RAMROD MMOS

IMPORTANT INFORMATION

READ THIS before you install your new RAMROD board in your computer.

First we will discuss the layout of the board. Referring to fig. 1 notice there are five 24 pin sockets numbered Z6 thru Z10 at the top of the board. Also note the two dip switches towards the bottom of the board labeled S1 and S2. Now being familiar with these locations, we will give detailed information about each one.

SOCKETS

Z6:

This socket is addressed as memory locations F000-FFFF by the computer. If you are using the operating system ROMs from your original board, remove the C014599 ROM from A401 of the old board and install it in this location. Refer to S1 switch settings for proper operation. OSN EPROM may be installed in this socket.

Z7:

This socket is addressed as memory locations E000-EFFF by the computer. If you are using the operating system ROMs from your original board, remove the C012499 ROM from A403 of the old board and install it in this location. Refer to S1 switch settings for proper operation. OSN EPROM may be installed in this socket.

Z8:

This socket is addressed as memory locations D800-DFFF by the computer. This is the floating point ROM. Remove the C012399 or FASTCHIP ROM from the old board and install it in this location. There are no switch setting requirements for this location.

THE OMNIMON! ROM MAY BE INSTALLED IN SOCKET Z9

Z9 and Z10:

These sockets are addressed as memory locations C000-CFFF by the computer. These are the memory locations that are not currently addressed by the stock machine. Refer to S2 switch settings for proper operation. Your board should come with the switch settings correct, but double check them to be sure they have not been changed. Also see updates at back.

Z9 and Z10 Options:

1.Z9-2K RAM, Z10-2K RAM
2.Z9-2K RAM, Z10-2K ROM or EPROM
3.Z9-2K ROM or EPROM, Z10 2K RAM
4.Z9-2K ROM or EPROM, Z10 2K ROM or EPROM
5.Z9-4K ROM or EPROM, Z10 BLANK
6.Z9-BLANK, Z10 BLANK

SWITCH SETTINGS

S1:Z6 and Z7 control.

ATARI ROMS	EPROM (OSN)
1 on	1 off
2 off	2 on
3 on	3 off
4 off	4 on
5 off	5 on
6 on	6 off
7 off	7 on
8 on	8 off

S2:Z9 and Z10 control.

4&8K EPROMs OMNIMON etc	Z9 and Z10 RAM or 2K RAM, 2K EPROM.
1 off	1 on
2 on	2 off
3 off	3 on
4 on	4 off

IF ALL SWITCHES ON S2 ARE IN THE OFF POSITION, THE RAM OR ROM IN SOCKET

Z9 WILL BE MASKED FROM THE SYSTEM. (SEE UPDATES AT BACK)

INSTALLATION

In all cases, the component side of the board FACES THE BACK OF THE COMPUTER and ALL I.C.'s HAVE THE NOTCHES FACING THE OUTWARD EDGE OF THE BOARD.

Take the ramrod board after you have configured it and push it into the socket where the original operating system board was. This method allows free air movement around the components for better heat dissipation. If the board does not stand up straight after installation, this indicates that the socket on the motherboard is misaligned, and it should be corrected to insure trouble free operation. DO NOT POWER UP THE COMPUTER if misalignment causes the board to touch the metal casing of the computer.

WARRANTY

This product is warranted for a period of one year from date of purchase against defects in material and workmanship. Return to Newell Industries, 3340 Nottingham Ln., Plano, TX. 75074, with proof of purchase for warranty repair or replacement.

GENERAL INFORMATION

As you can see from the previous information, you have purchased a very flexible piece of equipment. Although the additional 4K of address space for ram or rom will be of great benifit to most people, we believe that the option of being able to modify your operating system or create your own is the most powerful feature of the RAMROD OS board. Because the OS is the controlling factor of the computer, it would be possible to put your software in this location that would immediately be present on power up with no loss of ram for program storage. See the feature sheet of the OSN operating system included with the RAMROD.

Some of the more simple changes would be to change some of the default parameters of the OS. These would include screen colors, margins, character set, messages, and what takes control of the system on power up. Someone may want DOS booted all the way, regardless of the cartridge('s) installed.

If there are any bugs in your current OS, you will be able to try and correct them. If you come up with a patch to correct a bug, write about it, there are several magazines that would be interested in publishing this type of information.

MODIFYING YOUR OS

There are several different methods for doing this, we will discuss a couple of these.

