ERAMCO
SYSTEMS

ES 83121A
ERAMCO MLDL - HARDWARE MANUAL

OWNER'S MANUAL

Februari 1984

83121A-M000002

POTENTIAL FOR RADIO AND TELEVISION INTERFERENCE (FOR U.S.A. ONLY)

The ERAMCO MLDL BOX "ES MLDL 1" generates and uses radio frequencies energy and, if not installed and used properly, that is, in strict accordance with the manufacture`s instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a class B computing device in accordance with the specifications in subpart J
of part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If your "ES MLDL 1" does cause interference to radio or television reception, you are encouraged to try to correct the interference by one or more of the following measures:

* Reorientate the receiving antenna.

* Relocate the computer with respect to the receiver.

* move the computer and "ES MLDL 1" away from the receiver.

If necessary, you should consult your dealer or an experienced radio/television technician for additional suggestions. You may find the following booklet prepared by the federal Communication Commission helpful: How to Identify and Resolve Radio- TV Interference Problems. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

Any words in RED UNDERLINING, have been added to make sure you read & understand them , otherwise the MLDL Ram (4Kb pages) may not function as expected.

INTRODUCTION

The **Eramco Systems** MLDL - Box is a thoroughly designed addition to your HP-41 system. A word of warning however, precedes its operating instructions. Read this manual from cover to cover before connecting the MLDL - Box to your system. The box makes extensive use of the capabilities of your HP-41 so make sure to understand the details of this manual before using this extension. A final but no less important warning: Do NOT, under any circumstance, connect or disconnect the box unless the calculator is switched OFF. Irreparable damage can result.

SETTING UP INSTRUCTIONS

Your **Eramco Systems** MLDL - Box is delivered in a plastic enclosure. To open the box remove the bottom part after removing the fastening screws. You do not have to replace the screws if you want to open your box often, it is advisable however to keep the screws in a safe place if the need might arise to close your box a bit more permanently. Once the box is open you will see the Eprom sockets and the Eprom enabling switches.

Two sockets are used for one Eprom set, coming to a total box capacity of 6 Eprom sets i.e. 24 k of program! The sockets touching each other at the small edge belong together and form the space for one 4k. Eprom set. You will see one Eprom set installed if your box was ordered with the ERAMCO MLDL Eprom.
If you want to insert new or other Eprom's please follow the instructions for carefull insertion of new Eprom's as ireparable damage can be done to the box, the Eprom's and eventually even to the calculator.
Each Eprom set can be enabled or disabled using the corresponding Eprom switch. The Eprom's are numbered starting at the switch side from 1 up to 6, under the Eprom sockets you will find the logical numbering ranging from 0 up to 5. To enable one set, set the switch with the corresponding number (1-6) to ON.
Do **not** switch any Eprom on yet, wait until you have read the rest of this manual.

INSTALLING NEW EPROM'S

All Eprom's have an identification indicating the pin-out, i.e. the location of pin 1. The identification can consist of a notch in the small edge of the Eprom or of a dot above pin one. This identifies pin one as the pin at the left bottom corner of the Eprom. An Eprom set consists of two Eprom's called the U2- and L8-Eprom. The U2 Eprom has the generic type number 2716 and has to be placed in the socket under the enabling switch, the L8 Eprom has the generic type number 2732 and has to be placed in the adjacent socket i.e. the socket under the print interconnecting cable. Insert the Eprom's so that pin one points away from the connecting cable to the calculator (see fig 1.1). You will notice that the sockets also have a mark to identify the location of pin one. Should the pins of the Eprom be bent outwards (as almost always will be the case using new Eprom's), place the Eprom with the pins of one edge touching a smooth surface (e.g. a table top) and slowly and **carefully** bend them inwards. Repeat this procedure for the pins of the other edge and try to insert again. Normally this will be sufficient to make the pins fit into the socket.

THE MLDL BOX PORT USAGE

It is important to know something more about your calculator and its internal mechanics to avoid potential problems.This additional knowledge concerns the user ports (1-4) and their addressing scheme (see fig 1.2). Each user port covers two pages of 4 k. bytes length. These pages are called the lower (even numbered) page and the upper (odd numbered) page. Almost all plug in Rom's delivered by Hewlett-Packard occupy only the lower page of a port, leaving the upper page unoccupied. This fact enables the user to use plug-in Rom's and the MLDL - Box simultaneously due to the design of the **Eramco Systems** MLDL - Box. In tabel 1.3 you can find which Eprom set is addressed where and how this places the Eprom image in your user ports.

Tabel 1.3

EPROM #	Address	Port
1	D	3-U
2	9	1-U
3	F	4-U
4	B	2-U
5	8	1-L
6	C	3-L

As you can see, two lower port pages being port 2 and port 4 (pages A and E) are left unused. The user can use this ports for a plug-in module and for the Card Reader. A few plug-in's have a special address placing their image outside the user ports. These plug-in's are Time-module, Printer (non IL and IL), IL-module, X-Memory module, Single Memory module and Quad Memory module. These modules can be plugged into each user port, occupying only physical space, no port-address space is used however. All other plug-in Rom's however occupy physical and address space. Should a plug-in Rom use the same address space as an Eprom set, **the plug-in Rom will be given preference** above the Eprom set, resulting in an apparent disabling of the Eprom set.

If you encounter this problem, switch the Eprom off or plug the plug-in module in another port. No damage will be done to the module by this double addressing but it is recommended to avoid these situations. After having adapted the configuration of the Eprom section of the MLDL - Box to your needs, make sure all unused Eprom sockets are disabled by switching the corresponding switch OFF. This enables you to use this address space for other plug-in's or for your Eprom pages.

THE MLDL - BOX EPROM PAGES

Turning your MLDL - Box around you will see the Eprom addressing switches and their enabling switches. Eprom stands for Erasable Read Only Memory i.e. the part of the MLDL-Box into which you can write your own data using the special MLDL-ROM instructions. To the calculator these pages have the appearance of a plug-in module (ROM). The rotary switch selects the page at which your Eprom is placed, the enabling switch permits you to use this page i.e. to read the contents of this page. Remember, the enabling switch is a read enable switch, **writing** into this page will **not** be inhibited by disabling the page! As pages Ø, 1, 2 and 4 are used for the HP-41 C(V) operating system it is recommended not to place the switches in the position corresponding to these addresses. As no two pages can occupy the same page a preference scheme also applies to the two Eprom pages. Eprom (1) (the left page and enabling switch) will be disabled automatically if Eprom (2) (the right page and enabling switch) is placed on the same page as Eprom (1). (See fig 1.3). Make sure that the pages of Eprom contain valid data, especially the last 16 Eprom words, before switching on-line any Eprom page. Use CLBL or switch the page to address 3 and use RAMWR.
If you take a closer look into the Eprom compartment you will also see the memory retention battery. This battery ensures the retention of all data placed in the Ram pages during approx. one year after installation. Replace or have this battery replaced before Eprom failure occurs. To be on the safe side replace the battery once every 9 to 12 months.

THE MOMENT OF TRUTH

Close your box. Make sure your calculator is turned OFF and connect the box to the system. After switching your calculator ON you have your Eprom sets and the Eprom pages at your fingertips. Should this not be the case, check the enabling switches, check your user ports for forgotten modules and check the Eprom address sing and enabling switches. See set up instructions for address sing conflicts.See also In Case Of Problems.
If your selection of used ports was carefull enough you will hardly ever have to switch any Eprom / Eprom off. If you have to, refer to the instructions for use to disable the conflicting Eprom / Eprom. Do not remove or install Eprom's too often as this will wear out your Eprom sockets. Use the Eprom pages for less

used software. An IL-cassette and the routines RROM / WROM provide you with this facility.

IN CASE OF PROBLEMS

As can be seen in tabel 1.3 only Eprom's 5 & 6 can cause direct problems. These problems occur if a plug-in module is placed in ports 1 or 3 and Eprom's 5 or 6 are inserted. Non problem causing plug-in's are Printer, HP-IL module, Time module, Single and Quad Memory and Extended Memory module.
Indirect problems can occur with 8k. plug-in modules. These are (apart from 8k. custom Rom's):
Navigation Rom, Real Estate Rom, Structural Analysis, Data Logger Rom, HP-IL Development Rom, Plotter Rom, Petroleum Rom, PPC Rom and a special case is the Autostart/Duplication Rom. All these Rom's have a size of 8k., except for the Auto/Dup Rom, and there fore occupy the lower as well as the upper page of a port. A port thus used can not be used also in the Eprom Box. Switch off the Eprom's corresponding to that port or remove the 8k. module, which ever seems needed. In the special case of the Auto/Dup Rom we have a 4k. Rom that uses an upper page in a port. Place this Rom in one of the ports 1 or 3 to get maximum addressing capability. Do not forget to switch off switches 2 or 1, depending on the port used.

