

Volume 2, Issue 7

December 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. G. Brooks
*
*      96 Banbury Road
*      OXFORD
*      OX2 6JT
*
*      Tel: Oxford 50822
*
*****
```



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

THIS OFFER APPLIES UNTIL MAY 1986 ONLY

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. 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 (DIS/VAR 80 or DIS/FIX 80), or in a form which is as legible as possible. Artwork 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
Daisey, Daisy. Post Particles. Data Protection Act. Support Your National Bulletin Board. Ivan's Bits: A Solution ?	
BULLETIN BOARD.....	6
Left-overs...	
COMBINING EDITOR/ASSEMBLER, DISK MANAGER, & TI-WRITER.....	7
By RICHARD J. BAILEY From The NEW HORIZONS USER GROUP	
PSEUDOCODE.....	9
Embrace A Dogma, And You Could Be Bitten	
DISK DRIVES.....	12
From OTTAWA TI-99/4 USERS' GROUP Newsletter, By LLOYD GALENZOSKI, Original By JIM NESS	
USER'S REVIEW.....	14
From ALLEN BURT, A Detailed Look At The MYARC Disk Controller	
STEPHEN SHAW'S SOFTWARE COLLECTION : AN UPDATE.....	21
The Title Says It All	
A LOOK AT GPLLNK.....	22
First Of Two Articles From R. A. GREEN Of OTTAWA TI-99/4 USERS GROUP	
LETTERS.....	24
From NEVILLE BOSWORTH, In Support Of Alan Davey's 4ABC	
A LOOK AT USING GPLLNK.....	25
Second Article From R. A. GREEN Of OTTAWA TI-99/4 USERS' GROUP	
CLOSE FILE.....	28
Grateful Thanks And Christmas Salutations	

E D I T O R I A L

PSEUDO-DAISEY! or Oops! This Idiot's Unhinged!

More than two weeks after DAISEY WHEEL contacted me to express her wish that an open letter to DOT MATRIX be published in TI-LINES, I received a letter which purported to be from someone called DAISY. It contained a submission to the Grand Jelly (Flavour Of Your Choice If You Win) Competition To Find Silly Things To Say Using The Letters O, T, I, And U, but did not contain a return address.

I suspect that the letter may be the work of ANN ONYMOUS-SCRIBBLER, spinster of this Hemisphere, who has been known to impersonate characters with silly names (Yes! This Means You, Ann Onymous-Scribbler!). Her entry of Occasion The Insomniac Unconscious ranks near to STEPHEN SHAW's Occidental Trappist Itinerants' Union, which incidentally he used when addressing a letter to me and the GPO actually delivered it. (See later for Moan About GPO Shredding My Mail Before I Get The Chance To).

I suspect that Ann simply opened a dictionary at random at pages covering the letters she required, and then added the date before subtracting the age of her dog. This would explain everything (My Father Was Not A Detective For Nothing, You Know).

However, I will be placing the matter in the hands of the Police just as soon as RICHARD SIERAKOWSKI comes over again in his Mobile Massage Parlour (as in "I'll be over in two shakes, well, two thousand").

You may have already spotted a clue - Ann didn't know how to spell Daisey, and spotted another clue: Ann couldn't spell Brooks (There Is No "E", Har, Har).

I also noticed that the signature was a very good forgery - but not good enough to get past Eagle Eyes Brooko (what was that about bald eagles?).

So, Ann Onymous-Scribbler, your days are numbered (starting at zero). Make sure you, Dear Reader, renew your subscription next May so that you can follow The Exciting Adventure as we track Ann down!

Quite apart from anything else, I don't think that I could find the "rhubard" flavour jelly which you claim as your prize...

~~~~~  
GPO SHREDDIES  
-----

For some reason, the months of October and November brought a number of large envelopes through my letterbox. However, on the way they had all been relieved of their sides - that is, something had chewed very carefully from one end of each envelope to the other, on both sides, leaving just a front, back, top and bottom to hold the contents in. For that reason I am unsure as to what exactly I have been sent, so a word of caution: if you send me anything which is a mixture of large and small documents in a large envelope, remember that my postman must get pretty hungry on his rounds of an Autumn morning...

## THE DATA PROTECTION ACT

The silly season is upon us again, and our MPs have carefully and with much thought and consideration passed an Act which not only doesn't protect the individual and is unworkable, but which literally forces firms to reconsider the use of computers when handling large quantities of data. Yes, the much-vaunted DATA PROTECTION ACT is now law, and applies solely to information held on computers. The riff-raff can hold anything they like on you as long as it is on paper, and you can't do a damned thing about it.

After I had received a circular from the DoHSS at work, and after having discussed this Act in my professional capacity as representative of my hospital on a local Data Centre group, I was informed that the Post Office had copies of the necessary registration forms and explanatory booklets.

I beetled round to the main office near my place of work and queued for twenty minutes, only to be stared at blankly by the counter staff when I asked for the pack of bumph. None of them knew what it was about, and even the young lad who dealt with me, supposedly of an age group who were taught all about computers at school, could only repeat words like Protector and Data what ? when I tried explaining what I wanted.

Now however, two weeks later, they finally know what it's all about. Well, almost. They now know what to get when you ask for the Data Protection Pack (and would you believe it, they'd had some delivered on the day that I had made my first enquiry, but no-one understood what they were for!).

