8:16

The Alternative Atari Newsletter

ATARI

£1 February/March 1991 Issue 11

Introduction to C Programming

The Black Box Reviewed

Buying a MIDI Keyboard?

Then read our hints and tips within Music Scene: MIDI
Also includes review of the Yamaha PSS-790

Atarifest '90

Full show review inside

Plus: All the regular 8:16 columns
Notice Board / Software
Roundups / User Group File

Gralin International

11 Shillito Road, Parkstone, Poole, Dorset

Atari Light Pen

with
AtariGraphics
ROM
Only £25.00

Atari XE Games Console

(Boxed with PSU)
Built in Missile Command
Joystick and 2 extra ROMs
Limited Quantity- £40 + P&P

The Newsroom

DTP for the Atari 8 bit

Special Offer - £14.95

Clip art disks available soon

Custom Built Computers

(Keyboards only - No PSU)

1MByte XE	£150.00
80 Column 800XL/65XE	£65.00
80 Column 130XE	£75.00
Complete with wordprocessor and comms software on disk	
RAMrod 130XE	£90.00
Complete with OSNXL & Omnimon XL	
80 Column Ramrod 130XE	£120.00

Cassette Software

£2.99 per set

Set 1: Star Raiders, Chess, Asteroids, Missile Command, Centipede, Airstrike 2, Realsport Tenis, Chop Suey, Realsport Football, Basketball

Set 2: Typo Attack, Missile Command, Star Raiders, Centipede, Realsport Tenis, An Invitation To Programming

ANTIC Software

We have aquired a number of ANTIC software titles, along with back issue magazines and disks. List available with catalog, or send wants list.

Software	£ 7.99 ea
	5 for £ 34.99
	10 for £ 59.99
Magazines	£ 0.99 ea
Magazine with disk	£ 2.99 ea

Touch Tablet (No ROM)	£9.95
Touch Tablet+RAMBrandt	£14.95
1020 Printer/Plotter	£22.50
850 Interface	£49.99
Multi-Viewterm+Interface	£29.99

130XE Computers

(Boxed without PSUs)

Cheapest and easiest memory upgrade!

Only £35.00 + P&P

Parrot II Sound Sampler

Only £49.95

P&P / Handling Charges

Software:	£1.50 for orders under £50
Printers & Computers:	£3.50 each
Hardware:	£2.50 per order

There is no P&P or handling charge for software ordered with printers, computers or hardware.
Overseas orders extra. Please ask for quote.

ICD Products

Action! language cartridge with Toolkit disk	£39.99
Action! Toolkit disk	£14.99
Action! Runtime disk	£14.99
MAC/65 assembler cartridge with Toolkit disk	£39.99
MAC/65 Toolkit disk	£14.99
P:R: Connection	£49.99
Printer Connection	£29.99
R-Time 8 cartridge	£29.99
Rambo XL memory upgrade board (no RAM)	£19.99
SpartaDOS Toolkit disk	£14.99
SpartaDOS X cartridge	£39.99
US Doubler 1050 enhancement	£14.99
US Doubler with SpartaDOS Construction set	£29.99

The Flight Crew!

Colin Hunt .... Starfleet Controller
Graham Broomfield .... Ground Controller
Thomas Holzer .... VCS Controller

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

8:16 is produced by BaPAUG Publishing.

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

Letters and articles (and we urgently need them) 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, Apple Macintosh SE and Laserwriter Plus. The software used was TariTalk and AtariWriter Plus on the 8 bit. 1st Word Plus, Gem Kermit, TariTalk and ST Xformer II on the ST. Microsoft Word, MacPaint, MacDraw, Mac Kermit and Aldus Pagemaker Version 4 on the Macintosh.

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.

This issue is dedicated to Aaron and Tracy, whom I missed so much while I was in the states, and to my wife, who recently stated that "You spend more time turning your computers on than you do me!" Oops!

Gavan, please write.

Copy date for the next issue of 8:16 is 22nd March 1991.
Issue date is 26th April 1991

All help welcome!
My wife is called Tracy!

Contents

Features

□ Introduction To C Programming 8
The third and last article in our C series covers parameter passing and linking assembler with C.
By David Watson.

■ Turbo-Info 11
This issue contains parts 3 a4 and covers the Turbo-OS and the 65C816 registers.
By Chuck Steinman.

■ The Black Box 14
Review by Simon Trew.

■ Did You Know 16
A look at device handlers and a simple boot sector protection scheme.
By Derryck Croker.

Atarifest '90 18
Report from the premier US Atari show.
By Colin Hunt.

□ Music Scene: MIDI 20
Guide to buying a Keyboard and review of the Yamaha PSS-790.
By Paul Brookes.

■ AtariWriter Plus Explained for People Who Don't Want to Read the Book 24
By Jimmy Boyce.

The VCS Page 26
Latest news and reviews.
By Thomas Holzer.

Departments

Notice Board 4
BaPAUG News 5
Dear 8:16, ... 6
2600 Review 23
Atari 8-bit Software News 25
Lynx Software News 25
User Group File 27

Advertisers

Frontier Software 28
Gralin International 2, 23

■ 8 Bit Articles   ST Articles □

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

Notice Board

RealThings

With the final release of the ST version of **Deluxe Paint** (Electronic Arts), **RGB Studios** have started to release their AnimKits for the ST. The first volume is a sampler disk available for only £9.95 with the special form within Deluxe Paint.

The sampler contains four amazing animation sequences, including Owl Tower, where an Owl flys through a tower, across the screen and towards you. An added bonus is the fact that the sampler can be viewed without Deluxe Paint and contains all the parts required to build three of the animations with Deluxe Paint.

Volume 2 of the AnimKits is two full double sided disks of anim 'cell' sets for many different bird species. A future volume contains human life forms.

RGB Studios, Gables, Buxted, East
Sussex TN22 4PP, England

WANTED

Do you produce Atari related products?
Want some free advertisement?
Then send a press release or
information to 8:16 and we will pin it
up on the Notice Board

Mug Shot & Geography Tutor

Gralin International are pleased to announce that they are now the sole UK distributors of **Mug Shot** & **Geography Tutor**.

Mug Shot allows you to build an infinite number of mug shots (faces) in monochrome or colour on any ST. Creations are built from parts, with additional part data disks available. They can be saved to disk (4 file formats), and printed in four different sizes.

Geography Tutor is an electronic atlas for your ST computer. Maps and databases are provided for every country of the world. Countries can be viewed on different maps in either monochrome or medium resolution colour mode. Information includes facts on countries major religions, languages, name of capital, population and more.

Both programs are available from Gralin International for a retail price of £29.99 each.

Updated 80: Device Handler

Simon Trew is planning to write an updated 80: device handler (Disk 20 within the BaPAUG PD library), but would like some more feedback before starting. On the disk you will find a questionare ready to complete, so come on, help Simon improve on an already excellent program and send those questionaires to him.

New Dean Garraghty Catalog

We have recently received the latest copy of Dean's PD Catalog - Update #8. All the disks are £2.00 each, and cover topics from digitized music to games, demos to utilities. If you want a copy of the catalog send a SSAE to Dean Garraghty, 62 Thomson Avenue, Balby, Doncaster, DN4 0NU.

Lynx Price Drop

After disappointing sales and an expensive advertisement campaign Atari have reduced the price of the Lynx by £50 to £129.99. One reason quoted for the low sales is the small range of titles available. This should soon be resolved with the release of approximately 20 titles over the next couple of months. Already released are Paper Boy and Slime world.

BaPAUG News

September: AGM
Reviewed by Colin Hunt

This years AGM proved to be our most popular AGM to date with a massive 12 members turning up. It also turned out to be our longest, taking over the whole meeting. Following on from previous AGMs the meeting was held in a semi-formal manner with no fixed agenda being presented.

A brief description of the groups activities over the last year was given by the secretary. The highlights being the running of the MIDI show in June and the creation of three special interest groups (SIGs), one each for MIDI, hardware and software.

The treasurers report showed that the group was financially sound and it was proposed that the membership fees do not change from those set last year. Thus it was agreed that the full membership should remain at £8 and the junior membership at £4.80. It was also reported that groups membership had fallen over the year by 10 members, while the number of subscribers to 8:16 had risen by 41.

Next came the election of this years committee. To speed the process up it was proposed that last years committee retain their current positions as there were no challengers and no resinations. This proposal was accepted by the members present. It was also proposed that a new committee position be generated called "Public Relations Secretary" to promote the group locally and hopefully counteract the downward slide in membership. After a brief discussion on whether the position should hold voting rights the proposal was accepted with the post being a full committee member. This years committee is therefore as follows:

Chairman	Ian Brooker
Vice Chairman	Graham Broomfield
Secretary	Ian Brooker
Treasurer	Colin Hunt
PR Secretary	Paul Brookes
8:16 Editor	Colin Hunt
ST Librarian	Paul Brookes
8 bit Librarian	Colin Hunt

+ a representative from each active SIG.

It should be noted that the Secretary position is currently non voting as Ian is already the Chairman, and that the 8:16 Editor and Librarian positions are permanent non voting position. SIG representatives are treated as full members of the committe and are therefore allowed to vote.

After the (re)election of the committee a discussion was held about the success of the three SIGs formed during the year. The consensus of the discussion being that a lot more effort was required to ensure each SIGs existence. In order to take steps in this direction it was decided that Peter Graham would take over control of the MIDI SIG and that the Hardware SIG and Software SIG would be combined into one SIG called The Development SIG with Neil Horder in charge.

The final discussion of the evening was to decide upon the meeting topics for 1991. Unfortunately not enough topics were agreed upon to complete the year, leaving the 'To Be Announced' meetings open for future suggestions (which should be presented to the committee). It was also agreed that one (or two) members or a SIG would be responsible for organising each meeting and that all meetings should start at 7.30pm (not 'from 8.00pm' which has recently become the norm), with members arriving from 7.00pm.

The meetings schedule for 1991 is as follows:

Date	Topic	Organiser(s)
4th Jan 91	PD Night	Paul Brookes
1st Feb 91	Communications	Michael O'Brien
6th Mar 91	Graphics Competition	Neil Horder
5th Apr 91	MIDI Night	MIDI SIG
3rd May 91	SIG Night	Development SIG
7th Jun 91	General Games Night	Committee
5th Jul 91	Emulators	Colin Hunt
2nd Aug 91	Adventures	Ian Brooker
6th Sept 91	AGM	Committee
4th Oct 91	TBA	
1st Nov 91	SIG Night	TBA
6th Dec 91	TBA	

October: Sound Sampling
Reviewed by Paul Brookes

Mastersound and ST Replay were demonstrated on the ST. Replay for the 8 bit was also demonstrated by Graham Broomfield and sounded very impressive considering it is only a 4-bit system! Some general discussion about Shannon's sampling theorem was invoked by the prototype 16-bit sampler and sample player that Neil Holder and myself have cobbled together for a CD-quality demonstration, despite the complete lack of filters! The sampler used the Mega-ST's bus for data output, and the cartridge port for data input. A non-mega version (with filters) is in the pipeline, designed around the DMA port, but lack of experience with this port has halted progress for the time being.

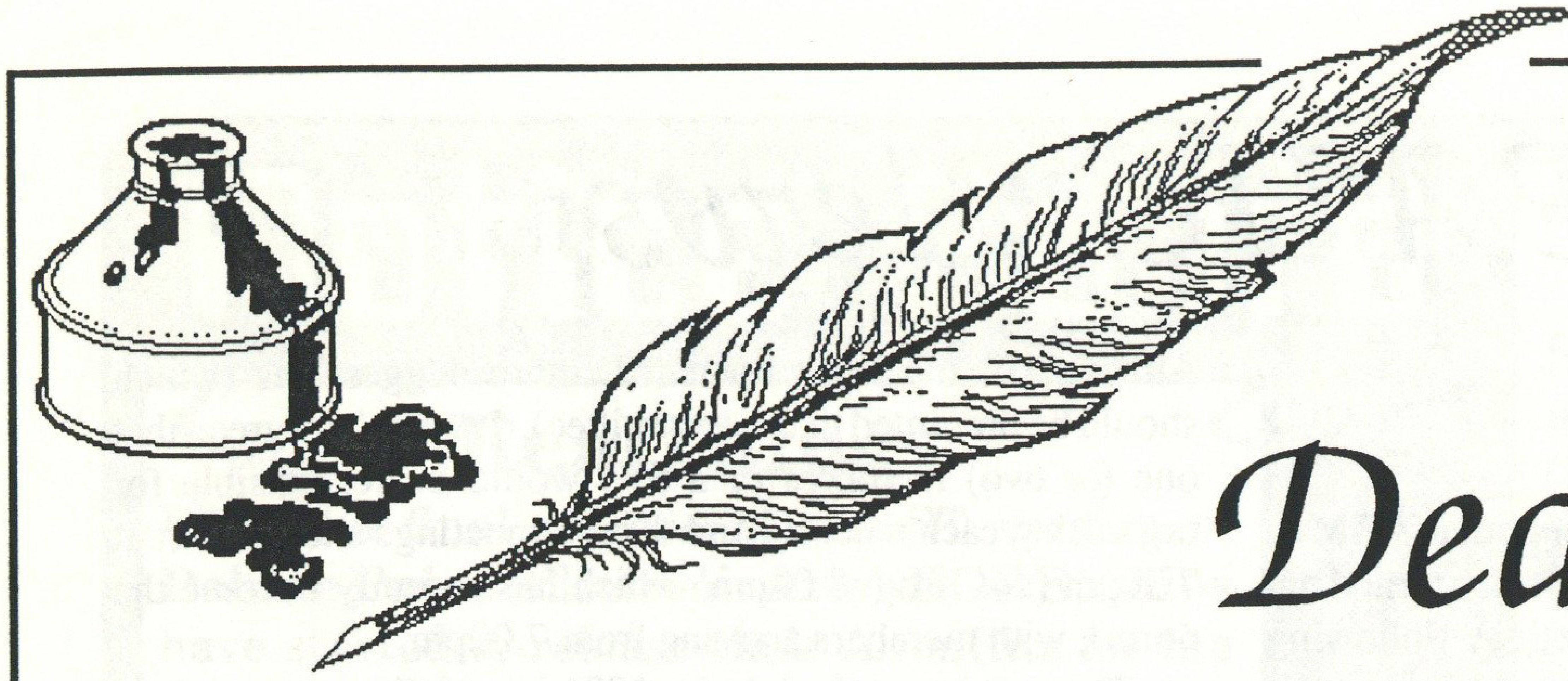
November: SIG Night
Reviewed by Paul Brrokes

The two SIG's were officially started (again - see September), with each SIG now having two people in charge. A list of names for each group is also available for those interested in participation.

