VOLUME 1 ISSUE 11 $1.50 JULY 1985

SUPER 99 MONTHLY

MYARC NEWS ON

PAGE 9

SOFTWARE

Review: COMPANION

STANDARD: 1A 2XB 4B 5A 6B 7A 9A 10A

Company: INTELPRO

Suggested Retail Price: $79.95

Overall Rating: 95 of 100

There are only a few word processors for the 99/4A that really work efficiently and COMPANION is definitely one of the best.

There are several strong points to the COMPANION system. In our opinion, COMPANION is the fastest word processor for the 99/4A and possibly for any microcomputer. Particularly impressive is the fact that all of the COMPANION Assembly Language routines are loaded in memory together -- the only subsequent disk access relates to document files, making COMPANION very efficient for processing large volumes of documents in a single session.

COMPANION is also extremely easy to use. Making wise use of a computer keyboard and the display screen are important features in developing an easy to use word processor. With COMPANION, nearly every command relates well to the corresponding key sequence and screen display. For instance, pressing <CNTRL> <L> yields a boldface down arrow on the screen, the symbol for a Linefeed ("L" is for Linefeed). Similarly, <CTRL> <P> displays a boldface reverse P to denote Paragraph (with automatic indentation).

COMPANION is also much easier to use than most word processors for sending printer commands and block graphics. Rather than using some meaningless code or redefining one or more characters, COMPANION uses <CTRL> <A> to SEND ASCII, which appears on the screen as a boldface A. SEND ASCII is used for non-printed instructions to the printer and is followed by the ASCII number that is to be sent, such as 7 to ring the bell on the printer. For printed characters, such as block graphics, a SEND GRAPHICS command, <CTRL> <G> (again, easy to remember), displays a boldface slash (/) and is followed by the ASCII number for the desired graphics character, which is usually 130 to 255 for most printers.

There is an abundance of other noteworthy features in COMPANION. One

-->

of the best is 8 sets of 100 horizontal tabs for a total of 800 tab positions! Also, a powerful set of Edit functions is available that allow quick string searches, manipulation of blocks of text, counts of strings, a count of the total characters (18000 allowed per file, with linkages between files accomodated) and much more.

COMPANION is very user-friendly and is next to impossible to crash. Selecting a menu option often evokes a prompt questioning whether the user is certain that the selection is desired, as some operations can permanently alter or eliminate existing documents.

Controversy surrounds COMPANION's use of a 40 column screen that does not scroll horizontally and does not appear as the printed page will appear. We have found that most of the critics of this system have not tried it and the problems are more imagined than real. INTELPRO refers to the system as a conceptual approach as opposed to a secretarial approach to word processing. Certainly, the keystroke savings of the system result in much faster word processing than is possible with most systems. After only a few sessions, users can grasp the conceptual approach and many users voice strong approval of the ease of use of the COMPANION method.

Certainly, no word processing system can be all things to all users and COMPANION is no exception. There are two significant drawbacks to COMPANION, but some users will not be concerned with either (and there is no rule that says a user must use a single word processor to the exclusion of all others).

The files generated by COMPANION are in DISPLAY/VARIABLE 255 format, which is not consistent with the file formats of the major modules from TI, such as the Editor/Assembler and Terminal Emulator II (DIS/80). Therefore, files from other sources are difficult to load into COMPANION. Also, it is not easy to generate files that could be used by other modules.

The second major drawback to COMPANION is that it does not allow right justification of documents (a term that means that the right margin lines up evenly, a simulation of typeset print). Many users do not care for this feature and are not at all disturbed to be without it.

We were very impressed with COMPANION. It is fast, easy to use, well-documented and packed with useful features. Although the price is higher than most of the current software for the 99/4A, COMPANION is priced comparably to word processors for other computers. Even if you already have a word processor that you think is a good one, COMPANION is worthy of your consideration!

For more information on COMPANION, write to INTELPRO, 5825 Baillargeon St., Brossard, Quebec, Canada J4Z 1T1.

HARDWARE

Expansion Box Expansion

by S. J. Szymkiewicz, MD

STANDARD: 1A 2MM (o) 5A 9A

The TI 99/4A has great potential as a hardware hacker's machine. The ability of the 9900 CPU to act as a controller simplifies the addition of new peripheral devices to the machine, even those unplanned for by TI. Also, the expansion box itself eliminates some of the problems encountered when hacking on small computers, such as building power supplies and providing a bus for multiple devices. In this article, using the control capabilities of the 9900 and meeting the hardware and software requirements of the expansion box will be explored.

