

The Little Black Bit Book.

Version 0.60.
Revision 7/11/06 21:43.

By Mr. Karl N. H. Meyland.
(ASS. DIP. COMP. SCI.)

Introduction

The Atari computer range has played a part, as I type this, in my life for about 8 years. Over the period of those 8 years, I have seen the Atari computers rise to a degree of recognition for their price, performance and usability. Nowadays this is changing with the price war between PC clones and Macintoshes being introduced, the advent of Windoze for computer illiterates, and a much wider market for the Atari range to compete in.

One thing that has never changed over those 8 years is the sad lack of available technical information on the computers. And further still, the lack of information that actually would be divulged to the Atari Public. Finally, if it was there, it would cost the user plenty!

With the introduction of the Atari Compendium we finally got some resources that the public can get hold of, but alas, that still came at a price. Here in Australia a copy of the Compendium would have cost me close to AU\$140! That meant that a copy of one book would have cost me almost as much as what a new 1 MEG STe would!

One problem I saw with buying a copy of the compendium was that it included very little on the hardware involved in the computers. Sure it showed you how to program the chips, but for the purposes of making upgrades such as memory expansions, it was pretty useless. However, in the last 3 years (December 1994 as I type) I have met some wonderful and innovative users who have provided me with a range of information on Atari computers. And so the photocopies have grown. Unfortunately, the copies deteriorate, and cannot be sent electronically, so it became apparent that I had to 'electrify' the documents I had for safekeeping.

Disclaimer

The hacks described in this 'book' can void your warranty, so you are on your own if you try to do them. I disclaim any warranties implied, whatsoever. And if you get knocked on your butt from the voltages involved in the computer, or lose your eyebrows, whatever, from playing about with these hacks, it's your problem. I warned you, so there.

Copyrights

Hmm. Hard to do this one. I retain **all** copyrights for this document, except for the copyrights of individual authors who have written their documents. If you want to keep a copy of this 'book' you must send me a donation, otherwise, delete it. If you make use of an article by another author included in this 'book' you have to send them their requested donation. As some of the articles are pretty old, it's up to you to determine if the authors' addresses are still current, but that's on your head. And you can't give this away on coverdisks, in magazines, disk mags, whatever, ***without my express written permission***. You must obtain it first, O.K.?

There, that about covers it. If you don't adhere to the above terms, look out.

Shareware

This stuff would never be covered in the Compendium, so I thought that I should release it. And so I am. As of release 1.0, this 'book' is Shareware. If you decide to print out a copy and keep it, I ask that you send me a donation of some sort, cash donations of \$5 or \$10 Australian would be nice, also if you have copies of documents that I do not have, I would be pleased to include them in a future revision.

As stated in the above paragraph about copyrights, individual authors retain their copyrights, so if you use one of their articles to perform an upgrade, send them their requested Shareware fee. And if you don't send their fee, well, welcome to the dark world of piracy you naughty people!

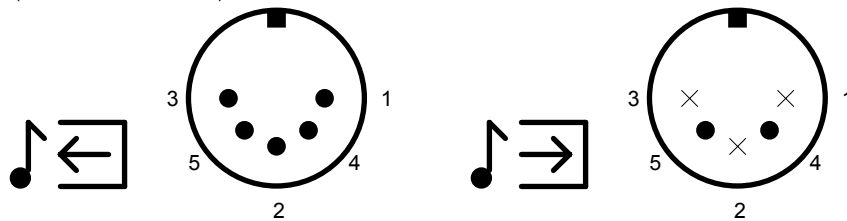
Donations, comments, articles and error corrections may be sent to:

Karl Meyland
Grange Road
POREPUNKAH
Victoria 3740
AUSTRALIA.

*A small piece of history - the first box of blank floppy disks I ever had the misfortune to use cost AUS\$99!
(3.5", DD Verbatim DataLife disks!)*

Atari ST/STe Series.

Atari ST/STe Midi Ports.
(DIN 5 Female)



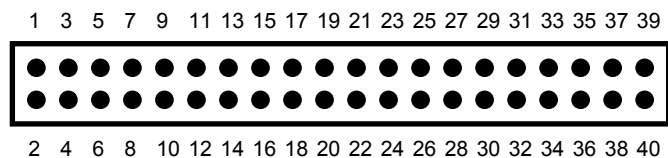
MIDI OUT

- 1 - THRU Transmit Data
- 2 - Shield Ground
- 3 - THRU Loop Return
- 4 - OUT Transmit Data
- 5 - OUT Loop Return

MIDI IN

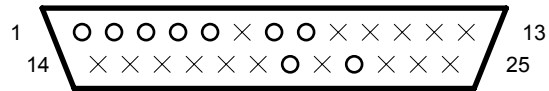
- 1 - Not Connected
- 2 - Not Connected
- 3 - Not Connected
- 4 - IN Receive Data
- 5 - IN Loop Return

Cartridge 1040 ST/STe



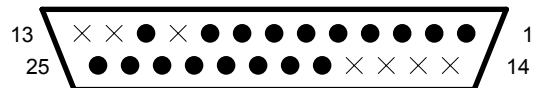
- | | |
|-----------------|------------------------|
| 1 - +5V DC | 20 - Address 15 |
| 2 - +5V DC | 21 - Address 8 |
| 3 - Data 14 | 22 - Address 14 |
| 4 - Data 15 | 23 - Address 7 |
| 5 - Data 12 | 24 - Address 9 |
| 6 - Data 13 | 25 - Address 6 |
| 7 - Data 10 | 26 - Address 10 |
| 8 - Data 11 | 27 - Address 5 |
| 9 - Data 8 | 28 - Address 12 |
| 10 - Data 9 | 29 - Address 11 |
| 11 - Data 6 | 30 - Address 4 |
| 12 - Data 7 | 31 - ROM Select 3 |
| 13 - Data 4 | 32 - Address 3 |
| 14 - Data 5 | 33 - ROM Select 4 |
| 15 - Data 2 | 34 - Address 2 |
| 16 - Data 3 | 35 - Upper Data Strobe |
| 17 - Data 0 | 36 - Address 1 |
| 18 - Data 1 | 37 - Lower Data Strobe |
| 19 - Address 13 | 38 - Ground |
| | 39 - Ground |
| | 40 - Ground |

Modem
1040 ST/STe



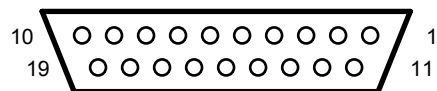
- | | |
|-------------------------|--------------------------|
| 1 - Protective Ground | 14 - Not Connected |
| 2 - Transmitted Data | 15 - Not Connected |
| 3 - Received Data | 16 - Not Connected |
| 4 - Request To Send | 17 - Not Connected |
| 5 - Clear To Send | 18 - Not Connected |
| 6 - Not Connected | 19 - Not Connected |
| 7 - Signal Ground | 20 - Data Terminal Ready |
| 8 - Data Carrier Detect | 21 - Not Connected |
| 9 - Not Connected | 22 - Ring Indicator |
| 10 - Not Connected | 23 - Not Connected |
| 11 - Not Connected | 24 - Not Connected |
| 12 - Not Connected | 25 - Not Connected |
| 13 - Not Connected | |

Printer
1040 ST/STe



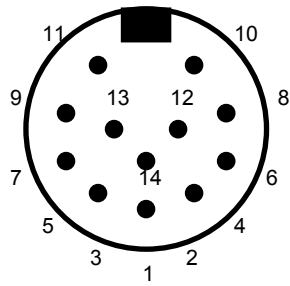
- | | |
|--------------------|--------------------|
| 1 - STROBE | 14 - Not Connected |
| 2 - Data 0 | 15 - Not Connected |
| 3 - Data 1 | 16 - Not Connected |
| 4 - Data 2 | 17 - Not Connected |
| 5 - Data 3 | 18 - GND |
| 6 - Data 4 | 19 - GND |
| 7 - Data 5 | 20 - GND |
| 8 - Data 6 | 21 - GND |
| 9 - Data 7 | 22 - GND |
| 10 - Not Connected | 23 - GND |
| 11 - BUSY | 24 - GND |
| 12 - Not Connected | 25 - GND |
| 13 - Not Connected | |

Hard Disk
Atari ST/STe



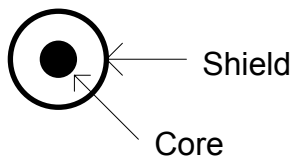
- | | |
|------------------------|-------------------|
| 1 - Data 0 | 11 - Ground |
| 2 - Data 1 | 12 - Reset |
| 3 - Data 2 | 13 - Ground |
| 4 - Data 3 | 14 - Acknowledge |
| 5 - Data 4 | 15 - Ground |
| 6 - Data 5 | 16 - A1 |
| 7 - Data 6 | 17 - Ground |
| 8 - Data 7 | 18 - Read/Write |
| 9 - Chip Select | 19 - Data Request |
| 10 - Interrupt Request | |

ST/STe Floppy Disk
(14 Pin DIN Female)



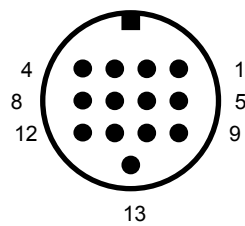
- | | |
|--------------------|----------------------|
| 1 - Read Data | 8 - Motor On |
| 2 - Side 0 Select | 9 - Direction In |
| 3 - Logic Ground | 10 - Step |
| 4 - Index Pulse | 11 - Write Data |
| 5 - Drive 0 Select | 12 - Write Gate |
| 6 - Drive 1 Select | 13 - Track 00 Detect |
| 7 - Logic Ground | 14 - Write Protect |

ATARI ST/STe Television (where applicable)
(RF Socket)



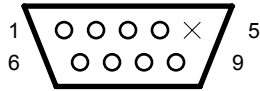
- Core - RF Modulated Video
Shield - Ground

ATARI ST/STe Video
(13 Pin DIN Female)



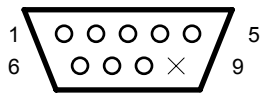
- | | |
|----------------------|---------------------|
| 1 - Audio Out | 8 - +12V Pullup |
| 2 - Composite Video | 9 - Horizontal Sync |
| 3 - Gen. Pur. Output | 10 - Blue |
| 4 - Monochrome Det. | 11 - Monochrome |
| 5 - Audio In | 12 - Vertical Sync |
| 6 - Green | 13 - Ground |
| 7 - Red | |

ATARI ST/STe Mouse/Joystick
(9 Pin DIN Female)



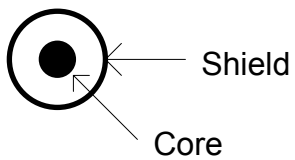
- 1 - Up/XB
- 2 - Down/XA
- 3 - Left/YA
- 4 - Right/YB
- 5 - Not Connected
- 6 - Fire/Left Button
- 7 - +5V DC
- 8 - Ground
- 9 - Joy 1 Fire/Right Button

ATARI ST/STe Joystick
(9 Pin DIN Female)



- 1 - Up
- 2 - Down
- 3 - Left
- 4 - Right
- 5 - Reserved
- 6 - Fire Button
- 7 - +5V DC
- 8 - Ground
- 9 - Not Connected

ATARI STe Audio Output
(Dual RF Sockets)



- Core - Audio
- Shield - Ground

APPENDIX C 1040ST SPECIFICATIONS

Computer

Processor	MC68000, 16-bit external architecture; 8 MHz clock frequency
Memory:	
4160ST	4,194,304 bytes of RAM; 196,608 bytes of ROM
2080ST	2,097,152 bytes of RAM; 196,608 bytes of ROM
1040ST	1,048,576 bytes of RAM; 196,608 bytes of ROM
520ST	524,288 bytes of RAM; 196,608 bytes of ROM
Graphics Resolution (selectable)	640 x 400 monochrome 640 x 200 x 4 colors 320 x 200 x 16 colors
Color	Palette of 512 colors
Interfaces	Midi In and Midi Out Ports Monitor Port (RGB Analog; high-resolution monochrome, composite video and audio) Parallel Interface Printer Port RS232 Serial Modem Port Floppy Disk Port (includes controller) Hard Disk Port (10 megabits per second DMA transfer rate) ROM Cartridge Port (128 kilobytes capacity) Mouse/Joystick and Joystick Ports TV Port (not included on some models)
Sound Generator	3 voices from 30 Hz to above audible range
Keyboard	94-key intelligent keyboard using 6301 microprocessor
Power Supply (built-in)	+5V @ 3A +12V @ 1A -12V @ 30mA
Power Consumption	95 watts (maximum)
Ambient Temperature	41° to 113°F (5° to 45°C), operating or idle -4° to 149°F (-20° to 65°C), storage -40° to 149°F (-40° to 65°C), transport
Relative Humidity (non-condensing)	20% to 80%, operating or idle up to 95%, storage or transport

Disk Drive

Track Density 135 tracks per inch

Storage Capacity 360 kilobytes per side (formatted MFM); 720 kilobytes total

Storage Medium

4160ST, 2080ST, 3¹/₂-inch microfloppy disk; double-sided, double density; 135 tracks per inch

1040ST

520ST 3¹/₂-inch microfloppy disk; single-sided, double density; 135 tracks per inch

Head Positioning Stepper Motor
Mechanism

Data Transfer Speed 250 kilobits per second

Physical Characteristics

4160ST, 2080ST, Maximum height 2³/₄"

1040ST Width 18³/₄"

Depth 11¹/₂"

Weight 9lb. 7oz.

Internal power supply

520ST Maximum height 2¹/₂"

Width 18³/₄"

Depth 9¹/₂"

Weight 4lb. 6oz.

Internal power supply

ATARI MEGA ST^E

The Atari Mega ST^E stands as a powerful force to be reckoned with in the world of serious business computers. With an increased processing speed of 16mhz as well as an on-board cache, 4096 colour palette, stereo digital sound and graphics co-processor, the Mega ST^E is unbeatable by any computer in its class.

The Mega ST^E retains all the advances that made the original Atari ST^{FM} so popular, such as a built-in MIDI interface, an RF modulator and GENLOCK support for receiving external video sources, plus parallel and serial ports for standard communication and printing requirements.

Equally important, the Mega ST^E offers a graphic user interface which is widely acknowledged as the most natural and intuitive way for humans to interact with computers. It also boasts a broad range of software spanning home, classroom and office applications. When it's time to consider which personal computer offers the most personal advantages, the Atari Mega ST^E leads the way.

Technical Specifications

Architecture

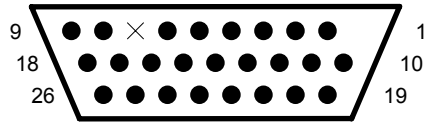
- CPU: Motorola 68000 running at 16mhz
- Bus: 16-bit external; 32-bit internal; 24-bit address
- FPU: Optional MC68881/82
- RAM: Optional 2 - 4 MB
- ROM: 256 Kbyte internal; 128 Kbyte external plug-in ROM

Input/Output Ports

- Internal A24/D16 VME card slot
- MIDI IN/MIDI OUT
- Audio out: 2 x RCA jacks for left and right channels
- Connector for colour or monochrome monitor
- 2 x modem/RS232C serial ports
- 1 x high-speed DMA serial port or LAN

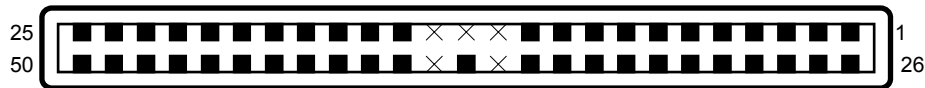
Atari Falcon 030 Series.

Falcon 030 DSP CONNECTOR
(DB26 Female)



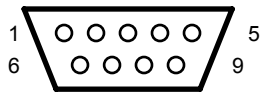
1 - GP0	14 - GND
2 - GP1	15 - SRD
3 - GP2	16 - GND
4 - P_DATA	17 - +12V
5 - P_CLK	18 - GND
6 - P_SYNC	19 - R_DATA
7 - Not Connected	20 - R_CLK
8 - GND	21 - R_SYNC
9 - +12V	22 - EXT_INT
10 - GND	23 - STD_INT
11 - SC0	24 - SCK
12 - SC1	25 - GND
13 - SC2	26 - EXCLK

Falcon 030 SCSI(II) CONNECTOR
(SCSI II Female)



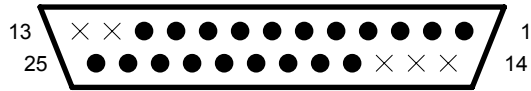
1 - GND	26 - SCSI 0
2 - GND	27 - SCSI 1
3 - GND	28 - SCSI 2
4 - GND	29 - SCSI 3
5 - GND	30 - SCSI 4
6 - GND	31 - SCSI 5
7 - GND	32 - SCSI 6
8 - GND	33 - SCSI 7
9 - GND	34 - Parity
10 - GND	35 - GND
11 - +5V	36 - GND
12 - Not Connected	37 - Not Connected
13 - Not Connected	38 - +5V
14 - Not Connected	39 - Not Connected
15 - GND	40 - GND
16 - GND	41 - ATN
17 - GND	42 - GND
18 - GND	43 - BSY
19 - GND	44 - ACK
20 - GND	45 - RST
21 - GND	46 - MSG
22 - GND	47 - SEL
23 - GND	48 - C/D
24 - GND	49 - REQ
25 - GND	50 - I/O

Falcon 030 SERIAL PORT CONNECTOR
(DB9 MALE)



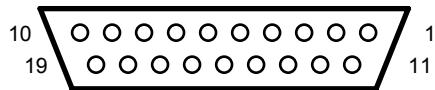
- 1 - Carrier Detect
- 2 - Receive
- 3 - Transmit
- 4 - Data Terminal Ready
- 5 - GND
- 6 - Data Set Ready
- 7 - Request To Send
- 8 - Clear To Send
- 9 - Ring Indicator

Falcon 030 Parallel Port Connector
(DB25 FEMALE)



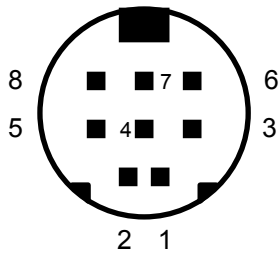
- | | |
|--------------------|--------------------|
| 1 - Strobe | 14 - Not Connected |
| 2 - Data 0 | 15 - Not Connected |
| 3 - Data 1 | 16 - Not Connected |
| 4 - Data 2 | 17 - Select |
| 5 - Data 3 | 18 - GND |
| 6 - Data 4 | 19 - GND |
| 7 - Data 5 | 20 - GND |
| 8 - Data 6 | 21 - GND |
| 9 - Data 7 | 22 - GND |
| 10 - Acknowledge | 23 - GND |
| 11 - Busy | 24 - GND |
| 12 - Not Connected | 25 - GND |
| 13 - Not Connected | |

Atari Falcon 030 Video
(DB19 Male)



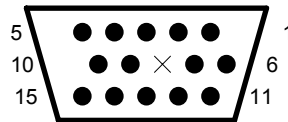
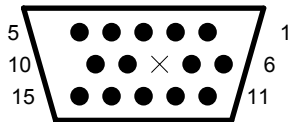
- | | |
|--------------------|-----------------------------|
| 1 - Red | 11 - Ground |
| 2 - Green | 12 - Composite Sync / Video |
| 3 - Blue | 13 - Horizontal Sync |
| 4 - Mono / Overlay | 14 - Vertical Sync |
| 5 - Ground | 15 - External Clock Input |
| 6 - Red Ground | 16 - External Sync Enable |
| 7 - Green Ground | 17 - +12V |
| 8 - Blue Ground | 18 - M1 |
| 9 - Audio Out | 19 - M0 |
| 10 - Ground | |

Falcon 030 SCC Connector
8 Pin Mini DIN Female RS-422



- 1 - Handshake Output (DTR RS 423)
- 2 - Handshake Input or External Clock
- 3 - - Transmit Data
- 4 - Ground
- 5 - - Received Data
- 6 - + Transmitted Data
- 7 - General Purpose Input
- 8 - + Receive

Falcon 030 Enhanced Joystick
(DB15 HD Male)



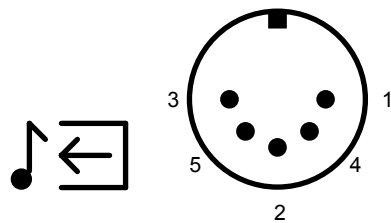
PORT A

- | | |
|----------------------|-------------|
| 1 - UP 0 | 9 - Ground |
| 2 - DOWN 0 | 10 - Fire 2 |
| 3 - LT 0 | 11 - Up 2 |
| 4 - RT 0 | 12 - Down 2 |
| 5 - PAD0Y | 13 - LT 2 |
| 6 - FIRE 0/LIGHT GUN | 14 - RT 2 |
| 7 - VCC (+5 VDC) | 15 - PAD0X |
| 8 - NOT CONNECTED | |

PORT B

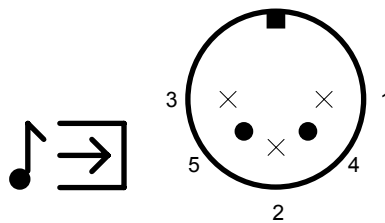
- | | |
|-------------------|-------------|
| 1 - Up 1 | 9 - Ground |
| 2 - Down 1 | 10 - Fire 3 |
| 3 - LT 1 | 11 - Up 3 |
| 4 - RT 1 | 12 - Down 3 |
| 5 - RT 1 | 12 - Down 3 |
| 6 - RT 1 | 12 - Down 3 |
| 7 - RT 1 | 12 - Down 3 |
| 8 - Not Connected | |

Falcon 030 MIDI Port
(DIN 5 Female)



MIDI OUT

- 1 - Thru Transmit
- 2 - Ground
- 3 - Thru Loop Return
- 4 - Out Transmit
- 5 - Out Loop Return

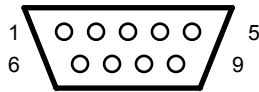


MIDI IN

- 1 - Not Connected
- 2 - Not Connected
- 3 - Not Connected
- 4 - In Receive
- 5 - In Loop Return

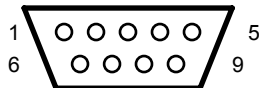
Atari TT030 Series.

TT-030 Modem 1
(DB9 Male)



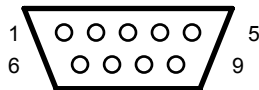
1 - Carrier Detect	Input
2 - Received Data	Input
3 - Transmitted Data	Output
4 - Data Terminal Ready	Output
5 - Ground	---
6 - Not connected	---
7 - Request To Send	Output
8 - Clear To Send	Input
9 - Ring Indicator	Input

TT-030 Modem 2
(DB9 Male)



1 - Carrier Detect	Input
2 - Received Data	Input
3 - Transmitted Data	Output
4 - Data Terminal Ready	Output
5 - Ground	---
6 - Data Set Ready	Input
7 - Request To Send	Output
8 - Clear To Send	Input
9 - Ring Indicator	Input

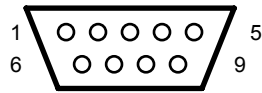
TT-030 Serial 1
(DB9 Male)



1 - Not Connected	---
2 - Received Data	Input
3 - Transmitted Data	Output
4 - Data Terminal Ready (%)	Output
5 - Ground	---
6 - Not connected	---
7 - Request To Send (%)	Output
8 - Not Connected	---
9 - Not Connected	---

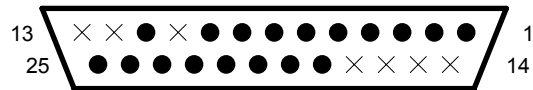
(%) Data Terminal Ready and Request to Send on this port are asserted whenever the TT power is on.

TT-030 Serial 2
(DB9 Male)



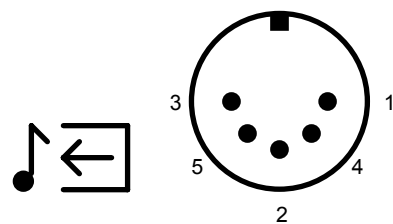
- | | |
|-------------------------|--------|
| 1 - Carrier Detect | Input |
| 2 - Received Data | Input |
| 3 - Transmitted Data | Output |
| 4 - Data Terminal Ready | Output |
| 5 - Ground | --- |
| 6 - Data Set Ready | Input |
| 7 - Request To Send | Output |
| 8 - Clear To Send | Input |
| 9 - Synchronous Clock | Input |
| -- - Synchronous Clock | Output |

TT-030 Parallel Printer
(DB25 FEMALE)

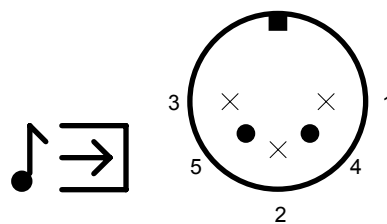


- | | |
|-------------------------|--------|
| 1 - Strobe | Output |
| 2 - Data 0 | --- |
| 3 - Data 1 | --- |
| 4 - Data 2 | --- |
| 5 - Data 3 | --- |
| 6 - Data 4 | --- |
| 7 - Data 5 | --- |
| 8 - Data 6 | --- |
| 9 - Data 7 | --- |
| 10 - Not Connected | --- |
| 11 - Busy | Input |
| 12 - 17 - Not Connected | --- |
| 18 - 25 - GND | --- |

TT-030 MIDI Port
(DIN 5 Female)

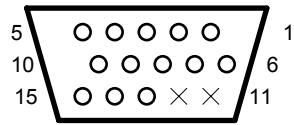


- MIDI OUT
- | |
|------------------------|
| 1 - THRU Transmit Data |
| 2 - Shield Ground |
| 3 - THRU Loop Return |
| 4 - OUT Transmit Data |
| 5 - OUT Loop Return |



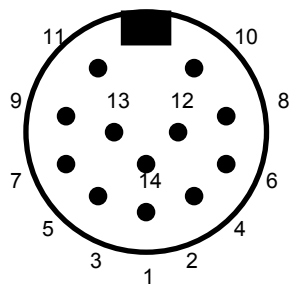
- MIDI IN
- | |
|---------------------|
| 1 - Not Connected |
| 2 - Not Connected |
| 3 - Not Connected |
| 4 - IN Receive Data |
| 5 - IN Loop Return |

TT-030 Monitor
(DB15 HD Male)



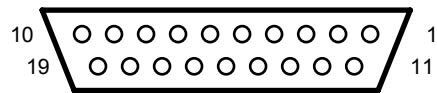
- 1 - Analog Red
- 2 - Analog Green
- 3 - Analog Blue
- 4 - High Resolution Monochrome (ECL)
- 5 - Ground
- 6 - Red Return
- 7 - Green Return
- 8 - Blue Return
- 9 - Monochrome Monitor Detect
- 10 - Ground
- 11 - Not Connected
- 12 - Not Connected
- 13 - Horizontal Sync (TTL)
- 14 - Vertical Sync (TTL)
- 15 - High Resolution Monochrome (ECL)

TT-030 Floppy Disk
(14 Pin DIN Female)



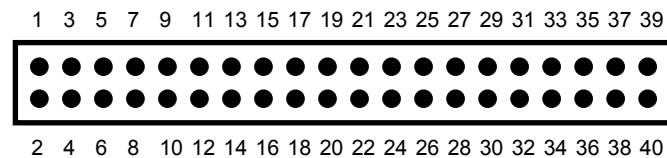
- | | |
|---------------------------|----------------------|
| 1 - Read Data | 8 - Motor On |
| 2 - Side 0 Select | 9 - Direction In |
| 3 - Ground | 10 - Step |
| 4 - Index Pulse | 11 - Write Data |
| 5 - External Drive Select | 12 - Write Gate |
| 6 - Internal Pull-Up | 13 - Track 00 Detect |
| 7 - Ground | 14 - Write Protect |

TT-030 ACSI DMA
 (ACSI Interface, DB19 Female)



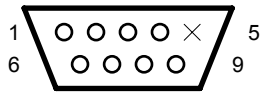
- | | |
|------------------------|-------------------|
| 1 - Data 0 | 11 - Ground |
| 2 - Data 1 | 12 - Reset |
| 3 - Data 2 | 13 - Ground |
| 4 - Data 3 | 14 - Acknowledge |
| 5 - Data 4 | 15 - Ground |
| 6 - Data 5 | 16 - A1 |
| 7 - Data 6 | 17 - Ground |
| 8 - Data 7 | 18 - Read / Write |
| 9 - Chip Select | 19 - Data Request |
| 10 - Interrupt Request | |

TT-030 Cartridge
 (40 Pin PCB Edge Connector, Female)



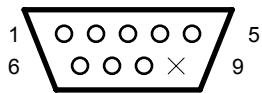
- | | |
|-----------------|-------------------------|
| 1 - +5V DC | 21 - Address 8 |
| 2 - +5V DC | 22 - Address 14 |
| 3 - Data 14 | 23 - Address 7 |
| 4 - Data 15 | 24 - Address 9 |
| 5 - Data 12 | 25 - Address 6 |
| 6 - Data 13 | 26 - Address 10 |
| 7 - Data 10 | 27 - Address 5 |
| 8 - Data 11 | 28 - Address 12 |
| 9 - Data 8 | 29 - Address 11 |
| 10 - Data 9 | 30 - Address 4 |
| 11 - Data 6 | 31 - *ROM Select 3 |
| 12 - Data 7 | 32 - Address 3 |
| 13 - Data 4 | 33 - *ROM Select 4 |
| 14 - Data 5 | 34 - Address 2 |
| 15 - Data 2 | 35 - *Upper Data Strobe |
| 16 - Data 3 | 36 - Address 1 |
| 17 - Data 0 | 37 - *Lower Data Strobe |
| 18 - Data 1 | 38 - Ground |
| 19 - Address 13 | 39 - Ground |
| 20 - Address 15 | 40 - Ground |

MOUSE/JOYSTICK 0
(DB9 Male)



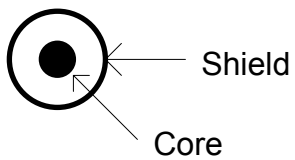
- 1 - XB/Up
- 2 - XA/Down
- 3 - YA/Left
- 4 - YB/Right
- 5 - Middle Button/Joystick Up
- 6 - Left Button/Fire
- 7 - +5V DC
- 8 - Ground
- 9 - Right Button/Joystick 1 Fire

JOYSTICK 1
(DB9 Male)



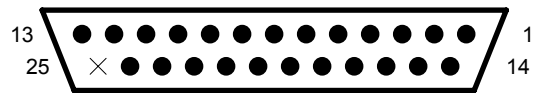
- 1 - Up
- 2 - Down
- 3 - Left
- 4 - Right
- 5 - Reserved
- 6 - Fire Button
- 7 - +5V DC
- 8 - Ground
- 9 - Not Connected

TT030 Audio Output
(Dual RF Sockets)



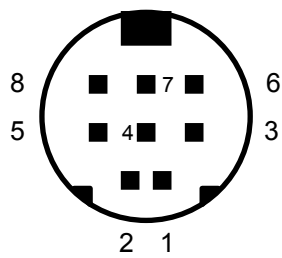
- Core - Audio
- Shield - Ground

TT-030 SCSI
(DB25 FEMALE)



- | | |
|-------------|--------------------|
| 1 - *REQ | 14 - Ground |
| 2 - *MSG | 15 - *C/D |
| 3 - *IO | 16 - Ground |
| 4 - *RST | 17 - *ATN |
| 5 - *ACK | 18 - Ground |
| 6 - *BSY | 19 - *SEL |
| 7 - Ground | 20 - DATA PARITY |
| 8 - DATA 0 | 21 - DATA 1 |
| 9 - Ground | 22 - DATA 2 |
| 10 - DATA 3 | 23 - DATA 4 |
| 11 - DATA 5 | 24 - Ground |
| 12 - DATA 6 | 25 - Not Connected |
| 13 - DATA 7 | |

LAN
8 Pin Mini-DIN Female RS-422



- | | |
|---------------------------------------|--------|
| 1 - Handshake Output | Output |
| 2 - Handshake Input or External Clock | Input |
| 3 - *Transmit Data | Output |
| 4 - Ground | --- |
| 5 - *Received Data | Input |
| 6 - + Transmitted Data | Output |
| 7 - General Purpose Input | Input |
| 8 - Receive Data | Input |

Falcon 030 Internal Memory Connector.

J6. 30 pin, dual row, upright male header.

1	GND	2	GND
3	VCC	4	MADDR 8
5	MADDR 7	6	MADDR 6
7	MADDR 5	8	MADDR 4
9	MADDR 3	10	MADDR 2
11	MADDR 1	12	MADDR 0
13	GND	14	VCC
15	VCC	16	GND
17	GND	18	VCC
19	MADDR 10	20	MADDR 9
21	WE	22	RAS 0
23	RAS 1	24	CAS0 H
25	CAS0 L	26	CAS1 H
27	CAS1 L	28	GND
29	VCC	30	VCC

J17. 50 pin, dual row, upright male header.

