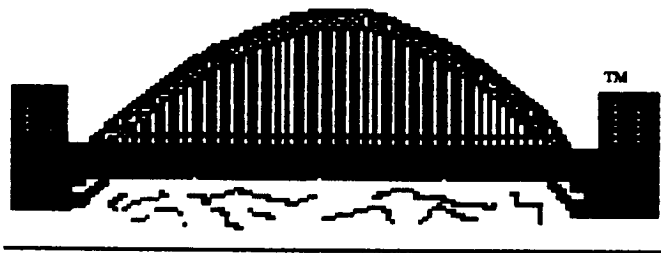
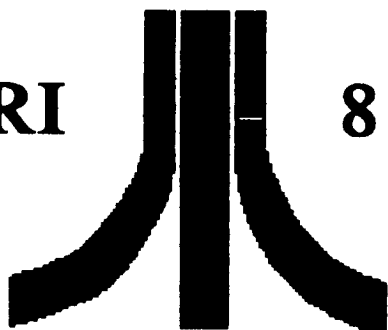


# *TYNE & WEAR*



ATARI



8 BIT

# *USER GROUP*

Issue 21

May/June 1996

# TWAUG NEWSLETTER

## Reminder

TWAUG NEWSLETTER is published bi-monthly, around mid-month of (Jan, Mar, May, July, Sept and Nov.)

It is printed and published by TWAUG, no other publishing company is involved.

Opinion expressed by authors, in this newsletter, is their own opinion and do not represent the views of TWAUG.

The Atari Fuji symbol and Atari name are the trademarks of Atari Corporation. The Fuji symbol on the front cover, is for informational purpose only.

TWAUG is entirely independent and is in no way connected with Atari Corporation or any associate company.

also the family commitments. With all this workload on John's shoulders, it is really a miracle that John managed to get through all that work, with some delays of course.

Well, it should be a lot easier now, we have a enthusiastic helper again, a volunteer, who is willing to dedicate his spare time to the Atari 8-bit cause and help us to keep the newsletter going.

Hopefully, in future, all correspondence will be dealt with much quicker than of late, as well as the PD orders sent out quickly too. We therefore apologize for any delays in dealing with your correspondence and PD orders that occurred in the last few months.

From now on, when in doubt and need to contact a TWAUG member by phone, do not contact John but contact:

Alan Turnbull on 01670 - 822492

and Max is still available on:

0191 - 586 6795

## Feed Back

Give us your support, write to us and give us your views on what you'd like to see in the newsletter, or write an article it will be most welcome, we do need your help.

TWAUG

PO BOX 8, WALLSEND

TYNE & WEAR, NE28 6DQ

---

## NEWS...

The TWAUG team is back to full strength. Since David's sudden death back in October 95, we at TWAUG have been working hard sorting through all the disks David dealt with. John worked really hard, trying to get acquainted with the PD library, answering the correspondence and sorting through nearly three thousand disks plus holding down a full time job. John's job is very demanding, he also has to work a lot of overtime, and there is

# TWAUG NEWSLETTER



## PUBLISHING!

This new look newsletter is set up with the Desktop Publishing program "TIMEWORKS 2", on the Mega 1 ST with 4 meg memory. Files are transferred to the ST with the Black Box transfer utility, as ASCII files. Those files are then imported into the DTP and printed with the Star/LC24-100 at 360dpi, the result is in front of you.

TWAUG

NEEDS



YOU

### TWAUG subscriptions

Home	1 Copy	£ 2.50
- DO -	6 Copies	£12.50
Europe	1 Copy	£ 2.50
- DO -	6 Copies	£13.50
Overseas	1 Copy	£ 3.50
-- DO --	6 Copies	£16.00

**Issue 22 is due mid-July**

## ISSUE CONTENT

REMINDER .....	2
CONTRIBUTIONS & CONTENT .....	3
DON'T LET BASIC BUG YOU .....	
Tutorial in Basic by Mike Bibby.....	4
130XE UPGRADE TO 320 .....	
by Scott Peterson.....	11
GAMES REVIEWS .....	
by Kevin Cooke .....	13
FOR SALE SECTION .....	17
MAIL BAG SECTION .....	18
B - TAPE .....	
Upgrading Data-corder .....	
by Jiri Bernasek.....	19
SECTOR SKEW - HIGH SPEED SIO ...	
What is Sector Skew ? .....	
by Bob Woolley .....	30
DISK CONTENT .....	33
USER GROUPS ADVERTS.....	
for LACE & OHAUG .....	34
ADVERTISEMENT .....	
for CHAOSI COMPUTER.....	35
ADVERTISING .....	
MICRO DISCOUNT .....	36

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU

**W**e saw last month how to label strings with variables. This meant that if we were using a string several times in a program we could use a variable instead of it.

For example:

A\$="AUSTRALIA"

means that, from now on, instead of

using "AUSTRALIA" in full in our programs, we can use A\$.

PRINT A\$

will print out AUSTRALIA for you. Of

course we

had to make room for the string by telling the Atari its maximum size with a DIM statement.

The labels we used last month were all single letters of the alphabet followed by \$. The dollar sign tells the computer that it is a

string we are labelling - such a variable is called a string variable.

**Expand your knowledge of programming with PART FOUR of MIKE BIBBY's guide through the micro jungle**

It is called a variable because the "contents" of a variable (in technical terms, its value) can vary throughout a program. Program I should illustrate the point.

As you will see when you RUN it, the value of

A\$ varies as we reassign it during the

program.

A\$ always takes the

## Now let's get these

last value assigned to it.

```
10 REM PROGRAM I
20 DIM A$(9)
30 A$="AUSTRALIA"
40 PRINT A$
50 A$="AMERICA"
60 PRINT A$
70 A$="AFRICA"
80 PRINT A$
```

Program I

You may wonder why on earth you would want to use the same variable for different things, rather than label everything separately. As we shall see, it can be extremely useful.

## down to work

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

So far we have restricted our string variable to single letters of the alphabet followed by the \$ sign, such as A\$, B\$ and C\$.

However there is no need for such a limit - provided we follow them with \$. String variables can be made up of several letters, even words. They must, however, be capitals.

Program II illustrates the point. It is our most sophisticated program to date, and is well worth having a close look at.

Incidentally, remember to enter NEW between programs.

Perhaps the first thing to remark upon is that our string variables,

grown into actual words. They've still got the \$ at the end, though, to show that they're string variables, or labels.

Also, notice that while our labels are in capitals, the strings themselves, inside the quotes, are a mixture of lower and upper case. You'll need some deft manipulation of the Caps key as you type it in.

As you'll probably remember, the PRINT CHR\$(125) of the line 30 clears the screen. It is good programming practice to use words for variables, since we can make the label describe what it is labelling. Programs make more sense this way.

```
10 REM PROGRAM II
20 DIM NAME$(20),FACT$(20),THREAT$(20)
30 PRINT CHR$(125)
40 NAME$="Mr. Smith"
50 FACT$="You owe me money."
60 THREAT$="Pay up or else."
70 PRINT
80 PRINT "Dear ";NAME$
90 PRINT FACT$;THREAT$
100 PRINT "cordially yours,"
110 PRINT "Mike"
```

Thus we use NAME\$ to label "Mr. Smith", FACT\$ to label "You owe me money", and THREAT\$ for "Pay up or else".

