

*Puget Sound  
1 Sept 86*

# TIOS99

NEWSLETTER OF THE PUGET SOUND 99ERS

P.O. BOX 6073  
LYNNWOOD, WA 98036

# CONVENTION ISSUE

**SEPTEMBER 1986**

**Vol. 5 No. 9**

### OFFICERS

PRESIDENT :	CHUCK WYNNE	745-3249	VICE PRESIDENT:	RICK LEWIS	568-0296
SECRETARY :	GEORGE SJAASTAD	522-1101	TREASURER :	DENNIS WOOD	641-0298
NEWSLETTER:	GENE CHRISTIANSEN	641-1032	LIBRARIAN :	JOHN UELAND	672-7593

### NEXT MEETING

DATE : THURSDAY, September 25, 1986      TIME: 7:00 PM  
PLACE: Kirkland Public Library, 406 Kirkland Ave, Kirkland

- |         |                               |   |
|---------|-------------------------------|---|
| AGENDA: | * Officer reports             | * Convention updates                    |
|         | * TI Artist demonstration     | * Extended BASIC Graphics by Rick Lewis |
|         | * Question and Answer Session | * General discussion session            |

IN THIS NEWSLETTER

The President Says. . . . . 3

RAM . . . . . 4

TI99/4A News

    Seattle TI99/4A Convention. . . . . 5

    Freeware updates. . . . . 6

    Myarc Extended BASIC IV v2.1 released . . . . . 6

    Updated Myarc disk manager III. . . . . 7

C-Review: Using C99 . . . . . 7

Assembly Language Tutorial: Part I. . . . . 8

Marvelous Time Savers . . . . . 10

Using the Interrupt Button: Pause program . . . . 10

Extended BASIC Programs!

    Fancy Text demo . . . . . 12

Bugs, Bugs, Bugs!

    Fat Bits drawing program. . . . . 15

    Spiral. . . . . 16

SOME LOCAL BULLETIN BOARD NUMBERS

<u>TITLE</u>	<u>PHONE NUMBER</u>
PUGET SOUND 99ERS BBS-SEATTLE	784-4142
QUEEN ANNE COMPUTER SHOPPE	361-0895
TI-COMM PLUS-BREMERTON	377-1845
NEW WORLD BBS	365-6938
RAINBRIDGE ISLAND	1-598-3228
BERDEEN	533-0951
ELETRAVEL	742-9034
SHUTTLE	885-INFO
ROTHGUARD-BURIEN	824-6757
RADIO 1	878-0158
Genie	1-800-638-8369

**THE PRESIDENT SAYS**

By Chuck Wynne

One of my TI systems has the Myarc 512K card. This card is wonderful on BBS's due to the fact that if you ask it to log to RAM disk, the Mass Transfer program will save all data so fast you wont even notice your saving every word. I recommend it. Mass Transfer V3.8 is out now. It has some nice improvements. It is in our library. I would suggest you send the author some money. By the way, the local newspaper said that in the Freeware department, 1 out of 2500 people send in their money. Thats a survey for all computers. That looks gloomy.

Recommended also is Barry Traver's Diskazine. His price was \$30 per year. Sounds like a lot? Not for what you get!!!! You are dealing with the real experts on the TI. His magazine on disk is great. He has an archiver program that will condense, say Mass Transfer, the documentation, and all files into one for uploading to a BBS. When someone downloads the big file, they run the archiver and transfer the file back to the original ones. The documentation being one, the program being another etc. The big file also takes up less space than the original and keeps the whole thing together if you want this to be used for backup. This program is copyrighted and uses the XXB routines also on the disk. These are routines that enhance Extended Basic with new calls you can use in your own programs. I haven't even touched the surface of what he has sent on the disks. Write to Genial Traveler, 835 Green Valley Drive, Philadelphia, PA 19128. You know, with computer, the back issues are just as important as the future ones.

The first program I ever typed in from HC Magazine was written by Barry. He is a minister also for your information. I met him in LA at the Fest. He was on our BBS just recently but left no message. I hope he will come to our convention.

**CALDWELL**



So that's why they call it a "cursor"!

The Myarc 512K card can be backed up by transformer. Someone asked if they could use an Atari transformer that puts out 500 milliamps. Well, if the current is steady enough, the card requires 230MA by my meter (9V + on the tip of the jack). Sounds like it would work fine. By the way, has anyone added 9V nicads to it as well for battery back up and a natural pad to smooth the supply? Let me know as this sounds like a good idea.

I was at the Comsoft show in Bellevue September 13th. No TI's were there. However, Chris Baker, a TIER in another club took one big chance. He bought a 10 Meg hard drive for about \$89. Well he called the company and found out it wasn't a 10, but a 20 MB. He took it over to Michael Auer's house where he runs a hard drive on his TI. It worked perfectly. What a bargain. This by the way can be hooked up to the MYARC controller card that has the quad density chip. It is the replacement for the personality card. The upgrade is about \$49. This will allow you the use 80 track drives. This gives you 2880 sectors per floppy which is 720K of storage. Wow! The 20 Meg gives you 80,000 sectors.

