LLEY
99ERS NEWS

TI 99/4A

HOME COMPUTER NEWSLETTER

NOVEMBER
1986

TEXAS INSTRUMENTS
Newcastle
& The Hunter Region
Home Computer
USERS' GROUP

REGISTERED BY AUSTRALIA POST
PUBLICATION No NBG8023

The Secretary - HV'99ERS
4 Arcot Close, TARRO - NSW.
Australia - 2322

YOUR COMMITTEE

PRESIDENT
Allen Wright
77 Andrew Rd.,
VALENTINE 2280
Ph. 468120

VICE PRESIDENT
Jim Grimmond
31 Jarrett St.,
TORONTO 2283
Ph. 595751

SECRETARY
Albert Anderson
6 Arcot Close,
TARRO 2322
Ph. 662602
Viatel [illegible]6626020

TREASURER
Brian Rutherford
9 Bombala St.,
DUDLEY 2290
Ph. 498184

SOFTWARE LIBRARIAN
Alan Lawrence
35 Bayview St.,
WARNERS BAY 2282
Ph. 486509

PUBLICATIONS LIBRARIAN
Paul Mulvaney
26 Marmong St.,
MARMONG POINT 2284
Ph. 583623

EDITOR
Brian Woods
9 Thirlmere Pde.,
TARRO 2322
Ph. 662307

COMMITTEE
Bob MacClure
75 Deborah St.,
KOTARA SOUTH 2288
Ph. 437431

Gary Jones
53 Janet St.,
JESMOND 2299
Ph. 573744

Tony McGovern
215 Grinsell St.,
KOTARA 2285
Ph. 523162

Peter Coxon
25 Reserve Rd.,
WANGI WANGI 2267
Ph. [illegible]

CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.
Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX 80 or DIS/VAR 80). A suitable Public Domain word processor program will be supplied if required by the club librarian Al Lawrence.
Please include along with your article sufficient information to enable the file to be read by the EDITOR eg. File Name etc.
The preferred format is 35 columns and page length 66 lines, right justified.
All articles printed in HV99 NEWS (unless notified otherwise) are considered to be PUBLIC DOMAIN. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.
Articles for publication can be submitted to.

THE EDITOR
HV99 NEWS
9 THIRLEMERE PDE.,
TARRO 2322

General address for ALL other club related correspondence.

THE SECRETARY
HV99 USER GROUP
6 ARCOT CLOSE,
TARRO 2322

DISCLAIMER

The HV99 NEWS is the official newsletter of the HUNTER VALLEY NINETY NINE USER GROUP. Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result of applying such information.
TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.
HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS

PRESIDENTS POOL-SIDE CHAT

Why won't the T.I.99/4A die?. This is a question that I keep asking myself. I am sure that no one person could honestly say that they had the entire answer to that question.

I will burden you with my thoughts on the topic, please feel free to add your comments via the Newsletter. First for the Basic and Extended Basic user, the machine if VERY User friendly. Those people who have tried some of the other popular Basics will be only to aware of the ease of use T.I. Basic offers. This is not withstanding the fact that T.I. Basic IS slow!

Extended Basic offers features that are generally not available on other Basics & when they are offer they are usually heralded as a major achievement! Some of these are the SPRITES, USER WRITTEN SUB PROGRAMMES and the ability to easily LINK to ASSEMBLY Language routines.

The construction of the machine is beyond doubt of a level that I cannot see ever being repeated until perhaps one day, when T.I. re-enters the Home Computer Market.

The same applies to the P.E. Box and it's associated cards manufactured by T.I.

On top of all this is the seige mentally which was created in all T.I. owners by the Popular Computer press here in Australia. Mis-informed, misguided, shoddy and sometimes down right biased reporting on the T.I. caused people who had purchased a T.I. to question their own judgement. Many owners opted out, others stuck with the little T.I. in the face of this and the unwanted comments from Commodore, Apple and owners of other more fashionable computer owners.

The upshot of all this is what has been left? A band of staunch survivors. They are in little pockets of Users, and can be found all around the world in Groups such as the H.V.99'ers. For the Groups to survive into the future the Groups must meet the needs of their members.

For User Groups to survive, be they for the T.I, Apple, Microbee or what ever, it is essential that the Group should always reflect the wishes, wants and aspirations of it's Members. That is a fact as basic as night follows day. The H.V.99'ers set this as a primary goal. Our Newsletter is the principal tool in our attempts to achieve this goal. An essential ingredient in the Newsletter doing just that is USER INPUT. Each person has something either to share or some question they would like to ask. The Newsletter is the ideal forum for this to occur as it maximises the sharing process. For our members who cannot get to the Monthly and weekly meetings the magazine is of critical importance.

Also important is that those Members keep in contact with the Group via letters to the Editor, small articles or asking questions for publishing in the Newsletter. The Newsletter also offers opportunities which are not often available ie to write an article for a Magazine. Writing an article is not an easy task but it is something that should be experienced. If each member contributed one article every 12 months, apart from making the Editors job a lot easier, the Newsletter could then be said to TRULY reflect the Group's Membership.

The principles I have expressed above also apply to the Committees of the any Community based Group. The Committee should be representative of the Membership. It is critical that each year some new people with fresh ideas and aims should be added to the Committee. I am not advocating wholesale changes to the Committee each year but a steady turn over. Each year a couple of new people in and a couple of oldies out. Doing this creates a continuity of purpose for the Group hd yet still produces new ideas and ims. To use one of my NO NO words DWEVER! these new people onto the bmmittee must be encouraged and llowed to express their views, pinions and ideas. It is the job f the Club President to ensure that his occurs, preferable in such a ay that it is done without notice o that is seems a natural and ormal process.

often think of a saying my father ho passed away 4 years ago used egularly. When referring to people n general he would advise me to 'always give a bloke a fair go". What did he mean?. Don't write anybody off without fair trial.

Well that is enough for now, from the idealistic President. I have to leave something to say in the Christmas issue to which I hope you will be contributing.

SECRETARYS NOTES

Hello once again from the mailbox and office of HV99. As summer approaches the usual drop-off in computer use is expected but as yet has not occurred in this part of Australia. Meetings have been well attended, articles for the newsletter come forth and both hardware and software fields seem to have their share of the to and fro's within the group.

The mailbox of HV99 received an upgrade during the month in the form of a larger more secure version as the old wooden structure just refused to sit on the post any more. A TKO by a kid on a skateboard didn't help either! Memberships continue to arrive, and to those new members we say welcome and as you've probably been earbashed about writing an account of your experiences as they happen with regard to the 4A by Mr. EDITOR, I won't repeat it; I shall just offer a nice invitation to do so. The experience is both rewarding to you and those lucky devils who happen to consume it.

We have openeo our doors a little wider as of this month and now have the priviledge of contact with our counterparts in West Germany through a group called Workshop Rheinland. To Mike Heuser and the folk in the FRG - welcome to HV99. I feel both groups should benefit from future exchanges of information. We are lucky in this respect that contact from this end can be translated by Mike. Letters and the like have so far arrived in English, thanks to Mike however I do expect some German written material, so if any-one could help with the translation could they get in touch with me. Also, not connected with Workshop Rheinland and quite by coincidence we have a new member from Germany. I'll take this opportunity to say welcome to Frank and Gisi Phillips in Heilsbronn, West Germany. We will be in touch soon with your requests.

Whilst on the overseas front, HV99 has been in touch with TI in Lubbock Texas and through their Cosumer Relations officer Donna Hall, HV99 is to be included on TI's listing of registered user groups. This will allow us access to releases of software and the like that are expected to be made available by TI in the near future. In the letter from Ms. Hall she says that this service will be provided to 'registered user groups' only, so if your group is not 'registered', better put pen to paper and write to Donna Hall, Consumer Relations, Texas Instruments - P.O.Box 53, Lubbock, Texas 79408. if you want to see what TI might have to offer in the future.

On to the local scene for a while! Gary Wilson, a new member from Menai in Sydney has provided us with some info on the availability of a suitable PC board for use with the 32k RAM expansion project. The board is from an ETI project which was for the memory expansion of the VZ2ØØ computer (I am only telling you where it came from!), and Gary informs me it suits our needs. The board is available from RCS Radio, in Bexley, Sydney and costs $8.9Ø + postage. Gary is also a HAM Radio operator and has linked his 4A to his rig by some devious means and would like to hear from any-one interested in this sphere of computer communications. An article from Gary on this subject is expected soon, so stay posted.

