CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC

COMMODORE

GEOSYNC:
Letters, News & Hard Drives

by D. Roderick Eamon

Like most folk, when reality chooses a different road for me than the one I planned to take, I get irritated. Seems it happens more often lately. Because of the rapidly changing nature of the software industry, I plan only one column ahead. This month I planned to answer reader letters and reveal a few interesting, undocumented features available to GEOS 2.0/REUsers. You know, the in-depth free-hand stuff columnists love to write about.

No such luck. Fate intervened with the arrival of a brown-wrapped package from Berkeley, California. From the far West, it came without warning: geoChart! Not the final version but a review copy of the application. If BSw remains true to form, any change between this version and the one at your local software supermart should be limited to bug fixes.

Berkeley Softworks doesn't release new products every month. Originally, I'd heard whispers about geoChart late last Spring, when it was still in larval stage. Because geoChart isn't a major release, as were geoPublish and geoProgrammer, this column may be the first you've heard about it. Notice any major fanfare during November, its official release month? I didn't.

Frankly, the idea of generating charts and graphs from raw data didn't excite me that much. Obviously, this is a hold-over from my early (tramatic) math training. So I plodded ahead, certain that I'd have to dust off the old high-school trig textbook and cram. Fighting my anxiety, I opened the new disk called geoChart, installed it to GEOS 2.0, and prepared for a long night with the 116 page manual.

Hard Data: Hardware Requirements

System requirements: GEOS 64 (1.2, 1.3 or 2.0); all Commodore C/D 64/128 computers (running in 64 emulation); TV or monitor; joystick, mouse, or other input device; 5.25" disk drive (1541/71 or compatible), and two or more disks.

Recommended options: REU or second 1541, -71 or -81 drive; printer; and any version of geoWrite, geoCalc, or geoFile.

The geoChart disk itself is sparsely populated for the GEOS application. Included ondisk, Side 1: geoChart and a copy of GEOS V-2.0's Note Pad desk accessory (this latest Note Pad allows cut and paste of text Pad pages). That's it. As a user accustomed to geoApplications disks—usually overflowing with sample files, printer drivers, and an odd utility or ten—finding only the minimum number of needed files is indeed a refreshing approach. Side 2: an auto-run demo of GEOS 2.0.

My personal recommendations: upgrade to GEOS 2.0, buy an REU (the 1750 if you run with 128), get a decent printer (an 80 dot per/inch printer), and a 1581 3.5" drive for data storage. These additions will increase your file handling powers and decrease time spent waiting for disk accessing and VLIR module swaps. And with GEOS 2.0, file handling and creating custom disks is a breeze!

Documentation

My first surprise was a sample file mentioned in the manual. Well, it certainly wasn't on either side of the disk! It's embedded in the geoChart program itself. Double-click geoChart's icon and a bar graph displaying "Widget" sales appears. Select "chart" on the Command Bar, then one of the nine selections, and geoChart will re-display this "default" data (in this case, the demo Widget data) in the graph form you've chosen. This built-in default is the best piece of tutorial documentation in the package.

Not that the manual is bad—it is actually quite good. A recent policy change at BerkSoft encourages GEOS Beta testers to suggest documentation revisions. On the whole these testers are a very informed lot: their suggestions work. The old saw, "A camel is a horse designed by committee," doesnt' apply. If I read the signs right, geoChart's User's Manual already shows benefit from this program. It also seems that they've paid attention to reviewers: the index and appendices are complete and easily used to look up nearly any problem encountered, including error messages.

Following the precedent started with the GEOS 2.0 manual, the important moves geoChart makes (and the moves you'll make too) are repeatedly mentioned in "Notes" at the end of each section. Included in Chapter 2—"Learning geoChart"—is an introduction to geoChart's data import and chart production techniques, followed by an extensive Tutorial that teaches how to produce chartable data using Note Pad and how to paste that data into the application. Each successive chapter marks a clear course through the more complex geoFile/Calc waters. Full instructions are provided on workable spreadsheet and database file design, how to alter Chart labels, fill patterns, and how to change values within the charts themselves.

The only area I found hard to understand was with the scatter chart (more on this later).

