



TT 99/4A

# HOME COMPUTER NEWSLETTER



### DISCLAIMER

The HUPP NEWS the n44irial the HUNTER VALLEY newsletter of NINETY NINE USER GROUP. Whilst every effort is made to correctness and accuracy information contained therin, be it of general, technical, or programming nature, no responsibility can be accepted by HV99 NEWS as a result applying such information.

TEXAS INSTRUMENTS trademarks, names and logos are all copyright to TEXAS INSTRUMENTS.

HV99 is a non profit group of TI99/4A computer users, not affiliated in any way with TEXAS INSTRUMENTS.

### CONTRIBUTIONS

Members and non members are invited to contribute articles for publication in HV99 NEWS.

Any copy intended for publication may be typed, hand written, or submited on tape/disc media as files suitable for use with TI Writer (ie. DIS/FIX RØ or DIS/VAR RØ). A suitable Public Domain word processor program will be supplied if required by the club librarian Al Lawrence.

Please include along with your article sufficient information to enable the file to be read by the EDITOR eq. File Name etc.

The preferred format is 35 columns and page length 66 lines, right justified.

All articles printed in HV99 NEWS (unless notified otherwise) are considered to be PUBLIC DOMAIN. Other user groups wishing to reproduce material from HV99 NEWS may feel free to do so as long as the source and author are recognised.

Articles for publication can be submitted to.

THE EDITOR HV99 NEWS 15 GAYTON CLOSE WARNERS BAY 2282 NEWCASTLE

General address for ALL other club related correspondence.

THE SECRETARY HV99 USER GROUP 25 RESERVE RD. WANGI 2267 NEWCASTLE

### YOUR COMMITTEE 1986

н.	MRIGHT	PRES.	PH. 468120
₽.	COXON	SECT.	PH. 751830
в.	RUTHERFORD	TRES.	PH. 488184
п.	LAMRENCE	LIBR.	PH. 488509
5.	TABLOR	EDIT.	PH. 487076
в.	WOODS	EDAS.	PH. 662307
T.	MC. GOVERN	TECH.	PH. 523162
в.	MAC. CLURE	MODL.	PH. 437431
G.	BONES	TECH.	PH. 573744
D.	WINTON		PH. 591883

# SECRETARY'S REPORT

\*\*

If

**es** 

as

Pr

ca

We

th

dr

th

in

re

Wi

th

Wi

CO

Th

a

fe

st

ca

ta

su

ev

Au

(E

th

re

it

ad

re

R.

Gr

Th

St

10

"HAPPY NEW YEAR" and welcome back. January was a very quiet time, but lots of people are ready to go now that February is here! especially your Committee.

We hope to bring you lots more with basic projects, together language course, examining Extended Basic and of course the Magazine and We ask that you demonstrations. contribute to this fine publication so that we can keep the standard high. If you have any news, views, articles games etc., interest your fellow user's, dont keep them to yourselves, share with your T.I. friends.

Don't forget the monthly meetings....The following dates are just to refresh your memories:-

COMMITTEE MEETING - First Tuesday of the month, 7.pm.

GENERAL MEETING - Second Tuesday of the month, 7.pm.

BASIC CLASSES - Most Tuesday's (check with Garry Jones)

ASSEMBLY LANGUAGE - See Joe Wright.

Don't forget our Newsletter/Book service for all Members. You may find some handy hints that are sure to help you with your programs.

HOPE TO SEE YOU ALL AT THE MONTHLY MEETING.

REGARDS, Peter C. \*\*\*\*\*\*\*



#### \*\* CASSETTE MODIFICATION \*\*

your cassette motor seems reluctant to start smartly, especialy when recording files such Rutherfords Mini Brian Processor then read on. Some cassette motors well when controlled by the console, this is due partly to some voltage |On a more positive note the drop from the forward resistance of also the switching transistor, additional interface lead, plug and socket as resistance and possibly age.

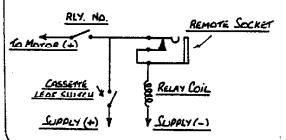
With a simple modification as per the circuit diagram your cassette will jump to attention whenever your console instructs it to go !!.

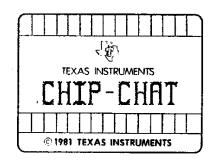
The alterations are simple requiring a two dollar little DICK relay and a few wiring changes, the relay can be stuck into any suitable space in the cassette with double sided adhesive tape (whatever would we do without such stuff?) one can be fitted in even crowded recorders such as the Audiosonic and its look alikes (Ehh!).

Instead of the computer controlling the motor it merely switches the relay and the cassette motor gets its full share of voltage and as an added bonus you can throw away that reversing plug.

#### R. Kleinschafer HV99 Grawin via Walgett

Thanks a lot for the tip Ron Steve





<del>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</del> THE ORPHAN CHRONICALS "The Orphan 185 page book Chronicals" was written bу Albright and is marketed by Millers Graphics.

The book traces the events leading upto the eventual scrapping of the TI 99/4A by Texas Instruments, and Word | features interviews with people who were directly or indirectly involved don't behave very such as Don Bynum (TI) and Charles LaFara (IUG).

features listings of user groups and bulletin boards as w ≥ 1 1 distributers of TI related products.

The book is now ready for shipment and can be ordered through Millers Graphics for \$US 9.95 plus approx \$US 5.00 for shipping to Australia. 04 interest particular is the authors purpose in writing the book which as he says he first envisioned as a project for a non-programer contribute something to the community, "I really mean it when Ī say that the book was a labor of love for the TI community. Being less than adequate programmer, felt like a user and not contributer when I ran all the great new software. The book is my way to try to give something back to the best bunch of computer junkies in the world - the 99/4A users!", The book is highly recommended and congratulations are in order to both Don Albright and Millers Graphics.

<del>\*</del> T199/4A FLIGHT SIMIL ATOR <del>\*\*\*\*\*\*\*\*\*\*\*\*</del> After years of waiting it appears that there may finally be a decent Flight Simulator released for our machine. The programme is written in machine code and is evidently with comparable the Microsoft version. I hope to have further details available for the next issue.

<del>\*</del> NEW COMPUTER DUE SOON At the November Chicago TI-FAIRE Lou Phillips of Myarc Stated that "Myarc is committed to producing the new machine and that they have always lived up to their commitments in the past".

The new computer which is yet to be named will feature the TMS9995 chip, 256K main memory, 64K VDP and 64K of ROM.

Ιt will bе approximately 95% compatible with existing TI software (will not support speech) and will allow you to use your existing Peripheral Expansion Box and all cards except the 32K card. It will also be compatible with the TI, CorComp and Myarc Disk Controllers. As an added bonus it will be supplied with the new Myarc Extended Basic and feature 2 Text modes and 8 Graphics modes.

If all goes well the new computer should be released in the US as you you are reading this and sell for around \$US 500.00

THORN-EMI SOFTWARE <del>\*\*\*\*\*\*\*\*\*\*\*\*\*</del> It now appears that software developed ЪУ Thorn-EMI and consequently shelved due to the scrapping of the TI99/4A may finally \* be released. Howard Greenberg of \* long period of negotiations with the Thorn-EMI management he may soon distribution rights to the previously unreleased software.

