Vol. 5 No. 10 DECEMBER 1987

(092)8712 New Horizons

This newsletter is published jointly by OH-MI-TI and New Horizons TI-99/4A Home Computer Users' Groups. Material may be reproduced without permission provided the author and source are acknowledged. For more information concerning TI Users' Groups in the Northwest Ohio area, contact:

Robert Peters
President, OH-MI-TI
225 S. Wheeling
Oregon, OH 43616
(419) 693-7934

Meeting; 11 Dec '87 Fri
Oregon #2 Fire Station
Time 7:00 Pm.

(419) 385-7484
TICOMM BBS
>>> 24-HRS <<<

SYSOPS
> TURNER - MILLS <

Don Turner
President, New Horizons
1690 Idlewood Street
Toledo, OH 43615
(419) 537-1454

Meeting; 12 Dec '87 Sat
Unity Church Secor Road
New time: 12:30 Pm.

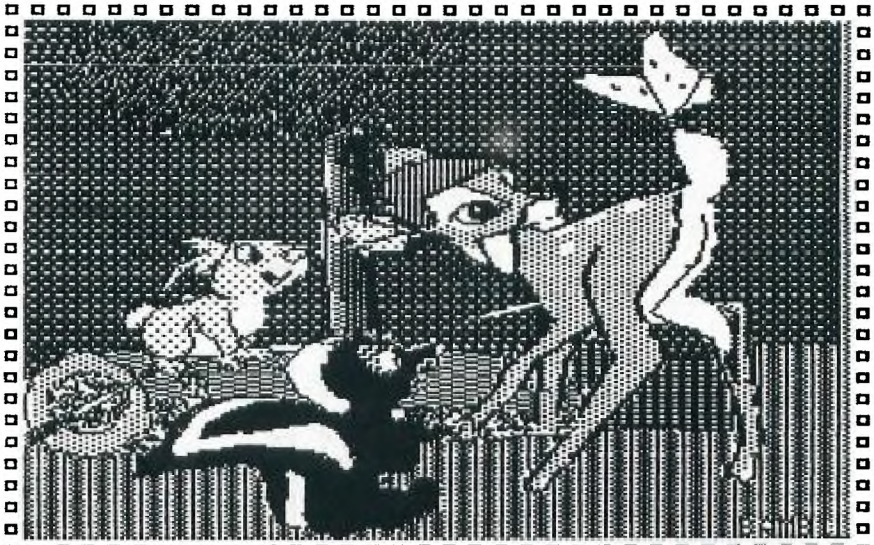
THE NEWSLETTER STAFF

Roger & Judy Feinauer

LOCAL CONTRIBUTIONS BY;

Kent Sheets Bill Tiep Rod Cook Hank Alvaro Don Turner

Merry Christmas 99ers



Bill Sager
612 Meadow Springs,
Maumee, Ohio. 43537

THE PRESIDENTS PAGE

PRESDENT'S CORNER
OH-MI-TI
By Bob Peters

First the news from last months meeting. We voted to remain a seperate club by a near unamious vote. We also had elections of officers. The vote on this was unamious. The officers are, Bob Peters, President, Ken Sheets, Vice-President, Dave Burket, Treasurer, Pat Huntsinger, Secretary and Bud Mills Board of Directors. Dues are $15.00 and should be sent to Dave Burket as soon as possible. His address is 1915 Bordo Rd. Northwood Ohio, 43619.

Our personal and club sympathy goes to the faimly of Phil Dennis. He was our former [illegible] editor and a friend to all of us and to the club, may he rest in Gods Peace and Love.

This months demo will be the MYARC 9640 Geneve, by our newsletter Editor, Roger C. Feinauer. This is an excellent opportunity to see the new machine in action as he now has completed DOS for it.

The door prize this month is the 994A Count. In order to get in the drawing you must bring a non-member or a perspective member with you. Bring cookies or other goodies with you for a Christmas Party. The club will supply coffee and other refreshments.

The meeting will be held at Oregon Fire Station #2 at 7:00 P.M. on Friday, Dec. 11. I would like to wish you and your family a Merry Christmas and a Happy New Year.

NEW HORIZON NEWS
By Don Turner

[illegible] holidays to everyone ! Have you been naughty or nice this year ? I thought so..... Last month we had 24 people attend the meeting. There were seven prizes won at this meeting. This month one of the many prizes will be TI-WRITER. Feel lucky this month ???

This month we will be meeting December 12th at **12:30** in Unity Church on 2525 Executive Pkwy. Please be sure to attend. We are planning a Christmas party for this month. there will be refreshments and a goodie table loaded with some of the finest treats a person can imagine. If you are planning to attend and want to donate [illegible] goodies to the goodie table please feel free to do so. The club will be providing the soft drinks and a party tray from one of the local deli's.

This months nominations for club officers turned into a quick vote. Russ Lee will be secretary for another term and Earl Hoffsis will be treasurer again as well. Jo Symington will be your new vice president and yours truly will be president. John and Chris Dew will maintain the club library for 1988 as well. I would like to congradulate the officers for 1988. Lida Lee is going to maintain the sign in table and Joni will be taking care of all sales again this year. I would like to thank everyone for their support last year.

please turn page -->

grateful for their help during the past year. I would like to include Burr Mallory and Bill Sager for keeping the exchange newsletters in order. Also I would like to mention Bill Tiep for his work on the newsletter. Finally I want to thank Roger Feinauer and his wife for the tremendous job they have done on the newsletter for us.

This month we will be featuring SUPER-CART. Paul Martin will be demo-ing his SUPER-CART and some SUPER-CART software. Roger Feinauer will be doing a demo on FONT WRITER II.

Club dues are due this month. Like last year they will be $15.00 for the year. Earl Hoffsis will be taking care of dues for 1988, be sure to see him before you leave the meeting. You can mail your dues to Earl if you are unable to make this months meeting.

In closing I would like to remind you that the new meeting time wil be at 12:30 pm. Also I would like to congradulate Earl Hoffsis on his retirement. Earl will retire in early January. Now he will have plenty of time to take care of the club !

EDITOR

This is the season to be jolly or it should be. It seems to be overrun with late articles this month. More and more people are turning in their articles later and later. So, to keep everyone punctual I've dedided to make a deadline for news articles. From now on, news articles (IN WRITTEN FORM ONLY) will be due on the second Saturday before the meeting.

Now, to all of you, I wish a Very Merry Christmas and a Happy and Safe New Year. I now have MDOS 1.0 and on the old HRD it will load system/sys and will find an autoexec,bat where it was booted from. The only bad thing I can see is it won't let me run the graphics formatter from font writer II. So, this means loading MDOS 99b. Other than this, everything works just fine.