Operational Considerations

After gaining a certain familiarity with geoChart, I discovered just how useful this program can be for occasional geoUsers. For example, if you use geoWrite, geoCalc, and/or geoFile to control your budget, geoChart provides a visual presentation of those numbers. If you run a small business, you can create daily, weekly, monthly or yearly charts to display sales. Graphic presentations of sales areas or items that require improvement become glaringly obvious. Any measureable data can be charted!

geoChart In Action

Entry/exit and mousing procedures are similar to other GEOS applications. But once in, the really interesting things begin to happen.

Let's say we've followed the instructions and now have a working Text Scrap on our disk that we want to translate into a chart. By selecting Create in the Create/Open/Quit dialog box, naming the new file, and finally clicking on paste from the Command Bar menu on the main screen the data is ported into the program.

The Data Mode display (Figure 1, page 284) appears. Rows and columns of tiny boxes (called the Grid) pop into screen center, along with two button icons (Clear and Chart), and a Data Series dialog box. The horizontal columns (Data Series) represent the strings (the geoWrite/Calc/File/NotePad lines of numerical data) we've imported. Each box (value—the actual data—and corresponding labels such as dollars, months, etc.) represent the axes of the data. The total number of rows and columns depends upon the volume of data in the series we've imported.

In Data Mode, we have an opportunity to decide how much data to show in the charts. A nice feature of Data Mode is that we can return to it at any time and re-select the data boxes as

continued on page 284

TEXAS INSTRUMENTS

TI FORUM

by Barry Traver and Jonathan Zittrain

Four "Programs For Programs" From Asgard

Asgard Software (P.O. Box 10306, Rockville, Maryland 20850) continues to support the TI/Myarc community with innovative programs. Charles Earl's PRESS will be reviewed in a future issue; this month's column will deal with four Asgard releases intended not as stand-alones but as programs to be used with other programs: QUICK-RUN, PRE-SCAN IT!, BATCH-IT, and EZ-KEYS PLUS. Most of the attention will be given to the two more recent releases, QUICK-RUN and BATCH-IT, introduced in fall-winter 1988.

QUICK-RUN and BATCH-IT are both unlike anything previously available commercially for the TI; both, however, are also marred (at least in the first release of the manual). Fortunately, as a result of information gained through helpful telephone conversations with authors Travis Watford and Charles Earl, this column is able to remedy those deficiencies.

QUICK RUN by Travis Watford ($9.95) has as its purpose the same purpose as Asgard's earlier PRE-SCAN IT! by J. Peter Hoddie ($10.00): to get your Extended BASIC program "up and running" as soon as possible. But there's an important difference: PRE-SCAN IT! did this by cutting down pre-scan time, while QUICK-RUN does it by cutting down initialization time, i.e., time ordinarily spent in reading data, performing calculations, redefining characters, etc. It does this by taking a "snapshot" of the program after such housekeeping work has been completed and then saving the result to disk.

The program is saved in a somewhat unusual format—DIS/FIX 64 files instead of the usual PROGRAM image—and requires use of a special loader to run the resulting program. Asgard, however, is to be commended for its policy that "programs saved with QUICK-RUN may be distributed without royalties" (Asgard Software fall '88 catalog) and that "the LOAD routine may be freely distributed with programs with the SAVE routine" (QUICK-RUN manual), which means that you can share with your friends programs that have been put through QUICK-RUN.

Asgard does warn that they do not guarantee that QUICK-RUN will work with all 99/4A consoles (and there are indications that it in fact does not do so in its first released version), but does offer what they call "a money back guarantee": "If you are unable to get this program to function with your console, we will give you a credit valid with the purchase of any other Asgard product." (I'm not sure that offering a credit voucher toward purchase of another Asgard product is exactly the same thing as offering your money back, but that may be a minor quibble).

