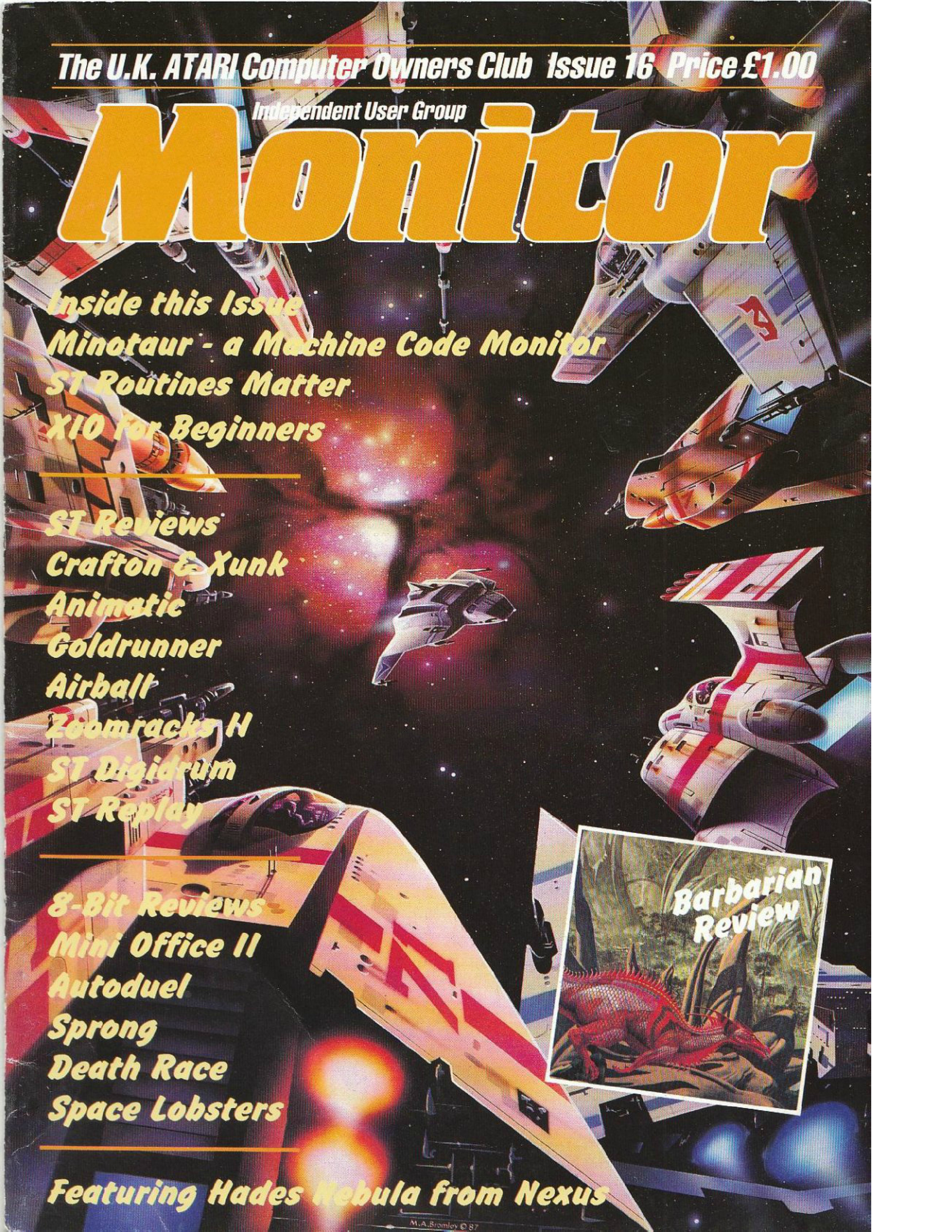
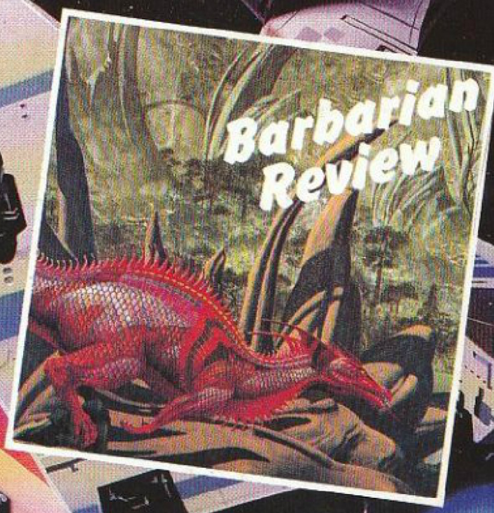
The U.K. ATARI Computer Owners Club Issue 16 Price £1.00

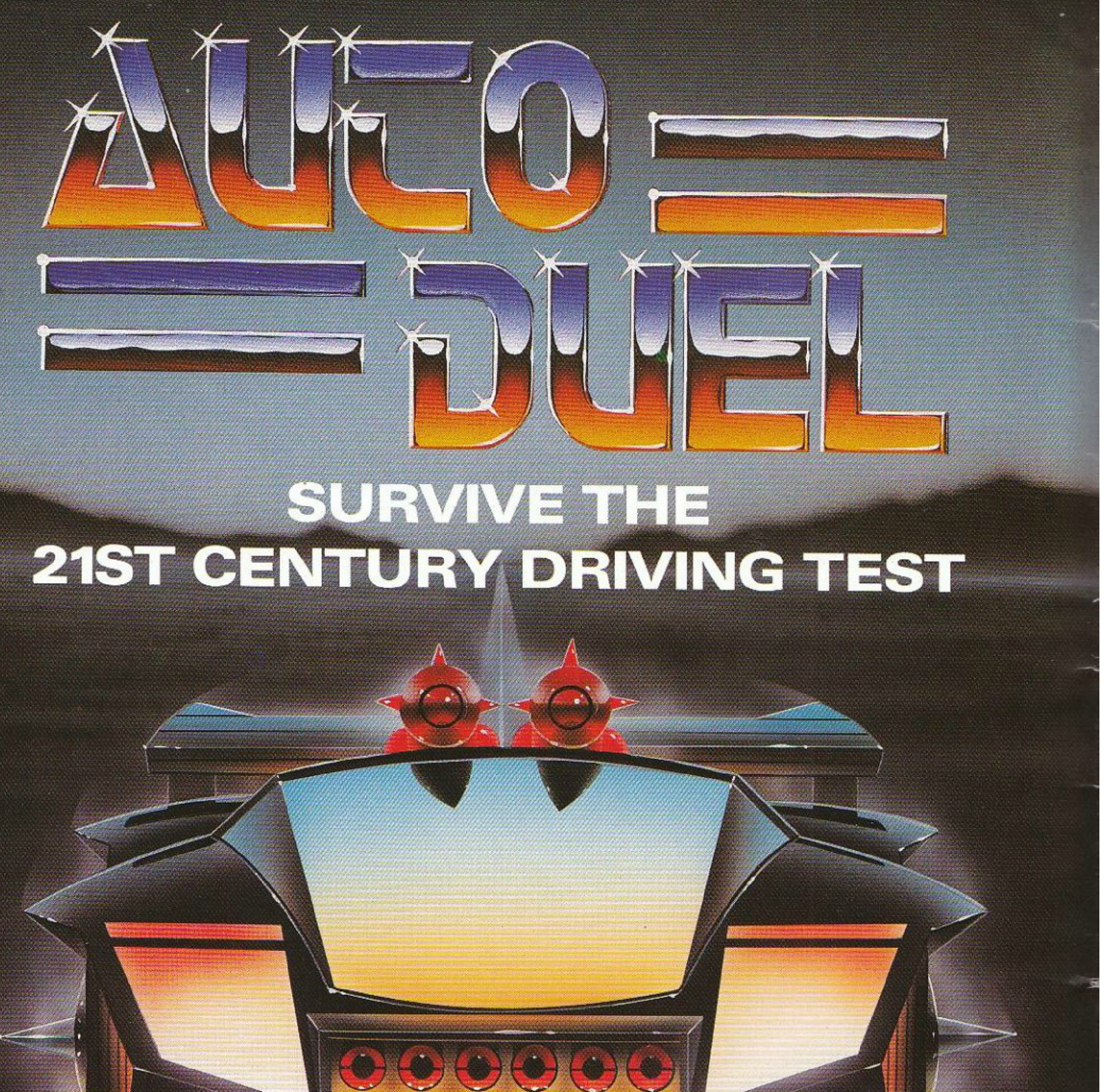
Independent User Group

# Monitor





Featuring Hades Nebula from Nexus



## Drive aggressively and give way to no one.

AutoDuel is a fast-paced, strategy rôle playing adventure set in the year 2030. A time when the American highways are controlled by armed outlaws, and when danger lurks around every bend.

Your aim is to earn fame and fortune. Compete in spectacular auto dogfights in true gladiator style – the prize money will buy you a powerful custom-built vehicle equipped with lethal weapons, including lasers, machine guns and flamethrowers. Undertake lucrative courier runs for the American AutoDuel Association or become a vigilante of the open road. Somewhere on your travels between 16 cities are the vital clues you will need to complete the final mission.

Only the most cunning road warriors are enlisted by the FBI to drive out the ultimate evil force. Will you gain honour and entry into the elite circle of AutoDuellists?

AutoDuel. Pick up the gauntlet. Available on disk for the Apple, Commodore 64, Atari, Atari ST and Amiga. Prices from £19.95.

Based on the award-winning Car Wars board game by Steve Jackson.

THE JOINT VENTURE IN EUROPE

ORIGIN SYSTEMS INC.

MicroProse Software Ltd. 2 Market Place. Tetbury, Gloucestershire GL8 8DA. Tel: (0666) 54326. Tlx: 434222 MPS/UKG

# CONTENTS

## WRITE FOR MONITOR

We are always looking for good articles, reviews and programs to publish in Monitor. If you think you turn out an interesting read or can write good programs, why not submit them to the Editor, you never know, he may be impressed!

Here's what to do; type your literary masterpiece on your word processor, but save it off in ASCII format. ST owners should use half-meg format. Then send it to the club address for the attention of the Editor, any program listings should also be on the disk. (It's also a good idea to include a contact phone number if possible.) Then when you next receive Monitor you may find your name in print!!

## MONITOR CLASSIFIED

From the next issue we would like to start a classified advertisement section. This will be for private individuals only (not companies) to buy and sell computer hardware, software, make contacts, find pen pals, etc. All adverts will be free up to 30 words, thereafter the charge will be 10p per word (cheques and Postal Orders made payable to the club). Send your advert to us by the 1st September for the next issue together with any payment necessary. Please mark your envelope 'Classified'.

## SUPPORT YOUR CLUB

We have stressed many times in this column that the club is what you make it. At the moment we are doing fairly well, but we could do a lot better. The club membership (same as a subscription to Monitor) could be a lot higher. It would help greatly if every member could promote Monitor in their area. How you say? Well that's easy, we're sure you know at least one other Atari computer owner who doesn't subscribe (probably reads your copy, right?) or maybe more at the local computer club you go to? Try to get them to join, refuse to let them read your copy or something. Or point out the benefits: Only £4 per year! Access to public domain programs. Interesting articles, tutorials, and reviews. But they know that anyway if they are reading your copy, don't they! Shouldn't take a lot to convince them.

## CREDITS

| | |
|---|---|
| Editor | Roy Smith |
| Art Editor | Greg Buckley |
| Technical Editor | Ron Levy |
| Technical Editor | Keith Mayhew |
| Adventure Editor | Steve Hillen |
| ST Librarian | Mike Stringer |

## SUBSCRIPTION RATES

| | |
|---|---|
| U.K. and Ireland | £4.00 |
| Europe | £7.00 |
| Outside Europe (Surface) | £7.00 |
| Outside Europe (Airmail) | £10.50 |

The above prices are in English Pounds Sterling and include postage and packing.

A subscription/membership fee to join the U.K. Atari Computer Owners Club is just £4.00 for four issues of the club magazine. All cheques/postal orders are to be made payable to the 'U.K. Atari Computer Owners Club'. Overseas membership is also available at slightly higher rates. Overseas members who use the Library service should include enough extra monies to cover return postage.

## ADVERTISEMENTS

Please note that the club cannot be held legally responsible for claims made by advertisers.

Cover: Hades Nebula from Nexus and Barbarian copyright Roger Dean and Psygnosis.

Club Address: U.K. Atari Computer Owners Club, P.O. Box 3, Rayleigh, Essex, SS6 8LR.

## Mini Office II

From Database Software
Review by Robin Anthony

In this fast moving world of mega plus-plus, new computers, new prices, new ideas coupled with the fact that by the time you get your new treasure home it's just been devalued by the emergence of yet another product, it comes as a pleasant surprise to find a new business program for the Atari 8 bit series.

In what has become the poor relation of the 16 bit computers, the launch of a non-arcade package could possibly be a shot in the arm for all of us who quailed at the thought of selling their 8 bit (can you find a buyer?) so we wouldn't be left behind in the software race.

Mini Office II from Database Software, a part of the Europa Press Group of companies is marketed as an integrated package consisting of the six most essential items for a small office. The following modules make up Mini Office II:

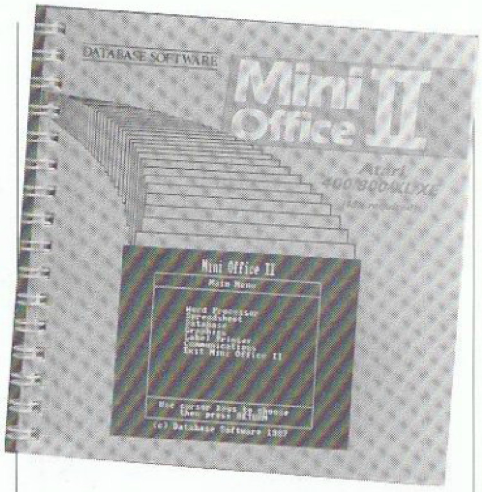
Word Processor
Database
Spreadsheet
Graphics
Communications
Label Printer

All modules are really stand alone packages which to some extent can be integrated with other sections. Each module is accessed from disk as and when required and because of RAM limitations will replace the previous module in memory.

Menu driven throughout, Mini Office II is aimed initially at a new user to business software by making the majority of user commands available from a series of menus. Most of the modules have tree structured systems, i.e. selections from a main menu open onto other menus. Retracing your steps is by the same method but the escape key is a failsafe which takes you back to the previous command and so on.

Before I go through the modules one by one, there a number of features available for all the modules. Screen colours can by changed by using the Start, Select and Option keys to cycle through the whole 256 colour range of the Atari. All the data saved to disk can be saved with a back-up file or overwritten as required. Each menu also has a mini-dos option and within this, disks can be formatted, files locked and unlocked, erased, renamed and directories of any drives read.

Taking the Word Processor first, I was amazed at the range of features available, the majority of which are accessed through the menu system. The main



option is the Editing screen, it's here where the bulk of the work is done. Text is entered in either 20 or 40 column mode which is rather unusual for word processors. The 20 column mode is really graphics mode 2 and consequently doesn't show lowercase letters.

To view text you'll need to return to the menu and select the Preview window, this shows your document in full 80 column mode with all the attributes shown such as margins, justification, line spacing, etc.

As you get used to using the word processor, the need to return to the menu can be almost eliminated as the program will allow you to embed commands to customise your layout. There are standard features within the module, Search and Replace, Insert/Overwrite mode, Cut and Paste, Justification, Ragged text, Automatic paragraphing, Indenting and Pagination just to name a few.

It seemed strange but I couldn't find underline anywhere, the Search and Replace option could only be operated through the menu system and rather disconcertingly it didn't hold the last search string in memory. There was a feature to see how fast your typing speed was, I only managed two words a minute regardless of the speed of my nimble fingers, I didn't think much of that feature! While I'm discussing points that annoyed me, the shortcut routine for writing the file to disk left the last character pressed displayed on the screen, which if you were not that observant may well get left in your text. One obviously well meaning feature, the Caps Convert had a verification sequence built in, this part was meaningless and irrelevant.

To finish off this section, lets look at the good points, of which I'm pleased to say outnumber the bad many times over. Headers and Footers can be defined easily, Page numbering was good, the length of the line could be up to 150 characters wide, page length up to 200 lines deep. Files too large for memory could be chained together for printing and files could be written to disk either in ASCII or Mini Office II format.

The print routine was an impressive feature, documents could be formatted in varying print styles to either Epson or 1029 printers. Mail merging was one of those little parts of this program which actually integrates between modules, information from the database can be merged into a document to produce personalised letters or perhaps data represented as a series of columns.

The database, next in line on the menu is of the 'Flat Card' type of storage system. Mini Office II has three main types of fields, Alpha, Numerical and Date. Database Software reckon that it has five types but these extra two are really only cosmetic changes to the numerical field, the numerical sphere is split up into Decimal, Integer and Formula. The first two speak for themselves but the latter is quite interesting for this type of database.

Formula will allow you to set up comprehensive calculations which can be processed on the screen, this I feel makes the database worthwhile. Working from the menu,(there are no embedded commands here) you can set up your database by defining your fields, saving them to disk and then 'opening' up your database so you can enter your data.

Screen format for the data is of a fixed style but this presents no problems, cursor keys move each record up and down and searching for data follows the same record view format but you can select from any field following a selection criterion of, equals to, not equal, less than, more than, wildcards are allowed either as single characters or whole strings.

Sorting can be across fields and follows a logical procedure ending in your data being indexed in ascending or descending priority. A clever part of this module is the 'Marked Record', this little feature will mark records of your choice to which a number of extra features can be associated with, for instance all marked records can be selected and saved to disk as a separate file.

The print routine for the database follows almost the same criterion as the word processor, there is a flexible printer editor in which you can 'stylize' your format. Records will be printed in tabular format or can be placed singularly one underneath each other, this routine is very flexible and I'm sure will suit most peoples requirements.

# Mini Office II

**6 powerful home and business programs in just ONE package – at a price that simply can't be matched!**

**Voted Business Program of the Year – 1985 AND 1986**
*Popular Computing Weekly*

**'This package is incredible value'**
*Daily Mail Home Computing Guide*

WORD PROCESSOR
**Compose a letter, set the print-out options using embedded commands or menus, use the mail merge facility to produce personalised circulars – and more!**

SPREADSHEET
**Prepare budgets or tables, total columns or rows with ease, copy formulae absolutely or relatively, use a wide selection of mathematical and scientific functions, recalculate automatically – and more!**

GRAPHICS
**Enter data directly or load data from the spreadsheet, produce pie charts, display bar charts side by side or stacked, overlay line graphs – and more!**

DATABASE
**Build up a versatile card index, use the flexible print-out routine, do powerful multi-field sorting, perform all arithmetic functions, link with the word processor – and more!**

COMMS MODULE
**Using a modem you can access services such as MicroLink and order a wide range of goods from flowers to software, send electronic mail, telex and tele-messages in a flash – and more!**

LABEL PRINTER
**Design the layout of a label with the easy-to-use editor, select label size and sheet format, read in database files, print out in any quantity – and more!**

DATABASE SOFTWARE **Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY**

ORDER FORM

Please send me Mini Office II for the Atari 400/800/XL/XE (48k required), on 5¼" disc for £19.95

I enclose cheque made payable to Database Software, or debit my Access/Visa card:

☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Exp. date ☐

Signed ........................................

Name ........................................

Address ........................................

........................................

........................................

*ORDER HOTLINE: TEL: 061-480 0171*

*SEND TO:* **Database Software, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.**

A018

Following on to the Spreadsheet module may be a little bit of a disappointment as it doesn't resemble a true spreadsheet profile. However if you've never used a spreadsheet before this won't worry you unduly. There were 52 columns and 84 rows which is not bad for a small office to get used to. Entering data was reasonably easy to do but to edit previously entered data was almost an impossibility!

There wasn't a 'Range' command in which to enable a series of cells to be formatted so consequently each cell had to be formatted individually, one way round this was to 'replicate' (copy) the cell to another position. In a change of direction, the majority of commands in this module were operated by control and letter keys, luckily a handy help screen was available to get you through the mire.

All this said, the module functioned adequately and I was able to demonstrate that even after all my spending I had some money left in my account. Worksheets could be saved as whole or partial, I couldn't see any way in which to append worksheets but this function is not always necessary.

As you'd expect, the printer routine worked well enough, style and presentation could be formatted and the procedural formulae could be sent to printer in columnular format. A handy feature worth mentioning is the graphics data transfer, this option allowed you to select up to 20 cells and save them in a format acceptable to the Graphics module.

This takes me nicely on to that section, here we have a well developed module that makes your data presentation look really good. Choose from Bar or Line graph and have up to three sets of data each with up to twenty values. Both of these graphs could display in 2D or 3D format, side by side or stacked. The Pie chart handled twenty values also but like any other pie chart only one data set at a time.

Again in a marked change from the total menu system, there was a different selection procedure encapsulated in the module. In keeping with the 'graphics' feel of things, Icons placed down the right hand side of the screen became the operating system. I have to admit that even though I liked the idea, they were hard to get used to, difficult to view properly and felt totally out of place in the 'integrated' package.

On a positive note the quality of charts and graphs both on screen and dumped to printer were excellent, as impressive as anything I've seen on the 8 bit. Files could be saved to disk for importing into other programs or for slide show presentations.

Telecommunications are a specialist subject and one which the majority of computer owners feel as though they ought to be part of it yet lack the resources needed to make it effective. Database Publications (Europa Press) have a lot of involvement in this field and subsequently this module reflects the area's linked to themselves.

Microlink/Telecom Gold at 300/300 and 1200/1200 baud are available, customised protocols for bulletin boards, macro key facilities are some of the major selections available. Missing from this module is the opportunity to link into the huge database of Prestel, however I feel this section was included as a support program for the main sections rather than as a module in its own right.

The final option is the Label Printer, using information from your database you can transform it into address labels, or even labels for your software collection. If you don't want to use the module as an integrated package it'll work well as a stand alone labeller.

Up to a 100 repeated labels can be printed at any one time, there are plenty of commands from within the menu (yes this one's back to the old system!) to show your own style and individualism. The label printer is quite flexible in it's approach allowing precise positioning of data regardless of whether it's imported from your files or typed in manually in the free form state.

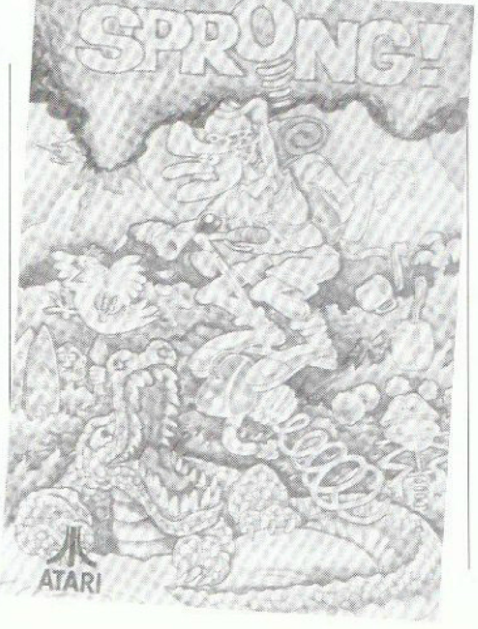
Any label format can be saved to disk thus allowing repeats to be quickly and efficiently carried out.

In conclusion, Mini Office II is a good value for money package, it's bad points are few and far between. The real value is not just a double sided disk, a small but well written 78 page manual, an amazing low price of £19.95 but in fact, is the support for 8 bit Atari owners at a time when business software is regarded as the domain of the ST. If you're looking for a cheap and cheerful program to help you get the best out of your computer then look no further, Mini Office II is ready and waiting!

## Sprong

From Red Rat
Disk Price £9.95, Cassette Price £7.95
Review by Otis Bey

Sprong was programmed by Paul Lay for Bignose Software and is being marketed by Red Rat. The subtitle of the game is 'the quest for the Golden Pogostick', which gives a good idea of what the game is about. It is a platform type game where you pogo your man across various hazards to the next screen all the while accompanied by the strains of 'Danny Boy' played on an organ. There are fifty screens to negotiate, each one getting progressively harder. Hazards include raging fires, laser beams, acid rain, lava flows and meteorites.



Your man has five lives per screen (a measure of the difficulty of play) and there is also a 2000 second time limit on getting across. Your man pogo's up and down automatically so you only have to move him left or right, plus to get an extra large jump you can press your fire button. When you are unfortunate enough to jump into the flames or fall into the water, pogoman gives you a scowl and promptly explodes into oblivion.

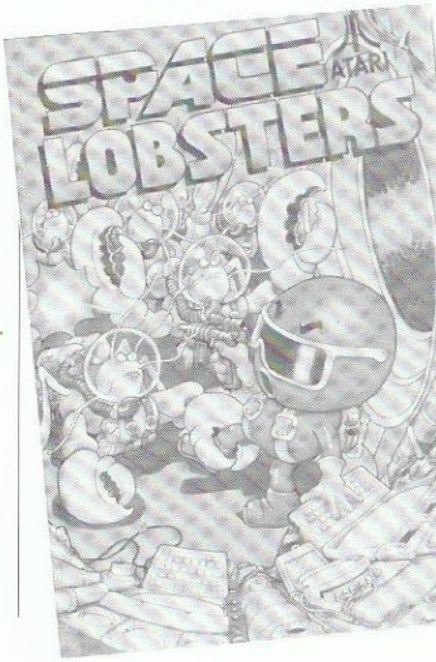
This game has obviously been well thought out, the graphics are very good, the playability is very difficult and therefore very challenging and there are nice little touches like each screen having a descriptive, almost poetical, one-liner about the scene displayed. It's very easy to get addicted to this one!

## Space Lobsters

From Red Rat
Disk Price £9.95, Cassette Price £7.95
Review by Mike Way

Space Lobsters is set on the drifting hulk of an abandoned space freighter, the nasties though, don't seem to realise it's abandoned! The amount of nasties per screen is just too much, and detracts from the playability. The screen is split about 60% control area to 40% play area. The control area displays a computer terminal screen and other statistics like ammunition, score and oxygen level.

The aim is to find terminals dotted around the craft and obtain 14 codes with which to escape in the escape pod.



Movement from level to level is accomplished by teleporter terminals, but apart from a slight changing in background colour it is not easy to tell that you have moved anywhere. This is the whole aim of the game, sorry, just another shoot 'em up to add to the extensive list.

Too little manoeuvering room makes this game short in play time and a lack of variety makes it short in interest time. The codes are not too easy to find, the packaging refers to over 100 screens, but I could not find anything like that many.

Over all the game lacks sophistication and all the hype has gone into the objective of the game, which does nothing to make me want to play the game for more than five minutes. Not one of Red Rats best.

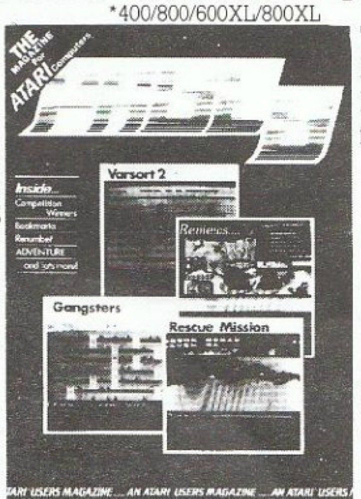
**PAGE 6** THE MAGAZINE FOR ALL **ATARI** COMPUTER* OWNERS

*400/800/600XL/800XL

**ATARI**

- NEWS
- REVIEWS
- TUTORIALS
- UTILITIES
- HINTS & TIPS

plus more


THE MAGAZINE for ATARI Computers
Inside...
Varsort 2
Gangsters
Rescue Mission
ATARI USERS MAGAZINE ... AN ATARI USERS MAGAZINE ... AN ATARI USERS
GAMES ATARI NEWS ATARI PROGRAMS ATARI REVIEWS

THE BEST PROGRAM LISTINGS from

- U.S.A.
- U.K.
- AUSTRALIA
- PUBLIC DOMAIN SOFTWARE LIBRARY
- SPECIAL OFFERS

**PAGE 6 is published bi-monthly.**
**Annual Subscription is £7.00. Send TODAY to:**

**PAGE 6, P.O. BOX 54, STAFFORD, ST16 1DR**

**Tel. 0785 41153**

### ATARI 400/800/XL/XE UTILITIES

**DISK-TO-DISK UTILITY:** A disk backup system that will enable you to rapidly backup your DOS disks, and also duplicate commercially available disks that are protected with CRC error **BAD SECTORS. DISK £14.95 Post Free.**

**DISK-TO-DISK PLUS:** Features the Disk-to-Disk utility as above, **PLUS** three further backup utilities; single-stage Tape to Disk, single-stage Disk to Tape, and single-stage and multi-stage Tape to Tape. **DISK £29.95 Post Free.**
(Current Disk-to-Disk users may return their disk and obtain a PLUS upgrade for £15 Post Free.)

**TAPE-TO-DISK UTILITY:** Transfer your single-stage machine code Tapes up to Disk and load them in a fraction of the time with 100% reliability. **TAPE £9.95 Post Free or on DISK £10.95 Post Free.**

**THE ALCHEMIST:** Transfer your single-stage **PLUS** many **MULTI-STAGE** machine code Tapes to Disk. The Alchemist will transfer to disk many games available only on cassette. (Current Tape-to-Disk users may obtain an Alchemist upgrade.) **S.A.E. for Details.**

**MENULOADER:** If used in conjunction with any of our Tape to Disk utilities, up to **TEN** transferred single-stage tapes may be placed onto **ONE** Menuloader disk, offering a convenient and economical method of storing your programs. Menuloader recreates itself, so you may create as many Menuloader disks as you require. **DISK £12.95 Post Free.**

**THE SECTOR INSPECTOR:** Scan your machine code disks for text. Adventure disks can be searched for clues and locations. Also features an Editor that enables you to replace existing text within a game with your own amusing or risque comments. Great fun! **DISK £8.95 Post Free.**

### SPECIAL OFFER

Buy any TWO of the following Tape backup utilities at the Special Offer price of ONLY £14.95 Post Free.

**TURBOCHARGER XL/XE:** Backup your single and multi-stage, Long and Short IRG Tapes, **PLUS a TURBOSAVE OPTION.** Your backup tapes will load faster than the originals. **TAPE £9.95 Post Free.**

**TAPE-TO-TAPE UTILITY:** This utility is similar to Turbocharger, but does not feature the Turbo facility. **TAPE £8.95 Post Free.**

**BINARY TAPE EDITOR:** Backup single and multi-stage, Long and Short IRG Tapes, **PLUS** an easy to use Editor that will allow you to **READ** and **ALTER** text within your machine code games. There is no longer any reason to be stuck on an Adventure, simply scan the tape for clues and locations. A Hackers delight! **TAPE £9.95 Post Free.**

**THE CASSETTE DUPLICATOR:** A tape backup utility written in Basic. The program may be LISTed. **TAPE £8.95 Post Free.**

**FREE OFFER:** Free machine code game with all orders over £20, state Tape or Disk. Overseas customers should add £2 to total order. Please enclose a S.A.E. with all enquiries. All software is supplied with full instructions.