Lastly, Kent Sheets asked me to let everyone know that he has a complete TI-99/4A system for sale and can be reached at home (419 836-7877).

"A spelling test? Surely they have software for that sort of thing!"

"Mind if I stroll leisurely alongside while you jog, dad?"

The New Adventure in by Roger Feinauer

There has been a lot that has happened for the 9640 sense the last time I wrote this article. First I got a copy of my art. This is the paint program I talked briefly about last month, also I have managed to get the Myarc 9640 operating system to boot from the ram disk with Ros 7.1 . And a friend in Detroit at least I think he is still a friend named Roger Brittin tells me *Dos 1.0* is on GEnie but he tells me it sill isn't finished. why!?

Lets start with fun the part of this article then go to the bummers. First of all Myart is a very good paint program with some interesting possibilities. When you first boot Myart from Dos I find that if you have a printer that prints slower then 200 cps. you need to create a batch file that tells dos to set up a spooler 128k. If you don't do this your system goes into La La land in an endless loop trying to talk with your printer. When you try to print full size pictures. I called Myarc about this and they said that at this time the 9640 dos vers. .97 thru getting ahead of the printer. They said that dos 1.0 has this problem solved. As of this time I don't have this version of Dos so I can't say one way or an other.

On the paint program you can select from a palette at the bottom of the screen of 256 colors which are about 1 bit col. wide by 16 bit rows high . You start with a black background and an icon in the shape of a pencil which is moved with the mouse Using the right button on the mouse you lay down the colors. There is an other mode called 512 mode that gives you a palette of 16 base colors . These colors act more like paint pots because you mix the color you need with a possible of mixing 512 colors but you can only put 256 of these colors on the screen at on time. Oh darn! There is also help screens built into the program by hitting either F9 or H key while doing your picture you are given a help screen with all the function then hit the Esc key gets back were you left off.

You have most of your basic paint program functions such as line - draw, circle, solid box, box unfilled,move and or copy portions of picture to the same picture. Here are some of the things I would like to see . More control over the program with the mouse. As of now most commands must be inputed from the key board . I think Myarc should take a look at other mouse paint programs such as Z-SOFTS PC PAINT for the pc machines. To see how to get the most out of their mouse. Also you only have one font and no way to load others in. Also there is no way to save or load block pictures. A block picture is a section of other picture as "instances" in Ti-Artist. I would also like Rle loader and saver for a larger selection of picture files this it's a good start but needs more "PS." It's still a good buy for a mouse and you get something to use it with.

To make a AUTOEXEC,BAT file for the Geneve 9640 would go as follows

```
A> COPY CON AUTOEXEC,BAT

REM This is a batch load for

REM MYART paint program

SPOOL 128

A:

A:MYART

 Z

 C
```

As you notice this is done out of Dos. Unlike the manual says, you can make batch files without Ti-Writer. The secret is the COPY command in dos. You see dos treats the key board as a device same as a printer which is PRN or disk drives as a: thru z: . Yes A: thru Z: for those who think I made a mistake read their manual. Of course you need some kind of fancy controller for this but back to the batch file. The first line of the batch you are telling the computer to copy from the key board to the default drive which in this case is a: a file called AUTOEXEC,BAT . After you do the "C prompt isn't seen any more . In the next 2 lines are REMARK statements for reference of what is going on in the batch file. The next command SPOOL 128 sets up a one hundred twenty eight K printer spooler for the the paint program. The lone A: is used to make sure the drive that is set is drive and the last line A:MYART tells drive to load MYART and all that's left is to tell dos to save the file. This is done with a end of file marker This is done with a control Z, this is what is meant with Z which means the same thing. Lastly, you need to get control C which cancels the copy command.

Copy con is also useful for other things let's say you want to get a rough draft of *Ti-Writer* file . And your in dos. Well you dont need to pull up Myword to print it. Just use the copy command as follows;

```
A>COPY A:filename PRN
```

Bingo your file is sent to the printer. This will even work with the ram disk "HRD"

Which brings me to a little bummer You see with the 9640 you have a new computer, Really you have 2 computers And each runs a little differently for each other. What am I leading to, well the Horizon ramdisk. It seems on the 99/4A you can have up to 5 ramdisks in the P-Box, and call them drives 1-9 Most of use label them above the number of our last drive . In the 9640 as of now you can have 2 and the first one set at CRU address of 1400 and must be called DSK6 and the other at 1600 and called DSK7. It is to my understanding as of now. The Geneve can only read or write to floppy disk to about 820k.

So this means you have at this time the capability to run two HRD at 720k each. At lease you should but the code at this time can only read out to about 1100 sectors on a 1mg. card with 360k of chips on it, which means it can read 275k. It seems the new card uses 16 bit words to to get out to 1mg.. Also the new card at this time wont se

please turn page -->

system/sys to boot the Geneve operating system because of the sixteen bit decoding sets the counter back to 0. So with the new card you have a 1100 sector ram disk to use until the decoding problem is solved. I would like to thank Bud Mills and Roger Britten which the latter is from Taylor Michigan for the information.

Now if you have one of the old ramdisks your in like flint. These use an 8 bit decoding of the sector numbers and works fine. In both cases you will need to use the 99/4A to load ROS 7.1 onto your ram disk. Just read the doc's that came with Ros 7.1 . You don't need to load the Menu program because MDOS is better and it probobly wont work anyway. Next resetup the Geneve system. Load your MDOS At the a promt type ASSIGN E= DSK6: Then press enter. If everything was done right. type on the a promt ASSIGN and you should get a listing of your drives. see below.

```
A:ASSIGN E=DSK6:

A:E:

E:

E:ASSIGN

E:A:DSK1

E:B:DSK2

E:C:DSK3

E:D;DSK4

E;E;DSK6

E;F;WDS2

E:G:WDS3
```

E: NEXT LOAD SYSTEM/SYS, E/A, MGR1, AND 2 to your ramdisk and your system should toot off the ram disk. As of now I haven't been able to get an autobat file to run off startup off the ramdisk but if I put a disk in drive one with a autoexec,bat file it will read it on start up. So this [illegible] I am with my system as of this time. So with this I will leave till next month.**DON'T FORGET**

CLUB DUES ARE DUE AT BOTH

CLUBS $15.00 IS NOT BAD FOR

WHAT YOU GET SO COME WITH YOUR MONEY AT THE [illegible]SETTING AND MERRY CHRISTMAS EVERYONE. roger

Hank Alvaro

THE FLIP SIDE

More and more people are discovering the joy of disk drives and the depression that comes when this great little device suddenly quits doing what you want. This seems like a good time to go over some of the old don'ts of the floppy and some things you probably havn't thought about. Mainly using the flip side.