Another problem often encountered, is the use of the same label by more than one Rom. Do a **CAT 2** to find duplicate labels in Rom. If there are any, please insert the wanted Eprom or Rom at the **lowest** possible address or disable / unplug the offending one. Should you want to use both at the same time it would be advisable to change the label name in the Eprom to a unique one. (See burning of Eprom's). A problem less often encountered but harder to detect is the use of the same X-Rom number by two Eprom's / Rom's. Check the Rom / Eprom user manual for the used X-Rom numbers. Your HP-41 C(V) will always use the **first** Rom / Eprom with the desired X-Rom number, disregarding all the others. E.G. Aviation Rom and Clinical Rom both have X-Rom 19 as identifying number. These modules, using the same X-Rom number are identified by the addition 'X' to the module name on the module case.
The Ram and Eprom pages can cause software crashes if not properly initialized. Make sure you have cleared or initialized the Ram page before switching it on-line. Should the system crash, switch the Ram or Eprom off. Try to wake up your calculator now. Unplug the box if necessary. Reinitialize the page or remove the defective routines in Ram or Eprom.

If all the possible causes have been checked, and you are still left in doubt as to the proper functioning of your system unplug the Eprom-Box and check your HP-41 C(V) for problems. If the Eprom-Box is suspected of causing the trouble send it in for service.

THE BURNING OF EPROM'S

In order to use your own personalized software at any moment without having to use your card-reader, wand or IL-mass storage device, you can order an Eprom set containing the routines you have written. This burning service is offered amongst others by **Eramco Systems.** Should you have problems, concerning bug's or duplicate labels / X-Rom numbers an Eprom revision service is also available at reduced rates. To use this service send your software and sofware verification data on magnetic medium (cards or cassette) or in barcode to **Eramco Systems** or one of the others. M-Code routines and User code routines can be mixed at will. If a complete Eprom set has to be burned, and is delivered in the standard ROM>REG, REG>ROM format, reduced burning costs will be charged in certain cases.

L1IMITED 180 DAYS WARRANTY

ERAMCO Systems warrants their MLDL - Boxes against defects in materials and workmanship for 180 days from the date of original purchase. If the unit is sold or transferred to other persons or institutions ownership warranty is automatically transferred to the new legal owner. During the warranty period ERAMCO Systems will repair or, at their option, replace at no charge a product that proves to be defective, provided the product is returned, shipping prepaid, to ERAMCO Systems or their official service representative.

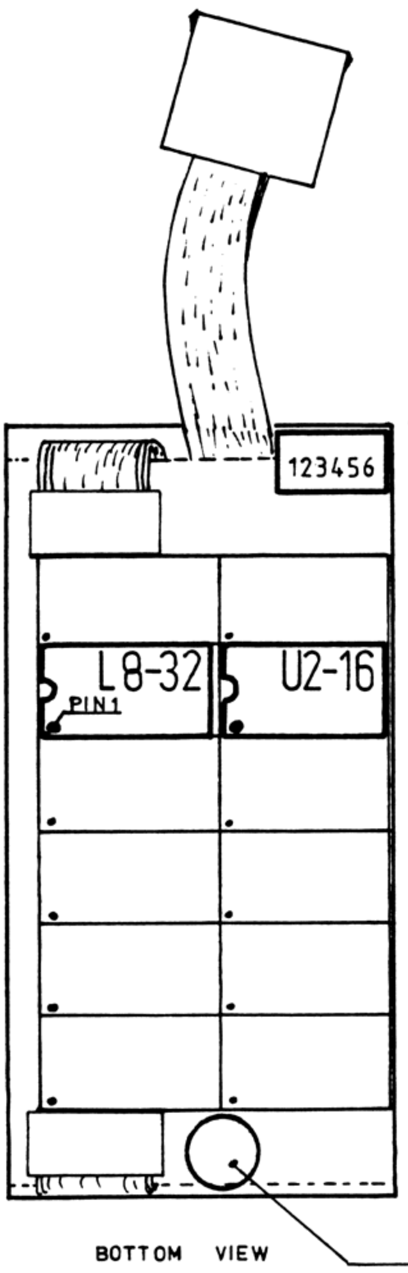
WHAT IS NOT COVERED

This warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other than ERAMCO Systems or their official service representative.

No other express warranty is given. Any other implied warranty of merchantability or fitness is limited to the 180 days period of this written warranty. In no event shall ERAMCO Systems be liable for consequential damages. This liability shall in no way exceed the catalog price of the product at the moment of sale.

ERAMCO - SYSTEMS
KROMBOOMSSLOOT 16
1011 GW AMSTERDAM
020 - 269838
THE NETHERLANDS

ERAMCO SYSTEMS



EPROM 1	ADDR. D	PORT 3-U
EPROM 2	ADDR. 9	PORT 1-U
EPROM 3	ADDR. F	PORT 4-U
EPROM 4	ADDR. B	PORT 2-U
EPROM 5	ADDR. 8	PORT 1-L
EPROM 6	ADDR. C	PORT 3-L

Fig 1.1

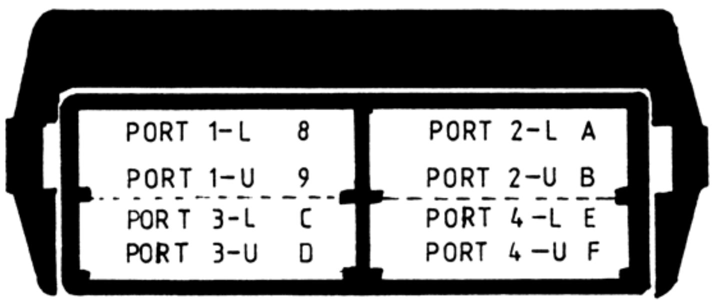


Fig 1.2

ERAMCO SYSTEMS

ON INITIAL MLDL SETUP:
The RAM Page addressing can be set to any page in the HP41C,CV,CX memory block. Except 0,1,2 which are the HP41's controlling ROMs.
But the READ ENABLE switches should be OFF. That is in the DOWN position.
Having plugged-in the lead into the 41 & switched ON, first off run the following:
1. In ALPHA put the Page letter for the 4Kb RAM block you wish to use. Switch ALPHA off.
2. XEQ 'CLBL' (Clear-Block), this will clear the chosen page from any crash codes or even any codes at all.
3. Now the RAM page is initialised & the ENABLE switch can be turned on (UP). Access is now possible for any programming you want.

