

# PEEKING AND POKING — TI'S NEW PORTABLE — THE CC-40



By Bill Gronos

9505½ SE 15 #B  
Midwest City, OK 73130

It's smaller than a 250 page hard-back novel (reference: "The Queen's Gambit" by Walter Tevis - a good book). It's solidly built and reeks of class. It's the Texas Instruments COMPACT COMPUTER 40.

If you are as addicted to computers as I am, you've probably wished you could pack up your 99/4A and tote it about with you. What a great way to pass time in the dentist's waiting room or during a daily commute to work.

But what would you use for a monitor? Trying to pre-empt someone's favorite soap opera with "Enhanced Basic" would likely net you a broken arm as soon as your fingers touched the channel selector. Instead of the dentist's waiting room, you would end up in the emergency room.

And where are you going to plug it in on the bus? My 99/4A "freight train" requires eight electrical outlets — the lights dim when I turn on the system. The 99/4 is definitely not a portable.

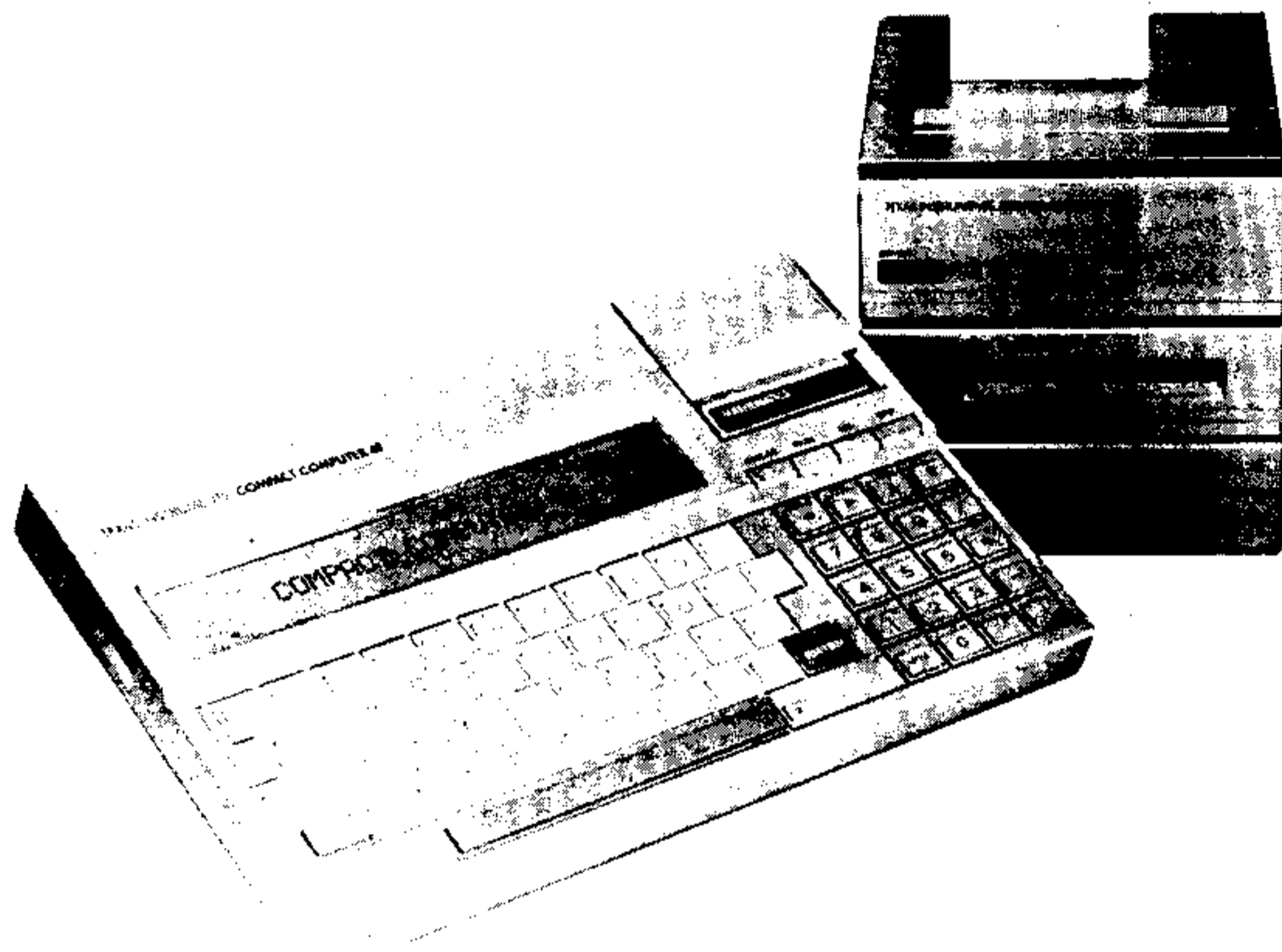
Seeing a market developing for portable computers, TI has thrown their hat into the ring along with Tandy, Sharp, Epson, et al. If TI can get some peripherals for the CC-40 on the shelves without their usual delays, they may just have themselves another winner.

