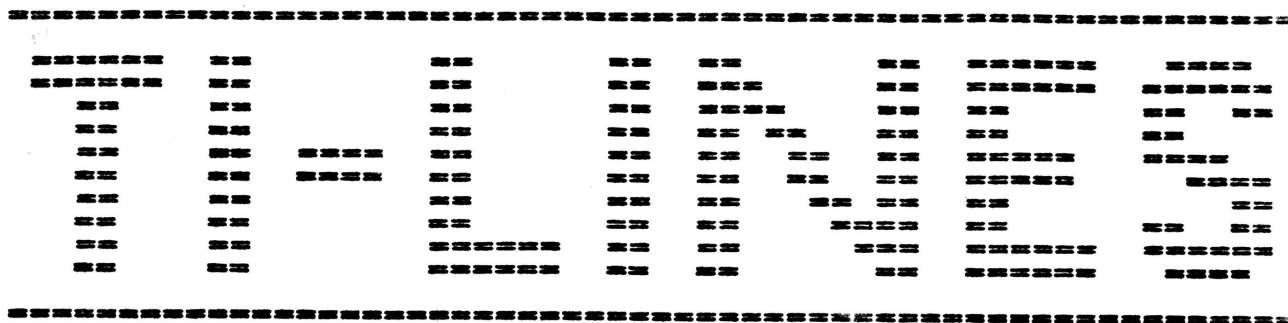


H A P P Y N E W Y E A R



Volume 1, Issue 10

January 1st., 1985

TI-LINES is the monthly newsletter of OXON TI USERS, and the form of presentation of any material remains the copyright of each individual author. OXON TI USERS reserves the right to submit any material to other newsletters for publication, with due credit to the author(s) concerned. Submissions are accepted on this understanding.

It is the responsibility of each author to ensure that no copyright infringement will occur by the publication of their articles in TI-LINES, and OXON TI USERS cannot be held responsible for any such infringements. Every effort is made to ensure that any information given in TI-LINES is correct.

TI-LINES is produced and published by:

Peter G. Q. Brooks
61 The Avenue
Kennington
OXFORD
OX1 5PP

Tel: OXFORD 730044. Please do not phone after 10 pm

Oxon residents are offered a subsidised subscription at £1.56 p.a., payable either as 12 second class postage stamps or by cheque.

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. Overseas subscriptions are by arrangement. Back issues are £2 including post and packing. New subscriptions begin with Issue 1 of the current volume, up to and including the current issue, regardless of the number of issues elapsed.

Contributions should be submitted either on diskette in TI-Writer compatible files, or in a form which is as legible as possible. Art work should fit within an A4 area and should not contain colour. Very high contrast line drawings are preferred, and these may be produced by arrangement with the publisher.

I n d e x t o a r t i c l e s

P a g e

EDITORIAL.....3

Gibbering. Faster COINCS. TI Forth. Sounds Familiar ?
 News & Views. Anorexic TI-LINES. A word or two from Outside.
 Update. Shock! Horror! PCW admits that 4A exists!
 Knocking on...

BOOK REVIEW.....8

Ira McComic's book on Assembly Language, reviewed for you by
 DAVID NEALE of the TI-99 Gebruikersclub Vlaanderen

DAVID BROWN.....9

After a nasty accident with a cassette tape, David's back
 (How about a few letters to David from the younger TI Users ?)

CONTACTS & BULLETIN BOARD.....11

Hungarian contacts in Baghdad

A LETTER TO THE TI*MES.....12

ALLEN BURT of TI-99/4A EXCHANGE on the subject of CTRL U and
 the transmission of control characters to his printer

ENHANCED BASIC - CALL H() GOES ON.....14

A deeper discussion of the HEADER subprogram. This will be
 a sticky one. Well, you can't say you weren't warned...

CLOSE FILE.....18

Hamming it up. RACTER and ELIZA Doolittle ? No, but the
 principle is almost the same. Fractal Graphics. Fancy your
 chances playing with WORMS ?

FEBRUARY'S ISSUE IS ALIVE AND WELL AND LIVING IN OXFORD...

E D I T O R I A L

GIBBER, GIBBER

Here am I, entering the New Year late and with a backlog of work as long as my overdraft, when I sit down to compose the (late) January issue.

It is my practice to quickly check back through previous issues before beginning an Editorial in order to seek out any items which have been updated in the intervening month, and invariably I come across at least one spelling mistake, or a misprint, neither of which was picked up during the proof-reading the month before.

Part of the way into the CALL H() article I spotted the dreaded PRINT and INPUT with pound signs instead of hashes, for which I apologise. The odd thing is that I seem to be the only one who notices these things which makes me wonder how many copies of TI-LINES end up on a nail in the smallest room down the bottom of the garden...