1	GND	2	VCC
3	MDATA 15	4	MDATA 14
5	MDATA 13	6	MDATA 12
7	MDATA 11	8	MDATA 10
9	GND	10	VCC
11	MDATA 9	12	MDATA 8
13	MDATA 7	14	MDATA 6
15	MDATA 5	16	GND
17	VCC	18	MDATA 4
19	MDATA 3	20	MDATA 2
21	MDATA 1	22	MDATA 0
23	DRAM 0	24	GND
25	VCC	26	GND
27	VCC	28	MDATA 16
29	MDATA 17	30	MDATA 18
31	MDATA 19	32	MDATA 20
33	MDATA 21	34	GND
35	VCC	36	MDATA 22
37	MDATA 23	38	MDATA 24
39	MDATA 25	40	MDATA 26
41	GND	42	VCC
43	MDATA 27	44	MDATA 28
45	MDATA 29	46	MDATA 30
47	MDATA 31	48	DRAM 1
49	GND	50	VCC

A value of 1 = open, 0 - closed.

DRAM 0	DRAM 1	
0	0	1MB (4 * 256k SIMMS)
1	0	4MB (4 * 1MB SIMMS)
0	1	14MB (4 * 4MB SIMMS)
1	1	RESERVED

Falcon 030 Internal Expansion Port.

The Atari Falcon030 has a full featured, internal expansion bus.

J20. 30 pin, dual row, upright male header.

1	D14	2	D13
3	D12	4	D11
5	D10	6	D9
7	D8	8	D7
9	D6	10	D5
11	D4	12	D3
13	D2	14	D1
15	D0	16	D15
17	GND	18	GND
19	GND	20	CPUBG0
21	EINT1	22	CPUBGI
23	500KHz	24	N/C
25	MFP IEI	26	MFP INT
27	EINT3	28	VCC
29	VCC	30	VCC

J19. 50 pin, dual row, upright male header.

1	GND	2	GND
3	BGK	4	AS
5	LDS	6	UDS
7	RXW	8	DTACK
9	FC2	10	FC1
11	FC0	12	BMODE
13	N/C	14	IACK
15	BG	16	BR
17	RESET	18	HALT
19	BERR	20	IPL0
21	IPL1	22	IPL2
23	CPUCLK	24	VCC
25	VCC	26	A23
27	A22	28	A21
29	A20	30	A19
31	A18	32	A17
33	A16	34	A15
35	A14	36	A13
37	A12	38	A11
39	A10	40	A9
41	A8	42	A7
43	A6	44	A5
45	A4	46	A3
47	A2	48	A1
49	EXPAND	50	N/C

RE-CASING THE ATARI - Put your Atari into a PC tower case.

This article was originally a series of letters between Ludo J. Sak and myself. Ludo is Dutch, and so English is a second language for him, so the English he used was a little confusing. Thankfully my Grandmother (Oma) is Dutch, so I still had no idea what he was talking about ☺, and here is the result.

It was originally intended for the Falcon, but there is no reason at all why you cannot do the same to your ST or STe.

Firstly, if you do go ahead and do this modification, it is good to know that all it does really is replace the dodgy ST-style case of the Falcon with a more elegant and functional PC-style tower casing, and adds a nicer, separable TT or Mega ST(e) keyboard to the machine. If you do it as follows, all you have to do to get your original Falcon/ST(e) back together is spend a few hours with a screwdriver and replace the case.

Building a tower case for the Falcon is very simple. If you have limited amounts of money to spend (as with most of us), it is much better to assemble your own Falcon-Tower, as the current batch of third-party Falcon cases are quite expensive, and it also allows you to get an insight on the workings of your bird ...

The first thing to do is buy a tower case. You will generally have to buy one of the bigger, full tower cases, as the Falcon motherboard has to fit in standing upright. You may be able to get away with buying a midi tower case(not mini, midi - it's somewhere between mini and full tower size.), but it could be a tight squeeze. I have also heard that it is possible to 'buzz' a $\frac{1}{16}$ or an $\frac{1}{8}$ of an inch off either end of the motherboard without any problems, but I don't know if I'd be doing that to my machine...

With the tower case, it needs to have a physically small power supply, so that it does not interfere with the motherboard. As for power rating, it's probably best to go for a 200+ watt supply, but considering the a ST runs at about 100watts maximum (that includes an internal disk drive), and even if you wanted to run your monitor off the same supply, you would most likely only need a 200 watt supply. However, if you are going to have 6 floppy drives, a 20" monitor, a CD-ROM, 3 hard disks and a powered sound amplifier running off the power supply, you may wish to consider a larger rated model. A 200W power supply should suffice for most applications.

In Australia, the tower case should cost about AUSS\$150 for the full tower model, the price is generally pretty stable between shop to shop, it's just the quality and features of the case that determine the price. (A really nice model has a flip-door on the side so that 2 screws allows full access to the internal workings, about AUSS\$150 if I remember correctly.)

In the Netherlands, Ludo bought his for about 275 guilders, and that had a 230 watt power supply.

Now comes the fun part. Open up your Falcon, and remove the plastic cover, the hard disk and the floppy disk drive. (Ludo failed to mention that it is probably a good idea to remove the RAM board too, as this reduces the amount of things that can get hit and thus broken.) Also detach the keyboard from the motherboard.

There are now two paths you may take. In his original design, Ludo drilled holes for all of the ports of his Falcon into the back of the tower case, and mounted it directly to the case. After some talking with Glen Elliott, we decided that a neater, safer option would be to mount the motherboard inside the case the same way as the PC does, and run connector cables from each of the connectors on the motherboard to their replicates on the external port of the case.

This second option has the added bonus that you may also add the Enhanced Joystick plugs to your case, and also the cartridge port, as with Ludo's method, these would either be lost, or turned into holes in the top or bottom of the case. The only problems with this second method is that the cartridge connector and the Falcon's SCSI II Honda-style plugs, while industry standard (well the Honda-style plug is anyway), they are expensive to buy. As of writing, cables with the Honda style plugs range from \$65 to over \$130 for a single cable. One way to get the cartridge port is to get a burnt-out 520 or similar computer, and steal the cartridge connector from that.

A way to make the cost of buying a Honda-style cable a once-only occurrence is to do what we did. Buy the cable, and hack off the end with the goofy-looking plug on it. To this attach a 25 pin plug and re-wire it. It involves a few more connections, and a little more work, but the end result it the machine is much easier to connect, and is pin-compatible to the TT SCSI connector.

One note however, keep the cables as short as possible. The more length you waste inside the machine, the less length you will have to position the devices connected to the machine. Some devices, such as hard disks, don't like being separated more than 2 feet (60 cm) or so from the machine, so using a foot of cable inside the machine means you only have a foot of cable left outside.

How to make the SCSI II to SCSI cable.

The Falcon has adopted a SCSI II Honda-Style cable connector, but if we look closely at it, only about 23 (individual) connections are made. This enables us to map it to a TT style SCSI DB25 Female Plug.

To achieve this, we would map the following:

TT-Style SCSI 25		FALCON-Style SCSI II 50	
*REQ	1	49	REQ
*MSG	2	46	MSG
*IO	3	50	I/O
*RST	4	45	RST
*ACK	5	44	ACK
*BSY	6	43	BSY
GROUND	7	24	GROUND
DATA 0	8	26	SCSI 0
GROUND	9	25	GROUND
DATA 3	10	29	SCSI 3
DATA 5	11	31	SCSI 5
DATA 6	12	32	SCSI 6
DATA 7	13	33	SCSI 7
GROUND	14	35	GROUND
*C/D	15	48	C/D
GROUND	16	36	GROUND
*ATN	17	41	ATN
GROUND	18	40	GROUND
*SEL	19	47	SEL
DATA PARITY	20	34	PARITY
DATA 1	21	27	SCSI 1
DATA 2	22	28	SCSI 2
DATA 4	23	30	SCSI 4
GROUND	24	42	GROUND
NOT CONNECTED	25	37	NOT CONNECTED

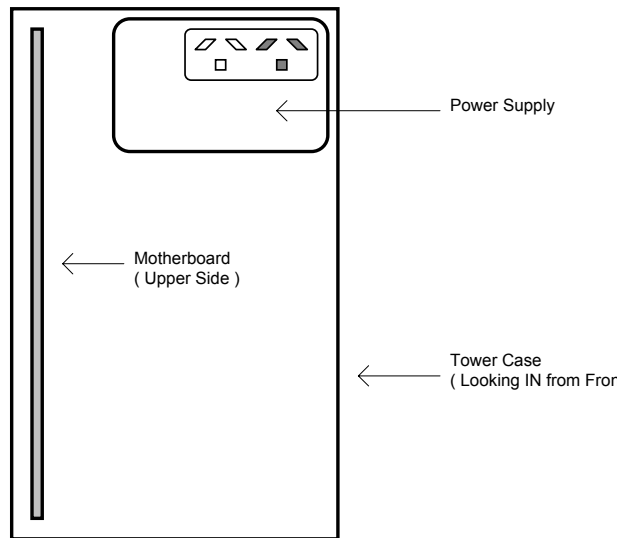
Ahh shit. What about 11,38- +5V.

If you do not take the cable-connect-up method, you're off to do the Ludo drill method. One word of warning though - if you do decide to do the Ludo Drill Method, you will most likely end up having to make one or two cable extensions anyway, as some ports (MIDI, Cartridge) are in inaccessible positions.

Once you have opened your machine and have removed all the loose components (HD, RAM Cards, Floppy Drive, etc.) it is time to start mapping out your holes for drilling into the PC Tower case. Once you have taken out the motherboard, you will notice that the shielding (metal sheeting) contains a map of all the holes to be drilled. You do not have to take the shielding off the bottom of the motherboard; in fact, it is probably safer that you don't as it provides some extra protection when doing the installation.

Now the hard work starts. Take the cover off the tower case and remove the front console. (This should be relatively easy to achieve - just a few screws.)

Now work out where you wish to place the motherboard. The best way to do this is left-down. By this, I mean that the motherboard goes into the left side of the case with the up side of the motherboard facing into the tower case.



Some cases may have the power supply in an awkward position, so you may have to move it. It's a simple procedure of unscrewing it from the case, drilling some holes and re-mounting it onto the case.

Once you have decided on the positioning of everything, grab the metal shielding and using it as a template, mark out where the holes for the connectors are to be.

Once you have checked that the motherboard fits into the new position, and that the corresponding holes line up, you can finally cut the holes out. The best way to do this is to carefully drill most of the waste metal out using a drill, then clean the holes up with a hacksaw and metal files. It is also a good idea to drill holes for the screw-connectors on the connectors (the sockets you screw the cables into for a secure fit), as these allow you to screw the motherboard in more securely.

After cleaning up all your holes, you can screw the motherboard in, and connect the power-supply to the motherboard. You will have to do a slight re-wire for the power plugs, as the PC and Atari standards are slightly different, so here's the color-combinations for you.

Purpose	PC Color	Atari Color
+12 Volts	BLACK	
12 Volts Return		
5 Volts Return		
+5 Volts		

If you have a Falcon030 with a 2.5" IDE drive, it is probably best to remount the hard disk into its original bracket. The floppy drive(s) can be mounted into mounting brackets and screwed into the bays of the tower case. The same goes for 3.5" hard drives, CD-ROMs, etc. If you have an external hard disk, it may be possible to mount it into the tower case, that possibility I'm afraid is up to you to determine. After everything else is connected in, you can connect an external keyboard to your new beast, either by extending the original keyboard's cable or by attaching a Mega ST(e) or TT Keyboard.

The connections to attach an external keyboard depend entirely on what sort of keyboard you have and what sort of machine you have. If I remember correctly, the connections for the TT and Mega STe keyboards are interchangeable - you can use a Mega STe keyboard with your TT and vice versa. To determine the connections, refer to the section *Attaching Mega ST(e), TT Keyboards to ST(e) and Falcon030*.

Tower cases also come with a MHz indicator. Unfortunately these are hard-wired via a set of jumpers to a set speed, so people with switchable speed machines (Mighty Sonic, Speed Resolution Card, ADSpeed 16, etc.) you cannot connect a certain pin to the indicator (although this should be possible via a small circuit board - maybe I'll do that in a later hack ...). Unfortunately you'll have to set the jumpers manually - if you buy an empty tower, you'll get the description of the jumper settings and it is not that difficult to set up.

The hard disk activity light you cannot connect via the board of your ST, because it is only supported on machines like the Falcon and the TT, who have internal hard disk support. You can connect your hard disk light to the tower LED though. If you have a newer style of hard disk, you can simply plug it on the connector of your harddisk. This will be a two-pin jumper, usually on top of the hard disk, don't confuse it with the SCSI connector or drive selection jumpers.

The reset on the back of your computer doesn't have to be modified. The reset can be tapped off from the keyboard connector, so reset is possible via the reset button of the tower.

The tower-case has a reset-switch with two wires connected to it. Connect these two wires to pins 7 and 8 of the internal keyboard connector. Connect the black wire to pin 8 (a ground wire) and the red wire to pin 7. (This works for the Falcon030, other computers may differ.)

The tower case also has a HD-activity light. On the Falcon030 you can connect this to pins 3 and 4 of the internal keyboard connector. This has the added bonus that any hard disk you access from a hard disk (not just the internal if there) will show up on the led.

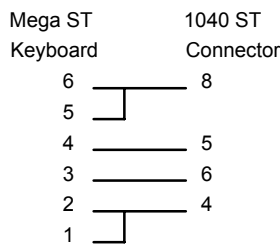
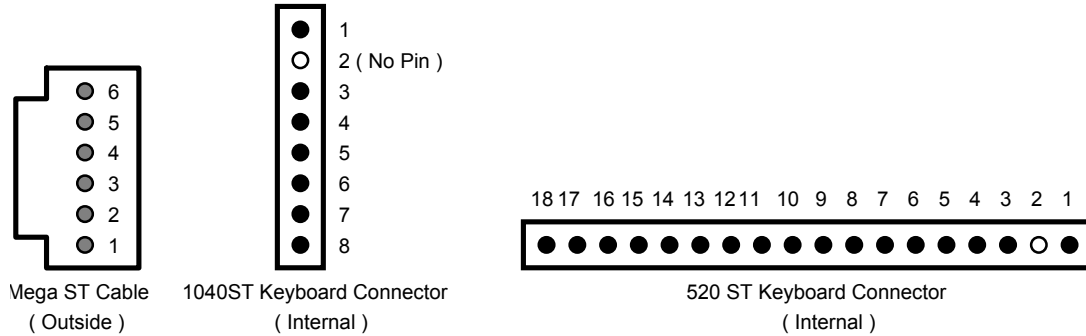
And that's it! After a lot of hard work, you have successfully (hopefully!) transferred your Atari Machine into a useful Tower Case!

Attaching Mega ST(e), TT Keyboards to ST(e) and Falcon030.

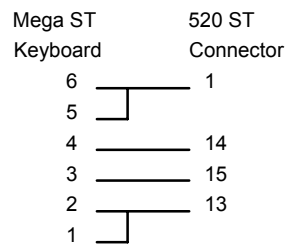
Whoever at Atari designed the ergonomics of the ST(e) and Falcon series should be severely beaten, but there is one way you can improve your RSI ratio while using your Atari - add an external keyboard.

Here's how to connect a MegaST keyboard to a 1040ST. It was taken from *ST-Computer*, October 1990, from the *Quick Tips* section.

It's probably best to get a 6-pin western connector plug and stick it in the right side of your computer's case, but you can solder the wires directly to the internal plug.



Mega ST to 1040ST



Mega ST to 520ST

Here's how to connect a TT Keyboard to a Falcon030. It was taken from a letter by Ludo J. Sak.

TT Cable	Falcon Connector	Falcon Signal
	o	10 -
	o	9 -
Black	o	8 - Ground
	o	7 - Reset
Red	o	6 - Receive Data
Orange	o	5 - Transmit Data
Yellow	o	4 - +5V
Green	o	3 - LED
	-	2 -
	o	1 -

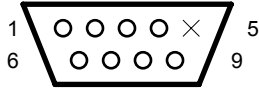
In the above table, the column marked "Falcon Connector" shows the pin-out of the keyboard connector in the Falcon itself. You will find this configuration when you disconnect the keyboard from the motherboard. The 'o' denotes a pin, the '-' denotes a pin is missing.

If you use this modification in conjunction with a tower casing, the ground and reset pin on the board can be connected to the tower reset button, so that pushing that button makes your machine reset. Red, orange, yellow and green are for the keyboard (soft reset on keyboard still works as well as the joysticks, mouse and led). If you use ground and reset for the reset button, then you can use another ground pin for working with the keyboard. If you wire the black wire to the lowest pin, it works fine.

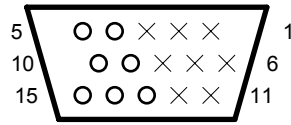
Note: The soft-reset (CTRL-ALT-DEL (warm), RSHIFT-CTRL-ALT-DEL (cold)) from the keyboard will only work on TOS versions that support it. On TOS 1.0 this does not work, but 1.62 and up does. Earlier TOS versions may work, but I only have TOS versions 1.0, 1.62 and 4.04 to test it on.

Atari TT to Monitor

Here's how to connect a 9 pin female D-Sub to a 15 pin male high density D-Sub such as when connecting a TT to a non-proprietary monitor.



9 Pin D-Sub (Female)



15 Pin HD D-Sub (Male)

Pin #	Signal	Pin #
1	ECL Video	4
2	ECL Video Return (Ground)	5
3	Horizontal Sync	13
4	Vertical Sync	14
5	Not Connected	
6	XECL (Not) Video	15
7	XECL (Not) Video Return (Ground)	10
8	Sync Return (Ground)	10
9	Monitor Detect (Ground)	9

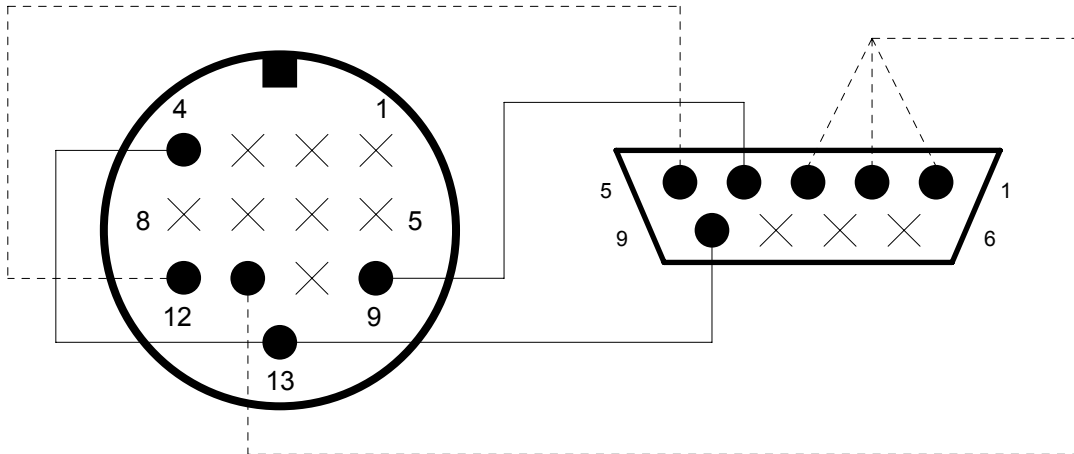
The following signals should be twisted pair lines:

1 - 2	ECL Video and ECL Video Return	28AWG, 75 ohms
6 - 7	XECL Video and XECL Video Return	28AMP, 75 ohms
3 - 8	Horizontal Sync and Ground	100 ohms
4 - 9	Vertical Sync and Monitor Detect	100 ohms

The TTM194 has the following specs:

Power Source	
Input Voltage	220-240 VAC
Power Frequency	50Hz
Power	64 Watts
Inrush Current	40 Amps peaks at 240 VAC

ST Mono to VGA Monitor

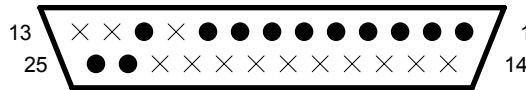


ST (e)		MS II	
+ - 4			
9	-----	4	(H/Sync)
11	-----	1	(Red)
		+ - 2	(Green)
		+ - 3	(Blue)
12	-----	5	(V/Sync)
+ 13	-----	9	(Ground)

To get monochrome, you combine R, G and B, hence the connection between 11 and 1,2,3.

To detect monochrome monitors, the ST(e) series have a Monochrome Detect pin which is just grounded inside the SM124/125/144 series, so ground it on pin 13 on the ST side.

Pin connections for 4 joystick adapter

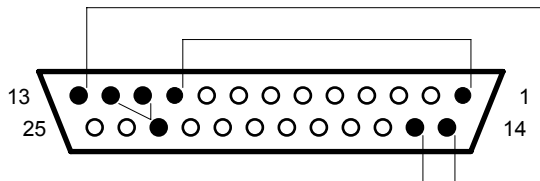


Atari Printer plug connections.

Printer Port	Joy 3	Joy 4
1	6	
2	1	
3	2	
4	3	
5	4	
6		1
7		2
8		3
9		4
11		6
24		8
25	8	

Fix for <ALT>+<HELP> Screendump Wait

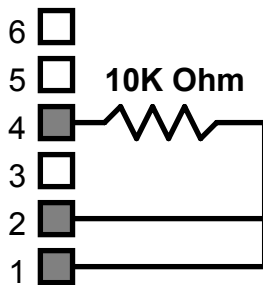
Isn't it annoying when you accidentally press the combination of <ALT>+<HELP> and you must wait ages for the computer to realize that it is not connected to a printer? Well, here is a fix for when you don't have your printer connected. Take yourself a Sub-D 25 pin male plug, and solder the pins as per the following diagram. (The diagram assumes you are looking at the solder cups.) Connect the plug to the printer port of your Atari, and now the computer will then receive it's own time-out signal, thus eliminating the very long wait.



<ALT>+<HELP> Screendump Fix

Fix for disconnected Mega ST keyboards.

If you remove the keyboard of the Mega ST and switch it on, the computer won't boot up and sits there sending alert-beeps to notify you that the keyboard is disconnected. If you need to disconnect your keyboard (i.e. for a secure BBS server, etc.) you can make up a plug to attach to your internal keyboard connector, and the Mega ST will boot up as normal.



Internal Connector

Falcon 030 MultiSync Video Adapter

When I initially bought my first Falcon 030, I had downloaded some demos from an FTP site to test out the beast's awesome graphics capabilities. I was indeed impressed by what I saw, but some of the demos flaked out on my monitor. I initially thought that I had dud copies of the demos, but later I found out that the Falcon has a neat hardware detect on it's video port.

The Falcon video port is a 19 pin D-Sub job, so there is a few extra pins to play with than your average 15 pin VGA adapter. Of note are pins 18 and 19. These pins, known as M0 and M1 determine what sort of monitor is attached to the Falcon. They work in a similar way to the Monochrome Detect pin in the ST(e) series, but as there are two pins, we can select a variety of type of monitor (O.K., 4 anyway) attached to the Falcon. The settings are as follows:

M0	M1	Monitor Type
0	0	ST Monochrome (SM124)
1	0	ST Color (SC1435 / SC1224)
0	1	VGA
1	1	Television

A value of 0 means the pin is grounded.

So with the adapter connected to the Falcon, the Falcon TOS (4.0x) determines what monitor is attached. Normally this isn't a problem, because the detection is actually a safeguard against blowing up your monitor. With normal VGA monitors, the sync rate cannot go low enough and so some games, etc. which try to get RGB video modes die. With normal RGB monitors, the sync rate cannot go high enough for certain resolutions, and so they die too. But many Atari users are not normal ☺, and have MultiSync monitors. MultiSync monitors are analog monitors which allow the displaying of resolutions down to 15.8kHz, which is what the Atari computers use for their 320*200*16 color modes. The MultiSync monitors are in effect, a RGB and a VGA monitor rolled into one.

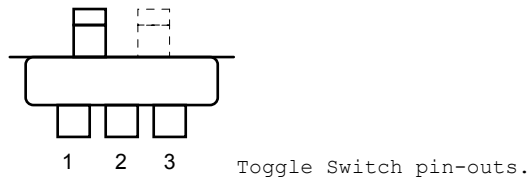
Now which monitors are MultiSync is hard to determine, you have to find that out from the specs of the monitor. Not all 'MultiSync' monitors are true MultiSync's however, and may be just Multi-Scan monitors. This is because NEC, who made the original MultiSync series (the MultiSync II and 3D are the models to look for ...) sort of took out a copyright on the MultiSync theory, anyway ...

When using the standard Atari VGA adapter, the Falcon TOS thinks that you have a VGA monitor attached to your machine, and treats it accordingly, not allowing certain frequencies out of the computer. As the M0 and M1 switches only don't allow for MultiSync monitors to be specifically selected, so we have to make a special cable to address all of the video modes supported by the Falcon.

The first part of the cable is to make a 'normal' VGA cable, connecting the lines from the 19 pin Falcon Video Adapter to a 15 pin VGA socket, as shown in the table below.

Signal	Falcon (19)	MultiSync (15)
Red	1	1
Green	2	2
Blue	3	3
Ground	5	5
Red Ground	6	6
Green Ground	7	7
Blue Ground	8	8
Ground	10	10
Ground	11	5
HSync	13	13
VSynC	14	14

The second part of the cable is to put in the switch to toggle between VGA and RGB modes. This is as follows:



Signal	Falcon (19)	Switch (3)
M1	18	1
M0	19	3
GND	11	2

The above switch is in the VGA mode, the shadowed position of the switch will put it in VGA mode. The toggling of the switch will alternately pull M0 and M1 low, resulting in a selection of either ST Color or VGA mode.

It is best to use all the ground lines possible (RED terminated with RED Ground, etc.) as this ensures the cleanest possible signal. You will however have to connect pins 5 and 11 of the Falcon to pin 5 of the MultiSync as the MultiSync is one GROUND line short.

You can mount the toggle switch in the DB19 case, but this is damn tricky to do, and you run the risk of having wires short together. I did this for my first cable, but for my second attempt, I made it into a separate switch box about 15cm from the Falcon plug with audio and video outputs as well.

The Falcon TOS checks the status of M0 and M1 every time a program does a change of video mode. (*Some demo writers do not follow this, and hard-switch the mode for some demos, as they know many people have MultiSync monitors - be careful about this 'feature' of some programs.*) This means that you can flip the switch *on-the-fly* when you are running the computer. For instance, when you are in a VGA mode (the best ones - no interlace!), and you want to run the latest demo, you would flip the switch before running the demo and it would then run correctly in it's native RGB mode. When you exit from the demo, you should flip the switch yet again so that the Falcon TOS does not try to set a RGB mode with the same settings as your previous VGA mode - this can lead to a screwed up screen.

And that's about it! What you need to build the cable is as follows:

```

1 * DB19 Female Solder Connector
1 * HD15 Female Solder Connector
1 * DB9/HD15 Hood
1 * DB19 Female Solder Connector
1 * HD15 Female Solder Connector
1 * 3 Pole, Double Throw Toggle Switch
Several strands of Colored wire - I used 25 Wire RS232 Cable

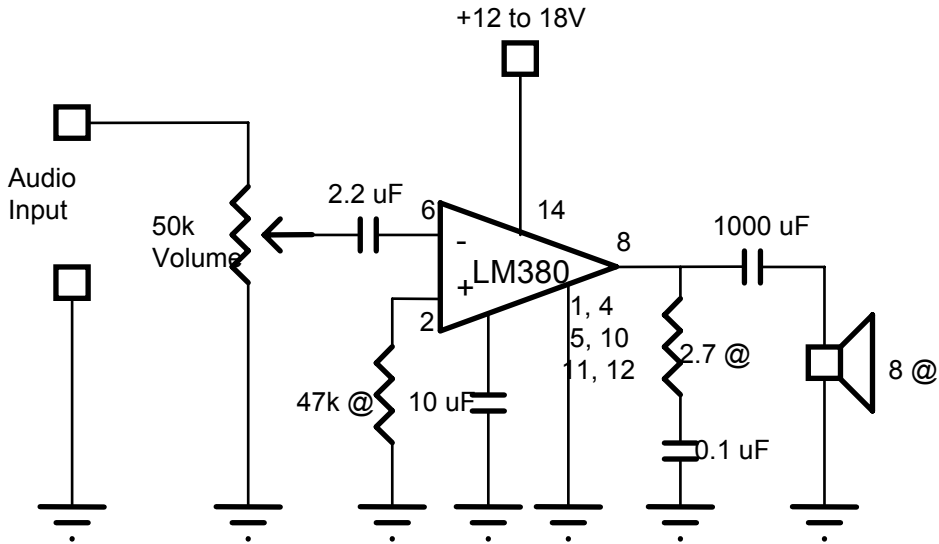
```

Happy Falconing!

ADD AMPLIFIED SOUND TO YOUR ST

Original Article by John Ogrizek

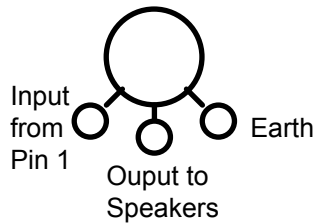
The sound output of the monitor on a ST is not all that good and with this simple modification to your ST you can run sound externally to a HiFi amplifier, a pair of amplified speakers (the type used for walkmans) or build a simple amplifier as shown in the circuit below.



1/8 " Socket



Rear View 50k Pot.



All that is required to modify your ST is a $\frac{1}{8}$ " Mono headphone type socket, a small amount of soldering and about half an hour of your time.

First of all, remove the screws on the underside of your ST and remove the top cover. Then remove the keyboard by removing the plug gently. To get to the monitor socket you will also need to remove the power supply which is held in by two screws. It is almost possible to access the monitor socket from here, but to make things easier, also remove the shielding. The shielding is removed by first removing the disk drive screws and then the screws on the shield as well as bending the metal tabs which hold the shielding on.

Once this is done, the next thing to do is to find pin 1 (sound out) on the monitor socket. Looking in from the board side it should be the top left hand pin. Cut the metal strip of pin 1 (on the back of the monitor socket) and carefully bend down one side so that the two strips are clear of each other. Now solder one wire to each side of the strip on the connector and one wire to the large earth pad on the motherboard.

Replace the shielding and run the three wires up through the shielding of the power supply. Now the $\frac{1}{8}$ " socket needs to have a hole drilled for it. I placed the socket so that it sits just above the shielding, be careful here in lining up the hole to be drilled. Once you have drilled the hole, insert the socket and wire it up as shown in the diagram. The socket is wired up in the manner so that when there is no plug in the socket the sound will come through the speaker in the monitor, but when a plug is inserted into the socket, the sound to the monitor is shut off and redirected to the external destination.

Reassemble your computer and make sure everything is normal without any plug in the socket. Now you can test it with a lead to your amplifier, HiFi, whatever. One idea is to use a pair of self-amplified speakers, such as those used with PC soundcards today.

If you have a pair of speakers lying around idle, you can also make your own amplifying circuit (read : *amplifier*) as shown in the diagram above. For a two speaker system, simply build two of the circuits. The sound output is good and there is plenty of amplification. You can also incorporate the power supply into the unit so that everything is in one box.

(An idea might be to incorporate all of this into the ST itself, and draw the power from the power supply inside the ST, but be careful - the ST supply is pretty, well, *crap*, to put it bluntly.)

Parts list for each unit of the amplifier as noted above:

- 1 - LM380 Operational Amplifier
- 1 - 2.7 Ohm Resistor
- 1 - 47K Ohm Resistor
- 1 - 2.2 microfarad electrolytic capacitor
- 1 - 10 microfarad 25V electrolytic capacitor
- 1 - 1000 microfarad 16V electrolytic capacitor
- 1 - 0.1 microfarad metallised polyester capacitor (greencap)
- 1 - 50K Ohm Potentiometer
- 1 - 8 Ohm Speaker
- 1 - 12 to 18V Power Supply

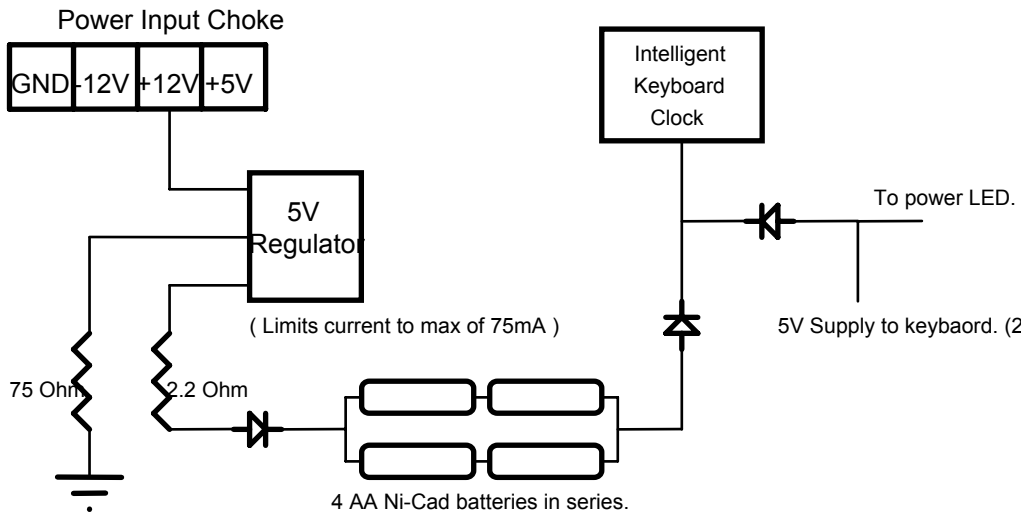
ADD A BATTERY BACKED-UP CLOCK TO YOUR ST.