## HOW PORTABLE IS IT?

Very! It will easily fit in a brief case with lots of spare room and if, like me, you aren't a suit-and-tie type, it will fit handily inside your backpack, foam packing and all. It travels very well.

I took the CC-40 along on a recent two-week business trip. I had often considered putting my 99/4A into a suitcase and taking it with me, but I always decided it just wasn't practical nor worth the risk of theft or damage. The CC-40 is the perfect computer for the businessman or traveling computer addict. I wish I'd had a CC-40 when I went to Saudi Arabia for two months.

Rather than read on my return flight, I uncased the CC-40 at 35,000 feet and added some features to the program I was writing for this column. Through no fault of the CC-40, this wasn't an easy task. The problem was



curiosity; The CC-40 drew a lot of attention. I was asked a lot of questions by my fellow passengers, which soon drew the attention of the airline stewardess. As she approached, I wondered if she was going to tell me that except for calculators, using electronic devices onboard an aircraft wasn't permitted. On a previous flight, I had a stewardess confrontation over my use of a computerized backgammon game; my argument that the game was no different than the calculator circuitry was in vain. It was different this time. The stewardess was very interested in the computer and asked more questions than the passengers did. The attention the CC-40 drew was a bit embarrassing.

Perhaps these computers will become this summer's status symbol, with Izod alligators and Gucci logos featured prominently on the keyboards.

The CC-40 will have far-reaching business applications when TI introduces their battery-operated modem. This will turn the CC-40 into a completely portable "smart terminal". An away-from-home executive can keep in touch with his electronic office from his hotel room or even a phone booth. I've always wished I could access my electronic mail service while I'm travelling — the CC-40 w/modem will make that wish come true.

The businessman's boon may well be the lawman's bane. The data sending capability of a completely portable computer/terminal system could likely be put to good use by drug dealers,

bookmakers or anyone else whose "business" requires secrecy. An easily written cryptographic program would make wiretaps virtually useless. Part of the cipher key could be memorized and only entered when the terminal needed to be used, reducing the hazard of compromise should the unit be seized. A one-button memory dump feature would destroy all evidence within microseconds in the event of a "surprise"; modern technology has made the spy's nitrocellulose-based flash paper pad obsolete. Hmmm...I must be reading too many spy books; let's get back to everyday uses.

Portability even has its uses while you're at home. Until I bought my 99/4A, one of my favorite hobbies was bicycling. Since I spend most of my free time on my computer, my Peugeot has been left to gather dust in the garage. I bought an exercycle to keep in shape, hoping I could pedal it and type at the same time: I never got around to building the special desk that would accommodate four feet of computer with room underneath for the exercycle, but I can easily use the CC-40 while I pedal. You can also sit outside on a lawn chair and get a tan while your computer is figuring tangents. The CC-40 certainly rates top marks for portability.

