CMD

Creative Micro Designs

CMD 1750/1750XL

User's Guide

& Programmer's Reference

CMD

Creative Micro Designs

CMD 1750/1750XL

User's Guide

& Programmer's Reference

Contents

General Information 4
Section 1: Getting Started 5
Section 2: The Utilities Disk 7
Section 3: RAMDOS 15
Section 4: C-128 Mode Use 19
Section 5: CP/M Mode Use 21
Section 6: Programmer's Reference 23
Technical Support 27
Warranty 27

3

General Information

Congratulations on your purchase! Your new CMD RAM Expansion Unit (REU) will provide you with the increased RAM capacity necessary to take full advantage of many of the high-powered applications that offer REU support. We hope that your new REU will greatly improve the speed and functionality of your Commodore C-64 or C-128 computer, and also act as a vehicle through which you will support the manufacturers and authors of new programs.

Compatibility

The CMD 1750 REU is fully compatible with the Commodore 1750 REUs, and will operate with all software packages written to recognize that device. Our 1750XL model has been designed in accordance with the most common standard available for expanded RAM devices to provide the maximum RAM available in a Commodore-compatible DMA RAM device.

Important Power Requirements

Your new cartridge is only a fraction of the size of the original 1750 REU, and in many situations will draw less power from your computer than an equivalent Commodore model. But contrary to the recommendations of previous 1750 clone manufacturers, CMD recommends the use of a heavy duty power supply on all C-64 or 64C computers equipped with REUs. Commodore specifies that not more than 100ma of current should be drawn from the cartridge port (using the stock power supply); most REUs draw about double that amount! In addition, many stock Commodore power supplies will see a vastly shortened life span even at the 100ma level, and when they fail will often cause damage to your computer. If you need a heavy-duty supply, you may order one by contacting the CMD order desk at 1-800-638-3263 (Mon.—Fri., 9AM—5:30PM EST). If you're calling from outside the United States or Canada, the number to call is 413-525-0023.

The Utilities Disk

We have included a group of utilities which we feel certain will make using your CMD REU easier and more productive. The RAMDOS and GEOS CONFIGURE programs included are enhanced to support the 2 MB capacity of the CMD 1750XL, and our extensive REUTEST utility will verify the operation of your CMD REU. Please be aware that these programs are covered under international copyright law, and may not be redistributed without permission from the copyright owners.

Section 1
Getting Started

Introduction

The CMD 1750 (512K) and CMD 1750XL (2 MB) RAM Expansion Units (REU) allow specially-written programs to store more data in RAM than previously possible with the stock memory (128K on the C-128, 64K on the C-64). Your computer has two kinds of memory, Random Access Memory (RAM) and Read Only Memory (ROM). RAM is used for temporary storage. Information stored in RAM remains there as long as the computer stays on. ROM stores the operating system of your computer, and cannot be changed. The CMD 1750 and 1750XL provide additional RAM for your C-64 or C-128. Like most RAM devices, CMD REUs cannot retain information after power is removed. Therefore, these devices need to be reloaded after turning your computer on. CMD's RAMLink provides a battery and power backup for your REU. That means that your programs and data remain intact when your computer is shut down. RAMLink also provides a permanent enhanced DOS, and lets you increase your memory up to 16 MB. Additional information on this product can be obtained from CMD.

All Commodore and compatible REU's contain *indirectly accessible* RAM, meaning that data or programs stored in the REU cannot be executed directly by the C-64/C-128—you must first move the program or data into the computer's main memory, which happens very quickly.

It's important to remember that RAM is not permanent storage; the information held in RAM is gone when you turn off the computer (unless used with a RAMLink or some other power backing device). If you want to save something permanently, be sure you store it onto a disk before turning off your computer.

The CMD 1750 increases your computer's memory by 512K, which is equal to 2048 disk blocks. The CMD 1750XL increases your computer's memory to an incredible 2048K (2 MB) or 8192 disk blocks. To demonstrate just how much memory is available, consider that an entire 1541 disk is only 170K. A 512K REU has 8 times the memory in a stock C-64, while a 2 MB REU contains 32 times the amount of memory found in a C-64.

An REU can vastly enhance the speed and functionality of programs written to make use of it. For example, GEOS uses part of the REU to store commonly accessed items like disk drivers, another part as a RAM disk, and can also store itself in the REU for fast re-booting. Other programs that use the REU offer different enhancements; some word processors and other programs store and retrieve segments of the program or data in the REU for quick access, while many terminal programs (Novaterm 9.6 for example) use it as a large capture buffer.

If you have a C-128 computer, there are three commands in BASIC 7.0 which have been designed to work with expanded RAM. These commands, STASH, FETCH, and SWAP, are dicussed in more detail elsewhere in this manual.

In CP/M mode the RAM Expansion Module acts like a very rapid disk drive. Large application programs, such a word processors or databases, can be stored in the REU at the beginning of a work session. From then on, you command the computer to find program utilities and modules in the RAM, rather than on an external disk drive. The CP/M operating system has its own RAM Expansion driver program and will control the REU transparently.

Your CMD REU may be used on any C-64, 64C, C-128 or 128-D computer, and in any mode available on these models. (*Note: CMD REU's may also be used on some SX-64 portable computers, however, it is impossible to say if the unit will work on a given SX-64 due to the differences in RF noise and other factors that affect bus operations in these machines.*)

Installation

To install your CMD 1750 or CMD 1750XL REU in your computer, follow these three brief steps:

Step 1: Before installing the REU, turn off your computer and all peripherals. Failure to do so may result in severe damage to the REU, the computer or both.

