T A B L E 0 F C 0 N T E N T S

1. INTRODUCTION

1.1 The Gemini GM833

The Gemini GM833 'RAM-DISK' board is an 8" x 8" 80-BUS and Nasbus compatible board designed for use as a high speed, 512Kbyte 'disk drive' with Gemini MultiBoard and Nascom computers.

1.1.1. Why a 'RAM-DISK' ?

Any method of expanding the address space of the Z80 suffers from the problem that it tends to be unique to a particular system, and no standard software will support it. Only software written specifically for the system would be any use, and it is likely that software like that would be extremely thin on the ground. So the approach being taken now, by a large number of manufacturers, is to use the extra memory in conjunction with disk operating systems (DOSs), and to make the memory appear as a disk. This solves virtually all the problems. The DOS handles the management problem of organizing data in the extra memory, and all that has to be added to the BIOS of the disk system, is a simple driver to convert a disk track/sector read/write request into a read/write request to a specific area of this extra memory. All the standard disk software still runs perfectly under the DOS, but any read/write request the 'Memory disk' will result in an immediate response. This is because it only takes a few microseconds to locate the wanted 'track' and 'sector' on the memory drive, as opposed to tens or hundreds of milliseconds on a mechanical drive.

1.1.2. The GM833 approach

The GM833 is not a conventional RAM board, but has been designed to use three Z80 I/0 ports, which may be regarded as 'track', 'sector- and -data'. This allows the board to be very simply interfaced to CP/M to appear as an extremely high-speed disk drive - over 30 times faster than a conventional floppy disk in certain applications. The GM833 'RAM-DISK' is supplied with 12K bytes capacity, and an on-board DIL switch allows multiple boards to be fitted to a single system. Also included on the leading edge of the board is an LED, an activity indicator which illuminates every time the board is accessed.

1.2. Guarantee

Your GM833 RAM-DISK board is guaranteed by the supplier (your Dealer) for one year from the date of purchase. However, being a system module, any faults attributable to incorrect installation or use of the board within your system will be fully chargeable as to both parts and labor.
Any queries regarding the implementation of your GM833 RAM-DISK board should be directed at your Gemini dealer.

2. THE GEMINI GM833 RAM-DISK

The Gemini GM833 RAM-DISK is not a conventional system memory board, but is arranged as a block of memory separate from the 80-BUS address lines, which the CPU can only access via three I/0 ports. To communicate with the memory, the CPU has to write an address to two I/0 ports, and then read/write data from/to another IO port.

If the RAM-DISK memory was addressed on a byte-by-byte basis, the data transfer rate would be slowed, and the board would be rather clumsy to use. However the board has been optimized for use with CP/M, and the interface has been made 'disk-like.

One can regard the three I/0 ports as 'track', 'sector', and -data'. The address applied to the 512K byte memory array is made up of three components: AO-A6 coming from a seven-bit counter, A7-A14 from the 'sector' latch, and A15-A18 from the 'track' latch. In addition, the four high 'address' lines from the 'track' latch are compared with an on-board DIL switch to provide an enable signal to the memory array.

The seven-bit counter is controlled in two ways: Whenever data is written into the 'track' register, the counter is cleared. Whenever data is read/written to the data port the counter is incremented. Thus the memory array can be regarded as a disk of sixteen tracks, (the low four bits of the 'track' register), with 256 sectors per track, (eight bits of the 'sector' register), and with 128 bytes per sector, (the seven bits of address from the counter). So transfers to and from the RAM-DISK occur in blocks (or 'sectors') of 128 bytes each.

This way of adding extra memory, as well as being more economical in its requirements for support software, is also economical in hardware. In the case of a paged or mapped memory board, the memory has to be designed to work at the full speed of the shortest Z80 memory cycle -the M1 and refresh cycle. With the I/0 approach the memory array only has to meet the more relaxed specification of an I/0 cycle, thus allowing slower RAMS to be used, and leading to a more reliable board with larger margins on critical timing paths.

3. INSTALLATION

3.1. General

Carefully unpack your GM833 and examine it for any mechanical damage. In the event of any damage please inform your dealer immediately.