This may seem long-winded, but it really does help to make your programs more readable, and hence easier to decipher. For example:

```
80 PRINT "Dear "NAME$
really tells you what the line
is doing, far more than:
```

```
80 PRINT "Dear "A$
```

Program II

Instead of being single letters, have

Similarly:

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

### PRINT THREAT\$

is more meaningful than

### PRINT B\$

The moral is, use words for variables (labels) as much as possible.

Actually, you can use capital letters and numbers intermixed for variable names. For example:

### NAME1\$

### R2D2\$

### C3P0\$

are all valid string variables.

However they must start with a letter - not a digit - and only capital letters are allowed. This means that:

### 1DAY\$

### 2MORROW\$

aren't valid.

Also, spaces aren't allowed, so:

### FIRST NAME\$

is illegal.

Variables shouldn't start with Basic keywords, as they confuse the Atari, so:

### PRINTER\$

is definitely out.

Try entering a program line such as:

### 10 PRINTER\$="EPSON"

Then LIST it - can you explain what happened? Steer clear of keywords

in variable names.

While we're at it, try entering:

### 10 WRITER\$-EPSON

Spot the deliberate mistake? Well, the Atari does and rejects the line - EPSON should have been in quotes. If you now enter LIST, you'll see the Atari has actually included line 10 as a program line - with ERROR in front of it.

This habit of the Atari can be rather irritating, but don't forget, you can get rid of a line by simply typing its number and pressing Return.

Although it's not likely to affect you at this stage, the Atari limits you to 128 variable names. The good news is that they can each be up to 120 characters long.

One advantage of using variables instead of directly using strings is that we can easily alter the output of the program.

In the case of Program II, if we want another victim to be the recipient of our letter, just change line 40. For example:

### 40 NAME\$="Mr.Jones"

From then on all uses of NAME\$ in the program will refer to Mr.Jones.

In this program it doesn't make a great deal of difference, but in larger ones, if you had used the string "Mr.

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

Smith" every time, instead of NAME\$, you would be in for a lot of retyping.

So far we have talked about string variables. However there is another kind of variable called a numeric variable.

Numeric variables are labels just as much as string variables are, only they label numbers in such a fashion that we can do sums with them. Try running Program III.

Line 30 uses the numeric variable A to label the number 10. Notice that for a numeric variable we can simply use a letter of the alphabet without following it with the \$ sign necessary for a string.

Also since it isn't a string, the value we are giving the variable doesn't have to be in quotes. Hence line 30 is simply:

```
30 A=10
```

```
10 REM PROGRAM III
20 PRINT CHR$(125)
30 A=10
40 PRINT A
50 PRINT 2*A
```

### Program III

Line 40 prints out, not A, of course, but the value that A labels, which is 10.

The most interesting part is line 50. Here we multiply the number that A labels by two, so that the line prints out 20.

That's the useful thing about numeric variables - you can do sums with them!

Try running Program III with the following versions of line 50:

```
50 PRINT A+8
```

```
50 PRINT A/4
```

```
50 PRINT A*A
```

```
10 REM PROGRAM IV
20 DIM A$(10)
30 PRINT CHR$(125)
40 A$="10"
50 PRINT A$
```

### Program IV

If you've been following what I've said so far you could be forgiving for thinking that string variables are for labelling words, and numeric variables for numbers.

Life is never that simple. You can, and often do, use string variables for labelling numbers - the point is that you can't do sums with them. Try entering Program IV, which is based on Program III, using the string A\$ instead of the numeric A.

Once you've entered it, try adding the following line:

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

### 50 PRINT 2\*A\$

As you'll soon find out, the Atari rejects line 50 out of hand. This is because you are attempting to do a sum with the wrong type of variable - string instead of numeric.

As with string variables, we do not have to (and should not) restrict ourselves to single-letter labels for numeric variables.

We can use words in a manner strictly analogous to string variables, save that we omit the final \$ sign. And, of course, we don't put what we are labelling in quotes, since it isn't a string.

Have a look at Program V. This is meant to be a cheery greeting for

```
10 REM PROGRAM V
20 PRINT CHR$(125)
30 DIM NAME$(10)
40 NAME$="MIKE"
50 PRINT "GOOD TO SEE YOU, ";NAME$
```

### Program V

someone when they RUN the program in the computer - the sort of thing I often used in my classes.

However as it stands it's a bit restricted - after all, only a small percentage of my students were called MIKE. What's really needed is some way for the Atari to find out the

name of the person so that it can tailor the message to suit.

Program VI fits the bill. The trick here is the use of the INPUT

```
10 REM PROGRAM VI
20 PRINT CHR$(125)
30 DIM NAME$(10)
40 PRINT "WHAT IS YOUR NAME";
50 INPUT NAME$
60 PRINT
70 PRINT "GOOD TO SEE YOU ";NAME$
```

### Program VI

statement in line 50. In Program V, line 40 put the value MIKE into NAME\$. In Program VI the variable isn't actually attached to a specific value - if you like, but neglect to tell it what it's labelling. Instead you type:

```
50 INPUT NAME$
```

When the Atari reaches this line it waits until you PUT IN, or INPUT, the value you want NAME\$ to have by typing the

value in.

To put it another way, when the computer meets an INPUT statement followed by a variable, it asks you what you want the variable to be - in fact, it actually puts a question mark on the screen.

You are then supposed to type in the



# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

answer followed by Return, which, as always, sends it to the computer, which then carries on with the rest of the program.

So when you run the Program VI line 40 asks: "WHAT IS YOUR NAME". Notice that we don't need a question mark - the INPUT statement of line 50 supplies that.

The micro then waits for us to type our reply and send it by pressing Return. Whatever we have typed in then becomes the value of NAME\$ - even if we have lied!

Line 70 then prints out the message after line 60 prints out a blank line.

The point of all this is that in Program VI, as opposed to Program V, the value of NAME\$ is not fixed

```
10 REM PROGRAM VII
20 PRINT CHR$(125)
30 PRINT "How old are you";
40 INPUT AGE
50 PRINT
60 PRINT "I don't believe you are "; AGE
```

### Program VII

Initially, but is decided during the program by the respond to INPUT.

This means that every student in the class can now run the program and have the message tailored to themselves.

Incidentally, line 40 is not strictly necessary, but it is only polite to tell people what kind of response you expect them to make. Otherwise they will be met with just a question mark - not too "user-friendly" as the jargon has it.

The semi-colon at the end of line 40 "glues" the question mark, or prompt, as it is known, to the preceding "message". Running the program with it omitted should make this clear.

Remember, when you run Program VI and it asks for your name, you must type your reply then press Return. If you omit Return the Atari won't receive your answer and will continue waiting. This could be incredibly boring!

If you make a typing mistake before you press Return, you can erase it with Delete. Once you've pressed Return, though, you're stuck with what you've typed.

