$2.00

VOLUME 2 · 1987
VER 21
APRIL

R/D COMPUTING

Dedicated to TI 99/4A and 9900 Computer Systems

Ryte Data

210 MOUNTAIN STREET,
HALIBURTON, ONTARIO K0M 1S0
(705) 457-2774

April... isn't Spring NICE. Hey, this issue reached you before the end of June? One of the minor goals was to get back "on schedule" here. Red faced and all. Hope you like it. There are some real gems here - with an article by Tony Lewis on memories.

What was I talking about? Oh yes, the world is full of three kinds of people. Those who make it happen, those who watch it happen and those who wonder what happened.

After numerous false starts Myarc has been shipping the first few machines. I would suggest that you place your order for Myarc's 9640 if you have any intention of buying this superior new computer. This will: tell Myarc how many to build, support the company, expand your computing power and enjoyment, help keep the entire TI 99/4A community alive, show your friends a new level of computing ability, reward you for hanging in there in the face of all adversity and, last but not least, give your local TI retailer a few bucks for supporting 99xx systems.

The best price seen to date is $415 US. This will translate to about $620 (CDN). List price is $530 with the keyboard thrown in. Watch for details over the next few weeks.

Being personally addicted to gadgets of all sorts (you should see the surroundings here), I feel that the Myarc 9640 represents an advance in computing. This is perfectly biased opinion. Several features stand out- independent Device Service Routines, enhanced graphics, 12mHz operation, multiple input choices; including the ability of the 9938 chip to accept graphics information from different sources, on chip memory & clock generator. 7 hardware & 16 software interrupts, DMA (direct memory access) & programmed I/O capability, pre-fetch instructions... PLUS the "old 99/4A" TRUE 16 bit intructions, memory to memory architecture etc.

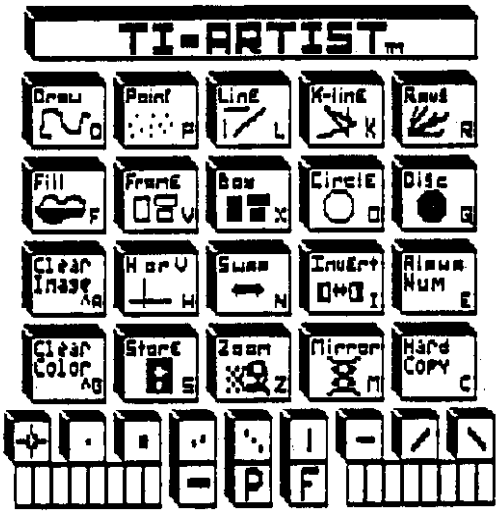
The STANDARD established by IBM does not reflect the state of the art. If only "IBM compatibility" meant truly advanced hardware design.

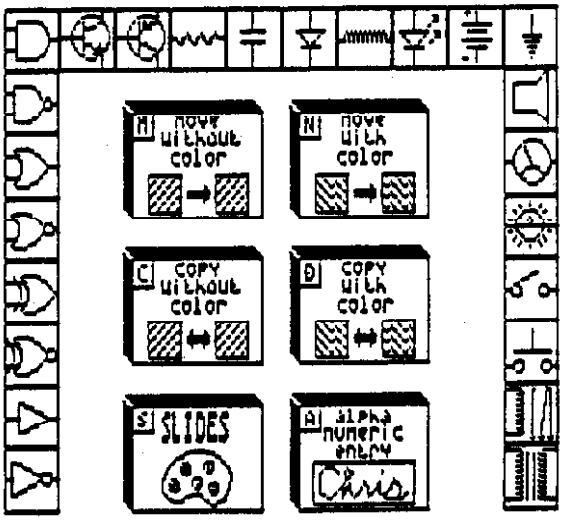
If ALL we needed was a "standard", no room or market for the MAC or AMIGA or ST would exist. To me, the whole argument is significant only in that it underlines the need for computers to advance in several directions at once. Anything else is just a little too predictable for my tastes. IBM did predict that only five mainframes would be necessary to supply the worlds computing requirements in 1975. Today you can put the power of a mainframe on your desk. Compaq effectively beat IBM to the punch by leapfrogging the 80286 technology with their 80386 machine. For a mere $10,000 you can put incredible power on your desk. This too, will take some time to have the software catch up... but it demonstrates what is happening with computers as tools.

In fact, it is my ferverent belief that the more access to information and the more power that folks have on-line, the better. It is only one alternative view to a totalarian "1984" scenario. A view, none the less, that holds that information, the use and the control thereof, is a human right.

Right! I was talking about memories. Well in this issue you will also find articles by Ron Albright, John Clulow and news in the circuit. Make it happen this year. Remember, Your orphan depends on your care!

TI-Artist uses picture menus to show and let you select the desired functions.





Pricing and ordering information:

TI-Artist, $19.95 Display Master, $14.95
Artist Extras, $6.45 All three, $35.00

Inscebot, Inc.
P.O. Box 260
Arnold, MD 21012

Minimum system configuration: Disk system, 32k memory --- and one of these: Editor/Assembler, Mini Memory, Extended Basic, TI-Writer. Fully compatible with TI, Myarc, and Corcomp systems.

Add $1.50 shipping/handling. MD res. add 5% state tax.

C99

C99 Part Two by Ron Albright Jr. Reprinted by permission.

I have been exploring c99 for the TI 99/4A of late. Written by Clint Pulley (38 Townsend Ave., Burlington, Ontario L7T 1Y6) and available as Fairware, the language is a full-featured version of "small c". I have found few limitations with the language (lack of floating point and math routines are the major ones), and have been able to do some nice routines with the language. [Editors note: these functions have been added to c99 with subsequent releases.] Briefly, C is a very popular programming language through which, it has been estimated, 70% of commercial software for other machines is written. So what makes it different? It is a " compiled" language. That means, once you have written your program in c99, you run a companion program called a compiler. The compiler takes your C source code and generates assembly language source code. The resultant code can then be run through the TI Assembler to generate object code, which executes just as fast as if you went through the strenuous (to me, anyway) task of writing assembly source code to start with. C is much easier to learn than Assembly Language and is efficiently compiled with the c99 compiler. I have seen some programs written with c99 alone (there are a few on Compuserve; a simple text editor and a word counter for TI Writer files by Warren Agee, a program similar to the TI Writer formatter, and a graphics demo by yours truly). They are indistinguishable from pure assembly code because the end product is just that. If there is some interest, I will address the language more in depth in some more starter level tutorials. I am no expert, by any stretch of the imagination, but I am learning and plan to spend a great deal of time with the language. Warren Agee, of Livonia, Michigan, has uploaded several tutorials to Compuserve which are mid-level in their scope, excellent in their content and expertise. These will help you as yo get further into the language. It is a marvelous programming tool and, hopefully, this simple file will help you to get started. Learning a new language is never easy, but it is time we all advanced beyond BASIC and started working in another environment. c99 provides a reasonable alternative. I could never think in reverse, so I gave up on Forth; I am too dense to learn assembly language. Pilot is too slow and requires too many disk accesses. Besides, C is used in so many other machines and for so many other applications, it has to be good. Let's begin by seeing what we have to work with.

