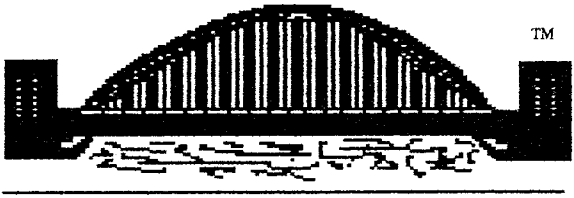
TYNE & WEAR



ATARI 8-BIT

USER GROUP

Issue 27

May/June 1997

Published by TWAUG

Edited & Printed by AMG

Publishing

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

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

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

Do you need to **Contact** anyone at TWAUG for a chat then phone

Alan Turnbull on: 01670 - 822492

or Max on: 0191 - 586 6795

Our Postal address:

TWAUG

c/o J.Matthewson

80 George Road

Wallsend, Tyne & Wear,

NE28 6BU

Max's Comments

I am afraid this issue will be rather a bit late in reaching you and I apologize for it. It was partially due through lack of material, for the newsletter, and I also had a visitor staying at my house for 14 days.

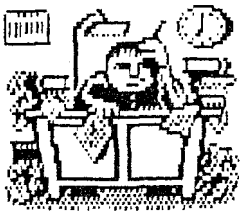
I had problems in assembling material for the newsletter lately, there doesn't seem to be much new material for the 8-bit classic in circulation at the present time. I remember the time when I had a lot of material on hand, but lately there is just nothing at all.

But do not despair, I've just received some stuff, which I believe will fill a couple of issues. The 8-bit Computer article starting on page 24 is part of it, but I haven't had the time yet to browse through the other material.

So stay with us and please do write and let us know what you think of my material choice, your letters will of course be published in the newsletter, unless otherwise requested.

I am hoping to receive some mail for the next issue.

TWAUG's address



PUBLISHING!

This new look newsletter is set up with the Desktop Publishing program "TIMEWORKS 2", on the Mega 1 ST with 4 meg memory. Files are converted to ASCII and transferred to the ST with TARI-TALK. Those files are then imported into the DTP and printed with the Canon BJ-30 Bubble Jet Printer at 360 dpi, with excellent result.

TWAUG

NEEDS  YOU

TWAUG subscriptions

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

Issue 28 is due mid-July 97

ISSUE CONTENT

ATARI8 SIO - RS232

These first two articles explain the porting of 8-bit material to the ST via the Serial port and also include a diagram to show you how to make your own Atari8 SIO to RS232 Serial Interface.
The articles and diagram have been contributed to TWAUG by Roger Lacey.

How to cheat at Comms with Your Atari 8-bit.

By Roger Lacey, LACE Secretary.

First published in BOOT the LACE Newsletter No.2/1992 Amended April 1997 by the author for TWAUG.

This article attempts to explain the inner workings of the Atari8 SIO to RS232 interface diagram attached. To be more specific, we are cheating the electrical interface standard of the V24 (RS232) connection. This is a European CCITT specified standard derived from the American RS232 standard for communication devices of 19200 Bits Per Second (BPS)and below. The electrical standard V28 specifies a voltage level of plus or minus three to twenty-five volts to indicate the Space or Mark or Binary O or 1 respectively. You may notice that this is inverse to normal logic notation where the Space and Mark is considered to be a Binary 1 and O respectively.

This brings me on to the components used in this 'cheat' serial interface. The MC1489 is intended to be used as a quad line receiver, receiving the V24 plus and minus voltages and converting them to logic levels of O Volts and plus 5 Volts and inverting them. This it does well, but it is also used to drive the V24 receive line as though it too were providing true plus and minus voltages. This is not the case of course as only the O Volts and 5 Volts are output. The reason it works is simply that a O Volts level is recognised by most modems and other communications devices as representing a negative voltage. Most devices use similar components to the one we are using and they all do the same thing to the negative voltage when it is processed, and that

is to short it to ground! Not directly, but via a diode and serial resistor which limits the current.

This then allows our 'cheat' to work and it is fine for modems and computers alike I use it on a modem, an AtariST and a Zenith PC. Only ancient devices will object to the lack of the negative drive voltage perhaps some printers or modems ten or more years old. The V24 lines lacking here are:-

1. Request To Send (RTS) which is often used for handshaking for flow control, mainly to indicate and control buffer status within the computer.
2. Data Terminal Ready (DTR) which under normal circumstances can control a modem's line status (on or off-hook) and can also be used for flow control depending upon the device attached. In our instance this is clamped high all the while the computer is switched on.
3. Ready For Sending (RFS or CTS) which is the Data Communications Equipment (DTE) acknowledgement of the RTS signal. But many computers do not support this line anyway (including the AtariST).

The other main component is the 74LS126, also a quad device but is a non-inverting buffer (the 1489 does the inverting) necessary to provide the Atari SIO with a tri-state interface such that no loading is imposed on the lines used when other devices are active. This design relies upon the fact that a cassette drive will not be attached to the SIO as the R: handler, of which there are several (Rverter etc.), steals the motor drive circuitry to enable and disable the SIO side of the interface.

The R: handlers are written for various speeds and purposes but load instead of the Atari 850 built-in R: handler. So basically this is a bare bones interface which is far from full specification but does allow most devices to communicate quite happily provided software flow control (DC1/DC3) is enabled, and even if it isn't unless you are working flat out at 19200 bps you probably won't need it. Try getting an Atari ST working flat out at 19200! You won't unless you are lucky. The Atari8 is faster in this respect!! Although this does depend upon the software used to drive the interface for which I cannot recommend highly enough, Robert Puff's excellent Shareware comma package Bobterm 1.21/1.22. Available from LACE or TWAUG.

