPLAY AT(3,20):USING "
##########":U :: V=MIN(V,MAX(
1,9-INT(U/5000))):: DISPLAY
AT(3,4)SIZE(2):10-V :: RETUR
N
470 X2=A(S,1):: X3=A(S,2)::
X4=A(S,3):: Y1=B(S,0):: Y2=B
(S,1):: Y3=B(S,2):: Y4=B(S,3
):: K=GR+S*4 :: RETURN
480 DATA 2,15,2,7,2,14,1,16,
4,11,4,4,4,5
490 DATA 0,0,0,1,0,2,0,3,0,1
,1,1,2,1,3,1,0,0,0,1,1,1,1,2
,0,2,1,1,1,2,2,1
500 DATA 0,1,0,2,1,0,1,1,0,1
,1,1,1,2,2,2,0,1,0,2,1,1,1,2
,0,1,1,0,1,1,1,2
510 DATA 0,1,1,1,1,2,2,1,0,0
,0,1,0,2,1,1,0,2,1,1,1,2,2,2
520 DATA 0,0,1,0,1,1,1,2,0,1
,0,2,1,1,2,1,0,0,0,1,0,2,1,2
,0,2,1,2,2,1,2,2
530 DATA 0,2,1,0,1,1,1,2,0,1
,1,1,2,1,2,2,0,0,0,1,0,2,1,0
,0,1,0,2,1,2,2,2
540 DATA FFFFFFFF,,FFFFFFFF,
,0F0F0F0F0F0F0F0F,0F0F0F0F0F
0F0F0F,,
550 DATA FFFFFFFF0F0F0F0F,,0
000000F0F0F0F,,00000000F0F
0F0F,0F0F0F0F,F0F0F0F0F0F0F0
F0,,0F0F0F0FFFFFFFFF,,F0F0F0
F,
560 DATA 0F0F0F0F0F0F0F0F,,0
000000F0F0F0F0,F0F0F0F,0F0F
0F0F0F0F0F0F,,F0F0F0F0F0F0F0
F,,0F0F0F0FFFFFFFFF,,0000000
0F0F0F0F,
570 DATA 0F0F0F0F0F0F0F0F,0F
0F0F0F,00000000F0F0F0F,,FFFF
FFFF0F0F0F0F,,F0F0F0F,,00000
0000F0F0F0F,,F0F0F0F0F0F0F0F
,F0F0F0F
580 DATA F0F0F0F0FFFFFFFF,,0
000000F0F0F0F,,0F0F0F0F0F0F
0F0F,0F0F0F0F,F0F0F0F,,FFFFF
FFF,,F0F0F0F0F0F0F0F0,,,0F0F
0F0F,F0F0F0F0F0F0F0F,F0F0F0F
590 DATA 00000000FFFFFFFF,,F
0F0F0F0F0F0F0F0,,0F0F0F0F0F0
F0F0F,0F0F0F0F,,F0F0F0F,FFFF
FFFF0F0F0F,,F0F0F0F,,0F0F0F
0F,,F0F0F0F0F0F0F0F,F0F0F0F
```

The following are the non-printing ASCII codes for the TI-99/4a in the p-Code/Pascal system. These were established with the following Pascal program;

```
Program Chart

Var CH:Char;
Begin
 Write('Enter a character.  To exit enter 1');
 Readln(CH);
 While (CH<>'1') Do
  Writeln('The ASCII value is ',Ord(CH));
  Writeln;
  Write('Enter a character.  To exit enter 1');
  Readln(CH);
 End;{of while loop}
End.{of Chart program}
```

With this program in operation it means that the p-Code/Pascal operating system is engaged so there are certain entries that will break the program or suspend its operation so these keystrokes were mentioned in this article.

Below is the listing of the non-printing ASCII codes. If the heading says "Ctrl" it means that the Control key is held down while the other corresponding key is pressed. In the case of the heading marked "Fctn" the Function key is held down while the other corresponding key is pressed. It did not make any difference if the alpha lock key was engaged or not.

Ctrl key held down.