First, equipment: you need the following - console, monitor, 32k memory expansion, at least one disk drive and controller, the Editor/Assembler package (cartridge or disk version) and, of course, the c99 system disk. A printer is nice (see below) but is certainly not imperative for programming purposes. Ideally, you would have two disk drives as this makes the work much easier, as does having at least double-sided drives (but ain't that always the case!) If you have double-sided drives, you can save yourself a lot of disk-swapping by, first, of course, making a backup of the c99 system disk and, secondly, copying from the Editor/Assembler disk, the files ASSM1, ASSM2 (the files for assembling source code) and EDIT1 (for the E/A Editor) on to the c99 system disk. But, if you have a single drive or single-sided system, don't despair... things will work just fine with what you have.

Once you have gathered your tools, you should get a disk directory printout of the c99 system disk1. Pulley even provides a disk catalog program on the system disk (called "SD" and running out of E/A #5 on my disk) but it doesn't print to the printer). You will notice that there are a long list of files in all shapes and 'colors' (D/V 80, D/F 80 and PROGRAM files). We will first go over what is important and what is not. Some of

Do you want to Easily create:

business cards, letterheads, advertisements, announcements, reports,
presentations, forms, flyers, note paper, book reports, greeting cards
newsletters, exams, school projects, term papers, cartoons,
books, transparencies, bookmarks, annual reports,

or anything else that combines words with pictures?

Then you need
Font Writer II

Font Writer II

- is written in assembly at crucial points for speed,
- lets you use TI-Writer to create the words you want,
- uses standard TI-Artist fonts and pictures (over 100 fonts available alone and literally thousands of pictures!),
- lets you use CSGD graphics too with simple conversion utilities,
- includes a Font Editor for creating that special font,
- includes a Font Manager for the amature or professional font designer,
- interfaces with popular graphics programs,
- includes a variety of pictures and fonts so you can begin right away,
- is a complete electronic publishing package that will continue to be updated in the years to come,
- is compatible with the Myarc Geneve 9640 computer,
- and is fully documented with a 40 page typeset manual written by a professional technical writer.

Requires a disk system, 32K, Extended BASIC, and an
Epson compatible/TI/Prowriter printer.

$24.95 includes shipping and handling

Why settle for less then the best?

Asgard Software
P.O. Box 10306
Rockville, MD 20850
(301) 345-2492

the files you will be using a lot, others seldom if at all, at least to start. Here are some of the files you should have and what they are for. I will list them in order of importance and probable frequency of use.

C99C, C99D, C99E

These are the compiler files. They are the heart and soul of the c99 system. These are PROGRAM image files and are run from Editor Assembler option #5. Unlike some PROGRAM image files, these CANNOT be run from option 3 of the TI Writer module. In my brief experiment, they could not be loaded with the Extended Basic Funnelwriter loader from Australia. The first thing I did with these files was to rename them UTIL1, UTIL2, UTIL3. Then, when you choose the LOAD AND RUN option from E/A (option #5), you only have to hit ENTER and the files will be loaded by that name as a default without having to type the names in.

CSUP

This file is very important. It is a D/F 80 (which always means it runs from E/A option #3) which MUST be loaded immediately after you load your completed, assembled program. We will discuss this more later, but suffice it to say that your c99 program will never run if you don't load this file after it and with it.

C99MAN1, C99MAN2, C99MAN3

These are the D/V 80 files that contain the documentation Clint Pulley provides with the c99 system. They are not going to go very far in teaching you how to program in c99. Like the manual TI provided with the TI Forth system, they are a simple, brief tutorial on how the different files work, and what they do, what the error messages mean etc. They are quite adequate for their intended purposes. Pulley tells you up front "This manual assumes a knowledge of standard C or the availability of a suitable reference." That translates into "If you have never programmed in C, go buy a book!" I will recommend a couple at the end of this article. Fair enough, Clint! If you have a printer, print these files out for future reference. If not, find a friend who does. You will need a hard-copy of these files.

C99ERRORS

This is a short D/V 80 file that contains a listing of the 30 or so error messages that the compiler will embed in your compiled code when it encounters one. It will only embed the error number. You will have to look in this file to find out what the number means. Print this out also.

C99SPECS

A terribly important D/V 80 file. This short file tells you what c99 supports and, more importantly, what it does not support compared to standard C. Why is this important? I have yet to find a book that addresses only "small c", the version of C (more limited than "big C") that c99 is modeled after. All the texts I am aware of cover the full C language. Small c and c99 do not have all the functions of C. When you look at program listings out of these texts, you will quickly become frustrated if you try to type them in verbatim as they are. Many program statements in C will give you errors in c99. You have to study this file when typing in program listings out of books to avoid these errors. For example, C supports "floating-point" arithmetic; small c and c99 do not. There are other examples covered in this file; print it out. You will need it. [Again, note that version 3.0 has added several features].

GRF1DOCS

This is the documentation for the graphics routines supported by the current version (3.0) of c99. Print it out.

ERRFIND1

This is a helpful file provided by

Clint. It is a PROGRAM file to be run out of E/A 5. Run this if you have run the c99 compiler on a source code file and received the dreaded "!!ERRORS!!" message. What it will do is prompt you for the compiled file's name (not the original source file!), read it very quickly showing the file on the screen as it reads it. You can stop to read the file yourself by holding down any key; releasing the key resumes the read. Then, after it has read the file, it will flash the lines again on the screen that contain the error message so you can (1) see where the error occured and (2) what the error message was. It is also nifty for reading ANY D/V 80 text file. It's purpose, though, was to help in debugging.

There are several other files that are, for the most part, files to be included in your c99 source code as you will use certain functions. We will go into this in some depth later, but you will use an "#include dski.filename" in your source file to copy these files into your source code. For example, if you used some graphics commands in your source file to draw some sprites or such, you would need to use "#include dski.grfrefs" in your source code as a line before you started using the graphics commands. Else, the compiler won't understand what they mean and give you multitudes of errors. If you use commands to acces disk files, you would have to use "#include dski.stdio" (which stands for "standard input and output") before you started opening and reading from disk files. Notice the use of lower case letters in these #include statements. The compiler can use lower case, unlike the E/A Assembler which only accepts upper case. Just keep the list of the other files as they will be used as you start to type in programs.

How does onw enter programs with c99? You can do it two ways. You can use TI Writer, but always use "PF" to print to disk rather than "SF". Throw in the "C DSKx.NAME" syntax to clean out all the control characters from the disk file. You, preferably, can use the Editor of the Editor Assembler package. We won't do a program this time, as you have enough to do for now.