Anyway, the program does break new ground, even though I myself have had some problems in getting it to work properly. Most of my problems, however, were due not to the program, but to the documentation. The one "QUICK-RUN example" provided in the first release of the manual does not (and cannot) work for a number of reasons. The "LINPUT #:A$" in line 120 of the second program is easy to spot and correct to "#1:," but the fact that a DIM A$(100) needs to be added to the first program is not at all obvious. In addition, if you follow the directions and do a CALL INIT before running the second program without doing a CALL LOAD ("DSK1.RUN"), the program cannot perform the CALL LINK ("MERGE", "1.TEST1").

continued on page 468

CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC

Texas Instr., TI Forum
continued from page 277

Here are corrected (and improved) versions of the programs, kindly supplied personally by Travis Watford:

```
100 DIM A$(100)
110 CALL CLEAR :: CALL INIT :: CALL LOAD("DSK1.SAVE") :: CALL LINK("SAVE", "1.TEST")
120 BREAK :: CALL CLEAR :: CALL INIT :: CALL LOAD("DSK1.SAVE"):: CALL LINK ("SAVE", "1.TEST")
130 FOR COUNT=0 TO 100 :: PRINT A$ (COUNT) :: NEXT COUNT
100 CALL CLEAR :: CALL INIT :: CALL LOAD("DSK1.RUN")
110 DIM A$(100)
120 OPEN #1: "DSK1.TEXTFILE" [supply an appropriate file of your own here]
130 LINPUT #1:A$(COUNT)
140 COUNT=COUNT+1
150 IF(EOF(1) OR COUNT=100) THEN 160 ELSE 130
160 CLOSE #1
170 CALL LINK ("MERGE", "1.TEST")
```

The directions for using these are much simpler than given in the manual: (1) RUN the first program (which will create DSK1.TEST), (2) after it BREAKs, RUN the second program (which will load and run the first program), and (3) after it BREAKs, enter CON (which will create a revised version of DSK1.TEST, which can be run from the LOAD program supplied on the disk). The contents of the array A$() are now defined (as the screen display confirms) and a permanent part of the program, no longer requiring any time for initialization! The manual states that "at a minimum, QUICK-RUN adds 26 sectors to the length of your file on disk," but (according to Travis) the real minimum is 12 sectors. Also, the warning not to put the CALL LINK for SAVE in a program loop because it would be executed each time the loop was executed is inaccurate: the line is changed to a REMark as soon as it is executed the first time (although it is still not a good idea to execute the SAVE routine within a loop, since that is where the resultant program will pick up execution when run).

Already well-known for his significant break-through public domain program MAX-RLE, Travis Watford has again produced something new. It may or may not work for you (I was unable to LIST the TESTAFTER program supplied on the disk: all that showed up was one line, and that was "garbage," although the program seemed to run okay), but for its price, you may want to give it a try (especially after

continued on page 469

Commodore, CP/M
continued from page 467

This method might seem a bit crude, but I'm open to any new suggestions. I have heard that some portion of the 128's memory may be shared with CP/M, but have been unsuccessful in locating it.

For any Lt. Kernal buffs that are interested in my menu program, please write to: Warren S. Knospe, 1 Slate Creek Drive #12, Cheektowaga, NY 14227.

My system is very complete and runs on the 128 in 80-column mode, and takes full advantage of the 128's graphic capabilities. (Lt. Kernal required, but I can modify the system to run without it.) It can currently catalog over 30,000 C-64, C-128 or CP/M files, complete with file descriptions, 5 user selectable windows, and is entirely joystick (ports 1 or 2) or keyboard-oriented. "System Maintenance" mode, a HELP window, a full set of database search/add/delete commands and the option to print or view the database catalog is included. I am also working on a feature that will allow C-Net Sysops to import filenames and descriptions directly from the U/D section of C-Net!

Warren Knospe

Though I have no idea how many of you are using the Lt. Kernal, I thought someone might find this utility interesting or useful. Contact Mr. Knospe for further information.

Dear Cheryl Peterson:

I felt after seeing how sparse the CP/M column was in the September issue of *Computer Shopper*, that I should write to let you know how much I've appreciated your column.

As a relatively new user of CP/M, yours is the only column I've read that made sense and helped me get through the turmoil of discovering CP/M.

CP/M has become a joy to use because of the tips and helpful hints in your column and I feel cheap because I've never thought about sharing my own discoveries with your readers. I hope to rectify this in the future.

I recently purchased a NEC 8500 laptop computer that runs CP/M 2.2, and will use it with my C-128 at home, to get a little more time to write, and hopefully do some real programming for the CP/M + on my Commodore.

