Ø45 8411

Boise 99'ers Computer Club

NOVEMBER 13, 1984

The Boise 99'ers Computer Club will hold its November meeting on Tuesday the 13th from 7:00 to 10:00 PM at the home of the Club President: Jerry Hough, 1331 Colorado Ave. Boise. Turn 3 blocks East of Broadway on Hale. The house is on the corner of Colorado and Hale.

The meeting place has been changed from Borah High school to the Hough house so we can use their computer for demonstrations, problem solving, and hands-on training.

We have received some more newsletters from other TI-99/4A Computer Clubs around the country that we will circulate at the meeting for your review. We also have many new programs added to our library. CHECK IT OUT...The price is right.

The meeting is for users of Texas Instruments' 99/4A Home Computers, and the public is welcome to attend. The following group objectives are defined in our operating constitution:

1. Education of members
2. Promote understanding and appreciation of TI micro-computers
3. Exchange of programs and ideas

We feel our most important purpose is to assist members in answering any questions they may have about TI micro-computers.

OFFICERS and COMMITTEE CHAIRPERSONS

PRESIDENT	Jerry S. Hough	344-1409
VICE-PRESIDENT	Joe D. Canning	342-1396
SECRETARY-TREASURER	Esther Smith	322-8004
LIBRARY COMMITTEE	Jeff Burkhardt	375-1968
EDITOR, NEWSLETTER	(((((open)))))	

ASSEMBLER SPECIAL INTREST GROUP

The special interest group for Assembly Language programing has been meeting Saturday mornings at Jerry Hough's home. The group has been studying the TMS 9900 Assembly Language using "Introduction to Assembly Language for the TI Home Computer" by Ralph Molesworth as a textbook. This is an excellent book which starts with the assumption that a person is reasonably familiar with the Basic language but knows nothing about Assembly Language.

There are several reasons for learning the Assembly Language. The most important is that programs written in Assembly can take full advantage of the tremendous power of the TMS 9900 chip. You are able to perform many operations directly that can only be done indirectly if at all in Console Basic or even Extended Basic. This leads to more direct lines of programing. It is also possible to write subprograms in the Assembly that can be called from Extended Basic to make more efficient and better running Extended Basic programs. Another reason is that the programs written in the Assembly run much faster than the same program in Basic or Extended Basic. Then there is one other reason. That is the personal satisfaction of being able to tell the computer exactly what you want it to do on a one to one basis. That is what you do as you tell it each step of the way what you want done and exactly how to do it. Your instructions in the Assembly Language program are translated in a one to one mapping into machine language instructions. You are no longer a slave to the programing language and its limitations. Programing in Assembly Language, you are in direct command of the computer and this gives one a feeling of mastery over the machine.

Anyone who has felt that the Basic or Extended Basic progams he has been writing are not doing what he wants them to, or that they just take too long to run, should investigate Assembly. The members of the Special Interest Group are making a good beginning in mastery of the language and the operation of the computer in this mode and will be able to help any newcomers. Those in the group are Jeff Burkhardt, Joe Canning, Dan Fullmer, Jerry Hough, Curtis Miller, and Bob Stanfield.

If you think that you would like to learn how to program in this fascinating language, contact Jerry Hough. His home telephone number is 344-1409 and at work he can be reached at 384-7738.

HEAR YE,
HERE YE! 1 AND 1 = 1

PRESIDENT'S CORNER

13 NOVEMBER 1984

>It's time to stand up and be counted. This year is coming to a close and a new year will be starting soon. That means your $12.00 Club assessment is due and the elections for new officers are near.

>Pay your assessment now. This will be your last notice. Continue to get your newsletter and be a part of the 1985 Boise 99'ers.

>Express your interest to the Nominating Committee to guide the Clubs destiny. New officers and committee's will be a part of the new year. You should be part of it, too

>NEW MEETING PLACE:
Last month we had a conflict at our Borah High school meeting room...It was Parents Day, and we were ousted. So, we all came over to my house. It worked out alright and we're going to do the same thing Nov. 17th. Call if you are lost...344-1409.

>ACCOLADES for Lou Duman for doing such a great job putting together the Club newsletter. All our thanks

>We need a newsletter editor. Any volunteers? This month was a cooperative effort of Jeff Burkhardt, Dan Fullmer, and myself. Thanks!!!