It was also reported that the MIDI SIG had held one meeting at a members home, and are planning to repeat this at regular intervals.

The new contact names are:

Development SIG:	Paul Brookes (0202) 871822 & Neil Horder (0202) 512062
MIDI SIG:	Jack Bartley (0202) 471860 & Les Fobelets (0202) 577114



Dear 8:16,

Worthy Praise?

I think that 8:16 is the best Atari Club magazine I have read. Yes! Even better than 'Monitor'.

Bryan Zillwood
Southampton

Constructive Criticisms

After receiving issue 9 of 8:16, I thought I'd drop you a line and let you know what I thought. Firstly, let me thank you for the copy of DOS 2.5/Turbo Basic which leads me on nicely to my first comment. I think it would be a nice idea if you included a disk with each issue containing all relevant type in programs along with various bonus programs/PD utilities etc. To finance this, you could raise the subscription to £1.25 (£5.00 for four issues)

My second point is that the inclusion of a contact section and letters page would enhance the 'feel' of the magazine, and finally, I would recommend abandoning the practice of including Atari ST features as I can't see the publication attracting many ST only owners. The page would be better spent on the 8-bit machine.

Mike Blenkiron
Doncaster

How do you perceive the future progression of 8:16? Should we add a cover disk? Would you pay for it? Should it be an addition to the newsletter with more information / listings etc (using the News Reader PD disk system) or a means of distributing PD software? Let us know, and depending upon the response we'll see what we can do.

Just as some background information we were planning to include Disk 21 free with the last issue. When I originally did the costings it looked OK, but as issue 10 took so long to produce and as we got closing to the new release date I re-costed the disk option and the total blew the budget. The copy of 8:16 you are currently reading (assuming its 32 pages) costs the following to produce and despatch:

Paper:	*£ 0.24*
Printing:	*£ 0.69*
Envelope:	*£ 0.05*
Postage:	*£ 0.28*
Total —>	*£ 1.26*

As you can see the total cost is already greater than the subscription rate, the extra being covered by advertisement revenues, software sales and 8:16 sales at computer shows which do not incur P&P charges to us. The above also assumes that every issue produced is sold at 95 pence, this however is not the case. BaPAUG members get 8:16 for 50 pence, about 15 to 25 issues are distributed free to various companies and about 10 copies are exchanged for newsletters from other user groups. We also have to produce more copies than required, but I've managed to keep the excess down to a minimal quantity. The inclusion of a disk will increase the production and P&P costs. I already know this because of the subscription disk.

To answer Mike's specific question about support for the ST. No, I've no plans to drop the ST support. The same applies to the VCS, Lynx and any other Atari machine. 8:16 is primary the newsletter for the BaPAUG, which supports all the Atari machines. It is therefore logical that its newsletter does the same. In fact, the BaPAUG now has more ST members than 8 bit ones, but realizing that the subscription base is biased towards the 8 bit, 8:16 will continue to reflect those needs. It should also be remembered that many 8 bit users also own and use the ST as well.

Wanted - Yorky Board

Does anyone know if there are any "Yorky" memory expansions still around, maybe for sale 2nd hand? I'm not that keen on taking my 800XL apart and this was a 256K expansion produced by York Computers which plugged into the expansion bus. York Computers stopped production when the cost of RAM chips made it too expensive to produce.

Did any other members belong to the Gladden House rental club - it closed a few months ago. They had available many programs which many reviewers said were only available directly from the US of A. I'd love to know if another company / group brought out the library of disks.

Brian Sheldon
Morecombe

Can anyone supply Brian with a "Yorky" expansion board. It sounds like the easiest 256K upgrade around. I've never heard of the Gladden House rental club - does anyone have any news on what happened to the disks?

New Electronics & Computer Club

Could you please tell your users that I am starting an Electronics and Computer Club in the Acton Area. We do not mind what micro they have and any one can join. Also we will give talks, issue a newsletter plus lots more.

P. Fasoli
10 Sunningdale Avenue
Acton
London W3 7NS

8 Bit & ST Compatible Monitors

Thinking ahead to the day when channel 5 arrives, those of us who use televisions instead of monitors will need to either have the modulators in our computers tuned to another spare TV channel or think about the expense of buying a monitor.

My question is this; is there a monitor on the market which will enable me to plug in my 8 bit 130XE and my 520STFM (not at the same time). I don't mind buying one monitor but I'm not flush enough to buy two. I am correct in assuming that there is not but perhaps a monitor with a 'scart' fitting may enable me to fix the problem.

Ray Pawson
London

Any monitor with a complete implementation of the SCART standard should be usable with both your computers. The 8 bit requires the composite video signals and the ST the RGB signals. If anyone has any experience using one monitor with both computers we would be pleased to hear from them.

From across the seas ...

First, I want to thank you for sending the disks and magazine. Your magazine was very informative, and I really look forward to trading with you. I would like to include some of your articles in our magazine if possible.

The disks you sent were great also. I thought we had about everything available for the 8-bit but I have not seen any of he material you sent me. It looks like the 8-bit is alive and well over there.

I am sending you a couple of extra GAG disk catalogs. Will you please send them to a couple of other clubs in your country and see if they would be interested in trading disks and newsletters? I also sent you some of our past magazines, and will add you to our mailing list.

Jerry Cross
President, Genesee Atari Group

Thank you for the disk magazines and catalogs. The spare catalogs I have sent to different users groups along with this issue of 8:16. I still have 4 duplicate copies of Great Lake Atari Digest (2 x June 90 and 2 x July 90), the official newsletter of CHAOS and GAG - the first four to send me a SSAE at least 7.5 x 9 inches in size, gets one each.

We have no objections to other publications reprinting articles from 8:16 provided the author and 8:16 are acknowledged as the original source and a copy of the publication is sent to 8:16. Articles that we have re-printed from other newsletters should be cleared for duplication from the original source. If you let us know which articles you are interested in I may still be able to send you files on disk.

and from under the channel.

Reading page 22 of issue 10 of 8:16, I have seen a little bug. In fact this is not ICD that no longer support the 8-bit, but this is Frontier Software who is preferring to concentrate on the 16 bit market.

This is what I am thinking, after reading the Gralin International advert on page 5 of the same issue. Am I right? Is there still a ICD products distributor in the UK?

For the occasion, I am asking if you could give me Derek Fern's address, because I cannot telephone him.

Carrodano Daniel
France

Firstly, yes I made a mistake, and you are correct. Gralin International are the sole UK distributors for ICD 8 bit products within the UK and it was Frontier that stopped importing these products. Derek Fern can be contacted at Micro-Discount, 265 Chester Road, Streetly, West Midlands, B74 3EA.

Viewdata Update #1

I've have just read Mike Falconers letter. Miracle Technology (now renamed) produced **Multi-Viewterm** written by Matthew Jones both as cassette and disk with a Datatari cable link (25 pin to Atari 13 pin). It features 300/300, 600 and 1200 also 1200/75 baud. Try them on 0473 233888.

R. Feldmesser
Shrewsbury

Viewdata Update #2

I am writing in answer to Mike Falconers letter in the "Dear 8:16" page of issue 10.

I can assure him that there is viewdata software for the 8 bit Atari. The program is called **Multi-Viewterm** and is, or was, marketed by Miracle Technology Ltd., of St. Peters Street, Ipswich, LP1 1XB. The program was written by Matthew Jones, a name which should be familiar to many Atari owners.

Multi-Viewterm only runs in mono but you can choose your own colour, I find a green screen best, and as it uses a special cable to connect the modem to the Atari serial bus no interface unit is needed, but the modem must be the last device on the chain.

While no up-load or down-load is available there is a buffer which you can use to 'capture' a series of frames. These can then be saved to disk while on-line or after you have logged off. The normal Atari screen edit is available on the Prestel mailbox screens so you can delete/backspace and move the cursor around as easily as you can on a word processor. There is a number of 'standard' command sequences built in such as repeat page etc and there is the facility for 9 user defined keys for sending passwords etc.

The program comes on disk and will run on a 130XE and a 48K Atari 800 without the Basic cartridge. I am once again using it on an 800 as my XE has died. I do not know if the program is still available as I brought mine in 1986 with a WS2000 modem but you could contact Miracle Technology and find out.

Finally, a plea for help. After 7 years with an 8 bit, which I shall still keep, I have brought a 520STFM upgraded to 1meg. Do you know of an Epson LX800 printer driver for 1st Word? The RX80 driver supplied works but not that well. It prints in Sans Serif in draft mode but Epson Roman in NLQ and selecting draft / NLQ from 1st Word is no guarantee of what it will print in.

Peter Boulter
Twickenham

Peter also supplied a telephone number for Miracle Technology but when I tried it I got the disconnected tone. Anyway, the information has been passed on to Mike who now has a better chance of locating a suitable program for his application. Now if someone could just answer Peter's question about using 1st Word with an Epson LQ800 we will have an information exchange bureau in place.

MYDOS & the XF551 Disk Drive

Hope this is of some use in helping with the problem of configuring the XF551 disc drive and could you please pass on the contents of the envelope to Mr. Paul Allen of Somerset.

P. Mattock
Essex

With pleasure. Included within the envelope was a print out of some of the MYDOS manual, a disk formatted for the XF551 and instruction on how to perform this task as follows:

First boot MYDOS master disk. When option screen appears select option (P) Set Density. At prompt Drive, density enter 1,D and hit return.

Then pick option (O) Change Config. At prompt Enter drive no enter 1 followed by return. At the following prompts enter as shown:

Remote drive:	N
Is Drive configurable:	Y
High capacity drive:	N
Is drive double sided:	Y
Tracks/side:	40
Step rate:	0

Now insert a blank disk into the XF551 and initialize. At prompt (A) or (Y) enter Y as A is for 1050 drives. Once formatted write DOS to disk. You should now have a double sided master disk for MYDOS that will run on the XF551. Just duplicate the disk for working copies.

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

Then write to:

The Flight Controller, 8:16
248 Wimborne Road, Oakdale,
Poole, Dorset BH15 3EF

Introduction to C Programming

By David Watson

This will be the last of my articles on C programming. I have now been writing them for a year, and a lot has happened in that time. Perhaps the most significant is that some of the commercial magazines have started their own series on C programming, and some even have the audacity to promote Sozobon C in the way that I did in my first article. All that I can say is - I was first. I have enjoyed writing the articles, but feel that I cannot compete with the commercial magazines because they come out monthly whilst 8:16 appears quarterly. This means that they can progress at three time the speed of my articles, whilst still maintaining a leisurely pace for their readers.

Hopefully I will still write the occasional article for 8:16; I've certainly enjoyed reading it. In some areas it is far better than the commercial magazines, particularly for those of us who are more technically orientated. However, enough of this idle chit-chat, on to the subject of C programming.

Parameter Passing

The first subject that I plan to discuss is the subject of passing parameters to programs. Fortunately C supports this directly, and eliminates the need for compiler dependencies, although this is only true to a limited extent as we will see shortly.

The trick is to define the function 'main()' with two arguments, as shown below:-

```
main(argc, argv)
int    argc;
char   *argv[];
{
 ... rest of program...
}
```

The first argument, argc, is an integer specifying the number of command line arguments the program was called with, and argv is a character array of the arguments. By convention, argv[0] contains the name of the program being run, so argc is always at least one. Now there's a problem here as not all C compilers put the name of the program being run into this array element because TOS doesn't make it too easy to do so. It is not safe to assume that K&R convention is being followed here.

The second problem arises when particularly long command lines are being passed to programs. Originally, when TTP programs were invoked from the desktop, there was only a limited amount of space to type in a command line, and Atari therefore only reserved a limited amount of space for command lines. However, with the advent of command-line interfaces such as Gulam or the Mark Williams shell, it became possible to pass command lines to programs which would overflow the reserved area. Mark Williams developed a solution to this, the writers of Sozobon C produced another, and Atari proposed a third. The last I knew, Atari were attempting to decide on a standard way of passing extended command lines to programs, given the three mentioned previously. Unfortunately I cannot the exact details of each method, so I'll have to skip over them here.

However, for the majority of cases, you will not need to worry about how long the command line is. Finally, on the topic of passing parameters to programs, here is an example of a program which will echo its name and arguments to the screen:-

```
main(argc, argv)
int    argc;
char   *argv[];
{
   int   i;

   for(i=0 ; i<argc ; i++)
      printf(“%s”, argv[i]);
   printf(“\n”);
   scanf(“\n”);
}
```

Okay, that's easy enough, isn't it. Now, it would be far more interesting if we could use our own routine to print characters to the screen, and to make things a bit faster we want it in assembly. The only problem is how do we link in modules which have been written in assembly?

