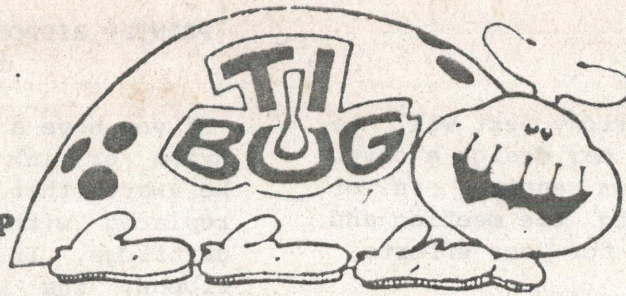


TIBUG
 TEXAS INSTRUMENTS
 BRISBANE USER GROUP



JULY
1987

Postal Address P.O. Box 57,
 Aspley. Qld. 4034.

BUG-BYTES

DISCOVER THE
 SENSE OF

EXCLUSIVELY

90 DAYS OLD
 OR LESS



COGNAC

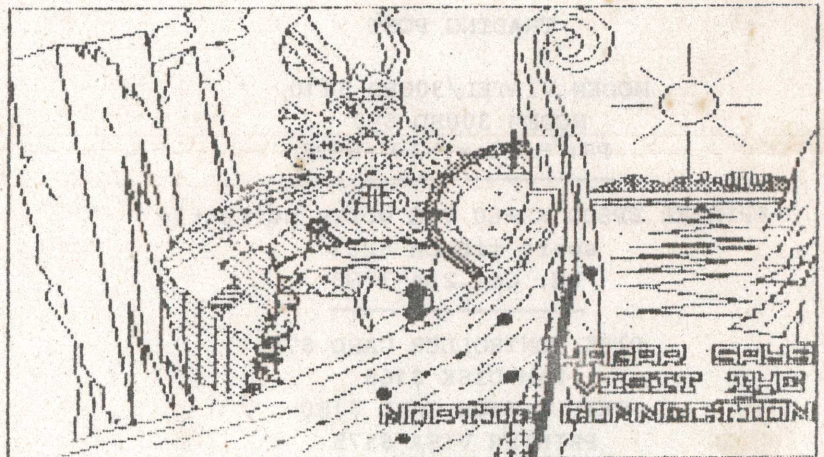
COGNAC

OLD TENNIS SHOE

DESIGN A COVER COMPETITION

DON'T FORGET!
 Design
 The
 Cover
 Comp
 SEE PAGE 2

DESIGN A COVER



CONTENTS

- MONITOR INTERFACE.....3
- LIBRARY UPDATE.....4
- ASSEMBLY TUTORIAL.....5
- SOFTWARE REVIEW.....10
- PROGRAMME.....14

DATE

31st JULY

PLACE

EAST BRISBANE STATE SCHOOL
 Cnr. Wellington Rd. and
 Stanley Sts.,
 EAST BRISBANE.

ROOM 4

July

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

The club meeting on Friday next will be the closing date for our design a cover competition so send your entries in or bring them with you to the meeting and maybe win a box of disks for your efforts.

For out of town members unable to make the monthly meetings but would still like to read the latest in newsletters from interstate and overseas newsletters then read on. The club is considering a newsletter circuit which would involve the club posting newsletters to member (a) who after a week would send them on to member (b) and so on, address labels will be included to make it easier. Members interested or members with any questions please write or ring as the soon as we get any replies we can start the circuit.

The June meeting had good attendance which made for an interesting meeting with lots of topics discussed and programmes exchanged. Chips for the new Ram Disk Cards were also available and went like hot cakes. John Campbell was elected as Vice President and Alan Saunders being nominated as Assistant Librarian.

At the July meeting there will demonstrations of the Ram Disk and Super Cartidge making this a very interesting meeting. The Ram Disk boards will also be available for those who have ordered them.

TRADING POST

MODEM VIATEL/300BD \$110
 MODEM 300BD \$80
 PH: Hillary 397 5926

PRINTER EPSON RX-80 VGC works perfectly
 spare ribbon \$299
 PH: Col 284 7783

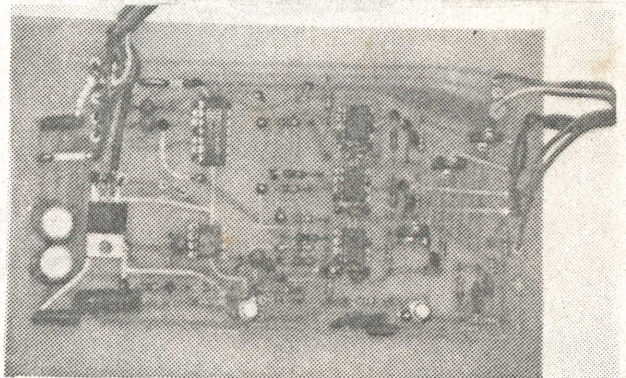
DISK CONTROLLER CARD \$75
 RAM DISK \$160
 TRIPLE TECH CARD \$160
 PRINTER VISA \$175
 MULTIPLAN UNOPENED \$85
 CONSOLE ASST. MODULES \$100
 EXPANSION BOX \$100
 TI-WRITER MODULE \$20
 EDITOR ASSEMBLER MODULE \$20
 PH: JOHN (071) 95 3771

TI WRITER MODULE & DISK with MANUAL \$25
 SPEECH EDITOR MODULE with MANUAL \$10
 TI LOGO II MODULE, DISK & CASSETTE
 with MANUAL \$30
 PH: Andrew 202 7499

PRINTER RIBBON REPLACEMENT
 by Chas Bagley

If you have a dot matrix printer having a nylon or ink type ribbon, you may already be aware that you can have the ribbon replaced without having to purchase a new cartridge. If your printer uses a carbon ribbon, the good news is that now it too can be replaced without having to buy a complete cartridge. This service is available from JANE'S COMPUTER SUPPLIES, 48 CRIBB ST., MILTON, 4064. If you take your cartridge there between 10-00 a.m. and 4-00 p.m., the service will be carried out while you wait. Cost to replace the carbon ribbon in the CP-80 printer is \$8-00, a saving of at least one third the cost of a new cartridge.. Nylon ribbons for most printers, including the EPSON, can be replaced in both black and colour, for approximately the same cost, or a little more. Units left outside the above hours, may be collected the next day.