Your GM833 will have been shipped to you fully tested and working. All that should be required is for the board to be plugged into the bus. However, as there are a number of link and switch options on the GM833 it should prove useful to read through this manual carefully first.

When plugging the GM833 into the bus please take great care, excessive force should not be required, any difficulty that is encountered will in all probability be due to the keyway of the edge connector not slotting accurately into the slot in the edge of the board. Ensure that the board is plugged in with the edge connector going in first and the correct way around, it should not be possible to plug the board in the incorrect way around because of the keyway. Power is automatically connected to the board through the bus - refer to the 80-BUS or Nasbus specifications for further details.

3.2. Switches S1 - S4

The standard GM833 board provides a memory drive of 512K (or 0.5Mbyte) capacity. If this is insufficient then further boards may be added. Current Gemini CP/M implementations allow up to four boards to be used, providing a capacity of 2 Mbytes. Several GM833 boards may be installed in a single system, but to use more than four boards requires custom software. (See Section 4 - Software.)

A 4 pole DIP switch is fitted to the GM833 to allow multiple boards to be installed in a single system. If only one board is to be installed then all four DIP switched should be 'ON'. If there are to be multiple boards then the following switch positions should be used.

Note: If the user wants to have GM833 boards in the system that will not be assigned as CP/M drive 'M', but will be used via user-written software, then the DIL switches) should be set to a higher number. See Sect. 4.1.

Board Number	Sl	S2	S3	S4
1	ON	ON	ON	ON
2	OFF	ON	ON	ON
3	ON	OFF	ON	ON
4	OFF	OFF	ON	ON
5	ON	ON	OFF	ON
6	OFF	ON	OFF	ON
7	ON	OFF	OFF	ON
8	OFF	OFF	OFF	ON
9	ON	ON	ON	OFF
10	OFF	ON	ON	OFF
11	ON	OFF	ON	OFF
12	OFF	OFF	ON	OFF
13	ON	ON	OFF	OFF
14	OFF	ON	OFF	OFF
15	ON	OFF	OFF	OFF
16	OFF	OFF	OFF	OFF

3.3. LK1 - /NASIO

LK1 is situated near pin 1 of IC70 and determines whether or not the GM833 generates a /NASIO signal. If the link is made a /NASIO signal will b generated. The /NASIO signal is only required for systems based on the Nascom 1 or Nascom 2 boards. This is because on the Nascom the I/0 ports are no fully decoded. Only one I/0 board in a system needs to generate /NASIO. This signal is NOT required with Gemini MultiBoard systems.

With Nascom 1 the I/0 internal/ external link (LK1) should be set to external and, because of a decoding error on Nascom 1, the on-board PIO (IC35 must be removed. With Nascom 2 the I/0 internal/ external switch (LSW2/8) should be set for external operation.

Note that the GM833 does not provide the Nasbus DBDR signal, and so if the board is to be used with a Nascom 1 it is necessary to implement additional circuitry on the Nascom to establish the required data-bus direction and switch DBDR accordingly.

3.4. LK2 - Clock
LK2 allows the clock input to the board to come from either the 80-BU CLK line (line 5), or the AUX CLK line (line 8). The GM833 requires a 4MH clock input. If the system in use does not have a 4M Hz system clock, then it will be necessary to cut the trace between 1 and 2 on LK2, and connect 1 to in order to pick up the clock input from the AUX CLK line, which will have t be provided with a 4MHz clock.

3.5. Port Addresses
The standard port addresses for the GM833 RAM-DISK are:

TRACK	OFBH
SECTOR	OFCH
DATA	OFDH

These are provided by the 833-1 PROM fitted in position IC65. This PROM also decodes ports OOH - 07H to provide the /NASIO signal. If it is require to decode the GM833 at any other addresses then custom PROMS may be ordered via any Gemini dealer.

3.6. LED - Activity Indicator
Situated on one edge of the GM833 is an LED. This will light on an access to the data port of the GM833, and for cased systems it is possible to disconnect the LED and bring it to the outside of the unit. Note that the circuitry associated with this has been deliberately designed so that the LED will illuminate regardless of whether the actual 'track' and 'sector' being selected correspond to memory on any specific GM833 board. Consequently an one LED will indicate activity on any GM833 board, and so if activity indication is required in a cased system, only one LED is required.