Cheques and P.O's to:

**J.R. DAVIS**
**10 INGRAM AVENUE, HOLMER,**
**HEREFORD. HR4 9RD.**

## Autoduel

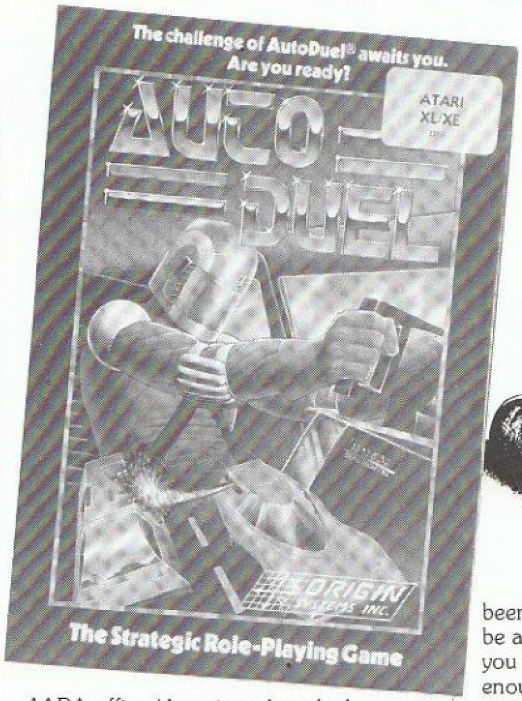
From Origin/Microprose
Disk for XL/XE Only, Price £19.95
Review by Brad Mountjoy

Autoduel is a rare thing in the world of eight bit Atari software at the moment. It's a really good program, a really thoughtful and well packaged game! Autoduel is from the Origin/Microprose stable and is everything we have come to expect from this simulation based software house. Autoduel comes on two disks (three sides) together with a 32 page manual/driver's guide, a road map of the northeastern seaboard of the USA circa 2030 and a toolkit, yes a genuine miniature toolkit!

The basic game is that you are a novice autoduellist in the 21st century, and you must improve your driving skills and fighting skills in order to survive, and to get out onto the road on courier missions or as a vigilante. As a courier you will carry valuable cargos from city to city, but as a vigilante you will be hunting road outlaws and cycle gangs. How do you get out onto the road? By going into the arena and fighting other cars, that's how! In this way you gain experience, but even more importantly you gain prestige and financial reward. With more money you can have yourself built a really powerful auto, with armour where you specify, weapons of your choice, a more powerful engine and so on. This is where Autoduel steps beyond a mere game into the realm of fantasy role-playing.

The manual goes into great detail on how to build your car, giving advice on vehicle design. This is where thoughtfullness and consideration go a long way, the vehicle you drive is a reflection of your skill, prestige and wealth. You can select your cars body style, chassis type, suspension, power plant (engine) and tyres, all of which will alter how you perform in the game. You can also select different weapons and where on your vehicle they will be mounted. Armour is an important choice and should not be taken lightly (no pun intended).

You start with no car of your own and are advised to make your way to the arena and put your name down for the Saturday night novice combats (a car is supplied by the arena). If you are good enough you can progress to the higher divisions and maybe one day take part in the 'city championship'. On the other hand after a few wins in the arena you may decide to get on the open road and visit some of the other 16 cities on the map (or one that isn't). Before you do however it is advisable to check at an


The challenge of AutoDuel® awaits you. Are you ready?
ATARI XL/XE
AUTO DUEL
ORIGIN SYSTEMS INC.
The Strategic Role-Playing Game

AADA office (American Autoduel Association) for road information, they will have the latest info on bandit activity. When you get to other towns there are plenty of things to do, apart from the arena contests of course. You can visit Joe's Bar and have a drink, but you are really there to listen for rumours (some of which will actually be true) or try to sell off any goods you have salvaged (I forgot to mention that after a combat victory you can ransack, I mean salvage, any weapons, armour, etc. for your own use) or maybe even sell the goods you've

been asked to courier (you don't have to be a goody goody in this game, mind you do lose prestige). Or if you have enough wealth you can visit the Gold Cross centre and have a clone of yourself made along with a braintape of all your skills up to date. Thus in the event you do met a faster, more ruthless autoduellist than yourself and you kick the bucket, your clone can take over. Or you can visit the casino and play poker or blackjack, a pleasant way to spend an evening.

All in all, Autoduel is great, complex, skilful, takes a time to learn to play but worth it, imaginative, well presented and a must for every 8 bit disk owner in the land!

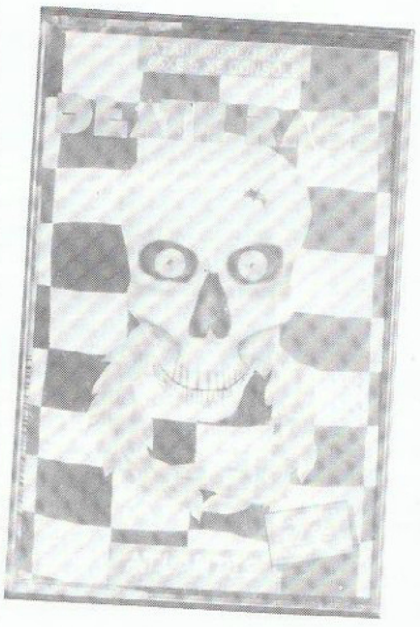
## Death Race

From Atlantis
Cassette, Price £2.99
Review by Otis Bey

Death Race is a straight forward road race game, similar to Pole Position. There are 70 cars in front of you and you only have 80 seconds to overtake them all. If you manage this without losing your 5 lives, extended play is awarded but this time you only have 60 seconds. Each extended play gives you bonus scores and an additional life. There are 5 different levels which can be pre-selected at the start.

Death Race gets no points for originality, but it is very playable as you can obtain simulated speeds up to 300 mph, and at these speeds the game is extremely hair-raising. Your reactions need to be lightning fast or a slight mistake will mean instant annihilation.

At £2.99 Death Race is very good value, the kids will love it! It is one of the better budget priced games available.



# CRACKING THE CODE

## by Keith Mayhew Part Twelve

### Character Mapped Modes

ANTIC mode numbers 2 through 7 provide the six different types of character display.

To recap, mode 2 is the standard text display mode which provides BASIC's GRAPHICS 0 mode. This offers a character cell eight scan lines high by eight pixels wide. The width of each pixel is the same as that in mode $F, GRAPHICS 8, (half a colour clock).

The colour scheme for this mode is exactly the same as that for mode $F with a bit set to 1 indicating that the luminescence value of COLPF1 should be used with the colour of COLPF2 for that pixel, normally you would want to keep the luminescence value up at maximum, 14 or 15 (bit 0 is ignored), so that character shapes appear virtually white.

Mode 3 is very similar to mode 2 but each cell is ten scan lines high.

Mode 4 has eight scan lines per cell and only four pixels in width. The data for each character is still eight bits wide but each pair of bits determine the colour of one of the four pixels, as in mode 3. Each pixel is one colour clock wide which makes each pixel twice the width of a pixel in modes 2 and 3.

Mode 5 is the same as mode 4 except that each pixel is made twice as high so that the whole cell takes 16 scan lines.

Mode 6, BASIC's GRAPHICS 1, has a cell 8 scan lines high by eight pixels wide with each pixel occupying one colour clock making this mode twice the width of mode 2. All the pixels in a character are the same, unlike modes 4 and 5 where four different colours can be displayed within each character.

Mode 7, BASIC's GRAPHICS 2, is the same as mode 6 but is 16 scan lines high.

### Character Sets

Before discussing the six character mapped modes any further we will quickly look at the common factor between them; the character set.

Register CHBASE, shadow CHBAS, is used to tell ANTIC where the character set is stored. CHBASE forms the high byte of the address with the low byte taken to be zero. Each character shape is stored in the character set as a group of eight consecutive bytes for all modes with the first byte being used for the top most scan line of a character cell. In modes 2 through 5, there are 128 such characters in each complete character set, thus the set occupies 1K bytes of memory. Modes 6 and 7 have only 64 characters in their sets and so only occupy 512 bytes.

As with players, display lists and screens, you can have as many character sets in memory at once and swap between them by changing the value in CHBASE. By using DLI's more than one character set can be displayed at once hence overcoming the limit of 64 or 128 characters for the mode being used. If you intend to make up many character sets then it is certainly worth having one of the character set generator programs which allow you to draw them on the screen with a joystick; it will be a lot easier and will save a lot of time.

### Boundaries

As with player/missile tables other data has to be aligned to certain boundaries in memory.

Character sets have to be aligned to 1K or 512 byte boundaries depending on which size character set is being used.

Display lists can start anywhere in memory but cannot cross a 1K byte boundary. Obviously it is best to start a display list on a 1K byte boundary so that you have the 1K maximum for your display list. Although it is possible to use the JMP instruction in a display list to cross a 1K byte boundary it has the side effect of creating a single blank line on the screen at that point. There is little use for this feature though, as all possible display lists will fit somewhere inside a 1K block.

The display data itself also has a restriction on where it can be placed. The Load Memory Scan, LMS, type instructions in a display list can cause data to be fetched starting from any location in memory, but as the data is fetched it must not cross a 4K byte boundary. It is often necessary to have more than 4K of screen data, e.g. BASIC's GRAPHICS 8 mode which requires nearly 8K. In such cases, the first LMS instruction is used as normal to start the display off, but when the 4K byte boundary is neared a second LMS instruction is used to re-load the counter to the next boundary. The problem this may now cause is that there is a gap in the screen memory at some stage, which can mean headaches when trying to calculate positions on the screen and their corresponding offsets into the display data. If this is a problem which you are faced with then a simple solution is to move the start of the display data so that the end of the 4K block corresponds to the end of one of the lines on the screen. The LMS instruction will then load with the next location, which is the start of the next boundary. By re-positioning the data in this way means that it can be kept contiguous, as if the boundary didn't exist.

If you are curious as to what happens if a boundary is violated the answer is simple; the data to be fetched after a boundary will actually be fetched from the start of the last block. That is they wrap around, analogous to when you increment a location from 255 to 0. As mentioned before, this was done to simplify the hardware design of ANTIC. It is something you will have to learn to live with!

### Displaying Data in Character Modes

The data stored in the screen memory for a character mapped mode consists of a series of bytes, each of which provide an offset into the character set to the appropriate character. A value of zero will display the character cell defined in the first eight bytes of the character set, one will display the next eight, and so on.

For example, BASIC's GRAPHICS 0 has forty bytes of data per line for each mode 2 instruction. Each value stored in this line represents the number of the character in the character set which should be displayed. The system has one character set which is stored at $E000 and extends to $E3FF, hence the value normally found in CHBAS is $E0. If you want to redefine just a few of the characters from the system set then your program can copy the system character set into RAM and then alter the appropriate characters and, of course, finally altering the value in CHBAS to point to the RAM version.

As an example of this, Listing 1 shows some code which copies the system character set down into RAM starting at location $4000. It then copies in two new characters which have been defined on lines 580 and 590. Rather than copy these two characters into the first location in the character set onwards it skips over the first character, i.e. $4008 onwards. This is simply because the first character is usually a space, i.e. eight

```
0100 ;Operating system shadows...
0110 CHBAS   =       $02F4   ;Character set base pointer.
0120 ;Operating system locations...
0130 SYSCHST =       $E000   ;System character set.
0140 SAVMSC  =       $58     ;Pointer to screen data.
0150 ;Page zero variables...
0160         *=      $CB
0170 SRC     *=      *+2     ;'source' pointer.
0180 DST     *=      *+2     ;'destination' pointer.
0190 ;Program equates...
0200 NEWCHST =       $4000   ;New character set address.
0210         *=      $0600
0220         PLA
0230         LDA     #SYSCHST&$FF ;System character set
0240         STA     SRC     ; pointer as source.
0250         LDA     #SYSCHST/256
0260         STA     SRC+1
0270         LDA     #NEWCHST&$FF ;New character set area
0280         STA     DST     ; pointer as destination.
0290         LDA     #NEWCHST/256
0300         STA     DST+1
0310         LDX     #4      ;Four pages to copy...
0320 NXTPAG  LDY     #0
0330 COPY    LDA     (SRC),Y
0340         STA     (DST),Y
0350         INY
0360         BNE     COPY    ;Copied page?
0370         INC     SRC+1   ;Next page.
0380         INC     DST+1   ;Next page.
0390         DEX
0400         BNE     NXTPAG  ;Finished all pages?
0410         LDX     #15     ;Copy sixteen bytes...
0420 CPYCHS  LDA     CHARS,X ;New character data.
0430         STA     NEWCHST+8,X ;Overwrite old data.
0440         DEX
0450         BPL     CPYCHS  ;Done?
0460         LDA     #1      ;Character 1.
0470         LDY     #0
0480         STA     (SAVMSC),Y ;Top left on display.
0490         LDA     #2      ;Character 2.
0500         LDY     #40
0510         STA     (SAVMSC),Y ;Under last character.
0520         LDA     #NEWCHST/256
0530         STA     CHBAS   ;Point to new char. set.
0540         RTS             ;All done.
0550 ;Data for two characters...
0560 ;When second character is placed under the first
0570 ; they (roughly) make a capital sigma sign.
0580 CHARS   .BYTE   $00,$00,$7E,$7E,$30,$18,$0C,$06
0590         .BYTE   $06,$0C,$18,$30,$7E,$7E,$00,$00
```

Listing 1.

bytes of zeros, so that the screen can be 'cleared' by writing zeros to it; as the system does.

The program finishes by placing the code 1 into the top left location of the screen (pointed to by SAVMSC) thus displaying the first of the redefined characters and the code 2 in the position under it, i.e. at 40 bytes further on from the first. Lastly, the shadow character set pointer CHBAS is set to the high byte of the address of the new character set, i.e. $40.

The result of the program is to display a bad attempt at a capital sigma sign (a Greek character in case you didn't already know). Listing 2 is, as usual, a BASIC program to load the machine code. After running it, the code can be executed by typing:

X = USR(1536)

## The Rest of the Story

Firstly, it is worth remembering that the system character set is not stored in ASCII order, or ATASCII, as it is known on the ATARI. For instance ASCII space, which is 32 decimal, actually occurs as character zero in the character set. Why the character set wasn't made to match up with ATASCII, I haven't the faintest idea! Anyway, the rule for converting ATASCII codes, which the operating system understands, into their character numbers is as follows:

| ATASCII | Character |
|---|---|
| \$00 - \$1F | \$40 - \$5F |
| \$20 - \$3F | \$00 - \$1F |
| \$40 - \$5F | \$20 - \$3F |
| \$60 - \$7F | \$60 - \$7F |

Thus when the O.S. is asked to print the upper case letter B, ATASCII $42, it has to convert it to $22 before saving the byte to the screen data so that a B actually appears. Note also, space, ATASCII $20, maps to $00 as mentioned above.

```
QZ 10 DIM HEX$(16)
CV 20 LINE=10000:TRAP 100:J=0:START=1536
VA 30 READ HEX$,CHKSUM:SUM=0
AA 40 FOR I=1 TO 15 STEP 2
ZG 50 D1=ASC(HEX$(I,I))-48:D2=ASC(HEX$(I+
   1,I+1))-48
KT 60 NUM=((D1-7*(D1>16))*16+(D2-7*(D2>16
   )))
LW 70 SUM=SUM+NUM:POKE START+J,NUM:J=J+1:
   NEXT I
LY 80 IF SUM=CHKSUM THEN LINE=LINE+10:GOT
   O 30
IN 90 ? "Checksum error on this line:"
VO 95 LIST LINE:END
YS 100 PRINT "Data in memory."
YB 10000 DATA 68A90085CBA9E085,1135
BS 10010 DATA CCA90085CDA94085,1077
SQ 10020 DATA CEA204A000B1CB91,1057
WY 10030 DATA CDC8D0F9E6CCE6CE,1732
FT 10040 DATA CAD0F0A20FBD4006,1086
JT 10050 DATA 9D0840CA10F7A901,864
IM 10060 DATA A0009158A902A028,764
AN 10070 DATA 9158A9408DF40260,949
BD 10080 DATA 00007E7E30180C06,342
XB 10090 DATA 060C18307E7E0000,342
```

Listing 2.

In ANTIC modes 2 and 3 the lower seven bits of a byte, select one of 128 characters. The eighth bit in these modes is usually used to give 'inverse-video' characters, that is the fore and background colours are swapped over. The actual operation when bit 7 is set to 1 is determined by the register CHACTL, shadow CHART. Bit 3, if set to 1, causes all characters, regardless of mode, to be displayed upside down, whether bit 7 is set to 0 or 1. This can be useful if you want a screen which mirrors the top half in the bottom half of the screen; you would copy the characters in reverse line order for the bottom half and use a DLI to flip bit 2 of CHACTL (not the shadow, of course) to a 1 to reflect the characters themselves. Bit 1 causes characters with bit 7 set to be displayed in inverse-video. The cursor on the standard text screen is implemented by flipping bit 7 of the character under the cursor while bit 1 of CHACTL is set to 1. Bit 0 of CHACTL causes the character, if it has bit 7 set, to become empty, i.e. space. Thus if you have a string displayed in inverse-video and flip bit 0 of CHACTL on every few vertical blank interrupts then the characters will be flashing on and off. Setting bits 0 and 1 to 1 will cause a character to be displayed as an inverse-video space.

Mode 3 lines are displayed in exactly the same way as a mode 2 line except that an extra two empty scan lines are added at the bottom of each character. However, if characters are used from the last fourth of the character set, i.e. codes between $60 and $7F and between $E0 and $FF, then the two empty scan lines are displayed at the top and the eight character data lines below it. This mode was added primarily so that lower case letters could have proper descenders.

Modes 4 and 5 , often referred to as multi-colour modes because they can have more than one colour inside a single character cell, allow upto 4 colours to be displayed in any one character. A pixel value of zero (from a pair of bits) causes COLBK to be used, 1 uses COLPF0, etc., as in the bit mapped modes which were discussed last time. Bit 7 is used in these modes to change the colour register used by a pixel value of 3. Normally, with bit 7 set to 0, COLPF2 would be used, but if bit 7 is set to 1 in a character then COLPF3 is used instead. Although there are still only four colours per character it is now possible to use a total of five colours for displaying character screens.

In modes 6 and 7 the lower six bits of a byte are used to select one of 64 characters. The top two bits determine the colour of the whole character. Values from 0 to 3 cause registers COLPF0 to COLPF3 to be used respectively. Where zero's occur in the character data itself, then COLBK is displayed.

## Scrolling

We will finish this time with an introduction to scrolling techniques; they will be covered in more detail next time.

To scroll data on most machines, including the ATARI ST, the actual data in memory must be moved around, usually by the processor, to achieve the effect of scrolling text or background for a game. This is not only costly on the processor time but is rather a waste of time in itself. With ANTIC, scrolling is made easy and quick by changing the position of the screen relative to the data, rather than the other way around. Thus to vertically scroll a screen, all that is necessary is to change the address after the first LMS instruction. For instance, adding forty to the address in BASIC's GRAPHICS 0 mode will cause data to be fetched from one line further on, thus the image will appear to have scrolled up the screen. It is up to you whether you prefer to think of the data scrolling up the screen or the screen scrolling down over the image!

Horizontal scrolling can be achieved in a similar way by just adding or subtracting one from the LMS address. Unfortunately, you will need an LMS on each line and to change each individually to do this properly, as we will see next time.

As you might have guessed such techniques will cause a coarse scroll of characters, i.e. a character at a time, or a pixel at a time for bit mapped modes. To scroll data more finely calls for the use of the registers VSCROL and HSCROL.

VSCROL, when enabled by setting bit 5 of a display list instruction to 1, causes that line to be scrolled up the screen by the number of scan lines specified. Thus with VSCROL set to 0 the line stays where it should, but as VSCROL is increased the line moves pixel by pixel up the screen. When the line is one pixel away from being at the next line's normal position, VSCROL should be set back to zero and a coarse scroll performed using the LMS.

HSCROL is used in a similar manner to shift a line pixel by pixel to the right, again using the LMS for coarse scrolling. The actual amount the data is shifted by is actually a colour clock, i.e. two GRAPHICS 8 pixels, as with players and missiles.

To fine scroll data in the opposite direction to that described above it is a simple matter of decrementing the value in the scroll registers instead of incrementing them and changing the LMS's appropriately. Fine scrolling can also be used in bit mapped display modes but their most common use is for scrolling characters.

Next time we shall look at how to put this theory into practice with a few sample programs.

ATARI ONLY*

**STOCKSOFT**

**SUPPORTING THE EIGHT BIT**

MAIL ORDER ONLY

**WORLD FAMOUS HAPPY!**
**DISC DRIVE ENHANCEMENTS**
**HAPPY REV 7**
This is the latest enhancement for the 1050 disc drive, to come from the world famous "HAPPY" people in the U.S.A. It is a replacement controller board for your 1050 drive, full fitting instructions supplied, which are basically remove drive cover, unplug 2 chips, plug in Happy.
GIVES THE FOLLOWING
Makes your drive Read/Write faster.
Back up including B/Sectors.
True Double Density
Fast Sector copier 128k
Warpo Speed D.O.S.
Compacter of Files.
Drive Diagnostic.
D.I.Y. Kit only £169 POST FREE
**1050 drive fitted with Happy only £299 post free.**

**CHEER-UP**
A replacement chip turns a 1050 duplicator into a Happy, complete with Rev 7 software, will still allow you to run dupes Rev 4 software **£49 post free**

**HAPPY "OPTIONAL EXTRA"**
1050 Controller **£49 post free** for use with Happy enhanced 1050 drives only controls R/W Speeds, Write protect, switches and L.E.D. no soldering – full instructions

**EXTRA P.D.B FILES**
for your HAPPY, a double sided disk by "DupeTec New York". Packed with new P.D. B. Files. Also runs on "Warp Lazer" and "Hyperdrive" enhancements **£20 post free**

**WARP DRIVE REV 3**
UK produced replacement control board for 1050 drive, fitting as Happy
**GIVES THE FOLLOWING:**
Back up including most types of bad sectors
True Double Density
PLUS True D/D Menu Maker
True D/D D.O.S.
NOW with the ability to run Archiver Progs. on the 1050 drive
PLUS P.D.B. Files DIY Kit only
**£80 post free**

*ABOVE ENHANCEMENTS SOLD SUBJECT TO NOT INFRINGING COPYRIGHT PROGRAMS.*

**1050 DUPLICATOR REV 4**
Based on a 6502 chip, is a replacement board for your 1050 drive. Easy 15 minute installation, compatible with both XL and XE.
Gives – read and write to B/sectors, custom formats, double sectors, also true double density, and turbo-drive giving up to 18 times faster read and write.
Comes complete with all hardware, software, fitting instructions **£159** Rush your order now! Or send large S.A.E. for further information.
1050 Drive with Dupe fitted only **£300 post free**
REV 4 UPGRADES for 1050 DUPE
now in stock **£20 post free**

**SPECIAL OFFERS**
The following Disk operating Systems
**Only £2 each post free**

| | |
|---|---|
| (1) D.O.S. 2 | (6) D.O.S. 4.00 |
| (2) D.O.S. 2.5 | (7) UTILITY D.O.S. |
| (3) D.O.S. 2.5F | (8) ART D.O.S. |
| (4) D.O.S. 2.6 | (9) MACH D.O.S. |
| (5) D.O.S. 3.00 | |

**FREE 24 PAGE CATALOGUE**
Games tapes or disks
Utilities and enhancements
everything for the 8 bit
send large S.A.E.

**STOCKSOFT**
**15 WOODBROOKE ROAD, BIRMINGHAM B30 1UE**

# Plotting in Zero

**By Ron Levy**

If you are writing yourself a program which makes use of a Graphics Zero screen to print text, it is often nice to 'jazz up' the presentation of your screens. You can do this by printing rows of characters at strategic points, such as borders, or perhaps underlining headings. You can use the standard characters for this such as asterisks, the dot or hyphen, etc., or you can use the special Atari Graphics Characters.

So how then would you put these rows of characters onto your Graphics Zero screen? Well, the obvious way is by using the PRINT command. For example, to print a line of asterisks on the screen you might do the following:

```
PRINT "******************************"
```

I hear you say: 'Okay, so it's messy, but it works!'

But, what if you want to produce a vertical line, or even a diagonal line, what a chore it then becomes!

Did you know that instead of using PRINT, you can use the PLOT and DRAWTO commands, just as if you were drawing on a pixel screen? To specify the character you want plotted you use the COLOR command. Take a look at Listing 1.

The program simply draws a horizontal line 40 asterisks across the top of the screen, and then a diagonal line of asterisks. The POKE 752,1 switches off the cursor. This is necessary because drawing on Graphics Zero tends to leave nasty cursor blocks all over the screen! The cursor is re-enabled by the POKE 752,0 command, or by pressing the BREAK or SYSTEM RESET keys. The other point to notice is the COLOR 42 command. The number 42 represents the ASCII value for the asterisk character. You can change this to any value you like from 0 to 255, but if you want to use a particular character you will need to know its ASCII value.

## Finding ASCII Codes

You can determine the value of a particular character by using the ASC command, or by looking in the back of your computer's manual, where there is (usually!) a table of these. Here is an example showing you how to use the ASC command in order to work out the value of the letter B:

```
PRINT ASC( "B" )
```

Which should, of course, print the number 66. In fact, instead of entering just COLOR 42 for line 30, we could actually change this to the following:

```
30 COLOR ASC( "*" )
```

## Graphic Characters

There are a set of graphic symbols which can be used to draw straight lines or patterns, and to help you to identify these and other ASCII characters, I have included Listing 2. This displays a table of the Atari's characters and their corresponding values. Remember, that the characters numbered 128 to 255 are the inverse video versions of those numbered 0 to 127, another point to watch out for is that some of the character set are screen control codes, they perform functions such as clearing screens, and inserting or deleting lines. In fact, these are the reason that I have printed character 27 before each character. This is the ESCAPE code, it disables the screen control codes to stop them disrupting the screen display.

As a final example, type in and run Listing 3. This shows you how to add a simple line border around the screen. See how easy it is using PLOT and DRAWTO on Graphics Zero!

```
JA 0 REM ....Program 1....
LP 10 GRAPHICS 0
WB 20 POKE 752,1:REM Cursor off.
VD 30 COLOR 42
KU 40 PLOT 0,0:DRAWTO 39,0
WW 50 PLOT 0,0:DRAWTO 20,20
VS 60 POKE 752,0:REM Cursor off.
```

Listing 1.

```
JT 0 REM ....Program 2....
JP 1 REM Prints A Table Of ASCII Codes.
NH 2 REM
IL 10 OLD=PEEK(561):DL=PEEK(560)+256*OLD
RH 20 DL1=DL-256
DC 30 FOR A=1 TO 6:POKE DL1+A,PEEK(DL+A):
   NEXT A
GF 40 FOR A=6 TO 50 STEP 2:POKE DL1+A,0:P
   OKE DL1+1+A,2:NEXT A
MB 50 FOR A=52 TO 53:POKE DL1+A,PEEK(DL+A
   -23):NEXT A
RP 60 POKE DL1+54,PEEK(561)-1
EH 70 POKE 561,OLD-1
DC 100 REM ..Now print the ASCII codes..
UA 110 PRINT CHR$(125)
SL 120 FOR X=0 TO 5
XB 130 FOR Y=0 TO 21
TT 140 C=X*22+Y
QR 150 POSITION X*6,Y
JG 160 PRINT CHR$(27);CHR$(C);"=";C;
MJ 170 NEXT Y
MB 180 NEXT X
```

Listing 2.

```
KM 0 REM ....Program 3....
WJ 1 REM Draws A Box Around The Screen.
LP 10 GRAPHICS 0
WB 20 POKE 752,1:REM Cursor off.
LD 30 COLOR 14:REM Top of screen.
KI 40 PLOT 1,0:DRAWTO 38,0
MM 50 COLOR 22:REM Right of screen.
MU 60 PLOT 39,1:DRAWTO 39,22
UG 70 COLOR 13:REM Bottom of screen.
MU 80 PLOT 38,23:DRAWTO 1,23
GC 90 COLOR 2:REM Left of screen.
LV 100 PLOT 0,22:DRAWTO 0,0
ID 110 POSITION 3,1
WK 120 PRINT "GRAPHICS ZERO DEMONSTRATION
   SCREEN"
OS 130 COLOR 13:PLOT 3,2:DRAWTO 37,2
XO 200 POSITION 17,10
LN 210 PRINT "DONE"
VY 220 POKE 752,0:REM Cursor on.
```

Listing 3.

# EIGHT BIT SOFTWARE

**Software Librarian - Roy Smith**

There are two ways to get programs from the library. You can use the donation scheme by sending in a disk or cassette of your own, or if you have a program of your own which you would like to add to the library you can exchange it for 3 programs of your choice. The rules are as follows:

## 3 FOR 1 EXCHANGE

1. Every program you donate entitles you to three programs in return.

2. The program you donate must be your own original and not copied.

3. Your donated program must be submitted on a cassette or a disk, programs in the form of print-outs cannot be processed.

4. If your program requires any special instructions they should be added in the form of REM statements within the program (or you may present them as instructions when the program is actually run).

5. BONUS. Every program submitted per quarter (between issues of the magazine) will be eligible to be judged 'STAR PROGRAM' for that quarter. This carries a prize of £10 which will be paid to the author. The programs will be judged by the Editorial Team and their decision is final. The Editorial Team are not eligible for the prize.

6. Please include 30p in stamps (or cash) to cover return postage.

7. The '3 for 1' exchange is only open to club members.

## DONATION SCHEME

1. Every club member can make a donation to the club, at any time, if he/she wishes to obtain a particular program(s).

2. There is no limit on the number of programs that can be asked for at any one time. (If you are asking for a lot of programs at once, please ensure that you send a sufficient number of disks or cassettes. It's better to send too many than not enough.)

3. Please include 30p in stamps (or cash) minimum to cover return postâge. If your parcel costs more than 30p to send to us, please include an amount equal to that of the postage, so that we may return your parcel to you without delay. Overseas members should add an extra £1 to cover postage costs.

4. The donation fee is £1 per program. Cheques or Postal Orders should be made out to the 'U.K. Atari Computer Owners Club'.

5. You should send in blank disks or cassettes, ensuring they are properly packed to prevent damage in the post. State which programs you require and remember to give your name and address. Also remember to include the fee and return postage.

6. The 'Donation Scheme' is only open to club members.

**THE LIBRARY SOFTWARE SERVICE IS FOR MEMBERS ONLY**

# *LIBRARY SOFTWARE TITLES*

Listed below are the software titles received by members for inclusion in the library since the last issue was published. As the library now contains over 350 programs, it is getting too big to print the entire list. For those of you who are new to Monitor and are unaware of what is available, then send for a photocopy of the complete list which is available from the librarian. There is a small charge for this service to cover photocopying costs. If you would like a list send 50p and a S.A.E. for return.