What still impresses me most about the TI is the lack of individuals selling their machines. When I decided on a computer, the best way to tell if it was good is to watch if people resold them. To this day there are very few in any paper for sale. Take a look. If you find one there for a bargain, and you call, it will be gone.

The first home computer to have "C" language written for it is the TI. Thanks to Clint Pulley of Canada. He did it on a DEC PDP 11/70 mainframe computer and transported it to the TI. Truly marvelous. Even the ATARI 520ST doesn't have it yet that I know of. I think we are better off now than if TI was still selling the 4A. Look at all the lovely support. Please support all those who support this machine. You will be supporting yourself!

## R-A-M!

By GENE CHRISTIANSEN

Hello, again!!! That month went pretty fast! Could it be that the anticipation is getting to me? Do I feel like a kid at Christmas? THE CONVENTION IS COMING, THE CONVENTION IS COMING!!! I am usually very "upbeat" at this time of the year anyway (my birthday is Sept 26th) but this year is just something else!!! This event could rightfully be THE BEST TI HAPPENING OF THE YEAR!!!

We have been witness to many astounding announcements and pronouncements this past year concerning "the computer that would not die".

It began in November, really. The news that an upgraded TI 99/4A (although with a new name) was hearing the BETA testing stage excited many. A demonstration of this machine was to happen at Chicago. Its announcement electrified the attendees and possibly provided greater turnout. It did not materialize. The bad press on Lou Phillips heated up again. People like Randy Ainsworth (of "rumor tag" fame) began to line up to cast the first stone. Lou promptly shot himself in the foot in Los Angeles by not having a working model of the Myarc machine there. This was followed by another self-crucifixion in New Jersey. It seemed that no matter where the Myarc band-wagon was to appear, it was doomed.

Sure, there were the smaller successes, like the introduction of Extended BASIC Level IV, the alk of the Module Emulator, and new disk controller. The 512K cards are now proving their worth. These all paled into insignificance when placed alongside the "big news"...the machine itself.

Then came Chicago again with a working model!!! This was followed by the news of the contributing personnel that is helping the development. Names like Charlton, Robbitt, and others.

Meanwhile, Barbara Wiederhold had started a ball rolling in Seattle. It was first thought to be the workings of an unsound mind. She could not even get help from local user groups! But Barb could organize and she did ever! Now we TI owners in the Great Northwest have our own event!!!

It also seems that by default (or the Will of God) we are to be the first to really showcase the Geneva, as it is now called. What timing!!!

If ever a case could be made for attending some sort of function, this is it!!!

And if having the Geneva at this convention were not inducement enough, take a gander at the list of speakers who are intending to attend: Craig and Sue Miller, Lou Phillips, Chris Bobbitt, Jim Horn (TI Forum on CompuServe), and Cheryl Whitelaw (C. Regena - COMPUTE).

Do you need any more reason to be there? How about it if you bring that friend of yours that has had a TI 99/4A in his closet for two years? How about the kid that grew up with TI but "upgraded" to the "Comme-dore" or "Trash-80"? Maybe you could bring along that neighbor who wants you to recommend a computer to use in his home or business.

Believe me, there will be something for everyone!!! So plan to come to the Sea-Tac Holiday Inn on the 26th of Sept. and find out what is really going on!!!

You can call Barb at 283-0753 (from 11 am to 5 pm, Mon-Fri) for further info on tickets, etc.

A sad note: after a year and a half, the Aberdeen BBS is going to be shut down. The sysop there (Randy Ainsworth) is citing not enough time to run it properly as the cause of the demise. This follows the tragic loss of the TIBBS run by Ed Durfee. I hope this isn't the beginning of a trend! I can still remember when it was said the TI 99/4A could not handle a BBS! Now there are more TI boards that are other single type!!!

I am still waiting to hear from those of you who may be unsatisfied with the way that this newsletter is published. Is it inconvenient to receive it at the meeting? I want to hear from you!!! Silence will guarantee tacit approval!!! I would hope that something like this would have generated SOME response!!!

The new promotion will continue for some time, yet. As you may recall, a member is responsible for bringing in two new members will receive a box of our high-quality disks!!! Can't beat a deal like that!!! Once someone joins then THAT person can "sponsor" new members and receive a box of his (or her) own!!! Remember, this is for a limited time only, so hurry!!!

This next item comes from a recent Reader's Digest. It had been submitted to the "All in a Day's Work" section by Priscilla J. Smith:

"A colleague was having a hard time operating her desktop computer. It was apparent that she had reached the end of her patience when she muttered at the screen, 'I know one thing. If I had a glass face like you do, I'd sure behave better.'"

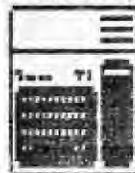
Our club now has a 1200 baud modem!!! We are now waiting for Chris George to arrange the time to get the software up and running. What with returning to school, I am sure he has been one busy young man!!! He has promised to get the system going very soon. It has become one of the most used boards in the area with calls being received from such far away places as Maryland and Pennsylvania!!!