The 31-character LCD display has very readable quarter-inch high digits and it scrolls to 80 characters. Working with a one line display wasn't as limiting as I first thought it would be. Included in the BASIC instruction set is the PAUSE

ALL command. This will automatically halt the program whenever the display is filled so you can view your text or calculations. Program execution continues when enter is pressed. PAUSE can also specify the number of seconds you want the program to halt, with automatic resumption when the time is up. The resolution of the PAUSE command is about a tenth of a second, with maximum values far beyond the limits of my patience.

---

**The CC-40 is the perfect computer for the businessman or travelling computer addict.**

---

The maximum line length of 80 characters is accomplished by scrolling the display sideways when the display is filled. Unlike the TI writer and Editor/Assembler, which have 40 character displays and imitate an 80 character line by making a 20 character jump to the next segment of the line, the CC-40 scrolls its display a character at a time. This is far easier on the concentration and makes data entry easier than "window" jumping. Still, some other handheld computers have multiple line displays, which makes me wish TI had used a similar display on their portable.

As expected, graphics are very limited. Only seven characters are user definable and you work with a 5x8 grid rather than an 8x8. TI did include a large number of pre-defined, special purpose characters including the greek alphabet and a set of Japanese Kata-Kana characters (which may give a hint of their marketing plans).

TI has definitely got plans to market the CC-40 in Europe. Included in its repertoire of Enhanced BASIC subprograms is "CALL SETLANG". This command selects the language in which system messages and errors are displayed. CALL SETLANG(1) conditions the CC-40 to respond in German and instead of "BREAK" you'll get "UNTERBRECHUNG". Languages other than English and German will be accessed with plug-in modules and the CC-40 manual indicates French, Italian, Dutch, Swedish and Spanish will be available.

The keyboard is of the push button variety similar to the 99/4 (plain). Since I'm a two-finger typist, I found I preferred this style of keyboard because it allows the use of an overlay that can define every keytop. TI Writer has five handy functions that I don't use because I never remember which key they are and since they aren't located on the top row, the slip-in insert doesn't cover them. Proficient typists might have other opinions about the keyboard.

They CC-40 keyboard has 69 keys (compared to 48 on the 99/4) that are

laid out in three sections: typewriter keyboard, numeric keypad and a four button section marked "break", "run", "on" and "off". Although it has 21 more keys than the 99/4, it has ten fewer symbols. While tilde, reverse slant and reverse apostrophe are no great loss, others will be missed. Especially the "at" sign (@) and underscore(\_). The "at" sign is used as a control character in the Editor/Assembler and TI Writer. I have plans on using the CC-40 for writing articles and assembler programs when I travel, saving them to Wafertape and using the Hex-Bus adapter to transfer them to my 99/4 when I return. Luckily, both the E/A and TI Writer have "replace" functions that will let me substitute the percent sign (%) for the "at" sign with only a small bit of inconvenience.

As compared to the 99/4A, the CC-40 suffers from "wandering" symbols. Both have "shift", "function" and "control" keys, but quotation marks, which were "function P" on the 99/4A, are "shift=" and the parentheses went from "shift 9 and 0" to "shift 8 and 9". Since I'm often switching from one computer to the other, the "hunt and peck" typing I use on the 99/4 becomes the "all points bulletin" and "push" method on the CC-40.

The keyboard only has one shift key, on the left side, and the enter key is where the right shift key would normally be. It was annoying to keep pressing enter by mistake and get "variable not defined" errors, but you get used to it.

Real typists aren't going to like the fact that the mathematics symbols (+ - \* /) on the numeric keypad aren't duplicated on the main keyboard. Happy hunting!