Anyway, I am a little wiser, and it now looks as though many of the early predictions about cost and duration of registration may have been way off the mark. I have yet to complete reading the lengthy booklet which comes with the pack, but it looks as though you are supposed to have a crystal ball. At one point you are reminded that you should cater for every eventuality and the suggestion is that you should foresee exactly what you will be processing a year, or two, or three from now, and declare those processes now! I will publish further details as time and space permit, but it looks as though it is going to be a right pig's ear of a piece of legislation, with all sorts of publications being produced to "help" you register correctly.

~~~~~

SUPPORT YOUR NATIONAL BULLETIN BOARD

ALAN DAVEY's 4ABC showcase TI Bulletin Board is starting to come of age, and I understand that it now has a dozen regular readers. There has even been the beginning of what Alan hopes may become a torrent of contacts from outside the UK. He has been advised that now that a TI Bulletin Board has been established in the UK, it will begin to form the focus for many TI owners outside the UK, particularly in Europe, and will act as a meeting place for many "wired" TI Users.

However, the equipment will not last for ever, and Alan makes no charge for accessing his board (BT do though!) and it would be helpful if interested parties were to dip into their pockets for a donation to the maintenance and replacement costs for 4ABC. If you feel that you would like to see the only TI board survive, and go on to expand and provide even more sources of information and contacts, then please send your contribution (no "crossed" items, please) to Alan at:

88 Halcombe Estate
CHARD
Somerset
TA20 2DU

Alan has expanded the board and it now offers a selection of adverts and helpful mini-articles and tips. It operates all day Sunday, and if you have a modem or acoustic coupler you should be able to access it. Call Alan during the week on CHARD (04606) 4511 to arrange the finer details, and do not be discouraged if you have difficulty. Talk it over with Alan and he will guide you through the procedures.

Users of Alan's board might like to note the following:

The first character that Alan's modem should receive is the carriage return, sent by pressing the ENTER key. However, if there is noise on the line, his equipment may interpret it as a character. In that event, you may receive an error message which advises you to either select a valid page symbol or press R to reselect. If you have not been via the instruction page, your screen colours will not have been reset, and you should press ENTER and ignore the error message. If you do not press ENTER, the screen colours will not be reset - if you are happy with the yucky green default colour, then you can carry on as normal.

Secondly, when you type the letter corresponding to the page you have selected, you may also type at least one space and then continue typing for up to 80 characters. This facility enables you to select a page and at the same time leave a message for Alan (all of your keypresses are fed either to his printer, so that he has a permanent record of your passage through his board, or to his screen where he may be monitoring your progress). In this way you can ask him, if he is available, to pick up the phone and have a voice-to-voice session.

However, newly-joined OTIUser NEVILLE BOSWORTH has discovered that you can leave a note - long or short - to Alan without having to also wait the 30 seconds it takes for a page to be transmitted. What you do is to type an invalid page selection - say, "S", - whereupon you will receive the error message on your screen, and you can then type out up to 80 characters of your note. In this way you can compile your note in double quick time. Contact Alan if you haven't understood my explanation!

Thirdly, some Users have been given pause for thought by what appears to be a bug in the TEII protocol. If you decide to dump a copy of your current screen contents to your printer with CTRL 2, when the cursor re-appears after the printout it will NOT be where it was before the CTRL 2. Instead it will have been relocated to the HOME position - the top left of your screen. The use of CTRL 2 will also not interfere with anything else that you may be typing to Alan.

Alan is currently looking at ways to enhance his board still further through the use of a "converse with Sysop" facility so that you may use your console rather like a teletype and "talk" to Alan without needing to break the computer/modem connection.

Alan can also advise you on the modem models and prices, if you should be looking to get yourself "wired".

~~~~~

IVAN MAY BE A BIT STUCK

I have had several responses concerning IVAN NIBUR's printer problem from last issue. The general consensus is that the least significant bit was stuck in the ON position in every byte acted on by the printer. This meant that any ASCII character with an EVEN decimal code (0, 2, 4, etc.,) became ODD, thus making all spaces (CHR#:32) into shrieks/shouts (CHR#:33) (i.e. "!") and resulting in the existence of that famous late Summer month, Sequences. Not to mention the sudden jump in the decade.

Now, I understand that Ivan may have a QuenData printer, and JOHN RICE tells me that from time to time, when he switches on HIS QuenData, it behaves in a peculiar fashion - although not in the same league as Ivan's!

The possibilities are not endless, and the most likely culprit is probably dust or dirt or other contamination of a contact somewhere, perhaps on the PEB motherboard. One solution is to replug the RS232 card (maybe even try a gentle cleaning of any contacts while the mains is off, and not with sandpaper) and power the system up again.

BULLETIN BOARD

SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

I still have a few things left over from the Show which weren't sold, and a few more items have appeared since then:

|                                  |          |                              |     |
|----------------------------------|----------|------------------------------|-----|
| "Gunshot" single joystick        | £ 4      | Adventure Module + cassette  | £16 |
| DSDD 40T Half Height MPI drive   | £75      | Extended BASIC V110          | £45 |
| Music Maker Module * 2           | each £12 | SSDD 40T Full height Shugart | £45 |
| 32K RAM Expansion card           | £65      | TI-Writer Module (+ manual)  | £20 |
| Parsec * 3                       | each £ 7 | TI Invaders                  | £ 6 |
| Early Learning Fun Module        | £ 4      | Addition & Subtraction 1     | £ 4 |
| SSDD 40T Full Height MPI drive   | £45      | 33 Programs For 4A (Book)    | £ 6 |
| Dynamic Games For Your 4A (Book) | £ 3      | Learning To Use 4A (Book)    | £ 3 |
| Get More From The 4A (Book) * 2  | £ 4      | Getting Started With 4A * 2  | £ 4 |
| 4A User's Handbook (NOT TI Ref)  | £ 7      | Editor/Assembler manual      | £ 8 |
| TI Forth + manual * 3            | £12      | TI Thermal Printer + 3 rolls | £60 |

Contact me at the usual number (it's on the front page...)

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

RICHARD OWEN wonders if anyone fancies the BURGERTIME module for £8. You can contact him on 0656 4972 if you do.

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

-----  
MAKING ONE MODULE THAT COMBINES E/A,  
DISK MANAGER AND TI-WRITER  
-----

Richard J. Bailey of the NEW HORIZONS USER GROUP, from an article published in the Summer newsletter, a joint publication of the OH-MI-TI and NEW HORIZONS TI-99/4A HOME COMPUTER USERS' GROUPS

If you've ever opened any of your command modules like the ones listed in the title of this article, you'll see the infamous GROM chips. The modules that use these GROMs have corresponding pins on all chips bussed together. The GROM itself determines the addressing and there can be one or more GROMs depending on the cartridge. There is a common reset line (pin 14) on all the chips and MULTIPLAN uses 5 of these GROMs, all in parallel. One thing all these cartridges have in common is the circuit board that has room for up to 5 GROMs.

If you were to take TI-Writer and E/A that both have only one GROM each, and put both those GROMs on one board, option 1, BASIC, would be O.K. However, option 2 would be garbage because each GROM is trying to display its function. To get around this problem you have to use separate resets to each of the GROM sets. To do this I removed the solder from all the blank i.c. locations on the TI-Writer board (or any board you'll be using that has one GROM) and removed the 100 ohm resistor which you'll later reuse. You now have to cut the reset line (the run connecting pin 14 on all chips) between the chip sets for the GROMs you are adding, i.e. DISK MANAGER II has 2 GROMs so cut the run on either side of where you insert those chips, but not in between.

