=##105-41+1=##105-41+1=##105-41+1=##105-41+1=##105-41+1=##105-41+1=##105-41+1=##105-41+1=##1 5-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-41·1-22105-4 5-41·J-22105-41·J-22105-41·J-22105-41·J-22105-41·J-22105-41·J-22105-41·J-22105-41·J-22105-41·J-22105-4

05-41

**AN OPERATING SYSTEM FOR THE HP-41** 

HANDHELD COMPUTER AND PERIPHERALS

BY

## -££105-4 5-41-1-12 C'105-4 5-41-1-11 -**T**(05-4 5-41-1-11 d:105-4 -41-1-12 105-4 5-41-1-12 C105-4 5-41-I-III £105-4 5-41-J-121 d'i05-4 -41-1-12 11:105-4 -41-1-12 11:05-4 -4[-]-12 1105-4 -4[·]-11'| 11:10S-4 -41-1-11 生105-4 -41-1-11 C'105-4 -41-1-121 £105-41 -41-1-1210 Ci05-41--41-1-1210 £105-41

-41-<u>1-1</u>10 1105-41·1-11105-41·1-11105--41·1-2'105-41·1-2'105-41·1-2 £\_105-4[+]=£\_105-4[+]=£\_105-4[ -41·1-12/105-41·1-12/105-41·1-12/10

C'105-41-1-11 -41-1-52105-41 ££105-41•]=££1 -41-1-0-105-41 **E**105-41-1-51 -41·J=02105-41 **EC105-41**-1-521 

E'105-41·1-5'105-41·1-6'105-41·1-6'105-41 -41+F52105-41+F52105-41+F52105-41+F521 [·]-12:105-41-]

E'105-4[·]=E'105-41 -41-1-01105-41-1-011 **E'i05-4[·]-<u>5</u>'i05-4**1 -41·1-12/105-41:1-12/1 **ECIOS-41**·1-5Ci 05-41 -41-1-12105-41-1-121

1105-41·1-11105-41·1-11105-41·1-11109 -41·J-12/105-41·J-12/105-41·J-12/105-41·J #105-41·J-#105-41·J-#105-41·J-#109

-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41.1-12105-41 -41·J=12'105-41'+J=12'105-4



TIOS-41.1-TIOS-41.1-TI A TASK-ORIENTED REFERENCE MANUAL WITH

41·J=52105-41:-J=52105-41

COS-41-1-11 OS-41-1-12 OVER 70 ORIGINAL AND USEFUL PROGRAMS 105-41-1-11 OS-41-1-12 -41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41 ۲٬۱۰۵-4۱۰۱-۳٬۱۰۵-4۱۰۱-۳٬۱۰۵-4۱۰۱-۳٬۱۰۲-۳٬۱۰۵-4۱۰۱-۳٬۱۰۵-4۱۰۱-۳٬۱۰۵-4۱۰۱-۳٬۱۰

105-4 1-1-11 105-41-1-12 05-4 105-4 1-1-12-1 i05-4 I I-III 105-4 1-1-121 105-41 105-4 1-]-[1] 105-41 [•]-[[[i 105-41 1-1-11 105-4 105-41 [•]-**[**[]i 105-41 1-1-11-1

-I-II



THOMAS W. BEERS

AND

TED W. BEERS

41.]-11.'i 2105-41 41-1-021 5-41 1-1-1 2105-41 41.1-11'i 2105-41

105-41

05-41·1-521 05-41·1-521 41.1-52105-411-1-52105-41 105-41·J-12'i 05-41·J-12'i

i05-41·1=22105-41:+1=22105-41

11-1-12/105-41-1-12/105-41-1-12/1

105-41-1-12105-41-1-12105-41

11-1-12-105-41-1-12-1 05-41-1-12-1

105-41-1-12105-41-1-12105-41

H1.J=0\_105-41.J=0\_105-41.J=0\_1 105-41-1-12105-41-1-12105-41

41.1-52105-41.1-52105-41.1-521



105-41 I-II'I Ľ105-41 -41-1-11 Ľi05-41 41-1-121 £4105-41 41-1-11

## 0S-41

AN OPERATING SYSTEM FOR THE HP-41

HANDHELD COMPUTER AND PERIPHERALS

Co-Authors:

Thomas W. Beers Professor of Forestry Purdue University

Ted W. Beers Student in Electrical Engineering Purdue University

© Copyright, 1985. Thomas W. Beers, Department of Forestry and Natural Resources, Purdue University, West Lafayette, Indiana 47907. All rights reserved. This book, or portions thereof, may be reproduced only with written permission of one of the authors.

#### PREFACE

The treatise which follows describes a body of information to which we have ascribed the name "OS-41 Operating System." It is not a dictionary of functions in the usual alphabetized fashion, but instead a guide which attempts to organize functions, techniques, and programs in a way that is task-oriented. Our ultimate goal is to make the HP-41 handheld computer as "friendly" as possible. We hope we succeed in this regard, but only you, the user, can decide that. In any event, this two-year "spare-time" project has been educationally rewarding to the authors.

We sincerely acknowledge and dedicate this book to our wives, Carolyn and Marion, whose continued forbearance was necessary and often pivotal. We were also privileged to have available the word-processing skills of Patty Karnehm, who typed the entire manuscript (draft, after draft, after draft!). We thank Patty for her patience and attention to detail, without which the project would surely have floundered.

The material contained in this book is supplied without representation or warranty of any kind. Neither the publisher nor the authors shall have any liability, consequential or otherwise, resulting from the use of the material included in this book.

-1-

# TABLE OF CONTENTS

PREFACE       1         INTRODUCTION       A. The OS-41 Operating System       4         B. Conventions Used       8         1. Function Prefixes       8         2. OS-41 Program Name Format       9         3. Other Conventions       10         4. Documented Example       13         C. Error Recovery       15         Chapter 1: THE MOVEMENT OF PROGRAMS
INTRODUCTION          A. The OS-41 Operating System       4         B. Conventions Used       8         1. Function Prefixes       8         2. OS-41 Program Name Format       9         3. Other Conventions       10         4. Documented Example       13         C. Error Recovery       15
A. The OS-41 Operating System
B. Conventions Used
1. Function Prefixes       8         2. 0S-41 Program Name Format       9         3. Other Conventions       10         4. Documented Example       13         C. Error Recovery       15         Chapter 1: THE MOVEMENT OF PROGRAMS
2. OS-41 Program Name Format
3. Other Conventions       10         4. Documented Example       13         C. Error Recovery       15         Chapter 1: THE MOVEMENT OF PROGRAMS
4. Documented Example       13         C. Error Recovery       15         Chapter 1: THE MOVEMENT OF PROGRAMS
C. Error Recovery
Chapter 1: THE MOVEMENT OF PROGRAMS
A. Introduction - A Procedure to Remove Superfluous ENDs 20
B. Functions, Programs, and Their Use
C. Program Movement Program Instructions
D. Program Listings
Chapter 2. THE MOVEMENT OF NUMERIC AND A PHANUMERIC DATA
A Introduction
$\begin{array}{cccccccccccccccccccccccccccccccccccc$
C Numeric and Alphanumeric Data Movement Program Instructions
D Program Listings and Tochnical Details
Index of Numeric and Alphanumeric Data Neuerent Preserve
Listing
2 The Data Meyement Algorithm
5. The VER bug
Chapter 3. THE MOVEMENT OF ACCTT DATA
A Introduction A Dreadure to Incent Decender into an ACOLI 5/1.
Records into an ASCII File . 214
D. Functions, Programs, and Ineir Use
C. ASCII Data Movement Program Instructions
D. Program Listings and lechnical Details
1. Index of ASCII Data Movement Program Listings 252
2. The Pseudo-ASCII Data Format
3. Instructions for ALSTO and SCROLL
Chapter 4. THE MOVEMENT AND DUDI TRATION OF ENTIDE MEDIA
A Introduction OF ENTIRE MEDIA
P. Europeione December and Their Use
D. Functions, Programs, and Ineir Use
C. Media Movement and Duplication Program Instructions 287
D. Program Listings and Technical Details
1. Index of Media Movement and Duplication Program Listings . 312
2. The "NaMe" File Format
3. The Packaging of Programs According to Function 313
Chapter 5: OPERATIONS TO CLEAR AND PURGE, AND THOSE RELATED TO KEY
ASSIGNMENTS, FLAGS, AND ALARMS
A. Introduction - The Master Clear Procedure
B. Functions, Programs, and Their Use
C. Clear and Purge Program Instructions
D. Program Listings

Chapter 6: THE CATALOGUING OF FUNCTIONS, PROGRAMS, AND FILES	
A. Introduction	362
B. Functions, Programs, and Their Use	364
C. Catalog Program Instructions	367
D. Program Listings	375
LITERATURE CITED	377
Appendix A: OS-41 FLAG AND DATA REGISTER USAGE	378
1. Flag Usage	378
2. Register Usage	382
Appendix B: THE USE OF OS-41 PROGRAMS AS SUBROUTINES	390
Appendix C: OS-41 SUPPORT PROGRAMS	<b>39</b> 2
Appendix D: BARCODE FOR PROGRAMS	394
INDEX	466

#### INTRODUCTION

#### A. The OS-41 Operating System

When the HP-41C Programmable Calculator was introduced several years ago, it represented a major step forward in the personal computing field. In a rather short time, Hewlett-Packard responded to user needs by providing software modules and other powerful peripherals to expand the HP-41 system. Perhaps the most significant addition is the Hewlett-Packard Interface Loop (HP-IL) Module, which makes the HP-41 the controller of an ever-growing family of portable (often battery-powered and briefcase portable) computing products such as thermal, impact, and inkjet printers, plotters, mass storage devices, modems, video monitors, digital multimeters, and data acquisition units. Truly the HP-41 has become a handheld computer.

Unfortunately, however, the introduction of new hardware for the HP-41 has outpaced the software needed to make the system work. Associated with this system is the HP-RPN programming language which, while fundamentally straightforward, becomes less amenable to spontaneous mental recall as more peripherals are added.

We originally started this project very simply to organize the numerous and very specific procedures available to move data and programs throughout the HP-41 system. However, as the project evolved it became evident that this aspect is but one part of what we have chosen to call the OS-41 Operating System. As perusal of the TABLE OF CONTENTS will show, "the system" is more than a collection of programs; rather, we have tried to incorporate state-of-the-art options, recommendations, warnings, and examples in the form of an educational treatise. Thus, in addition to the heart of the system, featuring OS-41 programs described in Chapters 1

-4-

through 6 which address the movement of programs, data, and entire media as well as media duplication, clearing, purging, and cataloguing, there are sections included (where appropriate) which deal with such topics as error recovery, removal of superfluous ENDs, a data movement algorithm, a serious "bug" related to the Card Reader VER function, programs to process pseudo-ASCII data, and identification and alphabetizing of backup mass storage media. Four appendices include various technical details: documentation of OS-41 flag and data register usage; how to use the OS-41 programs as subroutines; listings of several support programs; and complete barcode for all OS-41 programs.

We have tacitly assumed a modest knowledge of the HP-41<sup>1/</sup> system on the part of the user, and though it may be true that those who perceive a need for this text probably need little more direction in how to use it, we recommend that all users proceed as indicated below.

- Read the rest of this INTRODUCTION, especially Section B: Conventions Used.
- As needs or interests dictate, turn to the pertinent chapter and read its Introduction, noting especially the table(s) outlining the array of operations available.
- 3. Locate the specific cell of interest in this "operations table" and proceed to the indicated descriptions in Section B, where the various options open to you are described.
- Note that the Section B descriptions proceed from the basic functions, procedures, and programs and culminate (if appropriate)

-5-

<sup>1/</sup>Unless otherwise indicated "HP-41" is used throughout to indicate HP-41C, HP-41CV, and HP-41CX models.

in a brief citation of the OS-41 program written to satisfy the pertinent need. If no OS-41 program is appropriate, it will be so stated and we recommend use of the techniques already available (and indicated). If, however, an OS-41 program is cited, complete details on its use can be found in Section C in the same chapter; the program listings and other technical details are in Section D (same chapter); the program barcode is in Appendix D.

After some experience and an awareness of the coding scheme we employed to name the OS-41 programs, steps 2, 3, and 4 can frequently be circumvented by simply loading and executing the proper OS-41 program. Two examples follow:

- 1. To copy a program from main memory to mass storage, one should use the program M2MSP and be prompted for the necessary parameters, rather than recalling the name of the Interface Loop Module function (WRTP) and worrying about parameter details and such items as verifying and securing the file; M2MSP performs these operations automatically.
- To print the contents of a <u>d</u>ata file stored in extended memory, one should use the program XM2PD, rather than... (we'll spare you the messy details!).

Continued use of the OS-41 Operating System will make apparent the desirability of having many of the OS-41 programs readily available in a custom module, but such is not currently available. We provide in this text program listings and barcode and let the needs of the user dictate which programs to maintain in main or extended memory or to have in mass storage. It is, however, entirely feasible to group frequently used

-6-

programs into "packages" and move them as one unit. Details on packaging programs can be found in Chapter 4, Section D.

In the development of this text we have read, assimilated, learned from, and employed the published works of many HP-41 pioneers. Citations are made where appropriate and the detailed source is given in the LITERATURE CITED section. Special note, however, must be made of certain authors whose indicated publications represent true landmarks and should adorn the shelf of any serious HP-41 user.

- \*\* Keith Jarett, "HP-41 Synthetic Programming Made Easy" (1982) and "HP-41 Extended Functions Made Easy" (1983).
- \*\* Richard J. Nelson, founder of the Personal Programming Center (PPC), long-time editor of the "PPC Calculator Journal" and coordinator of "PPC ROM User's Manual" (1981).
- \*\* Cary E. Reinstein, "HP-41/HP-IL System Dictionary" (1982 and 1983).
- \*\* William C. Wickes, "Synthetic Programming on the HP-41C" (1980).

Lest the very knowledgeable reader wonder why, we have deliberately tried to minimize the use of "synthetic programming"<sup>2/</sup> (or other such "low-level" techniques) because of our feeling that most users will not be immersed in the subject to this extent.

<sup>2/</sup>Synthetic programming as defined by W. C. Wickes, the term's originator, is '...using quirks in the 4l operating system to create "new" functions not included in the intended design function set.'

# B. <u>Conventions Used</u>

# 1. Function Prefixes

Whenever functions are referenced throughout this text, a prefix is used to indicate the module or peripheral in which the function resides, as detailed in Table 1. For example, the Card Reader function WPRV is indicated by CR: WPRV.

Table 1. Function Prefix Terms.

Prefix Module or Peripheral		
no prefix	function built into HP-41	
A/D	Autostart/Duplication Module (41-15042)	
CR	Card Reader (HP 82104A)	
cx1/	function built into only HP-41CX	
IL	HP-IL Interface Loop Module (HP 82160A)	
Р	Peripheral Printer (HP 82143A)	
PL	Plotter Module (HP 82184A)	
PPC	PPC ROM Module	
Т	Time Module (HP 82182A)	
W	Optical Wand (HP 82153A)	
XF	Extended Functions/Memory Module (HP 82180A)	
XIO	Extended I/O Module (HP 82183A)	

1/Since the HP-41CX contains built-in versions of the Extended Functions/Memory and Time Modules, any function prefixed by "XF" or "T" can be considered a "CX" function as well. 2. <u>OS-41 Program Name Format</u>

OS-41 program names have one of two formats, depending on whether data are moved between two media or one medium is operated on. The two formats are

src2dsttype, and

srcoper,

where

src = source medium term (see Table 2), dst = destination medium term (see Table 2), type = data type (see Table 3), and oper = operation (see Table 4).

Tables 2, 3, and 4 below describe the possibilities for <u>src</u>, <u>dst</u>, <u>type</u>, and <u>oper</u>.

As examples of the two program name formats, the OS-41 program to copy numeric or alphanumeric <u>data</u> from <u>extended memory</u> to <u>mass storage</u> is XM2MSD, whereas the OS-41 program to <u>catalog mass storage</u> is MSCAT.

Term	Medium
A	ALPHA register
В	barcode (source only)
С	magnetic cards
D	display (destination only)
м	main memory
MS	mass storage
Р	printer (destination only)
x	X register
XM	extended memory

Table 2. Terms Used for Source and Destination Media.

Term	Data Type		
A	ASCII data		
D	numeric or alphanumeric data		
Р	program		
Ε	entire medium		

Table 3. Terms Used for Types of Data.

Table 4. Terms Used for Operations.

Term	Operation		
CAT	catalog		
DUP	duplicate		
DUPA	duplicate alphabetically		
х	expunge (purge)		
Z	zero (clear)		

# 3. Other Conventions

The following paragraphs are provided as reference material to be read in detail if and when the user is puzzled by other symbols used or procedures followed throughout the text. Therefore, they should be skimmed on first reading. On the other hand, the user prepared to implement a particular OS-41 program should read more carefully Subsection 4, Documented Example. a. Keys:

HP-41 keys are described either by function ("the multiply key") or by key name, enclosed in brackets ("press [SST] or [shift][BST]"). When enclosed in brackets, key names are as they appear on the HP-41 keyboard, not necessarily exactly like the functions they represent (for example, [CATALOG] instead of [CAT]).

b. Main memory SIZE vs. file size:

Unfortunately, Hewlett-Packard reference manuals refer to both the number of main memory data registers currently allocated and the amount of a medium used by a file with the same word, "size". To avoid confusion, throughout this text the number of main memory data registers is denoted by capital "SIZE", while file space used is denoted by lower-case "size". The only exception to this rule is when an OS-41 program must prompt for the size of an extended memory or mass storage file; display restrictions preclude lower-case "size". However, the prompt is always prefixed with other descriptive words to remind the user that a <u>file</u> size is requested.

c. Prompt and display terms:

Since the HP-41 has only a twelve-character display to prompt the user for data or to output results, certain abbreviations are necessary. This text uses several short terms to indicate media and other items as detailed in Table 5.

-11-

Term	Meaning
+R	continued record (output)
+REC	continued record (input)
bbb	beginning register or record
BCODE	barcode
CCC	character
CDS	magnetic cards
DST	destination medium
eee	ending register or record
II	increment
MM	main memory
MS	mass storage
PRGM	program
R	register or beginning of record (output)
REC	beginning of record (input)
RRR	temporary register
SRC	source medium
TMP	temporary storage
ХM	extended memory
XMS	extended memory modules

Table 5. Terms Used to Prompt or Output.

d. Temporary storage register usage:

Some OS-41 programs require that data be passed through one or more main memory data registers. To insure maximum flexibility, the user is allowed to specify the block of registers, known as temporary storage registers, to be used. The only stipulation is that registers already used by the program cannot be chosen for temporary storage. The user should be aware that specifying a large block of temporary storage registers will decrease program execution time, but of course will also destroy the previous contents of more data registers.

## 4. <u>Documented Example</u>

The following example of a <u>fictitious</u> OS-41 program illustrates the format used in the OS-41 program instructions. Boxed comments point out particular features worth noting in this hypothetical excerpt from the Example section of M2MSPL (<u>Main memory TO Mass S</u>torage, <u>Program Lines</u>).

Copy lines 10 through 15 of main memory program L1-3A to mass storage file "L1-3B" (which already exists) registers 10 and up. Use main memory registers 20 through 24 for temporary storage.

Note that file names are quoted to distinguish them from program names. This convention is followed throughout the text except in sections A and B of each chapter, where program names are quoted to distinguish them from functions.

-13-

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create mass storage file "L1-3B" with size 16 or more. Enter a program L1-3A into main memory with 15 or more lines.

#### [XEQ]"SIZE"025

Within examples, the ALPHA key is represented by quotes for brevity. The actual SIZE required in this example is 025 or greater.

#### 2. [XEQ]"M2MSPL"

1.

The OS-41 program must be in main memory to proceed. It may be copied to main memory from barcode (see Appendix D).

a. MM PROGRAM? L1-3A [R/S]

Actually any global label within a program can be used to specify the entire program "file" in main memory.

b.	MM bbb.eee=?	10.015 [R/S]	
с.	MS FILE?	L1-3B [R/S]	
d.	MS bbb=?	[R/S]	10.00

Whenever [R/S] is used to select a default value, the default is briefly displayed for verification.

e. TMP bbb.eee=? 20.024 [R/S]

If the temporary storage registers specified include any data registers already used by a program, a warning of the form "ROO-Rmax USED" is displayed and the prompt is repeated.

3. \*WORKING\* DONE

Many OS-41 program instructions include a Comments section to explain any additional information that may be needed to fully understand the operation of a program.

## C. Error Recovery

For many users, the concept of "error recovery" invokes images of losing all progress made in a problem, requiring a total restart only after deciphering and correcting the error at hand. However, a combination of the HP-41's built-in diagnostic capability and some common sense can turn almost any error into a minor nuisance that won't disrupt progress towards a solution. The general procedure for error recovery during an OS-41 (or any other) program is as follows:

#### <u>Step</u> <u>Instructions</u>

- 1. When an error message is displayed, don't panic! Wait until the PRGM annunciator (which indicates the program is running) disappears, then press [PRGM]. The HP-41 will display the program line where the error stopped execution.
- 2. Press [PRGM] again to exit PRGM mode. By combining knowledge of the error <u>message</u> and the instruction on the error <u>line</u>, correct the problem if possible (see clarification below), then press [R/S] to continue with the interrupted program. Note that any VIEWed or AVIEWed data prior to the error stop won't be displayed, but this rarely affects program execution.

The meaning of "if possible" above must be clarified. A program can be continued normally by pressing [R/S] after correcting the error only if these requirements are met:

- 1. The act of correcting the error must not alter any data used by the program (for example, the stack, ALPHA, or data register contents used).
- The act of correcting the error must not alter the position of the "program pointer", which determines where the program will restart when [R/S] is pressed.

In practice, requirement 1 usually means that any function or procedure that requires parameters and is programmable <u>cannot</u> be used for error recovery because stack, ALPHA, or data registers are frequently needed to enter parameters. However, any non-programmable function <u>can</u> be used for error recovery, since required parameters do not affect stack, ALPHA, or data registers. For example, if a program stops with the error message "NONEXISTENT" at a line with the instruction "STO 19", the XF: PSIZE function may not be usable to recover since it requires the new SIZE (20 or greater) in the X register. The SIZE function could definitely be used, however, since it is non-programmable and does not affect the stack, ALPHA, or data registers.

Requirement 2 simply means that any error recovery operation that uses or manipulates programs must be avoided. This includes cataloguing or purging main memory programs. For example, if a program stops with the error message "PACKING" followed by "TRY AGAIN" at a line with the instruction "GETSUB", purging a main memory program using the CLP function to make room is <u>not</u> a legitimate error recovery method, since the program pointer will be moved. Instead, the SIZE could perhaps be reduced with the SIZE function to make room in main memory for the program called by XF: GETSUB.

Experienced HP-41 users will recognize that the status of flag 25 (error ignore flag) has some effect on how the calculator responds to an error condition. Specifically, when flag 25 is set, the <u>next</u> error encountered is ignored; that is, the error-causing operation is not performed and the HP-41 does not stop. Flag 25, however, is cleared by the error-causing operation, so any subsequent error will cause the HP-41 to stop. Users should be aware that when flag 25 is used deliberately in this way they may have to look "back one error" when diagnosing a problem, since flag 25 set causes the first error to go undetected. Some OS-41 programs use flag 25 to extend the utility of a function. For example, the XF: SEEKPTA function is used with flag 25 set to determine if an extended memory file exists. Upon testing, then, if flag 25 is cleared (an "error"

-16-

occurred), the file doesn't exist, but if flag 25 remains set (no "error" occurred), the file does exist. In all OS-41 programs that use such methods, flag 25 is set only immediately <u>before</u> it is needed, and cleared immediately <u>after</u> so it won't interfere with normal error detection throughout the rest of the program.

One type of error is unique and warrants brief explanation. When the HP-41 attempts to execute a function that resides in a module or peripheral not currently inserted, it will display "NONEXISTENT", logically enough. However, pressing [PRGM] to view the line at fault will <u>not</u> display the function name. Instead, an XROM (eXternal <u>ROM</u>) number will be displayed. This is because the HP-41 does not store an external ROM's function in program memory by name but rather by a two-part number (which is more memory-efficient). To determine the module or peripheral corresponding to an XROM number, compare its prefix number (before the comma) with a table of known XROM prefix numbers. For convenience, all modules and peripherals mentioned in this text and their corresponding XROM prefix numbers are listed in Table 6.

An error caused by a missing module or peripheral <u>is</u> recoverable. Simply turn the HP-41 off, insert the required module or peripheral, then turn the HP-41 on and press [R/S].

Table 6. XROM Prefix Numbers.

XROM Prefix	Module or Peripheral
10	Autostart/Duplication Module (41-15042)
10 and 20	PPC ROM Module
17 and 18	Plotter Module (HP 82184A)
23	Extended I/O Module (HP 82183A)
25	Extended Functions/Memory Module (HP 82180A)

-17-

XROM Prefix	Module or Peripheral	
26	Time Module (HP 82182A)	
27	Optical Wand (HP 82153A)	
28	HP-IL Interface Loop Module (HP 82160A)	
29	Peripheral Printer (HP 82143A)	
30	Card Reader (HP 82104A)	

If, after checking all possibilities suggested on the preceding pages, it is determined that an error cannot be corrected, then it may be time to "start over at the beginning."

Occasionally, the HP-41 will experience a condition commonly referred to as a "trauma", or "crash". The usual symptoms are that the display does not change and pressing individual keys has no effect - a keyboard lock-up exists. The following trauma recovery tips, arranged in increasing order of severity, can be used to release the HP-41.

- 1. Press and hold [<-], press and release [R/S], then release [<-].
- 2. Remove the battery pack from the HP-41 for a few seconds, then replace it.
- 3. Users with the Card Reader can try this sequence: insert any card. If the card is not pulled through, remove the battery pack for a few seconds, then replace it. The card should be pulled through.

The above techniques should reset the HP-41 without resulting in MEMORY LOST. The next two techniques are for more serious traumas and will cause MEMORY LOST.

4. Press and hold [<-], then press [ON]; this is the Master Clear described in Chapter 5, Section A.

5. Remove the battery pack and reverse the individual batteries (not possible with the HP 82120A Rechargeable Battery Pack). Replace the battery pack, then press and hold [ON] for about ten seconds. Remove the battery pack again and reverse the individual batteries, then replace the battery pack.

Additional tips are provided in "HP-41 Extended Functions Made Easy" by Jarett (1983), from which the suggestions above were extracted.

## Chapter 1: THE MOVEMENT OF PROGRAMS

#### A. <u>Introduction</u>

Most HP-41 users are primarily interested in developing or running programs. Saving programs or keeping them readily available requires the movement of programs among the various media in the HP-41 system. Using functions and OS-41 programs, programs can be stored, recalled, copied, and printed. Table 1-1 shows the array of possibilities: functions available in modules such as the Extended Functions/Memory Module (like SAVEP and GETP); functions available in the HP-IL Module (like WRTP and READP); functions in peripherals such as the Printer (like PRP and LIST); and OS-41 programs written to streamline or extend these basic functions (like "M2XMP" and "XM2MSP").

Table 1-1 shows the common types of program movement along with the functions and/or programs to effect such movement. In Section B, additional details are given so the user can avoid most references to the HP-41 or peripheral manuals. <u>OS-41 program instructions</u> can be found in Section C.

Most of the cells in Table 1-1 contain two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

## A Procedure to Remove Superfluous ENDs

Many of the OS-41 programs detailed in Section C use the XM: GETSUB or IL: READSUB functions, which will insert a superfluous END statement before the program copied to main memory if the last program in main memory doesn't have its own END. This somewhat irritating occurrence is necessary to prevent the last program from being lost. Although the extra ENDs cause

-20-

9-	BARCODE (B)	25 "PRBC"				
ų	PRINTER (P)	24 P & IL: PRP P & IL: LIST "PRPR"	26 See text.	28 See text.	29 See text.	30 See text.
-4	MAGNETIC CARDS (C)	24 CR: WRPV CR: WALL See text.	26 See text.	28 See text.	29 See text.	30 See text.
ñ	MASS STORAGE (MS)	23 IL: WRTP IL: WRTPV IL: WRTA "M2MSE"(292) "M2MSP"(36)	26 "XM2MSE"(298) "XM2MSP"(42)	27 IL: RENAME XIO: COPYFL "MS2MSP"(48)	29 See text.	30 "B2MSP"(54)
-2	EXTENDED MEMORY (XM)	23 XF: SAVEP "M2XME"(288) "M2XMP"(34)	26 "XM2XMP" ( 40 )	27 "MS2XME" (302) "MS2XMP" (46)	29 See text.	29 "B2XMP"(50)
-1	MAIN MEMORY (M)	23 "M2MP"(32)	25 XF: GETP XF: GETSUB "XM2ME"(296) "XM2MP"(38)	26 IL: READP IL: READP IL: READA "MS2MP"(44) "MS2MP"(44)	28 CR: RSUB CR: MRG See text.	29 W: WNDSUB W: WNDLNK
KT SUBSECTION NUMBER	DESTINATION SOURCE	MAIN MEMORY (M)	EXTENDED MEMORY (XM)	MASS STORAGE (MS)	MAGNETIC CARDS (C)	BARCODE (B)
Ĥ		Ч	N	m	4	Ъ

Table 1-1. Functions and Programs to Move Programs Within the HP-41 System.

no harm to the HP-41 or main memory programs, they are annoying to see and consume valuable memory space. The following procedure can be used to remove one or more accumulated ENDs from main memory.

#### <u>Step</u> Instructions

- 1. Put the HP-41 in PRGM mode.
- 2. Press [shift][CATALOG][1], then press [R/S] immediately to stop the catalog listing.
- 3. Press [SST] repeatedly to step through the catalog until two ENDs are displayed consecutively (not including the permanent .END.). If no consecutive ENDs are displayed, main memory is free of superfluous ENDs, and the following "END purge" is not needed.
- 4. If two consecutive ENDs are encountered, at the second END press [<-] to display the first program statement.
- 5. If "Ol END" is displayed, press [<-] to delete the superfluous END, then go to Step 6. If something other than "Ol END" is displayed, the END represents an unlabelled program and should not be deleted. Instead, press [shift][GTO][.][O][O][O] and key in a LBL statement to name the program. Then go to Step 2.
- 6. Press [SST]. If a LBL statement is displayed, go to Step 2 and continue, otherwise go to Step 5.

Occasional use of this procedure will help insure that the maximum amount of main memory is available for programs and data.

## B. Functions, Programs, and Their Use

# 1-1.1/ MAIN MEMORY to MAIN MEMORY

a. To duplicate a program in main memory with all parameters prompted for, use the OS-41 program

"M2MP" (Main memory TO Main memory, Program).

## 1-2. MAIN MEMORY to EXTENDED MEMORY

a. Use XF: SAVEP (A:program, filename OR A:program OR A:, filename) $2^{2/2}$ 

to copy the specified main memory program (or the current program if none is specified) to extended memory under the specified file name (or the program name if none is specified). If a program file already exists with the specified file name, it will be replaced.

b. To copy multiple main memory programs - up to the entire main memory program memory - to an extended memory "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"M2XME" (Main memory TO eXtended Memory, Entire).

Refer to Chapter 4 for details.

c. To copy a program from main memory to extended memory with all parameters prompted for, use the OS-41 program

"M2XMP" (Main memory IO eXtended Memory, Program).

#### 1-3. MAIN MEMORY to MASS STORAGE

a. Use IL: WRTP (A:program, filename OR A:program OR A:, filename)

to copy the specified main memory program (or the current program if none is specified) to mass storage under the specified file name (or the program name if none is specified). If a program file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function.

b. Use IL: WRTPV (A:program, filename OR A:program OR A:, filename)

to copy a main memory program to mass storage in the same manner as WRTP, except the resulting file will be private; that is, protected from being listed, edited, or recorded on any medium.

<sup>1/</sup>Section numbers are keyed to Table 1-1 and indicate movement from row number
(given first) to column number.

 $<sup>2^{\</sup>prime}$  Throughout this text, the parameters needed in the stack registers and ALPHA (indicated by A) register are indicated in parentheses.

c. Use IL: WRTA (A:filename)

to copy all main memory information (including all programs and data registers) to mass storage under the specified file name. If a "write all" file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function.

d. To copy multiple main memory programs - up to the entire main memory program memory - to a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"M2MSE" (Main memory <u>TO</u> Mass <u>Storage</u>, <u>Entire</u>).

Refer to Chapter 4 for details.

e. To copy a non-private program from main memory to mass storage with all parameters prompted for, use the OS-41 program

"M2MSP" (<u>Main memory TO Mass Storage</u>, <u>Program</u>).

# 1-4. MAIN MEMORY to MAGNETIC CARDS<sup>3/</sup>

- a. To copy a program from main memory to magnetic cards, go to the desired program, put the HP-41 in PRGM mode, and insert magnetic cards into the Card Reader.
- b. Use CR: WPRV

to copy a private program from main memory to magnetic cards.

c. Use CR: WALL

to copy all main memory information (including all programs and data registers) to magnetic cards.

d. No OS-41 program is appropriate here since the above procedure and functions adequately copy programs from main memory to magnetic cards.

#### 1-5. MAIN MEMORY to PRINTER

Although the "printer" is stressed in all printer subsections, the discussion applies generally to a TV monitor if an HP-IL Video Interface is connected.

a. To print a program in main memory, use

P & IL: PRP

 $<sup>\</sup>frac{3}{Before}$  using the CR: VER function to verify magnetic cards, refer to Chapter 2, Section D for an explanation of the VER bug that can alter extended memory.

b. To print a selected section of a program in main memory, go to the desired program and line number, then use

P & IL: LIST

Specify the number of lines to list when prompted.

- c. No OS-41 program is appropriate here since the above functions adequately print or list programs in main memory. However, the OS-41 Support program "PRPR" (<u>PR</u>int <u>PR</u>ogram), discussed in Appendix C, can be used to print a descriptive header at the top of a program listing. The number of main memory program registers required and the number of bytes in the program are printed in a formatted manner.
- 1-6. MAIN MEMORY to BARCODE
  - a. Use the program "PRBC" (PRint HP 82162A BarCode)

to copy a main memory program to HP 82162A Thermal Printer barcode. "PRBC" is described in the Plotter Module Owner's Manual, page 172-173, and uses the PL: BCP function to generate in the ALPHA register a bit pattern representing each row of program barcode.

- 2-1. EXTENDED MEMORY to MAIN MEMORY
  - a. Use XF: GETP (A:filename)

to copy the specified program file from extended memory to main memory. If the last program in main memory doesn't have its own END, it will be replaced.

b. Use XF: GETSUB (A:filename)

to copy the specified program file from extended memory to main memory. Unlike when using GETP, the last program in main memory will never be replaced.

c. To recall to main memory all programs from an extended memory "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"XM2ME" (eXtended Memory TO Main memory, Entire).

Unlike most other "read all" procedures, "XM2ME" appends to, <u>not</u> replaces, the previous contents of main memory. Refer to Chapter 4 for details.

d. To copy a program from extended memory to main memory with all parameters prompted for, use the OS-41 program

"XM2MP" (eXtended Memory <u>IO</u> Main memory, Program).

- 2-2. EXTENDED MEMORY to EXTENDED MEMORY
  - a. To duplicate a program file in extended memory (<u>under a new</u> <u>destination file name</u>) with all parameters prompted for, use the OS-41 program

"XM2XMP" (eXtended Memory TO eXtended Memory, Program).

- 2-3. EXTENDED MEMORY to MASS STORAGE
  - a. To copy the entire contents of extended memory (program, data, and ASCII files) to a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"XM2MSE" (eXtended Memory TO Mass Storage, Entire).

Refer to Chapter 4 for details.

b. To copy a program file from extended memory to mass storage with all parameters prompted for, use the OS-41 program

"XM2MSP" (eXtended Memory TO Mass Storage, Program).

- 2-4. EXTENDED MEMORY to MAGNETIC CARDS
  - a. Automatic movement of programs in this fashion is difficult due to Card Reader restrictions. However, if one wants to copy a program from extended memory to magnetic cards, first use the OS-41 program

"XM2MP" (eXtended Memory TO Main memory, Program)

to copy the program to main memory, then use one of the options described in Subsection 1-4 to copy the program to magnetic cards.

- 2-5. EXTENDED MEMORY to PRINTER
  - a. Automatic printing of programs in this fashion is difficult due to Printer restrictions. However, if one wants to print a program in extended memory, first use the OS-41 program

"XM2MP" (eXtended Memory IO Main memory, Program)

to copy the program to main memory, then use one of the functions described in Subsection 1-5 to print or list the program.

- 3-1. MASS STORAGE to MAIN MEMORY
  - a. Use IL: READP (A:filename)

to copy the specified program file from mass storage to main memory. If the last program in main memory doesn't have its own END, it will be replaced. b. Use IL: READSUB (A:filename)

to copy the specified program file from mass storage to main memory. Unlike when using READP, the last program in main memory will never be replaced.

c. Use IL: READA (A:filename)

to recall all main memory information (including all programs and data registers) from the specified mass storage "write all" file. The previous contents of main memory will be lost.

d. To recall to main memory all programs from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2ME" (Mass Storage TO Main memory, Entire).

Unlike most other "read all" procedures, "MS2ME" appends to, <u>not</u> replaces, the previous contents of main memory. Refer to Chapter 4 for details.

e. To copy a program from mass storage to main memory with all parameters prompted for, use the OS-41 program

"MS2MP" (Mass Storage TO Main memory, Program).

- 3-2. MASS STORAGE to EXTENDED MEMORY
  - a. To recall the entire contents of extended memory (program, data, and ASCII files) from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2XME" (Mass Storage TO eXtended Memory, Entire).

The previous contents of extended memory will be lost. Refer to Chapter 4 for details.

b. To copy a program file from mass storage to extended memory with all parameters prompted for, use the OS-41 program

"MS2XMP" (<u>Mass Storage TO</u> eXtended <u>Memory</u>, <u>Program</u>).

### 3-3. MASS STORAGE to MASS STORAGE

a. Use IL: RENAME (A:old filename, new filename)

to change the name of a program file in mass storage.

b. To copy a mass storage file from the primary device to the one having the indicated IL-address, use

XIO: COPYFL (X:IL address to copy to; A:filename)

- c. For procedures to duplicate an <u>entire</u> mass storage medium, refer to Chapter 4.
- d. To duplicate a program file in mass storage (<u>under a new</u> <u>destination file name</u>) with all parameters prompted for, use the OS-41 program

"MS2MSP" (Mass Storage <u>TO</u> Mass Storage, Program).

#### 3-4. MASS STORAGE to MAGNETIC CARDS

a. Automatic movement of programs in this fashion is difficult due to Card Reader restrictions. However, if one wants to copy a program from mass storage to magnetic cards, first use the OS-41 program

"MS2MP" (Mass Storage TO Main memory, Program)

to copy the program to main memory, then use one of the options described in Subsection 1-4 to copy the program to magnetic cards.

## 3-5. MASS STORAGE to PRINTER

a. Automatic printing of programs in this fashion is difficult due to Printer restrictions. However, if one wants to print a program in mass storage, first use the OS-41 program

"MS2MP" (Mass Storage TO Main memory, Program)

to copy the program to main memory, then use one of the functions described in Subsection 1-5 to print or list the program.

#### 4-1. MAGNETIC CARDS to MAIN MEMORY

a. Use CR: RSUB (or simply insert a prerecorded magnetic card)

to copy a program from magnetic cards to main memory. If the last program in main memory doesn't have its own END, it will be replaced.

b. Use CR: MRG

to copy a program from magnetic cards to a specified location in the last program in main memory.

- c. To recall all main memory information (including all programs and data registers) from a set of "write all" magnetic cards, insert the cards into the Card Reader with the HP-41 out of PRGM mode. The previous contents of main memory will be lost.
- d. No OS-41 program is appropriate here since the above procedures and functions adequately copy programs from magnetic cards to main memory.

- 4-2. MAGNETIC CARDS to EXTENDED MEMORY
- 4-3. MAGNETIC CARDS to MASS STORAGE
- 4-4. MAGNETIC CARDS to MAGNETIC CARDS
  - Automatic movement of programs in these three fashions is difficult due to Card Reader restrictions. However, if one wants to copy a program from magnetic cards to one of these media, first use one of the options in Subsection 4-1 to copy the program to main memory. Then, use one of the options described in Subsection 1-2, 1-3, or 1-4 to copy the program to extended memory, mass storage, or magnetic cards, respectively.

## 4-5. MAGNETIC CARDS to PRINTER

- a. Automatic printing of programs in this fashion is difficult due to Card Reader and Printer restrictions. However, if one wants to print a program on magnetic cards, first use one of the options described in Subsection 4-1 to copy the program to main memory, then use one of the functions described in Subsection 1-5 to print or list the program.
- 5-1. BARCODE to MAIN MEMORY
  - a. Use W: WNDSUB (or simply scan the program barcode)

to copy a program from barcode to main memory. If the last program in memory doesn't have its own END, it will be replaced.

b. Use W: WNDLNK

to copy a program from barcode to main memory and execute it immediately. If the last program in main memory doesn't have its own END, it will be replaced.

c. No OS-41 program is appropriate here since the above procedure and functions adequately copy programs from barcode to main memory.

## 5-2. BARCODE to EXTENDED MEMORY

a. To copy a program from barcode to extended memory with all parameters prompted for, use the OS-41 program

"B2XMP" (Barcode TO eXtended Memory, Program).

- 5-3. BARCODE to MASS STORAGE
  - a. To copy a program from barcode to mass storage with all parameters prompted for, use the OS-41 program

"B2MSP" (Barcode TO Mass Storage, Program).

- 5-4. BARCODE to MAGNETIC CARDS
  - a. Automatic movement of programs in this fashion is difficult due to Card Reader restrictions. However, if one wants to copy a program from barcode to magnetic cards, first use one of the options described in Subsection 5-1 to copy the program to main memory, then use one of the options described in Subsection 1-4 to copy the program to magnetic cards.

## 5-5. BARCODE to PRINTER

a. Automatic printing of programs in this fashion is difficult due to Printer restrictions. However, if one wants to print a program in barcode, first use one of the options described in Subsection 5-1 to copy the program to main memory, then use one of the functions described in Subsection 1-5 to print or list the program.

# C. Program Movement Program Instructions

The program instructions found on the following pages should enable the user to apply the desired program. One or more examples are provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 1-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings.

M2MP (<u>Main memory <u>TO</u> <u>Main memory</u>, <u>Program</u>)</u>

Program No.: 41P1-1 Other Programs in This File: none

## <u>Purpose</u>

To duplicate a program in main memory with all parameters prompted for. M2MP is useful for creating a backup copy of a program under development.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Execute M2MP and follow the prompt:
  - a. MM PROGRAM?

Key in the name of the main memory program (six characters or less) to be duplicated and press [R/S].

 The HP-41 will display "\*WORKING\*" as the specified main memory program is duplicated. When completed, the HP-41 will beep, stop, and display "DONE".

#### <u>Examples</u>

1. Duplicate main memory program M2MP.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1.	MM PROGRAM?	[XEQ]"M2MP" M2MP [R/S]	
2.			*WORKING* DONE

## Comments

- 1. Since M2MP uses the XF: GETSUB function, a superfluous END may be placed before the duplicate program in main memory. The END can be removed with the procedure described in Section A.
- M2MP temporarily creates a dummy extended memory file named "..." of the same size as the specified program. Therefore, enough extended memory registers must be available to save the program. Also, if a file named "..." already exists, it will be replaced.
USE THIS PAGE FOR NOTES

M2XMP (Main memory TO eXtended Memory, Program)

Program No.: 41P1-2 Other Programs in This File: M2MSP

#### Purpose

To copy a program from main memory to extended memory with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Execute M2XMP and follow the prompts:
  - a. MM PROGRAM?

Key in the name of the main memory program (six characters or less) to be copied to extended memory and press [R/S].

b. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy the specified main memory program to and press [R/S]. To default to the name of the specified program, just press [R/S].

 The HP-41 will display "\*WORKING\*" as the specified main memory program is copied to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

#### <u>Examples</u>

 Copy main memory program M2XMP to extended memory under the file name "P1-2A".

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. a. b. 2.	MM PROGRAM? XM FILE?	[XEQ]"M2XMP" M2XMP [R/S] P1-2A [R/S]	*WORKING*
			DUNL

2. Copy main memory program M2XMP to extended memory under the same name.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. a. b. 2.	MM PROGRAM? XM FILE?	[XEQ] <b>"M2XMP"</b> M2XMP [R/S] [R/S]	M2XMP *WORKING* DONE

## Comments

1. If the specified extended memory file already exists, it will be replaced.

M2MSP (Main memory TO Mass Storage, Program)

Program No.: 41P1-3 Other Programs in This File: M2XMP

#### Purpose

To copy a program from main memory to mass storage with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Execute M2MSP and follow the prompts:
  - a. MM PROGRAM?

Key in the name of the main memory program (six characters or less) to be copied to mass storage and press [R/S].

b. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy the specified main memory program to and press [R/S]. To default to the name of the specified program, just press [R/S].

c. REPLACE? Y/N

Press [Y] (the multiply key) to indicate that an existing, secured mass storage file with the name of the destination file is to be replaced. Press any other key to insure the destination file will not replace any secured file.

2. The HP-41 will display "\*WORKING\*" as the specified main memory program is copied to mass storage, secured, and verified. When completed, the HP-41 will beep, stop, and display "DONE".

#### <u>Examples</u>

 Copy main memory program M2MSP to mass storage under the file name "P1-3A", replacing an existing, secured mass storage file named "P1-3A".

<u>Step</u>	Prompt	User Input	Display
prelim	inary -	"M2MSP,P1-3A" [XEQ]"WRTP" "P1-3A" [XEQ]"SEC"	
1. a. b. c. 2.	MM PROGRAM? MS FILE? REPLACE? Y/N	[XEQ]"M2MSP" M2MSP [R/S] P1-3A [R/S] [Y]	YES *WORKING* DONE

2. Copy main memory program M2MSP to mass storage under the same name, <u>not</u> replacing any existing, secured mass storage file.

<u>Step</u>	Prompt	User Input	<u>Display</u>
1. a. b. c. 2.	MM PROGRAM? MS FILE? REPLACE? Y/N	[XEQ]"M2MSP" M2MSP [R/S] [R/S] [N]	M2MSP NO *WORKING*
			DUNE

#### <u>Comments</u>

- The purpose of the "REPLACE? Y/N" prompt is to prevent unplanned loss of mass storage files. To this end, the user should press "YES" (the multiply key) <u>only</u> when a mass storage file is supposed to be replaced.
- If the specified mass storage file already exists and is <u>unsecured</u>, it will be replaced, regardless of the answer to the "REPLACE? Y/N" prompt.

XM2MP (eXtended Memory TO Main memory, Program)

Program No.: 41P2-1 Other Programs in This File: MS2MP

#### Purpose

To copy a program from extended memory to main memory with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Execute XM2MP and follow the prompt:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to main memory and press [R/S].

 The HP-41 will display "\*WORKING\*" as the specified extended memory file is copied to main memory. When completed, the HP-41 will beep, stop, and display "DONE". To <u>execute</u> the program just copied to main memory, <u>if</u> it has the same name as the extended memory file, press [R/S].

#### Examples

1. Copy extended memory file "M2XMP" to main memory.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary – Use M2XMP to copy itself to extended memory file "M2XMP".

1.			
a.	XM FILE?	M2XMP [R/S]	
2.			*WORKING*
			DONE

#### Comments

1. Since XM2MP uses the XF: GETSUB function, a superfluous END may be placed before the last program in main memory. The END can be removed with the procedure described in Section A.

# USE THIS PAGE FOR NOTES

XM2XMP (eXtended Memory <u>TO</u> eXtended Memory, Program)

<u>Program No.</u>: 41P2-2 <u>Other Programs in This File</u>: MS2MSP

#### Purpose

To duplicate a program file in extended memory (<u>under a new file name</u>) with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Execute XM2XMP and follow the prompts:
  - a. XM SRC FILE?

Key in the name of the source extended memory file (six characters or less) and press [R/S].

b. <u>file</u> PRGM?

where

<u>file</u> = source extended memory file name.

Key in the name of any program (six characters or less) in the source extended memory file and press [R/S]. If the program name is the same as the source file name, just press [R/S].

c. XM DST FILE?

Key in the name of the destination extended memory file (six characters or less) to copy the source extended memory file to and press [R/S].

 The HP-41 will display "\*WORKING\*" as the program in the specified source extended memory file is copied from extended memory through main memory back to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

1. Duplicate the extended memory program file "P2-2A", which contains the program M2XMP, under the new file name "M2XMP".

Step <u>Prompt</u><u>User Input</u>Display preliminary - Use M2XMP to copy itself to extended memory file "P2-2A". 1. [XEQ]"XM2XMP" XM SRC FILE? P2-2A [R/S] a. b. P2-2A PRGM? M2XMP [R/S] с. XM DST FILE? M2XMP [R/S] 2. \*WORKING\* DONE

2. Duplicate the extended memory file "M2XMP", which contains the program M2XMP, under the new file name "P2-2B".

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use M2XMP to copy itself to extended memory file "M2XMP" (or follow Example 1).

1.		LXEQJ"XM2XMP"	
a.	XM SRC FILE?	M2XMP [R/S]	
b.	M2XMP PRGM?	[R/S]	M2XMP
с.	XM DST FILE?	P2-2B [R/S]	
2.			*WORKING*
			DONE

#### Comments

- 1. Since XM2XMP uses the XF: GETSUB function, a superfluous END may be placed at the end of main memory. The END can be removed with the procedure described in Section A.
- XM2XMP temporarily creates a dummy main memory program of the same size as the source extended memory file. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. If the specified destination extended memory file already exists, it will be replaced.

XM2MSP (eXtended Memory TO Mass Storage, Program)

Program No.: 41P2-3 Other Programs in This File: MS2XMP

#### Purpose

To copy a program file from extended memory to mass storage with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Execute XM2MSP and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to mass storage and press [R/S].

b. file PRGM?

where

file = extended memory file name.

Key in the name of any program (six characters or less) in the specified extended memory file and press [R/S]. If the program name is the same as the extended memory file name, just press [R/S].

c. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy the extended memory file to and press [R/S]. To default to the name of the extended memory file, just press [R/S].

d. REPLACE? Y/N

Press [Y] (the multiply key) to indicate that an existing, secured mass storage file with the name of the destination file is to be replaced. Press any other key to insure the destination file will not replace any secured file.

 The HP-41 will display "\*WORKING\*" as the specified extended memory file is copied from extended memory through main memory to mass storage, secured, and verified. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

 Copy the extended memory program file "P2-3A", which contains the program M2XMP, to mass storage file "P2-3B", replacing an existing, secured mass storage file named "P2-3B".

<u>Step Prompt User Input Display</u>

preliminary - Use M2XMP and M2MSP to copy themselves to extended memory file "P2-3A" and mass storage file "P2-3B", respectively; note that "P2-3B" is secured by M2MSP.

1.		[XEQ]"XM2MSP"	
a.	XM FILE?	P2-3A [R/S]	
b.	P2-3A PRGM?	M2XMP [R/S]	
с.	MS FILE?	P2-3B [R/S]	
d.	REPLACE? Y/N	[Y]	YES
2.			*WORKING*
			DONE

2. Copy the extended memory program file "M2XMP", which contains the program M2XMP, to mass storage file "M2XMP", <u>not</u> replacing any existing, secured mass storage file.

<u>Step Prompt User Input Display</u>

preliminary - Use M2XMP to copy itself to extended memory file "M2XMP".