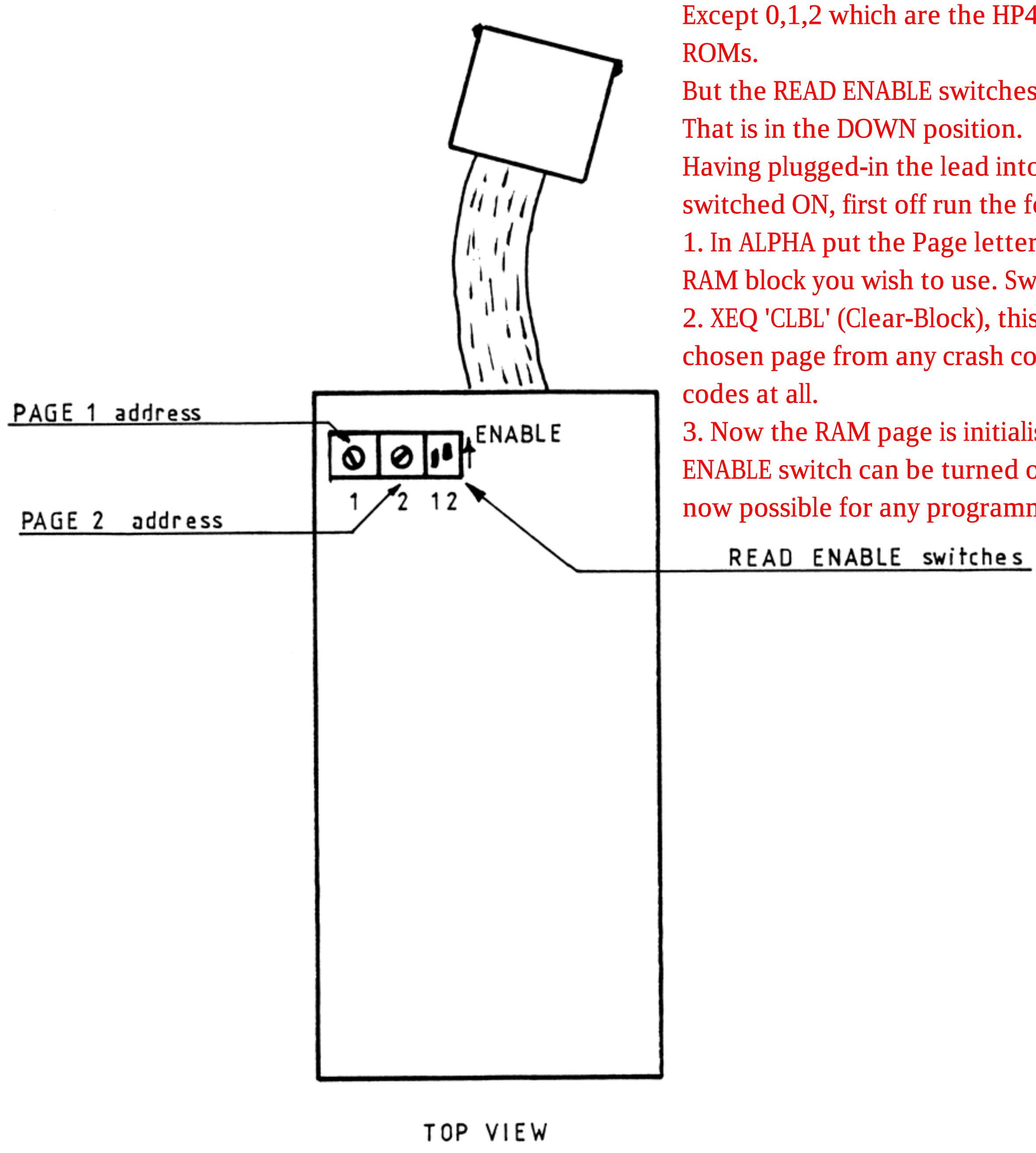


Fig 1.3

BATTERY NOTE: Memory retention is guaranteed through the use of a lithium battery that retains the memory contents, depending on how often the box is left connected to the HP-41. No current will be drawn from this battery as long as the box is connected to the HP-41. So the longer you leave the box connected to your system. The longer your battery will last. Here also lays one weak point of the ESMLDL1: to replace the battery you will need to unscrew the base, find the clip, & plug in a new 3 volt one. The lithium battery is a standard part & should be available in most countries of the world. You might even consider to replace this battery with two used HP-41 batteries (3 volts together). It is a nice way to get most out of the hardware you have to buy for your system.

ERAMCO
SYSTEMS

ES 83120A
ERAMCO MLDL-OPERATING SYSTEM EPROM

OWNER'S MANUAL

Februari 1984

83120A-M000002

CONTENTS

Introduction......................................3
Installation......................................3
Organisation of the instruction set...................3
Mldl write functions..............................4
Utility functions.................................10
User code programs................................16

Appendix A: Input / Output.........................18

Appendix B: Programmability........................19

Appendix C: Error Messages.........................20

Appendix D: XROM numbers...........................22

Appendix E: XROM and FAT structure.................23

Appendix F: Interrupting Points....................24

Function Index....................................25

Care and Warranty.................................26

User code programs................................27

INTRODUCTION

This manual deals with the **ERAMCO MLDL** operating system eprom. To get a full understanding of all the routines and functions in this eprom set, it is advisable to read through this manual carefully before operating any of the functions or routines.

INSTALLATION

Follow the instructions of your **ERAMCO MLDL-box** carefully when installing the eprom set in your box. It may be necessary to bend the feet of the two eproms slightly inward to make them fit easily into the epromsockets. Do not forget to enable the page where on which you insert the eproms (for more detailed information on how to insert the eproms, consult your hardware manual of the **ERAMCO MLDL-box**). A lower address is the most appropiate page for insertion of the eprom. This provides a quick access to the routines and functions available in the ERAMCO MLDL-eprom set.

ORGANISATION OF THE INSTRUCTION SET

As you will soon discover out the routines and functions in this eprom set are divided into three sections. The first section contains all the functions and routines that will change anything in the MLDL-RAM you are working on. So always be carefull when you use any of these functions. A single mistake can destroy the whole 4K RAM block that is under development.
The second section contains the functions that facilitates working with the MLDL-RAM. They do not change anything in the RAM but
will provide a quicker access to the RAM (LROM will tell you almost immediately where you can continue with writing in the RAM or where you can store a User-code program).
The third as well the last section contains some User-code programs that will make the manipulation of storing in and extraction of data from RAM a lot easier, especially when it concerns a lot of data (storing the whole block of 4K on a tape or read it from a tape).

Note : All inputs which has to be placed in the alpha-register are related to hexadecimal

RAMWR (RAM WRite)
XROM 11,01
This non programmable function allows the user to read every word in a ROM, EPROM, or MLDL-RAM (EROM). In case of MLDL-RAM it is also possible to change or write in this MLDL-RAM. The addresses and data are prompted for and given in hexadecimal form. This function will redefine the keyboard as long as it is used to make hexadecimal input easier.

After calling this function it will prompt for the absolute address in ROM. Now the following keys are active: 0-9, A-F, back-arrow and the on key. The back-arrow key is used in the usual way to correct the last given input. NULL will be displayed if you hold
the last input-key. When you release this key you will be prompted
again for the address. Pressing back-arrow without input causes the function to exit to normal operation of the HP-41.

In the display are shown the address and the data at this address (AAAA DDD). From now on the keyboard is defined as follows:

- -STO will give you the data at this and the following addresses. Each address and the data are displayed for about 0.5 sec. Pressing any key accept the R/S or the ON key, will slow down the listing of the data that is displayed. The R/S key will stop the listing at any desired place. The ON key will switch off the machine in the usual way.
- -TAN or BST decrease the address by one. This enables you to go through the listing by hand.
- -SST increases the address by one, making it possible to step through the listing by hand.
- -back-arrow asks you for a new address if there is no data input. Otherwise it will operate in the usual way to correct the last input.
- -"0", "1", "2", "3" (numberkey's 0,1,2,3) are interpreted as new data. In this way wrong data input is prevented, because the first character of a data word can only be 0,1,2 or 3. For the rest of the data input the hexadecimal keyboard is available again. Holding the last data key will NULL the input function and after releasing the key will prompt for new data. With the back-arrow key it is possible to correct the last given input. The address will be increased by one after completion of data input. This will facilitate the writing of long programs.

You can exit this function when you are in input mode, by means of pressing the back-arrow key twice.
WARNING: Be careful with the addresses from xFF4 up to xFFA. These addresses are scanned by the operating system of your calculator. It's possible that the calculator will crash when these adresses contain error data. For more information see appendix F.

MMTORAM (Main Memory TO RAM)
XROM 11,02
The function MMTORAM is used to copy a program from main memory in the calculator to the desired MLDL-RAM page in a MLDL-box. All the necessary translations for a good operation of the program are made automatically. The Function Access Table (FAT) is updated at the same time with the new Global Labels of the program. For good operation of this function it is necessary to initialise the MLDL-RAM and the program in the proper way.
Preparation of the program: Make sure that there is an END after the program (execute GOTO..) and compile the program with the help of COMPILE.
Preparation of the MLDL-RAM: You need a block of RAM words that is long enough to hold the desired program. The length of the program can be found with the help of CBT (see CBT). Add two to this number of bytes and you have the number of bytes that will be needed for the program when loaded into the MLDL-RAM. Now you must find a block in the RAM space that is large enough. Write down the starting address of this block. **BE CAREFUL** Addresses in RAM are given in hexadecimal form, but the length of the program (by CBT) is given in decimal form. Key in ALPHA the starting address of the block (it's handy to leave about 20 words between the starting address of the block where the program will be written and the first empty word in the RAM you have found for revisions) and execute COD The address is transferred into the X-register in a coded form. This form is not usable for any of the normal functions of the calculator. Be careful here. Any change of the value in the X-register can give a different and not wanted result.
User flags 0 and 1 can be set or cleared to achieve the desired private status

flag 0	flag 1	status
cleared	cleared	program open
cleared	set	program open, after COPY private
set	cleared	program private
set	set	program private

With the help of these two user flags it is possible to make the program completely private in the MLDL-RAM, e.g. you can not go into PRGM mode to examine the program and it is not possible to copy the program into the main memory with the help of the COPY function. A partly private status is also possible. In this state it is possible to examine the program, but after copying it into the main memory it will be private. The third option means no security at all. Programs are now free to examine and to copy (compare with e.g. the math module).Please note that changes in the program are only possible when it is stored in main memory (see the manual of the calculator for it's behavior when you are in rom).