A word of warning, if your printer is fitted with a carbon ribbon, do not replace it with a nylon ribbon as the print head will be damaged if you do.



MONITOR INTERFACE FOR THE TI-99/4A
 REPORT ON FOLLOWING PAGE

COMMITTEE

PRESIDENT - BOOK LIBRARIAN
 Garry Christensen - 284 1841

VICE PRESIDENT
 John Campbell - 351 3107

MINUTE SECRETARY
 Ralph Nielsen - 369 4062

TREASURER
 Col Christensen - 284 7783

PROGRAMME LIBRARIAN
 Larry Read - 821 1051

ASSISTANT LIBRARIAN
 Alan Saunders -285 5974

EDITOR
 Ross Whiley - 285 2173

MONITOR INTERFACE FOR THE TI-99/4A

By Chas Bagley

Most home computer owners, unless they happen to be well heeled computer buffs, are content to use a TV set as their visual display unit. This has the advantage of being moderately priced, readily available from any electrical discount warehouse, and able to be used as an ordinary TV set if required.

The tradeoff against these convenient features is a significant loss of screen image sharpness. This happens because the data output from the computer has to be modulated into an RF signal that the TV set's tuning circuit can recognise. This signal is subsequently demodulated, and is presented on the screen as a somewhat "over processed" image of moderate fidelity.

At least one manufacturer has marketed a TV set which also has an RGB input. The set referred to is a Phillips 34 cm. model CP 415 TV/COMPUTER MONITOR. There may be other brands available, and any set having an RGB input at TTL level could probably be used as a monitor for the TI.

Australian TV uses the Pal system and for this reason all TI-99/4A computers sold in this country have a TMS 9929 video display processor chip fitted instead of the TMS 9918 chip, as fitted to the computers sold in the US. This means that as the TMS 9929 chip has as its output Y, B-Y, and R-Y, an RGB converter (or decoder) is required between the computer console and the monitor. This has been a stumbling block in the past as, to the best of my knowledge, no such device has been commercially available.

Some months ago I was given a circuit diagram for such a device, but the drawing was of such poor quality that many of the component values were unreadable. A friend, Leigh Barker, redrew the diagram and then constructed a prototype unit to try out. As the original chip, as specified in the diagram is no longer available, a substitute had to be used. The results were quite disappointing as the image produced was of very poor quality. Another circuit, which uses a TDA 3560 PAL decoder chip was tried with even worse results. This chip, in theory should have the greatest potential to produce a good image, but we were unable to obtain sufficient information to get it to operate in the way we required.

For our next attempt, we tried a circuit taken from a computer magazine, and fairly similar to the one used for our first effort. This produced a much better image although there were still a couple of

minor faults. Leigh made a few modifications, designed a double sided printed circuit board layout, and then constructed the unit. This gave the result that we'd hoped for, a good sharp image with almost perfect color rendition. I have been using this unit for some weeks now and to be honest, I have become sufficiently spoiled to not ever want to go back to the old modulator.

Cost of the components, excluding the printed circuit board, is approximately \$40-00 TO \$50-00, depending upon where you buy them. As the unit draws very little power, there is virtually no heat generated and it can therefore be housed in any kind of case. Because of space limitations, I chose a compact diecast aluminium case costing \$21-00, but a cheap plastic case, included in the component costing, would be quite suitable. The board may also be left bare if preferred. My unit has approximately 15" of screened flex for both input and output, with a six pin DIN plug on each lead. When this unit is connected to your system, the modulator is not required.

A point to remember is that when your system is operating with the converter, the normal color and contrast controls on the TV set have no effect on the screen image. The brightness control is still effective but color adjustments are made using the three trimpots on the interface.

For the many people who use an ordinary TV set for their display unit, I am led to believe that it should be fairly straightforward to make up a small interface to connect between the RGB converter and the RGB connections on the TV set. This would then allow your TV set to be used as a monitor when required. We are investigating this possibility and hope to have something to report in the near future.

If anybody would like to purchase a PC board, orders for them will be accepted over the next month. Cost per board is \$23-00, which includes packing and postage, a circuit diagram, and a component list. All cheques should be made out to TIBUG, and should be sent with your order to: Chas Bagley, 4 Dakar Rd., ALGESTER, QLD., 4115.

Please allow several weeks for delivery, so that all boards can be produced in the one run.

As a final word let me say that I am very happy with the way my system is now working, and the above information is given in good faith. However, you will appreciate that no responsibility can be accepted regarding the outcome or performance of the converter or its associated equipment.

PROGRAM LIBRARY UPDATE

The following is a list of the latest software to be added to the users group software library. These disks are available for all club members. There is no charge to those members who bring disks to a meeting or copying day and duplicate the disks themselves. Two alternatives exist for remote members. Send either \$2.50 per disk and \$3.00 for postage and handling or include your own disks and send \$1.00 per disk copying fee and \$3.00 postage and handling. All requests for software can be sent to

A. SAUNDERS
9 MICHEL COURT,
PETRIE, 4502.

or phone 07-2855974 for further details.

All disks are single-sided.

226.....Sample sprite graphics created by JET SPRITE BUILDER V4 (disk 241). There is a program on the disk that allows you to view the sample sprites. It is loaded with the load program from extended basic. Author is John E Taylor.

227.....Turbo Pascal looks impressive, but the demo files and instructions are written in German. If anybody out there can read German please contact me and I will arrange for you to obtain a copy to interpret.

228.....This disk is comprised of four programs, two of which are updates.

- 1 SUPERCHEK - A management program for keeping track of your cheques.
- 2 RISING/SUN - An extended basic music program, not as good as AXELF (Axel Foley), but still well done. Original author Don Gilchrist.
- 3 PINBALL - Is the same as the original, but now has a HIGH SCORE facility. The list of high scores attained can only be viewed at the end of a game by pressing key nine. Once key eight is selected(start a new game)the high scores are not available. New sound effects have been added also.
- 4 ARCHIVER - Replaces the old ARCHIVER V2.1 . Option four of the menu (Catalogue ARC files) has been added to the menu. This option allows you to see what filenames have been compacted to form a packed file. There may be other changes that I am not aware of so get a copy and try it.

229.....Movie Pluto - A graphical moving picture of Mickey Mouse's faithful dog Pluto. Editor assembler option three,filename SHOW,program name START.

230.....Pulsar 1 - Assembly language routines written by Mike Amundsen. They allow you to write assembly language programs in a way very much like you would write a BASIC program. Three example programs are included on the disk.