Since your current OS is now in rom where it cannot be changed, we must get it to ram or disk where we can change it. This is really a very simple task to accomplish. DO NOT use DOS to save any ROM based memory locations. Because of the way burst IO works, all you will get is a file that will not load. If you already have an EPROM programer, more likely than not you will be able to change it's buffer locations and save the buffer to disk. Then using any of the availible disk editors you can change all or part of the code. You would then use an Eprom programer to write your new or modified OS into Eprom.

Another method would be to use the debug feature of an assembler/editor, move the code to ram, modify the part you want, and then save it to disk. At this point all you would need to do would be to use an Eprom programer to write your new OS into EPROM's. This would seem to be the easiest and most versatile method because of the speed with which you can work with ram.

EPROM REQUIREMENTS

1.4K EPROM - 2732 style pinout, 350 ns. max. access time
2.2K EPROM - 2716 style pinout, 350 ns. max. access time

RAM REQUIREMENTS

2716 compatible pinout, 2Kx8 bit static ram, 250 ns. max. access time.

OPERATING SYSTEM MODIFICATIONS

Here are a few changes you will want to make. These changes are for the REV. A or B roms, with the REV. B rom locations in parentheses.

First we will start with a one byte change at location E88E(E869). The original value is 6, and this controls the time interval for the repeat function of the keyboard. By changing this location to 3, the cursor will move twice as fast for repeating characters. This also means cursor movements using the control keys. This means faster editing of programs, because it is the O.S. it means twice the speed on any program, word processors, spreadsheets, etc. You will really like this change.

Next we will attack the debounce feature. At this time we will also shorten the time for the repeat feature to work. For this we will have to change two bytes. First at location FFE3(FFE3) change the 3 to a 1. This cuts the time by 2/3 but still operates without any problems. Next, change the byte at location FFEC(FFEC) from HEX 30 to 10. This will allow the repeat key to operate a lot faster. Two seconds is a long time when you're trying to move the cursor to edit something or just wanting to print a row of dashes. These three changes will approximately double the speed of the keyboard response time.

This next mod will shorten the warning bell. There are a couple of ways to do this. I took the one that will also shorten the click. At location FCD9(FCD9), change the value from 7F to 3F. This will cut the bell (buzzer?) time by more than 1/2. This still gives you plenty of time to quit typing before getting to the end of the line. This still allows for a click that you can hear when you hit a key. If you want to get rid of the click altogether, change the locations starting at F6FB(F6FB) from 20,D8,FC to EA,EA,EA. This NOPs the JSR to the click routine. With these changes, even the fastest typist shouldn't miss getting any keystrokes.

A few other simple changes are left margin default at location F174(F174) (now 2), right margin at location F178(F178) (now max at hex 27 [39]), default color tables starting at FEC1(FEC1) thru FEC5(FEC5), memo pad message starting at F0F3(F0F3) thru F10B(F10B), boot error message starting at F10D(F10D) thru F116(F116).

These next patches will increase the cassette baud rate to approximately 885 baud from the current 600 baud.(See OSN feature sheet) Also we will change the leader time from 20 seconds to about 10 seconds. This should give you about 50% faster load times on programs saved after these patches are implemented. This will also be compatible with the current O. S., so you will still be able to read old tapes, and people reading your tapes will have the benefit of the increase in speed. Note that the hardware will not support a baud rate much higher than what we will use. Experiment at your own risk.

CHANGES:
EB8A(EB86) from CC to 00 (lo byte of write record baud rate)
EB8F(EB8B) from 05 to 04 (hi byte)
EFA2(EFA2) from CC to 00 (baud rate in open routine)
EFA7(EFA7) from 05 to 04 (hi byte)
EFBB(EFBB) from 04 to 02 (leader time in open routine)
EF42(EF42) from CC to 00 (baud rate in init routine)
EF47(EF47) from 05 to 04 (hi byte)

OSN V6 OPERATING SYSTEM FEATURES

GRAPHICS MODES 12-15

Use the standard graphics call from basic to access the additional modes.
Graphics 12 - Full or split screen character graphics, upper, lower case, & control.
Graphics 13 - Full or split screen character graphics, upper, lower case, & control.
Graphics 14 - Full or split screen hi-res 2 color graphics mode.
Graphics 15 - Full or split screen hi-res 4 color graphics mode.