A = 1; B = 2; C breaks the program; D = 4; E = 5; F = 6; G = 7; H = 8; I = 9; J = 10; K = 11; L = 12; M = ?; N = 14; O = 15; P = 16; Q = 17; R = 18; S = 19; T = 20; U = 21; V = 22; W = 23; X = 24; Y = 25; Z = 26; (.period) = 27; (,comma) = 0; (-,/) = 187; 1 = 177; 2 = 178; 3 = 179; 4 = 180; 5 = 181; 6 = 182; 7 = 183; 8 = 30; 9 = 31; Zero = 176; (+,=) = 176.

Fctn key held down.

B = 190; D = 137(cursor right); E = 193(cursor up); H = 191; J = 192; K = 193; L = 194; M = 195; N = 196; Q = 197; S = 136(cursor left); V = 127; Y = 198; X = 138(cursor down); (<,,) = 184; (>,.) = 185; (:,;) = 189; 1 = 131; 2 = 132; 3 = ?; 4 = stops program; 5 = suspends program; 6 = ?; 7 = ?; 8 = ?; 9 = 143; zero = 188; (+,=) = 133; (-,/) = 186.

Thank you Ed Hall for this contribution to the West Penn 99'ers. JFW

A COMPARISON OF LANGUAGES
by Ed Hall

Since the theme for this journal is languages, I thought I would show a comparison of how to perform a particular routine by writing the steps involved in three different languages. I chose TI BASIC, TI EXTENDED BASIC (XB) , and ASSEMBLY using the TI EDITOR/ASSEMBLER (E/A). In order to run the programs I have included, it will be necessary to have a joystick which operates as number 1. To see the entire comparison it will also be necessary to have on hand EXTENDED BASIC AND EDITOR/ASSEMBLER packages.

In writing programs it is important to plan out the functions that you wish to perform as completely as possible. Figuring out how to fill in all the holes and squish bugs will be tasking enough while writing, so we should start with a pretty well-defined set of parameters. The parameters I have chosen will remain the same for each program. This is a must in comparisons. First I want to place a 0 somewhere near the middle of the screen. Next I want the 0 to be movable by the joystick in all eight directions. I also want the 0 to leave a trail of asterisks behind as it moves, and flash as it rests in one spot. Last, I want the asterisks to change color by stepping through all sixteen available upon the press of the fire button. Throughout each program, I want all action to be able to occur simultaneously.

Now that we have the basics chosen, let's move on to the writing. Remember that these programs are just what came out of MY head, so later if you would like, go ahead and write some of your own to compare their operation with these. The program listings can be found at the end of the article. The first program is in TI BASIC and as you will notice, it consists of 26 statement lines. Go ahead and enter this program into your system and save it. Then see how it runs and check to see if it meets all the parameters above.

Lines 100 and 110 set variables which will later be used to place the 0/*s on the screen. Line 120 invokes a resident subroutine to check the joystick port for activity. In basic "CALL" is used to run resident subprograms similar to using GOSUB ... to run subprograms within your main program. In this instance the 1 within the parentheses tells the routine to check joystick number 1 and the X and Y will be where the results are placed. Line 130 is used to allow the picking up of fire button activity. This is very poorly documented, but to check for the activation of a fire button, you must perform a key scan using split keyboard and search for an occurrence of 18 in the K variable. Line 140 places an asterisk on the screen. Line 150 modifies the variable X1 which is used in the horizontal placement of the 0/*s. The variables returned by the JOYST routine are 4/0/-4 so X/4 is used to modify X1. Line 160 is a trap in case X1 leaves the boundaries of the screen. Line 170 is the same as 150 except it modifies the Y1 variable. Line 180 is a trap for Y1 as 160 was for X1. Line 190 places the 0 using the modified coordinates. Line 200 checks to see if the fire button was pressed and if not sends the operation back to 120 where the loop continues. If the fire button was pressed, it falls through to 210. Line 210 increments the variable C and Line 220 checks to insure the color will be in bounds. If it is, it skips to line 240, but if not C is reset to 1 in 230 before moving to 240 where the COLOR subroutine is CALLed. Line 250 sends the program back as did 200. Lines 260 - 300 insure X1 stays within the bounds of the screen and allow direct wraparound. Lines 310 - 350 do the same for the Y1 coordinate.