Step 2: Insert the REU cartridge into the Expansion (Cartridge) port on the back of your computer. This port is located to the right rear of the computer. Make sure you insert the cartridge with the sticker facing up, and that it seats fully and squarely into the cartridge port socket. *Note: If you have a RAMLink, the REU should be installed in the RAM Port, with the REU sticker facing you.*

Step 3: Turn your computer and peripherals back on.

Your REU is now ready to use. However, before you begin using it with software, we recommend that you refer to Section 2 of this manual and follow the instructions for loading and running the REUTEST program included on the CMD REU UTILITIES disk. This program will verify that your REU is properly installed and that both the REU and the computer are in full working order.

Section 2
The Utilities Disk

The CMD REU UTILITIES disk contains programs to assist you in testing and using your CMD REU. These programs are copyrighted and are for your use only. The following is a list of the disk contents:

```
RAMDOS.BAS
RAMDOS64.BIN4.3
RAMDOS128.BIN4.5
FCOPY
REUTEST
ZAPREU
---GEOS FILES---    (note: this is just a separator)
CONFIGURE
128 CONFIGURE
```

All of the programs listed above are explained in this section with the exception of RAMDOS. RAMDOS is an REU disk operating system which requires detailed discussion. It is explained in detail in Chapter 3 of this manual.

REUTEST

This utility has been created by CMD to assist you in making sure that your REU is properly installed and fully operational. It tests the REC controller chip installed in your CMD REU, and also performs an extensive test of the installed RAM in your unit. *Note: If you have a CMD SuperCPU, disable it before turning on your computer to use this program. Disable other cartridges if you suspect they may be incompatible with an REU, and if you have a RAMLink, place it into DIRECT mode.*

To load the REUTEST program, insert the CMD REU UTILITIES disk in your floppy drive (device 8) and type:

```
LOAD"REUTEST",8
```

Be sure to press the <RETURN> key to complete the command. After the program has loaded, type RUN and press the <RETURN> key.

When the program begins, it will prompt you to make sure that you wish to test the REU. Answer yes by pressing the <Y> key, and the test will begin. As the test proceeds, it will first test the memory in your computer that will be used to verify

operation of the REU. It will then begin testing various function of the REC controller chip used in the REU. As each test is completed, the program will indicate that your unit has passed by placing an OK message next to the test description. Finally, the program will use various bit patterns to test the complete memory of the REU. This last test can take several minutes, but you'll see it change the bit pattern info as it progresses. You'll also see eight solid sprite blocks displayed along the bottom of your screen if you're using a 40-column display. (*Note: CMD recommends using a 40-column display for this test, as some hardware problems which do not affect the transfer of data can be detected as garbage data in sprites*).

When the test successfully completes, it will display a message to that effect. However, if at any point a failure occurs the test will end, indicating ERROR instead of OK for the status of the particular test that failed. If you get a such a failure, try the troubleshooting steps below before contacting CMD Technical Support.

- If you have any additional cartridge-port hardware attached to your system, remove it.
- Re-insert the REU directly into the cartridge port of your computer and run the test over again.
- If the test fails again, check the alignment of the contacts on your cartridge port, re-align them if necessary, and make sure they are clean. Test again if you found any deficiencies that you have corrected.
- Try testing the REU on another computer if you have access to one.
- Try testing another REU in your computer if you have access to one.

Be sure to take notes on what test the REU fails, and what the results were to each of the troubleshooting steps you took. Should you need to contact CMD Technical Support, this information will help speed determination and resolution of the problem. Above all, do not send the unit back to CMD until you have contacted CMD Technical Support for a Returned Material Authorization (RMA) number.

FCOPY

This program was created by CMD to fill the need for a file copier capable of copying any type of file between any two serial drives or a serial device and an REU. This program may also be used to scratch (erase) files, view directories, and send disk commands. Special support exists for devices which feature CMD Native Mode subdirectories, 1581 sub-partitions, 1541, 1571 and 1581 emulation mode partitions. These special features do not apply to an REU itself, but will be useful if you have other CMD devices with these features. The use of this program is mostly self-explanatory, but we have included a breakdown of the functions here in order to provide you with any details that aren't clear.

Set Source Device (F1)

Press the <F1> key to select the device that you wish to copy files *from*. The device number and type of device will be shown on the display. If the device that you want

to copy from does not appear right away, additional presses of the <F1> key will advance through the available drives. If any of the devices you use are not recognized by the program, question marks will be shown instead of the actual device type. This may occur with unrecognized third party disk drives; however, the program will usually work with these devices anyway, unless the DOS in the particular device happens to be highly incompatible with standard Commodore DOS commands and file handling procedures.

Set Target Device (F5)

Press the <F5> key to select the device that you wish to copy files *to*. The device number and type of device will be shown as the Target Device on the display. The use of this function is identical to the Set Source Device function.

Set Source Partition (F3)

If your source drive is a CMD partitionable device, use the <F3> key to select the partition you wish to copy files *from*. Additional keypresses will advance through the available partitions; you may also hold down the <SHIFT> key while using this function to move backwards through the partitions. The partition number, name and type will be shown as the Source Partition in the display.

Set Target Partition (F7)

If your target drive is a CMD partitionable device, use the <F7> key to select the partition you wish to copy files *to*. The partition number, name and type will be shown as the Target Partition in the display. The use of this function is otherwise identical to the Set Source Partition function.

Set Source Path (S)

If the source device is a CMD partitionable device and the source partition type is Native or 1581 Emulation, or if the source device is an actual 1581, you can press the <S> if you wish to enter a path for subdirectories or sub-partitions. After you press the <S> key, you will be allowed to enter the path information. When entering a nested subdirectory or sub-partition, each name must be separated by a slash (/). Press the <RETURN> key when the full path has been entered.