MODE	MODE TYPE	HORIZ.	VERT.S/S	VERT.F/S	COLORS
12	TEXT	40	20	24	5
13	TEXT	40	10	12	5
14	GRAPHICS	160	160	192	2
15	GRAPHICS	160	160	192	4

CASSETTE INTERFACE

On power up the cassette baud rate is set to $5CC (751=5, 750=204).
On ALL cassette handler calls the baud rate used is what is in these locations.
By changing locations 750 & 751 (dec.) you can set the cassette baud rate to any desired rate wanted (see note).
EXAMPLE: POKE 751,2 from basic sets the baud rate to 1250.
POKE 751,4:POKE750,0 sets the rate to that used in OSN V4.
NOTE: We recommend only changing location 751. This will be easier to remember. A 2 in this location will give you a baud rate of about 1250. Even though the OS will support baud rates far above this, your cassette will not. The maximum we could get out of a standard cassette was 1525, and this will very from one unit to the next. SYSTEM RESET will reset the baud rate to default.

KEYBOARD

The keyboard response has been doubled in speed, allowing faster cursor control and typeing.

SYSTEM POWER UP

To disregard cartridges during power up, the following sequences should be followed.
1. Install cartridge and power up in the normal manner.
2. Hold the SHIFT key down and power down and back up within 3 seconds.

The OS will then continue the boot process disregarding the cartridge installed. If you also hold the START key down, it will try and boot the cassette. Etc.

COMPATABILITY

OSN V6 meets all operating system entry requirements, and as many OS subroutines as possible were kept with the same entry points. All handler entry points in V6 are the same as the Atari OS. This allows compatability with almost all existing software.

COMPATABILITY PATCHES FOR OSN

Although OSN meets all OS entry requirments, there is at least one product that will not run with our OS. (There are a LOT more that won't run on the new computers). Believe it or not it is the old disk version of the Atari Word Processor. The reason for this is their use of two four byte tables in the OS. They broke their own rule in this case. If you use this product, we will give you the means to patch it so it will run with OSN and probably the new computers. This involves installing the tables in their code, and then changing their calls to the new loacations within the code.

There is a message in the code that says "Insufficient Ram, 48K Minimum Required". We will replace the word 'Required' with the two 4 byte tables that we need (ATAINT,INTATA), and then change the refferences to point to our tables.

First, using OMNIMON! we must locate this address in memory. Boot the software and then enter Omnimon. Then search memory for "Required". Make a note of the address of the first 'e'. This will be where we will put the tables. Now that we know where to put them we can continue with our patch.

Using Omnimon, read the first 128 sectors of the disk into memory (R1 700 80). Search memory in char. mode for ' Required' (note the space). Once found, go to hex mode and replace it with 9B 9B 40 00 20 60 20 40 00 60. The 9B's at the start are for the end of message. Now in hex mode, search for refferences to FEF6 (S700 F6 FE) and FEFA (S700 FA FE). When located, replace these addresses with the new table addresses that you have made. The first table starts where you made note of the 'e'. For the second, just add 4 (in hex). Remember that all addresses refered to are in low,high order.(ei.192A would be installed as 2A 19).

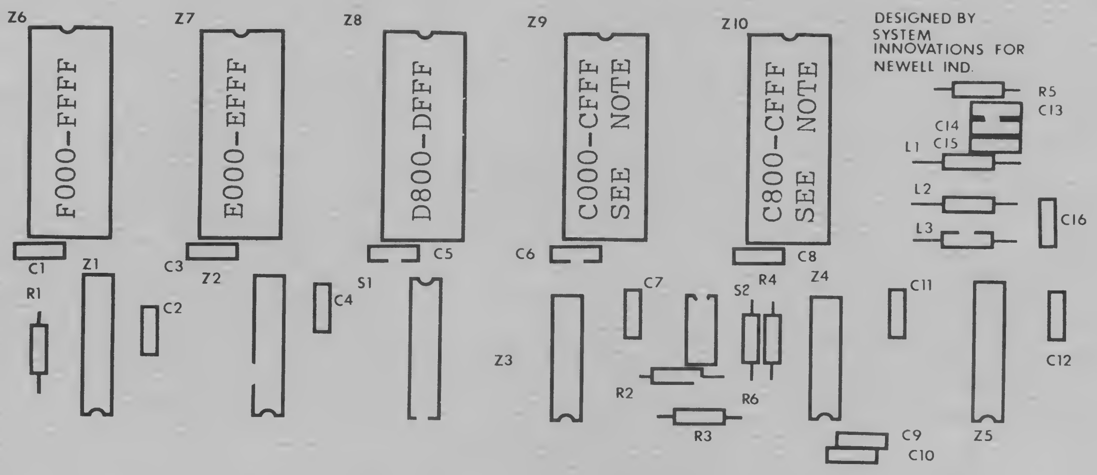
Now all that is left to do is to write the sectors back to disk (W1 700 80). If all is done properly you will have a program that will run under all operating systems to date.