No-No's: 1. Never touch the brown magnetic surface of the disk. 2. Never write on disks with a ball-point pen. 3. Keep disks away from heat. like the sun or your heater. 4. Never allow smoking near your disks or drives. It shortens their lives too. The tar gums the surface of the disk and cloggs the drive head.

5. Never store disks by your monitor. The 13-20,000 volts of the screen and transformers inside can do nasty things.

6. Never leave your disks near any kind of motor. They have magnetic fields or arcing that will blow your little bits away.

7. Never use your disks for a frizzbe.

9. Never stack your disks flat. The weight of the other disks will flatten the edges of the bottom disks, making them hard or impossible to run.

9. Reconsider using both sides of a disk by notching and flipping.

"But I've been doing that for centurys and never had any problems." - Lucky!

What I'm talking about is what the disk manufacturers call a dangerous act. The disk was only designed to run in one direction. It will run great in both directions but what's happening inside the disk? There's this neat liner inside the outer plastic covering that helps keep the inner disk clean. As a disk wears, the oxides come off and some are caught by your read/write head. Some build up on the pressure pads but most are trapped in the liner. It works like the brush you use to clean your records. As long as you move the brush in one direction on the record it picks up dirt and lint. But what happens when you notch and flip your disk? It is now running backwards. What happens to the dirt on your record brush when you go backwards? All the garbage you had just picked up will be deposited on the record.

This is what happens to your disk also. Reversing the disk lets the oxides break loose from the liner in clumps and build up quickly on the drive head and sometimes even scratch the disk. Don't forget about those pressure pads. Flipping the disk over on single sided drives exposes the 'good' side of your disk the oxides that have built up on the pad.

You might get away with flipping for awhile but you had better stock up on cleaning solution and make sure you have plenty of backups.

Also some single sided disks have thinner oxide coatings than double sided any by flipping you run the risk of having "bleed through". That's when the information on one side affects the data on the other side.

Now armed with this information maybe it will be easier to make your decision about handling your disks.

What was that record listeners used to say-"See you on the flip side."

LAWS OF COMPUTER PROGRAMMING:

1. Any given program, when running, is obsolet
2. Any given program costs more and takes long
3. If a program is useful, it will have to be chang
4. If a program is useless, it will have to be documented.
5. Any given program will expand to fill all available memory.
6. The value of a program is proportional to the weight of its output.
7. Program complexity grows until it exceeds the capability of the programmer who must maintain it.

TROUTMAN'S PROGRAMMING POSTULATES:

1. If a test installation functions perfectly, all subsequent systems will malfunction.
2. Not until a program has been in production for at least six months will the most harmful error be discovered.
3. Job control cards that positively cannot be arranged in improper order will be.
4. Interchangeable tapes won't.
5. If the input editor has been designed to reject all bad input, an ingenious idiot will discover

BUILDING TABLES

By: Don Turner

New Horizons U/G

One of the most useful ways to handle a large number of records is to put them into a table or array. Most sorting programs utilize tables, games use them for random picking and the list goes on. The TI can handle around 1600 elements in a table depending on the size of your program. The larger your program the smaller the table and the opposite holds true as well. The TI can handle up three dimensional tables as well. However these large tables will eat up a lot of memory.

To set your table up, the DIMension statement must be used for a table having more than 9 elements in it. Your statement would look like this : 100 DIM A(100). This will give you 100 elements in the table declared as A. The big advantage of this is you do not have to create 100 variable names. The varaible A now has 100 locations to store data in.

Let me better explain. Lets say you have 50 variables, you could name them A1,A2,A3 etc. Your program could not use a loop to load each variable. Therefore you would have to write at least 50 input statements to load the data when 3 lines would do the job just as well.

```
10 INPUT A1
20 INPUT A2
30 INPUT A3.......
500 INPUT A50
```

Using a FOR NEXT LOOP would work just as good and save time and most of all memory.

```
10 FOR I=1 TO 50    For the beginners
20 INPUT A(I)       the value of I
30 NEXT I           increments by 1 when
```

line 30 is executed then control is passed back to line 10 until I is greater than 50.

To retrieve the data in a specific

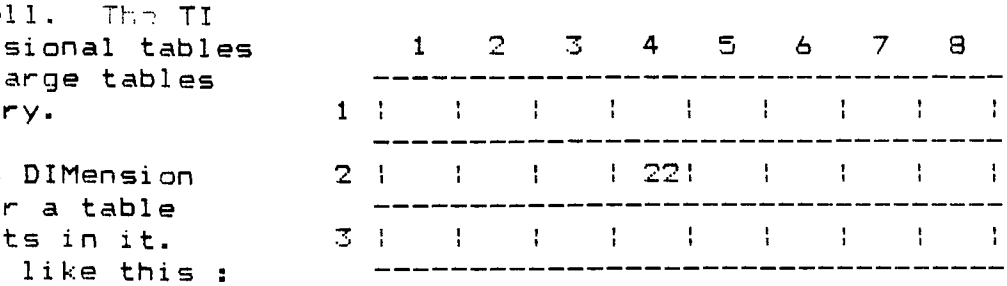
location in your table you would use the numeric value of that location to get that information. Lets say you wanted to get the 34th value of A(34). You would merely tell the computer :

```
10 PRINT A(34)
```

To create a two dimensional table the DIM statement would look like this:

```
10 DIM A(8,3)
```

This would give you a table 25X25 or 24 elements. Your table would resembl. a piece of graph paper having rows and columns. Note the figure below.

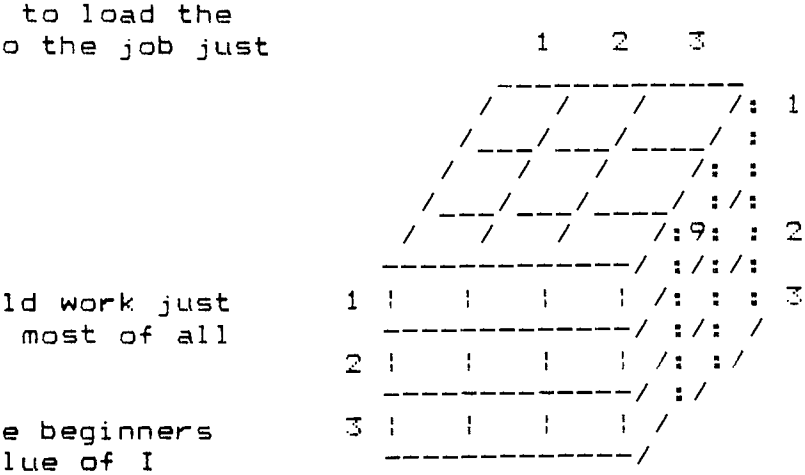


Look at the location 4,2 you will note the number 22, if you wanted to input or print that element you would use location A(4,2)

The most difficult table to use is a 3 dimensional table wich has rows column and depth or sides. As you guessed it the DIM statement would look like this

```
10 DIM A(3,3,3)
```

giving you 27 location



If you look at the figure above I put the value 9 in the location A(1,3,2) ROW 1, COLUMN 2, DEPTH 2. The value of a table dimensioned this way could

enable you to locate a reservation in (ROW,SECTION,THEATRE) or finding a persons name in a (STATE,CITY,STREET). Thes tables are difficult to master until you have used them for several applications. I still have to sit bac and draw them up on paper to figure ou what I need to do when I use them

INTERFACING TO THE REAL WORLD PART 2: THE I/O CARDS, TESTING THE CIRCUIT WITH SOFTWARE.

by Rod Cook
CH-MI-TI

This article will discuss the construction and checkout of the I/O cards plugged into the mother board discussed in Part 1 of this series. Part 1 is a prerequisite to this article and should be reviewed with regards to the addressing scheme used and the meaning of the various signals directed to the I/O cards.

