THE [illegible]

[illegible]

FROM THE SOUTHERN NEVADA USERS' GROUP

Vol. 6 — No. 9 September, 1988

NEXT MEETING

MONDAY, September 12, 1988 — 6:30 PM

NEVADA POWER BUILDING MEETING ROOM

[illegible] West Sahara, Las Vegas, Nevada

President's Message

Another month has passed and a few minor changes have been implemented by the new officers. From now on Cindy Mitchell will be giving you the rundown on the upcoming monthly meeting. This will free this space for announcements of new products, upcoming events, articles of interest, and in general allow me to babble, banter, burble, blurt and blab without fear of incurring Rudy's wrath for taking up too much precious space on the front page. Another exciting change will be starting in October. S.N.U.G. will be purchasing software to demonstrate at the monthly meetings, and then add it to our slightly stale raffle pot! This will allow members to examine potential purchases first hand, rather than taking the word of a reviewer as to it's worth, and who knows, you might even win something that you would buy! (It could happen...) Cindy is currently working up a list of potential purchases so if you have a preference, ask her to put it on her list. Software getting the most votes will be shown first. John Martin has recently purchased a copying machine and has agreed to let the group use it for the newsletter. We will pay for paper, toner, and maintenance with club funds, but we should still see a substantial savings in the cost of the newsletter. Thank John. (If you need anything copied, call John... grin.) This month I've submitted two articles for the newsletter. Opposite ends of the TI spectrum are addressed although both deal with fundamentals of computer use. If you would like to see an article on a particular subject, bring it up at the meeting. One member spoke up and PRESTO, an article appeared. Better still! Write about something that interests you and submit it to Rudy. (My fingers are worn out!)

Bob

LIBRARIAN'S REPORT (9/88)

WOW!! Do we have programs, or what? This is great. I started with a console, a TV and Beginners Basic Tutor and look at me now. I've got ten disk caddies full of disks that are full of programs for this little TI. I'm in HACKERS HEAVEN, surrounded by magnetic film!

So, I sez to myself, self what are we gonna put on the Disk O' the Month? Hmmm.... It's kinda like request time. I feel like a disc-jockey. (Get it!)

Well this month, we have Archiver version 3.02 for the people that like saving space on the disks and sending those programs over the telephone lines. Barry Boone has enhanced a program that I thought was unbeatable the way it was. Well, he out did himself this time. The program is compatible with your previously compressed and packed programs. This one comes with a little bit more documentation than ver. 2.4 did.

Remember Jawbreaker from last month's DOM, well I've got another EA option 5 game called Crossfire! This is cute.

Rudy Johnson demonstrated the cassette rapid loader last month at the membership meeting. It is also on this month's DOM.

Cindy Mitchell requested Picasso and I have included it. Use your new version of Archiver to decompress and unpack it!

Print is a utility for the Seikosha printer user. I haven't got one of those but it looked worthwhile.

Also included on this month's DOM is a TI-WRITER CHARAI file that gives the word processor an inverse video of the

control characters such as carriage return, line feed, etc. This really helps me out when I am using FunnelWriter. The screen shows all the control characters with a reverse of the actual screen colors making them stand out so that the user does miss an embedded character and mess up printing out that important document you just finished.

That's all until next month!

<< Dee >>

==========

Creating a batch file is E A S Y !

I saw a need for a MDOS command which when executed would copy a program or progams from one disk to another and then erase the file(s) from the first disk. After a little experimentation the final product (below) was produced. I named the batch file MOV because that is exactly what it does, it moves files.

(Editor's Note - The following text is in 60 column format. It originally was in 80 column format. The lines that are wrapped here are indented 2 spaces to the right. When typing them in they should not be wrapped, but cofntinued on the previous line to 80 columns!)

```
1.  ECHO OFF
    CLS
    ECHO FORMAT IS: MOV X:FILENAME Y:[new filename if
      entered]
    ECHO
--  ECHO FILE IN DRIVE X: WILL BE COPIED TO DRIVE Y:[new
      filename]
    ECHO
    ECHO BATCH COPY FORMAT: MOV X:%1 Y: -!- DISKCOPY
      FORMAT: MOV X:% Y:
    ECHO_____________________________
    ECHO FILE(s) IN DRIVE X: CAN THEN BE DELETED WITH A
      KEYPRESS
10. IF "%1"=="" GOTO END
    IF "%2"=="" GOTO END
    ECHO
    ECHO
    ECHO
--  ECHO
    ECHO TO COPY %1 TO %2 -!- ( C ABORTS)
    ECHO
    PAUSE
    ECHO
20. CLS
    COPY %1 %2
    ECHO
    ECHO
    ECHO
--  ECHO
    ECHO
    ECHO
    ECHO TO ERASE %1 -!- ( C ABORTS)
30. ECHO
    PAUSE
    ECHO
    CLS
    ERASE %1
--  ECHO
    PAUSE
    CLS
    :END
```

The original program was MUCH shorter. It functioned just as well but the possibility of operator error causing a deleted file loomed into the foreground. (I did it.) Fortunately the file that disappeared was named TEST and the contents read "THIS IS A TEST."
The easy way to make a quick test file follows:

From the drive prompt (in MDOS) type:
COPY CON TEST and press ENTER.
THIS IS A TEST (or whatever) and press ENTER.
Press CONTROL Z and the file will be written to the default (prompt) drive in DIS/VAR 80 format. A short batch file can also be typed in and saved this way, but the above file is too long and if you make a mistake you must start over since COPY CON does not allow full screen editing.

% What the batch commands mean %
Line 1 tells MDOS not to show the CLS command on screen.
Line 2 clears the screen.
Lines 3-9 display instructions and blank lines at the top of the screen. ECHO by itself displays a blank line.
Lines 10 and 11 check to see that the user entered parameters for the copy routine. %1 is a user variable for (source drive:filename). %2 is the variable for (target drive:filename). Yes, you can send the file to the target drive with a new filename and it will still be the same file. If you omit either of the variables the program branches to the :END LABEL and MDOS skips the following instructions. Nothing but the instructions and prompt are printed on screen and the program ends.
Lines 12-19 tell the user which files will be copied, give him a chance to change his mind and wait for him to press a key. The PAUSE command displays a little message of it's own, and waits for the keypress before continuing.
Line 18 does the actual copying.
Lines 20-33 warn the user that the source file is about to be erased and give him a chance to change his mind by pressing CONTROL C.
Line 34 deletes the source file.
Lines 35 and 36 let the user examine the screen full of garbage which tells him what has been done. (superfluous)
Line 37 cleans up the screen for exit back to MDOS.
Line 38 is the lable for emergency exit. You do not have to have an :END lable to exit though, after the last instruction is executed MDOS automatically exits.