We are still looking for someone familiar with or wishing to learn C-99. We have received many letters (and returned survey sheets) indicating that there is a growing need for help in language. Please call Chuck, Rick Lewis, or me if you can help out.

Well, that is about all for now. Don't hesitate to write! And see you at the meeting...AND THE CONVENTION!!!

## TI99/4A News

## Seattle TI99/4A Convention



The Seattle TI99/4A Convention is drawing nearer and nearer. It will be held at the Sea-Tac Holiday Inn. The night after our meeting, Friday, September 26, will be the President's Dinner. Reserve your seat now!! Barbara must send the number of seat reservations in on Wednesday morning. Speakers at the dinner include: Lou Phillips, Craig Miller, C. Regena, Chris Bobbit, Franz Wagenback, Al Lawrence, and others as time allows. There are currently about 75 people registered for the dinner and growing daily.

Saturday, September 27 is the day of the convention. It will be held in a 3300 square foot room. Featured booths will include:

Western Washington Computer Club, Spanaway, WA.  
 Cowlitz County Users Group, Longview, WA.  
 CompuServe Telecommunications, TI-FORUM, Jim Horn.  
 T.A.P.E. LTD. Franz Wagenback, German products.  
 Bits and Chips, computer shop, Al Lawrence.  
 Miller's Graphics, Craig and Sue Miller, San Dimas, CA.  
 C. Regena, Compute! magazine.  
 Myarc, Inc. Lou Phillips, New Jersey.  
 Queen Anne Computer Shoppe, Barb Wiederhold  
 Tigercub Software.  
 Asgard Software. Chris Bobbitt, Rockville, MD.  
 Vancouver BC Canada Users Group (TICC), Paul Demara, President.  
 Puget Sound 99ers, Chuck Wynne president.  
 Portland Users Group. Cricket Reyburn, president.  
 GENie Telecommunications. Scott Darling, Spokane, WA.  
 Eugene Users Group, Eugene, Oregon.  
 Washington D.C. Users Group.  
 Belgium Users group. Europe.  
 Tic Tac Users Group, Lake City, WA, Al Lawrence, President.  
 NOVA 99ers, Vancouver, WA, Daniel Lisson, president.  
 Los Angeles 99ers, Los Angeles, CA, Terri Masters, president.  
 Kitsap County Users Group, Bremerton, WA, Dennis Splett, Pres.  
 Boise 99ers, Boise, Idaho, Jerry Hough, president.  
 Tacoma 99ers, Wally Dubberly, President  
 Micropendium magazine  
 Tenex Catalog  
 Computer User



During the Convention, there will be a room across the hall where people will speak on new products for anyone to listen. There will also be tables with refreshments inside the convention room for your convenience. Remember to bring your coupon and your membership card to get \$1.00 entrance fee. You will pay \$3.00 without.

### Freeware Updates



There has been an influx of updates to freeware programs. A new PR-BASE 2.0, DM1000 3.3, Mass Transfer 3.8, FunLWriter, RLE Max, BA-Writer, and more. We only have the new PR-BASE, DM1000 3.3, and Mass Transfer 3.8 in our library at present. The Mass Transfer 3.8 has faster data retrieval, easier reconfiguration, and other modifications. The DM1000 3.3 now allows for display of display fixed 80 files along with display variable 60 files. The new PR-BASE 2.0 now loads from Extended BASIC, has easier to understand menus for data base and mailing label creation, and has double sided disk capability.

### Myarc Extended BASIC IV v2.1

The new Myarc Extended BASIC IV has just been released. We have not yet recieved it yet but we

are expected to see it at the convention. The new version will do integer arithmetic, DEF and SUB statements supported, and some of the bugs taken out.

### Updated Myarc Disk Manager III

The new Myarc Disk Manager has been released and we do have a copy. The new version allows for the new EPROM chips on the Myarc 128/512 K cards, so you can partition them for 128k OS (for Extended BASIC IV) and 32K OS. They have taken the bugs out of the Clone option where people have had troubles.

**SUBJECT: C99**  
**FROM: Ed Elliott**

MINIMUM EQUIPMENT: CONSOLE, MEM/EXP, 1 DISK DRIVE, ED/ASSEM

To start I'd like to present a quote from 'COMPUTER SHOPPER', July issue: "One of the cruelest realities of using an "ORPHANED" machine is the steady decline of commercial software support...as the ability of consumers to rely on third party software dwindles, the residents of the orphanage, then have a couple of choices: they can sell out and buy a fully supported machine, or they can take up the challenge and start writing their own software." I couldn't agree more!

A few months ago I was at a meeting of the TIC-TAC User's Group. At this meeting, the membership decided to try to produce a marketable piece of software. At that time I had reservations about this idea. In the past, the T.I. owner has had a choice of the following languages to undertake a project of this type.

BASIC: This is probably the language that everyone started out with.

After all, it's the language that came in the console. In T.I.'S case it's probably the best version to come in any home computer (STILL!). T.I. BASIC has the easiest graphics to use of any computer that I have tried and sprites are a snap. But it has its draw-backs. It is horribly slow and this reduces the utility of any program written in BASIC. From a businessman's standpoint, this alone eliminates BASIC.