In spite of these minor drawbacks, the CC-40 keyboard has several outstanding features. Ten of the keys, (0-9), are user definable and come in very handy for storing frequently used text, numerical calculations or even single line programs. Up to 80 characters can be stored for each keytop. These keys will hold their contents even after you turn the computer off. Simply press [function] [0-9] and your stored text reappears in the display as if you had just typed it in. If you define key 1 to be: "FOR X =1 TO 60:DISPLAY AT(15), X;:PAUSE 1:NEXT X:DISPLAY BEEP", you'll have a 60 second counter just by pressing [function] [1] [enter]. Another key could be coded as a stopwatch, a compound interest calculator or numerous other applications. Forgetful businessmen could find it helpful as a reminder board. User defined keys in no way interfere with stored programs.

The CC-40 has another outstanding keyboard feature: one button BASIC words. Instead of having to type "print"

a thousand times in every lengthy program, I simply press "function k". Twenty-seven of the most often used keywords in BASIC can be entered with a single symbol. Whether you're a rated typist or not, this feature really speeds up program entry.

While we're on the subject of BASIC, let's compare the 40's "Enhanced BASIC" to 99/4 Extended BASIC. The following functions have been added to Enhanced BASIC:

	FUNCTION PURPOSE
ACS	COMPUTES ARC COSINE
ADDMEM	ADDS EXPANSION RAM
ASN	COMPUTES ARC SINE
ATTACH	SAVES SUBPROGRAM VARIABLES
CLEANUP	DELETES UNUSED VARIABLES
DEBUG	ASSEMBLY LANGUAGE DEBUGGER
DEG	SETS ANGLE UNITS DEGREES
DELETE	REMOVES SPECIFIED PROG LINES
EXEC	EXECUTES ASSEMBLY LANG PROG FORMAT INITIALIZES DEVICES
FRE	RETURNS MEMORY INFORMATION
GETLANG	CHANGES PROMPT LANGUAGE
GETMEM	RESERVES MEMORY BLOCKS
GRAD	SETS ANGLE UNITS TO GRADS
INDIC	CONTROLS DISPLAY INDICATORS
INTRND	RANGES RANDOM NUMBERS IO SPECIAL PERIPHERAL CONTROL
KEY\$	HALTS PROG TILL KEY IS HIT
LN	COMPUTES NATURAL LOGARITHM

The extra memory commands were added to allow better use of the limited memory and to aid in adding assembly language programs and subroutines.

The extra math functions extend the CC-40's use as a calculator and it is excellent for this purpose. I own a TI 58 calculator and found I prefer to use the CC-40 instead. It's 31 column display allows you to keep better track of your long calculations and using variable names instead of register number decreases the chance of making mistakes. Also, my TI 58 will last less than three hours between charges, while the CC-40's batteries are good for 200 hours. The 40's LCD display is a lot easier on the eyes.

Having a built-in debugger is a great aid for one of my favorite hobbies

— exploring the inner worlds of TI's computer products that they seem so reluctant to tell you about. It made the writing of the program listed on page 37, which required figuring out how Enhanced BASIC is stored in RAM, far easier.

INTRND is a function that would be more useful in Extended BASIC than in Enhanced BASIC, since it is very useful in games. It will give you random integers between 1 and the number you specify. Here is a stock market analysis program for all you wheeler-dealers:

```

100 REM OULJA STOCK MARKET
    ANALYSIS ALGORITHM
110 OPTIONS(1)="BUY"
    :OPTIONS(2)="SELL" 120
    PRINT OPTIONS(INTRND(2))
    ;" ";
130 FOR X=1 TO 3
140 PRINT CHR$(INTRND(26)
    +64);
150 NEXT X
160 PRINT "PRESS ENTER FOR
    NEXT TIP":PAUSE:GOTO 120

```

If you believe in Fate, this program just may pick you a winner. However, if it comes up with "BUY WCI", you may want to run it again.