Back to your readers, in the August column Ted Chidester wrote that his 1571 was scrambling his dBase files when saved to a double-sided Commodore CP/M disk. I've run into this problem with some other file types, and I believe it's due to the bugs in the early 1571 ROM. He states that he has the upgrade ROMs in his drive, so I don't think that's his problem. I recommend he get hold of a program called C1571.COM. It disables the automatic verify on the 1571, speeds it up and has prevented problems for me. If that fails, another program XFORMAT.COM will allow you to format disks in Kaypro, Osborne, Epson, or IBM and these are MFM formats, the GCR BIOS routines will not be used. This will definitely eliminate your programs disk problems. As a bonus the Kaypro IV format will give you more storage than a double-sided GCR disk. The read and writes are a little slower, but worth it. One warning, don't use the C1571.-COM program with XFORMAT.-COM; it knocks the heck out of the drive.

In the September issue's column you had some information on WordStar for the C-128. I've been using the built-in WordStar on my laptop, and it's an excellent program. But, if you don't have $89 laying around, I'd recommend getting a copy of VDE2.31 by Eric Meyer. It's available public domain, and it has all the features of WordStar. I've installed it for my C-128 and it's what I prefer to use now for typing in source code and correspondence.

Also in the August issue, James Synnamon wrote about some "undocumented" features he'd found on his C-128 running in CP/M +. If read carefully, the CP/M section in the C-128 User's Guide has a wealth of information, including all the features James mentioned. Using the KEYFIG program on the system disk, I've completely revamped the keyboard assignments on my C-128. And you can make custom boot disks for different programs that each have the keyboard set up the way you want. It just takes a little experimenting. The console command characters listed in the user's guide are a boon to the programmer. They make it easy to control the display in BASIC or assembly, simply by sending the right characters to the console. I would also recommend the C-128 Programmer's Reference Guide, even to the non-programmer, just for the information on the console command characters.

If you have the 1571 disk drive, and a little time, you can use the extract feature of the help program on the system disk to create a printable copy of the help files. With a printed copy of them, you have a fast way to look up features when computing without looking through books, or having to load the HELP.COM to look them up.

To me, CP/M seems to have more going for it than MSDOS, though I've not really been exposed to the PC-compatible world. It makes more sense to me to find a program that does what I need running in a 58K TPA than to need 640K to just load a program that might be faster, but not really better.

My source for the PD programs has been Gil Cabral, 4 Terry Lane West, Wareham, MA 02571.

Tim Mihulka

I hope some of the readers can make use of this information and I look forward to any other tips you care to share. It's always nice when those who have used CP/M successfully try to "give back" some of the help they received early on. Thanks for writing.

That's it for this month. Keep those letters coming. Cherp! ●

CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC COMPUTERS CLASSIC

Texas Instr., TI Forum
continued from page 468

this review has rectified the deficiencies of the manual!).

BATCH-IT by Tom Bentley and Charles Earl ($19.95) is a batch language for the TI-99/4A and Myarc 9640. For those not familiar with the concept, a batch language allows you to write a program that will run other programs (including doing such things as providing entries or keypresses at appropriate prompts and making semi-intelligent choices based on what is going on). Simply put, for those tasks that are fairly routine a batch program can take your place at the keyboard, freeing you for other tasks that may require more intelligence.

How would you like a program that automatically loaded in DM1000, copied the disk in drive one to the disk in drive two, and (when the copying was complete), beeped and waited for you to press <enter> or "Q" to tell it whether to copy another disk or return to the main DM1000 menu? Well, here's a simple sample batch file I wrote to accomplish exactly that:

```
    DEFINE VAR1,1,"1"
    DEFINE VAR2,1,"2"
    DEFINE VAR23,40
    LOAD "DSK5.MG"
    RUN
NEWDISK
    KEY VAR2
    KEY VAR2
    KEY VAR1
    KEY VAR1
    KEY VAR2
LOOP
    GET 23,0,VAR23
    SEARCH "PRESS",VAR23
    ONMATCH CON
    GOTO LOOP
    BEEP
CON
    ONKEY 13,NEWDISK
    ONKEY 81,STOP
    GOTO CON
STOP
    CHAR 14
    CHAR 14
END
```

Since, unfortunately, the documentation for BATCH-IT is both brief and at times even inaccurate, let me explain the preceding program. VAR1 is defined as "1," VAR2 is defined as "2," and a VAR23 of length 30 is set up as a variable of length 40 to contain information on line 23 (the bottom line of the screen). (The purpose of these should soon become clear.) BATCH-IT then loads and runs DM1000 ("DSK5.MG" in my system), and does its own keypresses of 2, 2, 1, 1, and 2 to start the program making a bit map disk copy from drive 1 to drive 2.