## Home Entertainment

### CARD GAMES COMPENDIUM

by Les Howarth - Preston.
Seven different versions of Patience, 2 versions of Whist, plus Stud Poker and other card games.
Runs in 48K min. Disk only.
Requires 1 side of a dual density (1050) disk.
Quote XL or XE version.

### SUPERHEROS TOUCH TABLET SHOW

by E. Tilsley - Eastcote.
Picture show starring Superman, Spiderman, Ironman and others.
Runs in 48K min. Disk only.
Requires 1 side of a disk.

## Demos

***** STAR PROGRAM *****

### FRACTALS

by Adrian Cox - Reading.
Super fractal display, written in Turbo Basic.
Runs in 48K min. Disk only.
Requires both sides of a disk.
XL/XE machines only.

*****************

## Utilities

### CATALOGER & INDEX

by C. Rischbiter - Bury St. Edmonds.
A suite of programs designed for use in cataloging and indexing your disk collection.
Runs in 48K min. Disk only.
Requires 1 side of a disk.
XL/XE machines only.

### MINI DOS

by Adrian Barton - Welwyn Garden City.
Includes delete, rename, lock & unlock, format single and dual density, directory and print directory.
Runs in any size.
Disk only.

### RECORD COLLECTION DATABASE

by David Leyshon - Cambridge.
Keep track of your record collection. Comprehensive database includes sort and search facility.
Runs in 48K min. Disk only.
One side of disk required.

### SIMDOS

by J. T. Bean - Colwyn Bay.
Includes lock/unlock, rename, format, delete and directory.
Runs in any size. Disk only.

### TURBO FILER

by Erik Mareels - The Netherlands.
A super name and address filer written in Turbo Basic.
Runs in any size.
Disk only.

## Education

### EASY MATHS

by Basil Spooner - Fishguard.
Simple sums, including multiplication and division.
Runs in 16K Disk or Cassette.

## Music

### MUSIC FILES

by Trevor Skeggs - Milton Keynes.
Four tunes for use with the Atari Music Composer cartridge, 633 Squadron, the Magnificent Seven, High Noon and Myditty.
Runs in 16K Disk or Cassette min.

# ST PROGRAMMING

## By Keith Mayhew Part Two

### Workstations

As was stated last time, GEM consists of the VDI and the AES. VDI, the Virtual Device Interface, has a vast number of functions for drawing everything from circles, lines and text to filled self-intersecting irregular polygons! Furthermore, the VDI allows the output to go to any workstation you like. The term 'workstation' refers to a device, such as the screen or a printer, which is capable of producing graphical or text output.

On the ST, the screen workstation is opened by the AES, Application Environment Services, so that it can use the VDI to draw its windows, menus and dialogue boxes, etc. As a workstation can only be opened once, unless it had been closed, your program can't use the screen workstation. Well obviously that isn't exactly true, otherwise you wouldn't be able to write a GEM application!

The workstations are actually referred to as physical workstations; as they are physical devices. The screen physical workstation can also have virtual workstations opened. Each of the virtual workstations can be used totally independently of each other and each appears to act just like a physical workstation. What happens on the screen is that all the images from the workstations are drawn on the same area. It is therefore the responsibility of the programmer to ensure that only the 'correct' parts of the screen appear, otherwise output from one workstation will overwrite the output from another.

As an example, the AES uses the physical workstation to draw its window borders, i.e. outline, and it is the responsibility of the application program to send its output only to the inner area of that window. This is only a convention and no harm will come from drawing all over the screen; but it might well look messy! Also, don't forget that the AES allows more than one program to be in memory simultaneously in the form of one application program and none or several desk accessories. If programs are not written well, they will have a tendency to interact with one another producing annoying side-effects, so it is important to consider carefully how your program uses the system. If you are not writing a GEM application then you are probably going to take over the control of the system and so you will be free to do as you want.

The main advantage of having virtual workstations is that each of them can have different output parameters, such as colours, italic text, dashed lines, etc. In this way, your program can open several virtual workstations and set each of them up differently. This will save you having to keep on changing the parameters every time you want different effects. There is no real limit to the number of virtual workstations you can open at any one time; about eighty seems to be the limit(!), but most programs use just one.

### A Simple Program

To show the basic structure of a program which uses GEM, we will now look at a simple example. Listing 1 is a small file with some C definitions. It provides some types which might help if you want to transfer this program to another compiler; you should choose between 'char', 'short', 'int' and 'long' for your compiler so as to provide 8-bit, 16-bit and 32-bit storage for BYTE, WORD and LONG respectively. VOID will be used as a function type when it doesn't return a value. BOOLEAN will be used whenever a variable is to be used as a logical value, i.e. TRUE or FALSE.

The configuration given for the different types was used for the Megamax C compiler. If Metacomco have their new Lattice C compiler available in time, I will probably make sure that programs work with both compilers next time. It is advisable that you should use similar definitions in your own programs as you might want to transfer it to another compiler at a later date.

Listing 2 is our sample program. The first thing it does is to include the definitions file, assuming it's called 'defs.h'. Note that the global arrays 'contrl', 'intin', 'intout', 'ptsin' and 'ptsout' have to be declared because they are used by the GEM library supplied with C compilers (not just Megamax). The 'main' routine, where all C programs start up, starts and ends with calls to 'appl_init' and 'appl_exit' respectively. These are mandatory if you are going to use any AES routines and expect them to work properly. If you don't use the AES at all in your program then these are not strictly necessary but can still be included.

After the 'appl_init' call a virtual workstation is opened for the screen device and its 'handle' is assigned to the global variable 'vdi_handle'. The majority of the rest of the code uses the VDI to clear the entire screen and draw a box 16 pixels in from the edge of the screen. An underlined text string is then printed in the upper-left hand region of the screen.

It is worth remembering that you should always hide the mouse before drawing on the screen and show it again afterwards, otherwise you are likely to see a block of the old screen image appear when the mouse is next moved. This is done by the two calls 'mouse_off()' and 'mouse_on()' which are defined as macros to expand into the two AES calls 'graf_mouse(256, 0L)' and 'graf_mouse(257, 0L)' respectively; the macros are used because they are more readable. Before closing the virtual workstation and terminating, a call is made to the AES routine 'evnt_button' which waits for some type of click event to occur on the mouse. In this case it returns when the left mouse button is pressed.

The function 'open_vwork' used in 'main' asks AES for its physical workstation 'handle' by the 'graf_handle' call. A 'handle' is the term used to identify a workstation, once it has been opened, in all further calls to the VDI, it is just an integer number. The 'v_opnvwk' call is used to open the actual virtual workstation. It passes in the address of the variable holding the physical handle and afterwards it contains the new handle. Note that if you want to open any more virtual workstations you must pass in the physical handle again and

```
#define  VOID                /* No return type.     */
#define  CHAR      char      /* Character.          */
#define  BYTE      short     /* 8-bit integer.      */
#define  WORD      int       /* 16-bit integer.     */
#define  LONG      long      /* 32-bit integer.     */
#define  BOOLEAN   int       /* Boolean.            */

#define  FALSE     0         /* Boolean false.      */
#define  TRUE      1         /* Boolean true.       */
```

Listing 1.

```
   v_gtext(vdi_handle, 32, 32, "Test");   /* Draw test string. */
   mouse_on();                             /* Show mouse again. */
                                   /* Wait for a button event... */
   evnt_button(2, 1, 1, &dummy, &dummy, &dummy, &dummy);
   v_clsvwk(vdi_handle);           /* Close virtual workstation. */
   appl_exit();                    /* Let AES know were going.   */
}

/* Routine to clear an area of the screen i.e. fill a rectangle
   in background colour. Usually white in high-res.
*/
VOID            clear_area(x1, y1, x2, y2)
WORD            x1, y1, x2, y2;
{
   WORD            xy[4];

   xy[0] = x1;     xy[1] = y1;
   xy[2] = x2;     xy[3] = y2;
   vsf_interior(vdi_handle, 0);  /* Background fill colour.    */
   vr_recfl(vdi_handle, &xy[0]); /* Fill the area.             */
}

/* Function to return a virtual workstation handle.   */
WORD            open_vwork()
{
   WORD            work_in[11], work_out[57];
   WORD            handle, dummy, i;

   for (i = 0; i <= 10; i++)
      work_in[i] = 1;   /* Set default values to 1.            */
   work_in[10] = 2;     /* Co-ordinates are in raster pixels.  */
            /* Get the AES's physical workstation handle.      */
   handle = graf_handle(&dummy, &dummy, &dummy, &dummy);
   v_opnvwk(&work_in[0], &handle, &work_out[0]);   /* Open it. */
   return (handle);                            /* Return handle. */
}
```

```
#include "defs.h"

/* VDI global variables for C bindings. */
WORD            contrl[12];
WORD            intin[128], intout[128];
WORD            ptsin[128], ptsout[128];

WORD            vdi_handle;

#define  mouse_off()    graf_mouse(256, 0L)
#define  mouse_on()     graf_mouse(257, 0L)

/* Initialise as a GEM AES application and use the
   VDI to clear the entire screen and draw a box.
   Print a small message and wait for a button event
   before terminating.
*/
main()
{
   WORD            xy[10];
   WORD            dummy;

   appl_init();                /* Register with AES.             */
   vdi_handle = open_vwork();  /* Open a virtual workstation.    */
   mouse_off();                /* Hide the mouse.                */
   clear_area(0, 0, 639, 399);       /* Clear high-res screen.   */
   xy[0] = 16;          xy[1] = 16; /* Build co-ordinates of     */
   xy[2] = 639 - 16;    xy[3] = 16; /*  a border box.            */
   xy[4] = 639 - 16;    xy[5] = 399 - 16;
   xy[6] = 16;          xy[7] = 399 - 16;
   xy[8] = 16;          xy[9] = 16;
   v_pline(vdi_handle, 5, &xy[0]);                  /* Draw it. */
                               /* Set some text attributes... */
   vst_alignment(vdi_handle, 0, 5, &dummy, &dummy);
   vst_effects(vdi_handle, 0);
```

Listing 2.

must NOT use this altered value. The arrays 'work_in' and 'work_out' are used to pass values in and out of the VDI. The 'work_in' values must be initialised and set default colours, line thickness and so on for the new workstation. The important entry is 'work_in[10]' which sets the workstation to raster co-ordinates, i.e. the co-ordinates refer to the actual pixels on the device. The alternative is normalised device co-ordinates (NDC) which addresses a device with 32768 pixels in both directions and are scaled to the actual device co-ordinates, thus making programs easier to write but slower. A value of 0 in 'work_in[10]' selects NDC mode, but NDC won't work on ST's as the routines were never placed in ROM (GDOS releases might overcome this problem when it becomes available).

## Your Enquiries

We will stop at this stage so that you can try out the program on your C compiler and try some experimentation. In the future, comments in programs will be greatly reduced so that they are more readable and any relevant explanations will be given in the text. If you have any problems with this, or C in general, then write in and I'll do my best to answer them. Of course, I will try and answer queries on other problems too, so don't be frightened to ask!

David Derrick asked how the mouse form, i.e. its shape, can be changed. The answer is through the use of the AES's 'graf_mouse' call. The general form of the call is 'graf_mouse(type, address)' in C. Note that 'type' is a word, or 16-bit value with meanings as follows:

0 - Arrow.
1 - Text cursor or I-bar.
2 - Busy bee.
3 - Pointing hand.
4 - Flat hand.
5 - Cross-hair.
6 - Thickened cross-hair.
7 - Outlined cross-hair.
255 - User-defined form.
256 - Hide mouse form.
257 - Show mouse form.

Also note that 'address' is a long, or 32-bit value which is ignored except when 'type' is 255, when it contains the address of a user-defined mouse form, which might be explained next time. Note that calls to show and hide the mouse can be nested so that an equal number of each must be performed to display the mouse again.

The program below shows how to change the mouse form into an I-bar using ST BASIC, as requested. Unfortunately, I had to learn ST BASIC for this one; it is my first program in ST BASIC!!!

```
100 REM Change mouse form.
110 params# = gb
120 gintin = peek(params# + 8)
130 poke gintin, 1
140 gemsys(78)
```

The value in 'gintin' corresponds to 'type' in the C call. The 78 is the number of the function for 'graf_mouse'. Next time I might explain how these so-called bindings work, until then, have fun.

# READ ALL ABOUT IT

**By Bookworm**

## Atari ST The Advanced Programmer's Guide

### Author: Mark Harrison
### Publishers: Sigma Press

This book has no fewer than two hundred and ninety pages divided into twenty chapters and three appendices, including the obligatory ASCII character set! The book is described as a complete programming course for the ST, covering ST Basic, ST LOGO and C.

The first four chapters are virtually a re-write of the Owner's Manual which, very annoyingly, occupies no fewer than the first thirty four pages! The next chapter is devoted to GEM Architecture and TOS. If you were thinking that we can now get down to some serious reading, you are wrong. Only a single paragraph is devoted to the GEM VDI, whilst another is for the GEM AES! Four paragraphs are used to describe TOS and another couple to BDOS, BIOS and CCP.

The next eight chapters are devoted to ST Basic. Unfortunately, yet again, we are treated to another re-write, this time the ST Basic Reference Guide! No fewer than twenty eight pages are devoted to this epic re-run!

The next seven chapters are devoted to Strings and Character Manipulation, I/O Techniques, Computers, Numbers and Mathematics, Techniques for Sorting, GEM from ST Basic and Data Storage on disk.

The programming examples are sound, but I would hardly rate it as an ADVANCED COURSE!

The next chapter introduces the course on ST LOGO and, once again, we have another re-write of the ST LOGO Reference Guide! No fewer than thirty four pages are wasted in this. Quite honestly, it is getting as bad as watching the telly. I read books to get away from the box!

No fewer than two whole chapters and twenty nine pages later we have our Advanced Programmers Guide to ST LOGO dutifully and methodically covered. Or do we? You're right, we don't!

Now we are treated to the same guide to C. The first chapter goes into a brief description of Variables, Arrays, Strings and Pointers, etc., etc. It even finds time to squeeze in a crash course in C programming. Mind you, it wasn't hard to do, it only covers two pages, just!

Then, for some reason best known by the author, we are given a review of three C software packages. Annoyingly, it uses up another eight pages.

Now we get down to the nitty gritties of Advanced C Programming! A whole chapter covering C Libraries! Isn't this fun?

The next chapter demonstrates the writing of a GEM application. At last, you are thinking, we can now get down to some serious reading. Wrong again, we have reached the end of the book. Still, we can always refresh our knowledge of the ASCII character set!

Oh well, it is back to the telly. You would not believe it, but BBC are doing a re-run of the Waltons and ITV are transmitting a re-run of Sons and Daughters!

I would, if I was feeling particularly generous, give Mr. Harrison one mark out of ten for effort. This book should be re-titled 'a beginners guide to programming if you have mislaid your documentation and have nothing to watch on the telly'. I would then give him eight out of ten.

## The Atari ST Users Guide

### Author: John Heilborn
### Publishers: Osborne McGraw-Hill

This book has been written as a guide for newcomers to the wonderful world of the ST. Judging from the number of visitors to the recent Atari Show at the Novotel struggling out of the building clutching, grim faced and perspiration pouring, an ST and Monitor there are quite a few newcomers about, and this is the ideal book with which to unlock some of its magic.

Only about forty five, out of the two hundred plus pages are actually devoted to the ST! The remainder are almost entirely devoted to ST LOGO. Why this fact could not be announced on the front cover alludes me. ST LOGO has not had a great deal written about it. The documentation supplied with the language is basically a reference manual, it doesn't go very far into LOGO power. This little book goes a long way into correcting this. Like the Tim Knight book, there is a substantial amount of padding, but I feel in this instance it is justified.

The book is well written and printed. I am pleased to recommend this book to any newcomer, young or old.

## Understanding Atari ST Basic Programming

### Author: Tim Knight
### Publishers: Sybex

This very nicely produced book was kindly sent to us for review by Computer Manuals Ltd., 30 Lincoln Road, Olton, Birmingham. It is very well written, three hundred plus pages, which are divided into eight chapters and four appendices. It is a tutorial that covers many aspects of programming using the ST Basic language, which is part of the free package included with the purchase of the ST.

A good tutorial should always take the reader gently, but soundly, from the 'known' into the 'unknown'. Another feature that should be mandatory is the inclusion of a summary at the end of the chapter. Although it may seem rather strange, it is the summary that should be read first. The reason for this is to prepare the brain for new words, expressions and concepts. When the chapter, to which the summary refers is read, the brain has been conditioned, learning and understanding are greatly simplified.

It was very nice to see that this book did contain good summaries to the majority of its chapters. On a marks out of twenty concept, it already has earned five and we haven't even started to read it yet!

Unfortunately, no fewer than thirty pages are devoted to information that is to be found within the ST manual. This, to me anyway, is totally needless, wasteful and a bit of a cheek! It represents 10% of the book and 10% of the cost!

Now that the niggles are out of the way, let's get down to business. The chapters cover Fundamentals, Words and Numbers, Graphics, Sound and Music, Advanced Graphics and Text, Disk Drive and Printer, Programs and Problems. The Appendixes cover Keywords, Error Messages, Installation, the usual ASCII character codes and New Keywords.

The tutorial enables you to work alongside the author, where you will modify and enter programs covering the topics just mentioned.

I worked through most of the examples in the book and I could not fault the examples chosen, nor the manner in which I was being taught. I was disappointed that there was no attempt to cover MIDI. Perhaps this was accidentally omitted, while the padding of ST manual, ASCII character codes and Keywords was being written. Somehow, I think not!

I liked this book, apart from the above remarks. I feel the book is far better than the program under discussion. I do not like ST Basic, I never have and I never will! It was a labour of love to boot it up and run through the tutorial.

If, on the other hand, you like ST Basic and would like to improve your knowledge and understanding, then I would thoroughly recommend it. It is an excellent tutorial. Personally, I am waiting for Mr. Knight to be given a copy of Computer Concepts 'FAST BASIC' and write a tutorial for that. If he does, I sincerely hope that he omits all the padding!

## Mastering the Atari ST

### Author: John M. Hughes
### Publishers: Sigma Press

This is the second Sigma Press book in this short review of some of the books dedicated to the ST that I have covered. The same subject, yet they are worlds apart in quality and usefulness. Please do not think that this book is without its faults, it has them, but more about that later.

This book has two hundred and thirteen pages, twelve chapters and a single appendix. You will be relieved to find that it does not include an ASCII character set, but instead something much more useful, a brief tour round the Data Protection Act of 1984, in plain language.

Its aim is to give an insight to newcomers, in the main, of some of the common applications that the ST is capable of handling, in the home and business world. These are: Word Processing, Data Bases and the Spreadsheet.

The Software chosen as examples are 1st Word and 1st Word Plus from GST, LaserBase from LaserBase Int. and Professional VIP, released through Silica Shop.

I am only very familiar with three out of the four applications, the exception is LaserBase, so I am in a good position to appraise Mr. Hughes' arguments.

The first six chapters cover, in quite simple, but thorough explanations, the story of the computer, file organisation, GEM and Application Systems. This is not a re-write of the User's Manual but a fresh approach that does not take anything for granted with regards to what the user knows, or does not know. I do not think that any owner, or a user such as a secretary, would have any difficulty in following through these chapters without gaining a much clearer understanding of what is actually taking place.

Of the many types of application that are available, I suppose the most frequently used and abused is the Word Processor. A printer of some sort will usually be tacked into the system too.

The Author does an admirable job of explaining in some detail, how one can get the most out of the printer and consequently, the applications. Printer configuration and a thorough explanation of HEX coding, are tackled very well. Also, a good description of the types of printer that are available, including their individual problems and value.

There now follows a four chapter discourse on 1st Word and 1st Word Plus and I must say that the author has covered Word Processing in a very logical, concise and adequate manner. I believe that the Atari policy is not to bundle 1st Word with the ST now, but there are many purchasers who do have and use this WP. One of the main complaints that I have heard is the, almost, total lack of written guidance in using the program. Admittedly, there is a .DOC Guide, but you need to know how to use the program to get at it! It is a bit of a Catch 22!

GST have recently released 1st Word Plus with good documentation and the differences between the two programs are covered most effectively. Those readers who already own the Plus version and are sharp-eyed, will have probably noticed the clue that suggests this is the WP I use! (It is superb).

The next chapter deals, again very efficiently, with the value of the Data Base as part of an Application System. All of the common principles and concepts are, yet again, covered well and in some depth.

In the same manner, it can also be said of the final chapter on Spread Sheets.

The book does have its faults. They are small, but very niggly. I do not think they are the fault of the Author, but of the Editor and Publishers. They are centred around the readability of the book.

I am a fast reader. I cruise along at about seven hundred words a minute and one thing I do like to see in a book is rhythm. I like to see a balance of short sentences and long ones, short paragraphs and long ones. Punctuation marks and layout of the sentences and paragraphs should reflect rhythm. This book is printed with two line spacing between each paragraph! When you consider that each paragraph in the book is only four or five lines long, there is a hell of a lot of white space. This is of course just my own personal niggle.

This book is an outstanding buy. It is a very good tutorial. Instead of a Summary, the Author introduces each chapter with a Preview. Top marks here.

The important point to appreciate, although you may have a different set of application programs to those covered here, or you are intending to purchase others, it makes no odds. The principles covered and the knowledge you will acquire, apply to those as well.

## Atari ST Graphics and Sound Programming

### Author: Henry Simpson
### Publishers: TAB

This book has only a few pages that are dedicated to the features of the ST, so we are off to a good start. It is an excellent guide to Graphics, using ST Basic. This should not put off any prospective Fast Basic owners from purchasing it. It is not too hard to convert the programs over.

The Graphic coverage is divided into three distinct sections: Building Blocks, Character Graphics and Advanced High Resolution Graphics.

### Building Blocks

Fifty four pages cover basic topics, such as line drawing, colours, fills, palettes, to GEM VDI. By utilising the VDI (Virtual Device Interface) functions, they will considerably speed things up a bit, plus the added advantages of mouse control.

### Character Graphics

There are two very good chapters devoted to this often neglected aspect of programming and I found this section, in particular, to be very interesting.

### Advanced High Resolution Graphics

Of the three sections this was the most interesting. Three chapters cover Graphs, Plotting and Charting, Computer Art and Simple Animation. The only chapter which deals with Sound is the last one in the book. It covers basic sound statements to multiple voicing and WAVE commands. It did not have any MIDI program examples. This book reflects the authors enthusiasm for the ST and it does provide the reader with some very good programming examples, with the stress on good, modular programming techniques.

For any budding Tom Hudson, or David Hockney, you will get a great deal of pleasure and knowledge from this book. There are a great many program examples included. For a modest sum (£16) there are two disks available from the publishers, which contain all the examples.

## 1001 Things to do with your Atari ST

### Authors: Mark Sawusch and Linda Schreiber
### Publishers: TAB

This book has been very popular with Apple, Commodore and IBM PC owners for some time. Now the authors have done a conversion for the ST. There are no fewer than eleven chapters covering such diverse topics as: Personal Applications, Business and Financial Applications, Mathematical and Statistical Calculations, Technical and Scientific Applications, Educational Applications, Hobby Applications, Games and Recreational Applications, Control and Peripheral Applications, AI, Utility Programs and Miscellaneous Applications.

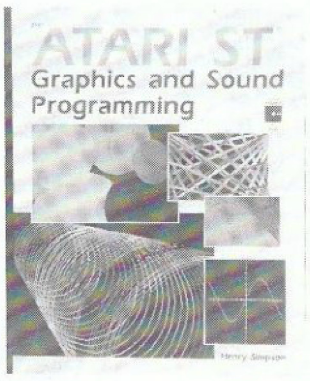
I still have not finished reading this book. I found it fascinating. There is a strong element of the porting across from the other systems, but in this instance, it was not unbearable. Again, there are a great number of program listings included which use ST Basic and I have tried many of them out. None, apart from one, showed any noticeable errors.

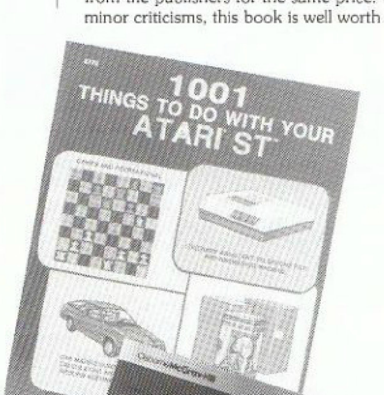
The one that did produce some head scratching was a small program for designing a Yagi Antenna. Antenna designing has been a hobby of mine for a great many years now and I compared this example with one of my own. I honestly think that a chunk of program was omitted from the draft copy!

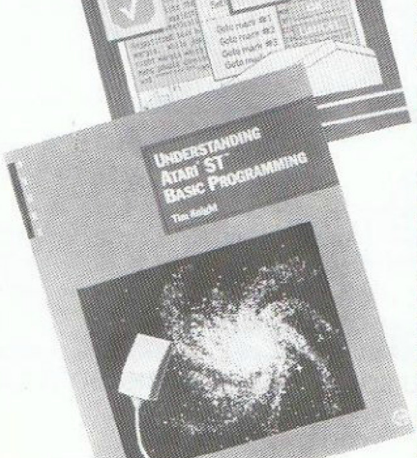
I mention this because the purpose of the book is to stimulate and also gives you a number (1001) of answers to that age old question that computer enthusiasts face from non-users, 'What can you actually DO with a computer?' Read this little book and you will certainly be able to amaze them.

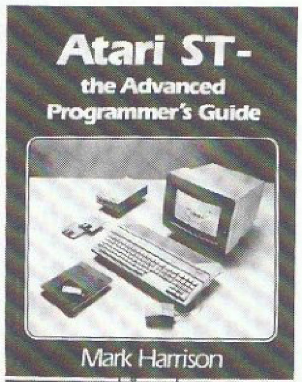
I was disappointed that the authors had not taken any apparent trouble to research the ST a little more carefully and include ideas and examples that are common to the ST and not those examples that are common to the other systems I mentioned earlier. There are no references at all to the built in interfaces: cartridge port, communications or MIDI. A great shame. This would have been an ideal platform with which to explore many other uses.

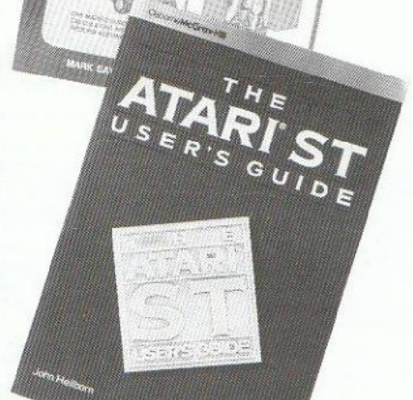
Please do not think that there are program listings for all examples, there are not. But, like the other TAB book, the listings are available on a disk from the publishers for the same price. With these minor criticisms, this book is well worth reading.

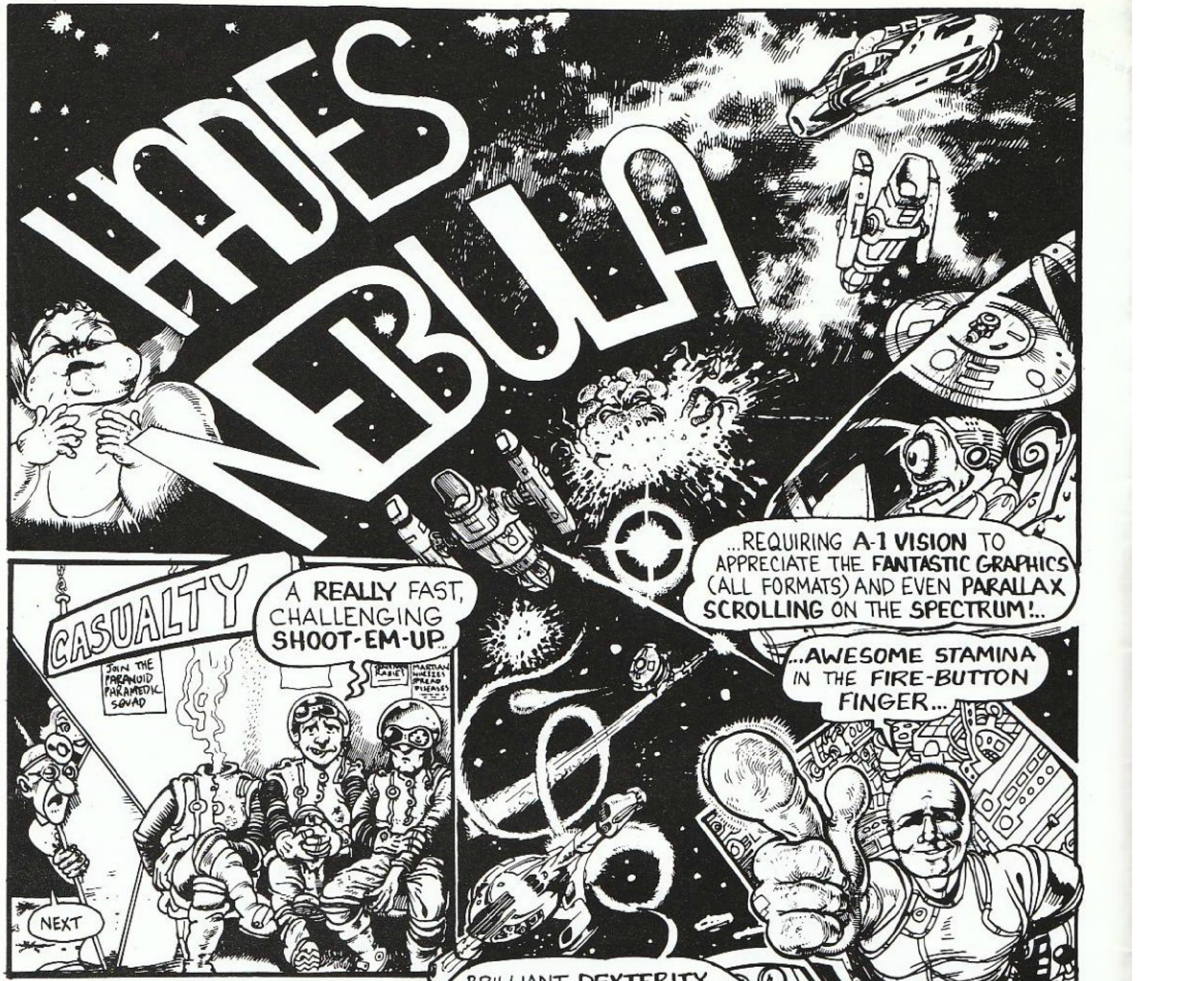











HADES NEBULA
CASUALTY
JOIN THE PARANOID PARAMEDIC SQUAD
MARTIAN MEASLES SPREAD DISEASES
NEXT
A REALLY FAST, CHALLENGING SHOOT-EM-UP...
...REQUIRING A-1 VISION TO APPRECIATE THE FANTASTIC GRAPHICS (ALL FORMATS) AND EVEN PARALLAX SCROLLING ON THE SPECTRUM!...
...AWESOME STAMINA IN THE FIRE-BUTTON FINGER...
...BRILLIANT DEXTERITY TO DODGE ALL THE NASTY AND PESKY BADDIES...