You now have to add a switch to connect the 100 ohm resistor to each of the GROM sets you are using. I'll leave it up to you what you use for a switch. If you only have 2 functions in a cartridge, then a 2 position miniature slide switch is ideal. If you have 3 functions, you could use a DIP switch but then you have to remember to only have 1 switch on at a time or you'll get garbage when you try to make a 2nd or 3rd selection. The 100 ohm resistor has to be connected from the common on the switch to the hole on the outside edge of the board where it was originally. The switch can be located directly over the location the resistor originally occupied and you will have to cut a hole in the top of the cartridge case so you have access to the switch slide or whatever.

It's important that you remember that any modification is your own responsibility and if anything goes wrong, you are the one to suffer, not me. Try to make sure that everything will physically fit and be VERY careful removing and handling the i.c.s. I would recommend using sockets on the board you are going to use (clean the i.c. pins of all solder before inserting in sockets) so you can easily replace them in the original cartridge (use socket here also) if you decide to undo the modification for any reason. Some of the cheaper modules like MunchMan have 4 or 5 possible i.c. locations. If you use the MunchMan cartridge, you can also include the 8K ram modification that Ron Gries and John Clulow came up with and have a really super cartridge!!

{See BRUCE CARON's article in last issue's TI-LINES. PB}

~~~~~

Make the most of your micro

MICRONET 800 is a fully interactive Database for Home and Business Micro Users.

There are over 30,000 pages of information on MICRONET, with News, and Views on the whole computer industry.

There is interaction between YOU, US and over 70,000 other users around the U.K. and the world.

There are magazines on computers and business, offering specialist information on everything from how to wire up a serial printer to the best investments on the stock exchange.

A Firm of Solicitors is on-line to give FREE legal advice.

AND MUCH MUCH MORE...

WE ARE OFFERING A SPECIAL DEAL IN THIS MAGAZINE ONLY, OF EIGHT POUNDS OFF THE FIRST QUARTER'S SUBSCRIPTION OF SIXTEEN POUNDS FIFTY.

THAT'S HALF PRICE!

Contact your Micronet Agent, MARTYN AMESS, on 04246 (Battle) 4813 (evening) or 0892 29783 (daytime) for more information on what Micronet can do for you and a special brochure and application form.

If you wish to join immediately, please fill in the coupon below, and send it off to us.

Micronet 800



Cut out the coupon
and send it to:

Micronet 800
8 Herbal Hill, London EC1R 5EJ
Telephone: 01-278 3143

TI USER SPECIAL OFFER

Name _____

Address _____

_____ Telephone No _____

Age _____ Micro TI99/4a _____

P S E U D O C O D E

P e t e r B r o o k s

D e c e m b e r

I doubt if many of you will have heard the term before, but PSEUDOCODE (syoo dough code) is the next best thing to sliced bread as far as programming is concerned.

You can forget the old dogma of "Flowchart Everything!" and embrace the new dogma of "Pseudocode Everything!".

That's really not a fair way to approach a discussion of pseudocode, but, as with most things, it is best taken with a pinch of salt.

Indeed, after you have read this article you'll probably exclaim that you've been pseudocoding for years and what's so special about it anyway ?

The answer to "What exactly is pseudocode ?" has to be a little vague since the question is of the category which includes "How long is a piece of string ?". Basically, pseudocode is anything you need it to be. Unlike BASIC, Pascal, Forth, Assembly Language, etc., the vocabulary and command structure are not predefined by somebody else (i.e. a manufacturer). The person who creates your pseudocode is you.

Now, so far I have danced around the edges of pseudocode without delving into it and giving examples. The reason for this is relatively simple: I can only give you examples of MY pseudocode. In fact, I have already done so; if you read my article on Using Your Joysticks in V1.8 (Nov. 84) you will see many examples.

Essentially, pseudocode is almost the same English (or French, Flemish, Dutch, etc.,) that you would use to describe a problem and a method for its solution. I say "almost the same", because a program written entirely in pseudocode is not pure prose - it does not "flow" the way that a manual or letter would, at least certainly not in the early stages.

Here's an example of my pseudocode describing a subroutine to mimic DISPLAY AT() in console BASIC:

1. First Entry Point: Clear line to be printed upon
2. Second Entry Point: Loop from beginning to end of string, using HCHAR or VCHAR to place the string, character by character, on screen, beginning at row and column specified
3. Return

That was a reasonably detailed pseudocoded description; a simpler example would be:

1. Display Data and return

The language - pseudocode - that I use is not fixed and varies from occasion to occasion, which for me is useful as I have a lousy memory!

The same routine might be described differently each time I invoke it; perhaps as DISPLAY MESSAGE or PRINT DATA (NO SCROLL) or whatever.

So how do you set about pseudocoding a program then? The simplest (and least helpful) answer is "any way you like"!

As with all programming, the planning involved is a little like making a shopping list or a list of "things to do before going on holiday". You can't begin to pseudocode a program if you don't have a clear idea about what you intend doing.

Begin by outlining whatever the "problem" may be. A "problem" ranges from "How do I write a bells and whistles Space Invaders?" to "How much VAT do I have to pay the VATman?".

If you have no clear idea of what Space Invaders involves, or of the current VAT regulations, you won't have much chance of immediate success, so make sure that you know EXACTLY what you intend doing. You CAN be vague about certain things in the early stages, but not for long.

Write down what you need to do - for example, in the case of VAT, let's say (for the sake of argument) that you will need to know:

- (i) What is the appropriate rate of VAT
- (ii) How much VAT to charge on a given price
- (iii) What the final VAT-inclusive price will be
- (iv) Given a VAT-inclusive price, how much is VAT, and what was the original price

There are other things - a running total of VAT paid, for example; - but we have enough to be going on with.

Let's re-express our problem in more pseudocode (or possibly spew-docode), and don't forget, YOU may prefer to use different words or expressions):

1. Find 15% of a value
2. Add that to the value
3. Find 100/115 of a value (= the original value prior to VAT)
4. Find 15/115 of a value (= VAT on the original value, and could also be obtained through simple subtraction)

Make sure that you understand completely any maths involved, especially where percentages are concerned (they seem to throw more people than do professional wrestlers!).

Make sure that any shortcuts you use are valid; minutes spent checking now may save days of work in the future.

I won't go into a detailed discussion of percentages or of basic algebra, as these subjects are lined up for a future series of articles in TI-LINES.

Now let's break the four steps down into finer detail, using more "plucked from thin air" pseudocode:

- 1.A Input Value
- B Multiply by 0.15
- C Do something with result (see what happens if you don't have a clear idea of what you intend doing ?)

- 2.A Multiply by 1.15 (the same as ADDING 15%)
- B Do something with the result

- 3.A Input Value
- B Multiply by 20, divide by 23 } Both methods find the
or divide by 1.15 } original Value
- C Do something with the result

- 4.A Multiply by 3, divide by 23 (finds the VAT)
- B Do something with the result

The pseudocode instruction "Do something with the result" is vague, simply because there are so many things that could be done, depending on the circumstances. For brevity you might use "DSWR" in each case.

Take it a stage further:

- 1.A.1 Prompt for Value
- 2 Input Value
- 3 Validate, branch to 1.A.1 on failure

- 1.B.1 $VAT = Value * 0.15$

- 1.C.1 DSWR

- 2.A.1 $VATINC = Value * 1.15$

- 2.B.1 DSWR

and so on. You can see how we are slowly approaching something resembling BASIC; however, with a much finer degree of detail we might code in Assembly Language.

The closer that the computer language lies to the "natural" language (i.e. English, French, etc.), the less pseudocode you need, and the more like pseudocode the computer language becomes (or vice versa).

If you prefer a more rigid structure, you might set your version of pseudocode down in detail on paper, giving it a syntax and vocabulary. You would then be just a step away from having created a computer language all of your own. (Perhaps a Sloane Ranger version of Forth, responding to ENTER with "OK, YAH ?")

By and large, if carried out with some care, pseudocode contains within it its own documentation, and unlike flowcharting, doesn't need to be totally rewritten every time a modification needs to be made.

