

99/48 OWNER/USERS GROUP
MONTHLY NEWSLETTER
Not affiliated with Texas Instruments, Inc.

DECEMBER 1983

VOLUME 1: ISSUE 12

DON'T FORGET

Our next meeting is on December 16; 7:00 to 9:00 PM; Woodstock High School. For those of you who have not been to a meeting at the high school before, we meet in a classroom just off the auto shop. Park in the rear parking lot and enter the building through the doors next to the greenhouse.

1984 MEMBERSHIP DUES

Your dues for the calendar year 1984 are \$15.00. Bring this amount with you to the December meeting. **NOTICE:** If your dues are not paid during the month of December, you will not receive your newsletter in January or the following months until they are paid. If your dues are not paid by February 1, 1984 your name will be dropped from our membership roster.

We intend to get our group up and working in 1984, learn some things and have fun. The January newsletter will outline our 1984 schedule. For those of you real computer hackers; there will be lessons in Assembly Language and how to build your own peripheral devices like an RS232 Interface, Disk Controller, Memory Expansion and more. For the game minded there is the continuing Entertainer Series for you to write software for. We'll also be looking at writing business software. So don't miss out on these activities coming in 1984.

THE ENTERTAINER SERIES

Last month at our meeting, we saw the excellent work done on our project by members Kathie Powers and Randy Erwin. As promised, I said I would show you how to merge graphics, screen displays, sprites and screen text in a single smooth running program with good quality music. You'll see all this and how it was done at the December meeting. In addition, you'll have a chance to see part of our first game "JUDGEMENT 12" in action. When we have at least three games for this package we will send it out to other user groups around the country for comment, and, hopefully get back their best work. The Entertainer Series is in extended basic and requires disk and memory expansion.

THE TEXAS INSTRUMENTS TI-99/8 PERSONAL COMPUTER

Rumor has it that TI will introduce the long awaited 99/8 as the low end entry into their professional computer division. The 99/8 is reported to be an 80K system with disk drive. It is thought to retail around 600-700 dollars at TI Data Product

dealers. Reasons stated for this move was to provide a competitive model to deal with the IBM "Peanut" to be introduced soon. The regular TI Professional Computer sells for \$2500 and up. A TI representative said that the 99/8 is compatible with the 4/A software.

TI had planned to introduce the new 99/8 at the Consumer Electronics Show last June, however, the ADAM computer, announced by Coleco, caused TI to withdraw since the price of the 99/8 was more than the ADAM.

TOUCH TONE PHONE FREQUENCIES

Did you know that just by playing the correct tones into the mouthpiece of a standard touch tone phone you can get it to dial and connect you to any number.

Here are the tones for CALL SOUND

- (1) CALL SOUND(100,1209,0,697,0)
- (2) CALL SOUND(100,1336,0,697,0)
- (3) CALL SOUND(100,1447,0,697,0)
- (4) CALL SOUND(100,1209,0,770,0)
- (5) CALL SOUND(100,1336,0,770,0)
- (6) CALL SOUND(100,1447,0,770,0)
- ~~(7) CALL SOUND(100,1209,0,852,0)~~
- (8) CALL SOUND(100,1336,0,852,0)
- (9) CALL SOUND(100,1447,0,852,0)
- (0) CALL SOUND(100,1336,0,941,0)

GAME TIMER BY RICK MIRUS (SUNCOAST 99er's)

"I was recently writing a game program when my wife decided it would be nice to have a clock display on the screen which could accurately limit the time allowed for a player to move. While a player is deciding on a move, the computer has to jump back and forth through various sections in the program to scan the joystick, scan the keyboard, check for valid moves and reposition a pointer on the screen. So how can you keep an accurate account of elapsed time while doing all of that? Well the problem can be solved easily. As you know, once a sprite is set in motion it continues to move across the screen in a smooth and regular fashion no matter what the rest of the program is doing. This feature, plus the CALL POSITION statement supplies you with everything you need for a timer.