How does BATCH-IT know when the copying is completed?

Simple: by checking the last line of the screen, to see whether it says "PRESS ANY KEY WHEN READY." It stays in the LOOP until it finds that the string "PRESS" is contained in the data it GETs from looking at line 23. BEEPs to signal it is ready for something else, and waits for someone to press either <enter>, i.e., CHR$(13), or "Q," i.e., CHR$(81). (The former causes the program to start copying another disk, whereas the latter causes it to return to the main DM1000 menu, since CHR$(14) is the same as FCTN-5.)

The BATCH-IT "manual" contains only three brief examples of batch files, and the longest (18 lines—see pages 11-12) contains two serious errors: there's a missing line (BREAK should be preceded by GOTO MAINMENU), and "MACRO" is used both for a variable name and a label, which is not permissible. One of the other examples is 12 lines long, and the remaining one shorter yet (8 lines). There are two additional examples on the disk not in the 14-page booklet, but (in my opinion) what is provided (the publisher in his catalog claims, "Heavily documented—comes with many sample batch files"!) is not really adequate for this excellent program. In short, the examples are too short, especially since using BATCH-IT to full advantage involves learning a new computer language.

The language includes no less than 29 commands (sometimes "explained" in a sentence or two), divided into five categories: assignment (DEFINE, GET, MOVE, VBASE, VPAGE), control (GOTO, GOSUB, RETURN, ONERROR, ONKEY, ONMATCH, BATCH, PAUSE, END), input/output (CLEAR, PRINT, INPUT, ENTER, KEY, LOOK, WAIT, LOAD, RUN), other functions (PATCH, CHAR, COMPARE, KEYMODE, BEEP). In addition there are two flags (ERROR,

continued on page 470

USUS STRUGGLES TO RECOVER

by Jim Horn

Recovery may not be the word for it. Survival is really the issue for the user group associated with the UCSD p-System, now called "Power System." It may be a measure of this once highly popular operating system software that yet a third term, "UCSD Pascal" is needed before a spark of recognition is raised in many people's eyes, if then.

Older as an operating system than MSDOS, the p-System was the earliest operating system software designed for inter machine portability to gain widespread acceptance. The University of Southern California in San Diego early in the days of the microcomputer persuaded the major manufacturers such as TI and Apple and later IBM to support what essentially was an optimized translator coupled with a compiler which generated a special hybrid code, called "p-Code." The first, and to a lot of people, the only language seen for this system was a version of Pascal, thus the early term "UCSD Pascal." Sadly enough, now that the multi language capability only dreamed of during the heyday of the p-System is here (COBOL, Fortran, Modula II) the world can barely stifle a yawn. At one time "UCSD Pascal" enjoyed name recognition on a par with Borland's Turbo Pascal, not all that hard, since UCSD Pascal predated the Borland product. The language was an early darling of hackers interested in moving to higher level full bodied programming code for micros.

Today, the USUS Newsletter is filled with references to "former glory" (Eli Willner), and "if we are to survive" (Henry Baumgarten) and similar remarks. While other members of the USUS Board of Directors express more optimism, USUS is obviously not in their salad days, and that is the good news, so we can omit the bad news for now. Actually, the soul searching generated by the vigorous new board has resulted in the discovery that the tough 99/4A community is now a significant part of USUS membership and are joining at a rate of about two a week. So reports the September/October issue of USUS Newsletter "All the News that Fits, We Print," edited by Sam'l Bassett, Editor in Chief, and William D. Smith, Associate Editor.