Construction

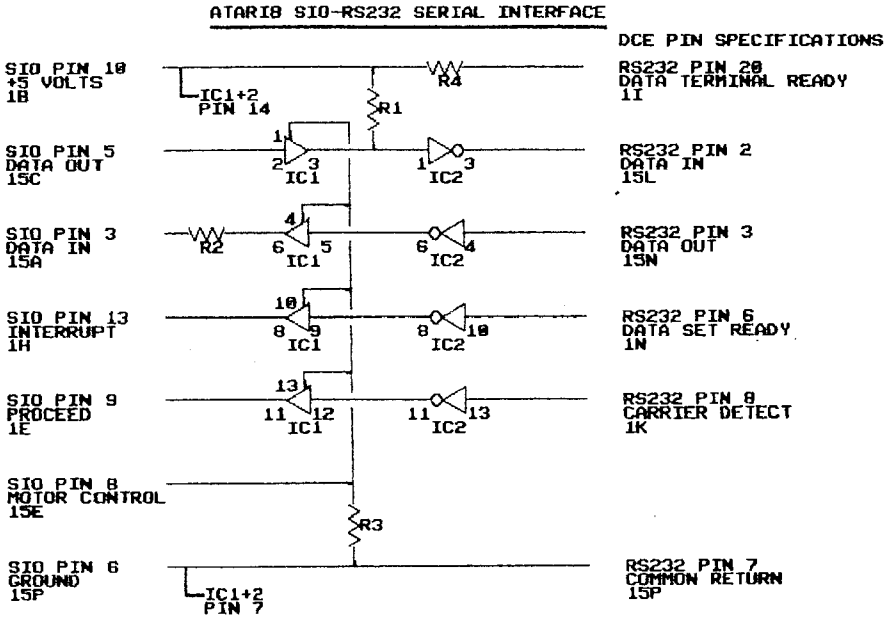
This little project needs normal

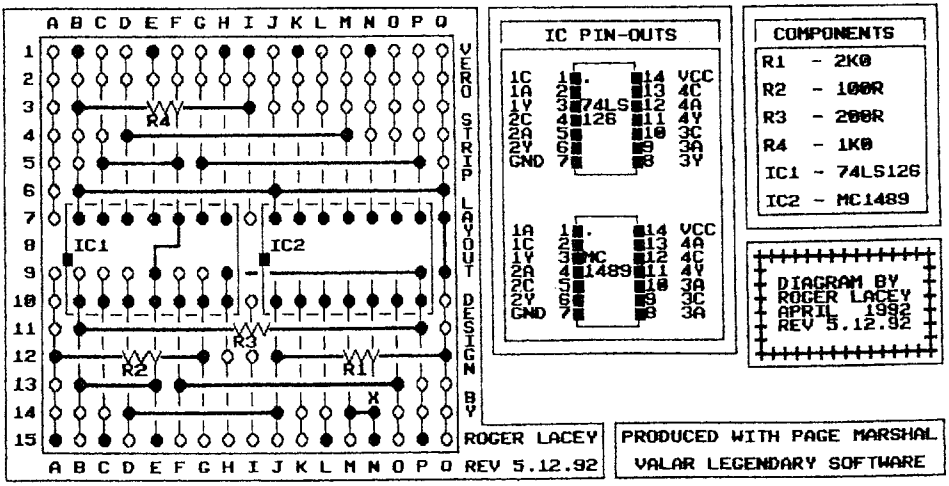
dexterity with a soldering iron and solder. A mistake here could do permanent damage to the Atari 8-bit computer you intend to connect to and no responibility can be accepted by the author or publisher of these details. Accepting that, this interface is by no means difficult to make and get working first time. I have used Vero Strip cut to size width-ways to wire the components as shown on the diagram. Vero Strip comes as a fixed size and has enough board to make three of these devices. This works better than plain Vero Board as most tracks need not be cut. Layout is far from critical and all components are very tolerant to abuse but IC sockets are recommended never-the-less. Don't forget to run the O Volt Lines before they are mounted.

I cheated and used Soldercon IC pins as opposed to sockets. These come in strips of one hundred or more and consequently can be used for ICs of any size. Vero pins are used to terminate the cables connecting to the SIO and V24 sides. These can be difficult to install as they require pressing into the hole with force, but once in place are very secure!

I recommend utilising an existing SIO lead and teeing into it by stripping the insulation on the cable centrally, and tapping the required wires by attaching spur wires. This can then be left in the SIO chain all the time and leaves no 'dead end'. The V24 plug is a standard 25-way 'D' type 'solder bucket' type with just the required pins connected via a multicore cable. A standard 25-way 'D' type shroud holds it together and provides the locking screws into the modem socket. That's about all there is to say really. Housing this board is left to your discretion as there is a large choice of Vero boxes and other containers: perhaps even building this inside the Atari would suit you better as there is plenty of room for this in all machines and substituting the 25-way plug for a 9-way, and mounting this on the case, is an acceptable practise. Many IBM PCs and clones use the smaller plug for this purpose and so does the Atari 850 albeit female!

Good communicating!


ATARI8 SIO-RS232 SERIAL INTERFACE
DCE PIN SPECIFICATIONS
SIO PIN 10
+5 VOLTS
1B
IC1+2
PIN 14
R4
R1
RS232 PIN 20
DATA TERMINAL READY
1I
SIO PIN 5
DATA OUT
1SC
IC1
IC2
RS232 PIN 2
DATA IN
1SL
SIO PIN 3
DATA IN
1SA
R2
IC1
IC2
RS232 PIN 3
DATA OUT
1SN
SIO PIN 13
INTERRUPT
1H
IC1
IC2
RS232 PIN 6
DATA SET READY
1N
SIO PIN 9
PROCEED
1E
IC1
IC2
RS232 PIN 8
CARRIER DETECT
1K
SIO PIN 8
MOTOR CONTROL
15E
R3
SIO PIN 6
GROUND
15P
IC1+2
PIN 7
RS232 PIN 7
COMMON RETURN
15P


A B C D E F G H I J K L M N O P Q
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
R4
IC1
IC2
R3
R2
R1
X
VERO STRIP LAYOUT DESIGN BY
ROGER LACEY
REV 5.12.92
IC PIN-OUTS
1C 1A 1Y 2C 2A 2Y GND
74LS126
1A 1C 1Y 2A 2C 2Y GND
MC1489
VCC 4C 4A 4Y 3C 3A 3Y
COMPONENTS
R1 - 2K0
R2 - 100R
R3 - 200R
R4 - 1K0
IC1 - 74LS126
IC2 - MC1489
DIAGRAM BY
ROGER LACEY
APRIL 1992
REV 5.12.92
PRODUCED WITH PAGE MARSHAL
VALAR LEGENDARY SOFTWARE

ATARI CLASSIC TO ATARI ST FILE TRANSFER

By Roger Lacey, LACE Secretary.

This article was first published in LACE Newsletter Number 5, June 1994. Updated by the author for TWAUG April 1997.

This is my method of connecting the two different Atari computers that I own. This allows files generated and held on one type of disk to be ported over to the other machine to be archived or otherwise worked upon.

Specifically, I transfer documents generated on the Classic to the ST to allow import into the Newsletter. Also I produce files with Bob Puff's Discomm 3.2 on the Classic, which allows a whole disk to be stored as a file image, then transfer them to the ST to be LH1 compressed and stored as archives on larger capacity ST disks. These archives can then be transfered back to the Classic at any time and uncompressed to re-create the original disk. This I have discussed before - onto what I do to achieve this.

Ymodem comms package

Bobterm 1.22, by Bob Puff, is my recommended Shareware comms package for the Classic. This drives my Black Box interface or my SIO-RS232 interface which featured in the LACE BOOT! magazine 2/1992 and which is included here for reference. Bobterm offers a selection of bit rates including full-duplex 19200 Bits Per Second (BPS) transfer rate. A lesser R: interface such as the Atari 850 will require a lower transfer speed of 9600 BPS or less.

This program will work happily with all forms of Atari 8 serial interface such as the ICD P:R: interface, MIO, Black Box etc. You just need to load the correct R: handler which is a small program that tells the computer where the serial port is and handles the translation options. The Black Box optionally automatically installs its own R: handler on installation (P: handler too) but the 850 and clone devices are booted from the comms program or run as an Autorun file, dependant upon the way the package handles them. Atari modems and Rverter (Datari in the UK) are also supported. The setup options should

ATARI CLASSIC TO ATARI ST FILE TRANSFER

be 9600 or 19200 bps, Atari translation. This ensures 8 data bits, 1 stop bit, no parity for file transfer compatibility.