The main requirement is an ability to sit back and think carefully and clearly about little bits of the problem/solution at a time, and to do it in a "structured", or carefully planned, manner. You may have heard of the "top down, bottom up" approach to programming - this is "Top down design, bottom up coding" - which is closely related to the approach I have used in demonstrating pseudocode here and in V1.8.

If you have any comments or observations I would be happy to receive them, and where appropriate, publish them.

DISK DRIVES

Taken from the October 1985 newsletter of OTTAWA TI-99/4 USERS' GROUP

This article was downloaded by LLOYD GALENZOSKI from the Chicago West Bulletin Board, telephone (312)766-2797. JIM NESS is the author and with the exception of minor editing changes, this is his article.

"It's funny (at least to me), but there are lots of people who seem to know lots of stuff about their computers, and all those tiny chips, and how the bits and bytes are handled, and there seems to be next to nobody who knows anything about disk drives, and how they work. Sensing this huge gap in Man's knowledge, I decided to figure out what makes them tick.

The great thing about disk drives is that they can find files buried randomly within a huge field of data, and they do it pretty fast. Actually, they can do it so fast because it's not at all random.

The mechanical concept is not all that complicated. A small motor spins at 300 rpm (at least in this country, with its 60 hz power supply), and there is a tiny stepping motor attached to a read/write head. A stepping motor is a common item in indexing applications, where you want a motor to move a precise distance and stop on a dime. The read/write head is just a smaller version of what you have on a cassette recorder.

The stepping motor "steps" the head from track to track on a diskette. The tracks are concentric circles, not a long spiral as you would have on an album.

All of this is ultimately controlled by the disk software provided with your computer. Usually this is located in ROM within the machine. In most machines, the ROM is only sophisticated enough to load in the official Disk Operating System (DOS) which is located on the disk in the drive when the machine is turned on. The DOS contains all the file handling software, copying software, etc., and because it is on disk, it can be easily modified and/or updated as time goes by.

Our friends at TI decided to put the whole thing in ROM, which has a few bad side-effects. First, it makes it hard to update and improve the software, which is located in the Disk Controller Card. Second, although the machine is a 64k machine, just like all others, TI has set aside so much memory for special purposes, that there is only 32k left to play with. They set aside 8k for cartridges, 4k for disk drive, 4k for RS232/PID cards, 4k for the Operating System (can't complain about that one), and 8k for various interfaces (speech, sound, VDP). Ok, those are all good applications to have, but if you don't use them, you still can't use that memory for other things.

Anyway, all of the controlling software for the TI99/4A is located in the ROM card, as I said. This software tells the stepper motor when to step to the next track, when to return to the beginning, etc.

There is no standard for how a computer keeps track of data. In the case of TI, there is a directory of existing files, and a map of where they are located, at the beginning of each disk. These files are not necessarily all in complete groups. If you delete a 12 sector file from a disk, there is a 12 sector gap recorded in the map. Then if you add a 20 sector file, the software will put the first 12 sectors in the gap, and put the rest in the next available spot. When you ask for a file that is broken up this way, you can hear the drive head scooting along to read each individual segment.

Because the disk drives themselves are pretty standard, there are a few things that don't change. For instance, there are 48 tracks per inch in most 5 1/4" systems. (There is a new 96 TPI system around, not compatible with the TI disk controller), and most systems only use 35 or 40 of the available 48 tracks. There are either 9 or 18 sectors per track (single or double density). Each sector holds 256 bytes of data, and the standard design allows 250,000 bits per second to be written.

Wow, you say, 250k! That is about 25k bytes per second, right? How come I cannot load a 25k program in one second, then ?

Where was I ? Oh, yes, I mentioned that most drives are capable of transferring data at about 250,000 bits per second, and you were asking how come your programs don't transfer that fast.

Two reasons. First, as I said, the transfer of data is actually controlled by the ROM software in the TI-99/4A, and to be as good as it is, it had to be a little bit slow. Not REALLY slow (anyone ever use a C64 disk drive ?), but not as fast as it could be. The second reason also has to do with software, but it is a universal problem associated with single density storage.

The major difference between single and double density storage is the way in which the data is coded. In order for the software to keep track of where the read head is located on a particular track, there are clock or synch bits laid down with the data bits. In the old fashioned single density format, a synch bit was laid down ahead of each "0" bit, so there were never two "0" bits in a row. That kept the software from getting lost if there were a lot of "0" bits in series. Putting all those synch bits on the disk took up a tremendous amount of space that should be used for data.

So, some genius came up with a way of encoding the clock bits in with the data bits, so that no unnecessary space was lost. Voila, double density storage was born! Double density, as used with the Corcomp software, is said to increase transfer speed by at least 80%, mostly because the number of bits to transfer is cut way down.

So much for the exciting story of double density versus single density. How about double sided versus single sided? Well, obviously, it requires two read/write heads in the drive. Did you know that when reading a disk, the software reads, first, a track from side one, then the opposing track from side two, and continues back and forth? You didn't know that? There is a simple reason for doing it that way.

The disk head needs something to keep the disk stationary against it. In a single sided drive, there is a small arm holding the back side of the disk against the head. In a double sided drive, that arm would be in the way of the back side read/write head, so the solution was to use the two heads, directly across from one another, to hold the disk in place. In order to keep them across from one another they alternate reading or writing as I said above. Very interesting, right? So if you wreck one side of a double-sided disk, you can kiss the whole thing goodbye."

Although not proven, there is some doubt that double-sided drives alternate from side to side as indicated by the author. It is said that the TI controller card does not function that way. Rather, it fills one side then moves to the other. (Confirmed by observation. PB)

Possibly the more technically inclined amongst us would be good enough to shed some further light on the subject.

MYARC DISK CONTROLLER CARD

A l l e n B u r t

D e c e m b e r

Whilst visiting the second TI-EXCHANGE Show at Brighton I had a rush of blood to the head and purchased the MYARC Disk Controller card. I know this pleased Howard (Arcade Hardware) but it did not give the same amount of pleasure to either the Bank Manager or the wife. To be honest I was not too sure when I arrived home as to the reason why I needed it.

However it was soon installed in "the box" and the TI card was carefully packed into the Myarc box ready for sale. One thing that did please me was that I now possessed a Disk Manager 2 module which would save my fingers a little bit of work - it does not require the same number of key strokes to get it moving as the earlier version.

The first action was obviously to test whether it would give me double density - on my SSSD disks !!! (it has been written in an earlier TI*MES that there is a sense of wonder when the sector count goes beyond 359). This turned to disappointment when it stopped at 639; surely twice 360 made 720 - who had pinched the other 80 sectors ? Perhaps now is the time to open the manual and read the instructions - yes, it does say that it is not true double density because it uses 16 sectors per track, not 18 - one problem solved.

Now for the big test: DOUBLE SIDED DOUBLE DENSITY (on SSSD disks !). Set the Formatter going and watch - there it is, 1279 on the sector count. I can now put almost the same amount of data on one disk that occupied four disks previously (only need to save on the purchase of 100 disks to pay for it !!!).

INBUILT SUBPROGRAMS

The next thing to try was the unique CALL subprograms included in the card.

