YESTERDAYS NEWS

VOLUME 2 NUMBER 6 Established 2016 JUNE 2017

30 Years Ago...

Historical Information taken From Bill Gaskills TIMELINE

JUNE 1987:

Danny Michael announces his departure from the TI Community in the Shoals 99er newsletter. He has sold his TI-99/4A system and purchased an IBM PC.

John Willforth of the Pennsylvania Users Group (PUG) author's an article about how to put Extended BASIC inside the TI-99 console.

Myarc receives an award from the Front Range 99er Computer Club of Colorado in appreciation for its support of the TI-99/4A computer community.

Word reaches the TI Community that Bruce Ryan of Ryte Data in Haliburton, ONT Canada is going to release a set of EPROMS for the TI DisK Controller which will provide support for up to quad density floppy disKs. The EPROMS are said to be priced at $45.00.

On June 10th the MBP ClocK/Calendar - ADC Kit price is reduced to $31.50. The company producing the hardware peripheral, MBP of 2 Cypress Drive Wichita, KS 67206, says that more than 200 of the devices have been shipped to TI owners worldwide.

Pioneer member of the TI Community, Doyle Bynum, co-founder of Softmail and Texas Peripherals, dies on June 25th in Lancaster, Texas at age 63.

Stu Olsen releases Mass Transfer v4.2 (ymodem capablities added) and v4.3 (same as v4.2 but with support for multiple ram disKs).

Ryte Data's R/D Computing publishes a 64K memory map modification project.

Ryte Data announces that it has a poor subscriber base for the R/D Newsletter, and that operations will cease if more interest is not generated.

Monty Schmidt releases GPL Assembler / LinKer for the 99/4A.

CorComp releases a TI to PC text transfer utility.

Marty Kroll Jr. releases CATLIB v1.5 (CATALOGING LIBRARY).

MICROpendium puts out feelers for a c99 columnist and a Geneve columnist.

Genisis Systems Software announces a freeware offering on how to create TI-Artist fonts. Company owner Lyle Thorogood, a previously unKnown entity in the TI Community, continues in that capacity as neither Genisis nor Thorogood are ever heard from again.

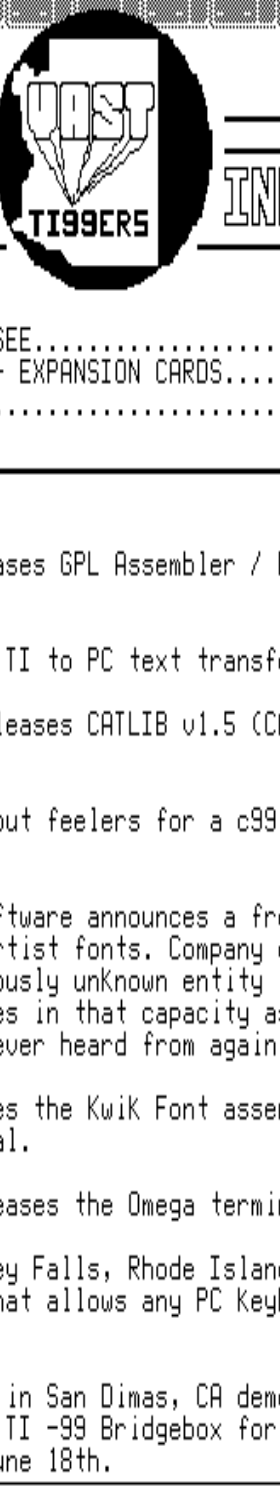
Wayne Stith releases the KwiK Font assembly language program and tutorial.

Travis Watford releases the Omega terminal emulator.