There are two parts to this problem, that of writing the routine in assembly in such a way that it does not interfere with the C compiler's code, and that of linking it in with the rest of the C program.

The first problem is solved relatively easily. If we preserve all of the registers on the stack using 'movem d0-d7/a0-a5,-(sp)' at the beginning of our procedure, and restore them using 'movem (sp)+,d0-d7/a0-a5' at the end of the procedure then we have all eight data registers and six of the address registers at our disposal. The two address registers which we have not touched have special uses.

As any 68000 programmer will know, a7 is used as a stack pointer, and it is clearly useless to preserve its value on the stack. This leaves us with the a6 register. The a6 register is commonly used to create stack frames which allow the user to dynamically allocate memory for local variables at run-time. All the programmer needs to do is work out how much memory is required by local variables, and reserve that amount of

```
_add:
   link      a6,#-2        ;Create space on stack for local
                           ;var 'c'.
   move.w    8(a6),d0      ;Store the integer passed in d0.
   move.l    10(a6),a0     ;Store the pointer to an integer
                           ;in a0.
   add.w     (a0),d0       ;Add the 2 integers, store result
                           ;in d0.
   move.w    d0,-2(a6)     ;Store result in destination integer.
   unlk      a6            ;Release stack space.
   rts                     ;Return from function.

Example 1:
```

```
BEFORE                                    AFTER

SP ->   TOP OF STACK                              TOP OF STACK
                                          A6 ->   A6.L
                                                  int a     <- #-2(A6)
                                                  int b     <- #-4(A6)
                                          SP -> int c       <- #-6(A6)
```

Figure One:

```
  4(A6)  :  int y (2nd local variable declared)
  2(A6)  :  int x (1st local variable declared)
   (A6)  :  Old value of A6 (before link instruction)
 -4(A6)  :  Subroutine return address
 -8(A6)  :  int a (1st parameter passed)
-10(A6)  :  int b (2nd parameter passed)
```

Figure Two:

memory on the stack when the procedure is called. An added advantage of this method is that the procedure is re-entrant, and that recursive calls to it will not destroy data from a previous call.

Let us assume that we wish to write our routine to print a string of null-terminated characters on the screen. From C we wish to call the routine in the following fashion:-

```
answer = add(a, &b);
```

We are using three variables here. The function will add two integers and return the result, which will be stored in the variable 'answer'. The first integer passed is passed by value, the second passed to the function is passed by reference. This example is a bit contrived, but it serves to illustrate my point without being over complicated.

Such a routine may be coded in C in the following manner:-

```
int add(a, b);
int   a, *b;
{
  int   c;

  c = a + *b;
  return( c );
}
```

A compiler may then code it as shown in example 1.

The complete parameter passing technique can be seen here. The first integer passed is stored in d0, the second is a pointer and it is accessed via an address register. As all of the variables being manipulated are integers they can easily fit into single registers, and this has simplified the compiler's task. However, if the data types had been more complicated then the compiler would have simply pre-calculated all of the necessary addresses and manipulated the data as necessary, using indirect addressing.

Let us consider the example above in which an add function is called:-

```
main()
{
  int   a, b, c;

  a = add(b, &c);
}

int add(a, b)
int   a, *b;
{
  int   c;

  c = a + *b;
  return( c );
}
```

This could be compiled as shown in Example 2.

The first step would be for the compiler to reserve memory on the stack for variables local to the current function - at the start of the program this is 'main()'. In the above example, we require enough space for three integers, which corresponds to six bytes. The first instruction in the program, "link a6,#-6", does this. The link instruction performs the following actions. Firstly, the value in the specified address register is stored on the stack and its address in memory stored in A6. Then the second argument is added to the stack pointer, thus reserving memory on the stack (for local variables). The second argument is either negative or zero; adding a positive number to the stack pointer may result in the stack being overwritten.

When the un-link instruction is then executed, the address in the specified address register is copied to the stack pointer, and the address register restored with its old value.

Now, to access these local variables it is necessary for the compiler to access them using register register indirect with displacement addressing, where a6 is the address register used to calculate the offsets. After the machine has executed the link instruction, the stack looks like Figure 1.

From this illustration it can be clearly seen how the stack is made up. The local variables appear on the stack in the order that they were declared (i.e. the first variable declared is pushed on to the stack first, and so on).

When dealing with passing parameters to the function, a similar process applies. This time, instead of accessing the data using positive offsets from a6, it is accessed using negative offsets. The important

```
_main:
    link      a6,#-6          ;Reserve memory for local vars.
    pea       -6(a6)          ;Copy pointer to stack
    move.w    -4(a6),-(sp)    ;Copy integer to stack
    jsr       _add
    move.w    d0,-2(a6)       ;Copy value returned to dest var.
    unlk      a6
    rts

_add:
    link      a6,#-2          ;Reserve memory for local vars.
    move.l    10(a6),a0       ;a0 = b
    move.w    8(a6),d0        ;d0 = a
    add.w     (a0),d0         ;d0 = a + *b
    move.w    d0,-2(a6)       ;c = d0
    unlk      a6
    rts
```

Example 2:

thing to remember here is that the parameters passed are pushed on to the stack in the reverse order that they were declared. Hence, if we declared our function as follows, the stack below would be obtained:-

```
int function(a, b);
int   a, b;
{
int   x, y;
.
.
.
}
```

gives the stack shown in Figure Two above.

Well, there you have it; all you need to know about passing parameters to functions at the assembly level. Needless to say you'll need to use some common sense in calculating offsets; integers will take up a word, pointers will take up a longword, and so on. Note that if you pass a 'char' then it will be sign extended to form a word before being put on to the stack. Do not fall into the trap of thinking that it only takes up a byte of space because it doesn't.

To finish up on the subject of parameter passing, consider the problem of passing a structure back to the calling function using the 'return()' call. The method used by Sozobon C is to reserve some memory which is large enough to hold the structure. When the 'return()' is reached, the variable to be returned is copied into the reserved space, and control returned to the parent function. The contents of the reserved space are then copied into the destination variable, and the program continues. It would appear to be more efficient to directly copy the information to the destination variable, but this has practical problems from the point of view of writing the compiler.

Linking Assembler With C

That's enough on the subject of passing parameters, however. The last topic I intend to discuss is how to link an assembly level routine in with a C program. As part of this topic I will cover how to link separate modules together to form a final, executable file. Let us consider the case of having two files which contain the following portions of program:-

file1.c

```
main()
{
  int  i;

  for(i=1 ; i<10 ; i++)
     printf(“%d squared =
%d\n”,i,square(i));
}
```

file2.c

```
int square(i)
int  i;
{
  return( i*i );
}
```

In this case our second file contains a utility function which will return the square of an integer. The first file accesses this function directly; the compiler assumes that the function will be in one of the other files included in the compilation, and will not complain that 'square()' is not in the same file. Of course, if it isn't in any of the files included in the compilation then it will fail at the link stage.

To compile this, we would type the command line:-

```
cc file1.c file2.c
```

This will compile both files, and link them together to form an executable file (exactly what it is called will vary from compiler to compiler; however you can specify what to call the executable file using the '-o' flag followed by the file name). Users of C compilers which use GEM environments are saved from the chore of all that tedious typing. What you want to do instead is select a 'compile and link' option from one of the drop down menus, and specify both 'file1.c' and 'file2.c' in the compilation.

And finally, the subject of including assembly modules in the final link. Actually this is easier than you may think, as all you need to do is assemble your code into an object module. If you have a choice of formats then DRI is probably a good choice. Indeed, if you use Sozobon C then all of the C object modules are in DRI format, and the linker can act upon it directly. Most of the commercial compilers will not let you include DRI object code directly, but provide facilities to either import it, or convert it into a suitable format. Consulting the manual is the best advice I can give you here. Having put the assembled object code into a suitable format then you simply include it in the final link, as before. So, if 'file2' is an assembled object module, then the command line will look like this:-

```
cc test1.c test2.o
```

Easy, isn't it!

What of in-line code, which avoids the problem entirely? Some compilers allow you to place code within C functions, surrounded by suitable identifiers so that the compiler recognises it as such. Some people don't like it at all, but I am not personally that bothered. I must admit that I don't really like the Sozobon C method of doing it, but the method used in the old Megamax C seemed fine. I think that the main gripe is that using in-line code gives you less flexibility, as assemblers are dedicated to the task whilst compilers are not, and assemblers can therefore offer more reliability in doing it right and providing useful features such as macros. You pay your money and you take your choice.

Well, that's it. This last article has been the most technical of the four. Hopefully it has given you an insight into the workings of compilers which can only be a good thing. I hope you've found the articles interesting and informative. All I can do now is wish you luck with your C programming.

Good luck.

Turbo-Info

Turbo-Info #3: What is the Turbo-OS and What Does It Do?

(Reprinted from Atari Interface Magazine, September 1989)

By Chuck Steinman (Dataque)

This is the third part of a multipart information article on the Turbo-816 (T816) from DataQue Software for the Atari XL/XE computers. The previous instalments went over the basic system and the memory and this article will explain some of the unique functions of the Turbo-OS.

What is Turbo-OS?

Some background on the Turbo-OS for those of you which have just tuned into the AIM Channel $816.... The Turbo-OS is an operating system for the Atari XL and XE computers which either can fully replace the Atari OS you now have or be used in addition to the Atari OS. The latter application requires the addition of a toggle switch to allow for user selection of which Operating System is active. A version for the 800 computer will be offered also. The Turbo-OS will only operate in systems with the T816 CPU adapter card installed.

My original intent was to have both the Atari OS and the Turbo-OS in one PROM. This would have reduced the amount of hardware needed and simplified installation.

Due to my not being able to acquire an acceptable agreement for the Atari OS, the Turbo-OS is a separate PROM, without the Atari OS included. This results in an additional 28 or so solder connections if the "Dual-Prom" option mentioned above is installed. Another option is to use one of the commercially available OS switchers, which include a PC board with multiple OS sockets and a selection switch.

There are two sections to the Turbo-OS, the normal CIO/SIO system interface and the new user interface. The latter is a menu system which allows the user to perform many new unique features. Also there are many new CIO vectors, which perform memory management and system control functions.

The Turbo-OS menu features include the following:

- A non-interactive Note Pad mode which is similar to the Atari 400/800 memo pad mode. While not too useful, kids get a kick out of it, and it does come in handy for certain applications.
- A full 16 megabyte hexidecimal and ATASCII dynamic memory dump utility. This will continuously dump the contents of a specified block of memory to the screen. There are several command keys to make moving about the large addressing range a breeze.
- A system configuration key, which will scan the addressing range of the Turbo-816 for applications and memory. If applications are found, they are installed into a menu, where they can be executed by a key selection. Up to 8 of these applications can exist.
- There is a memory option to clear areas of RAM including: Standard RAM, Banked RAM (extended), Application RAM (explicit), and Expanded RAM. There are several options as to exactly how the RAM is cleared.
- There are many built-in keyboard "hot" keys which can be made active from the Turbo-OS menu. An indicator shows the current status of the hot keys. These keys will allow the cursor to move about the screen quicker, and turn the video, key-click. and SIO noise on and off. Also key repeat and delay rate are adjustable from the keyboard.
- A built-in diagnostics screen will allow for verification of all memory types, including the Turbo-OS PROM. Also, there are provisions for testing the keyboard, joystick controls, paddle controls and other peripherals.
- There are also keys to select either a cold start of the system, or an option to return to the users application program if one is active.

The new CI0 (Central Input/Output) system calls include the following:

- CalcOS: Calculate and return the checksum of the OS ROM in register A.
- TRAMCk: Scan one (01 all) or' the three types of non-standard RAM.
- TAlloc: Allocate a specified amount of one of the foul types of RAM.
- TDaloc: De-Allocate a specified amount of one of the foul types of RAM.
- TExecu: Execute an installed Turbo-Application.
- Tlniti: Initialize an installed Turbo-Application.
- DoBank: Select one of the 16 possible Extended RAM banks.
- NatVec: Call an OS function from outside the 64k address range.
- CpyBuf: Copy up to a 64k block from-to anywhere in the 16MB range.
- FilPat: Fill a block anywhere in the 16MB addressing range with a pattern.

Another feature of the Turbo-OS is the improved floating point package. These routines are called by many application programs and languages such as the Atari BASIC and Atari Assembler/Editor cartridge. The amount of increase in speed ranges from 10% to over 300% and is being improved further. The improved routines were originally coded by Charles Marslett and were merged with the Turbo-OS under his direct authorization.

Some items, which should be noted about the Turbo-OS and compatibility with current applications, are:

- The C: (cassette) handler was removed to make room for the Turbo-OS enhancements and Turbo-menu. If the C: device is used, there is an installation option to allow the original Atari OS to be selected by a toggle switch. Anytime the cassette is needed, a flick of the switch will allow the system to communicate with the cassette.
- The extended character set table was also removed to free up more memory. This should not be a significant problem for domestic units, and like the cassette device, could be selected by switching in the Atari OS.
- Two routines to support the uploadable device linker from the 1090 interface were deleted. Their entry vectors terminate with an appropriate error. It was felt that there was no need to support this device in the Turbo-OS, and again, the switch option could be used if needed. Insufficient documentation would have precluded me from being able to write the routines from scratch anyway.
- A few games, and even fewer applications, scan the OS and compare it against the Atari OS. If any OS other than the Atari OS is resident, either an error message is printer or some other abnormal sequence follows. The switch option will allow these programs to run as normal.
- Some user written programs, and a few commercial programs, have used illegal entry points into the OS. The only valid entry points in the OS are on page $E4xx as listed in several reference documents. To simplify coding, or reduce code size, some people feel that using subroutines in the OS, other than the vectored ones, is a way to reach that goal. This is not as common as it was before the XL series was introduced.
- A very rare compatibility problem exists where a program either uses an unpublished 6502 op-code or feature. This has only shown up in two programs out of the several hundred programs tested by the beta testers. There is no easy fix for this, unless the program itself can be patched to remove the conflict. This would be evident if the program would not run with either OS activated.
- Finally there is a compatibility problem with certain pbi devices which do not buffer the address and data lines properly. The problems of this type are limited, but I am looking at modifications to the motherboard to correct the problem in all cases. The problem occurs only when the pbi device is attached to the Atari and is related to noise generated when that device is accessed.