4. SOFTWARE

The GM833 may be used either under CP/M or directly by a user program. If operation is to be under CP/M then Gemini CP/M BIOSs are available to support it. Alternatively the user may choose to modify their own CP/M. This section describes the Gemini CP/M BIOSs that support the GM833, gives an idea of how the user may modify their own CP/M, and gives details of how the board may be driven directly by an application program.

4.1. CP/M operation
Support for the GM833 has been provided in Gemini CP/M BIOSs Versions 2.3 and higher. The version number of the BIOS that you are running is displayed when the CP/M is first loaded.

The actual GM833 software implementation does vary slightly with different versions of the BIOS and the following summary covers currently available Gemini CP/M BIOSs. If you have older versions of the BIOS or CONFIG, then up-to-date versions are available via Gemini dealers for a nominal charge.

4.1.1. BIOS Versions 2.3 and 2.4
BIOS Version 2.3 supports one GM833 only, as drive 'M'. The BIOS is so arranged that it does not re-initialize the memory drive if you are forced to press Reset. The CP/M block size used is 2K, and the number of directory entries is 64. BIOS Version 2.4 has the same support for the GM833 as V2.3, but the number of directory entries has been increased to 128. In both cases it is assumed that the GM833 is set as board 1 (i.e. all switches ON.)

To set-up one of the above BIOSs to enable the 'M' drive it is necessary to alter one of the values in the patch area. Doing this is described in the manuals that were supplied with your CP/M and/or Gemini equipment. In this instance the byte that needs changing is XXA2, and bit 7 of this byte should be set. Version 1.2 of the program CONFIG has been extended to allow the user to do this more easily. It is only necessary to run CONFIG and select the option 'G' on the 'Memory Driver menu option. (This sets the most significant bit of the 'number-of-boards' entry in the patch area.)

4.1.2. BIOS Versions 2.8 and later.
BIOS Versions 2.8 and 3.0 support up to two GM833 boards (i.e. 1 Mbyte), and BIOS Versions 3.1 and later up to four GM833 boards (i.e. 2 Mbytes). These BIOSs support the GM833s as drive 'M' and are so arranged that they do not reinitialize the memory drive if you are forced to press Reset. Detection of the number of GM833 boards present is automatic, and the BIOS sets itself up automatically. It is assumed that RAM-DISK boards will have their DIL switches set sequentially from Board 1 upwards. If the user wants to have GM833 boards in the system that will not be automatically assigned as drive 'M', then the DIL switches) should be set to a higher number.

Following a 'cold-boot' of CP/M the BIOS looks to see if, and how many, GM833s are present, starting at board 1 and going up to board 4 (only up to board 2 with BIOSs 2.8 and 3.0 - incorrect operation may occur if more than two boards are present). If any exist then a drive 'M' is set up.

The following table shows the CP/M drive 'M' configuration that will: automatically be selected:

Number of GM833 boards	Total capacity	CP/M Block size	Number of directory entries
1	512K	2K	128
2	1024K	4K	256
3	1536K	8K	512
4	2048K	8K	512

Following initialization the CP/M BIOS performs two other operations Firstly it copies the CP/M system image from memory to a 'system track' on the 'RAM-DISK'. Thereafter all warm boots occur from the memory drive rather that drive 'A', considerably speeding up that operation. Secondly the BIOS patches the BDOS so that drive 'M' becomes the default drive that is logged of whenever the BDOS is reset. The net result is that a system with a GM83: installed will always start up with a prompt of 'M>' rather than the more usual 'A>'. As these CP/Ms also have CCPZ implemented, with its hierarchic search path, this means that programs on drive 'A' will automatically b, located and loaded (see the CCPZ manual supplied with the CP/M). If, however the more usual prompt of 'A>' is still required, this can be achieved by using the CONFIG program, and setting up an auto-execute command of 'A:'.

