GENEVE
FAQ

05.19.2001

Compiled By Dan H. Eicher

Contributions by:

Jerry Coffey
Charles Good
Tony Knerr
Ken Hastins
Bill Sullivan
Tim Tesch
Jeff White

Table Of Contents:

GENERAL [GFG000]:

WHAT IS A GENEVE [GFG001]?

The Geneve was a TI-99/4a upgrade produced by Myarc Incorporated of Basking Ridge, New Jersey. The Geneve is sometimes refereed to as a Myarc Geneve 9640 or Myarc 9640. The story behind the name.. Originally, Lou Phillips (President and Owner) wanted to call the machine something with 99, 990 or 9900, Texas Instruments asked him not to. So, he came up with the name 9640. 9 for the 9900 chip family and 640 for the amount of memory. Later, a programmer working on some of the original software told Lou, that a computer should have a name, not a number. This as they were working up the stairs at Lou's home, they spotted a painting on the wall. The name of the painting was Geneve - so the name stuck. The main differences between a Geneve and a TI-99/4a are:

Geneve	TI
Plug in expansion card	Stand alone console
9938 VDP	9918 VDP
9995 CPU	9900 CPU
Full XT style keyboard.	Slightly compressed keyboard built into console.

WHAT IS A GENMOD GENEVE [GFG002]?

A GENMOD Geneve is a standard Geneve that has a third party modification, that allows the Geneve to run at zero wait states, effectively doubling performance. The GenMod is a small daughter board that mounts to the Geneve. This modification was created by Dynamic Memory System (Ron Walters) and sold through Bud Mills Service. This modification MUST be used with a Memex card.

IS MYARC STILL IN BUSINESS [GFG003]?

No. Myarc ran as a full time business, until the late 80s, for a couple of years after that, Lou Phillips, attempted to run it as a side business. This didn't work so well as other demands for his time and attention out weighted Myarc business. The company was shutdown in the early 90s in good order. After the shutdown, remaining parts and inventory were transferred to a company called Cecure electronics, ran by Don Waldon. Don Waldon was active in the TI community, serving at one point as president of both the Chicago and Milwaukee users group. Finally, in the late 90s Cecure disappeared leaving many people holding the bag.

DOES ANYONE DO REPAIRS ON GENEVES [GFG004]?
You might try :
"Tim Tesch" <insane_m@hotmail.com>
Tim used to work for Cecure doing repairs. Tim sometimes does repairs as he has time. Many times repair is not possible, because the components that are bad are no longer available.

As a last resort, you might try Richard Bell <swim4home@aol.com> - he has been known to look at Myarc equipment on a best effort basis.

CAN I GET A COPY OF THE SCHEMATICS [GFG005]?
???

WHERE CAN I GET THIS, THAT OR THE OTHER PROGRAM [GFG006]?
Since the Geneve has been out of production for many year that availability of software and vendors very from year to year.

One vendor that has kept a good supply of TI and Geneve software is Ramcharged computers.

MDOS and GPL are freely available, check ftp.whtech.com.

IS THERE AN EMULATOR FOR THE GENEVE [GFG007]?
No. If you would like to write one, I'm sure there would be a couple of dozen interested people. At one time, a programmer in Canada got the Geneve to emulate a Coleco Adam and Barry Boone got the Geneve to run cartridges written for the Tomy Tutor. These conversions are available from RamCharged Computers.

WHERE CAN I FIND ADDITIONAL INFORMATION [GFG008]
A lot of documentation both TI and Geneve can be found on FTP.WHTECH.COM - This is service is provided free of charge to you by Don O'Neil. You can find Geneve Manuals, MDOS source code and hopefully soon the entire collection of 9640 NEWS (Courtesy of Beery Miller).

In the world of Linux the rallying cry is: "If you want to know look at the source code." - The same can be said for the Geneve!

9640 News - This was a Diskazine put out for a number of years by Beery Miller. It includes articles, news, original programs, tutorials. - Probably the best source of Geneve specific information.

Micropendium - Scans of this can be found on ftp.whtech.com. These are also being added as time permits to the CYC project in searchable PDF format. The CYC can be purchased from CaDD electronics. http://www.netcom.com/~mjmw.

MDOS [GFM000]?

WHAT IS MDOS [GFM001]?
MDOS is the native operating system of the Geneve. MDOS presents the user with a command line environment very similar to Microsoft MSDOS. The major feature one must get used to, is that MDOS maps certain drive letters to device, this happens automatically, but can be over written with a remap command. As an example A: = DSK1. For details, please see the documentation that accompanies MDOS. MDOS includes a portability layer called GPL MODE, (MDOS is sometimes called NATIVE MODE). Once GPL is invoked the Geneve functions very much like a TI.

WHAT IS THE CURRENT VERSION [GFM002]?
As of the writing of this document - 6.0A for MDOS and 6.0 for GPL - These two file should be used together and not mixed and matched with other versions.

IS THE SOURCE CODE AVAILABLE [GFM003]?
Yes, thanks to Beery Miller. Beery was a Geneve advocate who published a diskazine, called 9640 news. Paul Charlton wrote MDOS a senior electrical engineering project, it was also a work for hire program written for Myarc. At some point in the

relationship, Paul Charlton felt he had not gotten all of what Myarc owed him and he stopped producing upgrades and fixes.

At this point Beery got involved and with Lou Phillips blessing led a MDOS buyout. MDOS users contributed from 25+ dollars. Once enough money had been collected, the check was given to Paul Charlton, who gave the source code to Beery. Berry actually flew to New York to get the code in person.

Many different people have contributed to MDOS's development. For the last eight years, the primary maintainer has been Tim Tesch. The source code to MDOS is available at ftp.whtech.com.

NOTE! MDOS is probably the single largest program EVER created in the TI community! In order to set this program up for compile you must have either a SCSI controller or HFDC with at least a 5 MEG hard drive hooked up! =)

HOW DO I MAKE A CONTRIBUTION [GFM004]

Ok.. You have MDOS setup and compiling. Congratulations, treat yourself to a beer, your have earned it! If you come up with a fix, patch, hack, update etc.. please send a copy to Tim Tesch for inclusion in the next official release.