You can use INPUT with numeric variables as well as strings. Program VII demonstrates this.

When you get the prompt, try typing in a word rather than a number and see what happens.

A slightly more serious application of INPUT allows you to calculate the product of two numbers, as Program VIII demonstrates.

# TWAUG NEWSLETTER

## DON'T LET BASIC BUG YOU continued

Look carefully at line 70 and see if you can work out what's happening. FIRST isn't in quotes, and so the

quotes, so the sum is done and the answer printed out. Figure 1 should help to make this clearer.

```
10 REM PROGRAM VIII
20 PRINT CHR$(125)
30 PRINT "First Number";
40 INPUT FIRST
50 PRINT "Second Number";
60 INPUT SECOND
70 PRINT FIRST;" multiplied by ";SECOND;" is ";FIRST*SECOND
```

Finally, try altering Program VIII so that it adds or subtracts pairs of numbers.

### Program VIII

micro will print the number that FIRST labels. "Multiplied by" is printed literally since it is in quotes.

The numeric variable SECOND is not in quotes - it may have them on either side, but the quotes on the left are already paired with the quotes on the far left, so they don't count. The micro will therefore print out the value of SECOND.

"is" is printed literally, since it is in quotes. FIRST\*SECOND isn't in

We've covered an enormous amount of ground in this issue. I suggest that you spend a good while going over the programs. If you are having problems, re-reading the earlier articles will probably help.

Above all, remember it's a "hands-on" course - you can't expect the examples to make sense until you've typed them in!

```
70 PRINT FIRST ; "multiplied by"; SECOND ; " is "; FIRST*SECOND
```

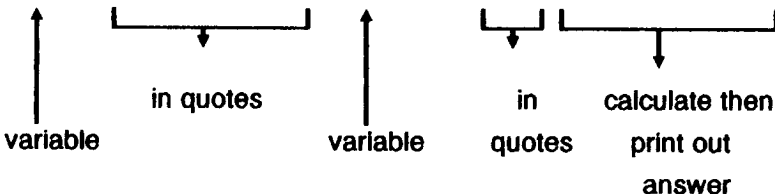


Figure 1: Mixing variables and strings in PRINT statements

# TWAUG NEWSLETTER

## THE 130XE - 320K UPGRADE

The 130XE/320K upgrade  
by Scott Peterson  
(downloaded from SIG-ATARI)

**A**fter both reading and building both the 800/288K upgrade (D. G. Byrd), and the 800XL/ 256K upgrade (C. Buchholz), I decided that there also had to be a way to upgrade the 130XE. There is, and thanks to the "Freddie" chip (CO61991) this modification is much easier to do than either of the other upgrades. To do the upgrade you will need a soldering iron, de-soldering tool, and some fine wire. See the parts list for the chips needed.

First, remove both the case and the metal shield to get down to the mother-board. Then remove the eight ram-chip U26 thru U33 (MT4264). They are the row closest to the TV RF module. Next, install Z2 thru Z9 in the place of U23 thru U33. These are the 256K ram-chips. You can solder them to the mother board, or use sockets. Now take a piece of wire approximately 12 in. long and run a jumper from pin one on each of the 256K ram-chips to the next. After you do this the wire will be connected to pin 1 on Z2 thru Z9 and you should have about 6 inches left over. Do this on the rear of the mother board and then snake the

wire thru the large hole near the ram chips. Next, desolder and remove U23 (CO14795), and replace it with a 40 pin socket.

Bend up pins 15 and 16 on U23 and insert it in the socket you just installed.

Take Z1 (74LS158) and bend up all the pins on it except pins 8 and 16.

Put this "piggy-back" on top of U20 (HD14050) and solder pins 8 and 16 of Z1 to pins 8 and 16 on U20.

Now solder a short jumper from pin 15 on Z1 to pin 8 of Z1.

Now, take a piece of wire about 4 in. long, solder one end to pin 30 on the chip marked "CO14805" on the mother board, and the other to pin 1 on Z1.

Next solder a wire to pin 15 (one of the two you bent out) of U23 and connect the other end to pin 2 on Z1.

Solder a wire to pin 16 on U23 and connect the other end to pin 3 on Z1.

Take R1 (33 ohm) and trim the leads to about 1/4 in.

Take the wire you connected to pin one on the 256K ram-chips and solder it to one end of R1, solder the other end of R1 to pin 4 on Z1.

Re-assemble the RF shield and case and you are finished.

# TWAUG NEWSLETTER

## THE 130XE - 320K UPGRADE continued

### ● PARTS LIST:

- ☆ Z1 74LS158 (2 to 1 Multiplexer)
- ☆ Z2-Z9 41256 dynamic RAM (150ns)
- ☆ R1 33 ohm 1/4 watt resistor.
- ☆ 1 40 pin socket.
- ☆ 8 16 pin sockets (optional).

The next page is a quick over view of the bit table and numbers to be used in location 54017 (PORTB). I have finished modifying a ramdisk handler for the extra ram. It uses a ram based OS so BASIC XE or XL can't be used. At present the best deal for this modification is to use MYDOS 4.0. This supports a very large single density ramdisk. With BASIC XE you can use a 1500 sector ramdisk and without it you can have about 2000 sectors. This upgrade has been built and tested on a BBS, it has run for days on end without a memory loss or error. If you need help or more information feel free to call the Peanut Gallery (408)-384-3906. 24HR, 300/1200 Baud. Leave mail to the Sysop (thats me). Good luck and let me know if you write a better handler.

Memory Control Register 54017  
(\$D301) 130XE/320K

Bit 7 6 5 4 3 2 1 0

D a b C c d B R

D=0 enable diagnostic ROM.

B=0 enable BASIC ROM.

R=1 enable OS ROM.

C=0 enable extended RAM.

abcd= memory control bits.

Bank #	Control #
Bank 0 ----->131	
Bank 1 ----->135	
Bank 2 ----->139	
Bank 3 ----->143	
Bank 4 ----->163	Basic= off
Bank 5 ----->167	OS = on
Bank 6 ----->171	ENH = on
Bank 7 ----->175	
Bank 8 ----->195	
Bank 9 ----->199	
Bank 10 ----->203	
Bank 11 ----->207	
Bank 12 ----->227 <--\	
Bank 13 ----->231  \	
Bank 14 ----->235  /130XE Banks	
Bank 15 ----->239 <--/	

If you are using MYDOS 3.016 and wish to use BASIC XE and a ram-disk at the same time, boot DOS and POKE 5275,163 and

# TWAUG NEWSLETTER

THE 130XE - 320K  
UPGRADE continued

5324,16. Go to DOS and write the new DOS. This will keep the two from "bumping" into each other. A similar poke can be done to DOS 2.5, it is POKE 4838,163. The handler I have will set up 192K of the extra ram as 2 SD ramdisks or 1 DD ramdisk. If you are a hot-shot programmer (I'm not), I think a print spooler that uses part of this ram would also be very nice. This mod is easy to do and perfect for running a BBS. One note, on compuserve there is a mod by Rich Andrews which should not be confused with this one, his uses 33 new chips and mine uses 9 new chips.  
Have fun. -Scott Peterson

## By Kevin Cooke

**Y**ep, it's time again, back for some more game reviews! Again, this one may not make it in time for the next issue of the magazine (issue 21) so if it doesn't, I apologise - still loads of college homework being piled upon me! Anyway, here goes once again!