To briefly recap, Part 1 explained the construction of the bus extender card for the P-box and the interface mother board. The bus extender card plugs inside the P-box and extends out the rear where a ribbon cable attaches to the card. The other end of the ribbon cable is connected to the mother board. The extender card serves the purpose of driving the ribbon cable and controlling when signals are sent down the cable to the external electronics, otherwise the cable is effectively turned off. There is also some decoding of the addresses done to generate a SELECT signal that is passed to the I/O cards. The mother board distributes the signals from the ribbon cable to the card-edge sockets that the I/O cards plug into. An external power supply also connects to the mother board to provide power to the I/O cards.

The main IC on each I/O card is a 40-pin chip called an 8255 Programmable Peripheral Interface (PPI). This chip has 24 I/O pins arranged into three 8-bit I/O ports called ports A, B and C. For the purposes of this discussion, these ports (24 pins) will be referred to as the external side of the 8255 implying these are the pins connected to the "outside world". The internal side of the 8255 are the remaining 16 pins and should be thought of as the computer side, the pins that connect to the computer. The computer side of the I/O cards turns out to be identical regardless of whether the cards are built for input (data into computer) or output (data from computer).

The computer side of the 8255 is comprised of 8 data lines that connect to the computer data bus. This can be thought of as the I/O port that the computer uses. The function of the PPI is to control how data is transferred between the internal data port and the three external ports. Of the remaining 8 pins, two are for power and ground and a third is called RESET and is tied to ground for this applicaton. The remaining 5 pins do the real controlling of the 8255. The chip's WR, RD, A0 and A1 pins are connected directly to the computer's WE, DBIN, A14 and A13 lines respectively. The remaining pin of the 8255 is the CS pin and is called the "chip select". All of the address lines and the SELECT line are decoded on the I/O card to generate the CS signal to activate the 8255 for a read or write operation. Depending on how the switches are set on the dip switch will determine at which address the CS signal is generated.

Notice that some of the line names have bars over them. This means that the signal is defined by negative logic. Positive logic is implied by the absence of a bar over the name. The physical signals we are talking about are either 0 volts or 5 volts and these signal levels are represented by logic values of 0 or 1 respectively. It is more common to refer to the voltages as low and high rather than to speak of specific voltage levels. A logic value of 1 corresponds to a high voltage and accordingly, logic 0 implies a low voltage. Under positive logic, a true state is represented by a logical 1 which in electrical terms is a high voltage. A low voltage corresponds to a logic 0 which represents a false state. For example, the SELECT signal from the bus extender card is defined by positive logic, (no bar over the name) and the interface is selected when SELECT is high (true). Negative logic is just the opposite. A true state exists when there is a low signal level, logic 0. For example, the CS pin of the 8255 has a bar over it. This means that the chip is selected (true) when the signal is low. The chip will not be selected when the signal on the CS pin is high.

Below is the truth table for the five control pins of the 8255. In the first line CS is a 1 which means the chip is not selected and the x's for the other signals means it doesn't make any difference what value they have the result is the chip is disabled. But when CS is 0, (true) then there are eight combinations of the other four signals corresponding to four write possibilities and four read possibilities. Actually there are only seven meaningful combinations. The last combination should not be performed because you can not read the program register, you can only write to the register.

CS	WR	RD	A1	A0	Function
1	x	x	x	x	chip disabled
.	.	.	.	.	
0	0	1	0	0	write to port A
0	0	1	0	1	write to port B
0	0	1	1	0	write to port C
0	0	1	1	1	write to Program Register
0	1	0	0	0	read from port A
0	1	0	0	1	read from port B
0	1	0	1	0	read from port C
0	1	0	1	1	invalid, can not read prog reg.

There are three control signals on the CPU bus that are used for control of memory operations between the CPU and other IC's either inside the computer or outside as in our case. They are WE (write enable), DBIN (read enable) and MEMEN (memory enable). WE and MEMEN are defined by negative logic) bars over their names. When the CPU is doing a memory operation, MEMEN will be low (true) otherwise it will be high. Same is true for WE, low for a write and high otherwise. But DBIN is the opposite (no bar-positive logic) the signal is high when doing a read operation, otherwise it is low. If you are doing a CALL LOAD which is a write operation, both MEMEN and WE go low and at the moment these signals are low, the address lines will contain the address to write to and the data lines will contain the value to put into memory. When you do a CALL PEEK, which is a read operatin, MEMEN will go low, DBIN will go high. Again the address lines contain the memory address and the data lines will contain the value at the memory address. The WE and DBIN signals are connected directly to their respective pins on the 8255 chip. Note, that DBIN is inverted on the Bus Extender card to make it conform to the logic definition for the RD pin on the 8255. MEMEN is used on the Bus Extender card to produce the SELECT signal. So, MEMEN from the CPU along with a signal generated by decoding the address lines are used to produce the CS signal.

An I/O card is selected when the CS signal is low. Refer to the schematic of the I/O card and notice that prior to the CS pin on the 8255 is a NAND gate. Two signals are input to the NAND gate, SELECT from the bus extender card and a signal that is common to seven XOR (exclusive or) gates. Let's call this the Board Select signal, BS. The truth table on the left for a NAND gate shows that its output is low (Chip is selected) when SELECT and the BS signal are both high. SELECT is high only when the address is between >9000 and >97FF (see Part 1 for the discussion on this). The BS input to the NAND gate is high only when the levels of the seven addresses, A5 thru A11, are opposite the level of the corresponding input from the dip switch on each of the respective XOR gates.