MMTORAM can be executed after these preperations. The function will prompt for the name of the program that has to be copied. It is enough to press ALPHA twice when the program counter is already set in the wanted program. Otherwise you must enter the name of the program in the same way as with CLP or COPY. The display will come back with the X-register after some time (dependant on the length of the program). A CAT 2 or a CAT x (x is the pagenumber of the MLDL-RAM where the program has been written on) will show you the updated FAT with the new labels. Noting down the start and end-address of the used block will allow you to make changes without address mistakes.

AFAT (Append FAT entry)
XROM 11,03
The function AFAT enables the user to update the FAT, e.g. to append the starting address of a routine that has been written in the MLDL-RAM. Functions are only accessable to the HP-41 when they have an entry in the FAT. This also holds for programs that are transferred to the MLDL-RAM. The function MMTORAM takes account of this automaticaly.
Input for AFAT is in the format UOPAAA. AAA is the start-address of the function within a page, P is the page number where it is loaded, O is an offset and U tells the HP-41 if the routine is a M-code routine or a User code program.

U=0 M-code function. The address points to the first word that is executable
U=2 User code routine. The address points to a Global Label

Example: AAA=3FF The start of the function or routine is found at X3FF

In order to understand the interaction of O and P it is necessary to realise that EPROM and MLDL-RAM can be placed at every wanted page, e.g. at any desired port. It must also be kept in mind that an EPROM or MLDL-RAM page contains only 4K. The value of P is only pointing to the page where the MLDL-RAM is positioned at at this moment. The value of P will also change when you address the MLDL-RAM to a different page. Opposite to this is the behaviör of the value from O. O is a constant, added to the pagenumber. It will not change when you place the MLDL-RAM at a different page. The constant O allows you the opportunity to execute functions and routines from another page than the one where the FAT entry is lodged. So it is evident that the page which is called must always be O pages further in the memory.

Example: The page that contain the FAT is at page 8, and the page that contain the routine itself is at page C, address is 490. We want to make an entry for a User-code routine with AFAT.

The value of O (the offset) is C - 8 = 4
The value of P (page containing the fat) is 8.
The value of AAA (start-address) is 490.
The value of U (M- or User code) is 2.
We do now need the following input for AFAT

248490

When we move the first ROM to another address we must also move the second ROM the same number of pages in the same direction if the value of O is something else then zero. Leading zero's in the input can be omitted

DFAT (Delete FAT entry)
XROM 11,04
The function DFAT is used when you want to delete an entry from the FAT. This function or routine will be invisible for the HP-41 after execution of DFAT. The XROM numbers of all the routines and functions that came after the deleted one in the FAT, will get one lower. Pay attention to this when you use functions or routines from the RAM you are working on. The same input format is used as with AFAT. The Difference is that you do not need to specify the value of U.
So the input format will be OPAAA (offset),(page),(address)
DFAT will search in the page with number P and delete the specified entry. Leading zeros may be omitted.

MOVE (MOVE ram block)
XROM 11,05
The function MOVE allows the user to move certain parts in a ROM, EPROM or MLDL-RAM to another place. Keep in mind that you can only move into MLDL-RAM. MOVE makes it possible to insert words or delete words at any place in the MLDL-RAM. It is also handy to copy only small routines or functions from another page to the MLDL-RAM page you are working on.
The input format in ALPHA is as follows : BBBBEEEEDDDD
BBBB gives the starting address of the block that has to be moved (it is the first word that will be moved).
EEEE gives the end-address of the block that has to be moved (it is the last word that will be moved).
DDDD gives the address of the first word of the block where the source block will be copied on to.
The function will accept a destination address within the original block.

This is the most necessary program to learn about & run FIRSTLY, as stated in the INTRODUCTION page, it basically clears junk values from a RAM 4Kb page, initialising it, before any personal programming is done. or even before the 4Kb page is switched on !!! Using the 'P' value in ALPHA is the quickest way to initialise/clear a RAM page before enabling it.

CLBL (Clear ram Block)
XROM 11,06
Clearing a block of MLDL-RAM is done with the help of CLBL. Input is in ALPHA in the format BBBBEEEE.
BBBB is the first word of the block that has to be cleared.
EEEE is the last word of the block that must be cleared.
Execution of CLBL puts zero in all the addresses between the given ones, including the start and end addresses. Another option of CLBL is to clear a whole 4K block at once. For this the input P in ALPHA. P represents the pagenumber of the page you want to clear. **** ATTENTION **** This last option is dangerous. It operates in the same manner as a MEMORY LOST, but in this case it is a memory loss of the specified MLDL-RAM page.
Valid 'P' values in ALPHA = a single letter for the RAM page you wish to use: 8,9,A,B,C,D,E,F !!!

COPYR (COPY Rom page)
XROM 11,07
The function COPYR enables the user to copy an entire page of ROM or MLDL-RAM on to another page of MLDL-RAM. This gives you the opportunity to change anything you want in the just copied block of ROM.
Input is in ALPHA and has the format SD.
S is the page from where the copy has to be made (Source).
D is the page to which the copy is destined (Destination).

ROMSUM
XROM 11,08
To check if a ROM is still in good order HEWLET-PACKARD has put a checksum in each ROM. With the function ROMSUM you are able to compute this checksum and put it at the proper place in the MLDL-RAM you are developing. The checksum is calculated by adding all the words on this page, take modulo 255 and put the remainder in xFFF.
The input is P in ALPHA. P is the page number of the MLDL-RAM you want to update the checksum.

REG>ROM (REGisters to ROM)
XROM 11,09
This function is the opposite of ROM>REG (for more information on this function see at ROM>REG). This routine will translate the registers with it's 5 words/register back into 5 different words and place them at the proper addresses in a MLDL-RAM page.
The input in the Y-register determines where the data will be put back in the MLDL-RAM. 3 different options are available to achieve this.

1. "Y"= 0	The block will be placed back at the same location as where the original stood (if the original was located from 83FF to 8456 it will be restored to the same address.

2. "Y"= P	P represents a page number that is made with the help of COD. The block will now be loaded at the same relative addresses from which it came from comes from but at a different page(if the original was located at 83FF to 8456 it will be restored at P3FF to P456).
3. Y = BBBB	Here BBBB represents the start-address where the block will be stored at (BBBB >= 0010). The block will be loaded starting at the address given by BBBB independent from the original start-address of the block.

The X-register must hold the number of the register that contains the first data words of the block that has to be read back (actually the first register contains a header that is used by REG>ROM and is made by ROM>REG).
Writing entire 4K blocks of MLDL-RAM from a storage medium is facilitated by the User-code programs 'WROM and 'RROM.

XROM 11,10
This is not a normal function. It does not do anything, when executed but it is used as a spacer from write routines and application routines within the MLDL-RAM . One possible application is to use it as a NOP. It will also terminate data input without raising the stack.

