Happy Thanksgiving from the VAST Users Group

Valley of the Sun Computer User Group

NEWSLETTER

VOL. 3 NOVEMBER 14, 1987 NO.11

****

CONTENTS

VAST BBS.......437-4335

8711 (Ø94)
Vast

SECRETARY'S SLATE

MINUTES

VAST User Group
October 10,1987

The meeting was called to order by Bob Nixon, President, at 10:02. Only 15 members were in attendance.

Bob passed out a questionnaire concerning vacant committee positions. It is hoped that members of the group will pick a committee that they would enjoy working with rather than being appointed to a position. Results of this questionnaire are unknown at the time this is being written.

In the area of "what is happening in the local computer world" arena, the topic of "The ONE Computer Show" was brought up. A three day computer show, running Oct. 7-10 is being held at the local Civic Plaza. Several members indicated they had attended the show. Reactions about the show ranged from slightly boring to quite interesting. If you were wondering what is happening right now in the computer industry, this show was the place to find the most current product lines.

Rene LeBlanc gave a talk on surge protectors. Rene had the most unfortunate experience of receiving a rather devastating "lightning attack" a few weeks back, common for the Phoenix area during this time of the year.

The majority of the electronic devices in his house suffered total failure as a result of this lightning strike. Most everything ranging from outdoor patio lights to modems were placed into a comatose condition. He was happy to report that although his Amiga was pronounced one of the casualties, his 99/4A survived the incident. Rene advised the members that a small investment into a few surge protector power strips would be money well spent.

Bob Nixon continued the meeting by announcing that plans should be made concerning a membership drive. Although the reasons are unknown, speculation was made concerning the low attendance of the past several meetings. It is possible that the recent change in the group's meeting location may be partly responsible for the attendance record. It was decided that a mailing should be done to remind non-attending members of the new location, time, date, etc.

Several members requested that a current membership list be distributed amongst the group. Newer members, when confronted with a question or problem, often do not know who to turn to for help. This list should be available at the November meeting.

The meeting concluded with a demonstration by Stu Olson in the area of "other computers". Stu brought one of his IBM compatible computers along with the "hood" popped so all could peek (and not poke) at the innards of the computer. Several software demos were done and a long question and answer period ended the presentation. Members attending the demonstration seemed genuinely interested and have no doubt, broadened their computer knowledge.

The meeting closed at approximately 11:20 A.M.

Stu Olson, Secretary,
VAST User Group

THIS MONTHS PROGRAM

BURGLAR ALARM

By Rick Lumsden
Winnipeg, Manitoba, Canada

(Reprinted from the <u>NORTH EASTERN 99'ers Newsletter</u> -- April, 1987)

The program listed allows you to use your spare TI CONSOLE as a burglar alarm with very little investment except for a bit of time.

The actual program is very simple and can be modified to suit you own particular needs. This particular version has a lot of statements that allow you to see what is going on in the program while running a demonstration, however they can be removed quite easily with no effect on the operation of the program. Just a few cautions though. Understand the program first before making any drastic changes. The other precaution is not to use your perimeter loop on the same joystick "direction" as the entry keyswitch. (eg. If you use the UP position for the keyswitch do not use this direction for the perimeter loop even if it is the opposite joysticks)

The program is set to use the UP position of joystick 2 for the entry keyswitch and the DOWN position of joystick 1 for the perimeter loop. It is also possible to use the other joystick directions (with appropriate program mods) to have more than one loop. Remember, this program will run as a stand-alone routine but is intended to be modified or totally rewritten by yourself to suit your particular job. The intention of this program is to be as simple as possible and not require any peripherals or modules. Most of us have a second console so here is a good use for it other than a paper weight.

To set up the alarm you will need the following:

1. TI console

2. Normally open magnetic or pushbutton switches for each door on the perimeter loop. (Radio Shack #49-495 or #49-497) with changes to the program (using the fire buttons and other positions) you may add other protection loops but you must insure that you have one switch per loop when using the normally closed switches. You may use as many switches of the normally open version on the loop as you wish.