---

**The CC-40 has (an) outstanding keyboard feature: one button BASIC words.**

---

I find the DELETE function to be help and hindrance both. The hindrance is not being able to delete a program line just by entering its number — you must enter [function] [delete] [line number]. The slight bother of this function is more than compensated by its usefulness: it allows you to delete entire blocks of line numbers with a single entry. This isn't an easy trick on the 99/4, requiring Extended BASIC (for the MERGE function), a disk drive and the right software. Any of you who have pressed the sequence [function] [down arrow] [function] [erase]...ad infinitum, ad nauseum, as often as I have will appreciate this feature.

Of all the extra functions that Enhanced BASIC has that Extended BASIC doesn't, the one that is of greatest value is KEY\$. Don't confuse this with CALL KEY, which both BASICs have. KEY\$ can be used as a string variable in any expression and causes a program to halt until a key is pressed. Let me illus-



trate the usefulness of this function by comparing two simple programs:

```

100 REM PROGRAM TO DISPLAY
    KEYS AS THEY ARE
    PRESED-EXTENDED BASIC
    VERSION

```

```
110 CALL KEY(A,K,S)
```

```
120 IF S=0 THEN 110
```

```
130 PRINT CHR$(K)
```

```
140 GOTO 110
```

```
100 ENHANCED BASIC VERSION
    OF KEY PROGRAM

```

```
110 PRINT KEYS
```

```
120 GOTO 110
```

As you can see, KEY\$ is a handy function.

Now let's look at the other side of the coin; what does Extended BASIC have that Enhanced BASIC doesn't?

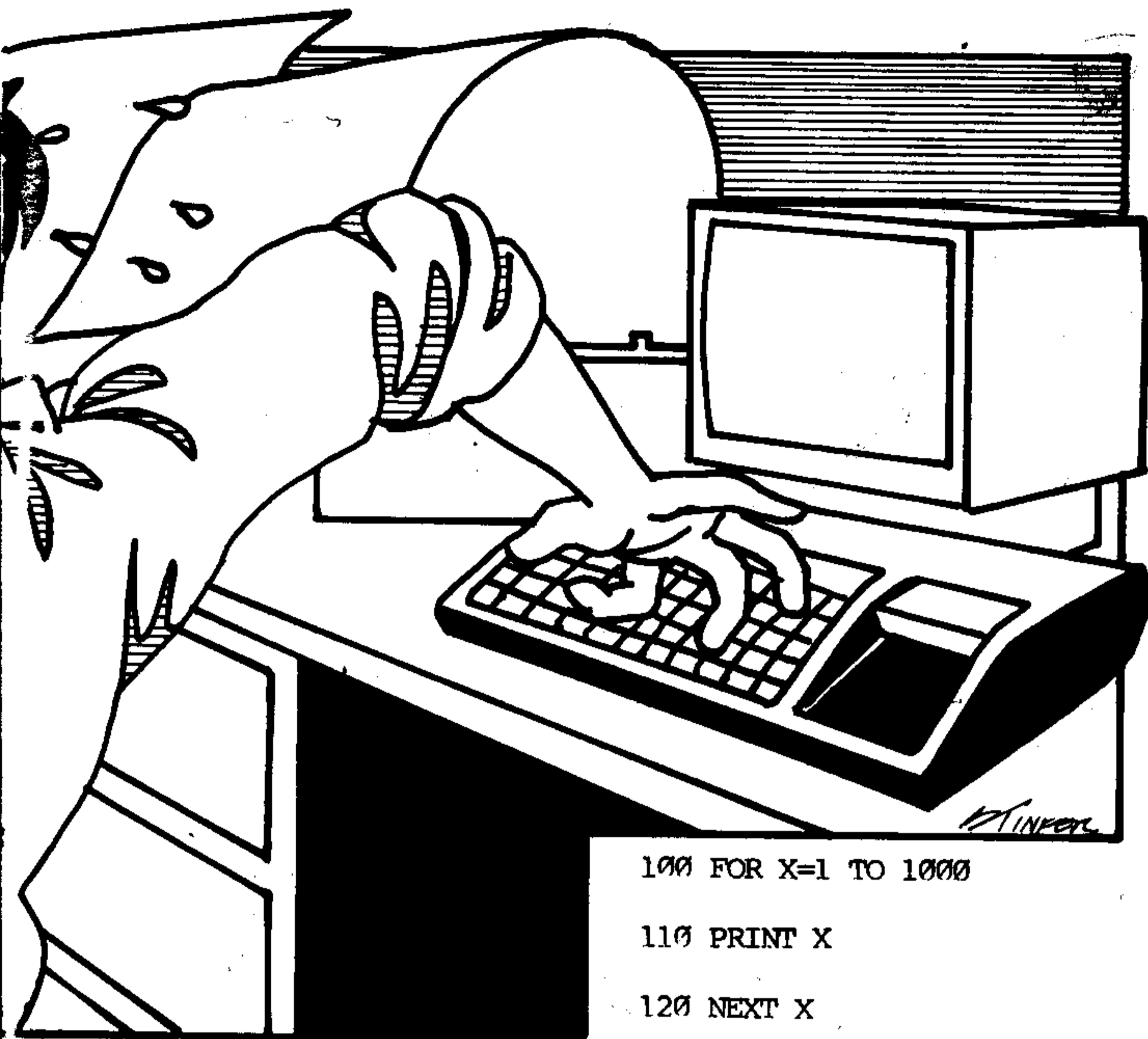
Well, since the CC-40 doesn't have a joystick port, there's no need for a CALL JOYST function. Likewise, since there are no Sprites, there are no sprite commands. The following graphics functions would be of little use in the