COMPILE
XROM 11,11
The function COMPILE places in every GTO and XEQ to a numerical label the distance to that label. Programs prepared with the help of COMPILE will usually run faster than programs that have to calculate these distance's while running. Two byte GOTO's that can not make the distance will be transformed to three byte GOTO's. This makes it possible that your program will be made longer by this routine and it is required to have at least three registers left after the program (.END. REG xxx with xxx not equal to zero).
Compile prompts for the name of the program you want to compile. Input is in the same way as with the main-frame function CLP. So if you are not in the program you want to compile, you must input the complete name. Otherwise it is possible to press ALPHA twice. The function will first pack the program (PACKING), then handle the two byte GOTO's (COMPL 2B G) and if needed (in this case compile has found a 2 byte GTO that can not make it and replaced it with a three byte GTO. This causes insertion of null bytes that have to be packed) repeat this sequence. After this is done it will continue with the three byte's GOTO's and XEQ's (COMPL 3B G/X). After the routine is finished it will put the message READY in the display. Labels not found will give the error condition NO LBL xx, with the number xx as the label not found. When you switch to program mode you will find the program step that caused the error condition.
Deleting steps or adding steps in a program, will change the status of the program into a decompiled one. Reusing the compiler will speed up the execution after the editing session.

LOCA (LOCAte word)
XROM 11,12
This function allows you to locate a given data-word in a 4K block of ROM, EPROM or MLDL-RAM.
The input format in ALPHA is as follows: BBBBDDD.
BBBB specifies the address from where LOCA starts searching in the 4K block. Actually it will start at BBBB + 1 to allow repeated search in the block. Flag 10 will be set when the wanted data (DDD) is not found in this 4K block. Whenever a data-word is found, it will be displayed with the address where it is located No change occurs in the status of flag 10. The data in ALPHA (adress + word) will be replaced with the data found.This makes it possible to continue on searching for the same word.

LROM (Last ROM word)
XROM 11,13
LROM searches <u>backwards</u> for the last non zero word in a block beginning at a given start-address. Input is AAAA in ALPHA. The display will give the address of the last non zero word and the value at this address. NONE will be returned when the block between the start address and the <u>beginning</u> of this 4K page doesn't contain any non zero word.
This function can be very handy when the end-address of the last program entered has to be found In this case the easiest way to put xFF4 into ALPHA and execute LROM. It will give you the address of the last word that is occupied by the program.

COD (CODe)
XROM 11,14
The hexadecimal number in the ALPHA-register is converted to it's -bit-representation and will be placed in the X-register. The contents of the ALPHA-register is unchanged. The stack will be rolled up and the value in the X-register before COD was executed is placed in the LASTX-register.

DECOD (**DECODe**)
XROM 11,15
The function DECOD is the opposite of the function COD. It will translate a -bit-representation in the X-register to the same hexadecimal form as is used with the function COD. The output is given in the ALPHA-register. When DECOD is executed manualy on the keyboard it will also give the hexedecimal representation in the display.

ROMCHKX (**ROMCHeck by X-reg.**)
XROM 11,16
This function enables you to check if a ROM or MLDL-RAM is still in good shape. Important though is the fact that a ROM or MLDL-RAM must contain a good computed checksum (see ROMSUM for the definition of the checksum). HP rom's will always contain a good checksum. During the test the XROM number is displayed along with the short form of the name and the revision number of the ROM. If the ROM or the MLDL-RAM doesn't contain this short name or the revision number, the display will show @@-@@ instead of the normal display of NN-RR.
Input in the X-register, the XROM number of the ROM or MLDL-RAM you want to test (an example is 30 for the cardreader). During the test XX NN-RR TST will be displayed. XX is the XROM number of the ROM that is tested, NN is the shortened name and RR is the revision number.

Output of ROMCHKX is the display XX NN-RR BAD (indicates a bad ROM) or the display XX NN-RR OK (indicates a good ROM) These outputs will be given only when the function is executed from the keyboard.
The behavior of ROMCHKX will be different when it is executed in a program; when a ROM is found to be good it will do the next step in the program. Else it will skip the next step (compare the function FS?. the rule do if true is in force).
When there is not a ROM present with the desired XROM number the message NO ROM XX will be displayed. Again it's behavior in PRGM mode is different. It will act as if the ROM is bad and skip the next line.

ROM>REG (ROM to REGisters)
XROM 11,17
All the credits for this function and its opposite go to to Paul Lind and Lynn Wilkins who have written these two routines.
ROM>REG places 5 words of 10 bits each in one HP-41 register. To avoid damage to the stored data it is saved as alpha data. This guarantees an optimal use of the available registers in the main memory of the calculator. Also it is now possible to store the routines and functions that are written in a MLDL-RAM on tape or cards and they make it easier to exchange M-code with other users.

The input for this function must be given in the Y-register. It has the form BBBBEEEE.
BBBB is the address of the first word to store.
EEEE is the address of the last word to store.
This input has to be in binary and right justified. This is achieved by putting the BBBBEEEE form in ALPHA and execute COD after this. The binairy representation can be transferred to the Y-register by means of keying in a number in the X-register. The X-register holds the number of the first data register that will be used as data store.
If the number of registers needed exceeds the number of free registers, you will get the error message NONEXISTENT.
There is also an output of this function. In the LASTX-register the last used register is given. By subtracting X from LASTX you will get the number of used registers and consequently the number of registers needed to store the desired MLDL-RAM block.

MNEM (MNEMonics)
XROM 11,18
This function will give in conjunction with DISASM the name of a M-code instruction that is fetched with DISASM. The mnemonics that are used are the so called HP-mnemonics (there are also PPC (Jacobs) mnemonics). The mnemonics are left as a string in the Z-register. Eventual surplus information (jump-distance, value, field specifications) is given in the T-register. In case of two word instructions the LASTX-register is used. The following User-code program makes it possible to translate every ROM that you want. It is available in this EPROM set.

```
01 LBL 'mdis        Name of program
02 CLST             initialize the stack registers
03 STO L            initialize the LAST X-register
04 SF 21            makes program stop at aview
05 'start add?      ask for start-address
06 AON              make ready for input
07 PROMPT           ask and wait for input
08 AOFF             leave the ALPHA mode
09 COD              put the start-address in X
10 LBL 01           start of the loop
11 DISASM           get the instruction
12 AVIEW            view the address, value and the character
13 MNEM             build the mnemonic in the stack
14 CLA              initialize the ALPHA-register
15 ARCL Z           get the first part of the mnemonic
16 '@               append a space
17 ARCL T           get the second part of the mnemonic
18 AVIEW            view the mnemonic
19 GTO 01           restart the loop
```

This routine is meant to be used in manual mode. For use with the printer it must be rewritten. The choice up to the user.

DISASM (DISASeMbler)
XROM 11,19
The function DISASM makes it possible to put the contents of ROM into the display. At the same time the character representation from the word is also given in the display.
Input: The X-register must contain the address of the wanted word (this can be done with the help of COD).
Output: The X-register will be incremented by one to make it easy to use DISASM in a loop. The Y-register holds the binairy value of the address and the data at this address (these values can be made visible with DECOD). The ALPHA-register contains AAAA WWW L

AAAA	is the address of the wanted word.
WWW	is the value of this word.
L	is the character representation of the word.

There are two ways to represent characters in the HP-41. One way is the use of the ASCII standard. The other way is derived from this standard by subtracting 40 [hex] from the codes in the range from 40 hex through 5F [hex]. This gives you codes that lay in the range from 0 hex to 1F [hex].

CAT (CATalog)
XROM 11,20
The function CAT gives you a selective CAT 2. This routine comes especially handy when you have to examine the catalog of a ROM that is located in a higher numbered port. When the system is loaded with a lot of roms it will take a long time before you arrive at the desired ROM (maybe you must go through the TIMER, PRINTER, IL-MODULE before you reach the wanted ROM). The function prompts in the same manner as does the CAT function of the HP-41. The prompt can be answered with the hex digits 0-F (CAT will redefine the keyboard in the same manner as with RAMWR). Digits 0-3 have the same result as the normal CAT function from the HP-41. Digits 5-F will start the catalog at the wanted page. For further details we refer to the manual of the HP-41.

CBT (Count BYtes)
XROM 11,21
This function counts the number of bytes that is occupied by a program. At the prompt the name of the desired program must be keyed in, or if you are already in the desired program press ALPHA twice (compare with the function CLP).
Output is given in the display only. The stack and the ALPHA-register are left undisturbed.

SYNT (SYNTesize)
XROM 11,22
With this function you can create two- and some three bytes instructions in the program memory without the bytegrabber. Data for this function is given in the X- and Y-register. The first byte of the instruction is decimal coded given in the X-register. The second byte is given in the Y-register. SYNT will place the instruction after the program line where the program counter is pointing at that moment. ATTENTION : this routine works both in PRGM and RUN mode. Therefore you must be very carefull when assigning SYNT to a key. Careless pressing the assigned key will produce an unwanted line in your program or even worse.
Example 159 ENTER^ 58 execute SYNT will give a TONE 8 in your program which is completely different from the normal TONE 8. An input of 247 in X and Y will give you a byte grabber.