3. Entry keyswitch (Radio Shack #49-515) or a hidden SPST toggle switch.

4. An audio amplifier and speaker(s) (your stereo amplifier will work just fine but the alarm will only be sounded in the house)

5. A cable to hook the audio output from the console to the amp. (If you have a monitor cable these will work fine. Some are available for the TI from Super Valu Stores for $10.95)

6. Joystick connector (Radio Shack #276-1538)

7. Hook-up wire

To run a simple demonstration of the program you will need two joysticks and your tv or monitor. First, you may want to set the delay variables in lines 150 and 160. Line 160 is the exit delay variable. This allows you time to leave the house after you turn on the keyswitch. If you mount the keyswitch outdoors,

THIS MONTHS PROGRAM CONTINUES

then set this variable to 1. The variable in line 150 sets the entry delay. This one allows you time to enter your home and disarm the system with the keyswitch before the alarm sounds. Remember to set this one on the fast side because it also delays in the event of a break-in.

When you type RUN, the words "PLEASE REMOVE ALPHA LOCK" and "PRESS 'C' TO CONTINUE" appear. Follow the insturctions and next comes "PERIMETER CHECK (Y/N)?". If you press "Y" the program jumps to line 700 and checks joystick 1 for any openings in the protection loop. If an opening is found (such as J1 in the center position) the program sounds a warning and tells you to check and remedy the situation. Do this by moving J1 to the down position and holding it there. Now push the "R" key and the program goes back to line 310 and sounds the OK chime.

The word "UNARMED" appears and tells you that the system is now ready for input from the keyswitch. When you turn the keyswitch on (by holding J2 in the UP position and J1 in the DOWN position) the program goes to the exit delay loop. This loop allows you to leave your home without triggering the system. Once this times out the program begins looping and checking each of the joysticks for a change in state.

If J1 suddenly becomes open the program moves to the entry delay loop. This delay allows you to enter your home and disarm the system with the keyswitch without setting it off. If the timer times out (eg. break-in) the program now sounds the alarm. You can simulate this by letting J1 return to the center position. Even if you were to close the door now it is too late, the timer is running down and the only way to stop it is to disarm the system.

Any number of changes and additions can be made to the program limited only by your imagination and your requirements. The intention of this routine is to give you an idea of what is possible and also to be as simple as possible. There are also heat detectors available that work on the normally open and normally closed switch principals so a fire alarm can also be added.

```
 ------------
 \ 1 2 3 4 5 /
  \ 6:7:8 9 /        JOYSTICK PORT
   - :::: --
     ::::               PINOUTS
     ::::_________
     :::         :   1 No Conn.
     :::         :   2 Right Grd.
     :::_____    :   3 Up
     ::     :    :   4 Fire Buttons
     ::  KEY-SWCH :  5 Left
     ::     :    :   6 No Conn.
     ::_____:    :   7 Left Grd.
     :           :   8 Down
     :__PER.LOOP__:  9 Right
```

```
100 REM BURGLAR ALARM PROGRA
M
110 REM FOR THE TI HOME COMP
UTER
120 REM A PUBLIC DOMAIN PROG
RAM
130 REM WRITTEN BY R.A.LUMSD
EN- WINNIPEG,MANATOBA,CANADA
140 REM 85/11 HUG-TIBBS
150 ENDEL=1000
160 EXDEL=1000
170 SKIPD=1
180 CALL CLEAR
190 PRINT "PLEASE REMOVE ALP
HA LOCK"
200 PRINT
210 PRINT
220 PRINT "PRESS 'C' TO CO
NTINUE"
230 CALL KEY(3,M,N)
```

CONTINUED PAGE 7 >

The Elements of Basic

Part 4

by Dave Howell

Courtesy "Erie 99'er Newsletter"

COMMANDS and STATEMENTS

Commands are different from statements. Commands are not part of the program, and they do not have line numbers. Instead, commands instruct the computer to do specific tasks. The TI BASIC commands are:

BREAK
BYE
CALL CLEAR
CALL CHAR
CALL COLOR
CALL GCHAR
CALL HCHAR
CALL JOYST
CALL KEY
CALL SCREEN
CALL SOUND
CALL VCHAR
CLOSE
CON