CC-40, so they were omitted: CHARSET, CHARPAT, COLOR AND GCHAR. There are no speech functions. If you want sound in your Enhanced BASIC program, you have to be content with simple beeps using lines like 100 DISPLAY BEEP BEEP BEEP.

Three useful Extended BASIC functions did get left behind: MAX, MIN and REC.

There are many more functions in the CC-40 that are too numerous to explain in a single article. Suffice it to say that this small computing marvel is not a stripped-down model. It gets top marks for programming power.

The CC-40 is no slacker on the technical side either. I was unhappy to learn that it does not belong to the "990" family, which means I'm going to have to learn another Assembly language, but there is good reason for this: battery life. The central processing unit (CPU) is a 2.5 MHz TMS70C20 microcircuit and was obviously chosen because of its CMOS architecture. CMOS stands for complimentary metal oxide semiconductor. CMOS' main advantage is low power consumption. I'm still on the original set of batteries. TI claims a 200 hour battery life and they might have understated it. Two hundred hours equates to 25 days of constant eight hour use. No telling how long the constant memory would last if the unit were never turned on.



```
100 FOR X=1 TO 1000
```

```
110 PRINT X
```

```
120 NEXT X
```

The 99/4 required three minutes to run this simple program, but the CC-40 did it in a mere 44 seconds. The reason for this is no mystery; the CC-40 doesn't have to scroll a memory-mapped monitor screen nor is time spent in Graphics Programming Language (GPL) interpretation. Having the display memory space directly available to the CPU gives the CC-40 a big software advantage when programs require extensive printing.

The display speed is a big plus, but could the CC-40 hold its own in mathematical computation?

```
100 FOR X=1 TO 50
```

```
110 PRINT SIN(.7853981634)
```

```
120 NEXT X
```

This program took 16 seconds on the 99/4 and 10 seconds on the -40. If we change line 110 to  $Y = \text{SIN}(.7853981634)$ , both computers will run the program in about eight seconds. Clearly, it is the printing time that is making the CC-40 appear to be quicker than the 99/4.