You need only one program line to set the clock and sprite in motion. You then place another line for updating the clock in all the various loops and subroutines during which you want the clock updated.

Here is a short program to demonstrate this...

```
100 Y=2 :: X=13
```

```

110 DISPLAY AT(22,1)BEEP ERASE ALL:
"TYPE A KEY OR MOVE JOYSTICK" :: DI
SPLAY AT(14,13):"TIME"
120 CALL SPRITE(#1,32,1,10,228,0,-1
)
130 CALL SPRITE(#2,42,2,90,124,#3,8
8,16,90,124)
140 CALL JOYST(1,A,B) :: CALL JOYST
(2,E,F)
150 CALL KEY(O,K,H)
160 CALL POSITION(#1,C,D) :: TIME=I
NT(D/3.75) :: DISPLAY AT(15,13):TIM
E :: IF (D>250)+(TIME=0) THEN 220
170 CALL MOTION(#2,-B,A,#3,-F,E)
180 IF H=0 THEN 140
190 X=X+1 :: IF X>20 THEN X=13 :: Y
=Y+1 :: IF Y>9 THEN Y=2
200 CALL SOUND(-1000,K*4,0) :: CALL
HCHAR(Y,X,K)
210 GOTO 140
220 CALL SOUND(500,2200,0) :: END

```

Line 120 starts the sprite in motion. Line 160 displays the time on the screen. In your program just put a line similar to line 170 in as many places as you like, so the time can be updated. If you don't do this often enough, your time may jump more than one at a time, but don't worry, it will still be keeping accurate time.

In line 120 the sprite is located in row 10 and column 228. Color 1 (transparent) is used so that the sprite will not be seen. Column 228 at a speed of -1 provides us with 60 seconds before the sprite reaches the edge of the screen. This could be changed to another number. For example, if you cut it almost exactly in half to 116, you would get a 30 second timer. In line 160 the column position of the sprite is divided by 3.75 to change position into seconds."

HOW TO GET INSTANT COLOR GRAPHICS ON THE SCREEN

Right off the bat, lets have a small lesson in reading the syntax of a statement. Look at the syntax (spelling) of the following statement...

```
CALL COLOR(character-set,foreground-color,background-color[,...])
```

What I want you to take special note of is the [,...] part of the syntax. When you see this in the syntax given for a statement in the manual it means that additional parameters may be added. For example, look at the program below.

```

730 FOR X=100 TO 108
740 CALL COLOR(X,5,1)
750 NEXT X

```

This looks like a rather simple program routine, and it is. None-the-less, you would be surprized how long this takes to execute. Extended basic can speed things up a bit.

```
730 FOR X=100 TO 108 :: CALL COLOR(X,5,1) :: NEXT X
```

Still this takes a lot of time and your graphics change color in a series of steps. We can still do better by understanding the full syntax of the statement.

```
730 CALL COLOR(100,5,1,101,5,1,102,5,1,103,5,1,104,5,1,105,5,1,106,5,1,107,5,1,108,5,1)
```

Now you are using the assembly language subroutine to its full advantage. Your graphic characters will all change color in what seems to be an instant. Try this out in your programs, you'll be pleasantly surprized.

CALL SOUND More than meets the ear!
By Jim Hindley (NORTHWEST OHIO 99er NEWS)

Most of you are probably familiar by now with the CALL SOUND statement (or command). Your manual gives a good explanation of the things you can do with it. But it's the things it does not tell you that you may find especially interesting. For example, did you know that any frequency you assign from 31953 to 43733 ends up exactly as 37287 Hz or that a volume of 8 is the same as a volume of 9?

I came across these strange occurences while developing a program for touch tone frequencies so my 99/4A could dial my phone. I looked in two different reference books to find the correct tone frequencies, and found they disagreed with each other! They were close, but not exactly the same. For example, one book said 697 Hz and the other 699.1 Hz for one of the frequencies. As it turned out, 697 Hz was correct, the other book was slightly off because it listed the tones generated by a specific frequency synthesizer circuit.