Set Target Path (T)

If the target device is a CMD partitionable device and the target partition type is Native or 1581 Emulation, or if the source device is an actual 1581, you can press the <T> key to indicate that you wish to enter a path for subdirectories or sub-partitions. This function is otherwise identical to the Set Source Path function, except that the path is intended for the target device.

Source/Target Directory (A/B)

You may press the <A> key to get a directory from the source device, or the <B> key to get a directory from the target device. The program will ask for a search pattern so that you limit the directory display to files which match a given name

and/or filetype. For directories read from a CMD device with a Real-Time Clock installed, the pattern will also allow the selection of files by time and date. If you want to see all files, press <RETURN> with the default pattern of an asterisk (*).

Select Files (F)

If you want to copy or scratch files, you must first select which files. To do this, set the source and target devices (and partitions if necessary) and then press the <F> key to read the filenames from the source directory into the selection buffer.

Before the buffer will load, you'll be asked for a pattern, in case you want to limit the files loaded into the buffer. If you want all files loaded into the buffer, just press the <RETURN> key with the default pattern of an asterisk (*).

File information is stored in the buffer using a dynamic method, so if necessary, you will be able to select from several hundred files if your source directory is that large. If your directory contains more files than can be viewed, you'll need to limit the selection by using a pattern which will match fewer files.

Once you have entered the file selection mode, the first few files that match your pattern will be seen in a window at the bottom of the screen, and an arrow will be pointing to the first file. You may scroll the arrow down by using the <CRSR-DN> key; you may also scroll the arrow up by holding down the <SHIFT> key in combination with the <CRSR-DN> key (this gives you <CRSR-UP>). If you scroll to the last selection, the list itself will begin to scroll, allowing you to access all the files in the buffer. Likewise, you can scroll the list up with <CRSR-UP> if you have scrolled it down.

To select an individual file for copying, scroll the arrow until it points to the desired filename, then press the <RETURN> key. An asterisk (*) will be placed next each file that has been selected. Pressing <RETURN> while the arrow is pointing to a selected file will de-select that file. Also, pressing the <T> key allows you to toggle *all* selections (<T> will select all unselected files, and de-select all selected files). As you select and de-select files, FCOPY will show you how many files and how many total blocks you currently have selected.

When you have finished selecting files, you can press the <C> key to begin copying files, or press the back-arrow key <←> to return to the main menu. If you elect to return to the main menu, you can select the copy function from there by pressing the <C> key, or you can scratch (erase) the selected files by pressing the <#> key (hold down the <SHIFT> key and press the <3> key).

Reselect Files (R)

After selecting files, you can return to the file selection mode without re-reading the directory. Do this by pressing the <R> key from the main menu mode. This allows you to change your selections before or after copying files.

Send Disk Command (@)

You may press the <@> key from the main menu mode to send a disk command to either the source or the target drive. *Important Note: Disk commands do not follow the source or target path, but are instead sent to the current directory. Therefore,*

it is wise to make sure you are currently in the correct partition, sub-partition, or subdirectory before sending a command intended for a certain area. When sending commands to a CMD device that supports partitions and paths, you may include partition number and/or path information directly in the disk command.

Begin Copying (C)

From the main menu mode or while in file selection mode you can press the <C> key to start the copying process. Only the files you have selected using the <F> and/or <R> options will be copied. When copying is complete, an option allows you to copy the same files to another disk (if your target drive is a floppy disk drive).

Begin Scratching (#)

At the main menu, press the <#> key (hold down the <SHIFT> key and press the <3> key) to begin scratching files you selected using the <F> and/or <R> options. *Please Note: Files can only be scratched from the source disk!* You will be asked "ARE YOU SURE (Y/N)?" before the scratching operation begins. Press <Y> or <N> to take the appropriate action.

Exit Program (←)

If you press the back-arrow key <←> while in the main menu, the program will quit. If you wish to use the program again, it must be re-loaded. The back-arrow key is also used to exit from file selection mode, so be careful to avoid hitting it twice!

ZAPREU

This utility allows you to browse through the contents of your REU, and should you so desire, fill your REU memory with a specified byte. This may be handy for users who have battery backed REU's. We've notice that when we use an REU in direct mode in RAMLink, some programs see the contents left previously and assume they are still intact. If you have since used some other program directly with the REU or through a DACC area assigned to the REU, the contents may no longer be completely valid. We've seen this when using CS-DOS after having used GEOS.

This situation was the reason for creating ZAPREU. To use ZAPREU, you must have an REU attached to your computer, either directly or via RAMLink. If it is through RAMLink, then you must switch RAMLink to DIRECT mode.

To use this program, load it by placing the CMD REU UTILITIES disk in your floppy disk drive (device 8) and entering the following:

```
LOAD "ZAPREU", 8
```

Be sure to press the <RETURN> key to start the load. After the program has loaded, enter RUN and press <RETURN>.

Once the program begins, the main options will appear on your screen, along with the default fill settings. The default fill settings display a start address of $00:0000. The first two zeroes are the bank number, and the four following the colon

are the actual starting address in that bank. These values are displayed in hexadecimal format. The end address is similarly displayed, but shows an address of $00:00FF. The addresses can only be incremented by even pages, so the last two places in these addresses never change. The amount of RAM selected for filling indicates .25 K, which is 256 bytes or one page of RAM. The default fill value is set to $00.

To change the start or end addresses, you may press the <S> or <E> key. Both use the same method of adjustment. The bank is changed up or down using the <SHIFT> key in combination with the <+> and <-> keys, and the page is changed using <+> or <-> alone. The start address is not allowed to go above the end address, and the end address is not allowed to go below the start address. As you change the addresses, the amount of RAM selected will change accordingly. The <RETURN> key is used to exit from these adjustments.