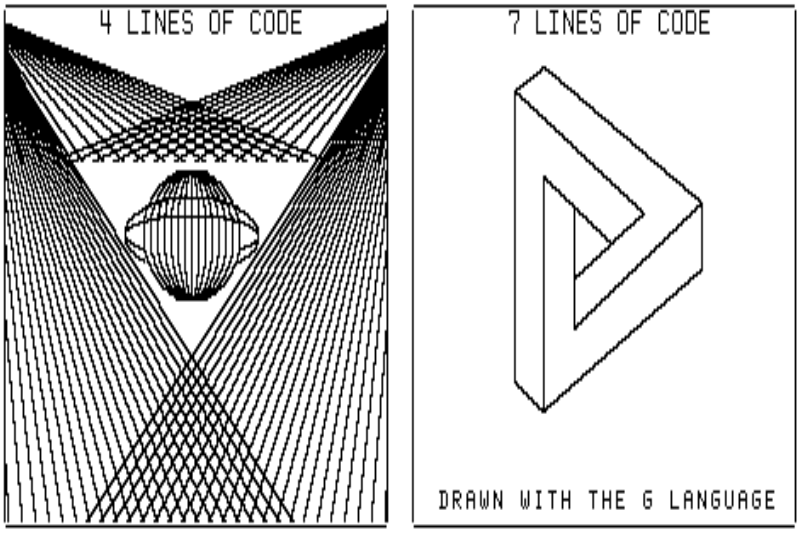
ML Systems of Valley Falls, Rhode Island releases an $80.00 interface that allows any PC Keyboard to be used on the TI-99/4A.

Craig Miller of MG in San Dimas, CA demos the Triton Turbo XT PC Clone with a TI -99 Bridgebox for the User Group of Orange County on June 18th.

INSIDE INFORMATION



to G or not to GEE



G - THE GRAPHICS LANGUAGE
from Adelaide South Australia

By Bob Warren

G can stand for many things including GRAPHICS, GREAT, and GOOD FUN. The graphics language G is all of these things. It was developed by Gene KrawczyK, one of the original members of ATICC, and he has made it available to the Club. It was written in Assembly Language, and so requires a disc and memory expansion to run.

G is a powerful graphics language with simple commands, which makes it ideal for children, but it is also sophisticated enough to provide a challenge for adults. UnliKe TI-ARTIST, which only produces still pictures, G allows a form of animation, and can rapidly change screens. A screen can be STOREd at any time, a new screen drawn, and then the old screen RESTOREd when required. A screen can also be SAVEd to or LOADed from a disc when needed.

I have edited Gene's instructions and commands to produce a beginners version of G which everyone can use. Details of this version are given below. Gene also included some commands which allow an experienced Assembly Language programmer to execute Assembly routines, including sprites(I thinK). When I have sorted these out and written instructions for their use, I will present an ADVANCED G.

I hope there will be sufficient interest for a regular column in our buletin to be devoted to G, both to answer queries and to exchange programing ideas.

THE UNDOCUMENTED FEATURES OF G

by MarK Schafer

Bytemonger, 1990 April, p3.

G is a graphic programming language written in machine language for the TI. It comes on disK with on-disK documentation.

However, I noticed that there is more to G than the documentation shows. What I did was looK at the program with a sector editor, and on the third and fourth sectors is a list of all the Keywords in G along with the syntactical symbols. Some of these were not described in the documentation. I set out to find out what they do.

First, a list is in order; these are the commands that I did not find in the docs:

GET, PUT, TEST, VPEEK, VPOKE, PEEK, POKE, VWTR, SARRAY AND LARRAY.

PEEK AND POKE
Starting with the most obvious, PEEK and POKE do just what CALL PEEK and CALL LOAD (with memory) do in Basic. They read and write values in memory respectively. Their syntaxes are as follows:

PEEK A VAR [VAR,VAR...]
POKE A B [C D E...]

Lower-case letters can be any numeric exprssion; var is any G variable. In other words, both commands can read/write in consecutive locations.

If you want to peeK or poKe into an address higher than 32767, you must subtract 65536 from it, just as you do in Basic. And, as far as I can tell, all values are in decimal.

VPEEK AND VPOKE
These do the same as PEEK and POKE but in VDP RAM. This enables you to read and write to the screen. Sure, you can write to it now using the documented commands, but only VPEEK can read it.

VWTR
VWTR is the next most obvious for you Assembly programmers since it happens to match the Assembly mnemonic for VDP Write To Register. However, this VWTR taKes only one argument and writes it to VDP Register 0, so it's not very useful.

GET AND PUT
For anybody familiar with IBM Basic, there's GET and PUT. No, they don't read and write files; they read and write

graphics, which is their other use in IBM-land. Their syntax is:

```
GET A B C D
PUT A B
```

GET's syntax matches that of CLEAR and INVERT. That is a,b is the upper left corner of a box, and c,d is the lower right corner. This area of the screen is buffered. When the PUT command is used, the last graphic that was buffered with the GET command is put on the screen with its upper left corner at the location given (a,b). Limitations apply, however. These commands will work perfectly if a, b, c, and d are all divisible by 8. If they are not, you may not get the exact area you specified.