IS THEIR A GUI AVAILABLE [GFM005]

Yes, a number of them.
One of the prettiest - but never finished was GEME by Lou Riccio.

Windows9640 - This is a Graphics Subsystem, that can either be used as full GUI or used in a runtime version with applications, written to its API - one example is CFORM. Written by Beery Miller.

GenBench - Written by Al Beard. This emulates a program launcher similar to one on the Amiga.

WHAT WERE SOME OF THE MILE STONE RELEASES [GFM006]?

4.0 - Added SCSI support, this version and all following, MUST have an extra 32K static ram chip installed to function.
6.0 - Added year 2,000 support.
6.0A - Added PDMA

MS-DOS HAS AN ANSI.SYS DRIVER, DOES MDOS [GFM007]
Yes and No.. Paul Charlton had a good set of ADM3A emulation routines that he had written for his terminal emulator Fasterm, so he included these routines in MDOS. On FTP.WHTECH.COM there is a copy of ADM3A manual which includes command codes. 9640 News also included some good tutorials on creating batch files with control codes in them.

ANSI - Tim Tesch and Mike Maksimik wrote a really nice ANSI viewer that you can take just about any ANSI control file and display in MDOS. Note, this is not a driver, but a stand alone programs. If you want to do control characters in your batch file you must use the more limited ADM3A.

===

SETUP ISSUES [GFS000]:

WHAT IS THE LATEST GENEVE BOOT EPROM [GFS001]?
.98 - regular Geneve.
.98m - GenMod Geneve.
Also, if you have a Geneve with a PFM modification,
you get much better boot control, you can boot from
PFM memory, or press the spacebar and load from
another device.

WHAT CARDS CAN BE USED WITH A GENEVE [GFS002]?
TI, Myarc and Corcomp Disk Controllers.
Myarc HFDC

- This card can support 1.4 megabyte drives, if it has a 9216B data separator chip.
- Some cards have a 9216 (no b) and are not reliable at densities greater than 720K.

Horizon Ramdisks.

Rave Ramdisks.
Myarc 512K card (*)
This card required modification to work in a Geneve environment.
Rave Speech Adapter.
TI, Myarc and Corcomp RS232s.
WHT SCSI Controller.
SNUG SCSI Controller.
MEMEX - Memory Expansion (*)
Defaults to 512K - you must have the GenMod to go up to 2MEG.

CAN I USE 1.2 MEG FLOPPIES ON AN HFDC IN A GENEVE SYSTEM [GFS003]?
The HFDC could be modified to support 1.2 Meg (500kbps, 360rpm, 80-track) drives. The density select inputs to the floppy data separator (SMC9216B) would have to be decoded to give the density select signal to the 34-pin connector. Most, if not all, 1.44 Meg (500kbps, 300rpm, 80-track) drives do not require the signal.

In reality, the density select signal is a speed select (300/360rpm). A 500kbps, 300rpm drive, be it 3.5" or 5.25" would appear the same to the controller (a 1.44Meg drive). Unfortunately, the coercivity of 5.25" high density media is not sufficient to store the same amount of data as 3.5" high density media. In fact, 300/360th (5/6) of 1.44 is 1.2. It is the faster speed (360 rpm) that brings the 5.25" high density media to usability at 500kbps.
Jeff White

From: "Jerry L. Coffey"
These analogies are useful for understanding the gross characteristics of IBM 80-track formats, but the reality is a lot messier.

1.2M -- This is formatted to 1,213,952 bytes in 2371 allocation units of 512 bytes each. NOTE that 2371 is not evenly divisible by 160 (which is the total number of tracks on both sides of the disk). Right! The number of sectors per track is DIFFERENT on the outer and inner tracks. So the formatting code has to change parameters as the head moves inward. Messy stuff!

720K -- This one also uses 96 tracks per inch ("80 tracks per side) but there have been two different versions. The first (up to DOS 3.x) used 512 byte allocation units, but then at about DOS 3.3, this was changed to 1024 byte allocation units (which could give the same capacity with fewer total bytes per track, but I think there was also a change in the head switching pattern at the same time).

Anyway, the 3.5" formats (at 135 tracks per inch) are a whole bunch simpler and more rationalized than the many IBM 5.25" formats.
Jerry L. Coffey, PhD. Mathematical Statistician

WHAT TYPE OF KEYBOARD CAN BE USED WITH A GENEVE [GFS004]

Any that is XT compatible. It is recommended, you use one that has an XT/AT switch, so you have access to Function-11/12, as some software likes to use these keys and they are not available on XT only keyboards.

WHAT TYPE OF MOUSE CAN BE USED WITH A GENEVE [GFS005]

The Geneve has a mouse port built in. Myarc produced plug in mice. This is NOT a serial mouse. There are instructions for modify a Logitech Bus mouse. You can also use a modified serial mouse hooked to the RS232. In both cases you will need a three button mouse as all Geneve GUI software requires the third button. Also, there are two different drivers, depending on which mouse type you use. Al Beard updated the Geneve mouse driver, adding extra API's for use with GenBench. The serial mouse uses a more basic API, it was written by Bruce Helstorm.

WHAT PRINTERS CAN BE USED WITH A GENEVE [GFS006]?

The Geneve will support any printer that the TI will at the application level. MDOS does include a company that will allow you to do screen dumps, printers supported by this are ????. MDOS also supports a built in print spooler that is XXK and is active in both native MDOS and GPL mode.

HOW DO I HOOK UP MY SPEECH SYNTHESISER [GFS007]?

You will need to use a RAVE Speech adapter. If you have problems with garbled speech, you may need an upgraded PAL available from Richard Bell. A Corcomp TripleTech card will NOT work, neither will the speech adapter designed by Winfried Winkler of the Berlin Users group.

HOW DO I HOOK UP MY MODULES [GFS008]?

Since the Geneve is a card to be plugged into the pbox, it doesnt have a cartridge slot. Because of this it is necessary to first dump all of your modules to disk before removing your TI and Flex/Interface cable. To help you do this Myarc provided a utility called CSAVE to dump modules to diskette.

WHAT KIND OF MONITOR CAN I USE [GFS009]?

