

LDOS" 5.3 Model 4 Interface Kit

Table of Contents

General Information	1
KI4/DVR - Keyboard driver	2
SET2RAM/CMD - Memory Management Handler	4
MEMDISK/DCT - RAM disk driver	5
BANKEK/CMD - Memory bank utility	8
Technical Information	9
Warranty and Customer Support12

LDOS" 5.3 Model 4 Interface Kit: Copyright (c) 1987 MISOSYS, Inc., All rights reserved. Published by MISOSYS, Inc., PO BOX 239, Sterling, VA 22170-0239.

The distribution and sale of this program is intended for the personal use of the original purchaser only and for use only on the computer system noted herein. Furthermore, copying, duplicating, selling, or otherwise distributing this product is expressly forbidden. In accepting this product, the purchaser recognizes and accepts this agreement. The purchaser is entitled to make as many working copies of this disk as is needed for his or her personal use.

General

The software in this package enables you to make use of certain Model 4 hardware features while using it in the Model III mode running LDOS 5.3. Specifically, it permits access to all keyboard keys, adds a bank switching memory management handler, and provides a driver to utilize one or two memory banks in a 128K Model 4 as a RAM disk (similar to the DOS MemDISK/DCT utility). Even though your Model 4 is capable of operating its video screen in an 80x24 mode, we have chosen not to support this operation in Model TII mode due to the inevitable conflict with most commercial software packages in existence for the Model III operation.

Normal Model III-type operation on your Model 4 uses a hardware mode where the machine duplicates a standard Model III - the first 14K of memory addressing is Read Only Memory (ROM). When you install the memory management handler provided with this package, it switches your machine to a mode where the first 14K is accessed as Read/Write memory. The major difference is that you now can alter the contents of this "ROM" image portion with pokes although it is not recommended that you do so.

The files on the accompanying DATA diskette may be easily copied to your working SYSTEM disk by means of the LDOS COPY utility. There may be a file named "README/TXT" on the disk. If so, this file will contain important information which may not appear in this printed documentation. You should read this file by issuing the command: LIST README.

KI4/DVR - Keyboard driver

This keyboard driver completely replaces the KI/DVR keyboard driver supplied with your LDOS 5.3. KI4/DVR is used to enable access of all keys on your Model 4 keyboard. The driver is easily installed using the syntax:

```
SET *KI [TO] KI4[/DVR] [USING] (parm,parm,...)
```

The allowable parameters are as follows:

Type - Activates the type-ahead feature.

Jk1 - Activates the screen-print feature.

Delay - Sets the delay until the first repeat.
This value defaults to 15.

Rate= - Sets the key repeat rate.
This value defaults to 1.

As this driver is established with the SET library command, it must be applied before any other *KI filters. When KI4/DVR is installed, the driver will reside in high memory. The amount of space varies with the installation of the modules necessary to satisfy the optional parameters (Type andJk1).

The installation and the use of this keyboard driver is virtually identical to that documented in your LDOS User Manual; thus, this documentation will discuss only the operating characteristics of KI4 which differ from the standard KI.

Function keys

The three function keys, <F1>, <F2>, and <F3>, each generate a distinct value in both the unshifted and shifted depressions. The key values generated are identical to the values generated under DOS 6.x. Note that these values remain constant when the Extended Cursor Mode (ECM) is toggled ON even though the <ARROW> keys will duplicate some of the values. This ensures that programs which expect a particular function key value will continue to "see" that value when you (or a program) toggle the keyboard to ECM. The Function key values nerated are as follows:

Key	Value	Key	Value
<F1>	81H	<sF1>	91H
<F2>	82H	<sF2>	92H
<F3>	83H	<sF3>	93H

Control key

The key marked <CTRL> is now used to specify "control" instead of the two-key combination, <SHIFT-DOWNARROW>. In fact, the latter combination will generate a key code value of 1AH. For the programmer, the code is the same value as that generated by <CTRL-Z> with the exception that it is not flagged as a control code. The flagging is only significant to those programs which make use of the Z80 flag state following a keyboard driver call.

Setting caps lock

The <CAPS> key is used to toggle the caps-lock mode ON and OFF. This replaces the <SHIFT-0> two-key combination as used by the KI/DVR. When caps-lock is ON, the twenty-six keys <a>-<z> generate capital letters <A>-<Z>.