Before jumping into a hardware project such as adding a new peripheral, there are two sources of information to be aware of. The TI-99/4A Technical Manual has schematics, timing diagrams, and explanations of

-->

how the console and expansion box function, including specifics of peripheral hardware and software interfacing. The Editor/Assembler Manual or another manual of 99/4A assembly programming is indispensable for writing assembly routines to use new peripherals. Both are needed for an understanding of interfacing with the 99/4A.

Understanding the bus of the expansion box is fairly straight-forward, although a few items need clarification. The pinout of the card-edge connectors is given in figure 1. Note that the even numbers are on one side and the odd numbers on the other side of the 60 pin, .100" spaced connectors. The bus allows access to the CRU unit, the memory control, address and data lines, some interrupt signals, and the expansion box power supply.

The 9900 controller ability stems from its CRU (communications register unit), a fairly unique feature of 9900 architecture. The CRU interface provides a separate I/O (input/output) port for single to sixteen bit data transfers without using memory space and the data bus. The SBO, SBZ, TB, LDCR, and STCR assembly commands are used to access the CRU. The only hardware needed is a simple combination of an address decoder (such as a 74LS138 3-to-8 decoder) and a bit addressable latch (such as a 74LS259). The CRU clock line goes true when CRUOUT is valid as the CRU output bit; use it to clock the latch onto the data. CRU input uses system clock phase 4 to indicate a valid bit on CRUIN. The TI Technical Manual has complete timing diagrams for these operations. Note that the line for address bit 15 doubles as CRUOUT while a separate line is provided for CRUIN.

The 99/4A uses the CRU for interfacing to peripheral devices. Each device has its DSR (device service routines, the assembly routines needed to use the peripheral) in the >4000 to >5FFF area of the memory map. The computer separates the DSRs by paging them in and out of CPU memory using the CRU's I/O. Consequentially, the CRU bit space is mapped into blocks, as illustrated in figure 2. The blocks not assigned for a particular peripheral are open for use. Each block has 128 input and 128 output bits to be used as needed, except for the base address output bit (the first bit on the CRU block) which is used to page the DSR in and out of CPU memory.

The address lines of the bus number from 0 to 15 with 0 as the most significant bit according to TI's convention (most other chip manufacturers use 0 as the least significant bit, so be careful to avoid confusion). These lines are driven by the CPU. The bus also has address lines A, B, and C. These high order bits are not connected to the CPU. The peripheral device should decode these lines during memory accesses, however, and assume the lines are high. Obviously, TI had more planned than the 32K expansion! These extra address lines could expand the 32K to 256K. The lines could be driven by the CRU of the 9900 or by a second CPU in the expansion box (judging from other lines available to the bus, a second CPU in the box was planned as a future upgrade for the 99/4A).

The data lines are numbered 0 to 7, again 0 is considered the most significant bit. Before these lines are available to both the CPU and the peripheral device, a buffer in the flex cable must be switched off using the RDBENA* line (remote data bus enable, active low). An open-collector signal must be used. The three conditions for activation of the RDBENA* from a peripheral device are: it must be a memory operation (MEM* is low), it must address the device's memory area, and the DSR enable bit must be high. A 2-input NAND, the 74LS03, provides an open-collector signal, or an open-collector inverter (74LS05) can be used to drive the line.

In addition to the buffer in the flex cable, a buffer for the data -->

lines should be provided between the expansion box bus and the device. An 8-bit transceiver, the 74LS245, works well for this. The RDBENA* signal can be used to activate the buffer and the WRITE* signal used to determine the direction of data transfer. In addition to this buffer , lines used by multiple chips (such as address lines) should have drivers (the 81LS45 8-bit line driver is an example) close to the bus.

The bus of the expansion box also contains access to the power supply. Lines 1 and 2 supply an unregulated +8 volts. Regulation must be provided by the peripheral card; a 7805 or LM340-5 voltage regulator can be mounted directly on the card to complete a +5 volt regulated supply. The maximum load should be less than 500 mAmps per card. Lines 57 and 58 supply an unregulated -15 volts to a maximum load of 30 mAmps. Lines 59 and 60 supply an unregulated +15 volts to a maximum load of 250 mAmps. Again, regulation should be added to use these as -12 and +12 volt supplies.