What about recommended books? I strongly suggest "C PRIMER PLUS" by Waite, Prata and Martin (Sam's Publishing, 1984). It is 500 pages and costs about $22. It is the "Going Forth (Brodie) for 'C'. It is easy to read, starts at a beginners level and is chock full of example programs. Some usable with the dialect of small c, some not (at least not without some conversions). I went through two other books on C before I found this tome. It is the best I have seen. If you know C, the bible (but much too advanced for me) is "THE C PROGRAMMING LANGUAGE" by Kernighan and Ritchie (Prentice- Hall, 1978). I found a back issue of Byte magazine also useful. The August, 1983 issue is devoted to C and contains some very nice articles and tutorials. You can still get a copy of this from Byte Subscriptions, PO BOX 597, Martinsville, NJ 08836-0597 USA.

There you have it. The second chapter in the "Beginners Guide to c9". If you are interested in more, let me know. I am just beginning myself and we can stumble through this together. My first swim in the waters was just great. Join in and learn c99.

EEPROMS

Electronics, like life, goes in cycles. Memory storage for computers was predominately in silicon chips prior to the mid-1970s. TI's big selling point, or so they thought, was their 'solid state software' cartridges, where programs were permantly stored in ROM and GROMs. But memory chips weren't cheap and most home computers in the early '80s had very little RAM (remember the TI ads bragging about its 16K memory?). Fortunately disk drives came down in costs and became an almost-necessity by the mid-80s. Hard disks (5-20 Mb) are coming within the financial reach of most TI owners in 1987. So now that most of the serious TI users have the majority of their programs on disk, why would anyone even consider a return to storage of programs on chips (aka solid state software) again?

Well, while the price of floppies and hard disks were falling, so was the price of memory chips. Now, thanks to non-volatile memory chips, it is possible to cheaply and permantly store commonly used programs and routines in silicon, and access them as soon as the computer is activated, and much quicker than disk drive speeds.

It was a big step for most owners to free themselves from TI's system of solid state software cartridges by purchasing disk drives. With drives, third party programs could be loaded and ran without swapping cartridges each time another program was loaded. The cartridges contain the programs in ROM (and GROM), which are permantly written, and cannot be modified by the user. Disk software can be modified as needed. But it must be loaded each time the computer is activated. And multiple programs, such as TI Writer or the Editor/Assembler require time consuming disk accesses to load and reload programs as you move from the Editor to Assembler/Formatter to Utility 1, etc.

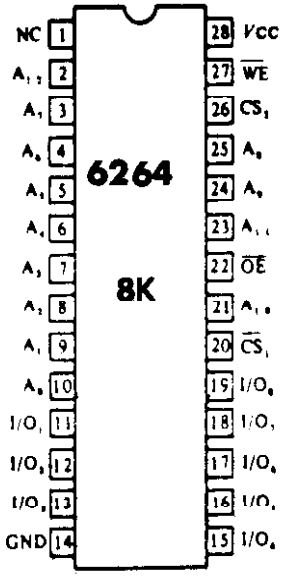
With non-volatile memory, it is possible to store your commonly used programs and routines in 2K or 8K chips which can be accessed upon computer startup. And they can be modified without removal of the chips from the circuit! These chips are called Electrically Erasable Programmable Read Only Memories (EEPROMs), and they are becoming more common and less expensive every day.

Introduction to EEPROMs

EEPROMs aren't new, but a review of the various memory chips available for computers may help some readers better understand current memory technology:

Type	Characteristics
ROM	read only memory, permantly programmed at factory
PROM	programmable read only memory, can be programmed once by user
RAM	random access memory, can be re-programmed an infinite number of times, loses stored data when power is removed
-Dynamic	requires special circuitry and controller that constantly re-stores/refreshes charge to memory cells, allows high density storage in a small chip
-Static	requires no refresh, CMOS versions can use 3V battery to retain data when main power is off; cannot store as much data per chip as Dynamic RAMs
EPROMs	Erasable Programmable Read Only Memory, can be reprogrammed several times; requires removal from circuit; programmer uses high voltages and has timing constraints; requires use of UV lamp to erase programmed data
EEPROMs	Electrically Erasable Programmable Read Only Memory, acts like static RAM, can be re-programmed in circuit - no special voltages or UV erasure lights; main constraint is a Write time takes 10 ms for each byte; can be rewritten 10,000 times, and retains data for 10 years (min) without battery or special circuits; Read times equivalent to most static RAMs

EEPROMs are quite similar to static RAMs. Fig. 1a and 1b compare the pinouts of a 6264 8Kx8 static RAM and a 2864A 8Kx8 EEPROM. They are almost pin for pin compatible, except for the extra CE+ pin on the 6264. Figures 1c-e are of a 2K, another 8K, and 16K EEPROMs. Commonly available EEPROMs run on +5V and draw currents similar



(Top View)

Fig 1b

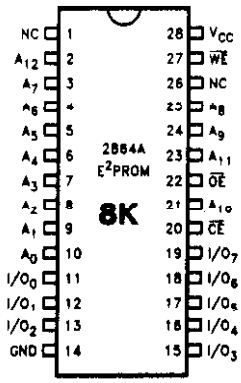
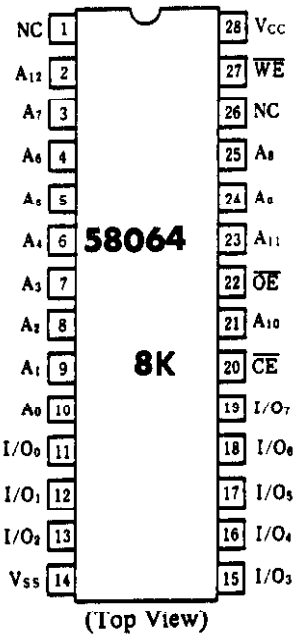


Fig 1a



(Top View)

Fig 1d

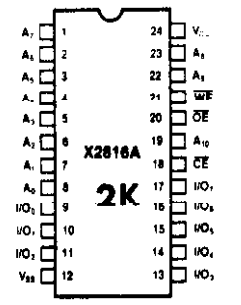


Fig 1c

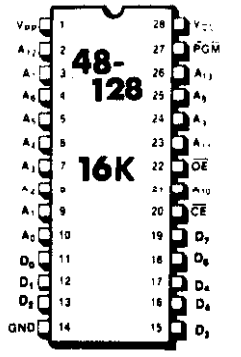


Fig 1e

to non-CMOS static RAM chips. Access times of 250-450 ns are common, although expensive chips with access times of <50 ns are available; this range of access times is compatible with the TI bus. As noted above, they can be rewritten 10,000 times (once a day for over 27 years) and will retain the programmed memory for over 10 years without a battery. Best of all, an EEPROM could be programmed, mailed around the world, and still retain its data. When programming, no high (13-22V) voltages, or special timing constraints are required for the WE or CE signals.