Hopefully Myark's BASIC will cure that, but for about \$250.00 more than the price of C99.

Assembler: Well here is the opposite end of the spectrum. You have complete control of the machine but that a slow learning curve. To learn Assembly Language, (for me anyway) is more than I care to attempt.

One thing you can say speed is no problem, but this is definitely for masochists!

Forth: Now were somewhere in-between. But what a strange language. I've bought the book 'Starting Forth' but after 2-3 months, I'm still not sure what's what. Reverse polish notation isn't that big of a deal, but it's trying to figure out what is going on - to quote Doug Rose, "Compared to Forth Assembler is easy...", (to understand that is).

Pascal: This looked like the answer. When I first got my P-Code card, I could hardly wait! But after a month learning the P-System and then Pascal I must say this was a very frustrating implementation of Pascal. It has the slowest compiler on earth and after your done? You have P-Code which is interpreted by the GPL Interpreter and is only mildly faster than BASIC. Back to square one.

C99: At last there is something that could be 'THE' language for the T.I. a quick comparison to the other languages puts it closest to Pascal- with bad punctuation, syntax wise that is. If you know Pascal, you can quickly grasp the fundamentals.

I've used C99 for about 3 weeks - not enough time to be a pro, but enough to find some good and some not-so-good features.

This is not a complete implementation of C. It has most of the key words and most of the

operators. What is missing here can be overcome by careful programming. What cannot be overcome is the fact that it can only handle D/V 80 files. This is its severest fault, but it does have the best graphics library of nearly all 'C's that I've seen. There is no 'MAKE' utility. If there is a small change to the source then the whole program must be re-compiled and Assembled with the Editor Assembler (not included of course). Also the closest that you'll come to a librarian is your copy of D/M 1000. Those files that are already assembled are included with the ASM- -ENDASM commands, this is a rare and valuable feature. Those files that must be compiled with the program use the standard 'INCLUDE'.

Its compiler, though fast on small projects slows geometrically with the addition of more include files and longer programs. Still for the money it is a kick to use. If it is updated to include better file handling, it could stand as it is for some time.

To use the program, first copy the following files to a blank, initialized disk: C99C,C99D,C99E. Rename them: UTIL1,UTIL2,UTIL3. Then copy: STDIO,PRINTF and CSUP. Next copy: ASSM1,ASSM2 (and the EDIT1 file if you are using the ED/ASSEM editor) from the Editor/Assembler disk.

Unfortunately you cannot use the FunLWriter Editor or ASSEMBLER, because they both lock up!

Write your program with the editor and save it to this disk, then go to option 5 of the Editor/Assembler and when prompted for a filename, press ENTER. This auto loads the UTIL1,2,3 files. The compiler asks for a filename and you give it the name of the file you saved with the editor. Then you are asked for an output filename. This will be an Assembly Language file so tack on a '/A' to remind yourself.

You are also asked if you want to include the C source, for small programs, to it. Your errors are flagged in this Assembler file at the C program lines (which are now comments in the file) directly below with the ^ sign and the word "ERROR", a thoughtful feature.

Upon successfully compiling this program (the one with the '/A' attached), you go to the Editor/Assembler and load the Assembler. Press ENTER at the prompts until prompted for a filename - give it the one with the '/A' attached. When it is done you load your program and CSUP then press ENTER at the third filename prompt. You are asked for a program name, type "START", then press ENTER and that's all there is to it! The wait is worth it, incredibly faster than BASIC and soooo much easier than Forth. A real thrill to see your program go that fast. So-Long!

## Assembly Language Tutorial: Part I

By Doug Rose

Before we start programming we need to know about the registers. We have 19 registers. Three which are Hardwired directly to the 9900 chip. These 3 are very special registers. They are the program counter (PC), workspace pointer (WS), and the status (ST). The concept of these registers is very important in order to learn assembly.

First, the program counter tells us the address of the instruction being processed. Our computer uses two memory locations at one time for each instruction. For example if our first instruction was at memory location 10 the next instructions would start at memory location 12 then 14 then 16 and so on until our program ended. Remember this is done all automatically. All we need to do is set the PC at the desired location. There is a few instructions that change the PC such as jump and branch. These are just like the GOTOs in BASIC. The jump instruction can only jump + or - 128 places. There are some other branch instructions that allow you to go anywhere in memory, but we'll wait to talk about these until another time. Most of the time the PC is self incrementing, meaning that as soon as one instruction is processed the next instruction in numerical order is processed with out any help from us.

The workspace register (WS) is a register that tells our computer where our 16 registers are.

Being that all information in the computer is processed in the registers, the WS is important for us to know about. There is an address on the CPU where there is 256 bits of memory. People like to use this space for their workspace when they need a program to run very fast. The reason this is very fast is the electrical impulses have a very small distance to travel to bring the information to the CPU (they are on the same chip). If you use the memory in the PEB then it has about ten feet to and back from memory. There will be times when you will need to branch to another routine in memory. When this happens you may leave these 16 registers and set up another area in memory for your registers. Then when you return back to your main routine you can restore your old register area with all information intact just the way they were before you left it. This is all done by changing the address in the WS to what ever address you decide to move your 16 registers to. You will see later when you write your own programs how all of this can save time, memory, and energy.