1.		[XEQ]"XM2MSP"	
a.	XM FILE?	M2XMP [R/S]	
b.	M2XMP PRGM?	[R/S]	M2XMP
с.	MS FILE?	[R/S]	M2XMP
d.	REPLACE? Y/N	[N]	NO
2.			*WORKING*
			DONE

#### Comments

- 1. Since XM2MSP uses the XF: GETSUB function, a superfluous END may be placed at the end of main memory. The END can be removed with the procedure described in Section A.
- XM2MSP temporarily creates a dummy main memory program of the same size as the extended memory file. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. The purpose of the "REPLACE? Y/N" prompt is to prevent unplanned loss of mass storage files. To this end, the user should press "YES" (the multiply key) <u>only</u> when a mass storage file is supposed to be replaced.
- If the specified mass storage file already exists and is <u>unsecured</u>, it will be replaced, regardless of the answer to the "REPLACE? Y/N" prompt.

MS2MP (Mass Storage TO Main memory, Program)

Program No.: 41P3-1 Other Programs in This File: XM2MP

#### Purpose

To copy a program from mass storage to main memory with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Execute MS2MP and follow the prompt:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to main memory and press [R/S].

2. The HP-41 will display "\*WORKING\*" as the specified mass storage file is copied to main memory. When completed, the HP-41 will beep, stop, and display "DONE". To <u>execute</u> the program just copied to main memory, <u>if</u> it has the same name as the mass storage file, press [R/S].

#### Examples

1. Copy mass storage file "M2MSP" to main memory.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary – Use M2MSP to copy itself to mass storage file "M2MSP".

<b>T</b> •			
a.	MS FILE?	M2MSP [R/S]	
2.			*WORKING*
			DONE

#### Comments

1. Since MS2MP uses the IL: READSUB function, a superfluous END may be placed before the last program in main memory. The END can be removed with the procedure described in Section A.

# USE THIS PAGE FOR NOTES

MS2XMP (Mass Storage TO eXtended Memory, Program)

Program No.: 41P3-2 Other Programs in This File: XM2MSP

Purpose

To copy a program file from mass storage to extended memory with all parameters prompted for.

Minimum SIZE: 000

Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive Extended Functions/Memory Module (HP 82180A)

#### Instructions

1. Execute MS2XMP and follow the prompts:

a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to extended memory and press [R/S].

b. <u>file</u> PRGM?

where

<u>file</u> = mass storage file name.

Key in the name of any program (six characters or less) in the specified mass storage file and press [R/S]. If the program name is the same as the mass storage file name, just press [R/S].

c. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy the mass storage file to and press [R/S]. To default to the name of the mass storage file, just press [R/S].

 The HP-41 will display "\*WORKING\*" as the specified mass storage file is copied from mass storage through main memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

 Copy the mass storage program file "P3-2A", which contains the program M2MSP, to extended memory file "P3-2B".

<u>Step</u>	Prompt	User Input	_Display
preli	minary - Use M2M file "F	MSP to copy itself P3-2A".	to mass storage
1. a. b. c.	MS FILE? P3-2A PRGM? XM FILE?	[XEQ]"MS2XMP" P3-2A [R/S] M2MSP [R/S] P3-2B [R/S]	
2.			*WORKING*

2. Copy the mass storage program file "M2MSP", which contains the program M2MSP, to extended memory file "M2MSP".

DONE

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use M2MSP to copy itself to mass storage file "M2MSP".

1.		[XEQ]"MS2XMP"	
a.	MS FILE?	M2MSP [R/S]	
b.	M2MSP PRGM?	[R/S]	M2MSP
с.	XM FILE?	[R/S]	M2MSP
2.			*WORKING*
			DONE

#### Comments

- 1. Since MS2XMP uses the IL: READSUB function, a superfluous END may be placed at the end of main memory. The END can be removed with the procedure described in Section A.
- 2. MS2XMP temporarily creates a dummy main memory program of the same size as the mass storage file. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. If the specified extended memory file already exists, it will be replaced.

MS2MSP (Mass Storage TO Mass Storage, Program)

Program No.: 41P3-3 Other Programs in This File: XM2XMP

Purpose

To duplicate a program file in mass storage (<u>under a new file name</u>) with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

1. Execute MS2MSP and follow the prompts:

a. MS SRC FILE?

Key in the name of the source mass storage file (six characters or less) and press [R/S].

b. file PRGM?

where

<u>file</u> = source mass storage file name.

Key in the name of any program (six characters or less) in the source mass storage file and press [R/S]. If the program name is the same as the source file name, just press [R/S].

c. MS DST FILE?

Key in the name of the destination mass storage file (six characters or less) to copy the source mass storage file to and press [R/S].

d. REPLACE? Y/N

Press [Y] (the multiply key) to indicate that an existing, secured mass storage file with the name of the destination file is to be replaced. Press any other key to insure the destination file will not replace any secured file.

 The HP-41 will display "\*WORKING\*" as the program in the specified source mass storage file is copied from mass storage through main memory back to mass storage, secured, and verified. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

 Duplicate the mass storage program file "P3-3A", which contains the program M2MSP, under the new file name "M2MSP", replacing an existing, secured mass storage file named "M2MSP".

StepPromptUser InputDisplaypreliminary - Use M2MSP to copy itself to mass storage<br/>files "P3-3A" and "M2MSP"; note that both<br/>"P3-3A" and "M2MSP" are secured by M2MSP.

1.		[XEQ]"MS2MSP"	
a.	MS SRC FILE?	P3-3A [R/S]	
b.	P3-3A PRGM?	M2MSP [R/S]	
с.	MS DST FILE?	M2MSP [R/S]	
d.	REPLACE? Y/N	[Y]	YES
2.			*WORKING*
			DONE

 Duplicate the mass storage file "M2MSP", which contains the program M2MSP, under the new file name "P3-3B", not replacing any existing, secured mass storage file.

Step Prompt User Input Display preliminary - Use M2MSP to copy itself to mass storage file "M2MSP" (or follow Example 1). [XEQ]"MS2MSP" 1. a. MS SRC FILE? M2MSP [R/S] M2MSP PRGM? [R/S] M2MSP Ь. MS DST FILE? P3-3B [R/S] c. NO d. REPLACE? Y/N EN]

#### Comments

2.

1. Since MS2MSP uses the IL: READSUB function, a superfluous END may be placed at the end of main memory. The END can be removed with the procedure described in Section A.

\*WORKING\*

DONE

- MS2MSP temporarily creates a dummy main memory program of the same size as the source mass storage file. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. The purpose of the "REPLACE? Y/N" prompt is to prevent unplanned loss of mass storage files. To this end, the user should press "YES" (the multiply key) <u>only</u> when a mass storage file is supposed to be replaced.
- 4. If the specified destination mass storage file already exists and is <u>unsecured</u>, it will be replaced, regardless of the answer to the "REPLACE? Y/N" prompt.

B2XMP (Barcode IO eXtended Memory, Program)

Program No.: 41P5-2 Other Programs in This File: B2MSP

#### Purpose

To copy a program from barcode to extended memory with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Optical Wand (HP 82153A) Extended Functions/Memory Module (HP 82180A)

#### Instructions

1. Execute B2XMP and follow the prompts:

a. GTO . . ?

Press [R/S] <u>only if</u> the last program in main memory has its own END; otherwise, press [shift][GTO][.][.] to insure the last program has its own END, and go to Step 1.

b. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy barcode to and press [R/S].

c. file PRGM?

where

file = extended memory file name.

Key in the name of the program (six characters or less) in the barcode and press [R/S]. If the program name is the same as the extended memory file name, just press [R/S].

d. SCAN BARCODE

Scan the first line of the program barcode to copy to extended memory. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

e. W: RDY nnn

where

nnn = next line of barcode to scan.

Scan barcode line <u>nnn</u> and repeat this step until all lines of the barcode have been successfully scanned. 2. The HP-41 will display "W: WORKING", then "\*WORKING\*" as the program in the barcode is copied from main memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

#### <u>Examples</u>

 Copy the program P5-2B in barcode below to extended memory file "P5-2A".

#### BARCODE FOR P5-28

Program Registers Required: 3

ROW 1: LINES 1 - 2

# ROW 2: LINES 2 - 5

# 

<u>Step</u>	Prompt	User Input	<u>Display</u>	
prelim	inary - Press [ last pr	shift][GTO][.][.] ogram in main memo	to insure that t ory has its own E	the END.
1. a. b. c. d. e. 2.	GTO ? XM FILE? P5-2A PRGM? SCAN BARCODE W: RDY 02	[XEQ]"B2XMP" [R/S] P5-2A [R/S] P5-2B [R/S] (Scan 1st line) (Scan 2nd line)	W: WORKING *WORKING*	

 Copy the program P5-2B in barcode in Example 1 to extended memory file "P5-2B".

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Press [shift][GTO][.][.] to insure that the last program in main memory has its own END.

1.		LXEQJ"B2XMP"	
a.	GTO ?	[R/S]	
b.	XM FILE?	P5-2B [R/S]	
с.	P5-2B PRGM?	[R/S]	P5-2B
d.	SCAN BARCODE	(Scan 1st line)	
e.	W: RDY 02	(Scan 2nd line)	
2.			W: WORKING *WORKING* DONE

#### Comments

- 1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.
- 2. B2XMP temporarily creates a dummy main memory program of the same size as the program barcode. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. If the specified extended memory file already exists, it will be replaced.

# USE THIS PAGE FOR NOTES

B2MSP (<u>Barcode IO Mass Storage</u>, <u>Program</u>)

<u>Program No.</u>: 41P5-3 <u>Other Programs in This File</u>: B2XMP

#### Purpose

To copy a program from barcode to mass storage with all parameters prompted for.

Minimum SIZE: 000

Required Peripherals

Optical Wand (HP 82153A) Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

1. Execute B2MSP and follow the prompts:

a. GTO . . ?

Press [R/S] <u>only if</u> the last program in main memory has its own END; otherwise, press [shift][GTO][.][.] to insure the last program has its own END, and go to Step 1.

b. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy barcode to and press [R/S].

c. <u>file</u> PRGM?

where

<u>file</u> = mass storage file name.

Key in the name of the program (six characters or less) in the barcode and press [R/S]. If the program name is the same as the mass storage file name, just press [R/S].

d. REPLACE? Y/N

Press [Y] (the multiply key) to indicate that an existing, secured mass storage file with the name of the destination file is to be replaced. Press any other key to insure the destination file will not replace any secured file.

e. SCAN BARCODE

Scan the first line of the program barcode to copy to mass storage. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle. f. W: RDY nnn

where

<u>nnn</u> = next line of barcode to scan.

Scan barcode line <u>nnn</u> and repeat this step until all lines of the barcode have been successfully scanned.

 The HP-41 will display "W: WORKING", then "\*WORKING\*" as the program in the barcode is copied from main memory to mass storage, secured, and verified. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

1. Copy the program P5-3B in barcode below to mass storage file "P5-3A", replacing an existing, secured mass storage file named "P5-3A".

#### BARCODE FOR P5-3B

Program Registers Required: 3

ROW 1: LINES 1 - 2



#### ROW 2: LINES 2 - 5

# 

<u>Step</u>	Prompt	User Input	_Display_
prelim	inary – Press [ last pr Use M2M file "P by M2MS	shift][GTO][.][.] ogram in main memo SP to copy itself 5-3A"; note that " P.	to insure that the ory has its own END. to mass storage "P5-3A" is secured
1. a. b. c. d. e. f. 2.	GTO ? MS FILE? P5-3A PRGM? REPLACE? Y/N SCAN BARCODE W: RDY 02	[XEQ]"B2MSP" [R/S] P5-3A [R/S] P5-3B [R/S] [Y] (Scan 1st line) (Scan 2nd line)	YES W: WORKING *WORKING* DONE

2. Copy the program P5-3B in barcode in Example 1 to mass storage file "P5-3B", not replacing any existing, secured mass storage file.

<u>Step\_\_\_\_Prompt\_\_\_\_User\_Input\_\_\_Display</u>

preliminary - Press [shift][GTO][.][.] to insure that the last program in main memory has its own END.

1.		[XEQ]"B2MSP"	
a.	GTO ?	[R/S]	
b.	MS FILE?	P5-3B [R/S]	
с.	P5-3B PRGM?	[R/S]	P5-3B
d.	REPLACE? Y/N	[N]	NO
e.	SCAN BARCODE	(Scan 1st line)	
f.	W: RDY 02	(Scan 2nd line)	
2.			W: WORKING *WORKING* DONE

#### Comments

- 1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.
- 2. B2MSP temporarily creates a dummy main memory program of the same size as the program barcode. Therefore, enough main memory program registers must be available to move the program through main memory.
- 3. The purpose of the "REPLACE? Y/N" prompt is to prevent unplanned loss of mass storage files. To this end, the user should press "YES" (the multiply key) <u>only</u> when a mass storage file is supposed to be replaced.
- If the specified mass storage file already exists and is <u>unsecured</u>, it will be replaced, regardless of the answer to the "REPLACE? Y/N" prompt.

## D. <u>Program Listings</u>

The program listings on the following pages were printed by an HP 82162A Thermal Printer set to MAN mode. Use the table below to locate the page number of any OS-41 program movement program listing or barcode.

0S-41 Prog	ram											L	Page isting	Number for Barcode
M2MP	•	•	•	•	•	•	•	•	•	•	•	•	58	395
M2XMP	•	•	•	•	•	•	•	•	•	•	•	•	59	396
M2MSP	•	•	•	•	•	•	•	•	•	•	•	•	59	396
XM2MP	•	•	•	•	•	•	•	•	•	•	•	•	60	398
XM2XMP	•	•	•	•	•	•	•	•	•	•	•	•	61	399
XM2MSP	•	•	•	•	•	•	•	•	•	•	•	•	62	401
MS2MP	•	•	•	•	•	•	•	•	•	•	•	•	60	398
MS2XMP	•	•	•	•	•	•	•	•	•	•	•	•	62	401
MS2MSP	•	•	•	•	•	•	•	•	•	•	•	•	61	399
B2XMP	•	•	•	•	•	•	•	•	•	•	•	•	63	403
B2MSP	•	•	•	•	•	•	•	•	•	•	•	•	63	403

Table 1-2. Program Movement Program Listing and Barcode Locations.

```
OS-41 Program Movement Program Listing: M2MP
```

```
Program Registers: 13
Bytes: 86
```

01+LBL "M2MP" 02 \*41P1-1\* 03 FS? 50 04 GTO 00 05 CF 21 96 \*MM PROGRAM?\* 07 AON **08 PROMPT** 09 AOFF 10 ASTO X 11 \* \*WORKING\*\* 12 AVIEW 13+LBL 00 14 CLA 15 ARCL X 16 "+,..." 17 SAVEP 18 "..." **19 GETSUB** 20 PURFL 21 FS? 55 22 SF 21 23 "DONE" 24 ASTO X 25 BEEP 26 CLD 27 END

# OS-41 Program Movement Program Listing: M2XMP, M2MSP

Program Registers: 29 Bytes: 203

01+LBL =M2XMP*
02 *41P1-2*
03 SF 05
04 GTO 00
05+LBL "M2MSP"
06 =41P1-3=
07 CF 05
08+LBL 00
09 FS? 50
10 GTO 03
11 CF 06
12 CF 21
13 "MM PROGRAM?"
14 AON
15 PROMPT
16 90FF
17 ASTO Y
18 CF 23
19
20 FC2 05
20 10: 00 21 ##S*
22 - FTIF2-
27 0NN
24 PPOMPT
25 ONEE
20 HOLL
20 10 20
21 ULH 90 EC9 97
20 FU: 20 D0 00C1 V
27 HKUL I 70 FC9C 97
30 FU(U Z3 24 OUTEN
31 HYIEM 70 00T0 V
32 H510 X
33 157 00
34 610 02
35 KT
36 "REPLHCE? Y/N"
37 HYIEN
38+LBL 01
39 RDN
40 GETKEY
41 X=0?
42 GIO 01
43 71
44 "YES"
45 X≠Y?
46 "NO"
47 AVIEN
48 X=Y?
49 SF 06
50 RDN

52+LBL 02 53 \* \*WORKING\*\* 54 AVIEW 55+LBL 03 56 CLA 57 ARCL X 58 FS?C 86 59 UNSEC 60 CLA 61 ARCL Y 62 °F,\* 63 ARCL X 64 FS? 05 65 SAVEP 66 FC? 05 67 WRTP 68 CLA 69 ARCL X 70 FC? 05 71 VERIFY 72 FC?C 05 73 SEC 74 FS? 55 75 SF 21 76 DONE. 77 ASTO X 78 BEEP 79 CLD 80 END

51 RDN

OS-41 Program Movement Program Listing: XM2MP, MS2MP

Program Registers: 16 Bytes: 110

01+LBL "XM2MP" 02 \*41P2-1\* 03 SF 05 04 GTO 00 05+LBL "MS2MP" 06 \*41P3-1\* 07 CF 05 08+LBL 00 09 FS? 50 10 GTO 01 11 **"XM**" 12 FC? 05 13 "MS" 14 \*\* FILE?\* 15 AON 16 PROMPT 17 AOFF 18 ASTO X 19 \* \*WORKING\*\* 20 CF 21 21 AVIEW 22+LBL 01 23 CLA 24 ARCL X 25 FS? 05 26 GETSUB 27 FC?C 05 28 READSUB 29 FS? 55 30 SF 21 31 R† 32 "DONE" 33 ASTO X 34 BEEP 35 CLD 36 RTN 37 GTO IND Y 38 END

# OS-41 Program Movement Program Listing: XM2XMP, MS2MSP

Program Registers: 36 Bytes: 247

AIAIRI "YN2YNP"	51 AVIEW
02 = 4102-2=	52 X=Y?
02 THE E	53 SF 06
03 51 55 04 CTO 00	54 RDN
04 GIU 00 054101 *#C2#C0*	55 RDN
ØJ♥LDL N32N3F 04 =41D7 7=	56+LBL 02
05 ~41P3-3~	57 <b>* *WOR</b> KING**
07 LF 03	58 AVIEW
08+LBL 00	59+LBL 03
09 FS? 30	60 CLA
10 GIU 03	61 ARCL L
11 CF 06	62 FS? <b>0</b> 5
12 CF 21	63 GETSUB
13 "XM"	64 FC? 05
14 FC? 05	65 READSUB
15 "MS"	66 ES2 <b>R</b> 5
16 *F SRC FILE?*	67 GTO <b>84</b>
17 AON	68 CLA
18 PROMPT	69 ARCI X
19 ASTO L	70 FS2C 06
20 "H PRGM?"	71 HNSEC
21 CF 23	7241 Ri 04
22 PROMPT	77 010
23 FC? 23	74 ODCI V
24 CLA	17 MKOL I 75 ML M
25 FC? 23	73 F) 72 ODCL V
26 ARCL L	(D HKUL A 77 500 85
27 FC?C 23	(( F3( 03 70 courd
28 AVIEW	78 SHYEM
29 ASTO Y	79 FU? 00
30 =XH=	80 WKIP
31 FC2 95	81 CLH
72 •HS*	82 HRCL X
73 • H DST FILE?•	83 FC? 05
	84 VEKIFY
75 ONEE	85 FC?C 05
76 OSTO Y	86 SEC
77 EC2 05	87 CLA
30 CTO 02	88 ARCL Y
70 D+	89 PCLPS
AG -DEPLOCE2 Y/N*	90 FS? 55
AI OVIEU	91 SF 21
	92 "DONE"
	93 ASTO X
44 CETVEY	94 BEEP
45 Y=92	95 CLD
тј 6-0: Ис сто ві	96 END
40 UU DI 47 71	
46 - 476-	
40 IL3 40 V4V9	
97 AFT/ 50	
-UN- NC	

# OS-41 Program Movement Program Listing: XM2MSP, MS2XMP

Program Registers: 36 Bytes: 249

01+1BI *XM2MSP*	51 X=0?
A2 *41P2-3*	52 GTO 01
03 SE 05	53 71
94 GTO 99	54 "YES"
95+1 RI =NC2XMP=	55 X≠Y?
96 =d197-2*	56 <b>"N</b> O"
00 4173 2 07 PE 05	57 AVIEW
07 UF 0J ROAIDI RR	58 X=Y?
00VLDL 00 00 FC3 50	59 SF 06
07 F3( J0 10 FT0 07	60 RDN
	61 RDN
11 LF 106	62+LBL 02
12 CF 21	63 * *WORKING**
13 "XM"	64 AVIEN
14 FC? 85	65+1 BL - 03
15 "MS"	66 CLP
16 •F FILE?*	67 OPCI I
17 AON	68 FS2 85
18 PROMPT	69 CETSUR
19 ASTO L	70 FC2 85
20 "H PRGM?"	71 DEGNGUR
21 CF 23	71 KEHUJUD 79 EC9 05
22 PROMPT	72 FU: 0J 77 CTO 04
23 FC? 23	73 610 64
24 CLA	(4 ULH 75 0001 V
25 FC? 23	70 HKUL A 74 FC00 A4
26 ARCL L	75 75/6 85
27 FC?C 23	77 UNSEC
28 AVIEW	78+LBL 04
29 ASTO Y	79 CLA
30 "MS"	80 ARCL Y
31 FC2 <b>8</b> 5	81 <b>-</b> F,-
32 *XH*	82 ARCL X
33 "F FILE?"	83 FS? 05
34 PROMPT	84 WRTP
75 DOFF	85 FC? 05
76 FC2 27	86 SAVEP
77 CLO	87 CLA
78 FC2 27	88 ARCL X
79 00°1 I	89 FS? 05
49 ECOC 27	90 VERIFY
40 TC:C 23	91 FS?C 05
42 ACTO V	92 SEC
42 HOLU A 47 ECO 05	93 CLA
43 FU? BJ 44 CTO 03	94 ARCL Y
44 GIU 02 45 D4	95 PCLPS
	96 FS? 55
40 KEPLHUE / T/N° 42 OUTEN	97 SF 21
97 HYIEW ADALDE DI	98 DONE
987LUL UI 40 DDN	99 ASTO X
97 KUN 50 Petvev	100 BEEP
JU GEINEI	101 CLD
	102 END

## OS-41 Program Movement Program Listing: B2XMP, B2MSP

Program Registers: 34 Bytes: 232

01+LBL "B2XMP" 02 \*41P5-2\* 03 SF 05 94 GTO 99 05+LBL "B2MSP" 06 \*41P5-3\* 07 CF 05 08+LBL 00 09 CF 21 10 FS? 50 11 GTO 02 12 CF 06 13 "GTO .. ?" 14 PROMPT 15 "XM" 16 FC? 05 17 "MS" 18 \* FILE?\* 19 AON 20 PROMPT 21 ASTO Y 22 \*\* PRGM?\* 23 CF 23 24 PROMPT 25 A0FF 26 FC? 23 27 CLA 28 FC? 23 29 ARCL Y 30 FC?C 23 31 AVIEW 32 ASTO X 33 FS? 05 34 GTO 02 35 R† 36 \*REPLACE? Y/N\* 37 AVIEW 38+LBL 01 39 RDN 40 GETKEY 41 X=0? 42 GTO 01 43 71 44 •YES\* 45 X≠Y? 46 "NO" 47 AVIEW 48 X=Y? 49 SF 06 50 RDN

51 RDN 52+LBL 02 53 "SCAN BARCODE" 54 AVIEW 55 WNDSUB 56 \* \*WORKING\*\* 57 AVIEW 58 FS? 05 59 GTO 03 60 CLA 61 ARCL Y 62 FS?C 06 63 UNSEC 64+LBL 03 65 CLA 66 ARCL X 67 °F,\* 68 ARCL Y 69 FS? 05 70 SAVEP 71 FC? 05 72 WRTP 73 CLA 74 ARCL Y 75 FC? 05 76 VERIFY 77 FC?C 05 78 SEC 79 CLA 80 ARCL X 81 PCLPS 82 FS? 55 83 SF 21 84 "DONE" 85 ASTO X 86 BEEP 87 CLD 88 END

Chapter 2: THE MOVEMENT OF NUMERIC AND ALPHANUMERIC DATA

#### A. Introduction

Data, generally defined as numeric information (but which can include alphabetic characters in groups of up to six, known as alphanumeric data), can be stored, recalled, moved, printed, or viewed, using a multitude of functions or OS-41 programs. Table 2-1 shows the array of possibilities: calculator functions (like STO and RCL); functions available in modules such as the Extended Functions/Memory Module (like SAVER and GETR); functions available in the HP-IL Module (like WRTR and READR); functions in peripherals such as the Card Reader (like WDTA and RDTA); and OS-41 programs written to streamline or extend these basic functions (like "M2XMD" and "XM2MSD").

Table 2-1 shows the common types of data movement along with the functions and/or programs to effect such movement. In Section B, additional details are given so that the user can avoid most references to the HP-41 or peripheral manuals. Subsections 5-5 and 5-7 also discuss functions to manipulate the Y, Z, T, and L stack registers. <u>OS-41 program instructions</u> can be found in Section C. Remember that all data movement functions and programs work with both numeric <u>and</u> alphanumeric data.

Most of the cells in Table 2-1 contain two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

-64--

Table 2-1. Functions and Programs to Move Numeric and Alphanumeric Data Within the HP-41 System.

L-	DISPLAY (D)	70 VIEW PPC: BV "M2DD"(96)	72 "XM2DD"(116)	74 #MS2DD"(136)	76 "C2DD"(156)	79 VIEW (ST) X See text.	81 "B2DD"(184)
-9	PRINTER (P)	69 VIEW P & IL: PRREG P & IL: PRREGX P & IL: PRΣ PPC: BV "M2PD"(94)	72 "XM2PD"(114)	74 "MS2PD"(134)	75 "C2PD"(152)	79 P & IL: PRX P & IL: PRSTK P & IL: PRBUF P & IL: ADV	80 "B2PD"(182)
-2	X REGISTER (X)	68 RCL PPC: MS "RCLD"(92) "M2XD"(92)	72 XF: GETX "RCEX"(113) "XM2XD" (112)	74 "MS2XD" (132)	75 See text.	78 X<>X STO (ST) X See text.	80 W: WNDDTA
-4	MAGNETIC CARDS (C)	68 CR: WDTA CR: WDTAX CR: WDTAX CR: WALL "M2CD"(90)	72 "XM2CD"(108)	74 "MS2CD"(128)	75 "C2CD"(148)	77 See text.	80 See text.
ñ	MASS STORAGE (MS)	67 IL: WRTR IL: WRTRX IL: WRTA "M2MSD"(88)	71 "XM2MSE"(298) "XM2MSD"(104)	73 IL: RENAME XIO: COPYFL "MS2MSD"(124)	75 "C2MSD"(144)	77 "X 2MSD" ( 168)	80 "B2MSD"(178)
-2	EXTENDED MEMORY (XM)	67 XF: SAVER XF: SAVERX "M2XMD"(86)	71 CX: RESZFL "XM2XMD"(100)	73 "MS2XME"(302) "MS2XMD"(120)	75 "C2XMD"(140)	77 XF: SAVEX "STEX"(166) "X2XMD"(164)	80 "B2XMD"(174)
Ţ	MAIN MEMORY (M)	66 XF: REGMOVE XF: REGSWAP PPC: BE PPC: BM PPC: BM "M2MD"(84)	70 XF: GETR XF: GETRX "XM2MD"(98)	73 IL: READR IL: READRX IL: READA "MS2MD"(118)	74 CR: RDTA CR: RDTAX "C2MD"(138) See text.	76 STO PPC: SM PPC: BI "STOD"(161) "X2MD"(160)	79 W: WNDDTX "B2MD"(172)
SUBSECTION NUMBER	DESTINATION	AIN MEMORY (M)	XTENDED MEMORY (XM)	ASS STORAGE (MS)	AGNETIC CARDS (C)	( (or ALPHA) ÆGISTER (X)	3ARCODE (B)
TEXT	Ū.	Ξ	5 2	ε	4	м Х ГГ	6 6

-65-

#### B. Functions, Programs, and Their Use

- 1-1.1/ MAIN MEMORY to MAIN MEMORY
  - a. Use XF: REGMOVE  $(X:sss.dddnnn)^{2/}$ , or use the <u>B</u>lock <u>M</u>ove program

PPC: BM (X:nnn; Y:ddd; Z:sss)

to copy the contents of the block of nnn registers starting with main memory register sss into the block of main memory registers starting with ddd. Note: the contents of the destination registers (i.e., starting with ddd) are lost.

b. Use XF: REGSWAP (X:sss.dddnnn)

to <u>exchange</u> the contents of the block of nnn registers starting with main memory register sss with the block of main memory registers starting with ddd.

c. Use the Block Exchange program

PPC: BE (X:bbb.eeeii; Y:bbb.eeeii)

to exchange two blocks of main memory registers, each described by the control words in the X and Y registers; bbb indicates the beginning registers, eee the ending registers, and ii indicates the increment values.

Note: The control words are the standard form used by the DSE and ISG functions; also, the indicated blocks can overlap, but usually are independent.

d. Use the <u>Block</u> <u>Rotate</u> program

PPC: BR (X: <u>+nnn;</u> Y: bbb)

to rotate up (+nnn) or down (-nnn) <u>one register</u> the contents of nnn main memory registers starting with register bbb. BR works on the storage registers in the same way as R $\uparrow$  and RDN do on the stack.

e. To copy data from one block of main memory registers to another with all parameters prompted for, use the OS-41 program

"M2MD" (Main memory TO Main memory, Data).

<sup>1/</sup>Subsection numbers are keyed to Table 2-1 and indicate movement from row number (given first) to column number.

<sup>2&#</sup>x27;Throughout this text, the parameters needed in the stack registers and ALPHA (indicated by A) register are indicated in parentheses.

- 1-2. MAIN MEMORY to EXTENDED MEMORY
  - a. Use XF: <u>CRFLD<sup>3/</sup></u> (X:nnn; A:filename)

to create a file of nnn registers with the indicated file name, then use either

(1) XF: SAVER (working file or A:filename)

to copy consecutive registers starting with main memory register 00 up to the file limit or to main memory register SIZE-1, or

(2) XF: <u>SEEKPTA</u> (X:rrr; A:filename), or

XF: SEEKPT (X:rrr)

to move the pointer to register rrr of the named (or working) extended memory file, followed by

XF: SAVERX (X:bbb.eee)

to copy main memory registers bbb through eee.

b. To copy data from main memory to extended memory with all parameters prompted for, use the OS-41 program

"M2XMD" (<u>Main memory TO</u> e<u>X</u>tended <u>Memory</u>, <u>D</u>ata).

#### 1-3. MAIN MEMORY to MASS STORAGE

a. Use IL: <u>CREATE</u> (X:nnn; A:filename)

to create a file of nnn registers with the indicated file name, then use either

(1) IL: <u>WRTR</u> (A:filename)

to copy consecutive registers starting with main memory register 00 up to the file limit or to main memory register SIZE-1, or

<sup>3/</sup>An underline is used throughout this text to indicate that the function is the <u>logical</u> one to use to make the file named in the ALPHA register the "working file" (see the Extended Functions/Memory Module Owner's Handbook, page 23).

(2) IL: <u>SEEKR</u> (X:rrr; A:filename)

to move the pointer to register rrr of the named (or working) mass storage file, followed by

IL: WRTRX (X:bbb.eee)

to copy main memory registers bbb through eee.

b. Use IL: WRTA (A:filename)

to copy all main memory information (including all data registers and programs) to mass storage under the specified file name. If a "write all" file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function.

c. To copy data from main memory to mass storage with all parameters prompted for, use the OS-41 program

"M2MSD" (<u>Main memory TO Mass Storage</u>, <u>D</u>ata).

#### 1-4. MAIN MEMORY to MAGNETIC CARDS<sup>4/</sup>

a. Use CR: WDTA

to copy all main memory registers to magnetic cards; copying starts with register 00 and stores 32 registers per <u>card</u> (16 registers per side).

b. Use CR: WDTAX (X:bbb.eee)

to copy main memory registers bbb through eee to magnetic cards.

c. Use CR: WALL

to copy all main memory information (including all data registers and programs) to magnetic cards.

d. To copy data from main memory to magnetic cards with all parameters prompted for, use the OS-41 program

"M2CD" (Main memory <u>IO</u> magnetic <u>Cards</u>, <u>Data</u>).

#### 1-5. MAIN MEMORY to X REGISTER

 To recall (copy to the X register) contents of individual main memory registers, use

 $<sup>\</sup>frac{4}{Before}$  using the CR: VER function to verify magnetic cards, refer to Section D for an explanation of the VER bug that can alter extended memory.
- (1) RCL, (2) RCL IND, or (3) RCL IND (ST)<sup>5/</sup>
- b. Use the Memory to Stack program

PPC: MS (R6:nnn)

to recall to the X, Y, Z, T, and L registers respectively, the contents of the five consecutive registers beginning with register nnn, the number presently stored in register 6.

- c. The utility program "RCLD" facilitates the recall of individual main memory registers with addresses over 99. This program is an adaptation of a routine called "RCLL" (in "STOL" by Lafoiet, 1982). It is suggested that "RCLD" be assigned to the [RCL] or the shift [RCL] key for convenience. For more details, see the Comments section of OS-41 program "M2XD".
- d. To copy data from main memory to the X register with all parameters prompted for, use the OS-41 program

"M2XD" (<u>Main memory <u>TO</u> X register, <u>D</u>ata).</u>

## 1-6. MAIN MEMORY to PRINTER

Although the "printer" is stressed in all printer subsections, the discussion applies generally to a TV monitor if an HP-IL Video Interface is connected.

- a. To view (and print, if the printer is connected) individual registers, use
  - (1) VIEW,

-----

- (2) VIEW IND, or
- (3) VIEW IND (ST) (see footnote 5)
- b. To print all registers in main memory, use

P & IL: PRREG

- c. To print a selected block of main memory registers, use one of the following:
  - (1) for registers bbb through eee, use

P & IL: PRREGX (X:bbb.eee)

 $<sup>5^{/}</sup>$  In indirect stack functions such as this, the "IND" is achieved, as usual, by pressing the shift key, the "ST" is obtained by pressing the decimal point key, and the cursor prompt must be satisfied by pressing a "stack letter" key, X, Y, Z, T, or L (not in ALPHA mode).

(2) for registers currently designated as the statistical registers ("normally" R11-R16), use

**P** & IL:  $PR\Sigma$ 

(3) for <u>automatic</u> printing (and viewing) of the <u>non-zero</u> contents of registers bbb through eee, <u>incremented by ii</u>, use

PPC: BV (X:bbb.eeeii)<sup>6/</sup>

d. To copy data from main memory to the printer with all parameters prompted for, use the OS-41 program

"M2PD" (Main memory TO Printer, Data).

1-7. MAIN MEMORY to DISPLAY

The display is commonly assumed to be synonymous with the X register and, indeed, they frequently are one and the same. However, when using VIEW (or AVIEW) they are quite distinct. It can be said that when not VIEWing register nn (or Y, Z, T, or L) or AVIEWing, the display defaults to viewing the X register.

- a. To view individual main memory registers, use
  - (1) VIEW,
  - (2) VIEW IND, or
  - (3) VIEW IND (ST) (see footnote 5)
- b. To automatically view the non-zero contents of main memory registers bbb through eee, <u>incremented by ii</u>, use

PPC: BV (X:bbb.eeeii) (see footnote 6)

c. To copy data from main memory to the display with all parameters prompted for, use the OS-41 program

"M2DD" (<u>Main memory TO D</u>isplay, <u>D</u>ata).

2-1. EXTENDED MEMORY to MAIN MEMORY

a. Use XF: <u>GETR</u> (A:filename)

to recall (copy) an entire named file from the extended memory register 00 to the corresponding main memory registers. Copying stops upon reaching the file limit or main memory register SIZE-1.

<sup>6/</sup> Registers containing zero are automatically skipped, and setting flag 09 doubles the viewing pause, while setting flag 10 achieves a stop between registers viewed.

b. To recall a block of extended memory registers to main memory registers bbb through eee, use either

XF: <u>SEEKPTA</u> (X:rrr; A:filename), or

XF: SEEKPT (X:rrr)

to position the pointer at the starting extended memory register rrr, then use

XF: GETRX (X:bbb.eee)

c. To copy data from extended memory to main memory with all parameters prompted for, use the OS-41 program

"XM2MD" (eXtended Memory <u>TO</u> Main memory, <u>D</u>ata).

- 2-2. EXTENDED MEMORY to EXTENDED MEMORY
  - a. HP-41CX users can use the function

CX: RESZFL (working file; X: +nnn)

to change the size of the working extended memory ASCII file to nnn. If nnn is positive <u>and</u> a reduction in file size will erase non-zero data, "FL SIZE ERR" will be displayed and the file size won't change. This protective feature can be deactivated by specifying a negative nnn.

b. To increase or decrease the size of an extended memory file, or to copy data from one block of extended memory registers to another block within the same file or another file, with all parameters prompted for, use the OS-41 program

"XM2XMD" (eXtended Memory <u>TO</u> eXtended Memory, <u>D</u>ata).

## 2-3. EXTENDED MEMORY to MASS STORAGE

a. To copy the entire contents of extended memory (program, data, and ASCII files) to a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"XM2MSE" (eXtended Memory <u>TO</u> Mass <u>Storage</u>, <u>Entire</u>).

Refer to Chapter 4 for details.

b. To copy data from extended memory to mass storage with all parameters prompted for, use the OS-41 program

"XM2MSD" (eXtended Memory <u>IO</u> Mass <u>Storage</u>, <u>D</u>ata).

- 2-4. EXTENDED MEMORY to MAGNETIC CARDS
  - a. To copy data from extended memory to magnetic cards with all parameters prompted for, use the OS-41 program

"XM2CD" (eXtended Memory <u>TO</u> magnetic <u>Cards</u>, <u>Data</u>).

- 2-5. EXTENDED MEMORY to X REGISTER
  - a. To recall to the X register the contents of register rrr of the named extended memory file, first use either

XF: <u>SEEKPTA</u> (X:rrr; A:filename) or

XF: SEEKPT (X:rrr)

to position the pointer at extended memory register rrr, then use

XF: GETX

After each execution of GETX, the pointer is advanced to the next register in the working file.

- b. To facilitate the recall process, use the program "RCEX" which makes much of the detail transparent to the user. "RCEX" operates just like RCL except that the register recalled is in the current working file of extended memory. The author of "RCEX" (Hill, 1983) suggests assigning the program to the [RCL] key for convenience. For more details, see the Comments section of OS-41 program "XM2XD".
- c. To copy data from extended memory to the X register with all parameters prompted for, use the OS-41 program

"XM2XD" (eXtended Memory TO X register, Data).

## 2-6. EXTENDED MEMORY to PRINTER

a. To copy data from extended memory to the printer with all parameters prompted for, use the OS-41 program

"XM2PD" (eXtended Memory TO Printer, Data).

## 2-7. EXTENDED MEMORY to DISPLAY

a. To copy data from extended memory to the display with all parameters prompted for, use the OS-41 program

"XM2DD" (eXtended Memory <u>TO D</u>isplay, <u>D</u>ata).

#### 3-1. MASS STORAGE to MAIN MEMORY

a. Use IL: <u>READR</u> (A:filename)

to recall (copy) the entire named file from mass storage register 00 to the corresponding main memory registers. Copying stops upon reaching the file limit or main memory register SIZE-1.

b. To recall a block of mass storage registers to main memory registers bbb through eee, first use

IL: <u>SEEKR</u> (X:rrr; A:filename)

to position the pointer at the starting mass storage register rrr, then use

IL: READRX (X:bbb.eee)

c. Use IL: READA (A:filename)

to recall all main memory information (including all data registers and programs) from the specified mass storage "write all" file. The previous contents of main memory will be lost.

d. To copy data from mass storage to main memory with all parameters prompted for, use the OS-41 program

"MS2MD" (Mass Storage <u>TO Main memory</u>, Data).

- 3-2. MASS STORAGE to EXTENDED MEMORY
  - a. To recall the entire contents of extended memory (program, data, and ASCII files) from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2XME" (Mass Storage <u>IO</u> eXtended Memory, Entire).

The previous contents of extended memory will be lost. Refer to Chapter 4 for details.

b. To copy data from mass storage to extended memory with all parameters prompted for, use the OS-41 program

"MS2XMD" (Mass Storage <u>IO</u> eXtended Memory, <u>D</u>ata).

## 3-3. MASS STORAGE to MASS STORAGE

a. Use IL: RENAME (A:old filename, new filename)

to change the name of a data file in mass storage.

b. To copy a mass storage file from the primary device to the one having the indicated IL-address, use

XIO: COPYFL (X:IL address to copy to; A:filename)

- c. For procedures to duplicate an <u>entire</u> mass storage medium, refer to Chapter 4.
- d. To increase or decrease the size of a mass storage file, or to copy data from one block of mass storage registers to another block within the same file or another file, with all parameters prompted for, use the OS-41 program

"MS2MSD" (<u>Mass Storage IO Mass Storage</u>, <u>Data</u>).

- 3-4. MASS STORAGE to MAGNETIC CARDS
  - a. To copy data from mass storage to magnetic cards with all parameters prompted for, use the OS-41 program

"MS2CD" (<u>Mass Storage TO</u> magnetic <u>Cards</u>, <u>D</u>ata).

- 3-5. MASS STORAGE to X REGISTER
  - a. To copy data from mass storage to the X register with all parameters prompted for, use the OS-41 program

"MS2XD" (<u>Mass Storage TO X</u> register, <u>D</u>ata).

## 3-6. MASS STORAGE to PRINTER

a. To copy data from mass storage to the printer with all parameters prompted for, use the OS-41 program

"MS2PD" (<u>Mass Storage IO Printer, D</u>ata).

- 3-7. MASS STORAGE to DISPLAY
  - a. To copy data from mass storage to the display with all parameters prompted for, use the OS-41 program

"MS2DD" (<u>Mass Storage TO D</u>isplay, <u>D</u>ata).

#### 4-1. MAGNETIC CARDS to MAIN MEMORY

a. Use CR: RDTA (or simply insert a prerecorded magnetic card)

to read one or more magnetic data cards into consecutive main memory registers starting with register 00.

b. Use CR: RDTAX (X:bbb.eee)

to read one or more data cards into main memory registers bbb through eee.

- c. To recall all main memory information (including all data registers and programs) from a set of "write all" magnetic cards, insert the cards into the card reader with the HP-41 out of PRGM mode. The previous contents of main memory will be lost.
- d. To copy data from magnetic cards to main memory with all parameters prompted for, use the OS-41 program

"C2MD" (magnetic <u>Cards TO Main memory</u>, <u>Data</u>).

#### 4-2. MAGNETIC CARDS to EXTENDED MEMORY

a. To copy data from magnetic cards to extended memory with all parameters prompted for, use the OS-41 program

"C2XMD" (magnetic <u>Cards TO</u> e<u>X</u>tended <u>Memory</u>, <u>D</u>ata).

- 4-3. MAGNETIC CARDS to MASS STORAGE
  - a. To copy data from magnetic cards to mass storage with all parameters prompted for, use the OS-41 program

"C2MSD" (magnetic <u>Cards TO Mass Storage</u>, <u>Data</u>).

- 4-4. MAGNETIC CARDS to MAGNETIC CARDS
  - a. To copy data from one set of magnetic cards to another with all parameters prompted for, use the OS-41 program

"C2CD" (magnetic <u>Cards IO</u> magnetic <u>Cards</u>, <u>Data</u>).

## 4-5. MAGNETIC CARDS to X REGISTER

a. Movement of data in this fashion makes little sense, because there is room for 16 data registers on each side of a card. However, if one wants to copy data from magnetic cards to X, first use the OS-41 program

"C2MD" (magnetic <u>Cards TO Main memory</u>, <u>Data</u>)

to copy the cards to main memory, then use the OS-41 program

"M2XD" (<u>Main memory IO X</u> register, <u>D</u>ata)

to recall the desired data value from main memory.

## 4-6. MAGNETIC CARDS to PRINTER

a. To copy data from magnetic cards to the printer with all parameters prompted for, use the OS-41 program

"C2PD" (magnetic <u>Cards TO Printer</u>, <u>Data</u>).

4-7. MAGNETIC CARDS to DISPLAY

a. To copy data from magnetic cards to the display with all parameters prompted for, use the OS-41 program

"C2DD" (magnetic <u>Cards TO Display</u>, <u>Data</u>).

- 5-1. X REGISTER to MAIN MEMORY
  - To store (copy) X-register data into individual main memory registers, use
    - (1) STO,
    - (2) STO IND, or
    - (3) STO IND (ST) (see footnote 5)
  - b. Use the <u>Stack</u> to <u>Memory</u> program

PPC: SM (R6:nnn)

to store the contents of the X, Y, Z, T, and L registers, respectively, to the five consecutive registers beginning with register nnn, the number presently stored in register 6.

c. The PPC ROM routine, BI (<u>B</u>lock <u>I</u>ncrement), can be used to advantage to store regularly "spaced" data (such as 0, 5, 10, 15, ...) in designated main memory registers:

PPC: BI (X:III; Y:BBB; Z:bbb.eeeii),

where	bbb	Ξ	beginning main memory register,				
	eee	=	ending main memory register,				
	ii	=	register increment value,				
	BBB	=	initial numeric value to be stored,				
and	III	=	amount added to the preceeding stored value; III can				
			be positive or negative.				

NOTE: if III = 0, BI loads a constant (BBB) into all registers in the block designated by bbb.eeeii, and if III = 0 and BBB = 0, BI becomes a <u>block clear</u> operation.

- d. The utility program "STOD" facilitates the storing of data from the X register to main memory registers <u>with addresses over 99</u>. This program is an adaptation of one called "STOL" written by Lafoiet (1982). It is recommended that "STOD" be assigned to the [STO] or the shift [STO] key for convenience. For more details, see the Comments section of OS-41 program "X2MD".
- e. To copy data from the X register repeatedly to main memory with all parameters prompted for, use the OS-41 program

"X2MD" (X register <u>TO Main memory</u>, <u>Data</u>).

## 5-2. X REGISTER to EXTENDED MEMORY

a. Use XF: <u>CRFLD</u> (X:nnn; A:filename)

to create a file of nnn registers with the indicated file name, then use

XF: <u>SEEKPTA</u> (X:rrr; A:filename), or

XF: SEEKPT (X:rrr)

to move the pointer to register rrr of the named (or working) extended memory file, followed by

XF: SAVEX

to copy the contents of the X register to register rrr in the working file.

After each execution of SAVEX, the pointer is advanced to the next register in the working file.

- b. To facilitate the storing process from the X register to extended memory, use the program "STEX", which operates much like STO, except that the X register contents are copied into the designated register of the current working file of extended memory. The author (Hill, 1983) suggests assigning the program to the [STO] key for convenience. For more details, see the Comments section of OS-41 program "X2XMD".
- c. To copy data from the X register repeatedly to extended memory with all parameters prompted for, use the OS-41 program

"X2XMD" (X register <u>TO</u> eXtended <u>Memory</u>, <u>D</u>ata).

Note that a "block" of registers can be one register only, therefore with proper input, "X2XMD" becomes a prompting version of "STEX".

## 5-3. X REGISTER to MASS STORAGE

a. To copy data from the X register repeatedly to mass storage with all parameters prompted for, use the OS-41 program

"X2MSD" (X register <u>TO Mass S</u>torage, <u>Data</u>).

## 5-4. X REGISTER to MAGNETIC CARDS

a. Movement of data in this fashion makes little sense, because there is room for 16 data registers on each side of a card. However, if one wants to copy data from the X register to magnetic cards, first use the OS-41 program

"X2MD" (X register <u>TO</u> Main memory, <u>D</u>ata)

"M2CD" (<u>Main memory TO</u> magnetic <u>Cards</u>, <u>D</u>ata)

to copy the data to cards.

## 5-5. X REGISTER to X REGISTER

- a. While it may seem trivial, it is occasionally appropriate to exchange the contents of X with itself; for example, whenever a "do nothing" (no-op) or a time-consuming operation is needed. The following calculator functions are acceptable no-ops; faster no-ops can be obtained using synthetic programming (see Jarett, 1982):
  - (1) X<>X
  - (2) STO (ST) X<sup>1</sup>/
- b. Various basic calculator functions facilitate the movement of data within the stack; the following are cited for completeness, and to emphasize certain features.
  - (1) ENTER ↑ puts a copy of the X register into the Y register and <u>disables</u> the stack lift; i.e., the stack will not lift when a number is now keyed-in or recalled.
  - (2)  $R \downarrow$  (the key) or RDN (the function) rotate (or "roll") the stack downward; Y replaces X, X replaces T, etc.
  - (3)  $R^{\uparrow}$  rotates the stack upward; X replaces Y, Y replaces Z, etc.
  - (4) X<>Y (the key) or X<>(ST) (the function) exchange the contents of X and Y or X and either X, Y, Z, T, or L, depending upon the response to the cursor.
  - (5) STO (ST) (see footnote 7) stores a copy of the X register into X, Y, Z, T, or L, depending upon the cursor response.
  - (6) RCL (ST) (see footnote 7) recalls a copy of the X, Y, Z, T, or L registers to the X register, depending upon the response to the cursor.
  - (7) LASTX moves a copy of the L register into the X register. Note that, similar to other "recall" functions, LASTX causes the stack to lift, unless of course either ENTER<sup>↑</sup> or CLX were just previously executed.

 $<sup>\</sup>mathcal{U}$ In such "stack" functions, the "ST" is achieved by pressing the decimal point key, and the X is obtained by pressing the X key (<u>not</u> the multiply key, and <u>not</u> in ALPHA mode).

## 5-6. X REGISTER to PRINTER

- a. The contents of the X register can be printed by either of the following printer functions:
  - (1) P & IL: PRX

to print contents of X register only, or

(2) P & IL: PRSTK

to print X, Y, Z, and T registers.

- b. The contents of X can be appended to the print buffer (in preparation to print) by
  - P & IL: ACX,

then printed<sup>8/</sup> by

P & IL: PRBUF (for left-justified output)
or P & IL: ADV (for right-justified output)

c. No OS-41 program is appropriate here since the above functions adequately copy data from the X register to the printer.

## 5-7. X REGISTER to DISPLAY

a. Use VIEW (ST) (see footnote 7) to view X, Y, Z, T, or L registers, depending upon the response to the cursor.

As in all VIEWing operations, the <u>stack does not lift</u>, the X register is not disturbed, and the display can be cleared by pressing [<-] <u>once</u>, after which the contents of the X register fill the display.

b. No OS-41 program is appropriate here since the VIEW (ST) function adequately views the X, Y, Z, T, or L registers.

## 6-1. BARCODE to MAIN MEMORY

a. Use W: WNDDTX (X:bbb.eee)

to copy data barcode into main memory registers bbb through eee.

 $<sup>\</sup>frac{8}{}$  The HP 82163A Video Interface automatically prints X upon execution of the ACX function.

b. To copy data from barcode to main memory with all parameters prompted for, use the OS-41 program

"B2MD" (Barcode TO Main memory, Data).

- 6-2. BARCODE to EXTENDED MEMORY
  - a. To copy data from barcode to extended memory with all parameters prompted for, use the OS-41 program

"B2XMD" (<u>Barcode TO eXtended Memory</u>, <u>D</u>ata).

- 6-3. BARCODE to MASS STORAGE
  - a. To copy data from barcode to mass storage with all parameters prompted for, use the OS-41 program

"B2MSD" (<u>B</u>arcode <u>TO</u> <u>M</u>ass <u>S</u>torage, <u>D</u>ata).

## 6-4. BARCODE to MAGNETIC CARDS

a. Movement of data in this fashion makes little sense, because there is room for 16 data registers on each side of a card. However, if one wants to copy data from barcode to magnetic cards, first use the OS-41 program

"B2MD" (<u>Barcode TO Main memory</u>, <u>D</u>ata)

to store the value(s) in main memory, then use the OS-41 program

"M2CD" (<u>Main memory TO</u> magnetic <u>Cards</u>, <u>D</u>ata)

to copy the data to cards.

