

TI 99/4A

HOME COMPUTER NEWSLETTER



YOUR COMMITTEE

ACTING PRESIDENT Jim Grimmond 31 Jarrett St., TORONTO 2283

Ph. 595751

SECRETARY
Albert Anderson
6 Arcot Close,
TARRO 2322
Ph. 662602
Viatel 496626020

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
7 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., KCTARA 2285 Ph. 523162

Peter Coxon 25 Reserve Rd., WANGI WANGI 2267 Ph. 751930

CONTRIBUTIONS

Members and non members are invited contribute articles publication in HV99 NEWS. hny copy intended for publication may be typed, hand written, or submited on tape/disc media 35 files suitable for use with Τi Writer (ie. DIS/FIX 88 or DIS/VAR 88). A suitable Public Domain word processor program will be supplied if required by the club librarian Bl Lawrence. Please include along with 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 (uniess notified otherwise) are considered to be PUBLIC DOMNIN. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author recognised.

Articles for publication can submitted to.

THE COLTOR

٧a

fo

ev

to

₫e

Ho

OH

th

Ot

ch

e x

p1

10

th

Ju

fe

fo

me

an

50

ea

ea

to

G≢

ne

fo

fo

th

th.

WA

wh

in

THE COLTOR
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

INSTRUMENTS.

DISCLAIMER

MELIS is official the newsletter of the HUNTER VALLEY MINETY WINE USER GROUP. Whilst every effort is made to ensure the correctness and accuracy of the information contained therin, be it ٥f general, technical. OF 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 AUVERIT computer users, not affiliated in any way with TEXAS

SECRETARYS REPORT



FROM ALBERT ANDERSON

Variety in this issue should provide have it. food for thought for delay in the April issue postage. Hopefully this won't happen again.

realised in Lucie Dorais from the Neil from us all. Ottawa group in Canada. Lucie has to chosen 'adopt' HV99 as her exchange user group and we pleased that you have.

In this issue you will find a loose leaf membership renewal for MX-80 Printer is available to allow the '87/'88 year which commences in this machine to perform in GRAPHICS July. Believe it or not, membership mode. A feature that this machine fees remain the same as last year could not access previously. for both our Australian and overseas has installed this in his machine members so please complete the form and is more than satisfied with the and send in your membership fees as results. For more info on this soon as possible. Renewing sub's contact an EPSON dealer near you. early makes it just that little bit Thanks Frank for this. easier for us.

Whilst I'm at it I would also like to remind you that the Annual General Meeting is to be held at the next meeting in June. Nominations for office bearers have been called for so please give this some thought. I might point out here that the first nomination received the name or whereabouts of was for our retired President (Joe) who is still keen to help with the interests of the group.

10

In last months issue an invitation for members to come along to the committee meetings was offerred and I would like to thank both Alan Franks and Ron Kleinschafer for taking up this offer. (yes that's right Ron from the black hole at Grawin - western NSW). This invite is open to all members so feel free to attend if you so wish.

For the local people non-computer night out has been arranged for later this month. notice elsewhere gives you the info and invites you to come along - Be in it! as the saying goes.

On the technical side the long awaited club disk drives (2 of them mounted as one unit) have come to fruition and these are available for loan at no charge to members with the usual proviso's. It's expensive Welcome once again to you all. equipment - look after it while you The drives are double almost sided and the unit is suited to everyone. Could I firstly apologise connect to the groups Corcomp or the to our mail-out followers for the TI. Expansion box. Another great project which has got past the grey matter and into the module port is Neil Quigg's 32k version of the To go with a special welcome to 'Super Cart'. An article from Neil our new members is the good news explains it all to you in this that our 100th member has been issue. Congratulations on this one

> More technical news comes from are Frank Phillips, one of our very conscientious members in Germany who writes to let us know that a conversion kit for the EPSON

> > If you have anything of interest or some news that can be passed on please let editor Brian or myself or someone know of it so we can have it included in this NEWSletter for all of our benefit.

> > This reminds me! Does anyone know person that cleaned out the BIG W store at Green Hills stock of TI cartridges? About 50 or so various

modules for a whole 30c each! Not only should he/she be in jail, he/she should be a member of HV99 to learn how to use them!

Well, that's about it for this session - get into the good stuff that follows! Bye for now....

Albert Anderson 4a4me

MID YEAR NIGHT OUT

A non-computer night out has been arranged for FRIDAY 22 MAY at the Imperial Palace Chinese Restaurant, Newton St, Beresfield (over Flemings Supermarket), commencing at 7.00pm. You and your spouse/friend/lover etc(?) are cordially invited to come along and join in the fun. If you intend coming, please contact Bob MacClure on 437431 as soon as possible, and by Tuesday 19 May at the latest so that the booking can be confirmed.

PE-BOX SOCKETS

Having trouble trying to get hold of the 60 way and 44 way edge connectors for the 3 Slot PE-Box at a reasonable price?

Try :- HY-COM UNITRONICS
7 President Lane
Carringbah (Sydney)
2229

Phone > 02-5247878

The 40 way edge connector - \$8.40 44 way edge connector (gold plated) - \$11.50 Postage - \$3.50

ANNUAL GENERAL MEETING

The Annual General Meeting of the Hunter Valley 97ers Users Group will be held on Tuesday, 9 June, 1987 at the Warners Bay High School.

ALL members are urged to attend and vote for a new Committee. Remember, the future direction of the Group is in your hands.

Members are also reminded that Annual Fees are due by 30 June. Please forward your renewal form along with the fee to the Secretary, or pay them in person to the Treasurer.

T

P۱

S

k

Joe Wright would like to read a book called EXPLORING FORTH if one of our readers happens to have a copy. If you can help Joe out, please get in touch with him. His address is: 77 Andrew Road, Valentine 2280 NSW Phone (049) 448120

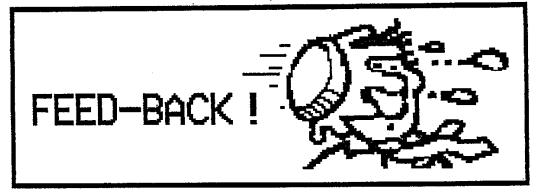
FOR SALE

- * Speech Synthesizer
- * Power Supply.
- * 4A keyboard.
- * UHF Modulator
- * 5 only 6264LP15 Memory Chips
- * LOGO 2 TE 2 ADVENTURE Mods

POA.

CONTACT: -

Phone > Ø89-270730 (Tas)



TO THE EDITOR.

FROM CHAOS MANOR. Confusion and of Purveyors Disruption, 24hr. service.

due consideration to STEVE TAYLOR'S suggestion that because non local members are at a disadvantage to having access the (see publications library T Secretary's report March 87). feel that the idea has considerable merit. A proposal that I think may work is that members wanting to read the newsletters etc, that the Group receives, request to be added to a list of "CHAIN MAIL" readers. work by having the first could listed "disadvantaged" member to receive, with their newsletter, any available material, then that member would be responsible for posting it Dear Sir/Madam, to the next member, with of course a time limitation that each member keeps the material, say, perhaps 1 week after receipt. This should be make further time to photocopies or notes of any material of interest to may be keep. (copyright permitting).

lok

et

The last member on the list would the return the material ta i f Publications Librarian and another member makes a request to be added to the member could be advised accordingly. I for one would be only too willing to pay for the postage to send any items on to someone else.

donate towards having newsletters etc specifically for circulation those "disadvantaged" members, such copyright laws permit arrangement.

that by the time I fully realise some members get to read articles etc it could be quite some months later but I feel that it is better to read them than NOT AT ALL, I also fully realise that this adds extra burden to an already overworked Committee, perhaps another local member could be "Volunteered" to organise photocopying and arrange to have the material put in with the newsletter to the first member on the list.

Perhaps others who would like to have this service could write and let their thoughts be known, I also request that the Committee consider this proposal as to whether it is a "Viable" proposition!! What do other members of the group think ???

Could you please add my name to the list of chain mail receivee's of any publications that the group receives.

> Yours Hopefully, R.Kleinschafer.

EDITORS COMMENT

The suggestion by Steve Taylor was welcomed by the Committee, and already some material has posted out to our non-local members, "list" then the last to be read and passed on to another member. Very shortly more material will be mailed out to other members. We aim to keep the number of 'links' in each chain to about 3 or 4 so that everyone can see tha mags I also would be quite willing to before they are out of date. these is no need to write to us requesting photocopied this service - out of towners are tolall included automatically, if unless you don't want to be included an in a chain, just sit back... HV99ers are working for YOU.

RANDOM **BYTES**

FROM BOB CARMANY

" 90 X It's time for another it. better do something with it!

We will start this column with 95 CALL CLEAR the BASIC loan calculation program 100 PRINT TAB(5); "CHECKBOOK that I promised you last time. Then BALANCE" we will get into a little bit of 102 FOR DELAY=1 TO 300 Forth and a few tricks and surprises 104 NEXT DELAY 110 CALL CLEAR as well. Here is calculation program:

100 CALL SCREEN(11) 110 CALL CLEAR 12Ø INPUT "ENTER AMOUNT OF LOAN ":L 13Ø PRINT 14Ø INPUT "ENTER NUMBER OF MONTHS " :T 15Ø PRINT 16Ø INPUT "ENTER INTEREST RATE ":R 17Ø FOR S=1 TO 6 18Ø PRINT 19Ø NEXT S 200 PAY=((R/12ØØ)*((1+R/12ØØ)^T*L)/ (((1+R/12ØØ^T)-1) 21Ø PAY=INT(PAY*1ØØ+.5)/1ØØ 22Ø PRINT 23Ø PRINT "MONTHLY PAYMENT= "; PAY 24Ø PRINT "TOTAL REPAY= ":(PAY*T) 25Ø PRINT "TOTAL INTEREST-"; (PAY*T)-L) 26Ø PRINT 27Ø FOR S=1 TO 6 28Ø PRINT 29Ø NEXT S 300 PRINT "ANOTHER LOAN (Y/N) * 3Ø5 CALL KEY(Ø, KEY, STATUS) 31Ø IF STATUS=Ø THEN 3Ø5 315 IF (KEY<>78) * (KEY<>89) THEN 3Ø5 32Ø IF KEY=89 THEN 11Ø 33Ø CALL CLEAR 34Ø PRINT "SEE YA NEXT TIME!" 350 FOR D=1 TO 1000 360 NEXT D

37Ø CALL CLEAR

38Ø END

Well, there it is! The program is so "user friendly" that it is not really necessary to give any sort of instructions with it. You simply enter the interest rate, amount of the loan, and number of months, and the program will return the monthly payment, total interest repaid, and the total amount paid.

Since are dealing with we I have FUNNELWEB all "cranked | "higher finance" this month, here is up" and ready to go so I guess I had a BASIC checking account balancing program for you.

> the load 120 INPUT "BANK BALANCE? ": BALANCE 13Ø DISPLAY "ENTER EACH OUTSTANDING" 148 DISPLAY "CHECK NUMBER AND AMOUNT. " 15Ø DISPLAY 160 DISPLAY "ENTER A ZERO FOR THE" 170 DISPLAY "CHECK NUMBER WHEN FINISHED. " 180 DISPLAY 19Ø N=N+1 200 INPUT *CHECK NUMBER ?": CNUM (N) 21Ø IF CNUM(N)=Ø THEN 25Ø 22Ø INPUT "CHECK AMOUNT? ":CAMT(N) 24Ø GOTO 19Ø 24Ø GOTO 19Ø 250 DISPLAY "ENTER EACH OUTSTANDING" 260 DISPLAY "DEPOSIT AMOUNT." 27Ø DISPLAY 28Ø DISPLAY "ENTER A ZERO AMOUNT" 290 DISPLAY "WHEN FINISHED." 300 DISPLAY 31Ø M=M+1 320 INPUT "DEPOSIT AMOUNT ?": DAMT (M) 33Ø IF DAMT(M)=Ø THEN 36Ø 34Ø DTOTAL=DTOTAL+DAMT(M) 35Ø GOTO 31Ø 36Ø NBAL=BALANCE-CTOTAL+DTOTAL 370 DISPLAY "NEW BALANCE= "; NBAL 380 INPUT "CHECKBOOK BALANCE? *: CBAL 39Ø DISPLAY "CORRECTION= "; NBAL-CBAL 400 END

XBA: her be

Exp

Mem

d i s mem

dis

an

retra Alt

Wyc eit inc

wor ďΦ the

spe an get for HCN sut

501 * S (one th abi to sui HCI

Th i 5 **"** = se ju

co

ogram not t of imply ht of and ithly and

with e is hcina

Now. for those of you with XBASIC and 32K memory expansion. here are some CALL LOADs that might be of interest:

CALL LOAD(-31868,Ø,Ø) Turn Mem Exp Off

CALL LOAD(-31868,255,231) Turn Mem Exp On

CALL LOAD (-31888, 63, 255) Turn disk drives OFF. Use NEW to free memory.

CALL LOAD (-31888,55,215) Turn disk drives ON.

CALL LOAD(-31931,0) Unprotect an XB program.

CALL LOAD(-31931,128) Protect an XB program.

Now. for some Forth re-definitions that 50 Wycove. translate TI Forth into do not appear in Wycove Forth. they are:

- : BASE->R BASE >R ;
- : R->BASE BASE R> ;

Now, for a little "soapbox" speech. A few months ago, there was method for Epson printers: an item in the HV99 newsletter about | .TL 126:27,67,70 FNTER> getting some sort of compensation When the ~ for HCM Computer (Home subscription. some people had already gotten a (the tilde) at the beginning of "settlement" of sorts. Well, I was file. one of the "few" (Ha!) unfortunates that found their subscription abruptly terminated. Well, I wrote the appropriate address and, surprisingly, got a response from HCM (now Home Computing Journal). The settlement that they offer is an issue or two of their quarterly "compost rag". From what I have seen, it is better than nothing but just barely! Well, 'til the next column. . .

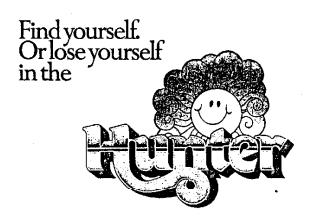
TI-MRITER TIP

This Question and Answer comes from the May, 1987 issue of the TISHUG News Digest, newsietter of Sydney TI Users Group.

I have been experiencing problems getting my TI-Writer to function properly with my printer when using I have used the A4 size paper. .PL 70 on the document command without success. Is there something I am missing? (from Percy H).