The status register tells us important information about the instruction that was just completed. For example: A R0,R1 adds what ever number we put in register 0 and 1 and stores the answer in register 1 (R=register, A=add). The status register is affected by this operation as it is by most operations. Bits 0 thru 5 of the status register contain information pertaining to the most recent arithmetic or logical operation such as the add instruction above and a few other type instructions. Each time any instructions are performed the status register is updated. For this reason the status register is very important for us to understand.

To explain the status register, it would be easiest explained by discussing the IBM punch cards that we may have seen. Our electric, gas, phone, and many other bills we see each month often use these cards. There is 78 places or fields across the card and 28 down the card. For an example we'll use the first 20 fields for first name and last name. The next one field will used for sex. Another two fields will be used for age and the last 5 fields for the zip code. Now with the card set up like this we could very easily do a search for someone in a specific area, specific age, and a particular sex. If we wanted everyone at zip code 98205, who was male, and between the age of 30 and 35, we would set up our card reader to all of this automatically. The card readers have little metal fingers that come down and touch the card in all of the available fields. If there is punched holes in the card these metal fingers will make contact and signal the card reader. If the sex field had a punch we would know it was male (no punch it would be female). If one age field had a punch in the 3 column and the other field had a punch in the 4 column we would know he was 34 years old. If we had the proper punches for the zip code, the card reader could kick the card in to the good pile and off we would go to process the next card.

The status register is much the same as the metal fingers. The information the fingers supply or the status registers supply may or may not be used. All processing in our computer goes through these electronic fingers. We only have to ask the status register what happened on the last instruction.

### Status Register

```

Bit  0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15
+-----+
:L>:A>:EQ:C :OV:OP:X :  RESERVED  :INT. MASK :
+-----+

```

Bit Number	Name	Meaning
0	L>	Logical greater than
1	A>	Arithmetic greater than

2	EQ	Equal
3	C	Carry
4	OV	Overflow
5	OP	Odd parity
6	X	Extended Operation
7-11	-	Reserved
12-15	INT. MASK	Interrupt mask



The status register is 16 bits long. In order to see what bit was turned on we will all need to know how to convert from hex to binary. (Converting from hex to binary is easily done and if you request it, I will write about it in the next article). The first 7 bits are used for the following: LOGICAL GREATER THAN, ARITHMETIC GREATER THAN, EQUAL, CARRY, OVERFLOW, ODD PARITY, EXTENDED OPERATION. The next 5 bits aren't used. The last 4 bits are INTERRUPT MASK VALUE. Most instructions modify the status registers but only in its pre-determined field. For example if we add 2 numbers together the logical >, arithmetic >, Equal, Carry, and the overflow bits will be affected. Either the bits will be turned on or off by this instruction, but no other bits will be affected. Odd parity, extended operation, and the interrupt will not be affected whether they are already on or not. We can find out which bits are affected by an instruction by looking at pages 80 through 202 in the Editor Assembler book. Next month ---- Explanation of the status register bits and the other 16 general register plus more.

## MARVELOUS TIME SAVERS

By Dave Miller



When I first purchased my 99 4/A in 1979..my goal was business and household efficiency. This digital wonder would give me total organization and additional leisure time. 7 years have gone by..let's look at what the little appliance has done for our family.

- \* Income Tax..better organization and improved refunds. More accuracy but not much savings in time.
- \* Word Processing..our family writes letters more often and I've seen an improvement in creative word usage. A word processor is much more tuned to human thought than typewriters.
- \* Financial Records..more orderly and accurate. We have a real budget and can monitor our net worth easily.
- \* Education..my children are computer literate and understand the value of rapid information access.
- \* Telecommunications..years of BBS usage has prepared my family for the future. Many experts predict cottage industries(work at home) will replace the present office environment.
- \* Fun..stimulating interactive games keep us alert and occasional arcade involvement helps relieve stress.
- \* Business..running a part-time business is made easier by computerized record keeping. Necessary paperwork is automated.

This isn't the entire picture but a realistic outline. On the negative side, many hours spent at the keyboard might have been spent hiking the beautiful Pacific Northwest outdoors. a great treasure our family enjoys. Oh well, life is a series of tradeoffs.

## Pause Button: Using the interrupt button

By Tom Wynne

I wrote the following Assembly Language program using routines from Danny Michael's screen dump

utility. It loads into memory so when you hit the interrupt button, the program will halt any module and wait for a key to be pressed and then continue. I made it mainly because if I am playing a game and something comes up that I have to leave it, I could press the interrupt button and halt it. I could come back later and hit a key and continue playing.

Unfortunately, the program will work only under certain situations and certain modules. The problem arises when the program passes control back to the module some things get reset, such as the program counter, workspace registers, GPL addresses, etc. The program did work on some modules I had such as Parsec, Donkey Kong, Chess, and it works with BASIC and Extended BASIC sometimes. I warn you, it is not guaranteed to work so save any data before trying it! The program also uses a CALL LINK("PAUSE") from Extended BASIC so you can issue this command anywhere in your program to wait for a key pressed.