Once the hardware requirements are understood, the next step is getting the 99/4A to use the peripheral device. In order for the device to be used under TI BASIC, DSR software must be written. When first turned on, the 99/4A searches for attached devices with DSRs by sequentially turning on the base CRU output bit of each CRU block to page in the DSR memory space. If a DSR identification is found, the device name is noted and any power-up routine performed as needed. Then the 99/4A continues its search. Once the name is noted, it can be used in file I/O under BASIC file commands. The DSR must support the proper error codes, of course. Provisions are made for multiple power-up routines, multiple main device routines, and multiple interrupt routines.

The DSRs the computer already has are easily explored. Simply insert the Mini-Memory cartridge and use the Easy Bug routines to set the desired CRU output bit with the Cxxxx command (xxxx is the hexidecimal address). Then the DSR can be viewed by the Mxxxx command or disassembled if a disassembler has been loaded (start it via the Exxxx command). Details of DSR organization and linkage to routines are given in the TI Technical Manual.

In order to write a proper DSR, one must be familiar with the file management of the 99/4A. The 99/4A uses PABs (peripheral access blocks) in handling data transfer. PABs are areas of VDP memory set aside to identify the location and organization of blocks of data. All the DSR must do is know where to find or where to put the data. PAB structure is explained in both the E/A Manual and the TI Technical Manual.

The expansion box is an opportunity for those who wish to begin to experiment with hardware additions. The set-up of the 99/4A with its CRU controller ability and the expansion box provisions of a bus and power supply offers a lot of potential. Once the CRU, the bus, DSRs and PABs are understood, hardware hacking becomes a simpler matter.

Figure 2

CRU Map

>1000	>10FE	Production line testing
>1100	>11FE	Disk Controller
>1200	>12FE	Modem
>1300	>13FE	First RS232 unit
>1400	>14FE	Not used
>1500	>15FE	Second RS232 unit
>1600	>16FE	Not used
>1700	>17FE	HEX-BUS™ Interface
>1800	>18FE	Thermal Printer
>1900	>19FE	EPROM Programmer
>1A00	>1AFE	Not used
>1B00	>1BFE	Not used
>1C00	>1CFE	Video Controller
>1D00	>1DFE	IEEE 488 Controller
>1E00	>1EFE	Not used
>1F00	>1FFE	P-Code Controller

Only even addresses are used as address line 15 is used as CRUOUT. Obviously, some devices planned for by TI never made it to production.

-->

Figure 1

Expansion box bus pinout

1 --- +8 volts, unregulated
3 --- logic ground
99/4A)
5 --- logic ground
7 --- logic ground
9 --- CPU indicator (1 for 99/4A)
11 --- RDBENA*
13 --- CPU hold request, active low
not supported by the 99/4A
15 --- Interrupt level A enable
17 --- Interrupt level A, active low
19 --- data bit 7
21 --- data bit 5
23 --- data bit 3
25 --- data bit 1
27 --- logic ground
29 --- address bit 14
31 --- address bit 12
33 --- address bit 10
35 --- address bit 8
37 --- address bit 6
39 --- address bit 4
41 --- address bit 2
43 --- address bit 0
45 --- address bit B
47 --- logic ground
49 --- logic ground
51 --- CRU output clock, active low
53 --- logic ground
55 --- CRUIN
57 --- -15 volts, unregulated
59 --- +15 volts, unregulated

2 --- +8 volts, unregulated
4 --- system READY (not used by
6 --- RESET* (system reset, active
low)
8 --- System Clock (not used by
99/4A)
10 --- Input audio
12 --- enable for circuit board
burn-in
14 --- IAQHA - interrupt or hold
acknowledge. Not supported by
99/4A
16 --- Interrupt level B enable not
supported by the 99/4A
18 --- LOAD* (not connected to 99/4A)
20 --- logic ground
22 --- data bit 6
24 --- data bit 4
26 --- data bit 2
28 --- data bit 0
30 --- address bit 15 / CRUOUT
32 --- address bit 13
34 --- address bit 11
36 --- address bit 9
38 --- address bit 7
40 --- address bit 5
42 --- address bit 3
44 --- address bit 1
46 --- address bit A
48 --- address bit C
50 --- CPU clock, active low, phase 3
52 --- data bus direction - high if
read
54 --- CPU write enable, active low
56 --- MEM* (memory request, active
low)
58 --- -15 volts, unregulated
60 --- +15 volts, unregulated