A few words of CAUTION! Entering MOV A:% B: ILL copy a disk and IF the COPY routine finds errors it will quit copying

and skip to the ERASE routine! This means that you could delete the disk before you back it up. You could add another line or two of instructions saying DON'T DO THIS and DON'T DO THAT, but people tend to not read L O N G instructions. Perhaps the next version of MDOS will have routines that allow better error checking but for now, read the screen prompts and all will be well.

When writing batch files you should use the Program Edit mode of Myword so that carriage returns are not added to the commands. Also, make sure that the "*End of file*" marker has no blank lines before it so that the last command is the last line of the file. Save the file with Print File and not Save File (with the name MOV) if you are using the regular editor, or tab and margin settings will be added to the end of the file and create problems. One more thing, don't use in the line numbers or spaces before the commands, MDOS doesn't like them...

Bob Sherburne

==========

GETTING THE MOST FROM YOUR CASSETTE SYSTEM - BY MICKEY SCHMITT NUMBER 2
LOADING AND SAVING PROGRAMS

WHILE LOADING AND SAVING PROGRAMS WITH THE USE OF A CASSETTE RECORDER IS NOT A DIFFICULT PROCESS IN ITSELF - READING AND UNDERSTANDING THE INSTRUCTIONS FOR THE VERY FIRST TIME CAN BE QUITE CONFUSING. WITH THAT THOUGHT IN MIND I HAVE TRIED TO KEEP THE INSTRUCTIONS AS SIMPLE AS POSSIBLE.

INSTRUCTIONS FOR LOADING PROGRAMS:

1. TYPE: OLD CS1
2. THEN: PRESS ENTER
3. FOLLOW THE DIRECTIONS AS THEY APPEAR ON YOUR MONITOR OR TV SCREEN:
   - 3.1 * [illegible] CS1
     THEN PRESS ENTER
   - 3.2 * PRESS CASSETTE PLAY CS1
     THEN PRESS ENTER
   - 3.3 COMPUTER DISPLAYS MESSAGE:
     * READING
   - 3.4 COMPUTER DISPLAYS MESSAGE:
     * DATA OK
   - 3.5 * PRESS CASSETTE STOP CS1
     THEN PRESS ENTER
4. WAIT FOR THE FLASHING CURSOR TO APPEAR IN THE LOWER LEFT-HAND CORNER OF YOUR MONITOR OR TV SCREEN
5. TYPE: RUN
6. THEN: PRESS ENTER

INSTRUCTIONS FOR SAVING PROGRAMS:

1. TYPE: SAVE CS1
2. THEN: PRESS ENTER
3. FOLLOW THE DIRECTIONS AS THEY APPEAR ON YOUR MONITOR OR TV SCREEN:
   - 3.1 * REWIND CASSETTE TAPE CS1
     THEN PRESS ENTER
   - 3.2 * PRESS CASSETTE RECORD CS1
     THEN PRESS ENTER
   - 3.3 COMPUTER DISPLAYS MESSAGE:
     * RECORDING
   - 3.4 * PRESS CASSETTE STOP CS1
     THEN PRESS ENTER
4. YOUR PROGRAM IS NOW SAVED - BUT YOU SHOULD GET INTO THE HABIT OF CHECKING ALL YOUR PROGRAMS TO BE SURE THAT THEY WERE SAVED WITHOUT ERROR.
5. CONTINUE TO FOLLOW THE DIRECTIONS AS THEY APPEAR ON YOUR MONITOR OR TV SCREEN:
   - 5.1 COMPUTER DISPLAYS MESSAGE:
     * CHECK TAPE (Y OR N)?
   - 5.2 TYPE: Y
   - 5.3 THEN: PRESS ENTER
   - 5.4 * REWIND CASSETTE TAPE CS1
     THEN PRESS ENTER
   - 5.5 * PRESS CASSETTE PLAY CS1
     THEN PRESS ENTER
   - 5.6 COMPUTER DISPLAYS MESSAGE:
     * CHECKING
   - 5.7 COMPUTER DISPLAYS MESSAGE:
     * DATA OK
   - 5.8 * PRESS CASSETTE STOP
     THEN PRESS ENTER
6. YOUR PROGRAM IS NOW SAVED - SAFELY AND WITHOUT ERROR. THAT'S ALL THERE IS TO IT!

NEXT MONTH'S TOPIC WILL BE KEEPING YOUR CASSETTE TAPES AND PROGRAMS ORGANIZED.

IF YOU NEED ANY HELP UNDERSTANDING HOW TO LOAD AND SAVE PROGRAMS - JUST GIVE ME A CALL (412-335-0163) AND I'LL TRY TO HELP.

==========

GETTING LOADED

At last nights meeting I asked the membership if they had ideas or topics for newsletter articles which they would like to see written. The suggestion to write an article on how to "load programs" came from one of our newer members. After thinking about the trouble I had five years ago trying to figure out the mysteries of getting programs to run, I decided to do it...

* BASIC programs are labeled PROGRAM in a directory. Almost all will load from console basic using OLD DSK1.FILENAME. (After loading them from BASIC OR EXTENDED BASIC type RUN.) If an error occurs while trying to run the program, list the offending line number. If the line contains a CALL LOAD or CALL LINK command, and there is only one command per line then the program must be loaded from the BASIC supplied with the Editor Assembler cartridge. (I will omit CorComp commands since no programs seem to have been written for them.) If the listing is extremly slow and strange

characters appear on screen then the program is probably EXTENDED BASIC and must be loaded and run with that module.

* EXTENDED BASIC (XB) programs also show PROGRAM on directories and load with the OLD DSK1.FILENAME command. (Type RUN) One program may be named LOAD on the disk. When the key is pressed to select XB, XB turns on disk drive 1 and looks for a program named LOAD. If it finds it, it runs it. If not XB will give you the READY sign and wait for a command. XB will run some BASIC programs but if characters above 143 are redefined or a CALL LOAD is executed without a prior CALL INIT the program will crash with a BAD VALUE or SYNTAX error. Again, edit the line where the break occurs and see if a CALL CHAR(x,"abcdetc.") sits by itself on that line. If it does or if there is only one command per line the program is probably BASIC.