# REVIEWS

## Title:

WLOCZKIJ "THE VAGABOND"

Sold by: Micro Discount,  
265 Chester Road, Streetly,  
West Midlands B74 3EA,  
ENGLAND.

Tel: 0121 353 5730

Price: 4 Pounds (+ P&P)

**I**n issue 17, Fred Meijer reviewed this piece of software. However, it's so good that I can't resist taking another look at it. Vagabond has, apparently, just got back home after being on a "big adventure" when his door bell rings. Answering it, he meets an alien who tells him that his space ship has crash landed. None of the alien occupants from it are injured but they are scattered over the Earth. Never one to turn down an adventure, Vagabond decides to help the alien. The disk first loads up a large picture of Vagabond (a small egg-like creature bearing a resemblance to "Dizzy" from a series of well known games on most other computers).

## REVIEWS

Next the main (text only title screen appears with a lively, well composed piece of music playing in the background. A press of the fire button brings up the first screen of this platform game.... and what a screen it is!!! Graphically, Vagabond is so impressive that you may almost.... well, do whatever it is that you do when you get considerably impressed!

The game looks STUNNING with around 6 colours (yes, colours, not shades of one colour!) being used on every different level! That's more like it! All of the sprites are VERY well defined, each being excellently animated and all looking as cute and cartoony as Vagabond himself (assuming he is a man - no offense intended, ladies!).

Graphically, the game just cannot be faulted. "But surely a game which is this graphically good cannot be playable?" I hear you ask! Wrong! Vagabond, in fact, has the playability of.... oh, all right then, it's just damn playable! The screen scrolling is smooth, enemies can be killed by jumping on their heads (yes, just like

in a console game!) and the difficulty curve is just right, making the levels a challenge without being too easy or too difficult. Music plays throughout the game (or suitable, sparse sound effects, if preferred, can replace the music) whilst the screen scrolling is excellent.

Excuse me if I don't say any more about Vagabond except IF YOU DON'T BUY THIS, YOU'LL REGRET IT. It could well be my favourite game of all time!!!

### Title:

ZONE PATROL

Sold at: Micro Discount (see above for address)

Price: 4 Pounds (+ P&P)

**T**his game was originally due for release by Atari but, as happened more than didn't happen, they decided to pull the plug on it. So did they have a good reason to do so? Zone Patrol is basically a Defender-style shoot 'em up, much like the more well-known Dropzone.

# TWAUG NEWSLETTER

## REVIEWS

As soon as you first press the fire button on your joystick, you may get a shock - a burst of very high quality digitised speech says "get ready" before the game screen appears. This nice interlude adds an air of quality to the game.

The main game screen is a little basic, albeit fairly colourful. Your character is a largish man with a jet-pack attached to his back. By moving the joystick, he can fly around quite freely. Gravity does take effect upon him however in that, once you stop moving the joystick, his jet-pack stops and he falls back safely to the ground.

Pressing the firebutton sends a lazer beam from his gun. Yes, you've guessed it, there's something for him to shoot at! Scattered randomly around the horizontally scrolling landscape are a number of aliens which it is your task to shoot. Due to the speed at which your man moves, you'll need fast reactions to stop flying and shoot the allens in plenty of time.

Three different speeds of play are selectable from the title screen. On

the fourth wave of allens (and presumably at regular intervals after that), a much larger flying saucer appears on the screen, ready to be destroyed.

I have to admit that I can't get past this level -those allens get me every time.

At the top of the screen you are informed of the number of lives which you have remaining, the number of smart bombs left (which destroy everything on the screen) and the number of allens left to shoot on that level.

At first, I wasn't going to recommend this game - it didn't seem as if there was an awful lot to do and, after playing Vagabond, it was a bit of a let down. However, I gave it a fair chance and ended up enjoying it quite a lot. Whilst not quite as good as Dropzone due to the lesser action element, it is very addictive.

Overall, not a classic but certainly one to consider adding to your collection.

## REVIEWS

### Title:

#### TOMMINGS DEMO DISK

Sold at: Micro Discount (see above for address)

Price: 1 Pound (+ P&P)

**T**ommings, in case you are wondering, was a Lemmings clone which, up until a very short time ago, was due to be released commercially. Unfortunately, the programmers decided to leave the Atari scene and the disk was never totally finished. Now, Derek Fern has released a demo disk of the program for us all to try.

If you've ever played Lemmings, you'll know what to expect from this. The object is to guide the small lemmings from the entrance to the level of the exit. Unfortunately, as lemmings are apparently stupid, they'll carry on walking constantly.... that includes over cliffs, into fire, into water, etc! It is your task to assign lemmings with certain skills to prevent the others from being killed. Skills include blocking other lemmings, digging through rock,

building bridges, etc. but you can only assign these skills to a limited number of lemmings per. level.

Graphically (excluding the very drab colourscheme), Tommings is far better than The Brundies. Every lemming is a much bigger sprite with far more animation. Everything is just a lot more clear, allowing you to see exactly which lemming is doing what, rather than guessing.

Tommings lets down a little due to the fact that mouse control is not supported and there is virtually no sound. However, for 1 pound you can hardly complain, can you?!! Overall, it's very good indeed.

Last time I spoke to him, Derek Fern said that he has a level editor for Tommings and is attempting to work out how to use it in the hope of releasing a finished disk - currently, the disk only has something like 5-10 level. If you are interested in seeing a finished version, why not purchase a copy of this demo to show your support. Who knows, maybe we will see a finished version in the near future.



## REVIEWS

### Title:

UFO DEMO DISK

Sold by: Micro Discount (see above for address)

Price: 1 Pound (+ p&p)

**T**his disk is a demo of yet another game which was due for release but unfinished.

Side A contains a demo of the game (albeit without sound) with the options (normally available by pressing a key on the keyboard) not working. The game seems very much in the mould of star raiders or Star Raiders 2 only with better graphics. When you shoot, instead of firing a laser, heat seeking missiles fly through space at your target.

What makes the game so nice to look at is the fact that enemy ships are all 3D, yes, made with polygons for more realism.

Side B of the disk contains the loading screen which was due to appear on the game, + a VERY good demo in which each different type of space ship which features in the

game comes floating towards you, rotating around before it flies away.

If you think games like Starwing on the Super Nintendo look good, this shows that the Atari Classic could probably do the same thing! Overall, if you are interested in seeing what the Atari Classic can do or even if you collect software, this disk is an essential purchase!

That's all for this time, folks! I'll see you next time (metaphorically speaking, of course!) for some more reviews. Hammer that fire button in the meantime!!!

## FOR SALE

---

Modem WS4000, Miracle Technology with manual and power unit, cable to use with Black Box.