#### 6-5. BARCODE to X REGISTER

a. Use W: WNDDTA (or simply scan the data barcode)

to copy data from barcode to the X register. Scanning two or more rows of data barcode loads the stack in the same way as keying in two or more numbers.

b. No OS-41 program is appropriate here since the above operation adequately copies data barcode to the X register.

### 6-6. BARCODE to PRINTER

a. To copy data from barcode to the printer with all parameters prompted for, use the OS-41 program

"B2PD" (<u>B</u>arcode <u>TO</u> <u>P</u>rinter, <u>D</u>ata).

## 6-7. BARCODE to DISPLAY

a. To copy data from barcode to the display with all parameters prompted for, use the OS-41 program

,

"B2DD" (<u>B</u>arcode <u>TO D</u>isplay, <u>D</u>ata).

## USE THIS PAGE FOR NOTES

## C. <u>Numeric and Alphanumeric Data Movement Program Instructions</u>

The program instructions found on the following pages should enable the user to apply the desired program. One or more examples are provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 2-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings and an explanation of the data movement algorithm used extensively in these programs. M2MD (<u>Main memory TO Main memory</u>, <u>Data</u>)

<u>Program No.</u>: 41D1-1 <u>Other Programs in This File</u>: none

## Purpose

To copy data from one block of main memory registers to another with all parameters prompted for.

Minimum SIZE: (ending main memory register moved to or from) + 1

Required Peripherals: none

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2MD and follow the prompts:
  - a. SRC bbb.eee=?

Key in the source beginning (bbb) and ending (eee) main memory registers and press [R/S]. If eee is less than bbb, only register bbb will be copied.

b. DST bbb=?

Key in the destination beginning register bbb to copy the source beginning main memory register to and press [R/S]. To default to register 00, just press [R/S].

3. The HP-41 will display "\*WORKING\*" as data are copied from the source block of main memory registers to the destination block. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Copy main memory registers 10 through 15 to registers 20 through 25.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. 2. a. b. 3.	SRC bbb.eee=? DST bbb=?	[XEQ]"SIZE"026 [XEQ]"M2MD" 10.015 [R/S] 20 [R/S]	*WORKING* DONE

2. Copy main memory register 10 to register 00.

<u>Step</u>	Prompt	User Input	Display
1. 2. a. b. 3.	SRC bbb.eee=? DST bbb=?	[XEQ]"SIZE"011 [XEQ]"M2MD" 10 [R/S] [R/S]	0.00 *WORKING* DONE

## <u>Comments</u>

- 1. M2MD is a prompting version of the PPC ROM routine BM.
- 2. If the two blocks of main memory registers specified overlap (i.e., SRC bbb  $\leq$  DST bbb  $\leq$  SRC eee), the overlapping source registers will first be overstored then copied to their corresponding destination registers.

M2XMD (<u>Main memory <u>TO</u> e<u>X</u>tended <u>Memory</u>, <u>D</u>ata)</u>

Program No.: 41D1-2 Other Programs in This File: M2MSD

## Purpose

To copy data from main memory to extended memory with all parameters prompted for. M2XMD checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: (ending main memory register) + 1

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2XMD and follow the prompts:
  - a. MM bbb.eee=?

Key in the beginning (bbb) and ending (eee) main memory registers to be copied to extended memory and press [R/S]. If eee is less than bbb, only register bbb will be copied.

b. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy main memory registers bbb through eee to and press [R/S].

c. XM bbb=?

Key in the beginning extended memory register bbb to copy the beginning main memory register to and press [R/S]. To default to the beginning of the file (register 00), just press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named extended memory file is created (if necessary) and data are copied from main memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Copy the contents of main memory registers 10 through 15 to extended memory file "D1-2A" (which already exists) registers 00 through 05.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use X2XMD to create and load extended memory file "D1-2A" with size 6 or more.

1.		[XEQ]"SIZE"016	
2.		[XEQ]"M2XMD"	
a.	MM bbb.eee=?	10.015 [R/S]	
b.	XM FILE?	D1-2A [R/S]	
с.	XM bbb=?	[R/S]	0.00
4.			*WORKING*
			DONE

2. Copy the contents of main memory register 10 to the same register in extended memory file "D1-2B" (which doesn't exist).

<u>Step</u>	Prompt	<u>User Input</u>	Display
1. 2.		[XEQ]"SIZE"011 [XEQ]"M2XMD"	
а.	MM bbb.eee=?	10 LR/SJ	
b.	XM FILE?	D1-2B [R/S]	
с.	XM bbb=?	10 [R/S]	
3.			*WORKING*
a.	XM D1-2B SIZE=?	11 [R/S]	
4.			*WORKING* DONE

### Comments

- As Example 2 demonstrates, the minimum size required for an extended memory file = (XM bbb) + (number of registers to copy).
- 2. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case one can resolve the difficulty and restart the program.

M2MSD (<u>Main memory TO Mass Storage</u>, <u>D</u>ata)

Program No.: 41D1-3 Other Programs in This File: M2XMD

## Purpose

To copy data from main memory to mass storage with all parameters prompted for. M2MSD checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (ending main memory register) + 1

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2MSD and follow the prompts:
  - a. MM bbb.eee=?

Key in the beginning (bbb) and ending (eee) main memory registers to be copied to mass storage and press [R/S]. If eee is less than bbb, only register bbb will be copied.

b. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy main memory registers bbb through eee to and press [R/S].

c. MS bbb=?

Key in the beginning mass storage register bbb to copy the beginning main memory register to and press [R/S]. To default to the beginning of the file (register 00), just press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage file to be created and press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named mass storage file is created (if necessary) and data are copied from main memory to mass storage. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

1. Copy the contents of main memory registers 10 through 15 to mass storage file "D1-3A" (which already exists) registers 00 through 05.

<u>Step \_\_\_\_Prompt \_\_\_\_User\_Input \_\_\_\_Display\_\_</u>

preliminary - Use X2MSD to create and load mass storage file "D1-3A" with size 6 or more.

1.		[XEQ]"SIZE"016	
2.		[XEQ]"M2MSD"	
a.	MM bbb.eee=?	10.015 [R/S]	
b.	MS FILE?	D1-3A [R/S]	
с.	MS bbb=?	[R/S]	0.00
4.			*WORKING*
			DONE

 Copy the contents of main memory register 10 to the same register in mass storage file "D1-3B" (which doesn't exist).

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. 2.		[XEQ]"SIZE"011 [XEQ]"M2MSD"	
a.	MM bbb.eee=?	10 [R/S]	
b.	MS FILE?	D1-3B [R/S]	
с.	MS bbb=?	10 [R/S]	
3.			*WORKING*
a.	MS D1-3B SIZE=3	? 11 [R/S]	
4.			*WORKING*
			DONE

## Comments

- As Example 2 demonstrates, the minimum size required for a mass storage file = (MS bbb) + (number of registers to copy).
- 2. Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file SIZE=?</u>" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case one can resolve the difficulty and restart the program.

M2CD (Main memory <u>TO</u> magnetic <u>Cards</u>, <u>Data</u>)

Program No.: 41D1-4 Other Programs in This File: C2MD

## Purpose

To copy data from main memory to magnetic cards with all parameters prompted for.

Minimum SIZE: (ending main memory register) + 1

## Required Peripherals

Card Reader (HP 82104A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2CD and follow the prompts:
  - a. MM bbb.eee=?

Key in the beginning (bbb) and ending (eee) main memory registers to copy to magnetic cards and press [R/S]. If eee is less than bbb, only register bbb will be copied.

b. INSERT DST

Insert track 1 of destination magnetic card 1 into the Card Reader.

3. If more than 16 main memory registers are being copied, follow the prompt:

a. RDY tt of nn

where

 $\underline{tt}$  = next track to be recorded, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

- 4. When all the destination magnetic cards have been recorded, the HP-41 will beep and prompt:
  - a. "VER"IFY CDS1/

Press [XEQ] "VER".

Here using the CR: VER function, refer to Section D for an explanation of the VER bug that can alter extended memory.

b. CARD

Insert each track recorded into the Card Reader. If the track was recorded properly, the HP-41 will display "TYPE D TR <u>tt</u>",

where

 $\underline{tt}$  = track number.

If the HP-41 displays "CARD ERR" or "CHECKSUM ERR", press [<-] and start over at Step 2, perhaps with new magnetic cards.

5. When all the destination magnetic cards have been verified, press [<-]. The HP-41 will display "DONE".

## Examples

1. Copy main memory registers 10 through 15 to a magnetic card.

<u>Step</u>	Prompt	User Input	<u>Display</u>
1. 2. b. 4a. b.	MM bbb.eee=? INSERT DST "VER"IFY CDS CARD CARD	[XEQ]"SIZE"016 [XEQ]"M2CD" 10.015 [R/S] (Insert track 1) [XEQ]"VER" (Insert track 1)	TYPE D TR 01
5.		[<-]	DONE

2. Copy main memory registers 10 through 50 to magnetic cards.

<u>Step</u>	Prompt	<u>User Input</u>	Display
1.		[XEQ]"SIZE"051	
2.		[XEQ]"M2CD"	
a.	MM bbb.eee=?	10.05 [R/S]	
b.	INSERT DST	(Insert track 1)	
3a.	RDY 02 OF 03	(Insert tract 2)	
	RDY 03 OF 03	(Insert track 3)	
4a.	"VER"IFY CDS	[XEQ]"VER"	
b.	CARD	(Insert track 1)	TYPE D TR 01
	CARD	(Insert track 2)	TYPE D TR 02
	CARD	(Insert track 3)	TYPE D TR 03
	CARD		
5.		[<-]	DONE

### Comments

- The destination magnetic cards may be inserted in any order in Steps 2, 3, and 4; the HP-41 automatically keeps track of where data reside on the cards.
- Any unused registers on a magnetic card track may be envisioned as filled with "null values" that won't overstore main memory registers in subsequent magnetic card to main memory operations.

M2XD (<u>Main memory TO X</u> register, <u>Data</u>)

Program No.: 41D1-5 Other Programs in This File: none

## <u>Purpose</u>

To copy data from main memory to the X register with all parameters prompted for. M2XD is convenient to use for main memory registers over 99, which aren't directly accessible by the RCL function.

Minimum SIZE: (target main memory register) + 1

### Required Peripherals: none

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2XD and follow the prompt:
  - a. MM RRR=?

Key in the main memory register RRR to be copied to X and press [R/S].

3. The HP-41 will copy the specified main memory register to X, beep, stop, and display the X register.

#### Examples

1. Copy the contents of main memory register 100 to the X register.

Step	Prompt	User Input	Display	
1. 2.		[XEQ]"SIZE"101 [XEQ]"M2XD"		
а. 3.	MM RRR=?	100 LR/SJ	x.xx (contents of	FX)

#### Comments

1. RCLD, a copy of which is shown below, can be used as an alternative to the M2XD program. When RCLD is assigned to the [RCL] key, a prompt of the form RCL\_\_\_ is obtained by pressing [RCL] in USER mode, enabling the recall of any main memory register. Note that three digits need not be provided in all cases. For example, to recall register 10 to the X register one would simply key 1 and 0; to recall register 08, key only 8; but to recall register 100, one must key 1, 0, and 0. As a further precaution, note that there is a programmed one-second pause between digits, therefore successive digits after the first must be supplied promptly to avoid erroneous results.

# 

ROW 5: LINES 17 - 22

ROW 4: LINES 9 - 16

# ROW 3: LINES 7 - 9

ROW 2: LINES 2 - 7

ROW 1: LINES 1 - 2

Program Registers Required: 9

BARCODE FOR RCLD, STOD

01+LBL "RCLD" 02 "RCL\_\_\_" 03 XEQ 00 04 X(> L **95 RCL IND L** 06 RTN 97+LBL \*STOD\* 08 •ST0\_\_\_-09 X(> L 10 RDN 11 XEQ 00 12 X(> L 13 STO IND L 14 .RTN 15+LBL 00 16 CF 22 17 AVIEW 18+LBL 01 19 PSE 20 FC? 22 21 GTO 01 22 END

Program Listing for RCLD:

Program Registers: 9

Bytes: 62

M2PD (Main memory <u>IO</u> Printer, Data)

<u>Program No.</u>: 41D1-6 <u>Other Programs in This File</u>: M2DD

## Purpose

To copy data from main memory to the printer with all parameters prompted for. M2PD can print one or more consecutive registers or registers separated by an increment value. If the printer is not connected, M2PD operates like M2DD.

Minimum SIZE: (ending main memory register printed) + 1

## Required Peripherals

Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 3. Execute M2PD and follow the prompts:
  - a. MMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) main memory registers to print and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be printed. If II is not specified, it will default to 1.

b. # DECIMALS=?

Key in the number of decimal places to be printed and press [R/S]. To default to 2 decimal places, just press [R/S].

4. The HP-41 will display "\*WORKING\*" and print each main memory register in the form

 $R_{XX} = yyy_yy$ 

where

 $\underline{xx}$  = register number being printed, and  $\underline{yyy},\underline{yy}$  = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.)

5. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Print the contents of main memory registers 10, 13, and 16 accurate to three decimal places.
 Printer Output

<u>Step</u>	Prompt	User Input	<u>Display</u>	(HP 82162A)
1. 2. 3. a. b.	(Set MMbbb.eeeII=? # DECIMALS=?	[XEQ]"SIZE"017 printer to MAN [XEQ]"M2PD" 10.01603 [R/S] 3 [R/S]	mode)	
4.			*WORKING*	R10= y.yyy R13= y.yyy R16= y.yyy
5.			DONE	

 Print the contents of main memory register 10 accurate to two decimal places.
 Printer Output

<u>Step</u>	Prompt	User Input	Display	(HP 82162A)
1. 2. 3. a. b. 4.	(Set MMbbb.eeeII=? # DECIMALS=?	[XEQ]"SIZE"011 printer to MAN [XEQ]"M2PD" 10 [R/S] [R/S]	mode) 2.00 *WORKING*	
5.			DONE	R10= y.yy

M2DD (<u>Main memory TO D</u>isplay, <u>D</u>ata)

Program No.: 41D1-7 Other Programs in This File: M2PD

## Purpose

To copy data from main memory to the display with all parameters prompted for. M2DD can view one or more consecutive registers or registers separated by an increment value. If the printer is connected, M2DD operates like M2PD.

Minimum SIZE: (ending main memory register viewed) + 1

## Required Peripherals: none

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2DD and follow the prompts:
  - a. MMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) main memory registers to view and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be viewed. If II is not specified, it will default to 1.

b. # DECIMALS=?

Key in the number of decimal places to be viewed and press [R/S]. To default to 2 decimal places, just press [R/S].

3. The HP-41 will display "\*WORKING\*", then display each main memory register in the form

 $R_{XX} = yyy, yy,$ 

where

 $\underline{xx}$  = register number being viewed, and  $\underline{yyy}$  = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.)

4. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

1. View the contents of main memory registers 10, 13, and 16 accurate to three decimal places.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. 2. b. 3.	MMbbb.eeeII=? # DECIMALS=?	[XEQ]"SIZE"017 [XEQ]"M2DD" 10.01603 [R/S] 3 [R/S]	*WORKING* R10 = <u>yyy.yyy</u> R13 = <u>yyy.yyy</u> R16 = <u>yyy.yyy</u> DONE

2. View the contents of main memory register 10 accurate to two decimal places.

<u>Step</u>	Prompt	User Input	<u>Display</u>
1. 2. a. M b. # 3.	Mbbb.eeeII=? ? DECIMALS=?	[XEQ]"SIZE"011 [XEQ]"M2DD" 10 [R/S] [R/S]	2.00 *WORKING* R10 = <u>yyy.yy</u> DONE

## <u>Comments</u>

 To stop the HP-41 after each display, set flag 00 before executing M2DD. Press [R/S] after each display to continue. XM2MD (eXtended Memory TO Main memory, Data)

Program No.: 41D2-1 Other Programs in This File: MS2MD

## <u>Purpose</u>

To copy data from extended memory to main memory with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory register) + (number of registers to copy)

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2MD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to main memory and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) or ending (eee) extended memory registers to be copied to main memory and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. MM bbb=?

Key in the beginning main memory register bbb to copy the beginning extended memory register to and press [R/S]. To default to register 00, just press [R/S].

 The HP-41 will display \*WORKING\* as data are copied from extended memory to main memory. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Copy the contents of extended memory file "D2-1A" registers 00 through 05 to main memory registers 10 through 15.

Prompt User Input Display <u>Step</u> preliminary - Use X2XMD to create and load extended memory file "D2-1A" with size 6 or more. 1. [XEQ]"SIZE"016 2. [XEQ]"XM2MD" a. XM FILE? D2-1A [R/S] XM bbb.eee=? b. .005 [R/S] c. MM bbb=? 10 [R/S] 3.

\*WORKING\* DONE XM2XMD (eXtended Memory <u>IO</u> eXtended Memory, <u>D</u>ata)

Program No.: 41D2-2 Other Programs in This File: MS2MSD

## Purpose

To increase or decrease the size of an extended memory file or to copy data from one block of extended memory registers to another block within the same file or another file, with all parameters prompted for. XM2XMD checks if the named destination extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: (ending main memory temporary storage register) + 1

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2XMD and follow the prompts:
  - a. XM SRC FILE?

Key in the name of the source extended memory file (six characters or less) and press [R/S].

b. SRC bbb.eee=?

Key in the source beginning (bbb) and ending (eee) extended memory registers and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. XM DST FILE?

Key in the name of the destination extended memory file (six characters or less) to copy source extended memory registers bbb through eee to and press [R/S]. To default to the name of the source extended memory file, just press [R/S].

d. DST bbb=?

Key in the destination beginning extended memory register to copy the source beginning extended memory register to and press [R/S]. To default to register 00, just press [R/S].

3. If the extended memory source and destination file names specified are the same, follow the prompt:

a. NEW SIZE=?

Key in the new size of the source extended memory file and press [R/S]. To default to the current file size, just press [R/S].

- 4. Follow the prompt:
  - a. TMP bbb.eee=?

Key in the beginning (bbb>7, or bbb>8 if changing file size) and ending (eee) main memory temporary storage registers to transfer data through and press [R/S]. If eee is less than bbb, only register bbb will be used.

- 5. The HP-41 will display "\*WORKING\*" as it checks if the named destination extended memory file exists. If it doesn't, follow the prompt:
  - a. DST <u>file</u> SIZE=?

where

<u>file</u> = destination extended memory file name.

Key in the size of the destination extended memory file to be created and press [R/S].

6. The HP-41 will display "\*WORKING\*" as the named destination extended memory file is created (if necessary) and data are copied from extended memory through main memory back to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Copy the contents of extended memory file "D2-2A" registers 01 through 05 to extended memory file "D2-2B" (which doesn't exist) registers 10 through 14. Use main memory registers 20 through 23 for temporary storage.

Step	Prompt	User Input	Display		
preliminary - Use X2XMD to create and load extended memory file "D2-2A" with size 6 or more.					
1.		[XEQ]"SIZE"024			
2.		[XEQ]"XM2XMD"			
a.	XM SRC FILE?	D2-2A [R/S]			
b.	SRC bbb.eee=?	1.005 [R/S]			
с.	XM DST FILE?	D2-2B [R/S]			
d.	DST bbb=?	10 [R/S]			
4a.	TMP bbb.eee=?	20.023 [R/S]			
5.			*WORKING*		
a.	DST D2-2B SIZE	=? 15 [R/S]			
6.			*WORKING*		
			DONE		

 Copy the contents of extended memory file "D2-2C" registers 00 through 05 to registers 10 through 15. Use main memory registers 10 through 20 for temporary storage.

<u>Step</u>	Prompt	User Input	_Display_
prelin	ninary - Use X2XM memory f	1D to create and File "D2-2C" with	load extended size 6 or more
1. 2. b. c. d. 3a. 4a. 5,6.	XM SRC FILE? SRC bbb.eee=? XM DST FILE? DST bbb=? NEW SIZE=? TMP bbb.eee=?	[XEQ]"SIZE"021 [XEQ]"XM2XMD" D2-2C [R/S] .005 [R/S] [R/S] 10 [R/S] 16 [R/S] 10.02 [R/S]	D2-2C *WORKING*

3. Decrease the size of extended memory file "D2-2D" from 20 registers to 10 registers while simultaneously moving<sup>1</sup>/ data from extended memory registers 15 through 19 to registers 05 through 09. Use main memory registers 10 through 15 for temporary storage.

Step Prompt User Input Display preliminary - Use X2XMD to create and load extended memory file "D2-2D" with size 20. 1. [XEQ]"SIZE"016 2. [XEQ]"XM2XMD" XM SRC FILE? a. D2-2D [R/S] b. SRC bbb.eee=? 15.019 [R/S] XM DST FILE? с. [R/S] D2-2D d. DST bbb=? 5 [R/S] 3a. NEW SIZE=? 10 [R/S]

> \*WORKING\* DONE

## Comments

4a.

5,6.

TMP bbb.eee=?

 The minimum size required for a destination extended memory file = (XM DST bbb) + (number of registers to copy).

10.015 [R/S]

2. Since flag 25 (error ignore flag) is used to detect if the named destination extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "DST <u>file SIZE=?</u>" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case one can resolve the difficulty and restart the program.

<sup>&</sup>lt;sup>1/</sup>The word "moving" instead of "copying" is appropriate here since the contents of the source extended memory registers will <u>not</u> be preserved.

- 3. As Example 2 demonstrates, it is not necessary to specify a number of main memory temporary storage registers less than or equal to the number of extended memory registers to copy. If fewer temporary storage registers are needed by XM2XMD, only those needed (in this example, 6) will be used.
- 4. When an extended memory file size is increased or decreased, XM2XMD temporarily creates a dummy file of the new size. Therefore, a number of extended memory registers equal to the new size must be available to increase or decrease a file size.
- 5. When an extended memory file size is increased or decreased, XM2XMD temporarily creates a dummy file named "...". If a file by this name already exists, XM2XMD will stop with an error message.

XM2MSD (eXtended Memory <u>TO</u> Mass <u>S</u>torage, <u>D</u>ata)

Program No.: 41D2-3 Other Programs in This File: MS2XMD

## <u>Purpose</u>

To copy data from extended memory to mass storage with all parameters prompted for. XM2MSD checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: (ending main memory temporary storage register) + 1

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2MSD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to mass storage and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) and ending (eee) extended memory registers to be copied to mass storage and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy extended memory registers bbb through eee to and press [R/S]. To default to the name of the extended memory file, just press [R/S].

d. MS bbb=?

Key in the beginning mass storage register bbb to copy the beginning extended memory register to and press [R/S]. To default to the beginning of the file (register 00), just press [R/S].

e. TMP bbb.eee=?

Key in the beginning (bbb>2) and ending (eee) main memory temporary storage registers to transfer data through and press [R/S]. If eee is less than bbb, only register bbb will be used.
- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage file to be created and press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named mass storage file is created (if necessary) and data are copied from extended memory through main memory to mass storage. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Copy the contents of extended memory file "D2-3A" registers 01 through 05 to mass storage file "D2-3A" (which already exists) registers 10 through 14. Use main memory registers 10 through 13 for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2XMD and X2MSD to create and load extended memory and mass storage files "D2-3A" with sizes 6 or more and 15 or more, respectively.

 1.
 [XEQ]"SIZE"014

 2.
 [XEQ]"XM2MSD"

a.	XM FILE?	D2-3A [R/S]	
b.	XM bbb.eee=?	1.005 [R/S]	
с.	MS FILE?	[R/S]	D2-3A
d.	MS bbb=?	10 [R/S]	
e.	TMP bbb.eee=?	10.013 [R/S]	
4.			*WORKING*
			DONE

 Copy the contents of extended memory file "D2-3B" register 05 to the beginning of mass storage file "D2-3C" (which doesn't exist). Use main memory registers 10 through 20 for temporary storage.

<u>Step</u>	Prompt	<u>User Input</u> <u>Display</u>
preli	iminary - Use X2 memory	XMD to create and load extended file "D2-3B" with size 6 or more
1.		[XEQ]"SIZE"021
2.		[XEQ]"XM2MSD"
a.	XM FILE?	D2-3B [R/S]

Step \_\_\_\_Prompt <u>User Input</u> Display d. MS bbb=? [R/S]0.00 TMP bbb.eee=? 10.02 [R/S] e. 3. \*WORKING\* MS D2-3C SIZE=? 1 [R/S] a. 4. \*WORKING\* DONE

## Comments

- The minimum size required for a mass storage file = (MS bbb) + (number of registers to copy).
- 2. Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file SIZE=?</u>" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.
- 3. As Example 2 demonstrates, it is not necessary to specify a number of temporary main memory registers less than or equal to the number of extended memory registers to copy. If fewer temporary storage registers are needed by XM2MSD, only those needed (in this example, only 1) will be used.

USE THIS PAGE FOR NOTES

XM2CD (eXtended Memory TO magnetic Cards, Data)

Program No.: 41D2-4 Other Programs in This File: MS2CD, C2XMD, C2MSD

## Purpose

To copy data from extended memory to magnetic cards with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers to copy)

### Required Peripherals

Extended Functions/Memory Module (HP 82180A) Card Reader (HP 82104A)

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2CD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to magnetic cards and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) and ending (eee) extended memory registers to be copied to magnetic cards and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through ((bbb + number of registers to copy) - 1) will be used.

- 3. The HP-41 will display "\*WORKING\*", then prompt:
  - a. INSERT DST

Insert track 1 of destination magnetic card 1 into the Card Reader.

- 4. If more than 16 extended memory registers are being copied, follow the prompt:
  - a. RDY tt of nn

where

 $\frac{tt}{n}$  = next track to be recorded, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

- 5. When all the destination magnetic cards have been recorded, the HP-41 will beep and prompt:
  - a. "VER"IFY CDS1/

Press [XEQ] "VER".

b. CARD

Insert each track recorded into the Card Reader. If the track was recorded properly, the HP-41 will display "TYPE D TR  $\underline{tt}$ ",

where

 $\underline{tt}$  = track number.

If the HP-41 displays "CARD ERR" or "CHECKSUM ERR", press [<-] and start over at Step 2, perhaps with new magnetic cards.

 When all the destination magnetic cards have been verified, press [<-]. The HP-41 will display "DONE".</li>

## Examples

 Copy the contents of extended memory file "D2-4A" registers 05 through 09 to a magnetic card. Use main memory registers 10 through 14 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2XMD to create and load extended memory file "D2-4A" with size 10 or more.

1.		[XEQ]"SIZE"015	
2.		[XEQ]"XM2CD"	
a.	XM FILE?	D2-4A [R/S]	
b.	XM bbb.eee=?	5.009 [R/S]	
с.	TMP bbb=?	10 [R/S]	
3.			*WORKING*
a.	INSERT DST	(Insert track 1)	
5a.	"VER"IFY CDS	[XEQ]"VER"	
b.	CARD	(Insert track 1)	TYPE D TR 01
	CARD		
6.		[<-]	DONE

<sup>1</sup>/Before using the CR: VER function, refer to Section D for an explanation of the VER bug that can alter extended memory.

 Copy the contents of extended memory file "D2-4B" registers 00 through 39 to magnetic cards. Use main memory registers 05 through 45 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2XMD to create and load extended memory file "D2-4B" with size 40 or more.

1.		[XEQ]"SIZE"046	
2.		LXEQJ"XM2CD"	
a.	XM FILE?	D2-4B [R/S]	
b.	XM bbb.eee=?	.039 [R/S]	
с.	TMP bbb=?	5 [R/S]	
3.			*WORKING*
a.	INSERT DST	(Insert track 1)	
4a.	RDY 02 OF 03	(Insert track 2)	
	RDY 03 OF 03	(Insert track 3)	
5a.	"VER"IFY CDS	[XEQ]"VER"	
b.	CARD	(Insert track 1)	TYPE D TR 01
	CARD	(Insert track 2)	TYPE D TR 02
	CARD	(Insert track 3)	TYPE D TR 03
	CARD		
6.		[<-]	DONE

### Comments

- The destination magnetic cards may be inserted in any order in Steps 3, 4, and 5; the HP-41 automatically keeps track of where data reside on the cards.
- 2. Any unused registers on a magnetic card track may be envisioned as filled with "null values" that won't overstore main memory registers in subsequent magnetic card to main memory operations.

# USE THIS PAGE FOR NOTES

XM2XD (eXtended Memory <u>TO X</u> register, <u>D</u>ata)

Program No.: 41D2-5 Other Programs in This File: MS2XD

## Purpose

To copy data from extended memory to the X register with all parameters prompted for.

Minimum SIZE: 002

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2XD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to X and press [R/S].

b. XM RRR=?

Key in the extended memory register RRR to be copied to X and press [R/S].

٠

3. The HP-41 will display "\*WORKING\*" as the specified extended memory register is copied to X. When completed, the HP-41 will beep, stop, and display the X register.

#### <u>Examples</u>

1. Copy the contents of extended memory file "D2-5A" register 05 to the X register.

<u>Step</u>	Prompt	User Input	<u>Display</u>
preli	minary - Use X2 memory Store	2XMD to create and / file "D2-5A" with l in register 05.	load extended size 6 or more
1. 2. a. b. 3.	XM FILE? XM RRR=?	[XEQ]"SIZE"002 [XEQ]"XM2XD" D2-5A [R/S] 5 [R/S]	*WORKING*
			1.00

## Comments

1. RCEX, a copy of which is shown below, can be used as an alternative to the XM2XD program. When RCEX is assigned to the [RCL] key, a prompt of the form RCEX\_\_\_\_\_\_ is obtained by pressing [RCL] in USER mode, enabling the recall of any existing register <u>in the working file</u> of extended memory. Note that the prompt need not be supplied with three digits in all cases. For example, to recall extended memory register 10 to the X register one would simply key 1 and 0; to recall register 08, key only 8; but to recall register 100, one must key 1, 0, and 0. As a further precaution, note that there is a programmed one-second pause between digits, therefore successive digits after the first must be supplied promptly to avoid erroneous results.

Program Listing for RCEX:

Program Registers: 10 Bytes: 67

13 SEEKPT 01+LBL "RCEX" 14 X(> L 02 "RCEX\_\_\_\_" 15 SAVEX A3 XE0 00 16 RTN 04 SEEKPT 17+LBL 00 05 CLX 18 CF 22 06 GETX 19 AVIEW 07 RTN 20+LBL 01 AS♦LBL "STEX" 21 PSE 09 X(> L 22 FC? 22 10 RIN 23 GTO 01 11 •STEX\_\_\_\_\* 24 END 12 XEQ 00

BARCODE FOR RCEX, STEX

Program Registers Required: 10

ROW 1: LINES 1 - 2

# 

ROW 2: LINES 2 - 8

# 

ROW 3: LINES 8 - 11

# 

ROW 4: LINES 11 - 15

ROW 5: LINES 15 - 24

# 

ROW 6: LINES 24 - 24

# 

XM2PD (eXtended Memory TO Printer, Data)

Program No.: 41D2-6 Other Programs in This File: XM2DD, MS2PD, MS2DD

## Purpose

To copy data from extended memory to the printer with all parameters prompted for. XM2PD can print one or more consecutive registers or registers separated by an increment value. If the printer is not connected, XM2PD operates like XM2DD.

#### Minimum SIZE: 003

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 3. Execute XM2PD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to print and press [R/S].

b. XMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) extended memory registers to print and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be printed. If II is not specified, it will default to 1.

c. # DECIMALS=?

Key in the number of decimal places to be printed and press [R/S]. To default to 2 decimal places, just press [R/S].

 The HP-41 will display "\*WORKING\*" and print each register in the form

 $R_{XX} = yyy_yy_y$ 

where

 $\underline{xx}$  = register number being printed, and  $\underline{yyy}.yy$  = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.) 5. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

1. Print the contents of extended memory file "D2-6A" registers 05, 08, 11, 14, and 17 accurate to two decimal places.

StepPromptUser InputDisplayPrinter Outputpreliminary - Use X2XMD to create and load extended memory file<br/>"D2-6A" with size 18 or more."D2-6A""D2-6A"

1.		[XEQ]"SIZE"003	
2.	(Set	printer to MAN	mode)
3.		[XEQ]"XM2PD"	
a.	XM FILE?	D2-6A [R/S]	
b.	XMbbb.eeeII=?	5.01703 [R/S]	
с.	<pre># DECIMALS=?</pre>	[R/S]	2.00
4.			*WORKING*

R5= y.yy R8= y.yy R11= y.yy R14= y.yy R17= y.yy

5.

DONE

XM2DD (eXtended Memory TO Display, Data)

Program No.: 41D2-7 Other Programs in This File: XM2PD, MS2PD, MS2DD

### Purpose

To copy data from extended memory to the display with all parameters prompted for. XM2DD can display one or more consecutive registers or registers separated by an increment value. If the printer is connected, XM2DD operates like XM2PD.

#### Minimum SIZE: 003

### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2DD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to display and press [R/S].

b. XMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) extended memory registers to display and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be displayed. If II is not specified, it will default to 1.

c. # DECIMALS=?

Key in the number of decimal places to be displayed and press [R/S]. To default to 2 decimal places, just press [R/S].

 The HP-41 will display "\*WORKING\*", then display each register in the form

 $R_{XX} = yyy_yy$ 

where

xx = register number being displayed, and yyy.yy = contents of register xx. (In this case, # DECIMALS= 2.)

4. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Display the contents of extended memory file "D2-7A" registers 05, 08, 11, 14, and 17 accurate to two decimal places.

preliminary - Use X2XMD to create and load exte memory file "D2-7A" with size 18	or more.
<pre>1. [XEQ]"SIZE"003 2. [XEQ]"XM2DD" a. XM FILE? D2-7A [R/S] b. XMbbb.eeeII=? 5.01703 [R/S] c. # DECIMALS=? [R/S] 2.00 3. *WORKIN R5 = y R8 = y R8 = y R11 = y R14 = y R17 = y</pre>	IG* ' <u>YY•YY</u> ' <del>YY•YY</del> ' <del>YY•YY</del>

Comments

 To stop the HP-41 after each display, set flag 00 before executing XM2DD. Press [R/S] after each display to continue. MS2MD (<u>Mass S</u>torage <u>TO Main memory</u>, <u>D</u>ata)

Program No.: 41D3-1 Other Programs in This File: XM2MD

#### Purpose

To copy data from mass storage to main memory with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory register) + (number of registers to copy)

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2MD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to main memory and press [R/S].

b. MS bbb.eee=?

Key in the beginning (bbb) or ending (eee) mass storage registers to be copied to main memory and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. MM bbb=?

Key in the beginning main memory register bbb to copy the beginning mass storage register to and press [R/S]. To default to register 00, just press [R/S].

3. The HP-41 will display \*WORKING\* as data are copied from mass storage to main memory. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Copy the contents of mass storage file "D3-1A" registers 00 through 05 to main memory registers 10 through 15.

<u>Step</u>	Prompt	User Input	<u>Display</u>
prelim	ninary - Use X2M file "D	SD to create and 3–1A" with size 6	load mass storage or more.
1. 2. a. b. c.	MS FILE? MS bbb.eee=? MM bbb=?	[XEQ]"SIZE"016 [XEQ]"MS2MD" D3-1A [R/S] .005 [R/S] 10 [R/S]	
3.			*WORKING*

DONE

MS2XMD (<u>Mass S</u>torage <u>TO</u> e<u>X</u>tended <u>M</u>emory, <u>D</u>ata)

Program No.: 41D3-2 Other Programs in This File: XM2MSD

## Purpose

To copy data from mass storage to extended memory with all parameters prompted for. MS2XMD checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: (ending main memory temporary storage register) + 1

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2XMD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to extended memory and press [R/S].

b. MS bbb.eee=?

Key in the beginning (bbb) and ending (eee) mass storage registers to be copied to extended memory and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy mass storage registers bbb through eee to and press [R/S]. To default to the name of the mass storage file, just press [R/S].

d. XM bbb=?

Key in the beginning extended memory register bbb to copy the beginning mass storage register to and press [R/S]. To default to the beginning of the file (register 00), just press [R/S].

e. TMP bbb.eee=?

Key in the beginning (bbb>2) and ending (eee) main memory temporary storage registers to transfer data through and press [R/S]. If eee is less than bbb, only register bbb will be used.

3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:

a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named extended memory file is created (if necessary) and data are copied from mass storage through main memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

 Copy the contents of mass storage file "D3-2A" registers 01 through 05 to extended memory file "D3-2A" (which already exists) registers 10 through 14. Use main memory registers 10 through 13 for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2MSD and X2XMD to create and load
 mass storage and extended memory files

 "D3-2A" with sizes 6 or more and 15 or

 more, respectively.

 1.
 [XEQ]"SIZE"014

 2.
 [XEQ]"MS2XMD"

 a.
 MS FILE?
 D3-2A [R/S]

b.	MS bbb.eee=?	1.005 [R/S]	
с.	XM FILE?	[R/S]	D3-2A
d.	XM bbb=?	10 [R/S]	
e.	TMP bbb.eee=?	10.013 [R/S]	
4.			*WORKING*
			DONE

 Copy the contents of mass storage file "D3-2B" register 05 to the beginning of extended memory file "D3-2C" (which doesn't exist). Use main memory registers 10 through 20 for temporary storage.

 

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2MSD to create and load mass storage file "D3-2B" with size 6 or more.

 1.
 [XEQ]"SIZE"021

 2.
 [XEQ]"MS2XMD"

 a.
 MS FILE?
 D3-2B [R/S]

b.	MS bbb.eee=?	5 [R/S]
с.	XM FILE?	D3-2C [R/S]

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
d.	XM bbb=?	[R/S]	0.00
е. 3.	TMP bbb.eee=?	10.02 [R/S]	*WORKING*
a.	XM D3-2C SIZE=?	1 [R/S]	
4.			*WORKING* DONE

#### Comments

- 1. The minimum size required for an extended memory file = (XM bbb) +
   (number of registers to copy).
- 2. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.
- 3. As Example 2 demonstrates, it is not necessary to specify a number of temporary main memory registers less than or equal to the number of mass storage registers to copy. If fewer temporary storage registers are needed by MS2XMD, only those needed (in this example, only 1) will be used.

USE THIS PAGE FOR NOTES

MS2MSD (<u>Mass Storage TO Mass Storage</u>, <u>D</u>ata)

Program No.: 41D3-3 Other Programs in This File: XM2XMD

#### Purpose

To increase or decrease the size of a mass storage file or to copy data from one block of mass storage registers to another block within the same file or another file, with all parameters prompted for. MS2MSD checks if the named destination mass storage file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (ending main memory temporary storage register) + 1

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2MSD and follow the prompts:
  - a. MS SRC FILE?

Key in the name of the source mass storage file (six characters or less) and press [R/S].

b. SRC bbb.eee=?

Key in the source beginning (bbb) and ending (eee) mass storage registers and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. MS DST FILE?

Key in the name of the destination mass storage file (six characters or less) to copy source mass storage registers bbb through eee to and press [R/S]. To default to the name of the source mass storage file, just press [R/S].

d. DST bbb=?

Key in the destination beginning mass storage register to copy the source beginning mass storage register to and press [R/S]. To default to register 00, just press [R/S].

- 3. If the mass storage source and destination file names specified are the same, follow the prompt:
  - a. NEW SIZE=?

Key in the new size of the source mass storage file and press [R/S]. To default to the current file size, just press [R/S].

- 4. Follow the prompt:
  - a. TMP bbb.eee=?

Key in the beginning (bbb>7, or bbb>8 if changing file size) and ending (eee) main memory temporary storage registers to transfer data through and press [R/S]. If eee is less than bbb, only register bbb will be used.

- 5. The HP-41 will display "\*WORKING\*" as it checks if the named destination mass storage file exists. If it doesn't, follow the prompt:
  - a. DST <u>file</u> SIZE=?

where

<u>file</u> = destination mass storage file name.

Key in the size of the destination mass storage file to be created and press [R/S].

6. The HP-41 will display "\*WORKING\*" as the named destination mass storage file is created (if necessary) and data are copied from mass storage through main memory back to mass storage. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Copy the contents of mass storage file "D3-3A" registers 01 through 05 to mass storage file "D3-3B" (which doesn't exist) registers 10 through 14. Use main memory registers 20 through 23 for temporary storage.

<u>Step</u>	Prompt	User Input	Display
prelim	ninary - Use X2MS storage	D to create and l file "D3-3A" with	oad mass size 5 or more.
1.		[XEQ]"SIZE"024	
2.		[XEQ]"MS2MSD"	
a.	MS SRC FILE?	D3-3A [R/S]	
b.	SRC bbb.eee=?	1.005 [R/S]	
с.	MS DST FILE?	D3-3B [R/S]	
d.	DST bbb=?	10 [R/S]	
4a.	TMP bbb.eee=?	20.023 [R/S]	
5.			*WORKING*
a.	DST D3-3B SIZE=	? 15 [R/S]	
6.			*WORKING* DONE

 Copy the contents of mass storage file "D3-3C" registers 00 through 05 to registers 10 through 15. Use main memory registers 10 through 20 for temporary storage.

<u>Step</u>	Prompt	User Input	_Display_
prelim	inary – Use X2MS storage	D to create and 1 file "D3-3C" with	oad mass size 6 or more
1.		[XEQ]"SIZE"021	
2.		[XEQ]"MS2MSD"	
a.	MS SRC FILE?	D3-3C [R/S]	
b.	SRC bbb.eee=?	.005 [R/S]	
с.	MS DST FILE?	[R/S]	D3-3C
d.	DST bbb=?	10 [R/S]	
3a.	NEW SIZE=?	16 [R/S]	
4a.	TMP bbb.eee=?	10.02 [R/S]	
5,6.			*WORKING* DONE

3. Decrease the size of mass storage file "D3-3D" from 20 registers to 10 registers while simultaneously moving<sup>1/</sup> data from mass storage registers 15 through 19 to registers 05 through 09. Use main memory registers 10 through 15 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create and load extended memory file "D3-3D" with size 20.

1.		[XEQ]"SIZE"016	
2.		[XEQ]"MS2MSD"	
a.	MS SRC FILE?	D3-3D [R/S]	
b.	SRC bbb.eee=?	15.019 [R/S]	
с.	MS DST FILE?	[R/S]	D3-3D
d.	DST bbb=?	5 [R/S]	
3a.	NEW SIZE=?	10 [R/S]	
4a.	TMP bbb.eee=?	10.015 [R/S]	
5,6.			*WORKING*
			DONE

#### Comments

- The minimum size required for a destination mass storage file = (MS DST bbb) + (number of registers to copy).
- 2. Since flag 25 (error ignore flag) is used to detect if the named destination mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "DST <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case one can resolve the difficulty and restart the program.

<sup>1/</sup>The word "moving" instead of "copying" is appropriate here since the contents
 of the source mass storage registers will not be preserved.

- 3. As Example 2 demonstrates, it is not necessary to specify a number of main memory temporary storage registers less than or equal to the number of mass storage registers to copy. If fewer temporary storage registers are needed by MS2MSD, only those needed (in this example, 6) will be used.
- 4. When a mass storage file size is increased or decreased, MS2MSD temporarily creates a dummy file of the new size. Therefore, a number of mass storage registers equal to the new size must be available to increase or decrease a file size.
- 5. When a mass storage file size is increased or decreased, MS2MSD temporarily creates a dummy file named "...". If a file by this name already exists, MS2MSD will stop with an error message.

MS2CD (<u>Mass Storage TO</u> magnetic <u>Cards</u>, <u>Data</u>)

Program No.: 41D3-4 Other Programs in This File: XM2CD, C2XMD, C2MSD

#### Purpose

To copy data from mass storage to magnetic cards with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers to copy)

### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive Card Reader (HP 82104A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2CD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to magnetic cards and press [R/S].

b. MS bbb.eee=?

Key in the beginning (bbb) and ending (eee) mass storage registers to be copied to magnetic cards and press [R/S]. If eee is less than bbb, only register bbb will be copied.

c. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through ((bbb + number of registers to copy) - 1) will be used.

- 3. The HP-41 will display "\*WORKING\*", then prompt:
  - a. INSERT DST

Insert track 1 of destination magnetic card 1 into the Card Reader.

- 4. If more than 16 mass storage registers are being copied, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be recorded, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

- 5. When all the destination magnetic cards have been recorded, the HP-41 will beep and prompt:
  - a. "VER"IFY CDS1/

Press [XEQ] "VER".

b. CARD

Insert each track recorded into the Card Reader. If the track was recorded properly, the HP-41 will display "TYPE D TR  $\underline{tt}$ ",

where

 $\underline{tt} = track number.$ 

If the HP-41 displays "CARD ERR" or "CHECKSUM ERR", press [<-] and start over at Step 2, perhaps with new magnetic cards.

 When all the destination magnetic cards have been verified, press [<-]. The HP-41 will display "DONE".</li>

## Examples

\_ \_ \_ \_ \_ \_ \_

 Copy the contents of mass storage file "D3-4A" registers 05 through 09 to a magnetic card. Use main memory registers 10 through 14 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create and load mass storage file "D3-4A" with size 10 or more.

1.		[XEQ]"SIZE"015	
2.		[XEQ]"MS2CD"	
a.	MS FILE?	D3-4A [R/S]	
b.	MS bbb.eee=?	5.009 [R/S]	
с.	TMP bbb=?	10 [R/S]	
3.			*WORKING*
a.	INSERT DST	(Insert track 1)	
5a.	"VER"IFY CDS	[XEQ]"VER"	
b.	CARD	(Insert track 1)	TYPE D TR 01
	CARD		
6.		[<-]	DONE

<sup>1</sup>/Before using the CR: VER function, refer to Section D for an explanation of the VER bug that can alter extended memory.

2. Copy the contents of mass storage file "D3-4B" registers 00 through 39 to magnetic cards. Use main memory registers 05 through 45 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create and load mass storage file "D3-4B" with size 40 or more.

1.		[XEQ]"SIZE"046	
2.		[XEQ]"MS2CD"	
a.	MS FILE?	D3-4B [R/S]	
b.	MS bbb.eee=?	.039 [R/S]	
с.	TMP bbb=?	5 [R/S]	
3.			*WORKING*
a.	INSERT DST	(Insert track 1)	
4a.	RDY 02 OF 03	(Insert track 2)	
	RDY 03 OF 03	(Insert track 3)	
5a.	"VER"IFY CDS	[XEQ]"VER"	
b.	CARD	(Insert track 1)	TYPE D TR 01
	CARD	(Insert track 2)	TYPE D TR 02
	CARD	(Insert track 3)	TYPE D TR 03
	CARD		
6.		[<-]	DONE

#### <u>Comments</u>

- The destination magnetic cards may be inserted in any order in Steps 3, 4, and 5; the HP-41 automatically keeps track of where data reside on the cards.
- 2. Any unused registers on a magnetic card track may be envisioned as filled with "null values" that won't overstore main memory registers in subsequent magnetic card to main memory operations.

# USE THIS PAGE FOR NOTES

MS2XD (<u>Mass Storage IO X</u> register, <u>D</u>ata)

Program No.: 41D3-5 Other Programs in This File: XM2XD

#### <u>Purpose</u>

To copy data from mass storage to the X register with all parameters prompted for.

<u>Minimum SIZE</u>: (main memory temporary storage register) + 1

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2XD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to X and press [R/S].

b. MS RRR=?

Key in the mass storage register RRR to be copied to X and press [R/S].

c. TMP RRR=?

Key in the main memory temporary storage register (RRR>1) to transfer data through and press [R/S].

3. The HP-41 will display "\*WORKING\*" as the specified mass storage register is copied to X. When completed, the HP-41 will beep, stop, and display the X register.

#### Examples

1. Copy the contents of mass storage file "D3-5A" register 05 to the X register. Use main memory register 10 for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create and load mass storage file "D3-5A" with size 6 or more. Store 1 in register 05.

1.		[XEQ]"SIZE"011	
2.		[XEQ]"MS2XD"	
a.	MS FILE?	D3-5A [R/S]	
b.	MS RRR=?	5 [R/S]	
с.	TMP RRR=?	10 [R/S]	
3.			*WORKING*
			1.00

# USE THIS PAGE FOR NOTES

MS2PD (<u>Mass Storage IO Printer</u>, <u>Data</u>)

Program No.: 41D3-6 Other Programs in This File: XM2PD, XM2DD, MS2DD

#### Purpose

To copy data from mass storage to the printer with all parameters prompted for. MS2PD can print one or more consecutive registers or registers separated by an increment value. If the printer is not connected, MS2PD operates like MS2DD.

#### Minimum SIZE: 004

#### Required Peripherals

#### Instructions

- 1. Insure that the minimum SIZE is set.
- Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 3. Execute MS2PD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to print and press [R/S].

b. MSbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) mass storage registers to print and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be printed. If II is not specified, it will default to 1.

c. # DECIMALS=?

Key in the number of decimal places to be printed and press [R/S]. To default to 2 decimal places, just press [R/S].

 The HP-41 will display "\*WORKING\*" and print each register in the form

 $R_{XX} = yyy_yy_y$ 

where

 $\underline{xx}$  = register number being printed, and  $\underline{yyy}$ .yy = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.) 5. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Print the contents of mass storage file "D3-6A" registers 05, 08, 11, 14, and 17 accurate to two decimal places. Printer Output

<u>Step</u>	Prompt	User Input	_Display_	(HP_82162A)
prelim	inary - Use X2MS "D3-6A"	D to create and with size 18 or	load mass sto more.	orage file
1. 2. 3. a. b. c. 4.	(Set MS FILE? MSbbb.eeeII=? # DECIMALS=?	[XEQ]"SIZE"004 printer to MAN [XEQ]"MS2PD" D3-6A [R/S] 5.01703 [R/S] [R/S]	mode) 2.00 *WORKING*	<b>R</b> 5= Y.YY
5.			DONE	R8= y.yy R11= y.yy R14= y.yy R17= y.yy

5.

MS2DD (<u>Mass S</u>torage <u>TO</u> Display, <u>D</u>ata)

Program No.: 41D3-7 Other Programs in This File: XM2PD, XM2DD, MS2PD

### Purpose

To copy data from mass storage to the display with all parameters prompted for. MS2DD can display one or more consecutive registers or registers separated by an increment value. If the printer is connected, MS2DD operates like MS2PD.

#### Minimum SIZE: 004

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2DD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to display and press [R/S].

b. MSbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) mass storage registers to display and the increment value II and press [R/S]. If eee is less than bbb, only register bbb will be displayed. If II is not specified, it will default to 1.

c. # DECIMALS=?

Key in the number of decimal places to be displayed and press [R/S]. To default to 2 decimal places, just press [R/S].

 The HP-41 will display "\*WORKING\*", then display each register in the form

 $R_{XX} = yyy, yy,$ 

where

 $\underline{xx}$  = register number being displayed, and  $\underline{yyy.yy}$  = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.)

4. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

1. Display the contents of mass storage file "D3-7A" registers 05, 08, 11, 14, and 17 accurate to two decimal places.

Step Prompt User Input Display preliminary - Use X2MSD to create and load mass storage file "D3-7A" with size 18 or more. [XEQ]"SIZE"004 1. 2. [XEQ]"MS2DD" MS FILE? D3-7A [R/S] a. 5.01703 [R/S] b. MSbbb.eeeII=? # DECIMALS=? [R/S] 2.00 с. 3. \*WORKING\* R5 =  $yyy \cdot yy$  $R8 = yyy_yy$ 

4.

## Comments

1. To stop the HP-41 after each display, set flag 00 before executing MS2DD. Press [R/S] after each display to continue.

 $R11 = \underline{yyy},\underline{yy}$   $R14 = \underline{yyy},\underline{yy}$  $R17 = \underline{yyy},\underline{yy}$ 

DONE

C2MD (magnetic <u>Cards TO Main memory</u>, <u>Data</u>)

Program No.: 41D4-4 Other Programs in This File: M2CD

## Purpose

To copy data from magnetic cards to main memory with all parameters prompted for.