231.....Super cartridge loader by David Romar. This includes assorted utilities and documents for loading the SUPER CARTRIDGE. Unfortunately this and the above disk were taken home accidentally by someone at the last meeting and if they could return these to me at the next meeting it would be appreciated. I am also told that during the school break Col Christensen made one of these cartridges up and it is intended that it be demonstrated at the next meeting.

232.....Super Cartridge Software No.1 - Includes instructions for making your own supercartridge and the object and source coding of two programs to test the supercartridge once assembled.

233.....Super Cartridge Software No.2 - Contains games, a disk fixer, Rom Raid and utilities for the module above.

234.....Super Cartridge Software No.3 † Contains

235.....Super Cartridge Software No.4 † games for

238.....Super Cartridge Software No.5 † Super

239.....Super Cartridge Software No.6 † Cartridge.

236.....Paragon Computing Enhanced Display Package V2.1 - A program to enhance your own extended basic programs by linking to new assembler subprograms such as Buffer, Char, Charpat, Charset, Clear, Clock, Color, Display, Flash, Gchar, Gtext, Hchar, Load R, Load V, Mode, Peek V, Reset, Screen, Scroll, Show, Steady, Text, Time, Vchar, Window, Intoff, Inton, Read, String and Write.

237.....A disk of three games written in extended basic. These games are very well done. Author Terry Staph.

1 JOKERPOKER - double or nothing is deadly, be careful with this option.

2 BLACKJACK - I have actually trebled my money on this one.

3 WHEEL OF FORTUNE - If you play this with your wife, don't tell her the answers. I did and being greedy I decided to keep going to build up my bank. Unfortunately on the spin after telling my wife the answers I was bankrupted allowing the wife to build her bank up.

240.....Horizon Ram Disk Op. Sys. V6.0 - Latest software to utilize your Horizon Ram Disk Card.

241.....JET SPRITE BUILDER - This disk is used to create sprites as in disk 226 above. It operates in assembly language.

That's all for this month.

REPRINTED COURTESY HUNTER VALLEY 99'ERS

LINPUT

AN EXPLANATION OF THE
EXTENDED BASIC COMMAND
BY "THE ROOSTER"

Extended Basic adds many additional commands and statements to those provided by Console Basic. One of these is the string input statement;

LINPUT

LINPUT NAM#

LINPUT "YOUR NAME ?":NAM#

LINPUT #12:NAM#

LINPUT only accepts string inputs, no editing is performed on the input stream of data.

A typical application for LINPUT is as the input statement in word processing applications. LINPUT accepts the entire line of text entered by the user. This includes leading and trailing blanks.

The following short programme demonstrates this feature. Type it in and try entering various input lines of text. eg. Leading blanks, names number sets etc.

```
100 CALL CLEAR
110 PRINT " ENTER A LINE OF TEXT
WITH": ":" LEADING BLANKS":
120 INPUT A#
130 PRINT : " NOW ENTER THE LINE
AGAIN":
140 LINPUT NAM#
150 PRINT : "THE FIRST LINE USED
INPUT":A#
160 PRINT : "THE SECOND LINE USED
LINPUT":NAM#
170 PRINT "TRY AGAIN? PRESS Y"
180 CALL KEY(0,K,S)
190 IF S<1 THEN 180
200 IF K=89 THEN 100
210 END
```


PUTTING IT TOGETHER

Loops and Things.

Well July is here already. The new financial year. Tax forms to fill out and pray that everything is in order. The tax always has me running around in circles so perhaps this would be an appropriate time to discuss looping in assembly language.

As in basic, loops allow the frequent execution of a section of code. This saves typing the code several times over. In BASIC the GOTO statement determines the location of the next statement to be executed. In assembly language this is done with the JMP instruction.

```
LOOP .           FIRST INSTRUCTION IN LOOP
.
.
.
JMP LOOP
```

Enough of the basics, its time to talk specifics. Loops can be of two types, controlled or uncontrolled. The uncontrolled loop is generally of little use in assembly because the system is locked into that series of instructions and cannot be made to do anything else.

LOOP JMP LOOP

This is an uncontrolled loop. It has the effect of locking up the computer. The only way to regain control is to switch off. For the more advanced programmers, if this instruction is preceded with a LIMB 2 instruction, the QUIT key is enabled. Do you know why (Hint look at the interrupt service routine).

The greater majority of loops are controlled, that is there is some method of leaving the loop. This is done when a condition is met. The loop must modify the control variable or allow input from the user or a device that will change the control variable.

What is the control variable. The control variable is the variable that records the condition for continuation of the loop. In the BASIC statement FOR A=1 TO 100 , the control variable is 'A'. When 'A' reaches a set value, in this case 100, the loop is terminated and the programme continues. In assembly language the value of the control variable is either a register or a memory location and must be adjusted within the loop. The FOR-NEXT loop can be performed manually in BASIC and the corresponding instructions apply in assembly:

```
10 LET A=1           LI R0,1   SET CONTROL VARIABLE
20 REM START OF LOOP LOOP .   START OF LOOP
.
.
.
70 LET A=A+1        INC R0    ADJUST COUNTER
80 IF A<= 100 GOTO 20 CI R0,100 TEST FOR CONDITION
JLE LOOP
```

The exact format for the loop can take several forms. The one above is called a 'REPEAT-UNTIL' loop and can be read 'REPEAT the following code UNTIL the condition is true'. That means that the loop will be executed at least once regardless of the initial parameters. The test for the termination of the loop is not until the end.

If the test is at the beginning, it is called a 'WHILE' loop, 'WHILE the condition is true do the following'. The test is made first

and if the parameters are correct, the loop is executed. The repeat-until loop is useful when the conditions are preset but when the conditions are determined during the execution of the programme, the while loop is most useful. It prevents the code within the loop from using the incorrect values. Assembly language has no error trapping. The programme must do the lot.

The following examples clear the screen. This is done by writing a space to each screen location. One uses a repeat-until loop and the other uses the while loop.

LI R0,>02FF START ADDRESS	LI R0,>0300	END OF SCREEN+1
LI R1,>2000 ASCII VALUE	LI R1,>2000	ASCII FOR SPACE
LOOP BLWP @VSBW WRITE	LOOP DEC R0	NEXT ADDRESS
DEC R0	JLT OUT	? FINISHED
JLT LOOP GOTO LOOP	BLWP @VSBW	WRITE
.	JMP LOOP	DO AGAIN
.	OUT	.