Pounds to anywhere in the world. The games are Submarine Commander, River Rescue, and Computer War and according to Howard are of the very highest quality.

Correct me if I'm wrong but Ι thought that Thorn-EMI also had a programme titled "747 PILOT" slated for release for the TI99/4A, now there's a program I personally would interested in. <sub>g</sub>pe

<del>\*\*\*\*</del> \* FREEWARE V FAIRWARE ¥ In the January issue of SHOALS TIdings John Taylor (JET)

the Newsletter Editor has some timely Now Someone Else has gone, and we comments on the "FREEWARE" concept. He states that during the several months Freeware has

under close scrutiny in more WAVE than one.

It turns out that the actual name "FREEWARE" i s the registered trademark of Headlands Press of Tiburon California. A registered trademark held by any firm individuals may not be used without their consent. For this and other reasons, a move is now afoot to come up with a new name to represent the "Try Before You Buy" type of software we have come to known as freeware. Until a standard new name has been decided upon, I will use the term FAIRWARE to describe the software. Fairware, because author has been fair to you by letting you try his program, and because you in turn have been fair to the author by rewarding him for his efforts.

I'm sure we all agree with John on this point, for too long people have been taking the FREE in FREEWARE far too literally, now it's time for us all to become a bit FAIR-DINKUM and give credit where credit is due, if you have a FAIRWARE product that you really get a lot of use out of and havn't sent the author any form of recognition give it a bit αf thought, put yourself in position, would you bother producing any more programmes ??.

"SOMEONE ELSE" the TI\*MES Group writes that after a \* We are saddened to learn recently of the death of one of our most valued members, Someone Else. Someone's passing created a vacancy that will His intention is to have all three be difficult to fill. Else had been games on one disk selling for 35 with us almost since our club was formed, and during that time, Someone did far more than a normal person's share of the Whenever leadership was mentioned, this wonderful person was looked to for inspiration as well as results. It was often said, "Someone Else can work with that group or committee". Whenever there was a job to be done or a meeting to attend, one name was everybody's list.... on Someone Else do it.". Someone Else was a wonderful person,

sometimes appearing superhuman. But a person can only do so much. Were the truth be known, everybody expected too much from Someone Else. wonder what we are going to do. past Someone Else left a wonderful come example to follow, but who is going

F1

F2 F3

> F4 F.5

F8

F9 F10

The cha 0n thi Whe

YOU fir key IBM

the IBM Rig The

on TED \*\*\*

¥ \*\*\* The LOG

con be by 500

HOU

to follow it? Who is going to do the things that Someone Else always did?

C. Raybern Portland User's Group

d

d

t

### \***\*** MULTIPLAN

<del>\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</del> If you have Multiplan and you wish to use books written for the IBM computer the following information may be of some help to you.

Multiplan books written for the IBM computer use call- outs for the F1, F2 etc., keys, as well as a few words that are not used on the TI. Here is a list of IBM keys and their equivalents fot the TI.

#### IBM ΤI MEANING

= CTRL-6 Next window

- CTRL-3 Next unlocked cells

= CTRL-7 Absolute reference

= CTRL-8 Recalculate

F5 and F6 have no reference

= FCTN-5 Edit word left

= FCTN-5 Edit word right

= FCTN-4 Edit character left

F10 = FCTN-4 Edit character right

The IBM uses REPEAT to repeat a character, such as the dashed lines. this.

When they call for TAB on the IBM, you can use CTRL-2 or CTRL-A. Ι find CTRL-A is handier because the keys are closer together.

IBM CTRL-Pg UP is CTRL-1 or HOME on the TI.

IBM END is CTRL-2 or FCTN-1 Right) on the TI.

The IBM ALT H is simply H for HELP on the TI.

BUG NEWS (California) TED WHOMSLEY.

### \*\*\*<del>\*\*\*\*\*\*\*\*\*\*\*\*</del> TI LOGO II

\*\*<del>\*\*\*\*\*\*\*\*\*\*\*\*</del> The latest TEX COMP price for TI LOGO II is \$US 19.95!!. Anyone contemplating purchasing LOGO would be foolish to let this bargain slip Assuming delivery ьу seamail the landed cost in Australia much would be approx \$A 50.00!

## ATTENTION DISK USERS

This article, by CHARLIE CHAN, originally appeared in the December issue of CHUG-A-LUG, the magazine of the Canberra TI Users Group.

I came across an interesting letter in the June 1983 edition σf Popular Computing. It reads: Му has a single sided disk computer I decided to try to use the drive. other side of a floppy disk. punched another notch and another sector hole in the envelope at the places and appropriate my disk capacity was doubled. To gain access to the second side, I flipped the disk over. Since I started flipping disks, have Ι not noticed any increase in disk errors or disk failures. Why don't manufacturers adventise this capability?

Single sided disks are certified only on one side. The other side either has never been tested or has been tested and found defective. Either way, you use the second at your own risk.

In a single sided disk drive, a pressure pad rests against the unused side of the floppy disk. Often bits of debris become caught between pad and the disk surface, producing concentric scratches. The scratches On the TI you must use COPY to do don't affect the use of the good side of the disk - unless of course, the disk. flip Then two things happen: you begin trying to use possibly scratched disk surface, and your original (good) disk surface is exposed Inow to the potentially damaging pressure pad.

> Damage to the disk envelope another problem. By punching a notch in the envelope, you void the manufacturers warranty on the disk media. Disk envelopes are made polyvinyl chloride, a material that shatters. So even though your cut appears clean, cutting the envelope actually produces many tiny particles that can scratch the surface.

> Sure, you can flip a floppy disk if you're careful. But don't put confidence in the disk afterwards.

out of the construction sequence the bit tricky so be patient object of the whole exersize and indeliberate with it. OOPSES that I have seen for a while. and the legs will slide That is to have this extra memory socket smoothly and without bent or after all chips into the expansion board kept to a minimum. sockets, so could you please insert the following after Step 16 and before Step 17 :-

16A - This step involves the actual insertion of the 4 HM6264 - LP15 memory chips into the 28 pin sockets in which they live. These devices are low power CMOS devices and all actually installed it might work a that we have been stressing about bit better - MY APOLOGIES. anti-static protection really counts now - so earth yourself properly and check it. Even with this protection it is important to try and not actually touch the IC pins during this process.

If you hav'nt inserted chips into sockets before it may be an idea to get someone that has to do it for you, as this is'nt really the type of chip to learn on.