Minimum SIZE: (ending main memory register) + 1

#### Required Peripherals

Card Reader (HP 82104A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2MD and follow the prompts:
  - a. MM bbb.eee=?

Key in the beginning (bbb) and ending (eee) main memory registers to copy magnetic cards to and press [R/S]. If eee is less than bbb, only register bbb will be loaded.

b. INSERT SRC

Insert track 1 of source magnetic card 1 into the Card Reader.

- 3. If more than 16 main memory registers are being loaded, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

4. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

1. Copy the contents of a magnetic card to main memory registers 10 through 15.

User Input Display Step Prompt preliminary - Use M2CD to copy 6 or more main memory registers to a magnetic card. [XEQ]"SIZE"016 1. [XEQ]"C2MD" 2. MM bbb.eee=? 10.015 [R/S] a. INSERT SRC (Insert track 1) b. DONE 4.

2. Copy the contents of magnetic cards to main memory registers 10 through 50.

 Step
 Prompt
 User Input
 Display

 preliminary - Use M2CD to copy 40 or more main memory registers to magnetic cards.

 1.
 [XEQ]"SIZE"051

 2.
 [XEQ]"C2MD"

 a.
 MM bbb.eee=?
 10.05 [R/S]

 b. INSERT SRC (Insert track 1)
 3a. RDY 02 OF 03 (Insert track 2) RDY 03 OF 03 (Insert track 3)
 4. DONE

#### Comments

- The source magnetic cards may be inserted in any order in Steps 2 and 3; the HP-41 automatically keeps track of where data reside on the cards.
- 2. If the source magnetic cards were recorded with fewer registers than needed, C2MD will copy only those available to main memory.

C2XMD (magnetic <u>Cards TO</u> e<u>X</u>tended <u>Memory</u>, <u>Data</u>)

Program No.: 41D4-2 Other Programs in This File: XM2CD, MS2CD, C2MSD

#### <u>Purpose</u>

To copy data from magnetic cards to extended memory with all parameters prompted for. C2XMD checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers to copy)

Required Peripherals

Card Reader (HP 82104A) Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2XMD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy magnetic cards to and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) and ending (eee) extended memory registers to copy magnetic cards to and press [R/S]. If eee is less than bbb, only register bbb will be loaded.

c. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through ((bbb + number of registers to copy) - 1) will be used.

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.
- 4. Follow the prompt:
  - a. INSERT SRC

Insert track 1 of source magnetic card 1 into the Card Reader.

- 5. If more than 16 extended memory registers are being loaded, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

6. When all the source magnetic cards have been read, the HP-41 will display "\*WORKING\*" as data are copied from the main memory temporary storage registers to extended memory, then beep, stop, and display "DONE".

#### <u>Examples</u>

 Copy the contents of a magnetic card to extended memory file "D4-2A" (which already exists) registers 05 through 10. Use main memory registers 10 through 15 for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Use M2CD to copy 6 or more main memory registers to a magnetic card and X2XMD to create and load extended memory file "D4-2A" with size 11 or more.

 1.
 [XEQ]"SIZE"016

 2.
 [XEQ]"C2XMD"

 a.
 XM FILE?

a.	XM FILE?	D4-2A [R/S]	
b.	XM bbb.eee=?	5.01 [R/S]	
с.	TMP bbb=?	10 [R/S]	
3.			*WORKING*
4a.	INSERT SRC	(Insert track 1)	
6.			*WORKING*
			DONE

 Copy the contents of magnetic cards to extended memory file "D4-2B" (which doesn't exist) registers 10 through 50. Use main memory registers 05 through 45 for temporary storage.

<u>Step</u>	Prompt	<u>User Input</u>	_Display_
1.		[XEQ]"SIZE"046	
2.		[XEQ]"C2XMD"	
a.	XM FILE?	D4-2B [R/S]	

<u>Step</u>	Prompt User Input	Display
b.	XM bbb.eee=? 10.05 [R/S]	
3.		*WORKING*
a.	XM D4-2B SIZE=? 51 [R/S]	*WORKING*
4a.	INSERT SRC (Insert track 1)	
5a.	RDY 02 OF 03 (Insert track 2)	
	RDY 03 OF 03 (Insert track 3)	
6.		*WORKING* DONE

- The source magnetic cards may be inserted in any order in Steps 4 and 5; the HP-41 automatically keeps track of where data reside on the cards.
- 2. If the source magnetic cards were recorded with fewer registers than needed, C2XMD will copy only those on the cards to extended memory.
- 3. As Example 2 demonstrates, the minimum size required for an extended memory file = (XM bbb) + (number of registers to load).
- 4. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.

C2MSD (magnetic <u>Cards TO Mass Storage</u>, <u>Data</u>)

Program No.: 41D4-3 Other Programs in This File: XM2CD, MS2CD, C2XMD

## Purpose

To copy data from magnetic cards to mass storage with all parameters prompted for. C2MSD checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers to copy)

#### Required Peripherals

Card Reader (HP 82104A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2MSD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy magnetic cards to and press [R/S].

b. MS bbb.eee=?

Key in the beginning (bbb) and ending (eee) mass storage registers to copy magnetic cards to and press [R/S]. If eee is less than bbb, only register bbb will be loaded.

c. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through ((bbb + number of registers to copy) - 1) will be used.

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. INSERT SRC

Insert track 1 of source magnetic card 1 into the Card Reader.

- 5. If more than 16 mass storage registers are being loaded, follow the prompt:
  - a. RDY <u>tt</u> of <u>nn</u>

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

6. When all the source magnetic cards have been read, the HP-41 will display "\*WORKING\*" as data are copied from the main memory temporary storage registers to mass storage, then beep, stop, and display "DONE".

# **Examples**

Step

 Copy the contents of a magnetic card to mass storage file "D4-3A" (which already exists) registers 05 through 10. Use main memory registers 10 through 15 for temporary storage.

Prompt User Input Display

preliminary - Use M2CD to copy 6 or more main memory registers to a magnetic card and X2MSD to create and load mass storage file "D4-3A" with size 11 or more.

1.		[XEQ]"SIZE"016	
2.		[XEQ]"C2MSD"	
a.	MS FILE?	D <b>4-3A</b> [R/S]	
b.	MS bbb.eee=?	5.01 [R/S]	
с.	TMP bbb=?	10 [R/S]	
3.			*WORKING*
4a.	INSERT SRC	(Insert track 1)	
6.			*WORKING*
			DONE

 Copy the contents of magnetic cards to mass storage file "D4-3B" (which doesn't exist) registers 10 through 50. Use main memory registers 05 through 45 for temporary storage.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1.		[XEQ]"SIZE"046	
2.		[XEQ]"C2MSD"	
a.	MS FILE?	D4-3B [R/S]	

<u>Step</u>	Prompt User Input	Display
b.	MS bbb.eee=? 10.05 [R/S]	
3.		*WORKING*
a.	MS D4-3B SIZE=? 51 [R/S]	*WORKING*
4a.	INSERT SRC (Insert track 1	)
5a.	RDY 02 OF 03 (Insert track 2	2)
	RDY 03 OF 03 (Insert track 3	)
6.		*WORKING*
		DONE

- The source magnetic cards may be inserted in any order in Steps 4 and 5; the HP-41 automatically keeps track of where data reside on the cards.
- 2. If the source magnetic cards were recorded with fewer registers than needed, C2MSD will copy only those on the cards to mass storage.
- 3. As Example 2 demonstrates, the minimum size required for a mass storage file = (MS bbb) + (number of registers to load).
- 4. Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.

٠

C2CD (magnetic <u>Cards IO</u> magnetic <u>Cards</u>, <u>Data</u>)

Program No.: 41D4-4 Other Programs in This File: none

#### Purpose

To copy data from one set of magnetic cards to another with all parameters prompted for. C2CD produces an exact copy of the source magnetic cards.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers on source magnetic cards) + 1

#### Required Peripherals

Card Reader (HP 82104A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2CD and follow the prompts:
  - a. TMP bbb=?

Key in the beginning main memory temporary storage register bbb and press [R/S]. To default to register 00, just press [R/S]. Registers bbb through (SIZE-1) will be used as temporary storage.

- 3. The HP-41 will display "\*WORKING\*" as the main memory temporary storage registers are initialized, then prompt:
  - a. INSERT SRC

Insert track 1 of source magnetic card 1 into the Card Reader.

- If more than 16 main magnetic card registers are being copied, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

- 5. When all the source magnetic cards have been read, the HP-41 will display "\*WORKING\*" as the number of registers required is determined, then prompt:
  - a. INSERT DST

Insert track 1 of destination magnetic card 1 into the Card Reader.

6. If more than 16 magnetic card registers are being copied, follow the prompt:

a. RDY tt of nn

where

 $\underline{tt}$  = next track to be recorded, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

- 7. When all the destination magnetic cards have been recorded, the HP-41 will beep and prompt:
  - a. "VER"IFY CDS1/

Press [XEQ] "VER".

b. CARD

Insert each track recorded into the Card Reader. If the track was recorded properly, the HP-41 will display "TYPE D TR  $\underline{tt}$ ",

where

<u>tt</u> = track number.

If the HP-41 displays "CARD ERR" or "CHECKSUM ERR", press [<-] and start over at Step 2, perhaps with new magnetic cards.

 When all the destination magnetic cards have been verified, press [<-]. The HP-41 will display "DONE". To start over at Step 3, press [R/S].

## Examples

1. Copy the contents of a magnetic card track to another magnetic card. Use main memory registers 00 and up for temporary storage.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary – Use M2CD to copy up to 16 main memory registers to a magnetic card.

1.		[XEQ]"SIZE"017		
2.		[XEQ]"C2CD"		
a.	TMP bbb=?	[R/S]	0.00	

Here using the CR: VER function, refer to Section D for an explanation of the VER bug that can alter extended memory.

<u>Step</u>	Prompt	User Input	Display	
3. _a.	INSERT SRC	(Insert source track	*WORKING*	
5. a. 7a.	INSERT DST "VER"IFY CDS	(Insert destination [XEQ]"VER"	*WORKING* track 1)	
b.	CARD CARD	(Insert destination	track l)	TYPE D TR 01
8.		[<-]	DONE	

2. Copy the contents of three magnetic card tracks to two other magnetic cards. Use main memory registers 05 and up for temporary storage.

<u>Step Prompt User Input Display</u> preliminary - Use M2CD to copy 33 to 48 main memory registers to magnetic cards.

1.		[XEQ]"SIZE"054		
2.		[XEQ]"C2CD"		
a.	TMP bbb=?	5 [R/S]		
3.			*WORKING*	
a.	INSERT SRC	(Insert source trac	k 1)	
4a.	RDY 02 OF 03	(Insert source trac	k 2)	
	RDY 03 OF 03	(Insert source trac	k 3)	
5.			*WORKING*	
a.	INSERT DST	(Insert destination	track l)	
6a.	RDY 02 OF 03	(Insert destination	track 2)	
	RDY 03 OF 03	(Insert destination	track 3)	
7a.	"VER"IFY CDS	[XEQ]"VER"		
b.	CARD	(Insert destination	track l)	TYPE D TR 01
	CARD	(Insert destination	track 2)	TYPE D TR 02
	CARD	(Insert destination	track 3)	TYPE D TR 03
	CARD			
8.		[<-]	DONE	

- 1. The source and destination magnetic cards may be inserted in any order in Steps 3 through 7; the HP-41 automatically keeps track of where data reside on the cards.
- C2CD uses the alphanumeric value "..." to identify the last main memory register into which the last source magnetic card register is read. Therefore, if any source magnetic card registers have this value, C2CD will produce erroneous results.
- 3. Any unused registers on a magnetic card track may be envisioned as filled with "null values" that won't overstore main memory registers in subsequent magnetic card to main memory operations.

C2PD (magnetic <u>Cards IO</u> Printer, <u>Data</u>)

Program No.: 41D4-6 Other Programs in This File: C2DD

### Purpose

To copy data from magnetic cards to the printer with all parameters prompted for. If the printer is not connected, C2PD operates like C2DD.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers on magnetic cards) + 1

#### Required Peripherals

Card Reader (HP 82104A) Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

# Instructions

- 1. Insure that the minimum SIZE is set.
- Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 3. Execute C2PD and follow the prompts:
  - a. # DECIMALS=?

Key in the number of decimal places to be printed and press [R/S]. To default to 2 decimal places, just press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through (SIZE-1) will be used as temporary storage.

- 4. The HP-41 will display "\*WORKING\*" as the main memory temporary storage registers are initialized, then prompt:
  - a. INSERT SRC

Insert track 1 of magnetic card 1 into the Card Reader.

- 5. If more than 16 magnetic card registers are being printed, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

6. When all the magnetic cards have been read, the HP-41 will display "\*WORKING\*" and print each magnetic card register in the form

 $R_{XX} = yyy_yy_y$ 

where

7.

xx = register number being printed, and yyy.yy = contents of register xx. (In this case, # DECIMALS= 2.) If there are less than 16 registers on the last track printed, only the number on that track will be printed.

7. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

 Print the contents of a full magnetic card track accurate to three decimal places. Use main memory registers 05 and up for temporary storage.

<u>Step</u>	Prompt	User Input	Display	Printer Output (HP 82162A)
prelim	ninary - Use M2CD magnetic	to copy 16 main card.	memory regi	sters to a
1. 2. 3. a. b.	(Set # DECIMALS=? TMP bbb=?	[XEQ]"SIZE"022 printer to MAN m [XEQ]"C2PD" 3 [R/S] 5 [R/S]	node)	
4. a.	INSERT SRC	(Insert track 1)	*WORKING*	
6.			*WORKING*	
				<b>R0=</b> y.yyy
				R1= y.yyy
				R2= y.yyy
				R3= y.yyy
				R4= y.yyy
				R5= y,yyy
				R6= y.yyy
				R7= y.yyy
				R8= y,yyy
				<b>R9= y.yyy</b>
				R10= y.yyy
				R11= y.yyy
				R12= y.yyy
				R13= y.yyy
				R14= y.yyy
				R15= y.yyy

2. Print the contents of a magnetic card recorded with 20 registers accurate to two decimal places. Use main memory registers 10 and up for temporary storage.

Step	Prompt	User Input	_Display_	Printer Output (HP 82162A)
prelim	ninary - Use M2C magneti	D to copy 20 main c card.	memory regi	sters to a
1. 2.	(Se	[XEQ]"SIZE"031 t printer to MAN	mode)	
3. a. b.	∦ DECIMALS=? TMP bbb=?	[XEQ]"C2PD" [R/S] 10 [R/S]	2.00	
4. a. 5a.	INSERT SRC RDY 02 OF 02	(Insert track 1) (Insert track 2)	*WORKING*	
6.			*WORKING*	<b>РЙ</b> = У,УУ
				R1= y.yy
				R2= y.yy
				R3= y.yy
				<b>₽4</b> = y.yy
				R5= y.yy
				R6= y.yy
				R7= y.yy
				R8= y.yy
				R9= y.yy
				R10= y,yy
				R11= y.yy
				R12= y.yy
				R13= y.yy
				R14= y.yy
				R15= Y.YY
				R16= Y.YY
				R17= y.yy
				R18= y.yy

7.

DONE

R19= Y.YY

- 1. The magnetic cards may be inserted in any order in Steps 4 and 5; the HP-41 automatically keeps track of where data reside on the cards.
- 2. C2PD uses the alphanumeric value "..." to identify the last main memory temporary storage register to print. Therefore if any source magnetic card registers have this value, C2PD will produce erroneous results.

C2DD (magnetic <u>Cards TO Display</u>, <u>Data</u>)

Program No.: 41D4-7 Other Programs in This File: C2PD

## <u>Purpose</u>

To copy data from magnetic cards to the display with all parameters prompted for. If the printer is connected, C2DD operates like C2PD.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of registers on magnetic cards) + 1

# Required Peripherals

Card Reader (HP 82104A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2DD and follow the prompts:
  - a. # DECIMALS=?

Key in the number of decimal places to be displayed and press [R/S]. To default to 2 decimal places, just press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>2) and press [R/S]. Registers bbb through (SIZE-1) will be used as temporary storage.

- 3. The HP-41 will display "\*WORKING\*" as the main memory temporary storage registers are initialized, then prompt:
  - a. INSERT SRC

Insert track 1 of magnetic card 1 into the Card Reader.

- 4. If more than 16 magnetic card registers are being displayed, follow the prompt:
  - a. RDY tt of nn

where

tt = next track to be read, and <u>nn</u> = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

5. When all the magnetic cards have been read, the HP-41 will display "\*WORKING\*", then display each magnetic card register in the form

 $R_{XX} = yyy_yy_y$ 

where

<u>Step</u>

Prompt

 $\underline{xx}$  = register number being displayed, and  $\underline{yyy}$ ,  $\underline{yy}$  = contents of register  $\underline{xx}$ . (In this case, # DECIMALS= 2.)

If there are less than 16 registers on the last track displayed, only the number on that track will be displayed.

6. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Display the contents of a full magnetic card track accurate to three decimal places. Use main memory registers 05 and up for temporary storage.

preliminary - Use M2CD to copy 16 main memory registers to a magnetic card.

User Input

<u>Display</u>

1. 2. a. b.	<pre># DECIMALS=? TMP bbb=?</pre>	[XEQ]"SIZE"022 [XEQ]"C2DD" 3 [R/S] 5 [R/S]	
3.			*WORKING*
a. 5.	INSERT SRC	(Insert track 1)	*WORKING* R0 = <u>yyy.yyy</u> R1 = <u>yyy.yyy</u> R2 = <u>yyy.yyy</u> R3 = <u>yyy.yyy</u> R4 = <u>yyy.yyy</u> R5 = <u>yyy.yyy</u> R6 = <u>yyy.yyy</u> R7 = <u>yyy.yyy</u> R8 = <u>yyy.yyy</u> R10 = <u>yyy.yyy</u> R10 = <u>yyy.yyy</u> R12 = <u>yyy.yyy</u> R12 = <u>yyy.yyy</u> R13 = <u>yyy.yyy</u> R14 = <u>yyy.yyy</u>
6.			DONE

2. Display the contents of a magnetic card recorded with 20 registers accurate to two decimal places. Use main memory registers 10 and up for temporary storage.

<u>Step</u>	Prompt	User Input	Display
prelimi	nary - Use M2CD magnetic	to copy 20 main card.	memory registers to a
1. 2. a. 3. a. 4a. 5.	<pre># DECIMALS=? TMP bbb=? INSERT SRC RDY 02 OF 02</pre>	<pre>[XEQ]"SIZE"031 [XEQ]"C2DD" [R/S] 10 [R/S] (Insert track 1) (Insert track 2)</pre>	2.00 *WORKING* R0 = <u>yyy.yy</u> R1 = <u>yyy.yy</u> R1 = <u>yyy.yy</u> R2 = <u>yyy.yy</u> R3 = <u>yyy.yy</u> R4 = <u>yyy.yy</u> R5 = <u>yyy.yy</u> R5 = <u>yyy.yy</u> R6 = <u>yyy.yy</u> R6 = <u>yyy.yy</u> R7 = <u>yyy.yy</u> R8 = <u>yyy.yy</u> R10 = <u>yyy.yy</u> R10 = <u>yyy.yy</u> R11 = <u>yyy.yy</u> R12 = <u>yyy.yy</u> R13 = <u>yyy.yy</u> R13 = <u>yyy.yy</u> R14 = <u>yyy.yy</u> R15 = <u>yyy.yy</u> R15 = <u>yyy.yy</u> R16 = <u>yyy.yy</u> R17 = <u>yyy.yy</u> R18 = <u>yyy.yy</u> R18 = <u>yyy.yy</u>
6.			DONE

- 1. The magnetic cards may be inserted in any order in Steps 3 and 4; the HP-41 automatically keeps track of where data reside on the cards.
- C2DD uses the alphanumeric value "..." to identify the last main memory temporary storage register to display. Therefore if any source magnetic card registers have this value, C2DD will produce erroneous results.
- 3. To stop the HP-41 after each display, set flag 00 before executing C2DD. Press [R/S] after each display to continue.

X2MD (X register <u>IO</u> <u>Main</u> memory, <u>D</u>ata)

Program No.: 41D5-1 Other Programs in This File: none

#### <u>Purpose</u>

To copy data from the X register to main memory with all parameters prompted for. X2MD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value.

Minimum SIZE: (ending main memory register) + 1

## Required Peripherals: none

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute X2MD and follow the prompts:

a. MMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) main memory registers to copy X to and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be loaded. If II is not specified, it will default to 1.

where

<u>rrr</u> = next register to copy X to.

Key in the numeric value (not in ALPHA mode) to be copied to main memory register <u>rrr</u> and press [R/S]; OR Key in the alphanumeric value (in ALPHA mode - six characters or less) to be copied to register <u>rrr</u> and press [R/S].

Repeat this step until main memory registers bbb through eee have been loaded with the desired data.

3. When completed, the HP-41 will beep, stop, and display "DONE".

1. Copy the constants Pi, 1, and  $e^1$  to main memory registers 10 through 12.

<u>Step</u>	Prompt	User Input	<u>Display</u>
1. 2. a. b.	MMbbb.eeeII=? R10=? R11=? R12=?	[XEQ]"SIZE"013 [XEQ]"X2MD" 10.012 [R/S] [Pi] [R/S] 1 [R/S] 1 [e <sup>X</sup> ] [R/S]	
3.			DONE

2. Copy the following table to every other main memory register starting with register 05, in "row major" form (i.e., row by row):

		ID	<u>Species</u>	
		1 2 3	Oak Maple Birch	
<u>Step</u>	Prompt	Use	er Input	_Display_
1. 2.	MMbbb.eeeIT=?	[XEQ] [XEQ] 5.01	]"SIZE"016 ]"X2MD" 502 [R/S]	
b.	R5=?	1 [R/	/S]	
	R7=?	[ALPHA]	OAK [R/S]	
	R9?	[ALPHA]	2 [R/S]	
	R11=?	[ALPHA]	MAPLE [R/S]	
	R13?	[ALPHA]	3 [R/S]	
	R15=?	[ALPHA]	BIRCH [R/S]	
3.				DONE

- As Example 2 demonstrates, it is acceptable to toggle back and forth (using the [ALPHA] key) between storage of numeric and alphanumeric data.
- 2. STOD, a copy of which is shown below, can be used as an alternative to the X2MD program. The number to be stored is first keyed, then when STOD is executed (for convenience assign STOD to the [STO] key and use USER mode), a prompt of the form STO\_\_\_\_ is obtained. Note that three digits need not be provided in all cases. For example, to store the contents of the X register in main memory register 10 one would simply key 1 and 0; to store X in register 08, key only 8; but to store X in register 100, one must key 1, 0, and 0. As a further precaution, note that there is a programmed one-second pause between digits, therefore successive digits after the first must be supplied promptly to avoid erroneous results.

Program Listing for STOD:

Program Registers: 9 Bytes: 62

01+LBL "RCLD" 02 "RCL\_\_\_" 03 XE0 00 04 X<>L 05 RCL IND L 06 RTN 07+LBL "STOD" 08 \*STO\_\_\_\* **0**9 X<> L 10 RDN 11 XEQ 00 12 X<>L 13 STO IND L 14 RTN 15+LBL 00 16 CF 22 17 AVIEW 18+LBL 01 19 PSE 20 FC? 22 21 GTO 01 22 END

(See page 93 for program barcode)

X2XMD (X register <u>TO</u> eXtended <u>Memory</u>, <u>D</u>ata)

Program No.: 41D5-2 Other Programs in This File: X2MSD

# Purpose

To copy data from the X register to extended memory with all parameters prompted for. X2XMD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value. X2XMD checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

# Minimum SIZE: 002

Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute X2XMD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy X to and press [R/S].

b. XMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) extended memory registers to copy X to and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be loaded. If II is not specified, it will default to 1.

- 3. If the named extended memory file doesn't exist, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. R<u>rrr</u>=? (if not in ALPHA mode) OR R<u>rrr</u>? (if in ALPHA mode)

where

<u>rrr</u> = next register to copy X to.

Key in the numeric value (not in ALPHA mode) to be copied to extended memory register <u>rrr</u> and press [R/S]; OR Key in the alphanumeric value (in ALPHA mode - six

characters or less) to be copied to register <u>rrr</u> and press [R/S].

Repeat this step until extended memory registers bbb through eee have been loaded with data.

5. When completed, the HP-41 will beep, stop, and display "DONE".

# Examples

1. Copy the constants Pi, 1, and e<sup>1</sup> to extended memory file "D5-2A" (which already exists) registers 10 through 12.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
preli	minary -	"D5-2A" 13 [XEQ]"CRFLD"	
1. 2. a. b. 4a.	XM FILE? XMbbb.eeeII=? R10=? R11=? R12=?	[XEQ]"SIZE"002 [XEQ]"X2XMD" D5-2A [R/S] 10.012 [R/S] [Pi] [R/S] 1 [R/S] 1 [e <sup>×</sup> ] [R/S]	
5.			DONE

2. Copy the following table to every other register of extended memory file "D5-2B" (which doesn't exist), starting with register 05, in "column-major" form (i.e., column by column):

		ID	<u>Component</u>	
		1 2 3	Diode IC Switch	
<u>Step</u>	Prompt		User Input	_Display_
1. 2. b. 3a. 4a.	XM FILE? XMbbb.eeeII=? XM D5-2B SIZE= R5=? R7=? R9=? B11=?	[X [X D5 5. ? 1 2 3 AI PH	EQ]"SIZE"002 EQ]"X2XMD" -2B [R/S] 01502 [R/S] 16 [R/S] [R/S] [R/S] [R/S]	*WORKING*

<u>Step</u>	Prompt	User Input	Display
	R13?	IC [R/S]	
5.	R15?	SWITCH [R/S]	DONE

#### Comments

- 1. It is acceptable to toggle back and forth (using the [ALPHA] key) between storage of numeric and alphanumeric data.
- As Example 2 demonstrates, the minimum size required for an extended memory file = (XM bbb) + (number or X values to copy).
- 3. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one may resolve the difficulty and restart the program.
- 4. STEX, a copy of which is shown below, can be used as an alternative to the X2XMD program. One first must make the desired extended memory file the working file, then key the number to be stored (i.e., put it in the X register). When STEX is executed (for convenience assign STEX to the [STO] key and use USER mode), a prompt of the form STEX\_\_\_\_\_\_\_ is obtained, enabling the storing of the X register contents into any existing register of the working file in extended memory. Note that the prompt need not be supplied with three digits in all cases. For example to store the contents of the X register into extended memory register 10, one would simply key 1 and 0; to store X in register 08, key only 8; but to store X in register 100, one must key 1, 0, and 0. As a further precaution, note that there is a programmed one-second pause between digits, therefore successive digits after the first must be supplied promptly to avoid erroneous results.

Program Listing for STEX:

Program Registers: 10 Bytes: 67

13 SEEKPT 01+LBL "RCEX" -14 X⇔ L 02 "RCEX\_\_\_\_" 15 SAVEX 03 XEQ 00 16 RTN 04 SEEKPT 95 CLX 17+LBL 00 18 CF 22 06 GETX 19 AVIEW 07 RTN 20+LBL 01 08+LBL "STEX" 21 PSE 09 X(> L 22 FC? 22 10 RDN 23 GTO 01 11 \*STEX\_\_\_\* 24 END 12 XEQ 00

(See page 113 for program barcode)

X2MSD (X register <u>TO Mass Storage</u>, <u>D</u>ata)

Program No.: 41D5-3 Other Programs in This File: X2XMD

# Purpose

To copy data from the X register to mass storage with all parameters prompted for. X2MSD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value. X2MSD checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (main memory temporary storage register) + 1

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute X2MSD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy X to and press [R/S].

b. MSbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) mass storage registers to copy X to and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be loaded. If II is not specified, it will default to 1.

c. TMP RRR=?

Key in the main memory temporary storage register (RRR>2) to transfer data through and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

4. Follow the prompt:

a. R<u>rrr</u>=? (if not in ALPHA mode) OR R<u>rrr</u>? (if in ALPHA mode)

where

<u>**rrr**</u> = next register to copy X to.

Key in the numeric value (not in ALPHA mode) to be copied to mass storage register <u>rrr</u> and press [R/S]; OR Key in the alphanumeric value (in ALPHA mode - six characters or less) to be copied to register <u>rrr</u> and press [R/S].

Repeat this step until mass storage registers bbb through eee have been loaded with data.

5. When completed, the HP-41 will beep, stop, and display "DONE".

## Examples

 Copy the constants Pi, 1, and e<sup>1</sup> to mass storage file "D5-3A" (which already exists) registers 10 through 12. Use main memory register 10 for temporary storage.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
prelim	ninary -	"D5-3A" 13 [XEQ]"CREATE"	
1. 2. a. b. c.	MS FILE? MSbbb.eeeII=? TMP RRR=?	[XEQ]"SIZE"011 [XEQ]"X2MSD" D5-3A [R/S] 10.012 [R/S] 10 [R/S]	
3.			*WORKING*
4a.	R10=? R11=? R12=?	[Pi] [R/S] 1 [R/S] 1 [e <sup>×</sup> ] [R/S]	
5.			DONE

2. Copy the following table to every other register of mass storage file "D5-3B" (which doesn't exist), starting with register 05, in "column-major" form (i.e., column by column). Use main memory register 05 for temporary storage.

ID	Component
1	Diode
2	IC
3	Switch

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. 2. a.	MS FILE? MSbbb.eeeII=?	[XEQ]"SIZE"006 [XEQ]"X2MSD" D5-3B [R/S] 5.01502 [R/S]	
c.	TMP RRR=?	5 [R/S]	
3.			*WORKING*
a.	MS D5-3B SIZE=	? 16 [R/S]	*WORKING*
4a.	R5=?	1 [R/S]	
	R7=?	2 [R/S]	
	R9=?	3 [R/S]	
	R11=? [	ALPHA] DIODE [R/S]	]
	R13?	IC [R/S]	
	R15?	SWITCH [R/S]	
5.			DONE

- 1. It is acceptable to toggle back and forth (using the [ALPHA] key) between storage of numeric and alphanumeric data.
- 2. As Example 2 demonstrates, the minimum size required for a mass storage file = (MS bbb) + (number or X values to copy).
- 3. Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one may resolve the difficulty and restart the program.

B2MD (<u>Barcode TO Main memory</u>, <u>Data</u>)

Program No.: 41D6-1 Other Programs in This File: none

## Purpose

To copy data from barcode to main memory with all parameters prompted for. B2MD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value.

Minimum SIZE: (ending main memory register) + 1

#### Required Peripherals

Optical Wand (HP 82153A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute B2MD and follow the prompts:
  - a. MMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) main memory registers to copy barcode to and the register increment value II and press [R/S]. If eee is not specified, only register bbb will be loaded. If II is not specified, it will default to 1.

b. SCAN Rrrr

where

<u>rrr</u> = next register to copy barcode to.

Scan the data barcode to copy to main memory register <u>rrr</u>. Repeat this step until main memory registers bbb through eee have been loaded with data. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

3. When completed, the HP-41 will beep, stop, and display "DONE".

# **Examples**

1. Copy the data barcode below to main memory registers 10 through 12.



<u>Step</u>	Prompt	<u>User Input</u>	_Display_
1.		[XEQ]"SIZE"013	
2.		[XEQ]"B2MD"	
a.	MMbbb.eeeII=?	10.012 [R/S]	
b.	SCAN R10	(Scan 1st barcode)	
	SCAN R11	(Scan 2nd barcode)	
	SCAN R12	(Scan 3rd barcode)	
3.			DONE

# <u>Comments</u>

1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.

B2XMD (<u>Barcode TO</u> e<u>X</u>tended <u>Memory</u>, <u>D</u>ata)

Program No.: 41D6-2 Other Programs in This File: B2MSD

#### Purpose

To copy data from barcode to extended memory with all parameters prompted for. B2XMD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value. B2XMD checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

## Minimum SIZE: 002

#### Required Peripherals

Optical Wand (HP 82153A) Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute B2XMD and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy barcode to and press [R/S].

b. XMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) extended memory registers to copy barcode to and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be loaded. If II is not specified, it will default to 1.

- 3. If the named extended memory file doesn't exist, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. SCAN Rrrr

where

Scan the data barcode to copy to extended memory register <u>rrr</u>. Repeat this step until extended memory registers bbb through eee have been loaded with data. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

5. When completed, the HP-41 will beep, stop, and display "DONE".

# <u>Examples</u>

1. Copy the data barcode below to extended memory file "D6-2A" (which already exists) registers 10 through 12.



preliminary - Use X2XMD to create and load extended memory file "D6-2A" with size 13 or more.

1.		[XEQ]"SIZE"002
2.		[XEQ]"B2XMD"
a.	XM FILE?	D6-2A [R/S]
b.	XMbbb.eeeII=?	10.012 [R/S]
4a.	SCAN R10	(Scan 1st barcode)
	SCAN R11	(Scan 2nd barcode)
	SCAN R12	(Scan 3rd barcode)
5.		DONE

2. Copy the data barcode in Example 1 to every other register of extended memory file "D6-2B" (which doesn't exist), starting with register 05.

<u>Step</u>	Prompt	User Input	_Display_
1.		[XEQ]"SIZE"002	
2.		[XEQ]"B2XMD"	
a.	XM FILE?	D6-2B [R/S]	
b.	XMbbb.eeeII=?	5.00902 [R/S]	
3a.	XM D6-2B SIZE	=? 10 [R/S]	*WORKING*
4a.	SCAN R5	(Scan 1st barcode)	
	SCAN R7	(Scan 2nd barcode)	
	SCAN R9	(Scan 3rd barcode)	
5.			DONE

# Comments

1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.

- 2. As Example 2 demonstrates, the minimum size required for an extended memory file = (XM bbb) + (number of barcode registers to copy).
- 3. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.
USE THIS PAGE FOR NOTES

B2MSD (<u>Barcode IO Mass Storage</u>, <u>Data</u>)

Program No.: 41D6-3 Other Programs in This File: B2XMD

### Purpose

To copy data from barcode to mass storage with all parameters prompted for. B2MSD can copy numeric or alphanumeric data to one or more consecutive registers or registers separated by an increment value. B2MSD checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

<u>Minimum SIZE</u>: (main memory temporary storage register) + 1

### Required Peripherals

Optical Wand (HP 82153A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute B2MSD and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy barcode to and press [R/S].

b. MSbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) mass storage registers to copy barcode to and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be loaded. If II is not specified, it will default to 1.

c. TMP RRR=?

Key in the main memory temporary storage register (RRR>2) to transfer data through and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. SCAN Rrrr

where

<u>rrr</u> = next register to copy barcode to.

Scan the data barcode to copy to mass storage register <u>rrr</u>. Repeat this step until mass storage registers bbb through eee have been loaded with data. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

5. When completed, the HP-41 will beep, stop, and display "DONE".

# Examples

 Copy the data barcode below to mass storage file "D6-3A" (which already exists) registers 10 through 12. Use main memory register 10 for temporary storage.



a.	MS FILE?	D6-3A LR/SJ	
b.	MSbbb.eeeII=?	10.012 [R/S]	
с.	TMP RRR=?	10 [R/S]	
3.			*WORKING*
4a.	SCAN R10	(Scan 1st barcode)	
	SCAN R11	(Scan 2nd barcode)	
	SCAN R12	(Scan 3rd barcode)	
5.			DONE

 Copy the data barcode in Example 1 to every other register of mass storage file "D6-3B" (which doesn't exist), starting with register 05. Use main memory register 05 for temporary storage.

<u>Step</u>	Prompt	<u>User Input</u>	Display
1.		[XEQ]"SIZE"006	
2.		[XEQ]"B2MSD"	
a.	MS FILE?	D6-3B [R/S]	
b.	MSbbb.eeeII=?	5.00902 [R/S]	
с.	TMP RRR=?	5 [R/S]	
3.			*WORKING*
a.	MS D6-3B SIZE=	=? 10 [R/S]	*WORKING*
4a.	SCAN R5	(Scan 1st barcode)	
	SCAN R7	(Scan 2nd barcode)	
	SCAN R9	(Scan 3rd barcode)	
5.			DONE

### Comments

- 1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.
- 2. As Example 2 demonstrates, the minimum size required for a mass storage file = (MS bbb) + (number of barcode registers to copy).
- 3. Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file</u> SIZE=?" prompt. The problem will become apparent when the file size prompt is repeated or an error message is displayed. In either case, one can resolve the difficulty and restart the program.

USE THIS PAGE FOR NOTES

B2PD (<u>Barcode <u>TO</u> Printer, <u>D</u>ata)</u>

Program No.: 41D6-6 Other Programs in This File: B2DD

# Purpose

To copy data from barcode to the printer with all parameters prompted for. If the printer is not connected, B2PD operates like B2DD.

Minimum SIZE: 000

### Required Peripherals

Optical Wand (HP 82153A) Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

# Instructions

- Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 2. Execute B2PD and follow the prompts:
  - a. # DECIMALS=?

Key in the number of decimal places to be printed and press [R/S]. To default to 2 decimal places, just press [R/S].

b. SCAN BARCODE

Scan the data barcode to be printed. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

3. The data barcode value will be printed in the form

BCODE =  $\times \times \times \times \times$ ,

where

xxx.xx = value of data barcode. (In this case, # DECIMALS= 2.)

When completed, the HP-41 will beep, stop, and display "DONE". To start over at Step 2b, press [R/S].

## <u>Examples</u>

1. Print the values of the data barcode below accurate to four decimal places.



<u>Step</u>	Prompt	User Input	_Display_	Printer Output <u>(HP 82162A)</u>
1. 2.	( 5	Set printer to MAN [XEQ]"B2PD"	mode)	
a.	# DECIMALS=?	4 [R/S]		
3.	JUAN JRU		:)	BCODE=3.1416
			DONE	
		[R/S]		
2b.	SCAN SRC	(Scan 2nd barcode	e)	RCODE=09-41
5.			DONE	00002-00 41

# Comments

1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.

B2DD (<u>Barcode TO D</u>isplay, <u>D</u>ata)

Program No.: 41D6-7 Other Programs in This File: B2PD

# Purpose

To copy data from barcode to the display with all parameters prompted for. If the printer is connected, B2DD operates like B2PD.

Minimum SIZE: 000

# Required Peripherals

Optical Wand (HP 82153A)

# Instructions

- 1. Execute B2DD and follow the prompts:
  - a. # DECIMALS=?

Key in the number of decimal places to be displayed and press [R/S]. To default to 2 decimal places, just press [R/S].

b. SCAN SRC

Scan the data barcode to be displayed. If the HP-41 displays a wand error message (prefixed by "W:") or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

2. The data barcode value will be displayed in the form

BCODE = xxx.xx,

where

xxx.xx = value of data barcode. (In this case, # DECIMALS= 2.)

When completed, the HP-41 will beep, stop, and display "DONE". To start over at Step 1b, press [R/S].

# Examples

3.1416+96

-14-SO

 Display the values of the data barcode below accurate to four decimal places.

# 

# 

<u>Step</u>	Prompt	<u>User Input</u>	_Display_
1. a. b.	<pre># DECIMALS=? SCAN SRC</pre>	[XEQ]"B2DD" 4 [R/S] (Scan 1st barcode)	
2.		[0/0]	BCODE = 3.1416 DONE
16.	SCAN SRC	(Scan 2nd barcode)	
2.			BCODE = OS-41 DONE

# Comments

1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.

# D. Program Listings and Technical Details

# 1. Index of Numeric and Alphanumeric Data Movement Program Listings

The program listings on the following pages were printed by an HP 82162A Thermal Printer set to MAN mode. Use the table below to locate the page number of any OS-41 data movement program listing or barcode.

Table 2-2. Numeric and Alphanumeric Data Movement Program Listing and Barcode Locations.

			Page Nur	mber for			Page Nu	mber for
0S-41 Prog	gram	L	isting	Barcode	OS-41 Pro	gram	Listing	Barcode
M2MD	• •	•	193	405	MS2XD	• • •	204	419
M2XMD	• •	•	194	406	MS2PD		205	420
M2MSD		•	194	406	MS2DD		205	420
M2CD	••	•	195	408	C2MD		195	408
M2XD		•	196	409	C2XMD		203	417
M2PD	• •	•	197	410	C2MSD		203	417
M2DD	••	•	197	410	C2CD		206	422
XM2MD	••	•	198	411	C2PD		207	423
XM2XMD	• •	•	199	412	C2DD		207	423
XM2MSD		•	201	415	X2MD		208	425
XM2CD	• •	•	203	417	X2XMD		209	426
XM2XD	• •	•	204	419	X 2MSD		209	426
XM2PD		•	205	420	B2MD		210	428
XM2DD		•	205	420	B2XMD		211	429
MS2MD	• •	•	198	411	B2MSD		211	429
MS2XMD	• •	•	201	415	B2PD		212	431
MS2MSD		•	199	412	B2DD		212	431
MS2CD	••	•	203	417				

## 2. <u>The Data Movement Algorithm</u>

In most OS-41 data movement programs, copying data from one medium to another involves one HP-41 or module function. For example, to move a block of data from main memory to mass storage requires only the single function IL: WRTRX. In fact, single functions exist to copy data from a source medium to a destination medium whenever main memory is either the source or the destination. When this is the case, data movement is simple and can be illustrated as below, where the contents of main memory registers 00-02 are copied to extended memory data file registers 02-04 using the function XF: SAVERX. Note that M2XMD makes the low-level operations transparent to the user.

	MAIN MEMORY	M2XMD		EXTENDED MEMORY
Reg	Contents		Reg	Contents
00	main memory reg 00	)	00	unchanged
01	main memory reg 01		01	unchanged
02	main memory reg 02		02	main memory reg 00
03	main memory reg 03	Y	03	main memory reg Ol
04	main memory reg 04	(	04	main memory reg 02

All OS-41 programs that copy data to or from main memory (including, indirectly, the X register, barcode, the printer, and the display) move data in this fashion. Unfortunately, there exist no functions to directly transfer data from one medium to another when <u>neither</u> is main memory. Therefore, it is necessary to use main memory temporarily as a "channel" through which data pass. All the OS-41 programs that copy data from extended memory, mass storage, or magnetic

-187-

cards to extended memory, mass storage, or magnetic cards use this technique.

To insure that no valuable main memory information is lost, the OS-41 programs allow the user to specify the size and location of the main memory channel, a block of one or more consecutive registers positioned anywhere within the current SIZE limits. The only exceptions to this flexibility are the OS-41 programs that have magnetic cards as the source or destination medium; due to Card Reader limitations, the size of the channel must match the number of registers copied to or from magnetic cards. The user still specifies the beginning of the channel, of course.

If the size of the channel is less than the number of registers copied, the channel must be used repeatedly with smaller sections of data until all data are copied. This is illustrated below, where the contents of extended memory data file registers 00-09 are copied to mass storage data file registers 00-09 through main memory channel registers 05-08. The functions XF: GETRX and IL: WRTRX perform the actual data transfer. Note that XM2MSD makes the entire three-step low-level operation transparent to the user.



The OS-41 programs accomplish the multiple transfer process with the algorithm illustrated in the program segment below, condensed from XM2MSD:

01 RCL 03	29 LASTX
02 INT	30 INT
03 LASTX	31 STO 03
04 FRC	32 X=0?
05 1 E3	33 GTO 04
06 ×	34+LBL 03
07 X<>Y	35 RCL 02
<del>0</del> 8 -	36 GETRX
09 1	37 WRTRX
10 X<=Y?	38 X(0?
11 +	39 GTO 05
12 RCL 02	40 DSE 03
13 INT	41 GTO 03
14 LASTX	42+LBL 04
15 FRC	43 RCL 02
16 1 E3	44 INT
17 *	45 STO Y
18 X<>Y	46 RCL 04
19 -	47 X=0?
20 1	48 GTO 05
21 +	49 +
22 X(=0?	50 i
23 SIGN	51 -
24 ABS	52 1 E3
25 STO 04	53 /
26 /	54 +
27 FRC	55 CHS
28 ST* 04	56 STO 02
	57 GTO 03

Main memory register 03 contains the location of the beginning (bbb) and ending (eee) extended memory registers in the format bbb.eee, while register 04 contains the location of the beginning (bbb) mass storage register in the format bbb. Register 02 contains the location of the beginning (bbb) and ending (eee) temporary storage (channel) registers in the format bbb.eee.

- Lines 01-11 calculate how many extended memory registers are to be copied.
- Lines 12-24 calculate how many temporary storage registers are available.

- Lines 25-28 calculate how many extended memory registers will remain <u>after</u> as many as possible have been transferred using the entire block of temporary storage registers repeatedly. This value is stored in register 04.
- Lines 29-33 calculate how many times the entire block of temporary storages registers will be used to transfer data repeatedly. This value is stored in register 03.
- Lines 34-41 transfer data from extended memory to the main memory temporary storage registers and from the temporary storage registers to mass storage. The extended memory and mass storage pointers are automatically updated. This sequence is repeated as many times as calculated in lines 29-33. If the control value in register 02 is negative, signaling all data have been transferred, the program segment is exited. Otherwise,
- Lines 42-57 calculate the proper control value to complete the transfer of the remaining extended memory registers. This value is stored in register 02 and made negative to signal the end of data transfer in lines 34-41.

This data movement algorithm is used with only minor variations in all the OS-41 programs that require data movement through main memory.

#### 3. The VER Bug

The existence of a bug in the CR: VER function has been documented by Jarett (1983). Any card reader with one of the following headers (obtained by pressing [shift][CATALOG][2]) has the VER bug: CARD READER CARD RDR 1D CARD RDR 1E CARD RDR 1F

Executing the VER function of a card reader that has this bug will change the contents of the <u>first</u> register of an extended <u>memory</u> module (not an extended functions/memory module) in <u>port 2</u> of the HP-41. The VER bug will also affect an extended memory module either in port 4 of a port extender or built into a dual extended memory module.

The result of the VER bug varies with the portion of extended memory affected. If the register changed is located in a data or ASCII file, the contents of the file will be incorrect. If the register changed is located in a program file, the program, when copied to main memory, will have several incorrect steps. Finally, and perhaps worst of all, if the register changed is part of a file header, the entire extended memory directory will be disrupted, making many extended memory files inaccessible.

Jarett details a program in "HP-41 Extended Functions Made Easy" that replaces the VER function, preventing loss of extended memory data. Unfortunately, the procedure is necessarily complex and likely to cause damage itself if casually used. <u>Therefore, it is recommended</u> that the user simply avoid using the VER function while an extended memory module is in port 2. If necessary, the program XM2MSE, described in Chapter 4, can be used to save the entire contents of extended memory before removing an extended memory module to execute VER. Later, MS2XME can be used to restore the contents of extended memory.