You will notice that the while loop is longer. You may also note that I counted down from the top of the screen image table to the bottom. The DEC instruction compares the result to zero automatically. This saves using the Compare Immediate instruction.

The basis for the loop is of course the ability to compare, and to choose the next instruction on the basis of the comparison. In BASIC the IF THEN ELSE statement does the same thing. In assembly, the flow of the programme is directed by the conditional jump instruction. In the above samples the conditional jump is saying 'IF THE SCREEN IS NOT CLEARED THEN DO AGAIN ELSE CONTINUE'. The conditional jumps are an important part of programming and should be understood thoroughly. If you don't then read about them in your E/A manual.

There are many times that several options must be tested for and the appropriate action taken for each. Consider testing for the arrow keys and enter in basic. the programme would look like this.

```

10 CALL KEY(0,K,S)
20 IF S<>1 THEN 10
30 IF K=13 THEN 1000 ! ENTER
40 IF K=11 THEN 2000 ! UP
50 IF K=9 THEN 3000 ! RIGHT
60 IF K=8 THEN 4000 ! LEFT
70 IF K=10 THEN 5000 ! DOWN
80 GOTO 10

```

In assembly this can be done in two ways. The first way emulates that of BASIC and the second uses the power of assembly in that we have direct control of the memory. This can be used to establish a list of possibilities and destinations. Compare the two (keyboard scanning will be covered at a later date so for the moment assume that the ASCII value of the key pressed is in register 1).

CI R1,13 ASCII VALUE	LI R2,LIST	POINT TO LIST
JEQ ENTER GOTO ROUTINE IF EQUAL	LOOP MOV *R2+,R3	GET ADD.
CI R1,11 ASCII VALUE	JEQ KEY	END OF LIST
JEQ UP ETC.	C *R2+,R1	TEST
CI R1,9	JNE LOOP	NEXT VALUE
JEQ RIGHT	B *R3	.
CI R1,8	.	.
JEQ LEFT	LIST DATA ENTER,13,UP,11	TABLE
CI R1,10	DATA RIGHT,9,LEFT,8	
JEQ DOWN	DATA DOWN,10,0	
JMP KEY GET NEW KEY		

The second version is a shorter programme and the list can be of any size. This sort of structure is similar to the CASE statement in many of the higher level languages. Assembly language allows for any data structure that the programmer can design. A little thought can produce some simple ways to emulate the higher level language statements.

Indexed addressing can be used to simulate the ON-GOTO that is found often in BASIC. Once again I will use the arrow keys to demonstrate the use assembly to emulate the BASIC statement.

```

10 CALL KEY(0,K,S)           AI R1,-8 VALUE IN R1
20 IF S<>1 THEN 10)         JLT KEY
30 IF K<8 OR K>11 THEN 10   CI R1,3
40 K/K-7                     JGT KEY
50 ON K GOTO 4000,3000,5000,2000 SLA R1,1 X2 FOR WORDS
                                B @LIST(R1) ON-GOTO
                                LIST DATA LEFT,RIGHT,DOWN,UP

```

Well thats about the end of the tutorial for this month. We have covered some of the principles involved in looping and looked at ways to emulate some of the higher level statements. Remember that you can exit a loop from anywhere inside it. I have only discussed the exits at the beginning and at the end. Remember also that if you forget to change the control variable inside the loop or if you jump to the instruction that sets the control variable instead of the beginning of the loop (a common mistake) then the loop will be infinite, no way out, a lock-up. Beware!

As usual, the answers to the last tutorial are provided, as are a few more problems. Go to it.

The Problem.

The question last month was to display the contents of register 3 on the screen. Lets build on that. Enlarge on that routine so that many values are displayed on the screen. That is, instead of setting register 3 to a particular value, move data from memory into the register then display it. Next get some more data and display it on the next line, and so on. Keep going until all 24 lines are used. To keep things standard, the area of memory that will be displayed will start at >A000.

The Solution.

The divide instruction uses 2 registers. The number is right justified in 32 bits. What does this mean? Lets divide 5 by 2 using DIV R0,R3. The value 2 is loaded into register 3 using the normal Load Immediate instruction. Register 0 needs a little more work. The number starts in register 0 but ends in register 1. The 32 bits really look like this:

```

0000 0000 0000 0000 0000 0000 0000 0101
      Register 0           Register 1

```

To do this we need to clear register 0 and load the value into register 1.

```

CLR R0
LI R1,>0005
LI R3,>0002
DIV R3,R0

```


As you know, registers can only hold whole numbers, so what happens to the answer of two and a half. This type of division is called modular division and is the same type that is taught in the earlier years of schooling. 5 divided by 2 equals 2, remainder 1. The dividend (2) is put into the left register and the remainder (1) into the right. Register 0 contains the value 2 and register 1 contains the remainder, 1.

Now take a look at the method of producing a string of numbers to represent a value. Lets use the number 123.

123/10 = 12 r3 (123 divided by 10 equals 12 remainder 3)

12/10 = 1 r2

1/10 = 0 r1

We stop when the quotient equals 0. Read the remainders in reverse order - 1, 2, 3. If we divided by 2 instead of 10, the answer would be in binary notation. Try that for 123 yourself and prove that it is true.

Here is the how your programme should look. Remember this is only one possible solution. If yours doesn't look the same, it doesn't matter to much as long as it works. No programme that works is wrong, its just that some are more efficient than others.

	DEF	START	DEFINE	ENTRY POINT
	REF	VMBW	I AM USING THIS UTILITY PROVIDED BY E/A	
START	LI	R4,BUFEND	POINTER TO BUFFER	
	LI	R1,>1000	NUMBER THAT WILL BE DISPLAYED	
	LI	R3,10	DIVISOR	
LOOP	CLR	R0	PREPARE FOR DIVISION	
	DIV	R3,R0	DIVIDE BY 10	
	SWPB	R1	BRING REMAINDER INTO THE MOST SIGNIFICANT BYTE	
	AI	R1,>3000	MAKE IT ASCII	
	MOVB	R1,*R4	STORE IN BUFFER	
	DEC	R4	POINT TO NEXT BYTE IN BUFFER	
	MOV	R0,R1	GET READY FOR NEXT DIVISION	
	JNE	LOOP	IF THERE IS SOMETHING LEFT THEN DIVIDE AGAIN	
	LI	R0,>0110	SCREEN ADDRESS FOR ANSWER	
	LI	R1,ANSBUF	WHAT TO WRITE	
	LI	R2,>0005	5 BYTES LONG	
	BLWP	@VMBW	DISPLAY ANSWER	
	LIMI	2	ENABLE QUIT KEY	
STOP	JMP	STOP	WAIT	
ANSBUF	DATA	>2020,>2020	BUFFER. USE >20 - ASCII FOR SPACE	
BUFEND	BYTE	>20	END OF BUFFER	
	EVEN		FORCE EVEN WORD BOUNDARY	
	END			

Well its a cold evening and my fingers are beginning to stiffen so I had best call it quits before they snap off completely.