TI-WRITER

Word Processor Dump: Part 3, Using the Gemini 10

STANDARD: 1A 2TW XB 3B 4B 5A 7A 9A
10A

Thanks to Joseph E. Bartle of Parish, New York, this month we have a modification for users with a Gemini 10 printer (not the more common 10X). We hope we get this listing right, as it came to us in a message as an ASCII transfer by modem. In line 25160, the RPT$ may be wrong. If you have any problems with it, please let us know!

Change or add only the lines shown as follows:

```
25065 TN=256+(T*8):: N2=INT(
TN/256):: N1=TN(N2*256):: !
MAY NEED TO ADD N2=N2*DE, NO
T TESTED FOR DENSITY.
25070 PRINT #1:".TL 1:27,65,
8,13,27,"&SEG$("7576",DE*2-1
,2)&","&STR$(N1)&","&STR$(N2
)
25160 TA$=RPT$(CHR$(2),T)::
FOR I=BR TO ER :: A$=CHR$(1)
&TA$ :: FOR J=1 TO 32
```

Mr. Bartle also advises that the Gemini 10 will center with a tab of

-->

14. Incidentally, some of you may be familiar with Joe's programs, as he wrote a popular program called "Jacket".

Back to the Epson compatibles, the error check at line 25020 was incorrect and should be as follows:

```
25020 IF (T<0)+(T>(80-(MIN(D
E,1)*44)))+(BR>ER)+(BR<1)+(B
R>24)+(ER<1)+(ER>24)THEN GOS
UB 25220
```

SPEECH

Child Talk

STANDARD: 1A 2TE 9A 12A

Though the program that follows is short and simple, we have found it to be very effective for introducing a very young child (as young as two years old) to computers. All the child has to do is press a key or, better yet, the joystick fire button.

The program can be very easily modified. Of course, you will want to insert your child's name in line 120 and the names of your family members in lines 320 and 330. To add more lists, add a line at 151 for the menu, increment the ASCII verification at line 170, add a branch at line 190, add data statements at 351 (technically, anywhere can be made to work), and add a RESTORE and RETURN at the end.

To allow the child to continue for as long as desired, the program has no end, so you must press <FCTN> <4> to end the program. Also, be sure to SAVE the program as soon as it is keyed in, as a young child will likely not take long to find the <FCTN> <QUIT> keys and wipe out the memory! That and avoiding severe damage to your keyboard is why the joystick is highly preferred! As the child grows more accustomed to being around the computer, switching to the keyboard will usually become easier, or at least one would hope so. Ha!

Lines 210 to 240 will vary the pitch of the computer's voice slightly, eliminating the monotone that quickly becomes boring. See page 34 of the TE2 manual for a discussion of pitch and slope.