The fill value can be change by pressing the <F> key from the main menu. The adjustment can then be made with the <+> and <-> keys. Use the <RETURN> key to exit from this option as well.

The <B> key, when pressed, will cause the fill to be performed according to the fill settings.

If you wish to examine the contents of your REU RAM, the <V> key can be used to enter view mode. This mode displays a page of RAM, either in hexadecimal or as CBM ASCII characters. The view mode is toggled between these two display methods by pressing the <H> key. The bank and page address currently being viewed is displayed at the top of the screen. You may step through pages of memory with the <+> and <-> keys, or through banks by using the <+> and <-> keys in combination with the <SHIFT> key. You may also set a bank and page to move directly to by pressing the <S> key to enter the set mode. This allows you to set the view address without fetching and displaying each page in between. Use the <+> and <-> keys to set the page, and the <+> and <-> keys in combination with the <SHIFT> key to set the bank. When you press <RETURN> to exit the set mode, the specified page of memory will be fetched and displayed. To exit from the view mode back to the main menu, press the back arrow key.

To quit from the program, press the <Q> key.

CONFIGURE & 128 CONFIGURE

These files are only used with the CMD 1750XL (2 MB) REU. If you have the CMD 1750 (512K) model, GEOS v2.0 and GEOS 128 v2.0 are already capable of recognizing and using your REU. You may also use the standard GEOS CONFIGURE files with a 1750XL, but only 512K of the memory will be recognized.

If you have a 1750XL and wish to have GEOS take advantage of the extra memory, you must replace the CONFIGURE file on you boot disk(s) with a version supplied on the CMD REU UTILITIES disk. For GEOS 64 v2.0, the file named CONFIGURE must be replaced with the file of the same name from the CMD REU UTILITIES disk; for GEOS 128 v2.0 the files are named 128 CONFIGURE.

The recommended steps involved in replacing your CONFIGURE file are as follows:

- Install you CMD REU if you haven't already done so
- Boot GEOS with only 1 disk drive (device 8) attached
- Locate and launch your CONFIGURE or 128 CONFIGURE program by double-clicking on it's icon
- Make sure that CONFIGURE indicates that you have 512K in the RAM expansion box (lower right)
- From the file menu, select save configuration
- From the file menu, select quit
- Locate and click once on the icon for the file deskTop (or 128 deskTop for GEOS 128)
- After a few seconds pause, click a second time on the deskTop icon (this should cause a 'ghost icon' to become attached to your pointer)
- Move the pointer until it's point is centered over the drive B icon on the right side of your screen
- Click once on the drive B icon (this will cause the deskTop file to be copied to your REU)
- When your pointer reappears and all disk activity has stopped on your disk drive, remove the boot disk and insert the CMD REU UTILITIES disk
- Click once on the drive A icon
- Locate and click once on the CONFIGURE or 128 CONFIGURE file icon, whichever it is you need for your version of GEOS
- After a few seconds, click again on the icon (this should cause a 'ghost icon' to be attached to your pointer)
- Move the pointer until it's point is centered over the drive B icon on the right side of your screen
- Click once on the drive B icon (this will cause the CONFIGURE or 128 CONFIGURE file to be copied to your REU)
- When your pointer reappears and all disk activity has stopped on your disk drive, remove the CMD REU UTILITIES disk and insert the your boot disk
- Click once on the drive A icon
- Locate and click once on the CONFIGURE or 128 CONFIGURE file icon, whichever it is you need for your version of GEOS
- After a few seconds, click again on the icon (this should cause a 'ghost icon' to be attached to your pointer)
- Move the pointer until it's point is centered over the blank area between the printer icon and the trash icon at the bottom of your screen (called the border)
- Click once in this area to move the file icon to the border
- After the file icon appears at the bottom of the screen, click once on it
- After a few seconds, click on the icon a second time (causing the ghost icon to re-attach itself to your pointer)
- Move the pointer until it's point is centered over the trashcan
- Click once on the trashcan to drop the file into it
- Click once on the drive B icon
- Locate and click once on the CONFIGURE or 128 CONFIGURE file icon, whichever it is you copied to drive B earlier

- After a few seconds, click again on the icon (this should cause a 'ghost icon' to be attached to your pointer)
- Move the pointer until it's point is centered over the drive A icon
- Click once on the drive A icon to begin copying the file to drive A
- After the copy process is complete, click once on the drive A icon, and verify that the file is there
- Shut down your computer for a minute
- While your computer is off, you may attach a second physical disk drive if you have one that you normally use with GEOS
- Turn your computer back on and boot GEOS
- Locate and launch the CONFIGURE or 128 CONFIGURE program
- Make sure your REU is recognized by showing 2048K in the RAM expansion box (lower right)

You may now reconfigure your system according to your needs. If you need to, refer to the section *Using More than One Disk Drive* in your GEOS 2.0 manual (pg. 77), and take note of the differences below.

- Your choices in CONFIGURE and 128 CONFIGURE will now include having 1581 RAM disks
- If you use only one physical disk drive, you may have two 1581 RAM disks; you may have only one 1581 RAM disk if you use two physical drives

Be sure to save your configuration when complete, and copy this version of CONFIGURE or 128 CONFIGURE to any other disks you use which contain your older copy of CONFIGURE or 128 CONFIGURE.

Section 3
RAMDOS

RAMDOS is software which allows you to use the CMD 1750/1750XL to emulate a disk drive. The software is installed into the computer, causing it to treat the REU as an additional disk drive. This dramatically increases the speed of disk operations with compatible software. The versions supplied support capacities of up to 2 MB.