Bye for now.....GARRY



REPRINTED COURTESY SAN DIEGO NEWS

SOFTWARE REVIEW: SPAD XIII By John D. Johnson

ONE OF THE FIRST things a student pilot learns is that the trick of flying is not taking off. It's landing. And so it is with the new flight simulator from Not Polyoptics. It is fashioned after the famous Spad XIII WWI French fighter.

DO NOT EXPECT to hitch up your joystick and zoom off in the clouds effortlessly blasting Fokkers. May, lad, you gotta learn how to fly first! No problem, Not Polyoptics have supplied a fine little manual which tells how and gives some interesting background on the SPAD. And even if you are a pilot, you have to get checked out on this machine.

ALSO DO NOT EXPECT Microsoft perfect graphics. The TI just doesn't have the memory to provide such displays. Nevertheless, if you have some imagination and patience, you can get a real idea of flight from this package. All, that is, except the sensations of gravity, "flying by the seat of your pants" that is. After I got used to the Spartan graphics, I was pleasantly surprised by this program.

IT PROVIDES SOME VARIETY (besides just dogfighting.) You can go fetch a spy from the destroyed town of Mareuil. Or fly over the Marne Hills and bomb big gun emplacements. Other diversions are flying to Paris to view le Tour Eiffel, or do aerobatics. Wow, was I surprised when I did a barrel roll upon taking off when I let the joystick get away from me. I vowed not to go dogfighting until I had learned to land. That took a few sessions. Then, when I did get in my first fight, I almost got my opponent -- but he put a bullet in my motor and I barely made it back to my field!

THE MANUAL Not Polyoptics provides gives hints on how to do loops, get out of spins and so on. It has an excellent keyboard template, and panel display.

THE RUDIMENTS OF FLIGHT are briefly explained. The battlefield map is also very helpful and can be viewed easily while flying.

DO EXPECT to spend time getting places. It takes about 10 minutes to get to the front lines and about 25 to get to the bombed town where you pick up the spy. If you can land on the jig saw puzzle streets! It has two actual hours flying time and you take that much time if you go on the longer missions!

THE EASIEST FLIGHT is to Paris to see the Eiffel Tower. Again, it takes several minutes of flying to get there.

The graphics of the tower are at first disappointing at best. There is no city, just the tower. But when you fly in close and circle it, it will loom up larger and larger on your CRT until it stands half the screen high and looks like a fair representation of the Eiffel Tower. Watch your gas, however, remember you need to get back to your base.

WORLD WAR TWO pilots had two sayings which are good to keep in mind and equally applicable to WWI:

"There are no old, bold pilots."

"Don't fly low and slow."

THAT LAST ONE is a good rule to follow in flying your Spad XIII. The enemy craft typically hover in the lower altitudes over the front lines. You can get the drop on them by going over the lines at altitude -- above 4,000 feet -- and then spiral down to engage them. Getting into dogfights over the front lines and near the German aero-drome put you in range of flak guns and you may want to disengage after a few tangles to avoid getting shot down. If you do get hit, get your nose down and try to jog the throttle to get back over your own lines. You can then land on open ground.

DON'T BE AFRAID to throttle back upon flying against the enemy. It is hard to keep the nose down otherwise. You have to get the bandit in your gun sights to hit him and if you're flying fast, it is hard to veer right and left, then get the nose down and also hit the fire and or "Q" button. It takes coordination but it can be done. You also have to constantly hit the 1 thru 6 keys to shift your view from front to back to either side and top and bottom.

ONE MINOR IRRITANT is the way the program ends if you crash or otherwise foul up. It simply goes to a blank screen and announces you should hit enter. You do and the disk is rebooted to start it up again. No, there is no gory crash! Also, if you lose a battle, you don't get to see what your score was. Despite all these short-comings, I still would not put it back in the hanger.

THE SOFTWARE requires EXB and 32K. It costs \$29.95. I got mine on sale, but I would pay that price! Vive le SPAD XIII. Vive la France. (I was just about to add, Vive Not Polyoptics, but it just doesn't sound right. Vive Not? Mais no, monsieur!) ***

"DOUBLE DENSITY" FROM YOUR TI DISK CONTROLLER

Reprinted from Hunter Valley 99ers

As reported in the April issue of our Newsletter, Secretary Albert had sent to Ryte Data for 2 EPROM chips that supposedly turned your TI Disk Controller Card into Double/Quad Density. Recently he received the chips (after a couple of months wait), and included was a sheet containing details of the chips. The following is a resume of those details.

BACKGROUND

The TI Disk Controller is capable of supporting up to 3 double sided single density disk drives. The total number of available sectors is 720 per disk.

Disk Manager II from TI was designed to support anything from 35 track drives all the way up to 80 tracks per side at double density. TI had designed a controller capable to supporting 4 double sided double density (DS/DD) 40 track or (DS/Quad Density) 77 track disk drives - to achieve a maximum of 630,784 bytes or 2464 sectors per disk! Obviously TI did NOT release this new disk controller.

With the pair of EPROMs in this package (from Ryte Data), which contain updated controller software, you can now support 80 track disk drives and achieve 1440 sectors per disk. This EPROM set will effectively provide 'double density' operation by using a QUAD density drive.

RYTE DATA 80 TRACK EPROM SET

1. The EPROM set is installed in the TI Disk Controller Card for the PE Box. You must remove the existing PROMs.

2. ALL other operations remain identical. You can mix drives and formats ie one 40 track drive and 1 or 2 80 track drives.

3. DM II from TI must be used to initialize the disks. Other disk manager programs will support the usual operations but not initializing disks for 80 tracks.