The USUS, or the UCSD p-System User Society, is the custodian of an enormous amount of software developed on Apples, IBMs, TI microcomputers, TI Pros, Strides, Sages, and a multitude of other machines you may or may not have heard of. Yes, as you will read later, Pascal software has been created on your trusty 99/4A plus the TI p-Code card. However, all of these programs, with exceptions will run on your TI99/4A regardless of the machine they were created on. I just received a call from long time TIFORUM member Carl Schuneman (CompuServe Member number 72446,3044) who is working through the huge library of USUS programs with his 99/4A and his p-Code card. If you have a p-Code card and would like to help in this task, give him a shout up on TIFORUM. He might be able to cut you a deal, because there is a lot of work to be done. Most of this effort is checking to see if some of the software does not make abnormal calls to routines written in assembly for another computer. USUS library programs are not supposed to make calls to machine dependent code, but you know how that goes. Also, some software is a template for other programs. In effect, there is a lot of dusting and cleaning to be done with this huge international treasure trove. However, at this stage, software tweaking is optional. Just knowing what will run and what will not represents major progress.

If you prefer to deal through the mail, it is probably best to contact the chairman of the 99/4A SIG, Ken Hamai, who can be reached by writing to the USUS address listed.

Ken recently did an excellent job at starting the process of identifying sources of public domain and shareware p-Code software. USUS library software is not public domain, even though cheaply available to members only, Ken asked to be advised (at the USUS address) of any other sources of public domain and shareware programs, so I am going to repeat his list, adding a few of my own comments. You are free to contact the shareware authors. Hamai suggested a donation of $10 is customary, recommending writing each person in advance to determine how each shareware author handles mailing and disk costs. Write ahead if you wish, but both parties will be firing a lot of paper back and forth. Better, I would recommend you send a check for $10, an initialized disk with a sturdy mailer, and a note inside along with enough loose stamps to get the disk back to you. In your note, tell the author that if he prefers to handle the disk action in another way, just throw the disk back in the mailer and into the mail preferably after putting your stamps on the container, of course. As an added inducement, use some kind of high quality disk, or colored disk. Tell your correspondent he should keep your disk and substitute his own if he has a supply of ready made disks. As you can see, I am big on pampering our software authors.

Now Comes The Orphan Software Project

USUS is also doing what right now may be unique but their project is something which deserves emulation. If you are aware of other projects of

continued on page 472

Texas Instr., TI Forum continued from page 469

MATCH). These commands make it a surprisingly powerful utility. The package is much more versatile, in fact, than I had expected, but I should have known better: Tom Bentley (author of PrEditor) and Charles Earl (author of Telco) had earlier already shown themselves to be very accomplished programmers.

BATCH-IT is intended primarily to work with Editor/Assembler option 5 programs. Since it is running these other programs, it needs a secure place of its own in which to hide, so to use BATCH-IT you will need a Mini-Memory module, a "supercart," or a Myarc 9640. As you can see, it is thus not intended for use with Extended BASIC programs on the TI-99/4A, but another worthwhile Asgard product—EZ-KEYS PLUS by Harry Wilhelm ($14.95)—performs a somewhat parallel purpose in that environment. (This program, an upgrade of the eariler EZ-KEYS, is not really a batch language, but its macros sometimes can accomplish similar results, and it is without question the best "KEYS"—type program that I have seen for the TI-99/4A.)

There are a few assembly programs with which BATCH-IT will not work (e.g., it will not work with some terminal emulator programs, although it will work with a patched Telco, and—at the time of this writing—it will not work within the FUNNELWEB editor, although it will work in many or all other areas of FUNNELWEB). In general, however, this is an exciting program with exciting possibilities, and patches may be found later to enable it to work with those few programs with which it does not work at present. Asgard admits their manual "is not a tutorial—it is a reference" (in my opinion, an incomplete reference with insufficient examples), but it is my hope that these deficiencies will be rectified by Asgard's plans to publish BATCH-IT tutorials and sample batch files in their magazine *KEY NOTES*. At any rate, the program itself is superb and definitely worth purchasing. I highly recommend it! Good programs like QUICK-RUN and especially BATCH-IT ought not to have to suffer as a result of inadequate documentation (non-working examples are particularly unfriendly to the customer!), so it is good news that greater care is expected to be taken with the manual for PRESS, a new word processor by Charles Earl, which may be Asgard's most significant release yet. (As Telco users know, Ruth O'Neill writes excellent documentation.) But more about that program in a future issue! ●