A couple of pages later, and I found 'program' instead of 'computer' on page 15. Duch. Maybe I ought to write a spelling checker for my TI-Written files - the only drawback is that most of the time I use slang and sometimes deliberately mis-spelled words, which would make such a task nearly impossible.

~~~~~  
SOMETHING NEW EVERY DAY...  
-----

While reviewing readers' submissions for Home Computing Weekly recently (I managed a marathon overnight stint from 3.30 pm through to 11.30 am), I came across a small tip from one DARYL COATES, a 14 year old from Stoke-on-Trent. He suggests that when using sprites in Extended BASIC it is faster and more reliable to use CALL PEEK(-31877, A) and then test (A AND 32) for equality with 32, as a substitute for CALL COINC(). I am a novice in such matters - although I have had my XB for a couple of years now, I have still not found the time to write anything extensive (no pun intended) with it (hangs head in shame). This may therefore be no news to you at all; however it was to me.

~~~~~  
TI FORTH

I have had TI Forth (Public Domain) for a while now without really being able to delve very deeply into it. My last experience of Forth was on the now-defunct Jupiter Ace, and on balance I must say that I prefer TI Forth. When and if I can find the time, I want to run a series on this marvellous language in TI-LINES. It will form the basis for one of a series of booklets which I have planned, to go out under the banner of QUINSOFT, my tentative publishing venture. If anyone has something which they want publishing, 25,000 words or so, and it is informative, please get in touch with me so that I can discuss outlines etc. with you. I have provisionally listed 30 titles, and I hope that the first booklet will be out in the Summer (retailing for under £3.50).

Recently a whole crop of Forths have appeared, but as far as I have been able to discover, they are all slightly different versions of the good old TI Forth.

An interesting thing about TI Forth. While wearing my TIHOME hat I am able to supply the disk and manual for £13.50 (which doesn't cover my costs, but then it is not intended to make a profit) and I therefore copy from the master disk whenever I receive a request for a copy.

Odd, then, that when I copied the Forth on my single-sided disk onto a formatted, double-sided disk, the Forth would not load properly. The usual "BOOTING..." information did not appear, and the menu did not follow on screen either. I vaguely remember, quite some time ago, a newsletter saying that Forth could not run from a double-sided disk.

Does anyone have any enlightenment ?

Incidentally, there is a bug in Forth which has been published before (in MILLERS GRAPHICS I believe) and which I had forgotten completely until I checked a copy which I was producing for recently-joined OTIUser RICHARD SIERAKOWSKI (through TIHOME). There is a typographical error on screen 72, about 6 lines down, where PAB_ADDR has been used instead of PAB-ADDR. It is a simple matter to boot Forth, load -64SUPPORT to get the 64 column editor, then 72 EDIT and change the "_" to a "-" (and also change the RS232 baud rate too while you're there, if you need to) and then FCTN 9 to escape from the editing, FLUSH, and Bob's your uncle.

If you don't do this, then when you attempt to load -PRINT or -FILE or indeed anything which uses SWCH (which defines the printer via the RS232 and also uses PAB-ADDR), then the Forth system will fail to LOAD the screen properly, throwing up PAB_ADDR ? in the process. MALCOLM HEDLEY has also received Forth from TIHOME, and he too had come across the error and notified me about it within half an hour of my finding it for myself.

TI*MES (newsletter of the expanding TI-99/4A EXCHANGE group) recently carried an article on Forth by PHILLIP MARSDEN, together with a routine to format disks in Forth. I quite like the power and speed of Forth, the more so since you can easily modify the basic system to add in your own commonly-used definitions. It's a little like having a BASIC that allows you to add your DEFINED functions to the built-in operating system (thus avoiding the need to type them in at the start of any program using them). I hope to begin the preamble to Forth later this year.

~~~~~

NOW, WHERE HAVE I HEARD THAT BEFORE ?

-----

January's YOUR COMPUTER carried a piece describing a micro thus:-

..."it could have made the (deleted) look like a worthwhile purchase rather than a deliberately underspecified machine designed to milk the purchaser for every last penny on peripherals."

No, you guessed wrong. It might have been the 4A, but it wasn't. It was the Acorn Electron. But it does have that certain ring to it...

~~~~~


NEWS AND VIEWS

From the Trade magazines comes information of a new onslaught by both ATARI and COMMODORE which looks set to make the price and facilities of the Sinclair QL look clumsy and gross by comparison. For about \$200 or less (sometimes much less) Atari unveiled a 64K - 128K micro which is portable, has a built-in 5" monitor, a 3.5" disk drive, a built-in music synthesiser, and runs at a startling speed with stunning graphics and all on the simple 6502 cpu - the same as that behind the Apple II, the BBC A and B and Electron, and the Commodore PET. All in all, Atari have launched a good half dozen or more machines, most vying with high-end machines like the Apple Macintosh for speed and facilities. And all costing much less than their current counterparts in the marketplace (it sometimes makes me wonder why I haven't changed machines!).

The new Commodore 128 has also been on show, with 128K RAM, an 80 column screen, capable of running CP/M (Control Program for Microprocessors - originally based on the Z80 cpu; the idea is that software written under this system is "portable", i.e. it can in theory be run on any machine which supports CP/M...MSX where are you? CP/M covers almost all of the most popular business software in existence, despite the fact that it is a horribly user-unfriendly environment) and for less than \$250.

Readers of the late Tidings might remember part of a Babble which concerned itself with video disk drives and the immense storage capacity which they confer. Now SHUGART have come up with the Write-Once laser drive, which allows you to write, but not re-write, your data to a disk with a 1Gbyte capacity...or 1024Mbytes or 1048576K!!!