At the ST/E end, nearly any comms program bar the Europress offering will be suitable. What we are looking for is support for a file transfer protocol called Y-modem batch to match the Classic end. Sometimes Y-modem is incorrectly refered to and is really X-modem 1K. If it doesn't support batch files it is not Y-modem. I use an old commercial Antic program called Flash 1.6 to perform transfers to and from Bobterm 1.22, but Uniterm which is a Public Domain program would be suitable and there are many other similar programs available. (Freeze Dried Terminal rings a bell). You do not need bells and whistles to talk to a computer three feet away! The setups here should reflect the speed of 9600 or 19200 BPS, 8 data bits, 1 Stop bit, No Parity checking.

Baud Vs. Bit

Often, comms programs incorrectly refer to the Baud rate as a measurement of Data TERMINAL (DTE) speed. It is important to differentiate between the two as the Baud is used as a measurement of LINE speed only and refers to the smallest signal element. Most modems use a line speed of up to 600 Baud but the DTE speed can be in the order of 38400 BPS with an actual modem speed of up to 14400 BPS. This is known as V32bis and is very common on more recent modems.

Cables

The link between the two computers is similar to the serial link between any two computers. The ST end is a standard RS232/V24 male D25 connector. (Note that the Mega STE is DM9). The Classic end is the same assuming my interface is used. Now all that is required is to link the Transmit Data pins to the Receive Data pins of the D connectors, connect the Common and perhaps link the control lines together, but usually this is not essential and depends upon the comms package in question. I have built two female D25 connectors into a small D25 purpose made box. This contains both connectors and the cross-over wiring. Connecting this to one computer and running this via a straight cable to the second completes the connection.

ATARI CLASSIC TO ATARI ST FILE TRANSFER

DB9F	or DB25F (Mega)(ST/E)			DB25F	or DB9M (850)
3	2	Tx Data	- Rx Data	3	4
2	3	Rx Data	- Tx Data	2	3
5	7	Common Return	- Common Return	7	5
*7	4	Req. To Send	- Clear To Send	5	8 *
*8	5	Clear To Send	- Req. To Send	4	7 *
*1	8	Carrier Detect	- Data Term Rdy	20	1 *
*4	20	Data Term Rdy	-		
Carrier Detect		8	2 *		

* Not necessary for most comms packages. But you may need them for serial printers and the like.

Before attempting a file transfer at this point it is worth actually setting both Terminal programs to terminal emulation and typing characters to confirm that a good connection is made, at the correct speed. Typing on one computer should reproduce on the others' screen.

Now selecting Y-modem batch upload and a batch of files to transfer on one computer, and selecting Y-modem batch download on the other, will result in the files being received and therefore duplicated on to the receiving disk drive automatically. Just make sure the receiving disk has sufficient space! Job done.

For the ST user with no Manual

For the new ST users here are some important points that will help you to run your machine.

What is GEM? Gem stands for Graphics Envirenment Manager. The GEM Desktop conveys and manages information by means of graphic images and words instead of words alone.

The GEM Desktop consists of the Menu Bar along the top of the screen, three Icons along the left side and the remaining screen space. The Menu Bar gives you access to the utilities and applications available from the GEM Desktop. With the Floppy Disk icons you control and manipulate the

For the ST user with no Manual

information stored on your disks. The Trash icon (Waste Bin) lets you dispose of your unwanted information.

If you are using a colour monitor or a TV and you switch your computer on the first time, without a disk in the drive, the screen will be in Low resolution, that's the default screen. That screen is alright for most of the games, but to get a wider screen you need to be in Medium resolution. Wordprocessors and Desktop Publishing program only run on Medium or higher resolution.

To change the screen to Medium resolution, move the mouse pointer onto Options in the Menu Bar. Click on Set Preferences option, a Dialog Box is displayed, three options will be highlighted in that box, two Yes's and one Low, click on the Medium box and then on OK to confirm it and close the Dialog Box. If you select Options again and click on Save Desktop option the screen will be save in file called DESKTOP.INF and every time you re-boot your computer with that disk in the drive the saved screen will be displayed.

If you have only one disk drive on your machine and you want to copy a file from one disk to another, is to open the disk drive highlight the file and drag that file or program onto drive B and make sure the drive is also highlighted. A Dialog Box will be displayed, click on copy and the file will be read, next you will be asked to insert disk B into drive A, just swap the disk with the one you want to copy to and click on copy, that's all there is to it.

To format a disk click on the Floppy Disk A, make sure a disk is in the drive, move the mouse pointer to File on the Menu Bar and select the Format option. When the first Dialog Box is displayed click on "continue", the next box displays the format options, single sided, double sided, how many sectors to the tracks and tracks. Most drives installed in the ST's will let you format your disks in double density, so select that option by clicking on it, the default sectors is 9 which will be highlighted and 80 tracks, that will give you 726016 bytes available on a disk formatted in double density and 9 sectors to each track of 80 tracks. With most drives you can format the disks with 10 sectors and 81 tracks.

An important mouse technique is "dragging". you can move an icon

For the ST user with no Manual

from place to another on the desktop. To drag an icon, select it with the mouse pointer press and hold the left mouse button while moving the mouse pointer. An outline of the icon follows the pointer and when you release the button, the icon appear at its new location.

How to control the mouse pointer with the cursor keys. How many of you readers read the manual right through, from the beginning to the end? I bet not many of you do that, and I also bet that not many read through it more than once. I read through every manual at least twice when I first receive it before I attempt to run the machine or program or whatever it is. Why I'm asking this question is, would you know how to control the mouse pointer when the mouse fails you. Only a couple of weeks ago my mouse packed up on me, I was working away and using the right mouse button a lot, when all of a sutton nothing happened when I pressed the button. I was lucky that every time I used the right button to verify the move an alert box came up the same time to click on with the left button for verification. It was the next day when using the mouse again, when I was opening a program, when the mouse failed me completely it was then I had to use the cursor keys to move the pointer and select the items. And here is how it's done:

Hold down the [ALTERNATE Key] and use the arrow to move in the direction indicated by the arrow, by eight pixels.

Hold [SHIFT] [ALTERNATE] and any arrow keys to move one pixel in the direction indicated by the arrow.

Hold [ALTERNATE] and press [INSERT] to select an icon or file, when pointer is above it. This combination is the same as clicking with the left mouse button.

Hold [ALTERNATE] [INSERT] and use any arrow key to drag icon or file in direction indicated by arrow. This is the same as pressing and holding the left mouse button and dragging.

[ALTERNATE] [CLR HOME] performs the right mouse button functions.

After my old mouse packed up on me and even so I'd read the manual at least twice I couldn't remember what combination of key presses did what, I put that down to old age.

Opening Windows:

When opening a window place the

For the ST user with no Manual

mouse pointer over the icon and click to highlight it then double click to open the window, or if your fingers aren't nimble anymore, move the pointer to the Menu Bar and move to the File option. The File Menu contains a number of options, click on Open and with the Floppy Icon still highlighted, the window will open displaying the content of the floppy disk A. This can be viewed as icons or as text. Go into View Option to change this option to your preference.

To size the window, move the pointer onto the small black box at the bottom right corner and drag to right, down, up or left and it will change accordingly. For quick enlarging to full size, move pointer onto top right small black box and click once and hey presto the window is enlarged to the full size of the screen, click again and it returns back to its previous size.

The left hand top small black box closes the window, or when a folder or a file is open it re-runs the disk to display the previous content, or full directory, depending what is being viewed.