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 PRINT #1:"HELLO TODD"
130 PRINT "1) LOVE LIST"
140 PRINT "2) NUMBERS"
150 PRINT "3) TOOLS"
160 CALL KEY(5,CHO,S)
170 IF (S<1)+(CHO<48)+(CHO>5
1)THEN 160
180 PRINT "OK"
190 ON CHO-48 GOSUB 360,380,
400
200 FOR I=1 TO 8
210 RANDOMIZE
220 SLOPE=INT(7*RND)+39
230 PITCH=32*(SLOPE*.1)
240 PRINT #1:"//"&STR$(SLOPE
)&" "&STR$(PITCH)
250 READ A$
260 CALL KEY(1,K,S)
270 CALL KEY(2,K,ST)
280 IF (S<1)*(ST<1)THEN 260
290 PRINT #1:A$
300 NEXT I
310 GOTO 190
320 DATA DADDY LUVS U,MOMA L
UVS U,PAW PAW LUVS U,MAW MAW
 LUVS U,CHEEK OH LUVS U,ANT
JINNY LUVS U,UNKEL AL LUVS U
330 DATA BABY BEAR LUVS U
340 DATA 1,2,3,4,5,6,7,8
350 DATA PLY ERS,HAMMER,SCRE
W DRIVER,RAKE,SHOVEL,TAPE,HO
E,LEVEL
360 RESTORE 320
370 RETURN
380 RESTORE 340
390 RETURN
400 RESTORE 350
410 RETURN
```

ASSEMBLY

Two Neat Load Interrupts

STANDARD: 1A 2TE (o) TX (o) EA 3B 4B 5A 6B 7A 9A 13B

Paul Charlton, the author of the —>

great terminal emulator program "FAST-TERM" ($10 FREEWARE, address is 1110 Pinehurst Ct., Charlottesville, VA 22901), gave us permission to print two routines he recently completed. The routines require what is known as a load interrupt switch (If you don't have one, we'll explain a good way to install one next month. Such a switch also comes in handy with Danny Michael's FREEWARE "Dump".). If you find any other way to implement the routines, please let us know as a load interrupt switch is not 100% reliable. Please note that we have not tested either routine, so let us know if you have problems.

The first routine will set the TE2 module into 1200 baud. Paul cautions that this one is particularly difficult to implement with the switch. Here are the steps to follow:

1. Assemble the source code listing below. Of course, this will only have to be done the first time.

2. Load the program from the E/A Load and Run option 3.

3. Without turning the Peripheral Expansion Box off, switch modules, changing from E/A to TE2.

4. Go into the TE2 as you normally would, ready for communicating, the cursor in the upper left hand corner of the screen.

5. Press the load interrupt switch.

Here is the source listing for the TE2 routine:

```
*
* LOAD INTERRUPT ROUTINE TO PUT
* TE2 CART INTO 1200 BAUD
*
REGS    BSS  32
ENTER   CLR  @LOADWP
        LWPI REGS
        CLR  R0
E1      DEC  R0
        JNE  E1
        LIMI 0
        LI   R12,>1340
        SBO  31
        LDCR @CNTRL,8
        LDCR @INTVL,8
        LDCR @RDR,11
        LDCR @XDR,12
        SBO  18
E2      STWP R0
        MOV  R0,@LOADWP
        RTWP
CNTRL   BYTE >83
INTVL   BYTE 1600/64
RDR
XDR     DATA >1A1
*
        AORG >FFFC
LOADWP  DATA REGS
        DATA ENTER
        END
```

The next routine is for use with the Tax Investment Record Keeping module. It allows use of a parallel ("PIO") type printer or, by changing the PAB in the listing, any valid printer name. Follow the steps as above, except you switch to the Tax module and proceed to the prompt for "PRINTER (Y/N)". Answer yes, then type RS232 or TP. Hit the load interrupt switch and you're ready.

```
*
* LOAD INTERRUPT ROUTINE TO PUT
* A NEW PRINT FILE IN TAX INV. RECORD
* KEEPING
*
REGS    BSS  32
NAMLEN  BYTE 3       ! CHANGE FOR
*                    ! ANOTHER PRINT
NAME    TEXT 'PIO'   ! "            "
        EVEN
ENTER   CLR  @LOADWP
        LWPI REGS
        CLR  R0
E1      DEC  R0
        JNE  E1
*
        LIMI 0
        LI   R0,>0589
        ORI  R0,>4000 ! FOR WRITE TO
*                     ! VDP
        SWPB R0
        MOVB R0,@>8C02
        SWPB R0
        MOVB R0,@>8C02
        LI   R1,NAMLEN
        MOVB *R1,R0
        SRL  R0,8
```

-->