The two main constraints to current EEPROMs are programming time and cost. EEPROMs use on-chip circuitry to generate the special voltage pulses needed to program the chip. The programming cycle requires that the chip not be written to for 10 ms after a byte is written, whereas static RAM bytes can be programmed as quickly as the access time will allow.

Here's two approximations of the total programming times for a 2Kx8 and a 8Kx8 EEPROM:

2048 bytes x 10x10E-6 sec = 20.48 sec
8192 bytes x 10x10E-6 sec = 81.92 sec
=1 min, 22 sec

This storage time constraint can be accounted for in either software or hardware, and is not considered a hardship since the EEPROM would probably be reprogrammed only infrequently. Some EEPROMs from SEEQ require that >FF (1111 1111) be written to each address prior to writing the actual data, thereby doubling the program time. Other chips from Intel allow a "page mode" where up to 16 consecutive bytes can be written per 10 ms storage time. Of course, the SEEQ chips are cheaper than the Intel versions.

The other constraint is cost. Fig. 3 shows the relative cost vs. function for common non-volatile memory chips. As of 1st quarter '87, 8kx8 EEPROM chips cost any where from $10 to $28 each, while 2kx8 chips run from $2 to $12. Both EPROM and battery backed static RAM are currently cheaper, but neither can offer the combination of in-circuit programming and portability.

SEND ORDERS TO: CAPTAIN'S WHEEL
17295 CHIPPENDALE
FARMINGTON, MN 55024
(612) 460-6348

SHIP TO: NAME _____

STREET ADDRESS _____

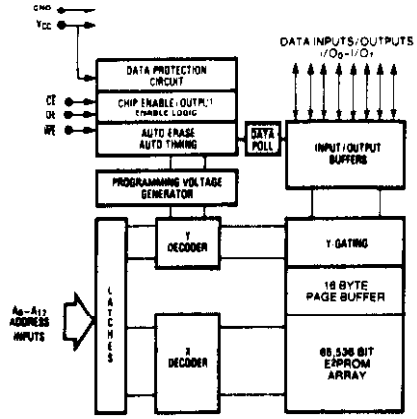
CITY/STATE/ZIP _____

PHONE NUMBER _____

##

ITEM	QTY	PRICE EACH	TOTAL
THREE SLOT EXPANSION KIT		35.00	
DISK DRIVE POWER OPTION		10.00	
32K MEMORY EXPANSION(Two or More)		39.00	
32K MEMORY EXPANSION	1	49.00	
LOAD INTERUPT SWITCH OPTION		10.00	
RESET SWITCH OPTION		10.00	
ADDITIONAL >2000->3FFF BANK OPTION		10.00	
ADDITIONAL >4000->5FFF BANK OPTION		10.00	
ADDITIONAL >6000->7FFF BANK OPTION		10.00	
ADDITIONAL >A000->BFFF BANK OPTION		10.00	
ADDITIONAL >C000->DFFF BANK OPTION		10.00	
ADDITIONAL >E000->FFFF BANK OPTION		10.00	
FILE UTILITIES LOADER SOFTWARE OPTION		10.00	
FILE UTILITIES LOADER SOFTWARE MODULE		15.00	
SHIPPING AND HANDLING			5.00
MN SALES TAX(ADD 6% FOR ANYWHERE IN MINNESOTA)			
C.O.D. (ADD $3.50 FOR C.O.D. ORDER)			
TOTAL ORDER			

NOTICE: If Two or more 32K memories are ordered with options you must state which options you want on each expansion unit idividually. ONLY THREE OPTIONAL BANKS PER UNIT MAX. (Optional software is considered as an optional bank.)



2864A Functional Block Diagram

Fig 2

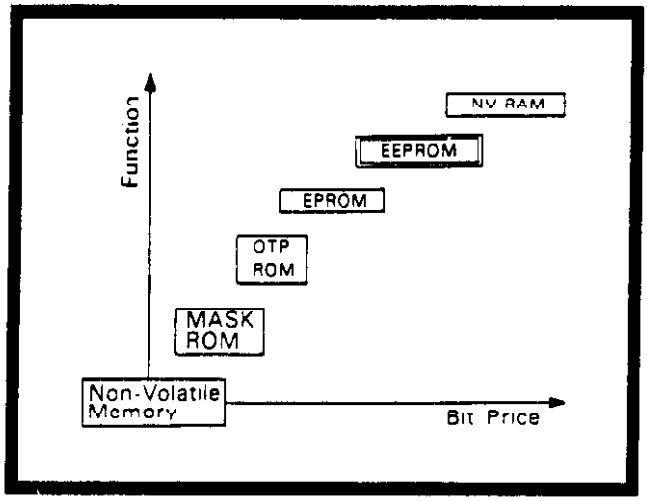


Fig 3
Relationship Between Function and Bit Price of Various Non-Volatile Memories

Comparison with other Non-volatile Memories

Below is a brief summary of the advantages and disadvantages of some non-volatile memory, as compared to EEPROMs:

ROM: advantages- retains memory indefinitely, less expensive in large runs; disadvantages- not user programmable, programmable only once.

PROM: advantages- retains memory indefinitely, user programmable, relatively cheap; disadvantages- programmable only once, requires high voltage/special timing for programming.

EPROM: advantages- user programmable, usually less expensive the EEPROMs; disadvantages- requires high voltage/special timing for programming, requires removal from circuit and UV light source for reprogramming.

Battery-Backed Static RAMs: advantages- acts as static RAM, cheaper per byte for storage if a large system if fed by a common lithium or NiCad battery, no timing or voltage constraints; disadvantages- non-portable, possibility of loss of data due to circuit/battery failure (lithium cell discharges or NiCad loses charge due to infrequent use).

Non-volatile RAMs (battery on board chip): advantages- same as above; disadvantages- cost considerably more than EEPROMs.