40 pounds plus p & p

Contact Max on 0191 - 586 6795

# TWAUG NEWSLETTER

## FOR SALE

1050 disk drive with SPEEDY fitted.  
Write protect switch. Complete with  
documentation, software and all  
leads.

50 pounds plus p&p

1050 disk drive with HAPPY fitted.  
Complete with software and all  
leads.

45 pounds plus p&p

1050 disk drive with U.S. Doubler  
fitted. Complete with all leads.

40 pounds plus p&p

Hayes 1200 Smartmodem. No  
manual but easy to use.

25 pounds plus p&p

Contact John Tel 0191 262 6897



Thank you so much Alex for your very  
nice letter, in opposite column.

Yes, we were thinking hard ourselves,  
wondering if the newsletter would fold,  
after David's death. But John and I had  
a really long discussion and decided that  
it would be a shame if, after all the  
work we'd done and experience gained  
was for nothing and besides, we owed it  
to David. We are also pleased that you  
like the new look missive, I always look  
for new ideas to improve the layout.

Max



## MAIL BAG

Dear Max and John,

I can't tell you what a shock it was to  
learn of the death of DAVID EWENS.  
We had developed a real comfortable  
relationship, aiding in the improvement  
of both THE OL' HACKERS A.U.G.  
and T.W.A.U.G. clubs. I know you both  
have suffered a great loss with his  
passing, but I must write this letter to  
commend you both for carrying on and  
doing it so well too.

I enjoy and look forward to each of  
your newsletter productions and read it,  
every single word. Your efforts along  
with a few other dedicated people, are  
the fire that keeps the old 8 BIT engine  
chugging along. You prove your  
dedication by your constant desire to  
improve as proved by the wonderful  
results you have obtained with your  
DTP in issue #20.

Please feel free to include this letter in a  
future issue if you desire.

Atarily,

Alex Pignato

President OL' HACKERS A.U.G.

# TWAUG NEWSLETTER

## -----B - T A P E -----

Version 1.0

for BW-DOS and Sparta-DOS 3.2

By Jiri Bernasek (BEWESOFT)

### INTRODUCTION

In the Czech republic (the country where I live) most Atari XL/XE users have no disk drive. At the beginning, the disk drives were not available here, and when they later appeared on the market, they were pretty expensive. Because of this, the XC12 Datacorder is the only data-storage device not only for game-freaks, but also for many "serious" users. Unfortunately, the original tape-system (the "C:" device) is no good for daily use, it is slow, and also unsafe. Because of this, several tape recorder upgrades were built in the eastern part of Europe during the earlier years.

All these systems are working with a little hardware modification inside the tape recorder. The most popular one is called "Turbo", and it was built in the Atari user club of Prague. The hardware isn't bad, but the first software was really awful, but - unfortunately - became very popular. It was just a little game-loader, with the file - format copied from poor Sinclair ZX Spectrum. Many programs were then modified (cracked) for this "system" - it was even necessary to add a quite long new tape handler to every "serious" program, because no CIO-handler was available. In fact, the original "Turbo" gave only high speed, without compatibility. Later the software was improved. After a few experiments with linking the new tape-system under CIO, then finally TT-DOS appeared. It is a DOS 2.5 clone with a high-speed tape handler "B:"

# TWAUG NEWSLETTER

## -----B - T A P E -----

included. In my opinion, this is the first version of "Turbo", goes behind the original "C:" handler, and gives something more. Not only high speed, but also a DOS-alike operation. And comfort - for example when the reading fails, there's no problem: Simply rewind the tape, and read the bad block again (no need to repeat reading of the whole file). A few months ago, they asked me to link this tape-system to my BW-DOS. So, now I'm releasing "B-TAPE". It is an even more improved version of this popular tape-system, compatible with BW-DOS and also SpartaDOS 3.2. It was made especially for czech users, but I decided to release also an English version. I don't think that a system like this can become popular in countries, where almost everyone has a disk drive, but may be someone will be interested... Well, you may wonder why I'm writing

all this. It is for you, "western" users, to understand why the "B-TAPE" was made, where it came from, and what is it good for. Now, a real "introduction" to the program follows...

B-TAPE is an advanced high-speed tape system for use with BW-DOS or SpartaDOS 3.2d, 3.2e, or 3.2f. The data-transmission is 5-10 times faster than the standard "C:" handler (depending on selected mode), and is safe (sometimes even more so than "C:"). The file-format is compatible with TT-DOS (an older czech DOS for both disk and tape). B-TAPE works only with a Datasort modified for "Turbo". It's a pretty simple modification, and it doesn't affect the original function. Look at chapter "Hardware" for more info. The B-TAPE package contains also a kind of Micro-DOS compatible loader for starting games etc. It can load almost every file-based

# TWAUG NEWSLETTER

## -----B - T A P E -----

games, demos etc. - also those with intros unpacking routines, and other things between loading. With B-TAPE, it's also possible to use BW-DOS with no disk drive. But it's clear that such a configuration is quite hard to use. For using BW-DOS with no disk drive, Ramdisk is strongly recommended.

### HARDWARE

B-TAPE needs a simple hardware-modification inside the tape recorder. This modification is called "Turbo", and is available from:

JRC, Chaloupeckeho  
1913, 169 00 Praha 6,  
Czech republic,  
tel./fax +42 2 354979,  
fax +42 2 521258.

They also sell modified new dataorders. For the Atari users who live too far, the documentation of the upgrade is included. I simply examined my

(modified) recorder, and wrote the following info.

Proceed at your own risk!

The basic idea is to add a second operation-mode to the recorder. It is only for reading, the recording section is almost unchanged. The new mode is selected by the COMMAND line "Low" at the same time with MOTOR line "High" (tape recorder on). (This occurs only while reading from tape, so the COMMAND line "Low" may not cause any troubles with other SIO devices -the DATA OUT line is off.) The signal from tape then goes (after amplification of course) directly do the DATA IN line (not via the decoding circuit). The tape dataorder Atari XC12 with upgrade is shown on picture "XCTURBO.PIC" - use any painter software for hi-res graphics (Basic mode 8+16) to view. The modification is as follows:

# TWAUG NEWSLETTER

## -----B - T A P E -----

- Remove C4, C6, C13, D3, D4, and R23.
- Add a 68kohm resistor between U1c inputs as shown on picture.
- Replace R24, R38, and R6 to match the values shown on picture.
- Add a transistor (for example BC546, almost any universal NPN type is OK) with a 47k ohm resistor and two diodes - as shown. You can use the diodes that you just removed from the positions D3 and D4.
- Open the I/O connector at the end of cable. Remove the wire from pin 11 (orange or red), and place it to pin 7.
- Add a 10nF capacitor between these pins (7 and 11).