CALL DIR()

This allows the cataloguing of disks without using Disk Manager. Just enter Call Dir(1) in either of the basic modes and the contents of the disk in drive 1 will be listed on the screen without harming any program in the console. The readout has more information than is given by Disk Manager - you are given the program size in bytes, (this does only apply to program format files) in addition to the number of sectors used as shown on the next page:

DISKNAME = TIWRITERXB
USED = 190 AVAILABLE = 33

FILENAME	SIZE	TYPE	P
CHARA1	9	PGM	2048
DPC	18	PGM	4106
EDITA1	33	PGM	8190
EDITA2	6	PGM	1046
FORMA1	33	PGM	8190
FORMA2	15	PGM	3342
FUNNELDOC	51	D/V	80
LOAD	25	PGM	5943

The listing cannot be dumped to printer unless there is a screen dump program held in the memory, but the controller can be used to control up to FOUR drives ... one cannot have everything.

CALL ILR

This is similar to CALL INIT available with Editor/Assembler but can be used without special modules plugged into the console i.e. in console basic - but there must be an expanded memory fitted for it to clear. It can be used in programs or in Immediate mode.

CALL LR

This is equivalent to CALL LOAD but again works without the need for special modules and can be included in programs. When used in the Basic modes it will load a program file as if in Editor/Assembler mode.

CALL LLR

Equivalent to CALL LINK and operates as the other CALL subprograms.

These CALLs however do not like being used in multi-statement lines. Only the first Call is read !

MYARC DISK MANAGER

Although MYARC had supplied the DISK MANAGER II module with the card they had promised to supply an enhanced version on disk. Howard was constantly being told that it should arrive within the next couple of weeks. It finally arrived four months later - but it was worth the long wait.

The program is what should have been supplied by Texas in the first place. It permits single screen editing of the disk information plus many more useful features. The program will partially run with the TI card - it will not Format or Copy but the other features can be used.

The program can be loaded from Basic or Extended Basic when used with the MYARC Card or with the Editor/Assembler module with the TI card. The title screen gives the following command menu:

SETUP

Pressing "S" will bring up the SETUP screen and a new menu appears at the bottom. The commands are:

SAVE EXIT CHANGE FOREGROUND BACKGROUND

FOREGROUND/BACKGROUND

By pressing the "B" key you can progress through the colour table in sequence changing the background colour. The colour number is displayed at each change. Pressing "F" will run through the same process for the foreground colours. You can have any combination of the sixteen colours you wish.

CHANGE

Pressing "C" will allow you to change the system defaults for the printer and disk drives. You can set up each drive with a default condition ranging from Single-Sided Single-Density to Double-Sided Double-Density. This is mainly used when formatting disks in the various drives.

SAVE

This command will allow you to save the new setup instructions to the disk.

CATALOGUE

Pressing "C" will bring a response asking for the Drive number - default value of "1" - pressing enter or "1" will list the contents to the screen of the disk in drive 1. If another drive is required it only requires the number to be pressed (i.e. pressing "2" or "3" will immediately read the disk in that drive). If there are a large number of files on the disk you can scroll using the Function or Control key with "UP" and "DOWN" keys in the normal way. Pressing "C" again will allow you to catalogue another disk.

EDIT

If the "E" key is pressed and there is a disk catalogue on the screen another menu appears at the bottom (if no catalogue is present you will be asked to enter the drive number to catalogue a disk).

New Menu:

COPY MOVE DELETE RENAME PROTECT UNPROTECT
VOLUME BACKUP SEE

All file utilities and disk name changes are handled on the one "catalogue" screen.

COPY

Pressing "C" will enter a "C" in the left hand field of the catalogue - any file with a "C" will be copied to another disk when the commands are executed later.

MOVE

This is similar to COPY but will DELETE the original file after copying.

BACKUP

This command places a "C" against all the files in the catalogue thus making a copy of the whole disk.

When the copy commands are executed you are asked which drives you intend to use and the disk manager will then read both disks and tell you if there is sufficient room on the copy disk to hold the files you wish to copy - you are told the last file it can copy if there is not enough room left. It will only copy a complete file so you do not have a partially loaded file as in the case of the Disk Manager module.

The copy program reads about 90 sectors at a time and then writes to the copy disk.

RENAME

Pressing the "R" key will place a blinking cursor on the file name - you then enter a valid name and press ENTER when complete.

VOLUME

This allows you to change the Disk Name without going to another screen.

PROTECT

Pressing the "P" key will automatically place the cursor in the right hand field where you may enter a "P" before dropping to the next line where another "P" can be inserted or the "arrow keys" will move to the next line if needed.

Pressing a "U" will UNPROTECT a program or file in the same way.

SEE

This is probably one of the most useful commands in the menu. By pressing "S" when the cursor is on a DIS/FIX or DIS/VAR file - the file is read and scrolled on the screen. It can be stopped by pressing the space bar and restarted in the same way. This gives an opportunity to check what is in the file without having to load TI-WRITER or EDITOR/ASSEMBLER to do it.

EXIT

This returns you to the main menu field where you can select the XECUTE command by pressing "X" and all the changes made during editing will be made to the disk.

If "X" is not pressed then no changes will be made and you are free to select another of the main menu commands:

UTILITY

This command will give you the UTILITY menu of:

TESTS CLONE FORMAT LOAD/RUN RAMDISK EDIT

TESTS

These are the disk tests similar to those in the Disk Manager Module.

CLONE

This is another copy routine but this time it makes an exact copy of the main disk. It will OVERWRITE anything left on the copy disk. This version does not copy the files in alphabetical sequence but in the exact sequence in which they exist on the main disk.

FORMAT

The program permits a choice of six formats:-

- Single Sided Single Density
- Single Sided Double Density - 16 sectors per track
- Single Sided Double Density - 18 sectors per track
- + Double Sided version of the above three

Thus the number of sectors per disk can range from 360 (SSSD) to 1440 (DSDD-18). This gives a storage capacity of 90K to 360K on a single disk.

The FORMAT menu also allows a choice of the interlace patterns. The interlace is the step interval used when verifying the sectors after formatting. The program has default values set to suit most drives, however, the instructions suggest that when using the newer drives (with 6ms or less track-to-track time), using a lower number for the step interval can speed up the process. The table below shows the times for formatting using an MPI Double Sided Drive.

Interlace Step					

	Sec/Trk	Sectors	3	4	5

Single Side SD	9	360	54.4	53.0	63.0

Single Side DD	16	640	46.2	60.8	61.2

Single Side DD	18	720	52.5	60.8	60.9

Double Side SD	9	720	104.5	104.5	-

Double Side DD	16	1280	84.8	116.8	117.1

Double Side DD	18	1440	100.4	116.7	117.1

Formatting times using DISK MANAGER 2 were as follows:-

	SSSD	SSDD	DSSD	DSDD
Sectors per disk	360	640	720	1280
Time in secs	54.1	60.0	101.0	1190

The MYARC Disk Controller card also has a set of DIP switches which can be set to make use of the faster track-to-track times on the newer drives.

When this was set the formatting was marginally faster.

		Interlace Step				
	Sec/Trk	Sectors	2	3	4	5
SSSD	9	360	35.9	52.0	52.1	61.6
SSDD	16	640	149.3	45.3	52.0	59.9
SSDD	18	720	-	51.8	52.0	59.9
DSSD	9	720	69.0	99.3	99.9	-
DSDD	16	1280	-	83.8	99.8	117.1
DSDD	18	1440	-	99.9	99.9	115.8

It appears that the optimum interlace step for the MPI drive is 2 for Single Density and 3 for Double Density. The optimum times for the MYARC version are approximately 2/3rds of the time taken by the Disk Manager 2 version.

LOAD/RUN

This command allows you to load and run an assembly language program without having to use the Editor/Assembler. The prompt asks for the file name and when this is entered it then asks for the program name - if it is an auto-run program just press ENTER here. You are then warned that it will clear the Disk Manager program from the memory and when you press ENTER it will load the program.

RAMDISK

This command allows you to format and use the MYARC RAMDISK if it is fitted.

This MYARC Disk Manager version is certainly novel and impressive in its action. It also shows what could have been done by TI if they had given enough thought during development of the system.