At this time let's start up the EXTENDED BASIC. After starting up XB, load in the program from above and run it. Could you tell a difference in speed? You should have noticed a marked improvement in the movement of the 0 and the change in color, even though it's the same program that was run in BASIC. Still it can be

continued on page 9

continued from page 8.

improved. XB adds the ability to create multiple lines which is both a memory saver and time saver in running a program. Our next step is to take advantage of this new ability. The program lines under listing #2 are more compressed, yet the program runs the same way. This XB program takes only 9 lines to equal 26 in BASIC.

Line 100 now performs the operations of two lines from the first program. Notice that C had to be initialized in this file. In the first program C was able to start off as 0 but in this one it has to be greater than 0 to work. Line 110 represents lines 120 - 160 of the first program. Line 120 represents lines 170 and 180. Line 130 represents lines 190 - 230. Line 140 is equal to lines 240 and 250. Line 150 represents 260 - 290. 160 equals 300 and line 170 replaces 310 - 340 with 180 equalling 350.

Once you've keyed this one in, see how it compares to the other one when run. Did it run faster? It should have, but it still isn't really fast, is it? Well maybe we can speed it up a little more in a while.

Let's regress for a moment and bring back up BASIC with either the MINIMEM or E/A in place and load in the original program from above. If you are using BASIC while the MINIMEM or E/A cartridge is in place additional resident subroutines are available. Among them is an alternate way to place characters on the screen called POKEV. CALLing POKEV(place,character ASCII#+96) writes the character directly into VDP RAM where the screen information is. Place is a number from 0 to 767 which directly represents a screen location from upper left to lower right respectively. The, character is whatever you choose to write to the screen. To alter the program to support this, change these two lines:

```
140 CALL POKEV(Y1*32+X1-1,13
8)
190 CALL POKEV(Y1*32+X1-1,14
4)
```

The expression is equivalent to using HCHAR even though the position is represented by a single number. Before we move on, make a mental note of the speed with which the 0 can be moved around the screen.

Now that we've seen what a little programming in BASIC or XB can do to meet our parameters, let's see how to accomplish them in ASSEMBLY. First off when using ASSEMBLY, we must take into account many more things than we do in BASIC. We need to access certain memory locations directly and place and manipulate data in precise ways. There are subroutines but they must called only after we have data in the correct places. You'll notice first that the SOURCE code has almost three times as many lines as the BASIC version. However, when this is ASSEMBLED it will run much faster and take up less space in memory. To run the following program you will have to type the whole program into a D/V 80 file as it is written and ASSEMBLE it using the E/A. If you are unfamiliar with the procedures for running the ASSEMBLER, after inputting and saving the text SOURCE code, perform these steps: 1) Plug in the E/A cartridge and press 2 twice to bring up the E/A menu; 2) With the E/A disk in drive #1, press 2 for ASSEMBLE; 3) Press Y at the LOAD ASSEMBLER? prompt and enter the file name you saved the text under as the SOURCE FILE NAME; 4) Enter the name you want the finished file to be called under the OBJECT FILE NAME; 5) If you want a listing of the code, enter a filename or printer under LIST FILE NAME; 6) Enter at a minimum the letter R under options. You might also add the letters CSL but do not add L or S if you've not given a name for the list. The above steps can be found in chapter 2.2 of the E/A manual.

Before we get into the program itself let me give a little extra info for the beginners to ASSEMBLY. The syntax of BASIC lines must be met in writing programs in those languages. In ASSEMBLY this same rule applies, however the syntax is much more straightforward. For an explanation of

continued on page 10

continued from page 9

the syntax refer to chapter 3.3 of the E/A manual. Whether you understand how the program works or not, go ahead and try to run it so you can compare its operation with those written in the BASIC languages. The source code is listed as PROGRAM LISTING #3 at the end of this article. When you type it in please observe spaces where they exist and do not place spaces where they don't exist in the code.