GE (Go to .End)
XROM 11,23
This function is a sort of replacement of the GTO.. function of the HP-41. It will put you at the end of the program memory, but it is not packing the memory. Furthermore it does not put an end to the last program in memory. When you do not know anymore where you are in main memory use GE and you are at a familiar place again.
Switching to PGRM mode will display .END. REG NNN
The latest revision of this routine will display 00 REG NNN, It also circumvents the line number bug in the HP-41 operating system.

XROM 11,24
This is not a function. It is only used as a spacer between the utility functions and the User code programs. For further remarks see the same function at page 10 (--- XROM 11,13).

'WROM (Write ROM)
XROM 11,25
The User-code program WROM saves the contents of an entire 4K RAM or ROM page on tape in a data file with a size of 824 registers. It uses the X-Function module for testing the number of registers that are momentarily allocated and if necessary change the size to the needed one. The functions SIZE? and PSIZE are used. The program stops with the error message PACKING , TRY AGAIN if the number of free registers is not large enough. The user should set the size manually if he doesn't have an X-Functions module. The size must be 206 or greater.
The program prompts for the name of the file in which you want the data to be stored, and also prompts for the page which must be saved. This pagenumber must be given in alpha in hexadecimal.

'RROM (Read ROM)
XROM 11,26
The User-code program RROM reads the contents for an entire 4K RAM or ROM page from tape from a datafile with a size of 824 or 823 registers. It uses the X-Function module for testing the number of registers that are allocated and if necessary change this number of registers to the needed one. The X-functions SIZE? and PSIZE are used. The program stops with the error message PACKING , TRY AGAIN if the number of free registers is not sufficient. The user should set the size manually if he doesn't have an X-Functions module. The size must be 276 or higher.
The program prompts for the name of the file you want the data to be stored in and also prompts you for the page to read into. This pagenumber must be given in hexadecimal.
'RROM can read both 823 and 824 size files. This is done because both types are frequently used at this moment.

'XRSYN (XRom SYNtesize)
XROM 11,27
This function together with SYNT enables you to create every wanted XROM AA,BB in a program. You do not have to have a certain rom module anymore, because you can create the calls to its functions and programs without the module.
Input: In Y is the AA part of the desired XROM AA,BB
       In X is the BB part of the desired XROM AA,BB
Output: You will get back two decimal numbers in X and in Y that enables you to create the XROM AA,BB in your program by executing SYNT.

One possible use of 'XRSYN is to create the XROM's in a program that is to be transferred from main memory to the MLDL-RAM with the function MMTORAM. Not only will it shorten the program, but also speed it up a lot, because the calculator does not have to search through the whole program area or the extended functions catalog (CAT 2).

APPENDIX A

XROM	NAME	INPUT	OUTPUT
11,01	RAMWR	0-F hex	word in RAM
11,02	MMTORAM	X beginaddress in RAM flags 0 and 1	stored program
11,03	AFAT	UOPAAA in ALPHA	FAT updated
11,04	DFAT	OPAAA in ALPHA	FAT updated
11,05	MOVE	BBBBEEEEDDDD in ALPHA	block is moved
11,06	CLBL	P / BBBBEEEE in ALPHA	block cleared
11,07	COPYR	SD in ALPHA	copied block
11,08	ROMSUM	P in ALPHA	romsum in xFFF
11,09	REG>ROM	0/P/BBBB in ALPHA first reg in X	data in RAM
11,10	---		
11,11	COMPILE	name of program	compiled program
11,12	LOCA	BBBBDDD in ALPHA	AAAADDD flag 10
11,13	LROM	BBBB in ALPHA	AAAADDD
11,14	COD	hex in ALPHA	binary in X
11,15	DECOD	binary in X	hex in ALPHA
11,16	ROMCHKX	XROM in X	bad / ok do if true
11,17	ROM>REG	BBBBEEEE in ALPHA first reg in X	data in registers last reg in LASTX
11,18	MNEM	AAAADDD in Y	mnemonic in Z and T
11,19	DISASM	BBBB in X	BBBB + 1 in X AAAADDD in Y
11,20	CAT	P at prompt	cat from page P
11,21	CBT	length of program	name at prompt
11,22	SYNT	X first dec. byte Y second dec. byte	instruction after pc.
11,23	GE	pc. at .END.	
11,24	---		
11,25	'WROM	name and page	4K in file on tape
11,26	'RROM	name and page	4K of tape in RAM
11,27	'XRSYN	Y xrom number X function number	coded xrom for SYNT

SHORT FORM LETTER	REPRESENTING
A	address digit
B	begin address digit
D	data digit or destination digit
E	end-address digit
O	offset digit
P	page number digit
S	source digit
U	user digit

APPENDIX B

PROGRAMMING AND THE MLDL EPROM SET

Most functions provided by the ERAMCO MLDL-EPROM can be entered in program whenever the eprom-set is plugged in an ERAMCO MLDL-box that is connected to the calculator. While the ERAMCO MLDL-box containing the eprom set is connected, program lines with eprom functions are displayed and printed as standard functions.

If the box is disconnected later, these program lines are displayed and printed as XROM functions with two identification numbers. The first number, 11, indicates that the functions are provided in the ERAMCO MLDL-EPROM. The second number identifies the particular function. The XROM numbers for the ERAMCO MLDL-EPROM are listed below.

Function	XROM Number	Function	XROM Number	Function	XROM Number
AFAT	XROM 11,03	DISASM	XROM 11,19	ROMCHKX	XROM 11,16
CAT	XROM 11,20	GE	XROM 11,23	ROMSUM	XROM 11,08
CBT	XROM 11,21	LOCA	XROM 11,12	ROM>REG	XROM 11,17
CLBL	XROM 11,06	LROM	XROM 11,13	'RROM	XROM 11,26
COD	XROM 11,14	MNEM	XROM 11,18	SYNT	XROM 11,22
COMPILE	XROM 11,11	MMTORAM	XROM 11,02	'WROM	XROM 11,25
COPYR	XROM 11,07	MOVE	XROM 11,05	'XRSYN	XROM 11,27
DECOD	XROM 11,15	RAMWR	XROM 11,01	---	XROM 11,10
DFAT	XROM 11,04	REG>ROM	XROM 11,09	---	XROM 11,24

Underlined functions are not programmable.

If program lines using the ERAMCO MLDL eprom are entered when the eprom set is not connected, the function is recorded and displayed as XEQ followed by the function name. Program execution will be slowed down by lines in this form because the calculator will first search for a program or program line with the specified label.

APPENDIX C

ERROR MESSAGES

This is a list of messages and errors relating to the functions provided by the ERAMCO MLDL-EPROM set. When any of these errors are generated, the attempted function is not performed, except as noted.

DISPLAY	FUNCTION	MEANING
NO LBL xx	COMPILE	The GTO or XEQ has no corresponding LBL in this program.
NONE	LROM	The whole block is empty.
	LOCA	There is no such word in the block from start-address up to the end of the page.
ROM	MMTORAM COMPILE CBT	The named program doesn't exist in main memory but is found in ROM
NONEXISTENT	-all-	The ERAMCO MLDL-EPROM set is not plugged in or is disabled or malfunctioning.
	ROM>REG	There are not enough registers available to store the specified block.
NO ROM xx	ROMCHKX	The ROM with the given XROM number is not plugged in or enabled.
xx NN-RR BAD	ROMCHKX	The ROM with the XROM number xx is bad.
xx NN-RR OK	ROMCHKX	The ROM with the XROM number xx is ok.
PACKING TRY AGAIN	'RROM 'WROM	There is not enough space in main memory to create enough registers.
COMPL 2B G	COMPILE	The 2 byte GTO's are handled.
COMPL 3B G/X	COMPILE	The 3 byte GTO's and XEQ's are handled.
READY	COMPILE	The compiler is ready.