Yes Percy, you do need one more thing. The command .PL 70 simply tells the FORMATTER the number of lines to each page of your paper. you can So far so good, but you also must tell the printer the number of lines Although they are not necessary in per page if you don't want it to Wycove (you can enter numbers in | print across the page perforations. either number base), I am going to This can be done in one of two ways. include the definitions of two more The first is with Special Character words that appear in TI Forth that | Mode of TI-Writer. You will find an Here explanation on page 98 οf TI-Writer Manual with the characters listed and defined on page 146. Function U is used to enter and leave the Special Character Mode. second method is with the The Transliteration command. Here is an ٥f the Transliteration example

(tilde) character is your unfilled portion of your found at the beginning of the file Magazine) it will set the printer to 70 lines It mentioned that per page. Don't forget to include ~



HANDLING CMOSIC'S

BY

RON KLEINSCHAFER CHADS MANDR

Lately a lot of TI users seem to be engaging in hardware modifications to their computers, also there has a lot of printed material appearing from time to time in various publications about the dangers and pitfalls of handling CMOS IC's and LSI's (and other chip technologies). This seems to put discharge would be dispersed to some fear into intending hardware yourself and not "jump" to the pins handling hackers about components, and much has been publicized about having the need for wrist straps, antistatic mats etc. The following should allay some of those fears and give a little confidence to the overly wary.

HANDLING LSI CHIPS.

- REMOVING CHIP FROM THE Α 1... SHIPPING CONTAINER.
- Before touching the a. by touching and holding antistatic material containing the chip pins.
- While holding the antistatic material, lift out the chip and hold lightly by the PINS.
- c. You can now safely carry chip, by holding the PINS.
- REPLACING THE CHIP IN ITS CONTAINER.
- a. While holding the PINS touch and hold the antistatic material replace.
- ANOTHER GIVING THE CHIP TO PERSON.
- While holding the PINS of the a. chip in one hand, make contact with the other person until he is holding the chip by its PINS.
- 4. REMOVE AND REPLACE IN BOARD.
- a. Before removing or replacing the chip, touch signal ground plane the component board.

While holding signal ground, remove or replace chip.

c. In general, something other than the chip (e.g., Your hand) should make first contact with the circuit.

CAUTION.

WHEN OPERATING ANY EQUIPMENT WITH COVERS REMOVED, KEEP PAPER AWAY FROM LOGIC TO PREVENT STATIC DISCHARGE FROM DAMAGING CHIPS.

The above is reprinted from a technical manual for technicians by the CENTRONICS DATA COMPUTER CORP.

Different to any thing you have been told or read EH! I suppose there is some logic behind it in the fact that there is less likelihood of static discharge to the pins if you are holding most of them, these ?? Although I personally do not do it this way, in many years handling such devices I have not had one one failure through static damage, I simply REFERENCE any tools, soldering irons, or hands to ground plane, then carry on. They are much hardier than is qiven credit for. In other words heaps of fancy equipment is not required just plain old COMMON SENSE!! Simply don't go sliding down the nylon chip carpet in your best track suit then reference yourself to the container showing the kids how well you can the juggle your 6264LP-15's.



COI an Hu is e x 32

*:

OD th us aπ pr o f

Th

of

to

C e a

b

t

ground,

er than Should ircuit.

WITH Y FROM CHARGE

rom a ns by ORP.

e been re is # fact od of if you any to Dins hot do o.f bt had Static any to They given

ps of

just

imply

nylon

then

can







A WORLD FIRST !!

AN ALL NEW SUPER IMPROVED 32K E-A MODULE

** MODULE RELEASE PRELIMINARY **

improved Editor Assembler command module has been designed and developed by a member of the Hunter Valley 99'ers. The module is completely compatible with the module existing E/A and adds 32kbytes of battery backed memory. The memory is arranged in 4 pages of 8kbytes and is totally available to the user. Use of the memory is only limited by the imagination of the programmer and should be very useful for the storage of and/or subprogrammes to be used by programmes that are run in the rest of the computers memory space.

The prototype module has been constructed and is presently undergoing evaluation trials by members of the H.V. 99'ers. It is envisaged that the trials will take approximately one month and the final version of the module will become available shortly after that time.

For further information on this device please enquire in writing to:

Hunter Valley 99'ers Module development, P.O. Box 84, HAMILTON N.S.W. 2303.

RS232 CARD REPAIR TIP

Patient :- TI RS232 Card. (PE.Box

type)

Symptoms: - RS232 operations function as normal BUT PIO will not output correct characters to printer when called on to do so. Example: -

PIM TEQT, 1014545890=QUEPTYUIMP-AQD instead of -

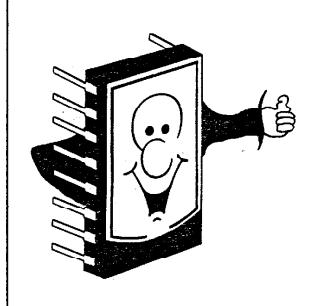
PIO TEST. 123456789Ø=QWERTYUIOP/ASD

This particular case would not return a carriage return and therefore would not line feed the printer.

Cure: Removal of the suspect 74LS245 Bi-directional buffer designated U3 on the RS232 Card and replacement (optionally in a socket) with a brand new specimen. Cost approx \$2.50 and the usual disclaimers on risking the health of your equipment apply. ie. Whatever you do is at your own risk.

Results: - Success! My PIO port now communicates with my trusty printer in a dialect that I can now interpret.

Albert Anderson - HV99



ASSEMBLY LANGUAGE

FOR THE LAYMAN

WITH ALLAN WRIGHT, HV99ERS

Oh well! One thing I can claim to pointer, know is how our little T.I. ticks, continues and that's far and away easier than workspace. understanding Human kind.

from last month a few thoughts and don't RTWP OUT. little bits to add about sub routines.

RT **

remember that RT is instruction. SOURCE CODE the ASSEMBLED CODE has! If you have please SHARE your returned will be the same as if you knowledge, that is what USER GROUPS had put

B *R11

that when

BL

Is used to BRANCH to a routine that in the sub routine workspace is RT MUST be used to return to the still intact. It will remain this calling TRUE. programme...NOT Instead of RT

B @ANYWHERE

could be used to take your programme! some place other than the instruction after the calling BL.

An example of this is to return programme flow to a Menu if some criteria has not been met. eg disk error, buffer full etc.

RTWP.

When a subroutine has been entered by BLWP, a new WORKSPACE is used by The the subroutine. RTWP instruction is used to return to the

May 1987! Well Well. I thought May instruction after the calling BLWP. be, this year would not passby as This reloads the main programme quickly at last. Ha! Wrong again, workspace address into the workspace and programme using the original However there may at some time be a circumstance when you continue using want to Before getting down to the exercises subroutine workspace. In this case

90

to th a

wh do re

> IN XX Αs

SU

Q.

iŧ

1

a

T

B CANYWHERE

Will take your programme wherever you so desire. At this point I As mentioned last month you will haven't found a use for this and a psuedo quite frankly I can't conceive one When included in your either BUT I will bet that somebody are ABOUT!.

The big advantage of using BLWP the use of the new WORKSPACE when In your code, after rereading the the subroutine is called. This next article from last month you could point I want to stress is very have been left with the impression obvious but sometime missed, a bit obvious but sometime missed, a bit like the trees in the woods thing.

> When programme flow returns from the subroutine after RTWP, all the DATA way until another BLWP or you change it in your programme.

> The subroutine workspace is different to any contiguous block of memory set aside. So any Data left in this area can be accessed from the Main programme.

> When setting aside this workspace area a LABEL is normally used. either with it EQUated to an address or BSS used to set aside a block (>20) of memory.

> The DATA left in this block of memory can be readily accessed by INDEX ADDRESSing using programme flow returns to the main programme. Again this is not the ONLY way to do this, BUT it is a

introduce the appd way to a further BLWP to a sub routine into a block of memory also, eg. which uses the same WORKSPACE is done before the DATA has been used. retrieved or what ever.

INDEX ADRESSING

Assume for this diskussion that WORKSPACE has been set aside for a sub routine as follows;

> SBRWS BSS >2Ø SUB RTNE WS

Also assume that during the subof the programme. The indicated by the LABEL SBRWS is no longer addressable as REGISTERS. (The WORKSPACE POINTER is set to another address).

The first byte of the memory word that the data is stored in (R3) be 6 bytes after SBRWS. Remember that the count starts from ZERO at SBRWS. So the following table can be shown.

SUB RØUTINE REG No. ADDRESS.

RØ SBRWS + Ø R1 SBRWS + 2 **R2** SBRWS + 4 RЗ SBRWS + 6

This can also be written.

R4 8 + SBRWS R5 1Ø+ SBRWS

at the above Looking table the concept of INDEXED ADDRESSING can be seen.

To MOV the data which had been left in R3 during the sub routine this code would be used. Firstly the offset from SBRWS should be loaded into a REGISTER (R7 in this case). then becomes the INDEX REGISTER. The following instruction will then MOV the data into R5.

MOV @SBRWS(R7),R5

don't think a great deal of thought is required to imagine this INDEXED ADDRESSING being used to pull a series of data from a block of memory.

each MOV is executed the value in R7 option is used.

next is incremented by the amount rquired topic. Always remembering of course to access the next piece of data. that the Data could be disrupted if | The same applies to loading data list of names.

> To round off this diskussion here are two MOV instructions as shown on page 72 of Ira McCOMIC's book A/L Programming. Both intruction a formatts accomplish the same task but required different programming techniques to apply them.

> > @Ø(R6),R5 VOM MOV *R6, R5

routine some data was placed into R3 In both cases a word is moved into which we now want in the main body R5, from the memory address which area has been put into R6. In the first case R6 is an INDEX register. the second Register indirect addressing is used. You will recall that register indirect addressing is used to MOV data into a sub routine eg.

> MOV *R11+,RØ (FOR BL)

*R14+, RØ (FOR BLWP) MOV

In the example;

MOV @Ø(R6),R5

Instead of using ZERO reference the value 6 could be used. this would then MOV the word at 6 + SBRWS

This is the address of the third the Registers in sub routine workspace.

WARNING.

When using INDEXED ADDRESSING, RØ cannot be used as the INDEX REGISTER.

Having dealt with sub routines fairly extensively over the last few articles the next logical step seems to be to show ways of adding these routines to your code.

The E/A package allows two methods for us to do this. The first and probably the easiest to understand and start with is with INSERT mode of the EDITOR.

INSERT. *****

To add some code to a file which is To do this R7 is set to ZERO. After already in the EDITOR the insert The operation of

LWP. ramme space flow ginal at n you the case

ever ht I and one body your ROUPS

i s when next very bit 9.

the DATA is this ange

no k of Deft from

bace sed, ress lock

of 1 Бу when main the s a

this is quite straight forward, 70.7 suggestion is to simply add the additional file to the bottom of your file for the first couple of Before attempting to assemble that goes. Once you get the hang of it code you must also have typed in the you will be placing code all over The result of this the place. procedure is that your source code This can become very lengthly. causes it to take a long time to load and also to page back and forth when modifying or through it correcting code.

This method does eat away space, when you INSERT a file into ************ your source code your are saving it This routine is a basic number check it again is just wasting month. valuable disk space. directive is a move efficient way to blanks from the input stream. add code during assembly.

COPY. ****

The COPY DIRECTIVE allows additional code to be added to a programme without the need to have the code as part of the SOURCE CODE. In the a very lengthy programme where buffer full is encountered code can be added to the assembled code directly from disk by filename, thus over comming the buffer full problem. It can also be used to add programme sub routines and and segments to your code very easily.

I ALWAYS have the Assembler produce a LIST to disk when I am assembling the COPY The use of any code. DEBUGing make DIRECTIVE can don't get difficult i f you listing.

Throughout these last few articles I have been stressing the need to write universal sub routines. This is essential to be able to take full advantage of COPY.

The cursor routine which was in the Newsletter has been December The following programme improved. alterations to includes seament code the CURSOR programme and required to include the number check sub routine and the clear screen sub routine in the object code. When assembled.

The Cursor routine is shown only to the LABEL EVALKI. The code from that point on is not changed from

December in the that shown Newsletter.

number check routine and the clear routine which are also listed. Each of these should be saved separately under the file names, NUMCHECK and CLEAR. If you have only one disk drive you will also need to alter the disk drive number in the COPY directive.

disk NUMBER CHECK.

a second time. That is, it already programme which follows the flow exists as a separate file and to diagram shown in the article last The routine includes а The COPY segment of code which strips leading section of code starts at the LABEL NUMCK2 and ends at LABEL NUMCK3.

> The section of code between NUMCK4 NUMCK7 is the NUMBER check and the flow routine which follows diagram.

NUMCK7 is the LABEL which starts the set not a number flag. While NUMCK8 through NUMCK9 transfers the number from the screen into BUFF1. first byte of BUFF1 has the count of characters in the number loaded into

CLEAR ROUTINE. **********

assignment from the second The article last month was to write a universal screen clear routine. The following code shows such a routine. It is in the programme above and is loaded using the COPY directive.

ASSIGNMENTS.

If you are following these articles the next step is to write a routine which will do the following.

It is to be universal and must be able to be added to the programme using the COPY directive.

- 1) Write to the screen at an address which can be loaded as data into the routine either of the following.
- a) PRESS SPACE BAR TO CONTINUE
- 6) DO YOU WANT TO TRY AGAIN Y/N
- Not leave the routine untill the correct key has been pressed.
- 3) Load into the routine using data the two test characters Y and N.