CONTINUE
DELETE
DIM
DISPLAY
EDIT
END
LET
LIST
NEW
NUM
NUMBER
OLD
OPEN
PRINT

PRINT#
RANDOMIZE
REM
RES
RESEGUENCE
RESTORE
RUN
SAVE
STOP
TRACE
BREAK
UNTRACE

TI BASIC statements, on the other hand, are usually found in program lines. They are:

BREAK
CALL CHAR
CALL CLEAR
CALL COLOR
CALL GCHAR
CALL HCHAR
CALL JOYST
CALL KEY
CALL SCREEN
CALL SOUND
CALL VCHAR
CLOSE
DATA

DEF
DIM
DISPLAY
ELSE
END
GOSUB
GOTO
IF THEN
INPUT
INPUT#
LET
NEXT
ON GOSUB

ON GOTO
OPEN
OPTION BASE
PRINT
PRINT#
RANDOMIZE
READ
REM
RESTORE
RETURN
STEP
STOP
UNTRACE

The commands and statements above are known as "reserved words." You can't use these words as variables in a program.

Many of the statements are identical to certain commands. Those statements become commands when used in the immediate mode. Compare the following:

Program Mode:

Statement:
```
     10   PRINT  "HELLO  THERE!"
```
Statement:
```
     20 CALL CLEAR
```
Statement:
```
     30 END
```

Immediate Mode:

Command:
```
     PRINT "HELLO THERE!"
```
Command:
```
     CALL CLEAR
```

VARIABLES

A variable can be defined as a name that can srepresent a value. Variables consist of a letter followed by additional letters and/or numbers. The following example program uses the variables A and B:

```
10 LET A=5.0
20 LET B=7.0
30 LET A=A+B
```

The variable A is initially assigned a value of 5.0 and B is assigned a value of 7.0. In line 30, the variable A is assigned a new value equal to the sum of variables A and B, which is 12. The previous value of 5.0 for A is erased.

The LET statement in the program above is used to assign a value to a variable. However, the LET statement need not actually be used in such programs on many computers including the TI. Both of the following statements have the same effect:

COMPUTER TUTOR CONTINUES.....

```
10 LET A=5
20 A=5
```

TI BASIC allows any group of up to 15 characters to be used as a variable name - as long as the first character is a capital letter and the name does not duplicate any of the reserved words (commands and statements listed above and the "functions" found below).

The following are examples of valid BASIC variable names in TI BASIC:

A A7 B2A AMOUNT PRICE CRAZY

The following are NOT valid variable names:

1A 3M 25PP7 LET RUN COS

Variable names can also be used to represent strings of alpha characters. These are known as string variables. To represent alpha characters, variable names must be followed by a string character ($). For example:

D$ D3$ PEP$ ADA$ MEN$ T777$

```
10 LET POPS$="DAVE"
20 X$="GRAMCRACKER"
```

Quotes are used around the strings much the same way as with PRINT statements.

Reserved words in TI BASIC also include the following functions:

ABS
ASC
ATN
CHR$
COS

EOF
EXP
INT
LEN
LOG

POS
RND
SEG$
SGN
SIN

SQR
STR$
TAB
TAN
VAL

D.H.