NO ROM	RAMWR	An attempt has been made to write on a page which doesn't have its XROM number set at the first address of this page.
NO WRITE	RAMWR	The data is not written at the desired address. It is impossible to write on a EPROM or ROM page. Also you can not write at a disabled page.
NO ENTRY	DFAT	This is not an entry in the FAT.
ENTRY>64	AFAT	There are already 64 entry's in the FAT.

APPENDIX D

XROM numbers range from 1 up to 31 inclusive. As quite a few ROM's are available at the moment of this writing it is advisable to choose an XROM number with care to avoid conflicts with other modules.

ROM name	XROM ID	ROM name	XROM ID
MATH	01	SECUR	19 *
STAT	02	CLINLAB	19 *
SURVEY	03	AVIATION	19 *
FINANCE	04	MONITOR	19 * +
STANDARD	05	STRUCT-B	19 *
CIR ANAL	06	C PPC 1981	20
STRUCT-A	07	ASSEMBLER 3	21
STRESS	08	IL-DEVEL	22
HOME MN	09	I/O	23
GAMES	10 *	IL-DEVEL	24
C PPC 1981	10 *	-EXTFCN	25
AUTODUP	10 *	-TIME-	26
REAL EST	11	- WAND	27
MACHINE	12	-MASS ST	28
THRML	13	(- CTL FNS -	
NAVIG	14	HP-IL MODULE)	
PETROL	15	-PRINTER	29
PETROL	16	CARD READER	30
PLOTTER	17	PPC ROM 2 ??	31
PLOTTER	18	ERAMCO-MLDL	11

+ Only a small number of this ROM, an early version of IL-DEVEL ROM, were made and are not stocked or sold by HP.

Those marked with an asterisks share their identifying number, and should not be used in the HP-41 at the same time. Of two functions with the same XROM ID, that at the lowest address (i.e. the lowest numbered port) will be accessed first and the other ignored. So be sure to choose to use discretion when choosing your own XROM number if you want to avoid these problems.

APPENDIX E

X R O M   S T R U C T U R E

XROM's are located at whole 4k blocks of addresses. The lowest addresses in an XROM, and a few of the highest have special functions. The remainder may be filled in any way. The locations in the 4k blocks must be filled by ten bit words, giving 2^10 different codes. They may be read as instructions, or as alpha-numeric data. The following summary, adapted from J. Schwartz' January 1983 PPC Conference paper, should be taken into account when studying an application ROM, e.g. the MLDL-ROM. A listing can easily be prepared by using the MLDL-ROM functions DISASM and MNEM.

Relative address (hex)	Function of code at that address
X000	The XROM ID number in hexadecimal digits.
X001	The number of functions in the XROM (m), including the XROM name.
X002-3	Address of XROM name
X004-5	Address of first routine, program, etc.
X005-7	Address of second routine, etc.
"	" "
"	" "
X002+2n	Address of n'th routine
X003+2n	
"	" "
"	" "
X002+2m	Address of last (m'th) routine
X003+2m	(m < 64)
X004+2m	Compulsory null - 000.
X005+2m	Compulsory null - 000.
"	" "
"	" "
Add. of name	Name of ROM (running backwards)
"	" "
"	" "
Add. of Fn# 1	Start of Fn# 1 code
"	" "
"	" "
Add. of Fn# 2	Start of Fn# 2 code
"	" "
"	" "
"	" "
XFF4-A	Special interrupt jump locations (see table).
XFFB-E	ROM name abbreviation and revision #.
XFFF	ROM checksum for diagnostic use

Word pairs containing function addresses:

```
      First word of pair:     b  Ø   0   0   0   0  a11 a1Ø a9 a8
      Second word of pair:    Ø  Ø  a7  a6  a5  a4  a3  a2  a1 aØ
```

This results in the following address in this 4k block if 0000 is zero:

```
     p3 p2 p1 pØ a11 a1Ø a9 a8 a7 a6 a5 a4 a3 a2 a1 aØ
```

Where pØ-3 is the bit representation of the 4k page number and aØ-11 represent the relative offset from the beginning of the page.When 0000 is not equal to zero it must be added to pØ-3. For more information see the function AFAT.

If the two words would read ØØ3, ØFF this would represent a starting address of a function at address X3FF (hex). The bit b in the first word indicates USER code or microcode. If set the address is the start of a USER code program (e.g. 2ØØ, ØA1 in the printer module is address 6ØA1, start of USER code program "PRPLOT")

APPENDIX F

THE SPECIAL INTERRUPT POINTS

xFF4	Interrupts during PSE loop.
xFF5	Interrupts after each program line.
xFF6	Wake-up with no key down.
xFF7	Interrupts when turned off.
xFF8	Interrupts when peripheral flag is set.
xFF9	wake-up with ON key.
xFFA	Wake-up after memory lost.

Do not use these points unless you know exactly what you are doing. Careless use of these points may cause CRASHES.

FUNCTION INDEX

FUNCTION	PAGE
AFAT	6
CAT	14
CBT	14
CLBL	8
COD	11
COMPILE	10
COPYR	8
DECOD	11
DFAT	7
DISASM	13
GE	15
LOCA	10
LROM	11
MNEM	13
MMTORAM	5
MOVE	7
RAMWR	4
REG>ROM	8
ROMCHKX	11
ROMSUM	12
ROM>REG	16
'RROM	14
'WROM	16
'XRSYN	16
---	10
---	15

CARE AND WARRANTY

Eprom care

Store the eprom set in a dry and clean place. Make sure that the feet of the eprom's are protected against bending. Some feet could brake from the eprom and make it worthless. Do not connect any external power supply to the eproms. Protect the eproms against static charges, otherwise irrepairable damage to the eprom's can result. Do not remove under any circumstances the labels on the eproms, for these labels protect the eprom's against losing there data by accident through too much U.V. light on the eprom's.

Limited 180 day's warranty

The 83120A ERAMCO MLDL-Eprom set is warranted against defects in materials and workmanship affecting electronic performance, -but not software content for 180 day's from the date of original purchase. If you sell your unit or give it as a gift, the warranty is automatically transferred to the new owner and remains in effect for the original 180 day's period. During the warranty period, we will repair or, at our option, replace at no charge a product that proves to be defective, provided you return the product, shipping prepaid, to ERAMCO SYSTEMS, or their official service representative.

WHAT IS NOT COVERED

This warranty doesn't apply if the product has been damaged by accident misuse or as the result of service or modification by other than ERAMCO SYSTEMS or their official service representative.

No other express warranty is given. Any other implied warranty of merchantabillity or fitness is limited to the 180 day's period of this written warranty. In no event shall ERAMCO SYSTEMS be liable for consequential damages. This liability shall in no way exceed the catalog price of the product at the moment of sale.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of manufacture. ERAMCO SYSTEMS shall have no obligation to modify or update products once sold.

USER CODE PROGRAMS

```
01 LBL 'wrom
02 XEQ 00
03 'from page?
04 AON
05 PROMPT
06 AOFF
07 ASTO X
08 +'000
09 ARCL X
10 +'559
11 XEQ 01
12 CLA
13 ARCL X
14 +'55A
15 ARCL X
16 +'AB3
17 XEQ 01
18 CLA
19 ARCL X
20 +'AB4
21 ARCL X
22 +'FFF
23 LBL 01
24 COD
25 1
26 ROM>REG
27 E3
28 ST/ L
29 RDN
30 LAST X
31 +
32 WRTRX
33 RCL Z
34 RTN
35 LBL 00
36 SF 25
37 SIZE?
38 FC?C 25
39 GTO 03
40 276
41 PSIZE
42 LBL 05
43 'filename
44 AON
45 PROMPT
46 AOFF
47 ASTO 00
48 823
49 CREATE
50 ,
51 SEEKR
52 RTN
53 LBL 03
54 '* no x-fun *
55 AVIEW
56 STOP
57 GTO 05
58 LBL 'rrom
59 SF 25
60 SIZE?
61 FC?C 25
62 GTO 03
63 276
64 X>Y?
65 PSIZE
66 LBL 04
67 'filename
68 AON
69 PROMPT
70 AOFF
71 ,
72 SEEKR
73 'to page
74 AON
75 PROMPT
76 AOFF
77 COD
78 XEQ 02
79 RDN
80 XEQ 02
81 RDN
82 SF 00
83 LBL 02
84 1,275
85 FS? 00
86 RDN
87 FS?C 00
88 1,273
89 READRX
90 REG>ROM
91 RTN
92 LBL 03
93 '* no x-fun *
94 AVIEW
95 STOP
96 GTO 04
97 END
```

```
01 LBL'xrsyn
02 RCL Y
03 4
04 /
05 INT
06 160
07 +
08 RCL Z
09 4
10 MOD
11 64
12 *
13 RCL Z
14 +
15 END
```

ERAMCO SYSTEMS

ADDENDUM ESMLDL-OS EPROM

This is the addendum for the ERAMCO mldl operating system eprom. This addendum belongs to your manual, so keep it with the manual. We will only describe the differences between the old and new functions if the way of operation is different of the way it is described in the original manual. Most of the functions in the ESMLDL-OS rom are the same. We also tried to keep the xrom numbers the same. This has been possible for all the original functions of the rom. Only the user code routines included in the rom are deleted. Instead of these programs we have added some new functions and two functions replacing the user code programs.