After the modification, the recorder should work in the same way as before. And of course, B-TAPE should work also. If it doesn't work, and you didn't find any mistake, try to check the values of components inside the recorder (and replace to match the values shown on picture). If you experience

noises on the data-line, try to place a small capacitor (22pF or so...) at the position of C6 (the original capacitor is removed to increase frequency-range, and so allow high-speed transmission). This simple upgrade decreases the volume of audio signal that goes from tape to TV-speaker. If you don't like this, leave the line from pin 11 of I/O connector to C12 unchanged (add no capacitor into connector, and don't link it to the added transistor), and connect the emitter of new (added) transistor directly to pin 7 of I/O connector using a special wire. Unfortunately, this wire must be placed at the outside of the original I/O cable.

### INSTALLATION

The high-speed tape system is supported by a new resident command "BTAPE". This software doesn't affect the original "C:" handler.

# TWAUG NEWSLETTER

## -----B - T A P E -----

- BTAPE [mode][speed][memory]
- (Resident command)
- mode ..... SS, SD, LS, LD
- speed .... 0-20
- memory ... B[bank][,address]
- O[address] or OX

This command installs a new tape handler "B:" that allows the high speed operation. All parameters excepting "memory" may be changed by starting this command again; in this case it isn't necessary to repeat the unchanged parameters. The function of the parameters is as follows:

- ☆ "mode" may be
- ☆ "SS", "SD", "LS", or "LD".

The first letter controls the length of gaps between blocks -

- ☆ "S" is for short gaps, and
- ☆ "L" means long gaps.

The second letter controls the function of double blocks -

- ☆ "S" means single blocks,
- ☆ "D" is for double blocks.
- ☆ Default is "SS".

The parameter "speed" allows you to select one of the 21 possible transmission speeds.

☆ Default is 12.

Both the parameters "mode" and "speed" are important only while recording data to tape.

While reading from tape, the B-TAPE software recognizes the used mode and speed automatically. More info about these parameters will come later.

The last parameter "memory" allows you to select, where the main part of BTAPE program will be in the memory. This resident is quite long (3kB), so installing it into low memory may cause trouble with too high MEMLO value. Because of this, the resident code of BTAPE may also be installed under OS-ROM, or into a Ramdisk bank. This reduces the size of resident code in low memory to minimum (110 bytes).

# TWAUG NEWSLETTER

## -----B - T A P E -----

Default setting (with no "memory" parameter) is to install the whole program into low memory. This increases the MEMLO value a lot, so many programs may not work in this mode. The DOS commands and Atari Basic are OK.

The parameter "/O" causes the resident code of BTAPE to be installed under OS-ROM. In this mode, the compatibility is like with SpartaDOS. MEMLO is OK, but programs like Turbo Basic or XLRDISK may not work.

Advanced users may add a hexadecimal address after "/O" (BTAPE will then be installed at the specified address (the length of the code is \$BFC - for version 1.0), and so it may work with programs that are using just a part of memory under OS-ROM). Under SpartaDOS 3.2, you must use parameter "/OX" while installing BTAPE under

OS-ROM.

The "AINIT" function then doesn't work, and the errors in CP are shown only as numbers (both the functions are removed to provide space for BTAPE). The parameter "/OX" may be used only under Sparta.

With the parameter "/B", the BTAPE resident code will be placed into a Ramdisk bank. (Advanced users may add a bank number (value for \$D301) and address when necessary.) This is the most universal mode (but you must have min. 128kB RAM).

The "RAMDISK" command must then be installed with a configuration file, to reserve a bank for BTAPE (see later). The default bank used by BTAPE is \$E3.

You can also use the "E" parameter, but in this case the capacity of Ramdisk will be reduced much more (64kB



# TWAUG NEWSLETTER

## -----B - T A P E -----

instead of 16).

### RAMDISK configuration

To reserve a bank of Ramdisk memory for another use (in this case BTAPE), you must use a configuration file every time you call the "RAMDISK" command.

In the B-TAPE package, there are four configuration files included. These files are reserving the bank \$E3 (that is used by BTAPE), and they are designed for the following memory-expansions:

- ☆ "B\_130XE.RD" is for Atari 130XE (64kB)
- ☆ "B\_COMPY.RD" is for a "Compy Shop" type expansion (256kB)
- ☆ "B\_RAMBO.RD" is for a "Rambo" type (256kB)
- ☆ "B\_NEWELL.RD" should work with a "Newell" 1MB expansion.

Advanced users may create another Ramdisk-configuration files using for example the

command "HEXEDIT". The file contains a list of banks available for RAMDISK use, each byte for one bank (values for \$D301).

### USING B-TAPE

The access to tape in high-speed mode is provided by device handler "B:". This handler uses filenames, so the full specification should be "B:name.ext". The syntax of a filename is the same as for disk-files under BW-DOS (including wild-cards). You can select the "B:" device as working drive in the CP (by typing "B:"). The length of tape-files under BTAPE is limited to 257040 bytes.

### Using the recorder under B-TAPE

B-TAPE routines are using simple sound signals for communication with the user during the I/O operations. Recording a file on tape starts with the well known "double beep" sound (known also from "C:"). It's necessary to find the correct place on tape, press <PLAY> and <RECORD> on the recorder, and

# TWAUG NEWSLETTER

## -----B - T A P E -----

then <RETURN> on computer. BTAPE will then records a sound signal "start of file" (a short melody for going up and for going down again) followed by the file itself.

Reading a file starts with a single "beep" (also known from "C:"). You needn't de- press any keys on the computer - the program waits for data immediately after the "beep", so just press <PLAY> on the recorder. BTAPE will search for a file and check its name. When the name doesn't match the file-specification, a sound signal "bad file" appears (exchanging two tones) - it's of course necessary to move the tape to the correct file. If you don't know where it is, abort the operation with <BREAK>, and use the "DIR" command for orientation.

When no readable block is found for 4 minutes, the operation is terminated (Error 138). The program may also ask you to move the tape to the correct block in the file - i.e. the first block. A simple melody will play, this indicates that you should use the <F.FWD> button (or check - if you are near the correct block), again a melody will play prompting you to use the <REWIND> button. While searching for the correct file

and block, the program accepts any file that match the file-specification. But once a block with correct name and serial number is found, the program goes into reading mode. Then, no blocks from other files are accepted - even if the filename is identical. The sound signals "F.FWD", "REWIND", and "bad file" may also occur during the reading process. One of the possible reasons is that reading of some block failed, and so the program asks you to rewind tape for retry. This may be also caused by bad length of the gaps between blocks (more info will come later). Another possible reason is a request from the application software to read from an other position in the file (the POINT function - it occurs for example with the command "OFFLOAD"). When the tape stops for a longer time (the reading is finished, or interrupted - for example some kind of intro is working while loading a game), it's good to press <STOP> on the recorder to avoid damage of the tape.

During the high-speed tape operations, the TV screen is off. While recording to tape, the screen is simply black, and while reading, it shows the signal from tape.

# TWAUG NEWSLETTER

## -----B - T A P E -----

Disabling the TV screen is necessary to allow exact timing of the I/O routines (no hardware support is available for the used format).

BTape installed under OS-ROM may also sometimes affect displaying of text modes (because of switching the ROM off for executing some of BTape routines). The problem is only on the screen; it may not cause any damage.