THIS MONTH'S PROGRAM CONTINUES FROM PAGE 5

```
240 IF N=0 THEN 230
250 IF M<>67 THEN 230
260 CALL CLEAR
270 PRINT "PERIMETER CHECK(Y
/N)?"
280 CALL KEY(3,L,T)
290 IF T=0 THEN 280
300 IF L=89 THEN 730
310 IF L<>78 THEN 280
320 CALL SOUND(1000,440,0,33
0,5)
330 CALL CLEAR
340 PRINT "UNARMED"
350 CALL JOYST(2,X,Y)
360 IF Y<>4 THEN 350
370 IF SKIPD>1 THEN 400
380 GOSUB 660
390 SKIPD=SKIPD+1
400 CALL JOYST(1,A,B)
410 IF B=-4 THEN 350
420 CALL CLEAR
430 PRINT "ALARM TRIPPED"
440 PRINT
450 PRINT "ENTRY DELAY INITI
ATED"
460 FOR ENTRDEL=1 TO ENDEL
470 NEXT ENTRDEL
480 CALL JOYST(2,X,Y)
490 IF Y=0 THEN 330
500 FOR LOOP=1 TO 5
510 FOR SIREN=700 TO 900 STE
P 10
520 CALL SOUND(-99,SIREN,0)
530 NEXT SIREN
540 FOR SIREN=900 TO 700 STE
P -12
550 CALL SOUND(-99,SIREN,0)
560 NEXT SIREN
570 NEXT LOOP
580 CALL CLEAR
590 PRINT "ALERT !!!!!!"
600 PRINT
610 PRINT
620 PRINT "ALARM TRIPPED"
630 PRINT
640 PRINT "PLEASE RESET"
650 END
660 CALL CLEAR
670 PRINT "EXIT DELAY INITIA
TED"
680 FOR DELAY=1 TO EXDEL
690 NEXT DELAY
700 CALL CLEAR
710 PRINT "ARMED"
720 RETURN
730 CALL CLEAR
740 CALL JOYST(1,A,B)
750 IF B=-4 THEN 320
760 CALL SOUND(1000,-2,0)
770 PRINT "BREAK IN PERIMETE
R CIRCUIT"
780 PRINT
790 PRINT "PLEASE CHECK"
800 PRINT
810 PRINT "PRESS  'R'  TO RE
CHECK       PERIMETER"
820 CALL KEY(3,K,S)
830 IF S=0 THEN 820
840 IF K=82 THEN 730
850 IF K<>82 THEN 820
```

EOF

NOVEMBER, 1987

IKES SUMMER REVIEW

LET'S DO IT!

By now all of you faithful readers of this Newsletter must be seriously wondering about the change in direction that this Group has taken recently. For the benefit of any newcomers, as well as our returning winter visitor members and any other members, here is a brief run down of what has been done and some of the reasoning behind it.

During this past summer, we found that because of the long delays in getting the 'New' Myarc computer out, the drastic reduction in price of most of the other computer systems and especially the availability of low cost IBM compatibles (IBM clones) a substantial number of members of the VAST 99 User Group had lost interest in the 99/4A and had switched to one of these other systems. The great surprise was, that almost all the Group's so called 'brains' had deserted the good old 99/4A in this manner. These were the same people, who had been very active in the Group and really had been the driving force in holding it together. They had written programs, given demonstrations, developed and built hardware projects, held classes in Programming and obtained all 99/4A related materials from such places as Compu-Serv or Genie, etc., etc.. Without further input and/or participation by these members, a quick death of the Group appeared assured.

Response to a questionnaire sent out to all members and former members let us know that there was considerable interest in keeping the Group alive however, even if it meant that in order to do so, other makes of computers would have to be supported. A lot of the 'Clone' owners felt that they could remain active in such a restructured Group and build upon the friendships developed during the TI-99/4A years. By staying or returning to the Group, their vast knowledge of the 99/4A system would not be lost, but still be available to the 'Hard Core' 99/4A users.

So there you have it. A brand new Constitution was written and approved. New Officers were elected and installed. Committee Chairpersons appointed and at this time we are in the process of getting additional Committee members interested in the various functions of the Group. In the past these functions were almost always 'handled' by just one and usually the same person. By establishing Committees, we hope to spread out these jobs and prevent 'Burn out', while getting a better functioning User Group, where people truly would get more out of. Being a member of one of these committees and taking an active part, therefore, is vital to the survival of the Group. If you are a 99/4A owner, especially, please do not shirk away from this and try to do your part.