The uses the 9938 Video Display Processor. The Geneve natively supports two types of monitors composite and RGB, but there are at least two other options.

OPTION 1:

A composite monitor you should be able to some what view 80 columns. If you use a color composite monitor, about the best you will be able to do is run things in GPL mode (40 column). Note MDOS can run in 40 column mode (issue the command MODE 40).

OPTION 2:

An analog TV grade RGB monitor. At one time, this type of monitor was common as it was used on Commodore 128s, Amigas, COCO 3s and Ataris. One of the more common model used by Geneve fans is the Magnavox 8cm515s. As of this writing some Kodak TV grade RGB monitors were available from Rochester Electronics (phone number).

OPTION 3:

The Video Turtle - this project was created by Jim Krych a TI and Amiga enthusiast. He created this device as he saw the pool of TV grade RGB monitors were being depleted. Basically the device takes the RGB out signals and converts them to S-Video. You can use any S-Video TV. Some are better than others. The unit I saw demonstrated was a wide screen (25) and the 80-columns out from a Geneve looked pretty good. The Turtle is sold through Texacomp LTD. (phone)

OPTION 4:
Jim Uzzel has successful interfaced an PC VGA monitor to his Geneve with a commercially available converter. Plans can be found (???)

WHAT IS AN EMULATE FILE [GFS010]?
By Bill R. Sullivan
An Emulate file, emulating DSK1 is an HFDC "thing". Emulate files can be created with MDM5 or with T & J Software's Emulate program for the Geneve. Once the binary file is created (it includes all the files that you copy to it up to file created size-max normally 2880, 3000 or so on Geneve) it can be set to active by MDM5 or T & J Software. The file will then be accessed as DSK1 as long as it is set to active. Many such files can be created on HDS1 only, but only one can be active at a time.

During emulate activation the physical DSK1 drive cannot normally be accessed--there are ways around this.

Tim Tesch writes:
The modified emulate creation utility I released allows you to make the files as large as 12,800 if memory serves correctly. One such emulate holds all of my Infocom and Adventure games :)

The Emulate 'bit' should ONLY be turned on by MDM5 or the Emulate programs for the Geneve. I don't believe I masked that bit when I wrote BITRESET - my goal was to eliminate the archive bit only.

As for Ken's assumption regarding the archive bit; yes, MDOS does change the status of this bit during file updates and possibly copies. I forget the current 'rules' MDOS follows.

BOOT UP [GFB000]:

Boot up is controlled by the Geneve's boot EPROM (controlling boot up is about all this EPROM does). Note the swan you see was designed by a young art student. She was senselessly killed in a subway shooting in the late 80's.

The document below is by Jeff White and explains how the Boot EPROM on the Geneve works:

The Geneve boot EPROM first determines if the floppy controller is a TI, CC, Myarc or other. If it is a TI, CC, or Myarc floppy controller, the built-in boot EPROM routine does sector i/o directly to CPU memory, not using the DSR on the card for sector i/o -- only the Myarc cards have direct CPU transfer capability and the timing issue.

If the floppy controller is "other" (like an HFDC), the sector i/o routine of the card's DSR is supposed to load a version of "LOAD/SYS" that will load "SYSTEM/SYS" from the device.

The "LOAD/SYS" file must be in the WDS1.DSK1. directory, an active DSK1-emulation or on floppy disk in drive DSK1. If the LOAD/SYS file is found on WDS1 (hard drive), the SYSTEM/SYS file will be read from WDS1.SYSTEM/SYS. If found on floppy, DSK1.SYSTEM/SYS is loaded.

Tim's trick to put a "LOAD/SYS" renamed "SYSTEM/SYS" on floppy to boot from SCSI was used because DSK1 directory- or file-emulation is not in the SCSI DSR. If booting from a recognized floppy controller, SYSTEM/SYS is the first file loaded.

I think that is a correct summary. "Deficiencies" in the DSR code (only VDP transfers or timing issues) are why the Geneve boot EPROM has its own sector i/o code. If the card is not recognized, but has DSK1 in its device list, the boot EPROM will load DSK1.LOAD/SYS (DSK1 directory or file emulation, or floppy) that should load SYSTEM/SYS from the device.
Jeff White

CAN I BOOT FROM ANY FLOPPY [GFB001]?

By default the Geneve's boot order is???
MDOS will only look on DSK1 - it will not search any other drives. The program must be called "SYSTEM/SYS". Note versions of MDOS later than 3.0 REQUIRE at least a Double Side or Double Density drive a TI Single Sided/Single Density drive is not large enough to hold later MDOS's.

CAN I BOOT FROM A HARD DRIVE [GFB002]?
(VERY SHORT FORM)
Yes, but you must do the following:

    Create a directory called DSK1. on your hard drive.
    Place a program file called LOAD/SYS (available from
        ftp.whtech.com) in HDS1.DSK1.
    Place SYSTEM/SYS in your root directory.

(LONG FORM TO SETUP FROM THE BEGINNING)
By Jeff White
First, to use an HFDC with the Geneve and a current (2.x-6) version of MDOS, it should be set to CRU >1000 (4-DIP in all ON positions) or CRU >1100 (4-DIP with 1 switch OFF, 2-4 switches ON). If you have a floppy controller other than the HFDC in the system, the HFDC should be at CRU >1000.

The HFDC EPROM version should be H11, which can be checked with MDM5 on a 4A system. There are some bad versions of H11 out there, and no way to easily tell if you have one. Other releases (H8-H10 and H12) are less certain than H11. All versions have bugs, but the H11 is the best bet. It is possible that the EPROM is labeled with a version. However, the number is often noted in hex- -- H10 labeled as "HA" and H11 labeled as "HB".

Format the hard drive on a 4A system with MDM5 (v1.41 is as good a choice as any) to 32 sectors per cylinder. For a 251, your 8x620x32 settings are correct. Accept the defaults for reserved directory sectors, etc.

Assuming successfully formatting, copy a few files to the root directory (ONLY directory) on the hard drive. Catalog the hard drive, delete the files, and swap the 4A out for a Geneve.