* XB MERGE files are shown as DIS/VAR 163 even though they are programs, and can be loaded and run using the format MERGE DSK1.FILENAME (See MERGE in XB manual, and don't expect much since MERGE is used as a programming tool.)

* GIANT XB files (more than 50 sectors) are shown as INT/VAR 254. They will load and run normally from XB.

* EDITOR/ASSEMBLER (EA) OPTION 5 program image files will show PROGRAM in directories also and they will usually be 33 sectors in length with the last letter or number of the group of files incremented. (i.e. TEST TESU TESV or UTIL1 UTIL2 etc.) These files will NOT even load with XB or BASIC and you will receive an I/O ERROR 50 error message. If you don't have EA then try Funlwriter or TI-Writer utility loaders. If the length is longer than 33 sectors then the file is probably used as data for a graphics program and will not run.

* EDITOR/ASSEMBLER OPION 3 files show DIS/FIX 80. The first type autostarts when DSK1.FILENAME is typed in, the other must also have a PROGRAM (link) name entered before it will run. (Try entering START and press ENTER) The second type can sometimes be run from XB using: CALL INIT :: CALL LOAD("DSK1.FILENAME") :: CALL LINK("STARTWORD"), or BASIC using: CALL LOAD("DSK1.FILENAME") on one line and CALL LINK("STARTWORD") later on. So recatalog the disk and look for a short PROGRAM with a related name. LIST it with XB and look for the CALL LOAD("DSK1.fileUwant2run") command. If you find it, run it from XB. If you get a SYNTAX ERROR, use EA BASIC and try again.

* CARTRIDGES which are saved on disk for use with Gram Kracker, 9640 etc. will directory as PROGRAM. They are almost always 34 sectors (some are shorter) in length, have incrementing file names, and can only be run from cartridge loaders.

* TEXT FILES show DIS/VAR 80 on the directory. These files can be loaded into TI-Writer (or other DIS/VAR 80 readers) and examined or "TYPE"ed from MDOS. These files cannot be loaded and run, however some programs such as MAX/RLE and PICASSO store their pictures in this format.

* ARCHIVED files are stored as DIS/FIX 128 P and cannot be run until they are unpacked. If no "P" (protected) is found then they must be uncompressed and then unpacked before you can use them.

* OTHER file types are data for a program, and cannot be run.

I know I have left out a few things (you long-time users will spot them right away but why confuse matters even more!) so if you still can't get a program to run, call Dee Wellman! (It's probably his fault since we don't have Bob Bieber to pick on anymore. (smile...) Now if I can just figure out how to save this file and leave MYWORD...

BOB SHERBURNE

==========

MULTI-COMM ?

by Dee Wellman

If you want a good, very inexpensive experience on a multi-user Bulletin Board, try MULTI-COMM.

This BBS is a local owned system on-line 24 hours a day. The system supports up to 24 users at one time! I have found it very entertaining.

It supports:

Electronic Mail	24 user Teleconferencing
Poker and Casino	Chess
Galactic Empire	Infinity Complex
Conquest of the Galaxy	300/1200/2400 baud (2400?)

with more coming!

The system consists of:

1 GST IBM AT 12mhz/40 meg hrddsk/1 1.2meg floppy
1 IBM Turbo clone/40 meg hrddsk/1 360k floppy
1 360/20 AT 20mhz/300 meg hrddsk/2 1.2 meg floppies
1 Multisync II VGA monitor
1 Power Supply (24hr. Battery Backup)
2 16 user MODEM cards (32 possible lines)

E-MAIL speaks for it's self, similar to SNUG's E-Mail, POKER is 5 card draw, and it only accomodates 6 players at a time, the pace is furious, the bets are big. CHESS speaks for it's self. GALACTIC EMPIRE, well now, you start with a spaceship stocked with men, fighters, food, etc. Your ship has missiles, torpedoes, phasers and all the things that made the Starship Enterprise great. Then there are planets that the commander of the spaceship must settle on and build more supplies. (Believe me you'll need them) Then there are your opponents such as DEATHSTAR, DRAGON, LOKI, GRIFFEN and me, DEEDELDOO (the Pirate). I love this game! With up to 24 players at a time flying through the Galaxy the dodging of missiles, torpedoes and phasers is fast and furious. This is a great way to enhance your typing skills and use your

wits to outfox the enemy, because in this game everyone is the enemy.

INFINITY COMPLEX is a brand new game just getting started. You are locked into an outpost on a planet, the outpost is INFINITY COMPLEX. you are controlled somewhat by the [illegible] computer MOTHER CONTROL. You must find a way to repair the computer while fighting off intruders and other players. Form a team and conquer. CONQUEST of the GALAXY is a multi player game that requires your skill at manipulating your planets resources and cash flow to outlast or conquer other planets. If you like being a brute and love pushing people and worlds around this is your game!

MULTI-COMM's rates are 25 cents per hour, with a $5.00 minimum to bring you on live. Live time players can utilize all the aspects of MULTI-COMM, but a user can play in some games without being live. The BBS automatically logs the non-live user off at the end of 15 minutes on-line time. That 15 minutes can be used to set up a REGISTER of your user log similar to that of SNUG's. I hope I have made you interested enough to call and check it out. You won't regret it!

MULTI-COMM
P.O. Box 81693
Las Vegas, Nv. 89180-1693
362-9224

rates:

$5.00 --- 20 hours
$10.00 --- 40 hours
$60.00 --- 300 hours

SECRETARY'S REPORT

MINUTES
AUGUST 8, 1988

The August meeting of SNUG was called to order at 7:PM by our new President, Bob Sherburne. Eighteen (18) members were present.

Bobs first order of business was a poll of the members' systems and special interests. Of the members present, nine members have expanded systems, three have expanded systems without a modem, and one with only a cassette recorder.

The majority of members were interested in learning how to identify and load the different type of files. There was also a discussion to help those members starting out in Basic and Extended Basic, Multiplan on a one to one basis or group session. It may be possible to hold these meetings here at the Nevada Power Co.

George Jilly brought in an Avatex 2400 modem for John Martin to test on the bulletin board. If the modem works out the club will purchase it. The present modem a Smart Dak 1200 will be in the September RAFFLE.