As you may have guessed, when you ask the 99/4A to put out 697 Hz, you actually get 699.1 Hz. The reason for the error is the same: The frequency synthesizer in the TI-99/4A cannot get every tone exactly right. It is programmed to get as close as possible, but as the range of highs to lows widens, it has a more difficult job to do.

A crystal oscillator running at 3579545 Hz is first divided down by 32 to 111861 then further divided down to the final frequency by a programmable divider. This final divider has 10 stages which means it can divide by numbers up to 10 binary places in length. In binary, 1111111111 is 2047 (decimal) and that's the largest number it can divide the input frequency (111861 Hz) by. In TI BASIC, however, the last stage is apparently not used so that the maximum is 1023. So let's say you specify a frequency of 698

(A above middle C) in a CALL SOUND statement. Here's what happens...

First, control passes to a machine language subprogram in ROM. The frequency you specified, 698, is divided into 111861 and the answer is 160.2593123. But the programmable divider can't handle fractions. So the computer decides to use 160 as the number to divide by. When the programmable divider is set up to divide the input frequency 111861 by 160, the result is a tone of 699.13125 Hz. Not exactly what you asked for, but probably close enough.

The lowest number the divider can apparently use in TI BASIC is 3. You may have noticed in the manual that the largest frequency allowable in a CALL SOUND statement is 44733. Why is that? Well, if you divide 111861 by 44733 you get 2.5006... which rounds off to 3 (remember the divider must be a whole number). Now if you divide 111861 by 3 you get 37287 and that's the highest frequency that can be output in TI BASIC.

As you can see, in the higher tones there are large jumps which get smaller as the frequency drops. Only when you get below 334 Hz do you finally get down to a 1 Hz jump. Also remember that the TI outputs square waves, not sine waves. Square waves are made up of odd harmonics, some of which are hundreds of times the tone's fundamental frequency!

The volume variable in the CALL SOUND statement has its interesting points too. It steps by TWO's so that you actually have 15 levels, not 30. The loudest level is produced by an attenuation of 0 or 1 and the softest by a 28 or 29. An attenuation of 30 dB is no output at all. In terms of output voltage at the DIN plug, it varies from .6 Vpp to .032 Vpp.

EDITORS COMMENT

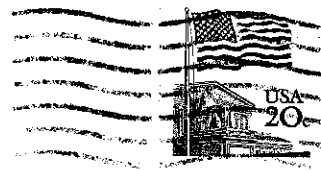
As you may know, in May of 1984 election of group officers will be held. As one of the founders and President this year of our group, it has been my responsibility to provide interesting and informative meetings for you. I have not always been able to do this, mainly because of development factors associated with the inception of any new group. This is our 12th month now and things are finally coming into place. Our 1984 schedule is just brimming full of interesting new things to see and do. Organization is the keynote for 1984. Many new things will be tried and many will be accomplished. We have added to our membership, skills that are not found in most other groups, and, these primarily center around our new ability to produce our own hardware. We feel that these skills are necessary to continue on with our major goal of providing solutions to your computer problems. Education will continue in 1984 with the inclusion of instruction of Assembly Language programming and programming in Extended Basic. We cannot be all things to all people, however, try as we might, some will be lost. To those who are lost I can only say, speak up when you don't understand. And lastly, and forgive me, if your not going to

participate in the meetings and that is to say your going to sit there like a bump on a log, then don't come. To this end, we will do our best to bring you out of your shell or whatever it is. I want to help, but I'm not a mind reader.

-Bob Eckert (815-653-9341) SOURCE TI4929

CX

99/4A OWNER/USERS GROUP
8602 DORR ROAD
WONDER LAKE, ILLINOIS 60097
SOURCE TI4929



MIAMI COUNTY UG
163 WEST THIRD
PERU, IN 46970

DATED MATERIAL
PLEASE DO NOT DELAY