NAND		
S	B	CS
0	0	1
0	1	1
1	0	1
1	1	0

S=SELECT
B=BS

The desired output of each of the XOR gates is 1 and this will occur when the input signals to the gates are different, see truth table at the left. The outputs of the XOR gates are open collector which allows all of them to be tied together. Open collector outputs require a pull-up resistor connected to 5 volts as is R1. When none of the output transistors of the seven gates are conducting, then no current will be flowing through R1 and BS will be high. But if any of the address inputs and the input from the dip switch match, then that output transistor will conduct allowing a current to flow through R1 which will pull BS low, and the 8255 will not be selected.

XOR		
A	D	O
0	0	0
0	1	1
1	0	1
1	1	0

A=ADDRESS
D=DIP-SWCH

The I/O card schematic shows the dip switch with all the switches open. The resistors, R2-R8, will pull the signal level of each of the XOR input lines to a high level. All of the address lines will have to be low to select this I/O card. Addresses A5 through A11 are low for address >9000 through >900F. In the article for Part 1, the addressing scheme for the interface was discussed. The middle two digits of the hex address correspond to the card address (or card number if you like.) So with all dip switches open, the I/O card is designated as card 0.

Close the switch corresponding to A11. Closing the switch causes the level on this input line to go low, (ground). Now A11 must be at a high and all the other addresses must still be low to satisfy the board select criteria. This will occur between addresses >9010 and >901F and this card can be designated as card 1. Close the second switch from the right in the schematic. Now both A11 and A10 must be high to select this card and this corresponds to addresses >9030 through >903F. This will be card 3. The four dip switches on the right in the schematic set the third digit of the hex address that will select the card and the left three switches will determine the second hex digit. Remember from the discussion in Part 1 that the second hex digit can only be 0 to 7. This is summarized in the following table:

ADDRESS RANGE	A 5	A 6	A 7	A 8	A 9	A 10	A 11	DIP SWITCH SETTING o = open, + = closed 7	6	5	4	3	2	1
>9000 >900F	0	0	0	0	0	0	0	o	o	o	o	o	o	o
>9010 >901F	0	0	0	0	0	0	1	o	o	o	o	o	o	+
>9020 >902F	0	0	0	0	0	1	0	o	o	o	o	o	+	o
>9030 >903F	0	0	0	0	0	1	1	o	o	o	o	o	+	+
>9040 >904F	0	0	0	0	1	0	0	o	o	o	o	+	o	o
. .	.	.	.	.	.	.	.	.	.	.	.	.	.	.
>90F0 >90FF	0	0	0	1	1	1	1	o	o	o	+	+	+	+
. .	.	.	.	.	.	.	.	.	.	.	.	.	.	.
>97D0 >97DF	1	1	1	1	1	0	1	+	+	+	+	+	o	+
>97E0 >97EF	1	1	1	1	1	1	0	+	+	+	+	+	+	o
>97F0 >97FF	1	1	1	1	1	1	1	+	+	+	+	+	+	+

The dip switch is not absolutely necessary for setting up the card address. If the card address is going to be changed frequently then perhaps the dip switch is needed. On the otherhand, if the card address will be set once and not be changed, then don't waste the money on the dip switch and resistors R2-R8. Wire the inputs to the XOR gates directly to +5v or ground, whichever is required. A 14 pin IC socket can be used, with a jumper wire installed from one side of the socket to the other to ground the input line. I used what is called a "double-row jumper header" cut to 7 pins long. To encode the card address I wire wrapped and shorting jumper across the header.

There is one signal generated on each I/O card that is supplied to the P-box. It is called RDBENA and it goes low (true) when a chip is selected. This signal is required by the P-box to enable the data bus to the console. Every peripheral whether in the box or outside must provide this signal when data is to be transferred on the data bus. The output of the gate supplying this signal must also be open collector, but a pull up resistor does not need to be supplied. If this interface is to be adapted to a computer other than a 99/4A, then this signal is not used.

Regardless whether the I/O card has 24 digital inputs (DI), 24 digital outputs (DO) or a combination of both, the computer side of the cards are all the same. Prior to talking about the external side of the I/O cards, let's cover the construction of the card up to this point. There is a diagram of the card with the layout I used and the material list for the I/O card is for those components that are common to both DI and DO cards. I wire wrapped the cards. There is alittle expense associated with the wire wrap sockets and terminal pins but I believe it's worth it in the ease of construction. The circuit card used is available at Radio Shack for $5 but it is on sale frequently for $4. It has a ground plane on the component side and is a nice clean board to use. Whether the ground plane is of any value I don't know, but it doesn't hurt anything. I had to trim about 1/4" from the edge of the board opposite the card edge connectors so that the right angle friction lock headers would fit the terminal housings. I spent alot of time trying to come up with some kind of inexpensive connectors where the external wires connect to the I/O card. There is nothing cheap (unless you own the company that makes them), so you might as well accept the fact that this stuff costs alittle more than you'd wish. But it is important to keep in mind that you want a connector system that is convenient. Don't use a 24 position screw terminal. Can you imagine having to loosen and remove 24 wires to replace a board that has gone bad, much less keep track of which wires go where? Do yourself a big favor, use terminal housings and headers. I've specified two of the 12-position type. The only better way to do this would be to use three 8-position terminals and headers to keep things on a byte format. The best source I've found of the major hobbyist suppliers for these terminals and headers is Digi-key. They have the best and most complete selection but sit down before you figure the cost.

Speaking of cost, I estimate the cost of a I/O card constructed with all the components given in the I/O card material list and the material list for the 24 digital input points will cost around $20 to $30. The I/O card wired for the 24 digital output points using the same components for the computer side of the card and the material list for 24 ouput points will cost between $25 and $35.

The 8255 interface chip is a Programmable. The function of each of the three I/O ports can be programmed through software as either input or output by writing a byte code to the program register. There are eight possible configurations of the three ports and they are summarized in the following table along with the byte code to be written to the program register. There are eight more port setups

PROGRAM REG. VALUE	PORT SETUP A	B	C
>9B	IN	IN	IN
>92	IN	IN	OUT
>99	IN	OUT	IN
>90	IN	OUT	OUT
>8B	OUT	IN	IN
>82	OUT	IN	OUT
>89	OUT	OUT	IN
>80	OUT	OUT	OUT

because port C is actually two 4-bit ports and each can be programmed separately. For this project, port C is treated as a full 8-bit port. The interested reader is referred to the 8255 data sheet for more details. To further wet someones whistle, there are three modes in which the 8255 can operate. This project uses mode 0 which is called basic I/O. Mode 1 is called strobed I/O and Mode 2 is called strobed bi-directional bus I/O. Perhaps those mean something to someone! Again the reader is referred to the data sheets. An interesting application in the data sheet is interfacing to a floppy disk controller chip in mode 2. Now thats an intriguing thought!

The 4th digit of the hex address specifies which of the three ports is accessed or whether the program resister is accessed. The following table summarizes this.