The shaded bar across is the move bar, place the pointer on the shaded bar and drag the window to its new position.

I hope this short article will help those without a manual to get going, with a little experimentation I am sure you all will find your way around the ST soon. But should you need more information on the working of your ST please do write and let me know and I'll see what I can do.

ST MAGAZINES

A few months ago I reported that all ST Magazines closed down. Fortunately a new magazine has emerged, it is called ATARI COMPUTING, it is published bi-monthily and the next issue is number 5, the price is £3.00 an issue and is only available by subscription.

There are still a number of outlets for the Atari ST available to purchase your software and/or hardware for upgrading your ST.

This section is for anyone thinking to upgrade to the Atari ST/e, TT or Falcon systems.

GRAPHICS MODES

I don't know what is going on at the moment it looks as if nothing is moving in the Atari scene, no new material at all, but we are hoping to receive some kind of articles in the next few days, hopefully.

In the meantime, while we are hanging about waiting, let us look at the Atari's various graphics modes and see what each is capable of displaying.

By Dave Russell.

You may have read some of this information before, or you may have discovered some of it by accident. If we're going to look at the Atari's graphics modes, we might as well start as the machine itself does - with Mode 0.

Get in the right Mode for screen displays

This is the default mode, the one which appears when the machine is switched on and no program is present. It's a large blue screen with a black border and the text appears on it in light blue.

Although Mode 0 is one of the graphics modes and is invoked by the command GRAPHICS 0 (or GR.0), it is more usually thought of as a text mode. It is used mostly for entering and displaying the letters and numbers which make up text, although it does have some 'graphic' capabilities as we'll see later in this series.

The blue rectangled screen can be thought of as the piece of paper on which you write your text. In fact, it's more like graph paper, divided up into little boxes. Many forms that we have to fill in have rows of boxes where you write your information and usually there is an instruction to write only one character in each box.

GRAPHICS MODES

A text screen is like that and Mode 0 has 24 rows of boxes, with 40 boxes in each row. However, you may have noticed that the word READY doesn't appear at the leftmost edge of the screen.

Unless you do something to alter it, the default setting is for 38 characters in a row. The 'missing' two characters form a margin down the left hand side of the screen.

```
10 FOR A=1 TO 4
20 PRINT "1234567890";
30 NEXT A
```

Listing I

```
10 GRAPHICS 0
20 FOR A=0 TO 15
30 SETCOLOR 2,A,8
40 FOR DELAY=1 TO 300:NEXT DE
LAY
50 NEXT A
60 GRAPHICS 0
```

LISTING II

It's quite easy to give yourself the 40 characters. The size of the left margin is held in memory location 82 and you can see how big it is normally by typing:

PRINT PEEK(82)

and pressing Return. This should print the value 2 on the screen.

We can alter the contents of a memory location using the POKE command, as long as we specify what number the enter there. So if you type:

POKE 82,0

and press Return, the word READY will appear at the very edge of the screen.

If you entered Listing I, it should still be in memory unless you've since typed New. Run the program again, and this time the 40 numbers will fit neatly across the screen.

We can work the same magic on the right hand margin using memory location 83. If you enter:

PRINT PEEK(83)

this will usually return the value 39, indicating the rightmost column. Remember, counting starts at 0, which is why location 83 doesn't contain the value 40.

If you enter:

POKE 83,10

and press Return, this will set the righthand margin to column 10. Assuming that location 82 still holds the value 0 that you Poked in earlier,

GRAPHICS MODES

the effect of this poke to location 83 is to give you a screen which is effectively only 11 columns wide.

The blue rectangle stays the same size - it's just that you can't type on so much of it now. to see the effect, try running Listing I again.

If you've been playing about with locations 82 and 83 and want to get back to the default values, you can always press the Reset key. This will (literally!) re-set the values without losing any program you had in memory.

Before leaving the text aspect of Mode 0, try:

POKE 755,4

Normal service can be resumed as soon as you return location 755 to its more usual value of 2 - or press Reset if you find it hard to type in Outback Mode!

The Atari has several registers which hold information about various things. Five of these contain information on colour in which the Mode 0 screen appears.

We could POKE particular locations to change colour, but Atari Basic offers us a more elegant method - the SETCOLOR command.

To use this command we need to know three things: the register number which we want to affect, the colour number which we want to put there, and how bright we want the colour to be.

These three parameters must follow the command in the order in which I've given them.

The default colour for register 2 is colour 9, the blue you know and love. To change this colour, all we need to do is key in:

SETCOLOR 2,4,4

If you've just entered this your screen is now ablow with colour 4, or pink as we call it. If you want to get rid of colour altogether, try:

SETCOLOR 2,0,0

This produces a very dark grey and the Mode 0 screen blends with the border to give the effect of a much larger screen. Of course, text can still only be entered in the area which is usually coloured.

If you enter Listing II and Run it, you'll see the screen cycle through the range of 16 colours available before returning you to the default colour. Notice that because Listing II uses the same line numbers as Listing I, it will overwrite it in memory.

GRAPHICS MODES

Only the second parameter, the one controlling the screen colour, is varied. The luminance remains at value 8 for each colour displayed.

Incidentally, the purpose of line 40 is simply to keep each colour on the screen long enough for you to see it. If you remove line 40 and run the program your screen will appear to flash as the colours are displayed at very high speed.

The colour information for the Mode 0 border is held in register 4 and we can alter this in the same way as the text screen. Enter:

```
SETCOLOR 2,0,0
```

and you will have a completely black screen. Now try entering:

```
SETCOLOR 4,9,4
```

This alters the register controlling the border colour so that it now contains colour 9 - the colour we usually associate with the text portion of the Mode 0 screen.

Well, we've had upside-down text so we might as well have the usual colour relationships reverse too! Reset will restore the registers to their default values...or maybe you prefer having white text on a black background.

We can make the border cycle through the available colours by simply changing line 30 in Listing II to read:

```
30 SETCOLOR 4,A,8
```

That is, by changing the colour in register 4, we alter the border colour rather than the screen colour.

The brightness of the letters on the screen is controlled by the contents of register 1. However, the colour of the letters is always the same as the colour of the text screen.

If we set the luminance parameter of register 1 to a bigger number than the luminance of register 2, then we (probably) get 'light' text on a 'dark'background.

If we set register 1's luminance to a smaller number than register 2's, we (probably) get 'dark' text on a 'light' background.

To see this effect, press Reset and then enter:

```
SETCOLOR 2,1,8
```

This produces a gold screen with paler text. If you now enter:

```
SETCOLOR 1,1,4
```

the screen stays the same but the text changes to a darker colour.

GRAPHICS MODES

So why use the word 'probably'? The luminance parameter can range from 0 to 14, but only even numbers are valid. If you enter an odd number, the luminance is set to the number you entered minus 1.

This means that if you set register 1's luminance to 9 and register 2's luminance to 8, then both are effectively set to 8.

'Thank the Atari for a decent reset function which many other micro owners would envy'

If you change line 30 in Listing II to read:

```
30 SETCOLOR 1,1,A
```

and run the program you'll see the text cycle through the luminance values. Strictly speaking you should change line 20 to read:

```
20 FOR A=0 TO 14 STEP 2
```

but it won't do any harm if you don't bother. Each luminance will be displayed for twice as long as each colour was displayed previously.