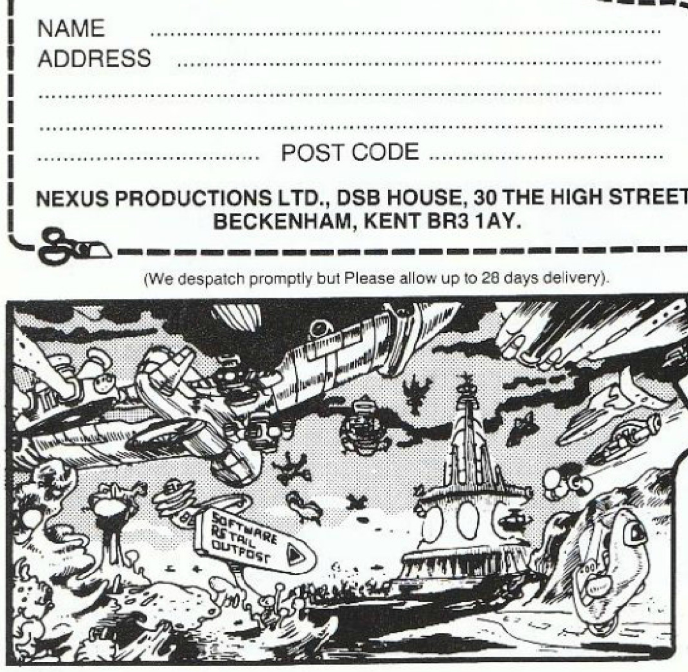
I am absolutely desperate to get my sweaty hands on a copy of **HADES NEBULA**, so here is £19.95 of my hard-earned cash!

NAME ..........................................................................
ADDRESS ....................................................................
..........................................................................................
..........................................................................................
............................... POST CODE ..........................................

**NEXUS PRODUCTIONS LTD., DSB HOUSE, 30 THE HIGH STREET, BECKENHAM, KENT BR3 1AY.**

...ACUTE HEARING TO ENJOY THE COMMODORE/ATARI FX, OR THE FUNNY LITTLE BLIPS ON THE SPECTRUM...
... BUT DON'T TAKE OUR WORD FOR IT... RUSH ALONG TO YOUR SOFTWARE RETAILER AND GET A LOOK-SEE!
ROBIN EVANS

(We despatch promptly but Please allow up to 28 days delivery).



**A PARANOID SOFTWARE GAME BROUGHT TO YOU BY**

**NEXUS**

**OUT NOW ON THE ATARI ST**

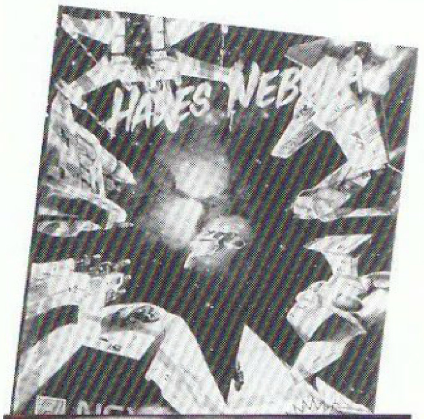
(And flippin' wonderful it is too!)
**ONLY £19.95 inc. P&P!**
(CHQ/PO. ONLY PAYABLE TO NEXUS PRODUCTIONS LTD.)

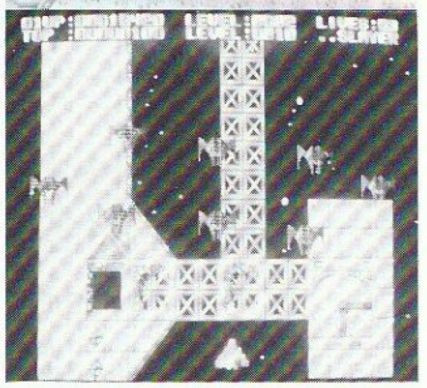
## Hades Nebula

From Nexus
Price £19.95
Reviewed by Stuart Rennie

It's the year 2124 and all the good things in life are at risk of being lost, but the World Energy Committee are sure that if only more energy sources could be found all would be well. Their plan is to send out ships through the Meason Accelerator (a 'go faster' device) to discover these new energy sources. However, unknown to the WEC, the evil Emperor Hades has despatched his entire battle fleet to intercept the Earth ships and make slaves of the crews. And this is just what happened, well almost! Your ship, Colony 7, had a computer malfunction in transit and materialised 2 years before the Hadian fleet arrived. Colony 7 started its mining operation in the Orion Nebula, little suspecting that Hades had already claimed the Nebula for himself and renamed it 'Hades Nebula'. Your peaceful energy gathering operation is suddenly converted into a fight for existence as the Hadian battle fleet comes to get you.

Hades Nebula is a scrolling landscape 'arcade quality' shoot 'em up with terrific graphics and a high level of difficulty. There are 16 (very long) levels and at the end of each level you have to negotiate your way through a meteor storm and fight and destroy one of the Hadian Battle Cruisers. This is not easy as it follows your every move and cannot be shaken off. The first level is fought over the surface of the planet you were mining, after that the scrolling backdrop consists of the shattered hulks of the Earth ships. This is where your luck





improves, hidden caches of weapons and shields are stored on board and you can shoot open these caches and pick up extra help in your fight against the Hadian fighters. You can pick up three lots of extra speed to give you more manoeuvrability, more powerful lasers (that pass through an opponent and may get another one behind), tri-lasers (which fire in three directions at once) and two sets of shields (to guard your left and right sides, and your front and back) which can both be used at once. You can also gain side ships, extra mini-ships which attach to your sides and give you extra fire power. With all these extra add-ons to your basic craft you command a really powerful fighting machine.

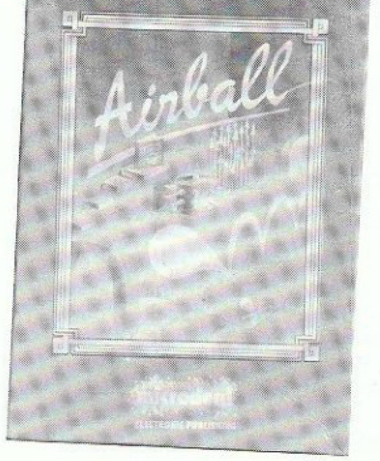
The object of the game is to get to the final level for a showdown with the Emperors Mothership, and to score as many points as possible along the way. You gain an extra life after every 10,000 hard won points. The game can be played with either a musical or sound effects background, on the demo copy I saw the sound effects had not been implemented so I cannot comment on them, but the music was there and it is very well done and catchy, so I can only speculate that the sound effects will be good too! Up to 20 players can take part in Hades Nebula, so you can have all your friends round to play.

Hades Nebula is a 'full screen' scrolling game that's a lot of fun to play. The graphics are top quality, especially the Battle Cruisers, and there is a novel name input score routine. If you're a 'shootist' freak you just can't afford to be without this one!

## Airball

From Microdeal
Price £24.95
Review by Mike Way

Wow!! Need I say more. Well I suppose I'd better. I think this is one of the best games I've seen on the ST this year. The visual look of the game is stunning, the playability is excellent and the game concept is novel to say the least. You control an airball which is pumped up at the start, but it has a slow puncture. Thus you have to move around the three dimensional maze collecting objects, such as tins of beans, crosses, pumpkins, Buddha and crates that assist you to overcome obstacles, before you deflate. Some rooms do have pumps



though, so you can jump on one and be inflated, not too much though airballs have been known to burst if over filled.

The object of the game is to discover the location of a Spellbook which will return you to human form. Unfortunately, the maze is full of prickly hazards and killer flag stones, so you must navigate carefully. You can control the airball by joystick, mouse or on the keyboard. If your airball is burst, it flies through the air in a frenzied fashion, just like a balloon does when it bursts. The further you get from the start room the more complex and dangerous the rooms and passageways become.

Airball is one of those games that is easy to become addicted to, indeed it could well turn out to be a classic!

## Animatic

From Soft Bits
Review by Mike Stringer

This very useful FAST BASIC programmer's tool does, as its title suggests, allow you to create animated objects for inclusion in your programs, or as animation demos.

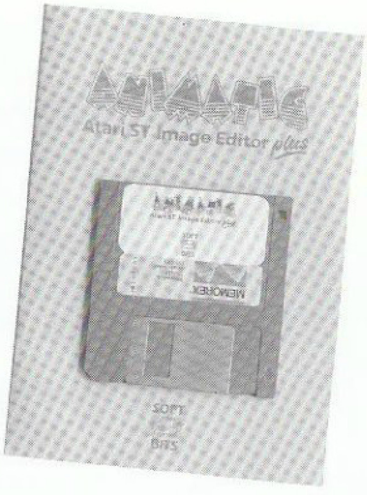
Using ANIMATIC does require some programming knowledge, particularly the use of PUT, GRAB, LOGBASE, PHYSBASE and BLIT. To help you familiarise yourself, there are some very good examples included on the disk which are very well documented but the program itself is very easy to use as well.

There are two types of file available, selectable from the menu, one covers NEOCHROME pictures and the other is ANIMATIC. In addition, included on the disk are six program modules which you can import into your own programs to perform numerous procedures. These include: animation cycling, reserving screens for display and buffers, setting up BLIT commands, PUT and GRAB commands, etc. These obviously take a lot of the tedious, routine programming away, a most considerate and useful feature.

### Neochrome Files

ANIMATIC gives you a number of options to use with NEOCHROME pictures; you may grab a thirty by thirty pixel portion of a picture and create animated effects, for individual use or inclusion in your files and as stand alone images.

The main edit screen has thirty boxes for the sequence and this is easily accomplished with the commands available using the mouse as the pen. The magnified sketch pad makes any editing a very simple task, with the final result being continuously seen.



### Animatic Files

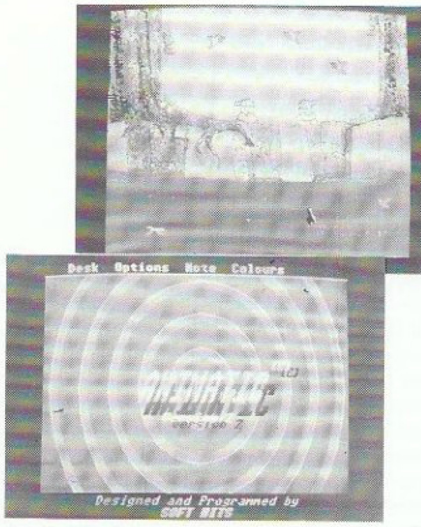
The procedure is similar to the Neochrome editing, the main difference is the colour values are changed, according to the file selected. In other words, the default values for Neochrome and Animatic are not the same.

### Editing

The functions available are: CLEAR, COPY, SQUEEZE, SCROLL, MIRROR and ROTATE. In use, one simply indicates, with the mouse, the frame you wish to edit and then press the button. It is as simple as that.

You can see the results as you progress simply by selecting the ANIMATE box. When this has been done, you indicate which is the first frame and then the last frame with the mouse. Further options are available to select static, animate and speed.

The manual, although small, is quite comprehensive and well written. The Animatic demos are superb. You do not need Fast Basic to use Animatic, unless you intend to incorporate them into your

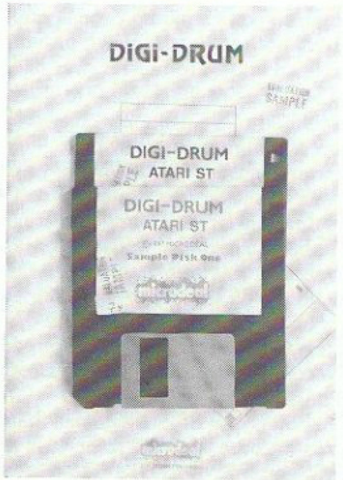


Fast Basic programs, or to experience the quality of the animation. I have tried to photograph one example that I am particularly fond of, I call it 'Watership Down', for obvious reasons. It features no fewer than ten animated objects.

Another very novel use of Animatic's abilities is included on the disk, it is called SMASHOUT. It is a spoof of that classic game Breakout and it is very hard to appreciate that the routines are all derived from this program. It is quite hard to play, but it is very addictive.

If you would like to see the files and the game first-hand, the author, Alan Knopp has kindly given us permission to include them in the ST Library. Remember, you do not need Fast Basic to use and appreciate the program, but you DO need Fast Basic to see and enjoy the Library files.

At the moment, the product is only available by mail order (see advert in this issue). The price is £9.95, I will repeat that, £9.95. It is exceptional value for money. Soft Bits is a new company, I wish them all the best in their venture and look forward to any future releases.



## ST Digidrum

From Microdeal
Review by Glissando

There is now an enormous amount of sampled musical and rhythm sounds appearing in ST games these days. Starglider, Gold Runner, T-t-t-t-yphoon and Star Trek are good examples for starters.

The main reason for this is due to the large amount of RAM that is available on the ST. A standard three and one half inch disk can hold over 400K of formatted data, whilst the double sided disks can hold almost 850K. It is possible to divide the programming into two, one half containing the program and the other the sound sampled accompaniment. It will not be too long before double sided program disks will be available for the 1040s and upgraded 520s.

Sound samples devour RAM as if there were no tomorrow! But, with careful programming techniques, quite a large amount of music can be handled. To assist the programmers, it is common to have short, high frequency samples that can be manipulated in a compacted format.

ST Digidrum is such a programmers tool. Unfortunately, the description on the package may be misleading, they use the expression 'SAMPLING'. To me, a sampling program will allow me to sample sounds. Some sort of interface to a microphone would be incorporated, but this is not so. The program has sampled sounds on the disk which you can manipulate in numerous ways. It would have been more accurate to describe it as a high quality sound sampled Drum Machine.

The basic disk contains a large number of sampled sounds, including a standard set, Toms, Bongo, etc. In addition they included a sample disk containing an additional 41 percussive sounds. This disk is most useful and considering it only costs £15, it should be considered if you purchase the main program.

Now that we have all these sounds, what can be done with them? To start off, no additional hardware is necessary, the sound will emanate from the monitor, or T.V. set. It can also be taken off the Replay cartridge, which is also marketed by Microdeal, for higher sound quality.

This device was not included in the review package, so I cannot make any comment.

On my eight bit set-up, I can get a high sound quality but I have not yet tried any internal modifications to the ST. Consequently, the sound quality is not very good as it stands. I will get round to it one of these days.

The program runs in high resolution black/white and on the colour output, too. I am always pleased when programmers take this into consideration.

From the list of sampled sounds, one can take up to sixteen and place them in the composing column. There are up to 99 song patterns available, although you can edit these to your own 'sound'. The signature is from four to thirty two beats per bar, and unfortunately these are rigidly fixed. It is sometimes very useful to break from conformity to introduce a fresh outlook on rhythm, plus of course, the ability to experiment!

This program is a very good attempt to bring to the percussion enthusiast a very workable and professional package. Because of the sound quality from my system, I will refrain from giving a critical appraisal of the sounds available. I wished that Replay was loaned, in order that I could have done so. There is no doubt that the sounds are very useable, although the cymbal family appeared to be lack lustre. But this may have been due to the ST and not the samples! Unfortunately, there did not appear to be any MIDI linkage or output, that is a shame.

The suggested price is £24.95 and is good value for money. Incidentally, the samples were obtained using Replay!

---

**SLUG**

**The ST London User Group**

We were the first and we are the best! Monthly meetings, quarterly newsletter, the very best public domain software at the cheapest prices, news, reviews, tutorials, gossip, games and hardware.

For a copy of our latest magazine, send a cheque for £1.00 made payable to the 'ST London User Group'

M. Mills (SLUG), 7A Ambleside Drive, Southend-on-Sea, Essex, SS1 2UT.

**Check us out!! You won't be sorry!!**

---

*Announcing*

**ANIMATIC**

**Atari ST Image Editor *plus***

For colour systems with Fast Basic.

AniMatic makes the creation of full colour animated sprite-like objects quick and easy. The objects can be incorporated into your own Fast Basic programs, moved and animated under program control.

Compatible with NEOchrome, AniMatic objects can be combined with a NEOchrome background. Or you can use the NEOGRAB function to cut sections from a NEOchrome screen and animate them.

The AniMatic disk comes with several demonstration programs, including the game 'Smashout'. Together with six program modules, written in Fast Basic, containing set-up and manipulation routines.

The modules can be inserted directly into your own Fast Basic programs.

Please note: AniMatic is specifically for users of Computer Concepts FAST BASIC. It is not directly compatible with other versions of Basic.

*only* **£9·95** including P & P.

Please send cheque or P.O. for £9.95 to:–

SOFT BITS Dept. M
5 LANGLEY STREET
LONDON WC2H 9JA
TEL. 01-836 2533

**SOFT BITS**

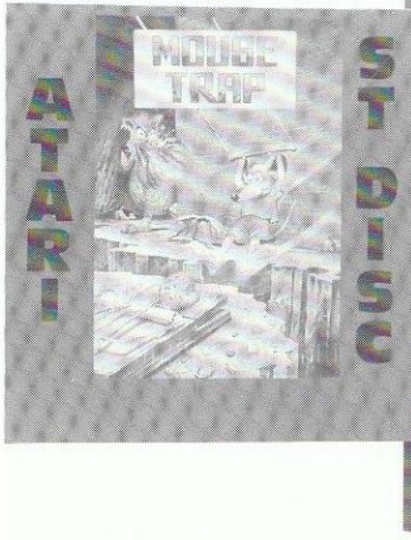
## Mousetrap

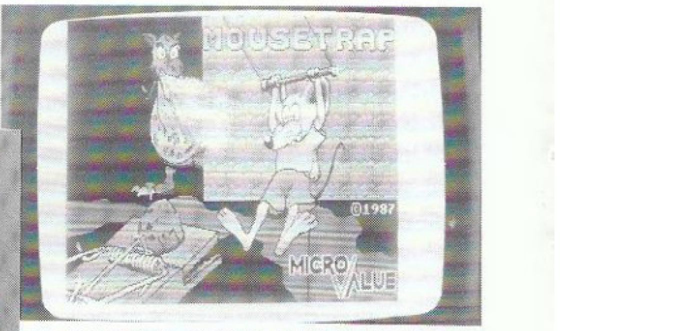
From Microvalue
Price £14.95
Review by Marvey Mills

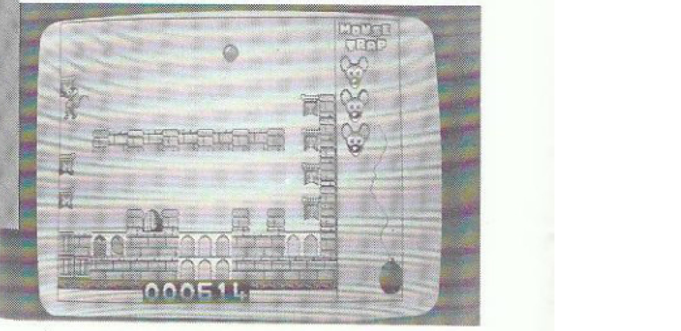
Lets get something straight from the start, I have NEVER liked platform type games. For some reason I've just never got on with them. Now lets get something else straight, this is one of the best games I've ever played on the ST and it changed my outlook on this type of game almost immediately! If the measure of a hit game is the amount of nice touches it contains then this will be a number one smash hit.

I defy you not to smile when you first see the way the nose on the cartoon like mouse bobs up and down as it runs around. The game consists of many screens in which Marvin the mouse (controlled using either joystick or keyboard) has to pick up all the objects from various platforms and ledges whilst avoiding contact with all manner of weird and wonderful baddies before he can proceed to the next screen.

The graphics are a joy to behold, the baddies range from teacup and saucers, through Jack In The Boxes to gruesome goblin (for want of a better word) type creatures. There are also bouncing balls, smelly cheese aromas, crushers and many many more.





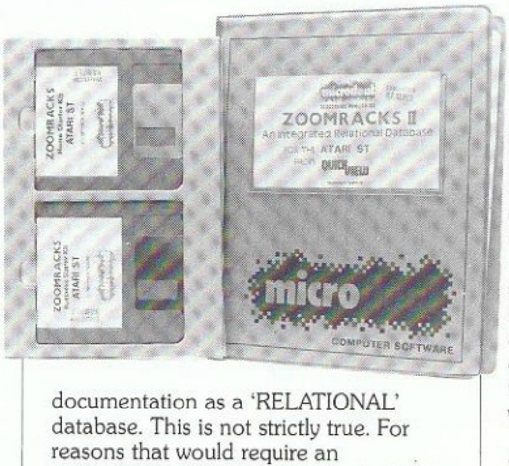


This is also the first game I've ever played that was easy to control from the keyboard, I didn't have a joystick when I first got it so I had to test it that way and believe me, even after I got a joystick I found myself resorting to keyboard for the tricky bits for the better control it yields. My one and only gripe is that like most other games around for the ST it doesn't save the high-score table to disk, but this is so minor I almost didn't mention it!

Microvalue have really lived up to their name with this one, it's out on a budget (for the ST) price but blows the socks off most games of the £20 to £30 mark. I hope this is a trend that continues. Nice one Microvalue.

## Zoomracks II

Produced under licence from Quick View by Microdeal
Price £69.95
Review by Mike Stringer



Zoomracks II (ZII) can be described as a Data Management System type of database. It uses a very novel system, in the computer world, of a device that has been in use in the industrial environment for many years, a card rack. It is similar to the rack found next to the clocking in device in a factory. The size of the rack depends upon the number of staff and holds a card for each employee. The employee sees only the top of the card, looks for his name and removes the card, pops it into the clock, bashes the button and pops it back into another rack for a repeat performance on leaving.

This is the outward appearance of ZII, but do not be deceived, it is much more powerful than a card indexing database such as the Card Filing Systems seen on the older, 8-bit Atari.

The zoom of the title simply indicates that you can easily zoom into the rack to retrieve your card. ZII is described in the documentation as a 'RELATIONAL' database. This is not strictly true. For reasons that would require an explanation of this, are outside the brief of a review. I would describe DBMAN, the DBASE III clone, as a good example of a 'Relational Database'. ZII does not come into that category, it is very close, but not quite! The definition, that is, not the program!

ZII uses a few unique descriptions that you will need to familiarise yourself with. Such as a 'QUICKCARD'. This is the description of an actual card, or record. The name given to a 'field' is a FIELDSCROLL.

One record, or Quickcard, can hold from one to twenty seven Fieldscrolls. Each Fieldscroll can hold up to two hundred and fifty lines of text, each line containing eighty characters. That is a lot of Data!

One rack can hold thousands of Quickcards and up to nine racks can be on-screen at any time. In reality, if an average Quickcard contains one hundred characters, an ST 1040 would manage four thousand records and an ST 520 would hold fifteen hundred.

This is based upon the formula taken from the manual:

Total No. Cards = Bytes available + (No. characters per card X 1.2)

The practical limit is, therefore determined by the RAM. This is the main draw-back to the program. I wish it were disk based, because with the very rapid data transference between the disk and the operating system, the momentary pause would be a small price to pay to give access to the file(s) space available on a disk or hard-drive.

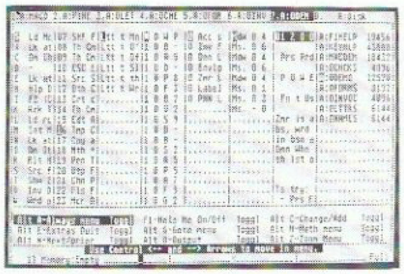
The pattern of Fieldscrolls for a specific purpose is termed a template and Microdeal have already released two disks containing about forty templates. These were sent with the review copy and I must say that they represent good value for money. Each retails for about £10. One disk is for the lady of the house with templates covering the Home. The titles of the rack include: Correspondence, Book library, Daily Calendar, Phone records, Freezer Inventory, Gifts, Pets, Record Library, Videotapes, etc.

The second disk is directed at the small business with templates for: Agendas, Correspondence, Purchase Orders, Customers, Cheques, Expenses, Employee Records, Address Book, etc.

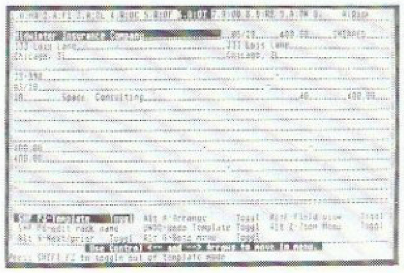
A number of similar disks are to be released soon; Mailing List, Academic, School, Accounting, Collectors, Writers, Manufacturing, Project Management, Sales, Computer and more.

Each item is intended as a starter template. Because they originate from the States, it is not difficult to change them to suit your own needs, because they obviously lean to that market and their requirements.

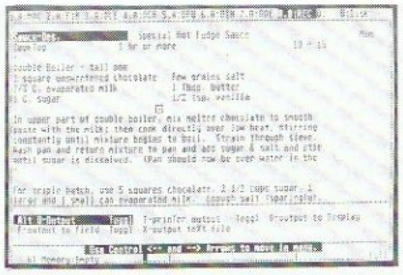
Now that we have established the tools, what can be achieved with this program? Well, the creation of templates is very easy, driven with useful menu commands. Sorting is equally easy and fast. It is possible to import and export



```
[illegible]
Alt E-Extras Pull  Toggl  Alt G-Goto menu   Toggl  Alt M-Math menu   Toggl
Alt N-Next/prior   Toggl  Alt O-Output      Toggl  Alt Z-Zoom Menu   Toggl
            Use Control <-- and --> Arrows to move in menu.
  11 Memory:Empty
```



```
Reminder Insurance Company          [illegible]
Spade  Consulting                                 40.        400.00
[illegible]
Alt F-Template     Toggl  Alt R-Arrange      Toggl  Alt F-Field view  Toggl
Shf F2-edit rack name     UNDO-mode Template Toggl  Alt Z-Zoom Menu   Toggl
Alt N-Next/prior   Toggl  Alt G-Goto menu    Toggl
            Use Control <-- and --> Arrows to move in menu.
Press SHIFT F2 to toggle out of template mode
```



```
Sauce-Des.               Special Hot Fudge Sauce                   Mom
Dom Top                 1 hr or more                       10 - 16

Double Boiler - Small pan
1 square unsweetened chocolate   few grains salt
2/3 C. evaporated milk           1 Tbsp. butter
1 C. sugar                       1/2 tsp. vanilla

In upper part of double boiler, mix melted chocolate to smooth
paste with the milk; then cook directly over low heat, stirring
constantly until mixture begins to boil.  Strain through sieve.
Wash pan and return mixture to pan and add sugar & salt and stir
until sugar is dissolved.  (Pan should now be over water in the

For triple batch, use 5 squares chocolate, 3 1/2 cups sugar, 1
large and 1 small can evaporated milk.  Enough salt (sparingly).
Alt O-Output       Toggl  T-printer output   Toggl  S-output to Display
F-output to field  Toggl  X-output text file
            Use Control <-- and --> Arrows to move in menu.
  11 Memory:Empty
```

files via ASCII codes. It is also possible, and I believe it to be unique, one can even insert DEGAS images into a quickcard! The ability to output data, via Output Forms, is very easy to set up, use and edit. In fact the editing of all aspects is very easy to perform. It is also possible to set up small MACROs and it is very encouraging to see this feature included, it extends the possibilities and power, even further.

The manual provided is a sturdy two ring binder. This is very handy for inserting any updates that come along periodically, as well as your own notes and comments. It covers most aspects quite well. Obviously, it cannot cover them all, because the total number of possibilities, created by you, are unknown.

I have now been using the program for over two months and I am very pleased with it. Please do not be put off with the adverts which suggest the price is over £150, I think that someone has got their lines crossed!

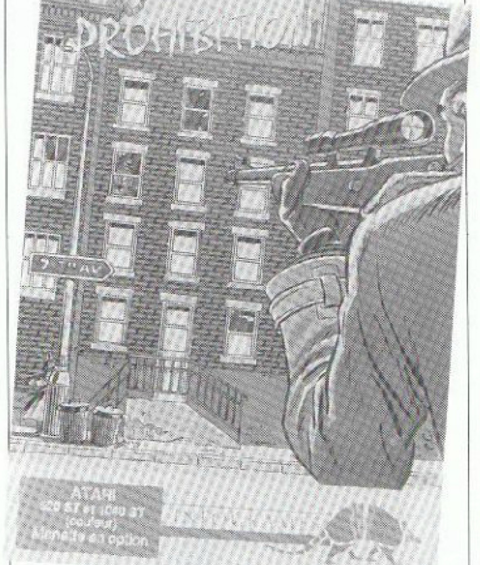
The disk also has a guide/tutor included, it is very useful for finding your way round the system for the first time. I believe that ZII is going to be of more value than most of the more well known and expensive databases that are being marketed at the moment. It is the most useful and easy-to-use that I have yet come across.

## Prohibition

From Infogrames
Price £19.95
Reviewed by Marvey Mills

I found Prohibition, a game from Infogrames, to have the same 'strange' feel to it that all of the French games I have seen so far have had. Although the idea is fairly original and the game well written I feel that in this, just like others, the French still have a way to go before they can produce a well-rounded, addictive and playable game.

The scene is set in downtown New York in the prohibition era when the mob ruled the city and the police were hard pressed to enforce the law. You, of course, play a sharp-shooting cop whose job it is to kill as many gangsters as you can. The gangsters are understandably upset with this and want you dead before you can say 'Go ahead, make my day'. The mouse controls a set of crosshairs which you use to aim your shots at the background which scrolls smoothly up and down, left and right. All of a sudden you catch sight of a villain pointing his gun at you. You have five seconds to take him out before he starts firing and unless



you kill him within this time limit, friend, you're a dead man.

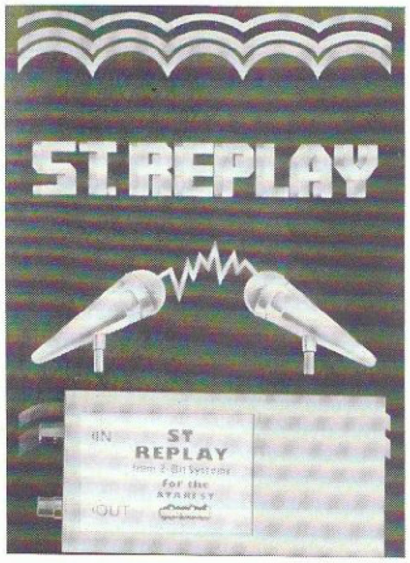
You don't get a chance to rest, because as soon as he's been dispensed with one of his friends pops up to avenge his death. This would all be very well if the playfield were smaller but all too often I found the gunman would appear a long way off the screen and I'd be frantically whizzing the mouse around trying to find him. There are little arrows which appear telling you in which direction the killer is but they were too small and grey coloured and merged into the background. Meanwhile your five second reprieve is ticking away.