Bob Sherburne demonstrated John Birdwell's DSKU Util V4.12. Rudy Johnson demoed Clyde Coolidge's High Speed Cassette Loader, which will be on the Sept DOM. For those of you using a recorder bring in a cassette to get a copy.

John Martin agreed to do another demonstration in telecommunications in the near future for those of you interested. In Sept John will show us a program by Roger Merrit, Jiffy Flyer, a screen graphic program that prints out flyers. John will bring in a few copies that sell for $10.

At the Sept meeting George Tilly will give us a demonstration of the much discussed TI Base, if you have been reading the News Letters which are available from the librian, Lance Wilson. George will also be bring in some leaflets for the Avatex 1200 modem selling for $69.

AUGUST RAFFLE WINNERS:

Jim Goodman chose the Tax/Investment pkg.
Cindy Mitchell chose a disk box.

Welcome to two new members. Gloria and Ken Shorter.

The meeting adjourned at 9:00 pm.

At the committe meeting we discussed purchasing software to be be demonstrated then placing the software in the Raffle Pot. We will take a poll on software that the members would be interested. Utility type, data bases, graphics, games, it is all availabe. If there is a program you would be interested to see, SPEAK UP!

Hopefully I will have a semi-complete list so some of the new members can see what is availabe no matter what system setup you may have.

SECRETARY
SNUG USERS GROUP

THE POOR MAN'S A to D CONVERTER (and what it's good for)

by John Martin

Just what is an A to D converter?? Well, normally, it is a device that will convert an analog signal into digital pulses and vice versa. That's about all I know about them because I've never used one or studied how they work.

In that case, where do I come off calling this a poor man's A to D converter? Good question, but I had to call it something and that seemed as good a name as any.

What this device does is to digitize an analog signal. It's not very elaborate. It doesn't have any fancy counters or frequency generators or anything else to help it decide when to send a pulse and when not to. It does have it's advantages, however.

What the poor man's A to D converter consists of is an array of opto-isolators. That's it. Just opto-isolators. An analog signal is fed into the LED side of the isolator. As the input signal makes the transition from positive going to negative going, one LED will stop glowing and another will start glowing. This causes the photo sensitive transistor switch associated with that LED to either open or close. This opening and closing of the switch represents a binary 0 or 1 to the computer.

Ok, so how do I get this signal into the computer and what's it good for anyway? Glad you asked. The signal can be connected through the joystick port on the computer. Since the "switches" are really transistors, the current will only flow one way through them. This means that the signal diodes that are normally required for a joystick will not be required for this device. Another plus is that since the analog signal drives the LEDs directly, there is no external power supply to build. The device is completely passive. With 2 joysticks decoded and 4 separate inputs per joystick, it is possible to have as many as 8 different analog signals monitored by this device.

So far, the main use I have come up with for it is as an input device for "watching" stereo music on my monitor. I feed in left and right signals from a stereo, break them up into left positive, left negative, right positive, and right negative signals. These are fed into joystick left, up, right, and down inputs of the computer. I have a number of programs (all in ASSEMBLY) that take the incoming joystick information and create different patterns on the screen. I haven't tried writing anything for the converter using BASIC or XBASIC, but other than speed, don't see why it wouldn't work as well as the ASSEMBLY stuff. Since I was wanting to create "real time" effects using music, I reasoned that the faster the sampling rate, the better and more representative the patterns would be. I am including the source code for one of the programs (called "LINETUNES") with this article.

Other possible uses for it might be to monitor or control various things around your home or business. In addition to responding to analog signals, it could as easily be used to monitor digital signals and provide the buffering needed between your voltage source and the computer. With 8 possible signals to input, the only thing to limit applications is your own imagination.

There is also a schematic for building your own "POOR MAN'S A TO D CONVERTER". As with anything of this nature, there is always some risk involved. If you plug this device into a high powered stereo, there is a chance that the signal could be too much for the LED'S. They could short and cause some damage to your stereo. I reccommend using a small, battery powered stereo for input. Neither this publication nor I will assume ANY responsibility for damage to your equipment. Proceed at your own risk. I have put many hours on two different stereo's with no ill affects, but will not guarantee that no harm will come to your equipment by building and using this circuit.

==========

NOTES FOR BUILDING "POOR MAN'S A TO D CONVERTER"

by John Martin

I have included a schematic with this article for building your own converter. With a lot of help from Rudy Johnson (actually, he did all the work while I just sat and told him what I wanted), we were able to draw a schematic of the converter with TI Artist. Thanks Rudy, I couldn't have done it without ya.

There were some things that we couldn't draw on the schematic due to the limitations of screen size etc. The main thing that didn't show up was how to connect the joystick extension cable to the converter. I have used a number of these extension cables for various projects during the past few years, and have found them to have inconsistant color codes. What that means is that you will have to use some sort of continuity tester to find which wire goes to which pin.

Cut your joystick extension cable about 10 inches from the male end (the end with the pins). Strip the wires back on the piece connected to the female end and test each wire for continuity to the socket. There is a diagram on the schematic that shows the pinout looking into the female end. Using this as a guide, write down the wire color for each pin.

When you assemble the converter, use both ends of the cable. Make sure that all the wires from the female end are conneted to the same color wires from the male end. This will let you keep your joysticks connected to the computer at the same time the A to D converter is connected. For Geneve owners, this has the added advantage of letting you have your joystick port in an easily accessable location. No more having to reach behind the P-box to connect and

disconnect your joysticks. I have had mine set up this way for several months and have had no problems with iterference between the converter and the joysticks.

The numbers on the output side of the schematic indicate which pins to connect to on the joystick cable. Refer to the color chart you have made and connect the appropriate wires to each point.

Input to the "Poor man's A to D Converter" can be either from speaker outputs or from a headphone jack. The schematic provides for speaker jacks on the converter so that if you use a headphone jack that automatically cuts off the speaker output to the stereo, you can plug the speakers into the converter and be able to hear the sound again. If you use a headphone jack, you will have to connect two of the input wires together (one from right input and one from left input) and run them to the common on the headphone jack.

Construction and parts layout is not critical. I used a plated perf board from Radio Shack (Part #276-149). This board has ample room for the components (all 3 of them) and still leaves room for future expansion.