Possible applications

As you can see, each of the other common types of non-volatile memory have both advantages and disadvantages as compared to EEPROMs. So when would you consider using

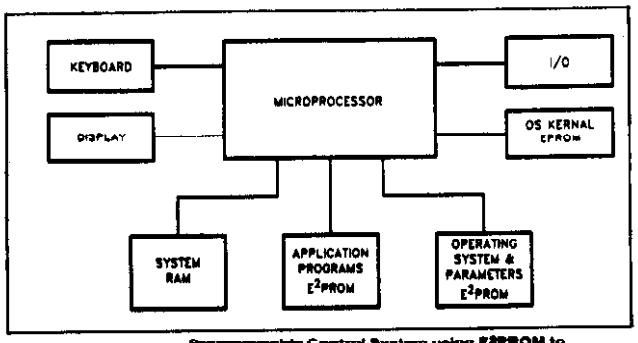
Fig 4

Feature \ Type	ROM	EEPROM	CMOS RAM	Advantage of EEPROM
Data Alteration		On board reprogrammability		System reconfigurability by software
				System reprogrammability by remote control
Data Retention	No need of battery			Free from battery maintenance

The Characteristics of EEPROM

an EEPROM? Probably when you want permanent storage of data/programs that would be reprogrammed on an infrequent basis (ie- you would expect to READ from the chip more often than you would WRITE to it), and the size of the information was relatively small (<64k). An excellent example of what you would not want to use EEPROMs for is a RAM disk. A RAM disk would see several read/write operations each time the system is accessed. While the EEPROM can be read as quickly as a 6264LP, it would take longer than a standard disk drive to store each 8k worth of data. The Horizon RAMDISK is a good example of use of battery-backed static RAM as non-volatile memory. While the Horizon unit can be purchased for less than $1 per Kbyte, an EEPROM unit of the same size would cost over $1.80 per Kbyte.

EEPROMs, as shown in Fig 5, are most useful in storing frequently used application programs/data or replacing operating system ROMs.



Programmable Control System using E²PROM to Store Operating System and Application Programs

Fig 5

Some specific applications might be:

*Replace ROMs: It is possible to swap the TI console and/or PEB card ROMs with EEPROMs that have been programmed differently. If they have a bug or two, pull them out, correct and reinstall. For these applications, the EEPROMs would be programmed by a separate programming circuit, since there are no WE (Write Enable) lines to ROMs to allow reprogramming in-circuit.

*Store Applications Programs: Frequently used utility or applications programs such as the Editor/Assembler, TI Writer, etc. could be stored in EEPROMs and made available upon power up without a disk access. If the program and/or data is revised, the new revision could be downloaded from disk, or even via the modem without disturbing the memory unit.

*Store DSR Programs @ >4000 and >6000: The 99/4A has an excellent Device Service Routine (DSR) system that allows polling of several memory chips for DSR routines, one at a time, at the >4000 address. Polling can occur at power up, interrupt, or while searching for a DSR routine or CALL subprogram. Several DSR locations were left undesignated by TI (see the TI Tech Manual) and potentially could be filled with 2k-8k chips with user defined routines. Likewise, modules like the Mini-Memory or SuperCart are candidates for EEPROMs at the >6000 location.

*Other Applications: Other applications are left up to the reader's imagination. Anytime you need a "permanent" memory device that is capable of being programmed (and reprogrammed in circuit) by the user, think of EEPROMs.

That's EEPROMs in a nutshell. A partial list of some of the companies that manufacture EEPROMs is given below. I am not an electrical engineer, so please forgive me if you spot a technical error or mis-statement. If you have questions, comments or criticism concerning this article, feel free to write to Ryte Data, or me directly:

Tony Lewis
409 Drolmond
Raleigh NC 27615

Or you can E-Mail me at CIS# 73357,1730. Last but not least, you can leave a message for me on the Bull City TIBBS, at (919) 383-8707; note that they support PC Pursuit. If reader response is favorable (and the space is available in the newsletter), an article on interfacing EEPROMs to the TI system may appear in a future issue.

EEPROM companies: SEEQ, Intel, Xicor, Hitachi, Exel.

FILE UTILITIES

John Gould of CAPTAINS WHEEL, 17295 Chippendale Ave., Farmington, MN 55024 has provided a demo copy of their "FILE UTILITIES SOFTWARE". The disk also includes an Extended Basic loader which catalogues and displays programs files. The loader will then load and run XB programs with a single keystroke.
This appears to be the only loader with sophisticated operation that does NOT require memory expansion.
The file utilities software performs some interesting functions to "bridge" the gap between cassette and disk based software. You can load and run normal object code files from disk or cassette. It also allows transfering files from one format to the other. All versions of TI loaders are supported except for some BASIC programs with CALL INIT, CALL LOAD from within the programs.

Captains Wheel also markets some interesting products for the 99/4A. See their ad this issue.

MAILINGS:

One of the small innovations you may have noticed is the postal machine stamp above your name that convinced the respective postal services to deliver this. Ahhh! No more hand pasting thousands of little pieces of paper... which was a REAL bottle neck. It was getting to the point where THAT factor ALONE was causing delays (beyond the OTHER factors). . Isn't technology wonderful? The only problem is that it is also expensive. The eternal trade off. Everything is relative in the Einsteinian universe.

SUBSCRIBERS:

We'd like to increase our size again during this next year. To do so, we need your input... articles, photos, drawings, ideas, comments, letters, news, programs, opinions etc.

In order to achieve this increase in size (and therefore an increase in associated costs), we are looking for new subscribers. Send us a new subscriber, send us the name of a likely 99er or have a friend write us for a free sample issue... you get the idea! An increase in our subscriber base will allow us to devote more pages to each issue, include more information and generally improve things even further.

Get em to mention YOUR name and address when they subscribe and we'll give YOU a disk of software to run, examine, use, copy or demo at your user group meeting! (There's some "neat stuff" on this disk).
As they say, in the party business, the more the merrier.

Please note that the date after your name is a "RENEW BY" subscription date. In other words, renew BY that time... please! WE are pleased to see that over 87% of the 1986 folks are still with us. Thankyou!

FINAL WORD: In examining the postal meter stamp, you probably noticed the cost of mailing this issue to you. Yes, our wonderful Canada Post has increased the costs AGAIN. Plans call for the size of R/D COMPUTING to increase to the allowable weight limit. In the mean time, our mailing costs have increased from the $.39 that each copy used to cost us before.

As a result we will have to raise the subscription cost to $16.00 US or $22.00 CDN as of May 30th. If you have a friend who would like to subscribe, please let them know of the cost increase. We will honour any subscriptions postmarked by May 30, 1987 at the current rates.

The supply of back issues has decreased over the past three months. For those of you who are interested in obtaining back issues, we would suggest that you place an order very soon. Once the existing stock is sold out we will not be reprinting back issues.
Plans do call for a "Best Of" publication covering hardware projects... but this will take place

AT LAST . . .

A TI99/4A keyboard interface with some "SMARTS".