So where do we go from here? Sometimes I hear people talk about the likelihood of 'clone people' taking over the Group. I also hear talk from some sources that they are not interested in or want to 'learn' about other systems. These last folks must be the same ones that clog up our Freeway Systems with there twelve mile per hour 1902 Buicks and 1918 Fords and on Saturdays spend all day cutting the grass around their house with a sickle. I don't believe that attitudes like that got us our computers to begin with. When I first joined this Group I was very much surprised at the number of older people active in the Group. In some instances, you had guys playing around with computers so large that they would have required a room full of equipment before their retirement two decades earlier. These folks were 'getting on' Bulletin Boards, were experimenting with programming

IKES REVIEW CONTINUES

and playing around with other than just the Basic programming language. They were into such things as Forth and Assembly. We even had some who were working with Ham Radio and Computers combined. The American principle of not being afraid (of innovations) and always being alert in looking for and finding improvements (rather than just sticking with the old), had brought these so called 'old timers' together in a group of people of all ages. It is precisely this phenomena which leaves me to believe that the bulk of our members will be genuinely interested in ALL other computer systems.

As far as one group taking over, that will strictly depend upon the degree of involvement in the group by ANY machine's followers. If the 99/4A owners decide just to sit this one out, the clone owners would probably disappear again and the Group would likely dwindle down to nothing. On the other hand, if all of us worked together, this VAST User Group has the potential of being the best one around. So let's do it! It's up to you now!

Ike Van Kampen

HINTS AND TIPS

TMS 9900 ASSEMBLY LANGUAGE TUTORIAL

Part 10
By Steve Royce
Courtesy WNY 99'ers

READING JOYSTICK INPUT

This article will present a simple example of joystick input in which the position of joystick 1 or its fire button status will be displayed on the screen. As in BASIC, the x and y positions are read as -4 for down or left, 0 for center position and +4 for the right or up position. The fire button is read as ASCII 18. We must also let the computer know which joystick we wish to access (as in CALL KEY(1,a,b). In this example, I have not attempted to check for key status, but that may be accomplished in a number of ways. That will be the subject for another tutorial when I get into keyboard input. The E/A manual does a good job in explaining the addresses which are involved in the KSCAN routine, so I will deal only with a practical example in this article. The source code is well documented, so just follow it through and type it in as you go.

```
          DEF    JOYST
          REF    KSCAN,VSBW,VMBW    Reference utilities
UP        TEXT  'UP'     )
DN        TEXT  'DOWN'   )
RT        TEXT  'RITE'   ) Text to be printed
LT        TEXT  'LEFT'   )
FR        TEXT  'FIRE'   )
FIRE      BYTE  >12          The fire button ASCII value
TEST1     BYTE  >04          Check up and right positions
TEST2     BYTE  >FC          Minus 4, down and left
KU        BYTE  >01
```

CONTINUED PAGE 10 ==>

NOVEMBER, 1987

HINTS AND TIPS CONTINUED....

```
JOYST MOVB %KU,%>8374 Place key unit 1 into >8374
                      to access joystick 1
KS     BLWP %KSCAN     Scan
       LI R0,>2FF      )
       LI R1,>2000     )Clear screen routine
CLS    BLWP %VSBW      )
       JNE CLS         )
       CB %>8375,%FIRE Is fire button pressed?
       JNE T1          NO, goto T1
       LI R0,>187      )
       LI R1,FR        )YES, print to screen
       LI R2,4         )
       BLWP %VMBW      )
T1     CB %>8377,%TEST1  Is X position RIGHT
       JNE T2            NO, goto next test
       LI R1,RT          YES, load RT into R1
       JMP WRITE         Goto the write routine
T2     CB %>8376,%TEST1 Is Y position UP
       JNE T3           NO, goto next test
       LI R1,UP         YES, load UP into R1
       JMP WRITE        Goto write routine
T3     CB %>8377,%TEST2 Test for X to be LEFT
       JNE T4           NO, goto next test
       LI R1,LT         Load LT into R1
       JMP WRITE        Goto write routine
T4     CB %>8376,%TEST2 Test Y to be down
       JNE BACK         No, skip write routine
       LI R1,DN         YES, Load DN into R1
WRITE  LI R0,>207       Screen position
       LI R2,4          4 bytes to write
*                       R1 holds data from tests, above
       BLWP %VMBW       Write it
BACK   JMP KS           Back to check again
       END JOYST        End and auto start
```

NEXT MEETING DEC. 12