At a cost of £5000 it is beyond the average 4A owner, but by the end of the decade Shugart reckon that the 1G drive will be readily available (provided we hang onto our micros long enough?). It is called the OPTIMEM 1000, in case you want to make a note in your 1999 planner...

However, running faster than an optical disk as far as data transfer goes is a goodie from Hitachi. It uses 500 (yes, 500, it says here) 8" floppies crammed into what is supposed to be a desk-top drive, and it handles 5Gbytes of data. Because it uses floppies it is much cheaper than the optical disk (see above). It will be interesting to see who wins through in the end.

How's this for a step forward? Can you guess what the next step in printers is going to be? Laser printers for the home, perhaps? Well, not quite. It's LCD. Yep, Liquid Crystal Diode, as seen on your average wristwatch. Take one light-sensitive drum, incorporate it into an advanced printing mechanism (not dot matrix, it seems) and place an LCD sheet between a light source and the drum. You get a printer which performs similarly to a laser printer as far as speed goes (which could be up to tens of thousands of characters per second) but is much, much cheaper. The PERSONAL COMPUTER WORLD item which discussed the new offerings from CASIO and EPSON termed the LCD sheets "shatters" - I think they mean "shutters".

Either way, an LCD printer can produce high quality and high density, and it prints a full page at a time (hence presumably the comparison with laser printer speed). The products are scheduled for a Spring launch. (Boingggg!)

And finally Cyril... CASIO are at it again. Know what a "Casio data bank read sensor" is ? It's a wristwatch which can recognise numbers and handwritten characters. Where are they written ? On the watch face with your finger... It has a 1K memory, arranged to store 50 names and phone numbers (8 letters, 12 digits). It is also a calculator, using the same character-recognition technology. It costs \$70, and is an advance on Casio's earlier character-recognition wrist watch, the existence of which I was totally unaware until now.

~~~~~

### SLIMMING DOWN

-----

As I pointed out last issue, future TI-LINES will not be as fat as they have been to date, as I have less and less time to write about all the things I want to. I'm still going to write about them, but it is going to take me a lot longer to come up with the articles. That places the onus for material on the shoulders of OTIUers, unless I am to simply copy wholesale from other newsletters. I am loath to do that, quite apart from the fact that it infringes their copyright. It would make my life a lot easier, but it would also mean that the same information (and the same mistakes) would be perpetually cycled round and round, with only one or two real sources of fresh information.

~~~~~

OUTSIDE INFLUENCE

Elsewhere in this issue there is a short piece from ALLEN BURT, who is not an OTIUer but who sent me a copy of his article which he submitted to CLIVE SCALLY of TI-99/4A EXCHANGE. Allen and I met some two years ago and we still keep in touch occasionally. Allen and RICHARD BLANDEN are, I believe, researching heavily into the 4A, and I hope to make a visit sometime between now (January 20th) and the end of February, in order to distil some of the results of their efforts and present them in TI-LINES.

There is also a book review from DAVID NEALE of the FLEMISH TI99 OWNERS CLUB which was kindly supplied in translation. Unfortunately David's original suffered at the hands of the GPO: his copy was on thermal paper and the GPO obviously use some kind of "hot" marker, because there were small dark grey dots all the way across his review. I have therefore retyped it to make it legible again.

~~~~~

### UPDATE

-----

During my (late) writing of this editorial I came across some bits and pieces in PERSONAL COMPUTER WORLD's February issue (and I haven't even got my January issue off the floor yet!) which might interest you.

Information Storage, a firm from Colorado Springs in the States, has just announced a 5.25", 100Mbyte, Write-Once optical disk drive. For those of you with disk systems, forget the paltry 96 tpi to which we aspire (all right, so maybe we don't; stop ruining the story), this little baby has 14,000 tpi. Imagine using Disk Manager 2 to try and initialise that!!! Prices are expected to drop to \$500 by 1986. Yes, I know that \$500 is ten times what you paid for your console. Don't be a wet blanket...

MARK LEE of the Maltby Users Group has had his name blazoned across the pages of PCW (monthly - not to be confused with the upstart Popular Computing Weekly, who have stolen the real PCW's initials. Well, made it more confusing for Initials Buffs like me -) in Rupert Steele's ACC NEWS section. In fact, PCW have taken the unprecedented step of actually publishing a program for the TI in the same issue. Yep, even I had to look twice and hack a leg off to make sure that I wasn't dreaming or suffering from hallucinations brought on by working 36 hours straight on the TI last weekend...

I remember MIKE O'REGAN battering for a year on the PCW doors, trying to get them to accept one of many of his submissions to them. The lucky chap this time is STEVE HUNT (I'm not familiar with his name, but a hefty pat on the back for him), and I don't think I'd be giving any secrets away by saying that, according to PCW's published rates of pay and my calculations, well, Steve must have walked away with £250 for the listing. Well, he will walk away with it, when they pay him. From my own experience with PCW, they take so long to cough up, most of the currency has been taken out of circulation...

~~~~~  
MRS WHITEHOUSE, WHERE ARE YOU ???