VALUE OF 4TH HEX DIGIT	RESPONSE
0 (1,8,9)	Port A is accessed
2 (3,A,B)	Port B is accessed
4 (5,C,D)	Port C is accessed
6 (7,E,F)	Program Register accessed

The primary values for the 4th hex digit of the address are 0,2,4 and 6. The values in the parenthesis can also be used because the bit pattern of the middle two bits of each four bit digit are all the same. Note nothing is implied about read or write access. This depends on how the three ports are configured. That is, which of the eight program values has been written to the program register. To say it one more

time, the program register can only be written to, not read from. I have read this address but only got a value of zero. Didn't seem to hurt anything though. For the sake of discussion, the input card will be wired for 24 DI points and the output card will be wired for 24 DO points. The first software operation to be done upon powering up will be to write the appropriate value to the program registers. If card 0 is the input card then >9B will be written to address >9006. If card 1 is the output card then >80 will be written to address >9016. The following BASIC statements will accomplish this. Being able to write the software in BASIC

```
CALL LOAD(-28666,155)    (>9006,9B)h
CALL LOAD(-28650,128)    (>9016,80)h
```

to operate the interface seems to be one of the nice things about using direct memory access techniques rather than CRU addressing. But I have not been able to do a CALL LOAD to a memory location between >9000 and >97FF with my Version 110 TI EXTENDED BASIC. I am limited to using E/A BASIC. I can CALL PEEK with XB in this area all day long but no writing into this area. Oh well!

The external side of the 8255 can be connected to a variety of different components depending upon the task to be done. The I/O ports of the interface can be connected to any TTL (0 to +5v) signal. If the ports are to be connected to additional IC's on the I/O board then the 8255 pins can be directly connected to the on-board chips without any isolation. But if signals from off the board (the real world, a model railroad) are being brought in to the ports then the 8255 should be isolated from these signals with an IC that the signal goes through before the port. This is to protect the $2 8255 chip from accidental shorts or high voltages. It's better to fry a 35 cent chip than the $2 chip.

For the 24 point digital input card, the input circuit schematic shows the input circuit for each input bit. The 24 point input material list itemizes the components needed. The inverter gate of the 74LS04 isolates the 8255 port from external damage and is sacrifices should the circumstances arise. The resistor pulls the input side of the inverter to a reference voltage so that unused inputs will have a stable value of zero or low.

For the 24 point digital output card the output circuit schematic shows the circuit for each output bit. The 24-point output bill of material lists the components for this I/O card. The non-inverting 74LS07 buffer uses an open collector output to drive the base of the I/O card output transistor (that's a mouth full). A high signal from the 8255 port will turn the transistor on, allowing a current to flow through the transistor. The open collector output can be used at higher voltages than exist on the output card. Part 3 of this series is planned to discuss connecting the outside world to the I/O cards. But I've included some external circuits to help you get a feel how these outputs are used.

An I/O card with 24 input points constructed as discussed draws about 70 milliamps from the power supply. A 24-point output card draws around 260 ma when all output are low or off. With all 24 outputs driving an LED (will be described in a moment) with 20 ma each, 740 milliamps will be required per output card. The 74LS07 chips are rather expensive, about $1 each and a 7407, which is sunstantially cheaper, can be substituted. But the penalty is that about 120 more milliamps are required upping the total to 860 milliamps. The point is, you should estimate you power requirements ahead of time.

The components for the input card and the output card should be mounted on the I/O circuit board between the 8255 and the 24-pin headers. Layout the chips for the computer side of the I/O card on the lower 1/3rd leaving 2/3rds of the circuit board for the external components. There will be no problem with space for the input circuit components. But there are *more components* required for the output circuit and you will have to work out the best solution for your application.

An output test card is constructed to check the function of the 24-point output I/O card. Once the output card is confirmed to be functional, a input test jumper is installed between the output card and the input card to be checked. A value is written to an output port. This value goes to the input port via the test jumper. The input port is polled and the value read is compared to the original value that was written. They should be the same. The material list for the output test card is for a 24-point output I/O card and contains 24 LEDs so all ports can easily be tested at once. The input test jumper provides for the connection of all 24 outputs to the 24 inputs. It will be tempting to disregard building the test apparatus, but it is worth your time. The output test card is ideal for developing software. I am already doing the preliminary software work for the railroad operating system I'll need and I haven't started construction of the layout that will be computerized. Guess what I am doing this winter! Following is a E/A BASIC listing that can be used to test the output card and another to test the input card using the output card and jumper.