EXTENDED BASIC SUB-PROGRAMS

By: J. D. Canning

Owners of Texas Instruments Extended Basic programming language have a very powerful option available to them. I consider this option easier to use than subroutines and usually consume less memory space. The device I am referring to are sub-programs. I'm sure you are not new to sub-programs since TI console basic and extended basic are full of built-in sub-programs (i.e. CALL CLEAR, CALL HCHAR, CALL VCHAR, etc.).

Subroutines are usually used to perform repeated tasks within a program. Often one must assign all variables to be used in the subroutine before executing the GOSUB statement. This process of variable assignment takes up room that is probably not necessary. Sub-programs have the capability of passing variables to and from an operation that is defined by the sub-program.

Let's look briefly at CALL SOUND. It has the syntax of CALL SOUND(T,F,V). Where T is the duration or time of the sound; F is the frequency; and V is the volume. Once these parameters are properly assigned, the computer goes off into its deep dark internal parts and executes a collection of commands that create a sound and then returns to the main program.

Let's look at CALL KEY. It has the syntax of CALL KEY(KU,V,S). Where KU is the key unit number (see a previous newsletter for a discussion of key unit numbers); V is the return ASCII variable; and S determines the keyboard status. Once the key unit number has been properly assigned, the computer once again delves deep into its inner workings and performs two rather miraculous things. It momentarily scans the keyboard for any key that has been pressed. It then assigns the ASCII code for the key pressed, if any, to V, and then assigns a certain value to S depending on what happened when the CALL KEY was executed. It then returns control to the main program. The interesting thing to note is that the computer is assigning variables that may be used in your program from somewhere within itself; all from a basic statement or command.

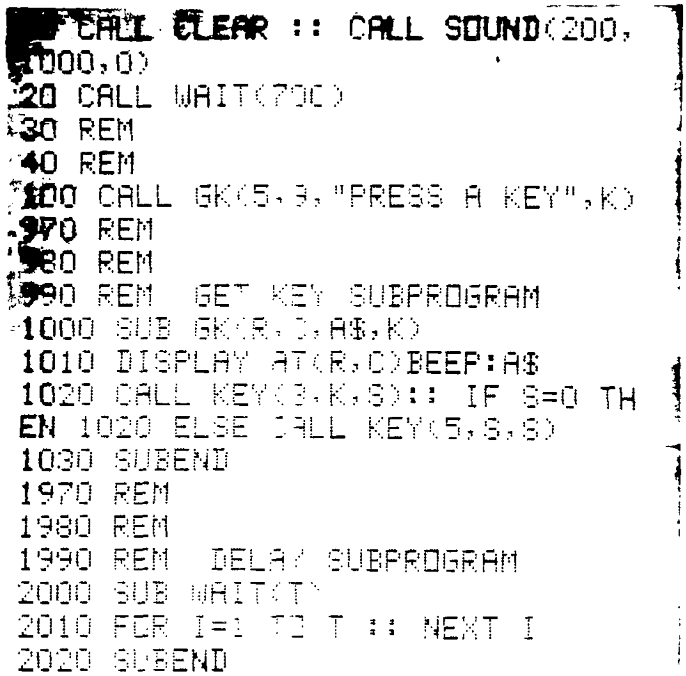
Now that we have discussed some built in sub-programs, let's look at some that we as users may create and customize. Below I have listed two sub-programs that might be helpful to show you how these and others may be designed. User designed sub-programs are called from the main program just like pre-defined sub-programs (i.e. CALL ------). The sub-program itself must be at the physical end of your main program, in other words actually occupy the highest line numbers. They have the general syntax of SUB ----- being the first line, where ----- is the sub-program's name and parameter list, if any. They must end with a SUBEND statement.

The sub-programs I have included are:

1. CALL WAIT(T) This sub-program causes the computer to pause or momentarily hold still for an amount of time determined by the variable T. This sub-program is the same as putting a FOR - NEXT loop in to create a delay.
2. CALL GK(R,C,A$,K) This sub-program is simply a glorified CALL KEY routine. When called, it will place a message defined by A$ at row R and column C on the monitor and suspend program execution until a key has been pressed. Then it will place the ASCII value of the key pressed in variable K and return to the main program.