As the chips are all the same

index finger. The legs should be all S L I MEPIT

aligned and if they are not they will have to be, so use a straight edge to align all legs together. Lay the legs of one side onto the top of it's respective socket and slightly engage the legs just enough to hold it in position - don't insert it completely yet. Now with a straight edge (I use the shaft of a small screwdriver) In last editions 32k Memory Expansion pull the other side legs toward the project article I managed to leave socket until they engage. This is a As they engage doing so created one of the better you can apply some downward pressure this work one must left out legs. As well as this the physically insert the FOUR memory fingers touching the chips has been

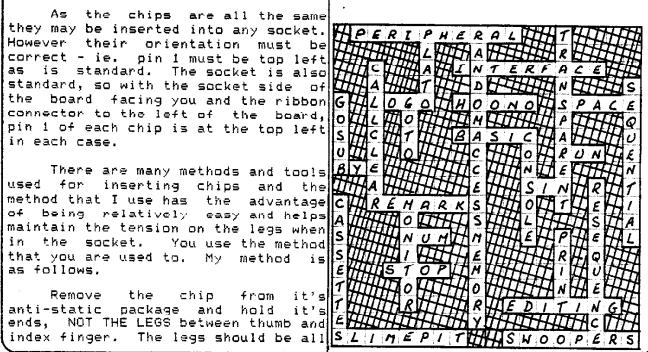
> Carry this out for the 3 remaining chips and theck the legs once more.

> > Proceed with step 17.

MoM with the memory chips

> Albert Anderson (4A - 4EVER)

### CROSSWORD PUZZLE SOLUTION



CAL ori

sub fro CAL

whi cre ve1 col

> num 25

to

CAL The reupp in



to

he

' s

i n

he

а

re

hø

he

en

3 95

ps.

а



# SPRITES

PART 2



BY BRIAN WOODS, H.V. SSERS

Last issue we saw how to create a Sprite, and now we move on to how to alter any of its characteristics.

### CALL MOTION

The speed of a Sprite is originally set in the CALL SPRITE subprogram and can only be changed from its original value by using the CALL MOTION subprogram.

A typical statement would be:CALL MOTION(#2,80,-50)
which would alter a previously
created Sprite #2 to change row
velocity to 80 (down the screen) and
column velocity to -50 (move right
to left)

Again, as in creating a Sprite, numeric variables etc, can be used as parameters:-

CALL MOTION (#B, C, C+ZO)

#### CALL LOCATE

The LOCATE subprogram may be used to re-locate Sprites, and puts the upper left hand corner of the Sprite in the location specified in the

statement:-

CALL LOCATE(#8,96,128) would change the position of Sprite #8 to about the centre of the screen.

CALL COLOR

Originally the color of the Sprite is set in the CALL SPRITE statement, but it may be changed at any time later by using the CALL COLOR subprogram. For example:-

CALL COLOR(#5,16)

sets Sprite #5 to have a. foregroung color of 16 (white). The background color is always transparent. You can define more than 1 Sprite color at a time in one statement:-

CALL COLOR(#1,5,#2,14,#3,16).

CALL DELSPRITE

The DELSPRITE sub-program removes Sprites from a program. Once it has been deleted, it can only be re-created by using another CALL SPRITE statement. You may remove 1, some, or all of the Sprites in one statement:

CALL DELSPRITE(#1) deletes
Sprite #1

CALL DELSPRITE(#2,#5) deletes Sprites 2 and 5

CALL DELSPRITE(ALL) deletes all existing Sprites

CALL DELSPRITE(#C) deletes a Sprite with a numeric variable = C

CALL PATTERN

The CALL PATTERN statement can be used to change the appearance of a Sprite without effecting any other characteristic. The CALL PATTERN sub-program appears as:-

CALL PATTERN(#1,65)

This statement would alter the shape of Sprite #1 from its present shape to ASCII 65.

The following example demonstrates how it may be used:-

100 CALL CLEAR:: CALL MAGNIFY(2)

110 CALL CHAR (65, "7E81A58181BD817E"

,66, "7E818181A599817E")

120 CALL SPRITE(#1,65,5,99,1,0,9)

130 CALL PATTERN(#1,45)

140 FOR A=1 TO 300::NEXT A

150 CALL PATTERN(#1,66)

160 FOR A=1 TO 30::NEXT A 170 GOTO 130

•

#### CALL MAGNIFY

The CALL MAGNIFY statement has only 1 parameter and it determines the size of ALL Sprites used. The parameter used must be an integer

from 1 to 4. If no CALL MAGNIFY is specified in a program the default is 1, ie all Sprites will be normal character size and use just 1 character position on the screen.

CALL MAGNIFY(2) means that each Sprite is defined only as 1 character, but takes up 4 character positions on the screen.

CALL MAGNIFY(3) causes all Sprites to take up 4 character positions (the same as CALL MAGNIFY(2)), but each Sprite is made up of 4 individual characters. defining a Sprite with magnification of 3 (or 4) the character number ASCII value) must be evenly divisible by 4 (eg. 80,84,100,120) because each Sprite uses the ASCII number as specified in the CALL SPRITE statement and the following also. For example, using ASCII 92 when creating the Sprite and using a magnification of 3 (or 4) causes the Sprite to consist of characters 92,93,94,95.

CALL MAGNIFY(4) Sprites are created as in CALL MAGNIFY(3) except that the final character is twice the size of 3X ie the Sprite takes 16 character positions on screen.

The example of a MAGNIFY subprogram in the Extended Basic manual (pi20) shows how the sub-program may be used.

#### CALL DISTANCE

This sub-program may be used to either find the distance between 1 Sprite and another, or the distance between a Sprite and a specific screen location. (The position of a Sprite is deemed to be its upper left hand corner).

There are 2 formats for this statement:-

- 1) CALL DISTANCE(#1,#2,A) to determine the distance between 2 Sprites.
- 2) CALL DISTANCE(#1,80,50,A) to determine the distance of a Sprite (#1) from a specific screen location (80,50), and give the value of that distance to a numeric variable, (A).

The value returned to the variable is the square of the distance between the 2 Sprites or the Sprite and the screen location. If the value computed is greater than 32767 then 32767 is returned.

#### CALL COINC

The COINC sub-program is used to detect a coincidence between:
1) 2 Sprites (#1,#2,TOLERANCE, VARIABLE)

2) 1 SPRITE AND A SCREEN LOCATION (#1,ROW PIXEL,COLUMN PIXEL,TOLERANCE, VARIABLE)

3) All Sprites (ALL, VARIABLE)

A tolerance (in pixels) coincidence is allowed which the possibility enhances detecting a coincidence. If the 2 Sprites or 1 Sprite and location come within the specified limit, then а coincidence and a value of detected assigned to the numeric variable. If no coincidence is detected the variable is assigned the value of O (zero).

Care must be taken when writing a program containing CALL COINC. If the Sprites are moving quickly this sub-program may miss a coincidence. This sub-program only checks for a coincidence when actually call, so it is possible that while a program is executing another statement, a coincidence may be missed.

This then is a brief explanation of Sprites and the sub-programs that make them work. The best way to learn more is, like all other facets of programming, experiment for yourself.

The program on page 175 of the TI Extended Basic manual gives a good example of all aspects of Sprites mentioned in these articles.

