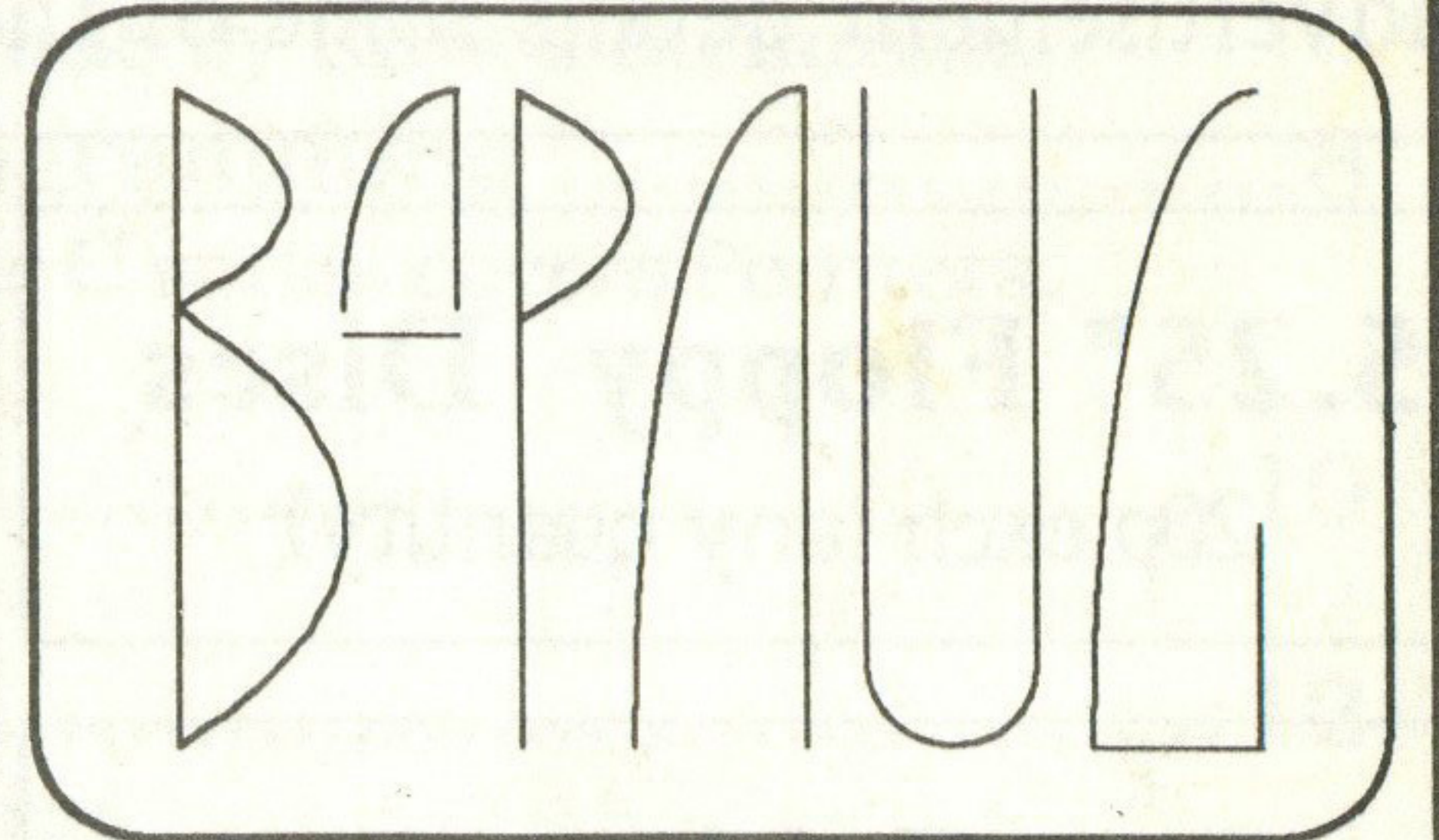
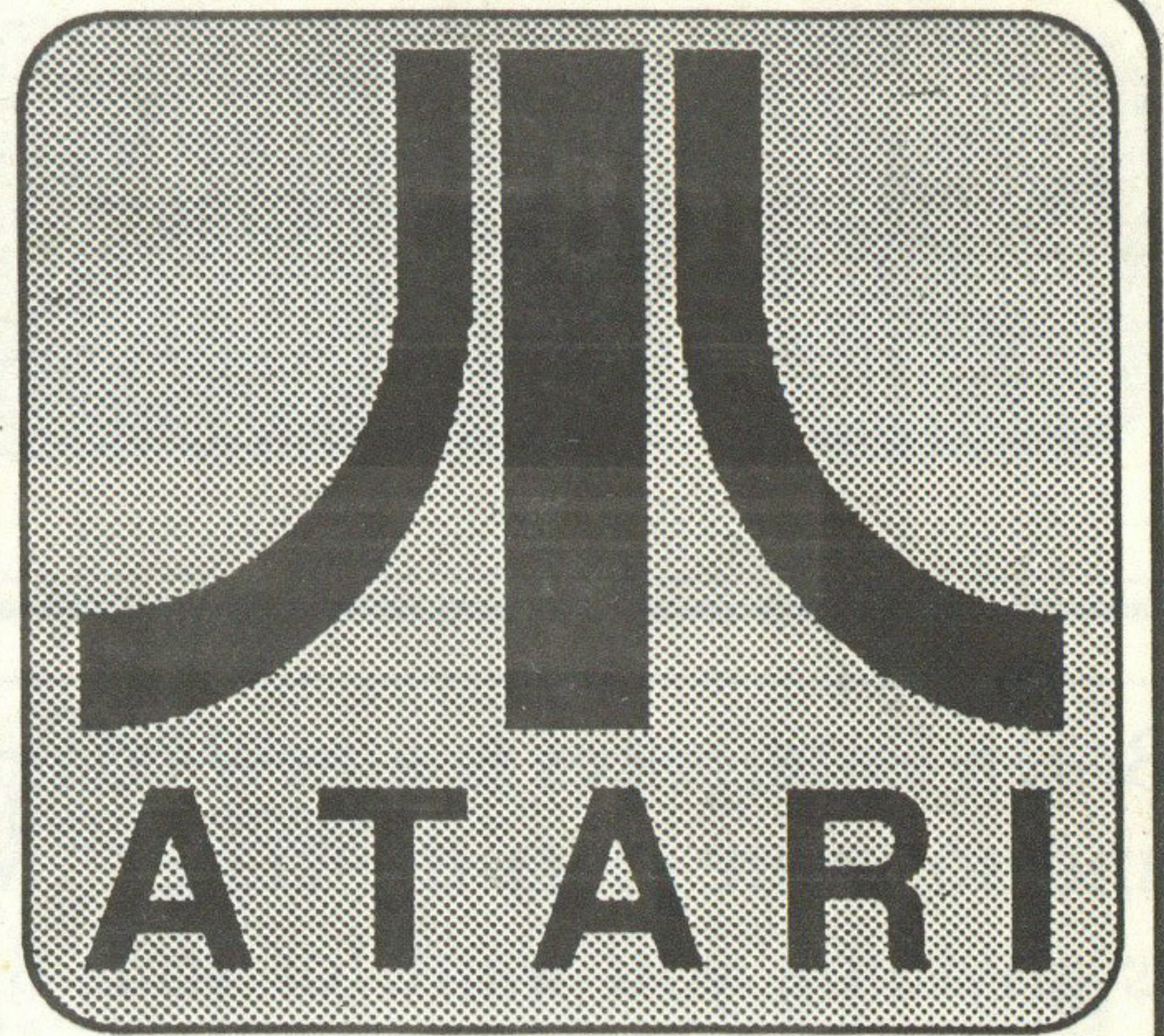


The Alternative Atari Newsletter



95p

September/October 1990 Issue 10

PROJECTS

**32,768 COLOURS FOR THE ST
GUMBY**

MIDIMASTER SIO DRIVER

REGULAR FEATURES

DEAR 8:16

NOTICE BOARD

SOFTWARE ROUNDUPS

USER GROUP FILE

REVIEWS

DARK CHAMBERS

DIAMOND GOS

DIAMOND PAINT

GATES OF ZENDOCON

PLASTRON

PRO SYSTEM 7800

ADVERTISEMENT

BREUN ENTERPRISES

DEAN GARRAGHTY

FRONTIER SOFTWARE

GRALIN INTERNATIONAL

SHARWARD SERVICES

TUTORIALS

ATARIWRITER PLUS

C LANGUAGE

TURBO 816

BRUEN ENTERPRISES LTD.

47 Mandeville Road, Southgate, London N14 7NJ

Please telephone **0831 166 429**

Bruen Enterprises is a new company supporting the 8 bit Atari. We are importing quality hardware and software from the states. We will provide any product or service for the 8 bit Atari. If there is anything you want for your 8 bit then please telephone us and we will try to get it for you. This advertisement details some of the products we can already supply.

5.25" Floppy Discs

20p each (any quantity)

Cables

850 to printer

850 to modem

2 metres long & only £8 each

Memory Upgrades

We will upgrade your computer for £60. XLs to 256K and XEs to 320K.

We can also supply 320K XEs for £120.

PD Software

* Cassette & disk based *

General

We buy, sell and part exchange all 8 bit Atari hardware. We have 1050 disk drives and 850 interfaces in stock. We can also supply parts and do repairs.

Cheques and postal orders should be made payable to **Bruen Enterprises Ltd.**

For further information on any of these products telephone 0831 166 429.

The Black Box

Only £200 for so much. It gives your 8 bit a **printer spooler**, **RS232 interface** with built in handler that takes no user RAM, **hard drive interface** and a button that dumps the contents of the screen to your printer when it is pressed.

There is also an add on that allows you to connect up to 4 standard high capacity drive mechanisms to your Black Box. The 4 drives are **super archiver compatible** with a capacity of upto 720K per disk. There is also software that allows the Black Box floppy interface to read and write to IBM compatible disks. You can have 3.5" or 5.25" floppies connected.

The Black Box needs no hardware modification. It just plugs into the parallel bus on your XL/XE.

Black Box and 20M Hard Drive

For £500 we can supply a 20M byte hard drive system including the Black Box (see above for details).

80M Hard Drive

* Please telephone for details *

XF551 Drive Modification

For only £10 we can supply a drive modification for the XF551 that allows you to connect a 3.5" drive mechanism and switch between it and the original 5.25" or just use the XF551 as a 3.5" drive. The modification is compatible with all DOSs (also fixes a few XF bugs) and supports high speed skew.

Turbo 816

£120 for a **16 bit replacement CPU** for the Atari 8 bit. It works with all existing hardware and software. It provides new instructions and addressing modes and allows you to connect upto **8MBytes** of directly addressable memory to your XL/XE (not bank switched). This is a great upgrade for machine code programmers.

Editorial

Colin Hunt
Graham Broomfield
Thomas Holzer

Team

D&E Controller
Graphics Controller
VCS Controller

All contributors are credited with their articles entry within the contents list.

8:16 is produced by BaPAUG Publishing.

All original articles published within 8:16 are the copyright of the author, but can be used by other publications provided the author and 8:16 are acknowledged as the original source and a copy of the publication is sent to 8:16. Articles re-printed from other newsletters should be credited to the original source.

8:16 and the BaPAUG cannot be held liable for any errors or claims made by advertisers.

Letters and articles should be sent to:

8:16
c/o 248 Wimborne Road
Oakdale, Poole, Dorset BH15 3EF

This issue of 8:16 was produced using an Atari 130XE, 1050 disk drive, 520STFM, SM124 monitor, Cumana 1M external drive, VIDI ST, Star LC24-10, Apple Macintosh SE and Laserwriter Plus. The software used was TariTalk and AtariWriter Plus on the 8 bit. Timeworks Desktop Publisher Version 1.12, 1st Word Plus, Gem Kermit and TariTalk on the ST. Microsoft Word, MacPaint, MacDraw, Mac Kermit and Aldus Pagemaker Version 3.5 on the Macintosh. More ST produced pages than ever before!

The opinions expressed within 8:16 are those of the authors and are not necessarily held by the BaPAUG.

The BaPAUG is a non profit making organisation.

Aaron, 'I can see ya!'

Thomas, thanks for The Halley Project: A Mission In Our Solar System, its great.

Copy date for the next issue of 8:16
is 15th October 1990.

Issue date is 19th November 1990

Glen expresses his regret that there is no SIGnews section due to the pressure of other work.



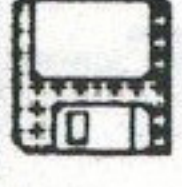
Contents

Page 3
8:16
Issue 10

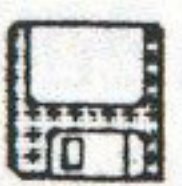
Features

	The Atari Pro System 7800.....	5
	The Rolls Royce of video gaming? Reviewed by Thomas Holzer.	
	The Gumby Upgrade	6
	A stereo POKEY upgrade by Chuck Steinman.	
	Introduction To C Programming	8
	The third article in our C series covers Pointers. By David Watson.	
	The Baby Theory	10
	By David Small.	
	Diamond GOS	12
	Reviewed by Colin Hunt.	
	Diamond Paint	14
	Reviewed by Colin Hunt.	
	8 bit Reviews: Plastron	15
	Reviewed by Ian Brooker.	
	Turbo-Info	16
	By Chuck Steinman.	
	32,768 Colour Support for the Atari ST	18
	By Barry Orlando.	
	VCS Game Reviews	21
	Dark Chamber and Double Dragon reviewed by Thomas Holzer.	
	AtariWriter Plus Explained for People Who Don't Want to Read the Book	24
	By Jimmy Boyce.	
	Lynx Review	27
	Gates Of Zendocon reviewed by Colin Hunt.	
	Midimaster SIO Driver	30
	By Graham Broomfield	

Departments

	Notice Board	4
	BaPAUG News	11
	7800 Software Roundup	21
	VCS Software Roundup	21
	Dear 8:16,	22
	XL/XE Software Roundup	26
	BaPAUG Stock Clearance	26
	BaPAUG 8 bit PD Library	28
	BaPAUG ST PD Library	29
	User Group File	31

 8 Bit Articles

ST Articles 

**The Bournemouth and Poole Atari User Group is a member of
The Association of Atari User Groups.**

Notice Board

Harlequin Release Schedule - 1990

The following details the 1990 release schedule for Harlequin's Atari 8 bit products (all Cassette £7.99 / Disk £9.99):

Plastron (Available now)

Fast moving, horizontally scrolling game in which the player pilots "scavenger one" on its mission to recover as much fossil fuel from the heavily guarded Omni Corporation planet - Plastron as possible. Dodging both ground and airborne attackers, the player must negotiate a maze of craters and traps, carefully conserving his retro boosts to allow him to leap to safety at crucial moments. Recover the allocated amount from a given zone and the land tank recovery unit races in and transports you to the next, more deadly zone.

Zero War (1st July)

All around the solar system the automated food manufacturing stations pump out the life resource of the galaxy. No human is involved in the process, the deep space stations are fully automated, until now. A strange virus has spread from satellite to satellite, the food supply has stopped, the peoples of the Galaxy have been brought to their knees. You streak through space, your course the first of the rogue space stations. The automated docking procedure lets you inside. Strapping yourself into your jet pack you proceed into the dimly lit interior.

Zero war is a first person, 3D shoot 'em up and puzzle game. You race against a constantly diminishing energy supply to locate the transference crystals and restart the food making process. Hordes of Droids and automated security checks stand between you and saving the lives of the peoples of the solar system.

Project Xanthein (1st August)

Cast adrift aboard the Cryogenic settler ship Xanthein, your engines gone and the Earth recovery ship 3 months away, you must protect the 64 sleeping inhabitants from the attacking aliens keen to dine on your precious cargo. You control the automated sanitation droid, the only functioning unit on the ship, in a desperate attempt to ward off the attackers and rendezvous with the Earth patrol vessel.

Other releases from Harlequin include the **Psygnosis** games **Menace** and **Shadow of the Beast** and two new titles called **Demon Wars** (the ultimate space adventure) and **Dino-Wars**.

Harlequin (The Entertainments) Ltd.,
76 Lincoln Way, Canvey Island, Essex SS8 9SL

Frontier Software - Product Prices

Frontier Software (0423-567140/530577) have recently sent us a complete list of their products along with their current prices - including the new lower prices for the Xtra-RAM range of memory upgrades. The current retail prices (including VAT) are:

Xtra-RAM ST Memory Expansion		Xtra-RAM STE Memory Expansion	
Unpopulated	£64.99	0.5Mb	£69.99
Populated to 0.5Mb	£79.99	2Mb	£169.99
Populated to 2Mb	£189.99	4Mb	£339.98
ST & STE Clock Cartridge		Printer Buffer (available October 1990)	
Forget-Me-Clock II	£24.99	Printer-Q 128K	£114.99
		Printer-Q 256K	£126.49
		Printer-Q 512K	£149.49
Disk Duplicator		Printer-Q 1Mb	
Mass Disk Duplicator	£249.99		£195.49

Disk Supplies

Here are the latest disk prices from Barnston Computer Supplies:

	5¼" DS/DD	
	48tpi	96tpi
10 off	£3.95	£4.45
25 off	£9.10	£10.25
50 off	£16.90	£19.40

100 capacity lockable disk storage boxes are also available at £7.95ea if purchased with disks. All prices include VAT and UK P&P.

Barnston Computer Supplies
16 Barnston Green, Barnston, Great Dunmow,
Essex CM6 1PH
Telephone: (0371) 874234

New UK ICD Distributor

Gralin International, a recently formed company specializing in the supply of hard to find Atari 8 bit hardware and software has recently become ICD's new UK distributors. Along with the change of distributor there is also a change in the pricing structure resulting in better value for money for the end users. Under the new agreement the new prices (including VAT) are:

Action with Toolkit	£39.99
MAC/65 with Toolkit	£39.99
Printer Connection	£29.99
P:R: Connection	£49.99
R-Time 8	£29.99
Rambo XL (without RAM)	£19.99
SpartaDOS X	£39.99
US Doubler	£14.99
US Doubler & SpartaDOS Construction Set	£29.99

Gralin International
11 Shillito Rd, Parkstone, Poole, Dorset BH12 2BN

8:16 Subscription Rates Annual (4 issues)

U.K.	£3.80
Europe	£8.00
Elsewhere (sea)	£8.00
Elsewhere (air)	£15.00

8:16 Commercial Advertisement Rates

Full page	£40.00
Half page	£20.00
Quarter page	£10.00

Phone (0202) 677895 for details.

The Atari Pro System 7800

Reviewed by
Thomas Holzer

The Rolls Royce of video gaming?

Read on and find out.

The year is 1984 and video games are going down the tube (literally), *CBS-Coleco* are stopping production, *Intellivision* are bust, *Vectrex* is long gone and *Warner Brothers / Atari Inc.* announces "the video game is dead, long live the video game". What am I talking? I'm talking about the **Pro System 7800**, released in Europe (including the UK) last year. That's right, originally released in 1984, the 7800 was Warner Brothers last attempt to rescue Atari, which was at the time, a rotten apple within a profit making giant company. I think every Atari user knows the story about Atari at that time so, I will not dwell on history.

Still, there it was and here it is now: the 7800, only six years too late. Have Atari missed the bus again? Well, not quite! It sold, not enough to save Atari, but it sold so well in the states that it is now the second most bought machine there, with only *Nintendo* beating it.

Lets now take a look at the machine itself, is it worth £70? Read on and find out.

The 7800 looks like a cross between a 2600 (VCS) junior and a slimmed down 5200 (another Atari curiosity). The joystick ports are in the front and the machine itself comes with two cheap horrible looking joypads (can't get used to them, terrible things). On top of the 7800 you'll find the power on/off button with a red light indicator, a handy pause button and the reset and select switches. At the back

you'll still find the A/B difficulty switches as well.

The 7800 cartridge slot is 100% identical to the 2600 slot and is fully compatible with all existing VCS games. This being the machines strongest selling point and its not just a sales hype - I've tried all of mine (around 200) and even games like *Space Shuttle* where you need to use the VCS B/W switch, work with no problem. That makes the 7800 the games machine with the biggest cartridge software library available.

So what? I hear you say. If I own a VCS, why should I buy the 7800 to play the same games?

Marie can shift any number of objects of any size in all possible directions . . .

Well, back in 1984, Atari designed a new graphics chip. This 48 pin IC, which was called **Marie**, includes 24,000 transistors and will blow the mind of any Atari 8 bit owner. Player Missile graphics, like in the XL/XE, is no more - Marie can shift any number of objects of any size in all possible directions on the screen - some Lady, eh! Take for example the 7800 version of *Robotron* where you have 70 robots running all over your TV screen and without any flicker. How about *Joust*, where the riders consist of 10 different colours, or the new version of *Asteroids* (build in, boots up when turning on without a cartridge plugged in)

which has new colours and pseudo 3D vision. *Ms. PAC MAN*, written by Bally exclusively for the 7800 is totally arcade quality and *Ballblazer* and *Rescue on Fractalas* (the first co-operation between Atari and Lucasfilm) beat the 8 bit version by miles. Of course, you can also get versions of *Galaga*, *Xevious*, *Food Fight*, not to forget, *Pole Position II* and just released for the XL/XE, *Desert Falcon*, Atari's first true 7800 release.

Less interesting (and most annoying -Ed.) is the fact that Atari re-designed the system and closed the slot on the side of the machine, where, in the states you can plug in a keyboard and the whole XL/XE range of add-ons like disk drives, printers or whatever, making it fully 8 bit compatible!

Don't forget, this all happened way back in time, 1984, so measured against today's machines (PC-Engine, 16 bit Sega) the Pro System is a bit dated. No doubt about that, but before you think of buying a VCS 2600, invest a bit more and buy the 7800 instead, or if you already own the VCS, chuck it away and buy the 7800, you'll never regret it.

Subscription Due?

If the number on the envelope after your subscription number (top right hand corner of label) is /10 your subscription is due.

Subscription details can be found on the Notice Board opposite.

The Gumby Upgrade

A STEREO POKEY UPGRADE *by Chuck Steinman*

This is upgrade version #1. It will only provide stereo output for software written specifically for this upgrade. It will not produce stereo output for existing software. Also, the keyclick will NOT be fed into the stereo outputs, as it does not originate from POKEY.

Parts Needed

- 1) Second Atari POKEY Audio Chip
- 2) 74LS14 / 74HCT14 Inverter
- 3) 1000 Ohm, 1/4 Watt Metal Film 2-5% Tolerance
- 4) Two RCA style phono jacks
- 5) Two 0.01 (or close) 16V (or more) bypass capacitors.
- 6) Two short (6-12" each) sections of shielded audio cable.
- 7) A standard dual RCA to RCA patch cable.
- 8) Optionally two 50k single turn trimmer pots.