A circuit modification to the intelligent keyboard power supply that allows the time of day clock to have uninterrupted power.

***** Warning! *****

This circuit may void your warranty! Or turn your ST into a toaster!!

The version of the operating system 7/85 has a bug that loads a predetermined time and date into the clock at power on and reset. (This bug was corrected in later TOS versions.)



The keyboard draws 20 mA, the batteries provide 500mA.

- (1) - Cut the positive wire of the LED and run this wire to the LED and connect.
- (2) - From pin 13 plug from motherboard to keyboard. Cut wire and add a diode in series with the wire as shown above.

The ST may be turned off for an unlimited time and the clock will still be powered, but you only have approximately 24 hours if the ST is unplugged from the wall (disconnected from the AC supply.)

STe Extra Joysticks

I always wondered why the hell they put those extra analog joysticks on the side of the STe series, and it wasn't until the Falcon came out that they started being used and I found out. Anyway, while browsing on the net one day, I found the following information about how to address these joystick ports.

To read the joystick, write a mask value corresponding to the row of keys/positions you wish to interrogate to 0xFF9202. Next, read back a WORD from either 0xFF9200 or 0xFF9202. Which address you read from is dictated by what you wish to interrogate. As indicated in the table, cleared bits mean that a key is being pressed or a joystick is moved in that direction.

Read Controller 0 at 0xFF9200

Write Mask	Bit 0 Clear	Bit 1 Clear
0xFFFE	Pause	Fire 0
0xFFFD	-	Fire 1
0xFFFB	-	Fire 2
0xFFF7	-	Option

Read Controller 1 at 0xFF9200

Write Mask	Bit 0 Clear	Bit 1 Clear
0xFFEF	Pause	Fire 0
0xFFDF	-	Fire 1
0xFFBF	-	Fire 2
0xFF7F	-	Option

Read Controller 0 at 0xFF9202

Write Mask	Bit 8 Clear	Bit 9 Clear	Bit 10 Clear	Bit 11 Clear
0xFFFE	Up	Down	Left	Right
0xFFFD	Key *	Key 7	Key 4	Key 1
0xFFFB	Key 0	Key 8	Key 5	Key 2
0xFFF7	Key #	Key 9	Key 6	Key 3

Read Controller 1 at 0xFF9202

Write Mask	Bit 8 Clear	Bit 9 Clear	Bit 10 Clear	Bit 11 Clear
0xFFEF	Up	Down	Left	Right
0xFFDF	Key *	Key 7	Key 4	Key 1
0xFFBF	Key 0	Key 8	Key 5	Key 2
0xFF7F	Key #	Key 9	Key 6	Key 3

To the same port you can also attach paddles, for games like PONG back in '82 or whatever. I cannot see a real use for using paddles, but you could use the port set up this way for temperature measurement, etc.

Two paddles may be plugged into each joystick port. Each paddle returns an 8-bit value indicating it's position (0 = full counter-clockwise, 255 = full clockwise) at the addresses shown below. Unlike joysticks, paddle positions are returned automatically with no need to write to an address prior to a read. Paddle fire buttons, however, are mapped and read in the same manner as the joysticks. See the explanation of joysticks above for an explanation.

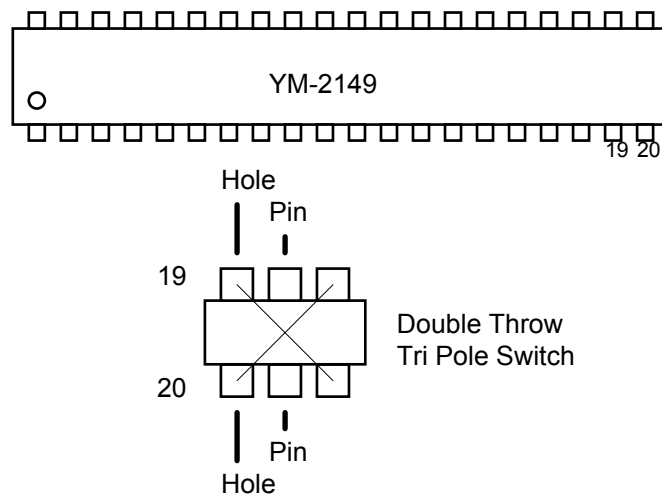
Reading Paddle Controller

Byte Address	Paddle
0xFF9211	X Paddle 0
0xFF9213	Y Paddle 0
0xFF9215	X Paddle 1
0xFF9217	Y Paddle 1

Finally, you can also attach light guns or light pens to the ports. These are more useful for shooting games like LETHAL ENFORCER and MAD DOG McCREE, so they're not a total loss ☺.

Joystick port 0 supports a light gun or pen. The position that the gun is pointing to is returned in the WORD registers at 0xFF9220 (X position) and 0xFF9222 (Y position). Only the lower 10 bits are significant giving a range of values from 0-1023.

B-BOOT MODIFICATION



Break pins 19 and 20 on the YM-2149 from the motherboard. (De-solder them, that is!) Connect the pins via wires to a double throw, three pole toggle switch. This then swaps the current drive to have the ID of 0, and hence the boot floppy.

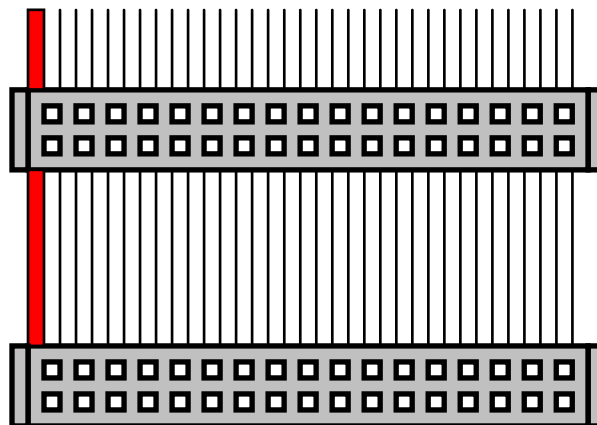
68000 Freeze Frame Button

To make a Freeze Frame Button, all you have to do is to short pins 16 and 17 on the MC68000 together, this will cause the internal PC to stop being incremented, effectively stopping the machine.

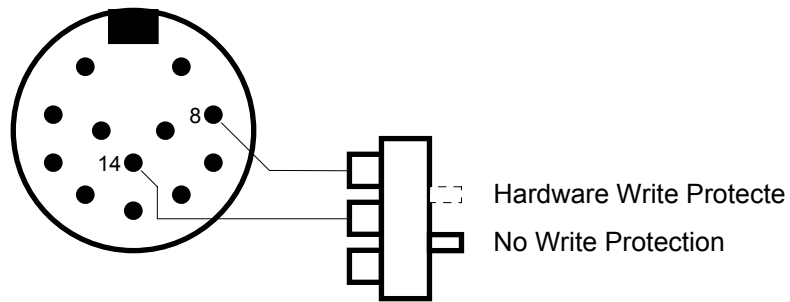
How to hook up a second floppy drive to a Falcon030!?

Sure, it's idiotically simple. Decide where you want the cable to come out of your case (I recommend the cartridge port, since it's already the right shape). Get a ribbon cable of the proper length to connect to the drive you want to add, plus the width of the Falcon and perhaps another 3 inches for luck. Digikey sells a 34-pin male IDC crimp-on header (rather expensive, IMHO, but hey, it'll do the job!) and you put that on the end of the cable; then add a female header right next to it, and another at the far end. Unplug your internal floppy, plug the new cable into the short on in the Falcon, plug the next female connector into your internal, route the cable, and plug the far end into your B: drive (change the jumper on that one to make it drive 2 (or 1 if the numbering scheme starts with 0). Test that it works, and put the case back on your bird.

It's not elegant, but it is simple. And there you have it. 2 DSHD drives.



HARDWARE VIRUS PROTECTOR.



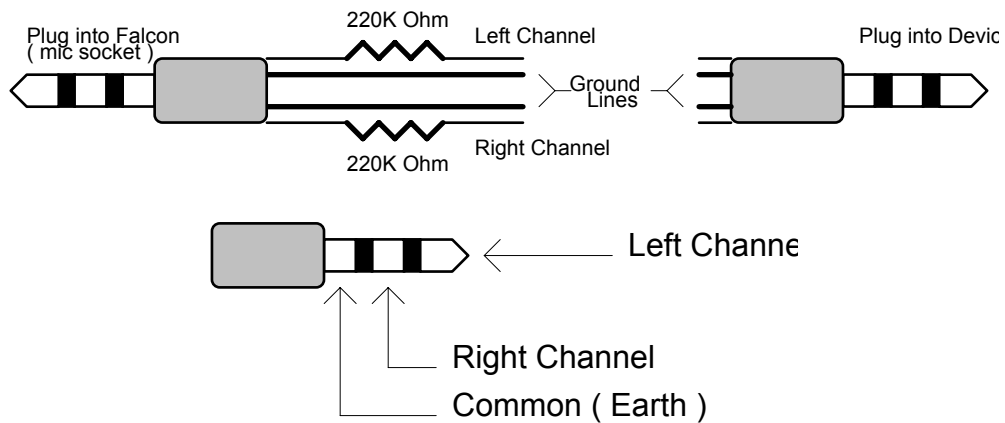
Idiotic, but effective. Ground Pin 8 of the Floppy Port, and the ST thinks that the floppy disk has the write-protect on.

FALCON RECORDING CABLE

With the initial batch of Falcon030's, the microphone jack was powered, for use with unpowered microphones, and so when connected to a 'Walkman' or the like, signal levels would be huge and cause great problems with the signal levels being way too high.

This problem is easily solved by using a specially adapted recording cable. You can make the cable yourself, as it's not very difficult and not expensive.

All what you need is a cable and two resistors, and connect it up as follows.

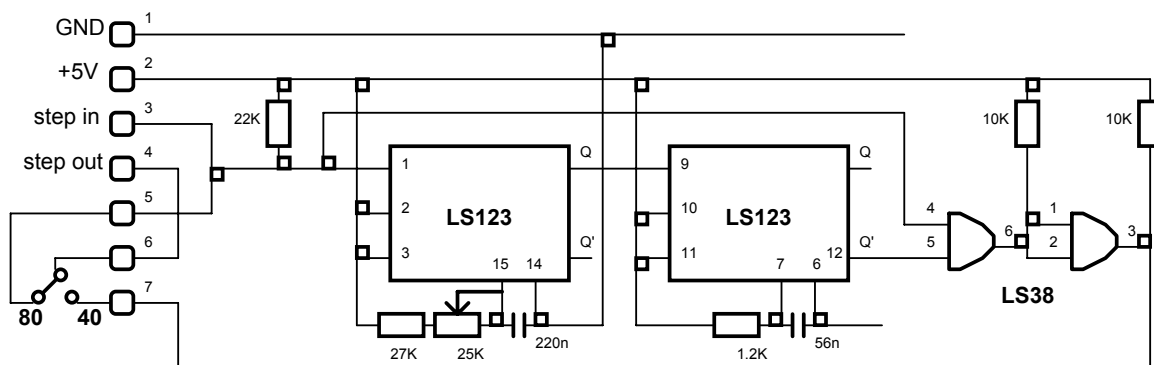


Hardware : Read IBM Disks with an ST.

This one comes from Michael Doerr, University of Kaiserslautern.
Taken from article 4513 of comp.sys.atari.st

This article is in response to the ever increasing number of people who want to hook an 80-track 5.25" drive to their ST, but want to read 40-track IBM-PC diskettes as well. Once upon a time (way back in 1986) the German magazine C'T published a solution called the step doubler. I built that circuit and can now happily read and write 360K MSDOS diskettes as well as making backups of my hard drive using cheap 5.25" diskettes holding 800K of data each. Below you'll see the schematic of the circuit.

If you carry out this modification, you're on your own - no warranty is implied if you fry your ST or your drive ...



N.B. Joins denoted by  .

Am Laufwerk ist die Leiterbahn, die von Pin 20 (Step) wegfuhrt, zu durchtrennen. Das Stck Leiterbahn, das noch mit der Steckerleiste verbunden ist, wird mit Pin 3 (Step In) verbunden. Die andere Seite wird mit Pin 4 (Step Out) verbunden. Zur Erhhung der Zuverlssigkeit sollte Pin 1 (GND) mit den ungeraden Pins (1-33) des Floppysteckers verbunden werden!

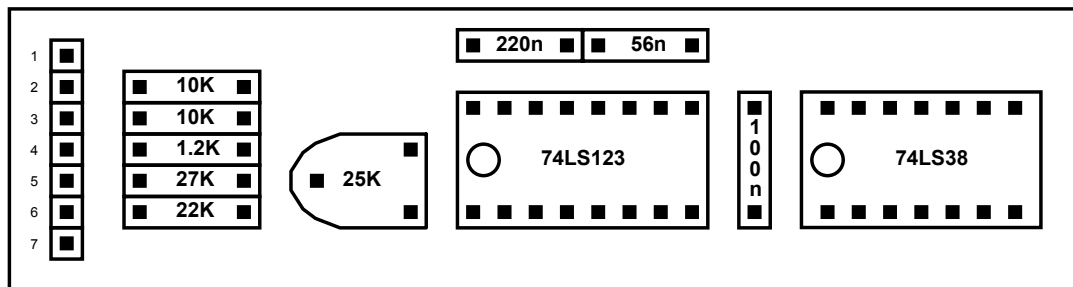
Note 1: Most 5.25" drives step flawlessly at 3ms but won't be able to do 2 steps in 3ms. That's why you should lengthen the step time of the switched 5.25" drive to 6ms. (You can do this by setting bits of an address in memory somewhere ... a variable at \$404 if I remember correctly.) With a Phillips X3134 you do not have to do this, as it is capable of doing 2 steps in 3 ms!

Note 2: While having a hot soldering iron, you should add some open collector TTL drivers (a 7407 is fine) to the following 3 signal lines:

- Pin 2 : Side Select
- Pin 5 : Drive 0 Select
- Pin 6 : Drive 1 Select

These lines are driven by unbuffered outputs of the Yamaha sound chip (YM2149) that isn't capable of driving heavy loads and may easily be damaged. All other outgoing lines on the floppy port are sufficiently buffered.

Layout:



68881 for the 1040

Author: Keith Sabine (keiths@cadence.com)

This describes how you can get a 68881 coprocessor to work with your Atari 1040 or similar.

Basically what you have to do is the following:

\overline{AS} is decode from address 0xFFFFA40. Only bits A23-A5 are used though. To do this I used two 74HC133 13 I/P nand gates, OR'ing the two outputs together using a 74HC32. Also FC0-FC2 are inputted to the 133's, decode as x01, i.e. do not use FC2 and invert FC1. A single 74HC04 will provide the inverters for the address decode and FC1.

\overline{DS} is just the AND of \overline{UDS} and \overline{LDS} . Other than this, the signals need to be connected just as shown in the 68881 data sheet (you'll need a copy of this, if only for the 68881 pinout). The following shows the connections I used, with MC68000 pins on the left and MC68881 pins on the right.

I soldered 2 26 way IDC cables with connectors right onto the 68000 pins (crude but my 68000 was not socketed, so there wasn't much choice). The use of IDC connectors is recommended as at least you can unplug your Atari and use as normal while you're checking your circuit. I built it on a small piece of board with plated-thru holes. If you can make PCB's so much the better, please send me one!

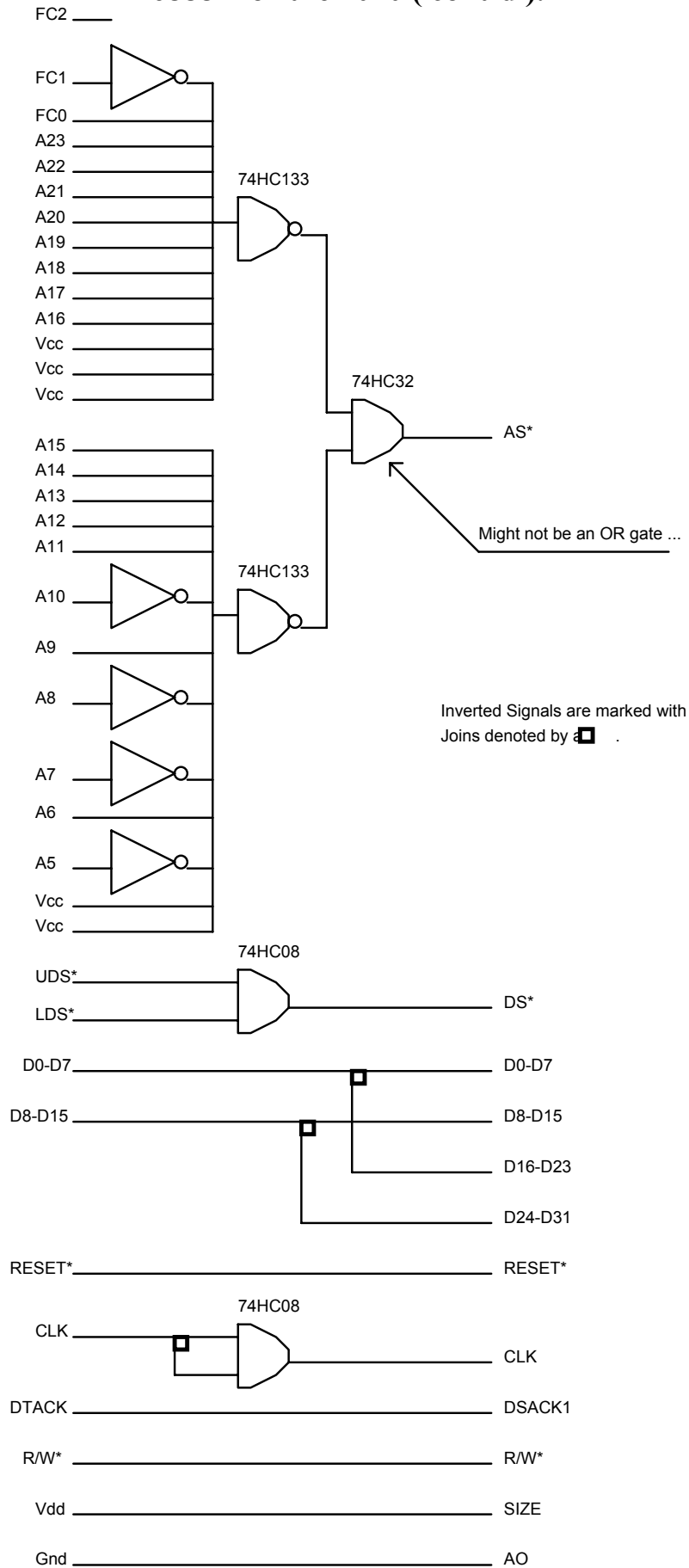
I buffered the clock from the 68000 using a spare AND gate in the 74HC08, since with ~8 inch leads the signal was a bit flaky, giving me errors at first.

Here's the complete schematic, there may be one or two 68000 -> 68881 signals that I have forgotten (I'm working from memory here, I'll check it over the weekend). If I have forgotten anything important I'll remail you.

I tested it with a simple Laser C program linked with the lib881.a library. A test of about 1000 log, exp etc. double precision operations took 7.00s with the standard lib (libc.a) and 0.53s with lib881.a.

Pretty impressive, huh?

68881 for the 1040 (cont'd).

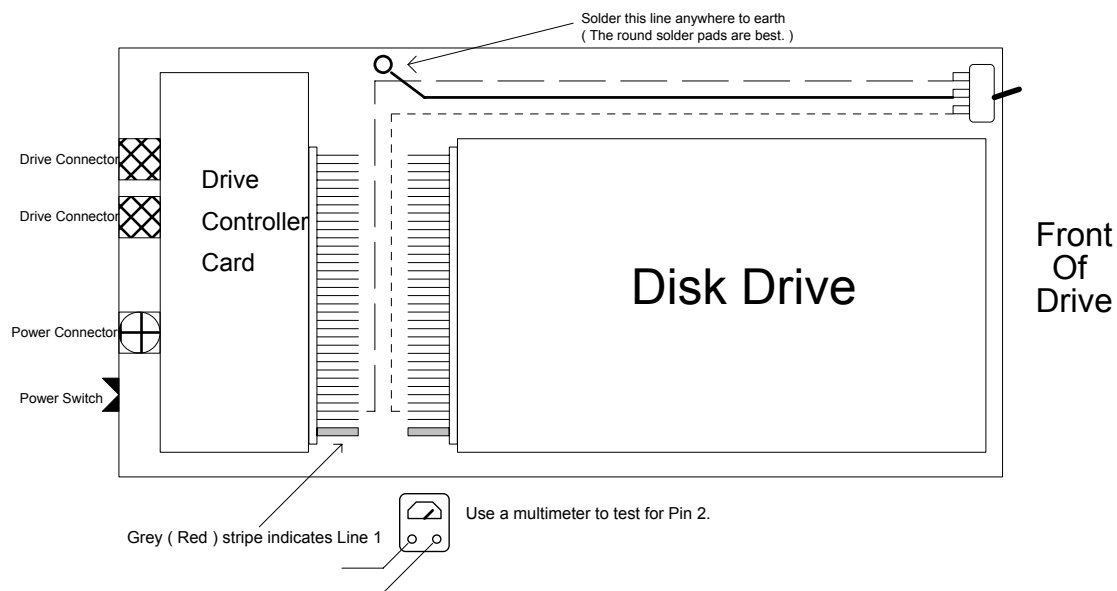


Double Sided / Double Density Drive Modification

In the early days of the ST, the 520ST came with an external disk drive. Many people thought that this was a nuisance, but in reality it was quite good - I mean, at least Atari got the right idea and made the mouse and joystick ports easy to access. Back in '85 when these types of computers were released, the floppy drives, the SF354 models, and most of the third party jobs were initially single sided. This trend continued for several years until the double sided drives, such as the SF314, phased the single sided drives out. There still posed the problem of single sided drives, and so for the best compatibility ratios, the software houses released their software on single sided disks for a time. Now with double sided drives as standard, it seems a waste to copy a single sided disk onto a double sided disk, and have a whole side of the disk wasted. To combat this problem, you can make a simple modification to your floppy drive(s) that allows two single sided disks to be backed up onto a double sided disk.

One warning though - make sure that you only format disks double sided on the top side, as formatting them on the bottom side can cause some bizarre effects ...

1. Remove the drive cover, etc.
2. Slice and cut the third (3rd) wire on the ribbon connector.
(Line one is marked with a red stripe.)
3. Solder fine wires, approximately 250mm long, to each of the pins on the pins on the toggle switch. Solder pins 1 and 3 of the toggle switch to each end of the cut wire.
4. Solder the middle (pin 2) pin wire anywhere on the motherboard to earth the switch. The best place to make this earth connection is on one of the circular pads on the motherboard where screws come through.
5. Fit the switch to either the front or the back of the drive casing (the front is usually the easiest to toggle) and solder the other wires as shown.



In some models, the red stripe indicating line one is on the opposite side to that shown in the diagram. Don't worry, just find the third wire from the red stripe, and this should be "Side-0-Select" which is what we are trying to change.

Switch Positions:

- Up - Normal = Double Sided or Single Side Top
- Down - Changed = Single Side Bottom

ONE-MEG (PIGGYBACK) UPGRADE FOR 520.

By Larry Rymal, <Z468252@SFAUSTIN.BITNET>, Newsgroups: comp.sys.atari.st

(This was REVISED AND TESTED by an anonymous engineer on Atari's development staff. The addition of the resistors should provide a long life to your machine, but the warning below is STILL IN EFFECT. This is not an official sanction of the modification. USE WITH CARE!!!)

WARNING: This is a hardware modification that will void the warranty of your 520ST. If you do not have the appropriate tools or experience you have a substantial chance of ruining your 520ST. Proceed at your own risk! This modification has been in my 520ST without any problems for 6 days now. However, I have (of course) not checked with knowledgeable sources at Atari to verify if this modification endangers the long term machine reliability and/or software compatibility (I suspect it may endanger their software compatibility if enough of us do it!)

Tools & components needed :

- 16 256k * 1 RAM chips, 150ns access time type, e. g. NEC 41256C-15
- A good quality, preferably temperature controlled soldering iron, with a miniature tip (tip should be narrow enough to avoid touching 2 I. C. pins at the same time). E. g. Weller type soldering station.
- Good quality resin core solder (thin).
- Approximately 4 foot of #24 AWG insulated wire and a good stripper for it and 2 feet of #22 AWG solid tinned copper bus wire. You will have to route 3 wires over a sequence of I.C. pins.
- Six (Nine?) 68Ω 1/4W ±10% carbon film resistors.
- Desoldering wick and solder suction tool.
- Philips type screwdriver (for opening your ST), tweezers, pliers, etc.
- A steady hand and self-confidence.

Explanation of the modification :

(Please read the rest of this document before starting. It may save you time and an 520ST.)

The current memory inside the 520ST consists of 16 256K*1 RAM chips. Address (A0..A8) lines are common to all those chips. The WriteEnable line is also common to all chips.

Data (in and out) lines are of course individual. The RAS (row-address strobe) line is common to all chips. The 8 chips forming the high order byte group have one common CAS line, and the 8 forming the low order byte group have one common CAS line (CAS is used as enable for write operations, such that WriteEnable can be common to both groups).

The high order group from MSB to LSB consists of U45, 44, 43, 42, 38, 34, 33, 32.

The low order group of U30, 29, 28, 25, 24, 28, 27, 26.

Note that all chips are adjacent, though the numbering has gaps. RAS0, CAS0H, and CAS0L are supplied from U1 pin 8,6 and 7 respectively (The 0 indicates bank 0) Bank 1 that you are going to build in will be "piggy-backed" on top of the current chips, where all pins of the new chips EXCEPT RAS (pin 4) and CAS (pin 15) are soldered to the old chips equivalent pins.

Thus they will end up sharing addresses, data, WriteEnable and power and ground with the existing chips.

All RAS pins of the new chips are wired together and will be supplied with the "RAS1" signal generated on pin 18 of U15 (the memory controller, marked 3H-2119C or so). The CAS pins of the 8 new high order byte chips (on top of U45..U32) are wired together and supplied from the "CAS1H" signal generated on pin 22 of U15. Analogously, the CAS pins of the new U30 to U16 are wired together and supplied with "CAS1L" from pin 21 of U15.

How to go about it:

- Step 1: Open up your 520ST, pull off the keyboard connector and remove the main circuit card from its top and bottom shielding. Make sure to remember which screws go where and note the keyboard connector orientation.
- Step 2: Desolder all of the capacitors adjacent to the existing RAM chips.
(*DO NOT SKIP THIS STEP. You'll lose time if you do, and worse, the modification will not be reliable since you can't solder pins obstructed by the capacitors reliably (if at all)*).
To desolder them, I found it easiest to heat the island on the non component side, and bend the wires straight. After doing that on each capacitor, turn over to the component side and heat the islands while pulling the capacitor out with the tweezers.
- Step 3: Open up the holes of all the desoldered capacitors, using a combination of de-soldering wick and suction tool. Do this from the non component side. If certain holes are difficult to open up, you may want to use a wood splinter. (Push it through while heating.) Be careful to remove all solder debris!! THE REASON for opening the holes NOW is that they will be less accessible once you've done the other steps! Patience is a virtue.
(*NOTE: Step 2 & 3 are the only ones that may damage your ST PC board. Be sure not to use excessive force while pulling out the capacitors. If you damage your PC board anyway, cure the problem now and not later*).
- Step 4: In this step we will piggyback the new RAM's on top of the old ones. Be sure to connect all pins except pin 4 (RAS) and 15 (CAS). The best way to go about this is to do chip by chip. First, bend the pins of the new RAM's such that they are perpendicular to the package (instead of having slightly spread "cowboy legs"). Use pliers to bend pin 4 and 15 such that the legs are 180 degrees from their normal position, so they stick up in the air above the plane of the top surface of the chips. Don't make an absolute sharp 180 degree bend since some manufacturers' pins may snap off. Leave a little curve in the leg, but insure that is above the plane of the top surface of the chip.
- Using #22 AWG to #16 AWG tinned solid copper wire you will form three buses along the top surface of the new DRAMs. Cut a #22 AWG solid copper wire the length of the 16 DRAMs on the PCB. The RAS bus is formed by soldering all the pin 4's of the new DRAMs to the solid copper wire. The bus wire must be seated against the top surface of the new DRAMs without a gap. This insures clearance between the top shield and the pins of the DRAMs.
- After soldering all 16 DRAMs to the bus clip off any portion of the pins that extend above the top of the bus wire. Now cut a #22 AWG solid copper wire the length of the 16 DRAMs. Place the bus wire along the top surface of the new DRAMs in contact with all the pin 15's of the new DRAMs. Solder every pin 15 to this bus and as above insure that the wire is seated solidly against the top surface of the new DRAMs. Cut off all excess pin length sticking up above the top of the bus wire.
- Using diagonal cutters remove the section of the bus connecting the new U30 pin 15 to the new U32 pin 15. This divides the bus in half with the new U16, 17, 18, 24, 28, 29 having a common pin 15. The new U32, 33, 34, 38, 42, 43, 44, 45 now have a common pin 15, separated from the other common bus.
- (*NOTE: until step 6 is finished, do not in any way apply power to your ST. This intermediate state of affairs will damage your memory chips!!*)
- Step 5: Remount all the desoldered capacitors. Bend the pins like they were before resoldering, such that they will not touch the lower shielding. Solder from the non component side.

Step 6: Orient the 520ST PCB so that you are looking at the solder side of the PCB (non-component side), with the row of DRAMs nearest you. Find the double square pattern of pads at the 68-pin socket of the memory controller, U15 (3H2119).

The following is a guide to locating the six memory controller pins necessary to complete the wiring. The socket is numbered counter-clockwise, starting with pin 1, the square pad (look closely) in the middle of the bottom outside row.

(NOTE: In the following, the symbol ↺ means to make a 90-degree turn counter-clockwise, i.e. around the corner.)

The sequence, moving counter-clockwise from pin 1, first on the outside square ONLY:

1, 3, 5, 7, 9 ↺ 10, 12, 14, 16, 18, 20, 22, 24, 26 ↺ 27, 29, 31, 33, 35, 37, 39, 41, 43
↺ 44, 46, 48, 50, 52, 54, 56, 58, 60 ↺ 61, 63, 65, 67

The sequence, moving counter-clockwise along the inside square only, and starting with the left side of the bottom row:

62, 64, 66, 68, 2, 4, 6, 8 ↺ 11, 13, 15, 17, 19, 21, 23, 25 ↺ 28, 30, 32, 34, 36, 38, 40, 42
↺ 45, 47, 49, 51, 53, 55, 57, 59

Six 68Ω 1/4W ±10% carbon film resistors must be added when adding memory. These series terminating resistors minimize undershoot which may damage BOTH BANKS of DRAMs if omitted. Solder a 68Ω resistor to pin 18 of U15, RAS1.

Solder a #24 AWG stranded wire from the remaining end of the 68Ω resistor to the pin 4 bus (RAS) of all the new DRAMs. that is the new U16, 17, 18, 24, 25, 28, 29, 30, 32, 33, 34, 38, 42, 43, 44, and 45.

Solder a 68Ω resistor to pin 22 of U15, CASH1. Solder a #24 AWG stranded wire from the remaining end of the 68Ω resistor to pin 15 bus (CAS) of the new U45, 44, 43, 42, 38, 34, 33, 32.

Solder a 68Ω resistor to pin 21 of U15, CAS1L. Solder a #24 AWG stranded wire from the remaining end of the 68Ω resistor to pin 15 bus (CAS) of the new U30, 29, 28, 25, 24, 18, 17, 16.

For best results in all three cases above solder the wires coming from the resistors to the middle of the three bus wires in a "T" fashion rather than at one end of the buses.

Use a continuity tester to find the following three traces -- do not depend on visual inspection.

Now install three 68Ω series terminating resistors in the original 512K bank of RAM. Be very careful while soldering to these narrow traces, since excessive heat can easily lift a trace from the board. Use an Exacto knife to gently remove solder mask from traces.