~~~~~

-----  
STEPHEN SHAW'S SOFTWARE COLLECTION  
AN UPDATE  
-----

#### NEW PROGRAMS IN:

SORGAN...A very interesting 'synthesiser' program which enables you to produce a wide variety of different sounds to be played from the keyboard. Nicely put together.

PR-BASE...At last, a FREEWARE Data Base program. Not simple to use but a worthy effort.

CATALOG...Yes, another disk cataloguer! This one can handle up to 2000 files, and is quite fast.

The above three are available together on two disks... PRBASE needs two disks (one for documentation) and the other two fill out the documentation disk!!!

-----  
A LOOK AT GPLLNK  
-----

R. A. Green

December

From the September 1985 newsletter of OTTAWA TI-99/4 USER' GROUP

The Operating System of the TI99/4A consists of code in ROM and in GROM. The code in ROM is assembler language. The code in GROM is TI's proprietary Graphics Programming Language (GPL).

The ROM code has three main functions: interrupt processing, floating point arithmetic and GPL code interpretation. The GROM code does everything else!

There are, in all this GPL code, several very useful routines that can be used by Assembler language programs. The Editor/Assembler and the Mini Memory modules provide a means, called GPLLNK, to access these routines in GROM. The Extended BASIC and TI-Writer modules do not provide a link to GPL.

I have developed a GPL link routine that will work for all modules. The Assembler source listing is shown below. The code for this routine is a bit tricky, so a few notes for those who want to understand the code may be in order.

1. The workspace registers are already loaded with some necessary values when RAGLNK is called.
2. The first, and only the first, time RAGLNK is called, it searches all GROMs until the hexadecimal value >OFFF is found.
3. The GPL operation code >OF is a call to an assembler language routine. The byte following the >OF, in our case, >FF gives the table number and entry number in that table. Table number 15 (>F) begins at >8300 in the console CPU RAM, and entry 15 in this table is at address >831E.
4. A GPL CALL stacks the current GROM address then branches to the routine to be called. A GPL RETURN unstacks a GROM address then resumes execution at that address. RAGLNK stacks the GROM address of the >OFFF instruction, then goes to the GPL interpreter to begin execution of the GROM subroutine. When the GPL subroutine does a RETURN, the >OFFF instruction is executed, causing GPL to exit to the assembler language routine whose address is at >831E. This brings it back to RAGLNK which returns to its caller.