```
LOOP1   MOVB *R1+,@>8C00
        DEC  R0
        JOC  LOOP1      ! LEN + 1 # BYTES
*                       ! TO REPEAT
        STWP R0
        MOV  R0,@LOADWP
        RTWP
        AORG >FFFC
LOADWP  DATA REGS
        DATA ENTER
        END
```

Paul sure does have a way of making the "impossible" seem easy!

"AS-HELP"

By Richard Mitchell

STANDARD: 1A 2XB EA 4B 5A 6B 7A 9A

Many thanks go to Mack McCormick, an outstanding 99/4A Assembly Language programmer, for one of the many sets of tips he recently offered to all on the TI FORUM. His tips reminded me of the one thing that has long bothered me about the E/A package and led me to a solution.

One of the primary functions of a computer is that it should allow the user to pursue increasingly meaningful tasks by simplifying the processing of lesser tasks. A realistic term for this seeking of simplified methods is laziness. My knowledge of Assembly Language has not progressed at the pace of my other learning experiences because until recently I was unable to find a suitable method for simplifying the task of writing Assembly code. To the rescue comes "Assembly Source - Hope for Extremely Lazy Programmers", "AS-HELP" for short, a method of writing Assembly code with minimal time and frustration. One certainly does not have to be lazy to use it, but a lazy person might not write many programs without it. Ha!

I have long had a theory about what Assembly programming should be:

1. The programmer should have sets of source module files grouped by compatibility so that modules in a set can be combined to form all or a significant portion of a program. Printed listings of the modules could be maintained in a notebook or file system for ready reference (even when the computer is not available). Each set of modules could even become a group project, with programmers sharing compatible modules.

2. It should be possible to combine the modules with a minimum of effort into a single source file so that a complete picture is available for debugging purposes and that single source file could be made available to others. This assumes that the combination of modules will assemble, but may or may not run or achieve the desired results and may also require additional comments.

Step 2 was where my problems arose. One can either use the Insert command or the COPY directive, but neither offers maximum coding speed and efficiency combined with a complete source file that can itself be assembled. Normally, debugging a series of COPY files requires that each file be modified and saved, which strikes me as a very cumbersome task. Insert'ing seems even less desirable, as only a few modules could occupy more lines on the screen than can be viewed at one time, making it difficult to follow the program flow. Finally, the solution occurred to me.

When Assembling, one can direct the LIST file to disk. The LIST file is very similar to a SOURCE file, but has some additional information. The obvious solution was to use the LIST file and eliminate the extraneous information.

Here is an Extended BASIC program that removes non-source information from a LIST file. I call the program "SRCRESTORE":

```
100 DISPLAY AT(5,1)ERASE ALL
:"NAME OF LIST FILE:":"DSK"
110 ACCEPT AT(6,4)SIZE(12)BE
EP:F1$
120 DISPLAY AT(8,1):"NAME OF
```

-->

```
 NEW SOURCE FILE:":"DSK"
130 ACCEPT AT(9,4)SIZE(12)BE
EP:F2$
140 OPEN #1:"DSK"&F1$,DISPLA
Y ,VARIABLE 80
150 OPEN #2:"DSK"&F2$,DISPLA
Y ,VARIABLE 80
160 LINPUT #1:A$ :: A$=SEG$(
A$,3,LEN(A$))
170 CALL CHECK(A$,FLAG)
180 IF (FLAG=1)*(EOF(1)=0)TH
EN 160
190 A$=SEG$(A$,17,LEN(A$)-16
)
200 IF A$<>"" THEN PRINT #2:
A$
210 IF (EOF(1)=0)*(POS(A$,"E
ND",1)<>8)THEN 160
999 CLOSE #1 :: CLOSE #2 ::
END
1000 SUB CHECK(A$,FLAG):: FL
AG=0 :: IF LEN(A$)<4 THEN FL
AG=1 :: GOTO 1050
1010 FOR I=1 TO 4
1020 IF (ASC(SEG$(A$,I,1))<4
8)+(ASC(SEG$(A$,I,1))>57)THE
N FLAG=1
1030 NEXT I
1040 IF POS(A$,"COPY",1)=24
THEN FLAG=1
1050 SUBEND
```

This places a whole new priority on the COPY directive, allowing you to take advantage of having a complete source file created through a minimum of keying, as if you had used Insert and COPY at the same time! Changes can be made to a single file and it can be shared with your friends as a single file. You (and your friends) can begin with just a source file of equates, such as the one we offered in March and build a set of compatible files to be shared. If you aren't proficient in Assembly, you might get there yet and now you can do it the lazy user's way!

CORCOMP CONTROLLER

Rapid Color Changes

STANDARD: 1A 2XB 4B 5A 6B 7A 9A

If you've thrown up your hands in despair in attempting to use Assembly Language, there is still hope for fast programs if you are using a CorComp Disk Controller.

Listed below is a program that changes the screen color and the colors of all character sets in an instant without using a loop (which is one of several reasons Extended BASIC is slow in executing this task)!

Here is an example of the code you could use with the routine. The CALL COLORS parameters are defined just like CALL COLOR parameters, but the result is that all character sets and the screen change colors.