Installing The Inverter

- a) Bend up all pins of the inverter except for pins 7 and 14.
- b) Cut off the narrow part of the pins which were bent up.
- c) Install the inverter over top of the existing 74LS14 (or another 74LS part if more convenient) on the motherboard. Make sure that the new chip has its locator notch / dot on the same end as the chip below it.
- d) Solder pins 7 and 14 of the inverter to the same pins of the lower IC.
- e) Run a small wire (wire wrap type works best) from pin-1 of the inverter to pin-13 of the CPU. The CPU is part number CO14806 on the XL/XE series.
- f) On the old POKEY there is a 3k pullup resistor connected between pin-31 and Vcc. You will need to un-solder this resistor and remove it.
- g) Run a small wire from pin-2 of the inverter to pin-3 of the inverter, and then on to POKEY pin-31. Note, you can use the

Disclaimer

Installation of this modification will void any warranty you may have on your computer. Chuck Steinman, DataQue Software, or GENie telecommunication service cannot be held responsible for the installation of this upgrade or any incidental or consequential damage to any equipment or persons using this upgrade or any variation of it. In other words you are on your own. This statement also applies to members of the BaPAUG and the editorial team for 8:16.

pad where the resistor was just removed. Be sure to get the correct one.

Installing POKEY

- a) Bend up all pins on the new POKEY which are marked with a minus on the diagram (see Figure 2). This includes POKEY pins: 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, and 29.
- b) Cut off the narrow part of each pin bent up.
- c) Tin each lead which was NOT bent up. This includes pins 1, 2, 3, 4, 5, 6, 7, 17, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39 and 40. These pins are marked in the diagram as *, >>, or <<.
- d) Now, bend up the pins indicated by the >> and << symbols. Do NOT cut these pins short.
- e) Place the new POKEY on top of the old POKEY in a piggy-back style.
- f) Solder the unbent pins of the new POKEY to the old POKEY. If your original POKEY was in a socket, then it is easier to connect the two if it is removed. Make sure no excess solder flows down the pins to the narrow part of the OLD POKEY. Re-insert both POKEYs into the original socket.
- g) Solder the 1k resistor from pin 37 to Vcc. The most convenient location to pick up Vcc is where the 3k resistor was removed earlier.
- h) Solder a wire from pin-31 of the new POKEY to pin-4 of the inverter.
- i) Mount the two RCA jacks on the rear of the case, preferably in an area close to the POKEYs.
- j) Solder a bypass capacitor to each of the centre conductors of the RCA jacks.
- *k) With the trim-pot knob facing you, pin 1 should be to the left side. Solder a wire from this pin on each trimmer, to a ground trace on the motherboard.
- *l) Connect the free end of the bypass capacitor to the centre pin of the trimmer (one capacitor to each trimmer).

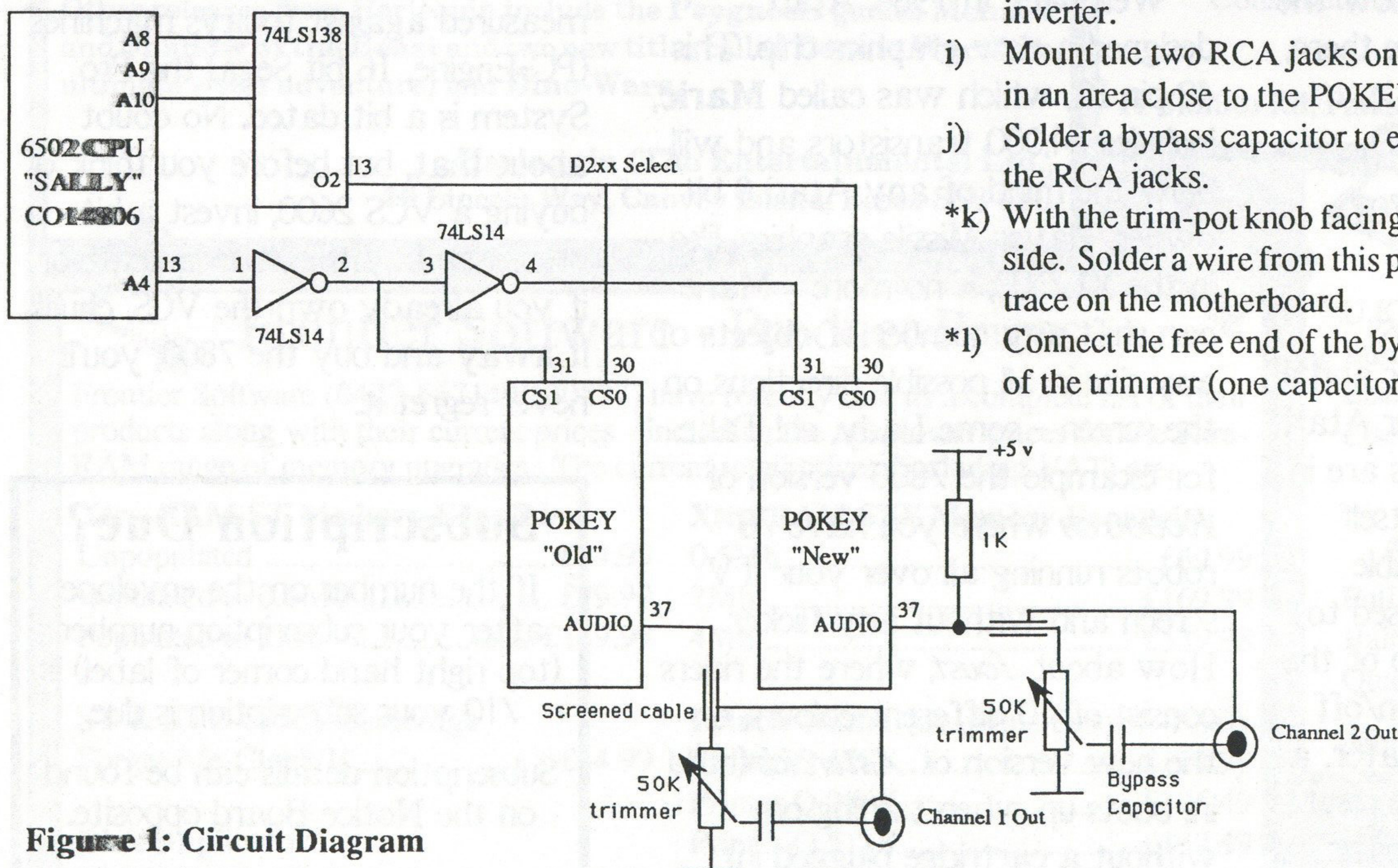


Figure 1: Circuit Diagram

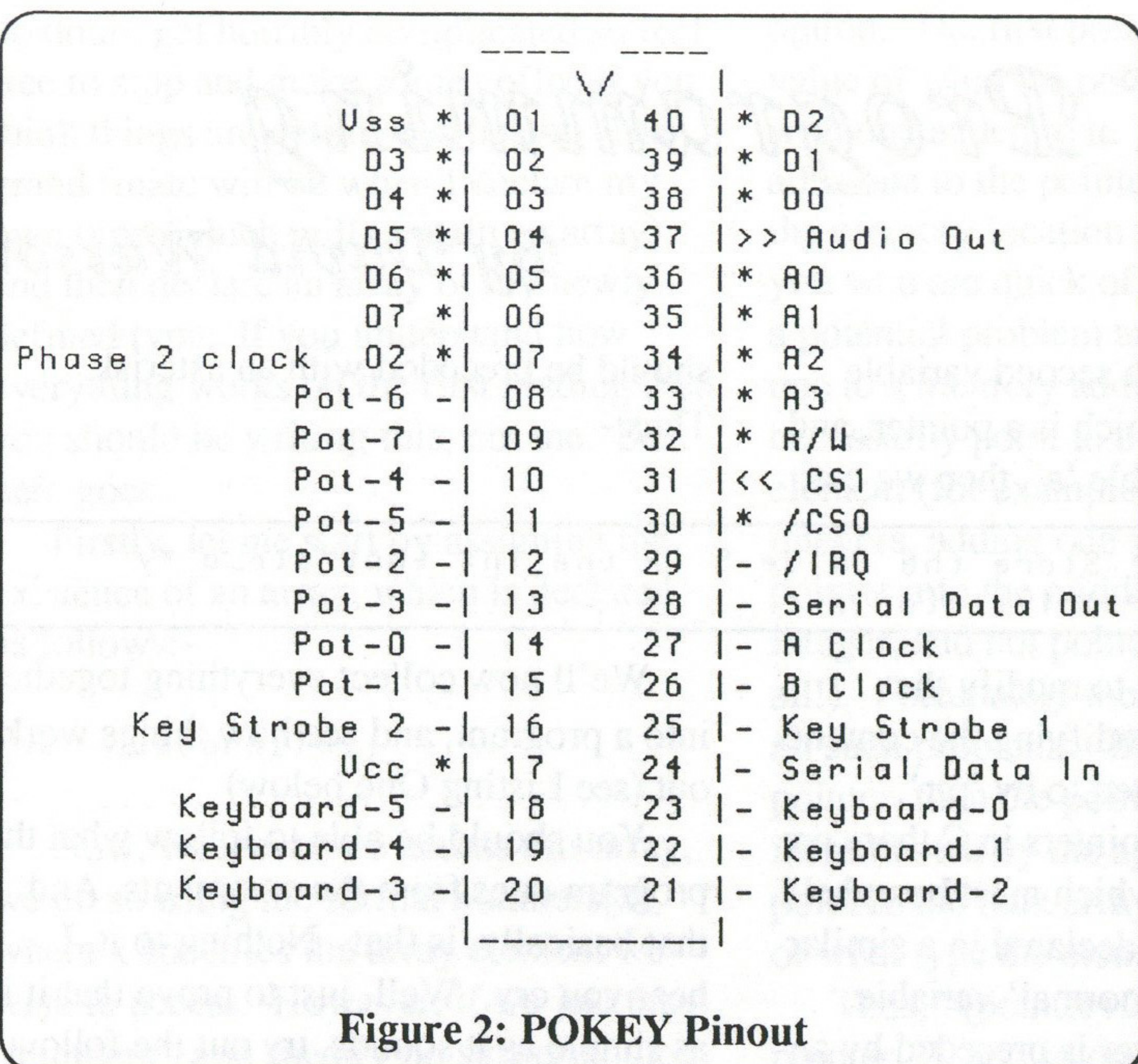


Figure 2: POKEY Pinout

- *m) Connect the shields of the audio cables to the provided solder lugs on each RCA connector, and the centre conductor of the free terminal of each trimmer.
 - n) Connect the centre conductor of the free end of the audio cable which is connected to the left RCA jack/trimmer/cap to pin-37 of the OLD POKEY.
 - p) Connect the centre conductor of the free end of the audio cable which is connected to the right RCA jack/trimmer/cap to pin-37 of the NEW POKEY.
 - q) The shield of the audio cable on the POKEY end, should be cut and taped (or heat shrunk) so that it does not touch anything.
 - r) Run a 18-20 AWG wire from the ground lug of the RCA jacks to the wide ground area on the motherboard. This normally makes contact with the shield box that covers the motherboard.
 - s) You will now be able to connect the two RCA cables to an AUX (or Tape) level input of a stereo or boom box.
 - *t) I would suggest centring the trimmers in their travel, and adjusting them as needed to get the best clarity. You may want to glue the trimmers to the back of the cabinet to keep them from moving around.
- * On my system the POKEY outputs worked fine without the trimmers. So I just connected the bypass capacitor on each

Location	Name	R/W	Function
\$D200	AUDF1	W	Audio Channel #1 Frequency (Divide F/n)
\$D201	AUDC1	W	Audio Channel #1 Control (Vol/Distort)
\$D202	AUDF2	W	Audio Channel #2 Frequency (Divide F/n)
\$D203	AUDC2	W	Audio Channel #2 Control (Vol/Distort)
\$D204	AUDF3	W	Audio Channel #3 Frequency (Divide F/n)
\$D205	AUDC3	W	Audio Channel #3 Control (Vol/Distort)
\$D206	AUDF4	W	Audio Channel #4 Frequency (Divide F/n)
\$D207	AUDC4	W	Audio Channel #4 Control (Vol/Distort)
\$D208	AUDCT1	W	Audio control for channels 1-4
\$D20F	SKCTL1	W	Serial Port Control
\$D210	AUDF5	W	Audio Channel #5 Frequency (Divide F/n)
\$D211	AUDC5	W	Audio Channel #5 Control (Vol/Distort)
\$D212	AUDF6	W	Audio Channel #6 Frequency (Divide F/n)
\$D213	AUDC6	W	Audio Channel #6 Control Vol/Distort)
\$D214	AUDF7	W	Audio Channel #7 Frequency (Divide F/n)
\$D215	AUDC7	W	Audio Channel #7 Control (Vol/Distort)
\$D216	AUDF8	W	Audio Channel #8 Frequency (Divide F/n)
\$D217	AUDC8	W	Audio Channel #8 Control (Vol/Distort)
\$D218	AUDCT2	W	Audio control for channels 5-8
\$D21F	SKCTL2	W	Serial Port Control

Figure 3: POKEY Registers

RCA jack to the appropriate audio cable center conductor. This was driving an AUX 350mV input of a Pioneer SPEC-1 preamplifier.

POKEY Registers.

I will only elaborate on registers used to produce sound in the stereo upgrade. All registers which were in the original POKEY will appear in the second POKEY 16 bytes higher in memory. The extra UART, key scanner, and pot scanner could be used for all kinds of neat projects.

Each Audio Channel Frequency Register is an 8-bit value which is a divisor of the primary frequency.

Each Audio Channel Control Register Controls the Volume and Distortion of each channel. The bits are assigned as follows:

- 76543210
- vvvv Volume control bits.
- Range controls volume as follows:
- 0000 lowest volume level
- ∖
- 1111 highest volume level
- s--- Volume only bit.
- Directly controls audio output:
- 0 Speaker output is off
- 1 Speaker output is on
- ddd----- Distortion code.
- Code is assigned as follows:
- 000 5-bit/17-bit poly noise
- 001 5-bit poly noise
- 010 5-bit/4-bit poly noise
- 011 5-bit poly noise
- 100 17-bit poly noise
- 101 pure tone
- 110 4-bit poly noise
- 111 pure tone

The AUDCTn register controls all channels. There are several functions assigned to this register as follows:

- bit-7 Makes 17-bit poly into 9-bit poly counter
- bit-6 Clock Channel-1 with 1.79 MHz (CPU rate)
- bit-5 Clock Channel-3 with 1.79 MHz (CPU rate)
- Bit-4 Join channel 1 and 2 to form 16-bit range
- Bit-3 Join channel 3 and 4 to form 16-bit range
- Bit-2 Insert filter in channel-1, clocked by channel-2
- Bit-1 Insert filter in channel-2, clocked by channel-4
- Bit-0 Switch clock base from 64kHz to 15kHz

The SKCTLn register controls various functions of the POKEY device, and only has to be initialized to a value of three to assure all four channels of POKEY are active.

To detect if the upgrade is installed, look at the 8 extra pots, they will be all zero. Also the key code register will be also zero. If you compare the keycode at \$D209 with that of \$D219, and \$D219 is zero, the upgrade is installed. You may want to mask IRQ's during the test for safety.

Let me know what wild things you all come up with for this upgrade.

Thanks, Chuck
(GENie User: DataQue.1)
Presented 12/16/1989

This article was down-loaded from The City BBS.

Introduction to C Programming

By David Watson

Pointers in C

The assembly language programmer is almost unique in that assembly language is virtually the only language which does not support data 'typing' in any way. If an assembly language programmer has stored information in a section of memory then he (or she) is free to treat it in any way - there is nothing to stop an assembly programmer from treating a string of characters as a series of numbers to be added, other than the fact that the result would not be of much use.

The high level language programmer - such as a PASCAL programmer - has the

```
int    *ptr1; /* Pointer to an integer */
float  *ptr2; /* Pointer to a floating point number */
char   *ptr3; /* Pointer to a character */
```

opposite problem however. There is virtually no way in standard PASCAL that a programmer can convert between a string and, say, a floating point number other than by some cunning programming or by using functions which may be non-standard.

The C programmer has the best of both worlds. Having defined an array of characters there is almost nothing to stop the programmer from storing integers in the array (although this can hardly be recommended). And with a small amount of knowledge of how the compiler handles different data types, the programmer can produce code which not only makes the most of the machine, but also makes good use of the compiler and of the C programming language. Perhaps the only real snag is that misusing data types is risky both in terms of introducing bugs into the program and in terms of being difficult to understand at a later date when the program needs modified. So although the potential to manipulate data types in any way you want is there, it should be used with care.

What almost everything hinges on are pointers. A pointer is a variable which 'points' to information which is stored in memory, usually by storing the memory address of the information (although things get a bit complicated when using mainframes). Given the address of the information, the machine can then go ahead and manipulate it as required. For example, we could have a variable (called 'a', say) which holds an

integer. If we have a second variable (called 'ptr', say) which is a pointer, and it points to the variable 'a', then we can

```
*ptr_a = 5; /* Store the value 5 in the int that ptr_a */
           /* points to */
```

instruct the machine to modify the contents of 'a' by modifying the contents of the memory pointed to by 'ptr'.

In order to use pointers in C there are a few simple rules which must be upheld. Firstly, pointers are declared in a similar way to declaring a 'normal' variable, except that the pointer is preceded by an asterisk. Thus:-

You can also declare your own types and declare pointers to them if you wish.

The second rule comes when initialising a pointer. Clearly if we wish to make our pointer point to a variable we must have some way of obtaining a value for the pointer. If we precede a variable by an ampersand (&) then the value obtained is the address of the variable in memory, which is just what we want. Thus:-

```
ptr_a = &val_a; /* ptr_a becomes a pointer to variable val_a */
```

Note that ptr_a must be a pointer to something the same type as val_a. We cannot make 'ptr_a' a pointer to a float, then initialise it using the above statement if 'val_a' is an integer. (Well actually you can, but more about that later. We'll keep things simple just now).

Finally, if you wish to access the contents of the pointer (as we did above when we initialised it to point to a variable) then we just put the variable name. If you wish to access whatever the pointer points to then the pointer's name

should be preceded with an asterisk.

Thus:-

We'll now collect everything together into a program, and see how things work out (see Listing One below)

You should be able to follow what the program does from the comments. And that basically, is that. Nothing to it, I hear you cry. Well, just to prove that it is as simple as it sounds, try out the following exercise before going on. Study the above example, make brief notes if you wish, then put this article to one side and write a program to do the following:-

You will have three integer variables (called a, b and c), and three pointers to integers (called x, y and z, which point to a, b and c respectively). Prompt the user for values of a and b (use the scanf function call), and then add the two integers and store the result in c. Finally, display the contents of c (using the printf function call) to make sure that the result is correct. The tricky bit is that you should perform the arithmetic through the

pointers, and not using a, b or c at all. Indeed, the three integers should only appear when initialising pointers or in scanf/printf function calls. Good luck.

Needless to say, things do get more complicated. Just to start the ball rolling, I will now talk about arrays in more depth. A direct consequence of this is pointer arithmetic. From there I will digress slightly with a brief discussion on how to reserve portions of memory, before moving onto the really fun stuff - arrays of pointers and so on. Things will

```
main()
{
    int  a;          /* Define 'a' as an integer */
    int  *ptr;      /* Define 'ptr' as a pointer to an integer */

    a = 5;          /* Assign '5' to the value of 'a' */
    ptr = &a;       /* Assign to 'ptr' the address of 'a' */
    printf("%d\n",a); /* Now display what is in 'a' */
    *ptr = 6;       /* Assign '6' to whatever 'ptr' points to */
    printf("%d\n",a); /* Finally, display what is in 'a'... */
    scanf("\n");    /* ...and wait for the user before finishing */
}
```

Listing One:

no doubt get horribly complicated so feel free to stop and make some coffee if you think things are getting too heavy. The grand finale will be when I declare my own type (which will contain an array), and then declare an array of my newly defined type. If you understand how everything works on the first reading then you should be writing this, not me. So here goes.

Firstly, let me start by assuming the existence of an array, which is declared as follows:-

```
int numbers[10];
```

Now, whenever we access this array, we do so using the format numbers[x], where x specifies the array element we wish to access. However, if we just take 'numbers', and specify no subscript then we effectively have a pointer to the first element of the array. Thus, '*numbers' is equivalent to 'numbers[0]'. Clearly it would be a good idea if we could address the second and subsequent array elements in a similar manner. Fortunately we can,

option. The first possibility returns the value of what the pointer was pointing to with one added to it. The second option adds one to the pointer, and then access the memory location pointed to. Those of you who are quick off the mark could see a potential problem arising in that adding one to a memory address would not necessarily point to the next array element (for example, in an array of integers, adding one would put the pointer into the middle of the first integer, and not point it to the second at all!). Fortunately the compiler handles all such problems. If you add one to a pointer, then the pointer will be incremented by the appropriate amount to point to the next array element, regardless of what type the element is.

Thus, *(pointer+2) is equivalent to pointer[2], and so on. Moving on again, suppose we wish to write a function which copies one string (i.e. character array) into another. To simplify this we will assume that the destination is always large enough to hold the string being copied. See Listing Two.

...which we can shorten to..

```
copy_str(s,d)
char *s, *d;
{
    *d = *s;
    while( *s != \0 ){
        *++d = *++s;
    }
}
```

...which can in turn be shortened to...

```
copy_str(s,d)
char *s, *d;
{
    while( *d++ = *s++ )
        ;
}
```

Quite impressive, isn't it? And to copy array_1 to array_2, where both are arrays of characters, all you need to do is use the call:-

```
copy_str(array_1, array_2);
```

In this case the arrays are being passed as pointers which are then subsequently modified within the function. It is quite safe to change parameters which are passed to a function, as the function works on it's own private copy of the variables. It is not possible to return parameters passed other than by using the 'return()' statement, which has the obvious restriction that only one parameter can be returned. The way round this is to pass pointers to the variables we wish to change, and modify them by accessing them via the pointers. To demonstrate this, consider the function in Listing Three in which two integers passed are added, and stored in a result which is accessed via a pointer.

To add variables x and y, and store the result in z, the following call is used:-

```
adding(x, y, &z);
```

Note that we do not pass z, but rather the address of z. This particular quirk offers many opportunities to forget the ampersand, or to forget to declare the pointer as a pointer (and hence inadvertently call it an integer), all of which cause run-time errors. There's no easy way to avoid these errors, unfortunately. How-



```
COPY OVER A CHARACTER
WHILE( WE'VE NOT REACHED THE END OF THE SOURCE STRING )
BEGIN
COPY OVER A CHARACTER
POINTER TO SOURCE STRING POINTS TO NEXT ELEMENT
POINTER TO DESTINATION STRING POINTS TO NEXT ELEMENT
END

copy_str(s,d)
char *s, *d; /* This is equivalent to 'char s[], d[]' */
{
    *d = *s;
    while( *s != \0 ){
        *d = *s;
        d = d + 1;
        s = s + 1;
    }
}
```

Listing Two:

with a bit of pointer arithmetic.