Well, there I was, quietly minding my own business (disassembling all 64K of memory with the TI Debugger just for a lark. Coo, don't it make the print-head hot ?) when a large envelope drops through the letterbox. (The time-scale is not accurate here, 'cos the postman comes at 8.30 am, and I was disassembling at 6 pm the next day. It's something called poetic licence. Your local Post Office has the application forms...)

Lo and Behold, tis the latest copy of TI*MES, and very welcome too, but, oh, what is this ? Do my eyes deceive me, or do my eyes deceive me ? . Brian. It is a naughty calendar, printed out using Extended BASIC (I could make a very, very rude joke here, but I'm not going to because you are all streets ahead of me) and entitled FOXY. Swoon, swoon. It is now hanging on the wall in my bedroom/study/dining room/Think Tank, and everyone who comes in admires it and asks me if I can do them a copy on my machine. Alas, no. Ahem. Er, I mean, no, of course not, disgusting little bleeders, what do they think I am...

~~~~~  
FEBRUARY'S TI-LINES  
-----

Because January's issue is late, this will have a 'knock on' effect for some issues to come, but I do not expect it to cause difficulties beyond MAY 1985. (Famous last words...) I can only apologise profusely and abase myself on the floor. If it helps any, this is being typed after yet another marathon stint - 7.30 am Tuesday to 11.55 am Wednesday...

-----  
B O O K   R E V I E W  
-----

D a v i d   N e a l e

J a n u a r y   1 9 8 5

LEARNING TI-99/4A HOME COMPUTER ASSEMBLY LANGUAGE PROGRAMMING  
-----

A new book by IRA McCOMIC

Paperback, 331 pages.

\$16.95

The fortunate owners of a fully expanded system who are able to use Editor/Assembler frequently complain about the standard of the Editor/Assembler manual. Mini-Memory owners have even more problems with the accompanying handbook.

This new book by Ira McComic offers an excellent solution to most of these problems. The reader is eased into Assembler by examining other languages, the structure of data, the structure of Assembly Language statements, the various addressing formats and the use of the Loader and Debugger.

This introduction is followed by a series of chapters in which the various Assembler instructions are examined in detail. Each chapter discusses a set of logically-related instructions such as Data Movement, Compare, Jump and Arithmetic Instructions, amongst others.

These chapters are followed by two chapters which discuss other aspects of Assembly Language (Expressions, Relocation, etc.) and Machine Code Formats.

Three appendices contain useful information. The first is a collection of instruction summaries in which detailed information about each instruction is listed alphabetically according to the corresponding mnemonic code. The second appendix contains Number conversion tables and the final appendix lists the ASCII characters and their corresponding binary, hex, and decimal codes. An Index is also included for easy reference.

"Learning TI-99/4A Home Computer Assembly Language Programming" is an excellent book and almost a MUST for anyone wishing to delve into the more rewarding regions of the 99/4A. Beginners will not be frightened off by a load of jargon and unnecessarily complex explanations; more experienced programmers will find the book a more than useful reference.

The book is published by WORDWARE PUBLISHING INC., 1104 Summit Avenue, Suite 104, Plano, TEXAS 75074, U. S. A. and costs \$16.95. I am not aware of its availability in the UK.

~~~~~

Hello again, I'm sorry about this column being missing from last month's issue of TI-LINES. Also missing from this issue is the promised program "The Pink Panther" This is the reason why there was no column in last month's issue-the tape with the program on it snapped and I refused to spend another six hours going through the sheet music working out the correct values etc(especially since I can hardly read music). Because of this, I was stuck for an idea. Instead, I am starting a short series on graphics programming-no promises this time. This issue has a program included that will use the TI's graphics capabilities to draw a 3D cube on the screen...

```

100 CALL CLEAR
110 CALL SCREEN(2)
120 FOR I=1 TO 16
130 CALL COLOR(1,16,1)
140 NEXT I
150 FOR X= 96 TO 116
160 READ A$
170 CALL CHAR(X,A$)
180 NEXT X
190 DATA 010618608,8060180601,0000000001061860,0000000080601806,00000000106186
200 DATA 01063608,8,01,609886818080808,000008060180601,000000000000008
210 DATA 0000000000000001,0000000106186080,061860800,8080808080808080,60180601
220 DATA 0101010101010101,80808080818698E,010618608,0101010181611907,
230 DATA 0619618101010101
240 FOR I= 1 TO 46
250 READ A, B, C
260 CALL HCHAR(A, B, C)
270 NEXT I
280 DATA 6,8,96,6,9,97,6,7,98,6,10,99,7,5,100,7,6,101,7,7,102,7,10,103
290 DATA 7,11,97,7,12,99,8,5,104,8,6,105,8,7,106,8,10,107,8,11,108,8,12,109
300 DATA 9,5,110,9,7,111,9,8,105,9,9,108,9,10,109,9,12,112,10,5,110,10,8,112
310 DATA 10,9,110,10,12,112,11,5,110,11,8,112,11,9,110,11,12,112,12,5,97,12,6,99
320 DATA 12,8,112,12,9,110,12,11,98,12,12,96,13,6,103,13,7,97,13,8,115,13,9,113
330 DATA 13,10,114,13,11,102,14,9,102,14,8,103,8,12,116,13,6,103
340 GOTO 340

```

.....