It would be much too time consuming and take up too much newsletter to describe in detail the lines of code for this ASSEMBLY language program. Instead I will describe the functions of areas in a more general sense. If further details are needed perhaps a discussion of the code can be arranged at one of our meetings.

The first few lines define the name the program will run under when loaded and set up the parameters for use within the program. Two resident routines will be used. These are VDP Single Byte Write (VSBW) and Keyboard SCAN (KSCAN). The first will place all characters on the screen and the second accepts input from the joystick and keyboard. Since ASSEMBLY does not automatically allow for a FCTN-4 to break program execution, I have allowed for the pressing of letter A to cause the program to stop. As used in this program, the letter A will actually send the computer to the title screen.

On the extreme left of the file you will see words like JOY and KEY. These are known as labels and must start in the first column of the text. The two lines at JOY set up values that will place the initial 0 near the center of the screen. The KEYSRC lines calculate the value to use in the placement of the 0. The first time through the values loaded at JOY are used but from then on the values created by the program are used. The lines at KEY actually place the 0 on the screen and read the input from the keyboard or joystick. The only inputs accepted from the keyboard are Q which equals the fire button and A which causes program termination. The only other inputs accepted are from the joystick. One of the inputs accepted is the fire button. Remember, the fire button is being used to change the color of the asterisks. The lines at KEY2 do the color change based on values from KEY. The lines at MOVCUR place the asterisks on the screen in place of the 0 and determine how to move the cursor if a direction has been given from the joystick. The lines at ADDY, XCHK and ADDX set up the values to be used by the next pass of KEYSRC in placing the next 0 on the screen. These routines also take care of wraparound. The last line of program execution is the EXIT routine which tells the computer to go back to the title screen if the letter A was pressed in KEY above. The statement END under the EXIT label has no function within the program, but is necessary to tell the ASSEMBLER to stop processing the file.

In order to run this program once it is ASSEMBLED, you will need to use option 3, LOAD AND RUN from the E/A menu. At the FILE NAME? prompt enter the drive and filename you have given the ASSEMBLED code and then press ENTER again. At the PROGRAM NAME prompt enter the name JOY. The program will immediately start. After running it for a while to check operation, press A to stop and return to the title screen. If you would like to run this from basic, you will need to change 3 lines in the source code and reASSEMBLE the file. The 3 changes are:

```
KEY    change >3000   to >9000
KEY2   change >0385   to >0311
MOVCUR change >2A00   to >8A00
```

With E/A or MINI-MEMORY in place you can use the following program in BASIC to run the ASSEMBLY program:

```
10 CALL INIT
20 CALL LOAD("DSK?.???????")
30 CALL LINK("JOY")
```

* ?.??????? is the drive and filename you gave to the ASSEMBLED file.

As you can see, there are more ways to write a program than there are languages. You can even combine a BASIC program with some ASSEMBLY routines held in memory and perform CALL LINKs to use them. Let me close with a thought for those who own MINI MEMORIES. How could the ASSEMBLY source code be changed to be able to type it into the line-by-line ASSEMBLER?