The term pointer arithmetic is slightly misleading, though. It is illegal to add, subtract, multiply or divide two pointers, as the results obtained are meaningless. However, we can add (or subtract, although strictly speaking this is also illegal) an offset to a pointer. This allows us to access the second and subsequent elements. The next problem arises as soon as we attempt to write the 'add offset to a pointer' instruction; there are two possibilities, show below. Which do you think is the correct one?

- a) *pointer + 1
- b) *(pointer + 1)

The correct answer is the second

Fortunately we can abbreviate this a bit with some of C's notational quirks.

Thus:-

```
copy_str(s,d)
char *s, *d;
{
    *d = *s;
    while( *s != \0 ){
        *d = *s;
        d++;
        s++;
    }
}
```

```
adding(a, b, result)
int a, b; /* The integers to add */
int *result; /* Pointer to the result */
{
    *result = a + b;
}
```

Listing Three:

ever, if your program crashes with two or three bombs then you should certainly look out for pointers going astray as they are a very common cause of this kind of error.

The next paragraph provides a brief explanation of how functions are implemented at the assembly level. If you don't understand 68000 assembly language, then skip it.

Whenever a C function call is made, the compiler usually provides code which surrounds the main function body. This code handles parameters which are passed to the function. What happens is that enough room for all parameters passed is reserved on the stack using a stack frame, A6 usually being the register used to keep track of the frame. Whenever these variables are accessed within the function, the copy of the variable in the stack frame are modified, not the actual variable at the calling level. As soon as the function ends, the variables in the stack frame are lost when it is unlinked. It is therefore necessary to access variables via pointers if lasting changes to their values are to be made.

However, one very important point. Just because `*(array+i)` is equivalent to `array[i]`, you should not be fooled into thinking that the following are equivalent:-

```
int array[5];
int *array;
```

The first reserves space for an array of five integers, the second only reserves space for a pointer to an integer. This is the kind of thing that will usually trip you up when you've stayed up too late and decided to try and write a program; be warned.

Multi-dimensional arrays are different from normal 1D arrays. Whereas a one dimensional array is simply a pointer to a portion of memory, multi-dimensional arrays are actually arrays of pointers which point to areas within the portion of vacant memory. Let us imagine that we have a 3 by 3 character array called `data[3][3]`. (Note that we write the subscripts as `[3][3]`, and not as `[3, 3]` which is more conventional. The reason for this is that in C a 2 dimensional array is represented as a 1D array, each of whose elements is an array). This will require nine bytes of memory, one for each character (plus, of course, memory for any pointers to the array). Then, if the array starts at location 'START', `data[0] = START`, `data[1] = START + 3` and `data[2] = START + 6`. Well, that's

the theory. In practise it is not a good idea to refer to other elements in the array by assuming that the memory is reserved in this way. The compiler writers may have decided to do things differently. Even if you can get away with this on your compiler, you may not be able to with another compiler. And besides, it's sloppy.

The practical upshot of all this is that if you want to pass a multi-dimensional array to a function, you'll need to do more than pass the array name. It is also necessary to pass the array of pointers which is being maintained by the compiler to keep track of where each 'row' in the array starts. So if we have the following array definitions, then we would pass them in the following manner:-

<u>Definition</u>	<u>Pass As</u>
<code>array[2]</code>	<code>array[]</code>
<code>array[2][4]</code>	<code>array[][4]</code>

As I've noted earlier, a 1D array name is actually a pointer, and we can safely pass `array[2]` as either `array[]` or `*array`. We can carry this further with 2D arrays; the following are interchangeable:-

```
array[][4] = (*array)[4]
```

The parenthesis around the second expression is necessary as square brackets have a higher precedence than '*'.
*'

Well, that's enough to be going on with. You should find that pointers give you a lot of power, but at a price. When your pointers start going astray you'll find that the machine crashes when you try to access memory and get bus errors or address errors, data in a completely unrelated part of your code mysteriously corrupts or you overwrite program memory. Life can become a nightmare for the user of RAMdisks. All that I can say at this point is 'Good Luck'!

In the next article I will discuss C's environment in more depth. Some of the topics covered will be how to pass parameters to programs (i.e. .TTP programs), how to link machine code modules into C programs and how to control the creation of programs which exist in several files of source code.

David Watson,
22 Lydbury,
Bullbrook,
Bracknell,
Berks.
RG12 3HH

The Baby Theory *by Dave Small*

Reprinted from Current Notes, March 1990

David Small is one half of Gadget By Small and is responsible for their Spectre range of Macintosh emulators available on the ST. Below is his "Baby Theory" as presented in his column "Small World" within the March 1990 issue of Current Notes.

It goes like this. If it takes nine months to make a baby, what makes you think anyone can get a product out any more quickly?

Sure, *sometimes* they do. But. Then they send the real documentation, the real release disk, and so forth later. Those count; it still ends up being nine months.

No matter what, it seems to take (at least) nine months. It's as though there was a universal constant that it takes that long to make something.

So why complain, lets say, about Gadgets taking too long to get the 030 accelerator out? Or *PC ditto II*? Or *Pagestream*, or or or ...

I also happen to know that it takes nine months to get a book out to the stores from the time the manuscript is accepted. Other professions accept this; why not us?

Related: Baby version 1.0 is full of bugs, diapers are needed for "patches" and so forth. Can't talk; the communication section doesn't work. Can't walk; it "crashes" if it tries. It requires years and years of user effort to get to Baby version 5.0, where all these problems are fixed, about five years later. In the meantime, tech-support (the doctor for serious bugs, mom for everyday ones) has to help out.

So why be hard on software / hardware companies that simply can't do better than four billion years of evolution can? Is that fair?

Naturally, I say this coming up on my fifth year of emulating Macs on an ST, and where the only bugs left are pretty obscure.

BaPAUG News

What's been happening at the BaPAUG monthly meetings?

In response to this popular question, Paul Brookes, takes a look at what's been going on at group meetings in recent months. Hopefully future issues of 8:16 will have similar reports from other group members for future meetings.

June: MIDI Show

This event was open to the general public as well as club members and approximately 50 people attended the evening. There was a good response to our adverts which appeared in the Bournemouth Advertiser and on 2CR local radio, producing around 40 new faces at the meeting. Nearly half of these people had STs and also owned keyboards, with about 10 people indicating that they would be buying some sort of MIDI expander box in the near future.

It was the intention of the evening to give a brief introduction to the MIDI specification and provide an insight into the various software packages available on the 8-bit machines and the ST. In addition some budget keyboards were made available by staff from Eddie Moors Music for those considering purchases to use with their ST's. I would like to thank Eddie Moors Music for their support during this event.

The evening went well, despite a couple of unfortunate events. The first of these was that yours truly lost his diagrams which were to have been the basis for a discussion on the MIDI interface. The second was that the KAWAI K1 synth decided to go into pure distortion mode half way through a demonstration, negating a weeks worth of preparation, but FM melody maker behaved, and a MIDI Special Interest Group was formed by Graham Jones.

On the 8 bit side Graham Broomfield demonstrated MIDIMaster from 2 Bit Systems / Page 6 and explained why he felt it necessary to re-write the MIDIMaster software, details of which can be found elsewhere in this issue of 8:16.

The final hour was spent as a general discussion hour, with individuals trying out the keyboards and asking questions about the software. In particular Score Writing software was in demand. Gary and Vince from Eddie Moors Music brought along Mastertracks which accepts PRO-12/24 song files amongst others.

July: SIGs Night

This meeting was dedicated to status reports from the three Special Interest Groups: MIDI; Hardware and Software. There was no report from the MIDI group, which has been organised by Graham Jones, due to his absence at the meeting. The Hardware group is being led by Neil Horder and myself for the ST, and Colin Hunt on the 8-bits. At the meeting, Neil's highly modified ST was on display and he explained his ideas behind designing a high resolution colour video digitizer. Due to the interest shown in the Hardware SIG and the clear indication that most people wanted to learn some electronics it was proposed that the club should design a standard interface system for both 8-bit (XL/XE only) and ST users, so that all machines can utilise the group's designs. Proposed projects include video digitising,

sound sampling, intelligent MIDI interfacing and general purpose input / output port design. Tutoring in electronics and computer interfacing will commence shortly, and key areas will be covered in forthcoming newsletter articles. The software group is still looking for a leader, so nominations or volunteers please come forward.

Finally, I was appointed to the position of promotions / publicity organiser, so contact me if you want to promote or publicise a club activity!

July Update #1

Colin, Neil and myself recently met to design the BaPAUG interface bus. After some thought we scrapped the idea of a common bus for ST and 8-bit machines, deciding to design the main part of each project jointly, then each sub-group could design the relevant 8-bit or ST interface. This will cut costs of kits if we decide to sell any of the club projects. More information will be available at the next club meeting, or by writing to me now that I'm the publicity organiser! If you write then please say what projects you would like to see and enclose an SAE if you'd like a reply.

Paul Brookes

July Update #2

I couldn't publish this article without highlighting the more specific reasons why the BaPAUG bus was scrapped, one hot summer night at my house! 8 bit users get ready to gloat, but first some history. Way back in February, Neil and I stayed late at that months meeting - we eventually got thrown out of the building and continued our discussion in the car park. The end result of this mammoth conversation was the initial design of the BaPAUG bus. The meeting at my place was taking place in order to finalize the specification and to design the first implementations for both machines. It was during this stage of the evening that it soon became apparent that the ST, yes the ST, couldn't hack it. Due to the lack of a R/W line on the cartridge port the design became over complicated and expensive. At the same time I was becoming annoyed (and blase) because of the special features of the XL/XE parallel bus I couldn't use, in order that the design could be implemented on the ST. Thus the bus was scrapped, but maybe one day it will, like a phoenix from the ashes, raise its head again - probably when the TT is released.

Colin Hunt

Meeting Schedule

September 7th:	Annual General Meeting
October 5th:	Sound Sampling
November 2nd:	SIGs Night
December 7th:	Traditional Games and Chess Challenge

The meeting schedule for 1991 will be decided at this years AGM, so members have until September to think up suitable topics. All suggestions will be put to the vote and the top nine selected, with two SIG nights and the AGM accounting for the other three months. If there are less than nine topics, meetings will be left blank and topics decided upon by the committee.

Diamond GOS

The Graphic Desktop for the Atari 8 Bit.

Reviewed by Colin Hunt

Everyone agrees that the user interface of the nineties is the WIMP environment, as found on 16 bit computers such as the Apple Macintosh and Atari ST. Reeves Software have recently released a product called Diamond GOS that provides this interface for the Atari 8 bit. So how does a 8 bit Graphical Operating System stand up to every day use?

Diamond GOS (Graphical Operating System) consists of a super cartridge which contains the GOS and desktop software, a utility disk which contains various utilities for customising the Diamond environment and two manuals, a user guide and programmers manual.

In order to use this new Operating System you have to configure a boot disk for your particular DOS, be it a DOS 2 variant, SpartaDOS or DOS XE. You also have to set up the required 'mouse' handler, for your input device - a standard ST mouse, joystick or touch tablet. If you already have a ST mouse you can experiment straight away by booting the system with the utility disk, with your ST mouse plugged into the second joystick port.

The Desktop

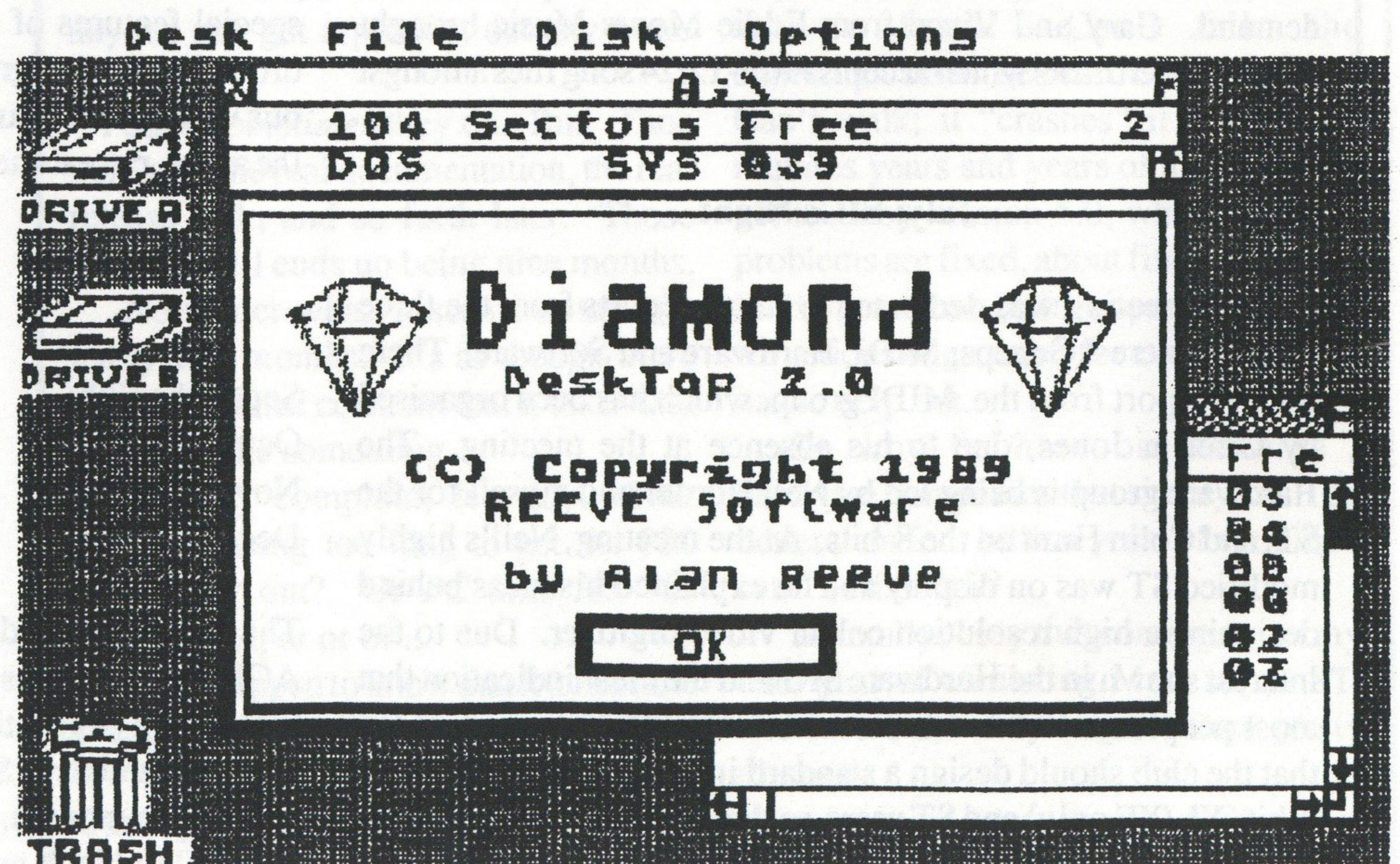
With the system booted you will be presented with a desktop remarkably like the one on the ST. Along the top is a menu bar with 4 options - Desk, File, Disk and Option. In the bottom left hand corner is the Trash Can and above is the disk icon for drive A (which is actually drive D1:). The first complaint I have about the desktop is the use of artifacting, which looks great on my 1200XL running NTSC, but on a UK system running PAL it only produces a murky blue / green vertical stripe. This can easily be resolved by turning the colour of your TV down, which result in a pleasing grey striped background. The second complaint I have is that the trash can does not invert when dragging files for deletion, though

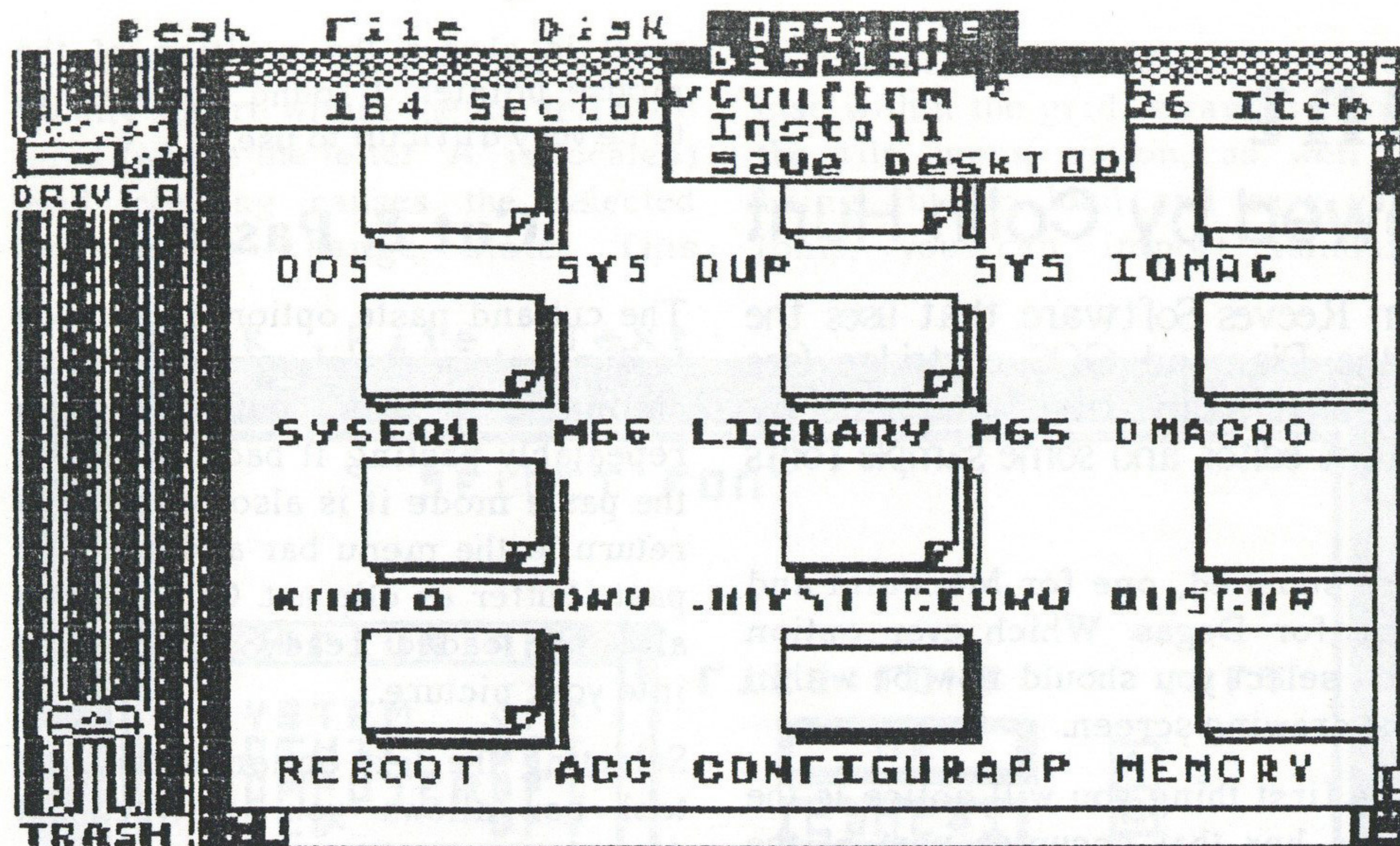
this isn't really important - I'm just use to seeing this happen on my ST.

Diamond GOS is very easy and responsive in use. Whenever the pointer is moved to an entry within the menu bar the resultant menu cleanly appears down the screen. Moving the pointer away causes the menu to close, unlike the ST which requires a mouse click outside of the menu area. When moving the pointer backwards and forwards along the menu bar the drop down menu generation and screen replace speed is very impressive. All menu items that can be selected are displayed in normal text, whereas all the invalid options are show in a form of light text formed by missing out every other 'on' pixel. The menu systems allows you to perform most of the functions found within the menu system of DOS 2.5 (although they may be under different names) and several specific functions required by Diamond - Display, Confirm,

Install and Save Desktop. Confirm forces the computer to ask for confirmation on dangerous activities (disk format, file deletion). Install is used to display (or remove) disk icons from the desktop. Upto 8 drives may be defined labeled from A to G for drives D1: to D8:. If you have to make any changes to the desktop in order to reflect your system, Save Desktop will save the current desktop settings to your boot disk and thus remove the tedious task of doing this each time you start the system. If you use SpartaDOS or DOS XE there is even a menu option that allows you to create directories. Of course, for users of the different variants of DOS 2 this option is in italics - DOS 2 doesn't support sub-directories. One DOS function that I could not find an equivalent for was H: Write DOS files, which means you have to leave the Diamond environment in order to produce any new boot disks.

The disk icons also operate in the same way as the ST with a double click opening a window containing the disk directory. This may be presented in a selection of ways - text or icon, and sorted by name, extender, size or none by using the Display option within the Option menu. The window also displays the drive id, the number of free sectors and the number of items within the directory. You also have all the usual window control features, including sizing, scroll bars and close. With the directory displayed as icons you have to use both the horizontal and





vertical scroll bars in order to move the window over the three column display, unless you use a full window which has the disadvantage of hiding the disk and trash can icons. Whenever you use the scroll bars to move the contents within the window, which you will have to do as only a maximum 19 entries can be display in text mode and 12 when using icons, the update is slow. This occurs as the desktop is constructed on a Graphic 8 screen and therefore the contents of the window have first to be removed before displaying the new.

Re-configuration

In order to use Diamond you have to configure your own boot disk containing your favourite DOS. This is done by following your normal procedure for producing a boot disk; eg format disk followed by writing DOS when using DOS 2.5. You then have to run a BASIC program called DOSCONFIG.BAS that produces the configuration file CONFIG.OS which tells Diamond which DOS you are using. With this configuration Diamond is expecting to see the ST mouse in the second joystick port, a problem if you do not have one, as the only way of changing this is from within Diamond itself. This is done by running an application called CONFIGUR.APP. The manual calls this program MOUSECFG.APP, which probably reflects on the fact that the routine is now more powerful than a simple mouse config file generator. It also allows

you to load in a memory driver which allows Diamond to handle bank switching, with a upper limit of 16Mbytes. There are two drivers available, one for the XL and one for the XE, though as far as I can tell there is no documentation to accompany these drivers.