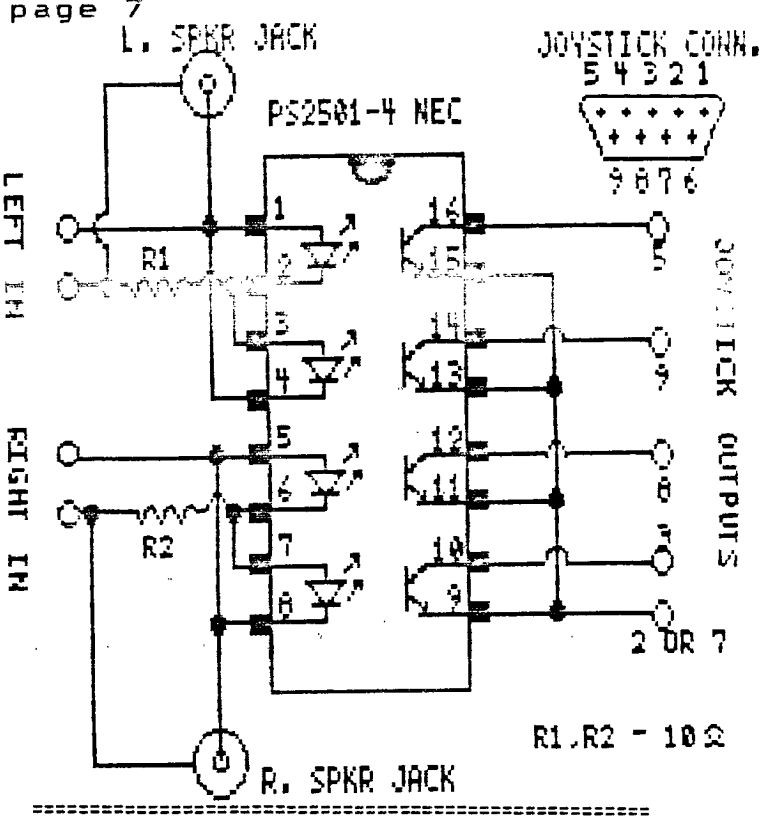
PARTS LIST

QTY	DESCRIPTION	PART #	SOURCE*	PRICE
1	quad opto isolator	PS2501-NEC	D-k	$ 2.38
2	10 ohm, 1/4 watt resistors	271-1301	RS	.39/5
1	Joystick extension cable	270-1705	RS	5.49
1	printed circuit board	276-149	RS	.99
1	project construct box	270-220	RS	1.79
2	speaker jacks	depends on speakers	RS	$1 to $5 ea
	Approximate cost			$12 to $16

* D-k is Digi-key, RS is Radio Shack

To order from Digi-Key, phone 1-800-344-4539. If there is no Radio Shack in your town, maybe you should move 'cause I can't find a phone number to order from them.

Since the type of speaker jacks you use will depend on your stereo, I cannot give an accurate cost for them, but they are inexpensive in any case. If you have trouble getting the quad opto-isolator, you could build it using 4 single opto-isolators, (Radio Shack calles them Optocouplers) but the ones available at Radio Shack are in packages of mixed types. They would probably work, but since the packages contain "one transistor, one darlington, and one other" type, I suspect that the sensitivity would differ between them. This would affect the symmetry of the patterns on the screen. If you can find 4 with the same part number, you should be able to get good results.



LINETUNES DOCS...

System Requirements:

TI 99/4A console or Geneve 9640
32K memory expansion (not required for 9640)
Disk drive
Editor/Assembler Cart

"Poor Man's A to D converter"

This program, as written, will run from Option 3 "Load and Run" from the E/A menu. The program should auto-start, but if the version you have does not, the start name is "RUN".

Be sure that the "Poor man's A to D converter" is connected to the joystick port on your computer and to the speaker or headphone jacks of your stereo. If you want to listen to the music while watching it, you must connect speakers to the jacks on the converter.