Miscellaneous

The <SHIFT-0> two-key combination now toggles the value generated by the <UPARROW> key between 5BH and 0BH. The "0BH" value is normal for a Model 4 operation while the "5BH" value is normal for Model III operation. The <CLEAR-SHIFT-0> three-key combination now generates a value of 6A0H (160d) which is the same as <CLEAR-SPACE>.

SET2RAM/CMD - Memory Management Handler

This utility switches your Model 4 computer, which is operating in the Model III ROM mode (or ROM-image mode), to a low-memory RAM mode (RAM from X'0000'-X'37FF'). It populates that RAM with an image of what was in ROM. Furthermore, it inhibits the use of Model III cassette I/O while it re-uses the memory space previously used by the cassette routines. The space is now used for a memory management handler (two service calls, @BANK and @EXMEM, used for bank switching) which enables you to easily program access to the extra 64K of a 128K Model 4. The SET2RAM utility should be usable on a Model 4, 4P, or 4D computer. The utility is easily invoked with the syntax:

```
|-----|
| SET2RAM |
|-----|
```

When the utility is invoked, it will perform a diagnostic check of your machine. If it does not interpret your machine as being a Model 4, 4P, or 4D, it will terminate with the error message:

```
Can't switch into RAM model
```

When SET2RAM determines your machine to be capable of Model III RAM-mode operation, it installs the @BANK handler for bank switching services and the @EXMEM handler for extended memory management. It also checks whether your machine has 64K or 128K of memory installed and establishes the proper values for the @BANK handler. The extra 64K, if present, is designated as banks 1 and 2; the upper 32K of the regular 64K of memory is designated as bank 0. Successful implementation of SET2RAM is observed by noting the display of the message:

```
Extended memory handler now loaded
```

The continued operation of your computer will be mostly transparent to this mode of operation.

Caution

When your computer is operating like a normal Model III, the first 14K of memory, which contains the BASIC interpreter and communications interface, can not be corrupted by being written to - it is unchangeable. Once SET2RAM has been installed, however, this memory region can be corrupted if written to. It is not normally the practice of programs to overtly attempt to write to this region. Unfortunately, some programs may have made use of the fact that this region could not be corrupted and have thus utilized a "write-to-ROM". One such program known to MISOSYS as writing to ROM during a verify function is the old Model III hard disk driver, TRSHD3/DCT, which was part of the Radio Shack Hard Disk Operating System (a version of LDOS 5.1.x). This program can be fixed by applying the following patch:

```
PATCH TRSHD3/DCT (D04,42=38:F04,42=00)
```

The patched copy is transparent to operation under LDOS 5.3 or earlier releases of the DOS. If you find strange behavior from other programs while operating in this RAM-image mode, please contact the publisher of the program for a fix.

MEMDISK/DCT

This driver makes use of the memory management handler installed by the SET2RAM utility. If you have the extra 64K of memory installed in your Model 4, you can establish it as either a 32K or a 64K disk drive emulated in memory. The RAM disk driver needs to use one of the eight logical drive slots available under LDOS. It is easily invoked with the command:

```
SYSTEM (DRIVE=d ,DRIVER=MEMDISK[ ,DISABLE])  
DRIVE=d specifies the logical drive number.  
DISABLE required if the "d" slot is active.
```

With this invocation, the following menu is displayed:

```
MemDISK 5.3a - LDOS 5.3 extended memory RAM drive  
Copyright 1987 MISOSYS, Inc., All rights reserved  
  
<B> Bank 1  
<C> Bank 2  
<D> Banks 1 and 2  
<E> Disable MemDISK  
Which type of allocation - <B>, <C>, <D>, or <E> ?
```

Note that there is no "<A>" selection. Under DOS 6.x, the <A> selection is used for emulating a small RAM disk in bank 0. Since the Model III mode of operation provides a smaller usable address space, this selection has not been

implemented; however, the remaining four selections have been designated through <E> to provide consistency with the memDISK selections under DOS 6.

Selections and <C> both provide a small 32K disk emulation in either bank 1 or bank 2 respectively. Selection <D> provides a 64K disk emulation using the full extra 64K (banks 1 and 2).