```
100 CALL INIT :: DELETE "LD-
CMDS" :: CALL COLORS(16,4)
110 CALL KEY(5,K,S):: IF S<1
 THEN 110
```

Here is the listing for the CALL COLORS subprogram (yep, one line!):

```
32000 SUB COLORS(FG,BG):: C$
=RPT$(CHR$((FG-1)*16+BG-1),3
2):: CALL LINK("VPOKE")(2048
,0,C$):: CG=BG-1 :: CALL LIN
K("WRTRG")(7,CG):: SUBEND
```

Well, we don't recommend tossing your E/A manual in the trash, but at least you can wipe the tears from your eyes and begin thinking about the exciting projects you'll be able to tackle with the Toolshed statements!

NEWS

A Chat With the President of Myarc

On July 22, we contacted Lou Phillips, President of Myarc, Inc. We thank Mr. Phillips for taking time from his very busy schedule to talk with us and for being very cooperative in answering our questions.

Of course, the most commonly asked question these days is whether Myarc will be releasing a computer. Mr. Phillips stated that Myarc will be announcing a new product by the end of

-->

the summer. Citing that details of the product are not final, Mr. Phillips stated that he preferred to not discuss the nature of the product, but said that Myarc would make "lots of noise at the right time". Our sources tell us the firm has been working on a computer prototype, but we'll obviously have to wait to see whether the promised new product is a computer.

The current news is that Myarc's new form of BASIC, which Mr. Phillips referred to as "Extended BASIC II", will be available by the end of July. The product will come with floppy disk, cartridge and manual. The floppy disk file(s?) contains the interpreter, which will require about 32K (comparable to the size of BASIC interpreters for other computers) and, due to the memory requirements after the interpreter is loaded, will require the new Myarc 128K memory card to provide adequate memory for program storage and workspace. The new form of BASIC will not use any GPL routines. GPL (Graphics Programming Language) is the language used by TI in their GROM (Graphics Read Only Memory) in both the console and the Extended BASIC module (as well as several other modules). GPL is the primary culprit in the slowness of TI Extended BASIC, as GPL must be interpreted in addition to BASIC being interpreted. Avoiding the use of GPL will allow the new BASIC to be very fast (sources tell us 3 to 4 times as fast as TI Extended BASIC). In addition to being fully compatible with TI Extended BASIC, the new BASIC will also be substantially compatible with GW BASIC, which runs on the IBM PC, with the major difference being in the DOS (Disk Operating System), which will, of course, be 99/4A compatible. Myarc's BASIC will also support many desirable features, such as 40 column mode and Hi-Res Graphics (Bit Image, including commands available in GW BASIC, such as Fill, Draw, Rectangle and Circle). Another new and useful feature will be integer support, which allows much faster processing of integer numbers. Also, the Accept function will have expanded capabilities, retaining information as to whether the acceptance key was the up, down or enter key, which is a very nice option, especially for use in creating a menu. The suggested retail price of the BASIC package is $84.95.

The Myarc Disk Manager has now been released. Though Mr. Phillips did not elaborate on the manager, it is anticipated that the manager will provide Myarc Disk Controller users with compatibility with all 99/4A disk formats.

When questioned about problems related to the Myarc 128K card, Mr. Phillips stated that the card is based on compatibility with the standard TI RS-232 and is therefore not compatible with the CorComp RS-232 card's PIO port. Myarc has a solid reputation for support of their products (several improvements were offered to hard disk owners over the past several years), but it appears the CorComp RS-232 may remain incompatible. Should other questions about the 128K card surface, we will check it out and report in future issues.

In response to our final question, Mr. Phillips made it clear that there is no relationship between Myarc and a certain Canadian firm. If you have been around those who pass gossip, you know what we are referring to. Otherwise, you are not missing anything.