There are a number of different commands for graphics available to the TI user these are:-

CHAR*	DELSPRITE	PATTERN
CHARPAT	GCHAR*	POSITION
CHARSET	HCHAR*	SCREEN*
COINC	LOCATE	SPRITE
COLOR*	MOTION	VCHAR*

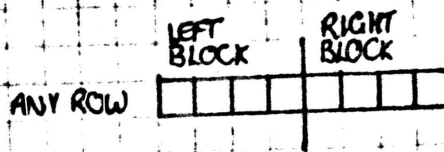
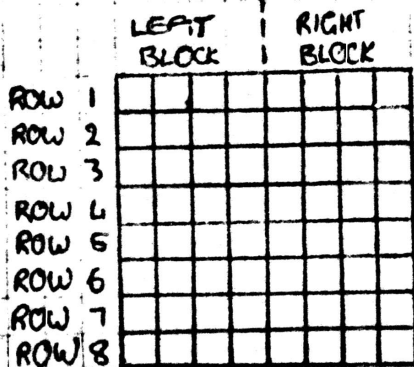
The ones with a dot after them are the statements that can be used in either BASIC or in EXTENDED BASIC, the rest can be used in Extended Basic only.

Here is a quick explanation of what each statement does (refer to the manuals for further details).

CALL CHAR(ASCII code, hexadecimal code)

This subprogram allows you to define special graphics characters. ASCII code can be a number between 32 & 156 (32 & 143 in XB). The hexadecimal code is a string which can be up to 16 characters in length, this defines the character. Each character is made up of 64 dots or PIXELS comprising an 8 by 8 grid as shown below

Each row is partitioned into two blocks of four pixels each...



To create a new character, you specify which pixels to turn off and which ones to turn on. The computer uses a binary code to show which of the pixels are on or off but we type in a hexadecimal code. The table below shows all the on/off conditions that are possible for the four dots in any one block.

	0 0 0 0	0
	0 0 0 1	1
	0 0 1 0	2
	0 0 1 1	3
	0 1 0 0	4
	0 1 0 1	5
	0 1 1 0	6
	0 1 1 1	7
	1 0 0 0	8
	1 0 0 1	9
	1 0 1 0	A
	1 0 1 1	B
	1 1 0 0	C
	1 1 0 1	D
	1 1 1 0	E
	1 1 1 1	F

CALL CHARPAT(ASCII code, string-variable)

The CHARPAT subprogram returns in string variable the 16 character code that specifies the pattern of ASCII code. It is the reverse of the CHAR subprogram.

CALL CHARSET

This subprogram restores character patterns back to their original color and there normal values if the ASCII value of the particular character is between 32 and 95.

CALL CLEAR

This clears the screen.

CALL COINC()

The COINC subprogram detects a coincidence between a sprite and another sprite or a position on the screen. The value returned is -1 if there is a coincidence or 0 if there is no coincidence.

CALL COLOR

The COLOR subprogram allows you to specify the foreground color of a sprite or the foreground and background color of a character set. In one statement, you can define the colors of sprites or character sets, but not both.

CALL DELSPRITE()

This subprogram removes sprites from further access by a program. After been deleted with DELSPRITE, a sprite can be redefined using the SPRITE subprogram.

CALL GCHAR(row, column, numeric-variable)

This subprogram reads a character from anywhere on the screen and returns the ASCII value for the character in the locations given in row and column through the numeric-variable.

CALL HCHAR(row, column, ascii code, repetition)

The HCHAR subprogram displays a character anywhere on the screen and optionally repeats it horizontally. If the repetition parameter is missed out then only one character will be displayed.

CALL LOCATE(sprite number, dot-row, dot-column)

The LOCATE subprogram is used to change the location of the given sprite(s) to the given dot-row(s) and dot-column(s).

CALL MOTION(sprite-number, row-velocity, column-velocity)

The MOTION subprogram is used to specify the speed at which a sprite travels. If the row and column velocities are zero then the sprite is stationary.

CALL PATTERN(sprite number, character-value)

The pattern subprogram allows you to change the character pattern of a sprite without affecting any other characteristics of the sprite.

CALL POSITION(sprite number, dot-row, dot-column)

This subprogram returns the position of the specified sprite(s) in the given

dot rows and dot columns.

CALL SCREEN

This changes the color of the screen to that which is specified. All portions of the screen that do not contain characters or have characters or bits of characters that are transparent are shown as the specified color.

CALL SPRITE(sprite-number,character value,sprite-color,dot-row,dot-column, row-velocity,column-velocity)

The SPRITE program creates moving graphic characters called sprites. They can be set in motion in any direction and at a variety of speeds, and continue their motion until it is changed by the program or the program stops. They move more smoothly than normal graphic characters which jump from one screen position to another.

CALL VCHAR

The VCHAR subprogram does exactly the same as the Hchar subprogram but will optionally repeat the characters vertically.

Next month, I will show a method which I use to draw complicated graphic displays on the screen.