Selection <E> disables a memdisk which had been previously installed. After being disabled, the memdisk can be re-enabled by another invocation of the SYSTEM command with the appropriate selection, , <C>, or <D>.

After making your selection of , <C>, or <D>, you need to select whether the "disk" is to be single density (SDEN) or double density (DDEN). The difference is that SDEN provides 1.25K per granule while the DDEN mode provides 1.5K per granule. You also get 64 file slots under SDEN versus 128 file slots under DDEN. Make your selection in response to the prompt:

Single or Double Density <S,D> ?

After you select the density, you have to "format" the memdisk if it is a new one (not one being re-enabled). Request formatting by answering the prompt:

Do you wish to Format it <Y/N> ?

Memdisk will verify the usability of the extra memory designated by your selection. While it is doing this, it will display the message:

Verifying RAM cylinder xx

with "xx" increasing in value starting from 00. Once the verification is complete, you will see the messages:

Verifying Complete, RAM good
Directory has been placed on Cylinder 1
MemDISK Successfully Installed

The memdisk has now been installed and is ready for service. The drive number to be used for it is the number you designated as "d" in the installation SYSTEM command.

If you select option <E>, your memdisk will be disabled and the following message will be displayed indicating its success:

MemDISK now disabled

Memdisk will check whether the selections you have made are legitimate. When it detects an error or other incompatibility with your command requests, it will display some appropriate error message. One or more of the following error messages may be displayed during your installation procedure. Each message is followed by some text appropriate to the underlying reason for the error.

Machine not in RAM mode! - This indicates that the SET2RAM memory management handler has not yet been installed. Memdisk/DCT requires this handler.

Must install via SYSTEM (DRIVER=! - MEMDISK/DCT was invoked without using the SYSTEM command of LDOS. You must use the SYSTEM command as specified in the syntax box discussed above.

Logical drive number required! - The SYSTEM command was invoked without specifying a "DRIVE=d" parameter. This parameter must be entered to assign the driver to a drive slot.

Can't specify SYSTEM drive slot! - Thememdisk driver cannot be initially assigned to drive 0 which is reserved for use as a SYSTEM drive. You should be able to BACKUP the system to an installedmemdisk then switch to it as drive 0; however, it first has to be designated as a drive numbered 1-7.

Unable to install MemDISK, requested bank in use! - You requested the use of a memory bank which has already been used and reserved by some other program.

MemDISK already Active - You attempted to install amemdisk; the installation program detected that amemdisk was already present in memory. This may occur in two cases: (1) Memdisk is already assigned to another drive slot, or (2) an installed memdisk was disabled via a SYSTEM (DRIVE=d,DISABLE) command; it must first be enabled with a SYSTEM (DRIVE=d,ENABLE) command.

Destination MemDISK contains Data, Do you wish to Format it <Y/N) ? - A disabled memdisk has been detected. If you want to reuse its contents, answer <N>. This allows you to recover the files which were written to the old memdisk provided the memory region was not overwritten with something else. If you want to proceed and re-initialize thememdisk, answer <Y>. You do not want to format the memdisk if you want to recover the files stored on thememdisk after having to re-BOOT.

MemDISK not present, installation aborted! - This error indicates that you specified <N> to formatting; yet no previously installedmemdisk was detected. You must format the first timememdisk is installed.

MemDISK type mismatch, format required! - You requested the re-use of a previously installed but disabledmemdisk by means of answering the Format? query with <N>. The installation portion ofmemdisk, however, detected the previous installation as being different from the one currently being requested (i.e. old was type and new is not , etc.).

Verify Error in Bank n at locationX'nnnn'! - The verification of memory which occurs during formatting has detected a memory error. The address where the error has occurred is displayed as nnnn'. The bank number is displayed as "n". You may have to have a memory chip replaced.

Target Drive not a MemDISK! - You selected option <E> (to disable amemdisk) yet the drive number designated in theDRIVE=d parameter of the SYSTEM command does not have a memdisk assigned to it.