Introducing the MLsystems Model KBM/99

In simple terms this interface enables the user to connect any IBM PC/XT or compatible keyboard to your TI99/4A computer.
Here are just a few of the features :

* FULL SIZE KEYBOARD LAYOUT
* SEPARATE NUMERIC / CURSOR KEYPAD
* JOYSTICK NOT AFFECTED BY ALPHA-LOCK
* LOAD INTERRUPT VIA THE KEYBOARD PrtSc key
* EASY HOOK UP, JUST THREE CONNECTIONS
* MANY T. I. WRITER FUNCTIONS NEED ONLY 1 KEY PRESS
* NO EXPANSION REQUIRED, CAN BE USED WITH A BARE BONES CONSOLE.
* FIFTY, COUNT EM' FIFTY, PRE-PROGRAMMED COMMANDS AND FUNCTIONS ALL INVOKED WITH A SINGLE KEYSTROKE :

"OLD DSK1."	"SAVE DSK1."	"PRINT "	"INPUT "
(ERASE)	"LIST "	"OPEN #"	"CLOSE #"
(BEGIN)	(PROCEED)	"CHR$("	"RETURN "
(AID)	(REDO)	"GOSUB "	"GOTO "
(BACK)	"RUN(cr)"	"ACCEPT AT("	"DISPLAY AT("
"CALL CHAR("	"CALL CLEAR "	"CALL COINC("	"CALL DELSPRITE("
"CALL COLOR("	"CALL GCHAR("	"CALL DISTANCE(#"	"CALL LOCATE(#"
"CALL HCHAR("	"CALL KEY("	"CALL MAGNIFY("	"CALL MOTION(#"
"CALL PEEK("	"CALL SCREEN("	"CALL PATTERN(#"	"CALL POSITION(#"
"CALL SOUND("	"CALL VCHAR("	"CALL SPRITE(#"	"CALL SAY("
"RUN "DSK1."	"MERGE DSK1."	"CALL INIT "	"PIO"
"CALL LOAD("	"RS232."""	"SIZE(cr)"	"DSK1."
"TRACE "	"UNTRACE "		

THE PRICE - VERY REASONABLE $80.00 - NOT INCLUDING THE KEYBOARD
(Compatible keyboards cost anywhere from $30.00 up)
FOR MORE INFORMATION OR TO ORDER CONTACT :

MLsystems
P. O. BOX 268
Valley Falls, Rhode Island 02864

* IBM & PC/XT ARE TRADMARKS OF INTERNATIONAL BUSINESS MACHINES

only after back issues are gone.

COMMAND DOS

by Monty Schmidt copyright 1987

OUT of the realm of the creative mind comes a new, very powerful program for your 99/4A computer SYSTEM. You'll say that this is "only logical", "why didn't someone think of this one years ago" or "Obvious!

COMMAND DOS is a very special 'memory resident' tool and "operating environment" for the TI 99/4A which gives you greater power and control over your system. It is a utility, a shell and an operating system all at the same time.

From 'DOS' you can perform a variety of tasks which normally require separate programs: load & run files, inspect and 'type' files to your screen or printer. With Command DOS you can FORMAT disks (in 40 or 80 tracks!), COPY files, DISKCOPY a whole disk, UNPROTECT or PROTECT a file, change a DISKNAME, ERASE a file and DIRectory out the entire disk.

Command DOS has a very special feature: with a file named AUTO-BAT you can create "BATCH" files which will automatically perform a number of tasks. Having this done under the control of the machine makes many jobs much easier. For example:

```
cls
rem Just press a key to start.
echo on
wait
more on
dir
rem Press CTRL S or CTRL Q to stop the
display.
vol dsk2.
dir
copy dsk2.DOS dsk1.DOS
beep
rem Place new disk in Drive #2.
copy dsk1.AUTO-BAT dsk2.AUTO-BAT
honk
```

So on and so forth. This unique program provides thirtyfour (34) new commands and six utilities. Command DOS gives you most of the features found in such systems as MS-DOS... running on the TI 99/4A. The program even has an EDIT40 command to run TI Writer and then return to DOS. This makes the program unique, functional and more than useful.

BATCH	HONK	RENAME	CMPDSK
BEEP	INIT	SETPRINT	DISKCOPY
CLS	LINK	TYPE	EDIT40
COPY	LOAD	UNPROTECT	FORMAT
DEL	MORE	VER	
DIR	ONKEY	VOL	
DISKNAME	OUTPUT	WAIT	
ECHO	PRINT	WIDTH	
ERASE	PROTECT	X	
FIX80	Q(uit)		
SEE	REF	AUTO-BAT	
HELP	REM	CHKDSK	

What's more, this program does not use up any of the standard 32k memory. Command DOS is written to occupy the memory space from >6000 to >7FFF. It requires the use of a GRAMKRACKER, MAXIMEM, GRAM CARD, SUPERSPACE, 8k EA MODULE, CACHE CARD or similar device.

The first version is being sold on disk for loading into one of these hardware devices. Package includes two disks, documentation and examples of how to use Command DOS. Available immediately for $19.95 US plus $2.00 shipping or $25.00 CDN funds plus $2.00 shipping.

With sufficient interest from the TI community, we would also put Command DOS on a module format for owners without any of the above devices. If you would like to see this, please write to indicate your interest!

8K DSR RAM CARD PROJECT

PART TWO: 8K Non-DSR RAM

by John Clulow

This article is a continuation of the project presented in the March issue. These instructions describe how to add 8K of battery backed RAM to the 8K DSR CARD. This RAM is selected only whenever any other PEB card is NOT selected. The idea for this project was developed by John Johnson for the 8k DSR RAM CARD.

It is a good idea to remove the battery from the 8K CARD before beginning this project.

You will need the following parts:
(1) 74LS259
(1) 6264LP-15
(10) 1N914 diodes
(3) 2K ohm resistors 1/4 watt
wire wrap wire (a different color)

1] Make sure your 8K DSR RAM CARD is working properly. Correction of any problems in the basic card will be made much more difficult after the 8K is added.

2] Remove U5 - the 74LS259.

3] Piggyback a new 74LS259 on top of the old one as follows:

A) Bend out pins 2 - 7 and 9 - 12 on the new 74LS259.

B) Place the new 74LS259 on top of the old 74LS259 making sure that pins 1, 8, and 13 - 16 make contact.

C) Apply a small amount of solder to connect pins 1, 8 and 13 - 16.

D) Remove any flux which may be deposited on the pins of the old 74LS259.

E) Replace the piggybacked U5 in the socket (you used sockets?!!)

4] Remove U7 the 6264LP-15. Minimize the potential for static discharge. Piggyback the new 6264LP-15 on top of the old one as follows:

A) Bend out pin 26 of the new 6264LP-15.

B) Place the new 6264LP-15 on top of the old one making sure the notches designating pin 1 line up and that all pins with the exception of pin 26 are in contact.

C) Apply a small amount of solder to connect all pins, except pin 26.

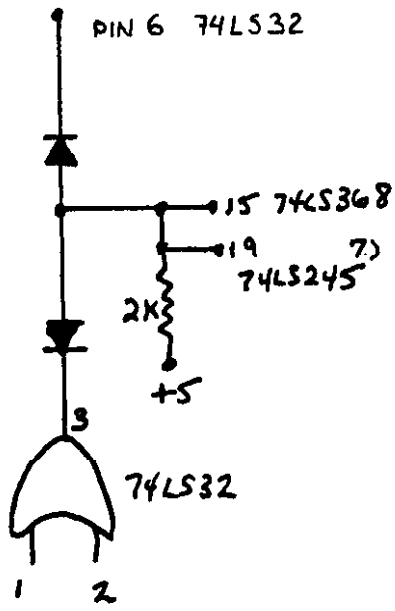
D) Remove any flux from the lower IC and replace the IC pair in the socket.