Like all applications run from Diamond, after the program has loaded you are presented with a new menu bar containing the application specific options. In this case you have two options: Desk and Configure. Desk contains the Help option, while Configure, as expected contains all the configuration options. We now come to another inconsistency. As soon as you load a new mouse driver, say one for the joystick, the new driver takes control thus forcing you to change your input device. When you change the port the mouse is to use, the change does not take effect until you save the new configuration, which also causes the system to re-boot. This also applies to the click rate which controls the delay used to react to successive presses of the mouse driver. It was while using this software that I encountered one of the few bugs. When you try to save a configuration file to a write protected disk the system hangs leaving the mouse active but the menus dead. The only course of action was a re-boot, losing your new configuration.

The Manuals

Diamond GOS is supplied with two manuals. The first, the user's

manual, supplies just enough information to start using the system (once you have read it and re-arrange all the sections into a more useful order). This manual could benefit from a major re-write.

The second manual, the Programmer's Guide, also falls short of requirements. The front cover proudly proclaims support for Atari BASIC, Turbo BASIC, Action and Assembly. The manual contains one section detailing all the functions available within Diamond along with the equates used within two of the MAC/65 library files supplied on the utility disc. Also on the utility disk is a file called DIABASIC.BAS which should be the Atari BASIC library. In fact this program contains the code required within your own program to set up a call to Diamond. The code is not very elegant and includes an exit from a READ loop via a TRAP on no data and a USR call that passes 17 parameters. The only way of finding the location of the Diamond system variables is to examine the MAC/65 library files. If you wish to write applications for Diamond you will have to carefully read these manuals and examine all the library files and examples provided.

Conclusion

Overall I was very impressed with this software. It is well error trapped and survived most of the silly tests like save the desktop when there no disk in drive one. Its major draw backs are the two manuals. Both need to contain more information, especially the programmers guide, if the system is to become the standard user interface of the nineties.

Availability

There is one final area I have yet to address - where can you buy Diamond GOS and how much? Currently, no UK supplier is advertising this product, so until one decides to import it you will have to purchase it directly from Reeves Software. Their address is 27W 150 Old Farm Lane, Warrenville, IL 60555, USA.

Diamond Paint

Reviewed by Colin Hunt

Diamond Paint is the first product from Reeves Software that uses the Diamond Environment based around the Diamond GOS cartridge (see separate review). The package consists of a 12 page manual and a floppy disk containing the paint application, a font editor and some sample fonts and pictures.

Limitations

Before I continue there are some limitations within Diamond Paint that have to be considered. The software only works satisfactorily if run on a 130XE or memory expanded XL/XE. With this hardware configuration you need to produce a new boot disk containing all the software supplied along with the appropriate memory driver (from the Diamond GOS Utility disk). If your hardware does not support this additional memory you will have to save your picture before returning to the Diamond Paint menu bar - a very time consuming and wasteful activity. You will also be unable to print your picture.

The Menu bar

When run, Diamond Paint, like all Diamond based programs, presents you with a desktop menu bar containing, in this case, three options: Desk, File and Text. The desk menu contains the info option and any desk accessories that were loaded at initial booting.

The File menu contains the options to paint a picture (New), modify a picture (Load), load and save clip art, import a picture, print, compression and exit. Text contains two options: Font and Style. Most of these options are self explanatory, those that contain special points will be highlighted within this review.

The Painting Screen

You have three ways of moving to the painting screen, these being paint (new), modify (load) and import a picture. The latter of these allows you to load an import module that can read picture files produced by other paint packages. Two modules

are supplied, one for MacPaint and one for Degas. Which ever option you select you should now be within the drawing screen.

The first thing you will notice is the tools box that occupies part of the screen and the mouse pointer that now looks like a pencil - see screen dump of sample picture below. The tool box can be moved about the screen, thus allowing you to work on the hidden area, by selecting it and dragging with the mouse button still pressed. Most of the common tools are available, including lines, mirror, circles, fill, text, erase etc. Whenever a tool is selected a yellow box is draw around it and the tool box removed when the selected tool is activated on the drawing area. Some of the functions, such as line and circle, draw ghost lines while you position them. In the case of circle the ghost circle update is very slow making the use of this function nearly impossible. The magnify function is also very impracticable in that it produces a 7x7 magnification window within one of the bottom corners, while leaving the pointer on the main screen. Within the magnification window the central

pixel is always the position of the mouse pointer. I found this system to be very difficult to use.

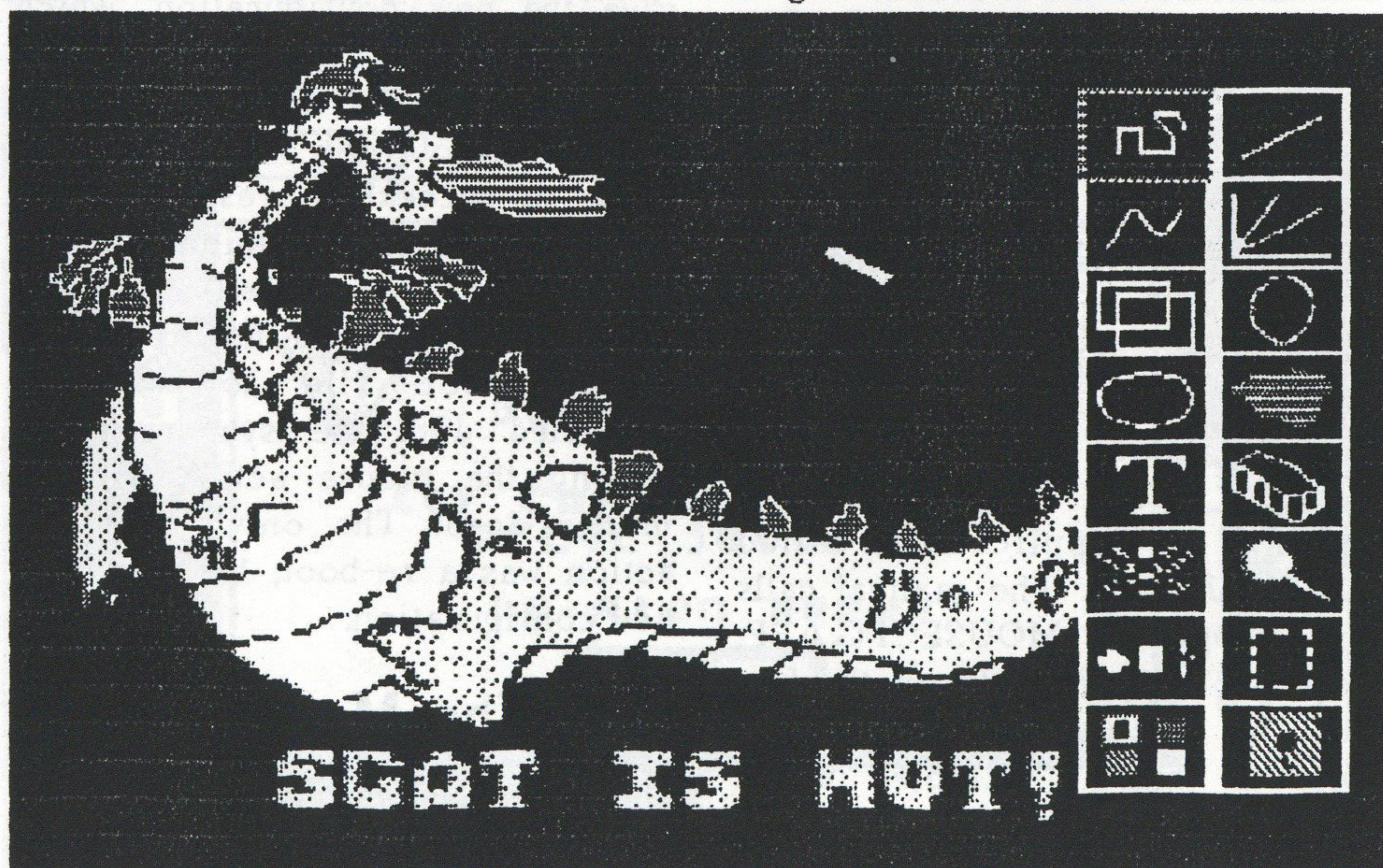
Cut & Paste

The cut and paste option allows you to duplicate parts of your picture by cutting a portion out and then repeatably pasting it back. While in the paste mode it is also possible to return to the menu bar and save the paste buffer as clip art. Clip art can also be loaded ready for pasting into your picture.

Selecting the text option within the tool box allows you to type text directly on to the screen using the pre-defined font and style, as defined within the text option on the menu bar. The size of the font is defined by the brush size currently being used. The disk comes complete with six fonts which are all available within eight styles: bold, italic, outline, underline, inverse, light, mirror and reverse. Some of these styles can be used together, such as bold, italic and underline. The main drawback with the use of the text option is that you have to return to the menu bar to load a new font resulting in multiple picture saves (unless you have a 130XE - see above). If none of the fonts supplied take your fancy the package comes complete with its own font editor.

The Font Editor

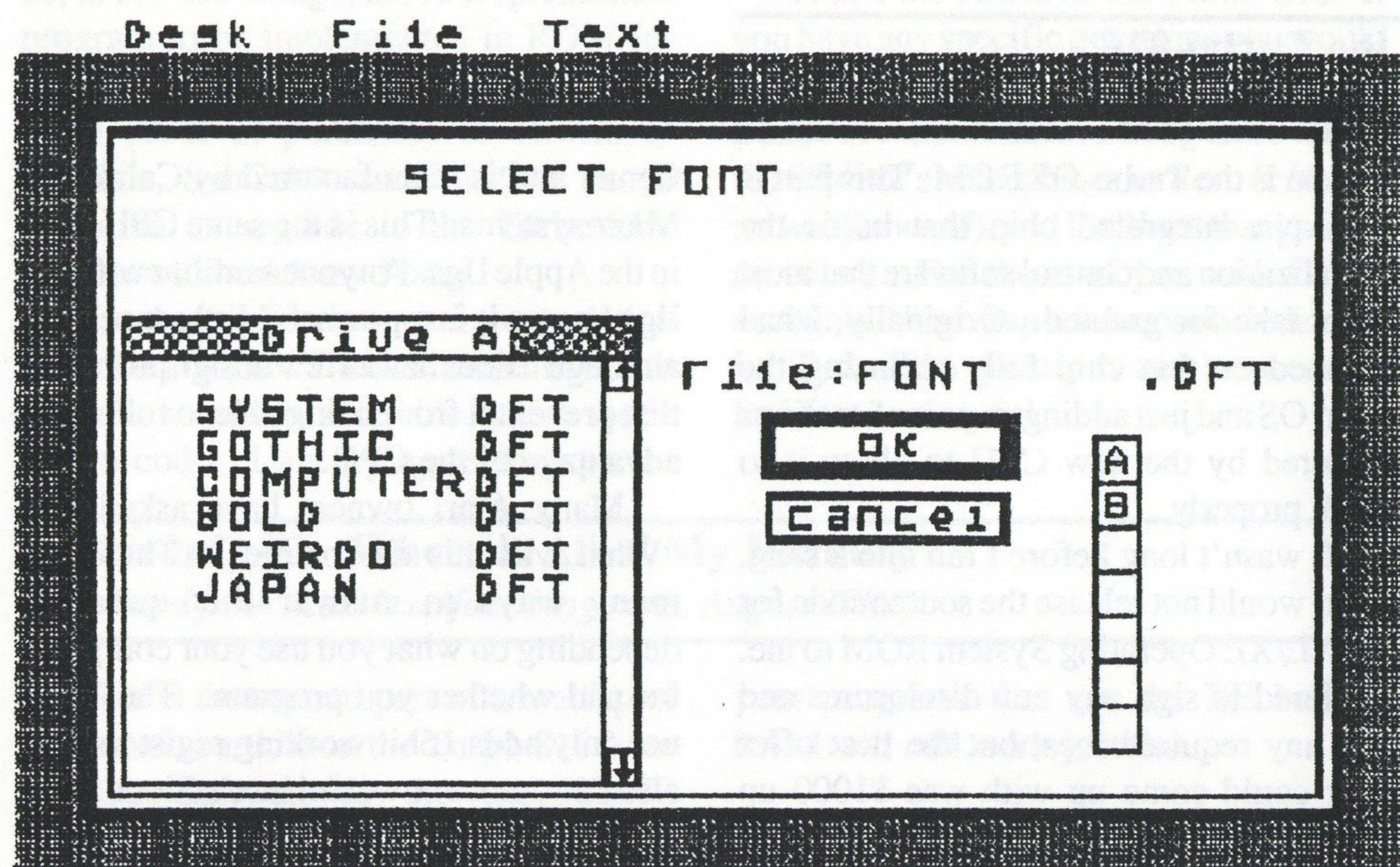
Running the font editor presents you with a menu bar containing Desk, File and Options along with a 24 x 24 grid with the letter 'A' within its



top left corner. Moving the pointer to any square within the top left 8 x 8 grid (where the letter 'A' is located) and clicking causes the selected square to change state. This

operation has no effect anywhere else within the grid, strange! Under the file menu option, as well as being able to load and save your fonts, you can import standard

Atari 9 sector character sets with the extension .FNT. Diamond paint uses 10 sector fonts with the extension .DFT. I was unable to get the options within the Option menu bar to work, these being Font Name and Edit Char. I was therefore unable to change the character being displayed within the grid. The situation was not helped by there being no documentation for the font editor within the Diamond Paint package.



Conclusion

Overall, the Diamond Paint package is a good example of what can be achieved using the Diamond environment. Some features could be improved and the manual needs updating to cover all the applications supplied.

8bit Review: Plastron

Reviewed by Ian Brooker

Plastron is the first of many new games for the Atari 8 bit from Harlequin Software, but will it have you software starved Atarians out there dashing to your nearest dealer, savings in hand. I'm not sure, read on.

Plastron is a small distant planet, of utmost importance to the universe, being the richest known source of fossil fuels. The planet has been mined for centuries by the Omni-Corp, making them the most powerful Mega-Corp throughout the Federated Galaxies. In a bid to undermine this stranglehold a band of pirates has been hired to steal as much fuel as possible from Plastron. As Surface Scavenger One your mission is to obtain as much fuel from the heavily defended mine zones of Plastron as quickly as possible. The game starts immediately after a shuttle has deposited you on the planets surface.

Before going into the game in detail a quick word about the disk loading sequence. Have you every watch other computers loading games

while displaying some stunning graphics - like the Amiga and the ST. Well, during the loading of Plastron from disk a vehicle, similar to a armoured personnel carrier trundles across the top half of the screen - very impressive.

Once loaded you control Surface Scavenger One, a two dimensional buggy, totally lacking in depth (in fact it looks like it has been squashed against a wall by an alien bulldozer), but is otherwise well animated, with knobby tyres and independent suspension. Surface Scavenger One travels along a horizontally scrolling planet surface littered with pits to avoid or jump, and dotted with fuel pods and jumps to collect. Fuel pod collection is vital to transport you to the next level and jumps are handy to help you over the pits, although the occasional ramp will help. just to make you task a little bit harder, Guardians roam the surface and air, hell bent on destroying you and thus taking one of your valuable lives.

Your final pitfall is a time limit, shown at the bottom right of the

screen by an ever exposing rib cage! Other information along the bottom includes lives remaining, jumps left, pods collected and of course the score.

The landscape is colourful as it scrolls smoothly from right to left, sound is limited to a few effects, but a catchy foot tapping tune follows you along your way. I felt that the buggy response to the joystick control to be too slow, making quick movements, necessary to dodge pits or Guardians or to catch a stray pod, frustratingly impossible.

Some of you may find this game challenging, but I fear the majority will find it all to hard and will give up all to quickly and load something else. Or will you prove me wrong? I hope so because anyone willing to write Atari games needs your support.

Graphics	70%
Sound	60%
Game play	40%
Value	50%

Turbo-Info

By Chuck Steinman (Dataque)

Turbo-Info #1: An introduction to the Turbo-816

(Reprinted from *Atari Interface Magazine*, July 1989)

Welcome to the introductory article in a series which will present the Turbo-816 from DataQue Software.

Initially, the features of the product will be presented and later the series will progress to actual applications and programming hints. If you have any comments or criticisms, I would appreciate it greatly if you would write a note to the editor of this fine magazine and express your thoughts.

For those of you not familiar with the Turbo-816, it is a hardware and firmware upgrade for the Atari XL and XE computers that expands the capability of those machines, while still maintaining compatibility with your existing software and hardware investment. There will also be a version for the original 800, although no release date has been set at this time. The XL/XE kit is called the Turbo-816x, and will sell for \$120 plus \$4.00 S&H within the US. Foreign destinations, COD and other special handling would be extra.

The main component of the T816 (short for Turbo-816) is the CPU adapter board. For those of you unfamiliar with computer jargon CPU stands for Central Processing Unit, which is the main "brain" of your computer. The CPU that came in your XL/XE computer is an 8bit 6502. The actual CPU in the Atari XL/XE computers is a slightly modified version of the 6502.

The T816 adapter board, along with a short ribbon cable, replaces your existing 6502 CPU with a new, more powerful 16bit CPU. This new CPU has the unique capability of also being able to understand the 8bit 6502 CPU instructions. That allows the adapter board to still execute your original 8bit programs.

The CPU adapter board is approximately 5 inches by 2-1/2 inches and it takes care of all timing and control functions, allowing it to fully emulate (or function like) the special Atari version of the 6502 while also extending the addressing range. One jumper must be added to your Atari XL/XE motherboard to make the system work. Installation and modification information is provided as part of the T816 kit.

The other component of the T816

system is the Turbo-OS ROM. This part is a 28-pin integrated chip that holds the initialization and control software that most users take for granted. Originally, I had planned on this chip fully replacing the Atari OS and just adding any new functions required by the new CPU to allow it to work properly.

It wasn't long before I ran into a snag. Atari would not release the source code for the XL/XE Operating System ROM to me. I offered to sign any non disclosures and pay any required fees, but the best offer they could come up with was \$1000 up front and 50 cents per unit sold, for the right to copy only certain sections of the Atari Operating System ROM, but no source code was to be included in that agreement.

I was not impressed with their support, so I proceeded to write the entire Turbo-OS from scratch. This not only cost several additional months of programming effort, but also increased the amount of time to test each and every function. The resulting Operating System is 100% compatible with software that uses the published, legal entry points into the Atari OS and legal RAM locations. Because of many programs making illegal calls to the OS or using reserved RAM. I suggest that BOTH the Atari OS and the Turbo-OS be installed in your machine and a toggle switch be used to select which is active. Details on how to make this modification are also included in the kit.

A popular misconception about the upgrade is that it will execute ST and/or IBM/PC programs. It will not.

I have never advertised or claimed it would, so I can only guess someone that was not familiar with assembly code, or the internal workings of computers, started this rumour.

The ST uses a 68000 processor, and the IBM uses an 80x86-based processor. The 65816 is not software compatible with either of those CPUs, and neither of those CPUs can directly understand 6502 code. If someone wants to write such an emulator for either of those CPUs, let me know when you have it done!

The T816 uses the WD65C816P5 CPU, which was designed by Western Design

Center and is manufactured by California Microsystems. This is the same CPU used in the Apple IIgs. Anyone familiar with the IIgs knows it is a powerful little machine, although it does have a few design problems that prevent it from being able to fully take advantage of the CPU.

Many Atari owners have asked me, "What will this do for me?" There are many ways to answer that question depending on what you use your computer for and whether you program. The T816 not only adds 16bit working registers, but also new, more powerful instructions and a 24bit address bus. Now with these new capabilities, more powerful new programs can be written to allow the Atari computers to compete again with the more expensive 16bit computers. The new addressing capability allows for up to 16 megabyte of memory (RAM) and the new instructions make smaller, faster, and more complex routines possible.

I have made MAC/65 Macros and other information available to all registered owners via my BBS and the GENIE telecommunications service, allowing owners of MAC/65, or any other macro assembler, to begin using the new 65816 instructions immediately.

To the non-programmer, at this time there are no programs which take full advantage of the T816. Current applications will run from 5% to over 300% faster depending on how much they use the Turbo-OS. Since existing software was not written to access the new memory or take advantage of the new instructions, that software can only gain performance by calls to the Turbo-OS.

I will have several new applications which will take advantage of the new memory and power, though. The first of these will be a spreadsheet program called the Turbo-S16. It will be similar to VisiCalc and SynCalc. The program will determine if a T816 and Turbo-OS are available, and then use any expanded or explicit memory, if available. Otherwise, it will run on a stock Atari 400/800/XL/XE using the normal RAM. Since it is not initially going to be designed as a concurrent application (i.e., it will not multi-task), the program

will absorb all available RAM for its use.

That means, if there is one megabyte of explicit or expanded memory available, the program will allow you to use up to that amount for the application. I plan on releasing this program in ROM, which will be, to my knowledge, the first spreadsheet program to be implemented in ROM for the Atari. A discount will be available (with proof of purchase) to owners of SynCalc, SynCalc+ and VisiCalc as an incentive to upgrade to the Turbo-S16. Details will be released later.

Another application nearing completion is the Turbo-A16. This is an assembler that will allow you to assemble native mode 65816 code. I am trying to incorporate

enough versatility to allow loading of source code from popular file formats. That will allow people to use their existing libraries without first having to convert them. It has not been decided whether this application will be in ROM or disk based.

That's the basics of the Turbo-816. If you have any specific questions you would like answered, please forward them to the Editor of Atari Interface Magazine - AIM (816 Editor's note: *please write to 8:16 and we will forward*), and I will see that you get an answer. For those of you with GENIE accounts, I have a special section dedicated for Turbo-816 questions and answers, so feel free to drop in!

Turbo-Info #2: Where is All of My Memory?

(Reprinted from Atari Interface Magazine, August 1989)

This is the second part of a multi-part information article on the Turbo-816 from DataQue Software, for the Atari XL/XE computers. The first instalment went over the basic system and this article will explain the various possible types of available memory in a T816 upgraded system.

Because of the limited addressing range of the 6502 CPU used in the Atari 8bit computers, there have been several techniques used to allow larger programs and more data to be stored. None of these techniques used to date have been efficient, and most are awkward at best.

The Turbo-816 changes all of that. Because of a lack of definition in the past of the various ways of describing the many ways the 6502 can address memory, I have made my own definitions as to how the 6502 and Turbo-816 can address memory. I have broken memory down into several basic groups.

Standard

This is the original 64K of memory that the 6502 can directly access without any gymnastics. In most systems, there would be a 16K section dedicated to the ROM space and I/O devices, and the remaining 48K is available for either RAM or cartridges.

Extended

This is the RAM that is available by using a bank selecting technique like the Atari 130XE. Many have upgraded their XL computers to also have a compatible bank selected RAM area. The Turbo-OS includes several routines to manage and report the amount of Extended RAM available. The Extended RAM is used and available exactly like it always has been, so

your programs that use the XE banked RAM will still work just fine.