Several popular programs by Synapse have also used illegal points in the operating system. Of the ones we have encountered, most problems are keyboard repeat key functions. Rumor has it that Synapse will correct this problem if they recieve enough complaints. For the time being, try searching the disk for the assemlbly code LDA $022B, CMP #$19. When found, change the $19 to $05. This will slow down the keyboard so you can use the program with OSN. Synfile by Synapse also does a LDA $FCD8, CMP #$A2 to determine if you are using an XL machine. To run on OSN change the $A2 to $A5, or replace their code with LDA $F11B, CMP #$78 to run on OSN and the standard OS.

If you encounter a problem with some software that does not work with OSN, contact the software company that produced it. They have the source code, and should be able to help you. As stated earlier, OSN meets ALL operating systems rules. If you cannot get the software company to help you, we will try to locate the problem. Although without the source code, we cannot promise to find the problem.

Most illegal calls to the OS are done with a JSR or JMP instruction (JSR=$20,JMP=$4C), although any reference to $E500-$FFFF are illegal. If you have a program that doesn't run, searching for illegal calls may locate the problem. A popular autorun program uses a JSR to the OS (3 bytes) to INC $09, RTS, also 3 bytes (makes a lot of sense doesn't it). If you have an XL and wonder why your autorun program doesn't work, guess what? Yes, replace the JSR with the proper code and it will work.

Figure 1
COMPONENT LAYOUT



NOTE: Socket Z9 can be addressed as C000-C7FF or C000-CFFF depending on the setting of S2

NEWELL INDUSTRIES
P.O. BOX 253
WYLIE, TEXAS 75098

RAMROD UPDATES

This information is provided for those who wish to use it, it is not needed for proper operation.

1.DISABLING OMNIMON WITH EXTERNAL SWITCH (two methods, a or b)

(a).Install a SPST toggle switch across S2 dip switch 4. Turn switch 4 off, leave 2 on. you can now control the availability of OMNIMON using the external toggle switch.

(b)To control sockets Z9 and Z10 as two 4k banks, with masking feature, install a SPDT(center off) toggle. Connect center post of switch to hole between S2 switch 4.(SEE DRAWING) Turn switch 4 of S2 off. Connect one of the other post to pin 18 of socket Z9. The remaining post to pin 18 of socket Z10. REMOVE 74LS138 I.C. from socket Z1. Install a 5K (approx.) 1/4 watt resistor per drawing 1b on back (if not already present). You can now have two 4k eproms(roms) on the board at the same time. The center off position will mask out both sockets. To install ram, you must use the center off position and reconfigure S2 to the proper settings, and install the 74LS138 previously removed.

Using this method you may install the 8K OMNIMON in Z9 and and still have another 4K bank left for other features (romdos etc.). It is possible to have 16k in 4k bankable blocks using this method.
NOTE:On serial numbers less than 00201, an additional 5K resistor must be installed between pin 18 of socket Z9 and +5V. This will be evident if only one resistor is to the right of the S2 DIP switch.