at some

This prograi exiting the E progra luxury

FINISH **** The to charac live i to cre

The be at th chapte

Readir E/A ADDRES

E/A DIRECT

I.McC

Thats

injini * 25 0002 × 7H (3¢07 **× 0**F 0004 × 31

0005 × Ta 990a * #**!** 0007 8008

> (1009 STAT 0010 DEVI 0011 KEY 0012 KES 0013 HES

OQLA ENT 0015 LEF oots RIG 9917 INS

0019 DEL BOTH ERA 0020 **584**

0021 ZEF 6622° Mir 0027 PEF

(#24 Fl.) (a)25 FL/ 0026 EE

.0027 *∃*¥1 0028 88

```
ember
           After all you may need to use 1 or 2 0029 RETURE BSS >2
                                                                                                   SAVE BRANCH RETURN ADDR
                                                                 0030 BUFF1 855 >20
           at some later date.
                                                                 0031
                                                                            EVEN
that
           This
                     routine
                                  is
                                                           the
                                                                 0032 * COPY TWO FILES
n the
                                        needed
                                                    for
                                                                 0033
                                                                            COPY "DSK2.NUMCHECK"
clear
           programme above to give a
                                                  means of
                                                                            COPY "OSK2.CLEAR"
           exiting the programme and return to
                                                                 0034
 Each
                                                                           MOV R:1, ORETUR!
                                                                                                   SAVE RETURN ADDR
ately
                                         Currently the
                                                                 0035 START
           the E/A
                       module.
                                                                            LWPI MYWSP
                                         allow you that 0036
                                                                                                  LOAD NEW HS ADDR
K and
                         will not
           programe
                                                                                                   CLEAR SCREEN
                                                                 0037 START2 9L @CLEAR1
disk
           luxury.
                                                                 0038
                                                                            DATA >2FF.0
alter
                                                                 0039
                                                                                                   BRANCH TO CUREOR
 COPY
           FINISH.
                                                                            BL GGETKE1
           *****
                                                                 0040
                                                                            DATA >4,>104
                                                                 0041
                                                                                                  - BRANCH TO NUMBER CHECK
           The topic for
                                 next
                                         month
                                                   will
                                                                            BL DNUMCKI
                                                                  0042
                                                                            DATA >104,>4
           character
                         definitions.
                                                 Where they
                                                                 0043
                                                                            CI R7,) FPFF
                                                                                                   IS FLAG SET?
           live in memory and redefining them
                                                                 0044
                                                                            JNE START2
           to create our own character set.
check
                                                                  0045
                                                                                R0,010A
                                                                                                   SCRN ADDR
                                                                            LI
 flow
                                                                 0046
                                                                            LΙ
                                                                                R1,#ESS1
                                                                                                   TEXT ADDRESS
           The best reading to do on this topic
 last
                                                                  0047
                                                                                                   LENGTH OF DATA
                                                MOLESWORTH
                                                                            LI
                                                                                82, >0
           at this stage is in
                                                                  0048
                                                                            BLMP OVMBW
                                                                                                   WRITE IT
ading
           chapter 5.
                                                                  0049
                                                                            LI RO.>205
                                                                                                   SCRN ADDR
 This
                                                                 0050
                                                                                                   TEXT ADDRESS
                                                                            LI RI, MESSZ
           Reading from this month;
LABEL
                                                                 0051
                                                                            LI 82, >17
                                                                                                   LENGTH OF DATA
           E/A
                   Manual.
                                              59
                                                     INDEXED
                                    page
                                                                 0052
                                                                            BLUP OVNEW
                                                                                                   WALTE IT
           ADDRESSING
                                                                  0053
                                                                            BL QGETRE1
                                                                                                   WALT FOR ENTER
UMCK4
                                                                 (0)54
                                                          COPY
                                                                            DATA >0,>22D
           E/A
                   Manual.
                                                229
check
                                    Dage
           DIRECTIVE.
                                                                  0055
                                                                            JMP START2
                                                                                                   GO BACK TO IT AGAIN
 flow
                                                                 0056
                                                                            MOV ORETURI, RIT
                                                                                                   RETURN TO E/A
                                                                  0057
                                                                            RT
           I.McCOMIC A/L PROGRAMMING page 70.
                                                                 0058 * STANDARD CURSOR ROUTINE
. the
                                                                 0059 BETKEL MOV *R11+.R3
                                                                                                   LOAD LENGTH OF ENTRY
           Thats it for this month.
UMCK8
                                                                 0000
                                                                            *0V *R11+-R4
                                                                                                   POSITION OF ENTRY
umber
                                                                  0061
                                                                            MOV R11, GRETUR2
                                                                                                   SAVE RETURN ADDR.
                                 Joe Wright.
 The
                                                                  0.052
                                                                            SLR ODEVICE
nt of
                                                                  0043
                                                                            ELR VSTATUS
into
                                                                            ELR R9
                                                                                                   CHAR COUNT
                                                                  111164
                                                                  0065
                                                                            MOV R3-R6
                                                                                                   LOAD ACCUMULATOR
           (904) * 25-04-87 FILENAME SOURCE=TEST
                                                                  0006
                                                                            MOV RAIRT
                                                                                                   LDAD ACCUMULATOR
           0002 withis programme is a Bemonstration
                                                                                                   LAST POS INTO RO
                                                                  0067
                                                                            Ĥ
                                                                                77. Rb
           3095 * OF THE USE OF THE COPY
                                                                            MOV R4,R7
                                                                                                   SAVE POS INTO R7
                                                                  0068
  the
           0004 * DIRECTIVE IN THE EA. PACKABE .
                                                                                                   CURSOR POS
                                                                  0069 SETK1 NOV R4,R0
te a
           8005 * TWO FILES ARE ADDED DURING
                                                                  0070
                                                                            ELWP DVSER
  The
           0006 * PESEMBLY "NUNCHECK" AND "CLEAR"
                                                                  0071
                                                                            MOVE RIVEDEPDAT
                                                                                                   SAVE EXISTING DATA
tine.
                     DEF START
           0007
                                                                  0072 GETKE2 MOV R4,R0
                                                                                                   NEW CURSUR POS.
  is
           0008
                     REF ABCAN, UMBH, USBN, UMBR, USBR
                                                                  0073
                                                                            LI Ri. >1E00
                                                                                                   LOAD CURSOR DATA
           0009 S7ATU5 EQU →837C
                                           STATUS REG EQU
                                                                  0074
                                                                            BLUP OVEBU
                                                                                                   MALTE CURSOR
                                                                                                   WALT FOR A WHILE
                                                                  0075
           0010 DEVICE EDU >8374
                                            M/BOARD SCAN TYPE
                                                                            RL DOELAY1
                                                                                                   LOAD DEFAULT DATA
                                                                            MOVE ODEFDATAR!
           0011 KEYVAL EQU >8375
                                            KEY PRESS RETURN
                                                                  0075
                                                                  0077
           0012 MESSI TEXT 'NOT A NUMBER'
                                                                            BLUP DVSBW
                                                                                                   WHITE DEPAULT CHAR
icles
           0013 MESS2 TEXT 'PRESS ENTER TO CONTINUE'
                                                                  0078
                                                                            LI R10,600
utine
           0014 ENTERK BYTE >00
                                            EQUATES FOR KEY CHECKS
                                                                  0079 BETKES DEC. RIA
           0015 LEFTK 8Y7E >08
                                                                            JEQ SETKEZ
                                                                  0080
it be
bove
           0046 RIGHTK BYTE >09
                                                                                                 GET GEY ATRONE
                                                                  9(61
                                                                            BLWF TKSCAN
           0017 INSERK 9YTE >04
                                                                  0.082
                                                                            IB ASPACE- ASTATUS
                                                                                                 ANY KEY PRESEED
lve.
           MOTE DEFETA BALE 002
                                                                            INE BETRES
                                                                                                   NO KEY 30 BACK
                                                                  W_{ij}^{(i)}
           0019 ERASEK BYTE 307
                                                                  0084 EVALKI IB - RKEYVAL, 98FACE
Bress
           0020 SPACE BYTE 020
b the
           0021 ZERO - 3Y7E >30
           0022 NINE BYTE >34
           0023 PER
                     577E >2E
           0024 FLAG1 - DATA 00001
  the
           [0025 FLA⊖2 | CATA >000Z
           0026 DEFRAT BSS >2
                                           SPACE FOR DEFAULT CHAR
           0027 MYWER 1995 120
```

WORKSPACE FOR PROGRAMME SAVE RETURN ADDR.

0028 RETURN ASS >2

```
0001 * CHECK FOR WUMBER ROUTINE.
0002 * STRIPS LEADING BLANKS.STOPS CHECKING AT FIRST BLANK FOLD
0003 * REF MUST BE INCLUDED FOR VSBR, VMBR
0004 * DATA FIRST SCREEN POSITION, LENGTH OF FIELD TO CHECK
0005 * R7 SET TO SEFFE FOR BAD NUMBER
(MAIN * NUMBER (5 PLACED IN BUFF)
0007 * FL46: PATA ×0001
0008 * FLAG2 DATA >0002
0009 * BUFF: ESS >20
0010 * ZERB DATA >3000
                              HEX FOR ZERO
doti * NIME DATA DSAGO
                              HEX FOR NINE
0012 * PER DATA >2E00
                              HEX FOR PERIOD
0013 * BUFF1855 >20
                              BUFFER AREA
0014 *
            āĻ
                 anumcki
0015 ×
            DATA >0104.4
                              FIRST SCREEN POS
OMIA WIMEKI NOV *R11+-RO
            MOV *R11+,R4
                              CHARACTER COUNT
0017
                              CLEAR PERIOD FLAG
0018
            CLR
                 Rò
0019
            CLR
                 Ř7
                              CLEAR FLAG REGISTER
0020
            CLR
                 R9
                              CLEAR ACCUM.
0021 NUNCK2 BLWP DVSBR
                              GET CHAR
9022
            CP 
                 RI, JSPACE
                              SPACE CHAR?
0023
            JAE MUNCKS
0024
            INC
                              NEXT CHAR POS
                 Řΰ
                 RO. R4
                              LAST POSITION
0025
            C
0026
            AME MUMCK2
                              NEXT CHAR
0027
            J.M.
                 MUNICKT
0028 NUMERS MOV
                 RO-R3
                              SAVE SCREEN START POS
HOZY MENCKA BLMP DVSBR
                              GET FIRST CHAR
                              NEXT CHAR POS
            INC RO
dote
0031
            INC 89
                              COUNT CHARS
0032
            38
                              SPACE CHAR?
                 RI-DSPACE
0(33)
            JEQ NUMCKS
tių34
            [8
                 RINDER
                              CHECK FOR PERIOD
0035
            JEQ NUMCKS
0036
            58
                 RI-#ZERO
                              15 IT > HEX 30
u(37)
                              SET NOT NUM FLAG
            JET MUNCET
B(10)
                 RI-ANINE
                              CHECK FOR (NINE
            39
0039
            JUT NUMERA
                              GET NEXT CHAR
            JMP NUNCK7
0040
                              SET NOT NUM FLAG
nii41 NINCKS E
                 89-9FLA61
                              FIRST CHAR?
0042
            JED NUMCKT
                              NOT A NUM SET FLAG
0645
                 R9, aFLAGZ
                              IS IT SECOND CHAR?
ឲ្យម៉ូនីនី
            JNE
                NUNCKS
                              RETURN VALID NUM
11045
                 Ro-AFILHGI
                              IS PEIGD FLAG SET?
0046
            JEQ NUMBER7
                              SET NOT NUM FLAG
0(47)
            110
                NUNCka
                              RETURN VALID NUM
0048 NUMCKA C
                 Ros PFLAGI
                              IS IT FIRST PERIOD
            JEG MUNCKT
0049
                              SET NOT NUM FLAS
(0.50)
            INC Ro
            385
0651
                 MUNC 44
0052 NUMEKT LI
                 R7.)FFFF
                              SET NOT NUM FLAG
0053
            JMP
                 MUMCK9
0054 NINCKB DEC R9
                              DEC CHAR COUNT TO STRIP ENTER
0055
            MOV R9.R2
                              LOAD CHAR COUNT
0056
            SMPB RG
            HOVE 89. SEUFFI
4)657
                              LOAD COUNT INTO BUFF
0555
            60V R3,80
0059
            LI RI-BUFF1+1
(i) (i)
            BLUP BUMBR
```

0061 阿配纳 87

0001 * CLEAR SCREEN SUB ROUTINE 0002 * 26-04-87 FILENAME SOURCE=CLEAP 0003 * FIRST DATA=LAST SCREEN FOS 0004 * SECOND DATA LOWER SCREEN POS ()(x)5 * BL GCLEAR1 JA:4 (i(a)) * 0007 * REF MUST INCLUDE VERM 0008 CLEAR: *OV *R11+-R0 特色V 本代記[中:資本 0009 LI R1:72000 BLANK CHAR DATA $\hat{H}(t|s)$ 0011 CLEARZ BLUF OVSBU WRITE BLANK! DEC RO DECREMENT SCREEN POS, 0012 CI RO.R4 0013NO? BACK TO CLEAR? 0014 JNE CLEAR2 0015 RETURN TO MAIN PROBRAMME RΤ



MISSING PERSONS -

Could MERV WALKER or any-one that may know of his where-abouts please contact me with a new address as we have some newsletters waiting to get to him.

Albert Anderson, Secretary.

GRAPHICS AND TI-WRITER

This article originally appeared in the August 1986 issue of TIMES, the newsletter of the Melbourne TI Users Group. The author is Rick Grissom.

Any command you can send to the printer from a BASIC programme can also be sent using the TI-WRITER Word Processor. This article will list several tricks to get more out of both the printer and Word Processor. I use an EPSON MX-80 (same as the TI printer). If your printer uses different Control Codes the procedures should still work, but you'll have to change the codes. The first section of the article will discuss how to send Control Codes. The second section will discuss using the Transliterate (.TL) command and the last section will show how to use the .TL command and graphics to custom design letters.

Printer Control Codes

TI-WRITER has a feature to allow you to send ASCII special character codes. See page 146 of the TI-WRITER manual for a complete list of the codes. CTRL U changes the cursor to an underline and allows ASCII codes from Ø to 31 to be entered. Pressing CTRL U a second time returns to normal. The control codes can be embedded in the text and do not need to be set on a seperate line.

To print double width the key sequence would be CTRL U N CTRL U. You should see a small letter e with a dot over it. To use the compressed mode enter CTRL U O CTRL U. A small f with a dot is shown. To turn off the compressed mode enter CTRL U R CTRL U. Most printer commands use the ESC (ESCAPE) code as part of the command. An example would be emphasised print. To turn emphasised on enter CTRL U FCTN R CTRL U E. FCTN R in the special character mode is ESC and looks like a small b with a line on the upper left. To turn off emphasised enter CTRL U FCTN R CTRL U F. Be sure that any letters (eg E,F) are UPPER CASE.

Using these commands you can bypass the Formatter and print directly using the PF (Print File) command of the word processor.

<u>Transliteration</u>

The Transliterate command is a Text Formatter command that assigns (transliterates) one or more ASCII character values to another ASCII character value. It must be at the beginning of a line with nothing else on the line. It is usually best to place the command at the beginning of a document. The format is:

.TL ni:n2,...nz

An example would be: .TL 35:7

ASCII 7 (bell) is now assigned to ASCII 35 (#). Each time # is encountered the bell will ring.

A better way of using this command is to assign several codes to a

that lease us we o get

P05,

BRHAME.

single character. If you wanted to print an item several times using double width, emphasised and italics, the command would look like this:

.TL 42:14,27,69,27,52

ASCII 42 is \star . 14 turns on double width. 27.69 turns on emphasised. 27,52 turns on the italics. So instead of having to type many commands all that is needed is \star .

Graphics

One of the few drawbacks to the EPSON MX-80 is that there is no way to download a special character set. Because I needed Greek letters, I used the Transliteration command and the graphics ability of the printer to design some.

To design letters draw a box like the one at the end of this article. Draw the shape of the letter by placing dots at the intersection of the rows and columns. On the left side of the box is the pin numbering system. The pins are numbered from the bottom up. For each dot in the column add the number on the left to obtain a total for the column. For example to fire the second pin send the printer a 2, to fire the fourth pin send an 8 to the printer. If you want to fire several pins, total the numbers of the pins to fire. To fire the third and fourth pins send 12 (4+8), to fire all the pins send 222 (1+2+4+8+16+32+64+128). Pin 1 is normally used only for descenders (the letters that descend partly below the line, such as p and y). For more complete explanation of the pin numbering system check the owners manual.

When the printer prints a standard letter it is nine columns by eight rows. The high density graphics mode (ESC L) gives the same spacing of columns as standard printing. When turning on the graphic mode you must tell the printer exactly how many columns to print. To print nine columns the format in BASIC would be ESC;L;9;0 (in ASCII it would be 27,76,9,0). ESC(27);L;76 turns on the graphics, nine is the number of columns to print, the zero would be used for printing more than 255 columns. Along with the command to switch on the graphic the sum of each column must be sent. The final graphic command would be (ASCII): 27;76;9;0;C1;C2;C3;C4;C5;C6;C7;C8;C9 The final step is to use the .TL command. Pick a character which is seldom used and look up its ASCII number.

The final format using ASCII 126 - ~ (FCTN W) - would look like this:
.TL 126:27,76,9,0,134,136,200,112,48,24,12,2,2

When using the .TL command the Formatter must be used. A caution, make sure that there are no typos in the final command. Typos can result in some very strange effects when the file is printed. Using the font to design letters may not get the exact form wanted. I suggest testing the letters from BASIC. This is much quicker than switching between Editor and Formatter.

Listed below are the .TL commands I use. After the .TL command is a comment. The comment shows the character, ASCII code and what is printed. In use, you could not print the comments as shown here, since anything at the end of the line would have to be on a seperate line.

Good luck! If you come up with some good character sets let me know.

DEL LAN THE PI NU

37

16

.TL

.TL

. TL

.TL

.TL

.TL

ALF

BE1

Thi Use

Do thr twi (Re (F)

off and spa wil

che

do a

Who Be Co: ty

be

```
using
 this:
          .TL 123:27,83,1
                               123={=SUBSCRIPT
          .TL 124:27,83,Ø
                               124= ; = SUPERSCRIPT
          .TL 125:27,72
                              125=}=SUB, SUPER, DBLE STRIKE OFF
          .TL 35:27,76,9,Ø,124,146,146,146,146,146,146,146,124
                                                                      35=#=THETA
ns on
          .TL 36:27,76.9,0,28,18,34,34,20,12,20,34,34
                                                            36=$=ALPHA
o type
          .TL 91:27,76,9,0,16,16,24,4,66,68,72,80,32
                                                           91=[=NU
          .TL 92:27,76,9,0,128,128,254,128,128,254,128,128,0
                                                                    92=\=PI
          .TL 93:27,76,9,Ø,6,1Ø,18,34,66,34,18,1Ø,6
                                                          93= ]=DELTA
          .TL 96:27,76,9,0,3,5,28,244,148,148,148,140,96
                                                               96= '=BETA
          .TL 126:27,76,9,0,134,136,200,112,48,24,12,2,2
                                                               126=~=LAMBDA
is no
Greek
         ALPHA = «
ity of
         BETA = A
         DELTA = A
         LAMBDA = \lambda
this
         THETA = 0
 the
         PI = I
box is
         NU = •
 up.
total
ter a
         128.
                               C1= _
                                           128.
ant to
          64.
                               C2=
                                            64.
                                                                  C2=
 the
          32.
                               C3= -
                                            32.
                                                                  C3=
nd 222
          16.
                               C4=
                                            16.
                                                                  C4=
enders
           8.
                               C5=
                                             8.
                                                                  C5=
 For
           4.
                               C6= __
                                                                  C6≃
pwners
                               C7≖
                                             2.
                                                                 C7=
                               C8= __
                                                                . C8=
                               C9=
                                                                  C9≠
ans by
            1234567
                                               12345678
same
aphic
  Tα
II it
                               TI-WRITER TIP
 the
more
 the
ild be
```

This article is reprinted from the newsletter of The Ottawa T199/4 Uses Group, November 1986, and was written by Jane Laflamme.

Do you need to use a reserved character such as the ampersand (&) through the TI-WRITER Formatter? It can be used but you must type it twice in your text - &&. I find it good practice to use the RS (Replace String) function to check this for me. Enter Command Mode (FCTN 9), type in RS <ENTER>, then / & / && / <ENTER>. You then should check them one at at time to make sure you do not inadvertently turn off an underline command; it shouldn't happen with the spaces in front and behind, but remember, that is exactly what it is looking for, a space, an ampersand and another space. If any of them are missing, RS will not find it!

Does two spaces after the period bother you? Do the same thing; let RS do the work for you. When in RS type in $/./.^/$ and again, do it one at a time rather than ALL.

When using RS, remember that it only checks from the cursor down-Before using RS, to make sure it checks the whole document, enter Command Mode by FCTN 9 (ENTER), then type S (for Show line), (ENTER) type 1 for the first line (ENTER) and then to have the cursor at the beginning of the line hit CTRL V.

e. et me

use pits

like

tion,

Using

than

nd is

at is

since

can

BASIC FOR BEGINNERS

The HV99ers BASIC Group Notes

BY PAUL MULVANEY

This is a summary of the first night's instruction of a TI Basic course being run for HV99er members.

The computer has three major sections; the Input section, the Central Processing Unit(CPU) and the Output section.

The Input section consists of the keyboard and allows the user to give the CPU the instructions that he/she wants performed.

The CPU is subdivided into three units; Memory, Arithmetic and Logic Unit(ALU) and the Control Unit.

Memory is further subdivided into Read Only Memory(ROM) and Random Access Memory(RAM). ROM is predetermined by the manufacturer and cannot be changed by the user, hence the name. ROM contains the set of instructions which form TI Basic and allow the electronic hardware contained within the console to recognise the instructions entered by the user and to act upon these instructions. RAM is the area where the user instructions(program) are stored and the data generated by these instructions.

The ALU performs all the arithmetic and comparisons required by the program.

The Control Unit directs the overall operation of the other units and controls the transfer of data between these units during program execution.

The Output section converts the electronic signals into a form that can be interpreted by a television set into the characters and shapes that we recognise.

Because of the time involved in entering a program into the computer via the keyboard provision has been made to store the instructions held in memory. Two types of mass storage devices are available, tape recorder or disk drive. The tape recorder is the cheapest form of mass storage but is very slow compared to a disk drive.

The TI99/4A is a digital computer and uses binary signals to carry out its instructions. The binary system uses two states, 1 and \emptyset or ON and OFF. Because there is no intermediate state it makes a good decision maker, there can be no maybe.

The computer is a logical device which never makes mistakes, we the programmers make the misteaks. To prove this point enter the following program.

```
100 CALL CLEAR
110 PRINT "REMEMBER"
120 PRINT "Computers dont make mistakes" :::::
130 PRINT "Press the number 2 and then enter"::
140 INPUT A$
150 PRINT ::: "I said 2, DUMMY" ::
160 PRINT "Please try again" ::
170 GOTO 130
180 PRINT ::: "WELL DONE"
190 END
```

When to en the progr told.

145 I

After

IMI

We he perform to type of it accepts when on the contract of th

encid VARI/ Numer

A number number sign avoit may Shor

A s spac numb in m

stri eg A

Vari LET the

Var The B=2! C=A Not

Mat

Add

Sub

5.00

Mul

Exp

When this program is RUN the expected result is not obtained. Trying to enter the information in upper and lower case also does not bring the required result. Before you resort to physical violence study the program and you will see that the computer is doing exactly what it was The problem is the ommission by the programmer of the following line is the problem.

145 IF A=="2" THEN 18Ø

course

After adding the line to the program the computer behaves as expected.

entral

IMMEDIATE MODE.

give

We next looked at the IMMEDIATE MODE. In this mode the computer simply performs the given instruction and waits for another instruction to be typed in. To carry out the instruction a second time we have to retype it again, we do not use the RUN command.

Logic

eg PRINT "A BASIC EXAMPLE"

When the enter key is pressed the sentence A BASIC EXAMPLE is printed on the screen. Note that when text is to be printed it must be enclosed in " ".

Random er and set of rdware ed by re the

VARIABLES were discussed next. There are two types of variables.

these

Numeric and String.

by the

A numeric variable can be used to store a numeric value. numbers are stored without a + sign but a space is left in front of the In the case of a negative number this space is filled with a -A space is left after both positive and negative numbers to sign. avoid concatenation of two or more numbers. Valid numeric variables may be upto fifteen characters long:-

and rogram eg A, X, NAME, RATE, CONVERSION_RATE Shorter names use less memory space.

Rt can that A string variable can be used to store a group of letters, numbers, spaces and symbols. The string must be enclosed in quotes. A group of numbers enclosed in quotes is a string NOT a numeric and CANNOT be used in mathamatical operations.

mputer s held tape mass String variable names always have a \$ sign as the last character:eg As, GROUPs, B4s, LISTs

out DN and tision Variables can have values assigned to then using the LET statement. LET A=25 assigns the number 25 to the variable A, LET A=="BOB" the characters BOB to the variable A\$.

we the lowing

Variables are stored in memory and can be used over and over again. They can be used to store the result of mathematical operations. B=25.4+3.39 stores the result of the addition(28.79) in the variable B. C=A+B adds the values in A and B and stores the result in C. Note that the LET was omitted, the use of LET is optional and is rarely used.

Mathematical operations use the following infix operator symbols;

Addition PRINT 235.6+19.34

Subtraction PRINT 8.94-4.0697

Multiplication ¥ PRINT 2.34*5.729

Division PRINT 25000/5671

Exponentation PRINT 8^2



More than one item can be printed at a time in the Immediate mode. To assist in positioning the items print-seperators are used. The three types available are semicolons, colons and commas.

ot

As Lh NU

10

13 14 Ru

Th BI Th

10

T

or

(1

8

21

30

•

ø

R

O

h

-1

Semicolons cause the items to be printed side by side with no extra spaces between the values. Care must be exercised if printing string variables as the words will be joined together if no extra spaces are added.

Colons cause the next item to be printed at the beginning of the next line.

Commas cause the next item to be printed at the next print zone. There are two print zones on the screen, the first starting at print column 1 and extending to column 14 and the second from 15 to 28. (See the worksheet provided)

To become used to these functions try these exercises.

A=25

B=78

C=22

As="JACK"

B\$="JILL"

PRINT A:B:C

PRINT A:B:C

PRINT A,B,C

PRINT A::B::C

PRINT "A=":A:"B=":B:"C=":C

PRINT A\$; B\$

PRINT A\$:B\$.

PRINT A\$,B\$

PRINT A#; A, B#; B

As further exercises try using multiple seperators.

To further assist in positioning of information on screen the TAB function is useful. The TAB function allows you to stipulate the column at which you wish to start printing:

PRINT TAB(5); A=

puts the word JACK on screen starting at column 5. If a number larger than 28 is inserted the computer keeps subtracting 28 until the number is smaller than 28. If the item won't fit on the line the computer moves to the next line and does not break the word.

And there ended the first lesson.

TI BASIC 2

At the Basic Groups second get-together we leave the Immediate Mode and start looking at writing a program. The main advantage of a program is that a number of instructions can be operated on one after the other, they can be made to happen over and over again by simply typing RUN and sections of the program can be altered without retyping the complete program.

A PROGRAM is a numbered list of instructions which are operated on in a set sequence which is determined by the programmer.

To make programming easier TI included the NUMber command. This command automatically provides the program line numbers. If NUM is typed in the default value is line 100 and subsequent numbers will increment by 10. If a different starting line number is required

de. To three simply type NUM then the starting line number. If an increment value other than 100 is required add a comma and the increment number.

As an introduction to a program and revision from the previous session the following program was entered:

b extra | NUM string | 100 A=36

string **ce**s are

110 F=="BILLY" 120 S=="BLOGGS"

130 PRINT F#; S#; A

14Ø END

next RUN

The result was: -

BILLYBLOGGS 36

There plumn 1 See the The BILLY and BLOGGS were joined because of the nature of string variables and the use of a semicolon. To correct the problem we now learn how to EDIT.

The first method is to type EDIT 110 and enter, this brings line 110 up on screen and allows us to move the cursor to the right using FCTN D (the right arrow). By positioning the cursor after the Y we can insert a space and retype the " or we can INSERT by FCTN 2, a space then enter. This INSERT mode is especially useful if you have to add to the middle of a line of code. The second method of editing is the shorthand way and involves typing the line number then FCTN E. A second method of altering the program so it prints correctly is by editing line 130 so it ends up 130 PRINT F#; ";S#;A

A lot can be learnt about the computer by experimenting with short programs like this.

REMEMBER you can't harm the computer if you make a mistake so be adventurous and TRY - the worst you can do is wipe your program by inadvertently QUITing. This can be achieved if you use the SHIFT key on the right side of the keyboard. Because of its closeness to the FCTN key it is possible (and highly probable) that when you want to type the + sign you will slip off the SHIFT key onto the FCTN key and that is QUIT. Many a programmer has discovered this the hard way - an hour or more of typing gone in one slip of the finger! Hence ALWAYS use the left SHIFT key.

e TAB te the

larger number mputer You will have noticed by now that the screen changes to green when the program runs. If you wish to change the colour of the screen during running it can be done by the CALL SCREEN command and a number up to 16. The relevant colours and numbers are given on the worksheet and in the manual. Try typing the following: 98 CALL SCREEN(12)

The screen will be yellow while the program is running.

Another useful command to add as the first line of a program is the CALL CLEAR command. This will fill the screen with spaces and therefore remove ANY distracting material from the screen. 96 CALL CLEAR

de and ram is other, UN and mplete

To see what your program looks like type LIST. The entire program is shown on screen in numeric order, even though it was entered in a different order. The RESequence command will renumber the lines from line 100 with increments of 10. This becomes useful when adding to a program that has already been written.

n in a

When you have finished with a program it can be erased by the NEW command. This wipes memory and allows you to start afresh.

This UM is will quired To allow us to give the computer information while a program is running we use the INPUT command. The data entered via an INPUT command is assigned to a numeric or string variable. If INPUT A is used a question mark is displayed and the program will wait for a numeric value to be entered before continuing. If invalid data is entered a

warning message is printed. An input prompt can be displayed so that know what you have to enter. If a prompt is used there must be a colon separating the prompt and the variable name.

SOFTWARE DEVELOPMENT.

Software development or the writing of programs should be divided a number of steps so that it is an orderly progression rather than a haphazard wandering. The steps are:

3

-,2

1

263

1,7 6,9

2.3

4 11 1

5 Q

XXXX

* 4

* 4 *

STAF STAF

Supe

44 F A C

los

5#C the

tit

bli wha

YOU

how 747

C QM

d i s

The

an. new

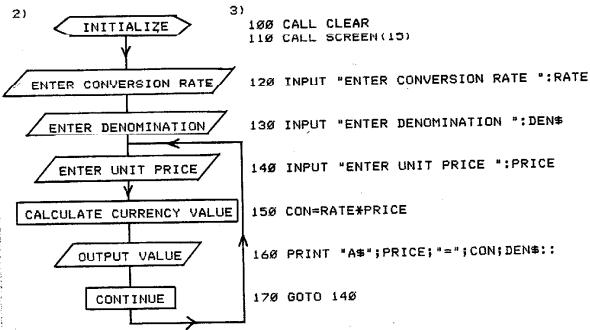
and

Mod

als

- 1) Problem Definition.
- 2) Program Design.
- 3) Coding.
- 4) Debugging.
- 5) Testing.
- 6) Documentation.
- 7) Maintenance.
- 8) Extension and Redesign. (For more information see HV99 Newsletter Nos 10 and JUNE 86 issues)
- i) The problem is:- You are going overseas soon and you want purchase a video camera, (\$2456) a zoom lens (\$173) and a stereo unit (±698). To ensure you are getting a good deal with your duty free shopping you want to calculate the value of each item in the currency of each country you will visit. The countries to be visited and the conversion rates are:

Malaysia M==1.76 Hong Kong HK#=5.55 Japan Yen=98.19



to create an endless This program uses the GOTO command Everytime we want to change the currency we will have to 'crash' program using FCTN 4 and then RUN it again. At the next get together reu we will refine the program and make it suit our problem more closely. LIN Try and design some changes yourself using your Users Reference Guide. War

Don't forget YOU are welcome to attend Paul's BASIC classes. From all reports the group is very enthusiastic and are always on the lookout Res for date of DPA for more starters. See the back page of this newsletter the next class.

that t be a

into than a

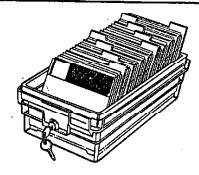
95)

nt to

unit

free rrency

the



SOFTWARE LIBRARIANS NEWS

BY BL LAMBENCE

99'ers. HI

*********************** **** S. T. A. R. *** ¥ ¥ ******************************

STAR by Michael Riccio (COM-LINK). STAR is the acronym for-Super TI Assembley Routines.

A collection of 53 routines, which ************************ loads from XB and hides away in LOW section of Memory Expansion. In fact the cursor just reappears after the title and ad screens, winking and blinking at you as if to say "Now what?" NO HINTS! NOTHING! for all you gung ho types who can work out how to run anything. There is no except to read the very comprensive documentation on the disk SO THERE!

These routines can be used to modify an XBasic program or help write a "VDP Memory", new one. "Colors", "String Handling", "Character and Sets" to name but a few. A11 routines acessed CALL via LINK("xxx").

Works with TI disk controller card and any Memory Expansion or RAMDISK Read, Alter, Rewrite Sectors same as DPATCH (not as good as the F'WEB Modified version) GRAPH 40 col TEXT also acessed by CALL LINK

A worthwhile disk in any Library to help in speeding up XBasic progams.

The disk includes a few sample demo progams and HELP print out the DOCS

POSITION VACANT. See ad elsewhere and fill in application form in the HV99'er. DISCover and FORMAT the year ahead. +++ FULL TIME +++ **************************

- ZIP - ZILCH -NIL- NONE -

MAY MADNESS STOCK CLEARANCE ¥ ONE NIGHT ONLY CASH ONLY ¥ * PERSONAL SHOPPERS ONLY ¥ ¥ TAPES ONE normal \$\$\$ * Buy TWO get one FREE ¥ Buy THREE get two FREE ¥ × HURRY WHILE STOCKS LAST ¥.

Cassette Data Base

After last months Disk Data Base Demo a few inquiries came my way to the available ones for Cassette.

A very good one by 8. Rutherford has been published in an earlier (July 86) HV99'er Newsletter and is in the library but in case you missed it or it was not quite what you required, included for all you Masocistic Typist Types eleswhere is a good one donated by J. Smart to HV99'er library when he moved to another brand of computer. This program was used by him to track his affairs and was named the MERLIN FILE (you may have heard of Wizard Merlin - well by the time you type this in you may wish he was still around. The program is available on tape from the library.

loop. the gether bsely. iide.

all a ook ou t te of



FUNNELWEB

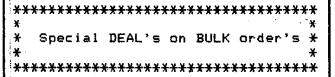
Why not update yours now - There so much more in it. If you are like some people who said its TOO much trouble to keep CUSTOMISING the LOAD, Then you have not been reading For those who have the FWDOC/REPT on MERGING the LOAD you have customised already. DO IT NOW.

All KNOWN bugs eradicated and a new CALL SAY("TRY AGAIN") feature included.

FULL INSTRUCTIONS. "READ 'EM".

Bugs, Speaking οf The Spyglass Award goes to Super Sluth of the Month J. the Curly Mystery of the MISSING (?) Ramdisk (to the system) all because HOW ABOUT THAT! it isn't just better of a "tiny" conducting hair. Full it is GOOD!. Investigateive Report elsewhere, big THANK you Joe.

All the Club Software is available on Demand. \$ 4.00 P/P. each Disk.



Happy Hacking

Al Lawrence.



IMPROUE YOUR SPEECH

BH DOE WRIGHT

the speech synthesizer and extended basic, here is something you may like to try. In the command mode type;

The computer makes a reasonable job of saying "try again".

Golden Now try this; type;

Wright who solved CALL SAY("#TRY AGAIN#")

This information has the The hash signs can be rommonly used been mentioned in many articles over the years. used with other phrases. That are listed in the back of your Ex. Basic hand book along with the list on vocabulary words for the speech synthesizer.

I have tried the following phrases and they all improve greatly;

> #GOOD WORK# **#HANDHELD UNIT#** HI WINH #READY TO START# #SUPPOSED TO# **#THAT IS INCORRECT# #THAT IS RIGHT# #TRY AGAIN#** #WHAT WAS THAT# #YOU WIN#

individual phrases can also be used inside the one statements to sentences for use in you programme. Some of the samples have tried and sound very good are;

CALL SAY("#THAT IS RIGHT#, #TRY AGAIN#")

CALL SAY("#GOOD WORK#, #TRY AGAIN#")

CALL SAY("#THAT IS INCORRECT#, #TRY AGAIN#")

CALL SAY("#GOOD WORK#.#THAT IS RIGHT#.#TRY AGAIN#")

.

2.4

4

. 2

10 A.F.

*

44

34

b ## 2

fee. Agai gran

244 204

SOM

OBE i **5** i nite WHE

res to 1111 cor 44.5

> Eg CLS

> > and top Som

to wr i

speech c, here try.

ble job

better on has 49V6 # can be used in the book bulary ter.

hrases

lso be ts to n you les I are;

AIN#*)

#TRY

(MAY-87/01) 1 3 ₹ 4 5 3 ò

SCR #51

STRUGGLING FORTH

HVPPERS MAY ARTICLE 1987

EXPLOPING VDP-RAM /PART2

-FUN WITH SCREEN IMAGE TABLE

-SUPER QUICK NEW CHARACTERS SETS 10 11 -WRITING TO VDPRAM WITHOUT USING

12 VEBW, VMBW

13 14

15

Well, we will soon be into winter, usually a productive time for computer buffs, but in my case my seasonable work load increases, and as I don't have a computer at home, the reverse applies.

top. Refer to your TI-FORTH manual, I've discussed Chapter 4, page 3. mentioned graphics/sprite/color.

Before we start, photostat the page by the variable USE. and stick it somewhere in front of screen 61 for such a routine you.

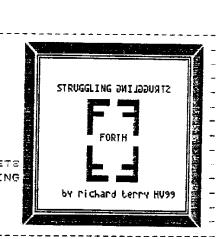
SCREEN IMAGE TABLE.

Think of the area marked "TEXT MODE | This occupies the next 32 bytes. SCREEN IMAGE TABLE" as a duplicate think its needed when information, whatever on the screen. Unlike the really scraping for byte space. rest of VDP-RAM, whatever you write the screen image table will STACK FOR VSPTR. immediately aobear מם thei

Eg try typing:

CLS Ø Ø AT 65 EMIT

and a capital A will appear at the I've covered the how and why of writing commands: VSBW, VMBW



VSBW/VMBW/VSBR/VMBR

These stand for VDP single/multiple byte write. They are blindingly fast in execution, as you have seen from last month's example. You can store all sorts of data/charts in the unused area of the VDP chip and Today we'll continue our inspection quickly write it to the screen as of the VDP chip, starting from the needed a la "shadow screen concept" before. I know I've develop your own routines here it's some of these things easy to see if they are working by before, but today I'll expand on a writing a routine to transfer what few and ask for help on a few. you think you've put in there back Again my discussions will exclude up to the screen image table, via an intermediate buffer area, such as one of the disk buffers pointed to Refer to CHECK-VDP.

VDP ROLLOUT AREA.

you TV screen sitting on the page. This floating point operations (someone occupies 960 bytes of space: ie it correct me if I'm wrong), so I guess is possible to fit 960 bytes of if your not using these within your or characters, or program its spare space if your

corresponding position on the screen Occupying 128 bytes I havn't got a clue what its used for PLEASE!!!

PABS (PERIPHERAL ACCESS BLOCKS)

top left hand corner of the screen. these superfically in a much earlier Sometimes it is quicker and handier article. For the uninitiated who to use the more primitive screen missed it, briefly this is a memory set aside to pass on information about any peripheral non console devices you may be accessing such as printers etc. Each PAB consists of where, name. Occupying 800 bytes, this into memory using the CHAR command IMPLY allocation is excessively generous as even with a printer and a few disk files in simultaneous use its hard to see how one would use more would give us a new capital A. than 100 bytes.

SPRITE MOTION TABLE.

Occupying 128 bytes, if your not into sprites (like me) and couldn't give a proverbial in which nether regions they are currently residing, top of this area with your data, but next bit in the memory! go no further when you get to HEX know the starting address Ø8ØØ(DEC 2Ø48). So, what's we'd all be in the dark without it, because it it the.....

PATTERN DESCRIPTOR TABLE.

Here resides the descriptions of all the characters you see (and some you don't on the screen. see) Conceptually they are like row after row of portraits hanging in a gallery, except in the murky depths[of the vdpchip they exist numbers, each 8 bytes long.

OK. Sit at the terminal and type:

DECIMAL 65 CHARPAT CR . CR . CR . CR

which will present you with

ie the character pattern for A, worked out the same way as in extended basic, except that in TEXT mode only the leftmost & bits are displayed (see CHAPTER6, PAGE 6), so defining new characters.

Knowing that there i s a new character every 8 bytes in VDPRAM, may sound dreadfully boring but it! has vast implications for your programming.

an earlier article you may

using a disk with 10 bytes + the length of the device already defined, we could load then

> ie 3844 447c 4444 4400 45 CHAR As th fract acces!

> use 1 I was never really happy with itsus wi slowness, so I sat down today andload worked out a quicker way to do it, chara almost intantaneously. takes

> > can

chara

typis

Refer to screens 52,53.

the p you can buildoze on right over the This code actually reads the patternneed descriptor table directly, bypassingon co CHARPAT command (whichsize the allocation is important, so hang on, presumably does the same). Since weyou we ofare a so character Ø is 2048 and there is aas special about that I hear you all new character every 8 bytes, we candon to chorus?. Well, it just so happens either save the entire table en masse, or from the start of oneWRITE letter up to what you desire.

> SAVE-PDT does this. buy or You Beg, borrow a Forth program withis po in save the 🎕 descenders it. To laboriously writing the result tocomme disk from paper as I did due to myscree faulty cerebration last time, wether a simply load it to the desired diskeven. block with an offset of 192 Bytesingut as (arbitary value) to leave us space of the top of the screen to writeNot code to reload it when needed.do Refer to SCR# 52. LOAD-PDT simplythe reverses the process and shoves the start characters back into the PDTable. Istore unable to print the actual which result here because one of symbols when sent to the printerstant obviously moonlights as a printer control character because it form Eg 5 fed my paper (thank god I was'nt out out out out of the room!) らし 単純文

The s The quickest way of reloading is not bytes via a definition, but simply to table include the code words in the still immediate mode on the screen, such you as Scr 55, so that they will execute Chang when the screen is compiled. Note var you must take this into account when in this example I have used VMBW. alter If you want to change your character track set first up on the boot screen, you south must use 2 SYSTEM instead of VMBW.

The word ;S is worth comment. I've situ never had occasion to use it before. typir It simply tells the compiler to what ignore whats on the rest of the screen and return to the calling Anywa remember my ASCII SNAFFLE routine, | routine, which might be for instance of 🕔

them then mand

ith its y and do it,

battern assing (which nce we οf e is a e can le en one

buy or with save ilt to to my e, we d diek Bytes space write eeded. simply es the le. I actual the rinter rinter

is not y to the such kecute Note VMBW. racter you MBW.

t form

nt out

I've fore. to of the Illing stance your program.

IMPLICATIONS FOR USE.

As this routine works in the brief fraction of a second it takes to access a single disk block we can use it before our program to present us with a nicer character set, or to sets of say graphics load new characters for our program. Ιt takes up absolutely no memory. One can also totally change the character set, within the running of the program if you had some special need, and re-load your original set on continuation. If you examine the size of the pattern descriptor table you will see the characters 128-255 are available for you to redefine, as well as any standard ones you don't use.

WRITING TO VDP WITHOUT VSBW/VMBR.

You may not be aware of it, but it is possible to write to any area of vdpchip using the ordinary your commands you would use on screen, such as EMIT, TYPE, etc, all the number formatting commands, and even accept data there using your input commands.

Not so hard really, all you have to do is to trick the computer to think the screen image table actually starts elsewhere than Ø, simply by storing into the CURPOS variable, which stores the current qbv position of the cursor, the VDP start address you want to write to.

Eg 5120 CURPOS ! will send any output in the program after this starting to 5120 in the vdp area. The system will treat the next $96\emptyset$ bytes of vdp as the screen image table for writing purposes, but will! still display on your monitor what write things say to the top and you were initially watching. alter the display, rather one loses area for your special routine, track of the cursor as its "gone south" into the bowels of the chip flashing unseen. If you are a good typist you can resuscitate the without rebooting situation bу typing blind and restoring things to what they should be.

Anyway, that aside, the implications of using ordinary commands to write

to vdp, are that one can work out routine such as to draw a table or something you want to present later, but store in vdp, on the monitor in front of you. When you later want to store it in the VDP chip you simply change CURPOS to the required address and execute the routine. which must of course include the restitution of CURPOS at the end. The uses are limited only by imagination.

As an example try SCR# 61.

After loading, type FILL to fill the screen with A's, then type CLS A->VDP. Notice nothing visible Next type CHECK-VDP. happens. comes back fast dosn't it!

SCROLLING PART OF THE SCREEN.

Power up your computer and try changing some of the following to see the variables effect on screen display:

SCRN-START = ist posit on screen SCRN_END = last posit on screen

By altering these we can change the points within which we write or scroll the screen.

Load Screen 60

and type SCROLL-CHANGE. You are now in a screen within a screen. A11 the ordinary commands will work. Type:

TEST

and watch it work only within Your new area.

Hence within a program, you can the bottom above what will become your Changing the SCRN_START and SCRN_END | new screen area, and then execute a variables doesn't seem to be able to word such as SCROLL-CHANGE, use this then change the variables back to their predefined 0.959 amounts, by a word like SCROLL-FIX.

> Well, thats all for this month. Next month I might dissect the full screen editor for you. seemingly tedious exercise is well worth while, as it then enables you to include in your programs an

editor of any size, anywhere on your screen, doing exactly what YOU want!

Forth programmers are often accused of never writing any REAL programs. Well, they do. In coming issues we will write quite a large and useful program from scratch to the end, using all the lessons in the last 12 months, which will enable you to design and save titlescreen images your program jazz uр presentation. It will make use of the full 40 * 40 column editor and some nice zappy windows as well. Following that, part of whole of a program that will allow you to take DisVAR8Ø files you have written and formatted with TI-WRITER, transfer these into the FORTH. environment and onto Forth screens, enabling you to build documentation for presentation within your Forth Program

That will keep us busy for some months to come!!!!!

ADDRESS FOR CORRESPONDENCE

RICHARD TERRY 141 DUDLEY RD WHITEBRIDGE 2290 049 436861/22450 AUSTRALIA

```
SCR #52
 O ( NEW CHARSET descenders
                                      6Mang7)
  i ( Saves pat descriptor table for from pdt adr for 95 charact
  2 ( to save more/less alter line 5,11
                                       ( exp, PDT adm, scr#-\save to )
  3 : SAVE-PDT
                                       ( address new characters are)
                  BLOCK UPDATE 192 +
  4
                                       ( 95 8 byte character codes )
  5
                  730
                                       ( deposit in disk block erea)
                  UMBR
                                       ( Flush these back to disk )
                  FLUGH
                                       ( leaves nothing on stack
  8
                                       ( exp, PDT adv, sard-91d from )
    דפק-פומטו:
                                       ( address new characters are)
                  BLOCK 192 ±
 10
                                        i stach new adrivado
 11
                  SWAP
                                       / PS 8 byta character codes )
                  760
 12
                                       ( write these to pot area
 13
                   UMEN
                                        ( leaves nothing on stack
 3.14
```

```
気の点 発売等
 O ( NEW CHARSET descenders
                  Mar687)
 1 55 BLOCK 192 + 2304 760 VMBW ; ;S
 15
SCR #40
O ( example to show scrolling part of a screen
 1 1 SCPOLL-CHANGE
                   I expects nothing on stack ?
          240 SCRN_START !
                  ( start at row 3
 3
          760 SCRN_END
                 t ( snd at new 18
 4
          O O SETEMY
 5
 : TEST
          100.0
          ÐÜ
 0
                   f print loop index
           "." TESTING"
 0
                   " print "testing"
10
           CR
11
          FOOR
12
13 : SOROLL-FIX
          O SCRN_START !
          PSP SCRN_END !!;
14
15
SCR #61
 O ( example to show writing to vdp with ordinary words)
 1 : FILL
                   f gs to how 0 col 0
 3
         DO 45 EMIT LCOP :
                  ( fill screen with A's
 : A->VDP
         5120 CURPOS !
                   ( start writing to HEX 1400 )
         FILL
 3
         O CURPOS! :
                  ( neset curson to sareen
10 : CHECK-VDP
         CLS
                   ( clear the screen
                                 •
: :
         5120
                   ( spot we put the A's
12
         USE 3
                   ! address in buffer area
15
         760 VNDR
                   1 mays A's here
1.2
         USE @ 0 940 VMBW 1
                   ( write back to screen
=
```

CASSETTE BASED DATA BASE

H PROGRAM BY

5 ! DONATED TO HUNTER VA 99'ers LIBRARY 1986	LLEY
10 DIM F\$(99.25).YDM(20)	, XDM
(20):: DIM NM\$(20):: ON : ING NEXT	WARN
100 CALL TITLE	
110 CALL CLEAR	
120 CALL SCREEN(5):: DIS	PLAY
AT(2,2):"[MENU]" 130 DISPLAY AT(4,1):"	1)
Design File	2)
Write File	3)
Delete File Save File	4)
14Ø DISPLAY AT(8,1):"	5)
Load File	6)
Print/Display File	フ)
Save File Format Load File Format	8) 9)
Exit"	7)
150 CALL NEON("Please cho	ose
option",K)	4 = ~
160 IF K<49 OR K>57 THEN 170 ON K-48 GOSUB 500,100	159
ØØØ, 25ØØ, 3ØØØ, 35ØØ, 45ØØ, 5	oo, l Gøøø
, 4000	
18Ø GOTO 11Ø	
500 CALL CLEAR :: CALL SO N(14):: DISPLAY AT(2,2):"	REE 1
LE DESIGNI"	•••
505 DISPLAY AT(4,1):"	To
design a file use the foling commands:"	1 ом
506 DISPLAY AT(6,1):"	'E'
MOVES cursor up	,×,
	'S'
moves cursor left moves cursor right"	, D,
507 DISPLAY AT(10,1):"	' 1
' enters text mode ctr ' exits text mode EN	1'E
'exits text mode EN	TER
enters file design ctr 'clears design"	1,C
508 DISPLAY AT(14,1):"	, A
' returns to menu	
<pre></pre>	7 1
o start".K)) £
520 X1, Y1=1 :: CALL CLEAR	
53Ø CALL GCHAR(X1,Y1+2,CH)::
DISPLAY AT(X1,Y1)SIZE(1) R\$(30):: CALL KEY(0,K,S):	: CH
ISPLAY AT (X1, Y1) SIZE(1):C	・ リ HR事
(CH)	

```
535 IF K=131 THEN 520
 536 IF K=65 THEN RETURN
 54Ø IF K=69 THEN X1=X1-1 ::
 IF X1<1 THEN X1=24
 55Ø IF K=88 THEN X1=X1+1 ::
 IF X1>24 THEN X1=1
56Ø IF K=83 THEN Y1=Y1-1 ::
 IF Y1<1 THEN Y1=30
57Ø IF K=68 THEN Y1=Y1+1 ::
IF Y1>3Ø THEN Y1=1
58Ø IF K=73 THEN 600
59Ø IF K=13 THEN 75Ø ELSE 53
600 CALL GCHAR(X1, Y1+2, CH)::
 DISPLAY AT (X1, Y1): CHR$ (30):
: CALL KEY(Ø,K,S)
601 IF K=133 THEN DISPLAY AT
(X1,Y1);CHR$(CH);; GOTO 530
602 IF K=131 THEN 520
605 IF K=8 THEN Y1=Y1-1 :: I
F Y1<1 THEN Y1=3Ø
607 IF K=9 THEN Y1=Y1+1 :: I
F Y1>30 THEN Y1=1
61Ø IF K>3Ø THEN DISPLAY AT (
X1,Y1):CHR$(K):: Y1=Y1+1 ::
IF Y1>3Ø THEN Y1=1
62Ø DISPLAY AT(X1,Y1):CHR$(C
H):: GOTO 600
750 SF=0 :: DM=1 :: CALL SCR
EEN(2)
76Ø FOR LN=1 TO 24
77Ø FOR CL=1 TO 32
78Ø CALL GCHAR(LN,CL,CH):: I
F CH=32 THEN 785 ELSE IF SF=
Ø AND CH()32 THEN XDM(DM)=LN
 :: YDM(DM)=CL :: SF=1
781 IF SF=1 AND CH>32 THEN N
M$ (DM) =NM$ (DM) &CHR$ (CH)
785 IF CH=32 AND SF=1 THEN S
F=Ø :: DM=DM+1
800 NEXT CL :: NEXT LN
81Ø MX=DM :: RETURN
1000 CALL CLEAR :: CALL SCRE
EN(3)
1010 DISPLAY AT(2,2):"[FILE
WRITING3"
1020 DISPLAY AT(4,2): "WHAT N
UMBER FILE IS THIS?"
1030 ACCEPT AT(5,2)BEEP SIZE
(2):FLN :: IF FLN=Ø THEN RET
URN
1040 CALL CLEAR :: FOR Z=1 T
D MX
```

1050 DISPLAY AT (XDM(Z), YDM(Z)):NM#(Z):: DISPLAY AT(XDM(Z), YDM(Z)+LEN(NM\$(Z))):F\$(FLN ,Z) 1060 NEXT Z 1070 FOR Z=1 TO MX-1 1080 ACCEPT AT(XDM(Z), YDM(Z) +LEN(NM\$(Z)))SIZE(-31):F\$(FL N,Z) 1090 NEXT Z 1100 CALL CLEAR :: DISPLAY A T(4.1): "WRITE ANOTHER FILE?(Y/N) " :: FOR D=1 TO 5Ø :: NE XT D 1110 CALL KEY(Ø,K,S):: IF K= 87 THEN 1000 ELSE IF S=0 THE N 111Ø 1120 RETURN 2000 CALL CLEAR :: CALL SCRE EN(9) 2010 DISPLAY AT(2,2):"[FILE DELETION 3 * 2020 DISPLAY AT(4,2): "WHAT N UMBER FILE IS TO BE DELETE D?* 2030 ACCEPT AT (7,1) REEP VALIT DATE(DIGIT)SIZE(2):FLN :: IF FLN=Ø THEN RETURN 2035 DISPLAY AT(8,2): "THE FI LE IS BEING DELETED* 2040 FOR Z=1 TO MX-1 2050 F\$(FLN,Z)="" 2060 NEXT 2 2070 CALL CLEAR :: DISPLAY A T(2,1): "FILE NUMBER"; FLN; "DE 2080 DISPLAY AT (4,1): "DELETE ANOTHER FILE?(Y/N)" :: FOR D=1 TO 5Ø :: NEXT D 2090 CALL KEY(0,K,S):: IF K= 89 THEN 2000 ELSE IF S=0 THE N 2090 2100 RETURN 2500 CALL CLEAR :: CALL SCRE EN(13) 251Ø DISPLAY AT(2,2):"[FILE SAVING) * 2520 DISPLAY AT(4,2): "WHAT F ILE DO YOU WANT SAVED?" 253Ø ACCEPT AT(6,1)SIZE(2)BE EP VALIDATE(DIGIT):FLN :: IF FLN=Ø THEN RETURN 2540 OPEN #FLN: "CS1", OUTPUT, INTERNAL, SEQUENTIAL, FIXED

255Ø PRINT #FLN:F\$(FLN,1),F\$ (FLN, 2), F\$(FLN, 3), F\$(FLN, 4) 2551 IF MX<5 THEN 2564 2552 PRINT #FLN:F\$(FLN,5),F\$ (FLN,6),F\$(FLN,7),F\$(FLN,8) 2553 IF MX<9 THEN 2564 2554 PRINT #FLN:F\$(FLN,9),F\$ (FLN, 10), F\$(FLN, 11), F\$(FLN, 1 2555 IF MX<13 THEN 2564 2556 PRINT #FLN:F\$(FLN,13),F \$(FLN,14),F\$(FLN,15),F\$(FLN, 16) 2557 IF MX<17 THEN 2564 2558 PRINT #FLN:F\$(FLN,17),F \$(FLN,18),F\$(FLN,19),F\$(FLN, 20) 2564 CLOSE #FLN 2565 CALL CLEAR :: CALL NEON ("...Hit [ENTER] to continue ..",K) 257Ø CALL CLEAR :: DISPLAY A T(4,1):"SAVE ANOTHER FILE?(Y /N) * :: FOR D=1 TO 5Ø :: NEX TD 2580 CALL KEY(0,K,S):: IF K= 89 THEN 2500 ELSE IF S=0 THE N 258Ø 259Ø RETURN 3000 CALL CLEAR :: CALL SCRE EN(16) 3Ø1Ø DISPLAY AT(2,2):"(FILE LOADING3" 3020 DISPLAY AT(4,2):"LOAD I NTO WHAT FILE NUMBER?" 3030 ACCEPT AT(5,2)SIZE(2)BE EP VALIDATE(DIGIT):FLN :: IF FLN=Ø THEN RETURN 3040 OPEN #FLN: "CS1", INPUT INTERNAL, SEQUENTIAL, FIXED 3Ø42 INPUT #FLN:F\$(FLN,1),F\$ (FLN, 2), F\$(FLN, 3), F\$(FLN, 4) 3043 IF MX<5 THEN 3059 3Ø44 INPUT #FLN:F\$(FLN,5),F\$ (FLN, 6), F\$ (FLN, 7), F\$ (FLN, 8) 3Ø45 IF MX<9 THEN 3Ø59 3Ø46 INPUT #FLN:F\$(FLN,9),F\$ (FLN, 10), F\$(FLN, 11), F\$(FLN, 1 2) 3Ø47 IF MX<13 THEN 3Ø59 3Ø48 INPUT #FLN:F≢(FLN,13),F \$(FLN,14),F\$(FLN,15),F\$(FLN, 16)

3Ø49 IF MX<17 THEN 3Ø59

3Ø5Ø INPUT #FLN:F\$(FLN,17),F \$(FLN, 18), F\$(FLN, 19), F\$(FLN, 2Ø) 3059 CLOSE #FLN 3060 CALL CLEAR :: CALL NEON (".....Hit (ENTER) to contin ue",K) 3070 CALL CLEAR :: DISPLAY A T(4,1): "LOAD ANOTHER FILE? (Y /N) " :: FOR D=1 TO 5Ø :: NEX T D 3080 CALL KEY(0,K,S):: IF K= 89 THEN 3000 ELSE IF S=0 THE N 3Ø8Ø 3100 RETURN 3500 CALL CLEAR :: CALL SCRE EN(9):: DISPLAY AT(2,2):"[FI LE DISPLAYI" 351Ø DISPLAY AT(4,2): "WHICH FILE TO YOU WISH TO EXAMIN E?" 352Ø ACCEPT AT(7,1)SIZE(2)BE EP VALIDATE(DIGIT):FN :: IF FN-Ø THEN RETURN 3530 CALL CLEAR :: FOR Z=1 T 354Ø DISPLAY AT(XDM(Z),YDM(Z)):NM事(Z):: NEXT Z 355Ø FOR Z1=1 TO MX-1 356Ø DISPLAY AT(XDM(Z1),YDM(21) +LEN(NM#(Z1))); F#(FN, Z1) 3565 NEXT Z1 3570 CALL NEON("Hit any key to continue....",K) 3580 CALL CLEAR :: DISPLAY A T(4,1): "EXAMINE ANOTHER FILE ?(Y/N)" :: FOR D=1 TO 5Ø :: MEXT D 3590 CALL KEY(0,K,S):: IF K= 89 THEN 3500 ELSE IF S=0 THE N 359Ø 3600 RETURN 4000 CALL CLEAR :: CALL SCRE EN(8):: DISPLAY AT(2,2):*[EX ITING1" 4005 DISPLAY AT(4,2): "HIT 'C ' TO CONFIRM" :: FOR D=1 TO 5Ø :: NEXT D :: CALL KEY(Ø,K ,S):: IF K=67 THEN 4Ø1Ø ELSE IF S=Ø THEN 4005 ELSE RETUR М 4010 CALL CLEAR :: END

4500 CALL CLEAR :: CALL SCRE EN(16):: DISPLAY AT(2,2):"[F ORMAT SAVING!" 4505 DISPLAY AT(4,2):"HIT 'C ' TO CONFIRM" :: CALL KEY(Ø, K,S):: IF K=67 THEN 451Ø ELS E IF S=Ø THEN 45Ø5 ELSE RETU RN 451Ø OPEN #255: "CS1", OUTPUT, INTERNAL, SEQUENTIAL, FIXED 452Ø FOR FMS=1 TO 2Ø STEP 2 4532 PRINT #255:XDM(FMS),YDM (FMS),NM\$(FMS),XDM(FMS+1),YD $M(FMS+1),NM${FMS+1}$ 4538 NEXT FMS 4539 PRINT #255:MX 454Ø CLOSE #255 455Ø CALL CLEAR :: CALL NEON ("Hit [ENTER] to continue... .. ".K):: RETURN 5000 CALL CLEAR :: CALL SCRE EN(15):: DISPLAY AT(2,2):"(F ORMAT LOADING]" 5005 DISPLAY AT(4,Z):"HIT 'C TO CONFIRM" :: CALL KEY(Ø, K,S):: IF K=67 THEN 5010 ELS E IF S=Ø THEN 5ØØ5 ELSE RETU RN 5010 OPEN #255: "CS1", INPUT , INTERNAL, SEQUENTIAL, FIXED 5015 FOR FML=1 TO ZØ STEP 2 5Ø2Ø INPUT #255:XDM(FML),YDM (FML),NM⊕(FML),XDM(FML+1),YD M(FML+1),NM\$(FML+1)5030 NEXT FML 5038 INPUT #255:MX 5Ø4Ø CLOSE #255 5050 CALL CLEAR :: CALL NEON ("Hit [ENTER] to continue... ..",K):: RETURN 10000 SUB TITLE

10000 SUB TITLE
10010 CALL CLEAR :: CALL SCR
EEN(3):: FOR C=0 TO 12 :: CA
LL COLOR(C,7,12):: NEXT C
10020 DISPLAY AT(10,9): "MERL
IN FILE"
10030 DISPLAY AT(12,8): "by J
ohn Smart"
10040 DISPLAY AT(14,5): "(Don
ated to HV99'ers)"
10045 DISPLAY AT(16,8): "(Lib
rary 1986)"

XXX

Th 501

AL Th st

si so fe

BO

Sa bo Li ca

SO Tw ta

CA Ca

Ca lo lo La

CO Gi ma

in



10050 CALL NEON("Hit [ENTER] to start....",K) 10060 SUBEND

1Ø1ØØ SUB NEON(M≢,K)

10110 M\$=SEG\$(M\$,LEN(M\$)-1,1
)&SEG\$(M\$,1,LEN(M\$)-1):: DIS
PLAY AT(23,1):M\$:: CALL KEY
(Ø,K,S):: IF S=Ø THEN 10110
10120 SUBEND



DISK REVIEW

This months disk is from the Lima Users Group in the states and has some interesting record keeping programs on it.

ALBUM FILE

This is a basic program that runs without memory expansion. It can store artist, titles and songs of about 300 albums and tapes on one single sided disk. Also it can search for albums with a particular song or search for all albums by a particular artist. Has quite a few features which can be read by listing the program, it is important to read all instructions. Could be very useful for anyone with a large record library.

BOOK FILE

Same program as album file only modified to keep track of all your books, has all the same features.

Like album file list the program and read all the instructions carefuly.

SONGBOOK & TAPEFILE

T.... / 1

Two more file programs one to record songs in your music books and tapefile to make a record of songs and their authors on your cassette tapes.

CALL PEEK

---- ---

Can be used to print out contents of memory locations by entering lowest and highest memory adress. Can also be merged with a program to look into memory adresses. Could be handy to anyone into Assembly Language.

COMPOUND INT

Gives you an idea what compound interest means and compares daily, monthly, quarterly and yearly growth. Could be a help to anyone investing money for maximum return.



FILE READER

A basic program which will read a DIS/VAR 80 file and print it to screen or send it to your printer. It is only a short program so it could be stored on disks with TI-Writer files on them to save loading TI-Writer to read them or print them out.

LOTTERY

A short basic program that selects seven random numbers and plays a tune. The tune has had a lot of effort put into it but the rest of the program needs tidying up.

PHONE BILL

Designed for phone charges in the States but could perhaps be altered for use here by anyone paranoid about their phone bill.

CHECK BOOK WRITER

A program that can only be used by those with a complete system.

Anyone who relies on their check book to operate a business or pay all their bills should get a copy of this program.

You can imprint your checks and print the amounts on the stubs with your printer. Sheets of the stubs can be saved in a notebook form for your record. The data is automatically merged into a disk file which can be printed out or displayed on the screen.

Its documentation is quite good and the program is more complex and has more features than I can list here.

Finaly I would like to mention the enthusiastic approach that PAUL MULVANEY has taken in teaching the BASIC group and say thanks to him on behalf of everyone.

Alan Franks.









So that more members can be involved in the production of our Newsletter it has been suggested that a disk to be reviewed plus a blank disk be sent to out of town members with a request to do a review (similar to Alan's above). The member could use the blank disk to write his review to and the disk to be reviewed would be his to keep in appreciation of his effort. It is hoped that this way everyone would feel that they are contributing something towards Newsletter.

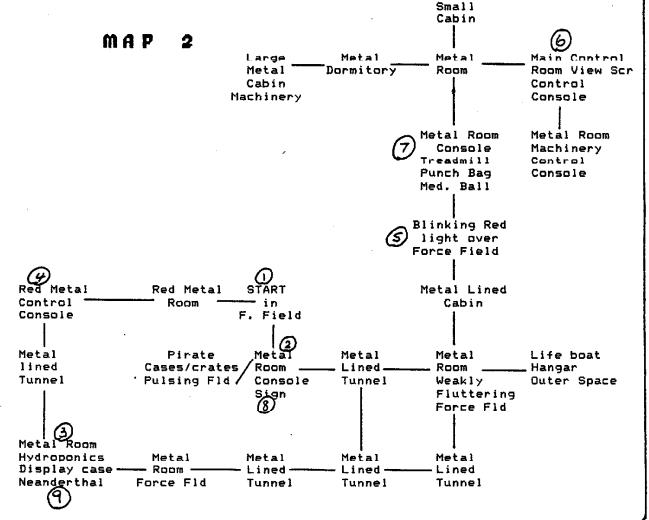
Adventurers° Corner

WITH "THE ADVENTURER"

rodney gainsford



SAVAGE ISLAND



to

b it Ming

s a the

ered

all

ith for ich

has

AUL

-..

our s a

iew to in uld

OUR COMMITTEE

There was a young man in the Hunter Who thought to become a great punter. Now this would, of course, depend on the horse, As all and sundry would know!

The young man, with great skill, (Without ever considering the bill!)
Thought to develop a plan.
"Now where do I start
To develop this art?" (asked he)

In a flash he saw where the answer would be, "Of course! Of course! Now I can see! The Hunter Valley 99ers for me!!"

Meek and mild, cap in hand, He came to join this merry band...

On arrival he did see A cherry nosed lot (all at sea!) First by the Com-it-tea he was met: "Stone the crows! Kill a chook! Everyone here looks like a crook!"

Alby Ando, tall and true Looking for an ear to chew

Behind his beard, so long and dark Edity Woods awaits your literary spark.

Long hours at night have all been fun, Killing bugs is Tony's run.

It came, it went, then came again, Libratious Al's beard is here to stay (Only the cook will have the say)

Looking round with great dismay "I just don't know how we'll pay. For this is something we can't afford!" Cries crafty Treasurer can'-a-tord.

Gentle Jim is in there too Carefully watching what they do...

Peter the Pom, God bless his heart, Runs the second hand 'puter mart.

Extended Basic is Jonseys dream Sometimes it makes him steam and steam.

In ironing out probs, McClure found his niche, A Clure for this and a Clure for that!! (Especially that fancy soldering act!)

Books to the left and books to the right Muncher Mulvaney strides thru the door Looking for a piece of the floor.

Now well may you ask
"Did the punter ever fulfill his task?"
With the answer to this I must be true Else he may hit me - with his shoe!
(Now that's a poser on which to chew!!!)

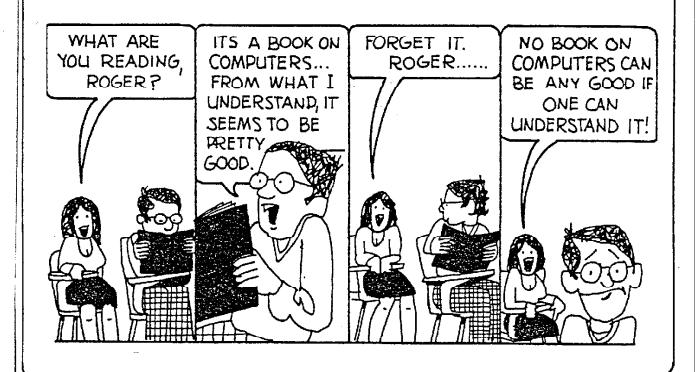
It is after all fair to mention
That the punter, his main intention
Was to become rich and famous And that was something he couldn't do,
At least, not with this motley old crew!

Off he wandered, dazed and shattered, His eyes staring stright ahead. Back and FORTH, ASSEMBLE and RUN, BASIC but EXTENDED, BREAK and RUN, and this bloody mob call all that FUN???

To the Doctor he finally went, (For he thought, "The whole lot are bent!") Into the surgery he ENTERed Where Richard the Fifth made him welcome.

This saga I now must end,
Before I too go round the bend.
But rest assured, my mental friend
Your torment here does not end THIS STORY HAS ONLY JUST BEGUN!!

The Rooster, Poet Laureate, HV 99ers



FORTH DATA BASE PROGRAM

BH

KEITH BRUCE

INTRODUCTION

This article is part two of two part discussion on a TI-FORTH program that enables data stored in a file to be accessed using an indexed list. The indexed list is an array of strings which are a subset of the data record, i.e. the key field. Together with the key field in the index list is the record number of the data record in the data file. Thus if the index list is sorted then the records in the data file can be accessed as thev were sorted. This overcomes the problem of sorting large data files with a machine that rather limited memory capacity.

Last months article gave an and at address 8324hex is the introduction into databases discussed the technique involved in index file access. Also presented If part of the index list contains was the first half of the program, the following: screens 240 to 263. Since that article I have transferred the source code for this program from the DS/DD disk to SS/SD disk. is so that it will be easy for me to it up with only one disk back club drive. Also it enables other members to get a copy of the code and be able to use on SS/SD only machines. Screens 24Ø to 263 correspond to screens 30 to 53 on S/SD disk I rearranged all and compiled (LOADed) only options I needed to support the made. this testing editing and ۵f program. The dictionary was BSAVE'd to screens 20 to 26 and together GET_DATRECORD will return with a little other TI-FORTH source record number of the record left with 6Ø code me that screens. Note are not a one to one correspondence, will be left in the buffer

The screen numbers presented article are 50 to 79. As stated above some screens have been edited from that presented last month. If the changes are essential to the understanding of the code presented this month then I have included them otherwise the amended code will be included on the disk in the club library.

of

upo

pre

pro

fie

out

Als

str wil

str

the

wit

mi:

RE

pr

pr

th

P&

nn

of

or

th

as

Er

SW

to ar

an

50

WO

29

i 🕊

Ha

SD

st

ŧa

th

ug

u

tř

11

u

B/

11

ti

u:

O

C

2

MAIN_WORDS

Following is a discussion on some of the words which important to the operation of the program.

FINDKEY (screen 52)

This word does a binary search of the index list to see if the string, whose address is on the stack, is in the index list. technique involved in a binary search was discussed in last months article. If the search does not find a match then it leaves -1 on the stack indicating that the key field was not found. Otherwise it leaves the element number of the matched item in the index list. the example below:

If the stack contains: 8324hex

and string: JONES, PETER

	Dat	a file
Element No.	<u>Contents rec</u>	ord no
1	AMOS, PAUL	4
2	BROWN, TOM	1
3	JONES, PETER	6
4	MORGAN, BILL	3
5	•	•
		_

the SS/SD disk. In converting to a Thus after calling FINDKEY in this the example it will return on the stack: source code for the TI-FORTH words 3 i.e. the element number in the the index list where the first match was

calling routine Мом the the the free data file. In the above example some this would return a 6. Now i f the rearrangement of the code was done data file is already open then by to add to the the word !" and that calling FREAD that record will be is why the DS/DD to SS/SD screens read from the data file. The record

Further processing on the contents upon what action is being done.

Note that no count byte prefixes most of the strings in this This is because all key program. fields in the index list are padded out with blanks (space character). Also the buffer containing string to be compared is padded out 36) moves the name of the file with blanks. strings is therefore performed on the file name length byte. the whole length of without any problems associated with are: mismatched string lengths.

REPORTS

Two basic report formats are provided which print to either the printer or the screen. Because these two devices have different page widths the report formats are not the same so as to take advantage of the extra page width on the printer. Both reports simply read the records in the data file in ascending index order.

Printer Report

The two standard TI-FORTH words SWCH and UNSWCH had to be rewritten to fit in with the different PAB area structure used in this program and also because of the absence of some of the TI-FORTH -FILE option F-D"). (i.e. assumes that the PAB for the printer is the first PAB in the PABs area. However I have already set aside space for two PABs starting at that stop. One for the data file and one for the index file.

Note that I have implemented the random access file records using normal TI disk files rather than PROGRAM MODIFICATIONS using the more traditional way of using FORTH screens. This is so that the information in the database is much more readily available for use with other languages (e.q. BASIC, XB and Assembler) on the TI.

PTRPAB Whenever the word executed 321 it (screen i s initialises the PAR pointers (PAB-ADDR, PAB-BUF and PAB-VBUF) for the printer PAB. Note that when using ALTOUT to redirect screen output to another device, in this case a printer, the PAB-VBUF must be exactly immediately one byte preceding the PAB.

As records are being read from of BUFR can now be done depending file, processed and then written to the printer the PAB pointers have to be reset each time before desired PAB, or file, is accessed. As previously stated F-D" was available so another word which did a similar function had be to MOV_PABNAME the created. (see name Thus any comparison of from a buffer into the PAB and sets the strings relevant buffers for the file names

Buffer	_Length_	Contents
INDNAME	27	Index file name
FNAME	22	Data file name
PTRNAME	6	Printer name

If the printer being used is other than PIO then the length of PTRNAME will have to be increased accordingly and the name changed on screen 34.

The reason F-D" was not was so that file names did not have to assigned before the program was compiled as is the case with F-D". Thus by using buffers and using MOV_PABNAME the file name can be entered by the operator when the program is running and the PAB set up accordingly. MOV_PABNAME assumes that the PAB pointers have been set Standard SWCH for the required file, e.g. PTRPAB, DATPAB or INDPAB have been executed, and the address of the file name string is on the stack. Note that a count byte precedes the filename string in these buffers. printer name could also be changed dynamically with the addition of extra code.

For the program to suit different record structure, screen layout or report format several applicable screens or words will have to be modified. These screens will now be discussed. Briefly the screens that are we are concerned with are 30, 43, 61, 63, 64, 65, 66, 75 and 76. However I will firstly discuss the record structure and what are the constraints on it.

RECORD_SIRUCIURE

Each record of the data file a one byte long buffer in VDP memory can be considered as a string of up to 255 characters depending upon the size of the constant BUFRLEN. The

ted Ιf the ted hem bе 1ub

his

ted

αn are the

rch the the The ary ths nat

on key it the See

his ck: the

NAS.

line the the ple the by bе

ord

FR.

buffer into which data file records stored for writing to and reading from file is called BUFR. Its length is set be BUFRLEN. The program is currently set up to find the key field at an offset of zero in BUFR i.e. at the start of BUFR. This can be altered if required and would be necessary if the program were to be modified to allow multiple index files with different key fields for each index list for the one database. A variable called KEY_OFFSET has been defined in the not fully but list, integrated into the program yet, which will allow a in BUFR at some other location location than the start of it.