SARRAY and LARRAY

SARRAY and LARRAY may be unique in the world of programming. The S and L stand for Save and Load respectively. So their argument is a file name.

SARRAY allows you to save the array (there's only one, remember) to disk. How much does it save? All 500 values, except the zero element (@(0)). The file takes up 15 sectors on the disk.

LARRAY loads in a previously saved array into the array in the program. These can be useful forprograms that have a large number of array constants.

TEST

And then there's that testy TEST. I haven't got this command figured out yet, but its syntax is:

```
TEST A B
```

a and b look like they can be any value. Small values don't seem to do anything. But weird things happen on certain values. You could compile a catalog of the effects. Mine starts out like this:

```
TEST 1195 1127    MESSES UP VIDEO.
TEST 125 1127     STRIPES THE SCREEN VERTICALLY.
TEST 196 1125     REPEATS FIRST 28 CHARACTERS IN FIRST 8
                  ROWS
TEST 128 -128+C   COLORS BACKGROUND COLOR C (0-7).
TEST 256 -32000   COPIES TOP THIRD OF SCREEN TO OTHER
                  TWO THIRDS
```

Some of these may have different effects according to what is already on the screen. It appears that this command is multi-purpose.

There are also symbols left out in the docs. They are the percent sign (%), comma (,), tilde (~) and number sign (#). I have figured out what two of these do.

The percent sign (%) is highly useful. It represents the CHR$ function in Basic. It will return whatever character corresponds to the ASCII value of the numeric expression that immediately follows it. An example is in order I think:

```
PRINT 0 0 %80
```

will put a P in the upper left corner of the screen since P is ASCII 80. But of course, % is more useful when a variable follows it. This allows you to echo user input to the screen.

The tilde (~) is just a variable. You can set it equal to something, print out its value, use it in a formula, or whatever you do with a variable. However, it shares a feature in common with the array: its value does not change from one run to the next. Therefore, it contains garbage if not set. The other variables are initialized to zero.

I have no idea what the other two symbols do. I've tried them in every way I can think of to no avail. My best guess is perhaps some future use.

There's still another interesting point to cover. Notice that G prints text. This is unexpected because G uses Bitmap Mode, which doesn't have text. So the G program must be converting the graphic definition of the characters to bitmap mode so they can be displayed. So where is it storing the character pattern table? In Low Memory expansion starting at >2000 (decimal 8192). This is where the space is located. The exclamation point follows that starting at >2000, and so on. They're in the same format as they are in Basic. But you have to remember, if you play with this area of memory, you must use decimal byte values. So the range is 0-255. If you POKE into this area of memory, you can change the look of the characters. And when you do, not only will it affect your program, but the editor as well. Any font change will be permanent until you change it again. The editor is in Text Mode, so it will only show the first six columns of each character.

Something you can do for fun is subtract the values in this area of memory from 255, and you will invert the character set. Or you can just do parts of it. This will allow you to put inverted text on the screen easily. Then you can change it back to regular text because, unlike Basic, changing a character's definition will only affect future displays; it will not change the looks of the ones already on the screen.

To change the definition of character X, use the following POKE statement:

```
POKE X*8+7936 A B C D E F G H
```

See "GEE", Page 3

GEE continues...

If you already Know what X is, change the formula to the appropriate constant. X is in the range 32-127.

While I'm on the subject of memory, let me just mention that the zero element of the array is stored at 9184. The 1 element is after that, and so on. Each element taKes up two bytes. The value of the tilde is stored at 9180. So another way to refer to it is @(-2). G does not checK for negative values of the array, so if you go low enough, you can change the character set by setting negative values of the array.

When you've got PEEK and POKE, you've got power. You can use these to access the joysticKs, scan other Keyboards, and even pronounce vocabulary words in the Speech Synthesizer! Watch out, though, if you write a program that accesses the joysticKs. G is constantly checKing to see if the user hits FCTN 9. When it does that, it does not change the Keyboard to be scanned, so it won't worK if you've changed it. You could immediately change the Keyboard backK to zero when you're through scanning it, or, type Z to escape while scanning joystickK 1 or N while scanning joystickK 2.