2.USING 8K EPROMS FOR OPERATING SYSTEMS

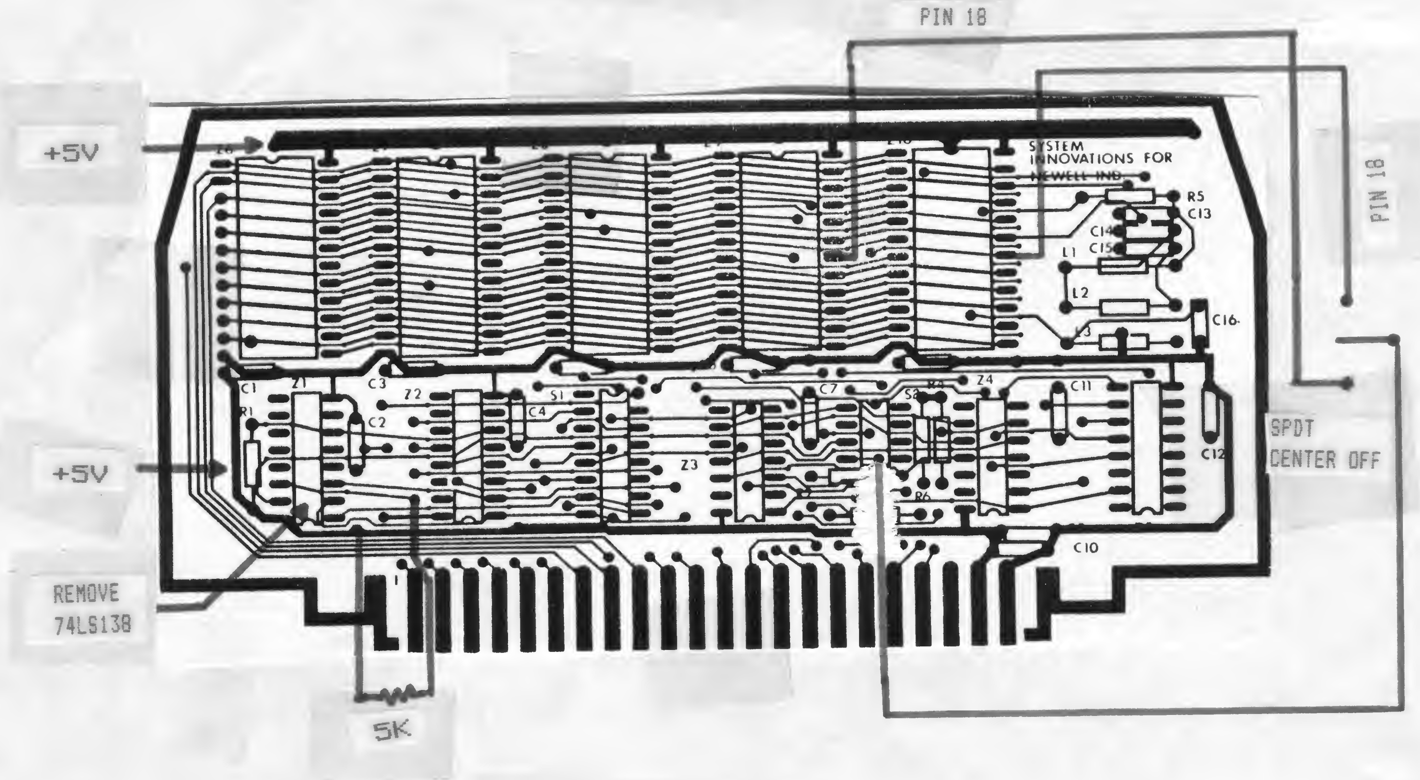
Following these instructions will allow the use of two operating systems on the board. These modifications will allow only the use of 2764 EPROMS in sockets Z6 and Z7. (SEE DRAWINGS)
A.Install a 5K 1/4 watt resistor between pin 18 of Z6 and +5V. Install another 5K resistor between pin 18 of Z7 and +5V. Or install these two resistors per the drawing. Either method works.

B.Using a single pole, double throw (SPDT) toggle switch, connect the center lug to pin 20 of Z6 or Z7 (these pins are common).
C.Conect one side of the switch to pin 18 of Z6, the other side to pin 18 of Z7.
D.After burning the operating system (E000-FFFF) desired into a 2764 EPROM, common pins 26,27, and 28 of the EPROM. Do this by CAREFULLY soldering a small wire accross the TOPS of the pins (there must be room to install pin 26 into the socket).
E.Solder a 3" insulated wire (small diameter) to pin 2 of the 2764s and plug the 2764 EPROMs into sockets Z6 and/or Z7. Make sure the exposed pins are at the top.
F.Pin 2 of the 2764 EPROMs must be connected to A12 on the ramrod board. A12 is located at the feedthrough hole between switch 8 of the S1 dip switch. Choose a routing method that will keep the wires out of the way.
G.Reset S1 switch as follows: 1,2,3,4,5,8 OFF; 6,7 ON.
H.Install the ramrod and test each operating system. If problems occur, recheck your modifications.