For the final disappearing act, enter:

```
SETCOLOR 1,1,4
```

Assuming you were back in the blue, the luminance in register 1 is now the same as that in register 2. This means that the text is now displayed at the same brightness as the background.

Unfortunately, the practical effect of this is that the text is rendered invisible. Unless you enjoy flying blind, press Reset once again.

And while you're pressing it, say a quiet 'thank you' to Atari for a decent reset function which many other micro owners would envy.

We now take a look at two other text modes, Graphics 1 and Graphics 2.

As you might expect, there are some differences between Modes 1 and 2, but they're similar enough for us to consider them together.

You may recall that the Mode 0 screen was like a piece of graph paper with 40 columns and 24 rows. If you imaging the columns stuck together in pairs, you've got something similar to Mode 1.

GRAPHICS MODES

TAKE A LOOK AT MODES 1 and 2

To be more accurate, the Mode 1 screen has two parts. The top part is like pairs of Mode 0 columns stuck together, while the bottom part is exactly the same as Mode 0 columns.

That is, Mode 1 has a 20 rows by 20 columns section, immediately underneath which are four rows of 40 columns.

The conceptual move to Mode 2 simply requires you to imagine the Mode 1 "fat" rows stuck together in pairs. This gives a 10 rows by 20 columns section, again with four rows of 40 columns beneath.

You can see the way these sections are separated quite clearly by typing GRAPHICS 1 and pressing Return. Assuming you were in Mode 0 before you did this, your screen has suddenly been transformed into a large black area with a small strip of Mode 0 blue at the bottom.

This strip is four lines deep and you should see the word READY on the second line and the cursor immediately beneath it on the third line.

If you press Return a few times, the word READY will move up the screen. On the third press, it will disappear.

If you had a program in memory and typed LIST now, the listing would be displayed on these four lines only. It's not very useful being able to read only four lines of a program at a time, which is why program writing and debugging tends to get done in Mode 0.

The four lines are what is sometimes called a "text window" and anything you PRINT will appear here. Try the simple exercise in Program I. When you run it, the cursor will end up on the bottom line.

```
20 GRAPHICS 1
30 PRINT "MODE 1 TEXT WINDOW"
```

Program I

This means that if you'd add another line of print, the first would scroll out of the window. Add a line 40 to Program I as follows:

GRAPHICS MODES

```
40 PRINT "THIS IS AN EXTRA LINE"
```

When you Run the program now, the first line of print scrolls off, leaving only the output from line 40 visible.

At this point, you might be tempted to ask what use the text window is if it can only display the last line to be printed.

We can make better use of it by causing the program to wait after it has written something to the text window. In this way, the text already there won't scroll off until we want it to do so.

For example, type in and Run Program II. You should now have three lines of text in the window and

```
10 DIM A$(1)
20 GRAPHICS 1
30 PRINT "MODE 1 TEXT WINDOW"
40 PRINT "THIS IS AN EXTRA LINE"
50 PRINT "PRESS RETURN"
60 INPUT A$
```

Program II

the cursor should be on the fourth line, just after a question mark. The reason the text stays there is because the program is waiting for something before it continues.

Line 60 tells the machine to expect an input from the keyboard and it will wait until it gets the signal that an input has been made - the signal that is sent when Return is pressed. If you now press Return, all the text scrolls out of the window to make way for the Ready message and the cursor.

I wouldn't want to claim Program II as a major contribution to programming, but it should illustrate one possible use of the text window. You can print simple messages and take inputs while all the main action is happening on the rest of the screen.

Oh yes, I'd forgotten about the top 20 rows of the screen which are what Mode 1 is all about! Getting something on there is almost as easy as getting it into the text window.

Program III shows how you can write some text to the main part of the screen. Instead of a simple PRINT, you must use PRINT#6. The #6 (read as hash-six) means "print to channel six".

If you Run this program you'll see our name in lights. Note the orange colour of the letters.

```
10 GRAPHICS 1
20 POSITION 5,5
30 PRINT #6;"ATARI USER"
```

Program III

GRAPHICS MODES

Now, type in line 30 again, only this time put our name in lower case as follows:

30 PRINT #6;"atari user"

When you Run this version, the name still appears in upper case (capital) letters, but the colour has changed to green.

Try typing our name in inverse upper case or inverse lower case and see what colours are produced.

This illustrates two aspect of Mode 1. Firstly, it's a five-colour mode, and secondly only half of the character set is available.

We'll take a closer look at the character set in the next issue, but for now let's look at controlling the colours.

'Mode 1: A five colour mode with just half the character set available'

Our use of PRINT #6 followed by upper case letters meant that register 0 was selected and this register defaults to orange.

By using lower case letters, register 1 was selected and this defaults to green.

If you typed the message in upper case inverse, this selects register 2 and so it should have appeared in blue.

Finally, using lower case inverse selects register 3 and so the message should have appeared in red.

You'll notice that no matter how you typed it in, our name still gets printed in upper case letters. That's what I meant by saying that only half of the character set is available.

We can access the other half and get lower case letters, but we'll leave that for next issue.

Register 4 controls the background colour and this defaults to black. You can change this using the SETCOLOR command as we did in Mode 0.

If you are in Mode 1, enter SETCOLOR 4,2,6 and press Return to see the effect of changing the background colour. The 4 selects register 4, the 2 selects colour 2 (orange) and the 6 sets the luminance.

The SETCOLOR command is the key to changing the other registers from their default values. You can see the

GRAPHICS MODES

effect of this by entering Listing IV and running it.

Because the message in line 60 is in

```
10 GRAPHICS 1
20 FOR A=1 TO 5
30 FOR B=0 TO 15
40 POSITION 5,5
50 SETCOLOR 0,B,6
60 PRINT #6;"ATARI USER"
70 FOR DELAY=1 TO 100:NEXT DELAY
80 NEXT B
90 NEXT A
```

Program IV

upper case, we can change its colour by altering register 0. Line 50 performs the alteration and the loop set up by line 30 cycles through all the available colours.

The luminance is kept at a constant value of 6, and line 70 simply keeps each colour on the screen long enough for it to be seen.

There is another way to get a character on the Mode 1 screen, by using COLOR and PLOT commands.

For example, with a clear Mode 1 screen (which you can get by pressing Reset and typing GR.1), type COLOR 65:PLOT 5,5 and press Return. The slightly confusing aspect of this method is that the COLOR command doesn't select a colour in Mode 1. Instead, it selects the character to be PLOTed, in this case character 65 or A as we know it.

Hence, if you entered the line as suggested, you should have an orange A at screen position 5,5.

You can still use SETCOLOR to change the colour of the character that you plot. If you change line 60 in ProgramIV to read:

```
60 COLOR 65:PLOT 5,5
```

and delete line 40 (because it isn't needed), you can see the A cycle through the same colours as our name did.

If you want to put a large piece of text on the Mode 1 screen, it's obviously easier to use PRINT #6 than lots of COLOR/PLOT combinations.

However, there are times when you might need to use COLOR, so Program V shows how to produce a familiar result with line 30 READing the character DATA from line 70.

Selecting a colour register is not quite as straightforward with COLOR/PLOT as it is with PRINT #6. I'll cover it in detail in next issue because the difficulties are connected with the missing half of the character set.