The screen will display a graphic representation of the relative left/right variations in the music in real time. The signal is broken up into 4 parts, left positive, left negative, right positive, and right negative signals. These signals are fed into the joystick port as left, right, up, and down. The lines on the screen are directly controlled by these signals. The relative length of the lines is determined by the frequency of the signal. The average length of the lines can be adjusted by pressing either "fire" button. The fire button on 1 joystick causes the lines to get longer, while the button on the other stick causes them to get shorter. Which joystick is which depends on whether you are using a TI or a Geneve.

```
*****************************************
*                                       *
*           LINETUNES                   *
*                                       *
*  by:    John Martin                   *
*         5908 W. Bartlett              *
*         Las Vegas, NV 89108           *
*                                       *
*  Stereo Music digitizing program      *
*         for use with the              *
*    "POOR MAN'S A TO D CONVERTER"      *
*                                       *
*  This program is compatible with      *
*    the TI-99/4A and Geneve 9640       *
*           computers                   *
*****************************************
```

```
*****************************************
*                                       *
*          REGISTER USAGE:              *
*                                       *
*  R0 \                                 *
*  R1  > USED BY VDP UTILITIES          *
*  R2 /                                 *
*  R3   POINTER INTO STACK              *
*  R4   TEMP VAR USED BY CALC           *
*  R5   TEMP VAR USED BY CALC           *
*  R6   DELAY FOR FIRE BUTTON           *
*  R7   POINTS TO CURRENT JOYSTICK      *
*  R8   CURRENT DOT COLUMN              *
*  R9   CURRENT DOT ROW                 *
*  R12  CRU BASE ADDRESS                *
*                                       *
*****************************************
```

```
VDPWA  EQU  >8C02       vdp write address
VDPWD  EQU  >8C00       vdp write data address
VDPRD  EQU  >8800       vdp read data address
*
       DEF RUN
*
MYWS   BSS  >20         set up workspace registers
ONE    DATA >1000       variable to change foreground color
HEX    DATA >F000       variable to check for white
*                         foreground
COL    DATA >2000       beginning color
PREVC  DATA 125         previous dot column location
PREVR  DATA 95          previous dot row location
COUNT  DATA >10         variable for current line length
PIXL   DATA 0           left variable
PIXR   DATA 0           right variable
PIXD   DATA 0           down variable
PIXU   DATA 0           up variable
VDPTMP BSS  2           temp variable for vdp utilities
LINE   DATA 0           variable for line length
PXTBL  DATA >8040,>2010,>0804,>0201   lookup table for
*                         pixel value in byte
*
TRAIL  BSS  >500        buffer for stack
       EVEN
*
RUN    LWPI MYWS        load up workspace
*
       LI   R0,>1B00    Sprite base address
       LI   R1,>D000    DISABLE SPRITES
       BL @VSBW
*
       LI   R0,>0701    SCREEN COLOR
       BL @VWTR         TO BLACK
*
*
*
*****************************************
*  SET UP BIT MAP MODE FOR DISPLAY      *
*****************************************
*
       LI   R0,>0002    SET BITMAP MODE
       BL   @VWTR       BIT
*
       LI   R0,>0206    SET BASE SCREEN
       BL   @VWTR       IMAGE TO >1800
*
       LI   R0,>03FF    SET BASE COLOR
       BL   @VWTR       TABLE TO >2000
*
       LI   R0,>0403    SET BASE PATTERN
       BL   @VWTR       DESC TABLE TO >00
*
       LI   R0,>0536    SET BASE SPRITE
       BL   @VWTR       ATTRIB TO >1B00
*
       LI   R0,>1800    FILL SCREEN
       CLR  R2
FILL   CLR  R1          IMAGE TABLE
FILL1  BL   @VSBW
       INC  R0          WITH 3 SETS
       AI   R1,>0100
       CI   R1,>0000    OF CHARS
       JNE  FILL1
       INC  R2          >00 TO >FF
       CI   R2,3
       JNE  FILL        (768 CHARS)
*
*
       CLR  R0
       CLR  R1          ZERO OUT PATTERN
PCLR   BL   @VSBW       TABLE
       INC  R0
       CI   R0,>1800
       JNE  PCLR
*
       LI   R0,>2000    ZERO OUT
       CLR  R1          COLOR
CCLR   BL   @VSBW
       INC  R0
       CI   R0,>3800
       JNE  CCLR
*
```

```
****************************************
* BIT MAP NOW SET UP.  CODE FOLLOWING  *
* INITIALIZES VARIABLES FOR LOOP       *
****************************************
*
      LI   R3,TRAIL+>49B  last address in stack !clear out
CL    CLR  *R3            clear it              !the stack
      DECT R3             next lower address    !before we
      CI   R3,TRAIL       top of stack yet      !use it
      JNE  CL             no? then do again     !
      LI   R8,125         starting column
      LI   R9,95          starting row
      CLR  R12            set cru address to 0
      LI   R6,>10         R6 used for delay in "FIRE"
*                           routine
****************************************
*  THE FOLLOWING CODE IS THE MAIN LOOP *
****************************************
*
LP    LI   R7,1           R7 used here as loop counter for
*                           2 joysticks to scan
*
      CLR  @PIXL          zero out
      CLR  @PIXR           pixel
      CLR  @PIXD            movement
      CLR  @PIXU             variables
      SBO  18             CRU bit 18 set to one for
*                           joyst #1, not set  for #2
A1    SBO  19             CRU bit 19 set to one ( scan set
      SBO  20             CRU bit 20 set to one     for
*                                                 joysticks
      TB   4              CRU bit 4 is joystick left input
      JEQ  RT             if not set, jump to next test
*
*
      DEC  @PIXL          decrament pixel left variable
*
RT    TB   5              CRU bit 5 is joystck right input
      JEQ  DN             if not set, go to next test
*
      INC  @PIXR          incrament pixel right variable
*
DN    TB   6              CRU bit 6 is joystick down
      JEQ  UP             if not set, go to next test
*
      INC  @PIXD          incrament pixel down variable
*
UP    TB   7              CRU bit 7 is joystick up input
      JEQ  P              if not set, go to print routine
      DEC  @PIXU          decrament pixel up variable
*
P     BL   @FIRE          check fire buttons
*
      SBZ  18             set to scan joyst #2
      DEC  R7             are we done with both joysticks?
      JEQ  A1             no? then go thru loop again
*
*
      MOV  @COUNT,@LINE   load current line length into
*                           variable
```

```
WRTLP LIMI 2              enable interrupts
      LIMI 0              disable interrupts