In closing, Mr. Phillips stated that Myarc is selling products at the pace they have projected, placing the firm in "great shape" financially, and he is "excited" about the firm's future. We are quite excited about the possibilities from the firm, too!

The "New" Old Computer

Word from several user groups is that the line of computers announced by a West Coast firm turned out to be the 99/4A in a new case. Clever, huh?

-->

99 POTPOURRI

News, Corrections, Updates, Editorials, Kudos, and Come-what-may

Last month, we received info on the "Explorer" program from Millers Graphics just before press time and our description of the product was therefore inadequate. The program includes a feature for CPU emulation, which means that the software appears to the user to be the 9900 hardware, generating an abundance of information for the user. Virtually any area of the console can be explored to determine information about your program, making "Explorer" one of the most powerful programs to be imagined, much less to exist. And, it comes with a 100+ page manual. User reaction has been that the program has the Craig Miller mark of quality throughout.

In May, we mentioned Dave Rose's "Character Sets and Graphics Design". There is now a "CSGD II", priced at $12.95. The address is 2781 Resor Road, Fairfield, OH 45014-5053. The new package increases the total number of character sets to 18 and many new graphics are available. As many of the character sets are proportional and the package allows many centering and spacing options, it is not fast, but it does a fantastic job on some very intricate printing. The programs have been tested on the following printers: Gemini 10X and SG-10, Prowriter, NEC 8023, Epson and BMC-BX80. Available on disk only, be sure (!) to mention your printer model (2 versions exist). 8 bit printer graphics capability is a must!

An organization known as the TI-99/4A Users Association of Canada has been formed. For more information contact Jane Laflamme, 83D Glen Park Drive, Glouchester, Ontario, K1B 3Z1.

In Sheldon, IL, Wayne Burgess is the Sysop of "The Programmer", a BBS operating on a TI-West program, including use of a 128K card. The number is (815) 429-3533.

The hundreds of letters we receive each month are appreciated and we do carefully read each one. While it is not possible to reply to every one, we answer as many as we possibly can and encourage your correspondence, as your feedback is the primary factor in the molding of the direction of this publication!

FREEWARE

Steve Lawless, 2514 Maple Avenue, The Cedars, Wilmington, DE 19808, has 128-Writer, which stores the Editor and Formatter of TI-Writer in Bank 3 of the Foundation 128K Card, returning to program control immediately after use. Steve's MASSCOPY is now available in Version 3.25.

From the Ottawa TI-99/4 UG, P.O. Box 2144, Ottawa, Ontario, Canada K7S 1W7, comes DM-1000, a fast menu driven disk manager and TI-FORTH 2D-Graphics.

As this issue has focused heavily on topics of interest to advanced users and owners of full systems, we will make an effort to include more programs and beginner level material next month.

SUBSCRIPTIONS (PER YEAR):

U.S. AND POSSESSIONS	
FIRST CLASS	$16.00
THIRD CLASS	$12.00
OTHER COUNTRIES	
AIR MAIL	$26.50
SURFACE MAIL	$16.00
INDIVIDUAL COPIES:	
U.S. SUBSCRIBERS	
FIRST CLASS	$ 1.35
THIRD CLASS	$ 1.00
CANADA SUBSCRIBERS	$ 1.35
OTHER	$ 1.50

(all back issues were available at press time)

SUPER 99 MONTHLY ORDER FORM

NAME_____

ADDRESS_____

CITY_____STATE_____

ZIP_____COUNTRY_____

For back issues, specify which:

_____

READER FEEDBACK: (Attach comments)

Super 99 Monthly is published monthly by Bytemaster Computer Services, 171 Mustang Street, Sulphur, LA 70663.

EDITOR
Richard M. Mitchell (CIS 70337,1011)

CORRESPONDING STAFF WRITERS

Barry A. Traver
Charles M. Robertson

STANDARD KEY

1	Computer	A	TI-99/4A
2	Cartridge	XB	Extended BASIC
		EA	Editor/Assembler
		TW	TI-Writer
		MM	Mini Memory
		TE	Terminal Emulator II
		TX	Tax Investment
3	RS-232	B	TI
4	Disk Drive	B	TEAC 55B
5	Expansion Box	A	TI
6	Disk Controller	B	CorComp
7	32K Card	A	TI
9	Monitor or TV	A	TV & RF Modulator
10	Printer	A	Gemini 15-X PC
12	Speech Synthesizer	A	TI
13	Modem	B	1200 Baud

Bytemaster Computer Services
171 Mustang Street
Sulphur, LA 70663

Bulk Rate
U.S. Postage
PAID
Sulphur, LA 70663
Permit No. 141

08/85 23

[redacted]

POSTMASTER: ADDRESS CORRECTION REQUESTED.

JULY
SUPER 99 MONTHLY