Using the interrupt button

Notice at the end of the program it has two statements that read:

```
ADRG >FFFC
DATA MYWS,CART
```

The ADRG >FFFC tells the Assembler to write the next code at memory location >FFFC. The code that is written via DATA MYWS,CART is the address of the workspace in my program MYWS and the address of the start of the program CART. When you hit the interrupt button, it performs a BLWP @FFFC and branches to CART. CART waits to make sure you have released the button and then resets it for the next time you press it. Type in the program below (remarks optional) and save it as PAUSE. Then assemble to the file PAUSE/OBJ using the R option for registers.

```
*****
* PAUSE BUTTON-USED WITH INTERRUPT BUTTON *
* PAUSES ANY PROGRAM AT ANY TIME          *
* BY TOM WYNNE                             *
*****
```

```
MYWS      DEF  PAUSE      For CALL LINK("PAUSE") from XB
          BSS   >20       Set aside memory for registers
KSCAN    EQU   >201C     Address of routine KEYSAN
KEYVAL   EQU   >8375     Address of key from KEYSKAN
STATUS   EQU   >837C     Address of GPL status
GPLWS    EQU   >83E0     Address of GPL registers
FLAG     BYTE  >00      place for BASIC or button
                       0=interrupt button
                       1=called from BASIC
PAUSE    LIMB  0        turn off interrupts
          LWPI  MYWS     Load my workspace
          SETO  @FLAG    put a 1 in FLAG (from BASIC)
          JMP   CHKKEY

CART     CLR   @>FFFC    entry point from button
          LIMB  0        (Danny Michael Screen dump
          LWPI  MYWS     by C.J. Daily)
          CLR  @FLAG     put a 0 in FLAG (from button)
          CLR  R0
```

```

LOOP2  DEC   R0           loop >FFFF to >0000 step -1
        JNE   LOOP2      not 0? GOTO LOOP2
        STWP  R0         store workspace pointer in R0
CHKKEY  MOV   R0,@>FFFC  move it to the address >FFFC
        CLR  @STATUS    clear status byte
        BLWP @KSCAN     execute routine KSCAN
        MOVB @STATUS,@STATUS key pressed?
        JEQ  CHKKEY     no, goto CHKKEY
        CLR  R0
        MOVB R0,@STATUS  clear status byte
        MOVB R0,@KEYVAL  clear key value byte
        LIM  2          turn on interrupts again
        MOV  @FLAG,@FLAG check flag
        JEQ  EXIT2     is it 0? yes, goto EXIT2
        LWPI GPLWS     load GPL workspace
        B    @>0070    go back to BASIC
EXIT2   RTWP          reset and return

        AORG >FFFC     tell assembler to put
        DATA MYWS,CART statements at >FFFC
        END           end of program

```

Type in the following program in Extended BASIC to load the Assembler program.

```

100 PRINT "LOADING PAUSE ROUTINE..."
110 CALL INIT
120 CALL LOAD("DSK1.PAUSE/OBJ")
130 PRINT::PRINT "PAUSE ROUTINE
LOADED..."
140 PRINT "PUSH INTERRUPT BUTTON TO"
150 PRINT "PAUSE PROGRAM...OR USE"
160 PRINT "CALL LINK("PAUSE")"
170 PRINT "TO PAUSE PROGRAM"
180 PRINT::PRINT "PRESS ANY KEY"
190 CALL LINK("PAUSE")
200 END

```



I would appreciate any hints on how to make this program work on other modules. Call 745-3249. Thank you.

## PROGRAMS, PROGRAMS, PROGRAMS

The following article is reprinted from the COMPUTER BRIDGE, the newsletter of the St. Louis 99ers Users Group.

### FANCY TEXT ON THE SCREEN

by Roy T. Tamashiro

Interesting screen presentations can catch the attention of computer users and bystanders alike.

The two routines below allow you to display text on the screen in a "fancy", attention-getting way. These routines are useful in creating title screens for your programs.

The first routine, CENTER BURST, makes each line of text emerge from the center of the screen(line), as though the letters were emerging from a geyser or a volcano. When the screen is complete, the letters "fall back" into the center of the row. The other routine, TELETYPE, displays text character by character, like a teletype machine. The text is centered on each line and lines are printed both left to right and right to left. The text is wiped from the screen using the same teletype action. Both routines include sound effects for dramatic effect. To use the routines, type in the CENTER BURST and TELETYPE routines and save them on cassette or disk. They may be typed and saved separately, or they may be combined into one continuous listing. When you want to include either (or both) routine(s) in your program, load the routine(s) into the computer's memory, and then add DATA statements to specify what you want displayed on the screen. The format of the DATA statements is as follows:

(Line number) DATA(Screen Row Number(1-24)), (Text)

For example:

250 DATA 1,"My Title Screen"

In this simple example, the message "My Title Screen" will appear on row 1. Note the following precautions:

- 1) The Row Number must be between 1 and 24.
- 2) The text may not be longer than 28 characters per row.
- 3) Be sure no other program lines follow the CENTER BURST and TELETYPE routines.