Sprites can be a little more difficult to learn than some other aspects of programming, but once the basics are mastered, you will find 1001 ways to use them in your programs.



by ill coat a that the sin

box

**typ** 

99/

D

men Ran Like Like RAM LOFF COM Men dat

rea

can

 $\subset OM$ 

to

mac

the

008

99/ mem Rea TI mem Ins dif is

box

the

eas

COM

# GRAM KRACKER REVIEW

BH.

DANNY MICHAELS. SHOALS 99-ERS

The GRAM KRACKER was designed Warren and marketed by Millers Graphics. The little box (approx the same size as a widget) contains 56K (expandable to 80K) battery backed programmable memory and plugs into the module port of the computer. On the surface this sound too exciting, but believe me, it is. What makes it so great is the type of memory that's Before we get into the let's look at the different|computer 99/4A.

The most familiar type σf memory to most computer users is Random Access Memory, or RAM. type of memory stores temporary data offs. like BASIC and ASSEMBLY programs and language RAM can easily be changed, and is of memory, or the patience to lost when you turn your computer The other common type computer memory is ROM, or Read Only Fortunately, Memory. This memory retains its language called assembly language to data even when power is removed, but make programming easier. Assembly can be stored in ROM. computer to store the operating system of the another program, called assembler, machine. the program that oversees the entire into machine language. operation of the computer.

Read Only Memory. GROM is unique to it still memory produced only ЬУ Instruments. What makes different from other types of memory personal then memory addressing becomes much computers. easier to understand. Each time the languages were developed computer retrieves or stores a value same purpose, to make it simpler

ROM or in RAM it must memory the address of the requested Addressing GROM memory different in that it only has to told the starting address for requested block of memory. When value read i s from GROM, it automatically increments itself the next address in sequence. computer is configured so that data from GROM is read through one normal CPU address. It's sort like having a post office box that contains another set of post office boxes. This allows the computer to actually have more memory than CPU can directly address. The only drawback is that since the CPU does not directly address the memory in GROM, it cannot directly execute programs stored there. I know this is supposed to be a review, but lets spend a little more time in tutorial mode and take a quick sook computer languages.

The only language the CPU in a really understands i 5 types of memory contained in the binary code, most commonly known as machine language. It is comprised only of binary values representing on and off states. Each instruction that the CPU understands is made up This of a combination of these On and To program in machine requires that the the value of variables. The data in programmer himself have a great deal up the code for each instruction he of wants the computer to somone developed as the name implies, you can only language uses mnemonics, or easy to read from it. No variable type data remember names, for all the CPU In most instructions. Once a program source systems ROM memory is used code is written by the programmer The operating system is is run which converts the mnemonics The machine language code produced by assembler can then be loaded In addition to RAM and ROM, the the computers memory and executed. 99/4A contains another type of Although assembly language is much memory known as GROM, or Graphics easier to use than machine language, requires an TI computers, as it is a type of knowledge of the inner workings of Texas the computer. Somewhere along the GROM trail of development of modern day computers many is the way it is addressed. If you languages have been developed, some think of memory as a set of storage for specific areas of interest. boxes such as post office boxes, others for more general use of However, all these for the

program the desired task. The most familiar of languages is Beginners All-purpose Symbolic to learn, and requires very little, if any, knowledge of the computers (or programming in Basic is speed. Since the instructions you use in limitation of the GPL interpreter, Basic cannot be directly executed by but a hardware design limitation of the computer, there is a middle man, known as the interpreter, who deciphers your instructions and in turn instructs the computer execute a set of machine language three, referred to as GROMs O, 1 and subroutines that accomplish desired task. In a lot of cases the the monitor, which takes care of the additional time required for the power up routine, and some of the interpretation of these commands is mathematical functions available in of no concern. Another trade-off is the computer. GROMs 1 and 2 contain the inability to access some of the the TI BASIC interpreter. internal features of the machine, such as bit image graphics and 40 plug in modules. Also reserved for column display.

Since the computer is it? Well, the folks at TI came up with yet another language called or Graphics Programming Language. This is an interpreted language. The OPL language closely resembles assembly language, and offers access to all the features of the machine. I'm not sure where the Graphics in the name came from, but commands that make it easier than assembly language when it comes to printing characters on the screen. The GROM memory in our computer contains programs written in GPL language.

basic language interpreter in ROM, that are written in GPL? Well..... our ROM contains a GPL interpreter. So where is the Basic interpreter? In GROM! Our Basic (and Extended This fantastic little box contains Basic) interpreter is written in 40K of simulated GRAM that resides GPL. That explains why our Basic is in GROM slots 3, 4, 5, 6 and 7. The slower than most other computers. memory is actually 8K RAM chips with The program that is interpreting some fancy electronics that make it your Basic programme is itself being appear to the computer as GRAM. And interpreted by the program in ROM. unlike GROMs, this GRAM uses up the This double interpretation takes entire 8K sections alloted for time. So much time in fact, that GROM/GRAM some instructions for the Basic contains 16K of normal interpreter are also stored in ROM. Which is bank switchable in 8K Written in machine language, these segments into the available module subroutines keep our basic from space. In addition, there are three being any slower thanit already is. empty sockets on board for optional The 99/4A has the capability of GRAMS to be used in GROM slots 0, 1

computer to do the up to 6K of stored data. GROM's contain only 4K of data, they always start at an even 8K boundary Instruction in the GROM memory map. This means Code or BASIC. Basic is one of the that there can be 48K ( $6K \times 8$ ) of easiest of all programming languages GROM data, but since there is 2K per GROM of unused space, the computer more correctly the The trade-off for ease of interpreter) can actually access 64K GROM. The unused space is not a the GROM memory chip itself.

Of the possible 8 addressable to GROM's, the TI console contains only the 2. GROM O contains what is known as The other 5 GROM slots are reserved for use in modules is a 8K slot in the computer memory map for "normal" memory such cannot as ROM or RAM. The TI EXTENDED The TI EXTENDED execute a program in GROM, what good BASIC module contains 24K of GROM (GROMs 3, 4, 5 and 6) and 12K of Then 12K of ROM is squeezed ROM. into the 8K available slot in the memory map by a method known as bank switching. Although there is 12K ROM, only 8K is active at any one time.

There is one other type of I suspect that it is because GPL has memory that our computer is capable of accessing. It's known as GRAM, or Graphics Random Access Memory. As RAM is to ROM, GRAM is to GROM. Unfortunately, TI didn't see fit to include any GRAM in the 99/4A. Boy, wouldn't it be nice to be able to make a few changes to some of those Where most computers have the programs, such as Extended Basic,

GRAM KRACKER to the rescue! use. unit also RAM memory, accessing 8 GROM's each containing and 2. There is another 8K of ROM

ca GR th CO ρl CO an Ca ma CO th mo me re th

Κſ

Ó۲

Ġ١

ar

CC

as

me

ir

Of

th

MC

рí me in as is bo Th

de

ty re 5 e ( COI in wh COL OX

to

un:

rea

105 mic the UPY P09 usi res mod mod "nc aut YOU

pos

reg 1 oa pog

memory that contains the KRACKER operating system. This program allows saving or loading the GRAM contents to disk or cassette, only if and also allows you to save the optional contents of GROM memory in a module OpSys/GRAMO switch selects Ther's also a nifty little the memory editor that can be used to console or the GRAM O chip in or GRAM, including the RAM memory in the console or P-Box. The saving of could make a few small changes? Just the character set. plug in the GRAM KRACKER, save its contents to disk, remove the module, and load the program into GRAM. You can then use the memory editor to the changes, reset and enjoy your "new" menu module! and since the GRAM KRACKER the memory is battery backed, you module remains intact even when you turn computer off. Now for details....