Let's put a fairly complicated math expression into the test. What would \$25,000 earn at eight percent interest compounded daily for a year? Change line 110 of the above program to:

```
110 Y=25000*(1+.08/365) 365
```

I got to wondering if the constant memory was the type that needed no current at all to store data. Such memories are called EEPROMs (electrically erasable programmable read only memories). To check this out, I pulled out all the batteries for a few seconds and then replaced them. The contents of memory remained intact. Further experimenting showed that the memory would be lost after about four minutes without batteries installed. Thus, the memory does use current, but it doesn't use much. Since there is a low battery indicator on the CC-40, and you have four minutes when batteries can be left out with no ill effect, your stored programs will remain in memory even if you change the batteries and be ready to run as soon as you turn the unit on. The CC-40 has a very respectable 34K of ROM for storing the system monitor and BASIC interpreter. There is 6K of built-in RAM memory with about 5700 bytes of RAM available for user programming. Add on memory cartridges will allow RAM expansion up to 18K.

Unlike the CPU in the 99/4, the CC-40's brain is an eight bit microprocessor. How does this affect the speed of execution? Well, it should definitely lag behind in the math department, so I ran a few simple programs to test this out.

First, the "idiot benchmark" was tested:

Since this program has no printing and is heavily mathematical, the 99/4 shows off its number crunching prowess and turns in a time of 13 seconds while the CC-40 uses 28 seconds — more than double the time.

Overall, the CC-40 is not as fast as the 99/4, but it is plenty fast enough for the purpose for which it was designed.

While the CC-40 is very advanced technically, the internal structuring is far simpler than the 99/4. By internal structure, I'm referring to how memory is allocated and high-level programming languages are implemented. The 99/4 internal memory handling is very complex. It has three memory systems: graphics memory (GROMS), video display processor memory (VDP RAM) and central processing unit memory (CPU RAM or ROM). The 99/4 BASIC programs are stored in VDP RAM — the system console contains a mere 256 bytes of CPU RAM; the 16K is VDP RAM. The 99/4 is doing constant memory shuffles when a BASIC program is executing.

The CC-40 has only good 'ole, easy to understand CPU RAM. This fact, combined with the PEEK and POKE functions allows you to do many programming shortcuts that are impossible in 99/4 console BASIC. The CC-40 BASIC program on page 37 illustrates one of these tricks.

**FAST FULL-SCREEN  
DATA PROCESSING  
WITHOUT  
DISK OR EXPANSION  
WITH THE  
MINI EDITOR!**

At last! A professional-type data processor for home-scale use with the TI99/4(A). ASSEMBLY LANGUAGE program for instantaneous editing of data as you review it on a FULL-SCREEN DISPLAY. Save and load data up to 12K long at 4K PER MINUTE. Combine texts in memory. Also offers complete formatting of texts for printing, with automatic word wrap and line centering, right justification, many more useful features designed to give you total control over printer. Great for all word processing, inventories, bookkeeping, catalogs, lists, etc. Demonstrated at the Pittsburgh Users Group. THE MINI EDITOR REQUIRES ONLY A CASSETTE RECORDER AND THE MINI MEMORY. Write for a free brochure or send \$25.00 check or money order for cassette and manual to:

**ADVANCED SOFTWARE**

181 Carnegie Av.  
Clairton, PA. 15025

PA residents add 6%

Suppose you wanted to write an inventory control program where you could input an item number and retrieve supply data and your product numbers were five digits long. With a large product line, access speed could be a big problem with such a program. Inventory statistics could be stored in DATA statements and you could do a read and compare to find the desired product. Part of your program might look like this:

```
220 INPUT "ENTER ITEM NUMBER"
:NUMBER 230 READ
PRODUCT,COST,QUANTITY
```

```
240 IF NUMBER=PRODUCT THEN
300 250 IF NUMBER=
"999999" THEN 400
```

```
260 GOTO 230
```

```
10000 DATA 10123,24.95,15
```

```
10010 DATA 10124,9.95,1123
```

```
11000 DATA 11123,45.95,8
```

One way to speed up the selection process is to group item numbers systematically under line number sections and RESTORE the data to a particular section, decreasing the number of comparisons required.