After the last line to be displayed on the screen add a final DATA statement with a number larger than 24 to indicate no further lines are to be displayed on the screen. For example:

750 DATA 99

Use CALL BURST or CALL TELETYPE following your last DATA statement to invoke the proper routine. The FANCY TEXT DEMO listed below illustrates how this routine is carried out.

```
*****
100 !*****
110 !* FANCY TEXT DEMO *
120 !*****
130 !Author: Roy Tamashiro
140 !Language: X-BASIC
150 !September 1985
160 DATA 1,"CENTER-BURST
TEXT"
170 DATA 2,"-----
----"
180 DATA 3,"BY ROY TAMASHIRO"
190 DATA 7."In this routine,
lines are"
200 DATA 8."written from
the center of"
210 DATA 9."the screen
*
* 260 DATA 1,"THE TELETYPE
MACHINE"
* 270 DATA 2,"-----
-----"
* 280 DATA 3,"By ROY TAMA
SHIRO"
* 290 DATA 5,"This routine
displays text"
* 300 DATA 8,"like a teletype
machine."
* 310 DATA 9,"Any line 28-
characters or"
* 320 DATA 10,"less is center
ed on the"
* 330 DATA 11,"screen on the
row you"
```

```

outward."
220 DATA 22,"Then the lines
    are erased"
230 DATA 23,"back into the
    center."
240 DATA 99
250 CALL BURST
* 340 DATA 12,"designate."
* 350 DATA 21,"Then the lines
    are erased."
* 360 DATA 99
* 370 CALL TELETYPE
* 380 END
*
*****
31000 !*****
31010 !* CENTER BURST *
31020 !*****
31030 !AUTHOR: ROY TAMASHIRO
31040 SUB BURST
31050 DIM R$(24)
31060 FOR I=1 TO 24 :: CALL
    CLEAR
31070 READ ROW :: IF ROW<25
    THEN READ R$(ROW) ::
    GOTO 31070
31080 FOR R=1 TO 24 :: IF
    R$(R)="" THEN 31160
    ELSE W#=R$(R)
31090 LLF=LEN(W$)/2 :: LRT=
    LEN(W$)-LLF
31100 LEFT$=SEG$(W$,1,LLF)
    :: RIGHT$=SEG$(W$,LLF+
    1,LEN(W$))
31110 CALL HCHAR(R,1,32,32)
31120 FOR I=0 TO LRT
31130 IF LRT-I/1 THEN DISPLAY
    AT(R,15):SEG$(RIGHT$,
    LRT-I,I+1)
31140 IF LLF-I THEN DISPLAY
    AT(R,14-I)SIZE(I+1):
    SEG$(LEFT$,1,I+1)
31150 CALL SOUND(-50,990,1)
    :: NEXT I
*
*****
32000 !*****
32010 !* TELETYPE *
32020 !*****
32030 !AUTHOR: ROY TAMASHIRO
32040 SUB TELETYPE
32050 DIM W$(24) :: CALL
    CLEAR
* 31160 NEXT R
* 31170 FOR R=1 TO 24::
*     IF R$(R)="" THEN
*         31300
* 31180 RIGHT$="" :: LEFT$=""
* 31190 FOR I=3 TO 16
* 31200 CALL GCHAR(R,I+14,B)
*     :: CALL GCHAR(R,I,A)
* 31210 LEFT$=LEFT$&CHR$(A)
*     :: RIGHT$=RIGHT$&CHR$(
*         B)
* 31220 NEXT I
* 31230 FOR I=1 TO LEN(LEFT$)
* 31240 CALL SOUND(-50,-2,1)
* 31250 DISPLAY AT(R,15)SIZE
*     (LEN(RIGHT$)+1):RIGHT$
*     :""
* 31260 DISPLAY AT(R,I)SIZE
*     (LEN(LEFT$)+1):LEFT$
*     :""
* 31270 LEFT$=SEG$(LEFT$,1,
*     LEN(LEFT$)-1)
* 31280 RIGHT$=SEG$(RIGHT$,2,
*     LEN(RIGHT$)-1)
* 31290 NEXT I :: CALL HCHAR
*     (R,17,32)
* 31300 NEXT R
* 31310 SUBEND
*
*****
32000 !*****
32010 !* TELETYPE *
32020 !*****
32030 !AUTHOR: ROY TAMASHIRO
32040 SUB TELETYPE
32050 DIM W$(24) :: CALL
    CLEAR
* 32180 FOR C=30 TO 3 STEP -1
* 32190 CALL HCHAR(R+1,C,30)
*     :: CALL SOUND(-10,
*         990,1)
* 32200 IF C<=START AND L1>=
*     (START-C+1)THEN DIS
*     PLAY AT(R+1,C-2)SIZE