4. The data is written on twice as many tracks, therefore it is NOT compatible with other disk formats, ie Myarc or CorComp double density.

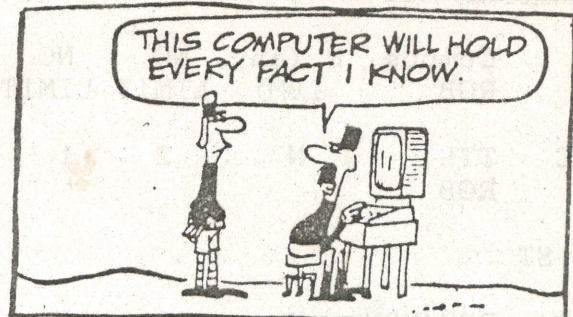
5. Half height 80 track drives are used to achieve the increase in storage capacity. These 80 track drives are listed as 96 tracks per inch (tpi) or as QUAD density.

6. You will be able to use these 80 track drives with the Myarc disk controller with QUAD density option installed, the same with any 40 track drives you already have.

7. This EPROM set has been tested with all these major software packages: TI Writer, MultiPlan, Extended Basic, Editor Assembler, etc. This does not guarantee that it is fully compatible with ALL software available, particular problems could possibly arise with software which accesses the disk controller on a direct level.

The bottom line is \$US45 for the 2 EPROMs. They are available from:-

Ryte Data
210 Mountain Street
HALIBURTON ONTARIO
KOM 1S0
CANADA



Comparison Charts

Based on maximum config of each machine

Machine	CPU RAM (K)	VDP RAM (K)	DISK DRIVES	HARD DRIVES	MOUSE PORT	Machine	Low Res	Colors	High Res	Colors
TI-99/4A	32	16	4 DS/QD	4 30mb	N	TI-99/4A	32x24	16	246x192	16
Myarc 9640	2024	192	4 DS/QD	4 30mb	Y	Myarc 9640	256x192	256 (512)	512x424	16
IBM PC	640	64	2 DS/DD	2 30mb	N	IBM PC	160x100	16	320x200	4
Atari ST 520	960	64	2 DS/DD	1 40mb	Y	Atari ST All	320x200	64 (512)	640x400	2
Atari ST 1040	960	64	2 DS/DD	1 40mb						
Apple 512 MAC	448	64	2 DS/DD	1 80mb	Y	Apple MAC All	512x384	2	512x384	2
Apple 512 PLUS	4032	64	2 DS/QD	1 80mb	Y					
Amiga	896	128	2 DS/QD	1 80mb	Y	AMIGA	320x200	64 (4096)	640x200	16

Note: 1 DS/DD Drive=360k
1 DS/QD Drive=720k

Note: Number in () is total Palette

Machine	Video Output	Keybrd Fixed?	RS232 Ports	PIO Ports	1000 \$\$	Machine	Text	Qual Output	Type
TI-99/4A	Compos RF Mod	Y	4	2	.4	TI-99/4A	40x24	2/3	Compos Color
Myarc 9640	Compos RGB	N (any IBM)	NO LIMIT	NO LIMIT	1.1	Myarc 9640	80x50	3/4	Compos Color RGB color/mono
IBM PC	TTL RGB	N	2	1	2.0	IBM PC	80x25	2/3	TTL RGB color/mono
Atari ST 520	RGB/MON	Y	1	1	1.2	Atari ST All	80x48	3/4	Only Atari Monitors
Atari ST 1040	RGB/MON	Y	1	1	1.5	Atari ST All	80x26	4	Built in
Apple 512 MAC	MONO ONLY	Y	2	1	2.0	Apple MAC All	80x26	4	Built in
Apple Mac PLUS	MONO ONLY	Y	2	2	4.5	AMIGA	80x25	1	RGB Color/ mono
Amiga	RGB/MON	Y	1	2	2.2				

Output Ratings
1-Poor 2-Good 3-Very good 4-Excellent

REPRINTED COURTESY SAN DIEGO NEWS

LOTS OF THINGS are happening in TI land:

ITEM: "Fortran is now available for the TI," was the announcement made by Rod VanOrden at our May meeting. Rod said a development package which includes a linker, compiler and debugger will be sold by Tenex Computer Express. See Tenex's Spring catalog. Rod commented, "This is a real step forward for the TI-99/4A!" The software's producer is listed as the LGMA Co. and the cost of the development package is \$50. So what good is Fortran? It's useful as a scientific development tool, notes Rod.

ITEM: The May issue of MICROpendium has an announcement that Texaments is offering TURBO-PASCAL. It lists \$79.95 as the software's price.

ITEM: Rod Van Orden again: Rod is compiling a c-99 Manual. This will be a complete listing of documentation files and program listings. It will amount to about 200 ++ pages of text and should be helpful to people who are struggling in learning our "small c" and want to converse with each other -- a sort of common reference.

ITEM: This may be a biggy. Tom Spillane of Digit Systems, a San Diego Co. producing an RGB Conversion kit (used for getting sharper video out of RGB monitors) has announced another product: An Advance Video Display Processor Card. Wha's that? It will be a card that fits in the P-box and provides more colors, more sprites, and 80 columns of text. Tom is using the same V9938 Yamaha chip that Myarc uses in its 9640 and which Mechatronics uses in its 80 col card, to achieve this advance. It will also make it possible to control or display VCR tape segments in programs. Home video fans will be able to install video overlays on their VCR tapes. Tom says an 80 column TI Writer will be made available. David Allen is working up demo software and is checking all existing TI software to see if it is compatible. To use this card, an analog RGB monitor* with an 80-column rating is required. Expected cost will be \$195. (*A Magnavox 8CM515 is an example of what is required.) So if the two other firms (Myarc and Mechatronics) are already providing this item, why should Digit? Tom says the card will fit in the P-Box whereas Mechatronic's is connected as a side-car attachment to the peripheral port. Also, that you have to buy the 9640 computer to get Myarc's. David remarks the graphics coordinates are simpler than the present TI chip the 9t9 now uses. Plain old x - y coordinates are used.

ITEM -- A low cost interface to connect a standard IBM keyboard to the 9t9's keyboard socket was being shown at the LA Fest West by Tom Freeman of the sponsoring LA group. Cost of the item was \$20 and it is being produced by a Rhode Island company. (More about this when we get the dope.)