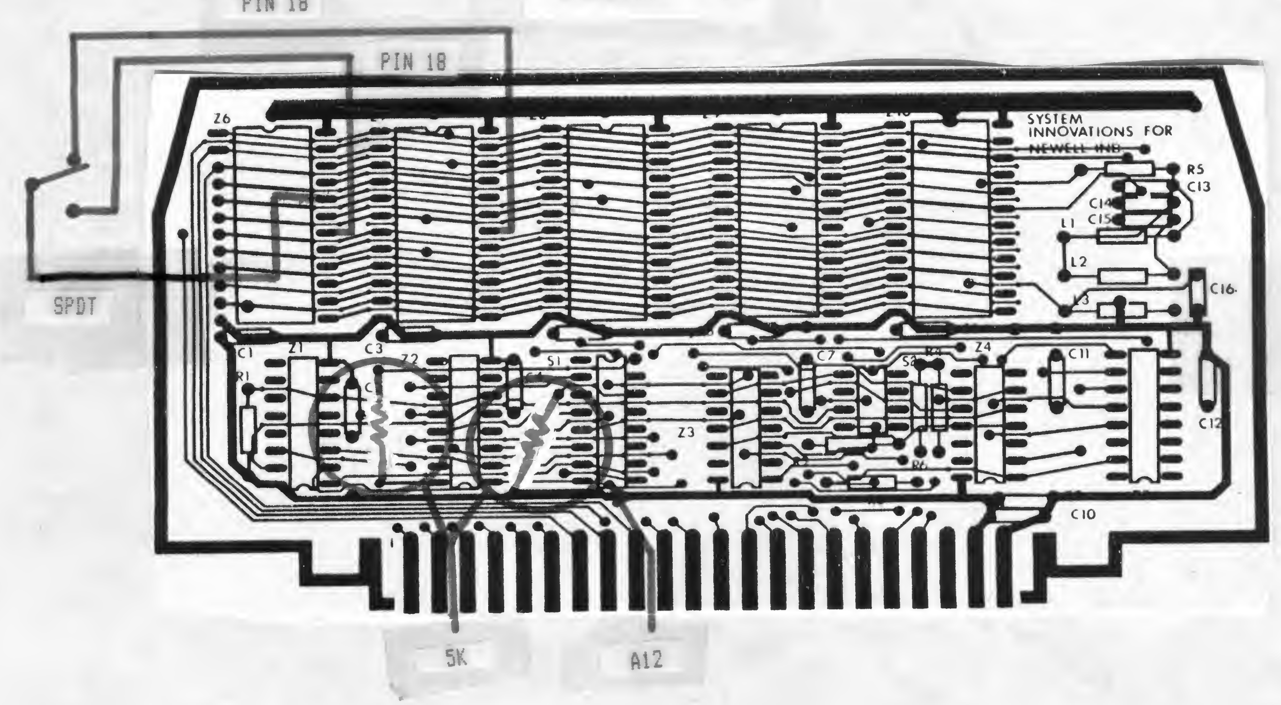
DRAWINGS ON BACK

NEWELL INDUSTRIES
P.O. BOX 253
WYLIE, TEXAS 75098

MOD 1(b)



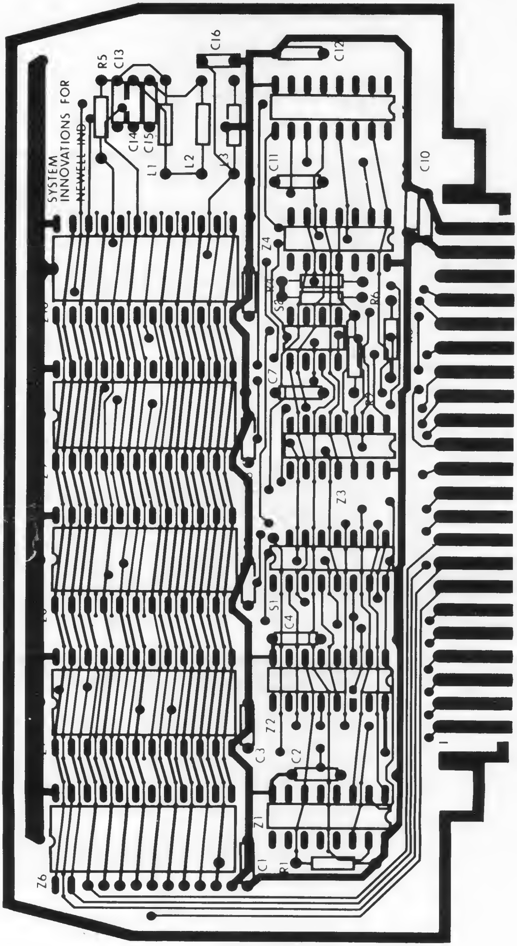
MOD 2



NEWELL INDUSTRIES
P.O. BOX 253
WYLIE, TEXAS 75098

NEWELL INDUSTRIES

RAMROD MMOS
CUSTOM OPERATING SYSTEM BOARD
FOR THE ATARI 800*



* Trademark of ATARI INC.