We still can't do sound or sprites in G since we can't control the interrupts. You can produce a sound, but you won't be able to stop it!

No article about a programming language is complete without a program, so I've included one here.

The name of this program is Display Races. It races color bars across the screen and gives you race results.

The three types of races are quicK, front-runner, and closer. The differences will be apparent when you run the program. The number of intervals can be from 0 to 9 and represents the number of calls the race will have. The increment can be from 1 to 8 and represents how many columns the racers will advance at one time.

The first three questions are all different colors just because it's so easy to do in G. In Basic it isn't worth the trouble.

At the end, the program will give you the order in which they arrived at each call and asKs you if want it inverted. If you do, then it will show you the order of finish with the position each racer was at each call, allowing you to see their progress. Each line will be in the color of the racer easily allowing you to follow across. You can forget about doing that in Basic.

I realize this article is long overdue, and maybe you've already discovered some of these things, but at least the program is new to you.

RANDOM GARBAGE

Boston Computer Society TI 99/4A User Group August 1987
By J. Peter Hoddie

DISK CONTROLLERS

There are 3 main disK controllers out there, made by TI, CorComp, and MYARC. They all vary in their capabilities, so let me first list the different features a disK controller can have. All disK controllers can support double sided disK drives. This means that if you have drives that can store data on both sides of the disKette, any of these controllers is capable of using that feature. Many disK drives can support what is called "double density." This is a method for packing double the data onto a disK. Most disK drives these days are 40 trackK, which is standard, while some can support 80 trackK. 80 trackK means that you can store twice the amount of data as on a 40 trackK disK. However 80 trackK drives usually require more expensive disKettes because the data is so compact on the surface of the disK. DisK drives vary in the time it takes them to access data. The slowest speed is about 20 milliseconds (MS). There are drives that are as fast as 3 MS, although these are more expensive.

The TI disK controller can handle up to 3 double sided, single density, 40 trackK 20 MS step time disK drives. In otherwords, the TI disK controller is the bottom of the line in all respects. Ryte Data current has available an EPROM set for the TI disK controller that will allow it to access 80 trackK drives, however I do not Know enough about the product at this time to maKe any comments on it.

The CorComp disK controller can handle up to 4 double sided, double density, 40 trackK 20 MS to 6 MS step time drives. This means that you can have one more drive than the TI controller, and each drive can hold double the data. The drives can also be accessed faster. The CorComp disK controller has some nice extra features including a good disK manager (it was the basis for DM1000), and a number of extra CALLs. The disK controller literally takes over the computer on power up, however, which causes some compatibility problems. This can be fixed by purchasing a new EPROM set from MG for about $35.

The MYARC disK controller can handle up to 4 double sided, double density, 40 or 80 trackK, 20 MS to 6 MS step time drives. This is essentially the same capability as the CorComp card. You can only use 80 tracks if you purchase a special EPROM from MYARC for about $50 that supports 80 trackK drives. The MYARC disK controller comes with probably the best disK manager program for the /4A, and has a built in CALL DIRectory command to catalog disK from BASIC and Extended BASIC. Also, the MYARC disK controller is noticably faster than the others because of the approach MYARC tooK in designing the card.

RS232 Cards

There is very little to say in this area. There are cards available from TI (very rare these days), CorComp amd MYARC. They all have 2 RS232 ports and 1 parallel (PIO) port. The CorComp will not work with the MYARC print spooler (more below on that), whereas the MYARC and TI will. The CorComp has what some describe as a "Kludgy" PIO port, however it works as well as the others. The MYARC supports some extra software commands to allow for 19.2K baud (the others stop at 9600, real slow <grin>, inverted Busy in software rather than hardware, and some other details. MYARC also has an EPROM that will make the PIO port act like the thermal printer (TP) if you need something like that. However, really all these cards are pretty much the same. Most people prefer the TI card, and shun the CorComp. The MYARC is probably the best and most readily available these days.