      CI   R8,240         are we at right side of screen
*                           yet?
      JGT  $+6            yes, then skip next instruction
      A    @PIXR,R8       no, then add value at right
*                            pixel variable to R8
*

      CI   R8,15          at left side of screen yet?
      JLT  $+6            yes, then skip next instruction
      A    @PIXL,R8       no, then add variable at left
*                           pixel variable to R8
*
      CI   R9,15          are we at top of screen yet?
      JLT  $+6            yes, then skip next instruction
      A    @PIXU,R9       no, then add value at up pixel
*                           variable to R9
*
      CI   R9,170         are we at bottom of screen yet?
      JGT  $+6            yes, then skip next instruction
      A    @PIXD,R9       no, then add value at down pixel
*                           variable to R9
*
*
      SETO @>83D6         disable screen time-out
*
      C    R8,@PREVC      check previous column
      JNE  CALC           same place?
      C    R9,@PREVR      check previous row
      JNE  CALC           also same place?  havent moved?
      CI   R8,125         yes? check horiz ctr of screen
      JEQ  RO             there yet?
      JLT  AC             left of ctr?
      DEC  R8             no, must be right... move left
*                           1 pixel location
      JMP  RO
AC    INC  R8             yes, then increment pixel
*                           location
RO    CI   R9,95          vert ctr of scrn
      JEQ  CALC           there yet?
      JLT  AR             above ctr?
      DEC  R9             no? must be below ctr... move it
*                           up 1 pixel
      JMP  CALC
AR    INC  R9             yes, then move down 1 pixel
****************************************
* THIS PART OF THE CODE FIGURES OUT    *
*      WHICH BIT TO TURN ON            *
* R8 IS COLUMN VALUE R9 IS ROW VALUE   *
****************************************
*
CALC  MOV  R8,@PREVC    STORE CURRENT COLUMN FOR NEXT PASS
      MOV  R9,@PREVR    STORE CURRENT ROW FOR NEXT PASS
*
      MOV  R9,R4        COPY ROW VAL INTO R4
      SLA  R4,5         MULTIPLY IT BY 32
      SOC  R9,R4        SET BITS FROM R9
      ANDI R4,>FF07     KEEP LEFT BYTE & LAST
*                       3 BITS OF RIGHT BYTE
      MOV  R8,R5        COPY COL VAL INTO R5
```

```
      ANDI R5,7         KEEP RIGHT BITS
*                       (REMAINDER OF DIV BY 8)
      A    R8,R4        ADD COL VAL OF PIXEL TO BE
*                         SET TO R4
      S    R5,R4        SUBTRACT R5 TO FIND POS IN PATTERN
*                         DESC TABLE
*
      MOV  R4,R0        GET READY FOR VSBR
      MOV  R4,*R3+      STORE OLD ADDRESS ON THE STACK AND
*                         POINT TO NEXT WORD
      BL   @VSBR        READ IN EXISTING BYTE
      CI   R3,TRAIL+>498 END OF STACK YET?
      JLT  PTB          NO? THEN CONTINUE
      LI   R3,TRAIL     YES? THEN RESET STACK VALUE
*                         POINTER TO INITIAL VALUE
*
PTB   LI   R7,PXTBL     LOAD INDEX TO BYTE VALUE
      A    R5,R7        INDEX INTO TABLE
      SOCB *R7,R1       SET CORRESPONDING BIT FROM TABLE
*                         INTO EXISTING BYTE
      BL   @VSBW        WRITE BYTE BACK TO SCREEN
*
      AI   R0,>2000     NOW GO TO COLOR TABLE
      MOVB @COL,R1      COLOR IT
      BL   @VSBW         WRITE COLOR
      AB   @ONE,@COL    NEXT COLOR READY
      CB   @COL,@HEX    WHITE YET?
      JNE  ERASE        NO? THEN SKIP NEXT STEP
      MOVB @ONE,@COL    RESET TO FIRST COLOR
*
ERASE MOV  *R3,R0       FIRST SCREEN LOCATION ON STACK
*                         (FIRST IN FIRST OUT)
      CLR  R1           ERASE THE BYTE
      BL   @VSBW        WRITE IT
      DEC  @LINE        DECREMENT LINE LENGTH VARIABLE
      JNE  WRTLP        IF NOT ZERO YET, GOTO WRITELOOP
*                         AGAIN
      MOV  @COUNT,@LINE RELOAD LINE VARIABLE WITH CURRENT
*                         LENGTH
*
      B    @LP          GO BACK AND DO IT ALL AGAIN
*
********************************************
* SUBROUTINE TO CHECK FIRE BUTTONS   *
********************************************
*
FIRE  DEC  R6           COUNTER TO SLOW REACTION TO
*                         JOYSTICK FIRE BUTTONS
      JGT  OUT          IF NOT ZERO, DON'T DO ANYTHING
*
      LI   R12,>0006    SET CRU BASE ADDRESS TO >0003
*                         (ADDRESS OF FIRE BUTTONS)
      STCR R6,0         STORE CRU BITS 3 TO 18 IN R6
*                          (16 BITS TRANSFER)
*                       BITS ARE INVERTED SO IF BUTTON
*                          PRESSED, BIT WILL BE 0
      ORI  R6,>7FFE     ONLY INTERESTED IN LSB & MSB SET
*                          ALL OTHERS TO 1
      INV  R6           INVERT BITS SO FIRE BUTTONS TRUE=1
      JEQ  RSTR         IF NONE SET, EXIT ROUTINE
      JLT  J2           MSB SET?  THEN JOYSTICK 2 TESTED
*
      INC  @COUNT       NO, THEN JOYST 1 FIRE PRESSED,
*                          ADD 1 TO LINE LENGTH
      JMP  RSTR         EXIT
*
J2    CI   R6,>8000     IS FIRE BUTTON PRESSED?
      JEQ  RSTR         NO? THEN EXIT
      DEC  @COUNT       YES, THEN SUBTRACT 1 FROM LINE
*                          LENGTH
      JGT  RSTR         IF GREATER THAN ZERO, THEN EXIT
      INC  @COUNT       IF ZERO, THEN ADD 1 TO LINE LENGTH
*                          (MUST BE >0)
*
*
RSTR  LI   R6,>11       RELOAD R6 WITH DELAY VALUE (MUST
*                          BE ODD NUMBER)
      CLR  R12          RESET CRU BASE ADDRESS TO ZERO
*                          BEFORE WE LEAVE
OUT   RT                RETURN TO CALLING ROUTINE
*
****************************************
*  THIS PART OF THE CODE REDEFINES    *
* SOME OF THE VIDEO ACCESS UTILITIES  *
* TO STAND ALONE CODE.  THIS SHOULD   *
*  ALLOW THE PROGRAM TO BE EASILY     *
* CONVERTED TO RUN AS PROGRAM IMAGE   *
* OR IN THE MODULE SPACE (SUPER CART) *
****************************************
*
VSBW  MOV  R0,@VDPTMP   TEMP STORAGE OF R0 VALUE
      ORI  R0,>4000     SET FIRST TWO BITS TO INDICATE A
*                          "WRITE" OPERATION
      SWPB R0           GET LSB FIRST
      MOVB R0,@VDPWA    WRITE TO VDP WRITE ADDRESS
      SWPB R0           NOW GET MSB
      MOVB R0,@VDPWA    WRITE IT
      MOV  @VDPTMP,R0   RESTORE ORIGINAL VALUE TO R0
      MOVB R1,@VDPWD    WRITE BYTE TO VDP
      RT
*
VWTR  MOV  R0,@VDPTMP   TEMP STORAGE OF R0
      ORI  R0,>8000     SET FIRST BIT
      SWPB R0           LSB FIRST
      MOVB R0,@VDPWA    WRITE IT
      SWPB R0           NOW MSB
      MOVB R0,@VDPWA    WRITE IT
      MOV  @VDPTMP,R0   RESTORE R0 TO ORIGINAL VALUE
      RT
*
VSBR  MOV  R0,@VDPTMP   TEMP STORAGE OF R0
      SWPB R0           LSB FIRST
      MOVB R0,@VDPWA    WRITE IT
      SWPB R0           NOW MSB
      MOVB R0,@VDPWA    WRITE IT
      MOV  @VDPTMP,R0   RESTORE R0
      MOVB @VDPRD,R1    READ BYTE FROM VDP
      RT
*
      END RUN
```