4.1.2.1. SUBMIT Operation.
With the automatic patches by BIOS 2.8 et se q., as described above this means that a CP/M disk reset command will log-in drive 'M', ant submit files will be looked for on 'M'. It is therefore necessary t, change SUBMIT and XSUB to work on 'M'. The exact modification will depend on the particular versions of the programs, but what it is necessary to do is to change the drive which is used for the $$$.SU file. Thus, search for occurrences of $$$ .SUB and change the preceding byte (which contains the drive number) to ODh (='M') (Currently this byte will be set to either 00 (for the default drive) o 01 (for drive 'A').) You may find that these changes have already bee made on your CP/M master disk, the resultant files being called, SUBMITM.COM and XSUBM.COM. The effect of the use of these BIOS versions along with these program changes, will be much faster SUBMIT operations.

4.1.3. Using the RAM-DISK
Using the RAM-DISK requires a careful approach to work to ensure that a the end of the day all changed and new programs end up back on permanent storage (disk). Copying a file to disk as soon as it is changed rather defeat one of the benefits of the RAM-DISK, but it has to be remembered that the RAM DISK is volatile. Though the Gemini BIOS does not obliterate files if you are forced to press the reset switch, it cannot protect you against accidental (o deliberate) powering down.

This is the time when the programmable function keys of the Gemini keyboard come into their own. At the start of a session one (or more) can b programmed up to provide a backup command, (e.g. PIP A:=M:*.MAC'M), and the this key can be pressed at idle moments when you pause for thought (or answer the phone), and, finally, at the end of the session.

In certain applications the volatility of the RAM-DISK will not matter. For example, if the application requires as rapid as possible access to any record in a very large database, then when the program is started it can copy a number of index files to the **RAM-DISK,** and use these to control access to the database. If the power fails nothing of importance will be lost.

4.2. Your own software

If you are writing you own drivers for the GM833 RAM-DISK, perhaps for your own BIOS or a custom program, it is best to regard the board as a 'disk' with the following characteristics:

16 Tracks per board
256 Sectors per track
128 Bytes per sector

As well as the drivers you will need a data management program that organizes what is stored where on the disk. If you are using CP/M or a similar disk operating system, this will already be taken off for you by the BDOS. If you are using the board direct in a non-disk environment, then this is a function your own program will have to handle.

This controlling program (e.g. the CP/M BDOS) passes read/write requests to the low level drivers (e.g. the BIOS) requesting a transfer of data between a block of memory and a track and sector on the RAM-DISK. The low level drivers for GM833 are very simple and would be:-

```
      Ld a,(sector)       ; Set the sector
      Out (sec port),a
      Id a,(track)        ; Requested track
      Out (trk port),a
      ld hl,dmaaddr       ;  Set HL=transfer address
      ld b,128            ;  128 bytes to move
      Id c,data_port      ;  Point at the data port
      inir                ;  Read the data
                          (Use OTIR for a write)
```

If you are driving the board directly, the controlling program can also be simple if you are dealing with a specific problem. For example, if the controlling program is dealing with data based on 128-byte records, the GM833 RAM-DISK can be considered as a remote array that holds 16 * 256 records. A driver totally written in a high level language can be used, using the INP and OUT statements (or similar) to read and write directly to the control and data ports of the RAM-DISK. However that approach loses the speed advantage of the RAM-DISK, as all the data transfers run at the rate of the language interpreter (or compiler).

A better approach is to use a small machine code routine and to set aside a small amount of memory as a buffer. The driver then can be called by using whatever mechanisms are provided by the high level language being used. (e.g. USR function or the CALL interface in some BASICs.)

Assembly language example for low-level drivers for interfacing to the GM833 RAM-DISK. This assumes the RAM-DISK is organized as 4,096 records of 128-bytes each. , On calling: DE = Record number (0-4095)
HL = Memory buffer for data transfer

```
read: ld    c,0           ; Indicate read
      jr    rw
write:      Id c,l        ; Indicate write
rw:   ld    a,e           ; Set the sector
out   (sec port),a
ld    a,d                 ; Requested track
out   (trk port),a
      Id    a,c           ;  Load flag
      ld    b,128         ;  128 bytes to move
      Id    c,data_port   ;  Point at the data port
      or    a             ;  Read or Write?
      jr    z,do$rd       ;  Skip if a READ
      otir                ;  Write data to the card
      ret
do$rd:      inir          ;  Read the data
ret
```