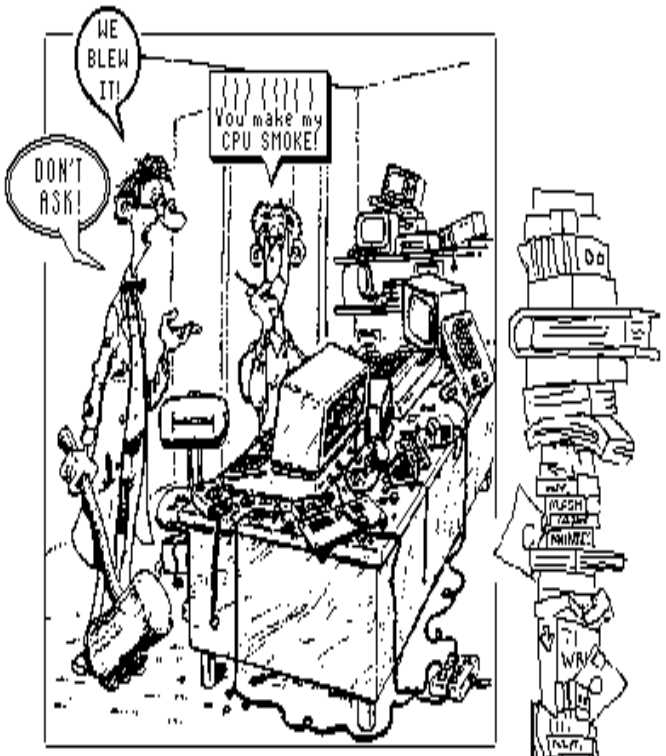
MEMORY CARDS and RAM Disks

There are more memory cards out there then almost anything else. TI made a 32K momery card. That was it. Most RAM disks, but not all, replace this card. If you just want a 32K card, they are available from MYARC and CorComp and there is on difference worth discussing between these two cards. They both seem to work reliably. Foundation made a 128K memory card that replaced the 32K memory expansion and gave you an extra 96K of memory that could function as a RAM disk. Unfortunately their RAM disk software was terrible. Quality 99 software and others have since released new software that makes this card acceptable, however since it is out of production it can't really be strongly recommended.

MYARC makes a memory card which replaces the 32K card, and comes with either 256K or 512K of memory. The memory beyond the first 32K can be divided between a RAM disk and a print spooler, although the print spooler will not work with the CorComp PIO port. For an additional $50 or so, you can get MYARC Extended BASIC II, which is a much faster, more powerful, and slightly buggier version of Extended BASIC that will work with the Foundation card or the MYARC memory card. CorComp makes a 256K and 512K card, and these both function as RAM disks. I don't Know much about these cards, except that they are reported to work quite well, so again I will make no comments.

The Horizon RAM disk comes either as a kit or assembled, and provided a very reliable 90K or 180K RAM disk. It can also be upgrated to 256K. It supports a very powerful operating system including replacing the TI title screen with a custom menu of programs. The Horizon RAM disk does not replace the 32K card and thus can be used along with a MYARC or Foundation RAM disk.

CAPTION ACTION



OMEGA:

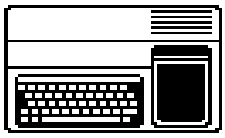
"SD" card! Not a "CF" card!!

SPARKDRUMMER:
(borrowing from Matt)

It worked...

... and then it crashed

TI GUY
by CIRO



Yesterdays News Information

Yesterdays News is a labor of love offered as a source of pleasure & information for users of the TI-99/4A & Myarc 9640 computers.

TI-99/4A HARDWARE

BlacK & Silver computer
Modified PEB
WHT SCSI card with SCSI2SD
Myarc DSQD FDC
Myarc 512K Memory Card
Horizon 1.5 meg RamdisK
TI RS232 card
Corcomp Triple Tech Card
1 360K 5.25 floppy drive
1 360K 3.50 floppy drive
1 720K 5.25 floppy drive
1 720K 3.50 floppy drive
80K Gram KracKer
Samsung Syncmaster 710mp

TI-99/4A SOFTWARE

PagePro 99
PagePro Composer
PagePro FX
PagePro Headline MaKer
PagePro Gofer
TI Artist Plus
GIFMania

PC HARDWARE

Compaq Armada 7800 NotebooK
Compaq Armadastation
Samsung Syncmaster 710mp

PC SOFTWARE

Dead,Dead,Dead Windows 98se
FileCap
prn2pbns
Irfanview
Adobe Distiller
Adobe Acrobat

Yesterdays News is composed entirely using a TI-99/4A computer system. It consists of 15 PagePro pages which are "printed" via RS232 to PC to be published as a PDF file.

Yesterdays News
c/o SparKdrummer
AtariAge forum
Phoenix, AZ. 85027