We have had several requests of late from people wanting to know how to go about video signal conversion for connection of both an RGB Drive monitor and converting the old tele to a monitor and using the Composite Video signal that is provided for the modulator. Some of our hardware hackers are working on these problems using previously published articles from TIsHUG's Techo Time and the 4a manual but we've hit some snags. Can any of you great user groups out there throw some light on this video signal processing area. Charles Bagley from Qld and many others would appreciate it.

The 3 Slot PE Box PC boards are expected any tick of the clock and as soon as I have them, those that ordered will be informed. We have ordered a couple of extras so if your interested get in touch with me. More detail on price etc. when I get the invoice in my paws.

Members will soon have access to a pair of double sided disk drives when they arrive from the supplier and stick them in a suitable box. This virtually sees HV99's full system realised - it's there for YOU to use; all we ask is that you look after it whilst you have it. If you have any doubts about what is available for you to borrow from the club just ask one of the committee. Also don't forget the publications library - theres heaps of other really good user group newsletters for you to read and keep up with the 4A explosion of information. What's more ITS FREE for you as part of your membership; use it!

While I'm appealing to the membership, if any-one has ideas of a venue for our end of year non-computer, get-together could they please SPEAK UP. There is a notice with regard to this elsewhere in this issue but the proposed date is around the 12-13 December.

OK, then I think that I have ran out of subject matter. Thanks once more to those that have helped and those that continue to help keep the old 4A firing on all 16.

WHATS ON

WORKSHOP-PARTY

Saturday, 29th November is the date of the final Workshop for the year. Judging by past Workshops, they are too good to miss. From what my spies tell me some people don't want to go home no matter what the time. It seems that Adventurers really are a breed apart - right Rodney and Geoff? The fun begins at 1pm at the Warners Bay High School with all the usual activities plus competitions for the kids. From about 5pm a bar-b-que will be held for all the family. Chips, lollies and drinks will be provided by the club for the children , the adults being asked to bring their own meat, drinks etc. It promises to be a great family afternoon, so come along and join in the fun.

DECEMBER MEETING

The final meeting for the year will be on Tuesday, 9th December commencing at 7pm. Gary Jones will show how to fit a Load Interrupt Switch in the Speech Synthesizer and then a demo of some screen dump programs will be given. Don't forget that this will be the last chance till February to borrow from the module and publications libraries, so make a point of being there.

MICROpendium NEWS

MICROpendium announced in their October magazine that REGINA, famous for her articles in the 99'er Magazine (R.I.P.) and COMPUTE! will be writing articles on BASIC commencing in January. This lady is absolutely brilliant, so don't miss reading her articles. MICROpendium is available each month from Paul Mulvaney, publications librarian each month.

While on the subject of MICROpendium, make the effort to read this mighty magazine , it contains all the latest news on the TI as well as articles and reviews on software and hardware available currently.

EXPLORING BASIC

The article this month will seem a little different than usual. Our regular writer Alan Franks is in the process of extending his family home. He will be back with a vengeance next month I can assure you. I can also tell you that Alan would love to get some feedback from his articles.

Down to business, our solution for last months problems are below.

Solution Problem 4-1.
****

```
10Ø DIM SCORE(1Ø)
11Ø CALL CLEAR
12Ø FOR A=1 TO 1Ø
13Ø PRINT "IMPUT No.";A
14Ø INPUT SCORE(A)
15Ø NEXT A
16Ø CALL CLEAR
17Ø FOR A=1 TO 1Ø
18Ø PRINT SCORE(A)
19Ø NEXT A
20Ø END
```

Solution Problem 4-2
****

```
10Ø DIM SCOPE(1Ø,2)
12Ø CALL CLEAR
14Ø FOR A=1 TO 1Ø
16Ø PRINT
18Ø   PRINT   "ENTER   STUDENT   A'S
RESULT";A
20Ø INPUT SCORE(A,1)
22Ø PRINT
24Ø   PRINT   "ENTER   STUDENT   B'S
RESULT";A
26Ø INPUT SCORE(A,2)
28Ø NEXT A
30Ø CALL CLEAR
32Ø PRINT " EXAM RESULTS"
34Ø PRINT
36Ø PRINT "STUDENT A","STUDENT B"
38Ø FOR A=1 TO 1Ø
40Ø PRINT SCORE(A,1),SCORE(A,2)
42Ø NEXT A
44Ø END
```

PROBLEMS.
****
Two problems have been set for this month.

5-1.
Modify the result of problem 4-2 to input and show results for 5 students.

5-2.
Now modify the programme again to allow each students name to be entered with his results. Also show in addition to each students name and results, his average score when printing the results.

LOOKING AHEAD.
****

Alan has asked me to mention that the Basic Group will be embarking on a project to write a programme to record information such as addresses, phone numbers etc. He intends to use this development of the programme as questions in his articles. The intention being over the following month to write the programme via the use of questions and answers which will be combined to form the complete programme. I would ask you to support Alan in the project. He will be showing the design requirement of the programme next month. This will be followed by the sectional and overall flow diagrams.

If you would like to have a say in what the programme will record then please do so, or if you think of something that has been missed then please pass the information on to Alan.

Until next month for and on behalf of Alan.

Joe Wright.

THE PRINTER BUFFER

GEOFF WARNER

The worst part of Word Processing, or in fact any printer-related task, is waiting for the printer to finish its work so that you can back on the job with your computer.

This wait can be quite long, and depends upon factors such as the length of your document (an obvious fact, you say), the speed of your printer and the print mode to which it is set. If you were logged on to a remote system that charged for access time this wait could also be quite expensive!

We can overcome this problem with a PRINTER BUFFER, a wonderful little device that allows you to make the time you spend waiting for your printer to finish more productive, by ACCEPTING ALL OF THE DATA TO BE PRINTED IN ONE GO, rather than piecemeal, at a rate dictated by the printer. When the buffer has accepted all of the data to be printed it releases the computer to your control and you can continue with something more useful or entertaining (or both!). Meanwhile the printer and the printer buffer carry on communicating with each other at a rate dictated by the printer, churning out hard copies of the fruits of your labour, oblivious to the fact that you are now creating a new Parsec record for the Southern Hemisphere!

Apart from this primary (and fairly obvious) feature, any printer buffer worth its salt should offer the following features:

COPY FACILITY - allowing you to make single or multiple copies of your document.

SOFTWARE PAUSE - where you include a special character in your text which halts printer output when detected and thus allows you to set up page lengths, change print styles etc. Ideally you should be able to over-ride this facility if you wish.

RESET - an obvious requirement which enables you to clear the buffer and make it ready to accept further data for printing.

SELF TEST - all of the funrtions of the buffer should be able to be checked via built-in diagnostic software routines to enable you to track down possible problems, without having to resort to mega-dollar service organisations.

EXPANDABLE BUFFER SIZE - to enable you to take advantage of any hardware upgrades without having to invest in a larger buffer.

SELECTABLE INPUT AND OUTPUT PORTS - to let you use any combination of serial and parallel computer and printer.

Does all of this sound like
(i) It's too good to be true?
(ii) It's what you've been looking for all your life?
(iii) It costs an arm and a leg?

Well, I have just finished construction and testing of the basic parallel-parallel version of a printer buffer that offers all of the 'essential' features listed above, PLUS optional Printer-sharing and computer-sharing facilities and a Hexidecimal Output Mode for those heavy programming perverts amongst us.

The final cost I haven't yet determined, as I have a well stocked junk box, but the basic version with 32K of buffer space installed would probably cost around $100 (at W.A. prices) providing that you shop judiciously and know which end of a soldering iron to hang on to (i.e. you don't stuff anything up).

Anyhow, onto the nuts and bolts of the thing. This buffer, hereinafter called "PBUFF" ('cause that's what its creator calls it) comes in a short-form kit which consists of a plated, single-sided PCB and an EPROM in one of three versions; 8K to 64K Buffer Memory size, 256K Buffer Memory size, or both. The latter option costs an extra $3.00. The producer will also sell you ALMOST anything else that u need to complete the project. me of his prices are excellent, me not so, and he is the first to mit it!

The assembly instructions that me with the PBUFF kit are written a way that ALMOST anyone should able to complete it with a nimum of problems. They are a ttle pedantic but they are aimed the average non-technical bbyist constructor, not us -technician types.

I have spoken to a guy who ran nto problems with his (he is a on-TI user of course) and spent ega$ on phone calls to Melbourne rying to find out what he had done rong. He informs me that the roducer of the kit was VERY helpful you can afford to be if someone lse is paying the STD rates!) and he problems were eventually sorted ut.

In the interests of economy, he PCB is single-sided, which means 2 links to be installed, but the oard is neat and reasonably well aid out, even if some of the esistors are standing on end. I ould suggest a photocopy of the omponent layout and some highlight' pens to help you keep rack of the links, since some of hem go underneath ICs (or their ockets) and would be a mongrel to it if you happened to leave one ut.

The instructions to get you up and running are spot-on, and utilize the built-in diagnostic routines to spot RAM faults etc. Using these instructions, the 'it doesn't work!' department, the short dissertation on Centronics signals and the pin-outs for chips supplied, you should be able to track down anything you may mess up and eventually end up with a working printer buffer.

So, what do I think about the PBUFF? In a word, GREAT! You install it in between your interface and your printer and forget about it. When you print, you get your computer back almost as soon as it takes you to check out the Data LED, work out what's going on and marvel at the speed in which a lengthy article like this one can be loaded.

If you want a single copy of your masterpiece, switch the copy switch on and then off again and you are on your way. If you want multiple copies, leave the copy switch on and your printer churns them out until you turn the switch off again, or run out of paper.

You want to check the PBUFF before you entrust your hard work to it? Press the Test switch and it outputs a sign-on message complete with buffer memory size installed:

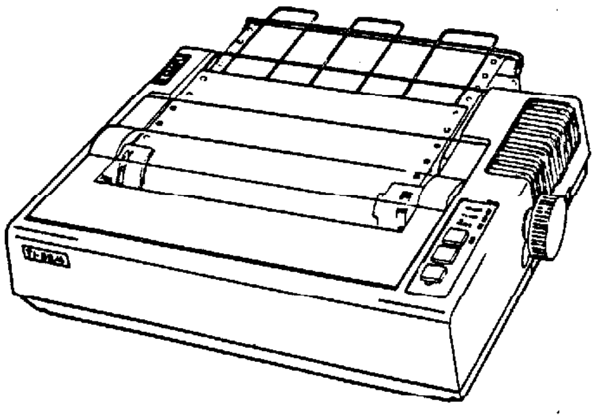
```
##PBUFF Version 2.2 ## by Don McKenzie.  Jan 86(C)
29 Ellesmere Cres., Tullamarine AUSTRALIA 3Ø43
Phone (Ø3) 3386286

32K RAM INSTALLED
```

In summary, a worthwhile project if you do a lot of printer-related computing and get tired of waiting for the printer to do its thing. I use it a fair bit in editing a newsletter for a Cricket Club and a Basketball Club. My next project is to build the optional Serial Board and thus eliminate one of my growing number of plug-in peripherals.

The kit is available from:
DON McKENZIE
29 ELLESMERE CRES.,
TULLAMARINE VIC. 3Ø43

phone (Ø3) 3386286

If you send him a self addressed envelope, he'll send you a price list and brochure on all of his products. If you send him $2.ØØ, he'll send the manual and deduct the $2 from the price of the kit if you buy it.



FORTH UTILITY PROGRAM

My aim in writing this article is to offer an alternate point of view on Forth. I will be discussing code that I have written that others may use as a guide to getting started or further into Forth.

This article is not intended to replace or take over the series of FORTH articles written by Richard Terry. However at this point I would like to thank Richard, and also all the other authors of articles for the HV-99ers News, for the great job they have been doing in producing so many interesting articles.

Why use Forth? Well one of the main advantages Forth offers to the programmer is its Modular programme structure. What this means is that once a routine, or word, has been defined and tested then it can be used by any other routine by passing to it any required parameters on the stack and then referencing it by its name to start it executing.

The programme shown in this article shows an example of two seperate routines, both comprising several words, being combined to produce a useful utility programme. Although these words are used in this programme they are not restricted to this programme. Once these words are compiled into the dictionary they can be used by any other words that are later defined.

EXPLANATION

For those people who have Corcomp or Myarc disk controllers you have available up to 36Ø FORTH screens per disk (1 screen=1Ø24 bytes) using the DS/DD capabilities of these controllers. The standard Texas Instruments disk controller and standard drives will support only 9Ø FORTH screens per disk. Thus for those people with controllers that support DS/DD disks each disk can store all the 8Ø odd TI-FORTH source code screens, your BSAVE'd compiled options (e.g. -EDITOR, -PRINT, -FILE etc), and then have up to 26Ø screens available for your own source/object code. This arrangement has the advantage of not needing to have a multi disk drive arrangement or not having to be swapping disks in and out of the drive on single disk drive set ups. The only disadvantage to date has been that the higher density disks are not compatable with other users who only have SS/SD and hence screens were not readily transportable between machines.

I therefore decided to overcome this problem and the resulting program is the subject of this article. This programme will transfer any contiguous block of screens from any format disk (assuming the appropriate disk controller is connected) to another disk of any format to any screen number. The programmes use is not however restricted to copying between disks of different formats but can also be used to copy a block of screens on a disk to another block on the same disk. Also it enables screens to be copied from one disk to another of the same format if it is a single disk drive arrangement. This will reduce the time spent in backing up work as the whole disk does not have to be copied every time .

Note that this programme is really only of significant benefit for single disk drive arrangements as the existing SCOPY and SMOVE words can copy from drive to drive if the disks they contain are of different formats.

The need for a programme of this type arises because on single disk drive arrangements all of the multi screen copy FORTH words will not allow you to change disks in the middle of the screen transfer. Also most of the single block, or screen, read/write words have their own error detecting routines and therefore abort if any sort of disk error occurs. So what is needed are two primitive words that read to or write from disk a block of bytes and return on the stack a flag dicating if the operation was ccessful or not. This type of erator is needed because, in the se of the Corcomp controller, if u change disk formats from one ad/write to another the controller ll not recognize this fact on the xt read/write attempt and will nerate an error. However it will rrectly read/write on the second y. Thus if disk formats are hanged in between a read and write he primitive operators described bove will on the first try give an rror, but the second attempt should rite correctly. If it is not uccessful on the second try then ome other disk error may be the ause and the programme will stop.

A listing of the programme is ncluded with article. As described arlier this programme is made up of wo routines. The way in which one outine ensures that the any support ords are loaded is shown on line 13 f screen 22Ø. This loads the ntaccept routine.

NTACCEPT

This routine was written and ested separately from MSCOPY. The irst draft of this routine was riginally written when I was efining basic input words.

It offers an alternative to the raditional method of getting an number from the operator. This technique is to receive all key strokes until the enter key is pressed and then check if the text entered can be converted to a valid integer. The standard FORTH text to number conversion routine does rot like alphabetic characters embeded in text that it is trying to convert to an integer and aborts if it finds any. Therefore if the traditional method is used any alphabetic characters will cause the programme to abort which I do not consider to be user friendly. The method I have used to overcome this problem is to trap all key strokes and reject all characters which would not normally comprise an integer. The penalty that has to be paid is that it generates more code i.e. uses more memory.

This routine could be combined into another programme by simply including line 13 of screen 22Ø.

If a routine is required to accept floating point numbers then INTACCEPT will do the job if the \ is removed from the start of line 8 on screen 235. This will allow a decimal point to be entered.

RUNNING THE PROGRAMME

To run this programme 12K of memory is required (1ØK for the screen buffer). To find out how much memory is left free enter:

```
SP @ HERE - .  <enter>
```

If there is enough memory free then type the following to run the programme:

```
MSCOPY <enter>
```

Then follow the instructions as prompted. If there is not enough free memory then probably some of the menu options that have been loaded will have to be forgotten.
e.g. type:

```
FORGET BOX <enter>
```

to remove the editor from the dictionary.

Improvements to this programme would be:
- to have the screen number displayed on the screen as it it was being read from or written to disk.
- use the spare and disk buffer VDP memory for the screen buffer instead of defining a buffer in the dictionary. This would overcome the speed penalty resulting because of the data transfer to expansion memory and back to VDP memory.

Please note that I have only tested this programme out on the Corcomp controller but I assume that it should work on the Myarc controller with no problems.

GENERAL

Throughout the listing I have used \ to indicate the start of a comment instead of the usual (. This is because \ does not need a end of comment terminator (e.g. (needs a) at the of the comment). This word is defined on screen 22Ø and operates by forcing the interpreter to stop interpreting the current line of that screen and continue on the next line.

Those members of HV-99ers who have DS/DD, DS/SD or SS/DD controllers and suitable drives and would like to take advantage of the extra screens that are available to them can do so by either getting the information from the librarian (see Sydney News Digest, July 1985 - Pages 6 and 7) or by contacting me and getting a copy of one of my formatted disks.

KEITH BRUCE   PH. 547258
14 PASADENA CRES.
CARDIFF   2285
NSW

```
SCR #220
( MSCOPY                                   27/05/86  )

: \   IN @ 64 / 1+ 64 * IN ! ; IMMEDIATE

: 2DUP  OVER OVER ;

: AT    GOTOXY ;              \ ( c r -- )

: BLOT  40 SWAP - SPACES ;    \ ( n -- )  n=start column
                              \ erase n chars from current line
: UPC   DUP 96 >              \ converts char to upper case only
        IF DUP 123 < IF 32 -  \ if it is a lower case char
        ENDIF ENDIF ;         \ ( n1 -- n2 )
230 CLOAD N                   \ load intaccept words

-->
```

```
SCR #221
\ MSCOPY                                   27/05/86
0 VARIABLE TEMP_BUF 10240 ALLOT
0 VARIABLE S.STRT 0 VARIABLE S.END 0 VARIABLE D.STRT

: POS1   0 22 2DUP AT 0 BLOT AT ; \ position cursor on cmd line

: Y/N                         \ ( -- T if yes F if no )
    KEY UPC DUP 89 =          \ is it Y
    SWAP DUP 13 =             \ is it ENTER key
    ROT  OR                   \ either Y or ENTER
    SWAP DUP 78 =             \ is it N
    ROT  OR IF                \ either Y or ENTER or N
         78 = 0=              \ yes, leave false flag if N
       ELSE MYSELF ENDIF ;    \ no, repeat until a valid key
-->                           \ is entered

```

```
SCR #222
\ MSCOPY                                            27/05/86

: READY?
     POS1 ."   Are you ready (Y/N)?"  \ display prompt
     Y/N ;

: INSERT_COPY   1 20 AT ." Insert Destination/Copy disk "
                READY? ;


: INSERT_MASTER 1 20 AT ." Insert Source/Master disk    "
                READY? ;



-->
```

```
SCR #223
\ MSCOPY                                            27/05/86

: READ_SCR TEMP_BUF + SWAP 1024 RDISK ; \ ( scr# --- flag )

: GET_SCRN 2DUP READ_SCR   \ ( scr# buffer.offset -- )
        IF READ_SCR       \ if it failed then try
           IF ." 2nd read failed "
              ABORT
           ENDIF
        ELSE DROP DROP    \ get rid of scrn no.
        ENDIF ;
\ GET_SCRN tries to read scr# from the disk twice. If it fails
\ on the 2nd try it prints an error message and aborts.
-->


```

```
SCR #224
\ MSCOPY                                            27/05/86

: WRITE_SCR TEMP_BUF + SWAP 1024 WDISK ; \ ( scr# --- flag )

: RIT_SCRN 2DUP WRITE_SCR   \ ( scr# buffer.offset -- )
        IF WRITE_SCR       \ if it failed then try
           IF ." 2nd write failed "
              ABORT
           ENDIF
        ELSE DROP DROP    \ get rid of scrn no.
        ENDIF ;
\ RIT_SCRN tries to write scr# to the disk twice. If it fails
\ on the 2nd try it prints an error message and aborts.
-->


```

```
SCR #225
\ MSCOPY                                         27/05/86
: COPY_OVER                  \ ( offset.scr.no --- )
 -  DUP 10 + S.END @ S.STRT  \ setup to copy 10 screens
    @ - 1+ MIN SWAP OVER     \ or until all scrns copied
    OVER 0 CNT !             \ reset counter
    DO I S.STRT @ +          \ source screen no
       CNT @ 1024 * GET_SCRN \ copy it into buffer
       1 CNT +!              \ increment counter
    LOOP
    INSERT_COPY              \ prompt operator for copy disk
    IF 0 CNT !               \ try to write it to copy disk
       DO I D.STRT @ + CNT @ 1024 * RIT_SCRN 1 CNT +!
       LOOP ELSE ABORT ENDIF
    INSERT_MASTER DROP       \ prompt operator for master disk
    0 0 GET_SCRN ;           \ test read source disk again
-->
```

```
SCR #226
\ MSCOPY                                         27/05/86
: SETUP_COPY                 \ ( no.scr --- )
       0 BEGIN               \ scr offset initially =0
           DUP               \ copy scr offset
           COPY_OVER         \ copy 10 screens
           10 + DUP          \ point to next block of 10 scrs
           S.STRT @ +        \ add start scr# to scr offset
           S.END @ SWAP - 0< \ take away end scr# & if <0
       UNTIL ;               \ stop else do again

: STRT_SCR 18 SWAP AT ." Start scr :" \ ( row# -- n )
           4 INTACCEPT ;
: END_SCR  18 SWAP AT ." End scr   :" \ ( row# -- n )
           4 INTACCEPT ;
-->

```

```
SCR #227
\ MSCOPY                                         27/05/86

: GET_SCRNS  CLS 5 1 AT ." MULTI SCREEN COPIER"
     1 4 AT ." Source disk"
     4 STRT_SCR DUP S.STRT !
     5 END_SCR DUP S.END !
     SWAP -    0< IF
           1 10 AT ." End is less than start - try again"
           1 11 AT ."    press <enter> to continue " Y/N
           DROP MYSELF ENDIF
     1 7 AT ." Destination disk"
     7 STRT_SCR D.STRT !
    18  8 AT ." End scr   :"
     S.END @ S.STRT @ - D.STRT @ + .
     ;
-->
```

```
SCR #228
\ MSCOPY                                          27/05/86

: MSCOPY  GET_SCRNS           \ get start and end screen no's
         INSERT_MASTER        \ prompt to insert source disk
         READY? 0=            \ everything okay?
         IF ABORT ENDIF       \ if not okay then abort
         SETUP_COPY           \ copy over screens 10 at time
         CLS 5 10 AT
         ." Enter  COLD  to restore system" CR CR
         ;
```

```
SCR #230
\ INTACCEPT - modified expect version             28/05/86
0 VARIABLE N    0 VARIABLE FP
0 VARIABLE CNT

: NONNUM  DUP DUP              \ ( chr -- flag ) True if numeric
         48 < SWAP 57 > OR ; \ check char is numeric

: STOREIT TIB @
         CNT @ + C! ;

: INCOUNT 1 CNT +! ;          \ increment char counter

: DECOUNT -1 CNT +! ;         \ decrement char counter


-->
```

```
SCR #231
\ INTACCEPT - modified expect version            28/05/86

: ?COUNT=1 CNT @ 1 = ;        \ ( -- flag ) true if cnt=1

: ?COUNT=N+1 N @ 1+          \ ( -- flag ) true if cnt<n+1
          CNT @ = 0= ;

: BLOCK1  ?COUNT=N+1         \ test if n no's have been entered
          IF
            DUP STOREIT      \ numeric entered - store it
            EMIT INCOUNT     \ and display it, increment count
          ELSE               \ if all numbers entered no more
            DROP             \ wanted so drop it
          ENDIF 1 ;          \ leave true flag on stack

-->
```

```
SCR #232
\ INTACCEPT - modified expect                    28/05/86

: ENTER   DROP               \ get rid of chr 13
        ?COUNT=1             \ enter key pressed
        IF 48 BLOCK1 DROP    \ if first key then number is zero
        ENDIF  0 ;

: BACKUP  ?COUNT=1           \ is this the first char
        IF   DROP            \ yes - do nothing
        ELSE EMIT 32 EMIT    \ no move cursor back one and rub
             8 EMIT          \ out char there
             DECOUNT         \ decrement char counter
        ENDIF 1 ;
-->


```

```
SCR #233
\ INTACCEPT - modified expect version            28/05/86

: NEGA    ?COUNT=1           \ minus sign entered - is it first
       IF   BLOCK1 ?COUNT=N+1  \ char - yes
       ELSE DROP 1             \ no
       ENDIF ;

: PCINT   FP @
       IF   DROP 1           \
       ELSE BLOCK1
            1 FP !
            ?COUNT=N+1
       ENDIF ;

-->

```

```
SCR #234
\ INTACCEPT - modified expect version           28/05/86
: INIT    N !                 \ store char count
          1 CNT !             \ initialize screen char counter
          0 FP !              \ floating point flag
          ;

: TIDY_UP 32 STOREIT          \ finish string off with a space
          DECOUNT             \ decrement counter
          CNT @               \ and store it at
          TIB @ C!            \ start of string
          ;


-->


```

```
SCR #235
\ INTACCEPT - modified expect version           28/05/86
: GETNUM  INIT                \ store chr count & initialize
     BEGIN KEY                \ get a char from keyboard
        NONNUM                \ is it non-numeric
        IF  DUP CASE
          13 OF ENTER  ENDOF \ enter key pressed
           8 OF BACKUP ENDOF \ left arrow key (fctn s) pressed
          45 OF NEGA   ENDOF \ minus sign entered
        \ 46 OF POINT  ENDOF \ decimal point entered
          ENDCASE
        ELSE BLOCK1           \ number entered - save & display
        ENDIF 0=
      UNTIL                   \ until true flag (non zero)
     TIDY_UP                  \ move count and string to TIB
     ;
-->
```

```
SCR #236
\ INTACCEPT - modified expect version           28/05/86

: INTACCEPT GETNUM            \ ( #chrs -- n )
            TIB @
            NUMBER            \ convert to a number
            DROP  ;           \ drop double no half










```

LIBRARIANS REVIEW

HI 99'ers,

****DISK CATALOGER****

Some people are still not sure what a Disk Cataloger is or does.

Put simply, it reads the contents of a disk and when all disks have been read, can store them (on disk) and, depending on type (Basic,X Basic, Assembly) list to screen, get a hard copy, Add, Delete, Sort, Exclude file names (eg.Load), Alphabetical, Disk by Disk in order of loading or file by file. Any number of permutations are available as are the number of Disk Cataloging programs around. Some simple ones capable of holding only a few disks with a small number of titles, which can suddenly fill up and your program bails up!!! You then say Sweetly Sunshine? count up to 99 and load one of the ones you paid heaps of the folding stuff for, but then it does the same! So Freeware,Fairware, Commercial, Price is no guarantee of perfection or satisfaction.

The listings can be read like the index of a book but instead of page, title or chapter you get Program Name, Disk Name, easily spot all the duplications, get a Disk report & the number of sectors free on the disk.

Having the need for a good catalog, I had tried several and found them not able to do all required in the one package and so persuaded El Presidente to personalize one to do all that was needed, which is the one I then used (and is available from the club library).

However, much to my shame, I had in the club library one of the best I have used, and it is Fairware by
** Marty Kroll Jr. **
** Version 1.4 **
and is one of the few that, while it is sorting the files prior to saving to disk, politely reminds you to send $10 to the Author (if you had not already done so).

Its options include:

A. Add disk to catalog.
B. Delete disk from catalog.
C. List disk summary.
D. List all programs.
E. Search for and list a disk.
F. Search for and list a program.
G. Print catalog disk by disk.
H. Print disk summary.
I. Print all programs.
J. Print all programs-no page division.
K. Search for and print a disk.
L. Change printer options.
M. Sort and save data.
N. Change foreground/background colours.
O. Select another set of data files
P. Terminate program.

It's fast and programmed in Assembly language - Opt 3. E/A Module but can be put on your Funlwriter disk to enhance it even further.

Other assets include:

* It catalogs those "Funny Sectored" disks.
* Informs you if the disk is already on file.
* Catalog up to 123 disks / 900 files.
* Allows storage of multiple data files on the same data disk.
* Comes with full documentation and listing of other Programs by the Author.
* Select 1. 2.or 3 columns for Printer output.
* Output to Screen or Printer.

In the Docs are instructions to change the defaults for colour, Printer name and formats with one of the Sector Editors. (NB. you may even make it return to Funlwriter on Termination!!!)

So if you are looking for a topline, fast Cataloger this is the one for you and is now available from the Club Library.

Printer Pictures.

A Program called MAX/RLE - loaded using Option 3 E/A Module or Util on Funlwriter (program name START) allows loading of any of the picture files on the disk and voila! you have some finely defined pictures. Press P to dump to printer, any mber key changes background lours, and A,C,D,E,F, changes lours. Pressing "S" will allow u to transfer the picture files to RAPHX format. A disk well worth aving for all picture freaks!

HANKS - My X-BASIC Module has eturned from its wandering trip.

Fairware for now,

Happy programming,

Al Lawrence.

MOVING IN THE RIGHT DIRECTION WITH SPRITES

GARY JONES

One of the main problems ncurred while trying to come to rips with the SPRITE instructions, s getting the sprite to move in the irection you would prefer and at he speed or velocity you would refer.

The velocity of a sprite is ontrolled by the values set in the nstrucutions as shown below.The max imits for the velocity are "-127 to 28" depending on the direction and Ø" if no velocity is required.

The direction is controlled by hree factors :-

(i) Whether the DOT-ROW is a ositive or a negative value.

(ii) Whether the DOT-COLUMN is positive or a negative value.

(iii) Whether the DOT-ROW or he DOT-COLUMN is set to zero.

Both direction and velocity are set by either the CALL SPRITE or CALL MOTION instructions

E.G.

```
CALL SPRITE(#1,65,5,8Ø,8Ø,-5Ø,Ø)
```

In this instruction a letter "A" will move up the screen at a medium velocity.

```
CALL MOTION(#1,Ø,Ø)
```

If then this instruction is used the letter "A" will both cease direction and velocity.

```
CALL MOTION(#1,2Ø,-2Ø)
```

If this instruction is then used the letter "A" will now move down and to the left of the screen at a faster velocity.

The tables enclosed in the magazine were constructed by the Extended Basic Class to act as a quick reference guide when calculating the direction of a sprite.

There are two tables, one for the CALL SPRITE pages and the other for the CALL MOTION pages of the reference manual.

"Frustration."

FOR THE ADVENTUROUS ONLY

RODNEY GAINSFORD

Hi! This month we have a map of THE COUNT and a map of the Puzzle Room in ZORK III. The object of this puzzle is to push a sandstone block with a ladder on it back under the starting point to climb up and out. Watch out for the Bumper Issue next month.

Now to this month's tips:

ZORK III

* When shadowy figure is defenceless, get hood and cloak.
* Enter the mirror.
* The Gold Machine is a Time Machine.
* The vial provides one or two moves with invisibility.
* Need:

    Staff
    Cloak/Hood
    Key
    Amulet
    Book
    Ring
    Lamp

THE COUNT

* Raise the Dumb Waiter/Lower the Dumb Waiter.
* When the bell rings go outside

ENCHANTER

The Spells

SPELL	FUNCTION	LOCATION
VAXUM	Make hostile creature friendly	Bedpost
ZIFMIA	Magically summon a being	Egg
OZMOO	Survive an unnatural death	Cell
EXEX	Make things move with speed	Gallery
KREBF	Repair wilful damage	Forest
CLEESH	Change a creature to an amphibian	Swamp
REZROV	Open locked or enchanted object	House
GNUSTO	Write spell into book	Book
BLORB	Safely protect a small object	Book
NITFO	Converse with beasts	Book
FROTZ	Cause something to light	Book
KULCAD	Dispel a spell	Engine Room
MELBOR	Protect caster from evil beings	Box
FILFRE	Create gratuitous fireworks	Map Room
GUNCHO	Banish to another plane	Terror Room

GHOST TOWN

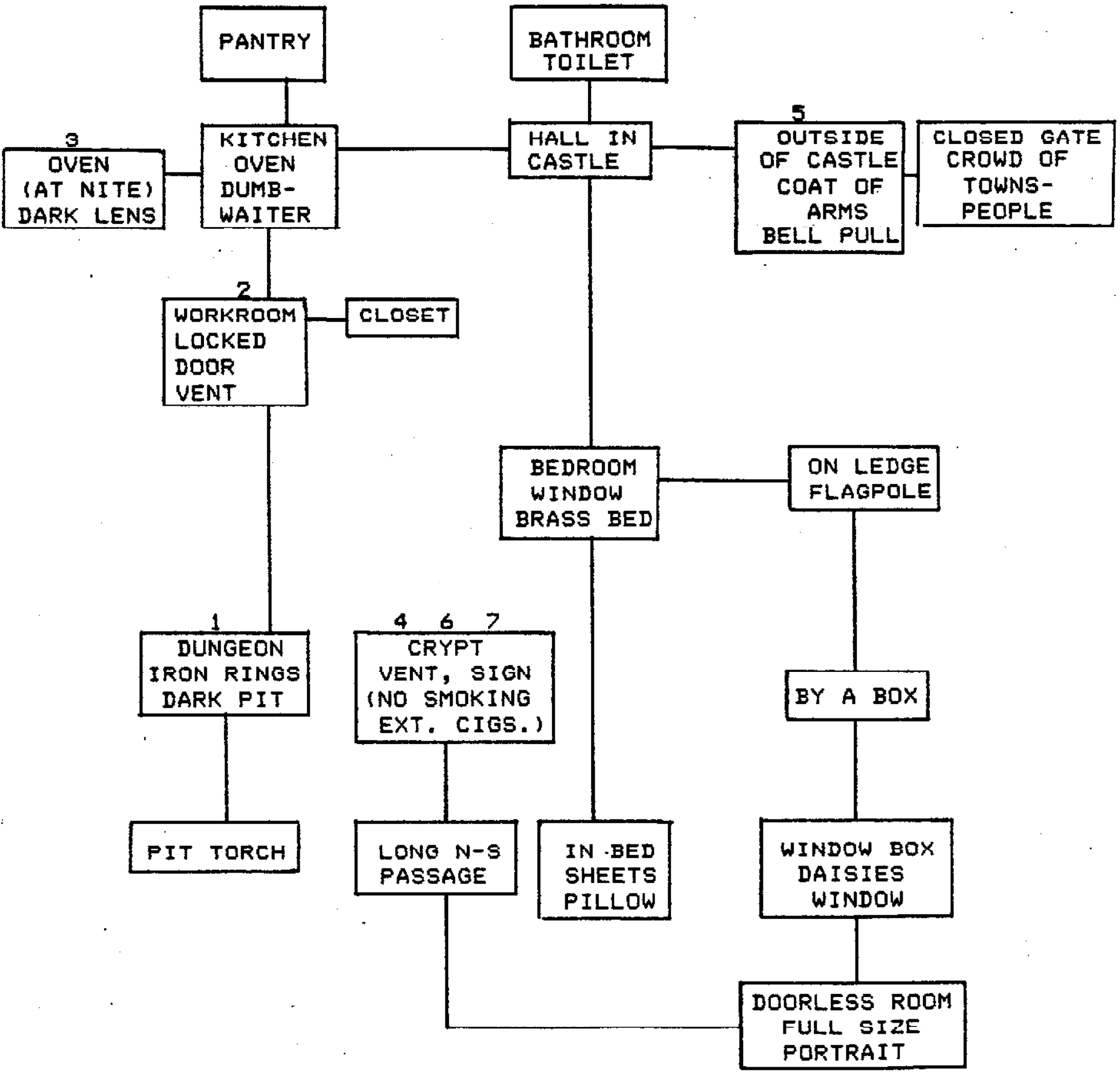
Treasure Summary

* Cash box
* Go board
* Derringer
* Necklace
* Gold dust
* Silver spurs
* $200
* Gold coin
* Silver bullet
* Gold nugget
* Silver cup
* Tom-Tom
* Furs

That's all for this month. Don't forget, please send any maps or hints to:

RODNEY G.
c/o The Secretary,
H.V. 99'ers.
6 Arcot Close,
TARRO N.S.W. 2322.

THE COUNT

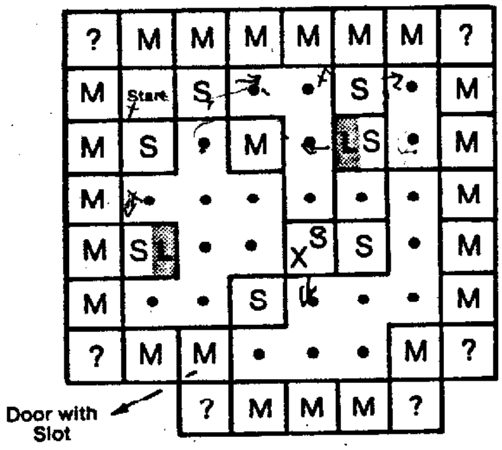

PANTRY
BATHROOM TOILET
3
OVEN (AT NITE) DARK LENS
KITCHEN OVEN DUMB-WAITER
HALL IN CASTLE
5
OUTSIDE OF CASTLE COAT OF ARMS BELL PULL
CLOSED GATE CROWD OF TOWNS-PEOPLE
2
WORKROOM LOCKED DOOR VENT
CLOSET
BEDROOM WINDOW BRASS BED
ON LEDGE FLAGPOLE
1
DUNGEON IRON RINGS DARK PIT
4 6 7
CRYPT VENT, SIGN (NO SMOKING EXT. CIGS.)
BY A BOX
PIT TORCH
LONG N-S PASSAGE
IN BED SHEETS PILLOW
WINDOW BOX DAISIES WINDOW
DOORLESS ROOM FULL SIZE PORTRAIT

Day 1:note/Clip
Day 2:Package-Letter
Cigarettes
Bottle of Blood

ZORK III

Room In A Puzzle

(Push a sandstone block with a ladder on it back under the starting point to climb out)



M=Marble Block

S=Sandstone Block (Movable) 2 have ladders

X=Depression in floor with Book in it

CHECKSUM

NEIL QUIGG

What is the checksum?

The checksum is a method of error detection used in data transmission. It is used in many forms ranging from simple summation of data to complex algorithms used in large computer systems.

How is it used?

When data is stored the checksum is calculated and stored within the data. Then when the data is read from the storage medium the checksum is recalculated and the value compared with the stored value. If the values are found to differ the computer will assume that a read error has occured and issues an error statement accordingly.

How does the TI calculate the checksum?

The E/A checksum value is calculated as the "two's compliment of the sum of the 8 bit ASCII value of each character in a record from the first flag through to and including the checksum flag". This may sound daunting however it is relatively simple in practice.

Firstly consider the 8 bit ASCII sum using the screen filling example from Octobers "ASSEMBLY LANGUAGE FOR THE LAYMAN" article. Each character including the 8 blanks has an 8 bit ASCII value.

eg. A = 0100 0001 or in hex 41.

for the purpose of this exercise and ease of calculations we will use hex values for each character. Now consider the first eight characters of last months object code.

```
char 0  0  0  4  4  (3 BLANKS)
hex 30 30 30 34 34 20 20 20
```

By adding the hex values we get >0158 continuing this process through the complete record including the checksum flag 7 we get a sum of >0C9B ,so far so good.

Now >0C9B equals 0000 1100 1001 1011 in binary and for two's compliment we invert each bit and add 1.

```
So         0000 1100 1001 1011
becomes    1111 0011 0110 0100
add one                      1
total      1111 0011 0110 0101
in hex        F    3    6    5  behold
```

our checksum.

Now back to last months example. After the object code is altered to produce the character B instead of A the loader issues a checksum error. This is because when the data is loaded the checksum calculated differs from that imbedded in the data. This problem was overcome by intoducing the checksum flag 8 which instructs the loader to ignore the checksum. The alternative is to correct the checksum value. To do this let us consider the actual data change made. The object code was altered from 41 to 42 so our 8 bit ASCII values changed from 34 31 to 34 32 so our ASCII sum would change from >0C9B to >0C9C.

```
>0C9C =    0000 1100 1001 1100
invert     1111 0011 0110 0011
add one                      1
twos com   1111 0011 0110 0100
HEX           F    3    6    4
```

Now if the fill screen example is changed from character A to B and the checksum value was changed from F365 to F364 and saved the OPTION 3 LOAD AND RUN should not issue an error.

ASSEMBLY LANGUAGE

FOR THE LAYMAN

ll another month is upon us. mmer is fast approaching, companied by all the things sociated with an Australian mmer. Flys as big as 747's, hot sterly winds on the east coast and r beautiful beaches. Ah! The ctar of life, an early morning roll along a deserted beach.

his month's article will be a bit orter than normal, I apologise for his and promise to have something r you to get you teeth into for e December issue.

st month I introduced the VDP RAM cess utility VMBW. Before oceeding any further READ pages 46 to 249 in the E/A Manual. These ges describe the Utilities vailable for you to read from and rite to VDP RAM. Referring to ROGRAMME 1 in last months article.

te that VMBW has been included in he REF on line ØØØ5. The next 2 ines contain the text which I want o display on the screen. On line Ø17 RØ is loaded with the address n VDP RAM where I want the Text to tart from.

n line ØØ18 the location address of he text to be written to VDP RAM is oadeu into R1.

n line ØØ19 the number of haracters to be written to VDP RAM s loaded into R2.

n line ØØ2Ø the Utility VMBW is ccessed using BLWP.

ach time the Utility is used the egisters RØ,R1, and R2 are loaded ith the correct data then BLWP to he Utility.

HE SCREEN.
****

ne of the things I do find ifficult is deciding where on screen to write the text. Then after deciding where, comes determining the correct memory location in the screen image table. Many articles have formulea showing how to calculate the screen image table location.

Unfortunately I also find this not entirely suitable either so I have invented the Joey Wright or Laymans method, a bit longer but effective. It is also patently simple.

I have a sketch of the screen image table on the wall to my right set out as follows.

SCREEN IMAGE TABLE.
****

LINE NUMBER	FIRST POS.	LAST POS.
Ø1	>ØØ	>1F
Ø2	>2Ø	>3F
Ø3	>4Ø	>5F
Ø4	>6Ø	>7F
Ø5	>8Ø	>9F
Ø6	>AØ	>BF
Ø7	>CØ	>DF
Ø8	>EØ	>FF
Ø9	>1ØØ	>11F
1Ø	>12Ø	>13F
11	>14Ø	>15F
12	>16Ø	>17F
13	>18Ø	>19F
14	>1AØ	>1BF
15	>1CØ	>1DF
16	>1EØ	>1FF
17	>2ØØ	>21F
18	>22Ø	>23F
19	>24Ø	>25F
2Ø	>26Ø	>27F
21	>28Ø	>29F
22	>2AØ	>2BF
23	>2CØ	>2DF
24	>2EØ	>2FF

Finding the address of a particular screen location is simply a matter of counting along the appropriate line.

Note that the first screen position is ZERO.

GETTING RID OF IT.
****

It is all very well to write to the screen but the time comes when you no longer need that screen full of text etc. Referring back to PROGRAMME 1 again, the SUB programme CLEAR1 which starts on line ØØ46 fills the screen image table with the blank character >2Ø. Thereby removing the screen display. This routine starts from the last character position >2FF and writes a blank into each position in the screen image table back to position Ø.

You should be able to determine how it operates by following the comments added on each line.

BL.
***

The clear screen SUB programme was accessed using the BRANCH and LINK instruction BL. Read page 1Ø8 of the E/A manual. The most important point to remember with BL is that it places the address immediately following the BL instruction into R11. This creates no troubles until you try to BL from within a SUB programme which you have already use BL to get into. Before using the second BL the contents of R11 must be saved so that it can be reloaded into R11 to allow the programme flow to return to the correct place when leaving the original SUB programme.

KSCAN.
******

KSCAN is an extended utility provided that allows us to use the routine that scans the keyboard. KSCAN must be included in the REF line. See line ØØØ5 of programme 1 from last month. KSCAN is accessed by BLWP @KSCAN.

As you will remember from Basic and Extended basic the keyboard can be scanned in several ways. That is the entire keyboard or each half. To select the type of scan to be done on the keyboard the correct valve is placed in the byte at memory location >8374. The ASCII value of the key pressed is placed in memory location >8375. If no key is pressed then >8375 contains >FF.

KSCAN Memory Locations.
REMEMBER THESE ARE BYTES NOT WORDS

>8374 Type of keyboard scan.
>ØØ Full keyboard scan.
>Ø1 Left side of keyboard and Joystick 1.
>Ø2 Right side of keyboard and Joystick 2.

>8375 ASCII value of key pressed is placed in this byte.

>8376 Y value of Joystick is placed in this byte.

>8377 X value of Joystick is placed in this byte.

>837C This is the GPL status byte. Bit two of the STATUS byte is turned on when a key different to the key pressed last time KSCAN was called is pressed.

Referring back to programme 1 from last month the KSCAN utility is used on line ØØ35. Note that the Status Register has been cleared on line ØØ34 before the branch to KSCAN. Then on line ØØ44 the Status Register is cleared again before leaving the Sub programme.

In programme 1 the KSCAN is used in a PRESS ANY KEY TO CONT loop. To cause the programme to loop until a key is pressed the STATUS Register bit 2 is checked on line ØØ36 using CB instruction. If bit 2 isn't set then line ØØ37 JUMPS back to line ØØ35 to the KSCAN again.

STATUS REGISTER.
****
The register lives at address>837C. Refer to pages 4Ø4/4Ø5 in the E/A MANUAL.

The KSCAN utility sets bit 2 when a new key is pressed. line ØØ11 in programme 1 contains the value >2Ø.

This value is used for the comparison on line ØØ36 to test for a key press.

BIT NUMBER.
Ø 1 2 3 4 5 6 7

BINARY VALUE held at ANYKEY line ØØ11.

Ø Ø 1 Ø Ø Ø Ø Ø

HEX VALUE.

2 Ø

THE END.
****

This where I will have to close for this month. Sorry I didn't get to the number crunching. That will have to wait until next month.

INPUT UTILITY PROGRAM

PAUL MULVANEY

This is a utility program demonstrating how to attract attention to an input on screen by alternately flashing the prompt in reverse video format.

The concept is to change the lower case characters to upper case and make them white characters on a black background. By alternately displaying the normal letters then the redefined letters a very eye catching input message is displayed.

Lines 1ØØ - 18Ø show the keys to use for the reverse video format. Lines 2ØØ - 24Ø alter the colour sets of the lower case characters and redefine the lower case characters as upper case. The first two examples are three lines long, the rest is to make a program out of the code. The first two examples presented are fairly slow in responding due to the FOR-NEXT delays. If faster response time is required use the third example which checks the keys after each DISPLAY which halves the response time.

By using the lower case characters it is easier to type the messages in but if the lower case are used in the program then more extensive use of the READ and DATA statements will be needed to redefine other characters. You only need to define the characters actually used in the message.

```
1ØØ ! AN EXAMPLE OF HOW TO HIGHLIGHT
AN INPUT ON THE SCREEN
11Ø ! USING REVERSE VIDEO CHARACTERS
12Ø ! FOR THE FIRST DISPLAY USE
UPPER CASE CHARACTERS
13Ø ! FOR THE SECOND DISPLAY USE
lower case CHARACTERS
14Ø ! FOR A SPACE FCTN C (')
15Ø ! FOR OPEN PARENTHESIS FCTN F
({)
16Ø ! FOR CLOSE PARENTHESIS FCTN G
(})
17Ø ! FOR SLANT FCTN A (¦)
18Ø ! FOR QUESTION MARK FCTN W (~)
19Ø !
2ØØ CALL COLOR(9,16,2,1Ø,16,2,11,
16,2,12,16,2):: CALL SCREEN(12)::
CALL CLEAR
21Ø DISPLAY AT(12,4):" REDEFINING
LOWER CASE CHARACTERS AS UPPER CASE"
22Ø FOR A=1 TO 5 :: READ C,K :: CALL
CHARPAT(C,A$):: CALL CHAR(K,A$)::
NEXT A
23Ø DATA 32,96,4Ø,123,41,125,47,
124,63,126
24Ø N=96 :: FOR A=65 TO 9Ø :: CALL
CHARPAT(A,A$):: N=N+1 :: CALL
CHAR(N,A$):: NEXT A
25Ø CALL CLEAR
26Ø REM ## FIRST EXAMPLE ##
27Ø DISPLAY AT(24,6):" ANY CHANGES ?
Y/N " :: FOR D=1 TO 1ØØ :: NEXT D ::
CALL KEY(Ø,KY,S)
28Ø IF S=Ø THEN 29Ø ELSF IF KY=89
THEN 3ØØ ELSE IF KY=78 THEN 32Ø ELSE
IF KY<>89 AND KY<>78 THEN 27Ø
29Ø DISPLAY AT(24,6):
"'any'changes'~''y¦n'" :: FOR D=1 TO
1ØØ :: NEXT D :: GOTO 27Ø
3ØØ CALL CLEAR
31Ø DISPLAY AT(12,2):
"'press'enter'when'changed'" :: CALL
KEY(Ø,KY,S):: IF S=Ø THEN 31Ø :: IF
KY<>13 THEN 31Ø
32Ø CALL CLEAR
33Ø REM ## SECOND EXAMPLE ##
34Ø DISPLAY AT(22,2):" PRESS ANY KEY
TO CONTINUE " :: FOR D=1 TO 1ØØ ::
NEXT D : CALL KEY(Ø,KY,S):: IF S=1
THEN 36Ø
35Ø DISPLAY AT(22,2):
"'press'any'key'to'continue'" :: FOR
D=1 TO 1ØØ :: NEXT D : GOTO 34Ø
```

```
360 CALL CLEAR
370 REM ## THIRD EXAMPLE ##
380  DISPLAY  AT(24,5):"  GO AGAIN ?
(Y/N) " :: FOR D=1 TO 100 :: NEXT  D
:: CALL EY(0,KY,S)
390  IF  S=0  THEN 400 ELSE IF KY=89
THEN 270 ELSE IF KY=78 THEN 430 ELSE
IF KY<>89 AND KY<>78 THEN 380
400            DISPLAY          AT(24,5):
"'go'again'~''{y!n}'" :: FOR D=1  TO
100 :: NEXT D :: CALL EY(0,KY,S)
410  IF  S=0  THEN 380 ELSE IF KY=89
THEN 270 ELSE IF KY=78 THEN 430 ELSE
IF KY<>89 AND KY<>78 THEN 400
420 REM ## END MESSAGE ##
430 PRINT " ''the'end''goodbye''" ::
FOR D=1 TO 800 :: NEXT D :: END
```

BUMPER CHRISTMAS ISSUE

We are receiving some good feedback from Australia and Overseas regarding the quality of our newsletter. It seems that everyone likes what we produce, articles that originally appeared in our newsletters are popping up everywhere, making our authors "world famous" - right Paul? -so be in it! As Joe says in his "poolside chat", give it a go and make the effort to write. Anything will do - reviews, comments, letters to the editor (keep them clean!) - it doesn't have to be an in-depth treatise pages long.

We hope to produce a bumper magazine for the December/January issue. This can only be done with YOUR help. If you do intend submitting something please get it to the editor as early as possible (ring me And I will pick it up) and certainly no later than Tuesday, 2nd December. Let's all pull together and make this an issue that the whole TI community will be talking about.

ENTOMOLOGY CORNER 9

Just a short article this month. I didn't really intend to write one at all as it was Will's turn, but it is 11th year exam time and writing an article just isn't on for him. Which isn't to say that he hasn't been on the machine, but less than usual. One thing that did get him a little excited was a letter and K-Town Newsletter from Joe White in Knoxville, Tennessee. This carried excited reports of of a disk copier that did the actual copy process in 28 seconds for a SSSD disk. That was exclusive of formatting and verifying of course which take longer and checking is essential for reliability. Nevertheless there is a speed record in there waiting to be shattered. I reminded Will that a disk spins at 300 rpm and he took it from there. A few hours later he could show a SS copy in 16 seconds, and the next day a DS copy in 32 seconds. He hasn't tried this on the original TI/Shugart drives. It works on the Chinons like a charm. These by the way may not be quite fast enough to keep up with a 6 ms track step time setting (possible with Myarc and Corcomp controllers) under all circumstancs. We have reverted to 20 msec after finding assembler problems on heavily fractured source files. He has done this with TI and Myarc controllers and it would be a trivial matter to add Corcomp to the list.

While I was able to get near the machine, I got down to work on various ideas and lo and behold FUNNELWEB Vn 3.4 is now with us. Note the renaming. The previous title Funlwriter has for some time seemed just too restricted in its implications. All of the improvements that accreted to Vn 3.3 have been retained and changes made to the user interface. Now the central menu screen has two parts, toggled by pressing just about any key. This replaces the 'switch" cycle which had outgrown its welcome. Functions of interest to programmers are concentrated on the second screen. There are various other detail improvements and a great deal of tidying up of internal code which had grown haphazardly as items were added to the switch list. It all adds up to an improvement so noticeable that we are reluctant to use Vn 3.3 any more.

Bob Boone told me from Ottawa that Vn 3.5 of DM-1000 was being released for the Chicago TI-Fair last weekend, and that they have incorporated the code for return to FWR as a permanent feature. We should have that soon to incorporate in the package. The good news is that the 1-sector file problem has been resolved. The 16/18 sector problem remains with Myarc controllers. We always use the Myarc DM to do 18 sector per track formatting in double density and this then leads to trouble with DM-1000 which detects the card and sets for 16 sector. A solution to this problem is at hand in that Will bashed up a routine that will format Myarc at 18 sectors as reliably as Myarc does it. When Vn 3.5 arrives I will look at incorporating this so that all DD disks are handled at 18 sectors.

The cooperation between Ontario and the Hunter Valley is helping to refine a very comprehensive and easy to use system. I certainly have nc desire to get into the complexities of emulating the file system and the DM module and more that Bruce Caron took on when he created DM-1000. It works the other way around too. There is only so much that can be done in the time available by any one person. I had a letter from the USA from someone who used IBMs at work and the 99/4a at home, and because of FWR much preferred the TI for ease of use.

We have in the last few days taken delivery of a Myarc 512K RAMdisk card for the PE-box. At this stage I can give only initial impressions as we haven't had time to dig into it yet. The main one is that it works like a charm. It helps to have a Myarc disk controller as the DM for that is aware of the details of the ramdisk even when it isn't emulating a regular disk. Physical construction is on a par with the Myarc disk controller - not quite up to the overkill in mechanical standards of the TI cards but perfectly adequate. The heart of the card is 16 256K dynamic RAMs and their TMS4500A controller. The TI 32k expansion is replaced by 32k of the ramdisk and the remaining 480k is bank-switched in 32k at a time under control of the DSR ROM. Up to 400K of this is available as ramdisk, the remainder being available for a print spooler. Why 400k ? The reason would appear to be that with the standard TI disk format which the ramdisk follows, the bitmap on sector 0 allows for this much worth of 256 byte sectors and compatibility at the sector addressing level is retained for normal software such as DPATCH or various cataloging programs. The card resides at CRU address >1000 so that it sees standard DSRLNK searches before the normal disk DSR. This means that when set up by the CALLs from Basic/XB or the Myarc DM to emulate an existing drive number, the ramdisk is found first. The only unfortunate choice that Myarc made was to have it call itself RD. in its native state rather than DSKR. which makes it less compatible with existing software than it could have been. The print spooler appears to work by having an interrupt routine flagged in the DSR ROM header which would be found by the normal VDP interrupt routine. This then goes to code which addresses the RS232 card and presumably spits out some of what has been stored in the spooler memory. We have not fully explored how this system works with programs. I would expect that if a program suspended VDP interrupts by LIMI 0 that spooling would cease. The TI-Writer Editor and FUNNELWEB enable interrupts during their keyscan routines, that is to say while the computer is waiting for user input. The Editor turns off the system reset by the appropriate bit in the system flag in PAD, and so allows other VDP interrupt functions such as screen blanking after a period of no activity to continue normally, and now the print spooling as well. We shall have to find out from someone else how it is affected by a hung PIO. The Myarc documentation is less than informative about the details. In fact the booklet follows the disastrous TI tradition of providing essentially no technical information beyond Basic programming advice, and much of it is a reprint of the TI disk manual, there being only one short section of hints for the Assembly programmer. The card appears to have a connector for an external power connection to the card but this is not documented, even by a inserted leaflet. Maybe this is just that they haven't got around to it, but the possibility remains that it doesn't work properly or even is a danger to other cards in the box. I just wish they would tell the story and give the necessary specification for the external supply if it is in fact usable. At least one well known program, DSKU does not react happily to the presence of the Myarc ramdisk.

What I have done is set up a special disk of FUNNELWEB and utilities configured to run from DSK5. At the start of a session the Myarc DM is loaded, either from Basic using the disk card CALL ILR and LR (they really are a bit more useful than I gave them credit for in an earlier article, but from command mode only) or else from E/A which is our normal module. This is then used to set the ramdisk to DSK5 and then to do a file copy of the disk into the ramdisk. Then LDFW is used to load FWB from the Myarc DM and we go from there. FWB and the RAMdisk are an ideal combination. It really is a flying finger exercise to switch from Editor to Assembler and back again. Assembling a program is a disk intensive activity and gives a handy speed comparison. The main FWB program used to take about 8 minutes on the TI controller in uncompressed object format so that it could be loaded with XB's CALL LOAD. Using the Myarc disk controller card with DD disk format and compressed object code (loadable with CALL LR) reduced this to under 4 minutes, and using the ramdisk has now brought this under 2 minutes. Apart from reduction of the tedium of the edit and assemble cycle the best thing is that you don't have to listen to the disk drives grinding away.

Currently the competition for Myarc is from Horizon and Corcomp. The Horizon looks to be the least intrusive and most flexible, being reprogrammable and using only the DSR portion of the memory map. It also has battery backup built in for its low power static rams. The only disadvantage apparent at the moment is that it is limited to DSSD equivalent size until the price of higher capacity static RAMs comes down to more reasonable levels. Also David Romer writes letters back, in contrast to my experience with Myarc so far, and has a very good engineering sense of what and how things should be done with the TI 99/4a system as it was bequeathed to us here in the orphanage.

Early reviews of the Corcomp 512K were not very encouraging, as there were a lot of firmware bugs apparent, and incompatibility with standard directory routines. It does have its own built in directory routine as does Myarc (CALL RDDIR) also, but what is really needed is the disk manager built in to one or other of the ramdisk or disk controller. Corcomp did have the good sense to give it the default device name DSKR. which is much easier to live with than Myarc's choice. I have heard from a couple of Corcomp owners since then and they seem rather more pleased with the device. From Australia I would wait until they had got their act together a bit more. I suspect each device has its own strengths and weaknesses, and more experience is needed to be able to report on their relative merits. Hopefully we will have some report on the Horizon card by next issue.

TONY MCGOVERN
FUNNELWEB FARM

SPRITE DIRECTIONS

	D-R	D-C
UP	-	Ø
DOWN	+	Ø
LEFT	Ø	-
RIGHT	Ø	+
UP LEFT	-	-
UP RIGHT	-	+
DOWN LEFT	+	-
DOWN RIGHT	+	+

VELOCITY LIMITS

Max: -127 to 128
Stop: Ø