CORCOMP has announced it is putting software on the market that will permit the 99-4/A to read IBM PC disks and then transfer the files to a TI disk. This was previously possible only with a null modem hookup between the computers or over a regular phone-modem hookup. No price is listed by CorComp. To use guessed it, CorComp disk controller is necessary along with two DSDD disk drives.

MYARC'S CALIFORNIA distributor, Les Merryman had a 9640 (Geneve on hand at the LA Fest West last month. He had very little to show on it. One machine had a multi colored dragon and was hooked up to a mouse. Using the mouse, one could move over the picture then click the fire button and zoom in on the graphics which were pretty good!

Bar chart

This little BASIC program comes from the Northern Alabama 99 Computer Users Group, Huntsville. It produces a multicolored vertical bar chart.

Here's the code:

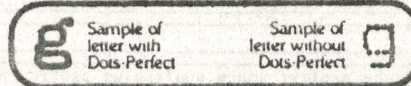
```
100 CALL CLEAR
110 FOR SET=1 TO 31 STEP 2
120 CALL COLOR(SET/2,8,8)
130 A=INT(RND)+1
140 CALL VCHAR(25-A,SET,SET*
8+31,A)
150 CALL VCHAR(24-A,SET+1,SE
T*8+31,A)
160 NEXT SET
170 FOR X=1 TO 16
180 CALL COLOR(X,X,X)
190 NEXT X
200 GOTO 200
```

Line 130 controls the size of the bars. Substituting the values in line 130 will change the height of the chart.

Introducing : Dots-Perfect™
 an upgrade kit for
 Epson® FX, JX,
 RX, and MX printers

sample of letters
 without Dots-Perfect
 ABCDEFG
 abcdefg

sample of letters
 with Dots-Perfect
 ABCDEFG
 abcdefg

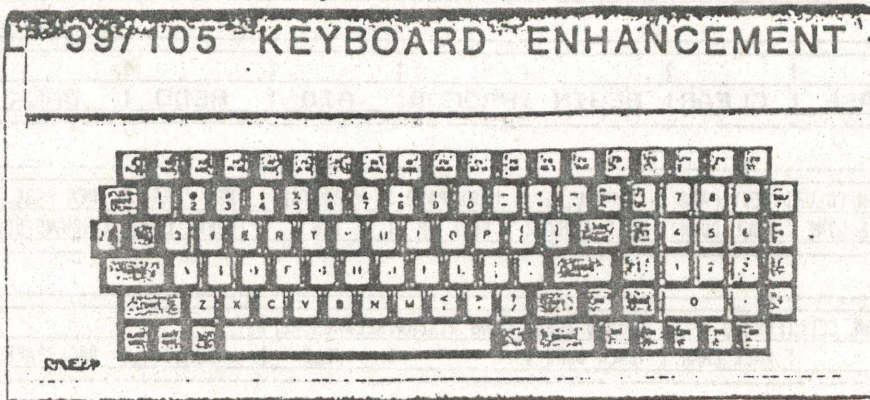


Dots-Perfect™ is here
 at Super Discount Prices
 Just \$63.00 + \$2.00 S/H
 For orders send check or money order to:

Myra D. White
 849 E. Bonita Ave.
 La Verne, CA. 91750
 (714) 592-6897 after 6PM PST

MODEL 99/105 KEYBOARD ENHANCEMENT - \$199.95

NEW!



INCLUDES custom keyboard.

THE FOLLOWING ARE STANDARD FEATURES ON THE MODEL 99/105 KEYBOARD ENHANCEMENT...

- * * INTERFACE LOGIC CARD WHICH INSTALLS EASILY IN CONSOLE WITHOUT SOLDERING.
- * * 105 KEY KEYBOARD WITH 24 FUNCTION KEYS.
- * * SUPPORTS SINGLE KEY ENTRY OF ALL TI-WRITER, MULTIPLAN, AND EDITOR/ASSEMBLER COMMANDS.
- * * ATTRACTIVE LOW PROFILE KEYBOARD WITH ADJUSTABLE TILT ANGLE AND 5 FT. CORD.
- * * DEDICATED NUMERIC KEYPAD WITH NUMERIC OPERATORS.
- * * DEDICATED CURSOR CONTROL KEYS.
- * * ENLARGED RETURN, SHIFT, AND CONTROL KEYS.
- * * OPTIONAL USER INSTALLED RESET AND LOAD INTERRUPT CAPABILITY FROM KEYBOARD.
- * * DETAILED INSTALLATION AND OPERATING INSTRUCTION MANUAL INCLUDED.

KEYBOARD INTERFACE CARD - \$149.95

- * * Interface card installs in 4A console in place of old keyboard and allows use of any IBM PC/XT compatible keyboard with your 99/4A. Interface card supports all 83 standard PC/XT key codes.
- * * Comes with complete, easy to follow installation and operating instructions. NO SOLDERING REQUIRED.
- * * Optional user installed RESET and LOAD INTERRUPT capability from keyboard. Complete instructions included.

NEW!

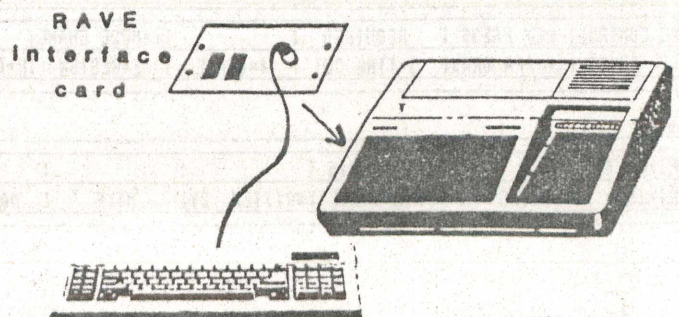
KIT AVAILABLE

- * * Includes printed circuit board, complete instruction manual, and RAVE 99 custom chips. Hard to find hardware also included. Requires only assembly \$ 92.00 and about \$10.00 in easy to find components to complete.

IBM & PC/XT are trademarks of International Business Machines

★ **RAVE 99** ★
 112 RAMBLING ROAD
 VERNON, CT 06066
 (203) 875-9833

EASY INSTALLATION



ADD 5% FOR HANDLING AND SHIPPING.
 CT. RESIDENTS ADD 7.5% SALES TAX.