-192-

# OS-41 Data Movement Program Listing: M2MD

Program Registers: 20 Bytes: 134

01+LBL "M2MD" 02 "41D1-1" 03 CF 21 04 FS? 50 05 GTO 00 06 "SRC bbb.eee=?" 07 PROMPT 08 CF 22 09 "DST bbb=?" 10 PROMPT 11 FC? 22 12 0 13 FC?C 22 14 VIEW X	51 DSE L 52 GTO 02 53 FS? 55 54 SF 21 55 •DONE• 56 ASTO X 57 BEEP 58 CLD 59 END
15 • +WORKING+•	
1741 BL AA	
18 X<>Y	
19 INT	
20 LASTX	
21 FRC	
22 1 E3	
23 *	
24 X()Y	
25 STO T	
26 -	
27 1	
28 +	
29 X(=0?	
30 SIGN	
31 HBS	
32 316N 27 000	
33 KUR 74 V/V3	
75 CTA 81	
36 LASTX	
37 ST+ Z	
38 +	
39 -1	
40 ST+ Z	
41 ST+ Y	
42 RDN	
43+LBL 01	
44 R†	
45+LBL 02	
46 RCL INU Z	
47 STU INU Z	
48 KUN	
49 514 Z	
31 + 16 <b>0</b> 6	

# OS-41 Data Movement Program Listing: M2XMD, M2MSD

Program Registers: 31 Bytes: 212

Giald: = MOVMD=
01*LUC HEAND 00 ##101.0#
02 4101-2 07 25 05
93 JF 93 AV CTO 88
N2♦FRF _USU2h.
N6 •4101-3*
07 CF 05
08+LBL 00
09 CF 21
10 FS? 50
11 GTO 01
12 "MM bbb.eee=?"
13 PROMPT
14 "XM"
15 FC? 05
16 MS*
17 "H FILE?"
18 AON
19 PROMPT
20 AOFF
21 ASTO Y
22 CF 22
23 "XM"
24 FC2 A5
25 - MS-
26 *F hbh=?*
20 1 000-: 27 DDANDT
21 FRUID 1 20 ECO 22
20 FU: 22 20 A
27 0 70 ECOC 00
30 FU(U 22 74 HTFU V
31 TIEM A 70 VEG 87
32 AEW 03 7741 Di Ai
33*LBL 81
34 LLH 75 0001 7
33 HKUL 2 34 OF OF
36 5F 20
37 F37 03 20 055VDT0
38 SEEKFTH
39 FU/ 80
40 SEEKK
41 FU/U 20
42 610 82
43 KUN
44 FS? 85
45 SHYERX
46 FC? 05
47 WRTRX
48 FC?C 05
49 VERIFY
50 FS? 55

51 SF 21 52 •DONE• 53 ASTO X 54 BEEP 55 CLD 56 RTN 57+LBL 02 58 TONE 0 59 \*XM \* 60 FC? 05 61 "MS " 62 ARCL Z 63 "H SIZE=?" 64 PROMPT 65 XEQ 03 66 CLA 67 ARCL T 68 FS? 05 69 CRFLD 70 FC? 05 71 CREATE 72 RDN 73 GTO 01 74+LBL 03 75 \* \*WORKING\*\* 76 AVIEW 77 END

Program Registers: 18 Bytes: 123

01+LBL \*M2CD\* 02 \*41D1-4\* 03 SF 05 04 GTO 00 05+LBL \*C2MD\* 06 \*41D4-1\* 07 CF 05 08+LBL 00 09 CF 21 10 FS? 50 11 GTO 01 12 "MM bbb.eee=?" 13 PROMPT 14+LBL 01 15 "INSERT " 16 FS? 05 17 "HDST" 18 FC? 05 19 "HSRC" 20 AVIEW 21 FS? 05 22 WDTAX 23 FC? 05 24 RDTAX 25 "DONE" 26 ASTO X 27 ""VER"IFY CDS" 28 FS? 05 29 AVIEW 30 FS? 55 31 SF 21 32 BEEP 33 FC?C 05 34 CLD 35 END

OS-41 Data Movement Program Listing: M2XD

Program Registers: 5 Bytes: 31

01+LBL "M2XD" 02 "41D1-5" 03 "MM RRR=?" 04 PROMPT 05 RCL IND X 06 BEEP 07 END 01+LBL "M2PD" 02 \*41D1-6\* 03+LBL "M2DD" 04 \*41D1-7\* 05 CF 29 06 FS? 50 07 GTO 00 08 CF 21 09 "MMbbb.eeeII=?" 10 PROMPT 11 CF 22 12 \*# DECIMALS=?\* 13 PROMPT 14 FC? 22 15.2 16 FC?C 22 17 VIEW X 18 \* \*WORKING\*\* 19 AVIEW 20+LBL 00 21 FC? 00 22 FS? 55 23 SF 21 24 ADV 25+LBL 01 26 FIX 0 27 °R° 28 X<>Y 29 INT 30 ARCL X 31 "+=" 32 FIX IND Y 33 RCL IND X 34 ARCL X 35 FS? 55 36 PRA 37 FC? 55 38 AVIEW 39 LASTX 40 RCL T 41 ISG Y 42 GTO 01 43 FC? 55 44 CF 21 45 ADV 46 "DONE" 47 ASTO X 48 BEEP 49 CLD 50 END

Program Registers: 21

Bytes: 145

# OS-41 Data Movement Program Listing: XM2MD, MS2MD

Program Registers: 26 Bytes: 179

	51 +
01+LBL "XM2MD"	51 V/-02
02 "41D2-1"	JZ AN-0:
03 SF 05	03 516M
04 GTO 00	04 HBS
05♦LBL "MS2MD"	55 +
96 *4103-1*	56 1
97 CE 95	57 -
98+1 RI 99	58 1 E3
80 CE 21	59 /
10 EC2 50	60 +
10 10: 00 11 CTO 01	61 FS? <b>8</b> 5
15 sV#x	62 GETRX
12 00	63 FC?C 05
13 FU? DJ 14 8008	64 READRX
14 "NƏ" 45 ml 51150m	65 FS? 55
10 "F FILE?"	66 SF 21
16 HUN	67 DONE
17 PRUMPT	68 ASTO X
18 AUFF	69 BEEP
19 ASTO X	70 CLD
20 "XM"	70 OCD 71 FND
21 FC? 05	ii che
22 <b>-H</b> S-	
23 "H bbb.eee=?"	
24 PROMPT	
25 CF 22	
26 "MM bbb=?"	
27 PROMPT	
28 FC? 22	
29 0	
30 FC2C 22	
31 VIEW X	
32 * *NORKING**	
34+1 BI 01	
75 010	
74 OPCI 7	
77 CTO 7	
70 V/\V	
30 AV71 70 EC3 05	
37 F32 0J 40 CEEKDTO	
40 DEENFIH	
41 FU? 00	
42 SEEKK	
43 INI	
44 LHSIX	
45 FRC	
46 1 E3	
47 *	
48 X<>Y	
49 -	
50 1	

# OS-41 Data Movement Program Listing: XM2XMD, MS2MSD

Program Registers: 84 Bytes: 582

		101 500
01+LBL "XM2XMD"	51 FC? 06	101 FRU 100 CT+ 07
02 •41D2-2•	52 GTU 02	102 31* 07
03 SF 05	53 RCL 08	103 LHSTX
04 GTO 00	54 FS? 05	104 IN!
05+LBL "MS2MSD"	55 GTO 10	105 STO 06
06 <b>*</b> 41D3-3*	56 CLA	106 X=0?
97 CE 95	57 ARCL 01	107 GTO 04
08+1 B! 00	58 <b>"</b> F, <b>"</b>	108 1
<b>A</b> 9 (F 21	59 ARCL 00	109 -
10 ES2 50	60 RENAME	110 1 E3
11 CTO 02	61 GTO 10	111 /
10 00 02	62+1 BL 02	112 STO 06
12 31 00 17 YEG 07	63 CL 9	113+LBL 03
13 AEW 07 14 OCTO 00	64 OPCI 01	114 CLA
14 HOLU 00 15 VCO AO	65 Ø	115 ARCL 00
10 AEW 00	60 0 66 CF 25	116 RCL 03
15 510 83	27 EC2 05	117 RCL 85
17 XEW 07	01 F3: 03 20 CEEVDTA	118 RCL 06
18 H510 Ø1	00 JEEKTIN 20 EC3 85	119 INT
19 XEQ 08	07 FU: 0J 70 CEEKD	120 *
20 STU 04	(0 SEENK 74 ECOC 05	121 +
21 RCL 00	71 FU/U ZJ 70 CTO 80	122 502 85
22 RCL 01	72 610 07 77 001 07	122 13: 03
23 X≠Y?	73 KUL 03	120 JULNIN 124 EC2 85
24 GTO 01	(4 IN)	124 FG: 00 195 CEEVD
25 CF 22	75 SIU 03	12J DEENK 102 DC1 00
26 "NEW SIZE=?"	76 LASTX	120 KUL 02 107 DC0 05
27 PROMPT	77 FRC	127 557 00
28 "SAME SIZE"	78 1 E3	128 GETRX
29 FC? 22	79 *	129 FU? 03
30 AVIEW	80 X<>Y	130 REHUKX
31 FC?C 22	81 -	131 ULH
32 GTO 01	82 1	132 ARCL 01
33 STO 08	83 X<=Y?	133 RCL 04
34 SF 06	84 +	134 RCL 05
35 ""	85 RCL 02	135 RCL 06
36 FS? 05	86 INT	136 INT
37 ASTO 01	87 LASTX	137 *
38 FC2 <b>8</b> 5	88 FRC	138 +
39 ASTO AA	89 1 E3	139 FS? 05
40+1 BI 01	90 *	140 SEEKPTA
41 "TMP hhh.eee=?"	91 X<>Y	141 FC? 05
42 PROMPT	92 -	142 SEEKR
47 STO 02	93 1	143 RCL 02
44 F92 96	94 +	144 FS? 05
45 Q	95 X(=0?	145 SAVERX
70 7 Al Erg Gl	96 SIGN	146 FC? 05
40 FU( 00 47 0	97 BBS	147 WRTRX
97 0 40 V\V0	98 STO 85	148 X(0?
90 A/I: 40 CTO 11	99 STO 87	149 GTO 05
47 GTU 11 50 VEC 12	100 /	150 ISG 06
50 XEQ 12	100 /	100 100 00

0S-41	Data	Movement	Program	Listing:	XM2XMD,	MS2MSD	(continued)	

151 GTO 03	201 CTO 05	251 EC2C 06
152+LBL 04	201 010 03 202 05TO 00	252 •F hhh=?•
153 RCL 02	202 HOTO 00	257 0
154 INT	200 010 01 204 Dri 00	255 5 254 DDOMPT
155 STO Y	204 KUL 00 205 1	254 (KONE)
156 RCL 07	200 1	230 FC20 22 252 VIEU V
157 ¥=0?	200 -	2JO TICH A 967 DTN
150 A-0. 150 CTO <b>05</b>	207 I E3	237 KIN 2504101 00
150 1	208 /	208+LBL 07
107 1	209 510 03	239 IUNE 0
100 1	210 INI	260 -051 -
	211 SIU <b>04</b>	261 HRUL 01
	212 RCL 02	262 F SIZE=?*
163 /	213 INT	263 PROMPT
164 +	214 ABS	264 XEQ 12
165 CHS	215 STO Y	265+LBL 10
166 STO 02	216 RCL 05	266 CLA
167 GTO 03	217 +	267 ARCL 01
168+LBL 05	218 1	268 FS? 85
169 FS? <b>0</b> 6	219 -	269 CRFLD
170 GTO 06	220 1 E3	270 FC? 05
171 CLA	221 /	271 CREATE
172 ARCL 01	222 +	272 GTO 02
173 FC?C 05	223 STO 02	273+LBL 11
174 VERIFY	224 RCL 08	274 "RAO-RO"
175 FS? 55	225 GTO 10	275 FS? 06
176 SF 21	226+LBL 97	276 • H8 USED*
177 "DONE"	227 *XH*	277 EC2 06
178 ASTO X	228 FC2 <b>R</b> 5	278 •F7 USED•
179 BEEP	220 **** 00 229 ****	279 DVTFU
180 CLD	230 FS2 06	280 TONE 0
181 RTN	271	200 TONE 0
182+LBL 06	201 1 0KC 272 EC2 04	201 10L 202 CTO 01
183 RCL 00	232 16: 00 277 *L DCT*	202 arto or 20741 Ri 12
184 RCI 01	200 F BOI 974 =L CTLE9=	203*LCC 12 204 = +U00VINC+=
185 X=Y?	234 F FILC: 275 000	204 THURNINGT
186 CE 06	230 HUN 974 DOMDT	203 HYIEW 204 END
187 X=Y2	236 FKUNF I	200 CMD
198 CTO <b>05</b>	237 HUFF 270 FC2 27	
199 (10	238 FU? 23	
199 CEN 199 ODCI 80	237 ULH	
101 EC2 05	240 FU? 23	
101 10: 00 100 DIIDEI	241 HKLL 00	
102 FOR C 107 ECO 85	242 FU?U 23	
104 DHDCE	243 HVIEW	
174 FURGE 105 DCI 00	244 RIN	
17J KUL 00 10/ * *	245+LBL 08	
170 107 OCTO V	246 "SRC"	
17( H3!U I 100 V-V2	247 FC? 06	
176 A=1: 100 CE 0/	248 DST"	
177 LF 105	249 FS? 06	
200 X=1/	250 "H bbb.eee=?"	

OS-41 Data Movement Program Listing: XM2MSD, MS2XMD

Program Registers: 55 Bytes: 380

	51 *	101 1
01+LBL "XM2MSD"	52 X<>Y	101 1
02 •41D2-3•	53 -	102 1 57
03 SF 05	54 1	103 1 C3 104 7
04 GTO 00	55 X(=Y2	104 /
05+LBL "MS2XMD"	56 +	103 T 102 CUC
06 <b>*</b> 41D3-2*	57 RCL A2	100 UN3 107 CTA 00
07 CF 05	58 INT	107 310 02 100 CTO 07
08+LBL 00	59 LASTX	100 GIU 03 1004) DI 05
09 CF 21	69 ERC	107VLDL 0J 110 CLO
10 FS? 50	61 1 53	110 CLH 111 ADC1 A1
11 GTO 02	62 *	111 MAGE DI 110 ECOC AS
12 SF 06	63 X<>Y	112 F3(0 0) 117 UEDIEV
13 XEQ 06	64 -	110 VERIFI 114 ECO 55
14 ASTO 00	65 1	114 FO: JJ 115 CE 01
15 XEQ 07	66 +	115 OF 21 117 *DONE*
16 STO 03	67 X(=92	116 DUNE 117 OCTO V
17 XEQ 06	68 SICN	117 HOLU A 110 DEED
18 ASTO 01	69 0BS	
19 XEQ 07	70 STO 04	117 ULU 100 DTU
20 STO 04	71 /	120 KIN 101-101 07
21+LBL 01	72 500	121+LBL 05
22 "TMP bbb.eee=?"	77 CT+ 04	122 "XM"
23 PROMPT	73 314 04 74 10CTV	123 FU? 85
24 STO 02	75 INT	124 <b>TH</b> S"
25 5	76 CTO 07	125 "F FILE?"
26 X>Y?	(0 310 03 77 V-00	126 HUN
27 GTO 09	(( A-0) 70 cto 04	127 PRUMPT
28 XEQ 10	(0 6!0 04 7041 DI 07	128 HUFF
29+LBL 02	(7#LBL 03 00 DCL 03	129 FC? 23
30 CLA	00 KUL 02 01 FC3 AF	130 CLH
31 ARCI A1	81 F57 80 00 CETRY	131 FC? 23
32 RCL 04	02 GE1KA 07 ECO AE	132 ARCL 00
33 SE 25	83 FL/ 83	133 FC?C 23
34 FS2 05	84 KEHUKA De rod de	134 AVIEW
35 SEEKR	83 F57 83 07 UDTDV	135 RTN
36 FC2 <b>0</b> 5	80 WK!KA 07 500 05	136+LBL 07
37 SEEKPTA	87 FU2 00 00 CONTRY	137 "XM"
38 FC2C 25	88 SHYEKA	138 FC? 05
39 610 08	89 X(0/	139 "MS"
40 018	90 GIU 05	140 FS? 06
41 9PCI 00	91 DSE 03	141 "⊢ bbb.eee=?"
42 RCI 93	92 610 83	142 FC?C 06
47 FS2 A5	93+LBL 04	143 "⊢ bbb=?"
44 SEEKPTO	94 KLL 02	144 0
45 EC2 05	95 INI	145 PROMPT
46 SEEKR	96 SIU Y	146 FC?C 22
47 INT	97 KCL 04	147 VIEW X
48 19572	98 X=0?	148 FC?C 05
49 FPC	99 GIU 85	149 SF 05
50 1 FZ	100 +	150 RTN
00 I L0		

OS-41 Data Movement Program Listing: XM2MSD, MS2XMD (continued)

151+LBL 08 152 TONE 0 153 **\*M**S \* 154 FC? 05 155 "XM " 156 ARCL 01 157 \* SIZE=?\* 158 PROMPT 159 XEQ 10 160 CLA 161 ARCL 01 162 FS? 05 163 CREATE 164 FC? 05 165 CRFLD 166 GTO 02 167+LBL 09 168 "R00-R04 USED" 169 AVIEW 170 TONE 0 171 PSE 172 GTO 01 173+LBL 10 174 \* \*WORKING\*\* 175 AVIEW 176 END

Program Registers - 54 Bytes: - 372

64 - 2 D2	51 SE 25	101VER-IFY CDS-
NI+FRF XWSFR	52 FS2 05	102 FS? 06
W2 *41U2-4*	57 CEFKPTO	103 AVIEW
03 SF 05	54 FC2 85	104 FS? 55
04 SF 06	55 CEEKP	105 SF 21
05 GTO 00	56 FC7 86	106 BEEP
06♦LBL •MS2CD•	57 CE 25	107 EC2C 06
<b>07 •41D3-4•</b>	50 ECOC 25	1 <b>0</b> 8 CID
08 CF 05	50 FC:C 20 50 FTA 85	109 RTN
09 SF 06	37 610 03 20 THT	110+1 RI 05
10 GTO 00	C1 LOCTV	111 TONE 0
11+LBL "C2XMD"	01 LH31A 69 EDC	112 *XM *
12 <b>*41D4-2*</b>	02 FRU 27 1 F7	117 FC2 05
13 SF 05	00 I C0 /4 +	114 - 149 -
14 CF 06	04 <b>*</b> /c 0/\0	115 DPC1 RR
15 GTO 00	63 X(7)	116 *F SI7E=7*
16+LBL "C2MSD"	66 - /7 9/_00	117 PROMPT
17 •4104-3•	D( A\-0; (0.0	118 YFO 07
18 CF 05	00 0 70 DCI 00	
19 CF 06	57 KUL 02 70 i	120 OPCI AA
20+LBL 00	(1) T 74 4 67	121 FS2 Ø5
21 CF 21	// I E0 70 /	122 CRELD
22 FS? 50	12 / 77 CT+ 00	123 FC2 <b>R</b> 5
23 GTO <b>0</b> 2	(0 0)T 02 74 DCi 09	124 CREATE
24 <b>*</b> XH*	(4 KUL 02 75 ECO 02	125 GTO 82
25 FC? <b>8</b> 5	(J F3: 00 7( CTA 07	12641 Ri 06
26 "MS"	(6 610 83 77 •INCEDT CDC+	127 - PAA-RA2 USED.
27 "H FILE?"	(/ INDER! DRU 70 OUTEU	128 AVIEN
28 AON	(0 HTICM 70 DBTOV	129 TONE Ø
29 PROMPT	(7 KUINO 00 VEN 07	130 PSE
30 AOFF	00 AEW 07 01 ECO 05	131 GTO A1
31 ASTO 00	01 FU: 00 09 COUEDY	132+LBL 87
32 <b>*X#</b> *	02 OMTERA 07 ECO 05	133 * *WORKING**
33 FC? 05	03 FU? 03 04 HDTDV	134 AVIEN
34 <b>"M</b> S"	04 MKIKA 05 (TR) G4	135 END
35 <b>"⊢ bbb.</b> eee=?"	0. 610 04 0(AID) 07	100 200
36 PROMPT	00¥LDL 00 07 ECO 05	
37 STO 01	00 CETRY	
38+LBL 01	00 UEIKA 00 ECO 05	
39 "TMP bbb=?"	07 FU? UJ 00 DEODDY	
40 PROMPT	70 KEHUKA 01 *INCEDT DET*	
41 STO 02	71 INDERI DDI 03 OVIEU	
42-3	72 NYICH 07 UNTOV	
43 X>Y?	20 MUTHA GALIDI GA	
44 GTO 06	05 CLO	
45 XEQ 07	70 ULM 02 ADM 80	
46+LBL 02	70 HKUL 00 07 ECOC 05	
47 CLA	21 FU:U UU QQ UEDTEV	
48 ARCL 00	20 TENII ( 00 TENII (	
49 RCL 01	100 OCTO V	
50 EC2 06	100 HOIU A	

### OS-41 Data Movement Program Listing: XM2XD, MS2XD

Program Registers: 25 Bytes: 173

Ø1+LBL "XM2XD" 02 \*4102-5\* 03 SF 05 04 GTO 00 05+LBL "MS2XD" 06 \*4103-5\* 07 CF 05 08+LBL 00 09 FS? 50 10 GTO 03 11 CF 21 12 "XM" 13 FC? 05 14 •MS• 15 \*+ FILE?\* 16 AON 17 PROMPT 18 AOFF 19 ASTO 00 20 "XM" 21 FC? 05 22 **\***#S\* 23 "H RRR=?" 24 PROMPT 25 STO 01 26 FS? 05 27 GTO 02 28+LBL 01 29 "TMP RRR=?" 30 PROMPT 31.2 32 X>Y? 33 GTO 04 34+LBL 02 35 \* \*WORKING\*\* 36 AVIEW 37+LBL 03 38 CLA 39 ARCL 00 40 RCL 01 41 FS? 05 42 SEEKPTA 43 FC? 05 44 SEEKR 45 RCL Z 46 FS? 05 47 GETX 48 FC? 05 49 READRX 50 FC?C 05

51 RCL IND X 52 FS? 55 53 SF 21 54 BEEP 55 RTN 56+LBL 04 57 -R00-R01 USED-58 AYIEW 59 TONE 0 60 PSE 61 GTO 01 62 END OS-41 Data Movement Program Listing: XM2PD, XM2DD, MS2PD, MS2DD

Program Registers: 44 Bytes: 307

01+LBL "XM2PD" 92 \*41D2-6\* 03+LBL "XM2DD" 04 \*41D2-7\* 05 SF 05 86 GTO 88 07+LBL \*MS2PD\* **08 \*41D3-6\*** 09+LBL "MS2DD" 10 \*4103-7\* 11 CF 05 12+LBL 00 13 CF 29 14 FS? 50 15 GTO 03 16 CF 21 17 CF 06 18 •XM• 19 FC? 05 20 MS\* 21 \*\* FILE?\* 22 AON 23 PROMPT 24 AOFF 25 ASTO 00 26 "XM" 27 FC? 05 28 "MS" 29 "Hbbb.eeeII=?" 30 PROMPT 31 STO 01 32 1 E3 33 🔹 34 FRC 35.01 36 X(Y? 37 SF 06 38 CF 22 39 \*# DECIMALS=?\* 49.2 41 PROMPT 42 FC?C 22 43 VIEW X 44 STO 02 45 FS? 05 46 GTO 02 47+LBL 01 48 \*TMP RRR=?\* 49 PROMPT 50 STO 03

51 4 52 X>Y? 53 GTO 07 54+LBL 02 55 \* \*WORKING\*\* 56 AVIEW 57+LBL 03 58 FC2 00 59 FS? 55 60 SF 21 61 ADV 62+LBL 04 63 CLA 64 ARCL 00 65 RCL 01 66 FS? 05 67 SEEKPTR 68 FC? 05 69 SEEKR 70+LBL 05 71 FC? 05 72 RCL 03 73 FS? 05 74 GETX 75 FC? 05 76 READRX 77 FIX 0 78 •R• 79 RCL 01 80 INT 81 ARCL X 82 "+=" 83 FIX IND 02 84 FS? 05 85 RDN 86 FC? 05 87 RCL IND 03 88 ARCL X 89 FS? 55 90 PRA 91 FC? 55 92 AVIEW 93 ISG 01 94 GTO 06 95 CF 05 96 CF 06 97 FC? 55 98 CF 21 99 ADV 100 "DONE"

101 ASTO X 102 BEEP 103 CLD 104 RTN 105+LBL 06 106 FS? 06 107 GTO 04 108 GTO 05 109+LBL 07 110 "R00-R03 USED" 111 AYIEW 112 TONE 0 113 PSE 114 GTO 01 115 END

### OS-41 Data Movement Program Listing: C2CD

Program Registers: 25 Bytes: 175

01+LBL "C2CD" 02 •41D4-4• 03+LBL 00 84 CF 21 05 FS? 50 06 GTO 01 07 CF 22 08 "TMP bbb=?" 89.0 19 PROMPT 11 FC?C 22 12 VIEW X 13 XEQ 04 14 X=0? 15-1 16.4 17 + 18 STO 00 19+LBL 01 20 RCL 00 21 ENTER† 22 "..." 23 ASTO X 24 SF 25 25+LBL 02 26 STO IND Y 27 ISG Y 28 FS? 25 29 GTO 02 30 RCL 00 31 "INSERT SRC" 32 AVIEW 33 RDTAX 34 XEQ 04 35 X<>Y 36 R1 37+LBL 03 38 RDN 39 RCL IND Y 40 ISG Z 41 X#Y? 42 GTO 03 43 RCL Z 44 INT 45 2 46 -47 1 E3 48 / 49 RCL 00 50 INT

51 + 52 "INSERT DST" 53 AVIEW 54 WDTAX 55 "DONE" 56 ASTO X 57 -- VER-IFY CDS-58 AYIEW 59 FS? 55 60 SF 21 61 BEEP 62 RTN 63 CF 21 64 XEQ 04 65 GTO 01 66+LBL 04 67 \* \* # MORKING\*\* 68 AVIEW 69 END

# OS-41 Data Movement Program Listing: C2PD, C2DD

Program Registers: 34 Bytes: 233

	51 DCi 7
01+LBL "C2PD"	JI KUL Z
02 •41D4-6•	JZ INI
Ø3♦LBL "C2DD"	53 2
04 •41N4-7•	54 -
05 05 00	55 1 E3
0J UF 27 07 FCD FD	56 /
05 F5/ 30	57 RCI 01
07 GIU 01	58 INT
08 CF 21	50 1
09 CF 22	07 T 20 ECO 00
10 *# DECIMALS=?*	00 FU: 00 /1 FCO 55
11 2	01 F3: JJ /0 CF 01
12 PROMPT	62 3F 21
13 FC?C 22	63 HUY
14 VIEW X	64 N
15 STO 00	65+LBL 04
16+1 BI 00	66 FIX 0
17 - THD MAN-7-	67 "R"
TO DOOMDT	68 ARCL X
10 FKUNFI 10 CTO 91	69 •⊢=•
19 510 61	70 FIX IND 00
20 2	71 ARCI IND Y
21 X>Y?	72 FS2 55
22 GTO 05	77 000
23 XEQ 06	10 FRH 74 ECO 55
24+LBL 01	74 FU? JJ 75 OUTEU
25.4	70 HVIEW
26 ST+ 01	76-1
27 RCL 01	77 +
28 ENTERT	78 ISG Y
20 cm	79 GTO <b>0</b> 4
70 OCTO V	80 FC? 55
30 H310 A 71 CE 35	81 CF 21
31 3F 23 Zow Di - 03	82 ADV
32*LBL 02	83 "DONE"
33 SIU IND T	84 ASTO X
34 ISG Y	85 REEP
35 FS? 25	
36 GTO 02	00 CLD 07 DTN
37 CF 21	00 A 101 05
38 RCL 01	00*LDL 0J 00 *000 DO1 UCCD*
39 "INSERT SRC"	89 -K00-K01 USED.
40 AVIEW	90 HVIEW
41 RDTAX	91 IUNE Ø
42 XEQ 06	92 PSE
43 X()Y	93 GTO <b>00</b>
44 R†	94+LBL 06
45+1 BL #3	95 * *WORKING**
	96 AVIEW
40 NUM 47 DOI TUB V	97 END
97 KUL IND ) 40 100 7	
48 156 Z	
49 X#Y?	
50 GTO 03	

```
OS-41 Data Movement Program Listing: X2MD
```

```
Program Registers: 12
Bytes: 82
```

01+LBL "X2MD" 02 \*4105-1\* 03 FIX 0 04 CF 29 05 FS? 50 06 GTO 00 97 "MMbbb.eeeII=?" 08 PROMPT 09+LBL 00 10 "R" 11 INT 12 ARCL X 13 LASTX 14 FC? 48 15 "⊦=" 16 **•**F?• 17 PROMPT 18 FC? 48 19 STO IND Y 20 FS? 48 21 ASTO IND X 22 FC? 48 23 RDN 24 ISG X 25 GTO 00 26 AOFF 27 "DONE" 28 ASTO X 29 BEEP 30 CLD 31 END

Program Registers: 43 Bytes: 296

01+LBL "X2XMD"
02 <b>*</b> 41D5-2*
03 SF 05
04 GTO 00
05+LBL "X2MSD"
06 *41D5-3*
07 CF 05
08+LBL 00
09 FIX 0
10 CF 21
11 CF 29
12 FS? 50
13 GTO 02
14 CF 06
15 <b>*XM</b> *
16 FC? 05
17 <b>"M</b> S"
18 •F FILE?•
19 AON
20 PROMPT
21 AOFF
22 ASTO 00
23 *XM*
24 FC? 05
25 MS
26 ⁺⊦bbb.eeeII=?*
27 PROMPT
28 STO 01
29 1 E3
30 *
31 FRC
32 .01
33 X(Y?
34 SF 06
35 FS? 05
36 GTO 02
37+LBL 01
38 TIMP RKK=?"
39 PRUMPT
48 510 82
41.3
92 X217 AD 6T0 92
43 610 85 44 950 67
99 ACM 0/ ASALDI DO
4J▼LOL 102 42 CLD
40 ULH 47 ODCI AA
41 NRUE 00 A0 Dri G1
49 CF 25
77 JE 20 50 ECO 05
00 (0) 00

101 PROMPT 102 XEQ 07 103 CLA 104 ARCL 00 105 FS? 05 106 CRFLD 107 FC? 05 108 CREATE 109 GTO 02 110+LBL 06 111 \*R00-R02 USED\* 112 AVIEW 113 TONE 0 114 PSE 115 GTO 01 116+LBL 07 117 \* \*WORKING\*\* 118 AVIEW 119 END

51 SEEKPTA

52 FC? 05

53 SEEKR

54 FC?C 25

55 GTO 05

56+LBL 03

58 RCL 01

60 ARCL X

61 FC? 48

64 PROMPT

65 FS? 48

66 ASTO X

67 FS? 05

68 SAVEX

69 FC? 05

70 STO IND 02 71 FC? 05 72 RCL 02 73 FC? 05 74 WRTRX 75 ISG 01 76 GTO 04 77 AOFF 78 CLA 79 ARCL 00 80 FC?C 05 81 VERIFY 82 CF 06 83 FS? 55 84 SF 21 85 "DONE" 86 ASTO X 87 BEEP 88 CLD 89 RTN 90+LBL 04 91 FS? 06 92 GTO 02 93 GTO 03 94+LBL 05 95 TONE 0 96 \*XM \* 97 FC? 05 98 MS \* 99 ARCL 00 100 "⊢ SIZE=?"

62 "⊦="

63 "H?"

57 "R"

59 INT

```
OS-41 Data Movement Program Listing: B2MD
```

```
Program Registers: 11
Bytes: 73
 01+LBL "B2MD"
 02 *41D6-1*
 83 FIX 0
 04 CF 21
 05 CF 29
 06 "MMbbb.eeeII=?"
 07 FC? 50
 08 PROMPT
 09+LBL 00
 10 "SCAN R"
 11 INT
 12 ARCL X
 13 AVIEW
 14 WNDDTX
 15 LASTX
 16 ISG X
 17 GTO 00
 18 FS? 55
 19 SF 21
 20 "DONE"
 21 ASTO X
 22 BEEP
 23 CLD
 24 END
```

# OS-41 Data Movement Program Listing: B2XMD, B2MSD

Program Registers: 44 Bytes: 304

01+1B( "B2XMD"
02 •41D6-2•
97 SE 95
94 CTO 99
9541 RI "R2MQTI"
06 = 41 DC_7=
00 4100 J 07 PC 05
0( UF 00 00410) 00
00 FIN 0
07 F1A 0
11 LF 23
12 CF 29
13 FS? 50
14 GTU 02
15 CF 06
16 "XM"
17 FC? 05
18 MS-
19 *F FILE?*
20 AON
21 PROMPT
22 AOFF
23 ASTO 00
24 "XM"
25 FC? 05
26 -MS-
27 ⁼⊦bbb.eeeII=?"
28 PROMPT
29 STO 01
30 1 E3
31 +
32 FRC
33 .01
74 X(Y)
35 SE 96
36 CF 27
77 EC2 05
70 CTA 82
30 GIU 02 7041 Di 01
37VLOL 01 AG =THD DDD-7=
AI DONNET
42 CTO 82
42 510 62
40 0 44 VV2
77 073: 45 CTO 04
40 610 00 AC VEN 07
40 AEV 01 A74101 00
47¥LDL 04 40 CLO
40 ULH 40 ODCI 00
47 HRUL 00 50 DC1 01
00 KUL 01

51 SF 25 52 FS? 05 53 SEEKPTA 54 FC? 05 55 SEEKR 56 FC?C 25 57 GTO 05 58+LBL 03 59 "SCAN R" 60 RCL 01 61 INT 62 ARCL X 63 FC? 05 64 RCL 02 65 AVIEW 66 FS? 05 67 WNDDTA 68 FC? 05 69 WNDDTX 70 FC? 05 71 CF 23 72 FS?C 23 73 ASTO X 74 FS? 05 75 SAVEX 76 FC? 05 77 WRTRX 78 ISG 01 79 GTO 04 80 CLA 81 ARCL 00 82 FC?C 05 83 VERIFY 84 CF 06 85 FS? 55 86 SF 21 87 DONE-88 ASTO X 89 BEEP 90 CLD 91 RTN 92+LBL 04 93 FS? 06 94 GTO 02 95 GTO 03 96+LBL 05 97 TONE 0 98 "XM " 99 FC? 05 100 "MS "

101 ARCL 00 102 "H SIZE=?" 103 PROMPT 104 XEQ 07 105 CLA 106 ARCL 00 107 FS? 05 108 CRFLD 109 FC? 05 110 CREATE 111 GTO 02 112+LBL 06 113 "R00-R02 USED" 114 AVIEW 115 TONE 0 116 PSE 117 GTO 01 118+LBL 07 119 \* \*WORKING\*\* 120 AVIEW 121 END

OS-41 Data Movement Program Listing: B2PD, B2DD

Program Registers: 20 Bytes: 136

01+LBL "B2PD" 02 "41D6-6" 03+LBL "B2DD" 04 \*41D6-7\* 05 CF 23 06 FS? 50 07 GTO 00 08 CF 21 09 CF 22 10 \*# DECIMALS=?\* 11.2 12 PROMPT 13 FC?C 22 14 VIEW X 15+LBL 00 16 FIX IND X 17+LBL 01 18 CF 21 19 "SCAN BARCODE" 20 AVIEW 21 WNDDTA 22 FS? 55 23 SF 21 24 FS? 23 25 ASTO X 26 ASHE 27 ASTO Y 28 ASHF 29 ASTO Z 30 ADV 31 •BCODE=• 32 ARCL X 33 FS? 23 34 ARCL Y 35 FS?C 23 36 ARCL Z 37 FS? 55 38 PRA 39 FC? 55 40 AVIEW 41 ADV 42 \*DONE\* 43 ASTO X 44 BEEP 45 CLD 46 RTN 47 GTO 01 48 END
USE THIS PAGE FOR NOTES

Chapter 3: THE MOVEMENT OF ASCII DATA

### A. Introduction

The difference between ASCII<sup>1/</sup> data, which is the subject of this chapter, and alphanumeric data, which is covered in Chapter 2, is only the manner in which it is stored and recalled. While alphanumeric data is stored in <u>registers</u> six characters long, ASCII data is stored in extended memory <u>records</u> of variable length. The Extended Functions/Memory Module provides many functions to manipulate ASCII data records and individual characters within records. ASCII data can be stored, recalled, moved, printed, or viewed, using a multitude of Extended Functions/Memory Module functions or OS-41 programs, as illustrated in Table 3-1.

Table 3-1 shows the common types of ASCII data movement along with the functions and/or programs to effect such movement. In Section B, additional details are given so the user can avoid most references to the HP-41 or peripheral manuals. <u>OS-41 program instructions</u> can be found in Section C.

Most of the cells in Table 3-1 contain two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

Since ASCII data can be stored in records only in extended memory,<sup>2/</sup> the OS-41 programs convert ASCII data to or from "pseudo-ASCII" alphanumeric data for operations with other media such as main memory and magnetic cards, using the pseudo-ASCII data format described in Section D.

-214-

<sup>1/</sup>ASCII is an acronym for American Standard Code for Information Interchange, a code that represents standard alphanumeric characters by integers.

 $<sup>2^{\</sup>prime}$  The exception to this rule is mass storage, which can contain ASCII files. The individual records, however, are inaccessible.

Table 3-1. Functions and Programs to Move ASCII Data Within the HP-41 System.

Err subsection Numeer       -1       -2       -3       -4       -5       -6       -7         Image: Construction Numeer       MIN       Errender (M)       MIN       Errender (M)       MIN       Errender (M)       MIN       Errender (M)       MIN       Errender (M)       MIN       France       MIN       France       MIN					_			
EXT SUBSECTION NUMBER         -1         -2         -3         -4         -5         -6           DESTLIANTION         MAIN         EXTENDED         MASS         MASS <th>-7</th> <td>DISPLAY (D)</td> <td>218 "SCROLL" (262)</td> <td>220 CX: ED "XM2DA" (240)</td> <td></td> <td></td> <td></td> <td></td>	-7	DISPLAY (D)	218 "SCROLL" (262)	220 CX: ED "XM2DA" (240)				
EEXT SUBSECTION NUMBER     -1     -2     -3     -4     -5       1     DESTINATION MAIN SOURCE     MAIN KEWORY     MASS REMORY	9-	PRINTER (P)	218 "SCROLL" (262)	220 "XM2PA" ( 23 8)				
EXT SUBSECTION NUMBER     -1     -2     -3     -4       1     DESTIMITION MAIN SOURCE     MAIN (M)     MAIN MEMORY     MASS (M)     MASS (M) <th>S-</th> <th>ALPHA REGISTER (A)</th> <th></th> <th>219 XF: GETREC XF: ARCLREC "XM2AA"(236)</th> <th></th> <th></th> <th></th> <th></th>	S-	ALPHA REGISTER (A)		219 XF: GETREC XF: ARCLREC "XM2AA"(236)				
EXT SUBSECTION NUMBER     -1     -2     -3       Image: Source construction main memory (m) construction memory (m) construction (m) constructio	-4	MAGNETIC CARDS (C)		219 "XM2CA" (232)				
TEXT SUBSECTION NUMBER     -1     -2       I     DESTINATION NEMORY SOURCE     MAIN NEMORY (M)     EXTENDED NEMORY (X)       1     MAIN NEMORY (M)     NAIN NEMORY (X)     EXTENDED NEMORY (X)       2     EXTENDED MAIN NEMORY (M)     218 (X)     218 (X)       2     EXTENDED NEMORY (X)     218 (X)     224)       3     MASS STORAGE (MS)     220 (Z256)     220 (Z24)       4     MAGNETIC CARDS     221 (C)     220 (Z24)       5     ALPHA REGISTER     221 (C)     221 (C)       6     BARCODE (B)     222 (C56)	ñ	MASS STORAGE (MS)		219 XF: SAVEAS "XM2MSE"(298) "XM2MSA"(230)				
IEXT SUBSECTION NUMBER     -1       I     DESTINATION     MAIN       NORCE     MAIN     MAIN       SOURCE     MORY (M)     MAIN       SOURCE     MORY (M)     218       Z     EXTENDED MEMORY (M)     218       MAIN     Z18     (226)       MASS STORAGE (MS)     218       MASS STORAGE (MS)     221       AMONETIC CARDS     221       MAGNETIC CARDS     221       MACODE (B)     MACODE (B)	-2	EXTENDED MEMORY (XM)	218 "M2XMA" (224)	218 CX: RESZFL "XM2XMA" (228)	220 XF: GETAS "MS2XME" (302) "MS2XMA" (242)	221 "C2XMA" (244)	221 XF: APPREC XF: APPCHR XF: INSREC XF: INSCHR XF: INSCHR CX: ED "TE" "A2XMA" (248)	222 "B2XMA" (250)
IEXT SUBSECTION NUMBER         I       DESTINATION         SOURCE       DESTINATION         I       MAIN MEMORY (M)         I       MASS STORAGE (MS)         I       MAGNETIC CARDS         I       ALPHA REGISTER         I       ALPHA REGISTER         I       BARCODE (B)	-1	MAIN MEMORY (M)		218 "XM2MA" (226)			221 "ALSTO" (256)	
	SUBSECTION NUMBER	DESTINATION	AIN MEMORY (M)	XTENDED MEMORY (XM)	IASS STORAGE (MS)	AGNETIC CARDS (C)	NLPHA REGISTER (A)	3ARCODE (B)
	техт	<u> </u>	-	ла С	m	4	ى ب	9

-215-

Although it is possible to copy pseudo-ASCII data from one medium to another without the use of extended memory by using the alphanumeric data movement programs detailed in Chapter 2, it is not recommended because such movement is very error prone. Instead, and as a general rule, ASCII data should always be processed using the ASCII data movement programs (which, of necessity, employ extended memory). For this reason, as Table 3-1 demonstrates, OS-41 programs have been prepared to move data only to or from extended memory. (The programs "ALSTO" and "SCROLL" are included because their usefulness warrants exclusion from this rule.) Of course, extended memory can be used as a middle step in movement between two other media. For example, to move ASCII data from the ALPHA register to mass storage, use OS-41 program "A2XMA" followed by "XM2MSA".

To insure simplicity of use, the ASCII data movement programs detailed in Section C use an append method for transfering data to extended memory. That is, all ASCII data moved to extended memory are appended to any existing records in the destination file. $\frac{3}{2}$ 

Note that although ASCII data are stored in extended memory <u>records</u>, the size of an extended memory ASCII file is specified in <u>registers</u>; each extended memory register assigned to ASCII data can hold approximately six characters. Furthermore, just as main memory data registers are numbered from zero, so are extended memory ASCII records and characters within records. Thus, the character numbered 5 in the record numbered 03 is actually the <u>sixth</u> character in the <u>fourth</u> record.

<sup>3/</sup>Mass storage is the exception to the append rule. Because of the way the XF: GETAS function works, any existing data in the destination extended memory file will be lost.

### A Procedure to Insert Records into an ASCII File

While the append method for ASCII data movement will suffice for most needs, on occasion it is necessary to <u>insert</u> ASCII records into an existing file. The following procedure, implementing only OS-41 programs, can be used for ASCII record insertion.

#### <u>Step</u> Instructions

 If the ASCII records to be inserted (the <u>source</u> records) reside in extended memory and are a subset of a larger file, copy them to extended memory dummy file "DUM1" with XM2XMA, then use XMZ<sup>4/</sup> to delete all non-source records from before and/or after the source records in "DUM1".

CAUTION: Delete the records after the source records <u>first</u>, then delete the records before the source records to prevent the source records from being renumbered and possibly inadvertently deleted.

If the source ASCII records do <u>not</u> reside in extended memory as a subset of a larger file, copy them to extended memory dummy file "DUM1" using M2XMA, XM2XMA, MS2XMA, C2XMA, A2XMA, or B2XMA, depending on where the data reside.

- 2. Make a duplicate of the extended memory file into which ASCII records are to be inserted (the <u>destination</u> file) under the name "DUM2" using XM2XMA.
- 3. Delete all the records <u>before</u> which the source records are to be inserted from the destination file using XMZ. If the source records are to be inserted at the beginning of the destination file, specify the "CLEAR ALL" option in XMZ.
- 4. Delete all the records <u>after</u> which the source records are to be inserted from the file "DUM2" using XMZ.
- 5. Use XM2XMA to append the source record file "DUM1" to the destination file.
- 6. Use XM2XMA to append the file "DUM2" to the destination file.
- 7. Purge the dummy files "DUM1" and "DUM2" using XMX.4/

 $\frac{4}{\text{See}}$  Chapter 5 for instructions for XMZ and XMX.

# B. Functions, Programs, and Their Use

# 1-2.5/ MAIN MEMORY to EXTENDED MEMORY

a. To copy ASCII data from main memory to extended memory with all parameters prompted for, use the OS-41 program

"M2XMA" (<u>Main memory TO eXtended Memory</u>, <u>ASCII</u>).

- 1-6. MAIN MEMORY to PRINTER (see Subsection 1-7)
- 1-7. MAIN MEMORY to DISPLAY
  - a. To display (or print if printer is connected) pseudo-ASCII data stored in successive main memory data registers to simulate prose (see subsection 5-1), use the program "SCROLL" described by Beers (1980). For a refined version of "SCROLL" and program instructions, see Section D.
- 2-1. EXTENDED MEMORY to MAIN MEMORY
  - a. To copy ASCII data from extended memory to main memory with all parameters prompted for, use the OS-41 program

"XM2MA" (eXtended Memory TO Main memory, ASCII).

- 2-2. EXTENDED MEMORY to EXTENDED MEMORY
  - a. HP-41CX users can use the function

CX: RESZFL (working file; X:<u>+nnn</u>)<sup>6/</sup>

to change the size of the working extended memory ASCII file to nnn. If nnn is positive <u>and</u> a reduction in file size will erase part of a record, "FL SIZE ERR" will be displayed and the file size won't change. This protective feature can be deactivated by specifying a negative nnn.

b. To increase or decrease the size of an extended memory file, or to copy ASCII data from one extended memory file to another, with all parameters prompted for, use the OS-41 program

"XM2XMA" (eXtended Memory TO eXtended Memory, ASCII).

 $<sup>\</sup>frac{5}{\text{Subsection numbers are keyed to Table 3-1 and indicate movement from row number (given first) to column number.$ 

 $<sup>\</sup>frac{6}{}$ Throughout this text, the parameters needed in the stack registers and ALPHA (indicated by A) register are indicated in parentheses.

- 2-3. EXTENDED MEMORY to MASS STORAGE
  - a. Use IL: <u>CREATE</u> $\frac{1}{2}$  (X:nnn; A:filename)

to create a mass storage file of nnn registers with the indicated file name, then use

XF: SAVEAS (A:XM filename,MS filename OR A:XM filename)

to copy the specified extended memory ASCII file to mass storage under the specified file name (or the extended memory file name if none is specified).

b. To copy the entire contents of extended memory (program, data, and ASCII files) to a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"XM2MSE" (eXtended Memory <u>TO Mass Storage</u>, <u>E</u>ntire).

Refer to Chapter 4 for details.

c. To copy ASCII data from extended memory to mass storage with all parameters prompted for, use the OS-41 program

"XM2MSA" (eXtended Memory TO Mass Storage, ASCII).

- 2-4. EXTENDED MEMORY to MAGNETIC CARDS
  - a. To copy ASCII data from extended memory to magnetic cards with all parameters prompted for, use the OS-41 program

"XM2CA" (eXtended Memory <u>TO</u> magnetic <u>Cards</u>, <u>ASCII</u>).

- 2-5. EXTENDED MEMORY to ALPHA REGISTER
  - a. To recall to the ALPHA register the contents of record rrr starting at character ccc of the named extended memory file, first use either

XF: <u>SEEKPTA</u> (X:rrr.ccc; A:filename) or

XF: SEEKPT (X:rrr.ccc)

to set the pointer at extended memory position rrr.ccc, then use either

XF: GETREC or

XF: ARCLREC

 $<sup>\</sup>mathcal{U}$  An underline is used throughout this text to indicate that the function is the <u>logical</u> one to use to make the file named in the ALPHA register the "working file" (see the Extended Functions/Memory Module Owner's Handbook, page 23).

to replace (GETREC) or append to (ARCLREC) the ALPHA register. The pointer is automatically advanced to the next character in the record.

b. To copy ASCII data from extended memory to the ALPHA register with all parameters prompted for, use the OS-41 program

"XM2AA" (eXtended Memory TO ALPHA register, ASCII).

2-6. EXTENDED MEMORY to PRINTER

Although the "printer" is stressed in all printer subsections, the discussion applies generally to a TV monitor if an HP-IL Video Interface is connected.

a. To copy ASCII data from extended memory to the printer with all parameters prompted for, use the OS-41 program

"XM2PA" (eXtended Memory TO Printer, ASCII).

- 2-7. EXTENDED MEMORY to DISPLAY
  - a. HP-41CX users can use the function

CX: ED (A:filename)

to display and change the contents of an ASCII file. Refer to the HP-41CX Owner's Manual, pages 228-234, for details about the use of ED.

b. To copy ASCII data from extended memory to the display with all parameters prompted for, use the OS-41 program

"XM2DA" (eXtended Memory <u>TO D</u>isplay, <u>A</u>SCII).

- 3-2. MASS STORAGE to EXTENDED MEMORY
  - a. Use XF: <u>CRFLAS</u> (X:nnn; A:filename)

to create an extended memory ASCII file of nnn registers  $\frac{8}{}$  with the indicated file name, then use

XF: GETAS (A:MS filename,XM filename OR A:MS filename)

to copy the specified mass storage ASCII file to extended memory under the specified file name (or the mass storage file name if none is specified).

 $<sup>\</sup>frac{8}{1}$  The Extended Functions/Memory Module Owner's Handbook suggests the following formula to estimate the minimum number of registers required for an ASCII file: nnn = (# of characters \* 1.2)/7.

b. To recall the entire contents of extended memory (program, data, and ASCII files) from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2XME" (<u>Mass Storage TO</u> e<u>Xtended Memory</u>, <u>Entire</u>).

The previous contents of extended memory will be lost. Refer to Chapter 4 for details.

c. To copy ASCII data from mass storage to extended memory with all parameters prompted for, use the OS-41 program

"MS2XMA" (Mass Storage TO eXtended Memory, ASCII).

- 4-2. MAGNETIC CARDS to EXTENDED MEMORY
  - a. To copy ASCII data from magnetic cards to extended memory with all parameters prompted for, use the OS-41 program

"C2XMA" (magnetic <u>Cards TO</u> e<u>X</u>tended <u>Memory</u>, <u>A</u>SCII).

- 5-1. ALPHA REGISTER to MAIN MEMORY
  - a. To facilitate the storage of pseudo-ASCII data in successive main memory data registers to simulate prose, use the program "ALSTO" described by Beers (1980). "ALSTO" is normally used to store the data which are then recalled in narrative fashion by the program "SCROLL" (see subsection 1-7). For these programs and instructions, see Section D.
- 5-2. ALPHA REGISTER to EXTENDED MEMORY
  - a. Use XF: <u>CRFLAS</u> (X:nnn; A:filename)

to create an extended memory ASCII file of nnn registers (see footnote 8) with the indicated file name, then use either

(1) XF: APPREC (A:text)

to append the contents of the ALPHA register to the end of the working file as a new record, or

(2) XF: <u>SEEKPTA</u> (X:rrr.ccc; A:filename), or

XF: SEEKPT (X:rrr.ccc)

to move the pointer to character ccc in record rrr of the named (or working) extended memory file, followed by

XF: APPCHR (A:text)

to append the contents of the ALPHA register to the end of record rrr, or

XF: INSREC (A:text)

to insert the contents of the ALPHA register before record rrr as a new record, or

XF: INSCHR (A:text)

to insert the contents of the ALPHA register before character ccc in record rrr.

b. HP-41CX users can use the function

CX: ED (A:filename)

to change and display the contents of an ASCII file. Refer to the HP-41CX Owner's Manual, pages 228-234, for details about the use of ED.

- c. The text editor program "TE" by Jarett (1983) emulates the HP-41CX function ED, but doesn't require the use of an HP-41CX. For more details, refer to "HP-41 Extended Functions Made Easy", by Jarett.
- d. To copy ASCII data from the ALPHA register repeatedly to extended memory with all parameters prompted for, use the OS-41 program

"A2XMA" (ALPHA register TO eXtended Memory, ASCII).

6-2. BARCODE to EXTENDED MEMORY

a. To copy ASCII data from barcode to extended memory with all parameters prompted for, use the OS-41 program

"B2XMA" (Barcode TO eXtended Memory, ASCII).

# C. ASCII Data Movement Program Instructions

The program instructions found on the following pages should enable the user to apply the desired program. One or more examples are provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 3-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings and an explanation of the pseudo-ASCII data format.

M2XMA (<u>Main memory IO</u> e<u>X</u>tended <u>Memory</u>, <u>A</u>SCII)

Program No.: 41A1-2 Other Programs in This File: none

# <u>Purpose</u>

To copy ASCII data from pseudo-ASCII form in main memory to extended memory with all parameters prompted for. M2XMA checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: whatever was required by XM2MA

# Required Peripherals

Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Execute M2XMA and follow the prompts:
  - a. MM bbb=?

Key in the beginning main memory register bbb containing pseudo-ASCII data and press [R/S]. To default to register 00, just press [R/S].

b. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy the pseudo-ASCII data to and press [R/S].

- The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press  $[{\sf R}/{\sf S}]$  .

3. The HP-41 will display "\*WORKING\*" as the named extended memory file is created (if necessary) and pseudo-ASCII data are copied from main memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

### <u>Examples</u>

3.

3.

1. Copy the pseudo-ASCII data starting in main memory register 10 to extended memory file "Al-2A" (which already exists).

 

 Step
 Prompt
 User Input
 Display

 preliminary - Use A2XMA to create and load extended memory file "A1-2A", then use XM2MA to copy "A1-2A" to main memory starting at register 10.

 1.
 [XEQ]"M2XMA" a. MM bbb=?

 10
 [R/S] b. XM FILE?

> \*WORKING\* DONE

\*WORKING\*

DONE

 Copy the pseudo-ASCII data in main memory register 00 to extended memory file "Al-2B" (which doesn't exist).

<u>User Input</u> <u>Display</u> Step Prompt preliminary - Use A2XMA to create and load extended memory file "Al-2AT" with size 5, then use XM2MA to copy "Al-2AT" to main memory starting at register 00. [XEQ]"M2XMA" 1. [R/S]a. MM bbb=? 0.00 b. XM FILE? A1-2B [R/S] 2. \*WORKING\* XM A1-2B SIZE=? 5 [R/S] a.

#### Comments

- 1. Refer to Section D for a description of the pseudo-ASCII data format.
- 2. To insure complete data transfer, the minimum size of a new extended memory file is the size of the file copied by XM2MA.

XM2MA (eXtended Memory TO Main memory, ASCII)

Program No.: 41A2-1 Other Programs in This File: none

# Purpose

To copy ASCII data from extended memory to pseudo-ASCII form in main memory with all parameters prompted for.

Minimum SIZE: (beginning main memory register) + 1.1 \* (number of characters/7), approximately

# Required Peripherals

Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2MA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to main memory and press [R/S].

b. MM bbb=?

Key in the beginning main memory register bbb to copy the first ASCII datum to and press [R/S]. To default to register 00, just press [R/S].

3. The HP-41 will display "\*WORKING\*" as ASCII data are copied from extended memory to main memory. When completed, the HP-41 will beep, stop, and display "DONE".

# Examples

1. Copy the ASCII data in extended memory file "A2-1A" to main memory registers 10 and up.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use A2XMA to create and load extended memory file "A2-1A" with no more than 24 characters.

1.		[XEQ]"SIZE"014 (	(or greater)
2.		[XEQ]"XM2MA"	Ū
a.	XM FILE?	A2-1A [R/S]	
b.	MM bbb=?	10 [R/S]	
3.			*WORKING*
			DONE

# <u>Comments</u>

- 1. Refer to Section D for a description of the pseudo-ASCII data format.
- 2. The actual SIZE required is highly dependent upon the actual ASCII data structure, and increases with the # records/# characters ratio. If "NONEXISTENT" is displayed during XM2MA, set a higher SIZE and press [R/S] to continue.