Note the upper limit of 255 a record of the required length. require this would Note: that to the program to records. modifications overcome by by-passing the disk DSR names as required. and writing a routine to directly access the controller chip. However Screen 63: TYPE_REC that is one task this author will not be attempting in the foreseeable the the record on the screen. future.

Screen 3Ø:

This creates the variables and screen 61. Any changes to buffers required to store the record screen layout or record structure and index list (also see screen 39 have to be reflected in this word. for the creation of the array INDEX). Following is a list of the Screen 64: GETFIELDS variables and a brief description of added to the value ALLOTed because of the 2 bytes set aside by the defining word VARIABLE.

Variable	Current	Comment
Name	Size	CONSTANT - length
INDLEN	3Ø	of the key field.
BUFRLEN	255	Length of BUFR and length of record in data file. (see discusion on record

etructure).

Screen 43: GET_KEY

This displays a prompt on the command line (line 20 of screen) for the operator to enter the key field. It fills the key field part of BUFR with blanks and then gets a string of up to INDLEN characters from the operator and stores it in BUFR. Ιt leaves on the stack the number of characters entered.

Sc

CC

ke

SE

e)

8

k€

tI

r

n

W

f

BI

t

If the location of the prompts are altered then the screen location at which the key field is obtained will have to be altered.

Screen 61: SET_PROMPTS

This positions the field name key field prompts for each field on the Each field name prompt screen. starts on a new line and they are spaced two lines apart. This allows contents to be for the field characters is not imposed by the displayed on the line underneath program or by TI-FORTH but by the each field name. This layout is not disk drive DSR. If you require more fixed and can be altered in position than 255 characters this limitation on the screen and also in order if could be overcome by using more than required. The name of the data file one random access record to build up is also displayed at the top of the screen together with the action e.g. modifying or deleting adding, To modify for another achieve it. This limit can also be record structure amend the field

This displays the contents of Ιt assumes that the the field names are already displayed on the screen and the record is in BUFR. The layout has already been described above in

the field These words get each. Note that 2 bytes must be contents from the operator for each field except the key field screen 43: GET_KEY). Each stores the string in BUFR at appropriate offset. GET# returns the number of characters entered and together with the offset into BUFR and the field length CHEK_CR fills blanks if the field with character count was zero. This is done so that no stray characters are left in the field such as a carriage Again if the screen layout return. or the field lengths are altered elsewhere then they have to be reflected in these words.

Screen 65: MODIT and REPOS

or d.

FR

ng

he

Ιt

of

ts ็ดก

ed

me

he pt

re

WS

on

ng

11 d

Ιt

re nd

ut

in

14

:h ee :s

he

hs

hd

FR

he

įs

re þе

ūŧ

e d

command line and then scans for a plans to adapt the code to allow the key to be pressed. It assumes that SET_PROMPTS and TYPE_REC have been executed and allows the operator to edit any of the fields except the key field.

If the key pressed is <ENTER> then the loop is exited and the sorted in more than just one way. record written back to file. If a key is pressed that is field can be edited. show the blanked out field. The new file and then sorting the field contents are then obtained fields. is then left at the end of the include it in a later version. command line prompt by REPOS ready for another field to be edited.

If the number of fields in the record is changed then the number of case options will have to be also altered.

Screen 66: GET_FILEDS

database the operator prompted to fill in all the fields, FORTH code is to write. This word simply calls each of individual get field contents words. Thus if the number or order of the KEITH BRUCE fields is changed then it has to be 14 PASADENA CRES. reflected here.

Screen 75 and 76: SCRN_REPORT PRNTREC and

These routines print out the contents of the record in a formatted manner on the screen or to the printer. Primarily if the record structure is altered then it should be reflected here. Otherwise the layout or formatting of the record printing can be changed in any way. Fields can be printed in any order or location and can even descending rather than ascending.

This concludes the section on modifying the program to another data structures' requirements. It can be seen that the program presented here is far from user friendly in this regard.

However this is my first attempt and MODIT displays a prompt on the it does actually work. I do have operator to define or specify the data structure without having to modify the source code (program) and recompile it to do it. Also I have plans to allow the program to support additional indexes. This would allow data to be presented or

This brings me to a point which within range then the corresponding I mentioned in last months article. The field in I said that if the index file was BUFR is first filled with blanks corrupted or deleted then it could then the record is redisplayed to be recreated by reading the data Well I have not included from the operator and stored in BUFR any code in the program at present at the correct offset. The cursor to do this, however I do plan to

Well enough TI-WRITEing for the moment, I'd better start writing code to include some of these enhancements to have something ready for next month. I don't expect anybody to type all 50 screens of this program in and so I have made available to the librarian a copy of this program for other users to have When a new record is added to a look at and to use if they wish. is Perhaps they might even see how easy

> PH., 547258 CARDIFF 2285



Due to a lack of space, only be left out. Record separators can 16 screens are included in this be inserted and the order of edition of the Newsletter. The printing of records could even be final 16 screens will appear in the June issue of the mag.

```
DATABASE
                                    29Feb87
     1234
       : MID
                                     \ ( -- (LO+HI)/2 )
            LO 3 HI 3 + 2 / i
            EY= \ ( adl adZ -- #lag ) true if string adl=adZ INDLEN 1- 0 DO DUP C0 ROT = IF 1+ SWAP 1+ ELSE DROP 0 LEAVE ENDIF
       : KEY=
    6789
            LOOP
   11
12
13
            SWAP DROP ;
       : L>H
                                    -\ ( -- flag ) true if LO > HI
   14 LO 9 HI 9 > ;
 SCR #51
    O \ DATABASE
                                     28Feb87
      : ?BRCH_FIN
OVER OVER INDEX
KEY=_DUP_MINI !
    234
                                        \ ( ad n1 -- flag ) true if key at addr is
\ the same as element n1 of index OR if
\ LO > HI
               L>H OR ;
    5673
   901Q555
     452
\ DATABASE
SCR
                                   28Feb87
                         ( ADDR -- N ) N=-1 => failed else N=key location
\ looks for Key in KList; returns location in KList
\ if found otherwise returns -1
!! C \ does it using a binary search
  20456769
     : FINDKEY
        HAXREC 9 HI ! C
BEGIN DROP
HID OVER
CVER INDEX $:
                               $<
             IF DUP 1 - HI ! ENDIF
              PSŘČH_FIN
 10
          UNTIL
SMAP DROP
MINI @ O= IF DROP -1 ENDIF
 11
12
13
 14
 15
SCR #53
0 \ DATABASE
                                   1Mar 27
   1234
     O VARIABLE MAXI
                                   : GET_DATRECNO
            INDEX +
   98
            Э
            1
 10
 121314
```

SCR #50

```
SCR #54
        \ DATABASE
                                              28Feb87
                                                                  \ ( ad -- flag )
\ see if key exists. return key loc
       : GETRECORD
             FINDKEY
              -DUP O
                             GET_DATRECNO
FREAD____
                 IF
                                                                  \ get rec no in dat file
\ read it from file
   6789
                 ELSE
                             RECNOTEND
                                                                  \ record not found
                 ENDIF
  11
12
13
SCR
       #55
      \ DATABASE

• PUTRECORD

FINDKEY
   01234
                                              28Feb37
                                                    \ (ad -- flag ) ad=addr of key in BUFR \ see if key exists. return key loc
             IF DROP MAXREC @ INDMAX < \ key does not exist in index
IF 1 MAXREC +! \ inc no of records
MAXREC @ DUP \ det new total no of records
FWRITE DROP \ write in new record
TABLEY NUD \ det addr of 'MAXREC' in index
   6789
                 FWRITE DROP | write in new records
FWRITE DROP | write in new record
INDEX DUP | get addr of 'MAXREC' in index
BUFR KEY OFFSET 3 + point to key in BUFR
SWAP INDESN CMOVE | move key into index
INDLEN + MAXREC 3 SWAP ! O | put rec no in index
MAXREC 3 1 > IF SCRTINDEX ENDIF
ELSE NO ROOM ENDIF
ELSE GET_DATRECNO FWRITE ENDIF;
 SCR #56
O \ DATABASE
                                              1Mar S7
                 : ADDRECORD
                 Ēİ SE
                               PUTRECORD
                 ENDIF
     3
   10 : DINDEX MAXREC @ 1+ 1 DO
11 CR I . 2 SPACES I GET DATRECNO . 2 SPACES
12 I INDEX 20 TYPE LOOP;
   11213
   15
 SCR
        #57
        \ DATABASE
                                             1Mar87
       : DEL_PT1
1 MAXI !
1 GET DATRECNO
MAXREC 1+ 2 DO
     5
     5789
                     DUP
                     T SET DATRECHD
SWAP 3
IF I MAXI !
DREFT DATE
   10
                             I GET_DATRECNO
                     ENDIF
                 LOOP
        --> <sup>i</sup>
```

```
O A DATABASE
                                     1Mar87
     : DEL PT2
MAXI a
             TTEM 0
             = ()=
             IF FREAD DROP
ITEM @ GET_DATRECHO
DUP_FWRITE
DROP
             MAXI @ INDEX INDLEM + !
  Î Î
12
13
            MAXREC @ ITEM @ DO
I 1+ INDEX
I INDEX INDREC
                                   INDRECL CMOVE
SCR #59
O \ DATABASE
                                       1Mar87
      : DELETE_REC
                                       \ ( ad -- flag ) ad=addr of 'key' in BUFR
             FINDREY
             DUP
                   ITEM !
             IF RECNOTEND
             ELSE
                  DEL_PT1
DEL_PT2
   8
                   -1 MAXREC +!
                                                 \ adjust total number of records
 ĪĪ
                                                 \ no error flag
 12
13
14
             ENDIF
 15 --->
SCR #60
   O N DATABASE
1 : PRESSZCONT
                             2Mar87
             CR

" Press any key to continue..."

KEY DROP ;
     : BLNK BUFR
BUFR INDLEN +
ROT +
SWAP BLANKS ;
                                                        \ ( offset #blanks -- ) fill bufr
\ with #blanks starting at offset
\ add offset
                                                        \ put n blanks into BUFR
 10
11
12
     : CHEK_CR
ROT O=
IF BLNK_BUFR
ELSE_DROP DROP
                                                       \ ( n offset #blanks -- )
\ If n=0 put #blanks into BUFR
\ fill bufr with blanks
\ do nothing
 13
 14
            ENDIF ; -->
SCR #61
      " DATABASE
                                      2Mar87
         SET_PROMPTS
CLS
O O AT
                                                               -- ) clears screen and puts
                                                        \ field names on screen
           O O AT
FNAME COUNT TYPE \ display file name
O 5 AT ." 1:Surname, first (k):"
O 5 AT ." 2:Fhone no.:"
O 7 AT ." 3:Address:"
O 9 AT ." 4:City/Suburb:"
O 11 AT ." 5:State:"
O 13 AT ." 6:Comments:"
   89
 10
11
12
13
```

CCR #58

```
SCR #62
0 \ DATABASE
                                                       2Mar87
         11
12
13
14
        #63
\ DATABASE
SCR
                                                       2Mar 87
        : TYPE REC
BUFR DUP
0 4 AT INDLEN TYPE
INDLEN +
0 6 AT DUP 9 TYPE 9 +
0 8 AT DUP 20 TYPE 20 +
0 10 AT DUP 25 TYPE 25 +
0 12 AT DUP 5 TYPE 5 +
0 14 AT 40 TYPE
                                                                                \ ( -- ) Displays contents of BUFR
\ on screen. Assumes field names
\ are on screen.
   13
           #64
DATABASE 2Marb/
GETFIELD2 BUFR INDLEN +
O 6 AT 9 GET$ 0 9 CHEK_CR ;
  SOR
           GETFIELDS DUFR INDLEN + 9 + 0 8 AT 20 GET$ 9 20 CHEK_CR;
           : GETFIELD4 BUFR INDLEN + 29 + 0 10 AT 25 GET$ 29 25 CHEK_CR;
     10 : GETFIELDS BUFR INDLEN + 54 + 11 0 12 AT 5 GET* 54 5 CHEK_CR ;
     11
12
13
            : GETFIELDS BUFR INDLEM + 59 + 0 14 AT 40 GET$ 59 40 CHEK_CR;
     14
   \ reposition cursor
        234
           ** MODIT
O 20 AT ." Enter number of field to modify"
BEGIN
KEY CASE 13 OF 1 ENDOF
50 OF O 9 BLNK BUFR TYPE REC GETFIELD2 REPOS ENDOF
51 OF 9 20 BLNK BUFR TYPE REC GETFIELD3 REPOS ENDOF
52 OF 29 25 BLNK BUFR TYPE REC GETFIELD4 REPOS ENDOF
53 OF 54 5 BLNK BUFR TYPE REC GETFIELD5 REPOS ENDOF
54 OF 57 40 BLNK BUFR TYPE REC GETFIELD5 REPOS ENDOF
        567.89
      10
     11
12
13
14
                   UNTIL SUFF PUTRECORD DROP ; -->
```

THE INFORMATION PAGE

IN YOUR NEWSLETTER THIS MONTH

Feed-Back on Chain Mail Newsletters

Random Bytes

A TI-Writer Tip on A4 page length

Handling CMOS IC's

Super Improved 32K E/A Module project

RS232 Card Repair Tip

Assembly Language - Sub Routines/Index Addr

Graphics and TI-Writer

TI-Writer Tip

BASIC for Beginners

Improve your Speech
Struggling FORTH - VDP RAM - part 2

Cassette Data Base - a program to type in

Söftware Review

Adventurers Corner - map of Savage Is. part 2

Oùr Committee - Humour in poetry

FORTH Data Base - Part 2 of a program to type in K.

R. Klienschafer

B. Carmany

Sydney Users Group

R. Klienschafer

A. Member

A. Anderson

A. Wright

Melbourne Users Group Ottawa Users Group

P. Mulvaney

A. Wright

R. Terry

J. Smart

A. Franks

R. Gainsford

P. Laureate

K. Bruce

PLUS MUCH MUCH MORE!!!!!

COMING EVENTS

Mid Year Night Out: Friday 22 May (7.00pm)
Next Committee Meeting: Tuesday 2nd June
Annual General Meeting: Tuesday 9th June

AGENDA FOR JUNE MEETING

Annual General Meeting - BE THERE!!

CLASSES AVAILABLE FOR MEMBERS

BASIC group conducted by Paul Mulvaney at the Warners Bay High on Tuesday 19th & 26th May.

ASSEMBLY group conducted by Joe Wright. Check dates and venue with Joe if your interested in attending.

FORTH group conducted by Richard Terry each Wednesday night. Check with Richard each Wednesday re venue.

ANNUAL SUBSCRIPTIONS

Subscriptions to the Group cover the period 1 July to 30 June following year. Membership enquiries are welcome; please address all enquiries to the Secretary.

The annual subscription is: Australian Residents...\$20

Overseas Residents....\$40 (airmail)

\$30 (surface)

Back issues of our Newsletter are available for \$1 plus postage

i

Ş