The GRAM KRACKER is a sturdy piece of hardware. It's housed in a metal enclosure that when inserted in the module slot extends out about as far as the Navarone Widget, bottom row of keys on the keyboard. resting your right hand on the blank section below the module slot. contacts on the edge card that plugs | The 2 into the computer are gold plated, which reduces the probability of switch. computer difficulties due oxidized contacts. There are 5 toggle switches on the front of the that are used for various write unit Let's take a look. reasons.

The first switch is more or an on-off switch. less In the middle of it's three positions is on, or "normal" mode. The upward position is "off". This using a module that does not have a rest line through the GRAM KRACKERS: module conector. The Atarisoft modules are of this type. With "normal" modules the GRAM KRACKER automatically turns itself off when you to select whether or not the The third|GRAM you plug the module in. position οf this switch performs a reset of the computer, and is spring the computer is first turned on or loaded to return to the normal position when you remove your

e

GRAM finger.

next two switches are used The you have installed the memory chips. The between normal GRAM O chip in the the inspect or change any portion of RAM GRAM KRACKER. Since GRAM O contains the power up routine for the computer, any program stored in GRAM module contents is facilitated by a 0 must contain a routine that takes cartridge connecter located on the over the operation when the computer GRAM KRACKER. Got a favorite module is turned on. The greatest (for me that you'd like even better if you anyway) use for GRAM O is to modify A11. patterns for characters used in BASIC, most . EXTENDED BASIC, and other modules are contained in GROM 0. Things like a slashed zero and a the true lower case characters are computer, select the program from possible by transfering the data in GROM O to GRAM O and then changing pattern codes. BASIC/GRAM 1-2 switch selects between the TI BASIC interpreter the contained in GROMs 1 and 2 and a GPL program stored in GRAMS 1 Using these GRAMS would allow modifications to console basic, or yu could replace Basic with another GPL program.

Next on the panel of the GRAM about 2.5cm higher than the KRACKER is another three position switch. This switch is used to This makes it slightly harder to select between the two banks of 8K type, especially if you are used to RAM in the unit, and also serve as a write protect for all memory in the The KRACKER except for GRAMS 1 and 2. banks of RAM can be independently selected by this Whenever a RAM bank is to selected in this manner, the write is switched protect for the unit off. When the switch is in protect position, the RAM banks can only be switched bν software. This switch has to be out of the write protect position in order to load data into the GRAMs. The loader in the GRAM KRACKER informs you when and where to position the switch for loading. position is necessary only when The write protect feature was left off GRAMs 1 and 2 to allow this space to be used as a buffer area.

> The last switch is labeled LOADER ON/LOADER OFF. this allows KRACKER loader selection appears on the main menu screen when If the loader is on, it reset. replaces the TI Basic selection

menu. Bγ selecting the GRAM KRACKER from the main menu load or save memory in the KRACKER, or save a module that is plugged in. from the GRAM KRACKER selection.

Now that you know a little more KRACKER at one time. about the probably wondering why you need one. copy modules. While it is true that your favourite module. you could obtain and run module software without buying the module borrow the module in order to copy it), there are a couple of rasons why this doesn't justify buying a GRAM KRACKER. First of all, module software is copyrighted, and it's against the law to copy them. Secondly, you probably already have most all the modules you'd really use, and with the falling price of cartridges you could most likely buy the ones you want for less money than you'd pay for a GRAM KRACKER. So what makes this contraption so great? I think one reason is because it allows you to enhance the modules that you already have. But not everyone is a programmer, right? Right. To modify module software, most of which is written in GPL, requires a pretty good knowledge of GPL and the internal workings of the This leaves everyone out computer. except for assembly language folks, WRONG! The manual that comes hight? with GRAM KRACKER includes detailed instructions on making some of the most commonly wanted modifications to the popular modules. Things like changing the default screen colours in Xbasic, Editor/Assembler and TI Writer, modifying the TE II cartridg! operate 1200 baud. at and changing the Tax Investment and Record Keeping module to print to a PIN connected printer. Also included are instructions for some modifications to "the operating system, such as skipping the title screen and mend on power-up and automatically executing item 2 on The manual I received was a temporary one, but it sufficiently detailed the operation of the unit. Also included was a utility diskette that contains the files for patching in other modifications to the Xbasic and Editor/Assembler modules. E/A mods add 5 new load and EDIT and ASSM files into GRAM for quick loading times, and the XB mods add 5 new CALLs including an on screen

clock. This disk you can temporary one. The final version is supposed ta contain modifications including a programme The memory editor ia also avrilable that allows you to load Extended Basic, Writer Editor/Assembler all into the GRAM Now that would GRAM KRACKER, you're be a super module! And Millers Graphics is not the only Most folks think it's main use is to offering instructions for modifying

The GRAM KRACKER has created so (provided of course that you can much traffic in the TI Forum on CompuServe that the SysOps there have dedicated a section of the database exclusively for information concerning it's use. This section, labeled "KRACKER HACKERS" contains the details for such things as how to disable the Quit Key on power-up, by-passing the auto disk load upon entering Xbasic, and many others. Millers Graphics has provided much of this information, but since the Forum is a hangout for many of the brightest programming minds in TI circle, there seems to be and endless οf tidbits supply of information on neat things the KRACKER can be made to do. really neat item provided by Millers Graphics was the code for adding a CALL CAT to Xbasic. This catalogs a disk to the without disturbing the program, and is available without having to load in a program every time you turn your computer on!

> As excited as I am about able to modify my much used modules, I think that the greatest thing that the GRAM KRACKER has to offer is the ability to have large programs that do not occupy any of the computers RAM memory. Imagine a full featured terminal or database program with 32K of data buffer! Millers Graphics has promised a GPL assembler and programming manual soon, it shouldn't be long before programs start to appear that take advantage this αf additional programming memory. I think that eventually the GRAM KRACKER will be thought of as an almost necessary peripheral for the serious TI user. The GRAM KRACKER is available from Millers Graphics, 1475 W. Cypress Ave., San Dimas, California 91773. The price is \$US 178.95 plus shipping costs, or \$US 192.45 for the unit including the optional GRAM chips. I think it's worth the price.