RAMDOS is a compatible subset of the Commodore disk operating system used on drives like the 1541, 1571 and 1581. It is much faster, however, because once the program is in the REU, accessing it no longer involves the slow methods used by mechanical disk drives.

Some applications (especially games) may not work with the RAMDOS installed. This is because the games depend on the physical aspects of the disk drive, sometimes executing proprietary code within the disk drive itself. Those applications that directly read and write tracks and sectors or attempt to execute proprietary code within Commodore disk drives cannot be run properly with RAMDOS. Other programs may not be compatible because they overwrite the portion of RAMDOS that resides in your computer. For greater compatibility between an REU and commercial software, you should investigate the addition of a RAMLink to your system. RAMLink allows compatibility with most commercial software programs that cannot recognize an REU.

Installing RAMDOS

To boot RAMDOS, insert the CMD REU UTILITIES disk into your disk drive (device 8) and enter the following command:

```
LOAD"RAMDOS.BAS",8
```

Be sure to press <RETURN> to enter the command. When you see the READY prompt, type RUN and press <RETURN>.

After the program loads, you are asked for a unit (device) number to assign to the REU. The default is 9, but you can change this to suit your own specific needs. You can also change the device number of the RAM disk later, using the U0>CHR$(n) or M-W commands. Try to avoid using a device number that would conflict with other devices on your system, and don't forget that you'll probably need at least one other device to transfer programs and/or data in and out of the REU.

Next, you must specify a page in the computer's memory where the RAMDISK INTERFACE PAGE will be located. Unless you have a good reason for putting it

elsewhere in memory, accept the default, which is 207 ($CF00) on a C-64 and 14 ($0E00) on a C-128. See the RAMDOS technical details section for further information.

You are then prompted to initialize (clear) the REU. Normally, you should answer yes by pressing <Y> to initialize the REU. *Always initialize the RAM disk after a power-up.* If your computer has crashed or been reset but hasn't been powered down, or if you use some means of power backup for your REU, your RAM expander may contain information from a previous session. Under those circumstances you can attempt to recover the contents of the REU by answering no (press <N>) in response to this prompt.

The program will print some messages while installing, then end.

General Capabilities Supported by RAMDOS

The following Commodore disk drive operations and/or enhanced capabilities are provided by RAMDOS.

- RAMDOS supports fifteen concurrent channels, specified by the secondary address. Channels 0 and 1 are treated like channels 2-14. Channel 15 is the command/error channel.
- All accesses to the directory are treated as the interpreted version, which has a memory format similar to a BASIC program (i.e., line numbers and links are included).
- Wild cards (? and *) are supported.
- Save with replace (SAVE"@FILENAME", n) is supported.
- RAMDOS supports file level activity (direct access to tracks and sectors is not supported).
- All standard filetypes are supported (PRG, SEQ, REL, and USR files).
- The DOS interface to the kernel is through the standard indirection vectors for the following standard kernel calls: OPEN, CLOSE, CHKIN, BASIN, BSOUT, GETIN, LOAD, SAVE, INMI (low level routines like TALK, TKSA, LISTN and SECND are not supported).

RAMDOS Command Channel Commands

RAMDOS behaves like any normal disk drive. It has a directory, and supports PRG, SEQ, REL and USR files. It has a command channel and an error channel, just like a 1541, 1571 or 1581 drive. However, not all DOS commands are supported, due to a different track and sector organization.

The following DOS Command Channel commands are supported:

Rename	(change the name of a file)
Copy	(only one file; no concatenation)
Scratch	(erase a file)
New	(format the RAM disk)
Init	(close all files)
Validate	(returns OK)
Pointer	(used for REL files)
UJ	(same as init)
UJ0	(full init)
UI	(same as init)
U0>chr$(n)	(set unit number, where n = unit number)
M-W	(Memory-Write, see note below)

Note: The M-W command may be used to change the unit number of the disk drive by using it to write to locations 119 or 120. The RAMDOS designates this single area to write to. Attempts to write to different areas in RAMDOS memory are trapped as syntax errors. Note that with the M-W command, a colon is mandatory. The following DOS commands are NOT supported:

&	Auto-Execute
B-R	Block-Read
B-W	Block-Write
B-A	Block-Allocate
B-F	Block-Free
B-P	Buffer-Pointer
M-R	Memory-Read
M-E	Memory-Execute

RAMDOS Technical Details

The RAMDOS software that has been supplied with your CMD REU is made up of the following files:

RAMDOS.BAS
RAMDOS64.BIN4.3
RAMDOS128.BIN4.5

RAMDOS.BAS is a small BASIC program that guides you through the process of installing RAMDOS into your computer. It provides work-arounds for more advanced users should software incompatibility problems arise.

RAMDOS64.BIN4.3 and RAMDOS64.BIN4.5 are the actual binary code files (for the 64 and 128, respectively) which are loaded into the computer by the RAMDOS.BAS program. After loading, RAMDOS is transferred into the REU itself, freeing up computer memory for other uses.

RAMDOS Operation

RAMDOS installation overwrites system memory. Once the DOS is installed, that system memory is again free for usage, with the following exceptions:

- Indirection vectors; these vectors now point to the RAM disk interface block, and the interface block contains the vectors' former values.
- RAMDOS interface page; this page of code may be installed anywhere in the range $0200 through $2000 in the C-128, or anywhere from $0200 through $CF00 in the C-64, provided the selected page is not lying under I/O or ROM. The default is the page starting at $CF00 in the C-64, and $0E00 in the C-128.