David Brown

C O N T A C T S

I have one new contact - the first for 1985:

KAROLY FOLDI PO BOX 3082, BAGHDAD, IRAQ

Karoly is one of six Hungarians working in Baghdad who own TIs, and all are keen to make best use of their machines. Karoly's interest lies in new hardware configurations, and in programming tricks. His written English is excellent, as is that of one of his colleagues, MIHALY HARASZTI (or Michael in English), who has also written to me. He has had hardware problems and wrote to TI in the States (who didn't bother answering) and then to Bedford (who did). It would appear that he has a faulty 32K card, and I will write to him shortly to try and obtain more detailed information.

B U L L E T I N B O A R D

I have acquired GARY HARDING's PRK module and Adventure cassettes. If anyone is interested, please drop me a line.

If anyone has either a TI-Writer module, Terminal Emulator II, Speech Synthesizer, or Extended BASIC for sale, please also drop me a line. I have a number of OTIUsers who are in dire need of such things. MARK LEE is willing to offer a MUSIC MAKER module plus cash balance for a TEII and a TI-Writer module.

17 Wagtail Close
Twyford
Reading
RG10 9ED

10th. November 1984

Dear Clive,

I enclose some information which may be of interest to some of the TI*MES readers. I make no claim as to the originality of the information, it is a collection of items gleaned from various sources plus some experimental work of my own.

For some time I have been dissatisfied with the TEXT FORMATTER in TI-WRITER. The first problem was that it performed a Form-Feed at the start -I have since discovered that this works fine if you start part-way down a page, in which case it just rolls to the start of the next sheet. But most people start printing on a new sheet anyway.

After having spent some time disassembling the formatter program and removing the offending form-feed I obtained a copy of TI-WRITER2 in which this had been done and has, perhaps what is the best modification, -true lower case screen letters...

The next problem was having to save to disk, -load the Formatter program, -reload the script and then print. Because the program does not stay in FORMAT mode it has to be re-loaded before each printing. As a result I tended to use the EDITOR mode for most of my work.

Prior to reading an article in H.C.M magazine (Aug 1984), I used to prime the printer in IMMEDIATE mode -to set print face and emphasized printing. It was possible to do some simple formatting by utilizing the codes listed on p146 of the TI-WRITER manual this was limited to changing to wide type or compressed type but that was about all.

The HCM article showed how to use the "CONTROL 'U' " function to obtain other print instructions . This was a great improvement on what I could do previously, I could now set up for emphasized printing (CONTROL 'U' FUNCTION 'R' CONTROL 'U'[to cancel mode] KEY 'E') -this is equivalent to putting CHR\$(27);CHR\$(69) into print instructions. I could change to Italic print in mid-line -reset the printer and double strike words.

Using this method it was possible to use all the printer instructions which only required two CHR\$ codes. I was not able to utilize the three-code instructions illustrated in the HCM. I tried many methods to utilize the underline codes[CHR\$(27);CHR\$(45);CHR\$(1)], but to no avail. It appeared that CHR\$(27);CHR\$(45) was a default instruction which activated the underline and it ignored the other CHR\$ codes used. The problem was finally solved with help from Richard Blanden and Stephen Shaw. On my printer -a STAR DFS10(Gemini 10) and parallel port -it is necessary to insert the third code in the control mode. (CONTROL 'U' FUNCTION 'R' CONTROL 'U' SHIFT -[the 'minus' KEY] CONTROL 'U' SHIFT 'A') and to cancel by repeating the process but ending with SHIFT '2'[CHR\$(0)].

There is very little problem identifying the codes because they appear in a special tiny form of HEX code -except for CHR\$(10) which is screened as LF and CHR\$(13) as CR.

CONTROL CODES FOR USE WITH TEXT EDITOR.

All codes are accessed by use of CONTROL key + 'U'.

<u>CHR\$</u>	<u>KEYS USED</u>	<u>CHR\$</u>	<u>KEYS USED</u>
0	SHIFT + 2	16	SHIFT + P
1	A	17	O
2	B	18	R
3	C	19	S
4	D	20	T
5	E	21	U
6	F	22	V
7	G	23	W
8	H	24	X
9	I	25	Y
10	J	26	Z
11	K	27	FUNCT R
12	L	28	FUNCT Z
13	M	29	FUNCT T
14	N	30	SHIFT 6
15	O	31	FUNCT U

All other CHR\$ are as ASCII codes and can be used in the usual way. The above table can be found on p146 TI-WRITER MANUAL and shows the character which is visible on the screen when using these codes.

These codes are useful when using the TEXT EDITOR mode of TI-WRITER. The text can be formatted by sending the appropriate codes for your printer.

For example if you want to overstrike a word or line of text place the codes on the preceeding line or in front of the words:

e.g TEXT EDITOR is achieved on my printer by keying "CONTROL 'U' FUNCTION 'R' CONTROL 'U' and KEY 'G' [CHR\$(27);CHR\$(71)]. The mode was cancelled by repeating the process but using 'H' in place of 'G'. ('bG TEXT EDITORbH is...[as seen on screen])

When using this method it is important to know that the codes do not take any printer space so a space must be left after 'H' otherwise the next word becomes joined to the one which is over struck.

The table above was typed on consecutive lines and the line spacing set to 1/4". (The coding on my printer is CHR\$(27);CHR\$(65);n -this sets the spacing to n/72").