Turbo-Info #4: New 65C816 Registers

(Reprinted from Atari Interface Magazine, October 1989)

This is the fourth part of a multi-part information article on the Turbo-816 (T816) from Dataque Software for the Atari XL/XE computers. The previous instalments went over the basic system memory and 0S. This article will cover some of the new 65C816 addressing modes.

The 65C816 is a 16bit processor, which also has the capability of executing 8-bit 6502 instructions. To allow for this, there are two operating modes tor the CPU, the EMULATION mode and the NATIVE mode. The emulation mode, as it implies, will fully simulate a 6502. The native mode opens up the full power of the new processor.

The 65C816 CPU

Even while in the emulation mode, many new instructions are available. Some are of little use because of being limited to the 64K addressing range of the 6502. Many of the new instructions allow for smaller, faster code. Other instructions offer power which was previously not feasible or practical.

Because of the limited 16bit addressing range of the 6502, the designers of the 65C816 realized that if their new processor was to survive, it must be able to address more than 64K while still maintaining 6502 compatibility. This was done by using multiplexing which adds another eight lines to the address bus. This new 24bit address bus allows for a full 16 megabyte range.

Obviously, there was no inherent way to control these extra eight lines with the 6502 instruction set, so new instructions and addressing modes would have to be added.

To maintain 6502 compatibility, there would be certain limits to the functionality of those lines while in the 6502 emulation mode. There must also be the addition of several new registers to allow for dynamic control of the state of the new address lines.

Features of the 65C816

There are two new registers in the 65C816 that control the eight new address lines. The B register controls the new lines during access to data memory. This would be any time that you would be loading, storing or doing manipulations to data external to the processor. This register is also known as the DATA BANK register.

The K register is used any time that program memory is being accessed. This would be when the CPU is fetching an instruction and its operands. This register is also known as the PROGRAM BANK register. Both of these registers are eight bits wide.

It should be emphasized that the B and K registers are not used in the emulation mode on the T816, so they could be used as general purpose registers. Because of this, while in the emulation mode, the CPU is limited to the base 64K of memory. This lowest 64K is where the stock XL/XE motherboard resides in the memory map. The CPU, and Turbo-816, power up in the emulation state, to maintain compatibility with the 6502.

In the native mode, the B and K registers are appended to the normal 16bit addresses. This allows for expanded addressing, even while still executing the standard 6502 instructions.

There are two other registers which should be mentioned at this time. In the 6502, the lowest 256 bytes of memory had special addressing modes. They were called "Zero Page Addressing."

There were several variations of zero page addressing depending on which, if any, index registers were used, and also if the data was loaded directly or indirectly. The 65C816 expands this addressing mode by allowing this same addressing mode throughout the base 64K. The register that allows this to be done is the D, or Direct, register.

The D register is 16bits wide. The contents of this register are added to the user supplied operand to form the 16bit address. This register is active whether in native or emulation mode, which causes some programmers grief, as they assume it is always on page zero while in the emulation mode.

The last register is really just an extension to an existing register. With the 6502, there was the limitation of only having a small 256 byte stack range. This limited use in applications where heavy parameter passing and stack usage were required. The 65C816 allows the stack to be located anywhere in the base 64K, and in the native mode the stack can be up to a full 64K in theory.

A common mistake made here is that programmers forget that the stack is always located on page one, and always 256 bytes in size while in the emulation mode. Also, people assume that in the native mode the stack wraps around within the current page.

Another feature of the 65C816 is that the index registers (X & Y) and accumulator (A) may be either 8bits or 16bits in size while in the native mode. The index registers are always the same size, but the accumulator may be separately sized. Of course while in the emulation mode, all three registers are limited to eight bits.

Any time you switch between native and emulation mode the registers will be forced to eight bits. The upper byte of the index registers will also be lost. The upper byte of the accumulator will be retained while in the emulation mode but only accessible through the XBA instruction. Memory accesses will always take on the size of the accumulator. This is easily overlooked, especially with the shift and rotate instructions.

Another control function is the E bit in the status register. This bit controls whether the Turbo 816 and CPU are in the emulation or native mode. This bit is changed by using the CLC or SEC instruction followed by the XCE instruction. This instruction exchanges the Carry flag with the emulation flag. In this way, you can switch modes easily, and also know which mode you were switching out of (by the returned carry).

There are several additional control pseudo-registers in the Turbo-816 that are controlled by the CPU and ANTIC. This allows access to the full 16Mb of address space with the CPU, while limiting ANTIC, GTIA, POKEY, and the PIA to the lowest 64K. These registers are not user accessible directly but are controlled via functions of the CPU and ANTIC hardware.

The Black Box

Reviewed by Simon Trew

Synopsis

The Black Box is an interface from the States that allows you to connect standard peripherals to an Atari 8-bit. This includes hard drives, RS232 devices, Centronics printers and standard floppy drives. It controls the computer via the parallel bus on the XL/XE range of micros.

I sent away for a Black Box and 20Mb Hard Drive system, enclosing my cheque for £500. Phew! The Black Box costs £200 and the Hard Drive is a further £300. The Black Box is certainly worth the money, but I think £300 is rather steep for a hard drive. The hard drive is a standard SCSI embedded drive (a Seagate ST-225N, to be precise) which could be got cheaper.

The box arrived and I unpacked it. First thing to notice was that the Black Box actually came in no box at all neither was the hard drive cased. Whilst this doesn't really bother me, I think that the title is a bit misleading. The Black Box (henceforth 'BB') is a PCB about the same width as the computer. It fits straight in a 130XE or via a cable into the 800XL. There are some DIPs and four switches. These control hard drive configuration, printer configuration and allow access to the BB menu and screen dumper. The 130XE cartridge port is brought out on the board, and on the 800XL you use the standard hole in the top of the machine for cartridges.

Printer

The BB can be configured to take the printer output that normally travels along the SIO and send it through its own parallel printer port. This seems to work very well, being entirely transparent to the computer. It can take output for all printers or just one printer - so that if you have more than one printer, one can be handled by the BB and another by the SIO. There are DIP switches to enable LF/CR or CR combinations, and the carriage return is automatically converted to CHR$(13). There is a push-to-make switch on the BB which, when pressed, dumps the screen to the printer. This can either be a text screen or a graphics dump - controlled by another switch. The graphics dump is, alas, only monochrome (not shaded), but is quite handy for graphics 8 screens and so on. The dump also takes account of the width of the screen (it handles narrow screens OK). When in text dump mode it also seems to translate graphics characters into approximate text representations, using plus signs, bars, negative signs etc. It is also possible to use extra memory on the BB board, or the internal memory of the 130XE or upgraded computers, for a printer buffer.

Serial

The BB also has an RS232 port. It has some advantages over the P:R: Connection or 850 interface. The RS232 handler is in BB ROM and so takes up no user memory. It also can operate at 19200 baud, whereas the 850 only operates at up to 9600 baud. The greatest advantage, though, is that it can read and write data to disk (or to a printer) whilst in concurrent mode, because it doesn't use the SIO. There is only one RS232 port on the BB. I have tried using the port with KERMIT and AMODEM7 without success, although my own terminal program and Mini Office 2 work OK. I would be interested to know if this is a fault of the programs eg not calling R: correctly or of the Black Box (or am I doing something wrong?)

Hard Drive

The main selling point of the BB, though, is the hard drive interface. This allows you to attach up to 4 SCSI controllers (8 drives) to your 8-bit computer. Because most DOSes cannot handle a disk the size of a hard disk in one go (SpartaDOS has a limit of 16Mb per disk), the BB allows the disk to be split up into partitions. The system can cope with 96 partitions across the disks. A partition can be up to 64Mb long, although SpartaDOS and MYDOS can only handle sector numbers up to 65535, ie 16Mb per partition. Each partition can then be assigned to any of 9 drives (that is D1: to D9:). Also floppy drives can be assigned to these drives, and need not be the same as the config number on the drive. For example, my disk drive is configured as number 1 but I have it on partition 9.

The BB normally operates 256-byte sectors. Most hard drives actually have 512-byte sectors, but no 8-bit DOS can handle this. So the BB automatically splits each sector into two. This does slow down disk writing but it speeds up disk reading a bit. To give you some idea of the speed of the device, SpartaDOS wrote 10000 bytes to a new file in 2.7 seconds, and read 10000 bytes in 0.9 seconds. It is also possible to have a partition set up as 'single density'. In this mode, only the first 128 bytes of a sector are used, rather wasteful of space, but this feature is really useful for sector copying boot disks (it can't copy protected disks of course). The system always boots from whatever disk or partition is assigned to D1:. It is possible to assign a floppy disk to any drive. The BB also features a built-in UltraSpeed SIO handler so if you have the US chip then all your disks can be configured for US, not just the SpartaDOS ones. It is also possible to assign a partition to more than one drive. This is useful under SpartaDOS so that one can have different directory paths set up for the same drive. This is useful for programs which won't allow you to put in drive paths, because you can have different paths set up on each of the logical drives.

The partition allocation and logical drive mapping can be changed at any time by pressing a push-to-make switch which takes you into the main BB menu program. This also allows you to change the drive configuration and printer configuration. There is also a monitor/disassembler program. This is quite useful for checking where programs get stuck or searching for bytes. However, it is not possible to 'single-step' or trace a program in any way. One can change memory but unfortunately memory cannot be dumped to disk - a serious omission, in my opinion.

There is a DIP switch setting to allow MIO compatibility. Users of ICD's MIO device can use their hard drives with the Black Box without the need to reformat them. The only difference in the storage format is that the MIO stores all data inverted, that is, if a computer sends a sector filled with $06 then the MIO stores it on the hard disk filled with $F9.

Robustness

When I received my BB, I eagerly connected it up to my 130XE. I powered on and - well, not much. The drive would not boot properly. I went into the menu and the screen was full of junk. I reformatted the disk, changed the partition, still getting lots of junk on the screen. I phoned Daniel Bruen, who was at all times very helpful, but neither of us could work out why it was failing. I suspected a faulty cable but a multimeter disproved this. Eventually I tried the system on Gavan Moran's 800XL and everything worked perfectly! I phoned Daniel Bruen and we both agreed this was a strange problem. He offered to sell me an 800XL which he had tested with the box, which I bought, and everything worked perfectly. I can't say why my 130XE wouldn't work so cannot offer any suggestions for checking yours before buying a BB. Any suggestions would be appreciated.

Since then, everything has worked fine. On one occasion, I thought I had lost all my data. Every directory on the disk was corrupted after the third entry! And every file was corrupted after about five lines of code! After a bit of lateral thinking I discovered that SpartaDOS had somehow flipped a bit telling it how many bytes there are per sector. This is the sort of thing to be careful about, when using untested software, because it can do a lot of damage. For this reason, there is a switch on the BB which can prevent writing to any partition on the hard disk. Also, each partition can be protected separately. I now use this switch before trying any new software. The disk supplied has utilities for parking disks, formatting disks and making MYDOS or SpartaDOS directories, but has no backup utility. It should have, in my opinion.

All the leads are of the IDC cable type. I am rather biased against this type of cable because there is no soldered connection between the cable and the socket or plug and after a lot of use the connections can start to break. I have had no problems yet. The power supply provided is well built but the leads connecting it to the Black Box are connected with terminal block. I expect a bit better than that for £500!

Documentation

The BB documentation is quite good. My copy was rather creased and was written over. It appears that Daniel Bruen had used it himself. If a manual needs amending then I would rather an erratum slip was inserted than a few comments penned on in biro. Whilst Mr Bruen is providing a good service and a fairly quick turnaround, it is these little points (terminal block, unboxed Black Box, shabby manuals) which in my opinion detract from the quality of the product.

The documentation provides information about setting up the hard drive, RS232 etc., and provides details of XIO calls for the RS232. I have done some comms and can follow the RS232 bit, but I think that someone new to RS232 might need another book to give a background to its protocol and so on. The documentation also provides lists of locations that can be accessed by the programmer. The BB is accessed via pseudo space addresses at $D500 to $DFFF. The BB ram is viewed through a window at $D600-$D6FF, and the BB ROM through a window at $D800-$DFFF, much like the banked RAM on a 130XE. The BB doesn't use any real memory, though. (I use SpartaDOS under the OS ROM without conflict). I have already written a utility that will show me all the partitions on logical drives without the need to enter the BB menu, it really is quite simple (it can even be done with Basic PEEKs and POKEs).

Summary

If you want a hard drive, The Black Box is probably your best bet. It's available NOW, is much cheaper than the MIO board or SupraDrive, and its RS232 and printer port remove the need for a P:R: connection or 850 interface. If you only want comms, then this interface is rather expensive. If you run a BBS, however, or use comms a lot for downloading or uploading software, the ability to communicate without the need to swap in and out of concurrent mode could be worth the extra cash. I wouldn't be without it!

Did You Know...

by Derryck Croker

Boot Sector Protection Scheme!

The following details how you can use the Atari boot sectors for storage of a program protection scheme, or to be more precise, how you can get the Atari boot process to load it for you.