Execution of System I/O Calls

All system I/O calls are made through vectors which point to locations in the DOS interface page. Code in the interface page determines whether the call is for the REU or another system device. If the call is not for the REU, control is passed on according to the values of the indirection vectors prior to DOS installation.

If the call is for the REU, code in the interface page swaps the DOS with some normal system RAM and passes control to the DOS. The DOS then processes your call (which can involve several swaps of DOS and user RAM), and upon completion returns control to the interface page code.

The interface page code swaps out the DOS, restores the user's memory map, and returns control to the calling routine.

Interrupt Operation

The DOS never creates interrupts. During the time the DOS is swapped in, IRQ's are disabled, to prevent DOS operation from interfering with your programs.

NMI's are not disabled. Any NMI code used by your programs should never access or reference data within the area under DOS control. To counter this, use a DOS image which does not use memory accessed by your NMI routine.

Note that the only NMI's caused by the operating system are for RS-232 and RUN/STOP RESTORE operation.

Concurrent RAMDOS and RS-232 operation is not possible because of the time required by the swapping process, which is not interruptible. Do not attempt to use the RS-232 to receive or transmit while accessing the REU.

RUN/STOP RESTORE

RAMDOS intercepts this operation and is unaffected when you press RUN/STOP RESTORE. During DOS calls, RUN/STOP RESTORE is disabled. You may have to press these keys several times to get it to work during intense disk activity.

Section 4
C-128 Mode Use

BASIC Commands

There are three BASIC commands built into BASIC 7.0 on the C-128 that were designed especially for use with the REU. These commands handle the three types of data transfers possible with the REU:

STASH   Stores data from the C-128 memory into the REU.
FETCH   Retrieves data stored in the REU and places it into C-128 memory.
SWAP   Exchanges data stored in the REU with data stored in the C-128 memory.

Each of these commands are followed by four parameters that define the block of data to be transferred. The parameters are the same for all three commands:

```
COMMAND bytes, insta, expsa, expba
```

Where: *bytes* = number of bytes involved in the transfer
       intsa = starting address of C-128 RAM (0-65535)
       expsa = starting address of expansion RAM (0-65535)
       expba = expansion RAM bank number (0—7 on the CMD 1750 and 0—15* on the CMD 1750XL)

These commands can cause the REU to wrap to the next bank of memory, unless that bank is on a 512K byte boundry, in which case a wrap to the first bank of that 512K segment will occur. This latter condition will only occur on the CMD 1750XL, since the CMD 1750 contains only 512K.

IMPORTANT NOTE: The BASIC commands STASH, FETCH and SWAP will generate an illegal quantity error if a bank number above 15 is used. This means you cannot use these commands to access banks 16—31 on the CMD 1750XL.

BASIC Example Program

The following is a short example of how you can transfer data in BASIC using the STASH and FETCH commands. The program will save a copy of the data displayed on the screen into the REU. Once this data has been saved, it will clear the screen, print a counter value, pause for 1 second, and then recall the data from the REU. The

program will loop through this sequence ten times, and then end. This example gives some insight into how simple animations can be displayed by using an REU.

Please follow the steps provided carefully, and be sure to check your typing before you RUN the program.

Step 1 Reset your computer in 40 column mode so that the power-up message is visible.

Step 2 Enter the following program lines. Be sure to press <RETURN> at the end of each line.

```
10 STASH1000,1024,0,0
20 FORX=1TO10
30 PRINTCHR$(147)
40 PRINTX
50 SLEEP1
30 FETCH1000,1024,0,0
40 SLEEP1
60 NEXT
```

Step 3 Type RUN and press <RETURN> to start the program.

This example shows how simple it is to save away sections of memory and bring them back when needed. Of course RAMDOS can also be used to SAVE and LOAD memory to provide similar results.

Section 5
CP/M Mode Use

Booting CP/M

If you are a CP/M user, you are familiar with the CP/M system disk packed with your C-128 computer. Early releases of the CP/M software packed with the C-128 computer do not support the REU. An upgraded CP/M system disk, that supports the REU was later released by Commodore. CMD has this version available if you need the upgrade.

To start CP/M you have two options:

1 Turn off the computer. Place the disk in the drive, then turn on the computer. CP/M will boot automatically.

2 Start in C-128 mode, with equipment turned on. Put the CP/M system disk in the disk drive and type the BASIC command:

```
BOOT <RETURN>
```

Once the system is booted, CP/M will function in the usual manner with up to four external disk drives labelled A through D and the virtual drive E. Follow the instructions in your C-128 Computer System Guide for using CP/M without the RAM expansion module. Provided below are instructions for using the RAM expansion module like a fifth drive.

Using CP/M Commands

Once the system is loaded, you can use the REU like a fast external drive simply by referring to drive M (Memory drive).

For example, to save time during a work session, you might want to store some of the more commonly used transient utilities in the REU. Here is the method:

1 With the CP/M utility disk in the external drive A, type:

```
PIP M: = A:FORMAT.COM [v] <RETURN>
```

In this example, the PIP utility would search the disk in drive A, locate the FORMAT.COM file, and store it in the RAM expansion module. The [v] verifies that the transfer occurred correctly. You can also use DIR M: to

determine what is stored in the REU.

Any filename could be used in place of FORMAT.COM. Be sure to use the correct filetype in addition to the filename if necessary.

2   When the computer displays the A> prompt, the transfer is complete. Now the FORMAT utility is stored in the RAM disk.

3   To compare the speed of execution from an external drive and from the REU, type the following at the A> prompt:

```
FORMAT <RETURN>
```

The disk drive will whir and eventually instructions for formatting a disk will appear on screen. Abort the format by pressing <RETURN>, then any key, then answering N to the question, "Do you want to format another disk?"