The key sequence was CONTROL 'U' FUNCTION 'R' CONTROL 'U' KEY 'A' CONTROL 'U' SHIFT 'R' (18/72 = 1/4") .('bA'2)

It is only necessary to use the CONTROL 'U' function to access CHR# lower than ASCII 31. All the other characters can be obtained direct from the keyboard. It is possible to obtain the CHR# above 96 by means of the CONTROL key mode -SHIFT 1 will print a lower-case (a) -SHIFT 'full stop' prints ""["CHR\$(126)].

I have not managed to do a RIGHT justification yet but will continue to try. My next investigation will be to try to find out how the TRANSLITERATE command operates and try to enable this to be used with the EDITOR and so avoid having to bother with the FORMATTER except for special applications. I hope that this help others to enjoy using the TI-WRITER module by just using a single mode of operation.

(A. D. Burt)

E N H A N C E D B A S I C

An enhancement of TI BASIC available through STATISTICS and PRK modules

Peter Brooks January 1985

References: TI Document ARCHIV.PRK.DOC.SUBRLS1 courtesy of TI
Articles by, and personal communication with, PAUL W. KARIS

CALL H() - THE HEADER SUBPROGRAM : DEEPER DISCUSSION

In the first part of this examination of CALL H(), we looked at a general description of what the HEADER subprogram does, and how it fits into the scheme of things. We briefly discussed the structure of the "reserved" area of memory, and how the header dictates its subsequent use.

This issue we will look a little more deeply at each of the facets of CALL H().

The form is:-

CALL H(R/W, INFO, FLD, V or V\$)

where:

R/W = READ or WRITE (Note: NOT R divided by W!)

INFO = the HEADER ITEM NUMBER (an integer between 1 and 14)

FLD = the FIELD NUMBER (1 to 15 on PRK
1 to 99 on Stats)

V or V\$ = a numeric or string item for transfer

R/W

Initially you will WRITE data to the HEADER (the area of memory which "describes" to the computer how the rest of the "reserved" area of memory is to be interpreted); later you may either READ or WRITE further data.

A zero (0) indicates that the contents of V or V\$ (or an "explicit" number or string - i.e., 123456 or "FORENAME") are to be written to the Header, while a one (1) indicates that an item is to be read from the Header and assigned to an appropriate variable. (Note that V and V\$ are simply used here to indicate the need for a variable - you don't need to use these specific variable names, and you can use arrays as well: V(N) or NAME\$(A*B+C-D/F,G+H,I/J)).

Summary

0 = Write data to header

1 = Read data from header

INFO

The parameter INFO tells the computer which item of Header Information you are working on. If you wanted to place a FILE NAME in the header, you would set the R/W parameter to zero (0) to indicate that you want to write something, and set INFO to one (1) - see the list of numbers below and also in last issue). A value of one for INFO indicates that the item to be written is the file name. This name can be between 0 and 9 characters long, and is placed in the "V\$" parameter.

Summary

To write a file name to the Header, use:

CALL H(0,1,FLD,"CLUBFILE")

remembering that you can use string variables (and numeric variables, where appropriate) here as well.

(See table I on the following pages)

Table I: Summary of the INFO values and their meaning

INFO VALUE	INDICATED ITEM	LIMITS
1	File Name	0 - 9 characters
2	Day or Month	1 - 31 } } To enable either European } or American date conventions } to be used
3	Month or Day	1 - 31 }
4	Year	0 - 99 This covers a century of use!
5	No. of FIELDS per RECORD	This item is automatically updated by the HEADER routine each time a new highest-numbered field is defined
6	No. of RECORDS	This item is automatically updated by the GETPUT routine (which will be discussed next issue) each time a new highest-numbered record is written
7	Length of the HEADER (bytes)	This item is automatically maintained by the HEADER routine
8	Length of each RECORD (bytes)	This item is automatically maintained by the HEADER routine
9	Name of FIELD	0 - 9 characters. From this point, FLD is actively involved.
10	Type of FIELD	1 = character 2 = integer 3 = decimal 4 = scientific notation

Table I: Continued

INFO VALUE	INDICATED ITEM	LIMITS
11	Width of FIELD	character : 1 - 15 integer : 1 - 10 decimal : 2 - 11 scientific notation : 8 - 13 (the width is handled automatically by the HEADER routine)
12	No. of decimal places for FIELD	character : 0 (handled by integer : 0 HEADER) decimal : 1 to width - 1 scientific notation : 0 to 5
13	Storage for FIELD in bytes	Handled automatically by the HEADER routine
14	Position of FIELD in the RECORD	Handled automatically by the HEADER routine

Items 9 to 14 inclusive are repeated for each FIELD (FLD) which has been defined.

FLD

The FLD or Field Number is a little awkward to discuss as it behaves a little peculiarly. The computer scans a CALL H() and expects there to be four parameters. Accordingly, although FLD (the FIELD) values do not play a part until INFO item 9, a value for FLD must be provided (say 0) each time CALL H() is used. It seems somewhat odd, considering that other subprograms in the series are capable of distinguishing between a subprogram with one additional parameter, and one with two (e.g., the CALL A() minimum and maximum values for automatic input validation).

Next issue we'll examine in detail how to set up a form and transfer its structure to the PERSONAL RECORD KEEPING module.