MasterCard VISA & Master Card Add 3% VISA

Electric Circuits

REPRINTED COURTESY HUNTER VALLEY 99'ERS

PART 3 OF A PROGRAM

BY PAUL MUELVANEY

This is the third and final instalment in a series. It is not a stand alone program, as the character redefinitions were done previously. Load the previous parts then type the code, the last line of the previous part will be overwritten.

The program puts a typical Direct-On-Line (DOL) Motor Starter circuit on screen. The motor contacts are operated by the energising of the contactor coil C1. The coil is energised through the Start button, the Stop button and the Overload. Once the coil is energised it holds itself in through the interlock C1. The coil is de-energised by opening the Stop button or tripping the Overload. If the Overload trips, the coil cannot be started until it is reset.

The concepts used in this program can be adapted for use in fields other than Electrical, eg Hydraulics and Pneumatics. I hope these programs may be of use to others. If anyone develops the idea further, I hope they share the information with the TI community.

```
940 CALL CLEAR
950 !
960 REM MOTOR CIRCUIT
970 !
980 CALL CLEAR :: RESTORE 13
20
990 DISPLAY AT(3,5):"D.O.L.
MOTOR STARTER"
1000 FOR L=1 TO 3 :: READ R,
C,K,T :: CALL VCHAR(R,C,K,T)
:: NEXT L
1010 FOR L=1 TO 19 :: READ R
,C,K,T :: CALL HCHAR(R,C,K,T
):: NEXT L
1020 DISPLAY AT(8,4)SIZE(21)
:"START STOP O/L C1" ::
DISPLAY AT(15,5)SIZE(2):"C1"
1030 DISPLAY AT(20,4):"1 FOR
START, 2 FOR STOP" :: DISPL
AY AT(22,4):"3 TO TRIP OVERL
OAD"
1040 DISPLAY AT(24,4):"4 TO
RESET OVERLOAD"
1050 CALL KEY(0,N,S):: IF S=
0 THEN 1050 :: IF N<49 THEN
1050 :: N=N-48 :: IF N>4 THE
N 1050 :: ON N GOTO 1090,114
0,1190,1050
1060 !
1070 REM MOTOR START
1080 !
1090 RESTORE 1390 :: FOR L=1
TO 16 :: READ R,C,K,T :: CA
LL HCHAR(R,C,K,T):: NEXT L :
: CALL VCHAR(13,10,97,5)
1100 FOR L=1 TO 3 :: READ R,
C,K,T :: CALL HCHAR(R,C,K,T)
:: NEXT L :: GOTO 1050
1110 !
1120 REM MOTOR STOP
1130 !
1140 RESTORE 1450 :: FOR L=1
TO 13 :: READ R,C,K,T :: CA
LL HCHAR(R,C,K,T):: NEXT L :
: CALL VCHAR(13,10,110,5)
1150 FOR L=1 TO 6 :: READ R,
C,K,T :: CALL HCHAR(R,C,K,T)
:: NEXT L :: GOTO 1050
1160 !
1170 REM OVERLOAD TRIP
1180 !
1190 RESTORE 1510 :: FOR L=1
TO 12 :: READ R,C,K,T :: CA
LL HCHAR(R,C,K,T):: NEXT L :
: CALL VCHAR(13,10,110,5)
1200 FOR L=1 TO 4 :: READ R,
C,K,T :: CALL HCHAR(R,C,K,T)
:: NEXT L
1210 !
1220 REM OVERLOAD RESET
1230 !
```



```

1240 CALL KEY(0,N,S):: IF S=
0 THEN 1240 :: N=N-48 :: IF
N>4 THEN 1240
1250 IF N<4 THEN 1260 ELSE 1
270
1260 DISPLAY AT(6,2)BEEP:"TH
E OVERLOAD MUST BE RESET" ::
GOTO 1240
1270 FOR L=1 TO 4 :: READ R,
C,K,T :: CALL HCHAR(R,C,K,T)
:: NEXT L :: GOTO 1050
1280 !
1290 !
1300 REM MOTOR CIRCUIT
1310 !
1320 DATA 6,4,97,14,6,29,120
,14,13,10,110,5,12,5,98,2,11
,7,112,1,10,8,112,1,12,9,104
,1,12,10,109,1
1330 DATA 12,11,104,2
1340 DATA 12,13,128,2,12,15,
104,4,12,19,128,2,12,21,104,
4,11,25,33,1,12,25,35,1,11,2
6,34,1,12,26,36,1
1350 DATA 12,27,121,2,18,5,9
8,2,17,7,112,1,16,8,112,1,18
,9,104,2
1360 !
1370 REM START
1380 !
1390 DATA 10,8,32,1,11,7,32,
1,12,7,128,2,12,9,98,1,12,10
,99,1,12,11,98,2,12,15,98,4,
12,21,98,4
1400 DATA 12,25,102,1,11,25,
100,1,11,26,101,1,12,26,103,
1,16,8,32,1,17,7,32,1,18,7,1
28,2,18,9,98,2

```

```

1410 DATA 12,7,32,2,11,7,112
,1,10,8,112,1
1420 !
1430 REM STOP
1440 !
1450 DATA 12,13,32,2,11,13,1
12,1,10,14,112,1,12,15,104,4
,12,21,104,4,12,25,35,1,11,2
5,33,1,11,26,34,1
1460 DATA 12,26,36,1,18,7,32
,2,17,7,112,1,16,8,112,1,18,
9,104,2
1470 DATA 12,9,104,1,12,10,1
09,1,12,11,104,2,10,14,32,1,
11,13,32,1,12,13,128,2
1480 !
1490 REM OVERLOAD TRIP
1500 !
1510 DATA 12,19,32,2,11,19,1
12,1,10,20,112,1,12,21,104,4
,12,25,35,1,11,25,33,1,11,26
,34,1,12,26,36,1
1520 DATA 18,7,32,2,17,7,112
,1,16,8,112,1,18,9,104,2
1530 DATA 12,9,104,1,12,10,1
09,1,12,11,104,2,12,15,104,4
1540 !
1550 REM OVERLOAD RESET
1560 !
1570 DATA 10,20,32,1,11,19,3
2,1,12,19,128,2,6,1,32,32

```



**TEXAS INSTRUMENTS
BRISBANE USER GROUP**
P.O. BOX 57 ASPLEY QLD. 4034



A1987MPL00 277
G MORRIS,
P.O. BOX 371,
CLEVELAND, QLD. 4163.