Explicit

This is a 1 megabyte area for RAM or ROM which is installed by using Turbo-SRAM cards. It can be any mix of either Static RAMs or PROMS. The Turbo-SRAM card is compatible with many common sizes of RAMs and EPROMS.

The Turbo-OS checks the Explicit range from \$010000 to \$07FFFF every 32K for a pattern, to detect whether either ROM or RAM exists. The reason for dedicating an area for SRAMs and EPROMs was to allow for resident programs to be held in either EPROM, or RAM, and for those programs to optionally also have their own dedicated data RAM.

This would allow you to have several different application programs or device drivers installed in the system, and select from those programs without having to load them from a disk file or insert a cartridge. They would normally be dormant, and would be activated by a system call.

There are several operating system functions which support the Explicit memory. There are routines to allocate and de-allocate Explicit RAM. There are also routines to initialize and Execute application programs that reside in the Explicit memory range.

Expanded

This is an 8 megabyte area dedicated for use by user applications as either data or program RAM.

The Turbo-OS will scan the range of \$080000 to \$7FFFFFFF to determine how much Expanded RAM exists in a particular

system. Also, the RAM can be allocated in 256 byte blocks by applications. This will allow for multiple applications to share the available RAM. There is also a de-allocation routine to return allocated RAM back to the system.

Proto

This is an area defined for the prototyping card, which is available for any user developed applications. No production product should ever use this area. This two megabyte area runs from \$800000 to \$9FFFFFF and is not checked, or used, by the Turbo-OS.

Video

There is a 4 megabyte range of memory dedicated for video adapters, in the range of \$A00000 to \$DFFFFFF. This will allow for multiple video adapters to have their own RAM, without conflict. Also, this RAM is specified as dual-port DRAM, so unlike the ANTIC video RAM, no bus arbitration or DMA is required to access the RAM, either by the CPU or Video controller

This allows the CPU to run though video refreshes, unlike the current system in which ANTIC halts the CPU while it is reading the video RAM out to the shift registers.

DataQue

This is an area defined for in-house applications and products, and should not be used by any third party or user products. This one megabyte area runs from \$E00000 to \$EFFFFFF and is not currently checked, or used, by the Turbo-OS.

Turbio

This block is reserved for Input/Output (I/O) devices and control registers. It is located from \$F00000 to \$FFFFFF and can be used by various peripheral cards. There are specific ranges within this spec for certain devices.

At this time, the only device that uses this area is the 80 column video adapter. Other peripherals such as disk controllers, printer interfaces and co-processors would use this area for their access, to allow for maximum compatibility. Each peripheral and its supporting device driver should allow for several different locations, to allow for co-existence with other peripherals and also multiples of like peripherals.

Turbo-816 is currently being imported by Bruen Enterprises Ltd. You can find their advert within this Issue of 8:16.

32,768 Colour Support for the Atari ST

by Barry Orlando

For Public Domain, dated 9-9-89.

Printed by permission of the author.

Re-printed from Atari Interface Magazine, October 1989.

Here's a hardware modification for the standard Atari ST that increases the colour palette from 512 colours to 32,768 colours (32 shades of gray), while still maintaining compatibility with existing Atari ST software (including SPECTRUM 512). This upgrade is software compatible with JRI's 4096C colour board and will also be compatible with the recently announced 4096 colour STE (Enhanced ST).

It does this by adding increased bits of resolution so as to increase the ST's 16 read/write colour palette hardware registers from their standard 9 bits each to 15 bits each. Each of the standard colour palette registers have the following bit arrangement:

FEDCBA9876543210
•••••RRR•GGG•BBB

where, R, G and B are the red, green and blue components.

The higher the value for any of these grouped bits, the more intense the resulting colour component. The above bits marked '•' are ignored.

The JRI 4096C colour board and the upcoming 4096 colour STE add an extra bit of resolution at bit locations 3, 7 and B, however these new bits are least significant bits so as to maintain compatibility with existing software.

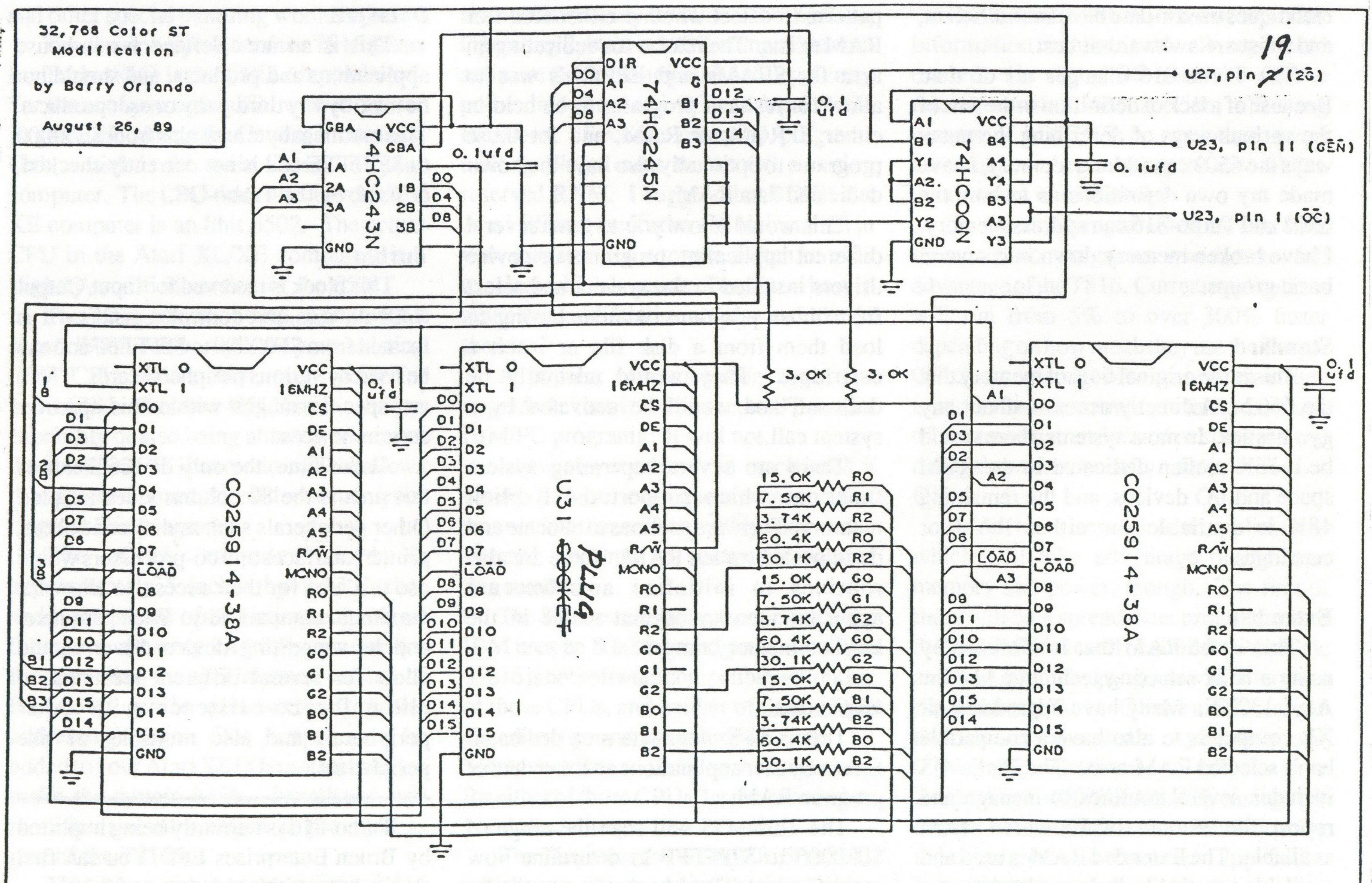
The 32,768 colour modification goes a step further by adding a fifth (and less significant) bit to each colour component at locations C, D and E. The new arrangement is then:

FEDCBA9876543210
•RGBRRRRGGGGBBBB

As with the JRI 4096C colour board, this modification adds an additional Shifter chip. However this modification is slightly more involved in that it uses three additional integrated circuits and by passes nine 5% tolerance resistors used by the Shifter chip, in favour of using nine 1% tolerance resistors.

Disclaimer of Liability

This modification should only be attempted by someone with experience repairing or building digital electronic circuits. Performing this modification will be done at your own risk and may void the warranty on your computer.



Parts Used

(Note: Wherever possible 8:16 have added details for a UK supplier)

- | Qty | Description / Source / Cost |
|-------|--|
| 1 | Component Perfboard (3" x 52.5") cut from Radio Shack #276-147 (\$2.99). |
| 2 | 3M Board Mounting Interconnectors (straight single-row male). Digi-Key Corp., Part # 92983408-36 (\$2.05 each). |
| 1 | Quad 2 input NAND Gate, High Speed CMOS (74HCOON). Digi-Key Corp., Part # MM74HCOON (\$0.28). Maplin Electronics, UB00A (£0.24) |
| 1 | Quad Tri-State Transceiver, High Speed CMOS (74HC243N). Digi-Key Corp., Part # MM74HC243N (\$0.78). |
| 1 | Octal Tri-State Transceiver, High Speed CMOS (74HC245N). Digi-Key Corp., Part # MM74HC245N (\$0.82). Maplin Electronics, UB67X (£0.60) |
| 5 ea: | 3.76K, 7.50K, 15.0K, 30.1K, 60.4K ohm metal film resistors, 1%, 1/4 watt. Digi-Key Corp., Part #s 3.76X, 7.50X, 15.0X, 30.1X, 60.4X (each value: 5 for \$0.50). |
| 2 | 3.0K ohm carbon resistors, 5%, 1/4 watt. Digi-Key Corp., Part # 3.0E (5 for \$0.25). |
| 5 | 0.1 ufd Monolithic Ceramic capacitors, Radial, 50V, 10%, or equivalent substitution, DigiKey Corp., Part # P4525 (\$0.19 each). Maplin Electronics equivalent 0.1 ufd 100v order # RA49D (£0.38) |
| 2 | Shifter Chips, Atari Part # C025914-38A. One chip is obtained from your ST's motherboard, the other from your local Atari Authorized Service Center (\$30.00) |

Digi-Key Corp., 701 Brooks Ave South, P.O. Box 677, Thief River Falls, MN 56701-0677. Telephone 1-800-3444539 for a free catalogue.

Maplin Electronics, P.O. Box 3, Rayleigh, Essex, SS6 8LR. Telephone 0702-554161

Installation of the Board

Located on the ST's motherboard is mounted a sheet metal box which houses the Shifter Chip. This box has a hinged lid which can be opened. Inside this box you'll find the Shifter Chip mounted in a 40 pin socket.

The board simply plugs into this socket so that the board is elevated above the ST's



```

/* 32,768 colour Board Tester for Palette Register */
/* at location $FF8240 */
/* by Barry Orlando in Mark Williams C */
/* This program displays the 32 grey levels and then */
/* the 32,768 colors in sequence using background palette 0 */
#include <stdio.h>
#include <osbind.h>
#include <aesbind.h>
#define peekw(wp)((*(int*)wp))
#define pokew(wp,w)((*(int*)wp)=w)
/*****main*****/
main()
{
    int g, b;
    unsigned int c, r;
    long i;
    for (r = 0; r > 23; r++) printf("\n"); /* clears screen */
    for (r = 0; r < 32; r++) /* red, green and blue values= r */
    { /* c format is RRRRRGGGGGBBBBB */
        c = r < 10; /* red */
        c += r < 5; /* green */
        c += r; /* blue */
        c = cvrt_m(c); /* convert to machine code */
        printf("grey level: %d \n",r);
        clr_write(c); /* set grey level */
        for (i = 0; i < 65000L; i++); /* added delay */
    }
    for (r = 0; r < 32; r++)
    { for (g = 0; g < 32; g++)
      { for (b = 0; b < 32; b++)
        { c = r < 10;
          c += g < 5;
          c += b;
          c = cvrt_m(c);
          clr_write(c); /* set palette color */
          printf("\nred: %d green: %d blue:%d ",r,g,b);
          for (i = 0; i < 30000; i++); /* Added delay */
        }
      }
    }
    gemdos(0x01);
}
/*****clr_read*****/
clr_read() /* reads color palette register */
{
    long save_esp;
    int o;
    save_esp = Super(0L);
    o = peekw(0xFF8240L);
    Super(save_esp);
    return (o);
}
/*****clr_write()*****/
clr_write(o) /* writes to color palette register */
unsigned int o;
{
    long save_esp;
    save_esp = Super(0L);
    poke(0xFF8240,o);
    Super(save_esp);
}
/*****cvrt_m()*****/
cvrt_m(a) /*converts readable code to board compatible code*/
unsigned int a; /* bit pattern in: RRRRRGGGGGBBBBB */
/* bit order in: M321LM321LM321L */
/* bit pattern out: RGBRRRRGGGGBBBB */
/* bit order out: LLL1M321M321M32 */
/* bit: FEDCBA9876543210 */
}
unsigned int f;
f = (0x400 & a) < 4; /* red bit L */
f += (0x20 & a) < 8; /* green bit L */
f += (0x1 & a) < 12; /* blue bit L */
f += 0x800 & a; /* red bit 1 */
f += (0x40 & a) < 1; /* green bit 1 */
f += (0x2 & a) < 2; /* blue bit 1 */
f += (0x7000 & a) > 4; /* red bits M,3,2 */
f += (0x380 & a) > 3; /* green bits M,3,2 */
f += (0x1C & a) > 2; /* blue bits M,3,2 */
return(f);

```



motherboard with sufficient clearance to allow closing the hinged lid.

Three wires are installed from the colour board to three vias (trace thru holes) located nearby on the motherboard.

Additionally jumpers are installed across each of three 3.6K ohm resistors on the motherboard. These resistors are easily traced from the Shifter socket pins 21, 24 and 27.

Board Assembly

Because the sheet metal box (discussed above) does not allow a lot of head clearance all components are mounted directly to the perfboard. If you use chip sockets the hinged lid will not fit back on without modification.

The diagram shows the orientation of components. I soldered all the components to the perfboard then soldered all the connections using 30 gauge Kynar wire wrap solid conductor wire (Radio Shack 278-502).

```

}
/*****curt_b()*****/
curt_b(b) /* converts back to readable code */
unsigned int b;
{
    int d;
    d = (0x4000 & b) > 4; /* red LSB */
    d += (0x2000 & b) > 8; /* green LSB */
    d += (0x1000 & b) > 12; /* blue LSB */
    d += (0x800 & b); /* red bit 1 */
    d += (0x80 & b) > 1; /* green bit 1 */
    d += (0x8 & b) > 2; /* blue bit 1 */
    d += (0x700 & b) < 4; /* red bits M,3,2 */
    d += (0x70 & b) < 3; /* green oits M,3,2 */
    d += (0x7 & b) < 2; /* blue bits M,3,2 */
    return(d);
}

```

There is a slight modification to the circuit if you have any RAM chips installed in your ST which are 150 nanosecond (i.e. slower than 120 nanosecond). Pin 1 (DIR) of the 74HC245N must be tied to ground instead of to the R/W line on the Shifter. This will only disable the read capability for bits C, D and E of each colour palette register mentioned above.

Software

Included with this article is a program to display all gray levels and colours. This is a very crude program but was intended to only prove that the circuit works. The C source code provides the engine for converting basic colour intensities to board compatible bit patterns.

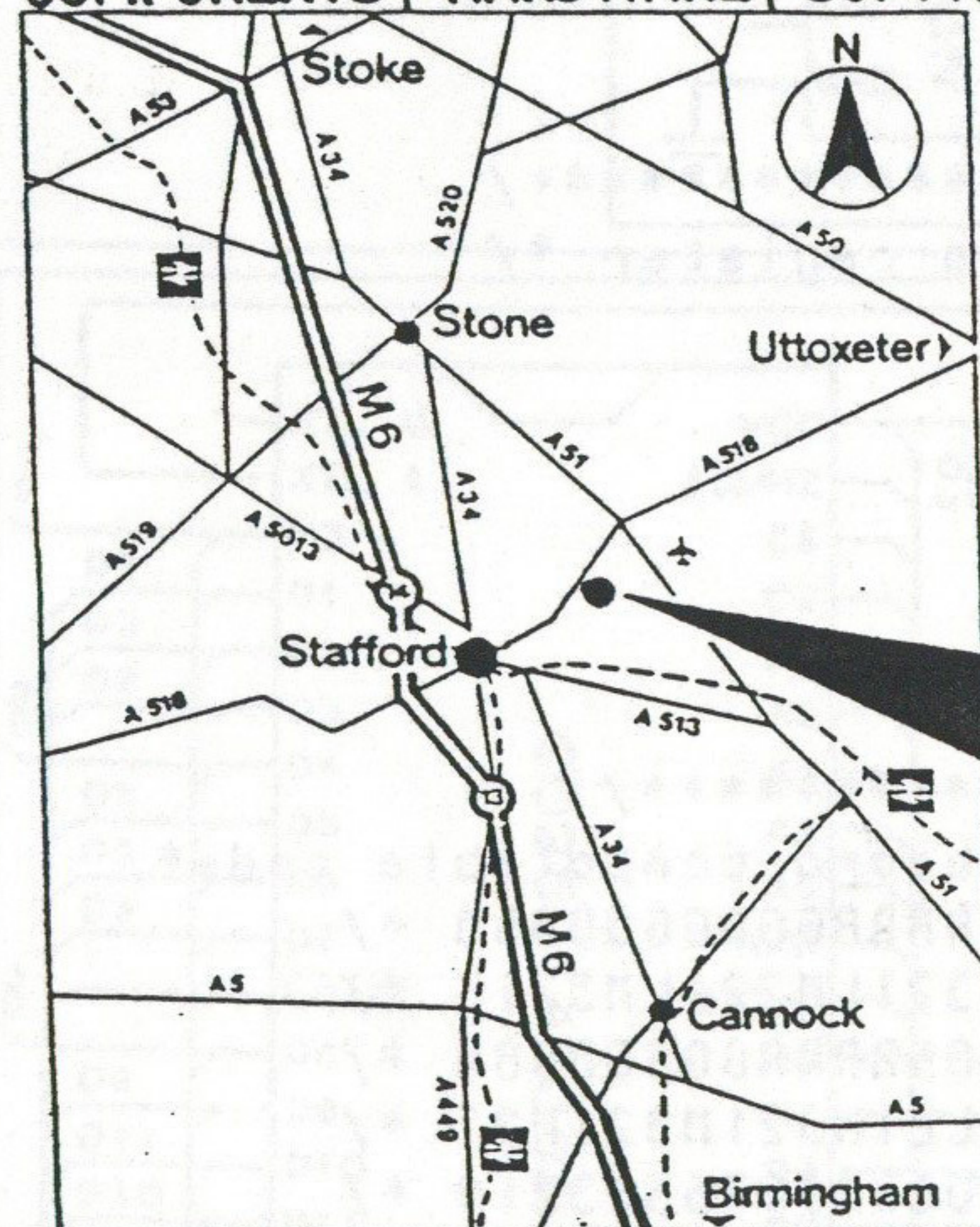
Sharward Services

PRESENTS

AMS4

THE FOURTH ALTERNATIVE MICRO
SHOW, RADIO RALLY AND
ELECTRONIC FAIR
FEATURING

COMPONENTS / HARDWARE / SOFTWARE / DISK DRIVES / RADIO



ELECTRONICS / BOOKS
TOOLS / COMPUTERS
MEDIA/BRING AND BUY

A
GREAT TIME
TO BE HAD
WITH ALL
THE FUN
OF THE
MICRO FAIR

Staffs
Show
Centre

VENUE - BINGLEY HALL, STAFF SHOW CENTRE
DATE - SATURDAY 10TH NOVEMBER 1990
TIME - 10AM TO 5PM
PARKING - FREE

FULL CATERING SERVICE AVAILABLE THROUGHOUT THE DAY
THE FOURTH ALTERNATIVE MICRO SHOW IS

ORGANISED BY:
Sharward Services

TELEPHONE 0473 602460 FAX 0473 602410

GRALIN INTERNATIONAL

11 Shillito Road, Parkstone, Poole, Dorset BH12 2BN

Cheques / Postal orders only. Please add £1 P&P to all orders.

Overseas orders please add £2.50 per order. Please allow 28 days for delivery.

APX Clearout

APX, which stands for Atari Program eXchange, was an idea by Atari in the early 80's whereby users could write programs for other users. The programs within APX tend not to be commercially viable, but were better than most PD. Unlike PD, all the programs come fully documented - in some cases this is over sixty pages. Below is a selection of titles available - send SAE for complete list.

Cassettes: £1.50 each; 5 for £7.00; 10 for £13.00

Attank / Babel / Banner Generator / Block 'Em / Counter / Decision Maker / Downhill / DSEMBLER / Dice Poker / Extended WSN (Turtle) / Instedit / Look A Head / Morse Code Tutor / Phobos / Pro Bowling / Salmon Run / Solitaire / Sound Editor / Space Chase / T: Text Display Device / Terry / Text Formatter (Forms) / Ultimate Renumber Utility / Yahtman

Diskettes: £2.50 each; 5 for £11.75; 10 for £22.00

BASIC Renumber Utility (RENUM) / Blackjack Casino / Blackjack Tutor / Block 'Em / Codecracker / Computerised Card File / Counter / Downhill / Family Cash Flow Rev. 2 / Insedit (Microsoft BASIC Version) / Insomnia (A sound Editor) / Meltdown / Phobos / Player Piano / Stereo 3-D Graphics Package / Typo Attack / Variable Changer / Wizards Gold

DISK BASED SOFTWARE CLEAROUT

£4.99 each: Asylum / Aztec / Bishop's Square & Maxwell's Demon / Hard Hat Mack / Hardball / Jeeper's Creepers / Juggler / Juno First / Lapis Philosophorum / Maltese Chicken / New York City / One On One / Pacific Coast Highway / Phantom / Protector II / Quasimodo / Realm Of Impossibility / Rochocet / Serpentine / Slime / Stella Shuttle / Survivor / Ulysses And The Fleece / Zaxxon

Send SAE for complete list of APX and disk based software

7800 Software Roundup

Ace Of Aces - Atari £12.99 Accolade classic. Faster than the 8bit version.	Fight Night - Atari £12.99 Funny boxing game.
Ballblazer - Atari £12.99 Lucasfilm classic, originally developed for the 7800 with special sound chip built into the cartridge.	Food Fight - Atari - £9.99
Centipede - Atari - £9.99	Hat Trick - Atari £12.99
Choplifter - Atari £9.99 Better than the VCS cartridge - no flicker, smooth animation. Fore runner of 'Gauntlet'.	Joust - Atari £9.99 Ride your Ostrich and let them have it. Very nice game graphics.
Dark Chambers - Atari £12.99 Again, much better than the VCS game of the same name.	Karateka — Atari £12.99 Six year old Broderbund title, ninja type game. Still stands up after all these years.
Desert Falcon - Atari £12.99 Dig Dug - Atari £9.99	Mario Bros — Atari £12.99 Good graphics and even better game play.
Donkey Kong - Atari £12.99 High price for an old game.	Ms Pac Man — Atari £9.99 Don't need to say anything about this one.
	Pole Position II — Atari £12.99
	Xevious — Atari £9.99

ATARI XL/XE OWNERS

Public Domain and commercial software on disk and tape!