The boot portion of Atari DOS occupies the first three sectors of the disk and are loaded by the resident disk handler in the OS. Once these three sectors have been loaded, control is given to the program starting at the sixth byte from the boot address, which in the case of Atari DOS is $0700, with the boot continuation code starting at $0706. It is this program's responsibility to load the DOS.SYS file, which then allows the user to load and save named programs in the familiar way. Boot, VTOC and directory sectors are automatically written to the destination disk as part of the format option from the DOS menu, but not if the disk is formatted using the DCB. Now the interesting part is that there are 3 boot sectors, of which ONLY TWO are required. So that means we can hide a protection scheme in the 3rd sector. This scheme relies on the fact that the OS boots a disk by reading each sector into the cassette buffer (at $0400), and then copying it to the intended load address. It therefore follows that the last sector read will be held in the cassette buffer, and this becomes the place to put for example, a series of RTS instructions for your program to JSR to. A normally formatted DOS disk will, if addressed in this way, surely crash the calling program. Simply fill the 3rd sector with $60, the assembler code for an RTS. The portion of memory that would normally have been loaded by this sector ($0800) will have the correct bytes added as part of the initialisation process.

The beauty of this method is that it is within the reach of most people, even BASIC programmmers with a USR call of

```
X=USR(ADR("hL[CTRL-,][CTRL-D]")):REM PLA,JMP
$0400
```

whereupon BASIC will transfer control to the machine language program starting at $0400. If the RTS instruction is present, control will return to BASIC, else it will crash! Of course, this is a simplistic approach, and the USR call could simply be REMed out, but there are other checks that you could add with some expansion to the machine language and USR call. (Just make sure that you do not allow the use of the cassette)! To all intents and purposes, disks using this method will appear to be ordinary DOS disks, and you will be able to freely copy their files to other disks.

Device Handler Table

Now we are going to look at what is perhaps one of the least used features of Atari 8-bit machines. One of the tasks that is carried out by the Operating System during power-up and Reset is to download a table held in the OS ROM to RAM. This table, which Atari calls HATABS, holds the start address of a table of vectors in the ROM which deal with fetching data to and from the screen editor, as well as the screen, keyboard, printer and cassette. Additional addresses for the handlers for the disk drive and the RS232 interface are loaded directly into HATABS if either or both are present when the machine is booted. We're going to concentrate on the E: handler, which is the one that is used whenever your programming language asks for a line of input from the keyboard for either deferred or direct mode commands. The E: handler accepts input from the keyboard, echoes it to the screen and stores it into a buffer, which Atari calls LBUFF. It is at $0580 and is $80 bytes long. Your programming language will go on to either tokenise and store this into your program if it starts with a line number, or if it is a direct mode command such as DOS, LIST, RUN etc. it will be carried out immediately. The direct mode is the one in which we are interested, since it provides us the method to introduce extra commands. Unfortunately we cannot easily provide extra programming commands since the language will not be able to find the instruction in its table of known commands, and will therefore create the well-known syntax error message. The technique that we are going to discuss applies equally well to BASIC as well as to the MAC/65 and Atari assemblers, and probably others.

Lets take another look at those tables. As we have already seen, there are 5 of them held in ROM, from $E400. During power-up and Reset the address for the start of each handler's ROM table is downloaded into HATABS which starts at $031A. Each device entry occupies 3 bytes, the ATASCII code for the device ie E, followed by the 2 byte address of the start of the vector tables for the routines in ROM which deal with Open, Close, Get, Put, Status, Special and a JMP to the initialisation routine for that handler. We are going to modify the table so that we can intercept the Get vector to point to the portion of our program that will deal with our new commands, but since these are not directly alterable we will have to download them into a safe area of RAM first. We will also need to repeat the modification routine every time the Reset key is pressed, as well as re-initialise DOS. A safe area of RAM can be found by moving the MEMLO pointers ($02E7 and $02E8) to just beyond the end of our program, and the whole should be assembled just above DOS, say $2000. Follow this chart for the steps required.

a) Store the addresses found in DOSINI ($0C and $0D) into the operand of a JSR REINIT instruction.

b) Load the address of the JSR REINIT instruction into DOSINI.

c) JSR to the address of our routine that downloads the ROM vectors into RAM.

d) Store the address of the end of the end of our program into MEMLO.

This portion of the program should end with an RTS instruction. Follow it with a JSR REINIT and a JSR INIT instruction. The first JSR will now point to the DOS initialisation routine as stored during a), and the 2nd should be made to point to the address of the routine that starts at b). A final RTS ends this stage. Now whenever the Reset key is pressed, the OS will JSR through DOSINI to initialise DOS, then through steps b) to d). The final RTS completes initialisation, and will pass control to the programming language.Our program now continues with the routine pointed to by c), as per this next chart.

a) Search through HATABS to find the E entry. This loop should end with an RTS if the entry is not found, unlikely but good programming practise. Try to modify external handlers which may have failed to load at boot-up and you will be grateful for this exit.

b) With the E entry found, store the address of the buffer that will hold the ROM copy of the Open, Close etc. vectors into the next two bytes of HATABS. Imaginatively enough, I'll call this buffer NEWTAB.

c) Move the 15 bytes that make up the ROM vectors for the E: handler into NEWTAB. This is the first table in the ROM and therefore starts at $E400 as we have already seen.

d) Store the contents of $E404 and $E405 into the operand of a JSR instruction, which I'll label FETCH. For reasons that do not concern us here, you'll need to add one to the contents of $E404 before storing.

e) Store the address of JSR FETCH into the 4th and 5th bytes of NEWTAB. This portion of the program should end with an RTS.

If you're still with me, you will realise that this transfers control to d) in chart one. What we have done, then, is to change the 2 byte address of the HATAB entry for the E: handler to point to a copy of the ROM table, and changed the Get vector in this RAM copy to point to one of our own, so forcing this routine to pass through our JSR FETCH. Now we can examine all input from the keyboard before deciding whether to pass it through or process our new commands. Here's another chart.

a) Call JSR FETCH. This will fetch one byte in the Accumulator from the keyboard, echo it to the screen, and store it in LBUFF.

b) Compare the Accumulator to a carriage return ($9B), which signals the end of the entry. Exit with an RTS for another Get if not found.

c) With a carriage return found, save the Y register, which holds any error that occurred during the Get, the X register, which holds the IOCB number, and the Processor status register. Examine LBUFF for the presence of one of our new commands. If one is not found, reload the Y, X and status registers and the Accumulator with $9B and exit with an RTS. Your language will now take this line for syntax checking, tokenisation, and possible execution.

d) Process the new command. Here we find a problem, since when we exit from this process the new command will still be in LBUFF. Read on for a way round this snag.The easiest way to deal with new commands is to build a table. Each entry should start with the ATASCII representation of the command, followed by the 2 byte address of the routine that you have prepared to deal with it. It is then a simple matter to compare LBUFF with this table and to extract the relevant routine's address. Remember that we are dealing with direct mode commands, which will start at the first byte of LBUFF. Then all you have to do is JSR to this routine, before exiting as in c) in the preceding chart. You will also need to deal with the contents of LBUFF, which will generate a syntax error on return to your language. Probably the best way is to load a full stop into the first byte of LBUFF before exiting. BASIC will accept this and tokenise as a REM statement. You will have to modify this for other languages, for example a semicolon for assemblers.How about an OLD command? All you need do is to look for the NEW command in LBUFF, store all the Page 0 pointers to the variable names etc in a buffer, and pass on the command unchanged. Your OLD command simply reloads these pointers into Page 0 before exiting as in the previous paragraph. It is a little known fact that the NEW command resets only these pointers, it does not clear memory.

I have used a major part of this technique as an aid to program entry by simply changing the contents of LBUFF to uppercase before passing it to the language otherwise unchanged, which allows program entry in lower or uppercase. Don't forget that you will need to skip over the contents of strings, which means that save, load or list filespecs will need to be entered in uppercase from the keyboard. Did you know that you can leave off the closing quote marks when entering a filespec in BASIC?

One further problem exists with this technique in that calling the DOS menu with this new editor handler installed will result in a crash, since the menu will overlay the area of memory occupied by this routine. Ensure, therefore, that you either do not use DOS, or that you have a MEM.SAV file active and make arrangements to reload the pointers to the original handler routine on detection of the DOS keyword in LBUFF.

AtariFest 90

Once again I find myself apologising for 8:16 missing its publication date. This time however there was absolutely no way 8:16 was going to be on time as I was in the States for the whole of October - the month most of the work should of been done. However, while I was in the states I managed to attend the Washington Area Atari Computer Enthusiasts (WAACE) AtariFest 90 show at the Reston Sheraton Hotel on October the 6th and 5th. The following is quick review of what I found.

Who are WAACE?

Before I start my review a few words about WAACE are in order. Firstly, WAACE is not a user group, instead its a collection of user groups located within the Washington DC area. It has an elected committee, responsible for organising event like the AtariFest. In the early days it was also the driving force behind Current Notes, which is now an independent publication in which any Atari User Group can affiliate - you'll find a special subscription form elsewhere within this issue of 8:16. Probably the most well known member of WAACE, from the UK viewpoint is Novatari - the Northern Virgina Atari User's Group. I certainty recall many moons ago wishes there was such a group in the Bournemouth and Poole Area, long before the BaPAUG was formed.

The Location

Previous AtariFest had been held at the Fairfax High School, but due to circumstances this could not be the case this year, so a new venue had to be selected. This turned out to be the Sheraton Reston Hotel which had a suitable number of conference rooms which were put to good effect. As well as this a special arrangement was made with the hotel allowing reduced rates on the room along with a two day ticket to the show.

Expectations & 1st Impressions

I'm one of those people that never expects too much from a computer show, however, the WAACE AtariFest have, over the last few years, built up an extremely good reputation as being a well balanced, friendly event. So one cannot help but turn up with some form of high expectations.

I must say my first impressions of AtariFest 90 were a bit mixed. There appeared to be no one around as I registered 5 minutes before the doors opened on the Saturday. Though, those people already waiting were talking enthusiastically about meeting old friends from previous shows, talking to developers and having, in general, a good time. I remember standing there, looking through the show program, searching for the Frontier Software stand, hoping that their enthusiasm would bear fruit. At the allocated time, the doors were opened and I started the long walk from just outside the Seminars room inside the Hotel lobby to stand 14 where Frontier Software were located.

The Seminars

Throughout the two days, seminars were held within one of the conference halls. It had been my intention to attend at least three of them, but I'm afraid I failed miserablely and missed them all. Some of the seminars looked very interesting with several guest speakers coming from Atari US for seminars entitled "Atari Corporation: Technically Speaking" and "Atari Corporation : Live!". Many of the seminars were about specific products, including one each for NeoDesk 3, Timeworks DTP and Wordflair.

UK Representatives

As you may have already guessed, **Frontier Software** had decided to attend this years AtariFest in order to test the response of their products within the US and to meet new US distributors. As well as Frontier Software, **HiSoft** also had a stand at the show for roughly the same reasons. HiSoft were displaying the range of ST products that have been seen at the recent UK shows.

Frontier Software concentrated on supplying Forget Me Clock 2s, a product I highly recommend, and bare board XtraRAMs. They also had on display a prototype of their multiple disk copier, which received quite a lot of attention. I must at this point thank Martin and Andrew for their hospitality throughout the show and I hope they enjoyed themselves as much as I did.

One of the surprises at this show was the complete absence of any reporters from the UK Atari magazines (excluding me of course and Martin from Frontier who has written an excellent report for Micro Computer Mart). All the ST magazines in the UK report activities at German shows, but appear to ignore the US - I suspect due to cost.

Atari US

One of the highlights of the show, for me anyway, was finding out that Atari US were present. After visiting Frontier, this was my next port of call. On approaching the stands there was a huge crowd building up around one man who towered above the rest of us. The person in question turned out to be Bob Brodie, Manager of User Group Services. Yes, Atari US have someone whose sole role is the interest of Atari user groups - how about it Atari UK.

On display, were some excellent demos showing of the features of the STE, a Portfolio transferring data with an Atari PC, a Lynx and the recently released Lynx carrying case. On Sunday, Atari also gave away a small number of American number plate holders (with the Atari symbol on them) and a large number of bulsa wood planes with the logos *Flying High* and *Power Without The Price*. As well as Bob there were several technical 'gurus' on hand to answer any questions.

While at the stand I took the opportunity to ask Bob why Atari had dropped the Atari 8 bit. The answer, was as I expected - "a computer system consisting of a computer, floppy, monitor and harddisk is far cheaper starting with a 520ST than it is with a 130XE. So whose going to buy the 130XE?"

Another disappointment was the non appearance of the TT. The americans will have to wait a bit longer to see the new wonder machine.

Show review by Colin Hunt

Special Interest Rooms

One of the nice features of this show were the six special interest rooms situated in a long sweeping arc around the hotel swimming pool. The Games room was almost entirely devoted to the Lynx, with there going on 2 dozen available for use, including several multi-player systems. The Swap room consisted of trellis tables where users could display any items they have for sale - if I didn't have to fly home I would of spent a small fortune in this room alone. All the rooms contained something of interest and throughout the day demonstrations were performed, such as Spectre GCR within the Mac & IBM emulation room. Other rooms covered Productivity & DTP, MIDI and Education.

User Groups & BBS's

User groups present at the show included Novatari (hi Geoffrey - thanks for the PD disks, hope ours were OK), Maryland Atari Computer Club, whose newsletter 'MACC News' had a front cover headline of 'Atari Fights Off Takeover Bid From Iraq'. Finally, there was a user group called Atari Users Association which supports the St only and provided me with a very professional looking 'Preferred Vendor Kit', which contained a disk, information and membership form. I also picked up a newsletter called 'The Atarian' by the North East Atari Team.