HUNTER ELECTRONICS

4N370 PINE GROVE
BENSENVILLE,IL.60106

USER EDITION
SEPT.1988

BY APPT.ONLY
312/766-9503

MYARC HARD DISK CONTROLLER

MYARC FINALLY HAS RELEASED ITS LONG AWAITED HARD CONTROLLER FOR THE 99/4A AND THE GENEVE'. YOU CAN NOW RUN MULTIPLE FLOPPY DRIVES(AS LONG AS THE ACCESS TIME IS AT LEAST 6 MS. OR FASTER), AS WELL AS MULTIPLE HARD DRIVES UP TO 100M EACH. OUR PRICE IS ONLY $299, AND DELIVERY **NOW**.

HARD DISK DRIVES

IF YOU NOW NEED AN EXCELLENT, LOW POWER DRIVE, CALL US ABOUT THE **KALOK** CORP. BRAND. THEY HAVE BEEN WRITTEN UP RECENTLY IN THE JULY 25 COMPUTER SYSTEM NEWS WITH THE FOLLING ADVANTAGES:

A. LOWERED PARTS COUNT BECAUSE OF NEW TECHNOLOGY, BETTER RELIABILITY
B. FEWER MOVING PARTS-GIVING BETTER RELIABILITY
C. MEAN TIME BETWEEN FAILURE-43500 HRS.(COMPARED TO SEAGATES 20,000)
D. LOW PRICE-ONLY $249 FOR 20M,5-1/4"(40M DUE OUT IN SEPT.)
E. DESIGNER OF THIS DRIVE ORIGINALLY DESIGNED SEAGATE'S ST225

MYARC PRODUCTS

GENEVE-$519		DS/DD FLOPPY CONTROLLER-$159	
RS232 - 115		512K MEMORY	279

MONITORS

MODEL 8762-COLOR COMPOSITE(80 COLUMN)	$259
MODEL 8CM873-RGB COLOR ANALOG(WORKS GENEVE)	549
MODEL BM7622-AMBER COMPOSITE(WORKS GENEVE)	99

NOTE: CALL FOR QUOTES ON SPECIAL CABLES FOR GENEVE HOOKUPS

USED EQUIPMENT SALE (FULL WARRANTY ON MOST ITEMS)

EXPANSION BOX ITEMS		STAND ALONE EQUIPMENT	
PASCAL CARD	$100	DISK CONTROLLER-$75	
RS232 CARD	60	RS232	60
EXP.SYSTEM	150	32K MEMORY	50
AXIOM PRINTER	40	SPEECH SYNETH.	35

USED SOFTWARE SALE

LOGO II	15	EDITOR ASSEMBLER	25		
T.E.11	10	TUNNEL OF DOOM	10	ZORK 1, 2	10
EXT.BASIC	35	PAINT N PRINT	15	MINI-MEM.	30

ASGARD SOFTWARE SALE (MENTION THIS AD-GET $3 OFF ANY OF THE BELOW)

FONT WRITER II-$22		LEGENDS - $25		CALENDAR MAKER-$19	
DOOM GAMES	9	MISSILE WARS-10		ARTIST INSTANCE	25

LATEST RELEASE II BASE BY TESTAMENTS

A NEW DATABASE THAT PERMITS 5 DATABASES TO BE OPEN AT THE SAME TIME. YOU CAN MAKE A BASE OF AT LEAST 8000 ITEMS-EASILY. $22

NEED HELP OR ASSISTANCE-CALL FOR OUR LOW RATES. 312-766-9503

SHIPPING-ORDERS OF SOFTWARE UP TO $100, ADD $2, AND $3 ABOVE THAT. ILL.

SOUTHERN NEVADA USERS' GROUP

The SNUGLETter is published monthly by the Southern Nevada Users' Group (SNUG). SNUG is a non-profit organization of individuals with an interest in all aspects of Texas Instruments' 99xx & 99xxx based computers including hardware and software by third party vendors. The GROUP meets 6:30 PM on the second Monday of the month - currently in the Nevada Power Company, Wengert Community Meeting Room., 6226 West Sahara Avenue. Visitors and guests are welcome to attend the meetings. Information on membership is available at the meeting. Articles may be copied from the SNUGLETter provided credit is given to both the author and the original source and that the article not be used for profit. (For-profit organizations wishing to use any articles from the SNUGLETter will need to make prior arrangements with the Executive of the Southern Nevada Users' Group.)

Southern Nevada Users' Group (SNUG)
P.O. Box 26301
Las Vegas, NV 89126-0301

TO:

SNUG TREASURER'S REPORT - 31 AUGUST 1988
(in lieu of 31 August 1988 bank statement)
Karen Rodgers--Treasurer

FIXED ANNUAL EXPENDITURES:

SNUGLETter (Estimated cost per 100)	
Publication Costs ($43/mo x 12).	$ 516.00
Postage ($25/mo x 12).	300.00
P. O. Box Rental Fee ($22 annually). . . .	22.00
Bank Account Service Charge ($8/mo x 12) . .	96.00
SNUG B/Board Phone Line ($11/mo x 12). . .	132.00
Long Distance Phone Calls (Estimated). . .	44.00
Miscellaneous Expenditures (Estimated) . . .	90.00
TOTAL ANNUAL OPERATING COSTS (Estimated)	$1,200.00
Annual Dues Collection: (Avg 30 mbrs x $18)	540.00
ANNUAL DEFICIT (Estimated)	
To be recovered through fund raisers, disc copying fees, special sales, etc.	$ 660.00

COLLECTIONS (during August 1988)

FUNDS BALANCE (as of 31 July 1988) (includes $8.38 July 1988 service charge)	$ 806.69
Membership Dues (Regular $18 x 1).	18.00
Membership Dues (Senior $12 x 1).	12.00
Bulk Diskette/Disc Box Sales	35.50
Disk of the Month Sales.	6.00
Miscellaneous TI Equipment Raffle.	19.00
Miscellaneous TI Equipment Sales	16.00
(Sub-Total)	106.50

EXPENDITURES (during August 1988)

Publication Cost (SNUGLETter)-R. Johnson	
August Issue	18.65
Postage (SNUGLETter Mailings)-R. Johnson	
August 1988.	29.30
Disks/Disk Boxes Purchased for Resale.	106.14
(Sub-Total)	154.09
FUNDS AVAILABLE (as of 31 August 1988)	$ 750.72
Checking Account (General Operation Funds)	$ 619.10
Funnelweb Donations Trust Fund	86.62
J.J. Menu Donations Trust Fund	55.00
(Bank Balance)	$ 750.72

PROJECTED EXPENDITURES:

Bank Account Service Charge (Aug. Estimate)	$ 8.00
Reimbursement for new B/B modem	175.00
Reimbursement for copy supplies	25.00
	$ 208.00