Just think if instead of RESTORE 10000 you could use a variable in the restore statement, as in RESTORE NUMBER. The BASIC line number could be the product number, which besides saving time would decrease the memory needed. The data in the above example would then become:

```
10123 DATA 24.95,15
```

```
10124 DATA 9.95,1123
```

```
11123 DATA 45.95,8
```

Your line number will take up the same memory space whether it be 1 or 32000. This is a far superior method, but RESTORE X just isn't legal.

BASIC, however, is an interpreted language. Each statement is re-evaluated each time it is executed. The POKE function allows us to "diddle" the memory in which our BASIC program is stored. What would prevent us from changing a line such as 180 RESTORE 1000 to 180 RESTORE 10123 just before the line is executed? NOTHING!

I have used this method in 99/4 BASIC with the Minimemory function POKEV (poke video display memory —

remember, 99/4 BASIC programs are stored in VDP RAM) and it works great. It requires knowing how BASIC stores a particular line and finding the addresses that contain the 1000 in RESTORE 1000. Don't go looking for the ASCII string for "RESTORE" in memory, 'cause you're not going to find it. BASIC programs are "crunched" and any function word is represented by a single byte value.

Briefly, the technique I use is to create a single line program like:

```
100 REM ZCZC
```

While I'm not going to find "REM", I will, with time, find "ZCZC". I picked that particular string because it isn't likely to be duplicated by accident within a program. Once you find the general memory area, you use another short program to decrypt the line you want to study:

```
100 REM ZCZC
```

```
110 RESTORE 1000
```

```
120 REM ZCZX
```

Once you know how and where RESTORE 1000 is stored, you will be able to convert it to RESTORE 10123 or any other line number. It is then only a short step to having your program alter the line based on a variable value and, viola!, you effectively create a RESTORE X function.

Instead of an inventory program, I built a quick access "address book" program around RESTORE X. This program is given at the end of this article. It's main feature is that you only have to key in the minimum amount of letters needed to make a name unique. Here is an example that will work with the DATA statements I've included in the coding. The entries you press are shown in square brackets:

```
[run]
```

```
[enter]
```

```
ENTER FIRST LETTER
```

```
[C]
```

```
CARTER; JACK
```

```
[H]
```

```
CHRIS CROSS 123-4567
```

```
[A]
```

```
CHARLIE LAFARA 948-1023
```

Since this was the name that you wanted, you press [enter] and the street will be displayed and press [enter] again to display city and state.

The DATA statement format is:

```
NAME AND PHONE #,STREET,CITY
STATE ZIP
```

The length of each item is flexible and you can use more than one data statement. The line DATA statement line number for a given name must be three digits preceded by the alphabetical sequence of the name's first letter. "Adams" could be 1001,1010,1234 etc. It could not be 10000 because the tenth letter of the alphabet is "J". I suggest you put your names into order by the frequency that they are used, within a particular letter's subsection. This will speed the retrieval time, which is nearly instantaneous for each letter pressed.

I've allowed a dual entry option that let's you select by first or last name without having to entirely duplicate all the data. The alternate format for the second entry is:

```
NAME, ,LINE NUMBER OF DATA
STATEMENT
```

The "" symbol tell the program the full set of data is elsewhere and it does another "RESTORE X" to the line number that follows the "".

Novice programmers should note the slick use of the ON ERROR function to alleviate having to put in REM DATA Z, etc. type lines when no names begin with that letter. Without the first line, the program would halt with a bad line error should a key be pressed for which there is no data.

One caution. Since precise memory locations are being altered, the program format must have exactly the same number of bytes up to the last RESTORE statement. To check the program entry, use the following commands directly from the keyboard:

```
CALL PEEK(8051,X,Y)
```

```
PRINT X,Y
```

```
CALL PEEK(7798,X,Y)
```

```
PRINT X,Y
```

Both prints must produce "3 232". If they don't, the program is entered incorrectly and running it could likely cause loss of the program by altering a critical portion of a line with an incorrect POKE.