Boot the Geneve from floppy:
(1) HFDC at CRU>1000, another floppy controller in system -- Put floppy disk with SYSTEM/SYS v2.x-6 in DSK1, turn system on, wait for up to 1 minute to see the MDOS prompt. If it does not boot to the prompt, something is wrong.

(2) HFDC at CRU>1100, floppy drive attached as DSK1, 20-pin cable attached to HFDC, but 34-pin edge connector detached -- Put floppy disk with LOAD/SYS and SYSTEM/SYS v2.x-6 in DSK1, turn system on, wait for up to 1 minute to see the MDOS prompt. If it does not boot to the prompt, something is wrong. If it did boot, connect the 34-pin edge connector and ribbon. Warm boot the system with Control-Alt-Del. [This last step is done in attempt to get MDOS to recognize the correct allocation unit size of the hard drive.]

Assuming that LOAD/SYS and SYSTEM/SYS were the only files on the boot disk, you should be at the A-prompt and HDS1 is the E-drive. Execute each of the following MDOS commands by typing it exactly on a line and pressing the enter key.

```
   E:
   MKDIR DSK1
   COPY A:SYSTEM"/"SYS
   CD DSK1
   COPY A:LOAD"/"SYS
```

If you get any error messages during any of the commands, something is wrong. Remove the floppy disk from the drive. Turn the system off and back on. If the system boots, you are now ready to risk your important data to the will of the HFDC. If the system does not boot, something is wrong.

WHY DOES BOOTING FROM A HARDDRIVE REQUIRE LOAD/SYS [GFB004]?
From: "Jerry L. Coffey"
As I recall the decision was based on space considerations. The HRD code was shoehorned into the EPROM code, but support of the

HFDC blew the byte budget, both for the code in SYSTEM/SYS and the space for boot code in the EPROM.

So a tiny snippet of code was added to the EPROM to call a second stage (external) routine called as DSK1.LOAD/SYS.

CAN I BOOT FROM A RAMDISK [GFB004]?
Yes, you will need to get a program called FORM3MEG. This will format Horizon Ram Disk and allow them to boot. Note: In order to setup a Horizon to boot, it can only be 800K (if any larger it will not boot). Horizon series 3000 can have a modification added called the Phoenix MOD (this is built into series 4000 boards) - this will effectively, take a large Horizon and create a 360K boot drive and leave the rest for larger drives.

WHAT IF I WANT TO BOOT FROM A LARGE RAMDISK [GFB005]?
(OR SOME PEOPLE ARE NEVER HAPPY!)
By Tony Knerr
Ok, here's how to make a Horizon over 800K bootable.

The boot EPROM on the Geneve only has routines for up to Quad density disks, so we will have to "fool" it when reading our "High Density" ramdisk so it can find the proper sectors where SYSTEM/SYS is located.

Format the ramdisk with FORM3MEG, choose Y for "Set disk to boot MDOS" and Y for "Load SYSTEM/SYS from disk to ramdisk" and go ahead and load the file on the ramdisk.

Now that SYSTEM/SYS is on the Ramdisk, you will need to find the File Descriptor Record for it. Load up "Disk Utilities" or some other sector editor. The sector will begin with the filename and will probably be sector >4 or >8. Write this sector down.

You'll also need to find the first actual data sector of SYSTEM/SYS. This sector includes the string "reassembling". Write this sector down also. It will probably be >200.

Now for the thinking part. We will be editing bytes >1C, >1D, and >1E of the sector with the FDR (either >4 or >8, you DID write that down, didn't you?) We will be interested in the 6 nybbles, or digits, of these three bytes. Let's number them 1 through 6 starting at the left digit of byte >1C.

Here's what the Nybbles mean:
Nybbles 4, 1, and 2, in that order, are the starting sector of the file.
Nybbles 5, 6, and 3, in that order, are the number of sectors of actual data in the file.

So if bytes >1C, >1D, and >1E are >40, >F0, and >1F, the FDR is telling us we have a file that starts at sector >040 and the data in it is >1FF sectors long (for any MDOS over version 2.50, at least). Well the length is ok, but the Geneve will load the wrong data, because it thinks the file starts at sector >040 instead of sector >200! You'll get a lock-up, because the wrong data was loaded in ram where SYSTEM/SYS is supposed to be.

So lets fix it. If we edit Bytes >1C, >1D, and >1E to be >00, >F2, and >1F, the we are telling the EPROM that the file starts at sector >200 and is >1FF sectors of data in length, where the EPROM will find the correct data.

If you've done everything right, the Geneve will now boot from the ramdisk! If not, you'll have to disable the ramdisk via the on-off switch, reboot from floppy, turn the ramdisk back on, then start over and do it correctly the next time.
It DOES work, I've been doing it for years.

For a 1.44m floppy, copy SYSTEM/SYS onto a "clean" disk. The FDR will be at sector >4. Edit Bytes >1C, >1D, and >1E to be >00, >F2, and >1F (again, this is for MDOS version 2.50 and greater).

Copy LOAD/SYS onto the floppy after you have copied SYSTEM/SYS.
The first six bytes of data for LOAD/SYS are :
>0000 >06DC >A000. You'll find these at the start of sector >400.

The FDR of LOAD/SYS will be at sector >8. Bytes >1C, >1D, and >1E will need to be changed from >00, >71, >00 to >00, >74, >00. The floppy should now be bootable. The same rules pertaining to SYSTEM/SYS as stated in the next paragraph also apply to LOAD/SYS on the 1.44m floppy.

WARNING! Do not perform any file operations such as copy, move or delete on the SYSTEM/SYS file after you have made these changes! You can copy other files to or from the disk, but file operations on SYSTEM/SYS will corrupt all sorts of things on the disk, just don't do it!

The information which enabled me to come up with this procedure can be found on pages 61 through 63 of the HFDC manual, "File Descriptor Records".

If you don't understand the above instructions, just ask and I'll try and address any problems you may have in this procedure.

CAN I BOOT FROM A SCSI DRIVE [GFB006]
Not at this time.

IS THEIR SUCH A THING AS A BOOT EPROM [GFB007]?
Yes and No. Initially Myarc advertised or stated, eventually MDOS would be stabilized and burnt into an EPROM. This never happened. Later Jim Schroeder who wrote FORM3MEG created the software, and Don Waldon came up with the design to use a FLASH EEPROM to function as a many RAMDISK, directly on the Geneve. With this package, one can load a current version of MDOS into the flash EEPROM and boot from it.

WHAT IS A JUMP BOOT DISK [GFB008]?
DR. Jerry Coffey devised a method to use a floppy to load at the maximum speed supported by all the pieces of a floppy subsystem CPU, Disk Controller, Floppy Drive etc. by optimizing the writing and placement of the SYSTEM/SYS file. The results is a floppy that will boot MDOS at about the speed of a ramdisk, and usually quicker than an HFDC. These disk were sold commercial.

From: "Jerry L. Coffey"
The most enthusiastic users of JumpBoot were developers who needed a fast robust method of rebooting while chasing bugs that made some parts of the system unavailable. Al Beard was one of them.

By the way, the last release of JumpBoot included a modified utility that would install the long (Hard-disk) version of SYSTEM/SYS on the disk (at a slight loss of boot speed) for those occasions when the boot files or other portions of the hard disk are corrupted and you have to fall back to floppy.

The 9640 floppy boot routine does a binary search for the SYSTEM/SYS file. The starting point is the largest power of two that is less than the total number of files (= number of pointers in sector 1).

So if you want a hit on the first try, put SYSTEM/SYS in that position (by juggling other filenames). I did this on the Jump Boot distribution disk to impress the troops the first time they ran it. <grin>
Jerry

HOW CAN I TELL OF MY MDOS FILE IS GOOD [GFB009]?
Tim Tesch has written a program called CRaCkIT that will read through a SYSTEM/SYS file and will compute a CRC value. This CRC value will then be compared to an original CRC value that is embedded in that release of MDOS. CRaCkIT will then tell you if your SYSTEM/SYS is corrupt.

=-

COMPATIBILITY [GFC000]:
WILL ALL 99/4A PROGRAMS RUN ON THE GENEVE [GFC001]?
The answer is Yes and No!
In general, yes - but their are a few gotchas....
Some of the most common:
1. On a TI, the scratch pad area is 256 bytes long and responds at multiple address in the >8000 range. On the Geneve, this 256

bytes is not shadowed. So if someone was careless and used one base address, and then switched to another and expected the same data to be there, they are out of look.
2. Writing code "to the bare metal" - an example would be someone that wrote code that speaks directly to the 9901 to read the keyboard instead of using KSCAN, these programs would fail.
3. People that got sloppy with GROM Base Address.

Some of these programs have been fixed. Mike Maksimik who wrote the SCSI code for MDOS fixed many modules to run on the Geneve.

=-

MISC TIPS [GFM000]:

TRANSFERRING SOFTWARE BETWEEN A PC AND A GENEVE [GFM001]?
Any floppy drive hooked to a Geneve can be put into 80 track mode with the command (). This is especially handy if you have a Myarc/Corcomp controller and a program called PC-Transfer (RamCharged) as this will allow you to write a 3.5 720K MSDOS floppy which can be read on a 1.4MEG PC drive.

NOTE: The MYARC HFDC IS NOT SUPPORTED BY PC-TRANSFER!!!

Here are some notes from Bill Sullivan on how to do this:
Configure Corcomp to DSDD for all drives that support 80 tracks.
In your AUTOEXEC file put the following statements:
SETDSK 108 (for DSK1. - most 80 track drives will work with "0" for step rate & 8 is for 80 tracks) SETDSK 208 (for DSK2., etc.)
To use PCT with 720K PC diskettes, format the diskettes on a PC!!!   The PCT FORMAT utility only formats to 360K.

I'VE COPYED A FILE FROM MY GENEVE TO A DISK ON MY TI AND NOT IT WON'T RUN OR DELETE [GFM002]!
You have ran afoul of the Geneve's archive bit!!! Myarc decided to use an unused bit in the file descriptor to determine if a file had ever been backed up or not - just like in MSDOS. According to TI spec's this bit was reserved, so programs should mask it's value. Unfortunately, many DSR's do not mask the bit,

so what you end up with is a file descriptor that screws up many programs.

From:"Bill R. Sullivan"
Corcomp & BwG FDCs, along with Ramdisk have the problem when the archive bit set. The TI FDC has a lesser problem with it. Myarc FDC & HFDC do not have any problems with it. Therefore, many believe it to be a Myarc created "bug" for their non-Myarc FDCs and HRDs.

Jeff White points out..
Bill, pp62,67 of the HFDC manual, 3rd ed, give the bit definitions of the file status flags of floppy and hard disk File Descriptor Records. Bit 4 of the file status flag is the backup bit that could be causing your trouble.

With a sector editor, you should be able to tell if it is the potential source of the problem.

Byte 12 of an FDR will normally be >0x or >8x, for fixed or variable length files. If byte 12, the file status flags, has a value of >1x or >9x, the backup bit is set.

A DSR that incorrectly handles the FDR byte 12 could cause problems.

You can check the file status flags with a BASIC program, as well:

```
  100 OPEN #1:"DSK1.",INPUT,RELATIVE,INTERNAL
  110 IF EOF(1) THEN 200
  120 INPUT #1:F$,A,B,C
  130 PRINT F$,A,B,C
  140 GOTO 120
  200 STOP
```

The F$ variable is the filename, of course. A, B, or C is the file type -- only one of them, but I do not recall which. The file type may or may not be the same as the file status flags

byte in the FDR. I.e., the flag/status byte(s) used in standard PAB access are not bit-for-bit equivalent to the FDR byte 12.
Jeff White

Your probably thinking, that all this conversation is well and good, how do I fix it, if your handy with a sector editor you can do it that way, or...

The program is called "Backup Bit Remover".
I think the archive was called "!BITREMOVR" or "!BBR"

It is a very small program; I think I even released the source code with the program.
Tim Tesch, S&T Software

WHY CAN'T MY HFDC READ SOME SINGLE DENSITY FLOPPIES, IS THIS A BUG IN MDOS [GFM003]?
No, this is a well known problem. The Corcomp disk controller, does not add enough gap bytes for all controllers. The HFDC's disk controller chip is very sensitive to this. There is a way around this problem though...

Bill,
Here's something for what its worth. When Mike Dodd was perfecting the format routines in Hypercopy, he spent months tearing his hair out trying to make things work correctly on the Corcomp card. Even when he tamed the problem for my early CC card, another beta tester with a later card was still having problems.

I had always assumed that the low-level formatting code supplied by CC was perhaps a bit sluggish and that was why they ran out of track before all the bytes in the recommended track image were written. But Mike was controlling the WD1773 directly, so maybe it was a quirk of the chip.

Anyway, with some help from Paul Charlton, Mike succeeded in his goal of formatting, reading and writing compatibly between any of the three controllers (Myarc, CC, and TI) in all supported formats.

So if you have a copy of Hypercopy, use it to format and copy the disks on the Myarc controller. You may be able to read the result on the WD1773 card. No guarantees, but a lot of beta testing suggests it will work.
Jerry L. Coffey, PhD. Mathematical Statistician

I HAVE A SERIAL PRINTER, HOW CAN I GET PRINT SCREEN TO WORK [GFM004]?
You'd have to rewrite a portion of the screen print routine within MDOS. The XOP which handles Screen prints is hard-coded to used "PIO.CR" as its output device. There is no padding, so the best you could do would be to sector edit this to "RS232." If you have the MDOS source code, check file "L6\VX2B2CS". This is where the change should be made....
Tim

==

MDOS PROGRAM DEVELOPMENT [GFPD000]

WHAT TOOLS ARE AVAILABLE TO DEVELOPMENT NATIVE MDOS APPLICATIONS [GFPD001]?

Abasic
This is native MDOS mode interpreted language. It is 40 or 80 column and includes a rich set of features. Jim Uzzell the maintainer of Abasic has written many power applications in Abasic. Abasic is a pure assembly program, since it is running on a much quicker machine compared to a TI-99/4a, you can expect speeds equal to programs written in the GPL language on a TI-99/4a.

GenProg
This was a commercial package released by Paul Charlton, included an extensive linker, librarian and single pass assembler.

MDOS Assembler
This was a shareware program, released by Clint Pulley and Al Beard, it allows the user to write native MDOS programs in assembler - it also needed for the following package.

TIC
This is a mostly K&R compliant C package. It is very powerful in the write hands, Mike Maksimik wrote CFORM - a power graphic hard drive format utility in this language.

Fortran9640
This is a commercial package written by Al Beard as a big brother to his Fortran99 package. Very polished and professional. Al used it to write an 80-column Fractals package and several sector editor type utilities.

HOW MUCH GRAM MEMORY DOES THE GENEVE HAVE [GFPD002]
From: Jeff White
Subject: TI99: Geneve GRAM Banks.

The Geneve has 8 pages of 8K RAM dedicated to GRAM in GPL mode. These are pages >38-3F. The two ROM banks (flipped with writes to >6000 or >6002 in write-protected mode) occupy pages >36 and >37. Thus, the Geneve can handle 16K cartridge ROM (per XB-type banking) and 64K GRAM. It would be possible to load an MDOS utility that would put several cartridges in memory that could be swapped in and out via software in GPL or MDOS modes, but such a system would not allow both cartridges to be available at once (like having TE II speech available from XB on the SNUG SGCPU system or CaDD PC99).

...Of course, hooks could be put into the GPL interpreter to copy GRAM's in and out of >38-3F, but that would slow things down. Another option is to write a GPL interpreter that runs GRAM from RAM in native MDOS mode.

WHAT BANK SWITCHING SCHEME DID MYARC USE TO ALLOW MORE THAN 64K [GFPD003]?
Tim Tesch writes:

Take a deep breath and plunge into the waters of the geneve memory map! I'll do my best to keep the waters clear and temperature "just right!"

Mapper "Page"	Memory Range
0	>0000 - >1FFF
1	>2000 - >3FFF
2	>4000 - >5FFF
3	>6000 - >7FFF
4	>8000 - >9FFF
5	>A000 - >BFFF
6	>C000 - >DFFF
7	>E000 - >FFFF

The Geneve has a TOTAL addressable memory of 2 MB, allocated in 8K chunks called pages. Each page is 8K of the total 2MB, giving us 256 mappable pages. The mapper, eight contiguous 1-byte values, determines which of the 256 memory pages are located in the 64K area, pages 0 through 7, as shown above. This 64K is what is termed the "visible" memory range as all pages must be mapped here to be "seen."

How does mapping work? Mapping memory can be thought of as an 8-CD Changer. Your CD library may contain up to 256 CDs, but you can only put 8 CDs into the changer at one time. The changer (mapper) does not know you have more than 8 CD's (pages), and it is quite content to play those 8 CDs over and over and over again. To play something else, you need to change the CD (page) in the changer (mapper).

Only one page may be mapped into the a mapper page at a time; i.e., you could not have page >a0 and page >B0 mapped into the mapper slot 0 (>0000->1FFF). trying to do so would be like placing two CDs into a single slot in your changer - they both won't fit!

However, you may map a single 8K page into all 8 mapper slots, resulting in "copies" of the same 8K of data in all 8 mapper slots - you could do this with CDs if you had 8 copies of the same CD! (For a more picturesque look at the mapper and pages, run MEMTEST on the geneve.)

So, if my mapper said I should have pages 0-7 listed as follows
>20,>21,>22,>23,>24,>40,>41,>42,

page >20 would be mapped into >0000-1FFF. page >40 would reside at >A000-BFFF and so on.

That aside, the Geneve only loads these 8 slots with a program image file. This, like the TI, is based upon the first three words of each file. The address portion of the file is only 1 word, which means a maximum address of 64K, another "limitation" imposed this time by the software, not the hardware. The other two words dictate the starting position (also 64K limitation) and the file type.

The Geneve, like the TI, can use overlays. With the Geneve it is a little simpler as the modules can be stored in CPU memory. How? The program can reserve as little as 8K or as much as whatever the system has free - the programmer must decide this. The program can then load the overlays into the reserved memory. Or, like the TI, can load them from disk and place them in the right spots.

To use our CD changer analogy, this is comparable to having only 1 CD left in your library because other people (programs) had borrowed your CDs (pages of memory). The more CDs in your library, the more possible to swap out; a Geneve with 2MB of memory can swap more pages than one with 548K.

MDOS loads a bit differently. The EPROM and LOAD/SYS both load MDOS into contiguous 8K pages, starting at page 0, until MDOS is loaded. For MDOS 5.00, pages >00 through >1F are used - 128K. MDOS is then "started" and it unravels itself a bit, takes up a few buffer pages, and moves some code around. This is why MDOS, though only 128K in size, takes up much more.

HOW DO I ACCESS SPEECH FROM NATIVE MDOS MODE [GFPD004]?
Jeff White writes:
Accessing speech is possible on the Geneve in native mode. Recall that the speech addresses are mapped to >9000-97FE (even bytes) on the 99/4A bus. Note that the 8K pages corresponding to 99/4A bus memory are >B8, >B9, etc. This gives the well-known result that the EPROM DSR pages fall at page >BA. That is, page >BA corresponds to 99/4A memory >4000-5FFF. Of course, on the Geneve, page >BA can be on any 8K boundary or not mapped at all.

Speech is within the >8000-9FFF memory of the 99/4A. This external bus memory corresponds to page >BC on the Geneve. Thus to access speech while in MDOS mode, map page >BC to any 8K page within >2000-DFFF. The offset within the page will be >1000-17FE. If I remember correctly, only four addresses are needed for speech (>9000, >9002, >9400, >9402 on the 99/4A).
These are offsets of >1000, >1002, >1400, and >1402.

If you use mapper register >F111 to put page >BC at logical >2000-3FFF, the speech ports corresponding to >9000, >9002, >9400, and >9402 on the 99/4A would be >3000, >3002, >3400, and >3402 for Geneve native mode.

If you want to map page >BC so that you can use the same addresses as you would on the 99/4A, use mapper register >F114.

Using page >BC also works with the ForTI music card. It normally maps to >8400-87FE on the 99/4A, and >8600-87FE on the Geneve in GPL mode. This address space is also used by the MBP cards and other so-called clock cards. Again, the cards can be accessed in Geneve native mode by simply calculating the offset in the 8K logical memory page that physical page >BC occupies.

Hope this explains it.

HOW FAST IS THE GENEVE COMPARED TO A TI [GFPD005]
By Jeff White

The 9900 in the 99/4A works at 3MHz, 4 wait states per word access (6 cycles). CPU PAD is 2 cycle access. Thus, the 8-bit bus has a burst rate of 3 million/6, or 500000 words/sec, the 16-bit bus 1.5million words/sec.

Assuming an average of 5 full bus cycles per instruction, 8-bit memory would be 100K instructions per second (500000/5), and 16-bit memory would be 300K instructions per second. I.e., 0.1-0.3 MIPS for the 99/4A.

The 9995 in the Geneve works at 3MHz, 1 wait state per byte access (2 cycles), except SRAM at 0-wait per byte and CPU RAM at 0-wait per word.
Thus, the 8-bit bus bursts at 1,500,000 bytes/sec, or 750000 words/sec, and 1,500,000 words/sec for 0-wait. Assuming an average of 6 full (8-bit) bus cycles per instructions, 1-wait memory operates at 250000 (1.5M/6) instructions per second, 0-wait memory at 500000 instructions per second, and CPU RAM memory at 1 million instructions per second. I.e., 0.25-1 MIPS.

Because the 99/4A and Geneve use a mix of memory speeds (wait states), with the fastest (16-bit memory) of both limited to 256 bytes, the upper end is seldom reached except for the most speed critical operations.

HARDWARE MODIFICATIONS [GFHM000]:
ARE THERE ANY HARDWARE HACKS WORTH DOING
Remember, many of the Geneve's parts are unavailable, there is no company that will make repairs for you. So, if you screw up anything you are pretty much on your own. Warning number two, the Geneve is a multilayer board, it's pretty easy to trash an inner trace and not be able to figure out what is wrong.

ADD 32K of FAST STATIC RAM [GFHM001]?
From: Tim Tesch
Subject: TI99: Geneve-32K Upgrade Instructions

A few people have asked how to modify their Geneve with the extra 32K SRAM, to give them extra memory and let them use MDOS 2.50 and above. So... here we go...

Requirements:
- Soldering iron, 15-20watt max.
- Solder, use rosin core NOT organic core
- 8" piece of wire-wrap wire.
- (1) 32Kx8 SRAM (Hitachi 62256 compatible)

DISCLAIMER: If you are not comfortable modifying your equipment, DO NOT PERFORM THIS UPDATE! I will NOT replace blown gate-arrays or 9901s for free! Enough said.

STEP 1:

Carefully remove 32K chip from Geneve, noting the ORIENTATION. The square/half-circle should be facing the back of the card (joystick/mouse/video connectors.) If you are not sure, use a Sharpie or piece of tape to keep track of the position. Use a small, wide-blade, flathead screwdriver. Apply gentle pressure to the TOP of the memory chip with your thumb, while gently "prying" between the memory and its socket. DO NOT PRY ONE SIDE COMPLETELY. Doing so will bend the pins on the opposite side. Pry the chip a bit at a time, switching from one side to the other until the chip is free.

>> This chip is located to the left of the battery, directly below the 12volt regulator. If you do not recognize this chip - STOP! Send in your card :)-

STEP 2:

Stack the new 32K chip on top of the existing memory. Place the newchip on top of the old chip. The square/half-circles MUST lay "on top" of one another. BEND pin #20 enough so that it does not contact the pin underneath. You may do so using a small-blade screwdriver.

Here is a visual representation:

```
  1 1 1 1 1
  4 3 2 1 0 9 8 7 6 5 4 3 2 1
 +-----------------------------+
 |                             |
 |       HM62256-10           (
 |      (upside down)          |
 +-----------------------------+
  1 1 1 1 1 2 2 2 2 2 2 2 2 2
  5 6 7 8 9 0 1 2 3 4 5 6 7 8
            ^
            ^
            ^

 * DO NOT SOLDER THIS PIN TO ITS NEIGHBOR BELOW *
```

One additional note: - TAT 09.13.1999
Before performing STEP #3, you need to solder the pins from the top chip to the pins on the bottom chip, leaving pin#20 unsolderd.

STEP 3:

Insert stacked chips into socket. chip should be inserted as shown above, half-moon to the right. Make sure pin #20 clears the EPROM so that you can solder the wire to it as shown in step #4.

STEP 4:

Solder wire to memory. Strip and tin one end of the wire. Solder to pin #20, taking care not to short it with (a) any adjoining pins or (b) the EPROM directly below.

Take the wire and bring it to the underside of the board. You can go around the "top" or "bottom". DO NOT GO THROUGH THE HOLES NEAR THE EDGE CONNECTOR! While tempting, you run the risk of stripping wire and shorting something. worst-case, you short the wire with +16 volts! This would not be a good thing.

STEP 5:

Solder wire to underside of Geneve - GATE ARRAY. This step MUST be done carefully and exactly. If you make a mistake, you risk blowing out your gate array. This chip will cost you about $45.00 to replace.

Strip the wire, exposing no more than 1/8". We will solder the wire to the gate array as follows:

```
<<<- mouse/joystick/video<<<
             o o    Gate Array     o o
             o o  (underside of    o o
             o o       board)      o o
             o o                   o o
             o o                   o o
             o o o o X o o o o o o o o
               o o o o o o o o o o o
```

"X" marks the spot. Carefully solder the wire to this pin. Make sure that you do NOT short any of the three surrounding pins. In case you wish to count, it is the FIFTH pin from the left, TOP row.

STEP 6:

Test the geneve
Replace clamshell and put into your PEB/Case. Turn on the power and see what happens :)   If all went well, you should boot without any problem. Once booted, type CHKDSK at any valid disk. You should see:

589824 bytes of total memory.

STEP 7:

Pat yourself on the back - you've successfully modified your geneve! Go try MDOS 6.00.

* Tim Tesch, S&T Software

ADDITIONAL NOTES:

(By TAT)
Anyone adding the 32K SRAM memory to their Geneve to make it compatible with MDOS 5.0 should be careful when following the instructions. One of the Geneves I took home for repair would not run certain programs, like Directory Manager or Port. So I tried the following:

   MEMTEST reported 576K - the proper memory
   CHKDSK reported the same
   MDOS 5.00 loaded properly
   GPL appeared to work properly

I checked the traces and wire running to the 32K socket - all good. It was only after I stepped away for some ice cream did I realize what the trouble was... The user had stacked the two 32K chips properly with one very important exception!! He soldered ALL 28 pins, including pin 20, which should only be connected to the gate array on the solder side. By doing so, the two chips "overlapped", their contents became one and the same. Talk about double trouble!

What's the proper orientation of the Geneve when doing the 32k upgrade?

Place Geneve on a table as follows:
1. keyboard,joystick,mouse,video connectors to LEFT
2. LED(red light) to RIGHT
3. Edge connector CLOSEST to you
The gate array will now be in correct position. Make certain you do NOT short out any adjoining pins.

ADD 64K of VIDEO RAM [GFHM002]

ADD A BATTERY HOLDER [GFHM003]

A good battery holder replacement for the Geneve is: and can be ordered from?

Ok.. I have replaced my battery and my Geneve STILL won't keep the time! Now what?

FIXING A CLOCK THAT WON'T KEEP TIME [GFHM004]
Ken Hasting writes:

It's the SIP resistors! I had 2 that were 1K (correct) and 2 that were 10K (incorrect). I battled that problem for MANY months, asking advice from Don O'Neil (maybe he remembers it) and also of Don Walden. Walden suggested that all the SIP resistors should be 2.2K, and when I installed those - the problem got REALLY bad! I decided to install 1K SIP resistors, and THAT took care of the problem. I don't remember which 2 had the wrong value (originally), so I can't say which one specifically is tied to the clock circuit (although, I could get out my Geneve schematics and trace it out)!

If you need to replace a SIP resistor (yourself), and you don't have a desoldering iron that can heat all the pins at the same time, use a pair of small, wire (or nippy) cutters and cut the old SIP resistor between each pin, so you can unsolder and remove the resistor network one pin at a time! That's cutting right through the resistor itself, not the pins.

You won't be able to get that close to the PC board, or you might damage a trace under the SIP resistor.

After removing the old SIP resistor, make certain that all 10 (or 6, depending on which one) holes are free of solder.

Insert a SIP machined-pin socket array, and solder it into place. This will allow you to remove a SIP resistor in the future, should it be necessary. Also, it allows for more flexibility, so the SIP resistor does not get broken when handling the card, or plugging or unplugging connectors on the back of the Geneve. The 10-pin SIP resistor nearest to the back of the card, is well-known for breaking, if the card is flexed too much while plugging (or unplugging) connectors to the Geneve. Usually, the crack is so slight, it can't be seen with the naked eye. Using a magnifying glass and plenty of light, you should be able to see if any of the SIP resistors are cracked. When the SIP resistor nearest to the back of the card gets cracked, the most common problem is with the Geneve's mouse - it will act very erratically!

The transistors nearest to the clock chip could also be the problem. I have seen many different values on those transistors on different Geneve's. I don't know why the same exact value was not used in different batches of Geneve's produced, but maybe the differences are so slight, that it doesn't matter. Values I've seen are: 2N4126 & 2N3906.

I don't remember what the different values are (I have a reference book, but I'm not going to dig it out, unless you want to know what the values are), but it may only be that one might have a higher mv tolerance.

Sorry for the long message. I hope this info is of help.
- Ken

P.S. 10K SIP resistors are also marked with a 103, and 1K with a 102. If you need to buy new SIP resistors, don't forget to get the SIP machined-pin sockets. They will usually be a long strip of about 30 pins (used for SIPP memory). Use a sharp knife to cut them to the right number of pins you need. DON'T use nippy or wire cutters to cut the SIP sockets - that will usually mash one or both of the pins (and the plastic insulator) on either side of the cutter's jaws!

WARNING!!! Be VERY careful, if cutting the SIP resistors! They are (usually) ceramic, and will "shatter" when cutting, sending bits and pieces around at high speed!!! Wear goggles or close your eyes (and mouth) just prior to snipping the resistor. Use caution! :-)
=====================

<E.O.F.>