Many BBS's used the show the promote themselves, though none of them had any systems up and running. The following is a small sample: The Washington MIDI Users Group (703) 532-7860, Bit Heaven BBS (301) 969-1138, A.R.M.U.D.I.C (703) 450-3910, Pizza's Place (301) 325-1027, Inner Limit (301) 356-5112 and Joppa Computer (301) 679-5809.

8 Bit Support

One of the joys of this show was finding several stands supporting the Atari 8 bit. These included Alpha Systems, Best Electronics, ICD, Toad Computers and of course the User Groups, who were located in the room next to the productivity and DTP room. It was also nice to meet the people I talk to regularly on the telephone.

Alpha Systems were selling their entire range of Atari products, including some end of line titles such as COS - the Cassette Operating System which is similar to Transdisk from Page 6. Titles that sold out included the Parrot II sound sampler, Pop'N'Rocker trivia quiz game and the books on Atari protection techniques.

Best Electronics had on display just about every spare part you would need to build an Atari 8 bit computer, and several items I had never heard of - the most interesting being the Microport XL experimenters kit which plugs into the parallel bus, the SIO link kit that allows more than one SIO device, that doesn't have a through port, to be connected to your computer - such as an 410 cassette recorder and MIDIMaster, and the Atari light pen with AtariGraphics cartridge. All these items are now available from Gralin International within the UK.

ICD were selling all the products currently available from Gralin plus the MIO boards in 256K and 1M versions.

The most disappointing aspect of the show for Atari 8-bit users was the complete lack of new commercial software. I say, commercial, as the latest version of the PD program R-Draw was available at the Novatari stand along with its author demonstrating the latest new features. We had a long and interesting discussion on different drawing programs and techniques while I demonstrated TBM Draw, which happens to overscan on NTSC TVs.

ST Support

As expected, the majority of the stands supported the Atari ST. Due to the lack of space I will only cover those I found to interesting or important.

Firstly, **Double Click Software**, authors of many excellent PD programs were demonstrating DC Desktop and DC Utilities. DC Desktop, like Neodesk, is a desktop replacement with all the additional features one would expect, plus a few nice ones - such as the ability to show NEOchrome, Degas, Tiny, Spectrum, Art Director and Mac pictures without having to load the appropriate art package - just like showing text files on the standard desktop. DC Utilities is a collection of programs that allow you to compress/un-compress files in super fast times. I use DC Sea already for our PD library as it produces self extracting arcs. DC Squish even allows the compressed programs to run!

Another excellent utility was Diamond Back II, available from **Data Innovation**. If you own a hard disk, you won't go far wrong using this backup utility. Another interesting utility is HyperLink from **JMG Software International**, which manages any number of HyperLink applications and modules.

The biggest draw back of the Stacy is it short battery life. **MultiByte** have a solution to this - their own Casey Internal Pack that provides 25+ hours of run time. Also available are 'Casey' cases, also for the Stacy.

Gadgets by Small

Of all the ST developers at the show, this one outfit deserves special mention and I would like to award David Small the *Doing The Unthinkable* Award for 1990, for thinking about it and then doing it. If you're currently lost, let me put you out of your agony - the product in question is Spectre GCR, the Mac emulator that runs faster than a Mac Plus, has a bigger working screen area than a SE/30 and reads Mac disks - in my humble opinion the best product this side of Betelgeuse Seven. Gadgets have also produced two other products - a Megatalk card that allows Spectre to talk to Apple servers using Appletalk, and a 68030 accelerator.

Summary

Unlike computer shows within the UK, this show was never over crowded - at no point did I feel like getting away from the madness that prevails here. The enthusiasts that attended where willing to talk to everyone about everything Atari. I for one will try my upmost to attend next years event.

Music Scene: MIDI

By Paul Brookes

Buying a MIDI instrument for use with an ST or 8-bit can be a formidable task. The range of keyboards, synthesizers and expander boxes is extensive. Within this article, Paul Brookes takes a look at the steps involved in choosing a MIDI instrument, and then examines the latest, budget offering from Yamaha, the PSS-790.

Musical Application

A 'computer musician' generally uses a combination of computer and electronic instruments, each equipped with a Musical Instrument Digital Interface (MIDI). On the 520 and 1040 ST's the MIDI interface is located next to the cartridge port on the left hand side of the machine. Mega ST owners are slightly unlucky as their MIDI sockets are crammed into the space between the monitor socket and printer cable on the rear of the machine. The MIDI connections consist of two DIN sockets, one marked 'IN' and the other 'OUT'. Here we come across the first anomaly. The 'OUT' socket is not wired according to the MIDI standard. It actually carries MIDI OUT and MIDI THRU port signals. Normally MIDI OUT and MIDI THRU have their own individual DIN connectors, but ATARI opted to save space by combining them into one socket!

For the beginner, this is nothing to worry about, as it is likely that only one external MIDI device will be connected to the computer. I'm not going to detail the different methods of connecting MIDI instruments together, as there are plenty of cheap books on this subject and most of the ST magazines have already run verbose series on MIDI. Anyone wishing to build or buy an OUT/THRU splitter lead for their ST should get in touch with BaPAUG's Development SIG, who can supply the parts and instructions.

Approximately half of the BaPAUG's MIDI SIG own "home keyboards", the other half owning synthesizers, organs or expanders. The choice of MIDI equipped instruments is huge. Many first-time buyers opt for the "home keyboard" because of the convenience of having built-in speakers and accompaniment facilities, but put up with the small keys used to cut costs. Some full-sized key keyboards are available, but many of these look poor value when compared with budget synths.

For many, the small keys are not a problem as they slave the keyboard from another MIDI device, commonly their ST. A widely used ST program within the club is Music Studio. It has its limitations, (especially only 99 presets), but it costs about £10, and there is a huge library of Public Domain songs available in Music Studio song file format, ideal for customising for output on a "home keyboard".

There are many second-hand keyboards available for under a hundred pounds. Technology within the field of electronic music advances at incredible speeds, so if you buy something second-hand then make sure you pay a suitably low price! £200 will buy you a brand new keyboard with sampled sounds, drums, rhythms and probably a built-in sequencer. Alternatively you could buy a second-hand expander box for between £150 and £250, such as the Yamaha FB01, TX81Z, Roland MT32, or a synth such as the Yamaha DX100 (small keys) for around £120, the DX27 (large key version) for around £190, Casio CZ3000 (£225), CZ5000 (£325), Roland D10 (£420) etc.

TIP: If you buy a second-hand synth/keyboard then check that every key operates correctly, and that the sound produced is not distorted with certain keys. Also check that all buttons and LED's operate correctly. Make sure that the audio and MIDI sockets are in good order and that plugs fit tightly into the sockets. In general avoid equipment that has been 'gigged', even if the price includes a battered filghtcase!

There are plenty of bargains to be had, so happy hunting!

Choosing the Polyphony and Number of Instrumental Parts

The polyphony of an instrument indicates how many notes may be sounded at the same time. 8 notes polyphony may sound good at first but don't forget that the sounds often decay after a note or chord has been played, and overlap the start of the next note/chord. Thus the polyphony required will depend on the type of sound used, the tempo, and the number of notes played simultaneously. A pianist playing quickly and using the sustain pedal may cause 16 notes or more to be sounding simultaneously, and that is just one instrumental part! If you want to play drums, strings, bass and piano from one MIDI unit, then the polyphony required is the total of the individual polyphonies used by each instrumental part required to play at the same instant as another.

Most good keyboards/expanders allow 8 (or more) different instruments to be played at once. For most users this is sufficient. Be careful, because some older keyboards only allow 8 monophonic instruments to be played simultaneously.

Some manufacturers allow the polyphony to be specified for each instrumental part. If 6 notes are needed for chords (i.e. two 3-note chords which overlap) played on strings say, and the polyphony is 16 notes, then 10 notes remain for use by the other 7 instrumental parts. Percussion may require only 2 notes polyphony leaving 8 notes for the remaining 6 channels and so on. Even with 16 notes polyphony, 8 parts can soon use up all of the notes.

My advice is to look at the music you want to program, decide how many instrumental parts there are which are played simultaneously, work out the polyphony for each part, and total these up. Don't buy an instrument which is monophonic in multi-timbral mode, unless you have an expander and are buying the instrument for its particular sound generation merits (the CZ101/1000/3000 are good examples of synths with limited MIDI facilities but easy to use for sound creation, and complement the "sampled" expander boxes such as the Roland D110, CM32P). (Omni-off Mono: MIDI mode 4 of the original specification). Buy a keyboard/expander which supports Multi Mode (Not originally in the MIDI specification) if you intend to use it as your main sound source.

Keyboard or Synthesizer?

I think of a synthesizer as being an instrument in its own right, i.e. it has its own sound creation facilities built into it and is fully programmable.

My definition of a keyboard is something which:

(i) comes with a set of pre-programmed sounds that cannot be altered,
(ii) is designed to imitate other instruments,
(iii) has some sort of built-in rhythm section.

Synthesizers (synths for short) are often more flexible than the fixed voice keyboard when considering the sound creation abilities of the two. Consequently, synths are more complex to program and are less suited to the beginner, who needs to be more experienced before delving into sound creation, and often just wants to be able to have access to a large variety of good quality, factory programmed sounds, of the type which are increasing available on the latest budget keyboards.

Synths provide better long-term facilities, and should be recommended for the more serious musician, who doesn't want his/her music to be limited by the factory pre-sets. Look for a synth with external ROM/RAM card slots as this will have even more sound creation/storage expandability.

Keyboards on the other hand, are ideal for those who don't need the custom sound creation facilities offered by the synthesizers, and in some instances, for those on a tighter budget!

Cost

Typically synths do cost more than keyboards. One or two synths offer truely excellent value for money at the moment. In my opinion these are:

(i) **Kawai K1**
Discounted to under £400 due to the arrival of the Mk2 version which offers a separate drum section and some limited digital effects.
Features: Full-sized, velocity and pressure sensitive keys (5 octaves) 8 part multi-timbral (MIDI multi mode), 16 notes polyphonic. Pitch bend and modulation wheels. Points for: Good range of sound creation abilities. Many RAM cards available. Nicely constructed and very portable. Points to watch: The pressure sensitive keys require more pressure than most other keyboards I have tried. The editing takes a lot of getting used to. Some of the PCM samples are rather noisy. (If this is a problem then buy the K4).

(ii) **Casio VZ-1**
Discounted to less than 350!
Features: 5 Octave full-size, velocity and pressure sensitive keyboard, 16 Note polyphonic, 256 sound and 256 combination memories. 8-part multi-timbral via MIDI (omni-off mono, not multi mode) MIDI IN/OUT/THRU, Pitch bend plus TWO assignable wheels, RAM card slot. Backlit LCD display and easy menu editing and set-up.
Points for: An excellent value master keyboard, for use with 'black box' expanders. Offers lovely analogue style voice creation. Pressure sensitivity slightly better than the K1. Well built. Points against: Multi-timbral mode too limited if this is likely to be your only instrument.

Keyboards:

(i) **Yamaha PSS-790**
Around £230
Probably the best value keyboard to date. Casio better find an answer quickly! See the review of this keyboard elsewhere in this issue!

Most towns and cities have at least one music shop which can demonstrate a range of keyboards from Yamaha and Casio. Some also have other makes such as Technics and Kawai. Prices range from £60 to over £400.

The Yamaha PSS-790

Whilst recently checking out the latest synths and best buys at my local music shop, I came across a nice little but LOUD keyboard made by Yamaha, the PSS-790. Whilst this wasn't exactly what I intended to listen to, it was definitely worth a listen. In fact I was so impressed with its Advanced Wave Memory (AWM) (sampled) sounds that I managed to borrow it for an evening to find out more. Thanks to Gary from Eddie Moors Music for arranging that. The following review details what I found.

Specifications

The 61 mid-sized keys cover the range C1 to C6 and are just about playable. In addition there are 8 assignable percussion pads (not velocity sensitive). 100 AWM voices (0-99) any four of which can be blended via the vector synthesis mode, (like the Kawai K1's joystick blend), are supported by 50 different pre-programmed accompaniment styles (0-49).

There is a recessed pitch bend wheel with programmable range for 'expressive control', programmable vibrato and reverb effects, plus the usual tempo, transpose and tuning controls.

Connections on the rear panel include MIDI IN/OUT/THRU, a stereo Headphones/Aux output, and a DC power input. Power is supplied either from the internal batteries (six 1.5V, R14 ('C'-size)) or an optional AC power adaptor (9-12V).

Two, circular, car-radio-type speakers are fed from a stereo 3W internal amplifier giving a truely excellent sound from a budget keyboard. To quote Gary "The PSS-790 sounds better through the internal speakers than it does through ____'s PA"! If you want to amplify the outputs, don't let that worry you though, I tried it with all sorts of hi-fi including book shelf, floor-standing and active speakers. The only problem you might encounter is with any family / neighbours who don't like bass drums thundering at 1am in the morning.

MIDI Thru Problem

A word of warning before you turn 'the wick up'. Make absolutely sure that you haven't got a MIDI loop. The 790 has an annoying software MIDI THRU (as well as the usual (and useful) hardware THRU), which operates between the IN and OUT sockets. If the sequencer you are using defaults to a software THRU function on the Atari (Steinberg's Twelve does this) and you have both MIDI leads connected, then you achieve MIDI feedback, and believe me it is just as destructive as the 'microphone too close to the speakers' problem. Hitting any key will cause a loop to start.

During testing, I continually achieved MIDI feedback (and heart failure as 560 Watts of PA took the sound levels to well over 112dB!). Every time Twelve loads a song from disk, the software THRU function re-enables itself, so be warned! With the 790 and both MIDI leads connected, Twelve crashed frequently. Maybe there is a problem with spill-over from the MIDI buffer. The 790's software THRU function really is bad news. The only other time I have experienced a crash with Twelve is when it was connected up to a Korg M1 and that was a one-off, never to be repeated occurrence (probably just a power glitch).