## ASSEMBLER SOURCE LISTING:

```

*TITLE:      GPLLNK Subroutine
*AUTHOR:      R. A. Green
*FUNCTION:    Provides access to the GPL routines, no matter which cartridge
*             you are using.
*LINKAGE:    Same as described for GPLLNK in E/A or MM manuals, except that
*             the GPL STATUS byte need not be reset before calling:
*             BLWP @RAGLNK
*             DATA GPL-routine-address
*NOTES:      This routine depends upon finding the value )OFFF somewhere in
*             GROM. This value occurs at least 3 times in the console GROMs
*             in my machines. )OF is the GPL opcode to call an assembler
*             routine.
*
      DEF RAGLNK
RAGLNK DATA WSP,$+2      Linkage/Transfer Vector
      MOV RO,R0          Do we have an address of )OFFF?
      JNE STACK         Jump yes
* Find an occurrence of )OFFF somewhere in GROM
      MOVB RO,*R3       Set the GROM address to zero
      MOVB RO,*R3
      JMP $+4
SRCH1  INC RO           Increment our GROM address
      MOVB *R4,R1       Get next GROM byte
SRCH2  CI R1, )OFF0    Is it the start of our value?
      JNE SRCH1        Jump no, keep looking
      MOVB *R4,R1       Get the byte after )OF
      CI R1, )FF00     Do we have )OFFF?
      JEQ STACK        Jump yes, EUREKA!
      INC RO           Bump our GROM address past the )OF
      JMP SRCH2        And keep looking.
* Notice that the above loop will not end if we
* do not find an occurrence of )OFFF
* Put our GROM address on the GPL subroutine stack.
STACK  INCT *R7         Bump GPL stack ptr at )8373
      MOVB *R7,@REG2+1  Get stack ptr into )83xx
      MOV RO,*R2        Our GROM address to the stack
      MOV *R6,R9        Save contents of )831E
      MOV R5,*R6       Put address of BACK into entry )F of table )F.
* Get GPL routine address from CALLER
      MOVB R10,@)837C   Reset GPL STATUS byte
      MOV *R14+,R8      Fetch the GROM address
      LWPI )83E0       Switch to the GPL workspace
      MOV @REG8,R6     R6=next GROM address to interpret
      B @)0060        Go to GPL interpreter
* Hopefully GPL will come back here
BACK   LWPI WSP        Switch back to our workspace
      MOV R9,*R6       Restore value in )831E
      RTWP           Return to calling program
* Our workspace registers loaded with interesting stuff.
WSP    DATA 0         R0=our special GROM address
      DATA 0         R1 LSB is zero
REG2   DATA )8300    R2=GPL subroutine stack address
      DATA )9C02    R3=GROM write address address
      DATA )9800    R4=GROM read data address
      DATA BACK     R5=address for GPL to come back to
      DATA )831E    R6=address of entry )F of table )F
      DATA )8373    R7=pointer to GPL subroutine stack
REG8   BSS 2         R8=GROM addr of GPL routine
      BSS 2         R9=saved contents of )831E
      DATA 0       R10=ZERO
      BSS 10        R11-R15
      END

```

-----  
L E T T E R S  
-----

Dear Sir,

I recently purchased from Arcade Hardware a Nightingale modem to extend the capabilities of my trusty TI. Unfortunately, in making the decision as to which type of modem to buy, and how to interface it to my rig, I was entirely without help. Luckily for me, I seem to have made a reasonably good job of it, but others may not be so lucky.

ALAN DAVEY this reason that I am writing to commend to you the services of Alan Davey (Tel. 04606 4511) who runs the 4/ABC bulletin/news board from his home in Somerset. Regardless of whether you already have a modem or whether you need advice on purchasing and attaching a modem from scratch to your TI he is willing to help in any way that he can. I myself have found him to be an endless source of encouragement, without which I may well have given up the attempt to "Baud Walk" around Britain's bulletin boards. He has also put me in touch with other TI freaks (both with and without modems) around the country who I would not otherwise have known existed.

Some of you may not be aware that there is software in the user group library that does away with the need to purchase the TE2 module, and that is easy and reliable to use. There is also a program that allows you to access Prestel at 1200/75 baud for the price of a local phone call. Believe me when I say that it is worth buying a modem just to see the brilliant colours that the TI can produce when connected to Prestel. By the way, Alan Davey can fix you up with the phone numbers and user codes to access the free trial section of Prestel, which is in itself well worth seeing.

While logged on to bulletin boards around the country I am constantly frustrated to find special sections for Commodore, Spectrum, Beeb etc. but nothing for us. A special section on a board such as Micronet or Microlink would allow TI modem users to chat together for an hour for only 40p! There could be advice surgeries etc. and it would be possible to electronically deliver a message to as many as 500 TI users at once via the mailbox facility.

We cannot afford to let technology leave us TI users behind, and therefore, I beg to suggest that you assist Alan in determining just how many TI owners have modems, or intend to purchase modems in the near future so that the viability of a TI network can be investigated. In the meantime, if people can be found who are willing to contribute to his news board with hints and tips or hitherto unpublished articles it would greatly assist in his labours to create an "Electronic Cottage" in which the TI can live and grow.

Yours faithfully,



NEVILLE BOSWORTH



-----  
A LOOK AT USING GPLLNK  
-----

R. A. Green

December

Taken from the October 1985 newsletter of OTTAWA TI-99/4 USERS' GROUP

In a previous article in this newsletter I gave a general purpose GPL link routine. This time I will describe how to use some of the GPL subroutines.

Some of the descriptions of the GPL routines in the E/A and MM manuals are not complete and there are no actual examples of coding. In the following I will describe some GPL routines in detail and then give a coding example using them. Remember, if you are using the E/A or MM GPLLNK, you must set the GPL STATUS byte at )837C to zero just before calling GPLLNK.

```
*-----*  
* GPL Routine )0016 *  
* Load Standard Character Set *  
*-----*
```

This routine is slightly misnamed. The character set loaded is the 8 pixel high upper case characters used on the title screen. The patterns for characters 32 to 95 are loaded at the VDP address specified in FAC ( )834A). Character 32, the first one loaded, is a blank. Character 95, the last one loaded is the underscore.

When using the Extended BASIC setup for the VDP, the pattern for the blank is loaded at VDP address )0400. When using the Editor/Assembler setup for the VDP, the pattern for the blank is loaded at VDP address )0900.

```
*-----*  
* GPL Routine )0018 *  
* Load Small Character Set *  
*-----*
```

Again, this routine is slightly misnamed. The character set loaded is the "standard" one used by BASIC, XB, E/A. etc. The characters are the 7 pixel high upper case characters. The patterns are loaded for the same characters as for Routine )0016.

```
*-----*  
* GPL Routine )004A *  
* Load Lower Case Character Set *  
*-----*
```

At last, this routine is named correctly. It loads the patterns for characters 96 to 126 into the VDP at the address specified in FAC. Character 96 is the single quote and character 127 is the tilde. When using the Extended BASIC setup for the VDP, the pattern for the quote is loaded at VDP address )0600. When using the Editor/Assembler setup for the VDP, the pattern for the quote is loaded at VDP address )0B00.

```

*-----*
* GPL Routine )0034 *
* Accept Tone *
*-----*

```

This routine issues a BEEP. It makes use of the sound list player in the system ROM. The sound list player can process a sound list in either VDP RAM or in GROM. The sound list for BEEP is in GROM so no VDP memory is used. As with playing any sound list, interrupts must be enabled at least once every 1/60th of a second. The sound list will not even begin to play until the next VDP interrupt occurs.