XM2XMA (eXtended Memory <u>TO</u> eXtended Memory, <u>A</u>SCII)

Program No.: 41A2-2 Other Programs in This File: none

#### Purpose

To increase or decrease the size of an extended memory file, or to copy ASCII data from one extended memory file to another, with all parameters prompted for. XM2XMA checks if the named destination extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: 004; 005 if changing file size

### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2XMA and follow the prompts:
  - a. XM SRC FILE?

Key in the name of the source extended memory file (six characters or less) and press [R/S].

b. XM DST FILE?

Key in the name of the destination extended memory file (six characters or less) to copy the source ASCII data to and press [R/S]. To default to the name of the source extended memory file, just press [R/S].

- 3. If the extended memory source and destination file names specified are the same, follow the prompt:
  - a. NEW SIZE=?

Key in the new size of the source extended memory file and press [R/S].

- 4. The HP-41 will display "\*WORKING\*" as it checks if the named destination extended memory file exists. If it doesn't, follow the prompt:
  - a. DST <u>file</u> SIZE=?

where

<u>file</u> = destination extended memory file name.

Key in the size of the destination extended memory file to be created and press [R/S].

5. The HP-41 will display "\*WORKING\*" as the named destination extended memory file is created (if necessary) and ASCII data are copied from extended memory to extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

### Examples

Step

Prompt

 Copy the ASCII data in extended memory file "A2-2A" to extended memory file "A2-2B" (which doesn't exist).

User Input

Display

preliminary - Use A2XMA to create and load extended memory file "A2-2A" with size 5.

1. 2. a. b.	[XEQ]"SIZE"004 [XEQ]"XM2XMA" XM SRC FILE? A2-2A [R/S] XM DST FILE? A2-2B [R/S]	
4. a. 5.	DST A2-2B SIZE=? 5 [R/S]	*WORKING* *WORKING*

2. Increase the size of extended memory file "A2-2C" from 10 registers to 15 registers.

<u>Step Prompt User Input Display</u>

preliminary - Use A2XMA to create and load extended memory file "A2-2C" with size 10.

1.		[XEQ]"SIZE"005	
2.		[XEQ]"XM2XMA"	
a.	XM SRC FILE?	A2-2C [R/S]	
b.	XM DST FILE?	[R/S]	A2-2C
3a.	NEW SIZE=?	15 [R/S]	
4,5.			*WORKING*
			DONE

### Comments

- 1. If the destination extended memory file is not large enough to hold the entire contents of the source file, as much data as possible will be copied and then XM2XMA will end normally.
- When an extended memory file size is changed, XM2XMA temporarily creates a dummy file of the new size. Therefore, a number of extended memory registers equal to the new size must be available to change a file size.
- 3. When an extended memory file size is changed, XM2XMA temporarily creates a dummy file named "...". If a file by this name already exists, XM2XMA will stop with an error message.

XM2MSA (eXtended Memory <u>TO</u> Mass <u>S</u>torage, <u>A</u>SCII)

Program No.: 41A2-3 Other Programs in This File: MS2XMA

### Purpose

To copy ASCII data from extended memory to mass storage with all parameters prompted for. XM2MSA checks if the named mass storage file exists and, if necessary, creates it with the user-specified file size.

### Minimum SIZE: 002

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2MSA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to mass storage and press [R/S].

b. MS FILE?

Key in the name of the mass storage file (six characters or less) to copy extended memory ASCII data to and press [R/S]. To default to the name of the extended memory file, just press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, follow the prompt:
  - a. MS <u>file</u> SIZE=?

where

<u>file</u> = mass storage file name.

Key in the size of the mass storage file to be created and press [R/S]. To default to the size of the named extended memory file, just press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named mass storage file is created (if necessary) and ASCII data are copied from extended memory to mass storage. When completed, the HP-41 will beep, stop, and display "DONE".

# Examples

 Copy the ASCII data in extended memory file "A2-3A" to mass storage file "A2-3B" (which doesn't exist), using the same file size for the mass storage file.

Prompt User Input Display Step preliminary - Use A2XMA to create and load extended memory file "A2-3A" with size 5. 1. [XEQ]"SIZE"002 [XEQ]"XM2MSA" 2. A2-3A [R/S] a. XM FILE? b. MS FILE? A2-3B [R/S] \*WORKING\* 3. MS A2-3B SIZE=? [R/S] 5.00 a. \*WORKING\* 4.

 Copy the ASCII data in extended memory file "A2-3C" to mass storage file "A2-3C" (which already exists).

DONE

<u>Step</u>	Prompt	<u>User Input</u>	_Display_
prelim	inary – Use A2X file "A steps:	(MA to create and 1 A2-3C" with size 5. "A2-3C" 5 [XEQ]"CREATE"	load extended memory , Then follow these
1. 2. a. b. 3,4.	XM FILE? MS FILE?	[XEQ]"SIZE"002 [XEQ]"XM2MSA" A2-3C [R/S] [R/S]	A2-3C *WORKING* DONE

### Comments

 Since flag 25 (error ignore flag) is used to detect if the named mass storage file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "MS <u>file</u> SIZE=?" prompt. The problem will become apparent when an error message is displayed, indicating a need to resolve the difficulty and restart the program. XM2CA (eXtended Memory <u>IQ</u> magnetic <u>Cards</u>, <u>ASCII</u>)

Program No.: 41A2-4 Other Programs in This File: none

# Purpose

To copy ASCII data from extended memory to pseudo-ASCII form on magnetic cards with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + 1.1 \* (number of characters/7), approximately

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) Card Reader (HP 82104A)

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2CA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to magnetic cards and press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>1) and press [R/S].

- 3. The HP-41 will display "\*WORKING\*", then prompt:
  - a. INSERT DST

Insert track 1 of destination magnetic card 1 into the Card Reader.

- 4. If more than one track is required, follow the prompt:
  - a. RDY tt of nn

where

 $\underline{tt}$  = next track to be recorded, and  $\underline{nn}$  = total number of tracks required. Insert track  $\underline{tt}$  into the Card Reader and repeat this step

until <u>tt</u> = <u>nn</u>.

- 5. When all the destination magnetic cards have been recorded, the HP-41 will beep and prompt:
  - a. "VER"IFY CDS1/

Press [XEQ]"VER".

b. CARD

Insert each track recorded into the Card Reader. If the track was recorded properly, the HP-41 will display "TYPE D TR  $\underline{tt}$ ",

where

 $\underline{tt} = track number.$ 

If the HP-41 displays "CARD ERR" or "CHECKSUM ERR", press [<-] and start over at Step 2, perhaps with new magnetic cards.

 When all the destination magnetic cards have been verified, press [<-]. The HP-41 will display "DONE".</li>

### <u>Examples</u>

-----

1. Copy the ASCII data in extended memory file "A2-4A" to a magnetic card. Use main memory registers 05 and up for temporary storage.

<u>Step Prompt User Input Display</u>

preliminary - Use A2XMA to create and load with two 12-character records extended memory file "A2-4A" with size 5.

1. 2. a.	XM FILE?	[XEQ]"SIZE"012 [XEQ]"XM2CA" A2-4A [R/S]	
ь. З.	TMP bbb=?	5 LR/SJ	*WORKING*
a. 5a.	INSERT DST "VER"IFY CDS	(Insert track 1) [XEQ]"VER"	
b.		(Insert track 1)	TYPE D TR 01
6.	en le	[<-]	DONE

1/Before using the CR: VER function, refer to Chapter 2, Section D for an explanation of the VER bug that can alter extended memory.

# Comments

- 1. Refer to Section D for a description of the pseudo-ASCII data format.
- 2. The actual SIZE required is highly dependent upon the actual ASCII data structure, and increases with the # records/# characters ratio. If "NONEXISTENT" is displayed during XM2CA, set a higher SIZE and press [R/S] to continue.
- 3. The destination magnetic cards may be inserted in any order in Steps 3, 4, and 5; the HP-41 automatically keeps track of where data reside on the cards.

.

USE THIS PAGE FOR NOTES

XM2AA (eXtended Memory <u>IO</u> <u>A</u>LPHA register, <u>A</u>SCII)

Program No.: 41A2-5 Other Programs in This File: none

# <u>Purpose</u>

To copy ASCII data from extended memory to the ALPHA register with all parameters prompted for.

Minimum SIZE: 002

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2AA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be copied to ALPHA and press [R/S].

b. XM RRR.CCC=?

Key in the extended memory record RRR and beginning character CCC to be copied to ALPHA and press [R/S]. Remember that records and characters are numbered from zero.

3. The HP-41 will copy up to 24 ASCII characters from record RRR to ALPHA, starting with character CCC. When completed, the HP-41 will beep, stop, and display RRR.CCC in the X register. To view the ASCII data copied to the ALPHA register, press [ALPHA].

### Examples

 Copy the ASCII data in extended memory file "A2-5A" record 03 to the ALPHA register, starting at character 5, the <u>sixth</u> character in the string.

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use A2XMA to create and load extended memory file "A2-5A". Store the string "ABCDEFGHIJKL" in record 03.

1.		[XEQ]"SIZE"002	
2.		[XEQ]"XM2AA"	
a.	XM FILE?	A2-5A [R/S]	
b.	XM RRR.CCC=?	3.005 [R/S]	
3.			3.005
		[ALPHA]	FGHIJKL

USE THIS PAGE FOR NOTES

XM2PA (eXtended Memory <u>IO</u> Printer, ASCII)

Program No.: 41A2-6 Other Programs in This File: XM2DA

#### Purpose

To copy ASCII data from extended memory to the printer with all parameters prompted for. XM2PA can print one or more consecutive records. If the printer is not connected, XM2PA operates like XM2DA.

Minimum SIZE: 001

Required Peripherals

Extended Functions/Memory Module (HP 82180A) Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Set the printer to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 3. Execute XM2PA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to print and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) and ending (eee) extended memory records to print and press [R/S]. If eee is less than bbb, only register bbb will be printed.

4. The HP-41 will display "\*WORKING\*" and print each record in the form

 $R_{xx} = text$  (if the beginning of a record) OR + $R_{xx} = text$ , (if the continuation of a record)

where

 $\underline{xx}$  = record number being printed, and  $\underline{text}$  = ASCII data in record  $\underline{xx}$ .

5. When completed, the HP-41 will beep, stop, and display "DONE".

1. Print the contents of extended memory file "A2-6A" records 03 through 06. Printer Output Step Prompt User Input Display (HP 82162A)

preliminary - Use A2XMA to create and load extended memory file "A2-6A" records 00 through 06; store strings longer than 24 characters in records 04 and 05.

1.		[XEQ]"SIZE"001	
2.	(Set	printer to MAN mode)	
3.		[XEQ]"XM2PA"	
a.	XM FILE?	A2-6A [R/S]	
b.	XM bbb.eee=?	3.006 [R/S]	
4.		*WORKING*	
			R3=text

R4=text +R4=more text R5=text +R5=more text R6=text

5.

DONE

XM2DA (eXtended Memory TO Display, ASCII)

Program No.: 41A2-7 Other Programs in This File: XM2PA

### <u>Purpose</u>

To copy ASCII data from extended memory to the display with all parameters prompted for. XM2DA can display one or more consecutive records. If the printer is connected, XM2DA operates like XM2PA.

Minimum SIZE: 001

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2DA and follow the prompts:

a. XM FILE?

Key in the name of the extended memory file (six characters or less) to display and press [R/S].

b. XM bbb.eee=?

Key in the beginning (bbb) and ending (eee) extended memory records to display and press [R/S]. If eee is less than bbb, only register bbb will be displayed.

 The HP-41 will display "\*WORKING\*", then display each record in the form

 $R_{xx} = text$  (if the beginning of a record) OR + $R_{xx} = text$ , (if the continuation of a record)

where

 $\underline{xx}$  = record number being displayed, and  $\underline{text}$  = ASCII data in record  $\underline{xx}$ .

4. When completed, the HP-41 will beep, stop, and display "DONE".

### <u>Examples</u>

1. Display the contents of extended memory file "A2-7A" records 03 through 06.

<u>Step Prompt User Input Display</u> preliminary - Use A2XMA to create and load extended memory file "A2-7A" records 00 through 06; store strings longer than 24 characters in records 04 and 05. 1. [XEQ]"SIZE"001 [XEQ]"XM2DA" 2. XM FILE? A2-7A [R/S] a. XM bbb.eee=? 3.006 [R/S] b. 3. \*WORKING\* R3 = textR4 = text+R4 = textR5 = text

 $+R5 = \frac{\text{text}}{\text{R6}} = \frac{\text{text}}{\text{text}}$ 

DONE

4.

#### Comments

1. To stop the HP-41 after each display, set flag 00 before executing XM2DA. Press [R/S] after each display to continue.

MS2XMA (<u>Mass Storage IO</u> e<u>X</u>tended <u>M</u>emory, <u>A</u>SCII)

Program No.: 41A3-2 Other Programs in This File: XM2MSA

# Purpose

To copy ASCII data from mass storage to extended memory with all parameters prompted for. MS2XMA checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

# Minimum SIZE: 002

### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2XMA and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be copied to extended memory and press [R/S].

b. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy mass storage ASCII data to and press [R/S]. To default to the name of the mass storage file, just press [R/S]. Note that, because of the way ASCII data are stored in mass storage, any existing data in the specified extended memory file will be lost.

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S].

4. The HP-41 will display "\*WORKING\*" as the named extended memory file is created (if necessary) and ASCII data are copied from mass storage to extended memory. When completed, the HP-41 will beep, stop, and display "DONE". 1. Copy the ASCII data in mass storage file "A3-2A" to extended memory file "A3-2B" (which doesn't exist).

Prompt User Input Display Step

preliminary - Use A2XMA to create and load extended memory file "A3-2A" with size 5, then use XM2MSA to copy "A3-2A" to mass storage under the same name.

1.		[XEQ]"SIZE"002	
2.		[XEQ]"MS2XMA"	
a.	MS FILE?	A3-2A [R/S]	
b.	XM FILE?	A3-2B [R/S]	
3.			*WORKING*
a.	XM A3-2B SIZE=?	5 [R/S]	
4.			*WORKING*
			DONE

2. Copy the ASCII data in mass storage file "A3-2C" to extended memory file "A3-2C" (which already exists), replacing its previous contents.

<u>Step</u>	Prompt	User Input	Display
prelin	ninary – Use A2 file ' copy '	2XMA to create an 'A3–2C" with size 'A3–2C" to mass s	d load extended memory 5, then use XM2MSA to torage under the same name
1. 2. a.	MS FILE?	[XEQ]"SIZE"00 [XEQ]"MS2XMA" A3-2C [R/S] [R/S]	2 A3-20
3,4.			*WORKING* DONE

#### Comments

- 1. Since flag 25 (error ignore flag) is used to detect if the named extended memory file exists, it is possible that other errors (i.e., named file exists but is too small) will cause the "XM file SIZE=?" prompt. The problem will become apparent when an error message is displayed, indicating a need to resolve the difficulty and restart the program.
- 2. The loss of existing destination extended memory ASCII data inherent in the use of XF: GETAS can be prevented by using MS2XMA to copy the source ASCII data to a temporary extended memory file, followed by XM2XMA to append the ASCII data in the temporary file to the destination extended memory ASCII file.

C2XMA (magnetic <u>Cards TO</u> e<u>X</u>tended <u>Memory</u>, <u>A</u>SCII)

Program No.: 41A4-2 Other Programs in This File: none

# Purpose

To copy ASCII data in pseudo-ASCII form on magnetic cards to extended memory with all parameters prompted for. C2XMA checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

# Required Peripherals

Card Reader (HP 82104A) Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute C2XMA and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy magnetic cards to and press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>1) and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. INSERT SRC

Insert track 1 of source magnetic card 1 into the Card Reader.

a. RDY <u>tt</u> of <u>nn</u>

where

 $\underline{tt}$  = next track to be read, and  $\underline{nn}$  = total number of tracks required.

Insert track <u>tt</u> into the Card Reader and repeat this step until <u>tt</u> = <u>nn</u>.

6. When all the source magnetic cards have been read, the HP-41 will display "\*WORKING\*" as pseudo-ASCII data are copied from the main memory temporary storage registers to extended memory, then beep, stop, and display "DONE".

#### Examples

 Copy the pseudo-ASCII data on a magnetic card to extended memory file "A4-2A" (which already exists). Use main memory registers 05 and up for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Use A2XMA to create and load with two
 12-character records extended memory file

 "A4-2A" with size 10, then use XM2CA to
 copy "A4-2A" to a magnetic card.

 1.
 [XEQ]"SIZE"012

 2.
 [XEQ]"C2XMA"

 a.
 XM FILE?
 A4-2A [R/S]

b. TMP bbb=? 5 [R/S]
3. \*WORKING\*
4a. INSERT SRC (Insert track 1)
6. \*WORKING\*
DONE

 Copy the pseudo-ASCII data on a magnetic card to extended memory file "A4-2B" (which doesn't exist). Use main memory registers 02 and up for temporary storage.

Step Prompt User Input <u>Display</u> preliminary - Use A2XMA to create and load with two 12-character records extended memory file "A4-2BT" with size 10, then use XM2CA to copy "A4-2BT" to a magnetic card. 1. [XEQ]"SIZE"009 2. [XEQ]"C2XMA" XM FILE? A4-2B [R/S] a. TMP bbb=? 2 [R/S] b. \*WORKING\* 3. XM A4-2B SIZE=? 10 [R/S] \*WORKING\* a. 4a. INSERT SRC (Insert track 1) \*WORKING\* 6. DONE

### Comments

- 1. Refer to Section D for a description of the pseudo-ASCII data format.
- 2. The actual SIZE required is highly dependent upon the actual ASCII data structure, and increases with the # records/# characters ratio. If "NONEXISTENT" is displayed during C2XMA, set a higher SIZE and press [R/S] to continue.
- The source magnetic cards may be inserted in any order in Steps 4 and 5; the HP-41 automatically keeps track of where data reside on the cards.

USE THIS PAGE FOR NOTES

A2XMA (ALPHA register <u>TO</u> eXtended <u>Memory</u>, ASCII)

Program No.: 41A5-2 Other Programs in This File: none

### Purpose

To copy ASCII data from the ALPHA register repeatedly to extended memory with all parameters prompted for. A2XMA checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

#### Minimum SIZE: 001

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute A2XMA and follow the prompt:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy ALPHA to and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists and determines the next record to be loaded. If the file doesn't exist, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

# 4. Follow the prompt:

a. REC <u>rrr</u>? (if the beginning of a record)
OR
+REC <u>rrr</u>? (if the continuation of a record)

where

<u>rrr</u> = next record to copy ALPHA data to.
Key in up to 24 characters (signalled by the ALPHA-full tone) to be copied to the beginning of (REC <u>rrr</u>?) or appended to (+REC <u>rrr</u>?) record <u>rrr</u> and press [R/S]; OR To proceed to the next record, press [R/S] when "+REC <u>rrr</u>?" is prompted. Repeat this step until finished, then press [R/S] when "REC <u>rrr</u>?" is prompted, to stop data entry.

5. When completed, the HP-41 will beep, stop, and display "DONE".

#### Examples

1. Copy the strings "HELLO", "ABC...XYZ", and "GOODBYE" to extended memory file "A5-2A" (which doesn't exist) records 00, 01, and 02.

<u>Step</u>	Prompt	<u>User Input</u>	_Display_
1.		[XEQ]"SIZE"001	
2.		[XEQ]"A2XMA"	
a.	XM FILE?	A5-2A [R/S]	
3.			*WORKING*
a.	XM A5-2A SIZE=?	10 [R/S]	*WORKING*
4a.	REC 0?	HELLO [R/S]	
	+REC 0?	[R/S]	
	REC 1?	ABCDEFGHIJKLMNOF	PORSTUVWX [R/S]
	+REC 1?	YZ [R/S]	
	+REC 1?	[R/S]	
	REC 2?	GOODBYE [R/S]	
	+REC 2?	[R/S]	
	REC 3?	[R/S]	
5.			DONE

2. Copy the strings "HI AGAIN" and "THE END" to the end of extended memory file "A5-2A" (which already exists from Example 1).

<u>Step</u>	Prompt	User Input	<u>Display</u>
1. 2.		[XEQ]"SIZE"001 [XEQ]"A2XMA"	
a. 3.	XM FILE?	A5-2A LR/SJ	*WORKING*
4a.	REC 3? +REC 3? REC 4? +REC 4? REC 5?	HI AGAIN [R/S] [R/S] THE END [R/S] [R/S] [R/S]	
5.			DONE

B2XMA (<u>Barcode TO</u> e<u>X</u>tended <u>Memory</u>, <u>A</u>SCII)

Program No.: 41A6-2 Other Programs in This File: none

#### Purpose

To copy ASCII data from ALPHA barcode to extended memory with all parameters prompted for. B2XMA checks if the named extended memory file exists and, if necessary, creates it with the user-specified file size.

Minimum SIZE: 001

#### Required Peripherals

Optical Wand (HP 82153A) Extended Functions/Memory Module (HP 82180A)

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute B2XMA and follow the prompt:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to copy ALPHA barcode to and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as it checks if the named extended memory file exists. If it doesn't, follow the prompt:
  - a. XM <u>file</u> SIZE=?

where

<u>file</u> = extended memory file name.

Key in the size of the extended memory file to be created and press [R/S]. The HP-41 will display "\*WORKING\*" as the file is created.

- 4. Follow the prompt:
  - a. SCAN REC <u>rrr</u> (if the beginning of a record)
     OR
     SCAN +REC <u>rrr</u> (if the continuation of a record)

where

 $\underline{rrr}$  = next record to copy ALPHA barcode to.

Scan the ALPHA barcode to copy to extended record <u>rrr;</u> OR To proceed to the next record, press [R/S] <u>twice</u> when "SCAN +REC <u>rrr</u>" is prompted. Repeat this step until finished, then press [R/S] <u>twice</u> when "SCAN REC <u>rrr</u>" is prompted, to stop data entry. If the HP-41 displays a wand error message (prefixed with "W:"), or doesn't respond to a scan, try again, perhaps varying scan speed or wand angle.

5. When completed, the HP-41 will beep, stop, and display "DONE".

#### <u>Examples</u>

 Copy the ALPHA barcode below to extended memory file "A6-2A" (which doesn't exist) records 01 (first barcode) and 02 (second and third barcode).

# 

# 

<u>Step</u>	Prompt	<u>User Input</u>	_Disp]ay_
1.	[	XEQ]"SIZE"001	
2.	[	XEQ]"B2XMA"	
a.	XM FILE?	16-2A [R/S]	
3.			*WORKING*
a.	XM A6-2A SIZE=? 1	LO [R/S]	*WORKING*
4a.	SCAN REC 0 (Sc	an 1st barcode)	
	SCAN +REC 0	[R/S][R/S]	
	SCAN REC 1 (Sc	an 2nd barcode)	
	SCAN +REC 1 (Sc	an 3rd barcode)	
	SCAN +REC 1 [	[R/S][R/S]	
	SCAN REC 2	[R/S][R/S]	
5.			DONE

### Comments

1. The Optical Wand Owner's Manual recommends protecting barcode while scanning with one of the transparent sheets supplied with the Wand.

# D. Program Listings and Technical Details

### 1. Index of ASCII Data Movement Program Listings

The program listings on the following pages were printed by an HP 82162A Thermal Printer set to MAN mode. Use the table below to locate the page number of any OS-41 ASCII data movement program listing or barcode.

Table 3-2. ASCII Data Movement Program Listing and Barcode Locations.

\_\_\_\_

0S-41 Progi	Pag ram Listin	e Number for g Barcode
M2XMA		432
SCROLL	269	433
XM2MA		434
XM2XMA		435
XM2MSA		437
XM2CA	273	439
XM2AA		440
XM2PA	275	441
XM2DA		441
MS2XMA		437
C2XMA		442
ALSTO		444
A2XMA	278	445
B2XMA	279	447

#### 2. The Pseudo-ASCII Data Format

The OS-41 programs that convert ASCII data to or from pseudo-ASCII data for storage in media other than extended memory use a format inspired by Jarett (1983). In this format, ASCII records are broken into groups of up to six characters and stored in consecutive medium registers. The end of each record is indicated by a string of less than six characters in a register. If the number of characters in a record is a multiple of six, the end of that record is indicated by the null string (""). A register containing the null string at the beginning of a new record indicates the end of the file.

As an example, consider the conversion of an extended memory ASCII file to pseudo-ASCII data in main memory registers 00 and up, using XM2MA as illustrated below.

		— XM2MA -		
		MA	IN MEMORY	
Record	Contents		Reg	Contents
00	OS-41		00	"OS-41"
01	AN OPERATING SYSTEM FOR THE HP-41		01	"AN OPE"
02	BY T.W. AND T.W. BEERS		02	"RATING"
		I	03	" SYSTE"
			04	"M FOR "
			05	"THE HP"
			06	"-41"
			07	"BY T.W"
			80	". AND "
			09	"T.W. B"
			10	"EERS"
			11	""

In this example, the end-of-record markers are the short (fewer than six characters) strings in registers 00, 06, and 10. The end of the file is indicated by the null string in register 11.

#### 3. Instructions for ALSTO and SCROLL

The programs ALSTO and SCROLL make possible the storage and recall of alphanumeric characters strung together to simulate ASCII records. Using these programs, any narrative of reasonable length can be conveniently entered and copied to main memory in a pseudo-ASCII format (ALSTO) and subsequently displayed or printed (SCROLL) in a continuous fashion much like the appearance of scrolling ASCII records. Historically, the concept and development of the original programs represented a casual joint effort (in 1979) of the co-authors of the present text. These results were formally described by the elder co-author (Thomas W. Beers, 1980) who cited the independent but similar efforts of Meyer (1980). The refined versions of ALSTO and SCROLL described in the following pages were written by the younger co-author, Ted. W. Beers. USE THIS PAGE FOR NOTES

ALSTO (<u>ALphanumeric data STO</u>rage)

#### Purpose

To accomplish the automatic storage of alphanumeric prose (greetings, messages, limericks, etc.) entered in groups of 24 characters.

<u>Minimum SIZE</u>: (beginning main memory register) + 4\*INT(# of characters/24), approximately

Required Peripherals: none

#### Instructions

- Insure that the minimum SIZE is set. If the required SIZE is underestimated, leading subsequently to a "NONEXISTENT" message, use this procedure to recover:
  - (a) Press [ALPHA] to exit ALPHA mode
  - (b) Press [XEQ]"SIZE"nnn, where nnn = a larger SIZE
  - (c) Press [ALPHA] to enter ALPHA mode
  - (d) Press [R/S] to continue with ALSTO
- 2. Execute ALSTO and follow the prompts:
  - a. MM bbb=?

Key in the beginning main memory register bbb to copy prose to and press [R/S].

b. KEY IN WORDS

Key in up to 24 characters (signalled by the ALPHA-full tone) to be copied to main memory and press [R/S]. If the ALPHA-full tone is heard, make sure that no more characters are entered before pressing [R/S] or the first character(s) will be lost.

c. MORE? OR R/S

Key in up to 24 more characters to be copied to main memory as in Step 2b and press [R/S].

Repeat this step until the prose has been completely entered, then press [R/S] immediately when "MORE? OR R/S" is prompted, to stop data entry. Note that filling the end of the last prose line with spaces is <u>not</u> necessary; however the deliberate entry of end-of-line spaces is occasionally useful in any line to improve the appearance of the output <u>printed</u> by SCROLL. 3. When completed, the HP-41 will stop and display

bbb.eee=bbb.eee,

where

<u>bbb</u> = beginning main memory register containing prose, and <u>eee</u> = ending main memory register containing prose.

To execute SCROLL (automatically specifying the prose just entered), press [R/S].

#### Examples

1. Store the Preamble to the Constitution of the United States in main memory registers starting with 10. In case the user has forgotten, it goes like this:

We, the people of the United States, in order to form a more perfect union, establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessings of liberty to ourselves and our posterity, do ordain and establish this Constitution for the United States of America.

<u>Step</u>	Prompt	<u>User Input</u>	Display
1.		[XEQ]"SIZE"066	
2.		[XEQ]"ALSTO"	
a.	MM bbb=?	10 [R/S]	
b.	KEY IN WORDS	WE, THE PEOPLE OF THE UN[R/S]	
с.	MORE? OR R/S	ITED STATES, IN ORDER TO[R/S]	
	MORE? OR R/S	FORM A MORE PERFECT UNI[R/S]	
	MORE? OR R/S	ON, ESTABLISH JUSTICE, I[R/S]	
	MORE? OR R/S	NSURE DOMESTIC TRANQUILI[R/S]	
	MORE? OR R/S	TY, PROVIDE FOR THE COMM[R/S]	
	MORE? OR R/S	ON DEFENSE, PROMOTE THE [R/S]	
	MORE? OR R/S	GENERAL WELFARE, AND SEC[R/S]	
	MORE? OR R/S	URE THE BLESSINGS OF LIB[R/S]	
	MORE? OR R/S	ERTY TO OURSELVES AND OU[R/S]	
	MORE? OR R/S	R POSTERITY, DO ORDAIN A[R/S]	
	MORE? OR R/S	ND ESTABLISH THIS CONSTI[R/S]	
	MORE? OR R/S	TUTION FOR THE UNITED ST[R/S]	
	MORE? OR R/S	ATES OF AMERICA.[R/S]	
	MORE? OR R/S	[R/S]	
3.			bbb.eee=10.06
		[R/S]	(The Preamble
			will scroll t

5 :0 completion in about 3.25 minutes.)

2. I would like the calculator, when turned on by someone else, to display the following message:

If you have found this calculator, please be informed that it belongs to T. W. Beers, phone (317) 463-0807. If you've stolen it, may the "bad bugs" be with you.

To facilitate this, store the message in main memory registers starting with 70. (For the HP-41 to display the message when turned on, see the Comments section of the SCROLL instructions.) Note that since parentheses, apostrophe and quotation marks are non-standard ALPHA symbols I will substitute "< >", "-", and ":", respectively. Users having an Extended Functions/Memory Module or an HP-41CX can use an alternative, described in the Comments section.

Step	Prompt	<u>User Input</u>	Display
1.		[XEQ]"SIZE"098	
2.		[XEQ]"ALSTO"	
a.	MM bbb=?	70 [R/S]	
b.	KEY IN WORDS	IF YOU HAVE FOUND THIS C[R/S]	
с.	MORE? OR R/S	ALCULATOR, PLEASE BE INF[R/S]	
	MORE? OR R/S	ORMED THAT IT BELONGS TO[R/S]	
	MORE? OR R/S	T. W. BEERS, PHONE <317[R/S]	
	MORE? OR R/S	> 463-0807. IF YOU-VE ST[R/S]	
	MORE? OR R/S	OLEN IT, MAY THE :BAD BU[R/S]	
	MORE? OR R/S	GS: BE WITH YOU.[R/S]	
	MORE? OR R/S	[R/S]	
3.			bbb.eee=70.097
		[R/S]	(The message will scroll to

completion in about 1.5 minutes.)

#### Comments

1. Characters not available on the ALPHA keyboard, such as quotation marks, apostrophes and parentheses, can be generated easily during ALSTO with the function XF: XTOA, which makes use of the decimal equivalent of such characters, reproduced below for convenience. Note that  $[R\downarrow]$  must be pressed after each use of XTOA to restore the stack.

Code	ASCII	Display	Code	ASCII	Display	Code	ASCII	Display	Code	ASCII	Display
0		-	32	space		64	0	e	96	•	Ŧ
1		7	33	. !	; ·	65	Ă	я	97	а	Cu
2		8	34	"	12	66	в	В	98	b	Ь
3		8	35	#	Ħ	67	С	E	99	с	c
4		X	36	\$	5	68	D	Ð	100	d	d
5		Х	37	%	96	69	Е	E	101	е	L
6		7	38	&	2	70	F	F	102	f	8
7			39	,	,	71	G	5	103	g	8
8		<b>8</b>	40	(	<	72	н	Н	104	h	8
9		8	41	)	;	73	I	Ι	105	i	8
10		8	42	*	¥	74	J	ل_ا	106	j	器
11		8	43	+	÷	75	К	ĸ	107	k	×
12		'n,	44	,	,	76	L	<u> </u>	108	I.	8
13		á	45	-		77	М	11	109	m	8
14		8	46	•	•	78	Ν	N	110	n	8
15		8	47	/	1	79	0		111	0	88
16		æ	48	0		80	Ρ	P	112	р	鼦
17		8	49	1	1	81	Q	G	113	q	8
18		æ	50	2	2	82	R	R	114	r	22
19		8	51	3	Ξ	83	S	5	115	S	8
20		8	52	4	Ļ	84	Т	7	116	t	88
21		8	53	5	5	85	U	1_1	117	u	8
22		ES .	54	6	5	86	V	1	118	v	88
23		83	55	7	7	87	W	14	119	w	2
24			56	8	8	88	Х	X	120	x	88
25			57	9	9	89	Y	Y	121	У	E.S.
26		E.S	58	:		90	Z	ź	122	z	33
27		13	59	;	,	91	[	<u> </u>	123	{	20
28		8	60	<	<i>2</i>	92	\	`	124	1	36
29		2	61		-	93	]		125	}	8
30		13	62	>	<u>`</u>	94	^	<i>,</i> <b>'</b>	126	$\sim$	Ĺ
31		8	63	?	Ċ	95	_	-	127		+-

**Character Codes** 

Thus, in Example 2 one could have placed parentheses around the area code, an apostrophe in the contraction, and quotes around BAD BUGS by the following procedure:

- a. Enter the first three lines as usual, then enter data as follows.
- b. T. W. BEERS, PHONE [ALPHA]40[XEQ]"XTOA"[R↓][ALPHA][shift] [APPEND]317

- c. [R/S]
- d. [CLA][ALPHA]41[XEQ]"XTOA"[R+][ALPHA][shift][APPEND] 463-0807. IF YOU[ALPHA]39[XEQ]"XTOA"[R+][ALPHA][shift][APPEND]VE ST
- e. [R/S]
- f. OLEN IT, MAY THE [ALPHA]34[XEQ]"XTOA"[R+][ALPHA][shift][APPEND]
  BAD BU
- g. [R/S]
- h. GS[ALPHA]34[XEQ]"XTOA"[R↓][ALPHA][shift][APPEND] BE WITH YOU.
- i. [R/S]
- j. Complete the ALSTO program by pressing [R/S]; press [R/S] again to scroll the message, which is now stored in registers 70 through 97.

While the procedure to generate special characters may appear complex, it is not. Using Step b above, the reasons for the various keystrokes are shown below:

Keystrokes	Purpose
[SPACE]T.[SPACE]W.[SPACE]BEERS,[SPACE]PHONE[SPACE]	input normal characters
[ALPHA]	exit ALPHA mode
40	input decimal equivalent of left parenthe- sis
[XEQ][ALPHA]XTOA[ALPHA]	append "(" to ALPHA register
[R↓]	restore the stack as required by ALSTO
[ALPHA]	enter ALPHA mode
[shift][APPEND]	restore the append cursor
317	input normal characters to complete the line

Note that if a special character is needed in the first position in the ALPHA register, as in Step d above, [CLA] is required to clear the ALPHA register; otherwise the special character is appended to the "MORE? OR R/S" prompt.

# USE THIS PAGE FOR NOTES

#### SCROLL (<u>SCROLL</u> prose)

### Purpose

To display in scroll fashion (or print if printer is connected) the alphanumeric contents of contiguous main memory data registers. Using SCROLL, greetings, messages, limericks, etc., can be easily and legibly displayed (or printed).

<u>Minimum SIZE</u>: whatever was required by ALSTO, (ending main memory data register used for the prose) + 1

#### Required Peripherals

OPTIONAL: Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

#### <u>Instructions</u>

- Key in the beginning (bbb) and ending (eee) main memory registers containing the message to be displayed or printed, in the form bbb.eee. If SCROLL is to be called as a subroutine, first make sure bbb.eee is in the X register. If the printer is connected, set it to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 2. Execute SCROLL. The message will be displayed or printed (if the printer is connected), stopping upon completion of the indicated ending register.

## Examples

1. Display the message,

Now is the time for all good men to come to the aid of the party.

stored in main memory registers 00 through 11.

<u>Step Prompt User Input Display</u>

preliminary - Use ALSTO to store in main memory registers 00 through 11 the message to be displayed.

1. .011 2. [XEQ]"SCROLL" NOW IS THE TIME FOR ALL GOOD MEN TO COME TO THE AID OF THE PARTY. 2. Print the 23rd Psalm, which is stored in main memory registers 10 through 109.

<u>Step Prompt User Input</u>

Printer Output (HP 82162A)

preliminary - Use ALSTO to store the 23rd Psalm in main memory registers 10 through 109. Set the printer to MAN mode.

1.	10.109
2.	[XEQ]"SCROLL"

THE LORD IS MY SHEPHERD: I SHALL NOT WANT. HE MA KETH ME TO LIE DOWN IN G **REEN PASTURES: HE LEADET** H ME BESIDE THE STILL WA TERS. HE RESTORETH MY SO UL: HE LEADETH ME IN THE PATHS OF RIGHTEOUSNESS FOR HIS NAME-S SAKE. YEA , THOUGH I WALK THROUGH THE VALLEY OF THE SHADOW OF DEATH, I WILL FEAR N O EVIL: FOR THOU ART WIT H ME: THY ROD AND THY ST **RFF THEY COMFORT ME. THO** U PREPAREST A TABLE BEFO RE ME IN THE PRESENCE OF MINE ENEMIES: THOU ANOI NTEST MY HEAD WITH OIL: MY CUP RUNNETH OVER. SUR ELY GOODNESS AND MERCY S HALL FOLLOW NE ALL THE D AYS OF MY LIFE: AND I WI LL DWELL IN THE HOUSE OF THE LORD FOREVER.

3. Use SCROLL as a <u>subroutine</u> to display the chorus of "Hail Purdue", which is stored in main memory registers 15 through 54.

<u>Step Excerpt from Program Display</u>

preliminary - Use ALSTO to store the chorus in main memory registers 15 through 54.

(program segment)

1. 15.054 2. XEQ"SCROLL" HAIL, HAIL TO OLD PURDUE , ALL HAIL TO OUR OLD GO LD AND BLACK: HAIL, HAIL TO OLD PURDUE, OUR FRIE NDSHIP MAY SHE NEVER LAC K. EVER GRATEFUL, EVER T RUE, THUS WE RAISE OUR S ONG ANEW OF THE DAYS WE-VE SPENT WITH YOU: ALL H AIL, OUR OWN PURDUE.

(program continues)

#### Comments

1. The frequent interspersing of commas, colons, and periods with other alphanumeric characters will cause a slight irregularity in the display of a message because non-consecutive punctuation marks do not occupy a full space in the display even though they are considered by the HP-41 to do so. Printing of the message is not affected. 2. Obviously, punctuation rules must be relaxed since there are no semicolon, apostrophe, or quotation mark characters available on the ALPHA keyboard. However, those users having an Extended Functions/Memory Module (HP 82180A) or an HP-41CX can simply edit the pseudo-ASCII contents of the pertinent main memory register by using the XF: XTOA function; for this purpose a printout of the data register contents is very useful. The one shown below was obtained using the OS-41 program M2PD for registers 15 through 54, used in Example 3.

R15=HAIL,	R35=K. EVE
R16=HAIL T	R36=R GRAT
R17=0 0LD	R37=EFUL,
R18=PURDUE	R38=EVER T
R19=, ALL	R39=RUE, T
R20=HAIL T	R40=HUS WE
R21=0 OUR	R41= RAISE
R22=01 D G0	R42= OUR S
R23=LD AND	R43=ONG AN
R24= BLACK	R44=EW OF
R25=: HAIL	R45=THE DA
R26=, HAIL	R46=YS WE-
R27= T() ()	R47=VE SPE
R28=D PIRD	R48=NT WIT
	R49=H YOU:
R30=R FRIF	R50= ALL H
R31=NDSHIP	R51=AIL, O
R32= MAY S	R52=UR OWN
R33=HE NEV	R53= PURDU
R34=FR LAC	R54=E.
NAL PLANENCA	

For example, to change the contraction in registers 46 and 47 from WE-VE to the proper WE'VE, proceed as follows to correct the contents of register 46:

<u>Step</u>	User Input	Purpose
a.	[ALPHA][<-]	enter ALPHA mode and clear ALPHA
b.	[shift][ARCL]46	append register 46 to ALPHA
c.	[<-]	delete "-" character
d.	[ALPHA]	exit ALPHA mode
е.	39	input decimal equivalent of the apostrophe character
f.	[XEQ][ALPHA]XTOA[ALPHA]	append apostrophe character to the ALPHA register
g.	[ALPHA][shift][ASTO]46	store YS WE' in register 46

<sup>1/</sup>For a table of equivalents, refer to the Extended Functions/Memory Module Owner's Manual, page 14, or to the Comments section of ALSTO.

Note that here the generated special character was terminal in register 46; if instead, normal characters were to follow it, the keystroke sequences in Steps c and g would require modification to temporarily remove the normal characters following the special character and to restore them after its creation.

3. Experienced HP-41 users become aware of the use of flag 11 (automatic execution flag); one especially neat application is the following program which appears to make the calculator unusable:

01+LBL "LOKOFF" 92+LBL 00 03 SF 11 04 OFF 05 GTO 00 06 END

When LOKOFF is executed, the calculator is turned off. Subsequent attempts to turn the HP-41 on will result in the flag 11 tone and an immediate turn off! Only pressing and holding [R/S] before [ON] will allow the calculator to remain on. Assign LOKOFF to a key so it can be quickly executed prior to an anticipated period when the HP-41 won't be used.

In addition to the above, a "return to owner" message, such as described in Example 2 of ALSTO, can be displayed. Refer to the LOKMO program shown below. The program assumes the message is stored in registers 70 through 97, and that SCROLL is in main memory.

> 01+LBL \*LOKMO\* 02+LBL 00 03 SF 11 04 70.097 05 XEQ \*SCROLL\* 06 OFF 07 GTO 00 08 END

IF YOU HAVE FOUND THIS C ALCULATOR, PLEASE BE INF ORMED THAT IT BELONGS TO T. W. BEERS, PHONE (317 ) 463-0807. IF YOU'VE ST OLEN IT, MAY THE "BAD BU GS" BE WITH YOU. 4. SCROLL was purposely made to use the full column width available on any HP-IL printer. An example of HP 82905B 80-column printer output, using a Beers family composition "The Uninvited"<sup>2/</sup> stored in main memory registers 20 through 119 by ALSTO, is shown below.

### Printer Output (HP 82905B)

THIS WINTER OUR FOND WAS A BARREN PLACE, COVERED WITH ICE AND SNOW. NOW SPRING HAS COME, THE ICE IS GONE, AND THINGS ARE BEGINNING T O SHOW. FROGS ARE JUMPING, FISH ARE RISING, AND A MUSKRAT SWAM BY ONE DAY. THE FROGS AND THE FISH WERE WELCOME, BUT WE DIDN'T WANT T HAT MUSKRAT TO STAY. FOR MUSKRATS DO DAMAGE TO PONDS LIKE OURS, THEY BORE HOLES THAT MAKE THE FOND LEAK. THOSE CRITTERS BELONG SOMEW HERE IN THE WILD, BUILDING THEIR HOMES BY A CREEK. SO THOUGH IT WAS SAD, ALONG CAME MY DAD, TO DO WHAT HAD TO BE DONE. HE TOOK CAREF UL AIM, AND WITH A FEELING OF SHAME, HE SHOT THAT POOR SON-OF-A-GUN! ...

 $<sup>2^{\</sup>prime}$ C'mon now, be kind, can you think of an easier way to get your pet doggerel published?

# OS-41 ASCII Data Movement Program Listing: M2XMA

Program Registers: 25 Bytes: 169

01+LBL "M2XMA"	50 0
<b>8</b> 2 <b>*41</b> A1-2*	JZ 3 57 •
03 CF 21	J) 54 0
84 FS? 50	04 H EE 7
85 GTO 80	00 E
86 CF 22	56 L
07 "MM bbb=?"	07 M
68 6	58+L 50 T
09 PROMPT	59 1
18 FC?C 22	60 °
11 VIEW X	61 H
12 "XM FILE?"	62 *
13 AON	63 H
14 PROMPT	64 2
15 AOFF	65 L
16 ASTO Y	66 F
17 XEQ 05	67 L
12+1 RI BR	68 k
19 CLA	69 k
20 ARCL Y	78 0
21 A	71+L
22 SE 25	72 *
23 SEEKPTA	73 4
24 FC2C 25	74 E
25 GTO 84	
26 RIM	
27 .4	
28 +	
29 A	
20 D	
31 SE 05	
32 <b>♦</b> 1 RI A1	
33 RIN	
34 CLA	
35 ARCI IND Y	
36 ALENG	
37 X=0?	
38 GTO 02	
39 FS? <b>0</b> 5	
40 APPREC	
41 FC?C 05	
42 APPCHR	
43+LBL 02	
44 FS?C 05	
45 GTO 03	
46 X≠Y?	
47 SF 05	
48 ISG Z	
49 GTO 01	
50+LBL 03	

51 FS? 55 SF 21 DONE" ASTO X BEEP CLD RTN BL 04 TONE 0 •XM • Arcl z "⊢ SIZE=?" PROMPT KEQ 05 CLA ARCL T CRFLAS RDN RDN GTO 00 LBL 05 \*WORKING\*" AVIEW END

OS-41 ASCII Data Movement Program Listing: SCROLL

Program Registers: 8 Bytes: 55

01+LBL "SCROLL" **0**2 ADV 03+LBL 00 **9**4 CLA 05 ARCL IND X 06 ISG X 07 ARCL IND X 08 ISG X 09 ARCL IND X 10 ISG X 11 ARCL IND X 12 FS? 55 13 ACA 14 FC? 55 15 AVIEW 16 ISG X 17 GTO 01 18 FS? 55 19 PRBUF 20 RTN 21+LBL 01 22 FS? 55 23 GTO 00 24 2 25 -26 GTO 00 27 END

Program Registers: 19 Bytes: 132

	51	ES2 55
Ø1+LBL "XM2MA"	52	SE 21
02 *4192-1*	57	"TONE"
83 FS? 50	54	осто у
04 GTO 00	55	DECD
85 CF 21	50	DECF CLD
06 "XM FILE?"	50	CLD CND
07 AON	J/	END
08 PROMPT		
09 AOFF		
10 ASTO X		
11 CF 22		
12 •MH bbb=?"		
13 PROMPT		
14 FC2 22		
15 0		
10 0 14 ECOC 22		
10 (0:0 22 17 HTCH V		
10 # 10# 0		
10 ************************************		
19 HVIEN		
SN+FRF RR		
21.4		
22 +		
23 CLA		
24 ARCL Y		
25.6		
26 SEEKPTA		
27+LBL 01		
28 RDN		
29 SF 25		
30 GETREC		
31 FC?C 25		
32 GTO 03		
33 ALENG		
34 6		
35 /		
76 INT		
77 F92 17		
70 NCF V		
70 1 E7		
37 1 L3 40 7		
46 / 41410/ 00		
41VLDL 02		
42 HOLU IND T		
43 H5HF		
44 136 T		
45 15G X		
46 610 02		
47 GTO 01		
48+LBL 03		
49 CLA		
50 ASTO IND X		

OS-41 ASCII Data Movement Program Listing: XM2XMA

Program Registers: 44 Bytes: 305

	51+LBL 04	101 GTO 06
01*LBL ^ARZARH 00 *4109-9*	52 CLA	102 XEQ 08
02 "41HZTZ 07 CE 01	53 ARCL 00	103 CLA
03 UF 21 AV FC3 FA	54 RCL 02	104 ARCL 00
04 F5/ 30 af cto ao	55 FS? 06	105 PURFL
00 610 02 at as as	56 ISG X	106 RCL 04
06 LF 03	57 FS? 06	107 CRFLAS
07 TXM SKU FILE?"	58 INT	108 RCL 00
NS HUN	59 SF 25	109 RCL 01
09 PRUMPI	60 SEEKPTA	110 STO 00
10 HSTU UU	61 FC?C 25	111 RDN
11 "XM USI FILE?"	62 GTO <b>0</b> 5	112 STO 01
12 CF 23	63 GETREC	113 GTO 03
13 PRUMPT	64 RCLPT	114♦LBL 06
14 HUFF	65 STO 02	115 ""
15 FC? 23	66 ASTO X	116 ASTO X
16 CLA	67 ASHE	117 RCL AA
17 FC? 23	68 ASTO Y	118 X=Y?
18 ARCL 00	69 ASHE	119 PIRFI
19 FC?C 23	<b>70 A</b> STO 7	120 FS2 55
20 AYIEW	71 ASHE	121 SF 21
21 ASTO 01	72 ASTO 1	122 • DONE•
22 RCL 00	73 CLA	123 ASTO X
23 RCL 01	74 ARCL 01	124 BEEP
24 X≠Y?	75 RCL 03	125 CLD
25 GTO 01	76 SEEKPTA	126 RTN
26 SF 05	77 CLA	127+LBL 07
27 "NEW SIZE=?"	78 ARCL Y	128 TONE 0
28 PRUMPT	79 ARCL Z	129 BST -
29 510 04	80 ARCL T	130 ARCL 01
30+LBL 01	81 ARCL L	131 "⊢ SIZE=?"
31 XEQ 08	82 FS? 05	132 PROMPT
32+LBL 02	83 SF 25	133 XEQ 08
	84 FS? 06	134 CLA
34 FS? 05	85 APPREC	135 ARCL 01
35 ASTO 01	86 FC? 06	136 CRFLAS
36 FS? <b>0</b> 5	87 APPCHR	137 RTN
37 CRFLAS	88 RCLPT	138+LBL 08
38 CLA	89 INT	139 <b>* *WOR</b> KING**
39 ARCL 01	90 STO 03	140 AVIEW
40 0	91 FS? 17	141 END
41 SF 25	92 CF 06	
42 SEEKPTH	93 FC? 17	
43 FC?C 25	94 SF 06	
44 XER 07	95 FC? 05	
45+LBL 83	96 GTO 04	
46 -1	97 FS?C 25	
47 STO 02	98 GTO 04	
48 Ø	99+LBL 05	
49 STU 83	100 FC?C 05	
50 SF 06		

Program Registers: 31 Bytes: 216

01+LBL "XM2MSA"
02 "41A2-3"
03 SF 05
04 GTO 00
05+LBL ™MS2XMA™
86 <b>*4183-2</b> *
07 CF 05
08+LBL 00
89 CF 21
10 FS? 50
11 GTO 01
12 "XM"
13 FC? 05
14 mMS m
15 "⊢ FILE?"
16 AON
17 PROMPT
18 ASTO <del>80</del>
19 <b>"M</b> S"
20 FC? 05
21 "XM"
22 "⊢ FILE?"
23 CF 23
24 PROMPT
25 ROFF
26 FC? 23
27 CLA
28 FC? 23
29 ARCL 00
30 FC?C 23
31 AVIEW
32 ASTO 01
33 XEQ 83
34+LBL 01
35 CLA
36 ARCL 00
37 <b>*</b> F,*
38 ARCL 01
39 SF 25
40 FS? 05
41 SAVERS
42 FC? 05
43 GETAS
44 FC?C 25
45 GTO 02
46 CLA
47 ARCL 01
48 FS?C 05
49 VERIFY
50 FS? 55

51 SF 21 52 -DONE-53 ASTO X 54 BEEP 55 CLD 56 RTN 57+LBL 02 58 CLA 59 ARCL 00 60 FS? 05 61 FLSIZE 62 TONE 0 63 "MS " 64 FC? 05 65 •XM • 66 ARCL 01 67 \* SIZE=?\* 68 CF 22 69 PROMPT 70 FC?C 22 71 VIEW X 72 XEQ 03 73 CLA 74 ARCL 01 75 FS? 85 76 CREATE 77 FC? 05 78 CRFLAS