Un mi

11 NE th in

POL

Af t the 100 13 nen 911 whi 947 The its The fro

oth ret dod hee spe pho

pha mai Eve imp A11

eff gra it. My

## ANTERNAROR A - COPS!!

Unfortunately some missing from the ANIMATOR 2 program listing in the last issue of HV99 NEWS. I am at a loss to explain how this happened but apologise for inconvenience caused

2050 PRINT #2: "CALL MOTION(# 1,0,SP#6)":"FOR I=BE TO E":" CALL PATTERN(#1,60+1\*4)":" D EL=1 TO 40": " NEXT DEL": "NEX T I" 2060 PRINT #2: : : : 2070 FOR I=1 TO 20 :: IF E#( I)=RPT\$("0",64)THEN 2080 :: PRINT #2: "SPRITE No.-"; I; "--":E\$(I) 2080 NEXT I 2090 CLOSE #2 2100 RETURN

## JUNI OR SOFTWARE REUIEN

POWER WARS.

e d

М d s

e

19

SO.

15

a

en

nd

ad

rn

ng

at

he

at

rs

ed

th

CS

ind

it

ims

ige

ng

he

85

or

lΔM

rs

an

ce

is,

lng

After loading and running this game, the title screen appears and things look promising. A very catchy tune BIG HINTS is played, along with the authors |1) Never names being displayed. You game and then the fun begins.

The graphics are great but the game itself is very simple and boring. The object is to move a power 'droid CALL GCHAR(Y, X, C) :: SC(Y, X) = Cfrom one side of the screen to the NEXT X :: NEXT Y other to collect power and then you came while What whence dodging "photons" My main criticism is the array photons. With each journey the 5, photons get faster while the droid at SC(5,8) maintains it's steady Eventually impossible and frustrating.

All in all, it is quite a effort i f and

My rating - 7/10

By Scott Johnston.

### PROGRAMMING HINTS FOR EXTENDED BASIC

This article Rob ByJarrett. from Channel 99 User's originally appeared in the November issue of The CLUBLINE-99 Magazine

The following article gives some programming hints that I have useful in programming Extended Basic. It's divided up into 2 sections: and Little Hints. Big Hints would visibly improve your program, whereas Little Hints would just save memory, or improve the speed of your program a little.

use the GCHAR subroutine. are Instead, DIMension the screen into a given the option for instructions 2 dimension array. Then, before your which gives a basic run down of the main loop of the game, type the following:

FOR Y=1 TO 24 :: FOR X=1 TO 32 ::

this does put falling from the information on the screen 'SC'. For example, speed of the 'droid compared to the asterisk (ASCII 42) was placed at row column 8, then the number stored would be equal crawl. Just remember: whenever something on making the trip becomes the screen changes (say the of a monster), then the variable good would have to be changed. Whenever you like great you would use a GCHAR statement, like graphics and sound effects, go for CALL GCHAR(Y,X,C) :: IF C<>32 THEN 340, it would become IF SC(Y,X)<>32 THEN 340. This totally eliminates the GCHAR, improving the your program.

2) Never use calculations that would LITTLE HINTS be repeated over and over. For examble:

100 CALL CLEAR

110 CALL SPRITE(#1,42,2,97,1)

120 FOR X=1 TO 256

130 CALL LOCATE(#1,SIN(2\*PI/X)+97,X)

140 NEXT X

150 GOTO 120

the computer has to calculate the or retype the function you want sine of a number over and over again. performed. A much faster way of accomplishing this is to use an array again.

100 CALL CLEAR

110 FOR X=1 TO 256

120 Y(X)=SIN(2\*PI/X)+97

130 NEXT X

140 CALL SPRITE(#1,42,2,97,1)

150 FOR X=1 TO 256

160 CALL LOCATE(#1,Y(X),X)

170 NEXT X

180 GOTO 150

calculating everything 440 GOTO 300 beforehand, your program will execute

3) Instead of using numeric arrays, the characters of the data you're|4) Try not to use too many GOSOBs as using can be strung together into a they use valuable memory. They also string array. This does not improve can make your program very confusing the speed of your program, but it to debug. saves a lot of memory. Example:

100 DIM SC(24,32)

110 FOR Y=1 TO 24 :: FOR X=1 TO 32

120 CALL GCHAR(Y, X, C) :: SC(Y, X) = C

130 NEXT X :: NEXT Y

140 END

This would become:

100 DIM SC\$(24)

110 FOR Y=1 TO 24 :: SC\$(Y)="" :: FOR

X=1 TO 32

120 CALL GCHAR(Y,X,C) :: SC\$(Y)=

SC\$(Y)&CHR\$(C)

130 NEXT X :: NEXT Y

the first example, the computer had to set aside memory for 768 (24) variables. In example 2, the computer just had to set aside This article, by YVES aside memory for 24 string variables. appeared in the November, 1985 issue To access the data in the array, you of Clubline-99 magazine. would do the following:

200 C=ASC(SEG\$(SC\$(Y),X,1))

screen column.

1) Always try to scrunch as much you can on one program line. Yes, this contradicts what you learned at school or Computer Camp, but this increases the speed of your program and saves memory.

me

for

cai

WIT

rei

Fi

pro

edu

tea

WOI

WOT

pra

on

5 y 2

cou

wor

The

on.

dif

typ

sto

i s

sto

WÂY

int

i 5

to

450

w11

tre

bf :

whe

com

hec

Fi1

DPE

fol

DPE

[, 1 ,0

the

FIL

bet

res

Thi

tim

to OPE

lcor

DEV tha to.

\*RS

mod

Never use the DEF statement, or make up your own subroutines (such as CALL FRED) as these slow your program This is quite a slow program, as down too much. Instead, use GOSUB,

> 3) IF Use logical statements. Example:

430 IF C=32 THEN 440 ELSE 450

|440 CALL SOUND(25,1000,0) :: GOTO 300| 450 CALL SOUND(100,110,0) :: GOTO 300

The IF statement in line 430 is badly written. It should be:

430 IF C<>32 THEN CALL SOUND(100, 110,0) ELSE CALL SOUND (25,1000.0)

This example is faster, and uses less memory than the first.

As you can see, many of these hints are just common sense. you've seen some of the programs people write (and sell!!) it makes you think no one has any sense at a11. Anyway, it looks like this month's article is over, next time, BYE.

# FILE PROCESSING

#### GENERAL

A FILE can best be defined as a The variable C is the ASCII code information stored on some medium. for the piece of data on the screen, First let us discuss the difference is the screen row, and x is the between a PROGRAM FILE and a DATA The Program File contains FILE.

merely a collection of instructions a disk type file). Each of these for the computer. This type of file devices are described in sufficient can be used directly by the computer detail in their respective user without any other information being manuals. required. On the other hand, Data Files must be accessed БУ program. For example, i f educational the spelling of some 2000 specified words, the inclusion of these 2000 sequential) or RELATIVE. words in your program is not very means that the information will be practical and may not work depending read using an INPUT statement and the On the hardware configuration of your information will be read one after system. This is where a Data File the other. could be used to contain all 2000|relative files can be read in any words to be used by the program. order. This is accomplished by using These two files are stored separately record numbers; in other words you on a disk or cassette.