GRAPHICS MODES

```
10 GRAPHICS
20 FOR A=1 TO 10
30 READ X
40 COLOR X
50 PLOT A+5,5
60 NEXT A
70 DATA 65,84,65,82,73,32,85,83,
      69,82
```

Program V

However, to see the sort of thing you can do, change line 40 of Program V to read:

40 COLOR X+32

The message should now appear in green. Instead of adding 32 to all the numbers in the DATA statement, you can use the method selectively to produce a mixed colour display.

For example, retype line 40 as it originally appeared in Program V and alter line 70 to read:

70 DATA 97,84,97,82,105,0,85,11 5,69,114

If you compare the two versions of line 70, you'll notice that I've added 32 to some of the values, causing those characters to be printed in green while the remainder are still printed in orange.

So far, I've not discussed Mode 2 at all. That's because everything I've said about Mode 1 applies to Mode 2.

If you change the GRAPHICS 1 in all the programs to read GRAPHICS 2, you'll get very nearly the same results. The only differences will be the size of the text and its position on the screen.

As I said earlier, Mode 2 rows are like two Mode 1 rows stuck together, so Mode 2 characters are twice as high as those of Mode 1.

This in turn means that the command POSITION 5,5 will refer to a different point on the TV screen depending on whether Mode 1 or Mode 2 is in use.

Mode 1 and 2 also differ in terms of the amount of memory they require - Mode 2 needs less than Mode 1 - but that needn't bother us here.

The programs we'll use won't be long enough to worry about memory considerations!

Next time we'll look at how access the missing half of the character set and how to select a colour register with COLOR/PLOT.

ATARI 8-BIT COMPUTERS!

From: Michael Current <mcurrent@carleton.edu>

Newsgroups: comp.sys.atari.8bit,comp.answers,news.answers

Subject: Atari 8-Bit Computers:

Frequently Asked Questions

Supersedes: <atari-8-bit/faq_857894285@rtfm.mit.edu>

Followup-To: comp.sys.atari.8bit

Date: 8 May 1997 09:02:44 GMT

Organization: Carleton College

Approved: news-answers-request@MIT.Edu

Expires: 21 Jul 1997 09:01:48 GMT

Message-ID: <atari-8-bit/faq_863082108@rtfm.mit.edu>

NNTP-Posting-Host: penguin-lust.mit.edu

Summary: This posting contains a list of Frequently Asked Questions (and their answers) about Atari 8-bit computers.

X-Last-Updated: 1997/05/02

Originator: faqserv@penguin-lust.MIT.EDU

Lines: 2838 Xref: demon comp.sys.atari.8bit:20733 comp.answers:14965 news.answers:61520

Archive-name: atari-8-bit/faq

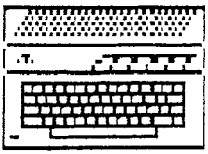
Posting-Frequency: 60 days

Last-modified: April 30, 1997

Welcome to the comp.sys.atari.8bit newsgroup!

Atari 8-Bit Computers

Frequently Asked Questions List

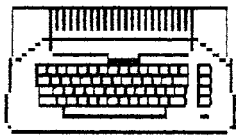


130XE

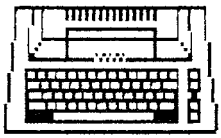
800XL

600XL

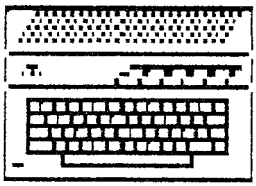
1200XL



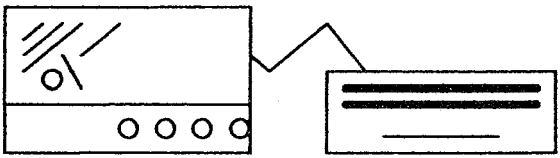
800



400



65XE



XE Game System

Additions/suggestions/comments/corrections are needed!

Please send to: mailto:mcurrent@carleton.edu

This FAQ list is in a constant state of development and comes with no guarantees. If you see any problems, I need to hear from you!

Please refer to the latest version of this FAQ list whenever possible! Available via these locations and Usenet FAQ archives everywhere:

http://www.faqs.org/faqs/atari-8-bit/faq/

ftp://rtfm.mit.edu/pub/usenet-by-group/news.answers/atari-8-bit/

news:comp.sys.atari.8bit news:comp.answers news:news.answers
Or you can always ask me for a copy at mailto:mcurrent@carleton.edu

---

Subject: 0.1) Table of contents:

Reader-suggested topics, unwritten as yet: (volunteers?!) How fast of a modem can my Atari display keep up with? -with Ultraspeed roms? -with XEP-80 -with any particular terminal program.

What is the XEP 80?

Video Upgrades

-Adding chroma\lumi to 800XL.

How can I make my commercial boot disk (tape, cart) into a load file or disk image?

What is APE?

Why does some European software (especially demos) flicker?

Subject: 1.1) What is an Atari 8-bit computer?

400, 800, 1200XL, 600XL, 800XL, 65XE, 130XE, XE Game System.
Here is a very condensed history of Atari, centered around their 8-bit computers.
1972 - Atari Inc. is formed by Nolan Bushnell; created the first coin-operated arcade video game with the introduction of Pong. The video game industry was launched and has shaped pop culture ever since.
1976 - Warner Communications acquires Atari Inc.
1977 - Atari launches the Video Computer System (VCS), giving birth to home video game systems.
1978 - In December, Atari announces the Atari 400 and 800 personal computers, using the 6502 microprocessor. The Atari 800 was code-named

"Colleen"; the 400, "Candy."

1979 - The Atari 400 and Atari 800 Home Computers debut at the Winter Consumer Electronics Show (CES) in early January; they begin shipping in October.

These MOS Technology 6502-based systems run at a clock speed of 1.79 MHz, offering 128 colors displayable simultaneously, up to 320x192 graphics resolution and up to 40x24 text resolution in 8 graphics modes and 6 text modes. Video may be displayed either on a composite video monitor in the case of the 800, or on a standard television for both systems. 4 independent sound voices are available through the audio output of the television or monitor, each with a 3½ octave range, plus there is a built-in speaker for key-click and other programmable sounds. The 800 has a second cartridge port and a full-stroke keyboard, while the 400 has a single cartridge port and a membrane keyboard. Each has 4 serial controller ports and an Atari Serial Input/Output port. Originally, both the 400 and 800 were sold with 8K RAM, but later most 800's were sold with 48K and 400's with 16K. Each includes the 10K Atari Operating System in ROM.

1981 - November: 400/800's begin shipping with the new GTIA chip in place of CTIA, increasing the palette of simultaneously displayable colors to 256 and adding 3 new graphics modes. CTIA is totally phased out by the end of the year.

1982 - The introduction of the 1200XL in 1982 marks the single largest advance in the 8-bit Atari system. The 1200XL runs most software and hardware designed for the 800 and 400, but now runs a slightly more advanced 6502C microprocessor, and includes a full 64K RAM. The single cartridge and monitor ports remain, along with 2 controller ports. In addition, the 1200XL includes 4 programmable Function keys and a Help key, built-in diagnostic and graphics demonstration programs, and probably the favorite keyboard of any 8-bit Atari computer. Clicks previously outputted through the built-in speaker are now heard from the television or monitor's speaker. The revised 16K Operating System offers many new features, including an alternate International Character Set.