BANKER/CMD - Memory Bank Utility

Banker will allow you to reserve or release a single memory bank or a range of consecutively numbered memory banks. BANKER will also display a map providing status of all memory banks. The utility is invoked by the command:

```
BANKER (Reserve=val,Free=val)

Reserve=val - specifies the bank(s) to reserve.

Free=val    - specifies the bank(s) to free up.
```

As can be observed, both parameters can be abbreviated to a single character. The "val" parameter can be either a single number or a number range entered as a string. For instance, the string "1-2" references banks 1 through 2 inclusive. You may also enter a single number as a string, as in "12". Thus, a typical command to reserve the first two external memory banks would be:

```
BANKER (R="1-2")
```

Both parameters may be entered with one command invocation. Reserve is processed first. Therefore, if your selections overlap, the released banks will be that specified by the FREE parameter. Note also that the status map will be displayed first with the existing status and will be subsequently displayed after each parameter is serviced.

BANKER does no error checking on your bank numbers other than ensuring that the starting bank number does not exceed the ending bank number in a string parameter entry. BANKER relies on the DOS to perform error diagnostics on the bank number passed in the @BANK service call (part of the memory management handler installed by SET2RAM).

The memory map displayed by BANKER looks like the following:

```
32K Banks avail = 03/03, In use = <--->
```

With the release of several memory expansion options available for the Model 4 (TRS-80), some enterprising MISOSYS customers may implement an @BANK handler for those extended memory boards to work with the SET2RAM memory management facility. BANKER will work with up to 32 banks of memory installed and accessible through an @BANK handler. 4 Interface Kit

Technical Information

This section is reference material designed for the programmer who wishes to use the memory management handler installed by SET2RAM. It also documents what parts of the ROMC memory image (X'3000'-X'37FF') are used by this package.

General

In investigating the ROMC code of various Model 4 computers, we have made the following determination. The address space from 3000H through 37FFH, known as ROMC, contains at least four regions fixed across ROM releases that are

reusable if the cassette routines are inhibited and the ROM RS232 driver is not used (it is not used under LDOS). These regions are:

1. 31COH - 31F6H cassette routine region
2. 3203H - 338DH cassette routine region
3. 35FBH - 36A9H RS-232 driver region
4. 36AAH - 37388 BOOT data moved to RAM

Other re-usable regions do exist but the addresses vary with ROM release; thus, MISOSYS is not using them. The SET2RAM utility inhibits the cassette and RS-232 vectors by setting the first byte of their jump address to a RET (0C9H). Region 3 and region 4 is used for the @BANK and @EXMEM handlers. The first two bytes (X'35FB' & X'35FC') contain the address of the last byte used by these routines. The @EXMEM handler comes next followed by the @BANK handler. If you are using some add-on memory board and wish to install your own handler, the starting address of @BANK is readily discernable by examining the memory region with DEBUG. Once you determine the start of the @BANK code, design your handler to overwrite this region. The @EXMEM extended memory management handler makes use of the @BANK routines but always through the vector address of @BANK.

The @EXMEM handler uses the first 256 bytes of Region 2 (32038-33028) as a low-memory intermediate page buffer. The capabilities added by @EXMEM will make it very easy for you to use the bank switching facility for data storage and retrieval since it takes care of the stack, interrupts, and buffer locations.

The region starting at 3303H is reserved for use by MemDISK.

@BANK - 3012H - Bank switching service

This service call deals with memory bank use. It's vector address is X'3012'. The top half of the first 64K block is bank 0, and the second 64K is banks 1 and 2. The @BANK handler provided with SET2RAM supports a total of 3 memory banks of 32K each (numbered 0-2). As the LDOS 5.3 architecture does not support automatic bank switching during device I/O and machine interrupt processing, ALL programs incorporating the @BANK handler must ensure that the machine's interrupts are disabled during program-controlled bank switching. Also, as the banked memory is imaged within the 8000H-0FFFFH address space, the stack pointer must be below 7FFBH.

If you are familiar with programming the TRSDOS 6.x @BANK SVC, this @BANK handler should behave identically except for the requirement that bank switching must occur only when interrupts have been disabled. In the interfacing register protocol identified below, register-B passes a function code.

In general, the registers affected will be AF and BC, and HL if a transfer is requested.

Bank Request [optional transfer]

- B => 0; Select bank in C.
- C => Bank number (0-7). Optionally set bit-7 to transfer to the address specified in register pair HL.
- HL => Optional address to transfer to in the new bank. This option is selected by setting bit-7 of register-C.
- B <= Returns a 0.