5] Remove the wire(s) connecting pin 3 of the 74LS32 and pin 15 of the 74LS368 and pin 19 of the 74LS245. (The connection between pin 15 of the 74LS368 and pin 19 of the 74LS245 may remain intact.)

6] Use two 1N914 diodes and a 2K ohm resistor to make an AND gate as follows. Make sure you test the diodes before using them:

A) Orient the diodes on the board with the cathode ends pointing away from each other. Solder the anode leads together.

B) To the anode juction of the diodes, connect one end of the 2K ohm resistor. Connect the other end of the resistor to +5 volts.



NOTE:
SEE PAGE 19 FOR 74LS32 DETAILS.

7] Connect this AND gate to the circuit as follows:

A) One of the diode cathodes connects to pin 3 of the 74LS32.

B) The other diode cathode connects to pin 6 of the 74LS32.

C) The anode connection of the diodes, in addition to being connected to one end of the 2K resistor, also goes to pin 15 of the 74LS368 and to pin 19 of the 74LS245. (The 368 - 245 part of the circuit should have remained intact.)

8] Connect 74LS32 pins 4 and 1.

9] Connect pin 5 (bent out) of the NEW 74LS259 to pin 5 of the 74LS32. This should be done by inserting a wire from the bottom of the board through the hole adjacent to pin 5 of the new 259.

10] Connect pin 5 of the 74LS32 to pin 6 of the 74LS368.

11] Disconnect the wire connecting pins 2 and 14 of the 74LS368.

12] Connect 74LS368 pin 2 to pin 9 of the 74LS368.

13] Connect 74LS368 pin 7 to 74LS32 pin 10.

14] Connect 74LS32 pin 8 to pin 14 of the 74LS368.

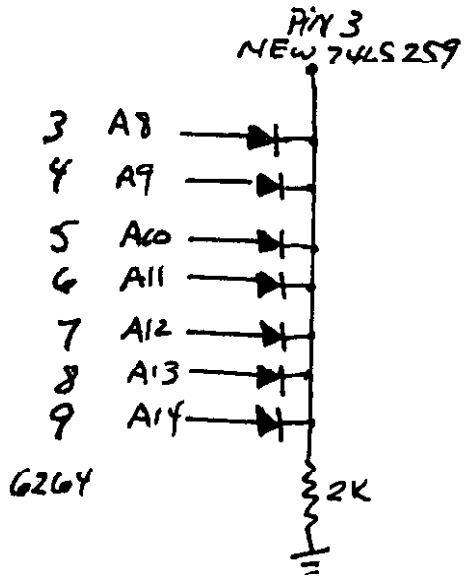
15] Connect pin 2 (bent out) of the NEW 74LS259 to GROUND.

16] Construct a 7-input OR gate from seven 1N914 diodes and a 2K ohm resistor as follows. Make sure you test all diodes before using them:

A) Line up the diodes on the board with the cathode ends facing the same direction.

B) Solder the leads of the cathode ends together.

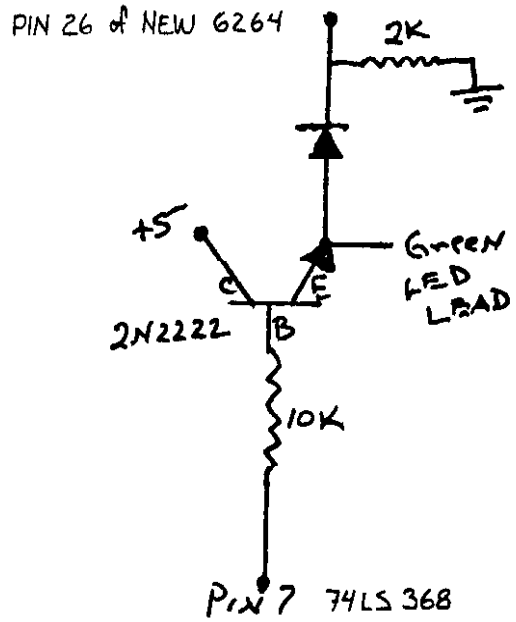
C) Connect one end of a 2K resistor to the cathode ends of the diodes and



connect the other end of the resistor to GROUND.

17] Connect the OR gate to the circuit as follows:

A) Connect the anode ends of the diodes to address lines A8 - A14. One convenient place to connect these lines is on 6264LP-15 pins 3 - 9. One address line should be connected to each diode anode.



B) Connect the cathode junction of the diodes to pin 3 (bent out) of the NEW 74LS259.

18] Add a 2N222 transistor, 2k

resistor, 10k resistor and 1N914 diode (check it first) to the circuit as follows:

A) Solder the leads of the 2N222 transistor to the solder pads on the board. Allow the body of the transistor to stand about 1/4" above the board.

B) Connect the base to one end of the 10k ohm resistor. Connect the other end of the resistor to pin 7 of the 74LS368.

C) Connect the collector of the 2N222 to +5 volts.

D) Connect the emitter to the GREEN lead of the bi-color LED. (The RED lead should be in use by the 8K DSR RAM.)

E) Also connect the emitter to the anode side of a 1N914 diode. Connect the cathode end of the diode to pin 26 (bent out) of the NEW 6264LP-15.

F) Pin 26 of the new 6264 IC and the cathode end of the diode also connect to one end of a 2K resistor. The other end of the resistor connects to GROUND.

19] Replace the battery.

This completes the project. You should observe the GREEN light to be on whenever the system is powered up and a PEB card (including the *K DSR RAM CARD) is NOT being accessed. The GREEN light should go out whenever another card is accessed.

You can use the MiniMemory or DEBUG to try writing to >4000 - >5FFF when the GREEN light is on (no CRU selected). You can also select various CRU base addresses to make sure the two rams on the card switch on and off at the appropriate times.

MAKE SURE THAT YOU REMEMBER TO TURN OF ONE CRU BASE BEFORE TURNING ON A NEW ONE!!! John Clulow April 6, 1987

EDITORIAL: The uses for this card are very interesting. With both the DSR and the 8K RAM installed, it is now possible to have continuous memory in the TI 99/4A system available from hex >2000 to >FFFF (using the 8K RAM in >6000 - >7FFF). With careful use, a variety of different programming applications can be achieved which provide more powerful applications with the machine.

This type of switching memory in and out of mapped locations is used by several other machines to achieve larger addressable memory specifications. MS-DOS will support up to 640k using this 'type' of bank switching technique. Each block of memory is switched from >0 to >F so that high memory, for example, would be designated >FFFFF.

The unfortunate aspect of the 99/4A is that no provisions were allowed in the operating system to provide for control of a similar technique with the memory map.