If required, your program can wait till the BEEP is finished by looking at the byte at )83CE (with interrupts enabled) until it becomes zero.

```

*-----*
* GPL Routine )0036 *
* Bad Response Tone *
*-----*

```

This routine issues a HDNK. It works in the same way as Routine )0034.

```

*-----*
*
* TITLE: GPL Routine Examples
* AUTHOR: R. A. Green
*
*-----*

```

\* PAD definitions

```

*
* FAC EQU )834A GPL Floating Accumulator
* SCTR EQU )83CE Sound list counter byte
* KBNO EQU )8374 Keyboard number for KSCAN
* KEYVAL EQU )8375 Key value from KSCAN
*

```

```

* REF RAGLNK GPL Link
* REF VMBW,KSCAN
*

```

\*EXAMPLE 1: Load character definitions for the BASIC or XB environment.

```

*
* LI RO, )0400 VDP address for blank
* MOV RO, @FAC To FAC
* BLWP @RAGLNK Call my GPLLNK routine
* DATA )0018 To load )20 to )5F characters
* LI RO, )0600 VDP address for quote
* MOV RO, @FAC To FAC
* BLWP @RAGLNK Call my GPLLNK routine
* DATA )004A To load )60 to )7E characters
*

```

\*EXAMPLE 2: Load character definitions for the E/A, MM or TI Writer environment.

```

LI   RO, )0900      VDP address for blank
MOV  RO, @FAC       To FAC
BLWP @RAGLNK       Call my GPLLNK routine
DATA )0018         To load )20 to )5F characters
LI   RO, )0800      VDP address for quote
MOV  RO, @FAC       To FAC
BLWP @RAGLNK       Call my GPLLNK routine
DATA )004A         To load )60 to )7E characters

```

```

*
*EXAMPLE 3:  Load large capitals definitions for game
*            title screen in the XB environment.
*

```

```

LI   RO, )0400      VDP address for blank
MOV  RO, @FAC       To FAC
BLWP @RAGLNK       Call my GPLLNK routine
DATA )0016         To load )20 to )5F characters

```

```

*
*EXAMPLE 4:  Prompt user with a BEEP then wait for
*            key press.
*

```

```

LI   RO, )02C2      Bottom line of screen
LI   R1, MSG1       "PRESS ENTER TO CONTINUE"
LI   R2, 23         Length of message
BLWP @VMBW         Write message to screen
BLWP @RAGLNK       Call my GPLLNK routine
DATA )0034         To start a BEEP
WAIT1 LIM1 2        Allow interrupts for BEEP
      LIM1 0        Stop them
      MOVB @KB, @KBND Set keyboard number
      BLWP @KSCAN   Check for key press
      CB @KEYVAL, @ENT Was key ENTER ?
      JNE WAIT1    Jump no, wait

```

```

*
*EXAMPLE 5:  Issue error message with HONK
*            and wait till it finishes
*

```

```

LI   RO, )02C2      Bottom line of screen
LI   R1, MSG2       "INVALID INPUT"
LI   R2, 13         Length of message
BLWP @VMBW         Write message to screen
BLWP @RAGLNK       Call my GPLLNK routine
DATA )0036         To start a HONK
WAIT2 LIM1 2        Allow interrupts for HONK
      MOVB @SCTR, RO HONK finished yet ?
      JNE WAIT2    Jump no, wait
      LIM1 0        Stop them

```

```

*
*DATA

```

```

*
MSG1  TEXT "PRESS ENTER TO CONTINUE"
MSG2  TEXT "INVALID INPUT"
KB    BYTE 0          Keyboard number
INT   BYTE 13        ENTER key value
*

```

```

END

```

-----  
C L O S E F I L E  
-----

It's been a mad rush as usual to get this issue of TI-LINES out before the turn of the century. I shall be away over the Christmas period which means that matters have no chance of improving in the early part of 1986, at least as far as delivery is concerned. I still have huge tracts of text which need proofing and rewriting several times, as well as half-formed plans for all sorts of exciting projects.

Talking of exciting projects, GORDON PITT of the West Midlands TI User Group is currently embroiled in several projects (you may remember that he is the source of supply for the cases and psus for my standalone drives). One is the liaison with Micronet, which may bear fruit in the not-too-distant future with a section devoted to the TI. Another is the advertisement campaign which he is conducting in the Midlands in an effort to shepherd more TI owners into the fold. Yet another is a project to produce a light pen capable of being operated from the joystick port. I am told that the first units have been built and are now undergoing field tests. I have no information on price, but I believe that software to run the pen will also be available. Contact Gordon on 0922 476373 for further details.

I would like to take this opportunity to thank Gordon and his right hand man, TREVOR DAVIES, for the unstinting work that they have done in promoting both OTIU and the TI generally.

Grateful thanks are also due to JOHN MATTHEWS, who has provided much-needed assistance in the production of TI-LINES, and without whose aid the larger issues would not have appeared.

JENNY KEARNE too deserves a massive vote of thanks, for without her typing skills I would certainly have been unable to produce many of the articles over the last twelve months.

The list goes on and on. STEPHEN SHAW, CLIVE & AUDREY SCALLY, PAUL DICKS, RICHARD BLANDEN, ALLEN BURT, IAN MARTIN, PHIL MARSDEN, STAN DIXON, ALAN DAVEY, NEIL LAWSON, IVAN NIBUR, MIKE BRICK, VINCE COHEN, JOHN RICE, and many, many more. It would take a whole issue to list those to whom thanks (and in some cases, cash!) are owed, and even then I'd probably miss someone out. Even my bank manager would figure in the list!

OTIU is currently 110 strong and still growing, bit by bit. I am slowly working through a mailing list, although the response has been very poor to date: 2 out of 360 have actually answered my circular.

We welcome some new members this month, too: a warm Hello to MUKUND RAJPARA, PETER CALCROFT, B. SPARROW, TONY SLATER, and P. ILLINGWORTH. I am told that there is likely to be another meeting next year in LEEDS and as soon as I have some firm details I will pass them on.

In the New Year I hope to establish contact with more User Groups overseas, and to provide more material for the novice User or minimal system owner.

Have a Merry Christmas and make a start on a Happy New Year,



Peter "All I Want For Christmas Is A Full Set Of Teeth, Some Hair On The Balder Bits, And Have You Got Anything For Piles ?" Brooks