The first operation executed on a disk drive after a tape-oriented operation will always begin with a short pause (1 sec). This is necessary to avoid crashing the first disk I/O by the tape recorder, that is unfortunately "active" also a while after turning it off (because of its capacitors). This function is very important while copying files from tape to disk. When the "XFSIO" command is used, it must be installed before BTape. Otherwise it may override the mentioned function (the pause).

### Selecting mode and speed

While recording a file to tape, the parameter "mode" and "speed" is important. The speed depends on the quality of the data recorder and tape. Mostly you'll be able to use the

highest speeds, but when there is a problem with the hardware-expansion, it is best to select a lower speed. The default speed 12 should be OK in most cases.

The function of "double blocks" means that every block will be recorded twice. In this mode, the file may survive small drop-outs on the tape, but the effective speed of transmission is decreased of course. This mode is especially good for very important files. The first block of every file is always doubled. This allows programs like "COPY" to find the file with the "DIR" function, and then start reading immediately, and it also helps in the case that recording overran into next file on the tape a little.

The length of gaps between blocks depends on the software, that will read the file later. In most cases, the mode of short gaps is good - this mode is for programs that read the whole file at once, without of pause. In this mode, the tape recorder doesn't stop between the blocks and the reduced length of gaps is saving time. When the application software needs a longer time for interpretation of the received data, the recorder will stop after a while automatically.

# TWAUG. NEWSLETTER

## -----B - T A P E -----

While recording to tape, there's no problem (you only get a longer gap), but while reading from tape, the recorder will overrun the beginning of the next block. In this case you'll need to use the <REWIND> button - this indicates that the mode of a longer gaps would be better. In this mode, the recorder stops after each block, and the gaps are long enough for the recorder to stop/start while reading from tape. It decreases the transmission-speed of course, but the application software has now unlimited time to interpret the data. This mode is good, for example while doing I/O operations using Basic-commands.

The binary program-files are a special case. Mostly the short gaps are OK, but there are programs (especially games and demos) that execute pretty long actions between loading - different intros, unpacking etc. Because this occurs only in a few places in the (mostly quite long) file, the mode of long gaps would mean a big time loss. In this case, you should copy the file to tape with the command "BINCOPY", that increases the length of certain gaps in the file. (More info later.)

### TAPE

The CP-commands of BW-DOS are mostly working on tape. The commands that don't work are those ones that use special disk-oriented functions:

MENU, UNERASE, BACKUP, MOVE, MSDOS, BOOT, CHVOL, XFSIO, FORMAT, PROTECT, UNPROTECT, RENAME, ERASE, CREDIR, DELDIR, CWD, DIRMAST, CHKDSK, HEXEDIT, and AUTOQWD.

Use of such commands on tape is a nonsense anyway... But these commands may become useful while working without a disk drive - when you use a Ramdisk.

The batch files and Hard Copy function works with tape, but without of a Ramdisk or a disk drive, it's almost senseless. Because the tape handler can't open more files at the same time, the working batch file will block the reading of commands and other files.

Some commands have different functions for disk and tape:

DOS COMMANDS ON

DIR [B:][filename]

# TWAUG NEWSLETTER

## -----B - T A P E -----

**DIRS [B:][filename]**

(Internal commands)

These commands are useful for the orientation on tape. Both the commands have the same function for tape: It'll read one block (any block that match the file-specification), and displays the name of the found file followed by serial number of the block in the file. When the file was recorded with double blocks, a little star will appear in front of the name. Because of compatibility, the listing keeps the syntax of a disk-directory - including the "FREE SECTORS" counter (it's senseless on tape, but there are programs that needs it while interpreting such a listing).

**COPY source destination**

(External command)

The "COPY" command works for tape-files too, of course. But it's important to use the new version, found on BW-DOS 1.30 master disk or in the "Update 2" package. The older version can't carry the original filename from tape (so you must always type both the filenames - source and destination), and it also can't copy directly from tape to tape.

This is also true for the internal "COPY" in SpartaDOS. While copying directly from tape to tape, you can only work with files that fit into the free memory at once. The maximum length depends on the version of DOS, installed resident commands, and also the status of internal Atari-Basic (or other cartridge installed).

To get an approximate number, you can subtract MEMHI - MEMLO - length of the "COPY" program. (First two numbers are shown by the "MEM" command, the last one comes from a disk-directory.)

### TAPE BOOT

BOOT is loading of the DOS after power-on. With B-TAPE, the boot may be also executed from tape.

### Boot loader

The first file on a boot-tape must be the Boot loader. It is a little program that loads the DOS itself and allows also loading of necessary residents, such as "BTAPE", "RAMDISK" etc. The Boot loader is stored in the same format as common tape-games to allow easy booting of the tape.

# TWAUG NEWSLETTER

## -----B - T A P E -----

### TAPEBOOT (C|T)

(External command)

This command records the Boot loader to tape. With the "C" parameter, you'll get a common boot-file in the "C:" format -such a file may be easily booted by holding down the <START> key during power-on.

The length of this loader is 14 blocks. The "T" parameter was included especially for czech users. With this parameter, the "TAPE-BOOT" command records the same loader in the format "TURBO 2000" - the very first version of the "Turbo" system. This is useful for the users, who own a cartridge with Turbo 2000 loader (these cartridges were sold here in the czech republic). The Boot loader is always the same, so you can transfer it with different tape-coplers.

Every files after the Boot loader will be in the B-TAPE format. The first of them must be a DOS file (a file that match the specification "X\*.DOS"). It may be any version of BW-DOS, or SpartaDOS 3.2d, 3.2e, or 3.2f.

The file is stored on tape without of any changes - you can copy the DOS from disk to tape and back

without any problems.

---

To be continued in the next issue of T.W.A.U.G.

We have a number of subscribers who are still using a Datacorder as a main source of saving data, this article should be of some interest to them.

In the follow on, in the next issue, we will also publish the diagram for this upgrade, we received with this article.

---

### SECTOR SKEW - HIGH SPEED SIO

**W**hat the heck is Sector Skew and High Speed SIO and all that Techie Stuff?

By Bob Woolley SLCC

There are a number of schemes out there to speed up the transfer of data from the disk drive to the computer. The absolute best is the RAMDISK. This technique does not need to get data from the disk drive at all, it only needs to look in memory for the data. Another modification is a parallel interface to

# TWAUG NEWSLETTER

## SECTOR SKEW - HIGH SPEED

modification is a parallel interface to the drive. By sending the data 8 bits at a time instead of 1 bit at a time, you can load data very rapidly. At the present time, only the ramdisks are available and they have some serious operating problems. They work, but are difficult to use on many types of programs. The most common and useful enhancement is the high speed SIO modification from ICD and others.

This type of upgrade allows the SIO hardware to function in the same manner as a normal machine, but at a much higher rate - almost three times faster. When a byte of data needs to be transferred, the controller loads it into a special register. From the register, a clock sends the data out over the interface 1 bit at a time. Every clock tick boots out another bit, eight clicks, eight bits. By increasing the clock frequency, you decrease the time it takes for the 8 bits to be transferred and this is essentially what these upgrades do. If that was all you did to your computer, though, you would be somewhat disappointed - the data would take just as long as before to load into your computer.