Utilities, digitized music/speech demos, graphic demos, sound fx, adventure games and a whole lot more.

Disk prices from only £1.50 a disk!

Also selected P.D. on one long tape.

All programs written by me, so it's unlikely you've got them already!

For a price list, send a SAE to:

Dean Garraghty,
62 Thomson Avenue, Balby, Doncaster,
DN4 0NU

VCS Game Reviews

By Thomas Holzer

Dark Chamber

Atari Corp. £12.95

You are on a dangerous search, deep underground. Wraiths, wizards and mouldering skeletons surround you. Even now a grim reaper stalks you.

The exits take you - where? Only deeper into the maze. Find weapons in the darkness and fend off your enemies. Claim the treasure and fight your way even deeper into the caverns. Your strength will out last the weaker ghosts, but are you a match for the strongest?

This cartridge is Atari's very own Gauntlet clone and can be played with one or two players. You control hero 1 and a friend can control hero 2 (wonderful names) to search for underground treasures and fights ghouls and ghosts. On your journey you can collect potions, weapons and various keys and treasures while being aware of poisons, traps and false doors. You also have to shoot *spawners* which generate ghouls and sometimes hide doors, keys and potions.

I can't really recommend this game because the screen flickers badly, the characters move too slowly and the whole game is not up to Atari's high standard. 1983 maybe, but 1990, no way! Its one redeeming factor is the nice title screen with rotating Atari logo.

Double Dragon

Activision £12.99

The black warriors have kidnapped your girl. That was their first mistake, challenging you and your twin brother in combat will be their second.

Double Dragon, from the arcades to your VCS 2600, is a one or two player game with the best ever graphics and sound (quote from the package).

I must admit, the graphics are very good, no flicker, smooth scrolling, even with four characters on the screen at once. Sound? Yes, there are two different tunes very clear with the title music being the better one. The game play music is a bit annoying after a while, never mind.

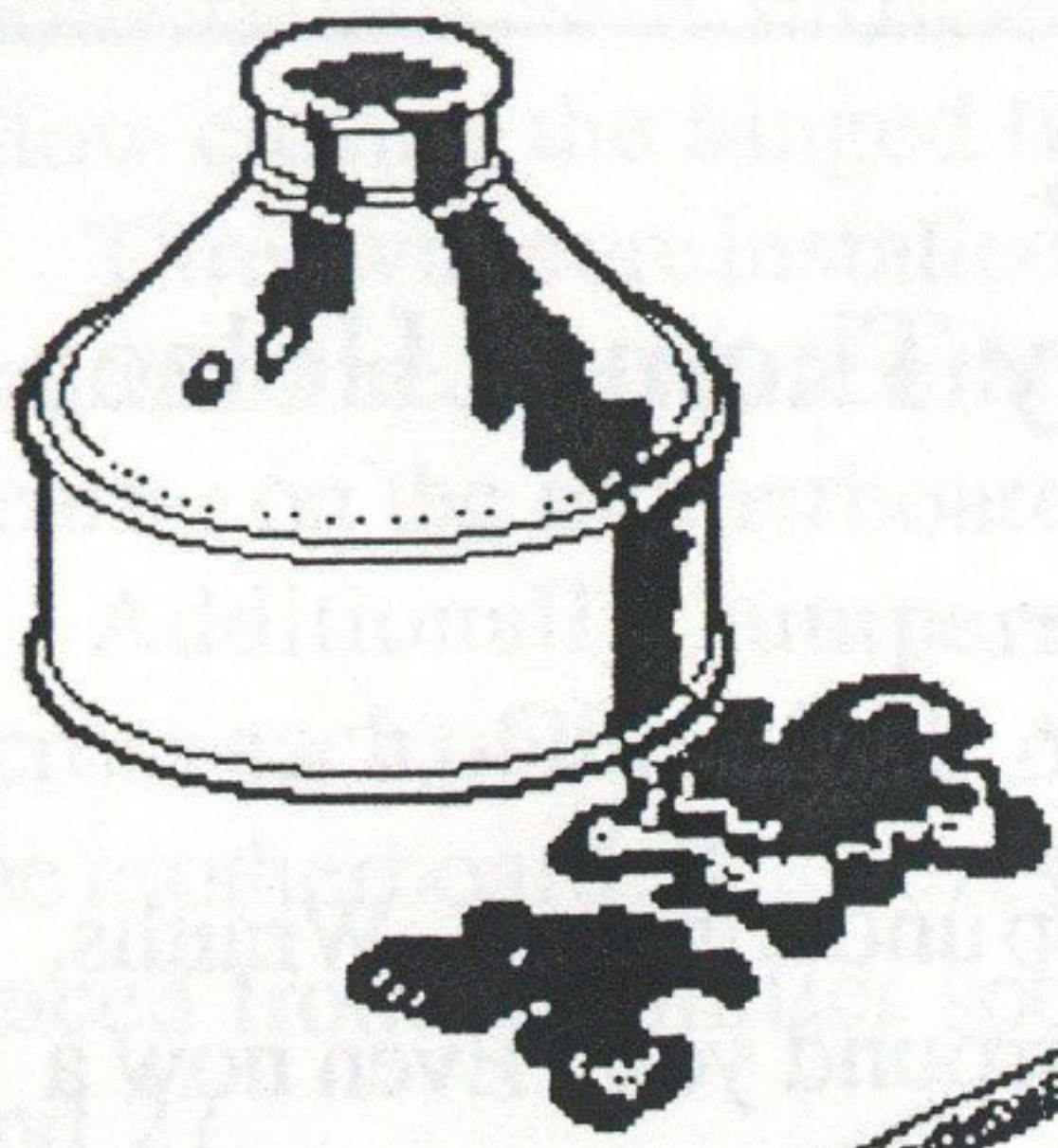
So, you and your friend guide Billy and Jimmy stroll down the streets to rescue your girl, beating up everyone that stands in your way (even nice blond girls).

There are four fighting moves and various weapons to collect. Also there are four missions and an ever decreasing timer to watch out for. You can also play this game alone, this makes it a bit harder, but is fun never the less.

If you know the arcade version of this game, you will like it. All I say is it's been a long time since a good arcade conversion made it to the VCS.

VCS Software Roundup

Defender II - Atari £9.99 Stargate under a different name.	Rock 'n' Rope — Prism/ Activision? £9.99 Old Atari 8bit game converted to the VCS
Real Sports Boxing — Atari £12.99 Good game for two players.	Super Football - Atari £12.99 Good 3D view of the game.



Dear 8:16,

Frontier & ICD Depart Company

Thank you for your recent order. As you may know, the item in question is manufactured in the USA by ICD. We have in the past acted as their distributor here in the United Kingdom. Sadly, for reasons beyond our control, we have this day reluctantly taken the decision to no longer represent them here in the United Kingdom. We regret that this decision means we are unable to fulfil your order. To this end, we have enclosed your original cheque or postal order since we have not processed any payment. If you intended payment via VISA or MASTERCARD please also be assured that no payment has been processed.

Should you wish to contact ICD direct, we have listed their postal address at the foot of this page.

Yours sincerely,

Martin Walsh Andrew R. Bennett
Marketing Manager Managing Director

ICD, Inc,
1220 Rock Street
Rockford, IL 61101-1437
United States of America

The above letter has recently been sent to all 8 bit users with outstanding orders for ICD products with Frontier Software. It is a real shame that ICD can no longer support the 8 bit, preferring to concentrate on the 16 bit market. I would like to take this opportunity to thank Frontier for all their support in the past and wish them all the success for the future.

Ribbon Refresh

I have been given to understand that there is a computer service in your area in Poole who supply a printer ribbon refresh for the 1029 printer. I have never heard of this before but I suppose anything is possible today.

Have you any experience of this particular commodity for replenishing printer ribbons. If you know of this company which trades under **Caspell Computer Services** could you let me have their address in Poole.

I would also be obliged to receive your up to date PD list, as my latest one indicates a number out of stock.

Could you also tell me the date of the next **Alternative Micro Show**.

Alfred Smith; South Humberside

Caspell Computer Services have been offering this product for some time now, though I have had no experience with it. Your best bet is to try it and find out. Their telephone number is (0202) 666155 and their address is 2A Sterte Industrial Estate, Sterte Road, Poole, Dorset (sorry I don't know the post code). Please let us know what you think so we can inform our readers.

Hopefully, you will find the latest PD catalogue within this issue of 8:16. Some disks (such as 13) have been removed as it now appears that they will never appear - before 'Out of Stock' meant I was still waiting for the master.

The next Alternative Micro Show (AMS4) is on the 10th November 1990 at the Staffordshire Show Centre (Bingley Hall) starting at 10am. For more information contact Sharwood Service on (0473) 602460. If you do decide to come along, look out for us, as we'll be there somewhere!

BaPAUG Awards

I was delighted to discover that you have given awards to STOS and Mandarin Software - these are very much appreciated. Sales of STOS will tip the 100,000 mark by the end of the year as a result of a bundling deal with Atari to include STOS with every 520ST sold.

We will continue to promote and support STOS throughout this year - we have just launched *Cartoon Capers*, the winner of the STOS Gameswriter of the Year Award 1989. This comes with a free double joystick and is selling well.

Christopher Payne; Managing Director

When producing any type of newsletter it is always very gratifying to see that companies like Mandarin (and GFA who telephoned me with similar remarks) take time to read the newsletter when they receive it.

Packet Radio?

Your name has been passed to me by Derek Fern who recently supplied me with an Atari 65XE computer and 1050 Disk drive.

My query is whether the radio equipment I have, Sangean ATS 803A radio and Realistic PRO 2005 scanner, can in some way be operated in conjunction with 8 bit computers or what software and hardware is required.

Brian Taylor; Glasgow

Is there any readers who can help Brian. I've asked some of our members but have so far drawn a blank.

Viewdata for the 8 bit?

I saw a reference to your group and magazine when I was online to the Ark BBS. I would like to ask you a couple of questions:

1) Is it possible for me to get a viewdata / videotext program for the Atari 8 bit? If I can then I can get online to Micronet and prestel. With the McHugh patch and all that it should be possible.

2) I have downloaded a file from the Ark BBS which was a binary file to un-arc any arced programs. I tried to binary run this program from DOS and it would not run. I tried to binary run from Turbo BASIC and it still would not run. I renamed it AUTORUN.SYS and it still would not run! The program loaded in all cases but would not run. What am I doing wrong? I seem to have successfully

downloaded it using Xmodem. Any thoughts?

Mike Falconer; Cheshire

In response to the first question, sorry, I know of no software. Is there anyone out there that knows different? Then please write in. The second question may be easy to answer. Did you boot WITHOUT BASIC? If not try that (press OPTION while switching on).

MYDOS and the XF551 Disk Drive

Firstly, I would like to thank you and 8:16 for making me a very happy person - because of the item in the news section of the last issue I was able to purchase a disk drive. Something I have been after for about 4 years!

I have frequently been told that if you have a problem you should get in touch with a user group, so here goes:

As stated above, I now own an XF551 disk drive (after four years of cassettes) and know nothing about disks. I had no trouble with DOS 2.5 but found that single sided enhanced density was not making very economic space of disks, because XF551/DOS 2.5 can't format the B side of the disk. I bought MYDOS from Page 6, but the manual is awful, leaving the beginner totally baffled. I need to know the exact answers to the reconfigure questions. So far I seem to be able to format a disk with anything from 1200 to 65,000 sectors ... still cannot write to side B.

When I first got the drive I converted some old BASIC cassette programs to disk, this worked OK. I even locked all the files. When I got MYDOS I tried to copy some over, but some of the filenames began with numbers and MYDOS will not

read them. On trying to unlock the files and rename (in both DOS 2.5 and MYDOS) I found to my horror that they would not unlock, and I kept getting errors. Also some files would not copy to MYDOS format. Please help me, I'm going mad. I suspect that I'm doing something wrong? How about a series on MYDOS like the excellent DOS 2.5 one?

How can I make Turbo BASIC compatible with MYDOS? I've tried writing the MYDOS files to the TB disk, this works fine until one accesses DOS, then it crashes.

Finally, I recently bought Datasofts Alternative Reality: The City. The game allows you to back-up disk 1/side 2 and all of disk 2 with the provided utility. Because I cannot format side B I have to use three disks.

Paul Allton; Somerset

The above is an abridged version of Paul's letter, I have answer some of his question in a letter already and have included it here as a lot of users are having the some problem with the XF551 drives. Firstly the XF551 drive is the first 8 bit drive released by Atari that is designed to write to both sides. The 810 & 1050 could only do this by notching the disk and turning it over

- thus turning one disk into two separate disks. You can also do this with the XF551 if your DOS does not support the XF551. If you have a disk with both sides formatted by the 1050 the XF551 is incapable of reading the B-side unless you turn the disk over (remember its two logical disks on one physical disk).

Can anyone help Paul with MYDOS. I do not own a XF551 so do not know the required configuration. Two DOSes that do work with the XF551 are DOS XE and SpartaDOS. The former being release by Atari, the latter by ICD. At the moment neither of these DOSes are available within the UK (though Gralin International will import DOS XE if requested). You may be able to obtain SpartaDOS second hand. Of the two only SpartaDOS (the X version with a 130XE) supports Turbo BASIC. I do not know how to make TB compatible with MYDOS - is there anyone out there who can help? Also, can anyone help Paul with his file transfer problems?

On your file lock problem could you please send the error messages as this would help greatly. Had you write protected your disks as well?

Lastly, see note above about 1050 formatted disks with the XF551. I suggest you notch your blank disk and format each side as individual disks.

Where's 8:16

Just a line or two to enquire about 8:16 - it seems a long time since issue 9 popped through the letter box, and it said that the next issue was due on the 4th May.

I'd also like to make a suggestion for a possible article. There are many tutorials in "C" running currently, but none offer any help for the BEGINNER (like me). If you have a PD "C" compiler all you get is incomprehensible stuff about emacs and -o. What does it mean, how do we use it and why? This I feel would be far more useful than discussions about scanf and at least would allow one to actually get started - even ST Format's series is guilty in this respect, as well as containing contradictory information in the first of the series.

Derryrk Croker; Watford

For the last year and bit 8:16 has been taking longer and longer and longer to produce. Issue 9 was released about six weeks late with issue 8 being about two weeks late. While producing issue 9 I started collecting articles and promises for articles intended for issue 10 (this issue) in order to try and get the release dates back on schedule. Because of this I was able to give a list of possible article intended for this issue within issue 9 - a very rare event with 8:16. If you

compare that list with the contents of this issue there is an approximate 50% match. The non appearance of the other articles is one reason this issue is late.

Another, very important problem, is the amount of free time I currently have spare to spend on producing 8:16. We are currently addresses this problem within the group and I hope soon to announce permanent 8 bit and ST editors who will remove some of the load.

Just to let you know where we currently stand, this issue is being released on the release date intended for issue 11. We have therefore skipped one quarter (an event which has happened once before). The release schedule for the next four issue of 8:16 (and this issue for reference) are:

<u>Issue</u>	<u>Copy Date</u>	<u>Release Date</u>
10	9th Jul 90	20th Aug 90
11	15th Oct 90	19th Nov 90
12	14th Jan 91	18th Feb 91
13	15th Apr 91	20th May 91
14	15th Jul 91	19th Aug 91

I will try my utmost to keep to this schedule. You may of noticed that we have included several re-prints from American newsletters - Current Notes and Atari Interface Magazine.

This is something we intend to do for future issues. It is also hoped that these newsletters will re-print articles from previous issues of 8:16.

Sorry I've turned this reply into an editorial, but I've had several letters asking about the whereabouts of issue 10 and this seemed the best way to explain.

As to your request for a tutorial for beginners I will try and sort something out for the next issue. There must be many other readers who have the same feelings about "C" and other topics, and without being told it is very difficult to decide where to pitch the level of each article.

**Any views on 8:16?
Any ideas for articles?
Any comments on computing in general?
Any requests for help?**

Then write to:

**The Editor, 8:16
248 Wimborne Road, Oakdale, Poole,
Dorset BH15 3EF**

AtariWriter Plus Explained for People Who Don't Want to Read the Book

by Jimmy Boyce (CACE)

Reprinted from Atari Interface Magazine, August & September 1989

It shall be my endeavour to rewrite the book and explain (from personal experience) what this word-processor is all about. STUDENTS, this will interest you, because I have used this program exclusively for ALL OF MY HOMEWORK at Jackson Community College and the Jackson Area Career Center. For statistical work and charts I use B-graph. That I will get into at a later date.

First of all, there is an interesting article in the July 1988 issue of ANTIC about the subject at hand. Secondly, I will be just sort of rambling through the handbook without any sort of order. Opinions about the various aspects of AtariWriter Plus (AW+) will be my own and not necessarily those of CACE.

I received my copy of AW+ as a birthday present from a very dear friend. To that friend, I say THANKS A THOUSAND TIMES OVER. One of the nicest parts about this program is that the disk has versions for the 800, 800XL, 1200XL and the 65XE on one side and for the 130XE on the other. For me this is super because I have an 800 and 130XE, so both computers are usable with AW+ and the spell checker works for all of the aforementioned computers. If you put in the wrong side (if you don't have an 130XE), it reminds you of that fact with a little message.

I am not going to go through the methods of setting up your equipment and how to load a disk. I figure you have already done that, so let's assume that you have got all of your equipment plugged in to the proper places and in the proper order, and you have turned on the system and booted up AW+.

After it has gone through all its grinding and flashing on and off, a menu appears on the screen. Lots of stuff with the first letter of each line looking strange when compared to the rest of the line. This stuff will begin the formal discussion of AW+. When ever you see <P#>, that is the reference page in the manual.

Create A File <P6> & <P13>

As the name implies, you are about to fill a blank page with some sort of literary genius of your own (or someone else's if you are copying). Stroke the letter "C" and

the screen divides itself with the top being blank and, at the bottom, is a second little rectangle with arrows. The arrows are for tab spacing (more about that later).

Next, you will see INSERT MODE. This means that as you type and discover that you have something additional to put in the text, you can go back to the place that you wish to insert that information and start typing as it pushes the rest of the text ahead of it and nothing is lost. If you wish to type over something, then hold down the CONTROL key and strike the INSERT key and you will see the message TYPE-OVER MODE.

Be careful of this arrangement; if you forget, you can lose a lot of hard work. LOWERCASE is next - by striking the CAPS key that word would change to UPPERCASE. If you want to type in only capital letters, then change it and you'll know from the screen that the change has been made.

The bottom line contains a number and BYTES FREE. This number varies according to the type of computer and disk drive you have. BE AWARE OF THE MAXIMUM SIZE FILE YOUR COMPUTER WILL LOAD!!! I learned the hard way and had to do some fancy editing to save what I had written (more about that later).

C=1 is the next thing you see, and that locates you on the vertical axis of the screen. L=1 locates you on the horizontal axis of the screen. Well folks, that locates you on the computer so you can begin your creativity with the best of them, even Bill Shakespeare.

You should be sitting with a blank screen in front of you, so experiment. It's fun and not too many computers have been destroyed with the normal finger tip pressure applied to the keys.

Don't tell me, you are sitting in front of that blank screen too terrified to touch a key! I didn't think so! I mean you might get some sort of a rare terminal (CRT) disease like dry eyeballs or something.

Edit File <P6>, <P13,14>

Once you have started a file and you need to do something else, like checking the formatting prior to printing, just hit

ESCAPE (the [ESC] key in the top left corner of the key board) and the Main Menu returns. After you have done whatever you wanted to do and have hit the [ESC] key again, the Main Menu has returned.

Folk's don't panic.- I know what goes through many people's minds when that four letter word jumps up, but in this case EDIT means return to the document you were working on. So, strike the letter "E" (remember it can be a lowercase E) and you are back to those little pearls of literary genius of yours - or hit the BREAK key (that's in the upper right hand corner of your keyboard) and you will be returned directly to your document.

Some of the handy edit commands are:
[SELECT][T] — This takes you directly to the top of your document. Handy when you want to make a change near the top.

[SELECT][B] — This takes you to the bottom of your document when you are done at the top and want to continue writing. Isn't nice how the commands make sense - T for top, E for edit, B of bottom?

[CONTROL] — By holding down the [CONTROL] key first and then holding down on an arrow key, you can scroll a line or column at a time to locate the CURSOR wherever you wish.

[SELECT][>] and [SELECT][<] — Holding down [SELECT][>] moves the CURSOR to the next word on the right which speeds up the process of moving the CURSOR; [SELECT][<] does the same thing in the opposite direction. I found this to be a bit awkward as a key stroke, but that's my opinion.

[CONTROL][A] and [CONTROL][Z] — These are handy commands - [CONTROL][A] moves you to the beginning of the line you are on and [CONTROL][Z] moves you to the end of the line you are on. This is nice when scrolling up and you are on the left side of the screen and want to get to the right side in a hurry or vice versa.

[OPTION] Key Commands — To get to the bottom of the screen (not the document) try using [OPTION] and the down arrow. Use [OPTION] and the up arrow to get to the top of the screen.

Deleting Text

If you goof and need to delete a letter, place the CURSOR over the letter and hold down the [CONTROL] key then hit the [DELETE/BK SP] key then type in the correct letter. The [DELETE/BK SP] key when struck alone works like a typewriter and deletes the letter to the left of the CURSOR.

There are ways of DELETING BLOCKS OF TEXT <P24>. That sure sounds terrible. Why would anyone want to delete something they just wrote? Who knows? Maybe you decided that particular bit of literary genius stinks and wish to try over.

Anyhow, the first thing to do is move the CURSOR to the first letter of the first word you wish to get rid of. Next, exert continuous pressure with one of your fingers on the [OPTION] key, while exerting yourself push the [B] key. Notice down in the lower screen where it used to say "INSERT MODE" that it now reads "BEGINNING MARKED".

If you have a lot of text to eliminate hold down the [CONTROL] key and push down on the down arrow to move the CURSOR down. At this time you will notice that as the CURSOR moves down everything is now in inverse print. If you go too far don't worry, because all you need to do is back up the CURSOR with your arrow key to the point you intended to stop. Notice how the inverse disappeared one letter at a time. Now you have all of your text to be deleted in inverse characters, right?