C L O S E F I L E

As another (hurried) issue gets put to bed, a few more questions have been answered, and another bundle have raised their heads. The mindful reader may remember some time ago during the short initial series on the Speech Synthesiser when I asked for information concerning a HAMMING WINDOW CIRCUIT (issue 6). I scanned through January's SCIENTIFIC AMERICAN and found a couple of interesting articles, one of which concerned the reliability of computer memories. It discussed in part an algorithm for protecting large memories from chip failures that could render a machine useless. TI's PHROM (Phrase ROM) at the heart of the Speech Synthesiser was, at the time, a large memory (32Kbytes) so this would have been of importance to them. The algorithm was for an error-correcting code which could cope with accidental loss of tiny amounts of data from a memory chip (perhaps caused by an "alpha particle" which is generated during the decay of small amounts of radioactive impurities in the chip). The originator was a RICHARD HAMMING of the Bell Telephone Laboratories in 1948, and the code is known as HAMMING CODE in his honour. Putting two and two together, I would guess that a Hamming Window circuit is related to the Hamming code - of which there are many varieties. It is likely that the other item which had me foxed - the LEROUX-GUEGUEN ROUTINE - is related to error-correction systems.

That article in SA is actually very interesting, the more so when you consider that I am by no means knowledgeable about hardware.

Another article concerned a program called RACTER. A contraction of the word RACONTEUR, - story teller -, it is a little like the ELIZA program which behaves like a psychiatrist and can hold a conversation with you about your problems. ELIZA is rather limited though, and its parrot-like responses to answers phrased in a certain way let it down often. RACTER however behaves rather like a schizophrenic, and judging by the article entitled COMPUTER RECREATIONS by A. K. DEWDNEY, could keep you enthralled for hours at a time. Indeed, the funniest moment came when RACTER was allowed to communicate with ELIZA; the former's exuberant, almost manic, garrulous output overcame the latter's ability to handle the quantity of information.

Now, the way in which I have described the above event may lead you to think that there are two highly intelligent programs with some vague manifestation of "human" qualities. I wish that I could reproduce the SA article here, for you would see the exchange for what it was.

Essentially these two programs are analysing written English, except that one is much more sophisticated than the other. In fact, RACTER is so good that a book has been produced, containing conversations, stories poetry, and aphorisms by RACTER, under the title THE POLICEMAN'S BEARD IS HALF-CONSTRUCTED, published by Warner Books Inc. I am hoping to get hold of a copy if at all possible; it sounds as though it has all the makings of an immensely enjoyable piece of literature.

In fact, RACTER has a companion program called INRAC, which is used to alter RACTER's personality and knowledge. Neither are available for the 4A, although they are for the IBM PC (\$69.95 and \$244.95 respectively).

In a similar vein, another program called SHRDLU, created in 1970 by TERRY WINOGRAD at the Massachusetts Institute of Technology, is more sophisticated than even RACTER, although it is limited to discussing its restricted "table-top" world of coloured blocks and other geometric objects. SHRDLU is a model for an "intelligent" robot, and it can understand "natural language" (e.g. English, German, French, etc.).

SHRDLU can manipulate the objects in its universe and then discuss the relationship of one item with another. For example, tell SHRDLU that there is a green cube next to a red pyramid, and then tell it to move the pyramid and place it on top of the cube. You can then ask it where the green cube is. It will respond with "beneath the red pyramid, which it is supporting". This kind of program is part of research into what is known as ARTIFICIAL INTELLIGENCE. Far from being a fearful prospect, there is a distinct advantage in having a computer or robot which can intelligently discuss logistic problems with humans, rather than simply passively executing programs to attempt to solve the problems.

Over the last few months I have been encountering several interesting articles from various sources discussing aspects of computing which lie at the forefront of research. Two examples of this are WORMs and FRACTAL GRAPHICS. A WORM is a machine code "under-cover agent", able to invade and take over areas of computer RAM. It began as a project to develop a program to allocate unused or disused RAM for memory-hungry tasks; the idea was to fashion the WORM so that it would free any unused RAM in a network of computers and thus make more efficient use of the system. There are now battles between highly-sophisticated WORMs who both "inhabit" the same block of RAM (say 64K) and who are both intent on achieving total control of that memory - obliterating the opposing WORM in the process.

Fractal Graphics on the other hand is a technique for producing highly realistic mountainous terrain on screen. It has developed out of a need to present credible mountain scenery in military training systems for pilots, and is making its way down to the microcomputer end of the spectrum. I have the bare bones of the algorithm for producing such graphics, but I do not yet fully understand the detailed operation: the original explanation was presented terribly, and I have yet to find an alternative information source in order to tie up the loose ends.

I had better close this particular issue now or it will never see the post box until next month, but not before I have welcomed three new names to DTIU: GRAHAM HILTON, CHRIS BAKER, and RICHARD SIERAKOWSKI. I hope that TI-LINES provides you with articles that you want to read, and if not, why not drop me a line telling me what you'd like to see written (or write yourself - no reasonable offer refused!).

A belated Happy New Year to all of you, and I look forward to the coming issues of TI-LINES - even though I know what's going to be in them!!

Peter Brooks

This page is blank