C <= Returns the previously resident bank number (0-7). If a transfer has been specified (via bit-7 set), bit-7 will remain set.
A <= Returns any error code if NZ condition.
NZ <= Bank not there.

Bank Release

B => 1; Reset bank in C.
C => Bank number.

Bank Availability Test

B => 2; Test if bank C in use.
C => Bank number.
NZ <= In use.

Bank Reservation Request

B => 3; Set bank in C.
C => Bank number.
NZ <= Already in use.

What Bank is Resident

B => 4; Return current installed bank.
C => A valid bank number.
A <= Returns the bank number (0-7) of the currently resident bank.

Note: The @BANK handler will not return an error state if you try to reset a memory bank that is "in-use" because it is not installed. The way in which bank-reset should be performed is to preserve the number of the bank you have reserved and made in-use and pass that number to the bank release function when you no longer need the memory bank. Note that even though @BANK permits you to reset a non-existent bank, if you try to enable it, you will get an error since the enabling routine will not permit the selection of a bank not installed.

@EXMEM - 3015H - Extended memory management

The extended memory handler allows for easier access of extended memory storage than through the normal @BANK handler. This facility gives you both character (one byte) and page (256 bytes) I/O by bank and relative address. As the @EXMEM handler provides its own control over both disabling of interrupts and stack management, the programmer can ignore those maintenance items. This makes it easy to use extended memory from languages which use a high memory stack such as BASIC and C.

The vector address of @EXMEM is X'3015'. The following information notes the protocol of the four extended memory management functions.

Get a character from a bank

Registers affected: AF, B
B => 1
C => Requested bank
HL => Relative offset in the bank, 0000H-7FFFH
A <= The character returned
Z <= Set if successful

Put a character into a bank

Registers affected: AF, B
B => 2
C => Requested bank
A => The character to put
HL => Relative offset in the bank, 0000H-7FFFH
Z <= Set if successful

Get a page (256 bytes) from a bank to your buffer

Registers affected: AF, B, HL, DE
B => 3
C => Requested bank
HL => Relative offset in the bank, 0000H-7FFFH
DE => Address of your 256-byte buffer
Z <= Set if successful

Put a page (256 bytes) from your buffer to a bank

Registers affected: AF, B, HL, DE
B => 4
C => Requested bank
HL => Relative offset in the bank, 0000H-7FOOH
DE => Address of your 256-byte buffer
A <= The character returned
Z <= Set if successful

Note that the relative offset can also be in the range 8000H-OFFFFH. The @EXMEM handler always sets the high-order bit of register H to ensure that the bank offset value is addressed into the bank swap image area.

Function 4, putpage, checks to make sure that your bank copy won't wrap past X'FFFF'; although getpage will permit you to read past X'FFFF'. There is no error checking on the address of your buffer in either function 3 or function 4. If your buffer is below X'5200', it is possible to corrupt the system; however, since you may be wanting to swap out low memory, @EXMEM does not prohibit it.

@EXMEM always does double buffering through a low memory intermediate buffer in case your buffer overlaps the bank swap image area. Although this may be a tad slower in operation when it is not necessary (i.e. when your buffer is totally below X'8000'), it enables the extended memory handler to be smaller; thus providing maximum space for the @BANK handler.

Warranty

This software program(s) is warranted to perform as documented when used on the specified hardware operating under the specified disk operating system as shown on the accompanying documentation. If within 90 days of the date of purchase the program is found to be defective due to a bug in the code, the publisher will, upon request, provide a patch to correct the bug or will update the program diskette with a corrected copy within a reasonable time period after return of the program diskette to the publisher. If within 90 days of the date of purchase the documentation proves defective due to missing pages, the publisher will provide substitutes for the missing pages upon request.

The publisher shall have no Liability or responsibility to the purchaser or any other person, company, or entity with respect to any liability, loss, or damage caused or alleged to have been caused by this product, including but not Limited to any interruption of service, loss of business and anticipatory profits, or consequential damages resulting from the operation or use of this program.

Customer Support

Customer service information on this product may be acquired by contacting MISOSYS, Inc., at the following address:

MISOSYS, Inc.
P.O. Box 239
Sterling, Virginia 22170-0239
703-450-4181