Now log on to the M drive by typing M: at the A> prompt. Load the FORMAT command from the RAM expansion module. To do this, at the M> prompt, type:

```
FORMAT <RETURN>
```

Almost instantly the format command is retrieved from the REU and displayed on the screen. Alternately, you can choose to execute the FORMAT command without logging to drive M> by typing M:FORMAT at the A> prompt.

This example should illustrate how the REU can speed up your operations. You can copy a number of files into the REU with a series of PIP commands. Or copy the entire contents of a disk:

```
PIP M: = A:
```

This will copy all the files from the disk in drive A into the REU.

Once the REU has been loaded with a program, use of the program is no different than if the program were in an external drive. If you are saving data files for later use, however, you may want to save them on an external drive instead of in the REU.

Using the reset button does not clear the expansion RAM banks, only the C-128 computer's main memory.

To avoid accidental loss of your work, periodically save programs or data files in progress on disk, as well as in the REU. A power surge, or an accidentally pulled electric cord, can have a devastating effect on your work if you only store it in the REU.

Section 6
Programmer's Reference

The RAM Expansion Controller

Memory transfers involving REUs are accomplished through the RAM Expansion Controller (REC), an I/O chip in the REU that maps into your computer's memory at address $DF00. You cannot directly access the RAM in the REU; instead, you instruct the REC to transfer data between the REU memory and computer memory. Programs can accomplish this by using POKEs, BASIC 7.0 commands, or via machine language instructions to manipulate the REC chip registers.

Remember that while you can use direct REC manipulation or RAMDOS to transfer memory contents between your computer and the REU, using them together isn't advised. RAMDOS uses all the available RAM space for itself, and direct REC manipulation could easily cause corruption of data stored by RAMDOS.

Memory Locations

In order for a safe transfer, you must be aware of the computers memory map. You are moving blocks of data in and out of main and expansion memory, so you must know the correct addresses to retrieve the blocks of data you intend to and store them in the proper places. You'll have to be careful not to put things in memory where they don't belong, e.g. over another program or memory allocated for a hardware function. If you transfer data into an area that is already occupied by a program or other data, you will overwrite the original data, which may affect the operation of computer. All the standard rules for C-64 and C-128 banking and I/O apply when programming REUs.

REC Operation

The REC has four primary operating modes:

1) Transfer a block of data from main memory to expansion memory.
2) Transfer a block of data from expansion memory to main memory.
3) Exchange a block of main memory with a block of expansion memory.
4) Verify a block of main memory with a block of expansion memory.

The REC has several internal registers which can be used to set up a particular operating mode. Modes can be selected by setting the appropriate bits in the Command Register (see Register Information). The starting computer address,

Direct Memory Access Operation

Direct Memory Access (DMA) is the process the REC uses to transfer data to and from the computer's memory. During DMA, the computer's processor is temporarily halted so the REC may access the computer's memory. There are two events which will cause the REC to begin a transfer sequence.

1 If the FF00 option in the Command Register ($01) is disabled, the DMA begins right after the execute bit is set in the Command Register.

2 If the FF00 option is enabled, the DMA begins immediately after a write to address FF00 is detected, providing the execute bit was previously set in the Command Register.

The purpose of the FF00 option is to allow the REC to access the RAM beneath I/O space. The REC registers are I/O mapped on the C-64 and C-128 expansion bus, which means that, without the FF00 option, I/O space would be enabled when DMA was initiated. This option, therefore, allows the user to bank out the computer I/O space, replacing it with the RAM underneath, before the DMA takes place. The FF00 option is cleared each time it is used.

Special Features

AUTOLOAD

When you select this option in the Command Register, the computers Base Address Registers, the expansion memory Base Address Registers, the expansion memory Bank and the Byte Counter Registers at the end of a transfer are automatically reloaded.

This is useful if one operation is to be executed repeatedly on one particular block of data. Note that if AUTOLOAD is selected in verify mode, the address where the verify error occurred is lost. Ordinarily, upon finding a verify error, the REC halts the DMA cycle and both Address Registers and the Bank Register point to one location above the address that failed to verify.

Address Control

Another special feature of the REC is the ability to hold a source and/or destination address fixed in any mode. This is accomplished by setting the appropriate bits in the Address Control Register ($0A). The default is that addresses will increment for both source and destination. It is possible, with a fixed computer address, to DMA to any I/O device except the C-128's 8563's internal RAM, the 6510/8502 I/O ports at $00 & $01, the C-128's 8722 MMU, and the REC itself.

REC REGISTER INFORMATION

Address	Bits	Function	
$DF00	7-0	**Status Register (read-only)**	
	7	Interrupt pending	1=Interrupt waiting to be processed
	6	End of Block	1=Transfer Complete
	5	Fault	1=Block verify error
	4	REU Size	1 on CMD 1750 and 1750XL
	3-0	Version	0000 on CMD 1750 and 1750XL
		Note: Bits 7-5 are cleared when this register is read	
$DF01	7-0	**Command Register (read/write)**	
	7	Execute	1=Transfer per current settings
	6	Reserved	
	5	Load	1=Enable AUTOLOAD option
	4	FF00	1=Disable $FF00 decode
	3	Reserved	
	2	Reserved	
	1-0	Transfer Type:	00=Computer to REU
			01=REU to Computer
			10=Swap REU and Computer
			11=Verify Computer with REU
$DF02	7-0	**Computer Base Address, LSB (read/write)**	
		Lower 8 bits of Computer Base Address	
$DF03	7-0	**Computer Base Address, MSB (read/write)**	
		Upper 8 bits of Computer Base Address	
$DF04	7-0	**REU Base Address, LSB (read/write)**	
		Lower 8 bits of REU Base Address	
$DF05	7-0	**Computer Base Address, MSB (read/write)**	
		Upper 8 bits of REU Address	
$DF06	2-0	**CMD 1750 Expansion RAM Bank (read/write)**	
		Expansion RAM Bank Pointer	
	4-0	**CMD 1750XL Expansion RAM Bank (read*/write)**	
		Expansion RAM Bank Pointer	
		Important Note: Bits 3 and 4 are write-only!	
$DF07	7-0	**Transfer Length, LSB (read/write)**	
		Lower 8 bits of Transfer Length	
$DF08	7-0	**Transfer Length, MSB (read/write)**	
		Upper 8 bits of Transfer Length	
$DF09	7-5	**Interrupt Mask Register (read/write)**	
	7	Interrupt Enable	1=Interrupts enabled
	6	End of Block Mask	1=Interrupt on End of Block
	5	Verify Error Mask	1=Interrupt on Verify Error
$DF0A	7-6	**Address Control Register (read/write)**	
		00=Increment both addresses (default)	
		01=Fix REU addresses	
		10=Fix Computer addresses	
		11=Fex both addresses	