Files are stored in different ways. typical program or a Program File is stored in Memory Image format, that FILE-TYPE : there are two file types is to say, it has been compacted and to choose from. way the computer stores internally. This particular format normally is normally only used by the computer alphanumeric information, in other to save and retrieve Program Files.

user defined, in other words, you called INTERNAL and is more efficient will determine how the file is to be for storing numbers, however, if used created and how it will be read. All to store characters, it will require of this information is established about the same space as DISPLAY. when the OPEN command is used. gives a program all necessary information about the Data CUTPUT UPPATE and these are INPUT, File.

DPEN STATEMENT The OPEN statement has the |following syntax:

OPEN #file numbersdevice filename [,file-organisation][file-type] [,open-mode][,record type]

A brief description of each of the above entities is as follows:

reserved for the keyboard and screen. than others then they will be padded. This file will be referred to every with blanks. Any that are longer time the file is read from or written will be truncated if FIXED is used. to (INPUT/PRINT commands). For every OPEN statement there should be a adjusted for each record, however, corresponding CLOSE statement.

"PIO" (parallel printer), manuals.

a program which is "DSK1.filename", "DSK.disk.file" (for

the FILE-ORGANIZATION : this parameter an specifies either SEQUENTIAL (default program is written to this means it only needs to be i f it is other than Sequential On the other can read the last record followed by the firs record (relative files are many also sometimes called RANDOM files For example, a and can only be used with disk).

The first stored on the medium in much the same DISPLAY, whic is also the default and it the most common. This file type is used to store or print words it stores exactly the same way it is displayed on the screen, hence The format for a Data File is the name DISPLAY. The second type is

> the OPEN-MODE : there are four open modes OUTPUT, UPDATE (default), and APPEND. The INPUT mode allows you only to file read a (using the INPUT command), the OUTPUT mode allows you to write to a file (using the PRINT) command), the UPDATE mode allows you to read and write from and to a file, the APPEND mode does much the same as UPDATE but it is used for vriable length records, whereas UPDATE is for fixed length records.

RECORD-TYPE : there are two record FILE NUMBER: is any valid number the record in the file are of the 1 and 255 with 0 being same length or if some are shorter this is not very efficient because DEVICE-FILENAME: this is the device be saved. If FIXED is used with no that you wish to read from or write record length the default is so to, "CS1" for cassette no. 1, characters for a diskette file. All "RS232" for RS232 no. 1 (printer or defaults are listed in the user

DEPARTMENT OF INDUSTRY AND COMMERCE



### Australian Customs Service

Reply to the Comptroller-General

Quote C84/4375

Edmund Barton Building Canberra A.C.T. 2600

Telephone 724826

// October 1984

Mr Stephen E. Taylor

Dear Mr Taylor

Senator Button has asked me to reply to your letter requesting details of the current import duty and sales tax rates on computer hardware and software from America.

Floppy disks and cassette tapes recorded with computer software are classified within item 92.12.100 of the Customs Tariff and are dutiable as follows:

- in respect of the carrying medium 30% to 5.7.85 25% from 6.7.85 to 5.7.86 20% from 6.7.86
- in respect of the remainder (i.e. software) Free

There is, however, insufficient information to enable me to give you the tariff classification and duty on the modules (read only memory cartridges, and the memory cards. Should you still require this information, I suggest you write to the Collector of Customs in Fremantle and supply him with as much detail as possible to assist him to meet your request.

The question of sales tax on the goods you are interested in should be referred to the Deputy Commissioner, Australian Taxation Office, 1 St. George's Terrace, Perth W.A. 6000 for consideration.

Yours sincerely

(P.J. LAWLER)

A/g Assistant Secretary

Import Operations

### MINI WORD PROCESSOR DISSECTED

 $\mathsf{CW}(\mathsf{L}f s(),\mathsf{L})$ , the subprogramme  $\mathsf{CW}$  is if the input was a null, and if it the subprogramme that looks after is then a SUBEXIT is performed back change word selection, although I to the RETURN at line 240 and then have called it change word, it will back to the main menu. change a whole string if you want to the user a chance to escape the also not just a word. And before I subprogramme if they pressed the to explain how subprogramme works I will explain is not entered then the computer some of the things that are needed waits for the user to input the "New in this subprogramme first.

Firstly we need to know the LI with a different prompts word changed, and the word or string that variable SL used to store the start is to replace it. Then the quickest line, and EL the end line for the way to do it is to check the length search for the old word, in each of the line of text and if it is case if a zero is returned from LI shorter than the word or string that then the programme loops back for a you wish to change then there is no valid text line number. In line 740 sense in looking to see if it is in the length of the old word is placed that line of text. After we have in the variable LE, then a loop is what looks like the string that we started from the start line SL to wish to change, we have to make sure the end line EL, and then the length that the string we have found is not  $\mid$  of the text line is compared to LE part of another string. For example the length of the old word, and if if we wished to change the string it is shorter than LE there is "test", to something else, then we need to check if that line of text dont want to change tester or contains the old word. If this retest, or any other string that is so then the programme branches to part of another. So first we have line 810, and the NEXT statement, if to check the position one place it was long enough then the value 1 before the string that has been is stored in the variable A. found by the computer using POS or line 750 the line of text SEG\$, and to keep the programme searched for the start position simple and to conserve memory I only the string OW\$, starting from considered a null for the start of position A, with the POS function, the line of text, or a space. and the position stored in the the line of text, or a space. and the position stored in the Others such as an open bracket would variable P. If OW\$ is not found in have probably been acceptable also, and there are probably others that branches to line 810 again. If P=1 are, that I have not thought of then OWst must be at the beginning of also. If anybody uses the programme the line and as there is no need to to write a lot of text like this it check one position before the start is not hard to add the extra logic. of OW\$ the programme goes straight We next have to examine the position to the next line, one place after the end of the greater than one the line continues string that the computer found to on to check one position before the see if it is acceptable. The only start of OW# and things I thought were acceptable was space the programme branches to line the null at the end of the line, a 770, where A is now given the value space or a punctuation mark and I $\int$  of 1+P, that is one position past only accepted three of them. Again where the string OW\$ was found in any others are wanted the logic! would not be hard to add. That is the main things that need to be thought about before I wrote the the code, and now lets have a look at position 23 then the "test" part of the code and how I did it.