There is one important change for all the routines that are asking for a hexadecimal input in the alpha register. Only the number digits 0-9 and the ascii characters A-F are valid data input. For every other character in the alpha register you will get "DATA ERROR"

We will discuss all the functions of the old rom and tell if there are any changes

11,01 RAMWR

There is one little change. If you are at address $0000 and you try to do a backstep, you will find yourself at $0001. This is done to avoid a wrap around to $FFFF or even worse to get the machine crashed. So if you really want to backstep to $FFFF you have to press backarrow once and continue at this address.

11,02 MMTORAM

In this routine we have made quit a lot of changes. First and most important change is the input of the starting address. The starting address is given in the alpha register as a four digits hexadecimal address.
To make the loading of subsequent programs even easier the first empty address after the loaded program is returned in the same format as the input format. This allows you to load the next program without having to find out first where to start loading.

The second major change in this function is the automatic compiling of the program before it is loaded into erom. Because it is not allowed to load the program that is at the .END the compiling routine will first make the .END a normal END. This automatic compiling and the input in alpha with the new address returned here and also the new function IPAGE (described later) are making the loading of user code programs a lot easier, for you don't have to worry about addresses anymore.

Because there are two compilers present in the rom now (about CMPDL you will find more later) you have the option to chose which compiler is used with user flag 3.
If this flag is set, it will make use of the normal compiler. When this flag is cleared it will make use of CMPDL. When this option is chosen, you can not copy the program to main memory anymore, for the labels are deleted and so the calculator does not know where to jump to if an GTO or XEQ is encountered.

Error messages : see the COMPILE and CMPDL functions

11,03 AFAT

11,04 DFAT

11,05 MOVE

11,06 CLBL

11,07 COPYR

11,08 ROMSUM

11,09 REG>ROM

11,10 ---

There are no changes made in these routines, except the general one that they will only accept real hex digits as input in alpha.

11,11 COMPILE

There are a few small changes in this routine. The first one is that if the program to be compiled is the one that is ended with the .END, it will change the .END to a normal end before compiling.
The second one is that we will pack after every change of a 2 byte instruction to a 3 byte instruction. This is done to avoid unnessassary changes of 2 byte instructions. It will make the program a little slower.
When a GTO or XEQ is encountered that does not have a corresponding label, you will get the normal error message. But now you will be placed at the step causing the error instead of the step after the error causing one.
When the compilation is finished the program counter is placed at the first step of the program.

11,12 LOCA

11,13 LROM

11,14 COD

11,15 DECOD

11,16 ROMCHKX

11,17 ROM>REG

11,18 MNEM

11,19 DISASM

These functions are not changed except for the general change that they will accept only real hex input e.g. digits 0-9 and A-F.

11,20 CAT

There is no change in this function, but sinds there are a lot of people using the HP-41CX we want to warn the users of the CX that the CAT function doesn't always behave in the same manner as with the CV or C. The port catalog will not always start at the desired point. It is possible, depending on some status information, that you will crash the calculator if you try this function. So be carefull when you use the CX.

11,21 CBT

11,22 SYNT

11,23 GE

11,24 ---

These functions are not changed.

We have discussed all the original functions of the rom and start now with five extra functions.

11,25 SAVEROM

This function replaces the old user code routine 'WROM. With this function you are also able to save the contents of an entire rom on cassette tape. The input format for this function is a name in the alpha register and the desired page number in x.
A file will be created on tape of 640 registers, occupying 20 records. The file identifier of this file is $07. It means that the files are presented in the DIR as :

ERAMCO SYSTEMS

```
                NAME          ??,S         640
```

We have chosen for a nonexistent file type to be sure that the data is not accidently destroyed. Therefore the file is also automaticcally secured after creation. This means that you save 7 records per file compared to 'WROM. Now you will be able to get the maximum number of roms on your tape (e.g. 24 files).

To get the maximum number of files on your tape it is recommended to do a NEWM with 27 file directory entry's. You can write 12 files on each side of the tape then. After having written 12 files you should protect the tape of rewinding from one side to the other by creating a dummyfile ENDTAPE of 300 registers.

Error messages :

PAGE > 15	if you tried to save a nonexistent pagenumber (> 15).
NO HPIL	if there is no il loop present in the system.
DUP FL NAME	if there is already a file with the specified name.
TRANSMIT ERR	see the il manual.

11,26 GETROM

This is the opposite of the SAVEROM function. Input format is the same, so name must be in alpha and page number must be in x.
It will read back the contents of the rom file and put it in the desired ram page. There is no checking done if the specified page is a ram page. This is to allow you to get a rom file to a page that is not switched on.

Error messages :

PAGE > 15	if you tried to read back to a nonexistent page (> 15).
NO HPIL	if there is no hpil loop present in the system.
FL TYPE ERR	if the specified file isn't a rom file.
TRANSMIT ERR	see the il manual.

11,27 CMPDL

This is in fact nearly the same function as the normal COMPILE. The only difference is that this function will delete the numeric labels in the program while compiling. This shortens the program and speeds it up. You can not use this function when the program has a GTO ind or XEQ ind in it. This is for the fact that these two functions need the labels to locate the location to jump to.
When this function is executed, it will make use of the user registers to hold the addresses of the deleted labels. Therefore make sure that the number of allocated registers is more then the number of labels in the programs. If you don't take care of this,

the calculator might crash.
To protect the compiled status as much as possible we change the .END to a normal END in case the program you are compiling is terminated by the .END.

Error messages :

GTO/XEQ IND you tried to compile a program that contains a gto or xeq ind. The program pointer is placed at the step causing the error.

For more error messages see COMPILE

11,28 IPAGE

This function sets up a ram page to load user programs and/or asembler code functions. The entire specified page is cleared and the specified xrom number and the name in alpha are written at the appropriate places.

Input : alpha holds the name of the rom
x holds the desired page to be initialized
At the prompt you fill in the wanted xrom number. Be sure to use only xrom numbers in the range 1 to 32.
There is no checking done on the input, because it is possible to use other xrom numbers, but you can not execute a function in a rom with a xrom number higher then 32.

The name that will be written to alpha consists of the first eleven characters in the alpha register when you have no more then 12 characters. If you have more then 12 characters in alpha, the name will be made of the first 11 characters that are left in the display when you would have displayed it. In other words the first 11 characters will be used of the last 12 characters in the alpha register.
When you have less then 11 characters, the last character will be an underscore.

Output of this function is in alpha the address of the first empty word as it is used by the function MMTORAM.

Error message :

PAGE > 15 if you want to use a nonexistent page

11,29 MKPR

This function allows you to make your programs private, even when you don't have a card reader. At the prompt you must fill in the name of the program that has to become private, or if you want to make the current program private press alpha twice (compare COPY of the normal instruction set).

Error message :

ERAMCO SYSTEMS

ROM	if you want to make a program in rom private.
NONEXISTENT	if you want to make a nonexistent program private or if you want to make a function private.

If you want to get an annotated listing of the ESMLDL-OS rom you can get it by ordering it from ERAMCO SYSTEMS. This has to be done by sending your order to :

ERAMCO SYSTEMS
W. van Alcmade str. 54 or Hugo de Grootkade 46-huis
1785 LS Den Helder 1052 LV Amsterdam
The Netherlands The Netherlands

The cost for this listing is $4 for shipment and $3.50 for the cost of the copy. Include this in your order in cash money and in USA dollars only. As soon as we receive your order, the listing will be shipped within 5 workdays.