I/O OUTPUT CARD TEST PROGRAM

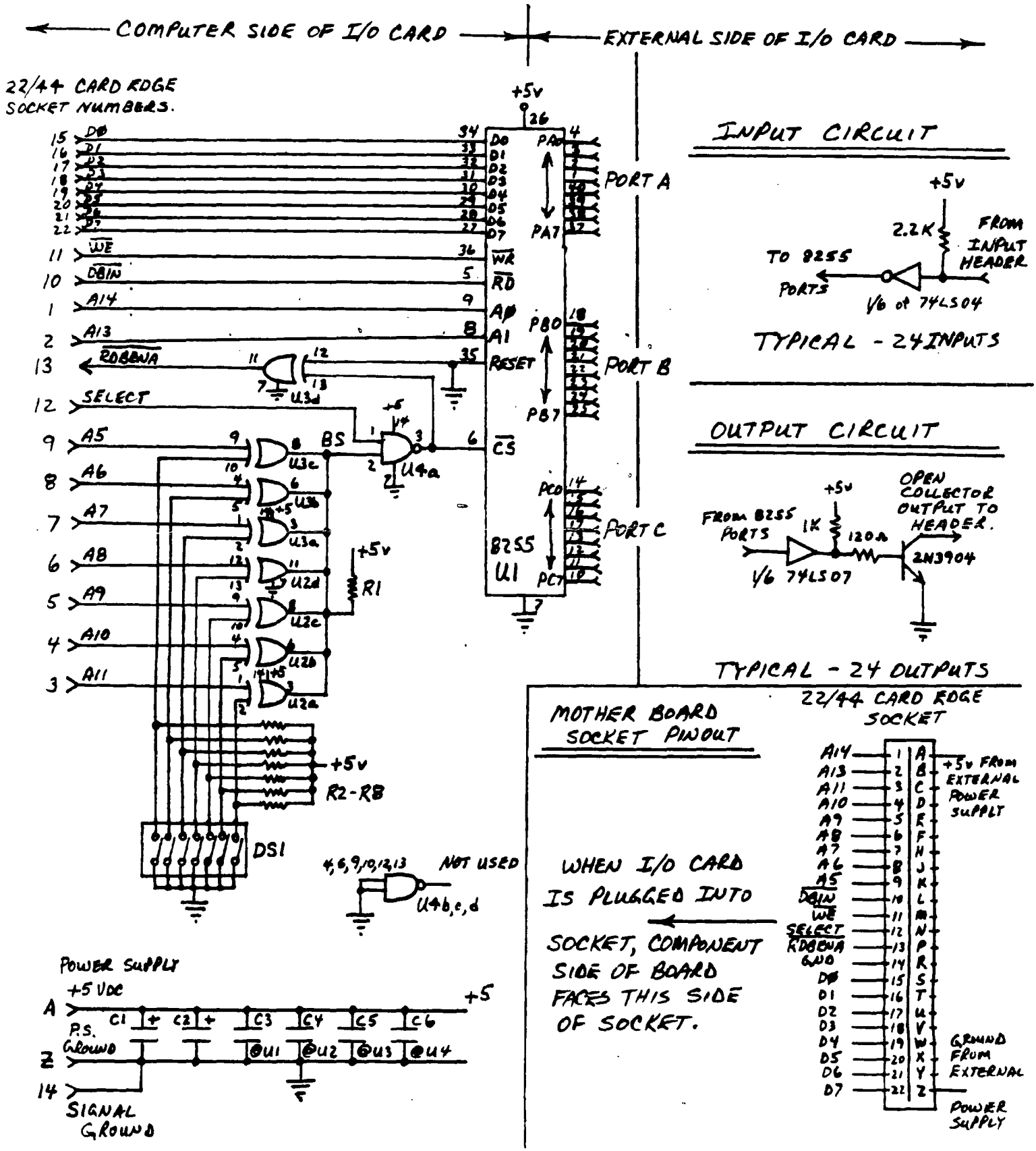
```
100 CALL INIT
110 A1=-28636            Port A at address >9010.
120 B1=-28634            Port B at address >9012.
130 C1=-28632            Port C at address >9014.
140 R1=-28630            Register at address >9016.
150 OUT=128              Setup as output card.
160 CALL LOAD(R1,OUT)    Write setup to Program Register.
170 FOR I=0 TO 255       Loop.
180 CALL LOAD(A1,I)      Write to port A  LEDs light.
190 CALL LOAD(B1,I)      Write to port B  LEDs light.
200 CALL LOAD(C1,I)      Write to port C  LEDs light.
210 NEXT I               Loop again.
220 CALL LOAD(R1,OUT)    This will RESET outputs to zero.
230 END
```

I/O INPUT CARD TEST PROGRAM USING OUTPUT CARD AND JUMPER

```
100 CALL INIT
110 A0=-28672            Card 0, addresses for
120 B0=-28670             ports A, B and C.
130 C0=-28668
140 R0=-28666            Program Reg. Address, card 0.
150 IN=155               Setup as input card.
160 A1=-28636            Card 1, addresses for
170 B1=-28634             ports A, B and C.
180 C1=-28632
190 R1=-28630            Program Reg. Address, card 1.
200 OUT=128              Setup as output card.
210 CALL LOAD(R0,IN)     Write setup to card 0.
220 CALL LOAD(R1,OUT)    Write setup to card 1.
230 FOR I=0 TO 255       Loop.
240 CALL LOAD(A1,I)      Write to port A, card 1.
250 CALL PEEK(A0,A)      Read port A, card 0.
260 IF A=I THEN 280      Is it correct? Yes, goto next.
270 PRINT "PORT A BAD READ"  No! print message.
280 CALL LOAD(B1,I)      Ditto for ports B.
290 CALL PEEK(B0,B)
300 IF B=I THEN 320
310 PRINT "PORT B BAD READ"
320 CALL LOAD(C1,I)      Ditto for ports C.
330 CALL PEEK(C0,C)
340 IF C=I THEN 360
350 PRINT "PORT C BAD READ"
360 NEXT I               Loop again.
370 CALL LOAD(R1,OUT)    RESET outputs to zero.
380 END
```

Finally, I am willing to try and answer any questions regarding this project. Please send a SASE along with your questions and I'll try and respond as my time permits. My name and address are: Rod Cook, 2314 Fallsburg Rd. NE., Newark, Ohio 43055. If you would like a copy of the TIW files for these articles, copies of the schematics along with the data sheet for the 8255 and any other info I may have on disk pertaining to this project, send me $4 and I'll return a disk with copies of the schematics and data sheets. The $4 covers my costs of copying, shipping, a disk and a little booster for me to reward my effort.

I/O CARD SCHEMATIC



PARTS LIST - I/O CARD - COMPUTER SIDE:

1 - PERF-BOARD W/GROUND PLANE RS 276-188
C1, C2 - 10 µf 25V TANT. CAPS.
60-70 T44 MINIWRAP WIRING TERMINALS JA T44-1
3 - 14 PIN WIRE WRAP IC SOCKET JA 14 PIN WW
1 - 40 PIN WIRE WRAP IC SOCKET JA 40 PIN WW
U1 - 8255A-5 PROG. PERIPHERAL I/O
U2, U3 - 74LS136, QUAD XOR - OPEN COLLECTOR
U4 - 74LS00, QUAD NAND
R1 - 1K 1/4 WATT RESISTOR
R2-R8 - 2.2K 1/4 WATT RESISTORS
C3-C6 - .1 µf MYLAR BYPAS CAP.
DP1 - 8 POSITION DIP SWITCH RS 275-1301
2 - 12 pin Rt. ANGLE FRICTION LOCK .1" HEADER DK WW 4310.
2 - 12 POSITION .1" CRIMP TERMINAL HOUSING DK WM 2010.
24 - CRIMP TERMINALS FOR .1" HOUSING. DK WM 2200

PARTS LIST - 24 INPUTS - EXTERNAL SIDE

4 - 74LS04 HEX INVERTOR
4 - 14 PIN WIRE WRAP IC SOCKET JA 14ANWW
4 - .1 µf MYLAR BYPASS CAPS.
24 - 2.2K 1/4 WATT

PARTS LIST - 24 OUTPUTS - EXTERNAL SIDE

4 - 74LS07 HEX BUFFER/DRIVER - OPEN COLLECTOR
4 - 14 PIN WIRE WRAP IC SOCKET JA 14PWWW
4 - .1 µf MYLAR BYPASS CAPS.
24 - 2N3904 NPN TRANSISTOR
24 - 1K 1/4 WATT RESISTORS
24 - 120 ohm 1/4 WATT RESISTORS