continued on page 11

continued from page 10

PROGRAM LISTING #1 (TI BASIC):

```
100 Y1=12
110 X1=16
120 CALL JOYST(1,X,Y)
130 CALL KEY(1,K,S)
140 CALL HCHAR(Y1,X1,42)
150 X1=X1+(X/4)
160 IF (X1)32)+(X1(1)THEN 26
0
170 Y1=Y1-(Y/4)
180 IF (Y1)24)+(Y1(1)THEN 31
0
190 CALL HCHAR(Y1,X1,48)
200 IF K()18 THEN 120
210 C=C+1
220 IF C(17 THEN 240
230 C=1
240 CALL COLOR(2,C,1)
250 GOTO 120
260 IF X1)32 THEN 290
270 X1=32
280 GOTO 120
290 X1=1
300 GOTO 120
310 IF Y1)24 THEN 340
320 Y1=24
330 GOTO 120
340 Y1=1
350 GOTO 120
```

PROGRAM LISTING #2 (EXTENDED BASIC):

```
100 C=2 :: Y1=12 :: X1=16
110 CALL JOYST(1,X,Y):: CALL
 KEY(1,K,S):: CALL HCHAR(Y1,
X1,42):: X1=X1+(X/4):: IF (X
1)32)+(X1(1)THEN 150
120 Y1=Y1-(Y/4):: IF (Y1)24)
+(Y1(1)THEN 170
130 CALL HCHAR(Y1,X1,48):: I
F K=18 THEN C=C+1 :: IF C=17
 THEN C=1
140 CALL COLOR(2,C,1):: GOTO
 110
150 IF X1)32 THEN X1=1 ELSE
X1=32
160 GOTO 120
170 IF Y1)24 THEN Y1=1 ELSE
Y1=24
180 GOTO 110
```

PROGRAM LISTING #3 (E/A SOURCE CODE):

```
       DEF  JOY
       REF  VSBW,KSCAN
KEYBRD EQU  >8374
JOYSTK EQU  >8376
YOFFST DATA 32
KEYCK1 DATA >0100
KEYCK2 DATA >1200
DIRCHK DATA >0000
JOY    LI   R8,12
       LI   R9,16
KEYSRC MOV  R9,R0
       MOV  R8,R6
       MPY  @YOFFST,R6
       A    R7,R0
KEY    LI   R1,>3000
       BLWP @VSBW
       LI   R3,>0100
       MOV  R3,@KEYBRD
       BLWP @KSCAN
       MOV  @KEYBRD,R3
       SWPB R3
       CB   R3,@KEYCK1
       JEQ  EXIT
       CB   R3,@KEYCK2
       JNE  MOVCUR
       AI   R5,16
       CI   R5,240
       JLT  KEY2
       CLR  R5
KEY2   MOV  R0,R4
       LI   R0,>0385
       MOV  R5,R1
       SWPB R1
       BLWP @VSBW
       MOV  R4,R0
MOVCUR LI   R1,>2A00
       BLWP @VSBW
       MOV  @JOYSTK,R3
       CB   R3,@DIRCHK
       JEQ  XCHK
       JLT  ADDY
       DEC  R8
       CI   R8,-1
       JGT  XCHK
       AI   R8,24
       JMP  XCHK
ADDY   INC  R8
       CI   R8,24
       JLT  XCHK
       AI   R8,-24
XCHK   SWPB R3
       CB   R3,@DIRCHK
       JEQ  KEYSRC
       JGT  ADDX
       DEC  R9
       CI   R9,-1
       JGT  KEYSRC
       AI   R9,32
       JMP  KEYSRC