The subprogramme starts at line 700 where I called the array L\$ again and the text line counter L. A dummy CAll KEY 5 is performed to turn on the full key board again, the next line displays the and prompts "Old word" and "New word", then accepts the "Old word" and We continue on at line 240 CALL stores it in OWS and checks to see This gives the wrong menu selection key. If a null word", which is then stored in the Then our old friend variable NW\$. or string that is to be called in line 720 and 730, with the the line then P=O and the programme but if P if it is not line the first time. the example if you wished to change the word "test" to some thing else and "retest" word started the word would start at position 25

when the position 24 was looked at to see if it had a space there, to make sure that "test" was not part of another word, which in this case it was, we increment A to one position past where OW\$ was found ready to search the text line further along. Though in this case we check the value of A against the length of the line of text we searching, to make sure we do not try to search past the end of line. If the position one place before the string was alright then at line 760 one position past the start of the string is looked at, and stored in the string variable CH\$, and if CH\$ is not acceptable HI in the line A is next incremented as I explained before Welcome back to the place past OWs. Then the full else took over for a few years. length of that line of text is added to the number of bytes of free space stored in L\$(0). Next the current | \*\*\*\*Adventurers\*\*\*\* line of text is changed to CHS and NW\$ and C\$, that is the first part HINTS-DEADLINE of the line up to the old word, then the new word and lastly the last How ever does one have the time After that at line 790 the length of in all the flowery Questions in the changed line σf text subtracted from the number of bytes Sleuths have spent 18 Manhours free, and the programme loops back are not much nearer an arrest. to 750. 770 as A does not need incrementing answers to our brief Questions. as POS will find the next ocurrence Here are clues we have Unearthed. of OW\$ in that line, as the last ocurrence has been removed. as I write this it makes me think to the garden and he is annoyed, it would have been better to him whats wrong. loop back to 770, or would it. Some thing for you to experiment with if |(b)Mr| McNabb will show you the holes not sure about this) instances that piece of a cup. Analyze it! method could miss an ocurrence of the old word. statement for the loop and when that drops a ticket stub if you scare is finished all the intermediate enough. strings are set to null to conserve questions. memory, and the the SUBEND is is performed back to line 240 and the RETURN that goes back to the main is Lo Blo. menu.

will probably finish the disection turn in the next months mag.

Brian R.

# LIBRARY NEWS AL LANRENCE

50

th

Whi

th

ar

We

Sa

Sp

an

D

We

Bas a j The

vei

аli

We

If

S

FUN

thá

For

reb

Th Ori

mar

Pro

the

Pro

dog

Dig

Wel

Thi

eve

now

Any

obt

in:

(a)

us

di

sen

dis

or

(b)

dis

dis

new Year and then the programme loops back to don't forget that-We are looking for try again. If every thing was NEW BLOOD- at the AGM. We have to alright then the programme branches have a GOOD Committee to keep our to 780 and CH\$ is assigned the part group active! SO Volunteer and of the text line up to but not Nominate to ensure we do. Now that including OWs, and Cs is assigned the Library is established I think it the rest of the text line from one would be good for the club if someone

and: part of the line past the old word. patience to be a Detective and type this is Infocom game of Mystery? 3 of us HV99 and. Who I did not loop it back to does not love reading the wonderfull

Though (a) Wait until after Mr McNabb goes

you wish, most times I think it may in the garden made by the ladder from be faster, but I think in some (I'm the Shed, DIG enough here and find

Line 810 the last (c) Front path-after lia.m.a paper line of the subprogramme is the NEXT appears, Read carefully, Ms.Dunbar her Get 1 t her

> (d)Dunbar's bathroom-In the cabinet

(e)Library-Get pad, pencil calender. I will finish at this point and Shade pad with pencil. Examine ru page Ωf ⊂alend**e**r. Go George's room and show it to him.

(f) I have heard in secret shelves.

the Will.

Who dunit?? Well we know it was the Butler, (there is none) There(the are only 6 people involved, By logic following) Idecided to go ahead and we have eliminated the Gardener Maid. fit them. Steve came down and soon So we will have to do some more we had connected the Disk control Spadework as I have lost my Licence cable to take 2 Drives. and been returned to Police School.

#### Disk

Base in XBasic which seems to be what P.E systems (one with a The Documention is on the disk and is and the later types have Rocker also a DEMO prog.included.

We have also updated DM1000 to V2.3 If you would like these let me know.

#### STOP PRESS

FUNNELWRITER V 3.0 is now ready and Opening the PE Box is easy, just Orignal and lots of improvments on leaving the Base Back exposed. other Commercial Processors. that it will also RUN heaps of other from the base. Programs, Utilities FORTH. documentation is included on Disk.

Well Done Tony Will McGovern.

This Disk should be a must everyones Library and is available heat sink. now by the usual means.

#### Library Access

Any clubs or indivuals interested obtaining any PUBLIC DOMAIN software bridges, in volume disks have 2 choices

- (a) Send blank initialised disks to output, us with return postage or send us Power down each time to do so. disks with programs on it and we will Everything should work so reassemble disks filled with programs requested secure. I made a or volume disks.
- (b) We can supply programs, or volume disks on our disks for the cost of Happy Computing. disk and PP<u>(\$4.00)</u>

one room about Double Update

Having decided to give my 99'4A extra storage and talking to a few others (g) Make sure you hear the reading of who had done so, there seemed to be no problems to it!!

Chinon F502 1/2 height DS/SD as not best way to go was choosen, and when 2 Chinons arrived The wires supplying the +12 +5 volts were paralleled fired up. Nothing!!#!!. We then tried everything on Steves PE System it WORKED.

The Big difference in the end We have received A PUBLIC DOMAIN Data traced to the fact that TI made TWO Push/Push a lot of people have been asking for on/off switch in the earlier Models, very clear to understand there is off switches) The +5v in the earlier model is supplied via a 7805C 1 regulator in the later one 78M05C .5 Amp regulator. Which allows the Volts to drop to 3.5 and this is is too low to enable the 99/4A to recognise that the Disk drives are connected.

the Delight in going from Editor to matter of unscrewing the screws at Formatter endless times not having to back, sides base a total of 12. reboot has to be seen to be believed. Remove all Cards from their slots This has all the features of the slide the front and sides Foward

Ward Remove the connectors to the Ai Power It has the added benefit Supply P.C.B. and unscrew the board The +5 Voltage New Regulator is Q2. Use Solder Suck the|braid and sufficient heat to remove solder from the 3 pins. Unscrew and the old regulator (78M05C # 8245) fit new regulator (7805C #8419-cost \$1.20 at Dick Smith). Use a dab of heat in transmitting paste when rescrewing to

I took the oppertunity to renew the noisy fan with a Quieter one. existing one is a 115V 60 Hz I put in a 240V 50Hz. Now I can almost hear the cursor blinking!

in After testing that there is no solder reconnect the P.C.B. connectors and screw it on the base again. Power up and check Voltage connect 1 Drive at a time

send at our cost an equal number of P.E.cover and fit the 2 drives in and template drilled 4 new holes while the cover was off.

Al Lawrence.

# ELASSIFIEDS

WANTED Ron Kleinschafer is after the following items. RS 232 Card, E/A and manuals or Mini Memory. Write to.

Ron Kleinschafer Grawin Via Walgett 2832

WANTED Joe Wright wants MICROSOFT MULTIPLAN. If you have a copy you wish to sell give Joe a call on 468120.

WANTED My JULY 1983 issue of Home Computer Magazine back. I lent it to someone and can't remember who!!!!!.....STEVE

WANTED Weight Control and Nutrition Module. Call Steve on 487076

DON'T FORGET THE A.G.M. 11/3/85

YOUR ATTENDANCE IS REQUESTED AT THIS VERY SPECIAL MEETING. IF YOU WOULD LIKE TO HELP OUT ON THE COMMITTEE FILL IN THE NOMINATION FOR PROVIDED WITH YOUR LAST NEWSLETTER