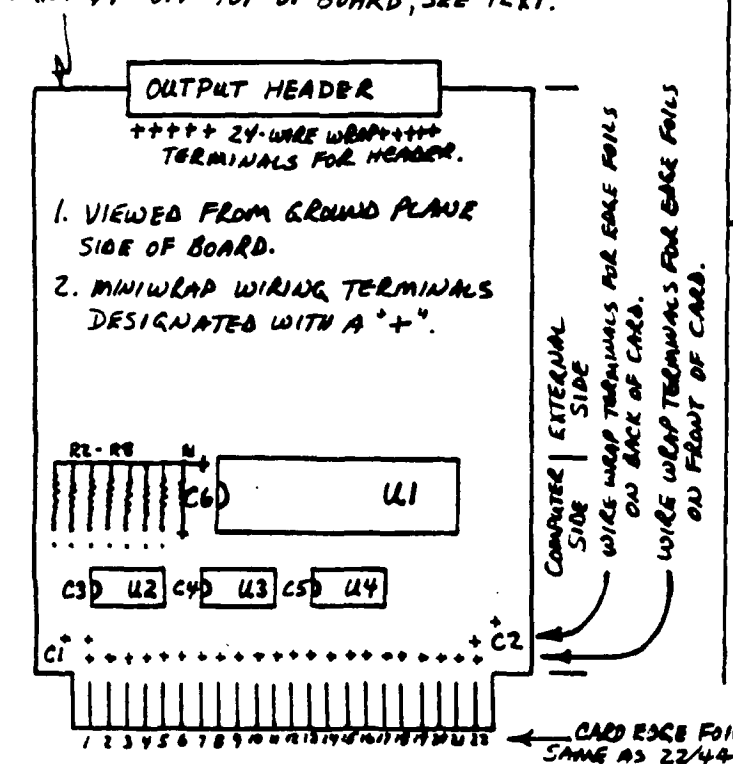
PARTS LIST - OUTPUT TEST CARD

1 - GRID BOARD W/SOLDER-RINGED HOLES RS 276-158
1 PKG. OF 10 - 3 CIRCUIT P.C. BOARD CONNECTORS DK WM 3300
24 - RED LEDS.
24 - 120 ohm 1/4 WATT RESISTORS.
1 LENGTH OF STRANDED WIRE.
1 ALLIGATOR CLIP.

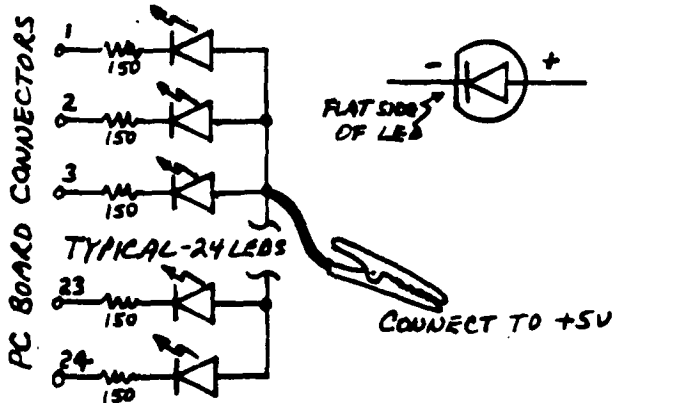
PARTS LIST - INPUT TEST JUMPER

4 - 12 POSITION .1" CRIMP TERMINAL HOUSING DK WM 2010
48 - CRIMP TERMINAL FOR .1" HOUSING DK WM 2200
24 - LENGTHS OF WIRE.

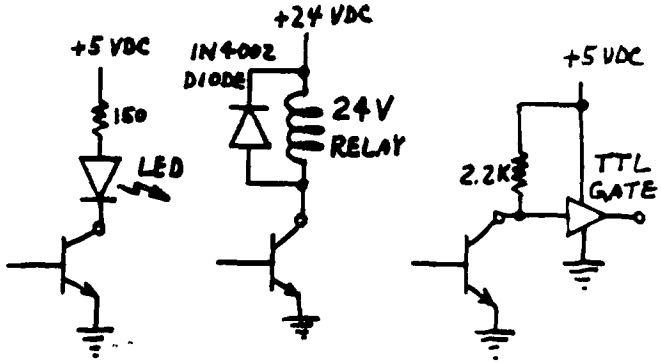
I/O CARD LAYOUT



OUTPUT TEST CARD



EXTERNAL OUTPUT CIRCUITS



The CLEVELAND AREA TI99 USER GROUPS had an interesting article in their November 1987 newsletter. The article appears on page 3 of the newsletter and it was written by Martin Smoley, President of the Northcoast chapter of the Cleveland Area User Groups. It's nice to know to have these guys as part of our local user groups.

. . . . . .

HORIZON COMPUTER
AND
BUD MILLS SERVICES
TESTIMONIAL

I am writing this short testimonial because I would like to show my appreciation for these courteous and helpful people. I purchased one of the green cards (which was one of the first) from Ron Gries of Horizon Computer soon after they came out. I purchased the balance of the kit from Bud Mills shortly thereafter. The parts were very easy to assemble, and the instructions were clear and complete, but I still found myself calling both Bud and Ron several times to check on the procedure they thought would be best. I received more help, more information, and more friendship at that time than I have ever received from any company. I realized later that at the prices they were charging, they couldn't have been making much profit, and therefore, were doing all this because of their love for the TI 99/4A. I talked to Bud Mills again this week, and I find that whether you spend money or not, he is as informative, helpful, and friendly as ever. He is as enthusiastic about the new improvement to the TI as when I first talked to him. For these reasons and many others, if you are thinking about purchasing a RAMDISK, I would recommend contacting Bud Mills Services for more information.

HARD FACTS FROM
BUD MILLS SERVICES
10/20/87

Due to the fact that there are so many rumors floating around concerning RAM Disks, I decided to check with Bud Mills Services, and get some facts. I talked to Bud last night, and here is as much information as I could take notes on, or remember.

Bud will be at the Chicago TI-Faire, and will be showing some really new and really big items. His standard RAMDisk kits will be approximately 104K, 192K, 256K, 384K, and 1 Meg. All of the new kits use HMS62256 32K memory chips. This makes it possible to expand any of them to as much as 3 Megabytes by adding more chips. The RDS/Menu program which will be supplied with the new cards is Version 7, and is capable of handling a card from SS/SD into the Megs without any modifications, and can partition a Megabyte into more than one drive for efficient use with no user effort. The 1 Meg will cost $399, which is a big byte for some, but the smaller SS/SD kits will probably start around $150. Remember that you can expand a 104K card up to 3 Megabytes whenever you get the urge. Bud also has an upgrade for the older boards which use 8K chips to bring the 192K up to 256K, for only $27, "That's Great!" And last, "but I think, almost best", Bud has a kit to put the 32K of RAM in the console and it is attached to the 16 bit (high speed) buss of the 9900. This kit is only $25, and as I write this article my kit is already on the way. If you are interested in any of these items, I recommend you contact Bud Mills Services at 166 Dartmouth Drive, Toledo, Ohio 43614. Phone (419)385-5946.

. . . . . .