Sounds & Samples

After removing the MIDI loop by disabling 12's software THRU, the 790 was put though its paces with the "Down South" demo file supplied with Steinberg's 12. The demo uses parts and will push some pretty good synths to the limits of their polyphony. The Yamaha, however, had no trouble since it has a massive 28 notes of polyphony!

The sounds are good. Most of the samples are very quiet, even when amplified externally. The range of percussion pre-sets should more than satisfy the average computer musician, and even includes a sampled voice saying "ONE", "TWO", "THREE", "FOUR".

For budding Pet-Shop Boys, there's even an orchestral hit patch, although it's a bit pathetic when compared to an S1000 sample. (Aside: Some Technics' keyboards I have encountered, have a good, sampled orchestral hit sound). String sounds are plentiful and include pizzicato strings / violin, 15 guitar sounds, harp, banjo, ukulele and more. This keyboard's greatest patch is number 00, the acoustic piano. Remember the EMT-10 piano module, whose sounds were pinched out of the £1000 plus Clavinovas? Well the '790's piano is not too far removed in terms of quality from the EMT-10. It is a long time since I've heard an EMT-10 mind you! Still, I have got a real piano to compare it with and for the market that this keyboard is aimed at, I don't imagine that there will be any complaints! What will they be selling this time next year I wonder? Also on the EMT-10, was a choir / chorus sound which many people love to use. The 790 has a similar chorus patch (#64).

The pipe organ sound doesn't rival my synthesized, CZ generated voice, but the other organ variations were all very welcome, especially the two jazz organs. That really tricky sound to imitate, the saxophone, comes in three forms on the 790, Soprano, Alto & Tenor. These are of limited use, offering a brave attempt at imitating an instrument which really requires a velocity and pressure sensitive keyboard of the type used on the Korg M1, which has the closest sax patch I have heard. (Roland's U20 has a usable sax patch but not M1 standard.)

The voice list is too long to detail here, but is split into 11 groups: Keyboard, Guitar, Other strings, Mallet, Brass, Woodwind, Chorus, Synth, Bass, Percussion and Others.

The style list is grouped into: Dance, Rock, Soul, Jazz, Latin, Waltz and Traditional categories. You can Rock, Pop, Dance, Rap, Boogie, Swing, Tango, Waltz, and March to the 790, to name a few. It is these auto accompaniment features which often attract people to the home keyboard instrument, rather than the synth.

If you can only play melodies, then with a bit of practice you should be able to master the Single-fingered chord technique. This will allow you to make best use of the auto accompaniment facilities which are really very good indeed. Until I saw this instrument I was stuck with the impression that auto accompaniment facilities were rather boring, often annoying rhythms of limited use to anyone. Now however, after spending a few agonising months learning some basic Boogie Woogie fingering, I find that I can select style 28 (Boogie Woogie) on the Yamaha, sit back and play single fingered chords and here an accompaniment with no off or missed notes, which with a few pushes of the tempo button, will rival Jools Holland himself (well almost). This does encourage a seriously lazy attitude to music making, so spend the time you would have had to have spent practising the 'left-hand', on reading a good music theory book and learn the wonders of music!

If you don't own an ST-based sequencer, then the built-in 8-track sequencer offered by the Yamaha will offer you a basic compositional aid. You can even dump songs into the Atari afterwards if you want to. I used the external MIDI sync mode on Twelve to trigger the record option on the ST every time I pressed play on the Yamaha. The only problem with this is the 2 bar count-in period used by 12, so I put 2 dummy bars in the 790 at the start of each track. If you use an accompaniment, then this is also transmitted via MIDI to the ST.

Conclusion

To summarise, the Yamaha PSS-790 is an excellent value, MIDI keyboard with all-round ability and a massive 28 notes of polyphony. It has one serious drawback, and that is the lack of a method to disable the software MIDI through. This means that the loop must always be broken at the Atari end if necessary. Not all software defaults to a soft-thru function like Twelve does. Once noted, this can be avoided but it does not excuse Yahama for implementing a function which really isn't necessary on a keyboard which has, after all, a separate hardware MIDI output. Perhaps it has something to do with the fact the the PSS-790's little brother/sister, the PSS-590, hasn't got a hardware MIDI THRU port? The two keyboards do share many features but the extra £80 spent on the 790 buys you better speakers, a MIDI THRU port, 6 more sequencer tracks and 5 more song memories, a pitch bend wheel, another 27 percussion sounds, 8 assignable drum pads, a vector synth mode and another octave of mini-keys.

Audition one soon.

Thanks to Eddy Moors Music, Boscombe, Bournemouth, Dorset for the loan of their one and only (at the time of writing) PSS-790, and Gary who learnt it with me!

Subscription Due?

If your subscription number (top right hand corner of address label) has /11 following it, your subscription is due. Send renewal form, along with £4 to:

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

Atari 2600 Review

By Thomas Holzer

Off The Wall
Atari Corp. - £12.99

The folks say that the legend of the grand master is centuries old. It begins soon after the great wall of China was built. Very near the wall stood an old, ugly wall in which mighty spirits lived. The people of China wanted to tear that wall down, but they were afraid. A grand master with the name of Kung Fu Lu was still a very young boy then and he had been granted five lives. One fine sunny day a mystical dragon appeared on top of the old, evil wall and threw a ball directly at him...

A wall? A ball?

Anyway, young Lu swung his walking stick (a young boy with a walking stick!) and smashed the ball back towards that old, ugly wall. Bricks shattered and magical objects hiding good fortunes and special powers fell from the wall. Kung Fu Lu tried to catch as many as possible and discovered that his fight against the dragon had a costly price. If he missed the ball with his walking stick (grin), he lost one of his lives. To make matters worse, a big cunning blackbird was guarding the wall as well, trying to prevent Lu from smashing the bricks.

But good old (young) Lu tried again and again to smash those bricks and kill the dragon until he became skilful in destroying the brick wall and moved on to gain the rank of master.

The legend tells us ...

To plug your joystick in the machine, shove in the cartridge and play breakout until your VCS explodes. Good sounds, gorgeous colours. Well done Atari, this one's a hit.

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

by **Jimmy Boyce** (CACE)

Reprinted from Atari Interface Magazine, October / November 1989

Well as I promised last issue we are done with editing — HOORAY!!!

Now, let's check the spelling before we print it. That way we won't waste printer paper doing it over again because we misspelled a word.

The Proofreader

The creators of AtariWriter+ call their spellchecker "ATARI Proofreader" <P49>. A rose by any other color still checks spelling. The dictionary disk is very small (only 36,000 words), but you can improve on that by creating your own dictionary disk. We will go over all of that in due time. Believe me, if you don't have your own dictionary disk and you use a lot of technical terms like I do, then I suggest setting one up. It will save you a wealth of time and trouble.

Proofreader is a menu-driven program just like AW+ and just as easy to use. So, let's get started! You have just finished your document and it is edited and saved. Go to the menu and gently stroke the letter [V].

Ah, the purr and whirr of the disk drive...and then, what to your wondering eyes should appear? No, not Santa Claus, but a little message that says,"INSERT PROGRAM DISK THEN PRESS RETURN." Hmmm. OK. stick in the dictionary disk and press RETURN.

Purr and whirr..."INSERT PROGRAM DISK THEN PRESS RETURN." Wait a minute. I did that! "Stupid program doesn't work." That was my first encounter with Proofreader, so I put the program disk back in, did the whole bit again and up at the top of the screen it said in bold and brazen letters "LOADING PROOFREADER" <P49>. And, what to my wondering eyes should appear but a whole new menu <P50>.

First off, let me say that nice key in the upper left corner of the keyboatd ([ESC] <P51>) will get you out of most any jam and get you back to the menu that you are presently staring at (assuming you are following along in AtariWriter+ while reading this. You are doing that, aren't you?).

This menu works just like the Main Menu. When you poke the key that matches the inverse letter on the screen, little messages are given at the bottom of the screen and you do what the messages tell you to do. Pretty simple actually, unless you are like me and can't or won't read the book until you get into trouble.

A couple things to remember, first this is a spell checker and that is it. Second, if the word is spelled correctly but is in the wrong context, Proofreader will say it is a valid spelling (i.e., their and there). Proofreader does not capitalize either, nor does it check punctuation.

This is the nifty one — if you spell a word correctly and that word is not in the Proofreader's dictionary, your word will be considered spelled wrong (another case for a personal dictionary). So, lets get on with it.

Correcting The Errors

The first item on the menu is CORRECT ERRORS <P53>. Lay a finger tip on the letter [C] and you are told where to put your dictionary disk. If you have a one drive system, then it is logical that you must place the dictionary disk in drive one. If you have a two drive system, you will be prompted to place the dictionary disk in drive two (if you have it turned on).

Now the fun begins — notice that your document is on the screen and a little menu is at the bottom of the screen. The little menu reads: C RETURN CORRECT WORD D RETURN SEARCH DICTIONARY RETURN KEEP THIS SPELLING and over on the extreme right hand side it says "CHOICE:"

As your masterpiece blips past your eyes, the words are in their regular appearance on your screen. Oops, what is this? All has stopped and there sits the word TAHT highlighted. Now you have a choice to make.

First choice: pounce on the letter [C] and change the spelling to THAT, and answer the prompts.

Second choice: let's say it is one of those days that you cannot remember how to tie your shoelaces and you cannot remember how to spell the highlighted word.

Bang on the [D] key and do as directed — type in at least two letters of the word, hit [RETURN] and a list of words with those beginning two letters will appear at the top of the screen (if there are any). Once you find your correct spelling, follow the directions at the bottom of the screen and type in the correct spelling.

Your third option is that TAHT is a name or something of that nature and you strike [RETURN] to keep the spelling of that particular word. Proofreader keeps in memory the word TAHT and will accept it for the rest of your document.

For reasons that escape me, you may just wish to see what words you have misspelled without correcting them. So, this is what you do. Load a file from your AW+ file disk and strike the [H] key with whatever finger you strike it with, and follow the prompts on the screen. If you want to quit this process just strike [ESC] <pg52>.

Another thing that you can do is print out a list of the words that you have mis-spelled. For you young people, this is a great feature because you can learn to spell those words correctly for future use.

Turn on your printer. If your printer is already turned on, and you have been using it, then switch it off and then back on. Don't ask me why, but the book <pg52> says, "DO IT!" so do it. When you have loaded the PR and the menu appears on the screen, depress the letter [P]. There goes the printer, and there is your list.

Searching for the Words

The next part may seem a bit complicated, but once you have been through it, it becomes fairly easy. This is the Search Dictionary <pg54> portion of this little treatise. Now, when you are correcting errors and you know the word is correct that you just hit <RETURN> and continue on, or you can hit <C> and correct it yourself and continue on, or you can hit <D> and start the following process: you will be asked what word you wish to search for. That is just like being told by your teacher to look it up in the dictionary. It drives me to distraction.

Finding the Words

However if you think you know at least the first two (2) letters of the word, then type them in and press [RETURN]. At this point, words will begin to appear at the top of the screen, and at the bottom of the screen it will say [RETURN] TO CONTINUE SEARCH, [C] TO CANCEL.

If you type in the letters"TH" then all the words in the PR dictionary will begin to appear at the top of the screen in little GROUPS, and as long as there are "th" words to be viewed, they will appear. When you find your word strike the [C] and you will be asked what word you want to search for, just hit return and then you will get the original menu with the words still at the top of your CRT. At that point you type in [C] again and you begin typing in the correct spelling and then [RF TURN].

Saving Your Masterpiece

As your disk drive whirrs and purrs, the computer continues its search for flaws in your ability to spell. If it should find another, then the whole process shall begin again until you are done with the file.

NOW, save that file! Yes, you can do it while in the PR part of AW+, and you can load your next file to be proofed also. Remember, any time you get stuck, lost or disoriented, just strike the [ESC] key and you are back to the PR menu.

Previews of Coming Attractions - Setting up, saving, and adding to your personal dictionary. See ya next issue!

Atari 8-bit Software News

HiTEC Software, are a budget software company who started on the Atari 8 bit with the re-release of the Boulderdash Construction Kit on tape. As it has sold well, they plan to release the following titles (lets hope these also sell well):

Top Cat	From the comic of the same name.
Yogi Bear and Friends	Features lots more comic stars.
Ruff and Ready	More comic fun.
Yogi's Escape	And even more comical fun.

Harlequin Software started on the 8 bit with an impressive title - **Plastron,** a nice colourful game. Due to the closing down of some mail order companies and no reviews in the major computer games magazines (they all had a copy) it didn't sell too well. I also think it was under rated in 8:16. Its Value should of been 85% and the Game Play 80%. But then, I know the cheat mode.

Anyway, **Harlequin** have told me that there will be no more Atari 8 bit in the future. This is sad because I have seen working versions of **Menace, Zero War** and **Project Xanthein** and they all look very impressive. However, are you ready for this? **Shadow of the Beast** will appear. I have seen the game, it is almost finished, and it looks fab. All the features from the 16 bit version are included, even the music is spot on. **Harlequin** says it might be Easter before its release, but the wait will be worth it and it will be their last 8 bit release unless it sells extremely well (squillions?)

I was going to give you an exclusive interview with some members of the company, but this is no pretty useless, as they will only be supporting the Amiga in the future.

Thomas Holzer

Lynx Software News

The following titles have been, or will soon be, released for Lynx:

Klax	Nice graphics and even better speech (I like that female voice).
Road Blaster	An old coin-op, fun never the less.
Ms PacMan	The original appeared on all Atari consoles years ago. Very addictive.
Xenophobe	Lesser know coin op. Big disappointment on home computers, but not bad on the Lynx.
Slimeworld	Best of the bunch. Very addictive and allow up to eight players connected at once.