```

```

32060 FOR I=1 TO 24 :: W$(I) * (1):SEG$(W$(R+1),L1-
    ="" :: NEXT I * (START-C),1) :: GOTO
32070 READ ROW :: IF ROW<25 * 32220
    THEN READ W$(ROW) :: * 32210 CALL HCHAR(R+1,C,32)
    GOTO 32070 * 32220 NEXT C
32080 FOR R=0 TO 23 STEP 2 * 32230 NEXT R
32090 IF W$(R)="" THEN 32160 * 32240 FOR R=1 TO 24 STEP 2
32100 START=INT(17-LEN(W$ * :: IF W$(R)="" THEN
    (R))/2) * 32260
32110 FOR C=3 TO 30 * 32250 FOR C=3 TO 30 :: CALL
32120 CALL HCHAR(R,C,30) :: * HCHAR(R,C,30) :: CALL
    CALL SOUND(-10,990,1) * SOUND(-50,-2,1) ::
32130 IF C>=START AND LEN(W$ * CALL HCHAR(R,C,32) ::
    (R))>C-START THEN DIS * NEXT C :: W$(R)=""
    PLAY AT(R,C-2)SIZE(1): * 32260 IF W$(R+1)="" THEN
    SEG$(W$(R),1+C-START,1) * 32290
    :: GOTO 32150 * 32270 FOR C=30 TO 3 STEP -1
32140 CALL HCHAR(R,C,32) * :: CALL HCHAR(R+1,C,
32150 NEXT C * 30) :: CALL SOUND
32160 IF W$(R+1)="" THEN 322 * (-50,-2,1) :: CALL
    30 * HCHAR(R+ 1,C,32) ::
32170 START=INT(16+LEN(W$(R+ * NEXT C
    1))/2):: L1=LEN(W$(R+ * 32280 W$(R+1)=""
    1)) * 32290 NEXT R
    * 32300 SUBEND

```

\*\*\*\*\*

## BUGS, BUGS, BUGS!!

### Fat Bits

By Tom Wynne



Someone brought to my attention that there was an error in this program. Sure enough, line 510 needed the carat (^) symbol. I also modified the program so you can erase dots by hitting enter on the dot that is on and it will turn it off. The asterisks '\*' are in front of lines that have been changed or added to the program.

TI Writer does funny things to BASIC program listings through the formatter. This is the main reason that these bugs arise in programs that we publish in the newsletter.

```

100 REM FATBITS--BY TOM WYNNE
110 REM PUGET SOUND 99ERS
120 REM
130 CALL LOAD("DSK1.DUMPIX")
140 CALL CLEAR
150 DISPLAY AT(4,7):"F A T B I T S"
160 DISPLAY AT(6,3):"BY TOM WYNNE - TIPS 99"
170 DISPLAY AT(12,1):"INITIALIZING...PLEASE WAIT"
180 DATA 00000000,0F0F0F0F,F0F0F0F0,FFFFFFF
190 FOR I=0 TO 3 :: READ A$(I):: NEXT I

```

```

200 DATA 00,01,02,03,10,11,12,13,20,21,22,23,30,31,32,33
210 FOR I=128 TO 143 :: READ B#
220 FOR J=1 TO 2 :: F#=F#%A#(VAL(SEG$(B#,J,1))):NEXT J
230 CALL CHAR(I,F#):: F#="" :: NEXT I
* 235 CALL CHAR(127,"F09090F0")
.240 CALL CLEAR
* 250 CALL SPRITE(#1,127,16,1,1,0,0)
260 R=1 :: C=1
270 CALL LOCATE(#1,R*4-3,C*4-3)
280 CALL KEY(0,K,S):: IF S=0 THEN 280
290 IF K=13 THEN 450
300 IF K=12 THEN GOSUB 540 :: GOTO 280
310 IF K>11 THEN CALL HCHAR(R/2,C/2,K)::C=C+1::GOTO 360
* 315 IF K<8 THEN 270
320 ON K-7 GOTO 330,360,390,420
330 REM LEFT
340 C=C-1 :: IF C=0 THEN C=1
350 GOTO 270
360 REM RIGHT
370 C=C+1 :: IF C>64 THEN C=64
380 GOTO 270
390 REM DOWN
400 R=R+1 :: IF R=49 THEN R=48
410 GOTO 270
420 REM UP
430 R=R-1 :: IF R=0 THEN R=1
440 GOTO 270
450 REM PLOT DOT
460 RR=R/2 :: CC=C/2
470 RE=ABS(RR=INT(RR))
480 CE=ABS(CC=INT(CC))
* 490 D=128+2^(3-(RE*2+CE*1))
500 CALL GCHAR(RR,CC,CH)
510 IF CH<128 THEN CH=128
* 520 CALL HCHAR(RR,CC,(D XOR CH)+128)
530 GOTO 270
540 !PLACE YOUR SCREEN DUMP
550 !ROUTINE HERE!
560 !CALL LINK("DUMP")!DANNY MICHAEL SCREEN DUMP
570 RETURN

```

## SPIRAL

It has been pointed out that we somehow chopped line 130 in last month's "SPIRAL" program. The correct line should read as:

```

130 FOR A=C1 TO C2 :: CALL HCHAR(R1,A,CH) :: NEXT A ::
    R1=R1+1 :: FOR A=R1 TO R2 :: CALL VCHAR(A,C2,CH) ::
    NEXT A :: C2=C2-1

```