OK, next key strokes are [OPTION] and [DELETE/BK SP]. Please take note of the little message down in the lower left hand side of your screen. It says something like DELETE BLOCK Y/N? This is the biggy, because when you gently touch that [Y] key, ZAP - it is gone! But not forgotten, at least not by the computer.

"Holy cow," you shriek. "What have I done?"

Never fear folks, just place the CURSOR where you want to replace all that lost composure (pardon the pun) and gently stroke [OPTION] and [X] and as in X marks the spot, there is your lost composure (OOPS did it again). It was saved in the failsafe buffer.

Now let's see, oh yes. dumping everything from the cursor to the end of the file. Let us suppose that you decided that your talents as a writer were in the toilet that day and what you had was garbage. Try this, push the [SELECT] and [DELETE][BK SP] keys. It is gone, and I mean gone - no failsafe buffet, nothing -

just gone! (For you 130XE people, this is just to the end of whatever memory bank you are in at the moment.)

Other Text Block Options

By the way, before I forget, this is the same procedure for pasting a block of text <P25>. Just place the cursor where you wish the first word of the deleted text to pop up at, and push down on the [OPTION] and [X] keys. Nifty, huh? (Attention 130XE users - you can go from memory bank to memory bank with pasting operations, just remember your failsafe buffer is 8K of memory. First get the block deleted, next press [START] and [B] to get to the memory bank of choice and then hit [OPTION] and [X].)

Suppose that you were not writing garbage - suppose that you were so proud of what you wrote that you wanted to see it twice! That is called DUPLICATING BLOCKS OF TEXT <P25>.

The main difference between deleting or pasting is that instead of punching [OPTION] & [DELETE][BK SP] at the end of the highlighted text you push [OPTION] & [E]. This leaves the highlighted text in place, removes the highlights and places a copy of the text in the failsafe buffet, where it sits until you place the CURSOR and strike [OPTION] & [X]. (130XE devotees can carry this action across memory banks as well.)

Another nice editing feature is the ability to take a random list of words and alphabetize them. Lets suppose you have a list of jobs that you have to do and you prioritize this list.

As it sets there on the monitor with the first priority job being the fifth one on the list and the second being the last, why not try this and they will be shuffled into numerical sequence. Press [OPTION] and [B], do the highlighting bit and then press [OPTION] and [A], a few seconds later — voila — they are in their new and proper sequence and you can run off to the garage knowing that the attic is next to be cleaned (or ignored because you discovered that the TV slated for the garage sale works).

To count the number of words in a file just type [OPTION] & [W], and down in the lower left, the program will tell you how many words are in your document. Go ahead and count them!

Search And Replace

Let's suppose that you have written an article about winter. Let's suppose you are now bored with winter. Let's suppose you have a magic wand (an Atari computer), and you are going to change winter into

spring. Well this is kind of silly I know, but how else can I lead into SEARCH AND REPLACE (S/R) <P26-27>.

According to the manual, you can search out and replace 21 characters at a time (130XE owners can cross memory banks with this procedure). Here is how you get rid of Old Man Winter. First, type [SELECT][T]. This puts the cursor at the top of your article. Next, hit [START][S].

Notice down in the lower left where INSERT MODE was? It now reads SEARCH STRING. Type in "Old Man Winter" without the quotes (or whatever it is you wish to replace with no more than 20 characters) and hit [RETURN]. Now type [START][R], and notice the lower left of the monitor screen reads REPLACE STRING. This is our chance to turn winter into spring, by typing in "Spring" and [RETURN].

You notice that the cursor has moved all by itself to the first occurrence of Old Man Winter! Press [SELECT][S] and the lower left hand side of the screen reads STRING FOUND. Now for final revenge! Your fingers pounce on the [SELECT][R] keys and Old Man Winter is replaced by Spring. Wouldn't it be great if we could do that with the weather?

What I have described is the case by case method of SEARCH and REPLACE. Another method is to simply hunt out the offending word(s) and use [SELECT][R].

When you are all done with your search and want to make sure that Old Man Winter no longer exists in your file try using [SELECT][U]. This will take you back up through your file looking for a last time for any appearances of Old Man Winter so you can enjoy your newly created Spring.

This may seem a bit tedious, so let's make it easy. Once you have gone through all of the first steps of defining the string and you have typed [SELECT][S] try [OPTION][G]. From where the cursors setting to the end of the document, Old Man Winter will be replaced with Spring and down in the lower left screen you will see the following: GLOBAL SEARCH/REPLACE COMPLETE. (130XE owners — this will not cross banks. However, you only need to find the first occurrence in each bank, place the cursor as described and strike [OPTION][G] in each bank.)

Here are a few little additional items to help with search and replace. If you have a word like NOT and you wish to replace it, be aware of the word. Those letters appear in other words like NOTHING for instance. So, when you use S/R you must put a [SPACE] in front of and behind the word so that the S/R function of AW+ does not

get confused and change parts of a different word. Another function is that S/R will change control characters for you, and you can use [?] as a wildcard.

One final note: you can use S/R as a search and delete function by typing [RETURN] instead of [START][R].

Let's wind up the section on editing with some of those odd ball things that can save you some time and frustration. Like merging a file <P27>. As editor of the CACE newsletter, I have to merge files a lot and this function has been of great help.

Merging Files Together <P27>

If you've seen the three column format of the CACE newsletter, you will notice that two or more unrelated items will fall in a single column. Since the number of contributing reporters has been growing, each article is saved on a disk as a separate file. But, when printed they must be merged to fill a column. This is what I do to accomplish that feat:

First I load a file, and then I set the cursor to the end of the file with [SELECT][B]. Now, I press [OPTION][L]. In the lower right of the screen is the prompt: FILE TO MERGE. I type in the file name (if the file is on Drive 2, then I preface the file name with D2:) and type [RETURN].

That's it folks, I have just merged a file. When merging files, please make sure that you have enough free memory to make the merge (130XE owners—you cannot merge across memory banks).

Saving Blocks of Text

Another handy little option is the ability to save a portion of a file <P28>. I don't know if you are like me, but on occasion I get to writing on the old 800 and forget that the file lengths are not as long as they are on the 130XE.

Next thing I know, I am stuck with a file that is too long to save. DO NOT PANIC! (I can say that because I did once, but now I don't.) Place the cursor at the beginning of the part you wish to save.

Remember when we talked about deleting blocks of text? Remember that command [OPTION][B]? Remember how you arrowed down using [CONTROL][-]? Well that is what you do here until you have got enough highlighted text to make a shorter file.

"OK," you say, "what's next?" Try [OPTION][S]! Check out the lower right hand side of your screen, it should read — FILE TO SAVE.

What I usually do is give my file a name and number (i.e., D2:TOOLONG 1). Then I hit [OPTION][DELETE][BK SP] and get rid of the saved portion. I repeat this process and save the next part as D2:TOOLONG2 and so forth, until my long file is all saved and I can recall it in logical sequence for hardcopy.

Additional Remarks

For 130XE people, here are a couple of notes: if you load up a memory bank press [START][B] to go to the next bank.—you have two more available. It will save across banks and will reload to the same banks.

Let's say you need to add something to

the middle of your file. Bank I is full, and that is where you need to place the additional information.

Try using [OPTION][F]. This command redistributes the file evenly across the three memory banks and leaves you space to make the insertion in Bank 1.

Put the cursor at the top of Bank 1 before you do it so you know where the cursor is when you are done. The first time I tried it, I had the cursor at the end of Bank 1 and finally found it at the end of Bank 3. Nothing serious, but it was unnerving.

Saving Text as ASCII

This next bit is kind of fun and interesting and, for serious writers or programmers, of great help. You can save your files in ASCII format by going to the Main Menu (by typing an [ESC]), pushing [CONTROL][S] and saving your file in the usual manner.

Now you can send this file via your modem to a friend with a different computer or a different word-processor, and they can receive it without the AW+ control characters. This will save them some time trying to unscramble your file. Make sure that you have deleted all AW+ global print format commands. These commands will screw up your file on any non-Atari program.

Programmers should save their files in the same manner, then ENTER the file into their compiler. Be sure to LIST your program. These files cannot be SAVED.

Primarily AW+ files are compatible with any DOS 2.0 and 2.5 formatted word-processors. So be it with editing. Bye for now.

XL/XE Software Roundup

Arkanoid — Hit Squad
Tape £2.99
Re-release of the Breakout clone.
Astro Droid — Byte Back
Tape £2.99
Re-release.
Hawkquest — Red Rat
Tape £9.95; Disk £19.95
A Xevious type shoot 'em up classic.

Head Over Heels — Hit Squad
Tape £2.99
Finally the Spectrum classic arrived in budget form.
Good graphics, fast game-play, a hit.
Kick Off — Anco
Tape £9.95; Disk £14.95

BaPAUG Stock Clearance

The BaPAUG have the following 8Bit items for sale. All are brand new. Quantities for each item are shown within the brackets. As you can see there are not many, so I would advise telephoning (0202) 677895 to reserve any items you want. Please note this is a once only offer, after these items have gone it is very unlikely that the BaPAUG will purchase Atari 8 bit products in bulk for re-sale through 8:16.

Item #1: BASIC XL (2) £24.95
Item #2: SpartaDOS Construction Set (1) £19.95
Item #3: The Designers Pencil (1) £3.50
Item #4: AtariWriter Plus (12) £8.50
Item #5: SpartaDOS X (1) £35.00

All prices include P&P. Once you have reserved an item please make out cheques / postal orders to BaPAUG and forward them to Colin Hunt, 248 Wimborne Rd., Oakdale, Poole, Dorset BH15 3EF confirming the item requested.

Note: We will only hold reserved items for 10 days.

Subscribe to 8:16

Only £3.80 for 4 issues

Cheques payable to BaPAUG

248 Wimborne Road, Oakdale, Poole, Dorset BH15 3EF

Lynx Review

Gates Of Zendocon

Atari - £24.95

Gates of Zendocon is one of the first games released for the Atari hand held games machine - The Lynx. The droids of your nemesis, the evil spider Zendocon have captured you and you have been sentenced to wander Zendocon's web of deadly universes. Thus begins your battle to fight through the 50+ interconnecting universes to win a duel to the death with the evil spider herself.

Gates of Zendocon is a single player horizontally scrolling shoot 'em up. In order to survive each of the universes you are supplied with a ship, located on the landing platform of the alien bases. This ship has two weapon systems, Neutrino Laser and Photon Bombs, and a destructor shield. The weapons have unlimited life, though the shield has a tendency to over-heat. The ship can survive three hits. The first destroys your shield, the second your laser and finally the third destroys one of your five lives.

As well as all the enemy life forms and armour that you have to destroy there are also friendly aliens. These are aliens held as slaves by Zendocon's minions and they will help you if you can free them. Once freed they follow your ship, aiding you when possible with their strange weapons. The four friendly aliens are; *Flying Eye-ball*, which hovers below your ship shooting fireballs. *Death Arising* which shoots laser beams straight up. *Cosmic Destroyer* which seeks out and crashes into aliens and *Sonic Dart*, the most powerful weapon, which emits deadly sonar waves. These alien follows stay with your ship until it is destroyed - so once you have their help be extra careful.

Each universe consist of a landing platform (where you start) and transporter gateways to other universes. In some universes you can fly past the first gate and thus pass through a second, or even a third, which transport you to different universes. When you pass through a transporter to a new base you have the option of landing on the upper platform and swapping over to the ship located on the lower platform. If your ship has taken any damage this is a necessity. Don't worry about the alien follows, they will swap ships as well.

When you start the game you are presented with a few options, one of these being hard or easy. As far as I can tell these options appear to be the wrong way round. The hard option produces a game which it

far easier to play, and gives you higher scores. Other options allow you to turn the music on and off and pause the game (a feature built into the Lynx). Upon entering the first universe you are shown the evil spider Zendocon, accompanied by a evil laughter through the headphones - press A or B to get away from this nightmare. Talking off headphones, I highly recommend playing this game using them as the sound effect, music is far superior than through the Lynx's built in speaker.

Lets go through the first eleven universes. By the way, universes are named by four letter words which can be entered at the start (once know) thus allowing you to continue where you last finished.

BASE: This is the first universe and consists of spaceships and a snake like creature that flows up and down on the screen. The best way to deal with these is continuous fire on the snake, while looking out for the spaceships.

ZYBX: Firstly, the landing platform at this base is lightning protected, with a little message daring you to try. If your ship took no damage don't bother, if it did you'll going to have to. Hover above the lightning, wait for it to stop and drop down quickly. The universe consists of red pulsating crosses which fire deadly missile, and I mean deadly as they fly straight through your shield.

XRXS: This universe consist of a green crystallized life form that grows across the screen, continually splitting and growing. Under no circumstances should you try and fly through or below this as disconnected section fall, and if your underneath or in the way, do I need to explain. To survive this screen go to the top and concentrate on using your Photon Bombs. At the end of this universe you will find a single life form under the transporter gate. Kill this and free an alien follower.

ANEX: This universe consists of a mysterious looking life form with one eye. Destroy the eye to kill the alien. There is one alien between the base and the first gateway, if you feel you can survive two more then you can attempt to reach the next gateway - a feat I have yet to accomplish after many attempts. Warning each new alien is bigger than the previous.

NEAT: Of the early universe this is the easiest, consisting of walls.. Fly straight through with continuous fire to clear a path.

YARR: This base is also lightning protected, so once again only land if you

have too. This universe is more crowded than the previous five, consisting of tentacles hooked to the bottom, flying missile platforms and an alien I can only call the green blob. The secret here is to go close to the top, below the missile platforms and to use continuous fire. If you have the follower from XRXS, lifes a lot easier. Completion of this universe will sometimes reward you with another follower.

EYES: The aliens within this universe consist of eyes paired together by a U shaped hook. Destroying one eyes causes the other to shoot off at high speed - either up or down depending which one from the pair it is. To maximise your score use both your laser, to shoot the upper eyes and your Photon bombs to catch the pair. If you shoot the hook both eyes shoot of. If you pass the first gateway and succeed in reaching the second you will be transported to BARE, by-passing the next three universes.

NYXX: This universe contains three alien types and is very straight forward to complete. Stay clear of the spiral hooks as they are not destroyed by your laser. Instead concentrate upon the huge green blobs and the eyes from the previous universe.

ZYRB: Within this universe you have to battle through a swarm of giant wasps and past missiles from missile bases that appear along the bottom. For maximum score go up from the base and destroy as many wasps as you can. As soon as the missiles appear move down in order that you can destroy the bases - this being easier than avoiding the missiles.

SRYX: The biggest threat within this universe is not the aliens, which are bouncing single eyes firing forward, but the rocks which when hit break into two smaller rocks (and then two more) and fly off in different direction - just like Asteroids. You can if you wish complete this universe without firing a single shot and claim the 50,000 points at the end for completing it. Alternatively you can slug it out and hope you don't get hit by to many stray fragments.

BARE: One huge snake. Hit the eye to destroy.

Overall Gates Of Zendocon is an excellent shoot 'em up. I have so far visited about a third of the universes while playing the game for several weeks, including non stop Burnley to Basingstoke while a passenger in a colleagues car. With games lasting over half an hour (when starting at BASE) you cannot complain about the value for money. Highly recommended.

BaPAUG 8 bit PD Library

NEW LOWER PRICES

(All disks XL/XE - SS/SD unless otherwise stated)

- Disk 1: Pokey Player III**£2.00
Pokey Player 'player' with graphic demo. Mouse controlled.
- Disk 2: The BaPAUG Slide Show**£2.00
Auto running AtariArtist slide demo. Slide show routine compiled Turbo BASIC.
.PIC Files: WATERS, STRIESAN, SO, MAY, MADMAX, SYD, HOLES, ASSASIN, DAVE, REED, HEROES, FINAL, MARKET, FLOYD, SUPPER, EYES, HEAD, ELOISE, ROBOTRIX, ENTER, ENDGAME, WISH, OTHER
- Disk 3: Digital Sound Demo**£2.00
Digitized music and scrolling graphics.
- Disk 4: The 8-bit Mouse PD Disk V2.0***£3.00
Disk 4m: with Printed Manual£4.00
Complete system for using the ST mouse and joystick with your 8 bit. Also includes a fully featured drawing program for the budding artist which also supports the touch tablet.
- Disk 5: The Softkeys Source Disk (SS/SD)** ...£2.00
Disk 5m: with Printed Manual£3.00
All you need to set up function keys on your Atari 8 bit. Includes source code and compiled object.
- Disk 6: The 8 bit Reference Manual**
(DS/eD)£3.00
A complete manual describing all the features of the Atari 8 bit range of computers by Bob DuHamel. Disk requires 1050 or XF551. Sections cover the use of the different device handlers, the disk operating system, system interrupts, floating point arithmetic, the hardware chips, display lists, player missile graphics, sound, the joystick ports, serial I/O and the changes within the XL/XE models.
- Disk 7: The Turbo BASIC Sector Editor*** £3.00
Disk 7m: with Printed Manual£4.00
A full feature sector editor written in Turbo BASIC by Gavan Moran.
- Disk 8: The Turbo BASIC Character Editor***£3.00
Disk 8m: with Printed Manual£3.50
Another superb utility written by Gavan Moran using Turbo BASIC. As well as comprehensive range of options this editor also uses a pop-up menu system for user input.
- Disk 10: Music Disk #1 (SS/SD)**£2.00
A collection of music files all written in Atari BASIC.
- Disk 11: Music Disk #2 (SS/SD)**£2.00
Another collection of music files all written in Atari BASIC.
- Disk 12: Device Handlers (SS/SD)**£2.00
Several device handlers including a 80 column display, 1020 80 column lister, multi-mouse and a null handler. Complete with documentation and source code.
- Disk 13: Just For Fun (SS/SD)**£2.00
Now finally available! Pride of place is FUNFACE, a real neat demo from Germany. Also includes thunder demo, two games of solitaire - SOLPEGS and SOLCARDS, VAMPBRAT, MINE and an excellent game called SUPERVER.BAS.
- Disk 14: The Turbo Collection**£2.00
A collection of Turbo BASIC utilities and demos. Includes a program that will recover crashed Turbo BASIC saved files.
- Disk 15: Technicolour Dream 1 (SS/SD)**£2.00
Slide show of pictures drawn using Red Rat's Technicolour Dream.
Pictures: CHURCH, POD, ROME, RAIN, CREST, ZSOFT, USGOLD, PARROT
- Disk 16: Technicolour Dream 2 (SS/SD)**£2.00
More pictures drawn using Red Rat's Technicolour Dream.
Pictures: ROCKS, MAN, PIGGY, HOUSE, HAYWAIN, GIRLFACE, DROID, SPHYNX
- Disk 17: Picture Show 1 (SS/SD)**£2.00
Micro Painter slide show.
Pictures: MICROMAN, ESCHER, GARFIELD, TVKIWI, SNOWMAN, HOUSE, FACE, EINSTEIN, SAILBOAT, TIGER
- Disk 19: Technicolour Dream 3 (SS/SD)**£2.00
Even more pictures drawn using Red Rat's Technicolour Dream.
Pictures: MTV, DESK, SANDRINE, PERLALU1, ARM, WINTER, DRAGON, ATARI
- Disk 20: Yet Another 80-Column Device Handler* (DS/SD)**£3.00
This disk contains Simon Trew's 80-column device handler. This handler is far superior to the one on disk 15. Complete with documentation, source code, character editor and some additional programs.
- Disk 21: Parrot II Sound Demo (SS/SD)**£1.50
SPECIAL ON THIS DISC: Send SSAE + disk and we will supply this disk FREE
Excellent disk demonstrating sounds sampled using the Parrot II sound digitiser. Demo includes two modes: Single sound tunes, select the sample then select the tune, and sequences consisting of several different sampled sounds.

Note: All disk titles marked with a * we give the author(s) a share of 20% of the disk selling price.

Cheques / Postal Orders should be made payable to 'BaPAUG' and sent, with order to:
BaPAUG, 248 Wimborne Road, Oakdale, Poole, Dorset BH15 3EF

BaPAUG ST PD Library

ST PD disk only £2 each
(All disks single sided unless otherwise stated)

- ST1: AniST**
Originally sold as Aegis Animator for £80, Ani-ST must now be the Ultimate PD bargain.
Files: ANIST.DOC, ANIST.PRG, CPSCR.TTP, EEL.STR, FISHLOOP.STR, FLUER.SCR, FRAGS.SCR, LITTLESEA.SCR, PLANTLOO.STR, REFF1.NEO, TETRA.STR
- ST2: PageStream Version 1.8 Demo DS**
Complete working version with all functions operational, except the print command, which, although operational adds diagonally across all pages 'Pagestream Demo Version'.
Files & <Folders>: <PSDOCS>, <PSFONTS>, <PSOTHER>, <PSPICS>, <PSTEXT>, PGSTREAM.DAT, PGSTREAM.INF, PGSTREAM.RSC, PGSTREAM.PRG
- ST3: Xformer Version 2.55 DS**
The only Atari 8 Bit emulator for the ST, version 2.55 being the final instalment. Works on a 520ST and faster than previous releases, two of which are also included on the disk.
Folders: XFORMER1.1, XFORMER2.1, XFORMER2.55
- ST4: GDOS 9 Pin Printer Fonts DS**
A selection of 9 pin printer and high resolution screen fonts in a selection of sizes for use with GDOS applications. Documentation also explains how to install into Timeworks DTP and Wordup.
Fonts included: Arbor Low, Ashbourne, Bakewell, Chatsworth, Crich, Dale Abbey, Derwent, Graphic, Haddon, High Tor, Kinder, Monsal, Snake Pass, Wye
- ST5: Double Click PD Sampler Disk**
This disk contains the demonstration version of the DC DESKTOP GEM desktop enhancement package from Double Click Software.
Files and <Folders>: <DCFORMAT.302>, <DCSHOWIT>, <DCXTRACT>, <DC_CLOCK>, <DESKEY>, <MYSTIC.100>, <STUFFER.090>, DCCONFIG.PRG, DCDSKDRP.PRG, DCDSKTOP.PRG, DCICE.PRG, DCINSTALL.PRG, DCSHOARC.PRG, DCSPPOOL.PRG, DESKTOP.INF, READ.ME, READTHIS.NOW, SAMPLE.PRG, SAMPLE.S
- ST6: ST Writer Version 3.8 DS**
Latest release of the ST version of AtariWriter. Also includes German and Spanish versions, Magniwriter which shows only 40 characters per line in high resolution, and a utility that converts files from / to ST Writer and 1st Word, Word Perfect and Word Writer.
Files & <Folders>: <ARTICLES>, <DOCUMENT>, <GERMAN>, <MGWRITER>, <PRINTER>, <SPANISH>, CHANGES.38, CHANGES3.DOC, CONFIG.TOS, README.DOC, SIFT.PRG, STWELENG.RSC, STWMAN.PRN, STWRITER.PRG, XYZZY.DAT, XYZZY.TXT
- ST7: Calamus Fonts #1**
Disk contains selection of fonts, with documentation, suitable for the WP/DTP program Calamus.
Fonts included: Broadway, Bullwinkel, Burlington, Calfont2, Calfont4, Calligraphy, Calamus Macro, Camelot, Chancery, Classical, Cupertino, Decadent, Oakville
- ST8: Games #1 DS**
This disk contains a selection of games for use in low or medium resolution. The first is CHECKERS, which as well as being stand alone also contains a desk accessory version. COREWARS is the ultimate programmers wargame - this disk contains the original and STOS version. DALEKS is a game for the thinkers among you, along with several versions of Solitaire and a hangman game based upon song titles. Finally this disk contains an excellent PACMAN clone called HACMAN.
- ST9: Astrology & Astronomy**
A selection of programs based upon the Astrology & Astronomy theme. All .PRG files are self extracting ARC's.
Files: ASTRO.PRG, ASTROMY.PRG, BAP_READ.ME, BIO.PRG, NAGELBIO.PRG, SAT.PRG, SUNMOON.PRG
- ST10: Games #2: Hero & Space Wars**
This disk contains two excellent games. Hero is a dungeon and dragons style shareware game written by the Winslow Programming Fund. Paying the requested \$10 fee should get you the HERO Dungeon Construction Set, and a hint book for the dungeon supplied. Space Wars is a two player game based upon the game of the same name that was in the arcades when Asteroids came out!
- ST11: Your Second Manual Version 4**
The last PD version of Your Second Manual. Disk also includes Desk Accessory to aid viewing.
Files: LMACC.ACC, LMACC.TXT, YR_2_MAN.V04
- ST12: CPM Emulator**
Atari's CPM Emulator for the ST. If your into CPM then you need this disk!
- ST13: Calamus Fonts #2**
More fonts for use with Calamus
Font folder names: CURSIV, FACADE, FURA, GAUDY, GILLA, GILLY, GREENY, HEBREWY, LEDGER, LUBBY2, MANVILLE, MOUSE_
- ST14: Spectrum Picture Show #1**
A selection of pictures produced using Spectrum 512.
Files & <Folders>: <AUTO>, SPSLIDE8.DOC, SPSLIDE8.PRG, BLADE1.SPC, BLADE2.SPC, KNIGHT.SPC, SHIP.SPC, STARTREK.SPC, STORMTRP.SPC, VENUS.SPC
- ST15: Calamus Fonts #3**
More fonts for use with Calamus.
Font folder names: MVG_DEM4, PEIGNOY, REVUE, SAVINGS, SILICY, SOUVENIR, SPOKANE, STEELY, STUDY

Midimaster SIO Driver

By Graham Broomfield

After buying MIDIMASTER from Page 6 and using it with my PSS680 keyboard, I was very disappointed with the software supplied. Its main disability is its inability to change what is going on. I therefore set about writing my own MIDI software for my 130XL (a 800XL upgraded to behave like a 130XE) using the MIDI interface supplied.