I hope the listing given with this discussion will further explain sub-programs. The important thing to keep in mind is that variable names used in sub-programs are local. That is they are only used by the sub-program. Thus one variable name may be used once in your main program and once in each sub-program without conflict or affecting each other.

This is very handy since sub-programs that are used often may be included or merged with any program you may be composing without fear of duplicating variable names. The computer passes information or data to and from the sub-program by the position of the data in the statement, not the variable name.



```
[illegible] CALL CLEAR :: CALL SOUND(200,
1000,0)
20 CALL WAIT(700)
30 REM
40 REM
100 CALL GK(5,3,"PRESS A KEY",K)
970 REM
980 REM
990 REM  GET KEY SUBPROGRAM
1000 SUB GK(R,C,A$,K)
1010 DISPLAY AT(R,C)BEEP:A$
1020 CALL KEY(3,K,S):: IF S=0 TH
EN 1020 ELSE CALL KEY(5,S,S)
1030 SUBEND
1970 REM
1980 REM
1990 REM  DELAY SUBPROGRAM
2000 SUB WAIT(T)
2010 FOR I=1 TO T :: NEXT I
2020 SUBEND
```

****

TI KEYBOARDS

By: J. D. Canning

For those people that like to tinker with their computers internal workings and want a heck of a deal on the most abused part of your computer make a note of this. All Electronics Corporation in Los Angeles has in stock at this time original replacement keyboards for the TI 99/4A Home Computer. They are the newer gray components with updated microswitches over the tacky "wire" switches that my console contains. They are priced at an unbeatable $6.50 each or two for $11.00. The catalog number is KP-48 from catalog number 384.

They also have original sound video modulators for the TI computer at $10.00 each. The catalog number is AVMOD.

You may place an order at:

All Electronics Corporation
905 S. Vermont Ave.
Los Angeles, CA. 90006
1-800-826-5432

10/10/84

JEFF
Greetings Mr. Burkhardt,
Club LIBRARIAN

Thank You very much for responding to my TI FORTH DATA DISK offer. Enclosed is your disk with my FORTH work on it. Some of the programs, as you will see, are done by others, but all are Public Domain, and I've included them for completeness.

There are two ways of using this disk. The First and simplest is to BOOT FORTH in the usual manner, then switch to the Data Disk, and type in 'EMPTY-BUFFERS 1 LOAD ' this will mark the disk buffers as empty, so that no residue of the initial BOOT-UP screens remain, and load screens #1 and #2 from the Data Disk. These screens contain an Auto Loader that I recently wrote in response to some comments I have received in the past month. Just select the letter of the program that you want to run, and follow the prompts, placing the appropriate disk in Drive #1, either TI's System Disk, or my Data Disk. After a program loads and runs, a message is displayed telling you to FORGET something. This is there because I could not place the FORGET word inside a colon definition, otherwise, I would have done so to make it as painless and simple as possible. Type 'FMENU' to get the Auto Loader back.

The SECOND way to use the disk, is to do manually what the Auto Loader does for you. I am sorry to say that several of the things contained on the Data Disk must be used in this manner. Typically, as instructed on lines #1 of each screen, you would load the needed option(s) from the System Disk, then enter the VDP Mode that the word will use, for example, GRAPHICS2 for Bit-Map Mode, then load the screen which contains either the start of the program, or the Demo Word. Finally, execute the appropriate word, in the case of a DEMO it may be QUILT, or DOTS. In the case of a program, it's designed to run itself. The RUN WORD is the last contained on the last screen of the program by itself.

I am totally sold on TI FORTH. It gives me as a programmer a new freedom and flexibility that I had not thought would be possible in a computer language. I can program on the primitive assembler level, or, and at the same time, program on a high level, like BASIC. In fact, my productivity had increased considerably since I started in FORTH.

You will notice that a few new things are added to the disk, that are not mentioned in the letter to MICROpendium. These are REAL TIME CLOCK, NUKE ATTACK, AUTO LOADER, and HALF-FAST COPYER. The reason for this is because I added them after writing the letter to the compendium.

In closing, I hope that these programs are not only practical, in that they give you pleasure, but that they are also practical in that you may learn from them and GO FORTH !