The reason for this is the sector skew of the disk. It has to be modified to take advantage of the higher SIO clock speeds.

Think about a Merry-Go-Round that has 18 wooden horses arranged around the outside. You are standing at the edge of the Wheel as it spins on its axis. On each horse sits a child with a number pinned to his shirt that you can read as he approaches your position. These numbers run from 1 to 18 and it is your job to remove the children from their horses as they pass by you. You must take them off in number order and carry them to the edge of the grass where you can put them down, OK? Here comes number 1! No problem. You lift him off his seat, walk over to the grass and place him gently on the ground. Easy. As you go back for the next kid, you realize that while you were handling number 1, number 2 has gone past your position. Now you have to wait for a whole revolution for number 2 to come back. Hmmmm. It should be obvious to you that if you can get back for the next child in less than one revolution of the disk, that changing the arrangement of the little fellows will allow you to get them all off more quickly.

# TWAUG NEWSLETTER

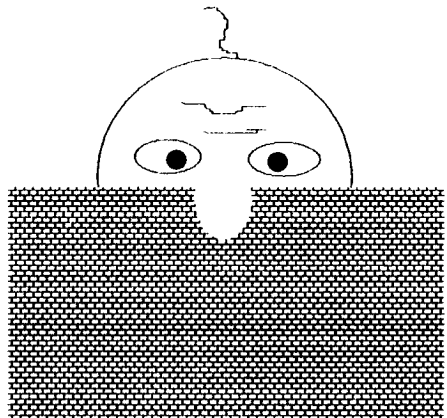
## SECTOR SKEW - HIGH SPEED

Let's arrange the kids so that they sit 1, 3, 5, 7, 9, 11, 13, 15, 17, 2, 4, 6, 8, 10, 12, 14, 16, 18. Now, if you can grab the proper child and deposit him safely on the turf before the wheel has moved one-half the way around, you won't have to wait hardly at all for the next number. You should be able to remove all the numbers in only 9 revolutions instead of 18, like the old sequence. Of course, if you can't get to the next number in less than half a revolution, it will take you even longer with this format than with a sequential one like you had!

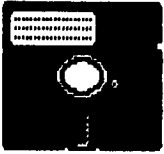
This is the basis for reading the sectors on the diskette faster than normal. You need to increase the speed at which you transfer the data to the computer AND you need to change the order in which they are written on the drive. Just by increasing the speed at which you can put the little monsters away does not get the job done any faster, does it? The XF551, for example, therefore has two ways to arrange the sectors (a SECTOR is just a place to put a block of data) on the disk - Normal and High Speed. Once the data is committed to reside on a certain "horse" during FORMAT time,

all the drive can do is to alter what is in the sector, not where it goes on the "Wheel". With the ICD upgrade, you can actually program your own sector sequence on a US doubler 1050 and customize your disks for your personal application.

Between the two changes then, you arrive at a useful and reliable method to load (or store - it works both ways) data much faster than the original system. Just be aware of the fact that the high speed sector skew can actually slow down your system if you are not reading it on a modified drive.







## **DISK CONTENT**

On Side A of this Issue disk you will find a nice little program called Menu Planner. With this program you can plan your Sunday Dinner, or your Lunches, or evening Meals. It is easy to use, all you do is use the space bar to cycle through the choices of vegetables or meats available and select your choices with the Return key.

Another good program is the Autoboot utility. If you are looking for a program to help you with setting up a basic program to an AUTORUN.SYS file, this is it, easy to use, just follow the prompts.

The program SOUND SYSTEM is for the person who loves to write music. With the menu on the screen, you've got the Editor, Player and the Writer options to choose from. I am not musically minded and so I am not able to say anything about this program, or what it can do. I can't even sing.

There is also a Chess Game, called Laser Chess, there aren't any instructions, but I am sure it's fairly easy to play.

Prisonball, when loaded you get five vertical bars on the screen and by pressing the start key some balls start bouncing about on either side of those bars. The aim is to break through the bars with the balls.

The one Demo on Side A and the four Demo's on Side B have been produced and dedicated to TWAUG by Mark Watson. Mark is still at school, he is 17 years old and they are his first demos. He is also looking for some contacts. Side B must be loaded without Basic, so for XL/XE users press Option when booting and then select by pressing the appropriate number.

You also find on Side B, the UNARC.COM utility to unarc the PDCATB.ARC file, this is an update on our PD Library list.

# TWAUG NEWSLETTER

## ADVERTISING USER GROUPS

### LACE

### O.H.A.U.G.

#### The London Atari Computer Enthusiasts

As a member of LACE you will receive a monthly newsletter and have access to a monthly meeting. They also support the ST and keep a large selection of ST and 8-bit PD software.

The membership fee is  
£8.00 annually

for more information contact:

Mr. Roger Lacey  
41 Henryson Road  
Crofton Park  
London SE4 1HL  
Tel.: 0181 - 690 2548

---

#### The OL'HACKERS ATARI USER GROUP INC.

O.H.A.U.G. is an all 8-bit user group in the STATE of NEW YORK.

They are producing a bi-monthly double sided disk based newsletter. The disk comes with its own printing utility, which lets you read the content of the disk, one screen page at the time, and/or you can make a hard copy of the disk, in one, two or three columns and 6 to 8 lines to the inch. A large PD Library is available.

Contact:

Mr.A.Pignato

O.H.A.U.G.

3376 Ocean Harbor Drive

Oceanside, N.Y. 11572

USA

TWAUG. NEWSLETTER

# ADVERTISEMENT

## CHAOS! COMPUTERS

PO BOX 30

MANCHESTER M19 2DX

Telephone: (0161) 737 1946

### THE HYPER DRIVE

Upgrade your *ATARI 1050* disk drive with a **HYPER DRIVE** enhancement from **CHAOS! COMPUTERS**.

The **HYPER DRIVE** is an easily installed hardware & software package for the *ATARI 1050* which will enable your disk drive to back-up most disks protected by unreadable or badly formatted sectors. Most copied disks can then be loaded on any 1050, whether enhanced with a **HYPER DRIVE** or not. The **HYPER DRIVE** enhancement also offers fast reading, writing, formatting and copying in single, medium or true double density formats (i.e. it is compatible with *RANA*, *PERCOM* and *INDUS* double density drives, and will read *U.S. DOUBLER* type format).

Fitting the **HYPER DRIVE** couldn't be simpler and requires no special tools or soldering. It simply plugs into socket on the 1050 circuit board. And with our **VERSION II** software package and full 28 page manual, it is one of the most versatile disk drive enhancements /copiers you can buy. **HYPER DRIVES** are available exclusively from **CHAOS! COMPUTERS** at a special introductory price of just £30.00 each.

Please make Cheques/Postal Orders payable

to 'P. HOLLINS'.

Prices are subject to change, from time to time, due to component costs, so wherever possible please *'phone to check'*.

# MICRO-DISCOUNT

Offers  
The complete Mail Order  
service for Atari 8 Bit



XL/XE users  
4 th September 1995