ADDX   INC  R9
       CI   R9,32
       JLT  KEYSRC
       AI   R9,-32
       JMP  KEYSRC
EXIT   BLWP @0
       END
```

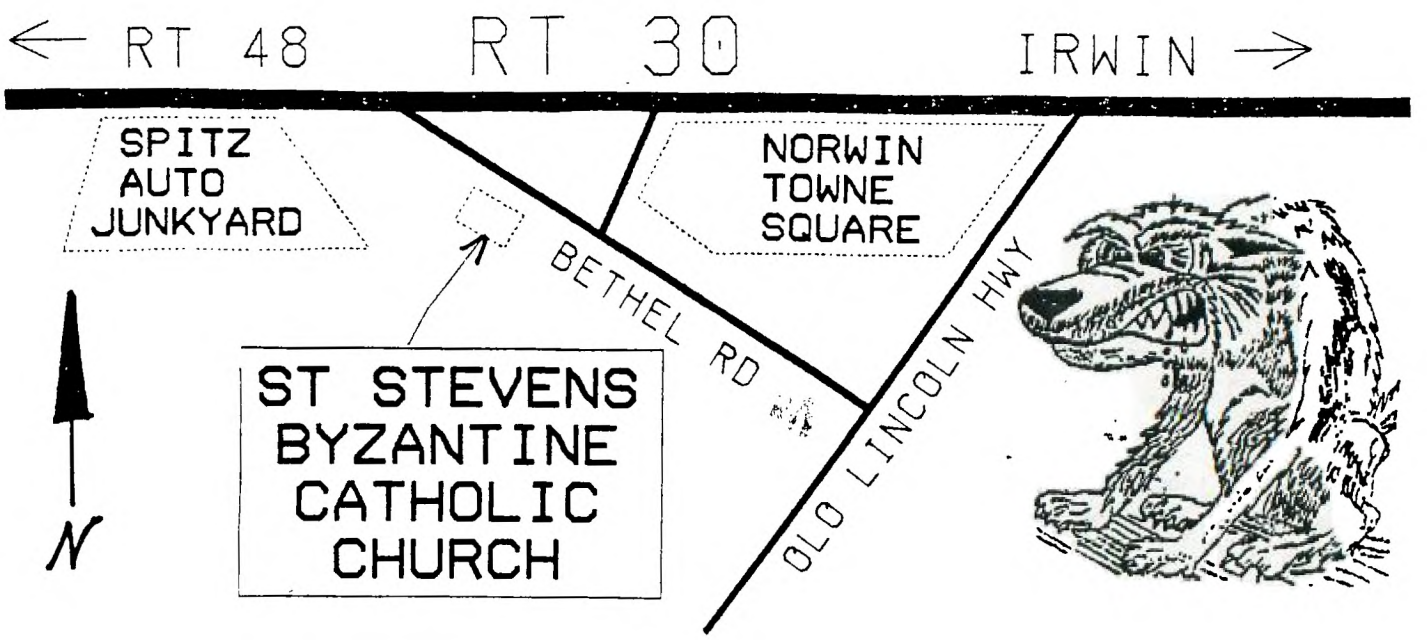
MUST SELL ENTIRE SYS.
TI console, PEB w/32K, *CorComp RS232, TI Disk* Cont. and Disk drive. External CorComp Dual-½ Height DSDD drives, Panasonic KXPI090 Pr., Speech Syn., Joy Stick cassette cables, many cart.'s, XBASIC, TE-II TI-LOGO II,TI-Writer, Disk Manager II, E/A, Super Extended Basic, Draw n' Plot, and many manuals. Call/Write:
Dwite Deithorne
849 Highland Ave.
Greensburg, PA
(412) 832- 9405
Many other items. B.O.
****

MUSIC PRO..Dave Caron, Lucie Dorais, and the Ottawa TI99/4A U.G.... A word processor for music. Create music by typing notes on staff. Save music scores, or print in sheet music form on Epson compat. printer. Music Pro can automatically take care of different voices w/ different durations & play them simultaneously. Comes with a detailed manual and sample songs, a keybrd overlay and kyb. note map. Req. 32K, XB, and disk system. $17.55
ASGARD SOFTWARE
P.O. BOX 10306
ROCKVILLE, MD 20850
(703) 255-3085
add &.75 S/H in USA
****

If you don't get every thing you want, think of the things you don't get that you don't want!
****

He who has a thing to sell and goes and whispers in a well, is not so apt to get the dollars as he who climbs a tree and **H O L L E R S.**
****

THE JULY MEETING WILL BE HELD AT ST. STEVENS BYZANTINE CATHOLUC CHURCH AT THE LOCATION INDICATED BELOW. SEE MICKEY'S OFFER BOTTOM RIGHT OF PAGE 2 FOR ANY WHO FEAR THEY WILL NOT BE ABLE TO FIND THE NEW MEETING PLACE. WE WILL HAVE BASICALLY THE SAME ADVANTAGES THAT WE HAVE ENJOYED IN THE PAST. I'D LIKE TO THANK SCOTT COLEMAN FOR THE MAP SHOWN BELOW.
SEE YOU ALL THERE ON JULY 18TH AT 7:00 PM AND DON'T BE LATE! OR WE'LL HAVE TO SIC THE JUNKYARD DOG ON YOU!



WEST PENN 99'ERS

% JOHN F. WILLFORTH
R.D. #1 BOX 73A
JEANNETTE, PA
15644

ISSUE NUMBER 7
JULY-1989

FIRST, FAST AND RELIABLE airmail par avion

EDMONTON 99'ers
P.O. BOX 11983 EDMONTON
ALBERTA. CANADA. T 5J3L1

NEW MEETING LOCATION ABOVE!