Learn Long and Program
John J. Volk
Rt. #1 Box 291
Van Buren, Arkansas 72956

Dear Club members,

We have expanded our club library directory once again with the help of the Pocatello Computer Club and the Lehigh 99'er Computer Group. The Lehigh 99'er Computer Group has given our club some great public domain software which I have not entered in the directory. The following programs are late additions:

MASS-COPY (E/A, Do, Mem)
A super copy program by Steve Lawless, copies a single sided disk in three passes or less. Even Proprietary disks!

@XRUN (Ex, Do, Mem)
This demo program is a spectacular display of the color graphics capibility of the TI 99/4A. A must for every member of the club.

GOTHICPRNT (Ex, Do, Pr)
This program converts the Gemini 10X and compatible printers into a 27 character Gothic printer. This is nice!

USERGROUPS (TIWRT, Do, Mem)
Prints mail labels with the TI-WRITER module, Use edit side: .LF .CR in printer description.

Let's see some interest in our library, we have something for everyone. Plus we have revised our library policy!

Your Librarian,
Jeff Burkhardt

****

A NEW TI ONLY PUBLICATION

By: J. D. Canning

I recently received a sample copy of a new publication for owners of TI 99/4A computers. It was 32 pages in length and was filled with general information and hardware and software reviews. It lacked any programs that could just be keyed in.

If you are interested, the annual charge is $12.00 and you may place an order at:

MICROpendium
P.O. Box 1343
Round Rock, TX 78680
(512) 255-1512

IN DEPTH WITH ACCEPT AT

By: J. D. Canning

Users of TI's Extended Basic have vastly improved capibilities of screen I/O. DISPLAY AT, ACCEPT AT and IMAGE provide a means to not only enchance a program's general professional appearance, but will also allow options not easily or efficiently achieved in console basic.

Today let's review some of the attributes of ACCEPT AT. Its general syntax is:

```
ACCEPT {AT(r,c)VALIDATE(datatype)BEEP ERASE ALL SIZE(x):}variable
```

All of the items within the { } are optional and any one or combination of parameters may be utiliized to input information from the keyboard. Please note that just ACCEPT may be used without any options. This use places two edge characters at the lower right of the screen.

Now let's look at each optional parameter individually:

AT(r,c) : Places the cursor at the screen position specified by r (row) and c (column); then waits for keyboard input.

VALIDATE(datatype) : Will only permit certain type of keyboard entries. These types are:
UALPHA : Permits all uppercase alphabetic letters
DIGIT : Permits the number keys 0 through 9
NUMERIC : Permits the number keys 0 through 9 and ".", "+", "-", and "E" (E for exponential numbers)
String Expression : Will only permit certain keys (i.e. if "YN12" is specified, the computer will only accept the keys Y, N, 1 or 2)

It is interesting to point out that if validate is used say with (DIGIT,"YN") the computer will accept all digits plus the keys Y or N.

BEEP : Sounds a short beep

ERASE ALL : Clears the entire screen immediately prior to data acceptance

SIZE(x) : x determines the maximum number of characters that the computer will permit to be entered into the input variable

The numeric sign of the "x" within the size option is very important. If it is positive, the entire input field is erased prior to permitting any keyboard input. If it is negative, the input field is not erased. This permits the computer, if desired by the user, to accept characters that are already in the input field without the operator performing any task other than pressing enter. Or the operator may optionally change the information. Thus the 99/4A now has default inputs. The default value may be placed at the screen location of the upcoming accept statement prior to executing the accept statement. If the "size" is negative that value will not be erased and is eligible for input, presuming that it does not conflict with the parameters given in the validate clause.

The information presented above is all contained within the manual provided with Extended Basic. I hope that having a different explanation of some of the extremely powerful options of the ACCEPT AT statement has uncovered some uses that you had not realized. I know from experience that almost every word within the TI programming manuals have a definate purpose that may sometimes be forgotten.

TI Gaming

By

J. Burkhardt

This article is for all you TI-gamers with the following TI game modules: Munch Man, Alpiner, Moon Mine, and Hopper. These modules have a test mode built into them which you can access. You may ask yourself, "So what?" Well, if you're like me, you can make it to a certain higher level and then you run out of turns. This means you have to start all over from the beginning until you reach that level again. This can be time consuming and aggravating. Now that I have your interest, you may ask yourself what you have to do to access the test mode. The answer is simple, all you have to do is: When the title screen appears, type *#* fairly fast. You will then see prompts on the screen for the level of play, screen #, and the number of turns you want. Well, I hope you use this information to further your skills at TI-GAMING.

****

TIGERCUB SOFTWARE

By: J. D. Canning

Jim Peterson is the heart of Tigercub Software. As I understand Jim was once an employee of Texas Instruments working as a programmer for the TI-99/4A. He now is selling original software for the 99/4A at $3.00 a program. Here is your chance to obtain software at reasonable prices without joining a national users group or program exchange.

You may obtain one of Jim's catalogs by writing to:

Tigercub Software
156 Collingwood Ave.
Columbus, Ohio 43213

Send $1.00 for shipping and handling charges. This fee is refundable with the first order of software. Our computer club also has a copy of his catalog which may be perused at the monthly meetings.

****

Copied from Lehigh 99'er Computer Group

TMS 9900 Microprocessor Data Manual (MP001 REV A) This thirty-eight page Strunk and White of AL manages to get the instruction set documented in eleven pages. With good drawings yet. The better 99/4A AL books tend to simply layout this material in their typefaces.... Want one? Send to TI., POBox 225012, M/S 308, Dallas, Texas 75265.

HI-LIGHTS OF THE NEWS(letters)

>Lehigh 99'er Computer Group lists some public domain programs that we sent away for. See Jeff Burkhardt about the additions to our library. They also provide an Associate Membership to their club for $3.00. You'll get their newsletter for a year. Such a deal!! One of the better newsletters, I've seen. Their contributors are both knowledgeable and interesting, and there is a wealth of information in every issue.

>Central Iowa 99/4A Users Group tells us about a new book from the Computer Shopper magazine. It consists of a series of articles they ran in their magazine and they call it "The Innermost Secrets of the TI 99/4A". I ordered it 10/22/84 to see what it's all about.

>Milwaukee Area 99/4 Users Group mentioned that if you have a Gemini-10X printer that drops part of the first character sometimes. the company knows about the problem and if you'll call them with the serial number of your printer, they will send you a new print head. Well, mine's been like that for the year that I've had it, the repairman said it would cost to much to fix, so I've just grudgingly lived with the problem...Until I read the note from MAUG. I called the company, and now have my new print head. It works fine.

>LA 99ers Computer Group published a session with a BBS, and contained within was some good things to be said about the new software series from Scott, Foresman called PROBE which teaches BASIC programming to youngsters by demonstration. Also, someone said the MICROpendium publication is worth having, and someone else said that the CORCOMP Disk Controller card is not compatable with the FOUNDATION 128K card nor their own RS232 card.

>Attleboro User's Group reports that they "have come up with at least five hundred data bases which they can call and use". They even talk about a dating service bulletin board. Sounds like fun!!!

****

FOR SALE by Jerry Hough -- phone 344-1409

(1) TI LOGO - A personal programming language for all ages. One that will easily allow you to communicate with the TI-99/4A Home Computer in a totally new audiovisual realm of applied imagination, exploration, and self-discovery. Includes Solid State Command Module, a language book, and a book of instructions for the sample programs which are included on both disk and tape. --------------------> $50.00

(2) TI standalone 32K expansion RAM - Required for TI LOGO and many other more sophisticated programs. You don't have to buy the Peripheral Expansion Box. It plugs directly into the side of the TI-99/4A Home Computer. ------------> $100.00

(3) SST EXPANDED BASIC COMPILER SYSTEM - A set of programs and documentation intended to translate a Basic program into TMS9900 machine language. ----------------------> MAKE OFFER

****

These X-Basic instructions will demonstrate a neat trick of displaying messages on the screen: SCROLLING---right to left.

```
>100 CALL CLEAR
>200 MSG$="                            THIS IS A LONG MESSAGE TO
    DEMONSTRATE THE TECHINIQUE OF SCROLLING RIGHT TO LEFT."
>300 FOR I=1 TO LEN(MSG$)::DISPLAY AT(10,1)BEEP:SEG$(MSG$,I,28)::NEXT I
>400 GOTO 100
```