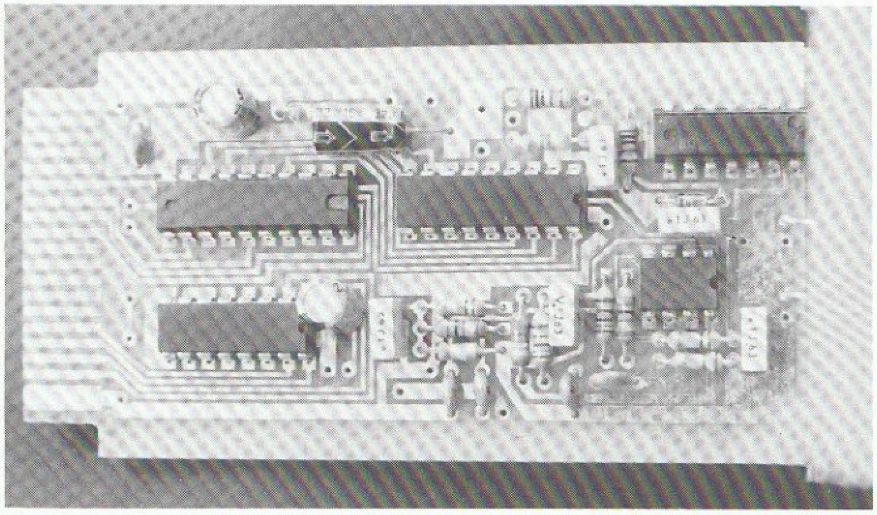
You are supposed to accumulate dollars for every hood killed but there was no running score on the playscreen nor was I told at the end so I had no way of knowing how well (or badly) I was doing which took some of the essential competitiveness out of it. All in all, not a bad game. A great idea but I feel that it could have been done better. It has good graphics and the now obligatory digitised sound (including some very good 'death-screams'!) but lacked the easy accessibility of something like Gold Runner.

## ST Replay

From 2-Bit Systems, marketed by Microdeal
Price £129.95
Reviewed by Mike Stringer

This cartridge based unit comes in a small cardboard box with a disk of programs and utilities and a twenty four page booklet which passes as the manual for this sound sampling and replaying system.





The cartridge plugs into the cartridge port of the ST via the standard edge connector which contains the circuit board and two phono sockets to the outside world.

The board is well made, although I would have thought at this price, the edge connectors would have been gold plated for better electrical contact and also to prevent oxidation of the contacts.

The circuit board comprises an Op-Amp and a very basic 8-bit A to D converter. On the sample reviewed there were a few modifications around the filter network. This part of the circuit was more pronounced on the input than the output. On the output was the circuitry involved with D to A. Altogether six ICs and a number of capacitors and resistors. Nothing very fancy at all, basic and uncomplicated. I have seen sound sampling devices for 8-bit machines that were of a better quality, both in design and construction but, importantly, less than half of the asking price.

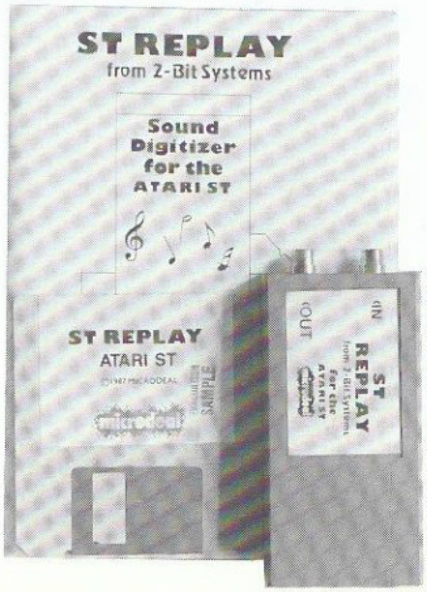
The specifications are not particularly impressive, either. For example, the signal to noise ratio is quoted as 48dB, not exactly Hi-Fi! If the sampler were sixteen bit and filtration were improved, this figure would increase and it would have been a superior product.

The ranges for sampling are very useful: 5kHz, 7.5kHz, 10kHz, 15kHz, 20kHz and 31kHz. With a standard 520ST, this equates to eighty seconds at 5kHz and thirteen seconds at 31kHz. With 1 megabyte of RAM available, these figures are more than doubled.

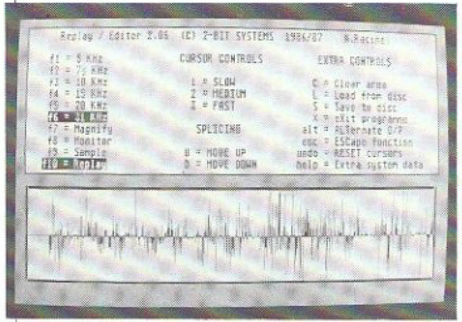
To evaluate the sampling quality, I used two microphones, a Shure 444, which is a standard communication microphone of very good quality for voice, a condenser microphone for voice and music, synthesiser and audio signal generator.

Voice quality was very good from 5kHz to 10kHz, as was to be expected and music/voice was very good from 10kHz to 31kHz. The deficiencies are no doubt due to the very poor signal to noise ratio. The output HAD to be put through a good quality graphic analyser. There was quite a lot of background noise without it.

The disk contains an Editor which has been deliberately kept very basic and of limited functions to preserve as much RAM for the end-user as possible. I did not find it very useful, but most restrictive, especially when I tried to splice samples together. I would have much preferred a more sophisticated Editor at the expense of RAM availability.



A number of sample routines for various Basics were included on the disk. These were well documented and I had no trouble in understanding the modus



```
Replay / Editor 2.06  (C) 2-BIT SYSTEMS  1986/87   N.Racine

f1 = 5 KHz          CURSOR CONTROLS        EXTRA CONTROLS
f2 = 7½ KHz
f3 = 10 KHz           1 = SLOW            C = Clear area
f4 = 15 KHz           2 = MEDIUM          L = Load from disc
f5 = 20 KHz           3 = FAST            S = Save to disc
f6 = 31 KHz                               X = eXit programme
f7 = Magnify           SPLICING         alt = ALTernate O/P
f8 = Monitor                            esc = ESCape function
f9 = Sample          U = MOVE UP       undo = RESET cursors
f10 = Replay         D = MOVE DOWN     help = Extra system data
```

operandi. Unfortunately, the same cannot be said for programmers who are using C, Modula II, Pascal or any similar language. For this price I would have expected to see these languages treated in the same manner as Basic.

Also I would have thought that the MIDI sampling format would have been included, but there does not appear to be MIDI implementation at all.

I must admit I was disappointed with ST Replay. Microdeals products are usually of the highest quality, but they seem to have gone a bit astray here. If this product was being sold for about £60, I would say it was good value for money, but for £130, I feel it is over priced. Basically it's a good product with some useful features, but for the money asked you would expect ST Replay to cover a lot more ground.

ST ST

# StockSoft

520STFM .................. lowest prices in the UK.
1040STFM .................. for latest prices see our price list

Public Domain Software: Games, Utilities, Art, Demo's, and Nudes.
Wide variety, all at only £5 per disk, post free.

## ST HAPPY

Happy computers of the USA who have been making superb disk backup utilities for the Atari 8 bit range since 1979, have now announced an ST backup cartridge which will back up 3.5" standard ST disks, also 3", 3.5", 5.25", and 8" disks form any other computer, provided the compatible disk drive is connected.

For further information apply below.

## BLANK DISKS

High quality GoldStar branded disks, tested and certified 100% error free. Single sided, box of 10, only £15, post free.
Double sided, box of 10, only £20, post free.

For further information on any of the above please send large S.A.E. to:-

STOCKSOFT

15 WOODBROOKE ROAD, BIRMINGHAM, B30 1UE.

MAIL ORDER ONLY.

ATARI ONLY.

## Barbarian

From Psygnosis
Price £24.95
Review by Tom Davis

It is difficult to catagorise Barbarian, but here goes. It is a platform type, fantasy role playing game with brilliant graphics, nice animation and absorbing game play (even if a little on the slow side). You take the role of Hegor who must enter the underworld of Durgan to find the lair of Necron the ancient enemy of Hegor's father Thoron. Once there you must destroy the crystal which gives Necron his powers. Once this is achieved, volcanic action starts to destroy the underworld and you must escape to the surface before you are crushed.

Along the bottom of the screen is a strip of controls, these are to give you movement of Hegor and other info such as lives left, time taken and weapons possesed. You can use the mouse, keyboard or a joystick to make the movements (Psygnosis recommend you use a mouse). In addition to the usual movements Hegor can jump (somersault if he is running), attack and defend (a sword stroke followed by a backward somersault) and pick up or put down objects.

As you get progressively deeper the hazards get more difficult, more monsters and guards come at you, and the armour encased knights are real tough. You can pick up arrows along the way just in case you find a bow to use. Using the mouse to control Hegor is a little clumsy at first but it is surprising how soon you get adept with it. Also surprisingly there is a lack of sound effects, apart from a few grunts from Hegor and growls from monsters.

Barbarian is presented in a stylish box that contains two disks (disk A only seems to have intro pictures on it). The intro pics show the Psygnosis logo, then the Barbarian logo and then a passable representation of the Roger Dean

landscape, which is also presented to you in the form of a poster. Also in the box is a story booklet which sets the scene and is well worth reading. Disk B is the program disk and is left in your drive whilst you play.

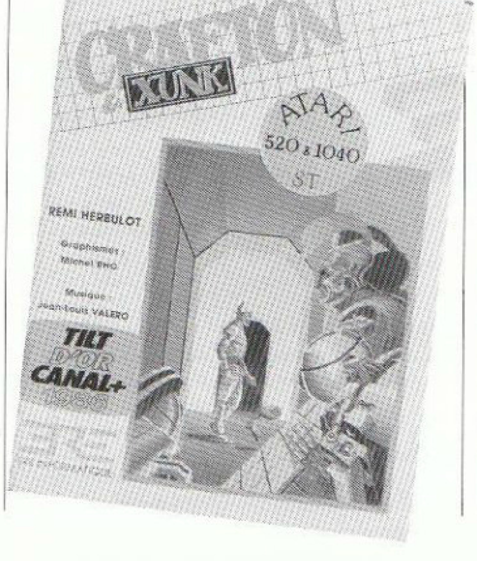
The only gripe I have is that when I first tried to load Barbarian I found that it would not load. So I tried loading it without the mouse plugged in and it worked, so I then had to fumble around plugging the mouse back in. Then after I was killed in the game it reloads and told me to press a key to continue, this worked sometimes but not always. Very frustrating!

I think Barbarian will be a top seller, it has all the ingredients to make it so. Excellent graphics, entertaining, a difficult goal to aim for, good game-play and a poster by Roger Dean, how can it fail!!

## Crafton & Xunk

From Infogrames
Price £19.95
Reviewed by Bill Dyer

Crafton and his little friend Xunk have to discover the 8 figure code that is the means to penetrate Zarxas room in the central computer for Galactic Control. They do this by finding scientists and interrogating them, each scientist has one part of the code. The 3D warren of corridors, laboratories and offices is an intricate maze full of objects and furniture, all in very good detail. Many objects can be picked up and used to destroy, reject or divert the guardians of the centre. Tables and chairs can be moved about to block the path of the patrolling robot guard dogs. When attacked energy is drained from Craftons



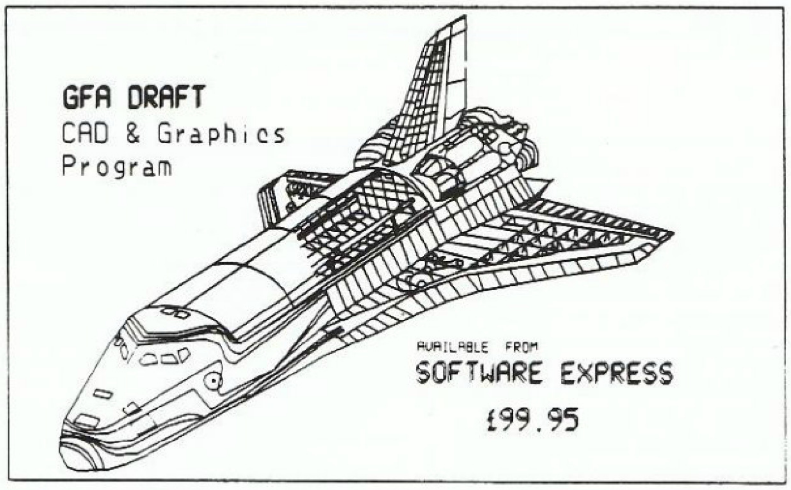
body until he turns green and expires. Other hazards include falling partitions, slippery floors and green haired Punks.

This program was written by Remi Herbulot, who is well known in France, and published by Ere Infomatique. Indeed the review copy packaging and instructions were all in french, making it a difficult product to review, because at first I was not aware of what had to be done in the game, let's hope the English version will be repackaged. There were a few instructions in English, but not really enough. Also frustrating was the fact that a small comic book of Crafton & Xunk's adventures was included, but I couldn't read it!!

Despite these minor problems, Crafton & Xunk is a very original game with lovely detailed graphics, lots of tricky rooms and fairly challenging game play.

**SOFTWARE EXPRESS**

514-516 Alum Rock Road, Alum Rock, Birmingham Tel.: 021-328 3585



Precision Software

£99.95

Superbase™ PERSONAL

RELATIONAL DATABASE SYSTEM

**VOGLER**

*Software*

An exciting new range of professional software for the discerning user.

JackFont
JackSpell
JackMake
HighJack
Shortcut

Please ask for an information sheet

Quality First!

OK

MERKANT

The BEST Stock Control and Merchandising program available for the ST

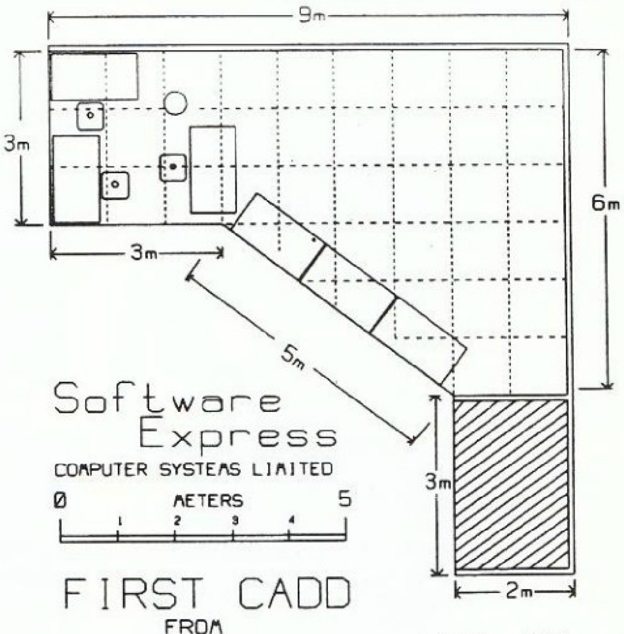
Ask for details

*Label Maker*

XLEnt Software

Undoubtedly the most useful Desk Accessory you will ever buy!

£19.95 Ask for details



FIRST CADD
FROM
GENERIC SOFTWARE

49.95

**Dealer and Overseas Enquiries Welcome**

**514-516 Alum Rock Road, Alum Rock, Birmingham**

**HOTLINE 021-328 3585**

# ROUTINES MATTER

By Marvey Mills

## It's Just Routine

Why use Assembly Language? Well the most likely reason for choosing to use this very low-level language at all is that it enables the software author to achieve the fastest possible execution speed. Since you are dealing with commands to the actual processor, what you write is exactly what is performed at run-time. The author has absolute control over what happens during the execution of his program. Compare this to BASIC for example. BASIC is an interpreted language. Each statement is examined by the BASIC interpreter, which then performs all the necessary functions by using some of its vast library of built-in machine code routines. The effect of this is to greatly simplify program writing but at the expense of running speeds; most BASICs are notoriously slow. For some applications the very fact that it is so high-level can be a disadvantage. With an interpreted language you are stuck with the functions that the language recognises with generally no chance of adding your own. For example, you may wish to format a disk within your BASIC program. If you are lucky there may be a command in that version of BASIC that allows you to do just that, but what if you only wanted to format one track on the disk, perhaps for some form of copy protection? The function for doing this is in GEM for the programmer to use, but if your version of BASIC doesn't support it then tough luck! Of course many versions of BASIC now include an assembler which is a reasonably good compromise. It will probably not be as good as a dedicated assembler program, but you still have the BASIC part to do all the tricky stuff with.

Using assembler for the first time on a new machine can be a very frustrating experience if you are not sure how to get started with it. You may, like myself, have learnt all about assembly language programming using one of the old 8-bit Atari machines and can't wait to start getting to grips with all that power that you know lurks within your ST, just waiting to be unleashed. One of the first programs I bought for my ST was GST's Macro Assembler and very good it is too, but I didn't actually get round to using it for a good 2-3 months simply because I just didn't know what one has to do to write a program that works.

In this article I will attempt to provide you with the necessary information and code that will build a basic framework within which you can write your own programs. There are certain basic functions that are fundamental to just about every program ever written, without them the program would be next to useless. I keep the basic 'blank' framework on a disk with all the routines included and when I start a new project I load it up and get to work without having to re-invent the wheel every time I want to clear the screen or get a user input for example. When I have finished the program, I delete those routines not used thus saving program space and yielding a much less complicated source listing. Note that the example I have chosen is a very simple one that does not use GEM. Programming in assembly using GEM and windows, etc., is VERY complicated and not to be undertaken lightly. For simple utilities a TOS program is quite adequate for most needs, and while not being as pretty as GEM, is nonetheless much more simple to use. What we will do is write a short and simple program, using as many basic functions as possible, that will turn the disk verify on or off. The advantage of turning the verify off is that the operation of writing information to a disk is speeded up because the computer does not have to re-read data just written to check it was written correctly. Of course this also means that data can be written to disk incorrectly, so use it with caution, and not when saving important work!

## Memory Management

Before starting to work on the actual coding for this program there is one very important area we must look at and that is something I call the 'Memory Management Header'. It is a short piece of programming that, whilst not essential, provides the correct way to start up any program be it a TOS application or a GEM application. See Listing 1.

It performs two functions. The first is that it tells the operating system where the user stack is in memory. The 68000 processor has two stacks, the Supervisor stack and the User stack. The operating system uses the supervisor stack for its internal functions leaving the user stack to us to use as we wish. The first two lines of the header firstly save the current stack pointer for later and then replaces it with a pointer to an area that I have called our_stack. This must actually be defined within the program itself, in other words the programmer must define an area in the program and call it 'our_stack' (or whatever name you have chosen). Secondly the header adds up all the memory space used by the program and tells GEM not to use it for anything else. GEM keeps a list of all the areas of memory that it cannot use and uses the rest for its own functions. Without reserving our bit we run the risk of GEM overwriting our program with potentially disastrous results. All this sounds

```
;** The Memory Management Header **

move.l    a7,a5           ;get old stack pointer
move.l    #our_stack,a7   ;and set stack pointer to our area.

move.l    4(a5),a5        ;add up all the various parts
move.l    $c(a5),d0       ;of memory that our program
add.l     $14(a5),d0      ;uses and tell GEM to keep them
add.l     $1c(a5),d0      ;safe for our program
add.l     #$100,d0        ;otherwise GEM could overwrite the
move.l    d0,-(sp)        ;program area during its
move.l    a5,-(sp)        ;operation.
move.w    d0,-(sp)
move.w    #$4a,-(sp)      ;uses SETBLOCK
trap      #1              ;call GEMDOS to execute operation
add.l     #12,sp          ;correct stack pointer on return
```

Listing 1.

complicated but it isn't really. All the programmer has to do is use the header at the start of every program and then forget about it. It is another one of those routines that I keep on disk in my blank framework.

## Printing to the Screen

We'll start with the 'print line' routine. As with all operating systems there is a ready-made facility for printing text to the screen built in. It is a trap #1 call (i.e a call to GEMDOS) and as with all traps, it expects to receive some parameters on the stack telling it what to do and with what. Traps are a rather neat way of calling operating system functions. Basically when a trap command is executed it causes the processor to go to a look-up table of addresses of particular parts of the operating system, find the one associated with that trap number and then choose a particular function within that section to execute. At this level of programming we will be using only trap #1 calls but traps #13 and #14 are readily available to the programmer yielding much more powerful functions and routines (the track format routine I talked of earlier is a trap #13 call). Advanced programming using the GEM windowing system uses trap #2 calls. To print to the screen we need to load two parameters on the stack, the address to start printing at and the function number of the print line function. See Listing 2.

Easy isn't it. But wait, what's this about correcting the stack pointer? Well yes, when the program returns from any trap call the stack pointer is set as if you had never jumped off and executed a complicated routine. In other words the trap call DOES NOT pull the parameters off the stack, it just reads them directly out of the stack area. I still don't know why this is, it's been puzzling me for ages! Anyhow, what you have to do to avoid the processor running out of stack space is to add up how many BYTES (remember one word equals two bytes, one longword equals four bytes) you have pushed onto the stack and add that value onto the stack pointer. It sounds a bit tricky but you'll soon get used to it. The print line function is generally used very many times in a program which uses up space repeating the code (short though it is) so I make it into a subroutine. See Listing 3.

Notice that the routine gets the address of the string from the A0 register. All you have to do now is load the text address into the register and then jump to the subroutine. See Listing 4.

You do not have to do any stack correction on calls to subroutines within your program. Nor do interrupts need stack correction, only trap calls. This routine will start printing from the address you have given it and will not stop until it sees the delimiter character. Otherwise known as End-Of-Line or EOL, its ASCII value is zero. As soon as the routine sees a byte with the value of zero it will stop printing to the screen and return from the trap call. It is therefore quite possible to have a text line with embedded linefeed and carriage return characters so that all the print formatting is done by printing just one line of text. Carriage return is ASCII value $0D and linefeed is value $0A. Consider how this example would look on the screen when printed using our print line routine.

'Hello',$0D,$0A,
'World',$0D,$0A,$0A,0

It would print 'Hello' on one line, 'World' on the next underneath it and then space two lines down the screen ready to print the next line.

## Clearing the Screen

Clearing the screen is a very simple matter now that we have a useable print line routine. When you are not using the VDI to perform complex screen manipulations of windows, etc., the screen conforms to what is known as the VT-52 standard. This means that if you attempt to print a value to the screen preceded by the ESCape value (decimal 27) the screen handler will treat the value as a command and will not actually print it as a value. It just so happens that the ASCII value for the capital letter 'E' is the VT-52 command for clearing the screen, so if you used our p_line routine to print the following string:

27, 'E',0

Now the screen handler would see the ESC value, recognise that the next value is a screen command for clearing the screen, execute this command and then look for something else to print. In this case we've put the End-Of-Line value next, so the routine would just return to the program. However, just like the carriage return and linefeed characters, you can put these codes in with your text and the whole lot will be acted upon.

27, 'E', 'Hello',$0D,$0A,
'World',$0D,$0A,$0A,0

If this was printed it would clear the screen before printing our 'Hello World' example. So you can do it either way if you like. I prefer to keep it separate from the text as I think the program reads better that way. Again I've made it into a subroutine, but this time you don't need to load any addresses into registers before calling it. See Listing 5.

Wait! Another little unknown has crept in there. What is that 'even' command? Well, it's not actually a command to the 68000 processor, it's what is known as a pseudo-opcode or pseudo-op. The problem is that ALL 68000 commands must begin on a word-boundary, but each character of

```
move.l   #text,-(sp)     ;push address of string on stack
move.w   #$09,-(sp)      ;push the function number for print line
trap     #1              ;call GEMDOS
add.l    #6,sp           ;correct stack pointer
```

Listing 2.

```
p_line
        move.l   a0,-(sp)       ;get address of text onto stack
        move.w   #$09,-(sp)     ;push the function number for print line
        trap     #1             ;call GEMDOS
        add.l    #6,sp          ;correct stack
        rts                     ;return from subroutine
```

Listing 3.

```
move.l   #text,a0      ;get address of text into a0
jsr      p_line        ;call our print line routine
```

Listing 4.

```
clr_scrn                              ;clear screen subroutine
              move.l   #scrn_code,a0  ;get address of ESC-'E' line
              jsr      p_line         ;call our print line subroutine
              rts                     ;return to program
scrn_code     dc.b     27,'E',0
              even
```

Listing 5.

text is only half a word long (one byte). Therefore, if by chance you have an odd number of characters in your string then the next real command in the program will begin half way through a word and not at the start of one. What 'even' does is to instruct the assembler program to check if this is indeed the case and put in an extra byte at the end of the string to force the character count to an even number. It will never appear on the screen since the last character of any string you put in to be printed will always be the EOL character, so GEM will never get as far as the dummy character when printing.

## Getting a Keypress

Another very useful subroutine to include is one to get a character from the keyboard and return it to the program for examination. Again, there is a ready-made one in GEMDOS that does exactly this. In fact there are several, each doing a slightly different thing. The one we are going to use is called 'CONIN'. The only parameter that needs to be pushed onto the stack is the function number itself (ASCII 1). This trap call differs from the one we used above in that it actually returns a parameter as well as receiving one. All trap calls that need to communicate data to the program do so primarily by means of the D0 register. The value returned can be a longword in some cases or a word value as in this case but NEVER a byte value. You may well ask why a byte value is not used in this case since no characters have a value that need more than a byte to hold them? The answer is that the call does not just return the character value but also the scancode value of the character. The scancode determines the physical key pressed not the character assigned to that key. For example, both the RETURN and ENTER key on the ST have the character value $0D (carriage return) but in order to tell the difference the RETURN key has a scancode of $1C whilst the ENTER key has a scancode of $72. Of the word returned from the trap call, the low byte is the character code and the high byte is the scancode. Unless you really need to you can ignore the scancode, for simple programs like this one it just isn't needed. This function does not behave like BASIC's INKEY$ command. Once called it waits until a key is actually pressed before returning to the program with the information. This feature makes it very useful indeed for menu selections and the like, which is in fact what we will use it for in the sample program. See Listing 6.

## Quitting a Program

In order to quit from a program we need to tell the operating system first so that it can release any memory we have grabbed. This is all handled by a simple trap #1 call, called 'TERM'. It is almost unique amongst all the trap calls in that you do not have to perform a stack correction (there is only one other, called 'KEEP PROCESS' which also quits a program but does not release the grabbed memory back to GEM) for the simple reason that it never returns to the program! Again, it only needs the function number on the stack to work correctly. See Listing 7.

## About the Program

Armed with just these few simple routines we can build up a useable program, see Listing 8. It only leaves the matter of actually writing the bit to turn on or off the disk verify, which although a simple thing to do, requires a slightly more complicated piece of programming. Location $444 contains the verify flag, if it is non-zero then verifies are done and if it is zero then they are not. The problem is that this location lies within an area of memory that GEM would not normally let us fiddle around with. It protects itself from 'unauthorised' tampering by giving an error. Normally, trying to change this location would result in eight nasty little bombs on the screen! However, GEM itself has to amend these locations, so how does it do it? Well the 68000 has two distinct modes of operation, User mode and Supervisor mode and GEM and the rest of the operating system run in supervisor mode whilst our programs run in user mode. Supervisor mode gives us unlimited powers to change protected memory at will, so what we have to do is make our program, or at least some of it, run in supervisor mode. Simple, all it takes is another trap #1 call, called 'SUPER' and we are running in supervisor mode until we make another trap #1 call to turn it back into user mode. I won't go into further detail at the moment since this is really outside the scope of this article but if you look at Listing 8 you'll see how it is done.

So that's it, a header, four simple subroutines and a bit of trickery and we have a simple, useful utility and hopefully some insight into assembly language programming on the ST.

```
get_key
          move.w    #$01,-(sp)      ;push function number onto the stack
          trap      #1              ;call GEMDOS
          add.l     #2,sp           ;correct stack
          rts                       ;return to program with character
                                    ;code and scancode in register D0
```

Listing 6.

```
quit
          move.w    #0,-(sp)        ;push function number on stack
          trap      #1              ;call GEMDOS
                                    ;and thats it- does not return
```

Listing 7.

Listing 8.

```
;***********************************************************
;* Disk Verify On/Off toggle program
;* Demonstrating some simple subroutines
;* By Marvey Mills for Monitor Magazine
;***********************************************************

lf      equ     $0a                 ;set up some constants
cr      equ     $0d                 ;for the print formatting
eol     equ     $0                  ;characters - makes program
esc     equ     27                  ;more readable

;** Memory Management Header **

        move.l  a7,a5               ;get old stack pointer
        move.l  #our_stack,a7       ;and set stack pointer to our area
        move.l  4(a5),a5        ;add up all the various parts
        move.l  $c(a5),d0       ;of memory that our program
        add.l   $14(a5),d0      ;uses and tell GEM to keep them
        add.l   $1c(a5),d0      ;safe for our program
        add.l   #$100,d0        ;otherwise GEM could overwrite the
        move.l  d0,-(sp)        ;program during its
        move.l  a5,-(sp)        ;operation
        move.w  #0,-(sp)
        move.w  #$4a,-(sp)
        trap    #1
        add.l   #12,sp

;******************************
;       PROGRAM CODE HERE     *
;******************************
start
        jsr     clr_scrn        ;first clear the screen

        move.l  #title,a0       ;get address of title string
        jsr     p_line          ;and call our print line routine

ask
        move.l  #prompt,a0      ;get address of the prompt
        jsr     p_line          ;and call our print line routine
        jsr     get_key         ;get a key press

        cmp.b   #'Y',d0         ;check if low byte equals ASCII 'Y' code
        beq.s   y_ver           ;if so branch to turn verify on
        cmp.b   #'N',d0         ;check if low byte equals ASCII 'N' code
        beq.s   n_ver           ;if so branch to turn verify off
        cmp.b   #'Q',d0         ;check if low byte equals ASCII 'Q' code
        beq.s   quit            ;if so branch to quit routine
        bra.s   ask             ;else ask again

y_ver                           ;sets disk verify on
        clr.l   -(sp)           ;go into supervisor mode
        move.w  #$20,-(sp)
        trap    #1
        add.l   #6,sp
        move.l  d0,save_ssp

        move.w  #$ff,$444       ;set verify on
        jmp     end             ;and go to end routine

n_ver                           ;sets disk verify off
        clr.l   -(sp)           ;go into supervisor mode
        move.w  #$20,-(sp)
        trap    #1
        add.l   #6,sp
        move.l  d0,save_ssp

        move.w  #0,$444         ;set verify off

end
        move.l  save_ssp,-(sp)  ;go into user mode
        move.w  #$20,-(sp)
        trap    #1
        add.l   #6,sp
        jmp     quit            ;then jump to quit routine


;*********************************************************************
;* START OF SUBROUTINE LIBRARY                                       *
;*********************************************************************
clr_scrn
        move.l  #scrn_code,a0   ;get address of ESC code for clear
        jsr     p_line          ;call our print line routine
        rts                     ;return to program
scrn_code
        dc.b    esc,'E',eol,eol      ;escape code for clear screen
        even

quit
        move.w  #0,-(sp)        ;push function number on stack
        trap    #1              ;call GEMDOS
                                ;and thats it- does not return

p_line
        move.l  a0,-(sp)        ;get address to start print
        move.w  #$09,-(sp)      ;push the function number on stack
        trap    #1              ;call GEMDOS
        add.l   #6,sp           ;correct stack
        rts                     ;return to program

get_key
        move.w  #$01,-(sp)      ;push function number on stack
        trap    #1              ;call GEMDOS
        add.l   #2,sp           ;correct stack

        and.b   #%11011111,d0   ;include this line if you want to force
                                ;the character into uppercase

        rts                     ;return to program with character
                                ;code and scancode in register d0

;************************
;* Text storage
;************************

title   dc.b    '*****************************',cr,lf
        dc.b    '* Disk Verify Toggle Program *',cr,lf
        dc.b    '*****************************',cr,lf,lf,lf,eol

prompt  dc.b    'Type <Y> to set verify on.',cr,lf
        dc.b    'Type <N> to set verify off.',cr,lf
        dc.b    'Type <Q> to quit with no change.',cr,lf
        dc.b    'Enter Choice>',eol

;************************
;* DATA STORAGE
;************************
                even
                ds.l    64      ;user stack area
our_stack       ds.l    1
save_ssp        ds.l    1       ;save supervisor stack pointer
```

# ST LIBRARY

**Librarian: Mike Stringer**

## Introduction

Allow me to tell you how the ST Library is going to be structured. Listed here are the disks currently available. I am expecting about thirty disks from North America, plus another dozen or so from some members over here. Still, we will be starting with a fair foundation upon which to build a very useful and valuable service to you, our readers.