79 GTO 01 80+LBL 03 81 \* \*WORKING\*\* 82 AVIEW 83 END

# OS-41 ASCII Data Movement Program Listing: XM2CA

Program Registers: 27 Bytes: 186

01+LBL "XM2CA" 92 •41A2-4• 03 CF 21 04 FS? 50 05 GTO 01 06 "XM FILE?" 07 AON **08 PROMPT** 09 AOFF 10 ASTO 00 11+LBL 00 12 "TMP bbb=?" 13 PROMPT 14 STO 01 15.2 16 X>Y? 17 GTO 05 18 \* \*WORKING\*\* 19 AVIEW 20+LBL 01 21 RCL 01 22.4 23 + 24 CLA 25 ARCL 00 26 0 27 SEEKPTA 28+LBL 02 29 RDN 30 SF 25 31 GETREC 32 FC?C 25 33 GTO 04 34 ALENG 35.6 36 7 37 INT 38 FS? 17 39 DSE X 40 1 E3 41 / 42+LBL 03 43 ASTO IND Y 44 ASHF 45 ISG Y 46 ISG X 47 GTO 03 48 GTO 02 49+LBL 04 50 CLA

51 ASTO IND X 52 INT 53 1 E3 54 / 55 RCL 01 56 + 57 "INSERT DST" 58 AVIEW 59 WDTAX 60 "DONE" 61 ASTO X 62 \*\* YER\* IFY CDS\* 63 AVIEW 64 FS? 55 65 SF 21 66 BEEP 67 RTN 68+LBL 05 69 "R00-R01 USED" 70 AVIEW 71 TONE 0 72 PSE 73 GTO 00 74 END

OS-41 ASCII Data Movement Program Listing: XM2AA

Program Registers: 9 Bytes: 62

01+LBL "XM2AA" 02 \*41A2-5\* 03 FS? 50 04 GTO 00 05 \*XM FILE?\* 06 AON 07 PROMPT 08 A0FF 09 ASTO 00 10 "XM RRR.CCC=?" 11 PROMPT 12 STO 01 13+LBL 00 14 CLA 15 ARCL 00 16 RCL 01 17 SEEKPTA 18 GETREC 19 BEEP 20 END

Program Registers: 21 Bytes: 144

01+LBL "XM2PA" 02 \*4102-6\* 03+LBL "XM2DA" 04 \*41A2-7\* 05 FIX 0 06 CF 29 07 FS? 50 08 GTO 00 09 \*XM FILE?\* 10 AON 11 PROMPT 12 AOFF 13 ASTO 00 14 "XM bbb.eee=?" 15 PROMPT 16 \* \*WORKING\*\* 17 CF 21 18 AVIEW 19+LBL 00 20 FC? 00 21 FS? 55 22 SF 21 23 ADV 24 CLA 25 ARCL 00 26 INT 27 SEEKPTA 28 LASTX 29+LBL 01 30 "R" 31 FS? 17 32 "+R" 33 INT 34 ARCL X 35 LASTX 36 "⊦=" 37 ARCLREC 38 FS? 55 39 PRA 40 FC? 55 41 AVIEW 42 FC? 17 43 ISG X 44 GTO 01 45 FC? 55 46 CF 21 47 ADV 48 "DONE" 49 ASTO X 50 BEEP 51 CLD 52 END

OS-41 ASCII Data Movement Program Listing: C2XMA

Program Registers: 30 Bytes: 206

AD:
NI+TRT -CSYNH.
02 <b>*</b> 4184-2*
03 CF 21
04 FS? 50
05 GTO 01
06 "XM FILE?"
87 AON
AR PROMPT
A9 ONFF
1(AID) 00
11VLOL UU 13 HTMD LLL-9H
12 INF 000-1
14 510 101
15-2
16 X>Y?
17 GTO 06
18 XEQ 07
19+LBL 01
20 CLA
21 ARCL 00
22 8
23 SF 25
24 SEEKPTA
25 FC2C 25
26 CTO 85
20 G10 00 27 DCI A1
21 KUL 01 90 A
20 .M 20 1
27 T 70 MINEEDT CDC:
30 TINGEKI OKU
JI HYILW
32 RUTHX
33 XEQ 07
34-6
35 Rt
36 SF 05
37+LBL 02
38 RDN
39 CLA
40 ARCL IND Y
41 ALENG
42 X=0?
43 GTO 03
44 FS? 05
45 APPREC
46 EC2C 05
47 OPPCHR
48♦iRI 97
49 FS2C 85
50 CTO 04
JU 010 01

52 SF 05 53 ISG Z 54 GTO 02 55+LBL 04 56 FS? 55 57 SF 21 58 "DONE" 59 ASTO X 60 BEEP 61 CLD 62 RTN 63+LBL 05 64 TONE 0 65 "XM " 66 ARCL 00 67 \*+ SIZE=?\* 68 PROMPT 69 XEQ 07 70 CLA 71 ARCL 00 72 CRFLAS 73 RDN 74 RDN 75 GTO 01 76+LBL 06 77 •R00-R01 USED• 78 AVIEW 79 TONE 0 80 PSE 81 GTO 00 82+LBL 07 83 \* \*WORKING\*\* 84 AVIEW 85 END

51 X≠Y?

```
OS-41 ASCII Data Movement Program Listing: ALSTO
```

Program Registers: 17 Bytes: 113 01+LBL "ALSTO" 02 "MM bbb=?" **03 PROMPT** 04 STO Y 05.4 06 ÷ 07 \*KEY IN WORDS\* **0**8 AON 09+LBL 00 10 PROMPT 11 FC?C 23 12 GTO 01 13 ASTO IND X 14 ASHF 15 ISG X 16 ASTO IND X 17 ASHF 18 ISG X 19 ASTO IND X 20 ASHF 21 ISG X 22 ASTO IND X 23 ISG X 24 "MORE? OR R/S" 25 GTO 00 26+LBL 01 27 AOFF 28 INT 29 1 30 -31 1 E3 32 / 33 + 34 FIX 3 35 \*bbb.eee=" 36 ARCL X 37 PROMPT 38 GTO "SCROLL" 39 END

Program Registers: 29 Bytes: 198

01+1 BI "02XNO"	51 LASTX
92 *4105-2*	52 °F?°
87 FC 21	53 PROMPT
00 01 21 04 ECO 50	54 FS? 05
04 F3: 30 AF 210 AA	55 GTO 03
00 510 00	56 FC2 23
95 TAR FILE/T	57 GTO <b>84</b>
U/ HUN	5841 RI 07
US PRUMPI	59 FC2 27
NA HOFF	60 ISC Y
10 ASTO 00	21 V/\ V
11 XEQ 05	01 AV7 A
12+LBL 00	02 FU: 20 27 PF 05
13 CLA	03 UF 03 74 FCD 07
14 ARCL 00	64 FU? 23
15 0	65 610 82
16 SF 25	66 FS? <b>0</b> 5
17 SEEKPTA	67 APPCHR
18 FS? 25	68 FC? 85
19 GTO A1	69 APPREC
20 TONE 0	70 FS?C 23
21 •XH •	71 SF 05
22 OPCI AQ	72 GTO 02
22 ANGL 00 27 *L CI7E-2*	73+LBL 04
20 F DIZE-: DA DONNET	74 AOFF
24 FRUNE I	75 FS? 55
20 AEW 00	76 SE 21
26 ULH	77 "DONE"
27 HKUL 00	78 ASTO X
28 CRELHS	79 REEP
29 0	
30 SF 25	00 0CD 01 DTN
31+LBL 01	0241DI 05
32 SEEKPT	02*COL 03 07 • +UADVINC+*
33 SF 25	
34 GETREC	89 HYIEN
35 FS? 25	85 END
36 ISG X	
37 X(> X	
38 FS? 25	
39 GTO 01	
40 FIX 0	
41 CF 29	
42 CE 85	
43 CF 23	
44 AUN	
45+1 RI - 02	
46 *PFC *	
47 F92 05	
40 *+DFC *	
AQ INT	
59 OPCI X	
ee nave o	

# OS-41 ASCII Data Movement Program Listing: B2XMA

Program Registers: 31 Bytes: 217

	F4 10070
01+LBL "B2XMA"	DI LHSIX
02 <b>*</b> 4106-2*	52 AVIEW
03 CF 21	53 "PRESS R/S"
04 ES2 50	54 WNDDTA
05 CTO 00	55 FS? 05
03 610 00 07 - VM ETIEO#	56 CTO 83
05 AM FILC?	57 50 27
N/ HUN	51 1C: 23 E0 CTO 04
08 PROMPT	JO 610 04
09 AOFF	274FRF 03
10 ASTO 00	60 FC? 23
11 XEQ 05	61 ISG X
12+LBL 00	62 X<>X
13 0 9	63 FC? 23
14 OPCI 99	64 CF 05
15 D	65 FC2 23
	66 CTO 02
16 57 23	27 EC2 05
17 SEEKPTH	07 F32 03 70 000000
18 FS? 25	00 HFFUNK
19 GTO 01	69 FC? 05
20 TONE 0	70 APPREC
21 <b>*XM *</b>	71 FS?C 23
22 BRCL BB	72 SF 05
27 • SI7E=?"	73 GTO 02
20   SIZE-: SA DONNOT	74+LBL 04
24 FRUNFI 25 VFD 25	75 ANFE
	76 FS2 55
26 ULH	77 CE 01
27 ARCL 00	(( )( 21 70 = DANC=
28 CRFLAS	(0 DUNE 70 00TO V
29 0	79 HSTU X
30 SF 25	80 RFFL
31+LBL 01	81 CLD
32 SEEKPT	82 RTN
33 SE 25	83+LBL 05
	84 * *WORKING**
75 ECO 25	85 AVIEW
30 F3: 20 7/ 100 V	86 END
36 136 A	
37 X(> X	
38 FS? 25	
39 GTO 01	
40 FIX 0	
41 CF 29	
42 CF 05	
43 CF 23	
44 AUN	
45+LBL 82	
46 "SCAN REC."	
47 FC2 85	
40 -CUUN TDEU -	
40 JUHN TREU 40 Iut	
97 181 50 0501 V	
20 HRCE X	

Chapter 4: THE MOVEMENT AND DUPLICATION OF ENTIRE MEDIA

#### A. <u>Introduction</u>

In order to have confidence in the HP-41 system, an HP-41 user must know his programs and files are safe from inopportune loss. This requires that not only individual programs and files, but entire media be copied or duplicated, thus insuring against catastrophic setbacks. Moreover, media duplication is an important means of distributing software to others. Using functions and OS-41 programs, entire media can be copied to other media or duplicated. Table 4-1 shows the array of possibilities: functions available in modules such as the Extended I/O Module (like MCOPY and MCOPYPV); functions available in the HP-IL Module (like WRTA and READA); functions in peripherals such as the Card Reader (like WALL); and OS-41 programs written to extend these basic functions (like "M2XME" and "MSDUP").

Table 4-1 shows the common types of media movement and duplication along with the functions and/or programs to effect such movement. In Section B, additional details are given so the user can avoid most references to the HP-41 or peripheral manuals. <u>OS-41 program instructions</u> can be found in Section C.

The entire media movement <u>functions</u> discussed in Section B create or recall "write all" files that contain (among other data) main memory "status information," which is defined as:

- (1) The contents of the X, Y, Z, T, and L registers;
- (2) The status of flags 00 through 43;
- (3) The location of the statistical data registers;
- (4) The SIZE;
- (5) The alarms in main memory (if any);
- (6) The position of the program pointer.

Table 4-1. Functions and Programs to Move and Duplicate Entire Media Within the HP-41 System.

- 4	MAGNETIC CARDS	283 CR: WALL	284 See text.	286 See text.	286 See text.
Ϋ́	MASS STORAGE (MS)	283 IL: WRTA "M2MSE"(292)	284 "XM2MSE" ( 298)	285 A/D & XIO: MCOPY A/D & XIO: MCOPYPV A/D: MSCOPY "MSDUP"(304) "MSDUPA"(308)	286 See text.
-2	EXTENDED MEMORY (XM)	283 "M2XME" (288)	284 See text.	285 "MS2XME" (302)	286 See text.
-1	MAIN MEMORY (M)	283 See text.	284 "XM2ME" (296)	284 IL: READA "MS2ME" (300)	286 See text.
T SUBSECTION NUMBER	DESTINATION SOURCE	MAIN MEMORY (M)	EXTENDED MEMORY (XM)	MASS STORAGE (MS)	MAGNETIC CARDS
TEX		н	2	m	4

On the other hand, the entire media movement <u>programs</u> discussed in Section B create or recall "pseudo-write all" files, which may or may not contain the entire contents of the medium they support. Furthermore, the contents of a "pseudo-write all" file created by one program can be properly accessed <u>only</u> by the program that copies media in the opposite sense. For example, a "pseudo-write all" file created by OS-41 program "M2XME" can only be used by "XM2ME".

The cells in Table 4-1 contain one or two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

# B. Functions, Programs, and Their Use

# 1-1.1/ MAIN MEMORY DUPLICATION

No duplication procedure is possible here since there can be only one main memory in the HP-41 system.

#### 1-2. MAIN MEMORY to EXTENDED MEMORY

a. To copy multiple main memory <u>programs</u> - up to the entire main memory program memory - to an extended memory "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"M2XME" (Main memory TO eXtended Memory, Entire).

#### 1-3. MAIN MEMORY to MASS STORAGE

a. Use IL: WRTA (A:filename) $\frac{2}{}$ 

to copy all main memory information (programs, data registers, key assignments, and status information) to mass storage under the specified file name. If a "write all" file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function.

b. To copy multiple main memory <u>programs</u> - up to the entire main memory program memory - to a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"M2MSE" (<u>Main memory TO Mass Storage</u>, <u>E</u>ntire).

## 1-4. MAIN MEMORY to MAGNETIC CARDS $\frac{3}{}$

a. Use CR: WALL

to copy all main memory information (programs, data registers, key assignments, and status information) to magnetic cards.

- $2^{/}$ Throughout this text, the parameter needed in the ALPHA (indicated by A) register is indicated in parentheses.
- $\frac{3}{Before}$  using the CR: VER function to verify magnetic cards, refer to Chapter 2, Section D for an explanation of the VER bug that can alter extended memory.

<sup>1/</sup>Subsection numbers are keyed to Table 4-1 and indicate movement from row number (given first) to column number.

#### 2-1. EXTENDED MEMORY to MAIN MEMORY

a. To recall to main memory all programs from an extended memory "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"XM2ME" (eXtended Memory TO Main memory, Entire).

Unlike most other "read all" procedures, "XM2ME" appends to, <u>not</u> replaces, the previous contents of main memory.

2-2. EXTENDED MEMORY DUPLICATION

No duplication procedure is possible here since there can be only one extended memory in the HP-41 system.

- 2-3. EXTENDED MEMORY to MASS STORAGE
  - a. To copy the entire contents of extended memory (program, data, and ASCII files) to a mass storage "pseudo-write all" file with all parameters prompted for, use the OS-41 program

"XM2MSE" (eXtended Memory TO Mass Storage, Entire).

#### 2-4. EXTENDED MEMORY to MAGNETIC CARDS

a. No direct entire media copying procedure is currently possible here. However, to copy a group of <u>programs</u>, first use the OS-41 program "XM2ME" described in Subsection 2-1 to copy the pertinent "pseudo-write all" file to main memory, then use the CR: WALL function described in Subsection 1-4 to copy <u>all</u> main memory information (including the programs in the "pseudo-write all" file just copied) to magnetic cards.

3-1. MASS STORAGE to MAIN MEMORY

a. Use IL: READA (A:filename)

to recall all main memory information (programs, data registers, key assignments, and status information) from the specified mass storage "write all" file. The previous contents of main memory will be lost.

b. To recall to main memory all programs from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2ME" (Mass Storage <u>IO</u> Main memory, <u>E</u>ntire).

Unlike most other "read all" procedures, "MS2ME" appends to, <u>not</u> replaces, the previous contents of main memory.
## 3-2. MASS STORAGE to EXTENDED MEMORY

a. To recall the entire contents of extended memory (program, data, and ASCII files) from a mass storage "pseudo-write all" file, with all parameters prompted for, use the OS-41 program

"MS2XME" (Mass Storage TO eXtended Memory, Entire).

The previous contents of extended memory will be lost.

#### 3-3. MASS STORAGE DUPLICATION

a. Use A/D & XIO: MCOPY, or

A/D & XIO: MCOPYPV

to duplicate the contents of the source mass storage medium 4' onto all other media in mass storages devices on the HP-IL loop. MCOPY makes all destination files non-private, while MCOPYPV makes all destination files private. The source medium must have no private files or both MCOPY and MCOPYPV will stop without copying any data.

b. To duplicate the contents of the source mass storage medium in a manner similar to MCOPY, but including the protective feature of indicating the source medium prior to duplication and allowing an abort before duplication, use

A/D: MSCOPY

Unlike MCOPY, MSCOPY assumes the source medium is in the first device on the HP-IL loop.

c. To make one or more exact duplicate or backup copies of the entire contents of a mass storage medium with all operations prompted for, use the OS-41 program

"MSDUP" (Mass Storage DUPlication).

d. To make one duplicate or backup copy of the entire contents of a mass storage medium, <u>arranging the destination files in</u> <u>alphabetical order by file name</u>, with all operations prompted for, use the OS-41 program

"MSDUPA" (Mass Storage DUPlication, Alphabetical).

<sup>4/</sup> The source medium is defined as the medium in the mass storage device that is the primary device (if in Manual I/O mode) or the first mass storage device following the primary device (only if in Auto I/O mode). See the HP-IL Module Owner's Manual, pages 46-48.

## 3-4. MASS STORAGE to MAGNETIC CARDS

a. No direct entire media copying procedure is currently possible here. However, one can first use either IL: READA or OS-41 program "MS2ME" described in Subsection 3-1 to copy the information to main memory, then use the CR: WALL function described in Subsection 1-4 to copy <u>all</u> main memory information to magnetic cards. Note that the function IL: READA copies all information comprising a "write all" file, whereas "MS2ME" copies only a specific group of programs.

### 4-1. MAGNETIC CARDS to MAIN MEMORY

a. To recall all main memory information (programs, data registers, key assignments, and status information) from a set of "write all" magnetic cards, insert the cards into the Card Reader with the HP-41 out of PRGM mode. The previous contents of main memory will be lost.

## 4-2. MAGNETIC CARDS to EXTENDED MEMORY

a. No direct entire media copying procedure is currently possible here. However, to copy a group of <u>programs</u>, first use the procedure described in Subsection 4-1 to copy the information to main memory, then use the OS-41 program "M2XME" described in Subsection 1-2 to copy the selected programs to a "pseudo-write all" file in extended memory.

## 4-3. MAGNETIC CARDS to MASS STORAGE

a. No direct entire media copying procedure is currently possible here. However, information on "write all" magnetic cards can be copied first to main memory using the procedure described in Subsection 4-1, then copied to mass storage by the function IL: WRTA described in Subsection 1-3.

## 4-4. MAGNETIC CARD DUPLICATION

a. No direct entire media duplication procedure is currently possible here. However, to duplicate the contents of a set of "write all" magnetic cards, first use the procedure in Subsection 4-1 to copy the information to main memory, then use the CR: WALL function described in Subsection 1-4 to copy the information to another set of magnetic cards. The entire contents of main memory will be changed.

## C. Media Movement and Duplication Program Instructions

The program instructions found on the following pages should enable the user to apply the desired program. An example is provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 4-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings, an explanation of the format of the ASCII "NaMe" file used by MSDUP and MSDUPA, and a description of the program "packaging" concept made possible by M2XME and M2MSE.

M2XME (<u>Main memory TO eX</u>tended <u>Memory</u>, <u>Entire</u>)

Program No.: 41E1-2 Other Programs in This File: M2MSE

### Purpose

To copy multiple main memory programs - up to the entire main memory program memory - to an extended memory "pseudo-write all" file (to be subsequently recalled by XM2ME), with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of programs to copy)

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

## Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2XME and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory "pseudo-write all" file (<u>five</u> characters or less) to copy main memory programs to and press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>0) and press [R/S]. Registers bbb through ((bbb + number of programs to copy) - 1) will be used.

- 3. Follow the prompt:
  - a. MM PRGM <u>n</u>?

where

 $\underline{n}$  = number of the next program to be copied to extended memory.

Key in the name of a main memory program (six characters or less) to be copied to extended memory and press [R/S].

Repeat this step until the names of all programs to be copied to extended memory have been entered, then press [R/S] immediately in response to the "MM PRGM <u>n</u>?" prompt to continue.

<sup>1/</sup>Refer to the Comments section for a description of the "pseudo-write all"
file format.

4. The HP-41 will display "\*WORKING\*" as the main memory programs are copied to the extended memory "pseudo-write all" file. If extended memory is filled before all programs have been copied, the HP-41 will display "XM FULL", then will display either

pram LAST,

where

prgm = the last program successfully copied to extended memory; OR

"NO MM COPIED" if no programs were copied. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

 Copy main memory programs E1-2A, E1-2B, and E1-2C to extended memory "pseudo-write all" file "E1-2D". Use main memory registers 05 through 07 for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Enter programs El-2A, El-2B, and El-2C
 into program memory.

1.		LXEQJ"SIZE"008	
2.		[XEQ]"M2XME"	
a.	XM FILE?	E1-2D [R/S]	
b.	TMP bbb=?	5 [R/S]	
3a.	MM PRGM 1?	E1-2A [R/S]	
	MM PRGM 2?	E1-2B [R/S]	
	MM PRGM 3?	E1-2C [R/S]	
	MM PRGM 4?	[R/S]	
4.			*WORKING*
			DONE

#### Comments

1. The "pseudo-write all" file actually consists of a <u>set</u> of extended memory <u>program</u> files with names in the form

"file.n",

where

 $\frac{file}{n} = extended memory "pseudo-write all" file name, and <math display="block">\frac{n}{n} = 1, 2, 3, \dots, up \text{ to the number of programs copied.}$ 

(For example, Example 1 creates three extended memory program files named "E1-2D.1", "E1-2D.2", and "E1-2D.3", conceptually within the "pseudo-write all" file named "E1-2D".) Any existing extended memory program files in the above form will be replaced.

 Since flag 25 (error ignore flag) is used to detect if extended memory is full, it is possible that other errors (i.e., a nonexistent program name was entered) will cause the "XM FULL" warning. The problem must be detected and corrected by the user. 3. The capability of M2XME to copy groups of programs to extended memory suggests that frequently used programs be classified and named ("packaged") by their function (such as "LOAD" or "LOOK") and moved within the HP-41 system as a group rather than individually. Refer to Section D for more information and sample packages of OS-41 programs.

USE THIS PAGE FOR NOTES

# -291-

M2MSE (<u>Main memory TO Mass Storage</u>, <u>E</u>ntire)

Program No.: 41E1-3 Other Programs in This File: M2XME

#### Purpose

To copy multiple main memory programs - up to the entire main memory program memory - to a mass storage "pseudo-write all" file (to be subsequently recalled by MS2ME), with all parameters prompted for.

<u>Minimum SIZE</u>: (beginning main memory temporary storage register) + (number of programs to copy)

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute M2MSE and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage "pseudo-write all" file (<u>five</u> characters or less) to copy main memory programs to and press [R/S].

b. TMP bbb=?

Key in the beginning main memory temporary storage register (bbb>0) and press [R/S]. Registers bbb through ((bbb + number of programs to copy) - 1) will be used.

- 3. Follow the prompt:
  - a. MM PRGM <u>n</u>?

where

<u>n</u> = number of the next program to be copied to mass storage.

Key in the name of a main memory program (six characters or less) to be copied to mass storage and press [R/S].

Repeat this step until the names of all programs to be copied to mass storage have been entered, then press [R/S] immediately in response to the "MM PRGM  $\underline{n}$ ?" prompt to continue.

<sup>1/</sup>Refer to the Comments section for a description of the "pseudo-write all" file format.

4. The HP-41 will display "\*WORKING\*" as the main memory programs are copied to the mass storage "pseudo-write all" file and verified. If mass storage is filled before all programs have been copied, the HP-41 will display "MS FULL", then will display either

prgm LAST,

where

prgm = the last program successfully copied to mass storage; OR

"NO MM COPIED" if no programs were copied. When completed, the HP-41 will beep, stop, and display "DONE".

## **Examples**

 Copy main memory programs E1-3A, E1-3B, and E1-3C to mass storage "pseudo-write all" file "E1-3D". Use main memory registers 05 through 07 for temporary storage.

 Step
 Prompt
 User Input
 Display

 preliminary - Enter programs E1-3A, E1-3B, and E1-3C into program memory.

 1.
 [XEQ]"SIZE"008

 2.
 [XEQ]"M2MSE"

a.	MS FILE?	E1-3D [R/S]	
b.	TMP bbb=?	5 [R/S]	
3a.	MM PRGM 1?	E1-3A [R/S]	
	MM PRGM 2?	E1-3B [R/S]	
	MM PRGM 3?	E1-3C [R/S]	
	MM PRGM 4?	[R/S]	
4.			*WORKING*
			DONE

#### Comments

 The "pseudo-write all" file actually consists of a <u>set</u> of mass storage program files with names in the form

"<u>file.n</u>",

where

<u>file</u> = mass storage "pseudo-write all" file name, and <u>n</u> = 1, 2, 3, ..., up to the number of programs copied.

(For example, Example 1 creates three mass storage program files named "E1-3D.1", "E1-3D.2", and "E1-3D.3", conceptually within the "pseudo-write all" file named "E1-3D".) Any existing extended memory program files in the above form will be replaced.

2. Since flag 25 (error ignore flag) is used to detect if mass storage is full, it is possible that other errors (i.e., a nonexistent program name was entered) will cause the "MS FULL" warning. The problem must be detected and corrected by the user. 3. The capability of M2MSE to copy groups of programs to mass storage suggests that frequently used programs be classified and named ("packaged") by their function (such as "LOAD" or "LOOK") and moved within the HP-41 system as a group rather than individually. Refer to Section D for more information and sample packages of OS-41 programs.

USE THIS PAGE FOR NOTES

XM2ME (eXtended Memory <u>TO</u> Main memory, <u>E</u>ntire)

Program No.: 41E2-1 Other Programs in This File: MS2ME

## Purpose

To recall to main memory all programs from an extended memory "pseudo-write all" file  $\frac{1}{}$  (created by M2XME), with all parameters prompted for. Unlike all other "read all" procedures, XM2ME appends to, and <u>does</u> not replace, the previous contents of main memory.

#### Minimum SIZE: 000

## Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

- 1. Execute XM2ME and follow the prompt:
  - a. XM FILE?

Key in the name of the extended memory "pseudo-write all" file (five characters or less) to be copied to main memory and press [R/S].

2. The HP-41 will display "\*WORKING\*" as the programs in the extended memory "pseudo-write all" file are copied to main memory. If main memory is filled before all programs have been copied, the HP-41 will display "MM FULL"; CAT 1 can be executed to determine which programs have been successfully copied. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

1. Copy extended memory "pseudo-write all" file "E2-1A", which contains a number of program files stored by M2XME, to main memory.

 Step
 Prompt
 User Input
 Display

 preliminary - Use M2XME to create and load extended<br/>memory "pseudo-write all" file "E2-1A".

 1.
 [XEQ]"XM2ME"<br/>a. XM FILE?

 2.
 \*WORKING\*

\*WORKING\* DONE

<sup>1/</sup>Refer to the Comments section for a description of the "pseudo-write all" file format.

## Comments

 The "pseudo-write all" file, with a name such as "E2-1A" in Example 1, actually consists of the <u>set</u> of extended memory <u>program</u> files created by M2XME with the names in the form

"<u>file.n</u>",

where

```
and \underline{file} = extended memory "pseudo-write all" file name,
and \underline{n} = 1, 2, 3, ..., up to the number of programs copied by M2XME.
```

- 2. Since XM2ME uses the XF: GETSUB function, a superfluous END may be placed before the first program copied to main memory. The END can be removed with the procedure described in Chapter 1, Section A.
- 3. Since flag 25 (error ignore flag) is used to detect the last program file in the extended memory "pseudo-write all" file, it is possible that other errors (i.e., named file doesn't exist) will cause XM2ME to end normally without copying all programs to main memory. The problem will be apparent when CAT 1 doesn't show all of the programs that should have been copied.

XM2MSE (e<u>X</u>tended <u>M</u>emory <u>IO</u> <u>M</u>ass <u>S</u>torage, <u>E</u>ntire)

<u>Program No.</u>: 41E2-3 <u>Other Programs in This File</u>: MS2XME

### <u>Purpose</u>

To copy the entire contents of extended memory (program, data, and ASCII files) to a mass storage "pseudo-write all" file (to be subsequently recalled by MS2XME), with all parameters prompted for.

## Minimum SIZE: 002

### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XM2MSE and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage "pseudo-write all" file (six characters or less) to copy extended memory to and press [R/S].

CAUTION: If the specified mass storage file already exists <u>and is too small</u>, the HP-41 will experience a "MEMORY LOST". See Comment 1 for required file sizes.

b. # XMS=? 0-2

Key in the number of <u>extended memory</u> modules inserted (do <u>not</u> count the Extended Functions/Memory Module) and press [R/S].

CAUTION: If this number is greater than the actual number of extended memory modules, the HP-41 will experience a "MEMORY LOST". If this number is smaller than the actual number of extended memory modules, the contents of the "unspecified" modules may be altered when MS2XME is used later.

3. The HP-41 will display "\*WORKING\*" as it checks if the named mass storage file exists. If it doesn't, it will be created with the appropriate file size (see Comment 1). In either case, the entire contents of extended memory are then copied to mass storage and verified. When completed, the HP-41 will beep, stop, and display "DONE".

<sup>1/</sup>Refer to the Comments section for a description of the "pseudo-write all"
file format.

## <u>Examples</u>

 Copy the entire contents of extended memory, which consists of the Extended Functions/Memory Module and <u>one</u> extended memory module, to mass storage "pseudo-write all" file "E2-3A", which does not exist.

Step Prompt <u>User Input</u> Display\_ preliminary - Insure that <u>only one</u> extended memory module is inserted and that mass storage file "E2-3A" doesn't already exist. [XEQ]"SIZE"002 1. [XEQ]"XM2MSE" 2. MS FILE? E2-3A [R/S] a. # XMS=? 0-2 1 [R/S] Ь.

> \*WORKING\* DONE

## Comments

3.

- 1. The "pseudo-write all" file is actually a mass storage data file; each data file register contains the contents of a corresponding extended memory register (regardless of what type of extended memory file the register belongs to). This method of storing extended memory on mass storage was suggested by Moak (1983). The first register in the mass storage "pseudo-write all" file contains the number of extended memory modules specified, which is required in the subsequent use of MS2XME. The size of the mass storage "pseudo-write all" file depends on the number of extended memory modules specified; no modules creates a file of size 129; one module, size 368; two modules, size 607.
- 2. Because the OS-41 program MS2XME reads from the "pseudo-write all" file the number of extended memory modules specified when XM2MSE was used, it is recommended that the file name specified indicate the number inserted (for example, "XM2" when two modules are inserted). This will help prevent errors when using MS2XME.
- 3. XM2MSE manipulates HP-41 status register "c"; an unfortunate consequence is that XM2MSE <u>cannot be single-step executed</u> or a "MEMORY LOST" will result.

MS2ME (<u>Mass Storage TO Main memory</u>, <u>Entire</u>)

Program No.: 41E3-1 Other Programs in This File: XM2ME

### <u>Purpose</u>

To recall to main memory all programs from a mass storage "pseudo-write all" file (created by M2MSE), with all parameters prompted for. Unlike all other "read all" procedures, MS2ME appends to, and <u>does not replace</u>, the previous contents of main memory.

### Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

### Instructions

- 1. Execute MS2ME and follow the prompt:
  - a. MS FILE?

Key in the name of the mass storage "pseudo-write all" file (five characters or less) to be copied to main memory and press [R/S].

2. The HP-41 will display "\*WORKING\*" as the programs in the mass storage "pseudo-write all" file are copied to main memory. If main memory is filled before all programs have been copied, the HP-41 will display "MM FULL"; CAT 1 can be executed to determine which programs have been successfully copied. When completed, the HP-41 will beep, stop, and display "DONE".

### **Examples**

1. Copy mass storage "pseudo-write all" file "E3-1A", which contains a number of program files stored by M2MSE, to main memory.

 

 Step
 Prompt
 User Input
 Display

 preliminary - Use M2MSE to create and load mass storage "pseudo-write all" file "E3-1A".

 1.
 [XEQ]"MS2ME" a. MS FILE?

 2.
 \*WORKING\*

\*WORKING\*

1/Refer to the Comments section for a description of the "pseudo-write all" file format.

### Comments

 The "pseudo-write all" file, with a name such as "E3-1A" in Example 1, actually consists of the <u>set</u> of mass storage <u>program</u> files created by M2MSE with the names in the form

"<u>file.n</u>",

where

```
\frac{file}{n} = \text{mass storage "pseudo-write all" file name,}
and \frac{n}{n} = 1, 2, 3, \dots, \text{ up to the number of programs copied by}M2MSE.
```

- 2. Since MS2ME uses the IL: READSUB function, a superfluous END may be placed before the first program copied to main memory. The END can be removed with the procedure described in Chapter 1, Section A.
- 3. Since flag 25 (error ignore flag) is used to detect the last program file in the mass storage "pseudo-write all" file, it is possible that other errors (i.e., named file doesn't exist) will cause MS2ME to end normally without copying all programs to main memory. The problem will be apparent when CAT 1 doesn't show all of the programs that should have been copied.

MS2XME (<u>Mass Storage TO eXtended Memory</u>, <u>Entire</u>)

Program No.: 41E3-2 Other Programs in This File: XM2MSE

#### Purpose

To recall the entire contents of extended memory (program, data, and ASCII files) from a mass storage "pseudo-write all" file<sup>1</sup> (created by XM2MSE), with all parameters prompted for. The previous contents of extended memory will be lost.

### Minimum SIZE: 001

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MS2XME and follow the prompt:
  - a. MS FILE?

Key in the name of the mass storage "pseudo-write all" file (six characters or less) to be copied to extended memory and press [R/S].

- 3. The HP-41 will display "\*WORKING\*" as the specified mass storage "pseudo-write all" file is accessed to determine its configuration, then prompt:
  - a. <u>n</u> XMS?

where

 $\underline{n}$  = number of extended memory modules specified when XM2MSE created the file.

If the number of <u>extended memory</u> modules inserted (<u>not</u> counting the Extended Functions/Memory Module) equals <u>n</u>, press [R/S] to continue. Otherwise, insert or remove extended memory modules as required to match <u>n</u> and press [R/S] or go to Step 2, specifying a different file.

CAUTION: If the number of extended memory modules inserted is less than <u>n</u>, during the next step "MEMORY LOST" will result. If the number of modules inserted is greater than <u>n</u>, during the next step the contents of the "unrecognized" modules may be altered.

<sup>1/</sup>Refer to the Comments section of XM2MSE for a description of the "pseudowrite all" file format.

4. The HP-41 will display "\*WORKING\*" as it copies the entire contents of a previous extended memory from the specified mass storage "pseudo-write all" file, replacing the current contents of extended memory. When completed, the HP-41 will beep, stop, and display "DONE".

## <u>Examples</u>

 Copy the entire contents of a previous extended memory from mass storage "pseudo-write all" file "E3-2A", which was created by XM2MSE with one extended memory module.

<u>Step</u>	Prompt	<u>User Input</u>	_Display_
prelim	inary – Use XM2M mass sto "E3-2A", extended that "E3	ISE to copy extend orage "pseudo-writ insuring first t memory module is 5-2A" doesn't alre	ed memory to e all" file hat <u>only one</u> inserted and eady exist.
1.		[XEQ]"SIZE"001	
۷.		LXEUJ"MSZXME"	
a.	MS FILE?	E3-2A [R/S]	
3.			*WORKING*
a.	1 XMS?	[R/S]	
4.			*WORKING*

DONE

### <u>Comments</u>

 MS2XME manipulates HP-41 status register "c"; an unfortunate consequence is that MS2XME <u>cannot be single-step</u> <u>executed</u> or a "MEMORY LOST" will result. MSDUP (<u>Mass Storage DUP</u>lication)

Program No.: 41E3-3 Other Programs in This File: none

#### Purpose

To make one or more exact duplicate or  $backup^{1/2}$  copies of the entire contents of a mass storage medium with all operations prompted for.

<u>Minimum SIZE</u>: 003; 005 if backup option is selected and Time Module is not inserted

### Required Peripherals

Extended Functions/Memory Module (HP 82180A)
Extended I/O Module (HP 82183A)
HP-IL Interface Loop Module (HP 82160A)
2 or more HP-IL mass storage devices, such as HP 82161A Digital
Cassette Drives
OPTIONAL: Time Module (HP 82182A), used if backup option is selected to
 skip prompt for current date

## Instructions

- 1. Insure that the minimum SIZE is set and that all mass storage devices have fully charged batteries.
- 2. Execute MSDUP and follow the prompts:
  - a. SET STANDBY?

Mass storage (and other) devices with standby modes that respond to the IL: PWRDN function can be turned off automatically at the end of MSDUP to preserve battery charge. To enable this option, set the devices to STANDBY and press [R/S]. Otherwise, just press [R/S].

b. BACKUP? Y/N

Press [Y] (the multiply key) to indicate that one or more backup copies of the source medium are to be made (see footnote 1). Press any other key to indicate that one or more exact duplicate copies of the source medium are to be made.

- 3. If the backup copy option is selected <u>and</u> the Time Module is not inserted, follow the prompt:
  - a. DATE?

Key in the current date (<u>eleven</u> <u>characters</u> or less) in a standard format such as MM/DD/YY and press [R/S].

<sup>1/</sup> The distinction between the duplicate and backup options in MSDUP is that the backup option updates header information in the source and destination "NaMe" files, as described under Comments.

## 4. Follow the prompts:

#### a. CLOSE ALL MS

If any mass storage medium access doors are open and/or don't contain media, insert source or destination media (if necessary) and close them, then wait for the HP-41 to continue. Otherwise, just wait for the HP-41 to continue.

b. OPEN SRC MS

Open the medium access door of the mass storage device containing the source medium, then wait for the HP-41 to continue.

CAUTION: Opening any medium access door except the source medium access door can cause the entire contents of the source medium to be destroyed.

c. CLOSE SRC MS

Close the medium access door of the mass storage device containing the source medium, then wait for the HP-41 to continue. The destination media will be all media in mass storage devices on the HP-IL loop except the mass storage device containing the source medium.

5. The HP-41 will display "\*WORKING\*" as it prepares to duplicate the source medium.

CAUTION: At any point up to Step 6, the duplication process can be stopped without causing loss of previous data in the destination media. At worst, the "NaMe" file of the source medium may lose data and have to be restored. When Step 6 has begun, permanent loss of previous destination data (if any) will result.

6. The HP-41 will display "FORMAT", "COPY", and "VERIFY" as all data from the source medium are copied to the destination media and verified. Then the HP-41 will display "\*WORKING\*" as the "NaMe" file in the source medium is updated (if necessary). When completed, the HP-41 will turn off all devices set to STANDBY, beep, stop, and display "DONE". After approximately ten minutes of inactivity, the HP-41 will turn itself off; when later turned on, the HP-41 will display "DONE" if, and only if, MSDUP was successfully completed.

## <u>Examples</u>

 The mass storage medium "OS-41 TEST" was last backed-up on January 27, 1984. Today, October 23, 1984, it is to be backed-up again. Before using MSDUP, MSCAT prints the following header:

> MASS STORAGE: 0S-41 TEST TIME: 12:00 PM, 10/23/84 LAST BACKUP: 01/27/84

After using MSDUP, MSCAT prints the following headers for "OS-41 TEST" and its backup:

#### "OS-41 TEST"

Backup

MASS STORAGE: OS-41 TEST TIME: 1:00 PM, 10/23/84	MASS STORAGE: OS-41 TEST TIME: 1:05 PM, 10/23/84 LAST BACKUP: NONE
LAST BACKUP: 10/23/84	LHOT DHCKOL TONL

#### Comments

- If the backup option is selected, MSDUP changes the ASCII file "NaMe" on the source medium to indicate the date backed-up, and changes the ASCII file "NaMe" on the destination media to indicate the media have not been backed-up. Refer to Section D for an explanation of the "NaMe" file format.
- 2. If the backup option is selected, the source medium <u>must</u> have the "NaMe" file on it for MSDUP to operate properly. If necessary, use MSCAT first to establish a "NaMe" file on the source medium.
- 3. If the backup option is selected, MSDUP temporarily creates the extended memory ASCII file "NaMe". Therefore, three extended memory registers must be available and "NaMe" must not already exist in extended memory or an error message will be displayed.
- 4. If the source medium contains any private program files (indicated by a "P" under the TYPE column in an MSCAT listing), MSDUP will stop before Step 6 and display "PRIVATE".
- 5. MSDUP should work with any HP-IL mass storage devices provided that they:
  - (a) have Accessory IDs in the range 16-31 (mass storage device class);
  - (b) return 0 in response to the IL: INSTAT function when idle with the medium access doors closed;
  - (c) return 20 in response to the IL: INSTAT function with the medium access doors open; and
  - (d) have Auto Address Unconfigure default settings of 2.

Refer to the individual mass storage owner's manuals to check these requirements.

USE THIS PAGE FOR NOTES

MSDUPA (<u>Mass Storage DUP</u>lication, <u>A</u>lphabetical)

Program No.: 41E3-3A Other Programs in This File: none

### Purpose

To make one duplicate or  $backup^{1/2}$  copy of the entire contents of a mass storage medium, arranging the destination files in alphabetical order by file name, with all operations prompted for. MSDUPA should be used only infrequently because of the extensive media manipulation it requires.

Minimum SIZE: (number of files on source medium) + 1

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)
Extended I/O Module (HP 82183A)
Time Module (HP 82182A), required only if backup option selected
HP-IL Interface Loop Module (HP 82160A)
2 HP-IL mass storage devices, such as HP 82161A Digital Cassette
Drives

## Instructions

- 1. Insure that the minimum SIZE is set, or let the HP-41 set it in Step 2a. Also, insure that all mass storage devices have fully charged batteries.
- 2. Execute MSDUPA and follow the prompts:

a. AP. # FILES=?

Key in number of files on the source medium, approximating to a larger number if desired, and press [R/S]. The HP-41 will set the required SIZE. To leave the SIZE unchanged (after manually insuring the minimum SIZE is set), just press [R/S].

b. SET STANDBY

Mass storage (and other) devices with standby modes that respond to the IL: PWRDN function can be turned off automatically at the end of MSDUPA to preserve battery charge. Because MSDUPA will run for a long time and probably be unattended, this becomes almost a requirement. Set the devices to STANDBY and press [R/S].

c. BACKUP? Y/N

Press [Y] (the multiply key) to indicate that a backup copy of the source medium is to be made (see footnote 1). Press any other key to indicate that a duplicate copy of the source medium is to be made.

<sup>1/</sup>The distinction between the duplicate and backup options in MSDUPA is that the backup option updates header information in the source and destination "NaMe" files, as described under Comments.

d. CLOSE ALL MS

If any mass storage medium access doors are open and/or don't contain media, insert source or destination media (if necessary) and close them, then wait for the HP-41 to continue. Otherwise, just wait for the HP-41 to continue.

e. OPEN SRC MS

Open the medium access door of the mass storage device containing the source medium, then wait for the HP-41 to continue.

CAUTION: Opening any medium access door except the source medium access door can cause the entire contents of the source medium to be destroyed.

f. CLOSE SRC MS

Close the medium access door of the mass storage device containing the source medium, then wait for the HP-41 to continue. The destination medium will be the medium in the mass storage device next on the HP-IL loop to the mass storage device containing the source medium.

- g. "NEWM" DST?
  - (1) If the destination medium is unused, or no files on it are to be saved, press [XEQ] "NEWM", then specify the number of file entries to allocate to the directory of the destination medium. This number should be approximately, but not less than, the number available on the source medium.

CAUTION: Up to this point the duplication process can be stopped without causing loss of previous data in the source and destination media. When this step has begun, permanent loss of previous destination data (if any) will result.

When the "NEWM" initialization process is completed, press [R/S] to continue.

- (2) If the files on the source medium are to be alphabetically <u>merged</u> to existing files on the destination medium, just press [R/S].
- 3. The HP-41 will display "\*WORKING\*" as it prepares to duplicate the source medium and reads the names of all files on the source medium. Then, all devices set to STANDBY will be turned off as the HP-41 alphabetizes the source file names (a lengthy process; possibly an hour or more). Eventually, all devices set to STANDBY will be turned on and the HP-41 will copy data from the source medium to the destination medium, then the HP-41 will display "VERIFY" as the destination medium is verified. Finally, the HP-41 will display

"\*WORKING\*" as the "NaMe" file in the source medium is updated (if necessary). When completed, the HP-41 will turn off all devices set to STANDBY, beep, stop, and display "DONE". After approximately ten minutes of inactivity, the HP-41 will turn itself off; when later turned on, the HP-41 will display "DONE" if, and only if, MSDUPA was successfully completed.

### Examples

1. The mass storage medium "OS-41 TEST" was last backed-up on December Today, January 31, 1984, it is to be duplicated in 31, 1983. alphabetical order. Before and after using MSDUPA, MSCAT prints the following directory catalogs for "OS-41 TEST" and its alphabetical duplicate:

<u>Before MSDUPA</u> ("OS-41 TEST") <u>After MSDUPA</u> (alphabetical duplicate)

MASS STORAGE: OS-41 TEST TIME: 11:00 AM, 01/31/84 LAST BACKUP: 12/31/83			MASS STORAGE: 0S-41 TEST TIME: 12:00 PM, 01/31/84 LAST BACKUP: 12/31/83
NOME	TYPE	PECS	NAME TYPE REGS
NEM1D	ΠÖ	20 20	NaMe AS/S 3
стмпы	DD C	52	ALPHA PR,S 3
	DD C	7	DEMIB DA 20
HLFHH DEMIC	TK)0 TO	0 00	DEMIC DA 20
DENIC N-M-	рн ле е	20	MASK KE 1
Nane	H373 VE	-0 -4	STMON PR.S 52
MHSK	KE	1	TCTT DO 607
TSTT	ÐĤ	607	1011 DH 001

#### Comments

- 1. If the backup option is selected, MSDUPA changes the ASCII file "NaMe" on the source medium to indicate the date backed-up, and changes the ASCII file "NaMe" on the destination medium to indicate the medium has not been backed-up. Refer to Section D for an explanation of the "NaMe" file format.
- 2. If the backup option is selected, the source medium <u>must</u> have the "NaMe" file on it for MSDUPA to operate properly. If necessary, use MSCAT first to establish a "NaMe" file on the source medium.
- 3. Because it is frequently accessed, the "NaMe" file is copied first to the destination directory by MSDUPA regardless of its alphabetical position.
- 4. If the backup option is selected, MSDUPA temporarily creates the extended memory ASCII file "NaMe". Therefore, three extended memory registers must be available and "NaMe" must not already exist in extended memory or an error message will be displayed.

- 5. If the source medium contains any private program files (indicated by a "P" under the TYPE column in an MSCAT listing), MSDUPA will stop after copying all non-private files prior to the alphabetically-first private file and display "PRIVATE".
- 6. If the source mass storage medium has any files with names <u>seven</u> characters long, MSDUPA will stop after copying all files prior to the alphabetically-first seven-character name file and display "FL NOT FOUND". The only exception is if there exists a six-character name file that matches the first six characters of the seven-character name file; then, MSDUPA will stop upon attempting to copy the second of the pair and display "DUP FL NAME".
- 7. As described in Step 2g, it is possible to alphabetically merge the files on the source medium to existing files on the destination medium by pressing [R/S] directly in response to the prompt, ""NEWM" DST?". However, if the destination medium contains any files with the same name as a source file (such as "NaMe"), MSDUPA will stop and display "DUP FL NAME".
- 8. MSDUPA should work with any HP-IL mass storage devices provided that they:
  - (a) have Accessory IDs in the range 16-31 (mass storage device class);
  - (b) return 0 in response to the IL: INSTAT function when idle with the medium access doors closed;
  - (c) return 20 in response to the IL: INSTAT function with the medium access doors open; and
  - (d) have Auto Address Unconfigure default settings of 2.

Refer to the individual mass storage owner's manuals to check these requirements.

# D. Program Listings and Technical Details

## 1. Index of Media Movement and Duplication Program Listings

The program listings on the following pages were printed by an HP 82162A Thermal Printer set to MAN mode. Use the table below to locate the page number of any OS-41 media movement and duplication program listing or barcode.

OS-41 Prog	ram	Page Nu Listing	mber for Barcode
M2XME	•••	 . 318	449
M2MSE	• • •	 . 318	449
XM2ME		 . 319	451
XM2MSE		 . 320	452
MS2ME	• • •	 . 319	451
MS2XME	• •	 . 320	452
MSDUP	• •	 . 321	454
MSDUPA	• •	 . 322	456

Table 4-2. Media Movement and Duplication Program Listing and Barcode Locations.

# 2. The "NaMe" File Format

Both MSDUP and MSDUPA change the ASCII file "NaMe" if the backup option is selected. The "NaMe" file is accessed by MSCAT to provide additional information about the medium catalogued (see Chapter 6). Specifically, the "NaMe" file contains the name of the mass storage medium (ten characters or less) and the date the medium was last backed-up by MSDUP or MSDUPA in the format MM/DD/YY, DD.MM.YY, or free-form. The date format is determined by the MDY/DMY setting when MSDUP or MSDUPA was last used, and is free-form if the Time Module wasn't inserted. If the medium has never been backed-up, the date will be "NONE".

The format of the ASCII file "NaMe" is the same whether it's in its permanent mass storage medium location or its temporary extended memory location, and is illustrated below.

ASCII file "NaMe" (size: 3)			
Record	Contents		
00	mass storage medium name		
01	date last backed-up or NONE		

#### 3. <u>The Packaging of Programs According to Function</u>

The development of the OS-41 programs M2XME and M2MSE makes possible the convenient copying of <u>groups</u> (or "packages") of programs to either extended memory or mass storage, which facilitates later recalling them to main memory with OS-41 programs XM2ME and MS2ME. This capability suggests the classification (according to their purpose) of frequently used programs into logical packages, which could then be copied to main memory by one operation. For example, the need to copy numeric or alphanumeric data to or from the X or ALPHA registers, main memory, extended memory, and mass storage would suggest grouping the OS-41 programs X2MD, X2XMD, and X2MSD into a "pseudo-write all" file called "LOAD", enabling the movement of the entire "LOAD" package rather than the individual programs. Similarly, one might group the print (or display) programs M2PD, XM2PD, MS2PD, C2PD, and B2PD into a "pseudo-write all" file called "LOOK". Other logical groupings of OS-41 programs are shown in Table 4-3.

The availability of the OS-41 program XM2MSE and its inverse MS2XME carries the procedure one step further, enabling the nesting of several packages of programs into a "master file" so that extended memory can be configured rapidly to a previous condition. For example, the "pseudo-write all" files "LOAD" and "LOOK" described above might be copied from extended memory to mass storage under the file name "LL2" using the program XM2MSE, then later recalled to extended memory from mass storage using the program MS2XME.<sup>3/</sup>

To demonstrate more clearly the concept of packaging programs, consider the following scenario.