1983 - In 1983 Atari replaced the 1200XL/800/400 line-up with the new 800XL and 600XL. These new machines include most of the features of the 1200XL minus the Function keys and the demo program. But now both the 800XL and 600XL have the Atari BASIC language built-in. In addition, these two systems offer the Parallel Bus Interface, providing direct memory access to the heart of the computer. The 800XL contains 64K RAM while the 600XL has 16K RAM.

1984 - Warner Communications

sells Consumer Electronics and Home Computer divisions of Atari Inc. to Tramiel Technologies.
Atari Corporation is formed by Tramiel Technologies and its products marketed under the Atari brand.

(Atari Games Corporation is also formed from the former coin-op division and markets coin-op games under the Atari Games brand.

Today Atari Games is an indirect, wholly-owned subsidiary of the Midway Games division of WMS Industries.
WMS Industries Inc.
3401 N California Ave
Chicago IL 60618-5899 USA
http://www.wms.com/
Midway Games Inc. (a division of WMS) http://www.midway.com/
Atari Games Corporation (a subsidiary of Midway Games) http://www.atarigames.com/)

1985 - The new Atari Corp. delivered on its promise to advance the 8-bit Atari system by replacing the 800XL/600XL with the new 130XE and 65XE in 1985. The 65XE is nearly identical to the 800XL in features, minus the PBI. The 130XE, however, offers 128K RAM, plus the unique (but rarely used) ability for the 6502C and the ANTIC to independently access RAM banks. In addition, the 130XE replaces the PBI port with the Enhanced Cartridge interface, continuing the powerful feature of direct memory access.

1987 - In a change of marketing strategy, Atari introduced the new XE Game System in 1987. Despite its label, the XEGS is a true 8-bit Atari computer system. It offers the convenience of a detachable keyboard and built-in Missile Command game, while offering 64K RAM and full compatibility with the 65XE.

1992 - Atari officially dropped all remaining support of their 8-bit computer line on January 1, 1992.

1996 - On July 31, 1996, Atari Corp. became a JTS Corp. subsidiary. JTS, with headquarters in San Jose, Calif., was founded in 1994 to design, manufacture and supply enhanced-capacity hard disk drives for the notebook and desktop personal computer market. The company currently employs more than 5,800 people worldwide including manufacturing facilities in Madras, India.

JTS Corporation

166 Baypointe Pkwy San

Jose CA 95134-1621

Phone: 408-468-1800

Fax: 408-468-1619

Atari Corporation (a JTS subsidiary)

Subject: 1.2)

What can I do with an 8-bit Atari?

What can you do with an 8-bit Atari computer system? Virtually anything you can do with any other type of computer!

Programming? Pascal, C, BASIC,

Logo, Pilot, Forth, Lisp, 6502 assembler... Plus powerful unique languages like Action! and QUICK...

Word Processing? Try AtariWriter, Letter Perfect, Paperclip, TextPro, Bank Street Writer, 1st EXLent, TurboWord, Cut & Paste, Letter Wizard, Panther, Superscript...

Database? Try TurboBase, TurboFile, Synfile, Data Perfect, MicroFiler, MegaFiler, Homebase, Super Data Base 1-2-3, Small Business System...

Speadsheet? Look at Syncalc, Visicalc, TurboBase, Calc Magic, TurboCalc, SAM Budget...

Communications? There's Express!, BobTerm, Kermit-65, Omnicom, VT850, Chameleon, Ice-T, FlickerTerm 80, Term80...

Graphics? Print Shop, Newsroom, Blazing Paddles, Video Title Shop, Virtuoso, Movie Maker, News Station, Publishing Pro, Awardware, Page Designer, ChromaCAD, Rambrandt...

Music? Virtuoso, Music Studio, Music Construction Set, Songwriter, Electronic Drummer, Music Painter, Music Composer, AtariMusic, MIDI-Track, Digital Music System, Chaos Music Composer...

Alternate Operating Systems? There's the Diamond Graphic Operating System; SpartaDOS X, the 64K DOS on a "supercartridge"; the S.A.M. (Screen Aided Management) 80 column Desktop System; the Ultra Speed Plus OS; the Turbo Boss! High Speed O/S...

Hardware? A plethora of upgrades and add-ons are available, realizing improvements in speed, memory, sound, graphics, storage media...you name it!

Education? Colorful graphics, exciting sounds and full-screen editing give rise to hundreds of quality educational software titles.

Entertainment? The 8-bit Atari has long been famous for thousands of great games.

If you want to do something with a computer, chances are you can do it on an 8-bit Atari computer. While slower than today's PC's, the 8-bit Atari is far less costly than any of these, is easier to program at the machine level or alter at the hardware level, and has been documented more thoroughly than any of the newer computing platforms. The relative simplicity in design of the 8-bit Atari also means that many people find the systems more reliable than their modern counterparts. The software for the 8-bit Atari is sometimes more powerful than on any other platform. For example, some Bulliten Board Systems are still run on 8-bit Ataris specifically because the BBS software available can be better than that for any other type of computer. There is a reason for the relative quality of software on the 8-bit Atari. People program commercially for the latest PC's to make money; people program the 8-bit

Atari because they want to -- they enjoy producing good software. It takes a team of specialized programmers to develop a major piece of software for those other systems; with the 8-bit Atari, a single person has the chance to learn the entire system, thereby developing greater pride in his programming abilities and his final product. The character of the programmer can shine through.

The 8-bit Atari owner can take pride that his/her computer platform was developed years before the IBM PC or Apple Macintosh were even on the drawing boards, but remains as useful today as it was in 1979.

For these reasons and more, the 8-bit Atari remains a popular alternative in today's home computer marketplace.

Subject: 1.3) What are the some of the performance features of the 8-bit Atari?

Some of this text by (Bill Kendrick).

CLOCK SPEED:

NTSC machines: 1.78979 MHz

PAL machines: 1.773447 MHz

SCREEN REFRESH RATE:

60 times per second (Hz) on NTSC Ataris

49.86 Hz on PAL machines

SEE PAGE 33 FOR ANTIC AND THE DISPLAY LIST:

GRAPHICS INDIRECTION (COLOR REGISTERS AND CHARACTER SETS):

Nine color registers are available. Each color register holds any of 16 luminances x 16 hues = 256 colors. (Four registers are for player-missle graphics).

Character sets of 128 8x8 characters, each with a normal and an inverse video incarnation, are totally redefinable.

PLAYER-MISSLE GRAPHICS:

Four 8-bit wide, 128 or 256 byte high single color players, and four 2-bit wide, 128 or 256 byte high single color missiles are available. A mode to combine the 4 missiles into a 5th 8-bit wide player is also available, as is a mode to XOR colors or blacken out colors when players overlap (good for making three colors out of two players!) Players and missiles have adjustable priority and collision detection.

DISPLAY LIST INTERRUPTS (DLI's):

Screen modes can be mixed (by lines) down the screen using the Display List - a program which is executed by the ANTIC graphics chip every screen refresh:

All other screen attributes (color, player/missile horizontal position, screen width, player/missile/playfield priority, etc.) can be ajusted at any point down the screen via DLI's.

SCROLLING: Fine scrolling (both vertical and horizontal) can be enabled on any line on the screen.