Thomas Holzer

The VCS Page

By Thomas Holzer

New titles for the 2600 ...

Road Runner	Atari £9.99	The hit from the arcades.
Doubledunk	Atari £12.99	Good basketball game.
Radar Lock	Atari £12.99	Flight shoot 'em up from the maker of Solaris.
Mouse Trap	Atari £9.99	Old CBS Coleco title. PACMAN clone.
Golf		Golf playing game from Australian company. You have to play it upside down and sing 'I should be so lucky'.
Venture	Atari £9.99	Again, old CBS Coleco title. Good adventure game.
Q-Bert	Atari £9.99	Very old Parker title. Arcade game.
Donkey Kong	Atari £9.99	Even older CBS title. Only two screens.

...and for the 7800 Pro System.

Meltdown	Atari £12.99	Eliminate terrorists and stop nuclear war.
Motopsycho	Atari £12.99	Good racing game.
Basket Brawl	Atari £12.99	Another basketball simulation.
Mat Mania Challenge	Atari £12.99	A good graphics wrestling game. More fun with two players.
Ikari Warriors	Atari £12.99	Arcade conversion shoot 'em up.
Impossible Mission	Atari £12.99	Maybe it arrives ... Maybe is doesn't!
Turrican		I haven't seen this one yet, but if it's like the computer version ... can't wait.
Commando	Atari £12.99	Vertical Ikari warriors type game.
Tower Topplers	Atari £12.99	(Nebulas) Hewson game. Miles better than the computer version.
Planet Smashers		Fast shoot 'em up with add on weapons.
Gauntlet	Atari £TBA	Available at last?????
Crack'ed	Atari £9.99	Platform type game.

Where to buy?

In answer to the question I get most from people, "Where do I buy 2600 or 7800 software", I will shed a little light into the dark and damp dungeons where a shady evil looking dealer hands outs (in exchange for souls) his latest software creations. Well, not exactly, as its slightly more easier than this, so read on ...

Firstly, most of the software stores either stock or will order VCS cartridges for you. My local computer store, here in Reading, always has a decent stock of both formats. And, if he doesn't stock, it he will order it. All computer shops who are connected with the *Software Link Chain* will order them for you. A look at the recent retail price list produced by Leisuresoft Limited lists over 70 titles.

Secondly, all the Toys 'R' Us shops have a huge display of 2600 and 7800 carts, and these shops are all over the country.

Thirdly, but not so effective, is car boot sales. Go around them on a sunny Saturday morning and you will be surprised a what you can find second hand at a very cheap price.

Lastly, use mail order. Telegrams in Leicester have a whole catalog on 2600 and 7800 stuff and they will even import titles from the states for you.

I hope this will help some readers out there. The next problem arises when it comes to Atari 8-bit software because a lot of shops don't bother any more. Again, mail order is a good choice, Miles Better Software and Gralin International, or shops linked with the software link chain. Forget Toys 'R' Us, except for a few £2.99 cassette title.

Pro 7800 Review

Galaga from Atari

Blast 'em!!!

You are under attack (again)! The vicious 'Galagans' are reclaiming planets lost during an old interstellar war, and now they want your home planet. But, here you are, ready to fight to the bitter end to protect it. Ready to join the intergalactic warrior fleet, and without training, ready to accept your first combat mission.

Shaking in your boots, you accept the assignment and board your intergalactic command ship. It's now up to you ....

Galaga is one of the first cartridges released for the Pro System and even if it's a little old (1984), it's still a beauty. You start the game with four battle ships: one at the line of battle, the others in reserve. You can move your ship to the right and left to dodge the Galagan fire, while at the same time firing intergalactic missiles to wreck havoc within the approaching wave. You lose a ship whenever it is hit by an enemy missile or by a collision with a Galagan warship. The Galagans attack in a series of waves with the flagships never flying alone, being flanked by protective escorts in different coloured uniforms.

An entire fleet of Galaga invaders begins flying in formation, then instantly plunges towards you bombarding your ship before they rush back to the formation. If you don't shoot them first, each wave the attacks are faster and faster.

The flagship deploy blue macro-beams that can capture your command ship. To rescue the hostage, hit the flagship only while it is attacking, or you will destroy your captured ship as well. Once you regain your ship you will have two ships moving and firing in sync, giving you more attack power. Every so often you will face a challenging stage. Five groups of eight enemy ships flying in various patterns. Shoot as many ships as you can before they leave the screen and score bonus points. Hit all 40 ships for a 10,000 bonus.

The game can be played with either joystick or joypad and you can play with one or two players with three difficulty levels: Novice, Advanced or Expert. The Galaga ships are nicely coloured and a few musical scores can be heard. Animation is also very smooth with absolutely no flicker no matter how many sprites are on the screen. If you play the game in expert mode it is fast, not only that, it is very fast.

All I can say is. "If you own a 7800, go and get it".

User Group File

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, Cornwell 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

Notice To All User Groups

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.

Special Interest Groups

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

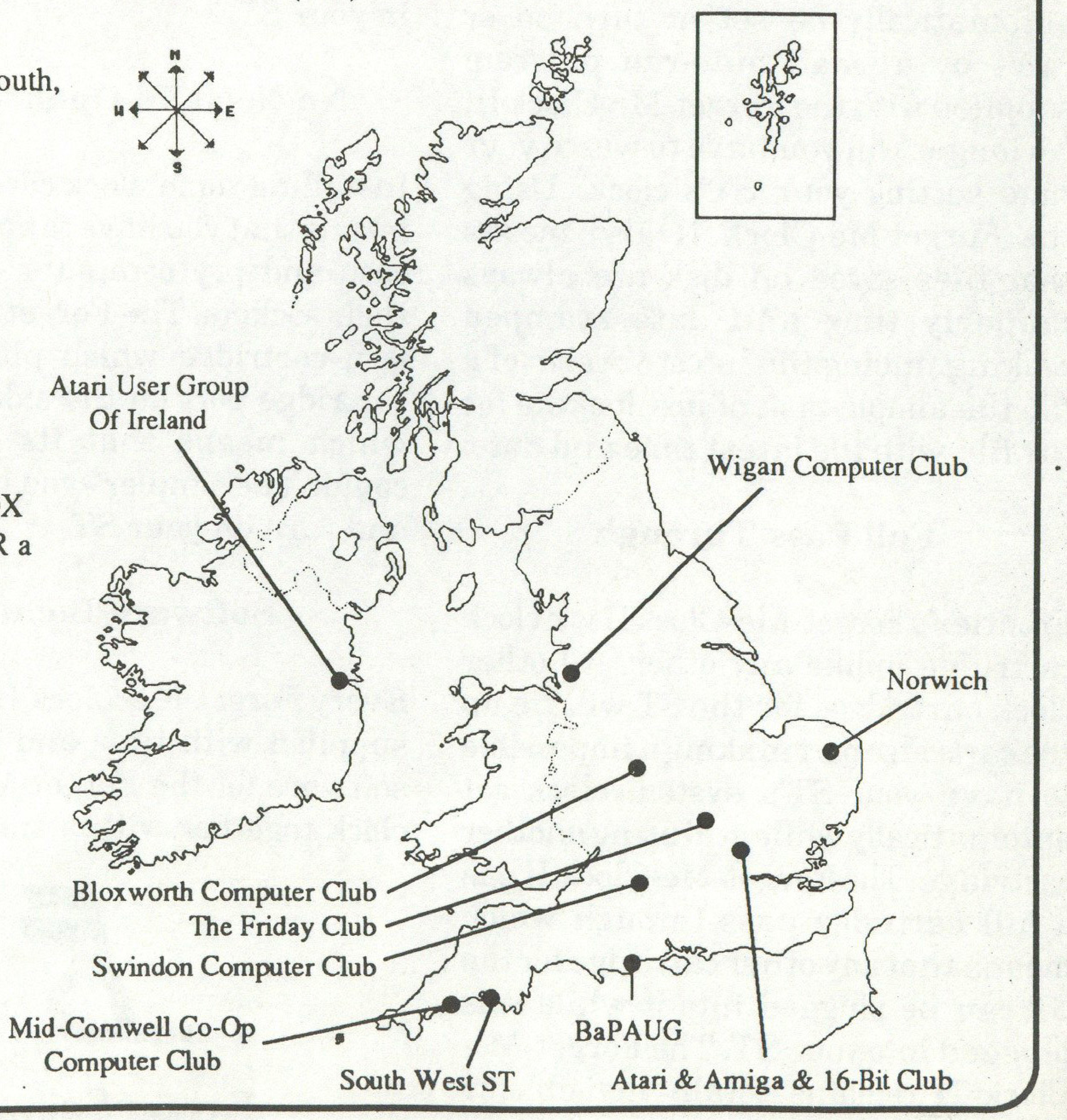
Name : **Johannesburg Atari Computer Enthusiasts (JACE)**
Address : 2 Whitehall Street, Hurst Hill, Johannesburg, South Africa, 2092

Name : **Maryland Atari Computer Club (MACC)**
Address : 8591 Wheatfield Way, Ellicott City, Maryland 21043, USA
Telephone: (301) 461-7556
BBS: Pizza's Place - (301) 325-1027
Inner Limits - (301) 356-5112
Newsletter: M.A.C.C. News

Name : **North East Atari Team (NEAT)**
Address : P.O. Box 18150 - 0150, Phila., Pa. 19116, USA
Newsletter: The Atarian

Name : **Northern Virginia's Atari Computer User Group**
Address : 8612 Thames St., Springfield, VA 22151, USA

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



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

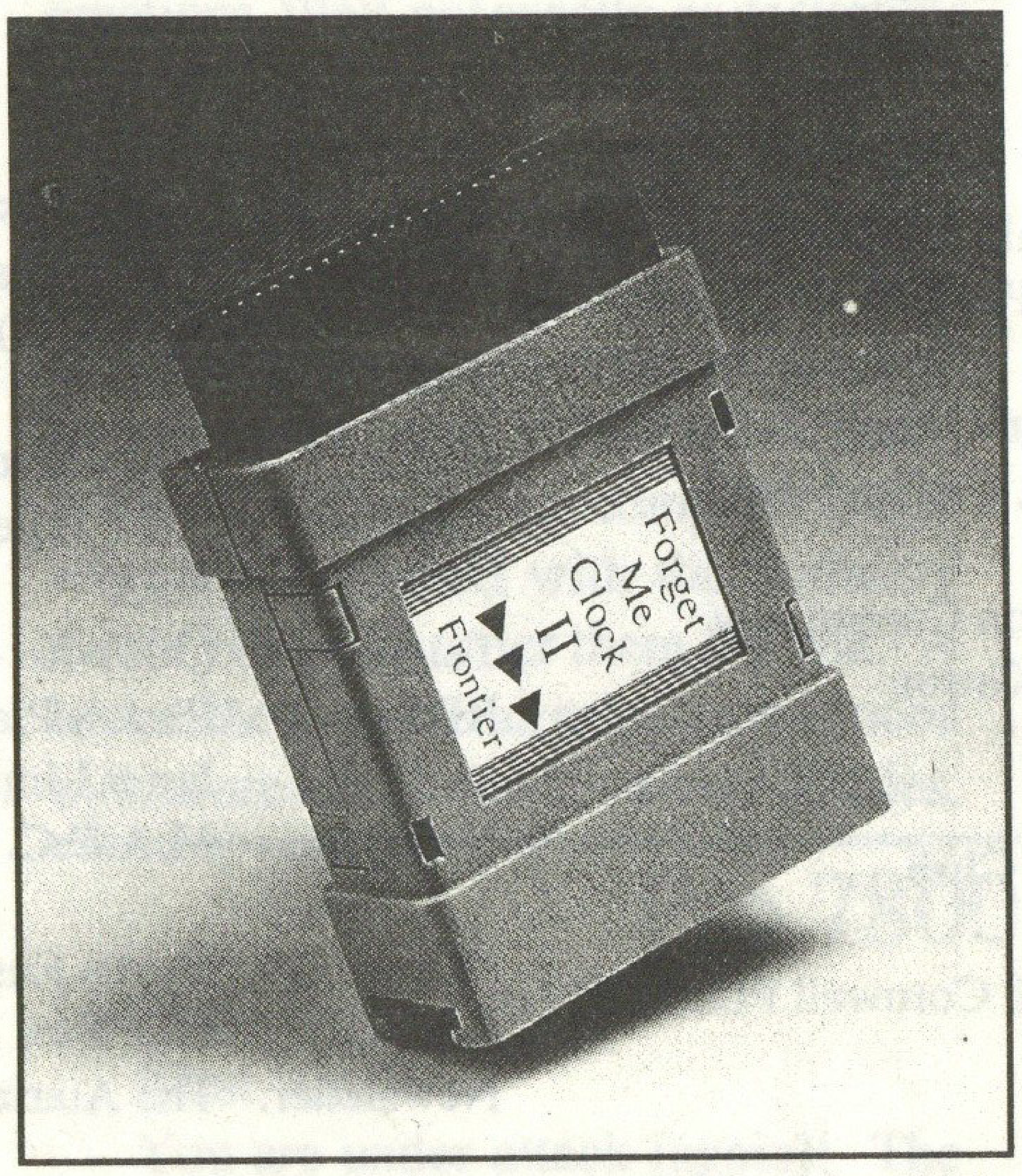
Frontier's Forget-Me-Clock II is the answer

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 Cartridge With 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 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

P.O. Box 113, Harrogate, North Yorkshire, England HG2 OBE. Phone (0423) 567140/530577. Fax (0423) 522874.