For many TI owners, the machine does perform all the tasks ever required. Sure, there are some limitations, but for the uses involved nothing "more" is needed. On the other hand, an expanded memory specification would allow more sophisticated software to be developed. John Johnsons MENU program is one example. As with many software products, you really do have to see it in operation to appreciate how powerful it is and what it does for the computer.

Even if you have never tackled a circuit project, I would strongly suggest that you invest a small amount of money and time to build this one. Once you are finished, send John Clulow an initialized disk, return mailer and $3.00 (or more!) to receive the loader, menu and clock software.

A parts kit can be obtained from Bud Mills Services, 166 Dartmouth Drive, Toledo, OH 43614. Send Bud $40.00 for the basic kit. The additional 8K RAM section may cost more.

TIGERCUB SOFTWARE

Over 130 original entertainment, education and programming utility programs in Basic and Extended Basic on cassette or disk. Only $3.00 each! Eighteen different full-disk collections, just $12.00 each! Descriptive catalogue $1.00 refundable. TIPS FROM TIGERCUB full-disk collections of 50+ programs and files from Tigercub Tips newsletters, Vol. I, II and III $15.00 each; any two $27.00; all three $35.00 postpaid. NUTS & BOLTS (#1) and (#2) full disks of 100+ utility subroutines in XBasic 'merge' format, ready to merge into your own programs: $19.95 each, both for $37.00 with documentation, postpaid. Orders to Tigercub Software, 156 Collingwood Ave., Columbus, OH 43213

SMART MODEMS FOR ONLY:
$39.00

Completely TI compatible. Add $6 if using CorComp RS232. Modems are auto dial/answer, come with RS232 cable, manual and telephone cable. Power supply is NOT included. Power requirements are fully documented. Power supplies may be available for a short period of time.
Write to: Smart Modems
14436 FENTON
REDFORD, MI 48239

Send US funds for foreign equivilent. Make check or money order payable to: Scott A Sorel. If a power supply is desired add $10. Money will be refunded if quantities are sold out upon arrival of order.

WANTED: TI DISK DRIVE CONTROLLER CARDS Have you or your friends upgraded to a Myarc, CorComp or Atronic DS/DD Disk Drive Controller Card? We would like to purchase your old TI controller card. We only require the card - and the Disk Manager II module if you still have it. No cables, stand alone units or power supplies. Contact Ryte Data (705) 457-2774 9-5 M-F.

YOUNGTREE

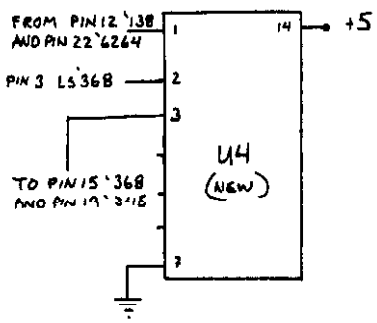
TI Keyboard $5.95
Cassette Cable $2.95
Oldies But Goodies $3.95
Beginners Basic Tutor $3.95
Paint N Print $24.95
Add $3.00 shipping handling
M/C VISA call 1-216-399-5463
YOUNGTREE
PO BOX 197
LEAVITTSBURG, OH 44430
Catalog sent with order or send $2.00 (refunded with 1st order)

FOR SALE: COMPLETE SYSTEM.

TI 99/4A Console, Peripheral Expansion Box, 1 Disk Drive w/ controller, 32k Memory Card, RS-232 Card, Joysticks, TI Writer, Extended Basic, Speech Synthesizer, Parsec, TI Invaders etc.)

Contact: C.P. Floyd
26 Chant Cresc.
Unionville, Ontario L3R 1Y8
(416) 477-3684

74LS32



NOTE: The diodes in the upper section of the diagram MARCH V.21 have been replaced with an LS32 chip as shown. Refer to drawing and associated text for connections. It is recommended to place the 74LS32 in the upper right - designated as U4. Note the IC's by name rather than U number.

SUBSCRIPTION COST:

$14.00 (US) for 12 issues $20.00 Cdn. all via First Class Mail.
$19.00 (US) overseas Airmail Delivery.

BACK ISSUES:

Back issues are available for $2.00 each, **subscribers** only for these issues —

V 2 sold out.
V 3 TMS 9995 Memory Map & specs numeric keypad project.
V 4 XB II **plus** by Mechatronic, Myarc 128k card, Autofire joystick project.
V 5 32k internal memory project, DS/DD Ramdisk, Auto power-up project.
V 6 Myarc 256k computer, "C" compiler, RAM/GRAM card.
V 7 Maximem review, EEROM programmer, Sense and control card.
V 8 Sold out.
V 9 Dual disk controller project, RESET switch project.
V 10-11 Double issue: LA TI FAIR, Bill Gronos on Assembly, Super load switch, EPROM programmer.
V12.5 Myarc 640k Geneve, Console speed upgrade, Video monitor filter, Gramkracker review, ATRONIC products.
V14.6 Expansion Box 8k 'module,' Super Clock Support, Basic Compiler, 9938 Video Chip.
V15 Video upgrade @ 15mHz, Quad. Density TI Disk Controller, 80 Column Display preview.
V16 P Box Modification, Multi-Module project.
V17 Tigercub, Eprommer update, Quad density update.
V18 Power Supply, RAVE 99 Keyboard, Year Review.
V19 Geneve, c993.0, RAM Disk backup, Hi-Speed cassette.
V20 99AT, T.O.D., Horizon modification, Rapid Copy 32k project, GRAM card software, etc.

V21 Expansion system, 8k DSR RAM Project, C99 by Ron Albright, TI Writer in Memory, AEI etc.

R/D COMPUTING is published monthly by Ryte Data in Haliburton, Ontario.

All material is from sources believed to be accurate. The publisher takes no responsibility for errors, omissions or misprints. Articles may be reprinted with credits giving source and address by users groups for publication.

Articles dealing with Texas Instruments 99/4A and 9900 based computers are published. Special attention is given to projects and information on upgrading and modifying the 99/4A console and system.

Please send any information on products, hardware, software, or modifications to our address. We cannot accept responsibility for materials submitted and, unless stated otherwise, will assign all manuscripts etc. for publication.

Only manuscripts with sufficient postage and self-addressed mailer will be returned.

Un-classified ad rates are $1.25 per 40 character line. Count ALL characters, spaces, etc. and submit with payment to our address. Ads are run in order received. Deadline is 21st of each month.

YOU MUST NOTIFY US OF YOUR CHANGE OF ADDRESS SIX WEEKS IN ADVANCE. PLEASE SEND A CHANGE OF ADDRESS CARD WITH YOUR OLD AND NEW ADDRESS.

Ryte Data
210 MOUNTAIN STREET
HALIBURTON, ONTARIO K0M 1S0
705 457-2774