SOUND: Four voices of 8-bit pitch-

resolution, 4-bit volume-resolution, 8-distortion sound can be produced. 2 voices (1 and 2, and/or 3 and 4) can be combined to make 16-bit pitch-resolution. Also 4-bit volume-only modes can be enabled for digitally sampled sound replay.
A fifth "voice" is produced by the internal speaker on Atari 400/800's (for keyclick and buzzer) and in the XL's and XE's this was (fortunately!) rerouted through the normal audio output, and the keyclick can be disabled.

Subject: 1.4) What is the internal layout of the 8-bit Atari?

The following text was written by Chris Crawford and appears in De Re Atari (Atari#APX-90008), a book published and copyright by Atari, Inc., 1981-1982. It has been very slightly modified here for generality.
"The internal layout of the Atari 8-bit computer is very different from other systems. It of course has a microprocessor (a 6502), RAM, ROM, and a (PIA). However, it also has three special-purpose (LSI) chips known as ANTIC, GTIA, and POKEY. These chips were designed by Atari engineers primarily to take much of the burden of housekeeping off of the 6502, thereby freeing the 6502 to concentrate on computations. While they were at it, they designed a great deal of power into these chips. Each of these chips is almost as big (in terms of silicon area) as a 6502, so the three of them together provide a tremendous amount of power. Mastering the Atari 8-bit computers is primarily a matter of mastering these three chips.
ANTIC ("Alpha-Numeric Television Interface Circuit") is a microprocessor dedicated to the television display. It is a true microprocessor; it has an instruction set, a program (called the display list), and data. The display list and the display data are written into RAM by the 6502. ANTIC retrieves this information from RAM using direct memory access (DMA). It processes the higher level instructions in the display list and translates these instructions into a real-time stream of simple instructions to GTIA.
CTIA ("Color Television Interface Adapter") / **GTIA** ("George's Television Interface Adapter") is a television interface chip. ANTIC directly controls most of GTIA's operations, but the 6502 can be programmed to intercede and control some or all of GTIA's functions. GTIA converts the digital commands from ANTIC (or the 6502) into the signal that goes to the television. GTIA also adds some factors of its own, such as color values, player-missle graphics, and collision detection.
POKEY is a digital input/output (I/O) chip. It handles such disparate tasks as the serial I/O bus, audio generation, keyboard scan, and random number generation. It also digitizes

the resistive paddle inputs and controls maskable interrupt (IRQ) requests from peripherals.

All four of these LSI chips function simultaneously. Careful separation of their functions in the design phase has minimized conflicts between the chips. The only hardware level conflict between any two chips in the system occurs when ANTIC needs to use the address and data buses to fetch its display information. To do this, it halts the 6502 and takes control of the buses."

The **130XE** and **XEGS** contain a small additional LSI called FREDDIE, a RAM address multiplexer. According to mailto:sup8pdct@closer.brisnet.org.au (James Bradford), "Freddy is a type of memory controller. It takes the address and clock from the CPU and multiplexes it with the appropriate timings and signals to use DYNAMIC memory. Freddy also buffers the system clock crystal and divides it down then feeds that to GTIA. The XEGS has a freddy but it doesn't have the extended RAM. Even if it did, you would still need the chip that does the REAL bank switching. It is a small 16-pin chip (Atari/Best Electronics catalog number CO25953: rev9/page 42). It gets RAS from freddy, the bank select bits from PIA, A14, A15 and the 6502 halt signal to control which bank of 8 chips RAS goes to. A14 and A15 then go to freddy for the address range of the extra memory bank (or normal address range with no bank switching). The ANTIC/6502 select bits in combination with the 6502 halt line, control the switching of the PIA bank number bits to A14/A15 and which bank of memory RAS goes to. Why people say freddy does the bank switching is beyond me. An 800XL can look like a 130XE with that 16-pin chip installed (That's right NO freddy) and an extra 8 RAM chips."

Hardware Arrangement (With thanks to mailto:peter@soemtron.sb.sub.de (Peter))

NOTES

* RAM: 400/800:8/16/48K, 1200XL/800XL/65XE/XEGS:64K, 600XL:16K, 130XE:128K
* ROM: 400/800:10K OS, 1200XL: 16K OS, all others:16K OS + 8K Atari BASIC
* CPU: 400/800:6502, all others: 6502C
* 800 includes two Cartridge Slots, all others include one
* early release 400/800 have CTIA instead of GTIA
* 400/800 have 4 Controller Ports, all others have 2
* PBI is on 600XL/800XL only
* ECI is on 130XE/800XE only.

ANTIC AND THE DISPLAY LIST

ANTIC Mode #	CIO/BASIC Graphics #	Display Type	Resolution (full screen)	Number of Colors
2	0	Char	40 x 24	1 *
3	-	Char	40 x 19	1 *
4	12 ++	Char	40 x 24	5
5	13 ++	Char	40 x 12	5
6	1	Char	20 x 24	5
7	2	Char	20 x 12	5
8	3	Map	40 x 24	4
9	4	Map	80 x 48	2
A	5	Map	80 x 48	4
B	6	Map	160 x 96	2
C	14 ++	Map	160 x 192	2
D	7	Map	160 x 96	4
E	15 ++	Map	160 x 192	4
F	8	Map	320 x 192	1 *
F	9 +	Map	80 x 192	1 **
F	10 +	Map	80 x 192	9
F	11 +	Map	80 x 192	16 ***

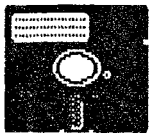
* 1 Hue; 2 Luminances

** 1 Hue; 16 Luminances

*** 16 Hues; 1 Luminance

\+ require the GTIA chip. 1979 - 1981 400/800's shipped with CTIA

++ Not available via the BASIC GRAPHICS command in 400/800's.



DISK CONTENT

With me being so very late setting up this issue, I though if I miss checking the disk content, John will get the master copy a little sooner to do the finishing touches to the newsletter. I sincerely hope you won't mind just this once.

There are various files and programs on this issue disk.

ADVERTISING USER GROUPS

LACE

The LONDON ATARI COMPUTER ENTHUSIASTS

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

The membership fee is £8.00 annually for more information contact:

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

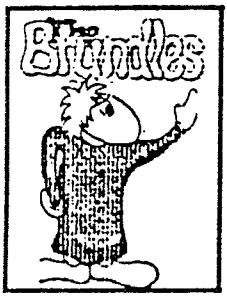
O.H.A.U.G.

The OL'HACKERS ATARI USER GROUP INC.

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

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

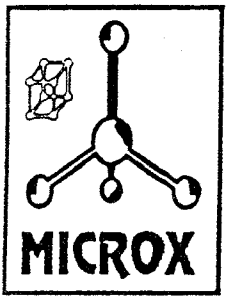
Contact:
Mr.Ron Fetzer
O.H.A.U.G.
Secretary & Treasurer
22 Monaco Avenue
Elmont, N.Y. 11003
USA



Are You Still Complaining about lack of New software for your Atari 8 bit ?
If you are you don't know about Mail Order From

MICRO-DISCOUNT

265, CHESTER ROAD.
STREETLY.
WEST MIDLANDS.
B74 3EA.
ENGLAND.



TEL 021 353 5730
FAX 021 352 1669

ADAX *HANS KLOSS*

DARKNESS HOUR