I first had to discover how the interface worked. This was achieved by taking the molex connector apart in order to check the pins used. With this information, the 130XE reference manual, MAC/65, a lot of spare time and many aborted attempts, the secrets were finally revealed.

The two main things I had to find was what rate the SIO timers had to be set to and how the interface worked.

MIDI transfers at a clock rate of 31.25KHz, thus with my oscilloscope looking at the output of the interface and loading different values into timer 3, I was able to find the correct divider to set up this frequency. The number I found was 21 and I still do not know how this relates to the way in which the main clock of 1.79MHz is divided down to 31.25KHz.

The interface is very simple, the main problem was how it worked, and why my first routines would not send any data out. Looking at the pins I found that 'motor drive' was used and then it came to me - the motor line had to be set first before data could be sent.

The next step was to write a machine code routine that can be linked with Turbo BASIC (my favourite language on the 8 bit). The object code for the final version of this routine is produced by Listing One. Type in the program and save it to disk before running. If everything is OK a file call SIO.OBJ will be created and saved to disk, if not re-check the listing.

Listing Two shows how to load and use the object file within your own Turbo BASIC programs. The routine will send MIDI codes and "note" data to your Keyboard, stored in BUFFER\$(lines 60 to 120 is the MIDI codes). The number of bytes to be sent must be stored in COUNT. BAUD is the value that will be loaded into the two Timers, this sets the rate at which data is shifted out of the SIO PORT (note only Timer 3 is set and Timer 4 is always set to zero).

Before the machine code routine can be called you must turn the motor line on, and then off when the routine returns to Turbo BASIC. This is done on lines 160 and 190. Also you must test to see if the data has been sent to the keyboard (line 180 does this). This is required as the machine code routine returns to BASIC after initialization, leaving a background task which sends the data out. You have to check that this task has finished before re-initializing. The program will play the drums on a PSS680 Keyboard. At present I still have a bug which hangs up the computer from time to time. I am still looking into this and I believe it has something to do with the way Turbo BASIC works, especially with the handling of interrupts.

In future issues of 8:16 I hope to have developed a full MIDI program with all the bells and whistle that ST uses get.

```

0 DIM A(10):DIM B(10)
1 OPEN #1,8,0,"D:SIO.OBJ"
2 TRAP 6:RESTORE 10
3 FOR A=1 TO 148
4   READ B:PUT #1,B
5 NEXT A:CLOSE #1:STOP
6 ? "FILE ERROR":END
10 DATA 255,255,0,177,141,177,76,6,177
,76,80,177,104,104,104,141,142,177,104
,133,51,104,133,50,104
20 DATA 141,146,177,104,141,145,177,17
3,142,177,141,4,210,169,0,141,6,210,17
3,12,2,141,147,177,173
30 DATA 13,2,141,148,177,169,177,141,1
3,2,169,3,141,12,2,169,255,141,149,177
,169,0,141,0,3
40 DATA 32,23,236,160,0,177,50,141,13,
210,96,152,72,230,50,208,2,230,51,206,
145,177,173,145,177
50 DATA 201,255,208,10,206,146,177,173
,146,177,201,255,240,11,160,0,177,50,1
41,13,210,104,168,104,64
60 DATA 169,0,141,149,177,173,147,177,
141,12,2,173,148,177,141,13,2,32,132,2
36,76,115,177

```

Listing One: Routine to produce SIO.OBJ

```

5 REM SIO/MIDI DEMO 6/8/90 G. B
10 POKE 106,$B0:REM MOVE RAM DOWN
20 GRAPHICS 0
30 BLOAD "D:SIO.OBJ":REM LOAD CODE
40 DIM BUFFER$(100):REM SPACE FOR DATA
50 BUFFER$(1,1)=" ":BUFFER$(20,20)=" "
60 BAUD = 21:REM BAUD RATE
70 BUFFER$(1,1)=CHR$(159):REM CHANNEL 15
NOTE ON
80 BUFFER$(2,2)=CHR$(64)
90 BUFFER$(3,3)=CHR$(64):REM VELOCITY ON
100 BUFFER$(4,4)=CHR$(159)
110 BUFFER$(5,5)=CHR$(54)
120 BUFFER$(6,6)=CHR$(64)
130 FOR N=38 TO 79
140 BUFFER$(2,2)=CHR$(N):REM KEY VALUE
150 COUNT=5
160 POKE $D302,52
170 X=USR($B100,BAUD,ADR(BUFFER$),COUNT)
180 IF PEEK($B195)=255 THEN GOTO 180
190 POKE $D302,60
200 BUFFER$(2,2)=CHR$(N+5)
210 COUNT=5
220 POKE $D302,52
330 X=USR($B100,BAUD,ADR(BUFFER$),COUNT)
340 IF PEEK($B195)=255 THEN GOTO 340
350 POKE $D302,60
360 FOR A=1 TO 400:NEXT A
370 NEXT N
380 GOTO 130

```

Listing Two: Sample player routine for PSS680 drums.

User Group File

Page 31

8:16

Issue 10

Local Groups (see map below)

Name : **Atari & Amiga & 16-Bit Club**
Contact : Michael Irish - 0707 327193
57 Rowans, Welwyn Garden City, Herts. AL7 1NZ
Notes : ST-Others, Meetings

Name : **Atari User Group Of Ireland**
Contact : Mike Casey
3 St. Kevins Park, Kilmacud, Co. Dublin
Notes : XL-ST-Others, Meetings, Newsletter, PD

Name : **Bloxwich Computer Club**
Contact : Edward Hunt - 0922 409291
29 Station Street, Bloxwich, Walsall, WS3 2PD
Notes : ST-Others, Meetings, Newsletter, PD

Name : **Bournemouth & Poole Atari User Group (BaPAUG)**
Contact : Ian Brooker; 163, Verity Crescent, Canford Heath,
Poole, Dorset, BH17 7TX
Meetings : 1st Friday every month at the Kinson Community
Centre, Pelhams, Millhams Lane, Kinson.
Newsletter : 8:16

Name : **Mid-Cornwall Co-Op Computer Club**
Contact : Mike Richards - 0726 890473
8 Victoria Road, Roche, St. Austell, Cornwall PL26 8JF
Notes : ST-Others, Meetings

Name : **Norwich User Group**
Contact : Ken Ward - 0603 661149
45 Coleburn Road, Lakenham, Norwich NR1 2NZ
Meetings : 1st Sunday every month. Contact Ken for time and
place.

Name : **South West ST User Group**
Contact : David Every
5 Turbill Gardens, Chaddlewood, Plympton, Plymouth,
Devon, PL7 3XF
Notes : XL-ST, Meetings, Newsletter, BBS, PD

Name : **Swindon Computer Club**
Contact : Mike Bird - 0793 539105
46 Eastcott Road, Swindon, Wilts. SN1 3LR
Notes : XL-ST-Others, Meetings, PD

Name : **The Friday Club**
Contact : Nicholas Bavington (0908) 612272
8 Byron Drive, Newport Pagnel, Bucks. MK16 8DX
Meetings: Every Friday at Ousedale School Physics Dept. OR a
members house.
Notes : XL-ST, Hardware & Software development.

Name : **Wigan Computer Club**
Contact : Alan Owen - 0942 212662
1 Lidgate Close, Wigan, Lancs. WN3 6HA
Notes : ST-Others, Meetings, Newsletter, PD

Special Interest Groups

Name : **Atari Portfolio User Group**
Address : 84 Cambridge Avenue, Gidea Park, Romford, Essex
RM2 6QU
Telephone : 0708 - 730764

Name : **GFA User Magazine**
Address : 186 Holland Street, Crewe, Cheshire CW1 3SJ
Telephone : 0270 - 256429

National Groups

Name : **Association of Atari User Groups**
Address : 45 Coleburn Road, Lakenham, Norwich NR1 2NZ
Telephone : 0603 - 661149

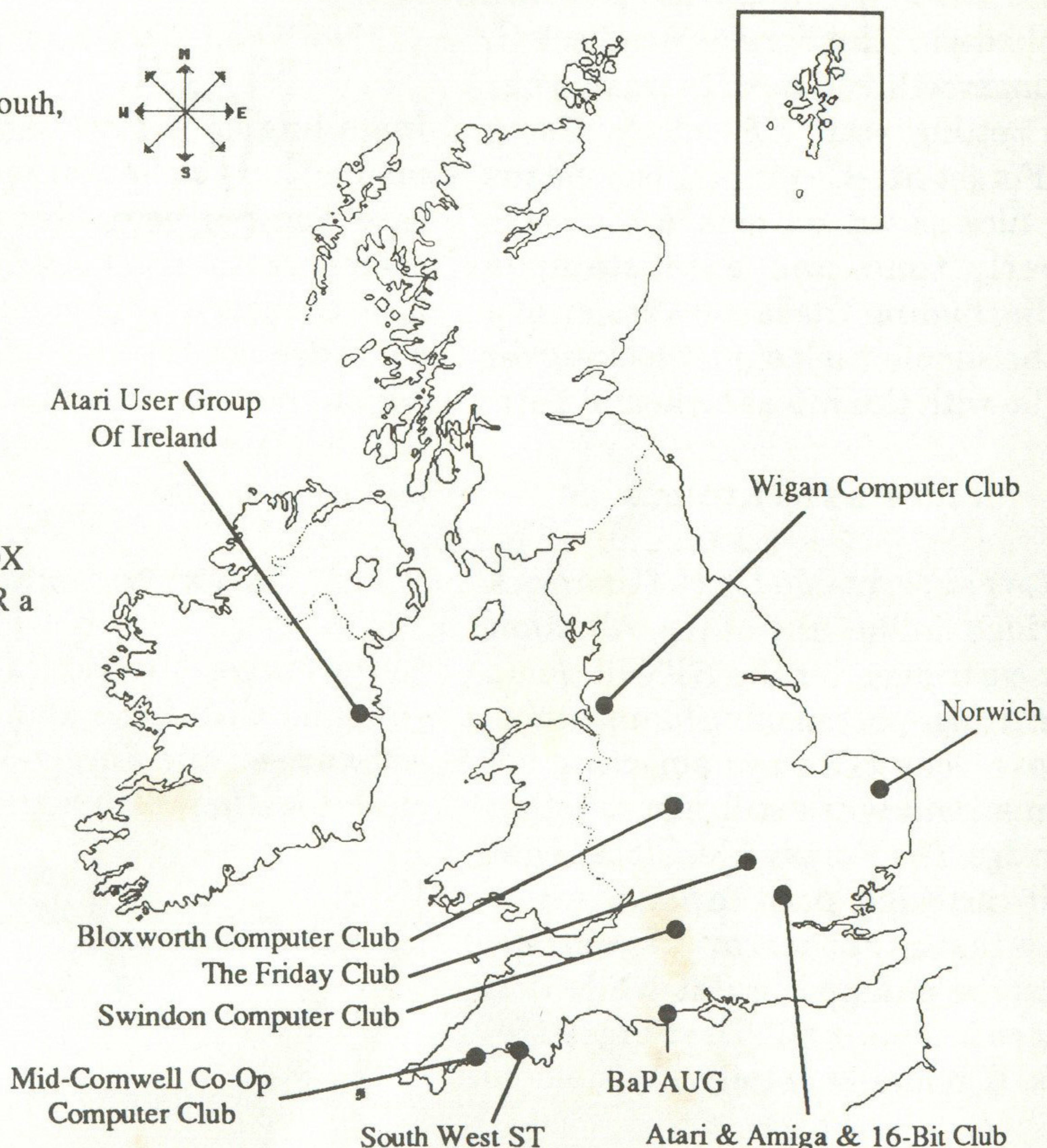
International Groups

Name : **Johannesburg Atari Computer Enthusiasts (JACE)**
Address : 2 Whitehall Street, Hurst Hill, Johannesburg, South
Africa, 2092

Name : **Pittsburgh Atari Computer Enthusiasts (PACE)**
Address : P.O. Box 13435, Pittsburgh, PA 15243, USA
BBS : (412) 571-0891

Notice To All User Groups

If you are a regular reader of 8:16 you will have noticed that the list of user groups within the *User Group File* has greatly changed. The groups now listed have been in contact with the BaPAUG within the last three months or have returned the Association of Atari User Groups survey to Ken Ward (Norwich User Group). If you run or belong to a user group that supports any of the Atari range of products and wish your group to be listed please forward details to the BaPAUG and AAUG.



Are you wasting your valuable time setting your ST's clock?

Frontier's Forget-Me-Clock II is the answer

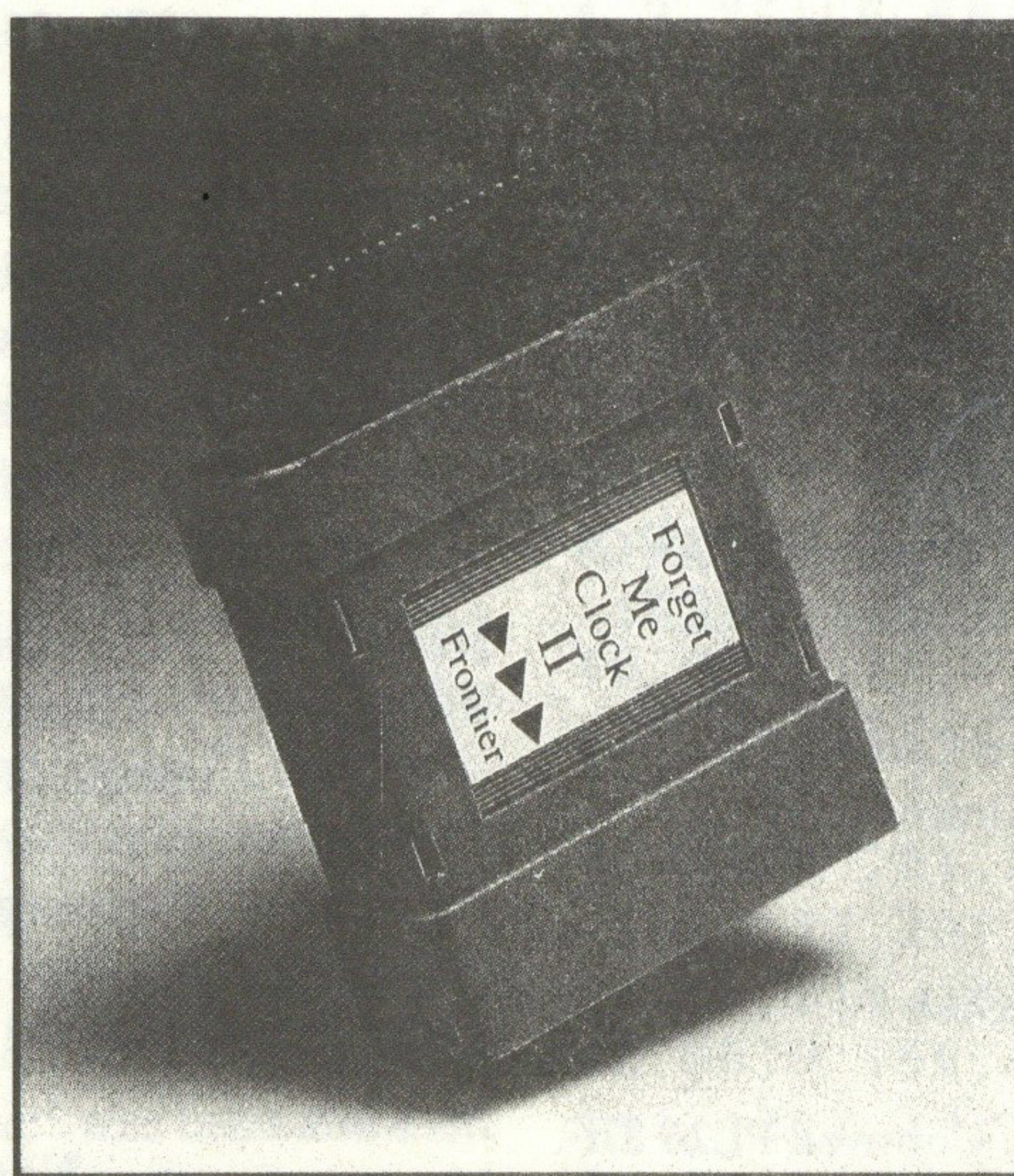
If you're one of those people who has an auto-run program annoyingly prompting you for the time and date every time you turn on your ST or ST^E or even worse, if you're one of those ST or ST^E users who doesn't set the system clock then Frontier's Forget-Me-Clock II cartridge is the solution you've been hoping for.

Just Plug It In And Go

With the Forget-Me-Clock II plugged into your ST or ST^E's cartridge port the system clock (used by the Control Panel) and keyboard clock will automatically be set at turn on or reset by a small auto-run program supplied with the Forget-Me-Clock II. No longer will you have to waste your time setting your ST's clock. Using the Forget-Me-Clock II also means that files saved on disk are always properly time and date stamped making finding the latest version of a file the simple task of just looking for the file with the latest time and date.

Full Pass Through

Frontier's Forget-Me-Clock II is a clock cartridge unlike any other. All other clock cartridges for the ST will tie up the cartridge port making it impossible to have your ST's system clock set automatically while still using another cartridge. The Forget-Me-Clock II has a full cartridge pass through which means that any other cartridge for the ST can be plugged into it while it is plugged into your ST. The Forget-Me-Clock II remains totally invisible so



Frontier's Forget-Me-Clock II Cartridge With Pass Through

that the other cartridge can be used normally, but it still automatically sets the system and keyboard clocks in your ST.

No Need To Open Your ST

Installing some clock cards for the ST means that you have to open your ST's case and pry computer chips out of their sockets. The Forget-Me-Clock II is a cartridge which plugs into the cartridge port on the side of your ST which means that its installation couldn't be simpler - you just plug it in and turn on your ST.

Software Included

Every Forget-Me-Clock II cartridge is supplied with time and date setting software for the Forget-Me-Clock II's clock together with a small auto-run

program which automatically sets your ST's system and keyboard clocks every time you turn on or reset your ST. Built into the setting software is the facility to stop the Forget-Me-Clock II's clock to save on battery life when the Forget-Me-Clock II is not being used.

Satisfaction Guaranteed

The Forget-Me-Clock II is supplied under Frontier's ten day money back guarantee, which means that if you don't like the Forget-Me-Clock II for any reason, you can return it for a full refund within ten days of purchase. The Forget-Me-Clock II has been designed to work with any model of ST whether it be ST, STM, STF, STFM, ST^E or Mega ST.

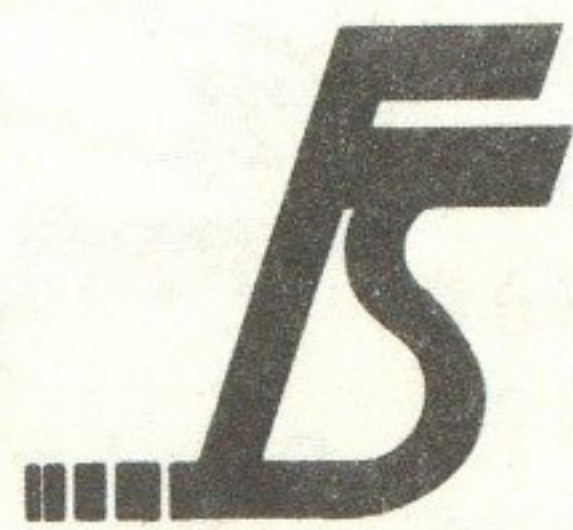
Two Year Guarantee

The Forget-Me-Clock II carries a full two year guarantee which includes the battery. Battery life has been tested to be many times the guarantee period. Frontier will supply replacement batteries outside of the guarantee period for a small charge.

Price

Forget-Me-Clock II Cartridge £24.99

Price includes VAT. Please add £1.15 for postage and packing to all orders under £50.00. Frontier accepts payment by Visa or Access. Price subject to change without notice. Goods subject to availability.



Frontier Software