The disks that I will be sending out are DS/DD but will be formatted for single sided use. Where the program requires 1 Meg formatting, these disks will be clearly marked and no additional fee will be requested. In other words, the fee will be the same, irrespective of the size of the program(s).

In some instances the files may be compressed. The necessary Archiving program will always be included on the disk, including the necessary info to allow you to convert them back to normal. In this way I will be able to put up to the equivalent of 500K of files on one, half-meg, disk.

In addition to the files, I will also include, if space permits, an up to date list of the library. The reason behind this is to keep you up to date at all times, you will not have to wait the three months, or so, for Monitor to arrive.

Because I have had very little response from you on how you want the Library to be structured, I have arranged it in the manner that seems the most logical and workable for me to provide a quick response to your requests.

Each disk will be filed under a heading according to the subject which the program/files relate. For example: LP1 is a Language disk, the subject is Pascal and it is the first in this particular section. Or, MMS1 is a MIDI disk containing files for Music Studio, again number 1.

There will also be a Support section which is intended to be used with programs/files for use with existing Commercial Software. For example, templates for VIP, Fonts for word processors or Printer Configurations and so on.

MIDI support files will be contained within the MIDI section because of the nature of the subject. I have given one example, but others already include Casio CZ Voices, 36 banks of voices for the Yamaha DX7 with the DXDROID, etc.

As other sections become available they will be introduced. Wherever possible, programs and files will be segregated to maintain integrity. If there is a demand for a mixture, I will try to oblige, this will be the exception, not the rule.

## What to do

The club has laid out a great deal of money to get the Library off the ground and in order to recoup these costs and to obtain new material, it is necessary to make a small charge. There are two services currently available. The first, you provide the disk with your request and the fee is £3.50. The second, we provide the disk (DS/DD) when the fee is £5.50. This includes all necessary return postage and packing.

Any member who submits material will have his disk returned, the contents having been copied into the Library, to be replaced by something very useful (or a request of your own) as a form of thanks. Please remember that if you do submit any material, it must qualify for the description of Public Domain, or something similar, i.e. no ripped off Commercial Software will be tolerated.

If at any time you wish to obtain the latest complete library list, just send a disk and £1.00, or just send £2.50 and we will supply a disk with the list recorded onto it.

## The ST Library is for subscribers only.

Let me thank those members who have kindly donated disks for inclusion. I must especially thank Marty and his group, Mat and his group, Sol and my friends in Canada and the U.S.A. The material they sent me is first class. In fact, some of the programs are outstanding. Of these Krabit on GAMES 2, the MIDI Sequencer on MSEQ 1, Cribbage on ED 1, Triple Yahtzee on ED 2 and Fastcopy on DISKUT 1 are worthy of a special mention.

Krabit, from Germany, is an outstanding chess game with fabulous graphics and a worthy opponent. The MIDI Sequencer is a thirty-two track program with features that are of equal quality to commercial programs costing £100's! For the card game enthusiasts, this version of Cribbage is very good. The same can also be said of that very popular dice game Yahtzee. This version allows up to six players with three cards per game. It is outstanding! Incidentally, it is written in GFA Basic. The graphics are excellent, it has a good randomizer - Yahtzee's appear with about the same frequency as they would with real dice. Fastcopy is an outstanding utility for backing up your disks. There are many interesting features, among which the FORMATTER is noteworthy. It is the only formatter I have come across which will format a disk up to eighty six tracks and up to ten sectors per track!! In real terms, a DSDD disk formats to almost 870K! There are so many good programs in the library it would take me too long to go into them all. You will not be disappointed!

Here follows are short description of the Library as it stands as we go to press:

### ADUTIL 1
EAMON Adventure System. Create your own adventure games!! Includes quite a few examples. 1MB, colour.

### ART 1
California Beach Girls. This requires 1MB and MONO! It is brilliant! (X-rated).

### ART 2
Portraits. 1MB colour. Very good! (X-rated).

### ART 3
Reveur 1. Art Nouveau. 1/2 MB colour.

### ART 4
Reveur 2. Art Nouveau. 1/2 MB colour.

### ART 5
Strip Show. 1MB. (X-rated).

### ART 6
Animation. 1/2 MB colour.

### ART 7
Perene. 1/2 MB colour. (X-rated).

### ART 8
Animation: (The Two Ronnies) 1MB colour and requires a RAM Disk.

### ART 9
Utilities, 21 items, 160K. It includes Palette, Triangle, Windpics.

### DESKACC 1
29 items, 230K.

### DISKUT 1
33 items, 313K. Includes Fastcopy and the special loaders for the SOUND SAMPLED demos mentioned in the SOUND section.

### ED 1
Includes Cribbage, Firstmath, Molecules. 1/2 MB colour.

### ED 2
Includes Barnyard, Trivia, Triple Yahtzee. 1/2 MB colour.

### GAMES 1
22 items, 340K. Includes Dragon, Pacman, Pool, Nightcrawler, etc. 1/2 MB colour.

### GAMES 2
Krabit. 1/2 MB colour/MONO.

### LASM 1
ABSMON Assembler/Disassembler, PD from FASTer.

### LMISC 1
Includes ASM, STLOGO, Tinybasic, XLISP. 1/2 MB colour.

### LFB 1
Animtic files. Brilliant. Requires Fast Basic. 1/2 MB colour.

### LFB 2
Fast Basic Mini Database and Printer utility.

### MIXT 1
33 items, 325K. Includes .ACCS's, Term and Game(Yahtzee).

### MSEQ 1
32 track sequencer and files. 1/2 MB colour/mono.

### MDXDROID 1
18 Banks of sounds for DX7.

### MDXDROID 2
A further 18 Banks of sounds for DX7.

### MMS 1
Music Studio SONGS. 1/2 MB colour.

### MMS 2
Music Studio SONGS. 1/2 MB colour.

### SOUND 1
Sampled OXYGENE. 1MB colour. Brilliant!

### SOUND 2
Sampled SMALLTOWN, requires LOADER. 1MB colour/mono.

### SOUND 3
Sampled GOING GETS TOUGH, requires LOADER. 1MB.

### SOUND 4
Sampled WIND HIM UP, 1MB autoboot, colour/mono.

### SOUND 5
Includes DLX Piano, Popminis. 1/2 MB colour.

### SOUND 6
Sampled FOREIGN AFFAIRS, 1MB colour and is GREAT!

### UTIL 1
34 items, 310K. Includes: RDV-COPY, DATFSH, SECED, SQUEEZE. 1/2 MB.

### UTIL 2
29 items, 320K. Includes: JOSHUA, MOSES, SNAPSHOT, SUPERFORMAT, LABLEMAKER, etc.

### UTIL 3
47 items, 316K. Includes: COPYDISC, FASTLOAD, MAKE512, STCALC.

### UTIL 4
22 items, 295K. Includes: ARCHIVER, AUTODATE, LASER, PICSWITCH.

Not a bad start, is it? Every effort has been taken to ensure that the disks are either Public Domain or Share Ware. If you find that any are commercial and copyright, please let me know IMMEDIATELY, so they can be removed.

Requests should be sent to me at the club address.

# Hexadecimal Converter

By Ron Levy

It is often desirable, when writing utility programs, to be able to print a number to the screen in hexadecimal form rather than in the usual base ten. The following machine code routine will take a one or two byte value and return a two or four character string representing its hexadecimal value. Whilst it is of course possible to write a BASIC subroutine to perform this function, such a routine would be fairly large, and more importantly, would be quite slow due to the number of floating point calculations involved.

This fast machine code routine is completely re-locatable, and the most convenient way of storing it in BASIC is by placing it into a string. For example, the demo program 1 simply reads each byte from data statements and uses the CHR$ command to force the byte values into H$. The routine is called with the command: –

X=USR(ADR(H$),ADR(H4$),N)

Where N is the number you want converted. H4$ will now contain the hexadecimal equivalent of N, and this will be a 4 character string. If you are only converting a single byte character (perhaps the result of a peek instruction) then you will of course only want a two character string, and this is accomplished by taking the second half of H4$ only, i.e. H4$(3,4). You must include the command H4$="1234", since this simply causes BASIC to set the length of H4$ to 4 characters. I have included the memory lister program as an example of how to use the routine in your own programs. Memory lister will display the contents of eight memory locations per line, giving the base address on the left, eight two digit values in the middle, then the eight printable CHR$'s of these values on the right. Notice that POKE 766,1 is used to prevent the screen control codes (clear screen, insert line, delete line, etc) creating havoc with the display when the CHR$'s are printed!

Instead of reading data statements to create the string you can actually equate H$ to a literal string at the start of the program, the only difficulty is that most of the characters in H$ are graphic symbols. Print H$ and you will see exactly what I mean. There is however a crafty way of including this into a program line.
Print H$, use the cursor control keys to insert spaces in front of the printed string, then type 20 H$=". Then move the cursor to the end of the string and print the closing quote ("). On pressing return BASIC will accept this as a valid program line, and you can then delete the FOR NEXT loop used to create the string, together with its associated data statements.

## How The Routine Works

The routine takes advantage of the fact that Atari BASIC allows you to pass any number of integers (value 0-65535) through with the USR call. In this case we are passing two integers, the first being the memory address of H4$, and the second being the number we actually want converted. BASIC will use the processor stack for this, and it will first push to the stack a single byte whose value will be the number of arguments in the USR call – in this case it will be 2. Our routine will first pull this byte off the stack, but it will not store it since we know that there will always be two arguments. The next two bytes pulled from the stack will be the address of H4$ which will be stored in two bytes of page zero memory, to be used as pointers when storing the results. The last two bytes pulled will be the number to be converted, and they will come off high-byte first, low byte last.

Each byte, is split into two 4 Bit Nibbles and 48 is added to obtain the relevant ASCII number character. A test is then performed to see if the nibble was between 10 and 15, and if so a further 7 is added to obtain the characters A to F. The result is then stored directly into its correct position in H4$.

```
QE 0 REM Hexadecimal Memory Lister
LU 1 REM To Demonstrate The Use of
NU 2 REM The Machine Code Routine.
HE 3 REM .        Ron Levy.
NJ 4 REM
BV 10 DIM H$(38),H4$(4),P(8)
IK 11 H4$="1234"
MH 12 FOR I=1 TO 38
GD 13 READ X:H$(I,I)=CHR$(X)
VS 14 NEXT I:REM Routine Now Loaded.
IG 80 POKE 766,1
FL 90 ? "Starting Location ";:INPUT I
QX 100 FOR I=I TO 1000
CY 110 X=USR(ADR(H$),ADR(H4$),I)
YJ 120 IF H4$(4,4)<>"0" AND H4$(4,4)<>"8"
    THEN I=I-1:GOTO 110
EG 130 PRINT H4$;" ";
DS 200 FOR J=0 TO 7
AO 210 P(J)=PEEK(J+I)
MF 220 X=USR(ADR(H$),ADR(H4$),P(J))
EQ 230 PRINT H4$(3,4);" ";
GK 240 NEXT J
OT 300 FOR J=0 TO 7
DC 310 IF P(J)<>155 THEN PRINT CHR$(P(J))
    ;
GH 320 NEXT J
LM 390 PRINT :I=I+8
FU 400 NEXT I
DS 500 POKE 766,0:STOP
IN 32000 DATA 104,104,133,213,104,133,212
    ,160,0,152,74,104,176,5,72,74,74,74,74
    ,41,15,24,105,48,201,58,144,2,105
XO 32001 DATA 6,145,212,200,192,4,208,228
    ,96
```

Program 1

```
0100 ; A Routine to convert a 2 byte
0110 ; integer into its Hexadecimal
0120 ; equivalent and dump this into
0130 ; a BASIC string.
0140 ; The routine is to be held in
0150 ; a Basic string and is called
0160 ; by the command:-
0170 ;
0180 ;    X=USR(ADR(H$),ADR(H4$),N)
0190 ;
0200 ; Where:-
0210 ;    H$  holds the routine.
0220 ;    H4$ will hold the result.
0230 ;    N   is the integer.
0240 ;          Ron Levy.
0250 ;
0260 ADRLO   =$D4
0270 ADRHI   =$D5
0280 ;
0290         *=$600
0300         PLA
0310         PLA        Pull address
0320         STA ADRHI  of H4$ from
0330         PLA         the stack.
0340         STA ADRLO
0350         LDY #$00
0360 ;
0370 LOOP    TYA
0380         LSR A  See if the Y reg
0390         PLA     is odd or even.
0400         BCS LOBITS
0410         PHA
0420         LSR A   Move the high
0430         LSR A   nibble over to
0440         LSR A   the lower
0450         LSR A            nibble.
0460 ;
0470 LOBITS  AND #$0F Mask low nibble
0480         CLC
0490         ADC #$30 Add the offset
0500         CMP #$3A
0510         BCC NOADD Add more if
0520         ADC #$06  required (A-F)
0530 NOADD   STA (ADRLO),Y
0540         INY
0550         CPY #$04 Check for end
0560         BNE LOOP   of loop.
0570 ;
0580         RTS
```

# MINOTAUR

## A Machine Code Monitor
## By Martin Hillen

## Introduction

Having decided that the awesome power of Basic is not quite sufficient to cope with your latest program to predict the weather, or scroll that player up the screen before it's time to put another coin in the meter, you might think of resorting to machine code subroutines. Right, so there you are with a string of numbers poked into memory, or a source file of cryptic mnemonics and a binary load file of object code (if you're lucky enough to have an assembler). What next? Quickly you type in the following line anticipating the space invader to scroll smoothly up the screen:

```
X=USR(1536,ADR(PLAYER$),5)
```

Only you find that the whole computer locks up! The usual method of solving this minor problem is to run a utility program called a monitor which allows you to step through the machine code one instruction at a time. Unfortunately, virtually all the commercially available monitors (apart from products like Omnimon, etc.) won't run with the Basic cartridge present. This article presents a machine code monitor utility which solves this problem.

## About the Program

The monitor is designed to run with Basic and Atari DOS 2.0 and uses a little under 4.5K bytes of memory. To create a copy of the monitor, type in the program in Listing 1. As always, save the program before running it. When you run the program it checks all the data statements and displays any line with an error on the screen for correction. When all the data is correct the program will prompt you to insert a disk with DOS 2.0 on it. The program will then create an AUTORUN.SYS file containing the monitor. To use the monitor simply boot this disk with Basic. Remember that the monitor must be loaded on power-up.

## Monitor Commands

When you boot the disk with Basic you should see 'Minotaur loaded.' appear above the Basic 'READY' prompt. This means that the boot was successful. To enter the monitor from Basic simply press the system reset key while holding down the option key. The monitor will prompt for input with 'Yes ?'.

The following conventions will be used to describe the commands available:

d - corresponds to a decimal number.
n - corresponds to an 8 bit hexadecimal number.
addr - corresponds to a 16 bit hexadecimal number.
[x] - corresponds to an optional parameter.

The monitor assumes all numbers entered to be in hexadecimal.

### Examine/alter Registers

Syntax: R or R [n,n,n,n,n]

Typing R on its own causes the monitor to display the latest values of the 6502's internal registers in the following order:

A - the accumulator.
X - the X register.
Y - the Y register.
C - the carry bit (see note).
Z - the zero bit (see note).
P - the status register.
S - the stack pointer.

Note: The carry and zero flags are not registers in their own right, but are displayed on their own for convenience. The states of these flags are taken from the status register.

Typing R followed by one or more 8 bit numbers separated by commas allows you to alter the value of one or more specific registers in the same order as given above without altering any of the others. For example, R 10,,4 causes the accumulator to contain $10 (16 decimal) and the Y register $4 but does not alter any of the other registers. Note that it is not possible to directly change the C or Z bits but they can be altered by changing the value in the status register.

### Examine Memory

Syntax: addr [,addr]

This command displays the contents of memory in both hexadecimal and ASCII format. The display on the screen consists of three regions per line. On the left is the hexadecimal address of the first data on the line. Next follows up to eight 8 bit numbers which are the contents of memory after the address. Lastly are 8 characters which are the ASCII equivalents of the data immediately preceding.

Typing just one address causes only that location to be examined. Typing two addresses separated by a comma causes the region defined by them to be displayed.

### Alter Memory

Syntax: addr;n [,n,n,n...]

This command takes the 8 bit data specified after the semicolon and deposits it in the addresses following the memory address given. It is possible to skip memory locations as for the register command by typing the required number of commas with no data between them.

### Disassemble

Syntax: [addr] L

This command displays 20 lines of 6502 mnemonics corresponding to the data in the address specified and the following locations. If the address is not specified then the disassembly starts from the last location of the previous disassembly. Note that data which does not constitute a valid opcode is displayed as both a hexadecimal number and an ASCII character.

### Step

Syntax: [addr] S

This command executes a single instruction at the address specified, or if no address is specified then from the next address after the previous instruction. Again, unknown opcodes are displayed in hex and ASCII. The instruction just executed is disassembled and shown on screen along with the contents of the registers.

### Trace

Syntax: [addr] T

This is the equivalent of repeatedly entering a step command. The trace continues until either a BRK instruction is reached, an unknown opcode is encountered or the user presses the break key.

### Execute Subroutine

Syntax: addr G

This command causes the monitor to pass control to the address specified. The monitor will regain control if one of the following conditions is met:

1 The routine executes an RTS instruction.
2 The routine executes a BRK instruction.
3 The user presses the break key.

When the monitor regains control all the latest values of the registers are displayed.

## Binary Load

Syntax: O D:FILENAME.EXT

This command takes the specified binary load file and loads it into memory. The monitor will display the addresses to which the load is occurring. If the file has an incorrect format or the filename is incorrect or if the load file would overwrite the monitor then the load is aborted with the message 'Bad file.'.

## Numeric Evaluations

Syntax: V $n [+d-$n-d+$n] or V d [+$n-d+$n+d]

This command evaluates simple sums and displays the result in both decimal and hexadecimal. Hexadecimal numbers should be preceded by the dollar '$' sign. To convert between decimal and hexadecimal simply enter one value to the command.

## Return to Basic

Syntax: Q

This exits from the monitor and returns to Basic. Alternatively, press the system reset key without having pressed the option key.

## Helpful Hints

1. Remember that if you are tracing code which calls any OS routines (such as CIO) then the monitor will doggedly spend the next 10 minutes tracing the OS rather than your routine. To avoid this simply step through your program (S command) and when you see a jump to the OS, execute the OS routine with the G command. When the OS returns to the monitor continue stepping at the address immediately after the OS call.

2. When you are passing parameters to your routine from Basic it is very useful to stop the routine immediately that it has received them. This is done by using the memory modify command to change a suitable instruction in the routine to a BRK instruction (0). Now since DOS does not trap a BRK instruction it is essential to have the monitor's vectors in place while the Basic program is executing. Having entered the monitor, run the Basic cartridge with A000 G. Now the BRK instruction will cause the monitor to take control from Basic and you can replace the patched instruction and trace your routine.

3. The monitor resides in memory from $1F00 to $2D44. Hence, if you wish to use the DOS menu you must have a MEM.SAV file on the disk. You may still call DOS otherwise but you must not use system reset or the system will lock up.

Listing 1.

```
EW 10 REM A monitor compatible with BASIC
DY 30 DATA 0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,
   0,0,10,11,12,13,14,15
LI 40 DIM DAT$(91),HEX(22):FOR X=0 TO 22:
   READ N:HEX(X)=N:NEXT X:LINE=990:RESTOR
   E 1000:TRAP 120:? "CHECKING DATA"
JG 50 LINE=LINE+10:? "LINE ";LINE:READ DA
   T$:IF LEN(DAT$)<>90 THEN 220
UU 60 DATLINE=PEEK(183)+256*PEEK(184):IF
   DATLINE<>LINE THEN ? "LINE ";LINE;" MI
   SSING!":END
LY 70 FOR X=1 TO 89 STEP 2:D1=ASC(DAT$(X,
   X))-48:D2=ASC(DAT$(X+1,X+1))-48:BYTE=H
   EX(D1)*16+HEX(D2)
FG 80 IF PASS=2 THEN PUT #1,BYTE:NEXT X:R
   EAD CHKSUM:GOTO 50
BG 90 TOTAL=TOTAL+BYTE:IF TOTAL>999 THEN
   TOTAL=TOTAL-1000
LR 100 NEXT X:READ CHKSUM:IF TOTAL=CHKSUM
    THEN 50
MO 110 GOTO 220
ZR 120 IF PEEK(195)<>6 THEN 220
EQ 130 IF PASS=0 THEN 170
US 150 PUT #1,224:PUT #1,2:PUT #1,225:PUT
    #1,2:PUT #1,68:PUT #1,45:CLOSE #1:END
RA 170 REM
NK 180 ? "INSERT DISK WITH DOS 2.0, PRESS
    RETURN";:DIM IN$(1):INPUT IN$:OPEN #1
   ,8,0,"D:AUTORUN.SYS"
FV 190 PUT #1,255:PUT #1,255:PUT #1,0:PUT
    #1,31:PUT #1,150:PUT #1,45
QS 210 ? :? "WRITING FILE":PASS=2:LINE=99
   0:RESTORE 1000:TRAP 120:GOTO 50
IP 220 ? "BAD DATA: LINE ";LINE:LIST LINE
   :END
WY 1000 DATA 5D2AFFFFFF2A06FF7E2A06FFFF2A
   06FF4E2AFFFFFF2A06FF602AFFFFFF2A06FF5A
   03FFFF09032DFF84032DFF09,943
SI 1010 DATA 032DFF4B03FFFFFF032DFF8D03FF
   FFFF032DFF8718FFFFFF1827FF7B1827FF5718
   27FF5118FFFFFF1827FF6618,3
PN 1020 DATA FFFFFF1827FF8A00FFFFFF0030FF
   810030FF570030FF5400FFFFFF0030FF9300FF
   FFFF0030FFFF36FFFF3C3639,471
VV 1030 DATA FF6FFF9FFF3C3639FF3F36FFFF3C
   3639FFA536A2FFFF36FFFF241E21FF241E21FF
   991E96FF241E21FF421EFFFF,696
JM 1040 DATA 241E21FF691E9CFF241E21FF120C
   FFFF120C15FF750C6CFF120C15FF4B0CFFFFFF
   0C15FF630CFFFFFF0C15FF0F,913
KV 1050 DATA 33FFFF0F331BFF72337BFF0F331B
   FF4533FFFFFF331BFF9033FFFFFF331BFF0824
   FFFFFF0000FF081008FFFF80,955
US 1060 DATA 80FF4022FFFFFF0404FF0882FFFF
   FF8484FF8024FFFF000000FF081008FF808080
   FF4022FFFFFF0404FF0882FF,260
ME 1070 DATA FFFF8484FF0824FFFFFF0000FF08
   1008FF808080FF4022FFFFFF0404FF0882FFFF
   FF8484FF0824FFFFFF0000FF,891
DQ 1080 DATA 081008FFA08080FF4022FFFFFF04
   04FF0882FFFFFF8484FFFF24FFFF000000FF08
   FF08FF808080FF4022FFFF04,230
ZQ 1090 DATA 0402FF088208FFFF84FFFF102410
   FF000000FF081008FF808080FF4022FFFF0404
   02FF088208FF848482FF1024,268
JV 1100 DATA FFFF000000FF081008FF808080FF
   4022FFFFFF0404FF0882FFFFFF8484FF1024FF
   FF000000FF081008FF808080,995
DB 1110 DATA FF4022FFFFFF0404FF0882FFFFFF
   8484FF414443414E4441534C424954434D5043
   5058435059444543454F5249,858
PM 1120 DATA 4E434C44414C44584C44594C5352
   4F5241524F4C524F525342435354415354585З
   545942434342435342455142,281
DC 1130 DATA 4D49424E4542504C425643425653
   4A4D504A535242524B434C43434C44434C4943
   4C5644455844559494E5849,642
LG 1140 DATA 4E594E4F50504841504850504C41
   504C505254495254535345435345445345495З4
   4158544159545358545841 54,155
LV 1150 DATA 585354594120FFFFD8A9448DE702
   A92D8DE802AD1FD0C903F00160A9008D5A2C8D
   572C8D642CA9808DC602A906,289
PB 1160 DATA A22BA04E205CE4A9208D0602A92B
   8D0702A22CA0B420EF2AA9018552A2509AA900
   8D572C20EB2A20E42AA2008E,792
DX 1170 DATA 502CA9C48D4403A92B8D4503A905
   8D4203A97F8D48038E4903205 6E430D1A0FFC8
   B9C42BC99BD0F88C4A2CC000,936
GV 1180 DATA F0C0A000B9C42BC920D01498AABD
   C52B9DC42BE8EC4A2CD0F4CE4A2C4C2E22C8CC
   4A2CD0DFA000B9C42BC93090,8
EI 1190 DATA 2AC947B026C941B004C93AB01EA2
   018E502C203F29AE442C8600AE452C8601C99B
   D008A9018D4B2C4C7E27A200,259
TF 1200 DATA DD962BF00EE8E00A90F6EE492C20
   AE2A4CEC218A0AAABDA02B8D462CBDA12B8D47
   2C6C462C4C9122C8B9C42BC9,647
OH 1210 DATA 9BD0F5AD502CF0F0AE9B2C9AA922
   48A9D648AD9A2C48AE9B2CAC992CAD972CEE5A
   2C286C000088D972C8E9B2C,41
EN 1220 DATA 8C992C688D9A2CBA8E9B2CA9008D
   5A2CA2509A4C95244C9122C8B9C42BC99BD0F5
   A0008C652C8C662CA9008D67,281
GT 1230 DATA 2C20242A208C2AA002B10099A12C
   8810F8AABD001FC9FFF040A8BD00208D632C20
   062AA200B900219D6E2CC8E8,772
YW 1240 DATA E003D0F4208223AD642CF006A900
   8D642C60A99B200D2A20DD2AAD582C300AEE65
   2CAD652CC914D0A64CEC218D,612
YY 1250 DATA 672CAE642C8E622C8D4B2C8D642C
   207E2720062AAE622CF001604C5723A9048D61
   2CA9082C632CF00160A9202C,435
UI 1260 DATA 632CF005A928200D2AA9102C632C
```

```
   F005A923200D2AA924200D2AA9402C632CF03B
   18ADA22C101549FF8DA22CA5,318
WW 1270 DATA 0038EDA22C8DA22CA501E9004CE8
   2365008DA22CA50169008D512CADA22C186901
   8DA22CAD512C69008DA32C20,530
BU 1280 DATA 172AAD632C100C20062AEE662CAD
   A32C20172AADA22C20172A20062AEE662CA904
   2C632CF00AA92C200D2AA958,987
TZ 1290 DATA 200D2AA9202C632CF005A929200D
   2AA9022C632CF00AA92C200D2AA959200D2A60
   4C74E44C9122AD502CD0F88D,922
ZW 1300 DATA 552CA204BD972C9D9C2CCA10F7C8
   203F29C99BF004C92CD0DEAD482CF00DC903B0
   D5AD442CAE552C9D9C2CB9C4,495
HF 1310 DATA 2BC99BF012EE552CAD552CC905D0
   D1C8B9C42BC99BD0B3A204BD9C2C9D972CCA10
   F720242AA22DA01E20EF2A20,35
AV 1320 DATA 2F2AA22CA07C20EF2A4CEC214C91
   22C8B9C42BC99BD0F5A22DA00A20EF2AA9FF8D
   642C20FF22AD672CF00AAD57,510
XS 1330 DATA 2CF00268684CEC21ADA12CA207DD
   1A26F005CA10F8302CA9008D512C8A4AAA9003
   EE512C8D22262D9A2CF002A9,226
DA 1340 DATA 01CD512CF0034C0B26ADA22C8500
   ADA32C85014C0B26C920D02E38A500E9018500
   A501E9008501BA8AA8AE9B2C,549
CP 1350 DATA 9AA50148A50048BA8E9B2C98AA9A
   ADA22C8500ADA32C85014C0B26C900D0223BA5
   00E9018500A501E900850120,928
LD 1360 DATA 2F2A20DD2AA000AD572CF0056868
   8C572C4C0B2BC94CD00DADA22C8500ADA32C85
   014C0B26C96CD019ADA22C85,333
CR 1370 DATA 00ADA32C8501A000B100AAC8B100
   850186004C0B26C940D00EBA8AA8AE9B2C9A68
   8D9A2C4CB325C960D01A8A8A,220
NA 1380 DATA A8AE9B2C9A688500688501BA8E9B
   2C98AA9A20062A4C0B26A002A9EA995B2C88D0
   FAAC662CB9A12C995B2C8810,193
XJ 1390 DATA F7BA8E562CAE9B2C9AAD9A2C48AC
   992CAE982CAD972C284C5B2C088D972C8E982C
   8C992C688D9A2CBA8E9B2CAE,389
DF 1400 DATA 562C9A202F2A20DD2AAD572CF001
   604CEC2190B0D0F01030507001028040 4C9122
   C8B9C42BC99BD0F5A9FF8D57,634
JW 1410 DATA 2C20EB2AA22DA00A20EF2AA9008D
   592CA511D00AC611A9008D572C4C0B2BAD582C
   30F320BE24EE592CAD592CC9,38
RP 1420 DATA 0CD0E0F0CF4C9122AD502CD0F88D
   512C8D522CC8B9C42BA2FFC92DD002E8C88E54
   2C884C9D26A2FF8E542CC99B,119
RY 1430 DATA F056C92BF008C92DD0CEE88E542C
   C8B9C42BC99BF0C2C924D007C8203F294CB326
   20A1298D532CAD542CF01318,680
UV 1440 DATA AD512C6D442C8D512CAD522C6D45
   2C4CDE2638AD512CED442C8D512CAD522CED45
   2C8D522CAD532C4C88262024,825
DC 1450 DATA 2AA9248D682CAD522C20752A8D69
   2CAD4D2C8D6A2CAD512C20752A8D6B2CAD4D2C
   8D6C2C2044DAAD512C85D4AD,228
QP 1460 DATA 522C85D520AAD920E6D8D8A0FFC8
   B1F3300599702CD0F6297F99702CC8A99B9970
   2C20DD2A4CEC21AD502CF011,168
GB 1470 DATA A9008D512CC8203F29C99BF007C9
   2CF0034C9122AD482CF010C903B0F498AAAC51
   2CAD442C91008AA8B9C42BC9,434
GS 1480 DATA 9BF005EE512CD0D04CEC214C9122
   A9008DFE028D4F2C8D4E2CAD4B2CD02AC8203F
   29AE482CF0E4A501CD452C90,570
IX 1490 DATA 09D018A500CD442CB011AD442C38
   E5008D4E2CAD452CE5018D4F2CA9008D4B2C20
   242AA91BA20E9D862CCACA10,765
DJ 1500 DATA F9208C2AA000B100C99BF006BEA4
   2C9D862C20752ABEAC2CAD4C2C9D6E2CAD4D2C
   9D6F2CC8AD4E2C38E9018D4E,711
CG 1510 DATA 2CAD4F2CE9008D4F2CC9FFD018AD
   4E2CC9FFD01120DD2AAD642CF006A9008D642C
   604CEC21C008D0B020DD2A10,860
ER 1520 DATA 034CEC21A5001869008500A50169
   0085014CB827A220A9039D4203A9049D4A03A9
   2B9D4503A9C59D44032056E4,727
WA 1530 DATA 10034C2729A9079D4203A9009D49
   039D48032056E4C9FFF0034C27292056E4C9FF
   F0034C2729A9009D48039D49,971
DZ 1540 DATA 032056E410034C27299D44032056
   E49D4503C9FFD00C2056E49D44032056E49D45
   03C91F9004C92D907F20752A,211
PH 1550 DATA AD4C2C8D302DAD4D2C8D312DA220
   BD440320752AAD4C2C8D322DAD4D2C8D332DA2
   202056E48D512C2056E48D52,347
ZI 1560 DATA 2CAD512C38FD44039D4803AD522C
   FD45039D4903FE4803D003FE4903AD522C2075
   2AAD4C2C8D352DAD4D2C8D36,539
NE 1570 DATA 2DAD512C20752AAD4C2C8D372DAD
   4D2C8D382DA22DA03020EF2AA2202056E44C77
   28C088F009A22DA03A20EF2A,961
ZH 1580 DATA A220A90C9D42032056E44CEC21A9
   008D442C8D452C8D482CB9C42BA205DD962BF0
   11CA10F8F00CC99BF008C92B,14
LV 1590 DATA F004C92DD00160A20FDDB42BF008
   CA10F868684C91220E442C2E452C0E442C2E45
   2C0E442C2E452C0E442C2E45,703
TH 1600 DATA 2C8A0D442C8D442CEE482CAD482C
   C905B0D2C8D0A9A9008D442C8D452C8D482CB9
   C42BC99BF008C92BF004C92D,770
NN 1610 DATA D00160A209DDB42BF008CA10F868
   684C91220E442C2E452CAD442C8500AD452C85
   010E442C2E452C0E442C2E45,516
UR 1620 DATA 2C188A6D442C65008D442CAD452C
   65018D452CEE482CAD482CC906B0C3C8D0A6E6
   00D002E60160AE612C9D6E2C,127
VV 1630 DATA EE612C6020752A200D2AAD4D2C20
   0D2A60A22DA9209D682CCA10FA60A000B9972C
   20752ABE702A9D6E2CAD4D2C,314
RK 1640 DATA 9D6F2CC8C005D0E9A017AD9A2CA2
   312901D002A2308A996E2CA019AD9A2CA23129
   02D002A2308A996E2CA99B8D,276
YY 1650 DATA 8E2C600E11141B1E48290FAABDB4
   2B8D4D2C684A4A4A4AAABDB42B8D4C2C60A500
   20752A8D6A2CAD4D2C8D6B2C,210
SF 1660 DATA A50120752A8D682CAD4D2C8D692C
   A93A8D6C2C60AD492CF00CA9008D492CA22CA0
   E54CEF2AB9C42BC99BD00488,105
JY 1670 DATA B9C42B498099C42BA22CA0F620EF
   2AA22BA0C44CEF2AA22CA0684CEF2AA22DA004
   4CEF2AA22CA0968E45038C44,458
TI 1680 DATA 03A200A9098D4203A97F8D48038E
   49032056E48C582C60086829EF482820242A20
   8C2A202F2A20DD2A4CEC21AD,217
WZ 1690 DATA 5A2CD0026840D8A9008D5A2C8E98
   2C8C992C688D972C688D9A2C6838E902850068
   E9008501BA8E9B2C584C0B2B,691
GN 1700 DATA AD5A2CF040A511D03CA0FF8411C8
   8C5A2CBABD01018D992CBD02018D982CBD0301
   8D972CBD04018D9A2CBD0501,291
MZ 1710 DATA 8500BD060185018A1869068D9B2C
   A90B9D0501A92B9D06014C5FE42C3B4C535447
   5152564F7E273E27F722AF24,988
FX 1720 DATA 2926AC223A2440246A2638283031
   32333435363738394142434445460000000000
   000000000000000000000000,637
FT 1730 DATA 0000000000000000000000000000
   00000000000000000000000000000000000000
   000000000000000000000000,637
FW 1740 DATA 0000000000000000000000000000
   00000000000000000000000000000000000000
   000000000000000000000000,637
FZ 1750 DATA 0000000000000000000000000000
   00000000000000000000000000000000000000
   000000000000000000000000,637
SU 1760 DATA 00004CF525000000000000000000
   00000000000000000000000000000000000000
   000000000000000000000000,995
UO 1770 DATA 000000000000000000000000009B
   00000000FF000000000000000001030507090B
   0D0F000306090C0F12152020,617
BY 1780 DATA 4D696E6F746175722C206279204D
   6F6E69746F7220313938372020202020202020
   202020202012121212121212,326
WP 1790 DATA 129B556E6B6E6F776E20636F6D6D
   616E642E9B53796E746178206572726F722E9B
   596573203F9B202020202020,400
TY 1800 DATA 2020202020202020202020202020
   41202058202059202043205A20502020539B46
   4646462D464646469B426164,908
AS 1810 DATA 2066696C652E9BA50C8DA921A50D
   8DAA21A9A8850CA921850DA9448DE702A92D8D
   E802A22DA06920EF2A186020,566
ON 1820 DATA 4D696E6F746175722206C6F616465
   642E9B00000000000000000000000000000000
   000000000000000000000000,263
```

# Adventure into the ATARI by Steve Hillen

*Part 4 in writing your own adventure.*

So far, we have discussed how an adventure is designed, how to compress the data needed, and how a program can 'understand' sentences. This time, methods of making the screen layout more interesting will be described. Before starting though, you will have to decide whether the extra memory used by the screen could be better used for the actual adventure. A more complex screen leads to more complicated programming as well.

If you have not been put off by the above drawbacks, then the program listed in this article may be of some help. It shows how you can set up a split screen display where the top few lines hold current information, and the lower lines scroll as per normal, i.e. just like a Scott Adams adventure. The actual method of doing this is quite complicated and involves a number of distinct stages.

The first task is to decide how many lines will be in each part of the screen. In this example, I chose 7 non-scrolling lines and 16 normal ones. Each of the non-scrolling lines takes up 40 bytes of memory, so my top screen needs a total of 280 bytes. I also need a new display list which itself needs a further 40 bytes. This is the total amount of memory that has to be reserved for our new display. Actually, reserving the memory is simple.

```
00010 ;Line eraser for AC15
00020 ;
00030 ;
00040          .OR $4000
00050          .TF "D:SCREEN.OBJ"
00060 ;
00070 PTR      .EQ 212
00080 ;
00090 ;
00100          PLA
00110          PLA
00120          PLA
00130          TAY
00140          PLA
00150          STA PTR+1
00160          PLA
00170          STA PTR
00180          LDA #0
00190 LOOP     STA (PTR),Y
00200          DEY
00210          BPL LOOP
00220          RTS
```

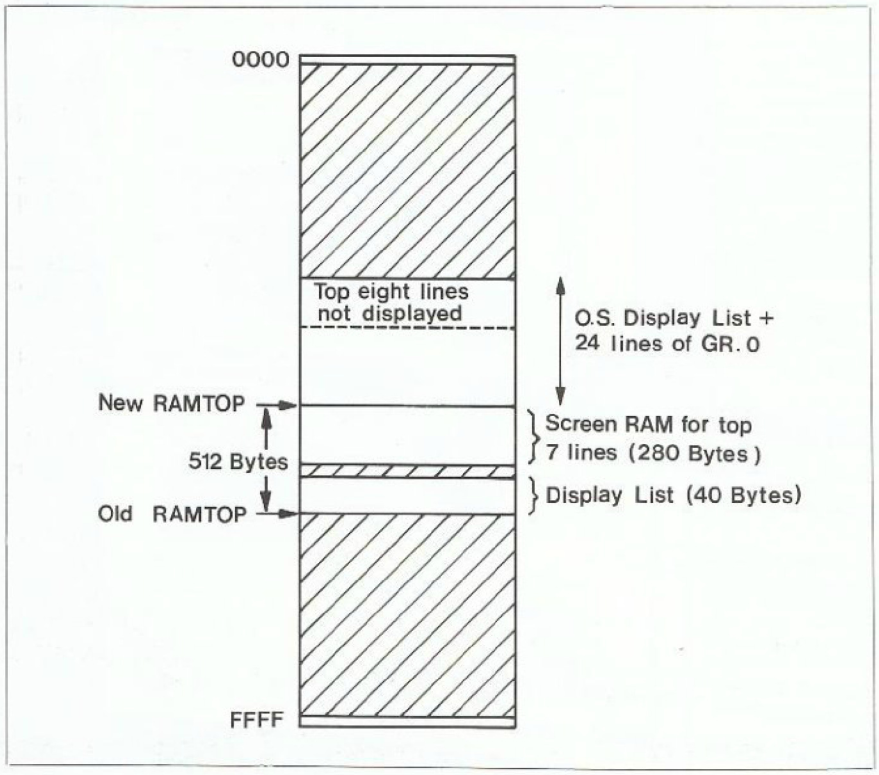

0000
Top eight lines not displayed
O.S. Display List + 24 lines of GR. 0
New RAMTOP
Screen RAM for top 7 lines (280 Bytes)
512 Bytes
Display List (40 Bytes)
Old RAMTOP
FFFF

Figure 1.

The number of pages (256 byte blocks) that are required for our total of 320 bytes is obviously two, so 2 is subtracted from the RAMTOP pointer as per lines 20 to 30. By calling Graphics 0 after changing RAMTOP, the normal screen is set up below our reserved two pages (see memory map, Figure 1).

The next stage is to sort out the lower scrolling lines. Since the normal Gr. 0 screen has 24 lines, we need to ignore the top 8 if only the bottom 16 are to be displayed. We must display the bottom 16 lines rather than the top 16 as this is the only way to ensure that the lower screen scrolls properly. The address of the 8th line down is calculated in line 90.

After this, the address of where the new display list is to be constructed is calculated (line 100). It has been put as high in memory as possible to avoid problems with the screen clear operation. Lines 105 to 120 actually POKE the new display list into the correct area. It is not really within the scope of this article to explain about display lists, but there are many sources of information on the subject. Basically, it is a list of numbers that determine what areas of memory are displayed on the screen. The data in line 120 can be broken into 5 sections. The first 3 numbers display the usual 3 blank lines, then 7 lines are displayed from our reserved area, then 1 blank line to act as a separator, then the standard 16 normal lines. Lastly, there is the instruction to jump back to the start of the display list (65).

Line 130 sorts out the first SEPARATOR (Load Memory Scan) operands which govern which part of memory is actually shown. In this case the address of our reserved area is given. Line 140 calculates the numbers that point to the second part of our display area, and line 150 completes the display list by supplying the jump back operands. Finally, the display list pointer is adjusted to point to our new display list (line 160). The split-screen is now being displayed.

Further down in the program are two subroutines starting at lines 500 and 520. The first one enables you to print directly to the top 7 lines, and the second swaps the printing back to the normal scrolling section. The cursor is switched off for the top screen, and re-enabled for the lower screen. There are a few limitations of this method of splitting the screen. Nothing written to the top 8 lines of the normal scrolling section will ever be displayed. Also, you must never try to write anything outside lines 1 to 7 of the top display, or call any of the editing keys, such as screen clear whilst in the top display as they will overwrite the display list and possibly crash the program.

The first problem is easily overcome

# THE O·S· CONTROLLER CARD

## HARDWARE

The O.S. Controller Card is a 'state of the art' hardware board, which allows you to have two operating systems in your XL/XE Computer. One is the standard Atari O.S., the other is programmable. The product has been designed to:
a) be extremely cost effective as it alleviates the necessity to purchase new Eproms every time you want a new utility (of course that is assuming that someone is even producing an Eprom with the utility that you require).
b) be easy to use – no programming knowledge at all is required to use the O.S. Controller Card.
c) allow enormous flexibility and potential to the point that the user may create a completely new operating system for the computer – even from Basic
d) be totally compatible with existing software and be easily up-dated if required.

## UTILITY DISK #1

This is the first of many utility disks and offers a host of utilities to get you started in the world of O.S. customizing.

**1) THE SNAPSHOT PRINTER DUMPER:** – at the touch of a button this will accurately reproduce whatever's on your screen to your printer in full A4 size, when the dump is complete your program will continue where you left off. Besides the normal draft quality we have included our unique SHQ™ mode, this is our graphics equivalent of NLQ – has to be seen to be believed. There is also a mode similar to SHQ™ that compensates for printer's with inaccurate line feeds. (Epson & Compatibles)

**2) THE DUMPER:** – an ideal utility to upgrade all your cassette software to disk or make disk to disk backups. The Dumper is not affected by any cass/disk protection at all! The Dumper can be activated repeatedly through-out your program making it ideal for software that has no save game option.

**3) O.S. BOOT MENU:** – a multiboot compatible menu that's O.S. resident, simply press the HELP key and you are instantly back at the menu, now either load a new file or press system reset and you are back to the beginning of your previous program.

**4) I/O ANALYZER:** – While you are loading a boot disk the I/O analyzer will constantly send updated information to the printer about the drive and the computer.

**5)** There are also such features to change the master O.S.character set, switch on or off auto scroll, noisy I/O, key click, inverse flash, fast cursor, alter the primary colour defaults and lock them, save any pre-XL/XE O.S. with or without customised ROMS.

## CONTINUED SUPPORT

1029 Snapshot Dumper £12.95.
SHQ Printer Link £16.95 (This will give you the SHQ finish to graphics program like Typesetter).
Ultraspeed Handler T.B.A.
Dumper Toolkit T.B.A.

## INSTALLATION

Installing the O.S. Controller Card yourself will require the insertion of 2 IC sockets and the soldering of 1 wire, alternatively have it installed by us free of charge but please add £5.50 p&p and ins. (pack securely).

## O.S. SWITCH PACK

The O.S. switch pack is an optional extra that when added to the O.S. Controller Card will give the you following features.-
1) Switch basic on and off (norm on)
2) Switch the board lock on and off. With this switch on, the O.S. Controller Card cannot be re-programmed thus eliminating any accidental tampering with your customised O.S. by other software
3) Switch between the Atari O.S., the Customised O.S., and the Permanent Ram O.S. even while your software is running.

## MAIL ORDER ONLY

PRICE LIST

| | |
|---|---|
| O.S. CONTROLLER CARD FOR THE 800XL COMPUTER | £69.95 |
| O.S. CONTROLLER CARD FOR THE 130XE COMPUTER | £79.95 |
| OPTIONAL O.S. SWITCH PACK | £ 9.95 |
| RETURN P&P & INSURANCE IF INSTALLED THROUGH COMPUTER HOUSE | £ 5.50 |

CHEQUES/POSTAL ORDERS SHOULD BE MADE PAYABLE TO:-
**COMPUTER HOUSE (Partnership) 'THE IMAGINATION STATION'**
**TEL: 01-731-1276**
**ADDRESS: 14 Romily Court, Landridge Road, Fulham, LONDON SW6 4LL**

simply by one POSITION statement, but erasing sections of the top screen is more complicated. There is a very short machine code subroutine in lines 170 to 200 to do the erasing quickly. It is called by the following:

X=USR(ADR(ER$),N,NN)

where N is the number of characters to erase (less than 128!), and NN is the address at which to start erasing, e.g. to erase lines 2 to 4 of the top screen, N = 3 * 40 = 120, and NN = AREA * 256 + 40 to point to line 2. In case you're still wary about using machine code, the alternative is to print a row of spaces to delete a line in the top screen.

Well that's about it for the program. Hopefully you'll be able to adjust it to your needs, and it should work well provided you respect its limitations. Next time, I'll try to put all the routines from the last 4 articles together to make a complete adventure.

```
HC 10 REM Reserve area for new screen
RD 20 AREA=PEEK(106)-2
FQ 30 POKE 106,AREA
LS 40 GRAPHICS 0
KE 50 REM Now sort out display list
VT 60 REM We are using 7 lines on top
HV 70 REM and 16 normal lines.
PB 75 NLINES=7
ZQ 80 LOW=PEEK(88):HIGH=PEEK(89)
PB 90 LMS2=LOW+256*HIGH+(NLINES+1)*40
WL 100 A=0:DL=(AREA+2)*256-40
NI 105 READ D:IF D=-1 THEN GOTO 130
VR 110 POKE DL+A,D:A=A+1:GOTO 105
BX 120 DATA 112,112,112,66,0,0,2,2,2,2,2,
   2,112,66,0,0,2,2,2,2,2,2,2,2,2,2,2,2,2
   ,2,2,65,0,0,-1
SV 130 POKE DL+5,AREA
UL 140 VV=LMS2:GOSUB 1000:POKE DL+14,LO:P
   OKE DL+15,HI
MM 150 VV=DL:GOSUB 1000:POKE DL+32,LO:POK
   E DL+33,HI
WX 160 POKE 560,LO:POKE 561,HI
US 170 REM M-c data for top screen clear
GW 180 DIM ER$(18):FOR A=1 TO 18:READ D:E
   R$(A,A)=CHR$(D):NEXT A:DATA 104,104,10
   4,168,104,133,213,104,133,212,169,0
TX 190 DATA 145,212,136,16,251,96
RJ 200 X=USR(ADR(ER$),40,AREA*256)
GC 220 GOSUB 500:POSITION 2,0:? "HI":GOSU
   B 520:? "HO":GOTO 220
YE 500 REM Write to top screen
AX 510 YYY=PEEK(84):XXX=PEEK(85):POKE 752
   ,1:POKE 88,0:POKE 89,AREA:RETURN
MN 520 REM Write to bottom screen
AK 530 POKE 84,YYY:POKE 85,XXX:POKE 752,0
   :POKE 88,LOW:POKE 89,HIGH:RETURN
LH 1000 HI=INT(VV/256):LO=VV-256*HI:RETUR
   N
```

Listing 1.

## 2 BIT SYSTEMS: MUSIC PRODUCTS FOR THE 48K ATARI

### REPLAY

Replay is a complete sound sampling system that allows you to produce real speech/music on any 8-Bit ATARI:

**Features:**
* Sample rate selectable from 6Khz to 21Khz
* Sample playback through TV/Monitor
* Allows samples to be used in your own Basic programs
* Supplied on cartridge, no user memory lost
* Records from HI-FI or external recorder
* New V2.0 Replay program with bigger option

Also included in the REPLAY PACKAGE

**Digidrum:** Digital drum sequencer (no hardware required), allows you to create your own Drum rhythms using 8 sampled drum sounds.

**Digisynth:** Simple sample sequencer (no hardware required) allows you to play tunes using sampled sounds (dog barks, guitars, voices etc.).

**Echo:** A real time Echo program to create special effects. Delay variable between 2mS to 1 sec.

**PRICES**
REPLAY system (cartridge, software, Digidrum and Digisynth) only £39.95.

### MIDI MASTER

A full feature MIDI interface for your ATARI, allows you to take full control of synths etc.

**Features:**
* MIDI in/out cables for easy connection
* 8 Track Real time sequencer with tempo correction
* Casio CZ series voice editor (allows you to edit and store voices on Disk/Tape)
* Yamaha DX100/21 series voice editor
* Music player program (allows you to play Music composer or AMS2 files, via MIDI)
* Now includes DX7 voice editor.
* CZ menu and split program

NB: We will be producing other voice editors for different synths, so if you own a different synth, get in touch.

**FUTURE RELEASES**
To enhance MIDI MASTER to include a 16 track polyphonic sequencer.

**PRICES**
MIDI MASTER (interface CABLE plus software) only £29.95.

### PERCUSSION MASTER

A high quality professional drumkit for the ATARI.

**Features:**
* External D/A and filter to ensure high quality sound, which plugs into joystick ports 1 & 2.
* 9 Sampled drum sounds.
* Polyphonic Rhythm editor using pull down menus and windows
* Capability to load new drum sounds from Disk
* Includes enhanced version of REPLAY software, allowing REPLAY owners to sample sounds with a far greater resolution.
* Includes digital echo program for use with the REPLAY cartridge.

* 3 Channel polyphonic
* 100 Song entries
* Audio output via hi-fi

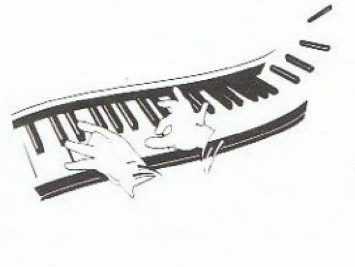
**PRICE**
PERCUSSION MASTER hardware & software £29.95

### DIGIDRUM II

A digital drum machine/sample sequencer that requires no extra hardware.

**Features:**
* 9 sampled drum sounds
* Step time pattern entry
* Capability to load Replay samples into any of the 9 sample positions
* KITS can be loaded or saved to disk or cassette
* 2 Channel polyphonic (any two drums)
* 100 Song entries (with complex looping)
* 32 Patterns (4-32 Beats/Bar)

**PRICES**
DISK £6.95
TAPE £5.95



**ORDER FORM**

Please send me:

☐ Replay System(s) on Disk/Tape*
☐ Digidrum II on Disk/Tape*
☐ Midi Master Interface(s) on Disk/Tape*
☐ Percussion Master System(s) Disk ONLY

I enclose a Cheque/Postal Order*
for £ (Payable to 2 Bit Systems)

* Delete as applicable

Send to: **2 Bit Systems, 44 Morcom Road, Dunstable, Beds. LU5 4EG.**

Name ..........

Address ..........

..........

..........

Postcode ..........

# XIO for Beginners

**By Ron Levy**

Most of you will have noticed this strange command (or set of commands, effectively), but many of you may be unsure of its use or the reason for its existence. Well let's put this right!

Your BASIC programming language has a variety of commands for controlling devices, or data input and output operations (I/O), such as OPEN, PRINT, GET, PUT, etc. and BASIC refers to each of these internally with a unique number. For example, the OPEN command is numbered 3, PRINT is numbered 9, and INPUT is numbered 5. You don't usually have to worry about these, since BASIC has these nice, easy to remember labels! However, not all of the commands and functions available have names assigned to them, probably because Atari ran out of space in the 8K BASIC ROMs. Because of this Atari designated the XIO command as being a means of accessing these unnamed functions.

## The Format of XIO

The XIO command has a fixed format, and this is as follows:

XIO,CMD,#CHAN,VAL1,VAL2,STRING$

These values associated with the XIO command will of course vary, and they depend upon the function you perform, but here is a quick explanation of their meaning.

CMD: This is the number which represents the command which is being performed.

#CHAN: This is the 'channel' number, same as you would reference in an OPEN command (i.e. #1, #2, #3, etc.).

VAL1 and VAL2: These two integers (value 0 to 65535) are not actually used in many of the commands but a few XIO commands use them to pass parameters to the operating system.

STRING$: This is a text area for the XIO command to pass text data such as filenames or that contained in a BASIC string.

## The Commands

So just exactly what commands are available? Well, the answer is not that simple, because the Atari 8 bit machine's operating system was designed to allow for easy expansion in terms of devices it will support. In plain english, this means that if you know how to you can actually add machine code to support more devices (or more functions for the existing devices) and provided that you have changed the relevant operating system tables, it will all be accessible from BASIC.

Take a look at Table 1. Here is a list of XIO commands which you will already have on your machine. If you examine Table 1 a little more closely however, you will notice that the Table is in fact for a machine which has a disk drive attached, and which has DOS loaded (Disk Operating System). This is an immediate example of what I have just explained about 'adding extra commands' to the system!

It's time now to take a look at some of these commands in a little more detail, and see how the XIO command works. You may have noticed that Table 1 includes several of the commands which are catered for more simply by BASIC. Rather than just skipping these I think it would be helpful to many of you if I gave some examples of how you can use some of the XIO commands in place of their BASIC counterparts. So here goes.

### XIO 3
### Open a file or device

The XIO version of Basic's OPEN command even looks very nearly identical! Take a look at Program 1. On line 20 you will see XIO shown as follows:

XIO 3,#1,8,0, "D:TEST.TXT"

Just as with the OPEN command, the 8 signifies 'open the file for output'.

### XIO 9
### Put record or 'PRINT'

Line 50 of Program 1 shows how to print a string of text to a file, and it has the following format:

XIO 9,#1,8,0, "TEST LINE"

Notice that we must place the 8 in the command, signifying an output operation. Instead of using a literal string (in quotes), we could in fact have printed the contents of a variable to the file, for example:

XIO 9,#1,8,0,A$

### XIO 12
### Close file

The CLOSE command in Program 1 is replaced by the close XIO, shown again below:

XIO 12,#1,0,0, "D:"

This has exactly the same effect as Basic's CLOSE statement.

## Reading our File

We can even read back our file using the XIO command, and once again, although it is unnecessary, it does help to demonstrate the use of XIO. Take a look at Program 2. The OPEN statement is the same, except that the '4' signifies that we want the file opened for input this time.

### XIO 5
### Get record or 'input'

The format for XIO 5 in our program is as follows:

XIO 5,#1,4,0,A$

We are of course using A$ to store the record read in from the file. Type in Program 1 and run it. You should then hear it creating a file, which will be called TEST.TXT. Now type in and run Program 2. This time it will read the file and print the string of text that it found. You should have the words 'Test line' on your

| COMMAND | OPERATION | EXAMPLE |
|---|---|---|
| 3 | Open a file | Same as BASIC's OPEN |
| 5 | Get record | Same as BASIC's INPUT |
| 7 | Get characters | Same as BASIC's GET |
| 9 | Put record | Same as BASIC's PRINT |
| 11 | Put characters | Same as BASIC's PUT |
| 12 | Close a file | Same as BASIC's CLOSE |
| 13 | Status request | Same as BASIC's STATUS |
| 17 | Draw a line | Same as BASIC's DRAWTO |
| 18 | Fill an area | XIO 18,#1,0,0,"S:" |
| 32 | Rename a file | XIO 32,#1,0,0,"D:OLD.DAT,D:NEW.DAT" |
| 33 | Delete a file | XIO 33,#1,0,0,"D:FILE.DAT" |
| 35 | Lock a file | XIO 35,#1,0,0,"D:FILE.DAT" |
| 36 | Unlock a file | XIO 36,#1,0,0,"D:FILE.DAT" |
| 37 | Point | Same as BASIC's POINT |
| 38 | Note | Same as BASIC's NOTE |
| 254 | Format a disk | XIO 254,#1,0,0,"D1:" |

Table 1.

screen, which are the contents of A$. Something you should have noticed is that your computer also printed the remaining 244 spaces in A$! Why is this?

The answer is quite simple. The XIO 5 command uses A$ as a kind of output buffer, it just dumps the received data into it without either checking that A$ is big enough, or changing the recorded length of A$! This is why I have to set each character of A$ to a space, this both cleared garbage characters from the string AND made Basic think its length was 255. Look at A$ now using the following command:

PRINT ASC(A$(11))

You should see that XIO 5 has even placed the Carriage Return code into the buffer!

## Beware

Something which you should be careful of if you decide to experiment with XIO 5 is that you never try to read a record which is longer than the DIMed length of your receiving string. If it happens then you will find that you will corrupt other strings or corrupt parts of your program, forcing Basic to 'lock-up'!

## Some useful commands

Let's move on now, and take a look at some more useful XIO's, the kind which you might need to use in your programs. I will assume that you have already run Program 1 by now, so that you have the file TEST.TXT on your work disk.

### XIO 32
### Rename file

By using XIO 32 your program can change the name of a file just as you normally would with the DOS menu. Type the following line, (while Basic is installed), and you will see what I mean.

XIO 32,#1,0,0,
"D:TEST.TXT,TESTA.TXT

Now go to DOS and look at the disk's directory, and you should see that the file has been renamed. Put it back to its old name (TEST.TXT) using the DOS menu.

### XIO 35
### Lock file

When you have returned to Basic type the following line:

XIO 35,#1,0,0, "D:TEST.TXT"

Go to DOS once again and list the directory, and you will see an asterisk next to our file, signifying that it has been 'locked'.

### XIO 36
### Unlock file

Now let's unlock our file. Type the following line in Basic:

XIO 36,#1,0,0, "D:TEST.TXT"

You should now find that the file is once again unlocked!

### XIO 254
### Format disk

This is a useful command, especially if you have run out of disk space and need to format a new disk so that you can save a program you are writing! In fact, once you have formatted your disk with this command you can write DOS files to it by typing SAVE "D:DOS.SYS". Instead of saving a BASIC program to disk, this will save the boot stage of DOS II (not the menu part), and if you are using K-DOS it will save the entire DOS. Phew, all from Basic, I bet most of you did not realise this was possible!

## Take Care

The last five XIO commands we looked at must each be given an UNUSED channel. For example, we have been using #1, but if our XIO commands were in a large Basic program where #1 was already in use, we would get some strange error numbers. You should therefore make sure that your channel numbers don't clash!

## Fill

The last XIO example I will give is the Fill, or XIO 18. An example of the format is:

XIO 18,#1,0,0, "S:"

The fill function is used to fill in shapes which have been drawn on a graphics screen using the PLOT and DRAWTO commands. It is, however, very limited in that the way it operates is very simple, in fact it is probably quite a big disappointment to most people when they learn how to use it for the first time!

Type in and run Program 3. Notice that it draws a line on the right hand side of the screen, and then proceeds to fill it partially.

### How Does it Work?

The process the computer follows when executing a fill is very simple indeed. It goes as follows:

1) Draws an imaginary line between the pixel last referenced to in a PLOT or DRAWTO, and the pixel refered to in the POSITION command.

2) Goes to the beginning of this line, and for each pixel along the imaginary line, draws a horizontal line out to the right, until it meets another lit (non-zero) pixel. If it does not meet another lit pixel, it will 'wrap around' the screen until it meets itself!

Following this logic, it should be easy to see that in order to draw a square, or filled-in box, you need only draw the right-hand edge, then give the top left and bottom left corner's coordinates in the PLOT and POSITION commands, the fill will take care of the rest. Try experimenting with different shapes, you will soon begin to see how limited Atari's fill is in practice! If you want to see a really good fill routine, take a look at Keith Mayhew's FAST FILL article and utility published in issue 9 of Monitor (back issues are available). This is a machine code routine you can add to your own basic program which will not only work many times faster than the Atari version, but will also fill even the strangest of shapes, even joystick drawn ones. If you like playing with graphics, I would heartily recommend trying it!

## Finally

Well, this concludes our tutorial on XIO. I hope that you are now confident enough to make more use of these commands in your programs, but if you are having difficulties with XIO then please write in, I'm sure I can help!

```
XI 1 REM ****** Program 1 ******
PV 2 REM Creates a file using XIO.
HC 3 REM .
YT 10 REM First create the file.
FH 20 XIO 3,#1,8,0,"D:TEST.TXT"
BA 30 REM
FV 40 REM Now 'print' to the file.
MQ 50 XIO 9,#1,8,0,"Test line."
BD 60 REM
NO 70 REM Now to 'close' the file.
LO 80 XIO 12,#1,0,0,"D:"
YX 90 END
```

Program 1.

```
YE 1 REM ****** Program 2 ******
FM 2 REM Reads a file using XIO.
HC 3 REM .
SV 5 DIM A$(255)
TQ 6 FOR X=1 TO 255:A$(X)=" ":NEXT X
BH 10 REM First open the file.
DH 20 XIO 3,#1,4,0,"D:TEST.TXT"
BA 30 REM
VR 40 REM Now 'input' from the file.
JU 50 XIO 5,#1,4,0,A$
BD 60 REM
NO 70 REM Now to 'close' the file.
LO 80 XIO 12,#1,0,0,"D:"
WR 90 PRINT A$
NQ 100 END
```

Program 2.

```
ZA 1 REM ****** Program 3 ******
JW 2 REM Using the Fill XIO (18)
HC 3 REM .
PP 10 GRAPHICS 8
YU 20 COLOR 1
ZN 30 REM First draw the right edge.
GP 40 PLOT 300,20
LH 50 DRAWTO 300,150
JF 60 REM Now define the left edge.
GY 70 PLOT 30,20
UP 80 POSITION 100,100
EO 90 REM Now give the fill colour.
LA 100 POKE 765,1
GA 110 REM And finally call the Fill.
PM 120 XIO 18,#1,0,0,"S:"
```

Program 3.

## MONITOR ON DISK

All the programs published in each issue of Monitor are now available pre-recorded on disk for you. No more need to spend frustrating hours of typing only to find the program won't run, and you are faced with the daunting task of bug hunting. The price is just £4.95 which includes postage and packing. Send a cheque/postal order made payable to the 'U.K. Atari Computer Owners Club' to Monitor Magazine, P.O.Box 3, Rayleigh, Essex. If you live in Europe add 50p, if outside Europe add £1.00. Please allow 14 days for delivery.

Monitor Disk 8.

Includes: Quickplot, a fast Graphics 8 Plot/Drawto handler. Nightmare Reflections, an exceedingly frustrating adventure. Matchbox, a concentration type memory game. Interrupts, 5 demo programs showing various uses of interrupts.

Monitor Disk 9.

Includes: Keyo, our own typing checker (these are the codes you'll see at the beginning of each line on the listings we publish). Multiboot Bootbase, database program for 'Multiboot disks'. Binload, binary loads from Basic. Happytyper, automatic line numbering. RAMdisk, for use with the 130XE. Fast Fill, a speedy shape filling utility.

Monitor Disk 10.

Includes: 3D Maze, escape from the maze in time if you can. PCB Paranoia, destroy your enemies before they get you. Disk Jacket, useful program for making your own disk covers. Chase, an excellent game, not to be missed.

Monitor Disk 11.

Includes: Hexadecimal Code Generator, better presentation for your programs. Cracking the Code, seven mini progs from the series. RAM Talker, with a little bit of hardware and this program, you can hear your own voice (for 400/800 only). HomeFM, a useful utility for use with Home Filing Manager.

Monitor Disk 12.

Includes: Another Boring Space Invaders Game, unlike its name, this game isn't boring at all. Get Motorised, four programs for use with the circuits shown. Mini-adventure, can you escape in one piece? Cracking the Code, Basic listing and assembler code for a drawing program. Opening Out, five useful programs for disk users.

Monitor Disk 13.

Includes: Demon, the Baron's demon has escaped and it's after you! Pageflipper, Basic and source code listings for page flipping techniques. Cracking the Code, Basic and source code listings for player/missile movement. Data compression techniques for adventure writers. Pengo, excellent Basic version of the well known Penguin game.

Monitor Disk 14.

Includes: Deathzone, superb action game in which you must kill the alien pods and escape from the Deathzone. Cracking the Code, display list program in Basic and source code. Adventure Column, excellent sentence analyser program. Motorway, novel fruit machine simulation but with a motorway theme.

Monitor Disk 15.

Includes: Whist, the well known classic card game. Cracking the Code, modify display lists in Graphics modes 9, 10 or 11. Programs to enable you to Enter commands directly. Turbo Basic, superb program gives enhanced basic for XL/XE machines. Guitar chord tutor, teach yourself to play the guitar.

Monitor Disk 16.

Includes: Minotaur, a comprehensive machine code monitor program, plus source coding. Split screen display for adventurers. Character set copying program. Plotting in Mode Zero demos. Using XIO demos.

## BACK ISSUES

Previous issues of Monitor are obtainable from the club for £1 plus 30p postage each. They contain many interesting and informative articles, hints and tips, program listings, reviews and practical advice. If you have missed out send for your copies of back issues today!! Please note that issues 1,2,3,4,5,6 & 7 are already sold out.

Number 8.

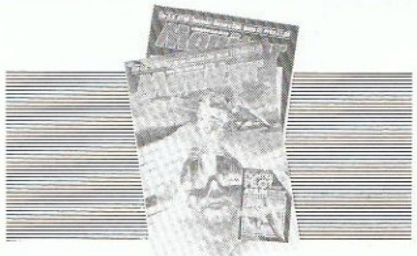
Includes: Cracking the Code. 2 new series; Opening Out and Starting from Basics. Horizontal and vertical scrolling. Mask of the Sun, Sorcerer, Conan, Alley Cat, Ghostbusters and Spy vs Spy all reviewed. Programs include Quickplot, Nightmare Reflections and Matchbox.

Number 9.

Includes: RAMdisk for the 130XE. Appraisal of the 130XE. Introduction to MIDI. Keyo typing checker. Binary loads from Basic. Reviews of TopDOS, Homeword and Mr. DO! Overview of FORTH. Programs include: Fast shape filler, Multiboot index and Happy Typer.

Number 10.

Includes: All about digitised pictures. How disk files work. Cracking the Code, Starting from Basics and What's MIDI all continue. Programs include: Disk Jacket,



PCB Paranoia and 3D Maze. American Road Race, Kennedy Approach, Asylum, Red Moon and Wishbringer reviewed.

Number 11.

Includes: RAM Talker for 400/800. Book reviews. MIDI programs. ST Hi-res Hat program. Hexadecimal Code generator. Reviews of Atariwriter Plus, Sidewinder, Koronis Rift, Electraglide, Mercenary, Fighter Pilot, Goonies and Alternate Reality. Plus Starting from Basics and Cracking the Code.

Number 12.

Includes: Add-on circuits for various motors. Disk file handling. Matrices and Arrays explained. Write your own adventure. Space Invaders program. Reviews of Technicolour Dream, Eidolon and Action Biker. ST reviews include DB Master One, Time Bandit and Menu Plus.

Number 13.

Includes: Omnimon and Ultimon compared. Data compression. Megamax C and Lattice C evaluated. Temper the sound of your 8 bit. Players and missiles explained. Programs include Graphics 8 page flipper, Demon adventure game. Reviews of Super 3D Plotter II, Planetarium, Price of Magik, Last V8 and Nuclear Nick. ST reviews include Cornerman, Cards and Major Motion.

Number 14.

Includes: Display Lists. Adventurers sentence analyser. In depth look at Happy Revision 7. Graphics Modes. Video digitiser mods for use with XL/XE machines. Deathzone, a superb arcade game. Reviews of Crystal Raider, Molecule Man, Domain of the Undead, Laser Hawk, Rick Hanson, Colleen Music Compendium and Spellbreaker. ST reviews include Music Studio, Starglider, TrimBase, Electronic Pool, Easy Record and Pinball Factory.

Number 15.

Includes: Player/missile priorities and interrupts. Turbo Basic commands and functions. 1050 write switch project. Enter commands directly in Basic. Whist card game for you to type in. DOS modifications. OS Controller Card evaluated. Reviews of Spitfire 40, Crumble's Crisis, Robot Knights and Replay. Intro to ST programming. ST Blitter. Reviews of Hollywood Hijinx, BCPL, K-Resource, Make, Micro-time Clock Card, Alternative, Trivia Challenge and Fast Basic.

# TECHNOLOGY SO ADVANCED ...

**MEMORY**
* 512Kbytes RAM (520ST-M, FM)
* 1024Kbytes RAM (1040ST-F)
* 192Kbytes ROM
* 128Kbytes external plug-in ROM option

**ARCHITECTURE**
* *Motorola 68000 Central Processing Unit (CPU) with a clock speed of 8MHz*
* *16-bit external data bus*
* *32-bit internal data bus*
* *24-bit address bus*
* *8×32-bit data & address registers*
* *7 levels of interrupts*
* *56 instructions*
* *14 addressing modes*
* *5 data types*
* *DMA (Direct Memory Access)*
* *real time clock as standard*

**GRAPHICS**
* *full bit-mapped display*
* *palette of 512 colours*

**Using Atari Monitors** *(on 520 & 1040):*
* *640×400 high resolution - monochrome*
* *640×200 medium resolution - 4 colours*
* *320×200 low resolution - 16 colours*
* *80 column text display (40 col low res)*

**Using Domestic TV** *(on 520):*
* *640×200 medium resolution - 4 colours*
* *320×200 low resolution - 16 colours*
* *40 columns × 25 line text display*

**SOUND AND MUSIC**
* *3 programmable sound channels*
* *frequency programmable 30Hz - 125KHz*
* *programmable volume*
* *wave & dynamic envelope shaping*
* *programmable attack, decay, sustain, release*
* *Musical Instrument Digital Interface (MIDI)*
* *MIDI allows connection of synthesisers etc.*

**MOUSE**
* *high precision*
* *2 button control*
* *free with 520ST-FM/1040ST-F*
* *non slip ball motion sensor*
* *removable ball for easy cleaning*

**STANDARD SOFTWARE**
* *GEM desktop + TOS operating system*
* *ST BASIC interpreter/language system*

**INPUT/OUTPUT**

| | |
|---|---|
| *MIDI out* | *(5 pin DIN) 31.25K baud* |
| *MIDI in* | *(5 pin DIN) 31.25K baud* |
| *audio out* | *1.0V DC peak to peak, 10K ohm* |
| *audio in* | *1.0V DC peak to peak, 10K ohm* |
| *RGB monitor* | *1.0V DC, 75 ohm* |
| *mono monitor* | *1.0V DC, 75 ohm* |
| *mono horizontal scan rate* | *35.7KHz* |
| *mono vertical scan rate* | *71.2KHz* |
| *sync* | *5V DC (active low) 3.3K ohm* |
| *modem/serial* | *RS232C, 50 to 19,200 baud* |
| *floppy disk* | *250 Kbits/s* |
| *hard disk* | *11.3 Mbits/s* |
| *mouse* | *standard Atari connector* |
| *joystick* | *standard Atari connector* |
| *cartridge port* | *128K capacity* |
| *RF output (520ST-FM)* | *for TV use* |

**OPERATING SYSTEM**
* *TOS with GEM environment in ROM*
* *hierarchical file structure with sub-directories and path names*
* *user interface via GEM, with self explanatory command functions*
* *multiple windows + icons*
* *window resizing, re-positioning and erasing*
* *drop down menus (selected by mouse)*
* *GEM virtual device interface*

**COMMUNICATIONS**
* *RS-232C serial modem port*
* *8-bit parallel printer port*
* *MIDI port (also for networking use)*
* *VT52 terminal emulation*

**KEYBOARD**
* *standard QWERTY typewriter format*
* *95 full stroke keys*
* *10 function keys*
* *18 key numeric keypad + cursor keys*
* *variable auto-repeat & key click response*
* *keyboard processor reduces CPU overhead*

## ... IT'S AFFORDABLE

## ONLY FROM SILICA

Finally, there's a personal computer that not only solves problems like other computers, but also solves the one problem that other computers have created. Affordability. Silica Shop are pleased to present the ST range of personal/business computers from Atari. The ST was designed utilizing the most recent breakthroughs in semiconductor technology, producing a personal computer that performs tasks with fewer parts. Which means it costs less to make. And less to buy. The latest ST computers now include built in power supplies and built in disk drives. The TOS operating system and GEM window environment are now on ROM chips which are already installed in the ST keyboard. This enables automatic instant booting when you switch on. Silica Shop are pleased to offer the complete Atari ST range. Our mail order department is situated in Sidcup and we have 4 retail outlets at Sidcup, Orpington, Lion House (Tottenham Court Rd) and Selfridges (Oxford Street). We have eight years experience of Atari products, longer than any other UK company, and are well established as the UK's No.1 Atari specialist. With a group turnover of over £5 million and in excess of 70 staff, we provide unbeatable service and support. We provide several facilities which you will find invaluable during your Atari computing life and most of these facilities are available **ONLY FROM SILICA**. We suggest that you read through what we have to offer, before you decide where to purchase your Atari ST.

### FREE STARTER KIT - *Only From Silica*

When you purchase any Atari ST keyboard, you will not only receive the best value for money computer on the market, but you will also receive the following from Atari Corporation as part of the package:

** BASIC Language Disk * BASIC Manual * ST Owners Manual * TOS/GEM on ROM*

If you buy your ST from Silica Shop, you will also receive:

** NEOchrome - colour graphics program * 1st Word - Word Processor*

In addition, we at Silica would like to see you get off to a flying start with your new computer, so we have put together a special **ST STARTER KIT** worth over £100, which we are giving away **FREE OF CHARGE** with every ST computer purchased at our normal retail prices. This kit is available **ONLY FROM SILICA** and is aimed at providing users with a valuable introduction to the world of computing. We are continually upgrading the ST Starter Kit, which contains public domain and other licensed software, as well as books, magazines and accessories all relevant to ST computing. Return the coupon below for full details.

### DEDICATED SERVICING - *Only From Silica*

At Silica Shop, we have a dedicated service department of five full time Atari trained technical staff. This team is totally dedicated to servicing Atari computer products. Their accumulated knowledge, skill and experience makes them second to none in their field. You can be sure that any work carried out by them is of the highest standard. A standard of servicing which we believe you will find **ONLY FROM SILICA**. In addition to providing full servicing facilities for Atari ST computers (both in and out of warranty), our team is also able to offer memory and modulator upgrades to ST computers.

**1Mb RAM UPGRADE:** Our upgrade on the standard Atari 520ST-M or 520ST-FM keyboard will increase the memory from 512K to a massive 1024K. It has a full 1 year warranty and is available from Silica at an additional retail price of only £86.96 (+VAT = £100).

**TV MODULATOR UPGRADE:** Silica can upgrade the 1040ST-F to include a TV modulator so that you can then use it with your TV set. This is an internal upgrade and does not involve any untidy external boxes. A cable to connect your ST to any domestic TV is included in the price of the upgrade which is only £49 (inc VAT). The upgrade is also available for early 520ST computers at the same price.

### THE FULL STOCK RANGE - *Only From Silica*

We aim to keep stocks of all Atari related products and our warehouse carries a stock of £½ million. We import many software titles direct from the USA and you will find that we have new releases in advance of many of our competitors. Unlike dealers who may only stock selected titles, we have the full range. In addition, we carry a complete line of all books which have been written about the Atari ST. A range as wide as ours is something you will find is available **ONLY FROM SILICA.**

### AFTER SALES SUPPORT - *Only From Silica*

Rest assured that when you buy your ST from Silica Shop, you will be fully supported. Our free mailings give news of releases and developments. This will help to keep you fully up to date with what's happening in the Atari market. And in addition, our sales staff are at the end of a telephone line to service all of your Atari requirements. If you purchase an ST from Silica and would like any programming or technical advice, we have a full time technical support team to help you get the best from your computer. Because we have both the staff and the systems specifically dedicated to providing after sales service on Atari ST computers, we are confident that our users enjoy an exceptionally high level of support. This can be received **ONLY FROM SILICA**.

### FREE CATALOGUES - *Only From Silica*

At Silica Shop, we recognise that serious users require an in-depth information service, which is why we mail free newsletters and price lists to our ST owners. These are up to 48 pages long and are crammed with technical details as well as special offers and product descriptions. If you have already purchased an ST and would like to have your name added to our mailing list, please complete the coupon & return it to us. The information service is available **ONLY FROM SILICA**.

### FREE OVERNIGHT DELIVERY - *From Silica*

Most orders are processed through our computer within 24 hours of receiving them. Most hardware orders are sent by the overnight METRO courier service FREE OF CHARGE to customers within the UK. This method helps to ensure minimum delay and maximum protection.

### PRICE MATCH - *Only From Silica*

We hope that the combination of our low prices, FREE UK delivery service, FREE Starter Kit and FREE after sales support, will be enough to make you buy your Atari equipment from Silica Shop. If however, there is something you wish to purchase, and you find one of our competitors offering it at a lower price, then please contact our sales department, providing us with our competitor's name, address and telephone number. If our competitor has the goods in stock, we will normally match the offer (on a 'same product - same price' basis) and still provide you with our normal free delivery. We realise that we are not the only company who will match a competitor's price. However, if you come to us for a price match, you will also be entitled to our after sales service, including free newsletters and technical support. This makes our price match promise rather special, something you will receive **ONLY FROM SILICA**. We don't want you to go anywhere else for your Atari products. So shop at Silica, the UK's No1 Atari Specialist.

**SIDCUP** (& Mail Order) .......... **01-309 1111**
*1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX*

**LONDON** .......... **01-580 4839**
*Lion House (1st floor), 227 Tottenham Court Rd, London, W1*

**LONDON** .......... **01-629 1234 ext 3677**
*Selfridges (1st floor), Oxford Street, London, W1A 1AB*

## 520ST-M

The affordability of Atari computers is reflected in the price of the 520ST-M keyboard, which is a mere £259 (inc VAT). This version of the ST comes with 512K RAM, as well as a modulator and lead for direct connection to any domestic TV. The price does not include a mouse. In addition, when you buy your 520ST-M from Silica, you will also receive the FREE Silica 'ST Starter Kit'. During 1987, many software houses will be producing games software on ROM cartridges, which will plug directly into the cartridge slot on the 520ST-M keyboard, giving instant loading without the expense of purchasing a disk drive. With the enormous power of the ST, you can expect some excellent titles to be produced, making this the ultimate games machine! If your requirement is for a terminal, then the 520ST-M can fulfill this role too. Leads are available to connect the ST to a variety of monitors, and with the imminent introduction of terminal software on ROM cartridge, the ST provides a low price terminal for business use. If you wish to take advantage of the massive range of disk software available for the ST range, you will need to purchase a disk drive. Atari have two floppy disk drives available, a ½ Mbyte model £149 and a 1Mbyte model £199. Full details of these drives, as well as the Atari 20Mbyte hard disk are available on request. If required at a later date, the mouse may be purchased separately.

**£259**

## 520ST-FM

The 520ST-FM with 512K RAM and free mouse, represents a further breakthrough by Atari Corporation in the world of high power, low cost personal computing. This model is the latest addition to the ST family, and is not only powerful, but compact. It is priced at only £399 (inc VAT) a level which brings it within the reach of a whole new generation of computer enthusiasts. When purchased from us, it comes with the FREE Silica 'ST Starter Kit' see paragraph on the left. To make the 520ST-FM ready for use straight away, Atari have built into the keyboard a ½ megabyte disk drive for information storage and retrieval, allowing you easy access to the massive range of disk based software which is available for the ST. This new computer comes with all the correct cables and connections you will need to plug it straight into any standard domestic television set. You do not therefore have to purchase an Atari monitor. If you do require a monitor however, these are available with the 520ST in the following money saving packages:

| | | |
|---|---|---|
| *520ST-FM Keyboard* | *Without Monitor* | *- **£399** (inc VAT)* |
| *520ST-FM Keyboard* | *+ High res mono monitor* | *- **£499** (inc VAT)* |
| *520ST-FM Keyboard* | *+ Low res colour monitor* | *- **£599** (inc VAT)* |
| *520ST-FM Keyboard* | *+ Med res colour monitor* | *- **£699** (inc VAT)* |

Because the 520ST-FM has its own power transformer built into the keyboard, there are no messy external adaptors to clutter up your desk space. You are left with only one mains lead, serving both the disk drive and the computer. You couldn't ask for a more stylish and compact unit.

**£399**

## 1040ST-F

For the businessman and the more serious home user, Atari have their most powerful model, the 1040ST-F with 1028K RAM. This low cost powerhouse can be introduced into a business environment as a stand-alone system, or can support a mainframe computer as a terminal. The 1040ST-F not only features twice as much memory as the 520ST-FM, but also includes a more powerful built-in disk drive. The drive featured on the 1040ST-F is a one megabyte double sided model. The extra memory facility of the 1040ST-F makes it ideal for applications such as large databases or spreadsheets. Like the 520ST-FM, the 1040ST-F has a mains transformer built into the console to give a compact and stylish unit with only one mains lead. The 1040ST-F is also supplied from Silica Shop with a free software package and 'ST STARTER KIT'. In the USA, the 1040ST-F has been sold with a TV modulator like the 520ST-FM. However, for the UK market, Atari are manufacturing the 1040ST-F solely with business use in mind and it does not currently include an RF modulator, this means that you cannot use it with a domestic TV (Silica Shop do offer a modulator upgrade for only £49). The 1040ST-F keyboard costs only £599 (inc VAT) and, unless a modulator upgrade is fitted, will require an Atari or third party monitor. There are three Atari monitors available and the prices for the 1040 with these monitors are as follows:

| | | |
|---|---|---|
| *1040ST-F Keyboard* | *Without Monitor* | *- **£599** (inc VAT)* |
| *1040ST-F Keyboard* | *+ High res mono monitor* | *- **£699** (inc VAT)* |
| *1040ST-F Keyboard* | *+ Low res col monitor* | *- **£799** (inc VAT)* |
| *1040ST-F Keyboard* | *+ Med res col monitor* | *- **£899** (inc VAT)* |

The 1040ST-F comes with a mouse controller and includes 1Mbyte of RAM. It has a 1Mbyte double sided disk drive and mains transformer, both built into the keyboard to give a compact and stylish unit, with only one mains lead.

**£599**

## ATARI ST

To: Silica Shop Ltd, Dept ATCOC 0387, 1-4 The Mews, Hatherley Road, Sidcup, Kent, DA14 4DX

**PLEASE SEND ME FREE LITERATURE ON THE ATARI ST**

**Mr/Mrs/Ms:** .......... **Initials:** .......... **Surname:** ..........

**Address:** ..........

..........

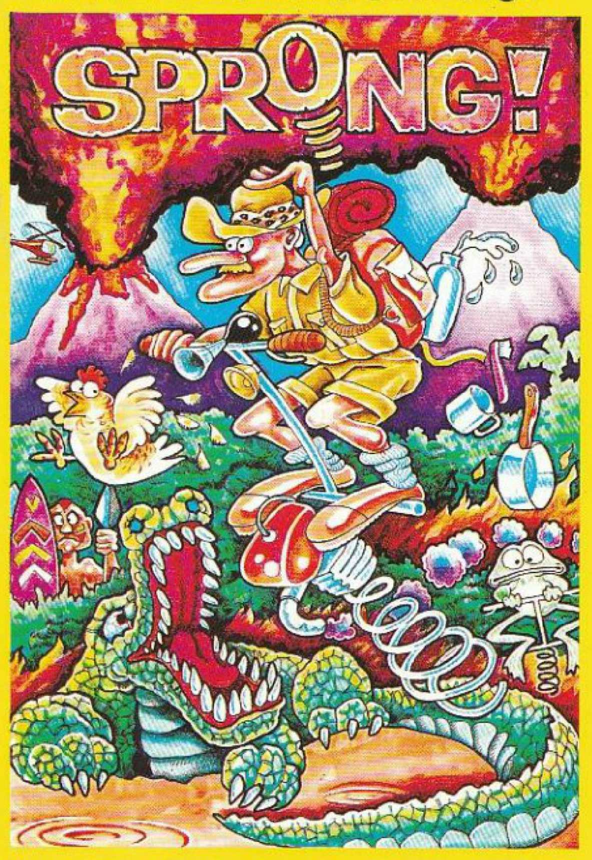
..........

.......... **Postcode:** ..........

**Do you already own a computer**
**If so, which one do you own?** ..........

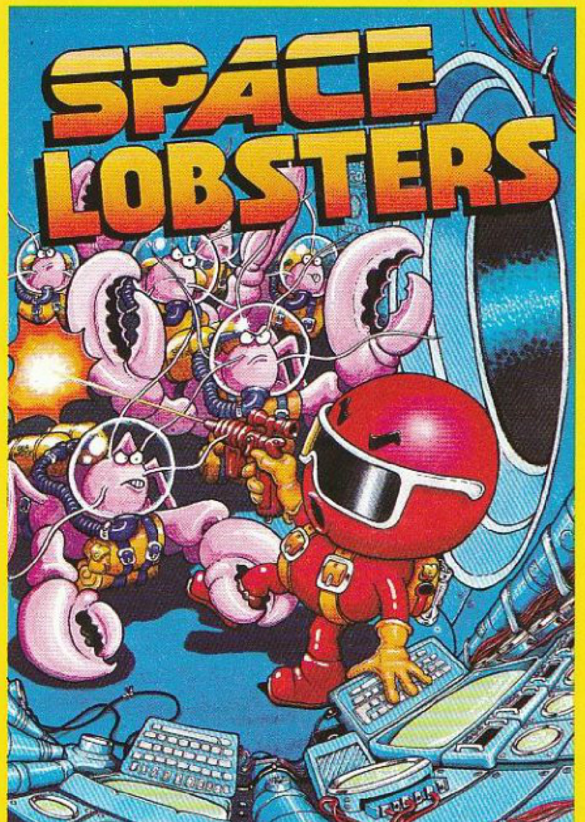
RedRat

*...where the playing gets tough !*



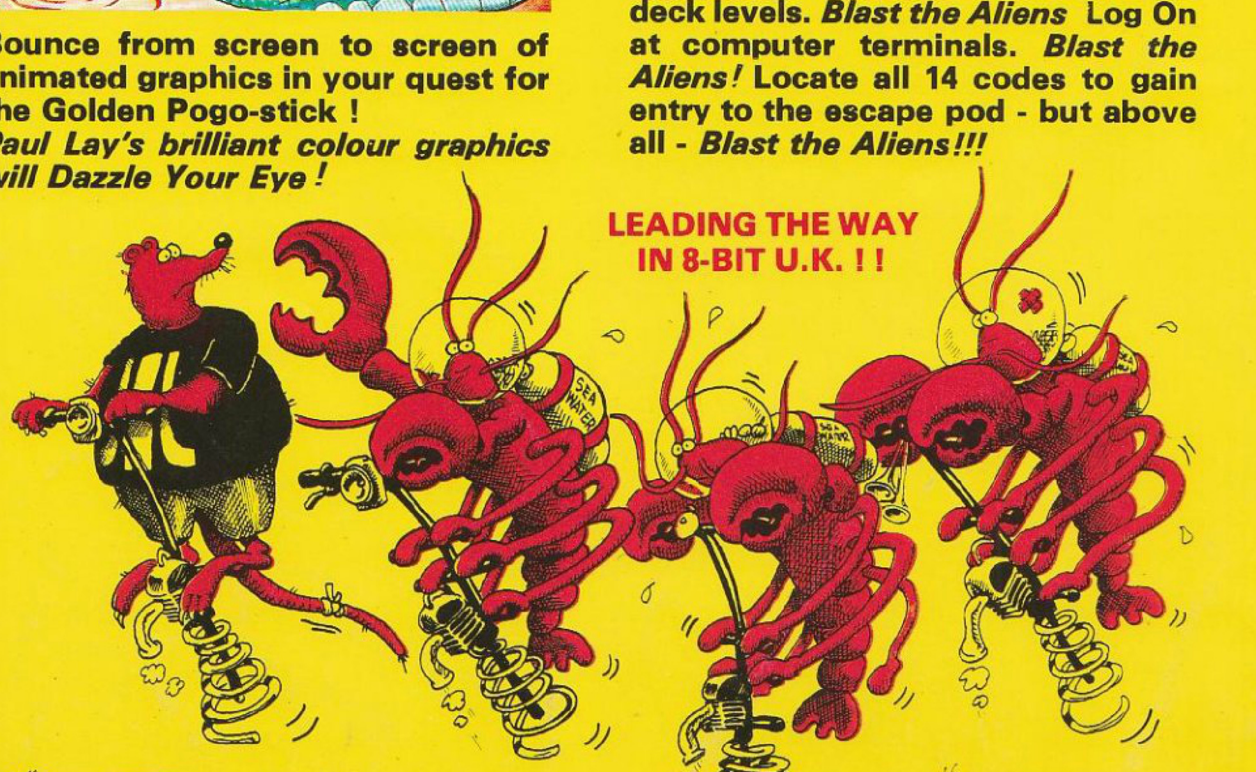
**Bounce from screen to screen of animated graphics in your quest for the Golden Pogo-stick !**
***Paul Lay's brilliant colour graphics will Dazzle Your Eye !***

**Atari XL/XE. Tape £7.95 Disk £9.95**



**Scour the Colossus space ship on your search for access codes. *Blast the Aliens!* Teleport to different deck levels. *Blast the Aliens* Log On at computer terminals. *Blast the Aliens!* Locate all 14 codes to gain entry to the escape pod - but above all - *Blast the Aliens!!!***

**LEADING THE WAY IN 8-BIT U.K. ! !**



**Red Rat Software, 11 Fennel Street, Manchester M4 3DU.**