Interrupts

An interrupt can be generated on an end of block condition or verify error by enabling interrupts and setting the corresponding mask bits in the Interrupt Mask Register ($09). In either case, the corresponding flag is set in the Status Register. Interrupt flags are cleared upon reading the Status Register. The Interrupt Mask Register bits stay as they were before the interrupt/clear sequence. For proper operation, the Status Register ($00) must be read at least once before a DMA is initiated with interrupts enabled. It is important to clear out the previous status before any mode is used with the interrupt option.

Operation Details

If a specified address range exceeds the address range in the selected REU expansion bank, a 'wrap' occurs into the next bank, This happens whenever the given byte counter exceeds the number of bytes left in the selected bank. The computer side always wraps to the beginning of the same bank.

Under normal operation (no AUTOLOAD, address increment), both Address Registers point to the next sequential memory location outside the selected transfer range at the end of the transfer. This is true for any mode and applies to both Base Address pointers except one that is held fixed. The Bank pointer is also updated (if necessary) at the end of the transfer. Also note that under normal operation, the Byte Counter decrements to the value 1. Care should be taken, therefore, to check the Transfer Complete bit in the Status Register and not rely on the Byte Counter value to indicate an end to the transfer condition. A Byte Counter of 0 results in a transfer of a full 64 Kilobytes. Again, wrapping occurs in all modes of operation.

Important Notes

The following notes should help you in using and programming the CMD 1750 and 1750XL REUs.

- The Z80 must not be in control of the C-128 when a DMA is requested.
- The C-128 MUST be in 1 MHz (SLOW) mode when a DMA is requested.
- The REC cannot re-configure the C-128's MMU during a DMA cycle.
- The BASIC commands STASH, FETCH, and SWAP can only be used to transfer and retrieve data in Bank 0 of the C-128 internal memory. To access Bank 1 of the C-128 it is necessary to alter a special control register on the C-128. This register is the RAM control register in the MMU at address $D506. Bits 6 and 7 of this register control both the VIC RAM bank pointer and the C-128 DMA bank pointer. To transfer into Bank 1 set bit 7 of the MMU register:

```
POKE DEC("D506"),PEEK(DEC("D506"))OR128
```

Then do the DMA command into Bank 1. To reset the VIC video back to Bank 0 use:

Interrupts

An interrupt can be generated on an end of block condition or verify error by enabling interrupts and setting the corresponding mask bits in the Interrupt Mask Register ($09). In either case, the corresponding flag is set in the Status Register. Interrupt flags are cleared upon reading the Status Register. The Interrupt Mask Register bits stay as they were before the interrupt line sequence. For proper operation, the Status Register ($00) must be read at least once before a DMA is initiated with interrupts enabled. It is important to clear out the previous status before any mode is used with the interrupt option.

Operation Details

If a specified address range exceeds the address range in the selected REU expansion bank, a "wrap" occurs into the next bank. This happens whenever the given byte counter exceeds the number of bytes left in the selected bank. The computer side always wraps to the beginning of the same bank.

Under normal operation (no AUTOLOAD, address increment), both Address Registers point to the next sequential memory location outside the selected transfer range at the end of the transfer. This is true for any mode and applies to both Base Address pointers except one that is held fixed. The Bank pointer is also updated (if necessary) at the end of the transfer. Also note that under normal operation, the Byte Counter decrements to the value 1. Care should be taken, therefore, to check the Transfer Complete bit in the Status Register and not rely on the Byte Counter value to indicate an end to the transfer condition. A Byte Counter of 0 results in a transfer of a full 64 Kilobytes. Again, wrapping occurs in all modes of operation.

Important Notes

The following notes should help you in using and programming the CMD 1750 and 1750XL REUs.

- The Z80 must not be in control of the C-128 when a DMA is requested.
- The C-128 MUST be in 1 MHz (SLOW) mode when a DMA is requested.
- The REU cannot re-configure the C-128's MMU during a DMA cycle.
- The BASIC commands STASH, FETCH, and SWAP can only be used to transfer and retrieve data in Bank 0 of the C-128 internal memory. To access Bank 1 of the C-128 it is necessary to alter a special control register on the C-128. This register is the RAM control register in the MMU at address $D506. Bits 6 and 7 of this register control both the VIC RAM bank pointer and the C-128 DMA bank pointer. To transfer into Bank 1 set bit 7 of the MMU register:

```
POKE DEC("D506"), PEEK(DEC("D506"))OR128
```

Then do the DMA command into Bank 1. To move the VIC video back to Bank 0 use: