SAN DIEGO COMPUTER SOCIETY

TI-SIG NEWSLETTER

NOVEMBER 1988

NOTES FROM WOODY'S DESK

IN THIS MONTHS Newsletter, we continue Tony McGovern's tutorials on squeezing assembly language programs: See pages 3 and 4. Our John Johnson describes his trials and tribulations while loading his new Horizon ram-disk and his Myarc ram: Look on page 4. Lutz Winkler contributes a nice article on TI Disk File Formats: See page 5. Woody (that's me) tells the how and why of last month's fiasco while trying to print the newsletter: Look on page 2.

NEW ADDITIONS TO THE DISK LIBRARY. Lutz Winkler has furnished us with a copy of "COMICSHOW". While this new program (to us) can reside on one DSDD disk, as is customary our library will have it on four single-sided, single-density disks. Since the documentation is not very extensive, you may have to experiment a little to get the full benefit from the programs. For example, nothing tells you that by using a number key or the = (equal) key you can control the speed of the bouncing balls or the rotation of the cube. (These are animated pictures.) Lutz is writing to the author to suggest further docs for the program.

I SHOULD MENTION THAT the program is by Thomas Opheys of West Germany. He does NOT want any money for it, but is interested in getting a letter or diskette telling of our problems with his software or perhaps a new show made by us. He also would appreciate suggestions on how to make the program better.

LUTZ ALSO FURNISHED another disk: "The Best of UK#1", "The Best of UK#2", "GEE", and "SING". The programs on this disk are all Archived, so if you do not have an Archiver, now is the time to pick up a copy from the library. AND DO NOT FORGET TO SEND IN YOUR FAIRWARE FEE TO THE AUTHOR! NOTE: Our librarian may choose to un-archive this disk; that is his option.

(Continued on page 2)

REPORT FOR THE SDCS

At the October 18th meeting J.D. Johnson reported on the September North Park Recreation Council Meeting. The change in our meeting room from the Game Room to the Craft Room is now a fact.

Woody Wilson will assume the position of Treasurer in addition to that of Vice-president. Gil Pico turned over the funds and receipt book to Woody. There was $179.15 in the treasury. As of Nov. 6th, there is $227.15.

Roland Anderson, Fred Eylar, and Gil Pico have renewed their memberships.

Woody Wilson reported on the FALL-4-A-SHARE-FAIR that was held at Placentia on October 9th, 1989. (This was covered in the September newsletter.)

Woody gave an update on the progress of the TI FEST WEST. The entry fee will be $4.00 plus $1.00 for each family member over 15 years of age. Under 16 will be free. Tickets good for two days, Feb. 18th and 19th, 1989. Free coffee and doughnuts. Tables free to vendors and User Groups. If possible, room fees to be reimbursed to attendees. Tickets bought in advance (minimum 6) will be $3.00 ea. Children MUST be accompanied by an adult to attend. The SCCG hopes to have two computers set up in the Hospitality Room so that children can play games.

Gil Pico "dissected" Bud Archer's Axiom parallel interface. Apparently there were different versions of the interface since Woody mentioned that the one he had at one time was not made the same way.

J.D. Johnson showed his minimum system setup. It is really quite a compact system and can do most of the things that you need from a portable.

Woody Wilson demoed Ray Kazmer's new program called "ARTCON+". This program is now sold by TEX-COMP. MAX-RLE version 2 is also on the disk. The high light of the program is Ray's program called "SHOWSPRITE" that is a talking utility that is used to show which CHARacters were used, at any given spot in an XB program for Sprites, HCHARS, etc. A Speech Synthesizer is required.

Woody also demoed "MINDEX" BY Francisco Garcia. The program as shown was used to index MICROpendium.

(Continued from Page 1)

THE TWO FILES FROM ENGLAND consist of FAIRWARE programs. UK #1 is from Roland Trueman and has five games on it (as near as I can figure) while UK #2 is from David Vincent and appears to be a VERY elaborate maze game set in an Abbey. Since I am not much of a game player, I just took a quick look at the programs. They seem to be OK.

THE GEE PROGRAMS ARE from Paul A. Dams, Editor of the Miami User Group. He has furnished some G-routines written in the relatively new GEE graphics language. You have to have the GEE language disk in order to use these programs.

THE PROGRAM "SING" CONTAINS four songs by Terry Atkinson. The songs utilize the TI "TEXT TO SPEECH-ENGLISH" PROGRAM by permission of Texas Instruments. Don't expect to hear "Opera" on this disk; the songs leave MUCH to be desired but it is singing of a sort. Expect to hear "Octopus's Garden", "I Saw Her Standing There", "Here Comes The Sun", and "You're So Vain".

PERHAPS SOMEDAY WE WILL GET a speech and music program that will make the voice sound a little more natural. What ever happened to the "digitized" program we supposedly have?

==

ON NOVEMBER 5TH, EIGHT MEMBERS of the TI-SIG met at Woody's house for a workshop on defective equipment. Present were Roland Anderson, Bud Archer, Fred Eylar, Julia Flanagan, Waldo Hamilton, John Johnson, Mike Marlow, and Woody Wilson. It turned out to be a most enjoyable session. Disk drives, PE-Boxes, Cards, and cords were checked and in some cases repaired. Waldo and John did most of the repair work with the rest of the gang adding their help whenever possible. There were so many items to fix that another session will have to be held. (Incidentally, the "workshop" made me clean up the garage and put a new top on the workbench. The "durn" thing looks so clean that I am afraid to use it!)

Before the next session is held, we hope to build some test equipment. John Willforth of the West Penn U.G. has been writing a series of articles on Disk Drives. In his second article he shows a tool to exercise and test most drives. John and Waldo are going to build one so perhaps we will be able to solve some of our disk drive problems.

==

SOME OF YOU MAY REMEMBER that last month I mentioned the problem I was having in printing the newsletter. Well, here is the long, long, tale of woe:

When I do the newsletter, I usually save the pages on to a new initialized disk before I print them. The format used is "OCT88NL;P1" to "OCT88NL;P5" (page 6 is already made up by Lutz Winkler).

Being inherently lazy, when I save each program I just change the LAST character of the filename. (Since FUNNELWEB keeps the filename in its "mailbox".) This is where the trouble started. Instead of typing the letter "O" I used the numeral "0" (zero). Normally if you do such a stupid thing you notice the difference when you catalog the disk. In this case all of the first letters of the filenames looked the same, so it was easy to overlook.

The next monster reared its head in the Utility program. Here is a program that I have been using for several years with no trouble, but one of Murphy's laws (I don't know its number, but it must be there!) finally caught up with me: "If you think you are smart, I'll make you stupid." Let me show you the program lines as they existed on that fateful night. (The old program had line numbers 100 to 650. This is just a piece of it.)

```
400 ! *** READ FILE *******
410 IMAGE "Line No.: ###
420 OPEN #1:F$
430 FOR I=1 TO 141 :: DISPLAY AT(14,1):
    USING 410:I
440 LINPUT #1:L$(I)
450 IF EOF(1) THEN 500
460 IF ASC(L$(I))>127 THEN 500
470 IF I>LIMIT THEN 490
480 NEXT I :: GOTO 500
490 DISPLAY AT(14,1)BEEP:"File is too
    long :: CLOSE #1 :: GOTO 330
500 CLOSE #1 :: RETURN
```

Now look on page 5 of this newsletter and check the defaults for a Sequential Display file; they are UPDATE, VARIABLE, 80 in this case. Well, the UPDATE is what cooked my goose (and it wasn't even Thanksgiving). In line 420 we open a file using the name I type in (In this case I typed "OCT88NL;P1" using the letter "O"). The program tried to find a file of that name, could not, so wrote an UPDATE file of 1 sector to the disk using the name I had entered. The program then failed with an error code 25 in line 440. That code tells me I am attempting to read past the end of a file. Actually I do not have a file!

The easy way to fix this would be to add the word "INPUT" to line 420. I did a little more. I added an error trap as well. Now, no more 1 sector files!

NOTE: THIS IS A REPLACEMENT PAGE. DUE TO A FORMATTING ERROR, PAGES 3 AND 4 PRINTED INCORRECTLY. PLEASE REPLACE!

THIS MONTH WE CONTINUE Tony McGovern's Tutorials on how to save bytes in your assembly language programs.

ASSEMBLY SQUEEZING Part 3

Tony McGovern -- Funnelweb Farm

Blocks to be moved aren't always of known length. A data type found in the many languages is the string, a collection of characters together with a length indicator. The most familiar form is that in TI Basic where a length byte precedes the number of bytes indicated by the length, up to a maximum of 255. Other string conventions are also used in various computer languages, but let's look at this one for the moment.

Your assembly program has to move STR1 to another STR2. Two things are clear from the start. Firstly the number of bytes to be moved has to be read from the first string (if it were already known the a simple block move would do) and secondly byte move instructions must be used as the byte is the unit that makes up strings.

A straightforward pass at the code with this in mind is

```
        LI   R0,STR1
        LI   R1,STR1
        MOVB *R0,R2
        SRL  R1,8
        INC  R8
LOOP    MOVB *R0+,*R1+
        DEC  R2
        JGT  LOOP
```

The length byte is read into a register, converted to a count word, and incremented to cover the length byte itself. This code takes 10 words and uses 3 registers. That's pretty expensive so let's make it shorter.

```
        MOVB @STR1,R1
        SRL  R1,8
        INC  R1
LOOP    MOVB @STR1-1(R1),@STR2-1(R1)
        DEC  R1
        JGT  LOOP
```

This uses the same indexed addressing in the loop as we did for the block move last time. It now takes 9 words and uses only 1 register. It is possible to shorten it by 1 word more by using a special property of the DEC instruction. DEC always sets the Carry status bit except when changing from zero to minus one (see the E/A Manual p100).

```
        MOVB @STR1,R1
        SRL  R1,8
LOOP    MOVB @STR1(R1),@STR2(R1)
        DEC  R1
        JOC  LOOP
```

The loop now executes one more time when R1 is zero, so the INC instruction is no longer necessary and the indexed addressing offset is also adjusted. So

ASSEMBLY SQUEEZING Part 4

Tony McGovern -- Funnelweb Farm

A second form of string that is sometimes found doesn't use a leading length indicator byte or word, but uses a null byte to mark the end of the string. This means that the string can be of any length but can't contain any null bytes as part of the string. Once again we look at the problem of moving a string from one known address to another.

```
        LI   R0,STR1
        LI   R1,STR1
LOOP    MOVB *R0+,*R1+
        JNE  LOOP
```

You can see why this form of string is used as the bytes moved set the termination condition as they are moved, and a separate countdown isn't needed. The code above takes only 6 words and uses 2 registers, but it is possible to do better still by using indexed addressing again.

```
        SETO R1
LOOP    INC  R1
        MOVB @STR1(R1),@STR2(R1)
        JNE  LOOP
```

The SETO initializes the index so that on the first pass through the loop the first byte is moved. This still uses 6 words but now trashes only one register. In either case the null byte is the last to be transferred and does not have to be attended to separately.

While we are at it, suppose we were reading in a definite number of bytes, say from GROM, and wanted to terminate them by a null byte to make this kind of string in CPU RAM.

(Continued on page 4)

(Continued from page 3)

```
LOOP   MOVB @GRMRD,*R1+
       DEC  R2
       JGT  LOOP
       MOVB @NULL,*R1
```

NULL is the label of a null data byte. You can save a word by replacing the last instruction with

```
       SB   *R1,*R1
```

Subtraction of a byte from itself is a handy way to clear a single byte as the CLR instruction works on full words only. In this instance

```
       MOVB R2,*R1
```

would also do the job as the loop exit condition leaves R2 containing a null word. From the point of view of someone trying to read the code, and that includes yourself at a later date, the previous form is preferable as it makes what is being done more obvious.

"HORIZON RAM + MYARC 512 + FUNNELWEB" BY JOHN D. JOHNSON IT'S AWESOME: Take one Horizon Ram Disk , add a Myarc 512K card and throw in the McGovern's Funnelweb (TI Writer) software and your little TI really comes alive.

AS THE CAREFUL READER already knows, the secret ingredient in this computer stew is the ROS program by John Johnson of the Miami Users Group. Boy, is he famous. That's my name too and when I meet anyone who's got a TI, they ask "Are you 'the' John Johnson who writes the programs?" I sadly shake my head and say, "No, I'm not that smart!"

THE ROS John Johnson designed only comes with the Horizon Ram Disk. It is the special Ram (Disk) Operating System which makes the HRD work -- and a whole lot more.

THE HRD can be purchased with 1MEG of ram on board. I already had the Myarc 512K card and decided to try it with the HRD. So far, it's a wedding made in heaven. YEA, verily, the wise men of Byte-ania say it is so! I only put together the minimum HRD card: six ram chips and that adds up to the equivalen of a SSSD floppy. Together with my Myarc I get a hefty 704K of ram including an 80K of printer buffer built into the Myarc. (Boy, did Lou Phillips miss the boat with his Ram Card. If he had put an operating system into it, his Ram card would no doubt be 'the' Ram Card, not the HRD. Let's not be too hard on Phillips: Corcomp also missed the boat, not to mention old Mr. T.I.)

CONFIGURING BOTH the ROS and the Funnelweb is a bit of a chore. I finally got them both sitting side by side as drive number three (HRD) and drive four (the Myarc). I have Funnelweb and a whole raft of games reposing in the Myarc. Both have DM1000 which runs from both. But best from the HRD -- it drops some functions when booted from the Myarc. I tried running FAST-TERM out of the Myarc but it didn't work. I (pause) just put DSKU in the Myarc and it worked -- partially. It took a little finesse and it was only a partial beta test. Trying to run the DSKU out of the Myarc "as drive" #4 didn't work. (Maybe it will but I didn't get it to.) But I simply went into basic and renumbered the Myarc as Drive #1 and the DSKU booted right in. But, the HRD didn't like the mix when I tried to access it from DSKU, as booted from the Myarc! The computer locked up when I tried it from Drive #4. But the ROS bounced right back after the restart and no files were lost nor damaged. That is a fundamental tribute to the Horizon Ram Disk's designers and to (the Miami) John Johnson's programming skill. As everyone in San Diego knows, if (the San Diego) John Johnson can't kill it, its bullet-proof.

COINCIDENTALLY: To keep the Myarc up between computing sessions, I use a 12VDC supply with a 7812 voltage regulator and a 470 uf capacitor and have some 6VDC batteries in series (big yeller ones) backing that up. If you rig up something like this, put an extra large heat sink on the 7812 because it gets hot only in the refresh -- call that repose -- mode, when the P-Box is off.

NOW, JUST WHY IS THE ROS so sweet? Because it is menu directed with just a few easy to remember basic "CALLS" which make it very versatile. Sweetest of all, is having all those programs on tap in an instant and the auto reboot of the menu. Smooth as silk is it. Love it. Funnelweb (rechristened) Funnyweb on my menu, is the frosting on the cake. And let us not forget DM1000!

NOTE FROM WOODY: Use a sector editor and change the drive #s in the affected programs to match the drive #s of the rams in which the programs reside! No trouble

NOTE FROM WOODY: You will notice that this page is short a few lines. Rather than re-write the articles, I have corrected only the formatter errors. It seems that somehow (gremlins again!) an old copy of the formatter was used. It does not LIKE the @ symbol and removes it and then duplicates the line three more times. So this is your corrected copy of pages 3 and 4 of the November 1988 TI-SIG newsletter.

TI DISK FILE FORMATS

```
File Organization
----------------->        Relative                         SEQUENTIAL
                             !                                  !
                      ---------------                   ---------------
File Type             !            !                    !            !
-------->          Internal     DISPLAY              Internal     DISPLAY
                      !            !                    !            !
                      ---------------                   ---------------
                             !                                  !
                     -----------------              -------------------------
Open Mode            !       !      !               !       !      !       !
-------->          Input  Output  UPDATE      Input  Output  UPDATE  Append
                     !       !      !           !       !      !       !
                     -----------------          -----------------      !
                             !                          !              !
                             !                          ------------ ---
Record Type                  !                          !         !  !
---------->                FIXED                      FIXED     VARIABLE

Record Length
------------->   Default  *)  80                        80        80
                 Maximum **) 255                       255       254
```

*) Applies to disk and RS232. Cassette & thermal printer are 64 & 32, respectively.

**) Note that variable files are limited to one less because sequential files need one byte for record-length.

The above schematic representation was originally done for my own edification. The explanations in the DISK MEMORY SYSTEM manual - as textual material - were ok but difficult to read and interpret. I hope those who, like myself, have had a problem understanding disk file setup (TI-style) will find this information helpful.

It provides a quick overview of the types of files available, default settings which are automatic (unless otherwise specified) and should make it easier to understand the contents of the TI manual referred to above as well as the information contained in the Extended Basic manual.

A quick glance at the chart has helped me many times to avoid using parameters which are defaults anyway (and thus save bytes). However, this does not mean that occasionally I have left one out which should have been there. After all, that would be too much to hope for.

Before I give a few examples of how to open files with only the minimun specifications, there needs to be an explanation why you will not be able to use the full 254 record-length for sequential/ variable files from Extended Basic. This is due to the maximum line length of XB. If you use a separate line for a string, say X$, you will find that the maximum length XB will accept is 154 characters. Any more and you'll get "LINE TOO LONG".

Now to the examples. To open a file, the OPEN statement can be very short or also very long, as we shall see.

```
OPEN #1:"DSKn.filename" is the same as
OPEN #1:"DSKn.filename",SEQUENTIAL, DIS-
        PLAY, UPDATE, VARIABLE 80
```

```
OPEN #1:"DSKn.filename",RELATIVE equals
OPEN #1:"DSKn.filename",RELATIVE, DIS-
        PLAY,UPDATE,FIXED 80
```

```
OPEN #1:"DSKn.filename",APPEND equals
OPEN #1:"DSKn.filename",SEQUENTIAL, DIS-
        PLAY,APPEND,VARIABLE 80
```

Note that only sequential variable files can be opened in the APPEND mode regardless of type (INTERNAL or DISPLAY).

There are times though, when none of the defaults apply and that means no shortcut. There is no way around entering

```
OPEN #1:"DSKn.filename",RELATIVE,INTER-
       NAL,INPUT,FIXED 124
```

if that's the type of file you need.

These examples should suffice and give you an idea of how to rid your programs of excess verbiage. All you have to know is what type of file you want to use and look for the parameters shown in caps. Then omit them.

Good luck.
EOF/LW

SAN DIEGO COMPUTER SOCIETY

TI-SIG

NOVEMBER 1988 NEWSLETTER

The San Diego TI-SIG is a special interest group of the San Diego Computer Society. It is neither affiliated with nor sponsored by TEXAS INSTRUMENTS, INC. We acknowledge all registrations, copyrights and other legal rights of the companies/products mentioned in this newsletter. Such references do not imply an endorsement or condemnation of these companies/products. The opinions expressed by the authors are not necessarily those held by TI-SIG or its officers. Articles from this newsletter may be reprinted, provided their source is credited.

Meetings: 3rd Tuesday of each month, at 7 P.M., in the Crafts Room of the North Park Recreation Center, 4044 Idaho St., San Diego

TI-SIG 1988/89 OFFICERS

President: John Johnson	297-9740
Vice Pres: Woody Wilson	264-6515
Treasurer:	
Secretary: Lutz Winkler	277-4437

TI-SIG
San Diego Computer Society
P.O. Box 5263
San Diego, CA 92105

TI-HOME COMPUTER GROUP
P.O. BOX 29863
DALLAS TX 75229