Cut the trace leading from pin 8, RAS0, of U15 near U15. Solder a 68Ω resistor in series with the trace.

Cut the trace leading from pin 6, CAS0H, of U15 near U15. Solder a 68Ω resistor in series with the trace.

Cut the trace leading from pin 7, CAS0L, of U15 near U15. Solder a 68Ω resistor in series with the trace.

Step 7: Sit back. Use Brain. Do you feel confident about the quality of your work? No mistakes? Check everything once again if you are but a little uncertain. Applying power with errors might make your ST into a decorative, nonfunctional piece of art. OK. Either rebuild your ST into its shielding and cabinet, or put it onto a surface clear of wires and solder remains and connect it to monitor, disk and supply.

Boot it. If it boots, you're probably there. Test if the new memory works by looking at the phystop variable (\$42E) with SID if you have the developer stuff. It should read \$100000 (1M hex). Also note that memcntl (\$424) now holds 5 instead of 4, and that v_bas_ad (\$44E) now holds \$F80000 (screen bitmap origin). If you don't have the developer stuff, try a single drive copy and check that you get the whole disk in one buffer instead of two.

If the new memory does not seem to exist, use SID to deposit and retrieve words on locations \$80000 and up (1/2 Meg hex). If bit errors occur, the ST boot ROM did not detect the extension (it checks all bits of 512 locations by testing a pseudo random sequence, before accepting a memory bank). Try to pin point the faulty chip(s) and remove the error.

If it doesn't boot, you're in trouble. I'm sorry. It is difficult to give hints on what to do here. So many possibilities. Desoldering the new chips probably won't work (if the old ones were functional, the ST would still boot).

Check for hidden short-circuit on the RAM pins. May also be that you have a flaky new pin connection.

That's all there is...

A 2.5MByte RAM Expansion for the Atari ST

By Christopher Hicks (cmh@uk.ac.cam.eng)

Version 2 - 31st March 1993

This version copyright March 1993, Christopher Hicks.
Non-profit distribution permitted and encouraged.

Introduction

This document details the addition of an extra 2MBytes of RAM to a standard 520/1040 STf/STfm. The motherboard I upgraded is marked C070789-001 REV D (1040 STf), but this procedure should be OK for other revisions, except that the IC numbers may be different.

The existing RAM consisted of thirty-two 256K by 1 DRAMs, mounted under the power supply. This procedure replaces sixteen of these with two 1MByte SIPs, giving a total of 2.5MBytes.

Note that 1MByte SIMMs can be used instead - it's just that I cannot get sockets for them at a reasonable price. The procedure is identical, since the pinouts are compatible.

Disclaimer

Although this procedure worked perfectly for me, I cannot guarantee that anyone else can perform this upgrade without damaging their computer. I therefore disclaim any responsibility for any damage that may occur as a result of attempting this upgrade. It will also void any warranty on your computer.

On a more positive note, there is no reason why someone who is experienced in wielding a soldering iron, and has done some electronic construction/troubleshooting, should not be able to perform this upgrade successfully.

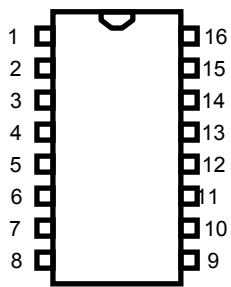
Summary

The ST's memory management unit (MMU) can support two banks of RAM (upper and lower). These may each contain 512KBytes or 2MBytes. In addition the upper bank may be empty. This arrangement gives rise to the following memory configurations:

	Lower Bank	Upper Bank	Total
1	512 kB	---	512 kB
2	512 kB	512 kB	1 MB
3	2 MB	---	2 MB
4	2 MB	512 kB	2.5 MB
5	512 kB	2 MB	2.5 MB
6	2 MB	2 MB	4 MB

This upgrade converts a machine from configuration 1 or 2 to configuration 5. An extra address line is required for the 2 MB bank. This is obtained from the MMU (U56) pin 64.

Device Pin Diagrams



256kbit DRAM



1MByte SIMM/SIP

256kbit DRAM:

Pin	Signal
1	AS
2	D
3	WE
4	RAS
5	A0
6	A2
7	A1
8	Vcc

Pin	Signal
9	GND
10	CAS
11	Q
12	A6
13	A3
14	A4
15	A5
16	A7

1MByte SIMM/SIP:

Pin	Signal
1	Vcc
2	CAS
3	DQ0
4	A0
5	A1
6	DQ1
7	A2
8	A3
9	GND
10	DQ2

Pin	Signal
11	A4
12	A5
13	DQ3
14	A6
15	A7
16	DQ4
17	A8
18	A9
19	NC
20	DQ5

Pin	Signal
21	WE
22	GND
23	DQ6
24	NC
25	DQ7
26	Q8
27	RAS
28	CAS8
29	D8
30	Vcc

NB Pins 26, 28 and 29 refer to the ninth bit required for IBM PC systems. This bit is not required on the ST upgrade.

Pin Functions

Pin Signals	Function
A0 - A9	Address Inputs (row/column multiplexed)
CAS	Column Address Strobe
RAS	Row Address Strobe
D	Data Input
Q	Data Output
DQ0 - DQ7	Data Combined Inputs/Outputs
WE	Write Enable Input
Vcc	Power (+5V)
GND	Ground (0V)
NC	Not Connected

Parts Required

1. Two 1M x 8, or 1M x 9 SIPs, 120ns or faster
2. Two 16 pin and two 14 pin IC sockets
3. A small piece of veroboard (big enough to mount the two SIPs)
4. A few dozen short bits of wire (a shredded scrap of ribbon cable is ideal - lots of colors for easy identification!)
5. Two 0.1uF ceramic capacitors
6. One 33 ohm resistor

Total cost - about 80 pounds including VAT!

Precautions

The motherboard and SIPs may be damaged by static electricity, so don't work on this project wearing synthetic clothes and rubber shoes on a nylon carpet etc... Seriously though, no damage should occur if you are reasonably careful (use a grounded soldering iron, and don't work on a highly waxed or polished surface, for example).

Procedure

1. Remove the motherboard from the computer, and identify the RAM chips (U 3>6, 10>13, 18>25, 27>30, 34>37, 39>42, 44>47 on my 1040STF). On a 520 only half of these will be present, and step 2 may be skipped.
2. Desolder and remove the upper bank of DRAMs. These are scrap, so don't worry about damaging them, but do take care not to damage any tracks on the motherboard. On my 1040 these were the right-hand 16 chips (nearest the FDD), though I know that on some boards (marked REV 4 ?) it is the 16 nearest the rear of the computer. (The way to tell if you are unsure is that all the pin 4s of each bank are connected together, but the two banks are separate.)

When these DRAMs have been removed, the computer should work as a 520ST. If it doesn't, the safest recourse is to buy thirty-two 16-pin IC sockets, and sixteen 256k x 1 bit DRAMs. Remove all the DRAMs from the motherboard, and replace them with sockets. Then fit one of the new DRAMs in each of sixteen sockets, and move them until you find a combination that works. You then have a 512k lower bank, and an empty upper bank as required.

3. Mount the four IC sockets on the veroboard, end to end, and butted up together, so as to make two 30-pin SIP sockets. The connections on these will, from now on, be referred to as SIP1 pins 1 thru 30, and SIP2 pins 1 thru 30.
4. Break the veroboard tracks between the following pins on SIPs 1 and 2: 2,3,6,10,13,16,20,23,25,26. This operation separates the data and CAS lines on the two SIPs.
5. Connect together the following veroboard tracks: 28, 29, 30. This disables bit 8 if present, leaving bits 0>7 for the ST.
- 5a. Connect one of the 0.1uF capacitors between veroboard tracks 1 and 9, and the other between tracks 22 and 30. These are to decouple the power supply to the expansion board.
6. Decide where the expansion board is to go. Mine is in the space normally occupied by the TV modulator and associated circuitry. Bear in mind that leads to the board should be no more than about 3 inches long to avoid capacitance problems. It cannot be stressed enough how short these leads must be if the upgrade is going to work properly.

7. Back to the motherboard... Using a continuity tester you should find that the CAS lines (pin 15) of the removed DRAMs are connected together in two groups of eight. Take a short wire from one of these groups of pin 15s to SIP1 pin 2, and another from the other group of pin 15s to SIP2 pin 2.

8. Take wires from the pin 2s of the first group of eight DRAMs to pins 3, 6, 10, 13, 16, 20, 23, 25 on SIP1. Similarly connect the pin 2s of the second group of DRAMs to pins 3, 6, 10, 13, 16, 20, 23, 25 on SIP2.

The databus is now connected up, and the complicated bit is over. Note that it is essential that CAS (pin 15) and D (pin 2) of each DRAM are taken to the same SIP.

9. Now wire A0 > A8, WE, Vcc, GND and RAS from the SIP board to the similarly named pin of any of the vacated DRAM slots.

10. Lastly wire pin 18 of the SIP board to pin 64 of the MMU (U56 on my board), via the 33 ohm resistor (in series). To aid identification, the part no of my MMU is C025912-38. Pin numbers are marked on the board.

11. Make sure there are no shorts, and check all the connections. Insulate the back of the SIP board (e.g. by taping a piece of card over it) and insert the SIPs. Also ensure that the resistor is well insulated.

12. Reassemble the computer and hey presto - a 2.5Meg ST.

13. Give a copy of this documentation to anyone you please, for them to try the same upgrade.

14. When you run out of memory again, repeat exactly the same procedure, replacing the other sixteen DRAMs, and you will find that you have a 4 Meg ST - what could be simpler?

If you are going to 4 Meg straight away, you can save a lot of work (and stray capacitance) by putting all four SIP's on the same piece of vero-board. This way you only have to connect up the data and address busses once (around 25 connections saved). It is probably best to keep all the other signals separate for the two banks.

If you experience problems with 'snow' on the screen, this may be alleviated by connecting a 68 ohm resistor in series with each of the CAS and RAS lines.

If you make use of this upgrade, please drop me a line at my email address, and I will send you details of any other hardware hacks I succeed with! Good luck...

Christopher Hicks.

A 2.5 Meg Socketed Ram Upgrade for the 1040ST

Shareware File by Barry Orlando. Dec. 28, 1988.

Introduction

This Shareware text file provides a procedure which details a 2.5 megabyte on board RAM upgrade project. The method described replaces 16 256K dynamic RAM chips with 16 one megabit dynamic RAM chips installed in chip sockets. Reinstallation of the ST's metal shield cover is also not impaired.

The beauty of this modification is that pin compatibility is simplified by the use of sockets instead of directly soldering the 1 MEG RAM chips (DRAMs) to the motherboard and by the fact that the removed bank of sixteen perfectly reusable 256K DRAMs can be resold by you to persons upgrading 520STfm computers to 1 megabyte.

If you use this upgrade method, I recommend that you also use the same high quality low contact resistance sockets that I used and have listed below, than to use any other standard sockets. These are the type and quality which are used on expansion boards made for the IBM clones.

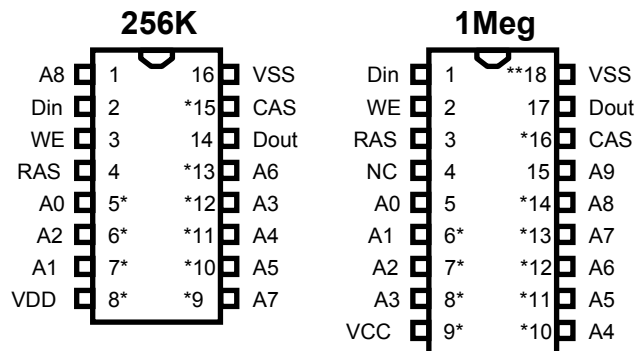
I originally wrote this procedure to aid me in upgrading my own 1040ST's memory to 2.5 megabytes because I didn't trust myself not making mistakes and possibly damaging any of the electronic components, especially the new DRAMs which I didn't consider cheap. I might note that my 1040ST's PC-board was marked Rev 4, but I don't believe that any other recent board revisions (if they exist) for boards with 32 256K DRAMs installed should affect this procedure.

Disclaimer of Liability

I make no claim that this modification will work for you. I can only say that it worked for me. This modification should not be attempted by anyone except by someone with experience repairing or building digital electronic circuits. Performing this modification will be done at your own risk and may void the warranty on your computer.

Background

The 256K dynamic ram chip differs from the 1 MEG dynamic ram chip by having one additional connection, namely the address input A9. The following chip diagram illustrates this and shows pin layouts:



Signal	Purpose
A0 - A9	Address Inputs
CAS	Column Address
Din	Data In
Dout	Data Out
RAS	Row Address Strobe
VDD	Power (+5V)
VSS	Ground (0V)
WE	Read/Write Input
NC	Not Connected
*	PC-board pin compatible
**	PC-board pin compatible on most chips

The 2.5 megabyte modification essentially reconnects all pins as previously connected on the lower of two banks of RAM chips plus adds a new line from the previously unused A9 pin on the ST's MMU (pin 64 of U15) to all new 1 MEG chip pins A9 via a new 33 ohm resistor. Some of the Address leads are interchanged, but this has no effect on the operation of the computer.

Major Tools

1. Desoldering iron (I recommend the spring loaded type that incorporates the head source.)
2. 15W grounded soldering iron (Radio Shack 64-2051 or similar)
3. Wire wrap clip and stripping tool
4. Exacto knife
5. 3-5 power magnifying glass or jewelers eyepiece.

Parts Required

1. One 33 ohm resistor, 1/4 watt, 5%.
2. 5 feet length of 30 gauge wire, Kynar wire wrap solid conductor (Radio Shack 278-502).
3. 16 double contact low profile dual-in-line I.C. sockets (D.C. Electronics catalog no. T02-18, cost: \$0.12 each).
4. Pack of 10 "Socket Wrap ID" (D.C. Electronics catalog no. 16-ID, cost: \$2.05).
You can get in touch with D.C. Electronics at the following location:

D.C. Electronics
P.O. Box 3203
Scottsdale, AZ 85257
(800) 423-0070

5. 16 one megabit dynamic RAM chips, 120ns (I used Hitachi Part no. HM511000-12, and paid \$24.00 each plus state tax and \$4.00 shipping from Ed at Computronix on Dec 16, 1988). You can get in touch with Ed at the following location:

Computronix
2151 A O'Toole Ave.
San Jose, CA 95131
(408) 943-0771

Work Area

Prepare your work area by placing a towel on your flat working surface and cover it with a layer of aluminum wrapping foil. This provides padding for the components and a conductor for static electricity.

Precautions

Since the slightest static discharge could easily damage components on your computers PC-board, always keep one hand on the metal foil bonded along the outer rim of the PC-board before and during touching any part of the PC-board or its components. Avoid working on this project when the humidity is low.

Procedure

1. After removing the PC-board from the insides of your computer, desolder and remove the front 16 256K chips (U33 through U45 and U16 through U30).
2. Using the Exacto knife, cut the following traces on the top side of the PC-board:
 - a. Between U30's pin 1 and U61's pin 1.
 - b. Between pin 14's of U33 through U45 and U16 through U30 and other pin 14's of U46 through U61.
 - c. Between the thru-hole (nearest to pin 1 in center of U32) of U32 and U46's pin 1. (Note: This thru hole traces to U32's pin 1 on the underside of the board.)
3. Using the Exacto knife, cut the following traces on the bottom side of the PC-board:
 - a. Between U30's pin 1 and the 32 ohm resistor R76.
 - b. Between U32's pins 2 and 14. Do the same for U33 through U45 and U16 through U30.
 - c. Between U30's pin 4 and U15's pin 8. (Note: Cut the trace after it goes through the board to the bottom side near resistor R91, but before it reappears on the top of the board via another thru-hole.
 - d. Between U32's pin 14 and the thru-hole located between U32 and U33.
4. Preparing New Sockets and Soldering Them In Place
 - a. On all 16 new sockets, remove the internals of pin 4.
 - b. On all 16 new sockets, bend all pins 1 and 17 flat against their bottom then up against their outer side.
 - c. On 14 of the new sockets, bend pin 18's flat against their bottom and pointing straight out to the side.
 - d. On the remaining two, bend both pin 18's flat against their bottom but back and down so it will just reach and fit into the PC-board hole of it's adjacent pin (pin 17). These two sockets will be installed and soldered with the socket's pin 18 going into the old 256K chip's pin 16 hole below the new socket's pin 17.
Trial fit these two sockets at this time.
 - e. Using a pair of scissors, cut pieces of "Socket Wrap-ID" plastic markers for sandwiching them between the new sockets and the PC-board during installation. These little plastic markers should be cut so as to electrically isolate the socket's pins 1 and 17 from leads and traces on the PC-board, and from pin 18 of the socket for the two separately prepared sockets above, but also so as not to interfere with seating the socket's four pads flush with the top of the PC-board upon installation.
 - f. Solder the two sockets with the bent back pin 18 and markers into U30 and U33.
 - g. Solder the remaining sockets with markers into U34 through U45 and U16 through U29.
Pin 18 of these socket's must be soldered to the top of the board at the nearest thru-hole that the pin 18 just barely reaches. Use plenty of heat as these thru-holes are big heat sinks.
5. Mounting The New 33 Ohm Resistor
 - a. Install a new 33 ohm resistor on the PC-board by drilling two small holes from the top at points near capacitor C55 and resistor R70. Make sure you don't drill through a trace on the other side by holding the PC-board up to a bright light and marking the spots with a pencil.

6. Wiring

Wire the following connections on the top side of the board:

- a. Between U32's new socket's pins 1 and 17 and a thru-hole on the PC-board between U32 and U33. This thru hole is the only one that has a trace that run to another thru-hole between U46 and U47 on the top side of the board.
- b. Between U16's new socket's pins 1 and 17, and pin 2 of U54. Solder the wire directly to the side of the chips pin 2.
- c. Between each of the remaining new socket's pin 1 and 17 and a thru-hole in back of each 256K RAM chip (i.e., U47 through U53 and U55 through U61). These thru-holes are the ones that you will find traces back to either pin 2 or pin 14 of U47 through U61 and then trace over toward U22, U23, U26, and U27. Actually, pins 2 and 14 are tied together on the bottom of the board for every 256K RAM chip still installed. So that's 14 more wires altogether (i.e., each new RAM chip's pins 1 and 17 will be connected to the pins 2 and 14 of the 256K chip directly in back of it.

Wire the following connections on the bottom side of the board

(Note: all pin numbers here refer to the pin numbers on the PC-board, not the pin numbers of the new sockets or new ram chips):

- d. Between U15's pin 64 and one side of the new 33 ohm resistor.
 - e. Between all pin 14's of U32 through U45 and U16 through U30, and the other side of the new 33 ohm resistor.
 - f. Between all pin 2's of U32 through U45 and U16 through U30, and the thru-hole near R91 which had its trace cut to U30's pin 4. This thru-hole traces back on top of the PC-board under the bank of resistors and over to U15.
 - g. Between U61's pin 1, U30's pin 4, and the cut trace side of resistor R76.
 - h. Between U30's pin 1 and U61's pin 3.
8. Inspect all solder joints with the magnifier for solder shorts and faulty connections. Verify that all connections were made correct.
9. Install the new 1 MEG RAM chips into their new sockets.
10. That's it! Clean isn't it? Now sit back and admire your work.
12. Reassemble the computer and reconnect the monitor and power cord and let it whirl!

Uhm, Barry, what happened to point 11?

A 3 Meg ST Upgrade

By Jeroen Scheerder (jeroen@phil.ruu.nl)

My ST definitely has 3.0 Mb. I agree, not long ago I thought this impossible too; however, in the magazine "ST Computer" 3/1990 (a German magazine), the author K. D. Litteck describes how it is done in an article called "Aus zweieinhalb mach drei" (an error correction appeared in ST Computer 4/1990). This magazine is published in Germany by Heim Fachverlag. It is in German, and my German is not too good, but I'll try to describe the article accurately.

How does it work? Well after determining memory configuration (see below) TOS starts clearing the amount of memory it thinks is available. Say you've got 3.0 Mb available (like I do); TOS at first thinks 2 banks of 2 Mb are available and sets out to clear 4 Mbytes. It then runs into nonexistent memory when it reaches the end of the three available megabytes. An exception of type buserror occurs, and TOS determines at which address the error occurred. This address is then written in the TOS variable 'phystop', thus setting the correct memory size. Now TOS boots on ...

This works in all TOS versions known to mankind. This is part 2 of an article about ST memory upgrading (technical part).

Before getting into the gruesome details, first some general ST memory theory: ST is equipped with Dynamic RAM, and for every memory location a single transistor is used. This allows for high speed and compact chips. However, the chips need to be refreshed about every 2 milliseconds, of which the MMU (among other things) takes care.

The DRAMs are accessed bitwise; therefore, the ST addressing in 16 bits, only banks of 16 chips can be used. Now the addressing signal is multiplexed (at one address line two addresses are located, that are consecutively transferred). In DRAM these addresses are then separated. This keeps the number of pins as small as possible. The multiplexing and sending control signals is catered for by the MMU too.

Now the multiplexing is the reason the DRAMs can only be increased in 4-folds; only 16 kBit, 64 kBit, 256 kBit, 1 MBit and 4 MBit chips currently exist.

Two control lines (called RAS and CAS) are of interest (both low-active). RAS transfers the low address of the multiplexed address line, CAS the high address. They work too as chip-selectors. The signal WE selects READ or WRITE mode. Furthermore, the ST divides CAS into CASL and CASH. CASL connects to the chips on address lines 0..7 and CASH to 8..15, so that memory can be byte-accessed too. On WORD-read or write both CASL and CASH are active.

The ST's MMU can manage two banks of memory. For bank 0 the control lines RAS0, CAS0L and CAS0H are used, bank 1 uses RAS1, CAS1L and CAS1H. These banks can be equipped with 64, 256 or 1024 kBit chips. On coldstart TOS writes patterns in memory and reads them back (all at well-chosen locations) to determine the memory configuration, and stores in the system variable memconf (address \$424) and the MMU's configuration register (\$FF8001) a configuration byte:

Bank 0	Bank 1	Value	Total Memory
512k	---	\$04	0.5M
512k	512k	\$05	1.0M
2.0M	---	\$08	2.0M
2.0M	512k	\$09	2.5M
512k	2.0M	\$06	2.5M
2.0M	2.0M	\$0A	4.0M

(I hope this is right - anyway, it's in all good ST literature)

On top of all this, MMU also manages to make memory contiguous: although the address space of each bank is 2.0 Mb, in case your bank 0 is 512 k and 1 is 2.0 M the MMU concatenates the total addressing space to a contiguous one (as TOS expects).

Now, that's all very well: now we want to fool the MMU to handle up to 5 (!) memory banks.

A scan of the electronics schematic in MacPaint format (you need it!) is posted too (binhexed and stuffed with Stuffit). This schematic is corrected for the error in the original printing.

A clarified copy of this diagram has been drawn into the next page ...

We need address line A9 (not used on 256 kBit chips). It is multiplexed and contains the real addresses A18 and A19. Both NAND-gates serve as OR-gate and should release the D-Latch 1 and 2 when CAS1L or CAS1H become active.

An IC 74LS375 demultiplexes and gets out of A9 the addresses A18 and A19.

With the D-Latch 3 A18 is obtained, with D-Latch 1 A19. A18 is also led over D-latch 2, being kept until CAS goes high. The IC 74LS139 (2-bit binary decoder) produces with A18, A19, CAS1L and CAS1H eight new CAS signals, of which we only need four. These four signals are fed to the memory banks (all formerly 0).

Now, some things to pay attention to:

- the 2.0 Mb should be bank 0;
- the RAS0-line must be interrupted;
- the assignment of RAS- and CAS-lines MUST be corresponding.

MMU pin configuration:

- 6: CAS0H
- 7: CAS0L
- 8: RAS0
- 18: RAS1
- 21: CAS1L
- 22: CAS1H
- 64: A9

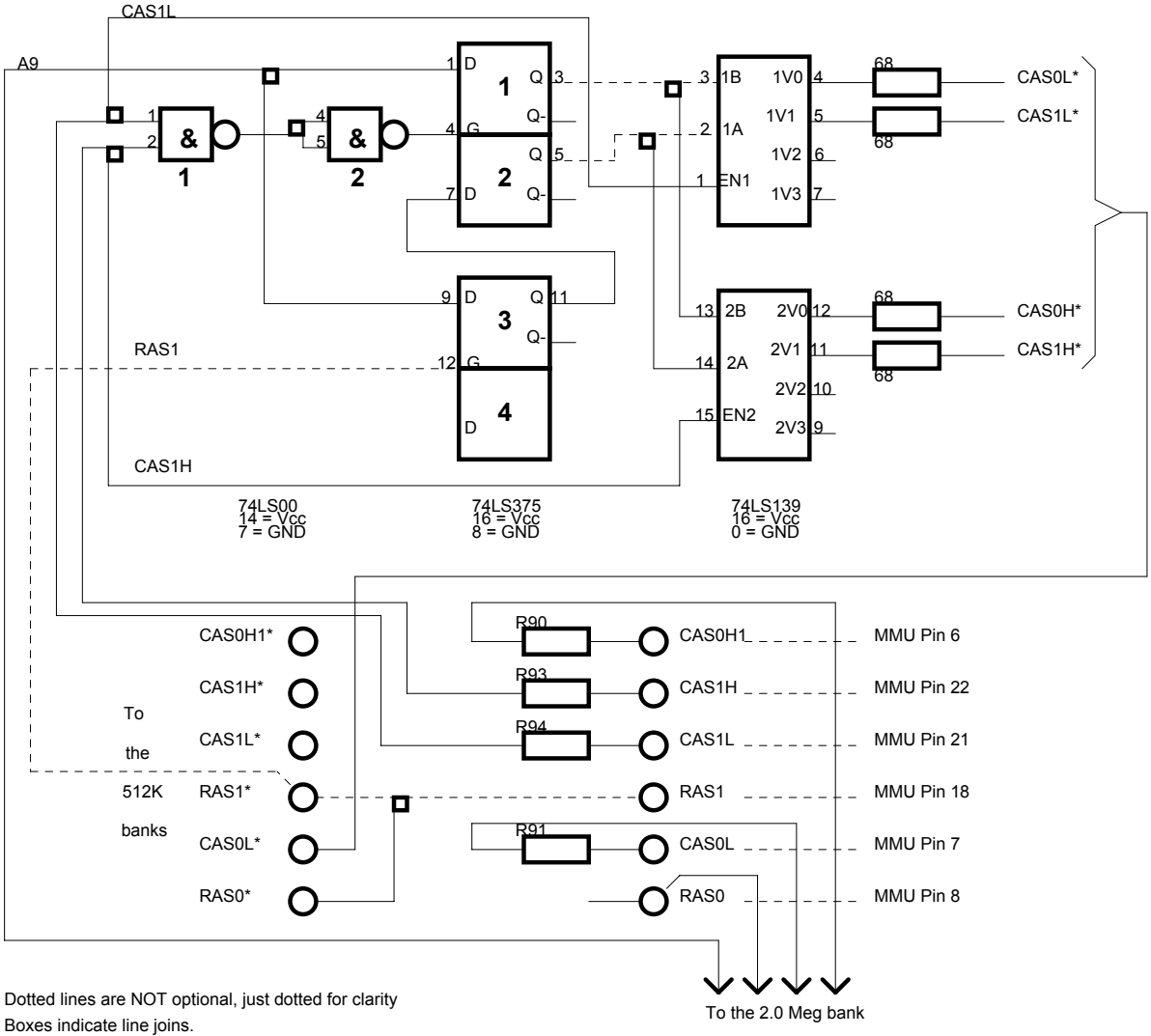
Components:

- 1 IC 74LS00
- 1 IC 74LS375
- 1 IC 74LS139
- 4 68 Ohm resistors
- 1 Block condensator
- 1 microFahrad wire or copper litze

I hope this is clear enough to you all (once you've seen the picture accompanying this, I hope).

Wish y'all good luck,

Jeroen.



Dotted lines are NOT optional, just dotted for clarity
 Boxes indicate line joins.

Introduction and Notes for adding High density disk drives to the Atari ST

The first thing you must do, is to verify that your existing HDD controller IC is a suitable one for the modifications. The controller on the Atari ST is a Western Digital WD1772 and carries a suffix (usually) of 00-02 or 02-02. As it is explained in the following texts, the original chip is designed to run at 10MHz and only the 02-02 chip will handle the higher clock speed of 16MHz, so ensure that you have a compatible HD controller first.

There are no major problems with running two disk drives either one of each or two high density drives but a little jiggery pokery is required to get it all rolling properly. The type of HD drive you select will depend upon you but some are easier to work with than others - it is essential that you get a drive that will handle the full Shugart specs and not one of the (increasingly common) PC type HD drives that has limited functions. The HD drive MUST have the ability to send out the HD select signal on it's interface connector. The PC does not use this normally as the HD select is controlled by the PC host card and tells the drive what kind of disk is inserted (this is the reason that the PC can mess up a HD disk by formatting the media with the wrong data rate).

For full compatibility you'll need to gate the HD select signal with the drive select lines so that you can mix both high density and normal density drives on the same bus.

Some of the later series of HD drives have a pull-up on the HD select line that will stop you accessing the normal 720k drive as a second unit. This can be overcome by swapping jumpers (assuming you get a drive that can be configured properly) or by changing the pull-up to a higher value and arranging gates so that it is driven and held low for non high density disk access.

You may also experience timing problems with some obscure makes of drive, so be careful to make sure you know what the drive unit you are buying can/can't do.

Finally, if you boot from the A: drive and you make this your high density one then be aware that you will need a small "autoboot" program to set the step rate BEFORE you can boot from a high density disk. This program MUST reside on the first track of the disk or the boot will fail for reasons explained below

How it works

Basically the 720k and 1.44 MEG disks drives are compatible in as much as that they both have 2 sides and 80 tracks. Now the important difference is the speed at which they write to the media. With 720k disks, the data is written out by the drive at 250k bits/second - this is known as "double density" and uses MFM encoding. A high density drive uses exactly twice this speed at 500kb/s and hence, writes twice as much data to the media. This is laid down by the format program as 18 sectors (instead of 9 on a 720k) per track of 512 bytes per sector. This faster rate is still MFM encoded and is known as "high density".

There is yet a faster rate again at 1Mb/s that is used on the IBM machines and creates disks of 2.88 MEG capacity but a new controller IC would be needed for this (anyway, the disks are too expensive yet) and it's known as "quad density" and there is an older standard known as "single density" which was half the data rate of the original 720k we discussed at 125Kb/s (remember the old CPM machines and things).

OK, you've swallowed all that (what you haven't....) now consider that since the physical interface is the same and only the data rate changes (a high density drive unit switches it's data rate up to 500kb/s when you plug in a disk with an extra hole) between 720k and 1.44 MEG disks, it becomes easy to see that if we send the data at the same speed as the high density standard, then we can talk to high density media. How do we do this - easy, double the clock rate to the disk controller chip. It's normally run at 8MHz, so run it now at 16 MHz and presto... you have your high density interface. You can pick up a 16MHz clock in the Atari from the video shifter.

One unfortunate side effect of doing this, to which I referred earlier was that since the data rate has changed and the speed doubled, then it now takes the HD controller chip half the time to carry out any operation than it did before.

Now this means that effectively, the STEP rate of the disk drive (a function of the controller chip) is now halved from 3ms to 1.5ms (and I haven't yet found a drive that'll go this fast). So the problem when booting from a HD disk, is that you must load a little utility that will change the STEP rate (programmed by the operating system) to 6ms (which when clock doubled becomes 3ms).

Now, this utility has to be small enough to fit on track 0 otherwise the disk drive will try and step out to the second track at 1.5ms (which it won't do) and the drive goes out of sync. Quite a number of programs incorporate the ability to change step rates and a small program can be written to do the job (I have written such a small utility if anyone wants it).

Compatibility

There are one or two commercial 720k disks that complain and refuse to load if tried in a converted machine with a 1.44 MEG as drive A: and if you're a games freak then it is probably better that you make the HD unit, your B: drive to maintain compatibility. Once written and formatted, TOS takes care of how many sectors per track there are and I have found no (serious) problems at all with two HD drives fitted to my Atari Mega 3. One last point to bear in mind is that a lot of (both commercial and PD) utilities were written to allow the use of 9, 10 or 11 sector formats, so these will not work with an 18 sector disk and any cleanup/repair/disk doctors that can't handle HD media, will most probably foul things up for you and make an even worse mess - you have been warned.

Oh yes, forgot to mention you can handle 5.25" HD drives as well but you need to fit an external switch (the 5.25" HD doesn't have an HD select line OUT).

Happy hacking.....

Martin GW6HVA @ GB7OSP

The trick is to double the clock frequency of the controller chip, so it will talk twice as fast to the drive: the bit stream has "high-density" format.

A clock of 16 MHz is well beyond specification (the WD1772 is rated to run at up to 10 MHz), but it will work most of the time.

The hack consists of two parts: hardware and software and works with high density drives only!
Now: the hardware.

The clock pin of the WD1772 is pin 18. It is normally hooked to the 8 MHz clock.

Cut this pin and connect it to the output of a simple circuit. feed the two inputs of this simple circuit with 8 MHz and 16 MHz clocks (16 MHz can be get at SHIFTER pin 39).

The "simple circuit" mentioned above could be a simple, manual switch. Now You are able to select two clock frequencies and therefore two "modes" for the controller chip: double density and high density.

The simple circuit could also be a multiplexed (74LS157, I think) controlled by the "I detected a HD-disk in me"-output of the floppy drive (the TEAC FD 235 HFN provides this output on pin 2 if a jumper is set; the 235 HFN has a second jumper that enables internal detection of HD disks: this must be set, too).

With the MUX, a disk inserted will select the appropriate mode of operation automatically.

Do not worry about two drives in different modes: when copying from one drive to the other, the clock rate will be switched once to twice a second. This will not trouble the controller (remember the clock to be in the range of Megahertz...).

If you want to connect one HD drive and one DD drive, you'll need two more TTL gates (one 74LS00 will do) to make sure, the controller is in DD mode for the DD only drive: invert the HD-detected-output (it is LOW for DD before, HIGH for DD behind the inverter) and feed it into a NAND together with drive select for the DD drive. Let a HIGH output of the NAND select DD-clock (8 MHz). Now "not-HD-detected OR DD-drive-select" will select DD mode, right?

With two HD drives, there is no problem at all :-)

Now for the software:

The controller uses its clock as base for all timings.

The pulse width and time between pulses for the step line are controller controlled.

To make it short: with the doubled clock the controller will try to make the head step twice as fast: he will send a step pulse every 1.5 milliseconds if set to 3 milliseconds (which is TOS standard).

Just set the step rate to 6 milliseconds to make the controller step 3 milliseconds in HD mode (well, the 6 milliseconds steps in DD mode will slow down floppy operation a bit, but not that much. Not stepping, reading and writing takes the time!).

How to set the step rate?

Well, there is a documented TOS variable for the step rate at 0x440 called seekrate.

The step rate in this variable will be used by the BIOS after a call of the routine, pointed to by "hdv_init" (the hdv_init routine seems to initialize the BIOS parameter blocks for the floppies). The pointer to hdv_init can be found at 0x46A (this is a supported TOS variable, too).

Look at the MWC-Assembler source below.

The second thing is to get a formatter that will format 18 sectors per track. HYPERFORMAT by Claus Brodt is such a thing, but a simple formatter hacked from the example in the Mark Williams C Lexicon did the same. It's included below.

For help try to write to hase@hase_1.mbx.sub.org, but it'll probably bounce; our mail service is fading fast... I'll try to reply to anything, I finally get.

hase

Atari ST High Density Disk Drives - Part 2

```
/* formath.c Formatter fuer High Density Disketten */
/* nur fuer angepasste Hardware! Floppycontroller und Laufwerk */
/* muessen High-Density tauglich sein! */
/* Hartmut Semken, Lupsteiner Weg 67 1000 Berlin 37 */
/* hase@hase_1.mbx.sub.org or hase@netmbx.mbx.sub.org */
/* 03-SEP-89 */
/* Modified by Karl, 30/12/1994. */
#include <stdio.h>
#include <osbind.h>
#include <time.h>

#define SLEEPTIME 1          /* 1: Zeitschleife, 0: Taste */
#define BLANK (0xE5E5)
#define MAGIC (0x87654321L)
#define BUFSIZE (20*1024) /* Platz fuer mehr als 18 Sektoren... */

#define DEVICE 0            /* 0 = Floppy A, 1 = Floppy B */
#define SIDES 2             /* je */
#define SECTORS 18         /* nach */
#define TRACKS 80          /*Belieben */

#define TOTSEC (TRACKS * SIDES * SECTORS)

extern int errno;

void sleep( int seconds )
{
    #if SLEEPTIME
    {
        clock_t t;
        for(t = clock();clock() < (t + CLK_TCK*seconds);)
    }
    #else
    {
        printf("Taste druecken\n");
        fflush(stdout);
        Crawcin();
    }
    #endif
}

void main()
{
    int track;
    int side;
    int status;
    short *bf;
    register char reply;
    short *middle;
    char buffer[512];

    printf("\033E\n");
    printf("Public Domain High Density Mini Formatter\n");
    printf("von H. Semken\nDer Autor garantiert in keiner Weise fr die
    Funktion\nDieses Programmes.\nBenutzung auf eigene Gefahr.\n");
    printf("\n\n");
    printf("\007\033pFormatiere Diskette in Drive %c\033q\n%d Seiten\n%d Sektoren
    pro Spur\n%d Spuren\n\n", (65+DEVICE), SIDES, SECTORS, TRACKS);
    printf("Wirklich formatieren? ");
    fflush(stdout);
    if ((reply = Crawcin()) != 'y' && reply != 'Y' && reply != 'j' && reply != 'J')
    {
        printf("Nein. Floppy nicht formatiert.\n");
        sleep(1);
        Pterm0();
    }
    printf("Ja.\n");
    printf("Diskette einlegen; Taste drcken...");
    fflush(stdout);
    Crawcin();
}
```

```

printf("\n");
bf = malloc(BUFSIZE);
for (track = TRACKS-1; track >= 0; track--)
{
    for (side = 0; side < SIDES; side++)
    {
        printf("Formatiere Spur %d, Seite %d", track, side);
        fflush(stdout);
        status = Flopfmt(bf, 0L, DEVICE, SECTORS, track, side, 1, MAGIC, BLANK);
        if (status)
        {
            middle = bf;
            printf("\t%d\n", status);
            while (*middle)
            {
                printf("\tDefekter Sektor %d\n", *middle++);
            }
        }
        else
        {
            printf("\tokay\r");
        }
    }
}
printf("\n\nAlle Spuren formatiert\n");
printf("Initialisiere Directory\n");
for (track = 0; track < (BUFSIZE>>1); bf[track++] = 0);
for (track = 0; track < 2; track++)
{
    for (side = 0; side < SIDES; side++)
    {
        if (status = Flopwr(bf, 0L, DEVICE, 1, track, side, SECTORS))
        {
            errno = -status;
            perror("Write Error");
        }
    }
}
Protobt(buffer, (long)Random(), 3, 0);
/* Prototyp Bootsektor fr * 80 * 2 * 9 Sektoren */
/* Prototyp Bootsektor fr das neue Format anpassen */
/* Bytes 19 und 20 enthalten die Sektoren pro Disk */

/* unteres Byte von TOTSEC */
buffer[19] = (char)((TOTSEC>>8)<<8)~TOTSEC);
/* oberes Byte von TOTSEC; es lebe das Intel int-Format */
buffer[20] = (char)(TOTSEC>>8);
buffer[24] = (char)SECTORS; /* Sektoren pro Spur */
status = Flopwr(buffer, 0L, DEVICE, 1, 0, 0, 1);
if (status)
{
    errno = -status;
    perror("Write Error (Bootsektor)");
}
status = Flopver(buffer, 0L, DEVICE, 1, 0, 0, 1);
if (status)
{
    errno = -status;
    perror("Verify Error (Bootsektor)");
}
printf("Diskette in Laufwerk %c formatiert\n", (65+DEVICE));
sleep(1);
Pterm0();
}
/* End main() */

```

TOS 2.06 Installation Instructions for 1040ST (V2.0)

Introduction:

This document contains a procedure for installing TOS 2.06 EPROMs into a 1040ST revision C motherboard. A simple circuit consisting of 3 TTL ICs wire and patience is all that is required. The total cost of the upgrade will be the cost of the TOS 2.06 EPROMs (\$60) plus \$1 for the TTL chips.

The OS upgrade should work on other STs as well but this upgrade has only been implemented on a revision C motherboard of a 1040ST (~1986 vintage).

This upgrade requires some careful soldering so don't try it unless you have had some PC board soldering experience (don't use a soldering gun!).

I would like to thank all who responded to my questions on notes for their help. Without the help, I would have had a slim chance of getting this thing to work.

I would like to suggest that anyone who wants TOS 2.06 to buy it from Codehead Technologies: (213)-386-5735 They send along a nice manual and a great utilities disk.

Background:

Most 520ST and all 1040ST computers were shipped with 6 ROMs that contained TOS (The Operating System). These ROMs are mapped into the address space 0xFC0000 to 0xFEFFFF. This is a 192K byte space. Each ROM holds 32K bytes. TOS 2.06 is larger than 192K for extra features and bug fixes so the address space had to be moved. The address space 0xFF0000 to 0xFFFFF is used by I/O ports on the ST so the existing space could not just be increased.

The new address space for TOS 2.06 is 0xE00000 to 0xE3FFFF. I might add that previous versions of TOS may have been here also but I went from TOS 1.0 to TOS 2.06 so I can't say for sure. Also, instead of putting TOS 2.06 into 8 32K byte ROMs it was placed into two 128K byte EPROMs. One part is called EVEN (supplies the high 8-bits) the other part is called ODD (supplies the low 8-bits).

When the 68000 comes out of the reset state, it reads the first 8 bytes in its address space (0,1,2,3,4,5,6,7). From this data it gets an initial value for the supervisor stack pointer and the initial value for the program counter. What the 520ST and 1040ST do (via the GLUE chip) is to remap addresses 0 through 7 to 0xFC0000 to 0xFC0007. The ROMs are programmed to have the start address in these locations. TOS 1.0 had the value 0xFC0020 while TOS 2.06 has the address 0xE00030.

In 520ST and 1040ST computers, the address space 0xE00000 to 0xEFFFFFF is not used or decoded. If a memory cycle is attempted in this range the data bus will time out and an bus error will result.

The requirements to get TOS 2.06 to work in a 1040ST are as follows:

1. Decode the address range 0xE00000 to 0xE3FFFF and have the TOS 2.06 ROMs drive their data addressed.
2. Generate DTACK for accesses to TOS 2.06 space.
3. Map the address range 0->7 to 0xE00000->0xE00007.
4. Mechanically connect the EPROMs and decoder logic to the motherboard.

The address range can be decoded with the high 4-bits of the address bus (A23->A20) to decode the address range 0xE00000 to 0xEFFFFFF. This will work fine because nothing else is in the range 0xE40000 to 0xEFFFFFF.

The DTACK signal in the ST is an open collector (wire OR) type bus. This means that any device that wants to assert DTACK may only do so with an open collector type interface. This interface basically is one in which the driver can either drive the line to GROUND or not drive the line at all. This is different than normal TTL which will drive outputs to GROUND or +5 (in reality 0.4 to 3.8V). The DTACK line has a pull up resistor on it which will bring the line high if none of the devices on the bus are driving it low. The DTACK signal can be derived from the decoder logic but must be buffered with an open collector buffer.

The GLUE chip decodes the 0->7 address range and the range 0xFC0000 to 0xFCFFFF and asserts the signal ROM2 when either of these two ranges are addressed. The interface can use this fact and OR in the ROM2 signal with the 0xE???? space to enable TOS 2.06 ROMs. This way the boot up sequence will be as follows:

1. 68K will read addresses 0->7 which will cause the ROM2 signal from GLUE to be asserted.
2. Data will be read from locations 0xE00000 through 0xE00007.
3. The 68K will then jump to address 0xE00030 and begin executing code.
4. At this point the ROM2 signal will never be asserted again and the TOS 2.06 decoder logic will do the work.

The TOS 2.06 EPROMs are a super set of the TOS 1.0 parts. This means that nearly all of the pins of the new parts can be plugged into the sockets of the old parts. All of the signals needed for the decoder are available off of the GLUE chip. Therefore the 3 TTL chips will be placed near the GLUE chip by glueing them to the motherboard with their pins bent up.

Before you attempt this upgrade make sure that you have enough room between the two ROM sockets to place the 32-pin parts into the sockets. On the Rev C mother board there is no problem. The TOS 2.06 chips will have pins 1,2,31 and 32 hanging out over the end of the socket. Basically if there is more than a quarter inch between U4 and U7 you are set.

Step by Step Instructions:

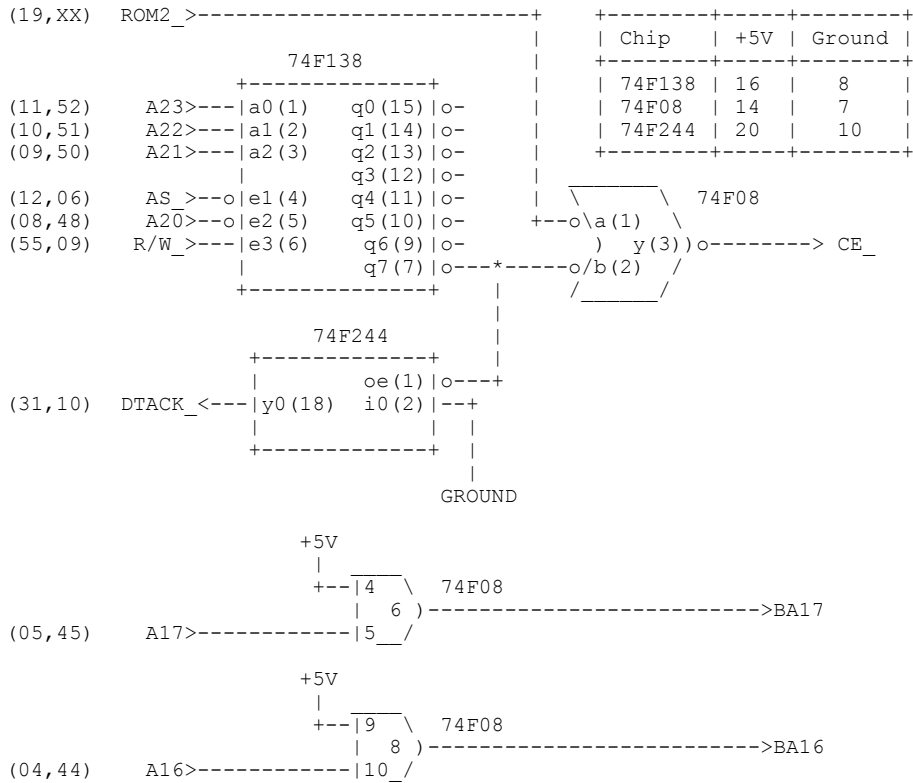
Please read through the remaining instructions before starting to determine if this upgrade is something you want to attempt. There are two main steps. The first step is the construction and testing of the address space decoder and DTACK logic. The second step is the insertion of the new ROMs into the system. Before you start you will need a disk with a program on it to read address 0xE00000. This program will be used to determine if the decoder and DTACK generator are working. Before the circuit is implemented the program will generate a bus error and display bombs on the screen. After the circuit is implemented the program should just read the value 0xFFFF if you do a 16-bit read.

```
main()
{
    unsigned int *ptr=(unsigned int *)0xE00000;

    printf("Value read was 0x%04x\n",*ptr);
}
```

Step 1 (Implement the Address decoder and DTACK Generator).

The schematic below (I hate ASCII schematics more than you do) shows the address decoder and DTACK logic. The circuit has 10 signals that connect directly up to the GLUE chip and 3 signals that connect directly to the TOS 2.06 ROMs. The GLUE signals are on the left and the TOS 2.06 EPROM signals are on the right.

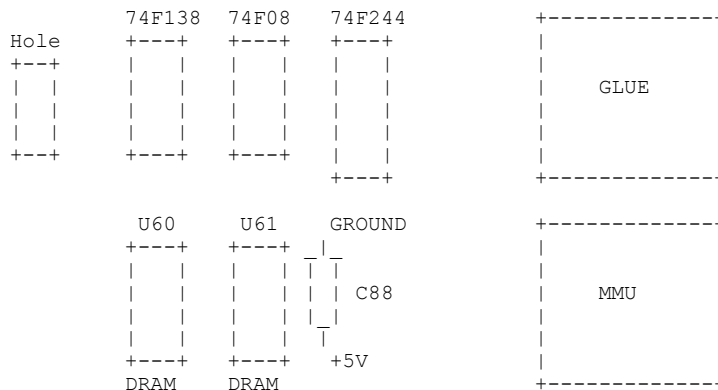


Numbers in parenthesis on the left are (GLUE pin #, 68000 pin #). The ROM2 signal is not connected to the 68K but can also be found on pin 20 of U4 and U7 (TOS 1.0 EPROMs). Parts needed are one of each of the following, 74F138, 74F08 and 74F244. Different families (LS, ALS, AS) may be substituted for the above parts.

The following is a description of how the circuit was constructed. The main deal is that the current TOS ROMs must be left in the ST. Don't take them out until this circuit has been verified.

Start out by removing the motherboard from the case and RF shield. Carefully remove the disk drive and power supply. Save the piece of black plastic for insulation to lay the PC board on while working on it and also when debugging.

Locate the area of the board shown below near the GLUE and MMU chips. The three TTL chips will need to be glued in the areas shown in the diagram below.

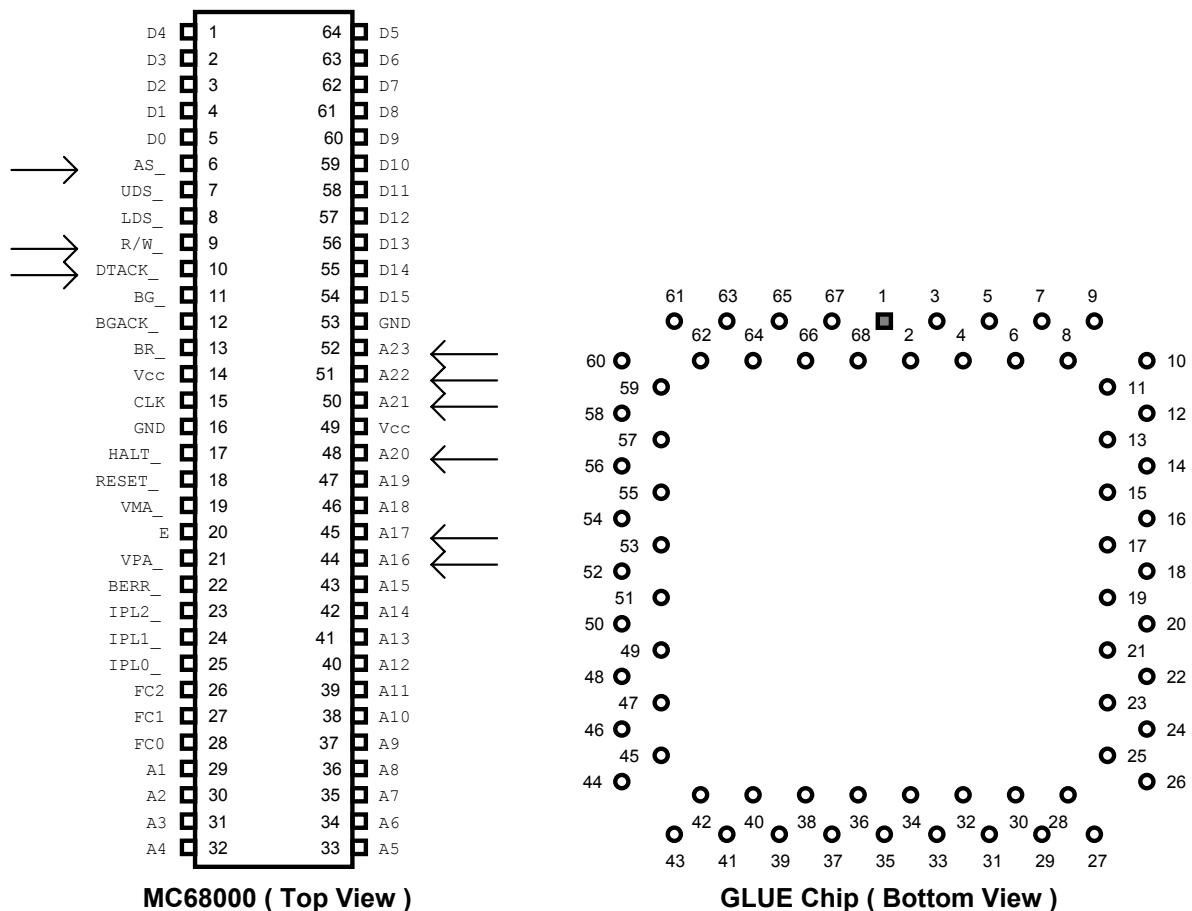


To make life a bit easier bend all the pins of the three chips up to change them from the pins going down to the pins going up. This way pin 1 will be the top left one. I used a rubber cement type glue to glue the chips to the PC board. Try to find spots without the silver pop throughs so that the parts are rock solid. Super glue should also work. Also, put the chips close to the hole since this is where the wires to connect to the GLUE chip will go. Keep in mind that a plastic shaft about half as wide as the hole will still need to go through it. You can always cut it off as a last resort.

Let the glue dry before soldering to the chips. After the glue dries, make sure that your computer still works. This can be done by placing the mother board on the black plastic and connecting up the power supply and disk drive. Just place a the disk in the drive with the small program to read 0xE0000. Connect up the monitor and blow off the keyboard for now. Switch on the power (be careful when you plug in the power cord not to get zapped). You machine should come up. If you are careful you can plug in the keyboard while the power is off but there is really no place to physically support it.

Now that you are convinced that you machine still works it is time to do some wiring. Use wire wrap wire for all of the connections. I believe that you can still get it at radio shack. First, connect up power to the TTL parts. As shown in the previous diagram +5 and GROUND can be obtained off of the capacitor C88. Next, connect up the logic in the circuit diagram. An Ohm meter should be used to verify your work.

The final step of the decoder circuit construction is to connect up the 10 wires to the GLUE chip. Remember that all but one of the 10 connections to the GLUE chip has an electrical connection to the 68000. The connections to GLUE will be made on the bottom of the board. Don't solder directly to the GLUE chip. Run the wires from the GLUE chip on the bottom of the board through the hole marked in the above diagram and then to the 3 TTL chips. Keep the wires as short as possible and keep them next to the PC board. The following two diagrams show the pin outs of the GLUE and the 68000 respectively. Check connections with an Ohm meter on the top of the board from the 68K to the 3 TTL chips. The ROM2 signal can be checked from pin 20 of the TOS ROM U4. Also, note that pin 1 of the glue chip has a square pad instead of a round one. There is a dot on top of the chip.

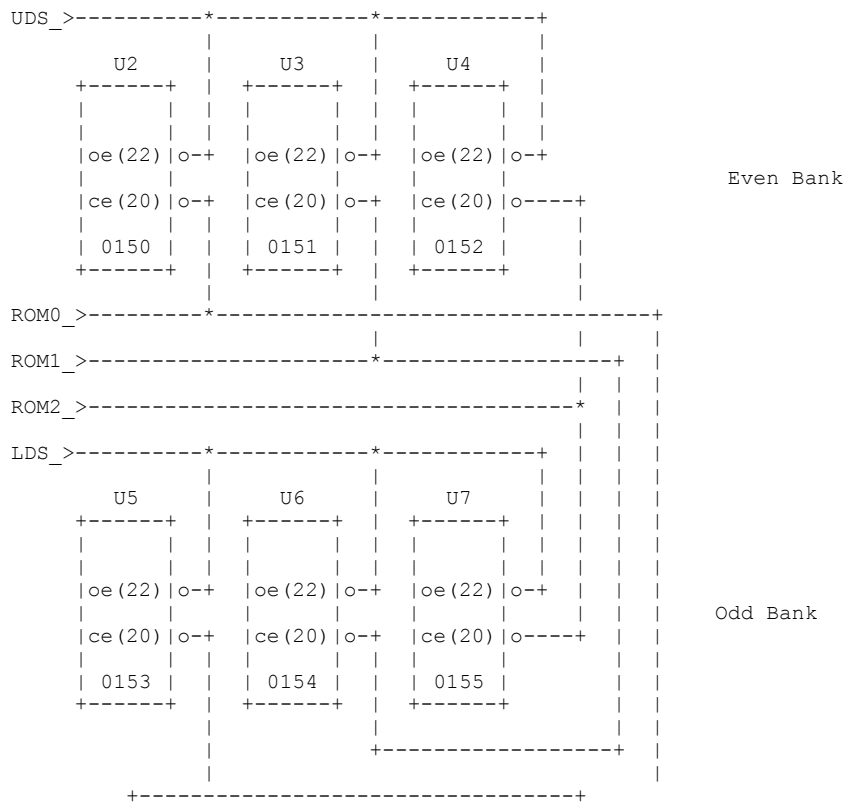


Now that the connections have been made double check your work and make sure that no pins of GLUE or the decoder are shorted. After you are sure connect up the power supply, disk drive, keyboard/mouse and monitor and power up you ST. If your machine does not come up turn off the power and check your work again. If all fails remove a wire or two until your machine boots. The DTACK_ line is the first one to remove when in doubt. If your machine did come up you are in pretty good shape. Try running the program described before that will do a read cycle to 0xE0000. If this program prints any result without a bomb on the screen you are just about done with the upgrade.

Turn off the power and get ready for the final step.

Step 2 (Yank out the old ROMs and bring in the new)

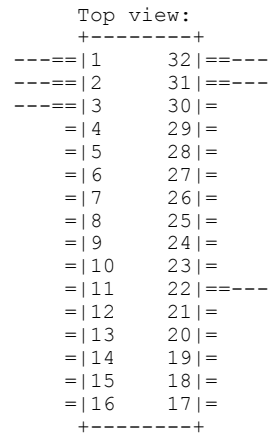
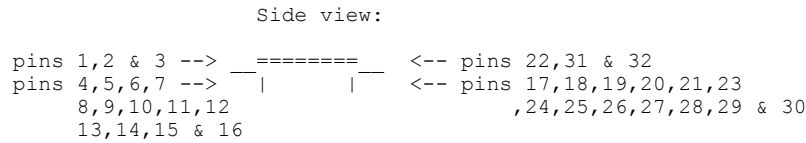
Now comes the part that you have been waiting for. The diagram below shows the location and diagram of the original TOS in the ST. Note that U4 is connected to bits D15 through D8 and U7 is connected to bits D7 through D0. Believe it or not the ROM2 signal really does select the low address ROM set. Basically what is going to be done in this step is to remove all six of the TOS ROMs and insert the EVEN TOS 2.06 ROM into the socket for U4 with the chip enable pin lifted. Likewise, the ODD TOS 2.06 ROM will go into the socket for U7.



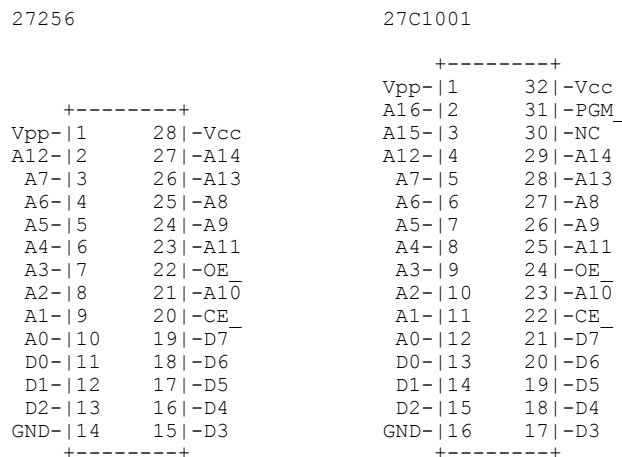
The 4 digit code at the bottom of each ROM is the last 4 digits of the part number. For example the complete part number of U7 is RP23256 0155. The ROMs are of the type 27256.

Carefully pull each of the six ROMs out using a flat tipped screw driver. Save the ROMs in case you need to go back to TOS 1.0. Make note of the position and part number of each chip.

Bend pins 1,2,3,22,31 and 32 of both TOS 2.06 EPROMs up 90 degrees so that the pins lie in the same plane as the ceramic package:



Place the EVEN TOS 2.06 EPROM into the socket for U4 so that pin 16 goes into the hole for pin 14 of the original TOS ROM. See the diagram below of the old (27256) vs. new (27C1001) chips.



The parts should not touch each other and the lifted pins should not be above the tops of the chips. Put the power supply back over the EPROMs (Rev C) board and make sure that the bottom plate does not touch the parts. It will look close but if you can see day light you are OK.

Connect lifted even EPROM pins 1,31 & 32 to even EPROM pin 30 (+5V).
Be careful not to drip solder down into the socket so that the part can be removed if desired.

Connect lifted odd EPROM pins 1,31 & 32 to odd EPROM pin 30 (+5V).

The remaining three connections are made from the EPROMs to the TTL decoder. Make the wires as short as possible and keep them next to the PC board. Thread the wires through the bypass caps to help keep the wires under control.

Connect even EPROM lifted pin 22 to odd EPROM lifted pin 22 then connect to pin 3 of the 74F08 (CE_). This is the new equivalent ROM2 signal.

Connect even EPROM lifted pin 3 to odd EPROM lifted pin 3 then connect to pin 8 of the 74F08 (BA16). This is a buffered version of address line A16.

Connect even EPROM lifted pin 2 to odd EPROM lifted pin 2 then connect to pin 6 of the 74F08 (BA17). This is a buffered version of address line A17.

Check your work with an Ohm meter and your eyes. Connect up the power supply and all the other junk and give it a try. If everything is happy you will see Atari's Logo on your screen when you turn on the power. The machine will sit for a bit and then start doing a RAM test. After the RAM test, the machine will place a bar on the screen and shrink it until it is gone then your machine will fire up the desktop.

You can bypass the memory test or the bar at any time by hitting the space key. If your machine works, great job. Put it back together and enjoy.

Complaints:

If you have complaints, questions or whatever, drop me some email:

rossin@fc.hp.com

Enjoy TOS 2.06

Ted Rossin.

Revision History:

Rev 2.0:

Thanks to great feedback from Arthur K Whitney the following changes have been made.

- a. The GLUE DTACK_ pin number has been changed from 50 to 31. Lets just say that I blew it when I drew the original schematic in my notebook.
- b. The circuit is drawn using a 74F244 instead of the 74F06 and the pull up resistor. The 74F06 part is a bit hard to find.

Rev 1.0:

Initial Revision.

Here is some help for all you guys that want a decent drive to put into your ATARI; either to replace a broken drive or as an after market item to replace your current ATARI drive.

You can put a nice bullet-proof TEAC drive in you computer even cheaper than you or your dealer can buy an ATARI drive. Your cost will be only \$57.00 give or take a buck or two plus the shipping if you mail order.

You can order these little beauties from several outlets listed in Computer Shopper or Byte or about any IBM oriented magazine. Shop the adds a little you should not have to pay more the \$55 to \$60. Ask specifically for a TEAC 720K floppy drive (model FD235F if you really want to impress the salesman).

When you receive the drive all you have to do is pull the RDY jumper from the drive and move the drive select jumper from D1 to D0 (FIGURE 1). Then you turn the drive over, heat up the old soldering iron and lift the inner tab on the high density select sensor box in the front left corner (check out the IMG diagrams with this file on where to find these).

Now pop this drive into your 1040 or MEGA and fly. On a 1040 you will have to flip the ribbon header over when you plug it to the drive. If you do plug in the drive and it doesn't work don't blow your cool. Just flip the cable over and try it again. It will go.

The only thing you have to be concerned about is that you do not cross plug the power connector. The TEAC drive only requires 5 volts to run. Not an entire hydro generating facility like the ATARI drives. If you want to be really safe clip the 12 volt line on the ATARI drive plug. That's the BLUE wire (you may want to check this with a meter to make sure... some computers may have a different color code... it is the outboard pin, farthest left looking at the back end of the drive) going to your drive that will keep the volt killers away.

You will have to either put spacers under the drive in the older 1040's to bring these beauties up to height or do what I do. Just remove some of the plastic stock from the lower portion of the case. Cut a rectangle out with the bottom edge at about the bottom of where the old drive button was.

A little side note... these drives will step out to 83 tracks with no problem. And if you have a hard drive that you boot from then you know how annoying it is when the drive A light stays on all the time. The TEAC drive activity light turns off on its own.

There you have it. Have fun. Course if you screw up doing this I will disavow any knowledge of your actions Mr. Phelps. That's my disclaimer line. You should feel quite comfortable though. I have installed these drives in ALL of my machines and sold them to a bunch of others and have yet to have one fail (OK, I had one factory defect) but the rest are running like champs.

If you want to do this to your machine or have any questions feel free to call and I'll give you all the technical and moral support you need.

These babies work just as good as "B" drives too. The drive jumper stays in the same place "D0". The secret is in getting the wiring right and following all the pin-outs properly (FIGURE 2).

I have done a couple of the ATARI case conversions and I wish I could say just plug them in or give you an easy conversion diagram but... time for sarcasm... ATARI is consistently inconsistent with their SF354 and SF315 drive layouts.

What I do is to go to my VCR shelf and rob one of the plastic cases from my least favorite tape (never from my porn collection, that is sacred) Then I cut a window out of one end with my trusty knife (dremel tools work, but they tend to melt plastic at high RPM) the size of the front of the drive... you know so the button, LED and all that is visible and easily operated (FIGURE 3). Cut out those little round things that go in the tape wheels too. Make a paper template of the bottom of your drive to help you guess where the holes should be drilled in the VCR case to bolt it down (I never seem to get these right on the first try... good thing VCR cases are cheap).

Next I dig out one of my 6 ft. ATARI drive cables and whack that thing in half. When you buy your drive you will get an adapter for IBM edge connectors with it. Solder the appropriate wires to the appropriate connections on the edge card. Use that diagram in my file for the appropriate connections (FIGURE 2). An OHM meter helps this along.

You should also receive a power plug that converts the big IBM plug to a little plug for your drive. Clip off the big plug. Now go find a wasted ATARI mouse and cut the cord off of it. If you don't have one then find a female 9 pin D-connector, about 2 ft. of stereo wire and wire pin 7 to the 5 volt (red) and pin 8 to the ground (black) of your IBM micro power plug (FIGURE 4). (You may have a joystick cord that might work but check it first. Most joysticks do not have pin 7 connected since most do not require 5 volts to run.)

Now cut two slots in the back of the VCR case. One just large enough for the drive cord and one just large enough for the power cord... oh, I forgot to tell you that was what you were making in the previous paragraph. Get a bunch of your favorite elastics and wrap them around your power and drive cords at a point just inside the VCR case. This will make a collar so the cords won't jerk out of the case if somebody (rug rat kids) pulls on them.

Now plug the drive plug into your outboard computer plug and the D-connector into your joystick port and test the drive. Hopefully everything works. If it does just squirt a little hot glue on the top of the drive and close the lid. This will hold the lid shut, but it isn't so permanent that you can't get back in if you need to.

This works great for my STacy and it eliminates the extra power pack. If you don't like using up a joystick port I have tapped connectors from the ST power supply direct to a plug which I mount on the back of the 1040 computer... still better than an extra power pack.

I did do one 1040 where I piggy backed two of these drives in. It worked (thank you Lord) but it wasn't as fun as I thought in would be. I recommend it only for sado-masochist.

If you mount one of these in an SF314 or SF354 the next best thing to do is to cut all traces from the in-bound plug on the drive's power/logic board. The TEAC will not use any of Atari's pre-drive circuitry. It will just screw it up. Now jump the appropriate connections to the header plug on the logic board (the header plug should also be isolated from Atari's logic board. Then make sure Atari's power plug is correct. Remember, 12 volts will kill a TEAC.

I wished I could be more specific about the ATARI external conversion, but I've seen so many different boards in them I'd have to write a dozen different schematics. I've done several and they take longer than the first method I told you about. However if you feel uncomfortable about trying one I would be glad to do it for you for \$20.00 including the shipping. Just send the board, not the whole drive.

Hope this helps a little. Have fun with it.

Brad Bradley
Box 20
Manti, Utah 84642
(801)835-8441
Email: BBRADLEY

PS: If you put one of these in and it works, well, if you want to chip \$5.00 for the info I won't be offended. This is for you to enjoy your ATARI as much as possible.

Everything you always wanted to know about desktop.inf, but were afraid to ask. Below is a typical desktop.inf file (with a few changes added for examples). Below each line is a definition of what each value represents.

#a000100

This is the first desk accessory, the rs232 config. Each 0 represents the first column of buttons on the set rs232 config dialog box. The first 0 is column 1 row 1, the second 0 is column 1 row 2, etc. A 1 value indicates that a button in the second column was chosen.

#b000000

This is the set printer config. It works the same way as the rs232 config.

#c7770007000600070055200505552220770557075055507703111103

This is the color palette. The color value is set using 3 digits at a time, representing the red, green and blue values. The 3111103 at the end deals with the keyboard repeat rate, and sensitivity. (None of the above three will do anything if the associated desk accessory is not loaded)

#d

This is apparently reserved for a fourth accessory, and does nothing at this time (so far as I can see).

#E C4 02 (#E=Extras)

This is two things. The first value has to do with both the Set Preference dialog and the mouse double click rate. The byte is broken down as follows with the indicated bit set performing the noted action:

Bit	7	6	5	4	3		2	1	0
Value	128	64	32	16	8		4	2	1
	displays	sorted	sorted	confirm	confirm		double click rate		
	TEXT	by	by	deletes	copies		all three bits, values		
	(if not set	size	date				may only be from 0-4		
	displays	\	/				5-7 will turn off the		
	icons)	both bits					mouse buttons		
		sort by							
		type							

If neither bit 5 or 6 is set the sort is by name

The second value is 03 for hi rez, 02 for medium rez, or 01 for low rez.

```
#W 00 00 02 0B 2A 0B 08 A:\*.*@ (#W=Window)
#W 00 01 0A 01 45 09 08 A:\TEST.C\*.*@
#W 00 00 0E 09 2A 0B 08 A:\*.INF@
#W 00 00 0F 0A 2A 0B 00 @
```

The above four are the window defs. The first number is how far over the horizontal slider is, the second is the vertical slider. The third number is the x coordinate of the left hand side of the window (this takes on even values, with odd values the same as next lowest even value). The fourth number is the y coordinate (this takes single increments). The fifth and sixth numbers are the width and height, respectively. The last number indicates where on the screen the window will open from. (The window opens with a different shape and from a different place with each number, but when you close the window, it will go to another place on the screen. I have not hacked at this long enuf to figure it out.) A zero or FF will not open the window. The text indicates which drive's contents will be displayed. If the drive does not exist, the window won't open, ie. drive bits not set.

Also, if the display validation is omitted, the window won't open. The second def above will display the contents of the folder TEST.C, while The third def above will open a window, and only display the .INF files on drive A. If you close and open the window, the files will display as defined in the file and program defs below. (NOTE: this only applies to icon images, every file will display in text -- except if the file bits are marked to be hidden, system, volume, read/write and whatever other bits there are, in which case it won't display at all, but if it is read only, it will display -- strange HUH?) The bottom-most open window in the list will be the active window.

```
#M 00 02 00 FF D FLOPPY DISK@ @      (#M=iMage (?))
#M 00 00 00 FF A DRIVE A@ @
#M 00 01 00 FF B FLOPPY DISK@ @
```

These describe the icon attributes. The first two numbers are the column and row position of the icon. The column can be from 0-7, the row 0-3. The third number determines the icon image which will be displayed. The image number is the same for this and the remaining defs in desktop.inf, as follows:

```
0= disk drive (drawer)          1= folder (sub-directory)      2= trash can
3= executable file (.PRG, .TOS, or .TTP)  4= text (stack of papers)
```

The fourth value doesn't seem to do anything, but must be a place holder for an unimplemented function. The single letter is the drive identifier, and the text is the drive name. The first @ indicates the end of the drive name. The second @ does nothing, but we can speculate as described below for the file identifiers. The order in the list determines the visual heirarchy of the icons, ie. which will display on top when moved over another icon.

```
#T 00 03 02 FF BLACK HOLE@ @      (#T=Trash)
```

This is the same as the disk drive. If you move a disk drive identifier below this in the list, it will display on top of the trash if moved to the same location. The trash has no identifier letter, but you can put one in.

```
#F 03 04 @ *.INF@ @      (#F=Files)
#D FF 01 @ *.C@ @      (#D=Directories)
```

These two determine which type of file or directory will be displayed, when displayed as icons. The first line will make GEM display only .INF files for use with the SHOW, PRINT, CANCEL alert box. If you delete this line, no icons will be shown for any file, except as defined below for programs. The second line does the same for sub-directories, only .C folder icons will show. When files are displayed as text, all files will be there, but if you single click on an 'undefined' file type, the system will reboot. If you double click, the name will be highlighted, but you can't do anything.

```
#G 03 FF *.APP@ @      (#G=Gem)
#G 03 FF *.PRG@ @
#F 02 04 *.TOS@ @      (#F=File)
#P 03 04 *.TTP@ @      (#P=Parameters)
```

The above four determine the types of files defined as executable images. Notice that the #F, #D, #G and #P all have two @ symbols. This is defined as: the text before the first @ tells GEM this is a def for an executable file, while text after the first @ and before the second @ tells GEM that this is a def for a SHOWable file (ASCII or object). Note that if both are

specified, like #G 03 04 1ST_WORD.PRG@ *.TXT@ then when you double click on any .TXT file, the 1ST_WORD.PRG file will run, and the .TXT file will be taken as a parameter.

The first number indicates the icon image for the file before the first @, and the second number indicates the icon image for the file between the @'s.

Note that the disk drive def also has two @, which would indicate that the FF is a def for the text in between the @'s, but I couldn't get anything to happen.

Try playing around with your desktop.inf by placing different values for the icon images (can be kinda fun). Be forewarned, tho, you can thoroughly confuse GEM, although no harm will result.

This concludes the desktop.inf tutorial (for disk based desktop.inf, there are a couple of desktop.inf's in memory which can do things, too).

There may be more to it, but I haven't figured it out yet.

The few lines of desktop.inf before and including some application installation information are :

```
#F FF 04 @ *.*@
#D FF 01 @ *.*@
#G 03 FF *.APP@ @
#G 03 FF *.PRG@ @
#F 03 04 *.TOS@ @
#P 03 04 *.TTP@ @
#P 03 04 d:\FASTBAS.PRG@ *.BSC@
```

The action of the desktop is clearly to try matching a 'double-clicked' file against these templates, IN REVERSE ORDER.

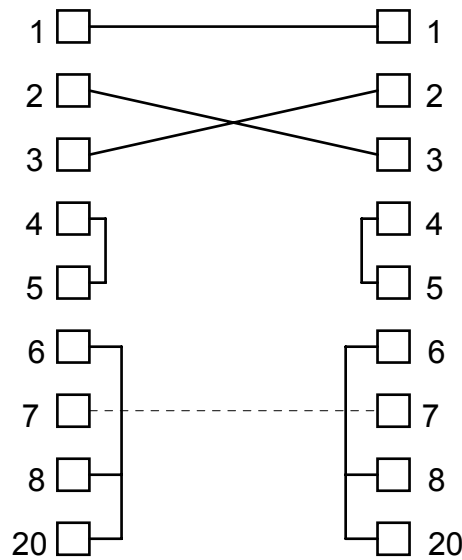
Once a match has been made, the desktop then takes either the specified action, or if none, the default action.

[Thus if the bottom template is changed to d:\DUMP.TTP@ *.*@ then an attempt will be made to dump ALL files. Interestingly, not only will .prg (etc) files not be executed, they will also not be displayed as executables on the desktop.]

If no other match is found, the top line wakes up the SHOW|PRINT|CANCEL alert box from the standard desktop. The simple solution is just to insert the name of one's favourite display routine, as below.

```
#F FF 04 d:\LESS.TTP@ *.*@
#D FF 01 @ *.*@
#G 03 FF *.APP@ @
#G 03 FF *.PRG@ @
#F 03 04 *.TOS@ @
#P 03 04 *.TTP@ @
```

Null Modem

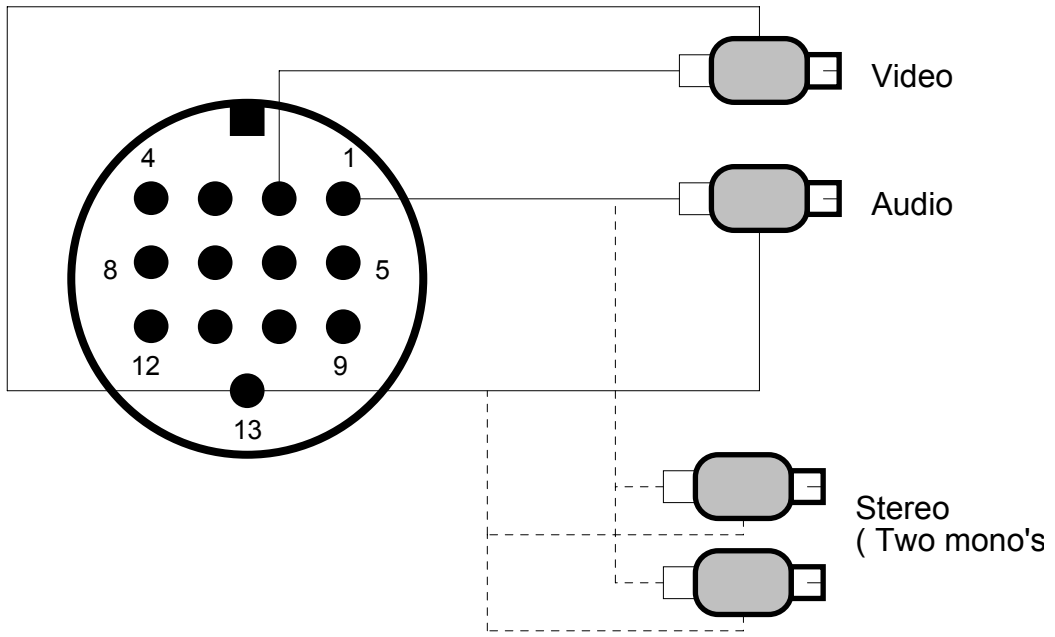


MSX1 Emulator for the Atari-ST

a MSX1 emulator for the Atari-ST is available from
`nic.funet.fi` or `ftp.funet.fi`
path: `pub/msx/emulators/`

Faulty DMA Chip in Early STe's.

With the early models of the STe, there was a problem with the DMA controller. The controller worked fine unless you had a Hard Disk connected to the STe. The MegaFile series were reported to work fine, but with the majority of Hard Disks there were severe problems. This was a stuff-up on Atari's part, and the faulty DMA chips were replaced by Atari free of charge. The STe's with the dodgy DMA controller had the serial number C025913-38. This faulty DMA chip was replaced by the DMA chip C398739-001A. One note - the DMA chips were not socketed originally, so it is a good idea to socket the chip for easy future access. It is also a good idea to use AHDI with a version of greater than 3.0, as this version of AHDI had problems also! (At the time of writing, the latest stable version of AHDI I used - with my Falcon - was 6.061)



If you can't understand this diagram, then you too don't understand STDoormat either ...

Stereo ST!?

Some time ago those twats in ST User printed the following article in their readers pages'. Now what the hell they think they are trying to do with directions like this I dunno, but I guess ST Doormat still has the crown for biggest bunch of wankers ...

If you want to read a ST mag, read ST Review - they at least know some of what's going on! Hi Ofir!

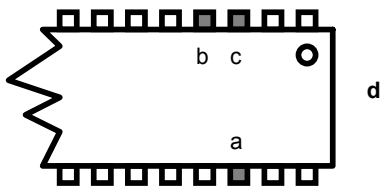
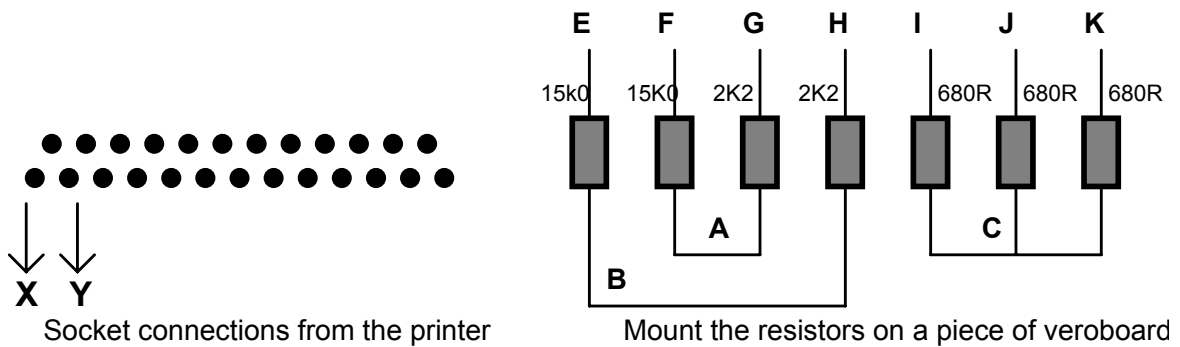
Let there be Stereo..

Regarding your answer to Eli Geller's as to whether his computer has stereo capability, well, it does in fact! The chip inside the ST, the famous Yamaha sound chip, is in fact, a stereo chip. Atari in its infinite wisdom however, decided to wire the circuit for Mono output.

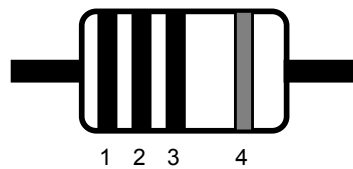
Here's a diagram, showing you how to get stereo output from your humble ST. **(What a load of bollocks, shite, whatever!! K.)** Of course, the circuit uses the printer port, but why anyone would write a program which prints and plays music in stereo at the same time, is anyone's guess!

Also of note is that the sound is fine when created using the chip, but if your program or game uses sampled sounds, there will be a little distortion if played through the chip's circuit. This is nothing to worry about, just use the monitor or TV's speaker instead.

A.J. Lee, Basingstoke, Hants.



Connections to the Yamaha sound chip (note the orientation notch)



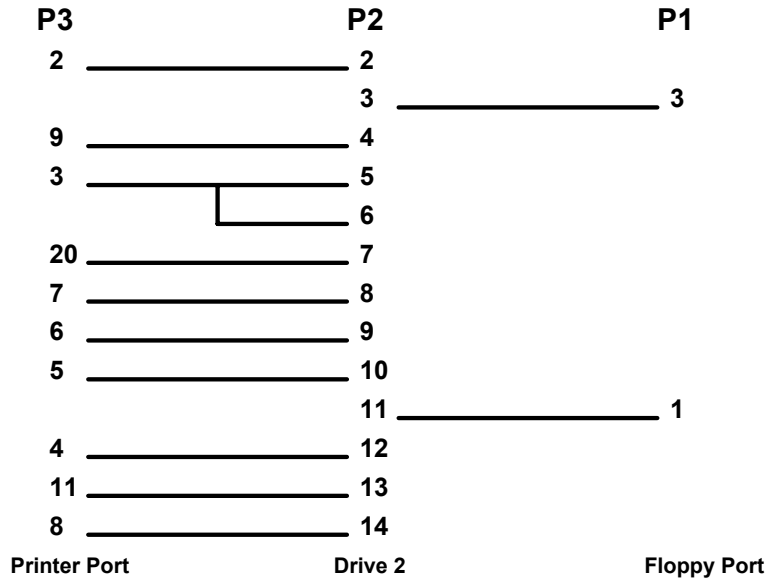
Resistors are color-coded to determine their value

Are these guys serious or what!? Ancient Hieroglyphs make far more sense

Blitz Copier Cable

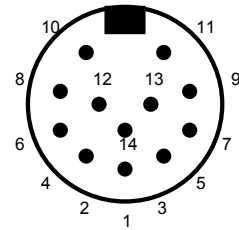
One of the neatest things about the Atari is the number of bastard copy protection schemes about. The Happy Discovery Cartridge will copy anything, but a much faster, and PD hardware hack is the Blitz Copier. This cable, when used in conjunction with the appropriate software, will copy almost anything, and faster than GEMDOS does a normal disk copy!

Here's the schematics on how to build yourself one:



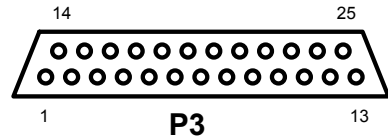
P1 & P2 are 14 Pin DIN Plugs (MALE)
P3 is a 25 PIN D Connector (MALE)

Shown with the plug in your hand, looking at the pins.



P1 & P2

Shown looking at the back (Solder Points)



P3

You should be able to get the software to drive this baby from many places; several FTP sites have it.

Internal MegaSTe HD Controller

Inf. from alexpp@rbg.informatik.th-darmstadt.de, who obtained it from Stephan_Kilian@k2.maus.de

5V	30		29	GND
5V	28		27	GND
5V	26		25	GND
Not Connected	24		23	HDINT
A0	22		21	RW
HDACK	20		19	HDCS
	18		17	Reset
	16	---	15	D7
	14	---	13	D6
	12	---	11	D5
	10	---	9	D4
	8	---	7	D3
	6	---	5	D2
	4	---	3	D1
	2	---	1	D0

In MegaSTe's without an internal HD, the lines with --- between them are connected via jumpers. This is jumper J402 if I remember correctly.

The lines that are grayed are connected to the DMA chip. I think that these are connected to the same lines on the DMA as the jumpers show. It is also possible to connect the MegaSTe internal HD controller to the ST's DMA port, as follows:

ST	MegaSTe	Pin		Pin	MegaSTe	ST
	5V	30		29	GND	
	5V	28		27	GND	
	5V	26		25	GND	
	Not Connected	24		23	HDINT	10
19	HDRQ	22		21	RW	19
16	A0	20		19	HDCS	9
14	HDACK	18		17	Reset	12
		16	---	15	D7	8
		14	---	13	D6	7
		12	---	11	D5	6
		10	---	9	D4	5
		8	---	7	D3	4
		6	---	5	D2	3
		4	---	3	D1	2
		2	---	1	D0	1

J404 I think is the Floppy Controller. I don't know about these connections, but here goes:

Pin	Signal	Signal	Pin
2	FDDS	GND	1
4		GND	3
6		GND	5
8	Index	GND	7
10	D0SEL	GND	9
12	D1SEL	GND	11
14		GND	13
16	MON	GND	15
18	D0IN	GND	17
20	STEP	GND	19
22	WDATA	GND	21
24	WGATE	GND	23
26	TR0	GND	25
28	WP	GND	27
30	RDATA	GND	29
32	S0SEL	GND	31
34		GND	33

Thanks to Mickey Boyd (boyd@math.fsu.edu) for that one.

WD1772 Floppy Disk Controller Pinouts:

Article 19155 of comp.sys.atari.st.tech:
From: Lonnie McClure <lmccclure@delphi.com>

Well, it's not much, but here's some data I had at hand on the WD1772 pinouts:

Pin	Signal
1	CS
3	A0
5	DAL 0
7	DAL 2
9	DAL 4
11	DAL 6
13	MR (Reset)
15	INTR
17	DD
19	Index
21	WD
23	MO
25	CLK
27	Step

Pin	Signal
2	R/W
4	A1
6	DAL 1
8	DAL 3
10	DAL 5
12	DAL 7
14	Ground
16	DRQ
18	WP
20	Track0
22	WG
24	RD
26	DIRC
28	Vcc

Article 16996 of comp.sys.atari.st.tech:
 Newsgroups: comp.sys.atari.st.tech
 From: ktessner@isle.waterloo-rdp.on.ca (Kevin Tessner)
 Subject: Re: ST Mouse --> PC

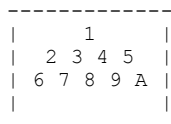
The signals on a bus mouse (Microsoft Inport) are fully compatible with the ST's mouse port. They can be used vice versa, but the pins must be adapted:

-----8<-----

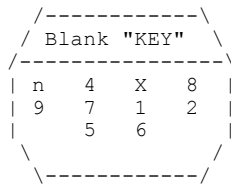
Atari ST	Microsoft Mouse	ST expected signal
1	5	XB
2	3	XA
3	8	YA
4	9	YB
5	N/C	N/C
6	2	Left button
7	N/C	+5VDC (N/C)
8*	6 and also tie to pin 1	Ground
9	7	Right button

* = I am not sure why I have pins 6 & 1 tied together, but it does work.

The Microsoft mouse connector looks like this. This is the solder side, with the leads pointing up.



If you can not use the connector, the pins are numbered as follows. This is the connector side from the Microsoft Mouse plug:



This is a view at the front of the connector. It is where the plug from the mouse would be. The pin labeled 'n' is not connected. The place labeled 'X' is filled, and does not have a pin.

-----8<-----

Source unknown, but it does work just fine. The view above is the view into the female connector, on the PC mouse's end of it is mirrored.

Article 17173 of comp.sys.atari.st.tech:
From: benkeen@aol.com (BenKeen)
Newsgroups: comp.sys.atari.st.tech
Subject: Re: Getting inside the Falcon ???

I've gotten into about 15 Falcons so far.. Anyhow, he're's how I did it:

First take out the 7 gold screws on the underside of the case. There will be one under a tamperproof seal. Do not take out any silver screws; they hold the floppy mount and the motherboard.

Next, what I do is put tape over the <countersunk> holes that the screws are in (after I have thoroughly loosened them) so I don't have to worry about loosing any when I take the top off.

Get the top of the case off and disconnect the keyboard. It is connected on the right. The bundle of wires may be glued to the case; just cut through the band of glue with scissors.

Once you've got the keyboard off, take out the dozen or so sheet metal screws and untwist the metal tabs that hold the main piece of the RF shield down.

Don't forget the three on the back of the power supply near the switch. Disconnect the speaker connector and slide the shield off.

Unscrew the three screws and unbend the tab holding down the shield over the floppy drive.

Take out the remaining screws that hold down the rest of the RF shield. The socket for the co-processor will be underneath the keyboard, obscured by the last piece of the RF shield. Normally you don't have to take this off to install a HD or a card or memory.

To get back to normal, reverse the steps. Just make sure that when you reconnect the keyboard that you line it up with the bottom of the wires that the connector fits with. There are more wires coming off the board that go into the connector.

Feel free to e-mail me for clarification.

Benjamin Keen<benkeen@aol.com>

Probleme mit dem Sound-Subsystem des Atari F030

Autor: Robert Schaffner

Robert Schaffner
Am Sch„ferpfad 10
63303 Dreieich

EMail: Robert Schaffner @ OF2
(MausNet)

Tel/Fax/Modem: 06103/67197

F030 - Platinenrevision B/C

Zu Bemerkungen w„re gleich am Anfang das mir dieser Fehler bis jetzt erst an zwei Falcons der Revision B und C aufgefallen ist. Diese Textdatei kann im allgemeinen nicht auf jeden F030 angewendet werden. Zumal es schon alleine nur im Soundsubsystemeingang schon zwei verschiedene Sound_In Beschaltungen und auch zwei verschiedene Schaltpl„ne gibt.

In jeden Fall sollte man bei Knack- und Kratzger„uschen w„hren dem Sampeln, dem Abspielen einer CD ect. zuerst sein Augenmerk auf die Klinkenbuchse werfen. Ein leichtes Nachbiegen der Kontaktfedern kann hier schon wahre Wunder ausl„sen.

Bei den beiden Eingangs erw„hnten F030-Computern hatten die Soundst„rungen jedoch zwei ganz unterschiedliche Ursachen, die am Anfang der Fehlersuche die gleichen Symptome aufzeigten.

Bemerkenswert erscheint mir die Tatsache das selbst ein Versuchsweises „ffnen eines F030 die Garantie erl„schen l„sst, also erst nachdenken ob man wegen diesem Fehler die Garantie verfallen l„sst oder eben seinen H„ndler daf„r konsultiert.

Nach dem man im Falcon die oberen Abschirmbleche entfernt hat kann man bereits alle Komponenten eines F030 in Augenschein nehmen.

Zum besseren Handling ist es aber sicherer wenn man die Bleche komplett entfernt. Dazu mu„ leider auch das Diskettenlaufwerk gel„st werden. Hier ist das Flachbandkabel recht vorsichtig zu bewegen!

Auch ist es gut, wenn man bei dieser Aktion den Lautsprecher vorsichtig absteckt, er hat recht kurze Anschlusskabel und ist jetzt nur hinderlich.

Zu ersten Messungen und Tests gen„gt es allerdings wenn das obere Abschirmblech und das Floppylaufwerk entfernt wird. Das Floppylaufwerk besitzt auf der Unterseite des Rechners drei lange Schrauben, die vierte Befestigung ist ein Abstandsbolzen aus Messing der gleichzeitig zur Zentrierung und zur „bertragung der Rechnermasse zum Laufwerk fungiert.

Ebenso so mu„ bei einem kompletten Ausbau der Hauptplatine das Schaltnetzteil und die interne 2.5" IDE-Festplatte ausgebaut werden. Man wird nicht extra Betonen m„ssen das die Festplatte auch in ausgebautem Zustand empfindlich gegen Ersch„tterungen und Schl„ge ist.

Da nun das Floppylaufwerk entfernt wurde kann man bereits die Sound_In-Buchse sehen, ebenfalls die drei Spulen mit Ferritmantel und den LM387 der hier als Analogverst„rker dient.

Viele der zum Anlogteil geh„rigen Bauteile befinden sich auf der Unterseite der F030-Hauptplatine, um dort messen zu k„nnen mu„ die Platine komplett ausgebaut werden. Also erstes kann nun mit einen ganz einfachen Digitalmultimeter an der rechten und linken Ferritspule getestet werden ob die Audio-Signale bis wenigstens am Ende der Spulen anliegt.

Der Falcon mu„ auch nicht unbedingt In Betrieb sein um diesen Fehler zu finden. Die Audiosignale lassen sich bei abgeschaltetem Computer leicht bis zum CODEC verfolgen. Ausserdem l„uft er auch mit ausgebautem Diskettenlaufwerk :-)

Auch nach den Spulen, in Richtung des LM387 sollte sich am Multimeter bei angelegtem Stereo-Audiosignal noch was tun. Nat„rlich zeigt das Multimeter keine genauen Werte an, sondern eben nur das ein Signal vorhanden ist. Das „ussert sich mit 'h„pfen' der Anzeige. Besser w„hre hier ein Oszilloskop, was aber eben leider nicht jeder hat.

Liegen die Stereosignale an den Spulenden an, ist man einen Schritt weiter. Die Klinkenbuchse und die Spulen k„nnen hier ausgeschlossen werden.

Die mittlere der drei Spulen hinter der Klinkenbuchse liegt „brigens im Masseanschlu„ der Klinkenbuchse.

Man sollte nun am LM387 an PIN2 (Microphone Left In) MLI und an PIN7 (Microphone Right In) MRI feststellen ob die zuvor an den Ferritspulen gefundenen Audio-Signale hier auch noch ankommen.

Ist das nicht der Fall verschwindet das Signal bereits auf dem Weg zu eben diesem Analogverst„rker. Das war bei einem Falcon der Revision B der Fall. Wenn man von oben auf den Rechner dr„ckte (Floppyseite) war der rechte Kanal mit Kratzger„uschen und Aussetzern verseucht. Nach ausbau der Platine konnte man sehen das sich die L„tanschl„sse der Spulen und einer der Klinkenbuchse bereits durch die Pappe einen Weg zum Masse f„hrenden unteren Abschirmblech gebahnt hatten. Was nat„rlich sofort zu einem Kurzschlu„ des rechten Audiosignals f„hrte.

Abhilfe konnte ich dadurch schaffen das ich die Bauteilebeinchen einfach entsprechen gek„rzt und die Bauteile erneut eingel„tet habe.

Die L„tbatzen waren an den Auflagestellen schon eingedr„ckt.

Das Floppy war in diesem Rechner übrigens nur original ganz leicht
Angeschraubt, ein festziehen der Floppyschrauben hätte den Fehler wohl eher
an den Tag gebracht. So z.B schon bei der Endkontrolle des Falcons!

Die Abschirmbleche liegen in der Mülltonne !

Bei dem zweiten Falcon hatte ein Aussetzter des linken Kanals die Rechneröffnung
zur Folge. Nach dem ich alle Schritte die ich bereits erläutert habe
durchgeführt hatte blieb ich wieder an der Befestigung des Diskettenlaufwerkes
hängen, auch dieses Laufwerk war nur lauwarm angeschraubt.
Bei korrekten Anschrauben und ca. 5 Min. Samplen war dem linken Kanal kein
Geräusch mehr zu entlocken. Um sicher zu gehen kann man die Ausgangssignale des
LM387 (U9) am F030 CODEC (U28) messen, und zwar ist PIN25 des CODEC über die
SMD-Bauteile C128, C124 und R133 für den rechten Kanal zuständig (MRO)
PIN27 des CODEC über die Bauteile C131, C129 und R145 für eben den linken Kanal
verantwortlich, erst der CODEC übernimmt die Wandlung in die digitale Form.

Bei diesem Falcon war es offensichtlich nun so das sich durch das ebenfalls
verspannt eingebaute Diskettenlaufwerk auf der Platinenunterseite die
Bauteile C40, C26 nicht mehr so ganz korrekt verhalten haben.
C40 hatte nur bei kaltem Rechner Verbindung mit seinen Lötstellen, C26 mußte
durch einen neuen SMD-Kondensator ersetzt werden.
Das Audiosignal des linken Kanals lag an der Spule L33 korrekt an, auch am
Spulenausgang in Richtung des LM387. Leider konnte man aber dort am PIN2
kein Audiosignal messen, also mußten alle Bauteile die sich in diesem
Schaltungszweig befinden unter die Lupe genommen werden.
Das mit der Lupe übrigens im wahrsten Sinn des Wortes !

Benötigt habe ich hierzu nur das Multimeter, bzw. den darin eingebauten
elektronischen Durchgangsprüfer. Das Oszilloskop habe ich erst nach der
eigentlichen Reparatur genommen um zu sehen das die Signale auf beiden
Kanälen auch noch nach dem LM387 (U9) korrekt aussehen.
Nur damit es nachher keinen hörbaren Unterschiede zwischen beiden Kanälen gibt.

Bei dieser Gelegenheit ist mir dann auch aufgefallen das beide Falconplatinen
auf der Unterseite aussehen als wenn die kein Fachman gelötet hätte..
Viele der SMD-Bauteile saßen entweder total schief auf ihrem Platz oder sind
so mangelhaft verlötet das man sich über sonst nichts zu Wundern braucht.

Eine genaue Beschreibung aller Bauteile die sich in diesen Signalwegen befinden
würde sicher den Rahmen dieser einfachen Fehlerbeschreibung bei weitem sprengen.
Deshalb hier nur kurz eine Zusammenstellung der Bauteile vom Klinkeneingang
bis zum CODEC (U28).

Linker Audio_IN-Kanal:

L33, R57, R58, C21, C26, R58, R87, C40, R55, LM387 PIN 2
vom LM387 PIN4 bis zum CODEC: C131, R133, C124 CODEC PIN25

Rechter Audio_IN-Kanal:

L35, R92, R90, C81, C79, R84, R88, C55, R87, LM387 PIN 7
vom LM387 PIN5 bis zum CODEC: R145, C129, R131 CODEC PIN27

Die Beschreibung dieser Bauteile bezieht sich auf die Revision B (OLD)

Zu Bemerkem wäre noch das sich neben PIN 25/27 am CODEC jeweils eine
Durchkontaktierung befindet an denen sich besser Messen lässt als an den
Anschlußbeinchen des Codec selber.

Sollte diese Aktion jemand nachstellen müssen so sei noch zu sagen das es
unumgänglich ist alle Sicherheitsvorschriften bezüglich der Netzspannung,
dem Umgang mit einem SMD geeigneten LötKolben dem Werkzeug und der
Schutzvorschriften bezüglich statischer Ladungen einzuhalten.
Ihr Falcon und die sechsfache Multilayerplatine werden es Ihnen Danken :-)

Alle Leiterbahnen der AUDIO_IN Schaltung befinden sich zum Glück auf der
Ober- oder auf der Unterseite der F030 Platine, angesehen als noch ein
recht einfaches und schnelles Unterfangen.

An dieser Stelle sein nochmals eindringlich vor dem Verbiegen, Verspannen oder
sonstwie stark mechanisch Belasten der Multilayerplatine gewarnt.
Viele andere Verbindungen liegen in den inneren Schichten..
Da hilft auch Anbohren nicht weiter..

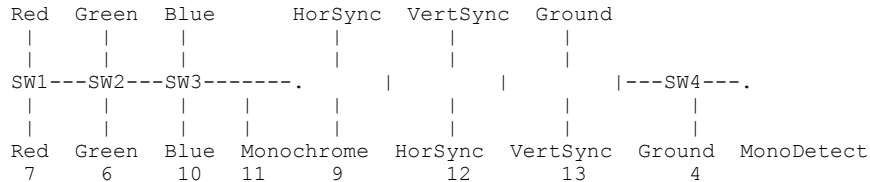
--EOF--

Article: 40874 of comp.sys.atari.st
 From: larserio@IFI.UIO.NO (LarsErikOsterud)
 Newsgroups: comp.sys.atari.st
 Subject: ST and Multisync

Multisync monitor switchbox and cabel (this is tested and works)

If you want monochrome your monitor must handle 71 Hz frequency NOT all monitors do this - To be sure - Test before you buy it!

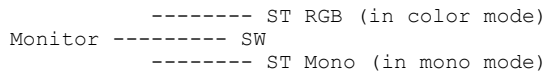
Multisync side (no pin number - they depend on your monitor outlet)



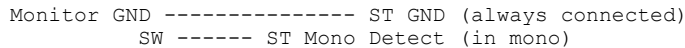
Atari ST, STE or MEGA STE side (number as in operating manual)

The switch you need must be of the type with 4 pins in that can be switched between 2 x 4 pins out (two choices) and works like this:

Switches 1,2,3: When in color red, green and blue passes straight
 ----- through without contact with each other or mono.
 When in monochrome they are all disconnected from
 ST RGB and instead all connected to ST monochrome



Switch 4: Open in color, monochrome detect not connected.
 ----- When in monochrome monochrome detect is connected
 to ground (this tells the ST it's in monochrome).



Don't worry, the worst bit isn't making this - it's getting the 13-pin Atari monitor connector for the ST side and solder it !!

Lars-Erik / ABK-BBS +47 2132659 / _____
 Osterud / larserio@ifi.uio.no / _____
 _____ / _____ / _____ Klubben, The norwegian ST
 _____ / _____ / _____ user association

Lars-Erik Oesterud (larserik@oslonett.no) once wrote:

> Does anyone know how to get a Quantum LPS 270 to work on a MEGA STE
> ?
> Is there a list of harddisks that will work with MEGA STE computers
> ?

> Lars-Erik Osterud Phone: +47 67131571 Fax & ABK-BBS: +47 67132659
> Atari Developer Pager: +47 96634222 Email: larserik@oslonett.no
> SysOp ABK-BBS

>

There is a jumper called PE (parity enable), leave the space empty !!!
Mine works great with a AdSCSI+ on a MST4.

```
--
+-----+-----+
| AOUNIS Stephane | Laboratoire de Physique Nucleaire et des |
+-----+-----+ Hautes Energies - Universite PARIS 6 |
|                               +-----+-----+
| Email: acounis@lpnhp3.in2p3.fr | Tel : (1) 44-27-43-62 |
| acounis@lpnaxp.in2p3.fr | FAX : (1) 44-27-46-38 |
+-----+-----+
"Ask a silly question and you'll get a silly answer"
```

Article 59575 of comp.sys.atari.st:
From: steve-j@teleport.com (Steve Johnson)
Newsgroups: comp.sys.atari.st
Subject: Falcon030 & Digital Audio
Date: 2 Mar 1994 02:55:20 -0800

Okay, here's a text file I got that some of you Falcon030 owners may be interested in:

Guys... do the following mods at your own risk. Remember the Falcon is loaded with surface mounted VLSI (and custom VLSI) so observe all static precautions!

To remove the bass boost from the Falcon's audio output:

Change C58 and C59 from 100pf to 1000pf
Change R123 and R130 from 10K to 15K
Change R59 and R68 from 100K to 4.7K
Change C95 and C90 from 0.33microFarad to 1K resistor

To reduce the built in speaker amplifier gain:

Add a 10K resistor between pin 2 of U21 (at R119) and AGND (at R111)

To reduce the gain of the Falcon's Audio Input by 6dB:

Remove R56, R57, R90, R92, C21, and C81
Change R66 and R87 from 100K to 47K
Add a 470pF NPO +/- 5% capacitor in parallel with R67
Add a 470pF NPO +/- 5% capacitor in parallel with R88
Add a 100pF NPO +/- 5% capacitor in parallel with R66
Add a 100pF NPO +/- 5% capacitor in parallel with R87

I believe that's the Steinberg-recommended way to 'fix' the Falcon030's audio circuitry.

--
Steve Johnson	"People will always be tempted to wipe
Email: steve-j@genie.geis.com	their feet on anything with 'Welcome'
or steve-j@teleport.com	written on it." -- Andy Partridge

```

* STE SIMM mixer by David Hoggan
* Lets STE configure 2.5 Megs properly
* MUST BE FIRST IN AUTO FOLDER!
* Correction 28 Dec 93 to work on MegaSTE by Mike De.Petris
*
* This program assumes 2Mb in bank 0 and 0.5Mb in bank 1.
* It has been found that the other way around doesn't work
* - the MMU configures to a 1Mb machine.

* Install your SIMMs as follows:
*
* (back of STE)
*
* |-- 1meg SIMM  --|
* |-- .5meg SIMM --|
* |-- 1meg SIMM  --|
* |-- .5meg SIMM --|
*
* (front of STE)

    clr.l    -(sp)
    move.w   #32,-(sp)      Switch to super mode
    trap     #1
    addq.l   #6,sp

    cmp.l    #280000,$42E   RAM <= 2.5 megs?
    ble     already        If so, nothing to do
    cmp.b    #9,$424       Is it installed properly?
    bne     already        If so, nothing to do

    move.l   #280000,$42E   Set phystop
    move.l   $4f2,a0        Find reset address (OS Header)
    move.l   4(a0),a0       Take correct reshander also at boot time
    jmp     (a0)            Reset system

already:
    move.l   d0,-(sp)       Returned stack
    move.w   #32,-(sp)     Switch back to user mode
    trap     #1
    addq.l   #6,sp
    clr.w    -(sp)         Back to TOS
    trap     #1

```


Repairing Dead SM124 Monitors

If you have an ATARI SM124 Mono monitor that has recently gone belly up, read on. In particular, this pertains to those of you who have tweaked the coils and trim pots in your monitor to get a full raster display, although it's quite possible that unmodified monitors have been affected as well. Also, if do have a modified monitor that's behaving fine, you still may want to consider the suggestions described in this article, as it could save you a surprise monitor failure that (you KNOW) will definitely happen at the worst possible time.

Finally, this text describes a problem particular to the monitor manufactured by "Gold Star Co", which has the Brightness, Contrast, and Off/On Volume controls recessed on the right side-panel. There may have been other manufacturers of SM-124's with completely different electronics, so if ANY of the physical component layout descriptions written here do not seem to jive, then all bets are off and the information given here may or may not have any practical value. In addition, the monitor 'fix' I'm going to describe should only be done by a trained electronic technician, or at the very least a knowledgeable electronics hacker with good soldering AND de-soldering skills!

Disclaimer:

You can easily get knocked squarely on your butt by the voltages present in your monitor. Whether or not it kills you probably depends on just what you fall on, your present health, and where the resulting flying monitor lands. In any case, your decision to implement the changes described herein is completely your own. I disclaim all responsibility for your health, safety, or any damage to equipment or property caused by fires, implosions or explosions, earthquakes, global flooding, or galactic imbalance resulting from your use of any information in this writing. I think that about covers it ;)

So... once upon a time, we all got our nifty HI-RES Mono monitors, powered them up, and the very first thing we all said was, "Ah.. MAN!!... This monitor is screwed up". Perhaps stronger language was used, but the fact was that we all thought something had to be wrong. Alas though, the tiny picture on that screen was exactly what the ATARI mystics had intended! Of course, it wasn't long before many daring hackers boldly removed the monitor's covers, cast their warrantees aside, and discovered a way to expand the tiny picture to full size. The information spread quickly through the worldwide internets, spider-nets, and fish nets, and soon all mere mortals had full screen displays and could actually read that 6 X 6 font! And so, the all lived happily ever after. But then one day years later, a curse fell upon the land, and darkness fell upon the faces of many of these monitors.

OK, enough with the story narrative. By now, you're beginning to get the picture (no pun intended). The modification to fill the screen does indeed place an additional stress on a portion of the monitors circuitry, which coupled with a design error quite common to many types of computer monitors, ultimately caused the failure.

The good news is that I've now seen quite a few monitors with this identical failure which is 100% repairable, and should leave you with 'good as new' results. The bad news is that you do need some technical skills, but even if you don't, finding a techie friend to help will be abundantly easier having a possible procedure for repair.

Brief technical summary. Right near the coil you adjusted to spread the horizontal width of your display, there's a 2.2 uF capacitor. More specifically, it is a non-polarized electrolytic capacitor, which is actually 2 polarized capacitors wired 'back to back' in a single enclosure. As the screen is widened, the peak voltage level across this part increases, thereby causing more internal stress. This situation is further aggravated by the type of part as well. All capacitor types have certain advantages and disadvantages and when it comes to electrolytic types, the advantage is smaller size. It's disadvantage however is that at higher frequencies, it has high 'leakage' current and acts more like a resistor. The Horizontal frequency passing through this capacitor is in fact quite high, and the resistive qualities at these frequencies cause the capacitors to generate significant heat. Over time, that heat will cause the part to fail even in unmodified monitors, but the modification certainly speeds up the process.

If your monitor has fallen victim to this failure, in a way your lucky. There's a fair chance that the problem (as said earlier) would happen one way or another anyway, and at least now you have a possible fix. First, you need to disconnect the monitor and open it up. If you have the chassis I'm describing there will be two screws on the upper sides, visible from the back; two more screws on the bottom; and a single screw on the back above the power plug, which is a removable 'spade' type AC connector.

After removing the screws, you can remove the back cover, but be sure to be careful of the speaker connections. The speaker is mounted to the inside of the cover, and connects to the monitor's main PC board via a small removable plug. It's a bit tough to get to, but trust me, it IS removable.

Once the cover is free, there are 2 fuses to check first. From now on, all references to right and left will assume you are behind the monitor, with the screen pointing away from you. There is a small PC board mounted vertically on the right side of the chassis which contains a 0.75 amp fuse. There's also a 2 amp fuse immediately to the right of the speaker plug you removed on the main board. Check both fuses with a DVM or other suitable instrument. Most likely, you'll find that it's the 2 amp on that is gone, but either one blown is possible. You can buy these tiny fuses at Radio Shack, and while your there, you can pick up a much more modern replacement for the capacitor we mentioned earlier. There's also a resistor that often cooks when this failure occurs, so you might as well get all this stuff at once. If the problem turns out to be something else, at least you'll be prepared for the eventuality. In any case, read on to do a 'walk through' before buying, to make absolutely sure your chassis physically matches the one we're discussing.

At your local Radio Shack, pick up the following items:

fuse	(pkg of 4)	3/4 amp	270-1048
fuse	(pkg of 4)	2.0 amp	270-1052
fuse	(pkg of 4)	5.0 amp*	270-1056
capacitor	buy 2!	1 uF, 200v metal	272-1055
resistor	(pkg of 2)	680 ohm, 1/2 watt	271-021

* optional purchase, see text.

Since we have plenty of extra fuses now, start by just changing them, and setting up the monitor (still uncovered) with your ST. If the display comes up normally, leave everything powered up and wait a few hours. If all goes well, you can either make the fix were going to describe anyway to save future grief, or put the other parts away, along with this text, in case the problem returns.

Assuming the problem did not go away (no display and/or fuse blown again), there are two paths you can take from here. You can just go ahead and change the suspected components without really knowing if they are at fault, since they'll probably need replacement eventually anyhow. The other possibility is that you don't want to do this until you're a bit more sure it's necessary. If that's the case, read the next few paragraphs. Otherwise, just skip over to "Making the Repair".

If you do want to do some checking, here is a possible (but not foolproof) procedure to see if the suspected capacitor is indeed the culprit. If the 0.75 amp fuse on the vertical board was blown, temporarily replace it with a 2 amp. If the 2 amp fuse on the main board has blown, temporarily replace it with one of the 5 amp fuses. Set up the monitor, still open, near your ATARI, and connect the monitor cable to your machine. Boot the ATARI with the monitor still off. We want to keep our 'powered up' time with the larger (wrong) fuses to a minimum. Now, hook up the power cord and turn on the monitor, being ready to instantly kill power at the first sign of any unsafe condition such as smoke or electrical arcing.

With the higher current fuses installed, the display will probably still be out, but the green pilot light on the front panel should light. If it doesn't, then either the new fuses have blown, or a problem exists somewhere else in the circuit. Of course, the pilot light could just be shot, but if the higher current fuses do actually blow, then forget about the remainder of this discussion and just try implementing the suggested repair. Also, if the screen display does appear with the larger fuses, don't be to happy yet! Twice now I've seen the 'crippled' capacitor seem to work fine for awhile before the final breakdown. So whether the screen display comes up or not, proceed to the next step as long as the fuses are not blowing.

Locate Capacitor C714. It will be the round cylinder on the left side of the main board (from the back), in-between the coil you adjusted when you increased the width and the flyback transformer. The coil is the only adjustable coil on the board, so you can't miss it, and hopefully you know that the flyback transformer is the thing-y with the wire that goes into the top of the picture tube! Locate the capacitor... but don't touch it yet! The flyback puts out thousands of volts, and if there is a flaw in it's insulation, it could arc right into you hand. Just leave the unit powered up for a few minutes, and then turn it off. Now, with your finger, move close to, and finally touch the capacitor from the top. What you're looking for is a HOT part, so be careful! If in fact the part is really hot, it probably has shorted and needs to be replaced.

Also, immediately to the right of the coil, also on the main board is a diode (D-709) and a resistor (R-221). Take a good look at the resistor to see if it looks stressed or burnt. If you want, you can do an in-circuit check with a DVM to see if it has opened. You can make this measurement from the bottom of the board if you want since it's pretty cramped on top. Even if it has not died, you should consider replacement if it appears stressed, or the underside of the board appears burned and discolored at the location of the resistor.

Making the Repair

Making the repair is simply a matter of changing the parts, BUT... there's some special considerations here. You may or may not be aware that radio shack does have 2.2uF, 50 volt non-polarized electrolytic capacitor in stock! But using that part would be dumb since the conditions that caused it to die in the first place have not changed AND, the part was a poor design choice to begin with! We're going to use the two 1.0 uF metalized film capacitors which are NOT electrolytic, AND have a working voltage 4 times that of the original part! We will have to parallel both capacitors to get 2.0uF, which is reasonably close to 2.2 and in fact works just fine. First though, you'll have to de-solder the existing C-714, which will probably be a bit of a chore. Do that, and also get rid of the existing R-221 if it shows the signs of heat stress (or is open) as we discussed earlier. You can just cut it out if you want, since we'll be soldering the new parts to the underside of the board.

Now, take the two 1.0uF capacitors, and solder them together in a neat parallel arrangement, that has a pair of leads bent to accommodate the spacing of the leads on the original capacitor. Look at the underside of the board and observe the original C-714 solder pads, as well as the clearance to the bottom when the cover is replaced. Give it a little thought and come up with a mechanically sound and visually sensible combination. The capacitors again, are not polarized so direction does not matter. And remember... parallel, NOT series!!! Solder the parallel combination to the C-714 pads under the board. Try to push the leads right into the holes left by the removal of the old capacitor.

R-221 should also be replaced from the bottom side of the board unless you're really patient, and have some small tools. It's much easier to just trim the leads and tack solder it in place under the board. If the board looks a bit burned in this area, you may want to clean it up a bit prior to installing the new resistor.

Finally, don't forget to put the proper fuse values back into BOTH locations. The original fuse values are important protection and do not normally blow unless there is a real problem. I definitely do not recommend leaving the larger fuses values in place. Do a bench check on the ST again, still with the cover off. You may need to re-adjust the horizontal width, or the position tabs on the neck of the picture tube to center and set up the display. Once this is done, you can put the cover back on, and you're back in business. If the fix does NOT help, just double check your soldering and connections to make sure you didn't wire something wrong.

Hope this fix was helpful. If so, I'd like you to drop me an EMAIL, since I'm curious as to just how many fried monitors out there had this problem. Also, if you're not able to make the repairs yourself and live anywhere in Islip, Long Island, NY area, send me E.Mail and maybe I can help you out. My GE mail address is R.CONSTAN, or Internet: rconstan@maestro.com.

-Randy Constan, Elfin Magic Co.

Mouse/Joystick Switch for ST Computers

Original Article By John Ogrizek

If you own a ST you will know how much trouble it is to change between a joystick and the mouse since the mouse/joystick ports are beneath the computer. Not only is it difficult to change between the mouse and the joystick but it can also result in damage to the cables. The unit has three ports on it. Port 0 is the mouse/joystick port (two connectors on the mouse/joystick switch) and is switchable. Port 1 is a straight connection from the computer. If you've had enough of cable swapping then this project is for you. It will only cost you half the price of a commercially available unit. All that is required is the ability to solder and a few hours of your time. The following parts are required:

- 2 x female DB9 connectors
- 3 x male DB9 connectors
- 1 x 6 pole 2 position rotary switch and knob to suit
- 1 x required length of ribbon or round computer cable (at least 14 strand)
- 1 x project box
- hookup wire, solder, etc.

If the wiring diagram below is followed carefully no problems should be encountered. The two DB9 connectors that plug into the computer can be of the crimp or solder type, the only difference being is that the solder type are cheaper than the crimp type. If you are using the crimp type then the cable can be crimped onto the connector as shown by using a flat object to press on the back of the connectors CAREFULLY, as the connectors are easy to break. (If you have the correct crimping tool it makes it a lot easier.)

Ribbon or round cable can be used, but the round cable makes it harder to line up the wires on the connector. At this stage a hole can now be drilled (for round cable) or filed (for ribbon) in the project box.

Next are the holes for the DB9 connectors, which is the most difficult part of the project. The connectors can either be placed from the inside of the box or the outside. If they are placed from the inside you must file out the hole in the box to exactly the shape of the connector, which requires a lot of patience. If the connectors are mounted on the outside of the box then filing out the holes is a lot easier as there is more room for error.

Once you have filed out the holes for the connectors the mounting holes must be drilled. Make sure the positioning of the switch does not foul the connectors.

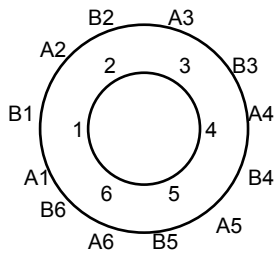
Now the connectors and switch can be screwed into place and the cable can be run into the box. You can either use a cable clamp to fasten the cable to the box or just use a bit of super glue to glue it to the box.

The wiring is next and should be done carefully following the wiring diagram below and checked on completion.

The only thing to be done now is to plug it into your computer and test it out, knowing that you'll never have to reach underneath your computer again.

Switch 1

2 Position, 6 Pole Switch



Mouse S.	Switch 1
1	A3
2	A4
3	A5
4	A6
5	N/C
6	A2
7	+5v
8	GND
9	A1

Joy 1 S.	Switch 1
1	B3
2	B4
3	B5
4	B6
5	N/C
6	B2
7	N/C
8	GND
9	N/C

Port 1 (Joystick) is not shown as it is a straight connection from socket to plug, connecting line one with line one, etc.

Port 0 (Joystick / Mouse) is wired via a switch so that either the mouse or the joystick may be activated without having to disconnect the other attached device. This could be a simple three pole, double throw toggle switch.

Atari Models - From the Hardware FAQ

Here is an incomplete list of all the Atari ST and up range. If someone has more details for it, or can notice a correction, please contact me, as I would like a complete list. I would especially like some rundown on the 8-bit stuff if anyone is interested, and all of the TOS versions from each computer, etc.

CPU's:

- 8 - Machine runs at 8 MHz.
- 16 - Machine runs at 16 MHz.
- 32 - Machine runs at 32 MHz.
- 8/16 - Machine switchable between 8 and 16 MHz.
- 16/32 - Machine switchable between 16 and 32 MHz.
- 68000 - Processor is a Motorola 68000.
- 68030 - Processor is a Motorola 68030.

RAM:

- S - Machine has SIMM slots for expansion.
- DB - Machine has a Daughter Board for RAM expansion.

Inbuilt:

- F - Floppy disk drive inbuilt.
- M - TV modulator inbuilt.

Extra:

- AV - 8 bit digital stereo audio, 4096 colors, hardware scrolling, analog joystick ports.
- MB - MegaBus processor-direct bus slots.
- 6U - 6U VME bus slots, second serial port.
- FC - Motorola 56000 DSP, 16 bit stereo audio, much enhanced video.

TOS:

Highest built in version of TOS.

Name	CPU	RAM	TOS	Inbuilt	Extra	Comment
130 ST	8 - 68000	128 K	1.0	M		Never Marketed
260 ST	8 - 68000	256 K	1.0	M		
520 ST	8 - 68000	512 K	1.0	M		
520 ST+	8 - 68000	1 MB	1.0	M		
520 STm	8 - 68000	512 K	1.2	M		
520 STfm	8 - 68000	512 K	1.2	FM		
1040 ST	8 - 68000	1 MB	1.0	FM		
1040 STf	8 - 68000	1 MB	1.2	F		
1040 STm	8 - 68000	1 MB	1.2	M		
1040 STfm	8 - 68000	1 MB	1.2	FM		
Mega 1	8 - 68000	1 MB	1.2	F	MB	
Mega 2	8 - 68000	2 MB	1.2	F	MB	
Mega 4	8 - 68000	4 MB	1.2	F	MB	
STacy	8 - 68000	1 MB	???	F		Laptop
STBook	8 - 68000	1 MB	???	F		Notebook
520 STe	8 - 68000	512 K, S	1.62	FM	AV	
1040 STe	8 - 68000	1 MB, S	1.62	FM	AV	
Mega STe	16 - 68000	1 MB, S	2.06	F	AV, MB, 6U	
TT	16 - 68030					Developers Only
TT	32 - 68030	S	3.06	F	AV, MB, 6U	Various Configs
Falcon 030	16 - 68030	14 MB, DB	4.04	F	AV, FC	
ATW	Inmos T800					
2080 ST	8 - 68000	2 MB				
2080 STe	8 - 68000	2 MB				
4160 ST	8 - 68000	4 MB				
4160 STe	8 - 68000	4 MB				
Medusa	64 - 68040					

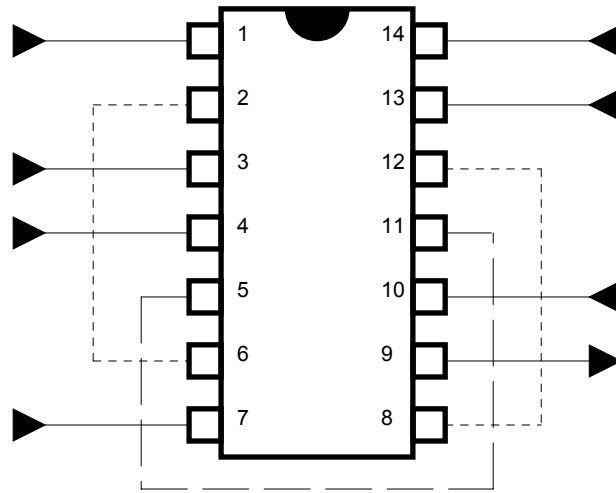
TRACKER-BALL FOR ATARI ST

by A. Schaffert

The use of a mouse for controlling a computer has become common practice. There may be a problem, though: space. It often happens that there is not enough room to move the mouse when the desk is littered with print-outs, pencils, notepads, books, diskettes and the odd cup of coffee. These space problems can be solved by the use of a tracker-ball, which is simply a mouse turned upside down.

The operation of a tracker-ball is basically identical to that of a mouse, but the ball movement is achieved via direct operation by the hand, rather than via the desk surface, or, if this is too smooth or slippery, the mouse pad. The principle of operation of the tracker-ball is clearly illustrated by the drawing in Fig. 1. Two spindles, fitted at right angles, drive encoder discs. Two optocouplers are fitted for each disk, to translate the ball rotation into a logic signal. A small auxiliary spindle (visible in the photograph of Fig. 5) is provided on the home-made assembly to allow adjusting the height of the ball. To avoid slip between the ball and the two main spindles, the auxiliary spindle supports the ball just below the 'equator' so that its own weight keeps it in firm contact with the spindles.

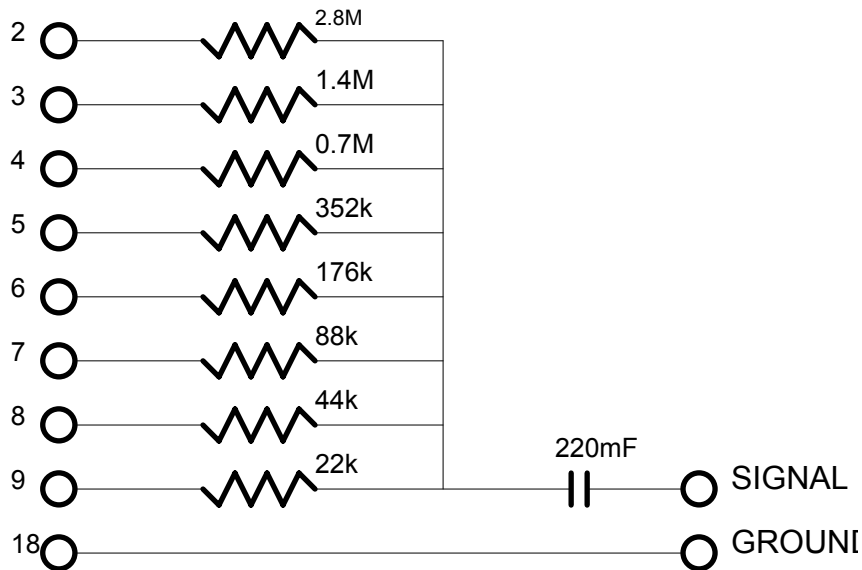
384000 bps fuer jeden ST/STe
Freeware-Schaltung (c) 06.04.1991 by Juergen Meyer
Jegliche Kommerzielle Nutzung (Abdruck, Einbau von Gewerbtreibenden) ist VERBOTEN



74LS74

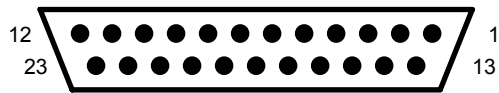
What is a SoundPlayer?

A soundplayer is a D/A converter that you plug into the printer port of the computer. It is compatible to the Covox Little Speech Thing. Here's how to build one:



The circuit consists of eight resistors and one capacitor. The circuit is basically an 8-bit D-A Converter. The capacitor is not essential, but it smoothes the signal coming out of the circuit, thus giving a better sound. The pin numbers on the left correspond to those of the RS232 port. For optimum performance, the circuit should be connected to a HiFi or an amplifying circuit.

A520 Modulator - 23 Pin D-Sub Female



Pin	Signal	Description
1	XCLK*	External clock input
2	XCLKEN*	External clock enable (active low)
3	RED	Analog Red output
4	GREEN	Analog Green output
5	BLUE	Analog Blue output
6	DI	Digital Illuminance (Blue Bit 0)
7	DB	Digital Blue (Blue Bit 3)
8	DG	Digital Green (Green Bit 3)
9	DR	Digital Red (Red Bit 3)
10	CSYNC*	Composite Sync (out only)
11	HSYNC*	Horizontal Sync (in/out)
12	VSYNC	Vertical Sync (in/out)
13	GNDRTN	Digital GND (Return for pins 1 & 2)
14	ZD*	Zero Detect (used for genlocks)
15	C1*	Clock Out (3.5X MHz)
16	GND	Analog GND
17	GND	
18	GND	
19	GND	
20	GND	
21	-12V	-12V (50mA) : On A1000 this is -5V, others -12V
22	+12V	+12V (175mA)
23	+5V	+5V (300mA)

Pins 11 and 12 are inputs with a Genlock otherwise they are outputs.

The above table is from the A500 manual, so is probably the video for the A500. Thanks go to Petri Pihkala (ppihkala@mits.mdata.fi).

NEC Multisync 3D User's Manual

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION

- 1.1 GENERAL INFORMATION
- 1.2 FEATURES OF THE MULTISYNC 3D COLOR MONITOR
- 1.3 CAUTIONS WHEN USING THE MULTISYNC 3D COLOR MONITOR

CHAPTER 2. USING THE MULTISYNC 3D

- 2.1 GETTING ACQUAINTED WITH YOUR MULTISYNC 3D COLOR MONITOR
- 2.2 ADJUSTING THE MULTISYNC 3D COLOR MONITOR'S CONTROLS
 - 2.2.1 USER CONTROL NAMES AND OPERATIONS
 - 2.2.2 SIGNAL IDENTIFICATION FLOWCHART
- 2.3 MAKING THE RIGHT CONNECTIONS
 - 2.3.1 WITH THE IBM PS/2
 - 2.3.2 WITH THE IBM PC, PC/XT AND PC/AT
 - 2.3.3 WITH THE APPLE MACINTOSH II VIDEO CARD
 - 2.3.4 QUICK OPERATION CHART

CHAPTER 3. TROUBLESHOOTING

CHAPTER 4. SPECIFICATIONS

IBM PC/XT/AT, IBM PS/2, MCGA, VGA and 8514/A are registered trademarks of International Business Machines Corporation.
Apple and Macintosh are registered trademarks of the Apple Computer Inc.
Macintosh II Video Card is a trademark of Apple Computer Inc.

CHAPTER 1. INTRODUCTION

1.1 GENERAL INFORMATION

Congratulations on purchasing a MultiSync 3D Color Monitor. We hope this User's Manual is helpful in the installation, operation and connection of your MultiSync 3D Color Monitor; if not, please let us know.

The MultiSync 3D Color Monitor, and Intelligent Monitor from NEC, is a microprocessor based, digitally controlled multiple frequency Color Monitor. The MultiSync 3D Color Monitor automatically adjusts to graphics adapter scanning frequencies between 15.5kHz and 38kHz (Horizontal), and between 50 Hz and 90Hz (Vertical).

As you can see, "design" was the top priority with the MultiSync 3D Color Monitor. Besides the dramatic cabinet design, the MultiSync 3D Color Monitor was designed with microprocessor based intelligence, allowing it to operate at each mode with the precision of a fixed frequency monitor.

The MultiSync 3D Color Monitor gives IBM PC, PC/XT, PC/AT, Personal System/2 (PS/2), Apple Macintosh II and compatible users crisp text and vivid color graphics displays when using the following graphics adapters (the MDA, CGA, EGA, PGC, MCGA, VGA and 8514/A) and Apple Macintosh II Video Card.

The MultiSync 3D Color Monitor can also be used with other "non-standard" graphics adapters to provide users with an extremely wide range of color monitor compatibility.

1.2 FEATURES OF THE MULTISYNC 3D COLOR MONITOR

- The MultiSync 3D Color Monitor is microprocessor controlled with a resident memory base of pre-programmed screen and input configurations and also digital controls.
- The MultiSync 3D Color Monitor automatically scans all horizontal frequencies between 15.5 kHz and 38 kHz, and all vertical frequencies between 50 Hz and 90 Hz.
- The MultiSync 3D Color Monitor is compatible with the IBM PC, PC/XT, PC/AT, PS/2, the Apple Macintosh II and compatibles.
- The MultiSync 3D Color Monitor is compatible with the IBM Monochrome Display Adapter, the IBM Color Graphics Adapter, the IBM Enhanced Graphics Adapter, the IBM Professional Graphics Controller, the IBM MultiColor Graphics Array, the IBM Video Graphics Array, the IBM 8514/A, the Apple Macintosh II Video Card and other compatible graphics adapters.
- The MultiSync 3D Color Monitor's wide compatibility makes it possible to upgrade adapters or software without purchasing a new monitor.
- The MultiSync 3D Color Monitor has a maximum horizontal resolution of 1024 dots and a maximum vertical resolution of 768 lines for superior clarity of display.
- The MultiSync 3D Color Monitor offers both TTL and ANALOG signal inputs, and in the ANALOG mode can display an unlimited palette of colors depending on the graphics adapter and software being used.
- The MultiSync 3D Color Monitor automatically adjusts to either a TTL signal input or an ANALOG signal input.
- The MultiSync 3D Color Monitor has a 14" diagonally measured, non-glare CRT that yields a 13" viewing area.
- A captive signal cable is included with a 15 pin mini (HD) D-sub connector that is configured for IBM's MCGA, VGA and 8514/A graphics adapters.
- Signal cable adapters for graphics adapters with 9 Pin D-Sub connector and Macintosh II Video Card are included.

1.3 CAUTIONS WHEN USING THE MULTISYNC 3D COLOR MONITOR

When setting up and using your MultiSync 3D Color Monitor, pay special attention to the following:

- To eliminate eye fatigue, do not use your MultiSync 3D Color Monitor against a bright background or where sunlight or other light sources will shine directly on the monitor.
- Your MultiSync 3D Color Monitor should be placed just below eye level for optimum viewing.
- Allow adequate ventilation around the monitor so that heat from the monitor can properly dissipate.
- Neither the monitor itself, nor any other heavy object, should rest on the power cord. Damage to a power cord can cause fire or electrical shock.
- Keep the monitor away from high capacity transformers, electric motors and other strong magnetic fields.
- Your MultiSync 3D Color Monitor should not be used in damp, dusty or dirty areas.
- Handle your MultiSync 3D Color Monitor with care when transporting.

CHAPTER 2. USING THE MULTISYNC 3D

2.1 GETTING ACQUAINTED WITH YOUR MULTISYNC 3D COLOR MONITOR

Refer to the table below to be sure that your MultiSync 3D Color Monitor package includes all of the items in this picture. Save the original box and packing materials in case you have to ship or transport your MultiSync 3D Color Monitor.

1. MultiSync 3D (JC-1404HME/E-1/EE/EE-1/ED/ED-1/R/N Color Monitor)
2. Power cord
In case of JC-1404HMEE/EE-1, the end of the power cord is as follows.
Green and Yellow : Earth
Blue : Neutral
Brown : Live
3. User's Manual
4. 15p-9p Adapter (for MDA, CGA, EGA, PGC)
5. 15p-15p Adapter (for Apple Macintosh II)
6. Tilt/Swivel stand: Enables you to position the MultiSync 3D Color Monitor at the best angle and tilt for maximum viewing comfort.

2.2 ADJUSTING THE MULTISYNC 3D COLOR MONITOR'S CONTROLS

Before connecting the MultiSync 3D Color Monitor with either an IBM personal computer or a compatible, the IBM Personal System/2, or the Apple Macintosh II computer, take time to familiarize yourself with the controls that you can adjust for the optimum display from the MultiSync 3D Color Monitor. Chapter 2.2.1 outlines the controls and power switch for the monitor. Chapter 2.2.2 explains the signal identification flowchart. Chapter 2.3.1 is for users of IBM's Personal System/2. Chapter 2.3.2 shows how to connect the IBM PC, PC/XT, PC/AT or compatible computers. Owners of the Apple Macintosh II should refer to chapter 2.3.3. Chapter 2.4 contains a Quick Reference Chart summarizing the steps necessary to connect and adjust the MultiSync 3D color Monitor for the optimum in high resolution color display viewing.

2.2.1 USER CONTROL NAMES AND OPERATIONS

❶ POWER SWITCH

Used to turn the power ON or OFF. When the power is ON, the LED power indicator, located on the front of the monitor, is lit.

❷ BRIGHTNESS CONTROL

Used to adjust the picture brightness of screen.

❸ CONTRAST CONTROL

Adjust the display to the contrast preferred by the user.

❹ COLOR SWITCH

When the monitor receives TTL color video signal, one of the four color configurations (AUTO, 8, 16, 64) must be selected.

COLOR SWITCH	COLOR MODE
Auto	IBM CGA, EGA or Compatible Graphics adapter
8	8 Colors
16	16 Colors with IBM Brown
64	64 Colors

This switch only works with TTL modes that the monitor identifies automatically. (See section 2.2.2)

❺ MODE SWITCH

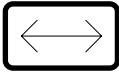
This switch selects either the monochrome mode or color mode with a TTL video signal input.

With an Analog video signal input and MODE SWITCH ON, the monitor will give the user an additional user programmable memory area.

MODE SWITCH	Input Video Signal	MODE
OFF	TTL	General Color Mode (CGA, EGA)
OFF	ANALOG	General Analog Mode (VGA, PGC, etc.)
ON	TTL	Monochrome Mode (MDA)
ON	ANALOG	User Programmable Memory Area

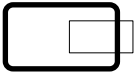
For more detail, see the signal identification flowchart. (section 2.2.2)

⑥ H. SIZE CONTROL



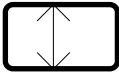
Adjust these controls for the preferred horizontal size of the display.

⑦ H. POSITION CONTROL



Adjust these controls for the proper horizontal position of the display.

⑧ V. SIZE CONTROL



Adjust these controls for the preferred vertical size of the display.

⑨ V. POSITION CONTROL



Adjust these controls for the proper vertical position of the display.

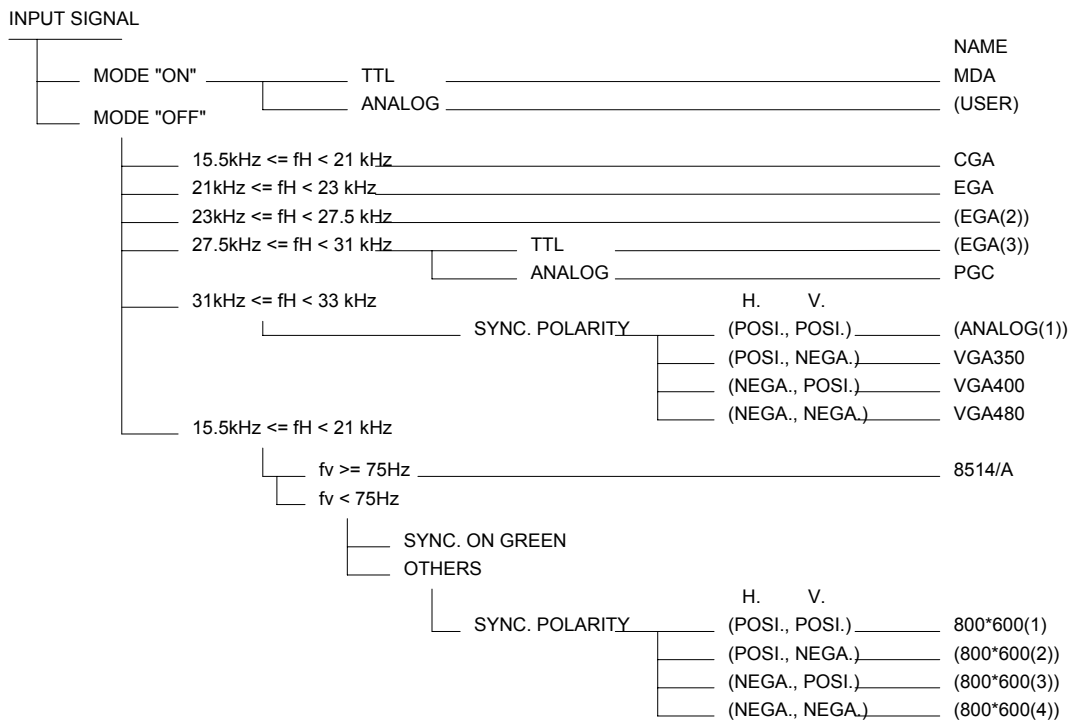
A few seconds after the user has adjusted the size and/or position, the monitor's CPU will save the new control data into its memory automatically.

NOTE: The monitor will display the screen configuration that was saved last.

⑩ MEMORY RECALL BUTTON

This button resets that data of user changeable memory and recalls the factory preset data.

2.2.2 SIGNAL IDENTIFICATION FLOWCHART



2.3 MAKING THE RIGHT CONNECTIONS

2.3.1 WITH THE IBM PS/2

The MultiSync 3D Color Monitor can be used with an IBM Personal System/2 in the following configurations.

For an IBM PS/2 equipped with the IBM MultiColor Graphics Array (MCGA), the IBM Video Graphics Array (VGA) or the IBM 8514/A adapter for models 50, 60, 70 and 80, follow these steps:

1. Make sure the power to the MultiSync 3D Color Monitor and the IBM PS/2 is turned OFF.
2. Connect the end of the MultiSync 3D Color Monitor cable to the connector of the PS/2's MCGA, VGA or 8514/A adapter.
3. Connect one end of the power cable to the MultiSync 3D Color Monitor and the other end to a power outlet.

2.3.2 WITH THE IBM PC, PC/XT and PC/AT

The MultiSync 3D Color Monitor can be used with IBM personal computers in the following configurations.

Using the IBM PC, PC/XT, PC/AT or compatible computer equipped with a non-IBM VGA board, super VGA board, the IBM MDA, CGA, EGA, PGC or compatible adapter, follow these steps:

1. Be sure the POWER to the MultiSync 3D Color Monitor and the computer is turned to OFF.
2. Install the video adapter according to the instructions accompanying it. If the video adapter gives you the option of installing for a multiple frequency monitor, set the dip switch on the adapter accordingly.
3. If your graphics adapter is the IBM MDA, CGA, EGA, PGC or compatible adapter, which has a 9 pin output connector, attach the 15p-9p Adapter to the output connector of the graphics adapter.
4. Connect the end of the MultiSync 3D Color Monitor signal cable to the output connector of the graphics adapter or the other end of the 15p-9p Adapter (if connected).
5. Connect one end of the power cable to the MultiSync 3D Color Monitor and the other end to a power output.

2.3.3 WITH THE APPLE MACINTOSH II VIDEO CARD

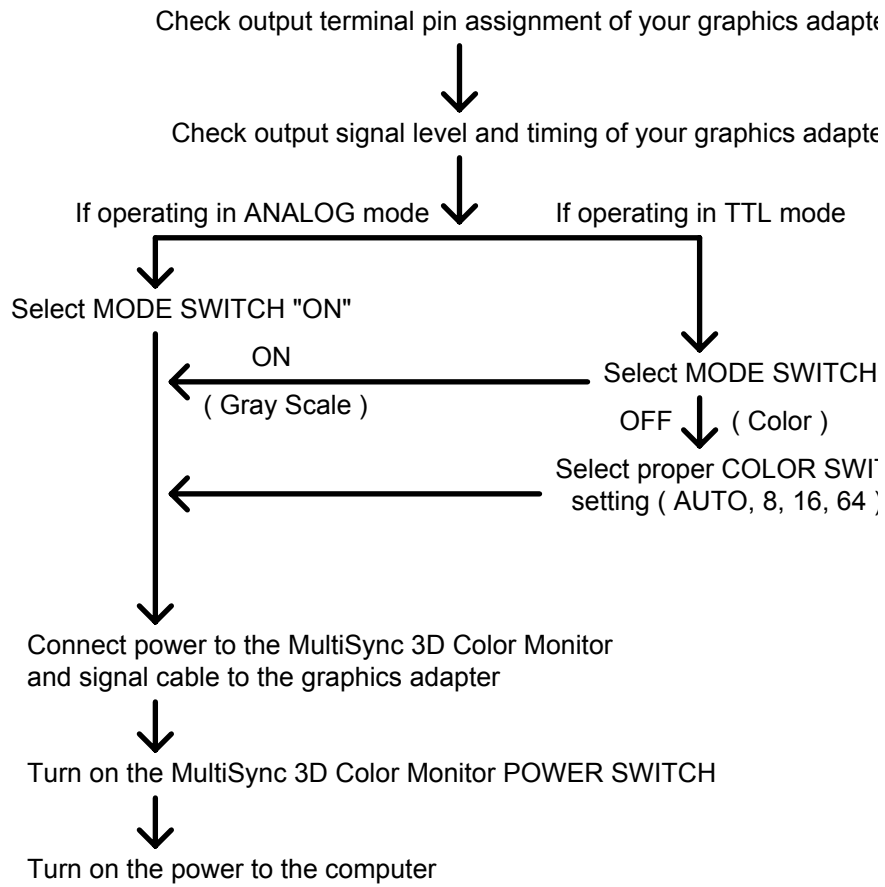
The MultiSync 3D Color Monitor can be used with the Apple Macintosh II computer in the following configurations.

Using the Apple Macintosh II or compatible computer equipped with the Apple Macintosh II Video Card, follow these steps:

1. Be sure the POWER to the MultiSync 3D Color Monitor is turned OFF.
2. Install the Apple Macintosh II Video Card according to the instructions accompanying it.
3. Attach the 15p-15p Adapter to the output connector of the Apple Macintosh II Video Card.
4. Connect the end of the MultiSync 3D Color Monitor signal cable to the other end of the 15p-15p Adapter.
5. Connect one end of the power cable to the MultiSync 3D Color Monitor and the other end to a power output.

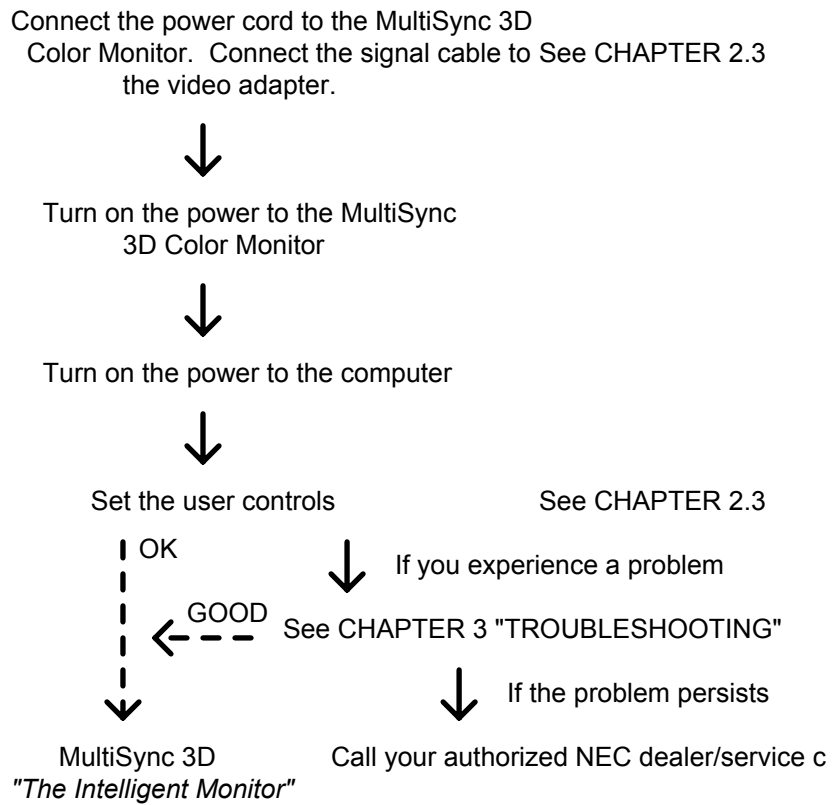
2.3.4 CAUTION WITH OTHER COMPUTERS

If your computer or graphics adapter is not IBM compatible, refer to the diagram below to determine compatibility or consult your authorized NEC dealer.



2.4 QUICK OPERATION CHART

To summarize the steps in connecting your computer and video adapter with the MultiSync 3D Color Monitor and in setting the necessary controls, refer to the chart below.



CHAPTER 3. TROUBLESHOOTING

Before calling NEC or an authorized service center please check that the following items are properly connected or set. If a non-IBM computer or graphics adapter, non-Apple Macintosh II computer or Video Card is being used make sure the signal timing and pin assignments meet the specifications outlined in APPENDIX B or C.

PROBLEM

CHECK THESE ITEMS

No picture

POWER SWITCH, signal cable
Computer power switch

Abnormal picture

1. Gray scale does not correctly display when connected to an IBM MDA board or compatible

MODE SWITCH should be "ON"

2. 16 or 64 colors do not correctly display when connected to an IBM CGA, EGA board or compatible

MODE SWITCH should be "OFF"
COLOR SWITCH should be "AUTO"

3. 8, 16 or 64 colors do not correctly display with non-IBM

MODE SWITCH should be "OFF"
8/16/64 colors should be selected with COLOR SWITCH to match the output of the graphics board

4. Display is too large, too small, or off-center

Push the MEMORY RECALL SWITCH when connected to the IBM computer, the Apple Macintosh II computer or compatible
Adjust H. SIZE, V. SIZE, H. POSITION, V. POSITION controls when connected to non-IBM, non-Apple computer.

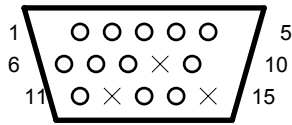
CHAPTER 4. SPECIFICATIONS

Picture Tube	13 Visual inches diagonal 90 degree deflection, 0.28mm Trio dot pitch Dot type black matrix, Not-long persistence phosphor, Dark bulb, Direct etch
Input Signal	Video : TTL Level Positive : ANALOG 0.7 Vp-p/75Ω Positive Sync : Separate sync. TTL Level Horizontal sync. Positive/Negative Vertical sync. Positive/Negative : Composite sync. TTL Level Positive/Negative : Composite sync. on Green Video sync. 0.3 Vp-p Negative (Video 0.7 Vp-p Positive)
Display Colors	TTL Input: 8/16/64 colors Analog Input: Unlimited colors
Synchronization	Horizontal: 15.5 kHz to 38kHz (Automatically) Vertical: 50 Hz to 90 Hz (Automatically)
Resolution	Horizontal: 1024 dots Vertical: 768 lines (interlaced)
Video Band Width	45 MHz
Active Display Area ₍₁₎	Horizontal: 240mm Vertical: 180mm
Misconvergence	Less than 0.6mm
Power Supply	AC 220V, 50 Hz
Power Consumption	82W
Dimensions	355(W) x 363(H) x 406(D) mm
Weight	16 kg
Environmental Considerations	Operating Temperature 0°C to +40°C Humidity 30% to 80% Storage Temperature -20°C to +60°C Humidity 10% to 90%

Technical specifications subject to change without notice.
(Like when you get a lightning bolt through your power supply, etc. ;^)

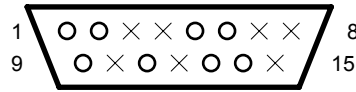
(1) - Active display area is changed by signal timing.

APPENDIX A. PIN ASSIGNMENTS



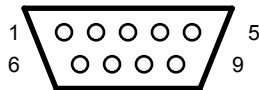
15 pin mini (HD) D-sub male *

- 1 - RED VIDEO
- 2 - GREEN VIDEO
- 3 - BLUE VIDEO
- 4 - GROUND
- 5 - GROUND
- 6 - RED GROUND
- 7 - GREEN GROUND
- 8 - BLUE GROUND
- 9 - *NOT CONNECTED*
- 10 - GROUND
- 11 - GROUND
- 12 - *NOT CONNECTED*
- 13 - H. SYNC
- 14 - V. SYNC
- 15 - *NOT CONNECTED*



15 pin D-sub male **

- 1 - RED GROUND
- 2 - RED
- 3 - *NOT CONNECTED*
- 4 - *NOT CONNECTED*
- 5 - GREEN WITH SYNC
- 6 - GREEN GROUND
- 7 - *NOT CONNECTED*
- 8 - *NOT CONNECTED*
- 9 - BLUE
- 10 - *NOT CONNECTED*
- 11 - GROUND
- 12 - *NOT CONNECTED*
- 13 - BLUE GROUND
- 14 - GROUND
- 15 - *NOT CONNECTED*



9 pin D-sub male ***

IBM MDA	IBM CGA	IBM EGA	IBM PGC
1 - GROUND	GROUND	GROUND	RED
2 - <i>NOT CONNECTED</i>	GROUND	SECONDARY RED	GREEN
3 - <i>NOT CONNECTED</i>	RED	PRIMARY RED	BLUE
4 - <i>NOT CONNECTED</i>	GREEN	PRIMARY GREEN	COMPOSITE SYNC
5 - <i>NOT CONNECTED</i>	BLUE	PRIMARY BLUE	<i>NOT CONNECTED</i>
6 - INTENSITY	INTENSITY	SECONDARY GREEN	RED GROUND
7 - VIDEO	<i>NOT CONNECTED</i>	SECONDARY BLUE	GREEN GROUND
8 - H. SYNC	H. SYNC	H. SYNC	BLUE GROUND
9 - V. SYNC	V. SYNC	V. SYNC	GROUND

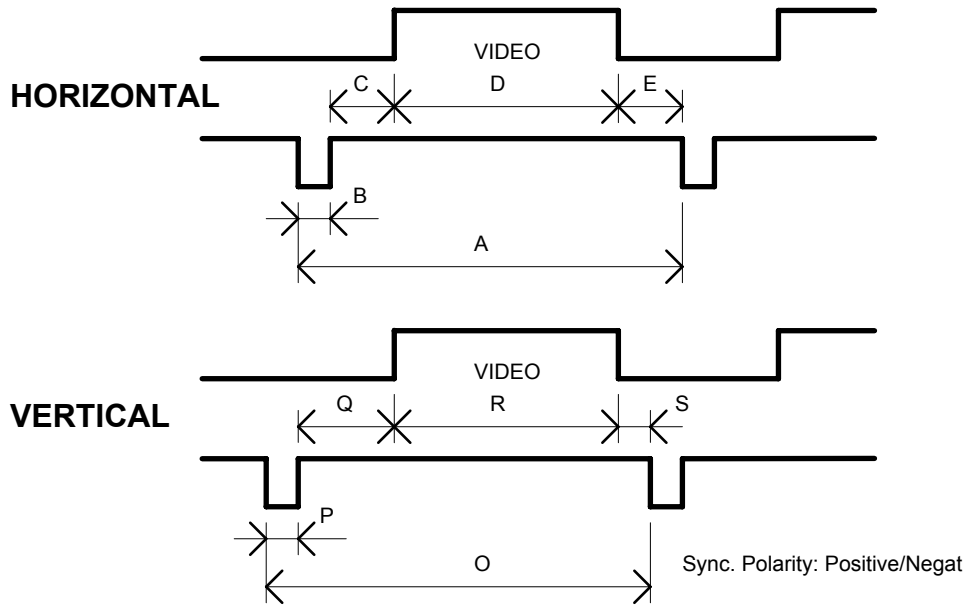
* 15 Pin mini (HD) D-sub male connector is on the captive signal cable, for use with the IBM VGA, 8514/A or compatible graphics adapters.

** 15 Pin D-sub male connector is on the 15 Pin mini (HD) to 15 Pin D-sub adapter, for use with the Apple Macintosh II Video Card.

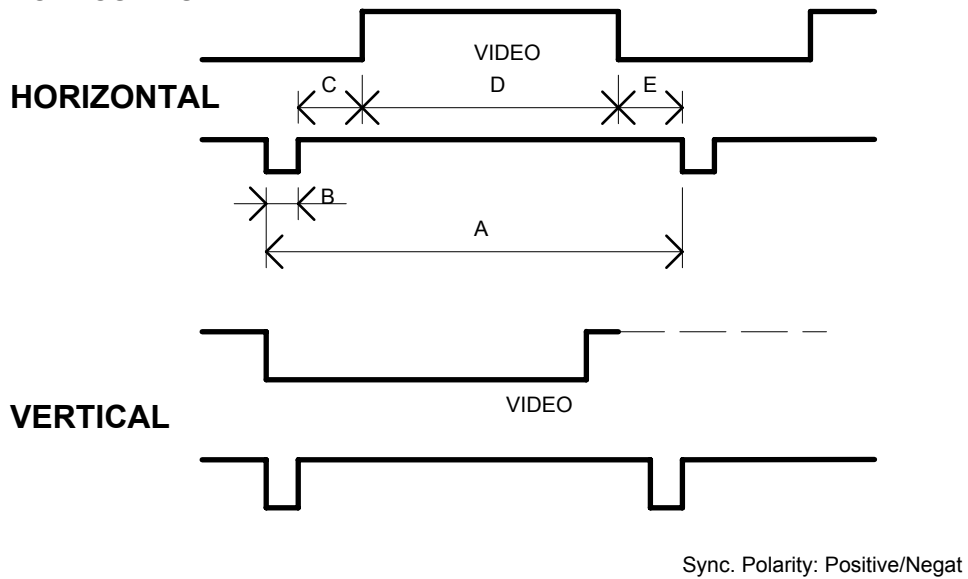
*** 9 Pin D-sub male connector is on the 15 Pin mini (HD) to 9 Pin D-sub adapter, for use with the IBM MDA, CGA, EGA, PGC or compatible graphics adapters.

APPENDIX B. TIMING CHARTS

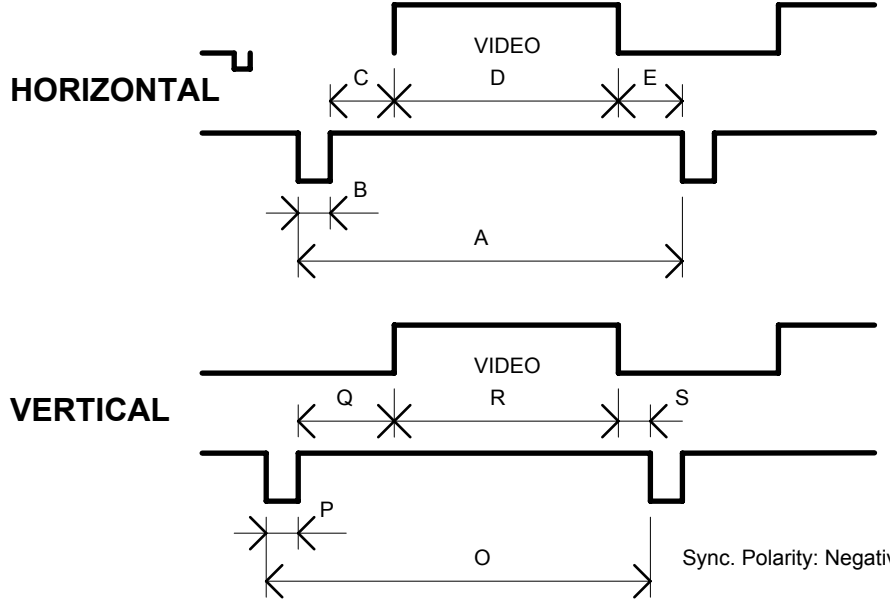
At last edit, separate sync is the only one correct.
SEPARATE SYNC.



COMPOSITE SYNC.



COMPOSITE SYNC. & VIDEO (SYNC. ON GREEN)



PRESET TIMING

	MDA Compatibl e	CGA Compatibl e	EGA Compatibl e	PGC Compatibl e	VGA/MCGA Compatible		8514/A Compatibl e	Macintosh II Video Card	800 x 600
f_H	18.43 kHz	15.85 kHz	22 kHz	30.48 kHz	31.47 kHz		35.52 kHz	35 kHz	35.16 kHz
$A_{\mu s}$	54.3	63	45.5	33	31.78		28.15	28.57	28.44
$B_{\mu s}$	8.3	4.2	4.9	4.5	3.81		3.92	2.1	2
$C_{\mu s}$	1.1	7.2	1.6	2.8	1.91		1.25	3.2	3.56
$D_{\mu s}$	44.3	45	39	25.6	25.42		22.8	21.5	22.22
$E_{\mu s}$	0.6	6.6	0	0.1	0.64		0.18	2.1	0.67
f_V	50 Hz	60.5 Hz	60 Hz	60 Hz	70.08 Hz	59.95 Hz	87 Hz	67 Hz	56 Hz
O_{ms}	20.1	16.52	16.68	16.6	14.27	14.27	16.68	11.5	17.78
P_{ms}	0.9	0.19	0.6	0.07	0.06	0.06	0.06	0.113	0.086
Q_{ms}	0.2	2.15	0.08	2.12	1.91	1.11	1.05	0.563	1.12
R_{ms}	19.0	12.6	16	13.05	11.12	12.71	15.25	10.81	13.7
S_{ms}	0	1.58	0	1.36	1.18	0.38	0.32	0.014	0.086
Remarks	TTL Video	TTL Video	TTL Video	Analog Video	Analog Video	Analog Video	Analog Video Interlace	Analog Video	Analog Video

NEC Multisync 5FG/6FG User's Manual

Connection to your personal computer (NEC, IBM and compatibles)

The MultiSync 5FG/6FG color monitor compliments NEC or IBM PC/XT, AT, 386, 486, PS/2 computer and compatibles.

Your System should have one of two configurations:

1. The video controller is built into the computer.
2. The video controller is in the form of a video card (sometimes referred to as a graphic card, video adapter, or graphics board.)

Both configurations have a video connector (or a CRT PORT on NEC laptop computers). If you are not sure as to which connector is the video connector, consult your computer or video card manual.

To attach the monitor to your system, follow the instructions below.

1. Turn off the power to the monitor and the computer.
2. If necessary, install the video card. For more information, refer to the manual accompanying the card.
3. Make sure the

PC - DOS Stuff

Changing Screen Attributes

remember : in CONFIG.SYS -
device=c:\dos\ansi.sys
or
devicehigh=c:\dos\ansi.sys

In Brief

To change screen attributes, use the ANSI set graphics mode (**m**) escape sequence. for example, the following command makes text bold:

```
ESC [ 1m
```

This command makes text magenta and bold:

```
ESC [ 1 ; 35m
```

The **m** escape sequence can have as many parameters as you want.

There are three kinds of screen attributes: text format, text color and background color. The text-format attribute specifies whether text is bold, underscored, blinking or hidden. Text-color attributes specify text color, Background-color attributes specify screen color. If you specify an attribute that your system doesn't support, MS-DOS ignores the setting and no change occurs.

You can use the ANSI set graphics mode (**m**) escape sequence to change the appearance of your screen. After you set an attribute, any new text displayed has that attribute. For example, the following command makes text bold:

```
ESC [ 1m
```

By default, text is white and the background is black. To return to the default setting, use 0 as the parameter, as in the following example:

```
ESC [ 0m
```

Changing Text Color

The following attributes specify text color:

```
30 Black
31 Red
32 Green
33 Yellow
34 Blue
35 Magenta
36 Cyan
37 White
7 Black text on a white background
```

The first ANSI escape sequence in the following example displays the word *WARNING* in red, bold, blinking letters:

```
ESC[31;1;5MWARNING
ESC[0m
```

The second escape sequence resets the attributes so that new text will have the default settings.

The following **prompt** command makes the drive and directory in your command prompt bold and blue, and the greater-than sign bold and red:

```
prompt $e[1;34m$p$e[31m$g$e[0m
```

Changing Background Color

The following attributes specify background color:

```
40 Black
41 Red
42 Green
43 Yellow
44 Blue
45 Magenta
46 Cyan
47 White
7 Black text on a white background
```