I, as a user, have an HP-41CV equipped with the Extended Functions/Memory Module, two extended memory modules, the HP-IL Interface Loop Module, and a digital cassette drive. I have frequent need to move individual programs within the system and therefore want to group M2XMP (which includes M2MSP), XM2MP (which includes MS2MP), and XM2MSP (which includes MS2XMP) into a conceptual package file called "PRGI", so that it can be moved as one unit. Furthermore, I occasionally want to call upon or move the group of programs that can perform this packaging operation: M2XME (which includes M2MSE) and XM2ME (which includes MS2ME). I've decided to refer to this conceptual

<sup>&</sup>lt;sup>3/</sup>At this point a precautionary caveat must be made to the unwary user. The programs XM2MSE and MS2XME, which use concepts employed by Moak (1983) and by Krabach (1984), involve synthetic programming (Wickes, 1980) to move the "curtain" between main memory data registers and program memory. Because of this internal manipulation, an "unexplained" "MEMORY LOST" can sometimes occur. Therefore, due care should always be taken to make backup files prior to using XM2MSE or MS2XME. See Section C in the INTRODUCTION for error recovery techniques.

Function	Suggested File Name <sup>1/</sup> of Package	Logical OS-41 Programs to Include
Load numeric or alphanumeric data	LOAD (55)2/	X2MD (12) <sup>2/</sup> X2XMD and X2MSD (43)
View or print data	LOOK (119)	M2PD and M2DD (21) XM2PD, XM2DD, MS2PD, and MS2DD (44) C2PD and C2DD (34) B2PD and B2DD (20)
Store or recall individual programs	PRGI (81)	M2XMP and M2MSP (29) XM2MP and MS2MP (16) XM2MSP and MS2XMP (36)
Store or recall packages of programs	PRGP (67)	M2XME and M2MSE (43) XM2ME and MS2ME (24)
Clear or purge data, programs, or files	CLR (100)	MZ and MX (18) XMZ (26) XMX and MSX (22) MSZ (34)
Process ASCII data – common	ASCC (134)	A2XMA (29) XM2AA (9) XM2MSA and MS2XMA (31) XM2PA and XM2DA (21) XM2XMA (44)
- uncommon	ASCU (88)	XM2CA (27) C2XMA (30) B2XMA (31)
Process pseudo-ASCII data	ASCP (69)	M2XMA (25) SCROLL (8) XM2MA (19) ALSTO (17)

Table 4-3. Examples of Logical Packages of OS-41 Programs According to Function.

1/This "pseudo-write all" file name should not exceed four characters, to facilitate subsequent movement of sub-files whose names are defined to be two characters longer than the package name.

 $^{2\prime} {\rm The}$  number of main memory program registers required is shown in parentheses.

package file by the name "PRGP". Since I do not want to continually occupy a considerable part of extended memory with "PRGI" and "PRGP", I want to store these two packages (and whatever else resides in extended memory at the time) in a conceptual mass storage file called "PRG2". The steps to accomplish my needs are detailed below.

<u>Step</u> P	rompt	User Inp	utDis	<u>play</u>
---------------	-------	----------	-------	-------------

 Insure that M2XME and the programs in the first group (M2XMP, XM2MP, and XM2MSP) are present in main memory program memory and that the minimum SIZE is set (in this case, 007).

2.		[XEQ]"M2XME"	
a.	XM FILE?	PRGI [R/S]	
b.	TMP bbb=?	2 [R/S]	
с.	MM PRGM 1?	M2XMP [R/S]	
	MM PRGM 2?	XM2MP [R/S]	
	MM PRGM 3?	XM2MSP [R/S]	
	MM PRGM 4?	[R/S]	
d.			*WORKING*
			DONE

The "pseudo-write all" file residing in extended memory conceptually under the name "PRGI" now contains the programs M2XMP (which includes M2MSP) under the file name "PRGI.1", XM2MP (which includes MS2MP) under the file name "PRGI.2", and XM2MSP (which includes MS2XMP) under the file name "PRGI.3".

3. Insure that the programs in the second group (M2XME and XM2ME) are present in main memory program memory.

4.		[XEQ]"M2XME"
a.	XM FILE?	PRGP [R/S]
b.	TMP bbb=?	4 [R/S]
с.	MM PRGM 1?	M2XME [R/S]
	MM PRGM 2?	XM2ME [R/S]
	MM PRGM 3?	[R/S]
d.		

\*WORKING\*

DONE

The "pseudo-write all" file residing in extended memory conceptually under the name "PRGP" now contains the programs M2XME (which includes M2MSE) under the file name "PRGP.1", and XM2ME (which includes MS2ME) under the file name "PRGP.2".

5. Insure that the program XM2MSE is present in main memory program memory. At this point extended memory contains the two program packages ("PRGI" and "PRGP") and perhaps other information.

StepPromptUser InputDisplay6.[XEQ]"XM2MSE"a.MS FILE?PRG2 [R/S]b.# XMS=? 0-22 [R/S]c.\*WORKING\*DONE

The "pseudo-write all" file stored in mass storage under the name "PRG2" is actually a data file containing the entire contents of extended memory at the time XM2MSE was executed. When "PRG2" is later recalled to extended memory by MS2XME, the number of extended memory modules inserted <u>must</u> be the same as when "PRG2" was stored (in this case, the Extended Functions/Memory Module plus two extended memory modules). Note that the <u>position</u> of the modules can be different than before, but as always, the two extended memory modules must not be in ports one above the other (i.e., in ports 1 and 3 or in ports 2 and 4). When "PRG2" has been copied to mass storage, future use would normally be as in the next step.

- 7. Copy "PRG2" from mass storage to extended memory using MS2XME then proceed, as desired, with any of the following:
  - (a) Copy the program package "PRGI" to main memory using XM2ME;
  - (b) Copy the program package "PRGP" to main memory using XM2ME;
  - (c) Perform both (a) and (b);
  - (d) Copy any of the individual programs in <u>either</u> package to main memory, using XM2MP to copy "PRGP.1" (i.e., M2XME), "PRGP.2" (i.e., XM2ME), etc.; or, perhaps more directly, use MS2MP (now in main memory if Step (a) has been performed) to copy the desired program (using its OS-41 name) from mass storage to main memory.

Program Registers: 43 Bytes: 300

01+LBL "M2XME" 02 \*41E1-2\* 03 SF 05 04 GTO 00 05+LBL "M2MSE" 06 \*41E1-3\* 07 CF 05 98+LBL 00 09 CF 21 10 FS? 50 11 GTO 02 12 "XM" 13 FC? 85 14 "MS" 15 \*\* FILE?\* 16 AON 17 PROMPT 18 AOFF 19 ASTO 00 20+LBL 01 21 "TMP bbb=?" 22 PROMPT 23 X=02 24 GTO 08 25+LBL 02 26.4 27 + 28 FIX 0 29 CF 29 30 1.4 31 AON 32+LBL 03 33 "MM PRGM " 34 ARCL X 35 •H?\* 36 PROMPT 37 FC?C 23 38 GTO 04 39 ASTO IND Y 40 ISG X 41 ISG Y 42 GTO 03 43+LBL 04 44 AOFF 45 \* \*NORKING\*\* 46 AVIEN 47 X()Y 48 -49 CHS 50 LASTX

51 INT 52 1 53 -54 1 E3 55 7 56 + 57 ISG X 58 "." 59 ARCL 00 60 ASTO Y 61 CLA 62 ARCL Y 63-1 64 AROT 65 ASTO 00 66 RDN 67 1.4 68+LBL 05 69 CLA 70 ARCL IND Y 71 "+," 72 ARCL 00 73 ARCL X 74 SF 25 75 FS? 05 76 SAVEP 77 FC? 05 78 WRTP 79 FC?C 25 80 GTO 07 81 CLA 82 ARCL 00 83 ARCL X 84 FC? 05 85 VERIFY 86 FC? 05 87 SEC 88 ISG X 89 ISG Y 90 GTO 05 91+LBL 06 92 CF 05 93 FS? 55 94 SF 21 95 "DONE" 96 ASTO X 97 BEEP 98 CLD 99 RTN 100+LBL 07

101 TONE 0 102 "XM" 103 FC? 05 104 "MS" 105 "H FULL" 106 PVIEW 107 DSE Y 108 X(> X 109 CLA 110 ARCL IND Y 111 \*+ LAST\* 112 INT 113-1 114 X=Y? 115 "NO MM COPIED" 116 AVIEW 117 PSE 118 GTO 06 119+LBL 88 120 "R00 USED" 121 AVIEW 122 TONE 0 123 PSE 124 GTO 01 125 END

Program Registers: 24 Bytes: 162

01+LBL "XM2ME" 02 \*41E2-1\* 03 SF 05 04 GTO 00 95+LBL "MS2ME" 06 \*41E3-1\* 07 CF 05 08+LBL 00 09 CF 21 10 FS? 50 11 GTO 01 12 •XM• 13 FC? 05 14 "MS" 15 \*F FILE?\* 16 AON 17 PROMPT 18 AOFF 19 ASTO X 20 \* \*WORKING\*\* 21 AVIEW 22+LBL 01 23 RCLFLAG 24 FIX 0 25 CF 29 26 1.4 27+LBL 02 28 CLA 29 ARCL Z 30 "⊦." 31 ARCL X 32 SF 25 33 FS? 05 34 FLSIZE 35 FC? 05 36 VERIFY 37 FC? 25 38 GTO 03 39 FS? 05 40 RDN 41 FS? 05 42 GETSUB 43 FC? 05 44 READSUB 45 ISG X 46 FS?C 25 47 GTO 02 48 TONE 0 49 "MM FULL" 50 AVIEW

51 PSE 52+LBL 03 53 FS?C 05 54 RDN 55 STOFLAG 56 FS? 55 57 SF 21 58 "DONE" 59 ASTO X 60 BEEP 61 CLD 62 END Program Registers: 44 Bytes: 307

A1+LRI "XM2MSE" 02 \*41E2-3\* 03 SF 05 94 GTO 99 05+LBL "MS2XME" 96 \*41E3-2\* 07 CF 05 08+LBL 00 09 CF 21 10 FS? 50 11 GTO 01 12 CF 06 13 CF 07 14 "MS FILE?" 15 AON 16 PROMPT 17 ANFF 18 ASTO 00 19 \*# XMS=? 0-2\* 20 FS? 05 21 PROMPT 22 XEQ 05 23+LBL 01 24 FS? 05 25 GTO 02 26 CF 06 27 CF 07 28 CLA 29 ARCL 00 30.0 31 SEEKR 32 READRX 33 RCL 00 34 RCLFLAG 35 FIX 0 36 CF 29 37 CLA 38 ARCL Y 39 \*F XMS?\* 40 STOFLAG 41 PROMPT 42 XEQ 05 43 RDN 44+LBL 02 45 1 46 X=Y? 47 SF 06 48 RDN 49.2 50 X=Y?

101 READRX 102 RDN 103 FC?C 07 104 GTO 04 105 "zi00+" 106 X(> [ 107 STO c 108 .238 109 FS? 05 110 WRTRX 111 FC? 05 112 READRX 113 RDN 114+LBL 04 115 RDN 116 STO c 117 CLA 118 ARCL 00 119 FS?C 05 120 VERIFY 121 FS? 55 122 SF 21 123 \*DONE\* 124 ASTO X 125 BEEP 126 CLD 127 RTN 128+LBL 85 129 \* \*WORKING\*\* 130 AVIEW 131 END Synthetic Text in XM2MSE LINE 80= F5:01:69:04:00 00: LINE 94= F5:01:69:20:10: 00: LINE 105=F5:01:69:30:10: 00:
OS-41 Media Duplication Program Listing: MSDUP

Program Registers: 50 Bytes: 348

01+L8L "MSDUP"	51 RCL 01	101 SF 21
02 *41E3-3*	52 SELECT	102 7
03 CF 21	53 FC? <b>0</b> 9	103 DEVL
04 FS? 50	54 GTO 02	104 PWRDN
05 GTO 01	55 "NaMe"	105 "DONE"
06 CF 09	56 3	106 ASTO X
07 -SET STANDBY?-	57 CRFLAS	107 BEEP
08 PROMPT	58 GETAS	108 CLD
N9 -BACKUP2 Y/N-	59 1	109 RTN
10 AVTEN	60 SEEKPTA	110+LBL 04
11+I BI BB	61 DELREC	111 TONE 9
12 GETKEY	62 "NONE"	112 AVIEW
13 X=02	63 APPREC	113+LBL 05
14 GTO 00	64 "NaMe"	114 -16
15 71	65 UNSEC	115 FINDAID
16 *YES*	66 SAVEAS	116 STO 82
17 X≠Y2	67 VERIFY	117+LBL 06
18 •N0•	68 SEC	118 RCL 00
19 OVIEN	69♦LBL 02	119 X()Y
28 Y=Y2	70 MCOPY	120 X=Y2
21 SF 89	71 NVERIFY	121 A
22 X±V2	72 XF0 07	122 1
27 CTO 01	73 FC2C 89	123 +
24 SF 25	74 GTO 03	124 SELECT
25 DOTE	75 1	125 -16
26 FC2C 25	76 SEEKPT	126 FINDAID
27 CTO 01	77 DEL REC	127 SELECT
28 •NOTF?*	78 CLA	128 INSTRT
29 DAVE.	79 SE 25	129 FS2 08
70 PPOMPT	80 DATE	130 20
71 ONEE	81 FC2 25	131 EC2 08
72 0510 07	82 ARCI 03	132 0
77 OCHE	83 FC2C 25	133 X≠Y2
74 OSTO 04	84 ARCI 84	134 GTO 85
75+1 Bi A1	85 RCLELAG	135 ES2 08
	86 FIX 4	136 RTN
37 NLOOP	87 X<>Y	137 RCL 02
78 STO 88	88 SF 25	138 RCI T
79 - CLASE ALL MS-	89 ADATE	139 X≠Y2
40 FF 08	90 X<>Y	140 GTO 06
41 XF0 04	91 STOFLAG	141 RTN
42 "OPEN SRC MS"	92 APPREC	142+LBL 07
47 SE 08	93 "NaMe"	143 * *WORKING**
44 XFR 04	94 UNSEC	144 AVIEW
45 RCI SEI	95 SAVEAS	145 END
46 STO 81	96 VERIFY	
47 "CLOSE SRC MS"	97 SEC	
48 CE 08	98 PURFL	
49 XF0 04	99 <b>•</b> LBL 03	
50 XE0 07	100 FS? 55	
ee new et		

Program Registers: 79 Bytes: 549

01+LBL -MSDUPA-	51 SELECT	101 -
02 *41E3-3A*	52 •R/S*	102 1 E8
03 CF 10	53 ASTO X	103 /
04 CF 21	54 ""NEWM" DST?"	104 X<>Y
05 FS? 50	55 PROMPT	105 2
06 GTO 01	56 XEQ 13	106 -
07 CF 09	57 RCL 01	107 1 E3
08 "AP. # FILES=?"	58 SELECT	108 /
09 CF 22	59 "NaMe "	109 +
10 PROMPT	60 FC? 09	110+LBL 05
11 1	61 GTO 82	111 1
12 +	62 3	112 +
13 FS2C 22	63 CRELAS	113 STO 00
14 PST7F	64 CETAS	114+1 Bi 86
15 "SET STANDRY"	65 1	115 RCL AA
16 PROMPT	66 SEEKPTO	116 RCL IND X
17 -ROCKUP2 Y/N*	67 DEL REC	117 ISG Y
19 OVIEN	68 "NONE"	118 **
10 HTLM 1941 Di GG	69 OPPPEC	119 PCI IND Y
	70 •NoMe •	120 Pt
20 GETKET 21 V-02	71 HNCCC	121 R+
21 A-0: 22 CTA AG	72 ENUENE	122 XFD 12
22 010 00	77 VEDTEV	122 NOR 12
20 (1 94 =VEC=	73 YERIFI 74 GEC	120 100 00
24 153 AF V40A	(4 3EU 754101 00	124 GIU 00 125 DCI 00
23 A+1: 24 #NO#	(JVLDL 02 7/ OCTO V	123 KUL 00 194 EDC
26 "NU" 27 OUTEU	(0 H3)U A 77 t 0	120 FRU 197 1 E.7
Z/ HYIEW Do V-VD	(( 1.7 70 CTO 7	127 I ETO 120 -
28 A=1 /	78 510 Z 70 CICN	120 - 120 V\02
29 SF 09	(7.516N 0041 DL 07	127 A/07 170 CTO OF
20 ONTOTO 20 ONTOTO	80+LBL 83	130 610 03
31 HUTUTU 70 HU 000	81 X() L	131 <b>FMKUF</b> 173 LACTY
32 NLUUM	82 DIKA 07 N 00	132 LHSTA
33 510 00	83 AF0/	133 +
34 "CLUSE WLL MS"	84 GIU 84	134 I ED
35 CF 108	85 SIGN	130 #
36 XEW NY	86 HSTU X	136 FRU
37 "UPEN SRC HS"	87 ISG L	137 1
38 SF 108	88 X=Y?	1.38 +
39 XEQ 09	89 SF 10	139 510 00
40 RCLSEL	90 X=Y?	140 NLUUP
41 STO 01	91 GTO 03	141 RCLSEL
42 CLOSE SRC MS*	92 ASTO IND Z	142 CLA
43 CF 08	93 ISG Z	143 ARCL X
44 XEQ 09	94 GTO 03	144 X=Y?
45 RCL 00	95+LBL 04	145 0
46 RCL 01	96 PWRDN	146 1
47 X=Y?	97 RCL Z	147 +
48 0	98 INT	148 SELECT
49 1	99 STO Y	149 -16
50 +	100 1	150 FINDAID

OS-41 Media Duplication Program Listing: MSDUPA (continued)

151 ANUM
152 SELECT
153 RDN
154 "NaMe"
155 FS?C 10
156 COPYFL
157+LBL 07
158 CLA
159 ARCL IND 00
160 COPYFL
161 ISG 00
162 GTO 07
163 RCL 00
164 MVERIFY
165 XEQ 13
166 FC2C 09
167 GTO 08
168 1
169 SEEKPT
179 DELREC
171 CLA
172 POIFIAG
173 DOTE
174 FT¥ 4
175 ONOTE
176 RIN
177 STOFLOC
179 =NaMe=
190 INSEC
100 00320 101 COVEDS
101 UEDIEV
102 TERIT
104 FUREL 10541 DI 40
10JVLDL 00 104 ECO 55
105 F3: JJ 107 CE 31
107 3F 21 100 7
100 ( 100 hCU)
107 JETL 108 DUDBU
170 FMKUN 101 -DONE-
191 - LUME 100 00T0 V
172 HOLU A
175 DEEF 104 CLD
174 ULU 105 DTN
170 KIN 107400 00
170*LDL 07
197 JUNE 9
198 HVIEW
199+LBL 10
200 -16

201 FINDAID 202 STO 02 203+LBL 11 204 RCL 00 205 X<>Y 206 X=Y? 207 0 208 1 209 + 210 SELECT 211 -16 212 FINDAID 213 SELECT 214 INSTAT 215 FS? 08 216 20 217 FC? 08 218 0 219 X#Y? 220 GTO 10 221 FS? 08 222 RTN 223 RCL 02 224 RCL T 225 X#Y? 226 GTO 11 227 RTN 228+LBL 12 229 RDN 230 RDN 231 CLA 232 ARCL Y 233 ATOX 234 ASTO Z 235 CLA 236 ARCL Y 237 ATOX 238 ASTO Z 239 X=Y? 240 GTO 12 241 X>Y? 242 RTN 243 RCL 00 244 RCL IND X 245 ISG Y 246 \*\* 247 RCL IND Y 248 X<>Y 249 STO IND Z 250 DSE Z

251 \*\* 252 RDN 253 STO IND Y 254 RTN 255+LBL 13 256 \* #NORKING\*\* 257 AVIEN 258 END

# Chapter 5: OPERATIONS TO CLEAR AND PURGE, AND THOSE RELATED TO KEY ASSIGNMENTS, FLAGS, AND ALARMS

A. Introduction

Even the casual HP-41 user has a periodic need to clear (set to zero) or to eliminate (purge or expunge) various registers, files, or media. Table 5-1 shows many of the options to perform these tasks: calculator functions (like CLRG and CLP); functions available in modules such as the Extended Functions/Memory Module (like CLFL and PURFL); functions available in the HP-IL Module (like ZERO and PURGE); functions in peripherals such as the Card Reader (like WDTAX); and OS-41 programs written to streamline or extend these basic functions (like "MZ" and "MSX").

In a similar fashion Table 5-2 shows functions and programs to set or clear, save or restore, and check the status of key assignments, flags, and alarms.

The cells in Tables 5-1 and 5-2 contain one or two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

# The Master Clear Procedure

The HP-41 can be manually reset to initial power-on conditions by using a procedure known as "Master Clear". Master Clear is the severest of all purging operations. While the reset state is occasionally established accidentally, it is sometimes convenient to apply Master Clear on purpose. The following procedure can be used to Master Clear the HP-41.

<u>Step</u>	Instructions	<u>Keystrokes</u>
1.	Turn the HP-41 off.	[ON]
2.	Press and hold the backarrow key.	[<-]
3.	Turn the HP-41 on.	[ON]
4.	Release the backarrow key.	[<-]

TEX	TEXT SUBSECTION NUMBER -1 -2			
		OPERATION		
	MEDIUM	CLEAR (SET TO ZERO) (Z)	PURGE (EXPUNGE) (X)	
1	MAIN MEMORY (M)	<pre>328 Data Registers: MASTER CLEAR CLX, ST0 CLRG CLΣ PPC: BC PPC: BI CX: CLRGX "MZ"(344) 329 Stack and ALPHA Registers: MASTER CLEAR CLX CLA CLST, + See text for others.</pre>	329 Programs: MASTER CLEAR CLP XF: PCLPS PPC: EP "MX"(346)	
2	EXTENDED MEMORY (XM)	330 MASTER CLEAR XF: CLFL XF: DELREC "XMZ"(348)	331 MASTER CLEAR XF: PURFL "XMX"(350)	
3	MASS STORAGE (MS)	332 IL: ZERO "MSZ"(352)	332 IL: NEWM IL: PURGE "MSX"(354)	
4	MAGNETIC CARDS	333 CLX, CR: WDTAX		

# Table 5-1. Functions and Programs to Clear and Purge Data, Files, or Programs Within the HP-41 System.

Alarms
and
Flags,
signments,
ey As
ž
HP-41
<b>t</b>
Related
Programs
and
Functions
Table 5-2.

			1		
ц Ч		DISPLAY	337 Press and hold key P & IL: PRKEYS CX: CAT 6 PPC: VK	339 FS? P & IL: PRFLAGS PPC: VF	341 T: ALMCAT CX: CAT 5
-4		RESTORE	336 IL: READK "GETK" PPC: RK See text.	339 XF: STOFLAG IL: READS See text.	341 T: XYZALM
Ŷ	OPERATION	SAVE	335 IL: WRTK "SAVEK" PPC: SK CR: WSTS	338 XF: RCLFLAG IL: WRTS CR: WSTS	340 CX: RCLALM
-2		CLEAR	334 MASTER CLEAR ASN [ALPHA][ALPHA] CLA, XF: PASN XF: CLKEYS PPC: CK	338 MASTER CLEAR CF XF: X<>F PPC: IF PPC: RF	340 MASTER CLEAR CX: CLALMA CX: CLALMX CX: CLALMX CX: CLRALMS T: ALMCAT, [c] See text.
R -1		SET	333 ASN PPC: MK XF: PASN PPC: 1K, +K	337 SF XF: X<>F PPC: IF	340 T: XYZALM
T SUBSECTION NUMBE		TOPIC	KEY ASSIGNMENTS	FLAGS	ALARMS
тех			ىر. س	Q	7

-326-

The HP-41 will display "MEMORY LOST", indicating that the Master Clear has reset calculator status as follows:

- 1. All main memory programs are purged.
- 2. All extended memory files are purged.
- 3. All alarms in main memory are purged.
- All main memory data registers, including the stack and ALPHA registers, are cleared.
- 5. All flags are reset to their initial power-on settings:

flags 00-20, 22-25, 27, 30-36, 38-39, and 41-54 are cleared; flags 26, 28-29, 37, and 40 are set; flags 21 and 55 are set only if the printer is connected.

 The SIZE is changed in one of two ways, depending on the HP-41 model:

> HP-41C and HP-41CV SIZE is set such that there are 46 main memory program registers; HP-41CX SIZE is set to 100.

- 7. All key assignments are cleared.
- 8. The Time Module date format is set to MDY.

Note that Master Clear <u>does not</u> reset the Time Module stopwatch, clock, or CLK12/CLK24 and CLKT/CLKTD formats.

A power interruption (including removing the batteries) of sufficient duration can cause a "power reset" with the same results as the Master Clear procedure, <u>plus</u> disruption of the Time Module stopwatch, clock, and CLK12/CLK24 and CLKT/CLKTD formats.

In Tables 5-1 and 5-2, MASTER CLEAR is shown in those cells where clearing or purging occurs during a Master Clear.

1-1.1/ MAIN MEMORY CLEAR OF DATA, STACK, AND ALPHA REGISTERS

- a. All data, stack, and ALPHA registers are cleared by MASTER CLEAR. See Section A for details.
- b. Use STO (X:zero)<sup>2/</sup>

to clear individual main memory data registers.

- c. To clear data registers while leaving the contents of the stack and ALPHA unchanged, use
  - (1) CLRG to clear <u>all</u> the currently allocated data registers, or
  - (2)  $CL\Sigma$  to clear only those six registers currently defined as the statistical registers. Unless otherwise designated by  $\Sigma REG$ , the statistical registers are 11 through 16.
- d. Use the <u>Block Clear program</u>

PPC: BC (X:bbb.eeeii)

to clear the block of registers defined by bbb as the beginning and eee as the ending registers, and ii as the increment value; specifying ii enables the clearing of alternate (ii=02), every third (ii=03), etc., registers. After executing BC, the X register is cleared, the L register is changed, and registers Y, Z, T, and ALPHA remain unaffected.

e. The <u>Block Increment program</u>

PPC: BI (X:III; Y:BBB; Z:bbb.eeeii)

described in Chapter 2, Section B, Subsection 5-1, can be used to clear the block of registers from bbb through eee, incremented by i; set III=0 (in X) and BBB=0 (in Y). When used to clear registers, the BI program is somewhat slower than BC; furthermore, in the stack only the T and ALPHA registers remain unaffected after BI has been executed.

<sup>1</sup>/Subsection numbers are keyed to Tables 5-1 and 5-2 to indicate the medium or status item affected and the operation performed.

 $<sup>^{2/}</sup>$ Throughout this text, the parameters needed in the stack registers and ALPHA (indicated by A) register are indicated in parentheses.

f. HP-41CX users can use the function

CX: CLRGX (X:bbb.eeeii)

to clear registers bbb through eee, incremented by ii.

g. To clear all, or a selected block of, main memory registers, with the necessary block limits prompted for, use the OS-41 program

"MZ" (Main memory Zero).

- h. To clear part or all of the stack and ALPHA registers, use the following as needed.
  - (1) CLX (or [<-] out of ALPHA mode if digit entry is terminated $\frac{3}{}$ ) to clear only the X register;
  - (2) CLA (or [<-] in ALPHA mode if character entry is terminated $^{2/}$ ) to clear only the ALPHA register;
  - (3) 0,  $\sqrt{x}$  to clear the X and L registers;
  - (4) 0, ENTER  $\uparrow$ , ENTER  $\uparrow$ , ENTER  $\uparrow$  or
  - (5) CLST to clear the X, Y, Z, and T registers;
  - (6) CLST, + (or \*, -, or  $\sqrt{x}$ ) to clear all stack registers (<u>including</u> L).

The function CLD is used in a program to clear the display of what is VIEWed or AVIEWed, replacing it with the "flying goose" character.

### 1-2. MAIN MEMORY PURGE OF PROGRAMS

- a. All programs and their key assignments (if present) are purged by MASTER CLEAR. See Section A for details.
- b. To purge programs individually, use

CLP

Specify the name of the program to purge when prompted. To purge the "current" program (the one observed in PRGM mode), at the CLP prompt press [ALPHA] twice <u>without</u> specifying a program name.

 $<sup>^{3/}</sup>$ Digit or character entry is terminated if the prompt character "\_" is not displayed at the end of the X or ALPHA register.

- c. To purge a selected group of programs from main memory, use either of the following:
  - (1) XF: PCLPS (A:starting program name)

to purge consecutive programs in main memory starting with the one specified in the ALPHA register through to the permanent ".END.".

(2) The Erase Program Memory program

PPC: EP

to purge consecutive programs in main memory following the program segment "//" in this form:

LBL "//" RCL b END + + + + + + +

If flag 14 is set or program "//" does not exist, <u>all</u> programs will be purged. Follow the execution of EP by PACK to reclaim the program registers used by the purged programs.

d. To purge consecutive programs in main memory starting with the user-specified program through to the permanent ".END.", use the OS-41 program

"MX" (<u>Main memory eXpunge</u>).

# 2-1. EXTENDED MEMORY CLEAR

- a. All of extended memory is cleared (actually purged) by MASTER CLEAR. See Section A for details. To preserve extended memory during a planned MASTER CLEAR, first remove any extended memory modules and the Extended Functions/Memory Module, perform the MASTER CLEAR, then re-insert the modules within approximately 30 seconds. Note that this procedure can't be used with an HP-41CX since its Extended Functions/Memory Module is built-in.
- b. To clear the contents of an extended memory  $\underline{data}$  or  $\underline{ASCII}$  file, use

XF: <u>CLFL</u>4/ (A:filename).

<sup>4/</sup>An underline is used throughout this text to indicate that the function is the <u>logical</u> one to use to make the file named in the ALPHA register the "working file" (see the Extended Functions/Memory Module Owner's Handbook, page 23.).

If the named file is a data file, all registers in the file are set to zero; if an ASCII file, the number of records is set to zero. Attempting to purge a <u>program</u> file using CLFL will result in a "FL TYPE ERR" message; see Subsection 2-2 to purge a program file.

c. To delete an extended memory ASCII file <u>record</u> (thus "clearing" the record), first use

XF: <u>SEEKPTA</u> (X:rrr; A:filename), or

XF: SEEKPT (X:rrr)

to move the pointer to record rrr of the named (or working) extended memory file, then use

XF: DELREC

to delete record rrr.

d. To clear all or selected registers in an extended memory data file or to delete all or selected records in an extended memory ASCII file, with all parameters prompted for, use the OS-41 program

"XMZ" (eXtended Memory Zero).

2-2. EXTENDED MEMORY PURGE

- a. All of extended memory is purged by MASTER CLEAR. See Section A for details. To preserve extended memory during a planned MASTER CLEAR, first remove any extended memory modules and the Extended Functions/Memory Module, perform the MASTER CLEAR, then re-insert the modules within approximately 30 seconds. Note that this procedure can't be used with an HP-41CX since its Extended Functions/Memory Module is built-in.
- b. To purge any file (data, ASCII, or program) from extended memory, use

XF: PURFL (A:filename).

The PURFL function in extended functions/memory modules with the header (obtained by pressing [shift][CATALOG][2]) "-EXT FCN 1B" contains a dangerous bug. After PURFL has been executed, any function that operates on the working file before the working file has been established (i.e., without specifying it in ALPHA) will disrupt the extended memory directory, making all extended memory files inaccessible. If this happens, a program detailed by Jarett (1983) in "HP-41 Extended Functions Made Easy" can be used to restore the directory.

c. To simultaneously purge one or more extended memory files with all parameters prompted for, use the OS-41 program

"XMX" (eXtended Memory eXpunge).

# 3-1. MASS STORAGE CLEAR

a. To clear the contents of an unsecured mass storage data file, use

IL: ZERO (A:filename).

All registers in the named mass storage file are set to zero. If the named file is any other file type, "FL TYPE ERR" is displayed.

If the file is secured, first use

IL: UNSEC (A:filename).

b. To clear all <u>or selected</u> registers in an unsecured mass storage <u>data</u> file, with all parameters prompted for, use the OS-41 program

"MSZ" (Mass Storage Zero).

#### 3-2. MASS STORAGE PURGE

a. To completely purge (i.e., initialize) a mass storage medium, use

IL: NEWM

Upon execution of NEWM, the user is prompted for the number of file entries anticipated, so that space for the directory can be allocated. Although any number up to 447 can be specified, directory size in the range of 100 to 250 is more logical since a smaller directory space enables faster access to files stored on the medium.

b. To purge an unsecured mass storage file, use

IL: PURGE (A:filename).

If the file is secured, first use

IL: UNSEC (A:filename).

The PURGE function removes the named file's entry from the directory and the file's space on the medium becomes available for the first new file that can fit into that space.

c. To simultaneously purge one or more mass storage files with all parameters prompted for, use the OS-41 program

"MSX" (Mass Storage eXpunge).

# 4-1. MAGNETIC CARDS CLEAR

- a. When the calculator is in PRGM mode, inserting magnetic cards into the Card Reader will, as expected, lead to the replacement of previous data on the cards. However, inserting pre-recorded cards indiscreetly with the calculator <u>out of</u> PRGM mode may have unplanned and undesirable or catastrophic results:
  - Inserting a set of "write all" magnetic cards inadvertently (they will be accepted as such even though READA has not been executed) will re-configure main memory, or, if the entire set of cards is not inserted, lead to MEMORY LOST.
  - (2) Inserting data cards inadvertently will copy the data from the cards to main memory data registers starting with register zero.
  - (3) Inserting program cards inadvertently will replace the last program in main memory if it does not have its own END.

In order to avoid the above possibilities, used magnetic cards of questionable content can be neutralized by this procedure suggested by Reinstein (1982): CLX followed by the function CR: WDTAX, that is

CR: WDTAX (X:zero).

The SIZE must be at least 001. The card track neutralized is actually loaded with the contents of main memory data register 00. At worst, subsequent inadvertent use of the card track will replace the contents of main memory data register 00.

# 5-1. KEY ASSIGNMENTS SET

All assignments of <u>functions</u> to keys require one-half register of main memory program memory. Therefore, enough main memory program registers must be available when assigning functions or the HP-41 will display "PACKING" followed by "TRY AGAIN". <u>Program</u> key assignments are stored in the LBL statement itself and hence require no additional main memory.

a. Use the non-programmable function

ASN

to assign any function or program to a key. The ASN function can be employed while in or out of USER mode to assign the desired function or program to any of 68 key positions. The only keys not assignable are [shift], [ON], [USER], [PRGM], and [ALPHA].

After the name of the function or program is provided at the prompt, the key to be assigned is pressed and the HP-41 briefly displays "ASN <u>function or program keycode</u>"; the variable <u>keycode</u> is a two-digit row and column number defining the position of the key assigned, and is negative if the key is shifted. For example, the key sequence [shift] [TAN] is keycode -25.

b. Use the <u>Make multiple Key assignments program</u>

PPC: MK

to make two-byte function assignments not possible with ASN. MK enables useful key assignments such as "STO 20" and "TONE 100". Several steps must be taken to use MK properly; refer to the PPC ROM User's Manual for details.

- c. Several options are available to make key assignments <u>under program</u> <u>control</u> (as well as manually):
  - (1) The programmable assign function

XF: PASN (X:keycode; A:function or program name).

See a. above for the definition of the keycode.

(2) The program <u>FIRST Key</u> assignment

PPC: 1K (X:keycode; Y:postfix byte; Z:prefix byte)

will make the two-byte key assignment indicated in the contents of the Z and Y registers to the key specified by the keycode in the X register. The prefix and postfix byte numbers for the function to be assigned can be determined by reference to a "byte table", published elsewhere (see Reinstein, 1982; Jarett, 1982; or Wickes, 1980).

(3) The program <u>ADD</u>itional <u>Key</u> assignments

PPC: +K (X:keycode; Y:postfix byte; Z:prefix byte)

can be used after 1K or MK to make more key assignments.

- 5-2. KEY ASSIGNMENTS CLEAR
  - a. All key assignments are cleared by MASTER CLEAR. See Section A for details.
  - b. To clear a single key assignment, use the non-programmable function

ASN

Press [ALPHA] twice without specifying a function or program name, then press the key to be cleared. A key is <u>not</u> cleared by assigning its standard function to itself. Note that any keys assigned a program are cleared when that program is purged from main memory.

c. To clear a key assignment <u>under program control</u> (as well as manually), use the programmable assign function with the ALPHA register clear:

XF: PASN (X:keycode; A:clear)

For example, the program sequence needed to clear a function or program assigned to the  $e^{X}$  key ([shift][ $e^{X}$ ]) is

-15 CLA PASN

Note that all functions that clear individual key assignments can create unused space in main memory that is unavailable for other purposes. To remove the unused space by "packing" the key assignment registers, occasionally use

PPC: PK

Unfortunately, PK clears all alarms in main memory.

- d. To clear <u>all</u> key assignments <u>under program control</u> (as well as manually), use either of the following:
  - (1) The clear key assignments function

XF: CLKEYS

(2) The <u>Clear Key</u> assignments program

PPC: CK

Like PK, CK clears all alarms in main memory.

## 5-3. KEY ASSIGNMENTS SAVE

When a main memory program is copied to another medium, any key assignment within the program is also saved, and may later be restored (see Section 5-4).

a. Use IL: WRTK (A:filename)

to copy all <u>function</u> key assignments to the specified mass storage file. If a key assignment file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function.

b. To copy all key assignments to extended memory, use the program

"SAVEK" (A:filename)

detailed by Jarett (1983). "SAVEK" uses synthetic programming techniques and requires special procedures to operate properly; refer to "HP-41 Extended Functions Made Easy" for complete instructions and cautions.

c. Use the <u>S</u>uspend <u>K</u>ey assignments program

PPC: SK (X:rrr)

to clear all key assignments <u>and</u> save them in coded form in main memory data registers rrr and rrr+1. The key assignments can be restored with PPC: RK.

d. To copy HP-41 status information (including <u>function</u> key assignments) to magnetic cards, use

CR: WSTS

e. The entire media movement functions discussed in Chapter 4 can be used to save (among other data) all key assignments. Refer to Chapter 4 for details.

#### 5-4. KEY ASSIGNMENTS RESTORE

When a program file is copied from a medium to main memory, any key assignment within the program file is restored if the HP-41 is in USER mode.

a. Use IL: READK (A:filename)

to copy all <u>function</u> key assignments from the specified mass storage file. Note that existing program key assignments are unaffected, and existing function key assignments are cleared.

b. To copy key assignments stored by "SAVEK" from extended memory, use the program

"GETK" (A:filename)

detailed by Jarett (1983). "GETK" uses synthetic programming techniques and requires special procedures to operate properly; refer to "HP-41 Extended Functions Made Easy" for complete instructions and cautions.

c. Use the <u>Reactivate Key assignments program</u>

PPC: RK (X:rrr)

to copy all key assignments from the coded form placed by PPC: SK in main memory data registers rrr and rrr+1. Note that all existing key assignments are cleared.

- d. To copy HP-41 status information (including <u>function</u> key assignments) from magnetic cards, insert the cards into the Card Reader with the HP-41 out of PRGM mode. To copy <u>only</u> the function key assignments, don't insert track 1 into the Card Reader. Note that existing program key assignments are unaffected, and existing function key assignments are cleared.
- e. The entire media movement functions discussed in Chapter 4 can be used to restore (among other data) all key assignments saved. Refer to Chapter 4 for details.

# 5-5. KEY ASSIGNMENTS DISPLAY

- a. To display the function or program assigned to a specific key without executing it, press and hold the key in USER mode. If the key is assigned a <u>program</u>, the character "T" will prefix the program name. If the key is unassigned, the HP-41 will display its standard assignment, which is generally the same as the mnemonic on the top of the key. If the assigned function requires a parameter (indicated by one or more "\_" characters), release the key and press [<-] to continue; otherwise, hold the key down for approximately one second (or until "NULL" is displayed), then release the key to continue.
- b. Use P & IL: PRKEYS

to print the keycode of each assigned key along with the function or program assigned to the key.

c. HP-41CX users can use the function

CX: CAT 6

to display key assignments in the same manner that PRKEYS prints them. Refer to Chapter 6 for more details.

d. Use the <u>View Key</u> assignments program

PPC: VK

to display only the keycodes of assigned keys. If the printer is connected, P & IL: PRKEYS is executed instead.

- 6-1. FLAGS SET
  - a. Use SF

to set any flag from 00-29.

b. Use XF: X<>F (X:flags 00-07 status)

to set (or clear) flags 00-07 in one step. The flags 00-07 status in X is in the form

status = 
$$\sum_{n=0}^{7} i * 2^{n},$$

where

i = 1 if flag n is to be set, = 0 if flag n is to be cleared.

X<>F replaces X with the previous flags 00-07 status in the form above.

c. Use the <u>Invert Elag</u> program

PPC: IF (X:flag to invert)

to invert (set if clear or clear if set) any flag from 00-55.

CAUTION: Inversion of flags 30-55 can cause unusual HP-41 behavior; refer to the PPC ROM User's Manual, page 217.

#### 6-2. FLAGS CLEAR

- a. All flags are reset to their default status by MASTER CLEAR. See Section A for details.
- b. Use CF

to clear any flag from 00-29.

c. Use XF: X<>F (X:flags 00-07 status)

described in Subsection 6-1, to clear (or set) flags 00-07 in one step.

d. Use the <u>Invert Flag</u> program

PPC: IF (X:flag to invert)

described in Subsection 6-1 to invert (clear if set or set if clear) any flag from 00-55.

e. Use the <u>Reset Flags</u> program

PPC: RF

to reset all flags to their default status (which clears most) with the exception that flag 37 is cleared and flag 38 is set, thus setting the display mode to FIX 2.

#### 6-3. FLAGS SAVE

a. Use XF: RCLFLAG

to recall to the X register in coded form the status of flags 00-43, which can be stored just as any numeric datum. The flag status can be restored with XF: STOFLAG.

- b. To copy HP-41 status information (including the status of flags 00-43) to mass storage or magnetic cards, use either
  - (1) IL: WRTS (A:filename), or
  - (2) CR: WSTS,

respectively. If a mass storage status file already exists with the specified file name, it will be replaced unless protected by the IL: SEC function. c. The entire media movement functions discussed in Chapter 4 can be used to save (among other data) the status of flags 00-43. Refer to Chapter 4 for details.

# 6-4. FLAGS RESTORE

a. Use XF: STOFLAG (X:flag status OR X:bb.ee; Y:flag status)

to restore the status of flags 00-43 or flags bb through ee if specified in X. The flag status is in coded form obtained by the XF: RCLFLAG function.

- b. To copy HP-41 status information (including the status of flags 00-43) from the specified mass storage status file or magnetic cards, use either
  - (1) IL: READS (A:filename) (for mass storage files), or
  - (2) Insert the pre-recorded "write status" cards into the Card Reader with the HP-41 out of PRGM mode. To copy all status information <u>except</u> function key assignments, insert only track 1 into the Card Reader.
- c. The entire media movement functions discussed in Chapter 4 can be used to restore (among other data) the status of flags 00-43. Refer to Chapter 4 for details.
- 6-5. FLAGS DISPLAY
  - a. Use FS?

to display the status of any flag. If the specified flag is set, the HP-41 will display "YES"; if clear, "NO".

b. Use P & IL: PRFLAGS

to print (among other data) the status of each flag. If a flag is set, "SET" will be printed next to the flag number; if clear, "CLEAR".

c. Use the <u>View Elags</u> program

PPC: VF

to display the numbers of all flags set, four at a time. Flag 50 (message flag) is always displayed as set because VF sets it.

# 7-1. ALARMS SET

a. Use T: XYZALM (X:alarm time; Y:alarm date; Z:repeat interval; A:text)

to set a tone, message, interrupting control, or noninterrupting control alarm.<sup>5</sup> Many options are available; refer to the Time Module Owner's Manual or the HP-41CX Owner's Manual for more details.

# 7-2. ALARMS CLEAR

- a. All alarms in the alarm catalog are cleared by MASTER CLEAR. See Section A for details.
- b. HP-41CX users can use the following functions to clear individual alarms or all alarms:
  - (1) CX: CLALMA (A:text)

clears the first alarm in the alarm catalog that matches the text in ALPHA.

(2) CX: CLALMX (X:alarm position number)

clears the alarm whose position in the alarm catalog is specified in X.

(3) CX: CLRALMS

clears all alarms in the alarm catalog.

- c. The function T: ALMCAT can be used to clear an alarm. Press [R/S] immediately, then press [SST] to position the alarm catalog to the alarm to be cleared. Finally, press [shift][c] to clear the alarm.
- d. All alarms in the alarm catalog can be cleared by removing the Time Module, turning the HP-41 on and off, and replacing the Time Module. Unfortunately, this also clears the time, date, and other settings.

# 7-3. ALARMS SAVE

a. HP-41CX users can use the function

CX: RCLALM (X:alarm position number)

to recall to the stack and ALPHA registers the alarm whose position in the alarm catalog is specified in X. RCLALM loads the stack and ALPHA in exactly the manner required to set the alarm with T: XYZALM. The alarm information can be stored just as any numeric

<sup>&</sup>lt;sup>5/</sup>The HP-41CX Owner's Manual refers to interrupting control and noninterrupting control alarms as control and conditional alarms, respectively.

and alphanumeric data in three to seven registers, depending on the contents of the ALPHA register.

b. The entire media movement functions discussed in Chapter 4 can be used to save (among other data) all alarms in the alarm catalog. Refer to Chapter 4 for details.

# 7-4. ALARMS RESTORE

- a. Alarms that have been saved with the function CX: RCLALM can be restored by using
  - T: XYZALM (X:alarm time; Y:alarm date; Z:repeat interval; A:text).
- b. The entire media movement functions discussed in Chapter 4 can be used to restore (among other data) all alarms saved. Refer to Chapter 4 for details.

#### 7-5. ALARMS DISPLAY

- a. To display all alarms in the alarm catalog, use either
  - (1) T: ALMCAT, or
  - (2) CX: CAT 5

CAT 5 acts identically to ALMCAT, displaying each alarm time and date chronologically along with its message (if any). Refer to Chapter 6 for more details.

# USE THIS PAGE FOR NOTES

# C. <u>Clear and Purge Program Instructions</u>

The program instructions found on the following pages should enable the user to apply the desired program. One or more examples are provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 5-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings.

MZ (<u>Main memory Z</u>ero)

Program No.: 41Z1-1 Other Programs in This File: MX

### Purpose

To clear (set to zero) main memory data registers with all parameters prompted for. MZ can clear one or more consecutive registers or registers separated by an increment value.

<u>Minimum SIZE</u>: (ending main memory register) + 1

Required Peripherals: none

### Instructions

- 1. Execute MZ and follow the prompt:
  - a. MMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) main memory registers to clear and the register increment value II and press [R/S]. If eee is less than bbb, only register bbb will be cleared. If II is not specified, it will default to 1. To clear <u>all</u> registers, just press [R/S]; the HP-41 will display "CLEAR ALL" as all registers are cleared, then end normally.

 The HP-41 will display "\*WORKING\*" as the specified main memory registers are cleared. When completed, the HP-41 will beep, stop, and display "DONE".

### Examples

1. Clear the contents of main memory data registers 10, 13, 16, and 19.

StepPromptUser InputDisplaypreliminary - Set the SIZE to 020 or greater.1.[XEQ]"MZ"a.MMbbb.eeeII=?10.01903 [R/S]

a.	MMbbb.eeeII=?	10.01903 [R/S]	
2.			*WORKING*
			DONE

2. Clear the contents of all main memory data registers.

<u>Step</u>	Prompt	<u>User Input</u>	<u>Display</u>
1. a. 2.	MMbbb.eeeII=?	[XEQ]"MZ" [R/S]	CLEAR ALL DONE

USE THIS PAGE FOR NOTES

MX (<u>Main memory eXpunge</u>)

Program No.: 41X1-2 Other Programs in This File: MZ

# Purpose

To purge (expunge) main memory programs with all parameters prompted for. MX will purge the specified program and <u>all</u> programs after it.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

- 1. Execute MX and follow the prompt:
  - a. START PRGM?

Key in the name of the first program to be purged and press [R/S].

2. The HP-41 will purge the specified program and all programs after it, then beep, stop, and display "DONE".

# Examples

- 1. A CAT 1 listing shows the following programs:
  - MZ MX PRGM1 PRGM2 PRGM3

Executing MX and specifying "PRGM2" as the first program will purge both "PRGM2" and "PRGM3".

# Comments

1. If MX is among the programs to be purged, the HP-41 will stop without beeping or displaying "DONE" after purging the specified programs.

USE THIS PAGE FOR NOTES

XMZ (eXtended Memory Zero)

Program No.: 41Z2-1 Other Programs in This File: none

# Purpose

To clear (set to zero) registers in an extended memory data file or to delete records in an extended memory ASCII file with all parameters prompted for. XMZ can clear/delete one or more consecutive registers/records or registers/records separated by an increment value.

Minimum SIZE: 002; 001 if clearing all registers/records

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute XMZ and follow the prompts:
  - a. XM FILE?

Key in the name of the extended memory file (six characters or less) to be cleared or deleted and press [R/S].

b. XMbbb.eeeII=?

Key in the beginning (bbb) and ending (eee) extended memory registers/records to be cleared/deleted and the increment value II and press [R/S]. If eee=0, only register/record bbb will be cleared/deleted. If II is not specified, it will default to 1. To clear/delete <u>all</u> registers/records, just press [R/S]; the HP-41 will display "CLEAR ALL" as the file is cleared, then end normally.

 The HP-41 will display "\*WORKING\*" as the specified extended memory registers/records are cleared/deleted. When completed, the HP-41 will beep, stop, and display "DONE".

# Examples

1. Clear the contents of extended memory data file "Z2-1A" registers 05 through 08.

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2XMD to create and load extended<br/>memory file "Z2-1A" with size 9 or more.

 1.
 [XEQ]"SIZE"002

 2.
 [XEQ]"XMZ"

		L. Church Langer Children	
a.	XM FILE?	Z2-1A [R/S]	
b.	XMbbb.eeeII=?	5.008 [R/S]	
3.			*WORKING*
			DONE

<u>Step</u> <u>Prompt</u> <u>User Input</u> <u>Display</u> preliminary - Use A2XMA to create and load extended memory file "Z2-1B" with 9 or more records.

1.		[XEQ]"SIZE"002	
2.		[XEQ]"XMZ"	
a.	XM FILE?	Z2-1B [R/S]	
b.	XMbbb.eeeII=?	2.00803 [R/S]	
3.			*WORKING*
			DONE

3. Clear all registers in extended memory data file "Z2-1C".

Prompt User Input Display Step preliminary - Use X2XMD to create and load extended memory file "Z2-1C". 1. [XEQ]"SIZE"001 [XEQ]"XMZ" 2. XM FILE? Z2-1C [R/S] a. CLEAR ALL b. XMbbb.eeeII=? [R/S] 3. DONE

# Comments

- 1. If eee < bbb and eee ≠ 0, XMZ will clear/delete only register/record eee. Furthermore, the common trick of specifying a large eee to insure that registers/records are cleared/deleted to the end of the file will not work with XMZ, because it clears/deletes registers/records in <u>reverse</u> order, starting with eee, to preserve record numbering during execution.
- 2. If records are deleted in an extended memory ASCII file, the remaining records are moved up in the file and renumbered accordingly.

XMX (eXtended Memory eXpunge)

Program No.: 41X2-2 Other Programs in This File: MSX

## <u>Purpose</u>

To simultaneously purge (expunge) one or more extended memory files with all parameters prompted for.

Minimum SIZE: 000

### Required Peripherals

Extended Functions/Memory Module (HP 82180A)

### Instructions

1. Execute XMX.

4.

- 2. Follow the prompt:
  - a. XM FILE(S)?

Key in the names of one or more extended memory files (each <u>seven</u> characters or less) to be purged, separated by commas, and press [R/S]. Up to 24 characters total (signalled by the ALPHA-full tone) can be entered. To indicate that more files are to be purged, put a comma at the end of the file list. Any spaces before or after a comma are interpreted as part of a file name.

DONE

- 3. The HP-41 will display "\*WORKING\*" as the specified extended memory file or files are purged. If a comma was put at the end of the file list (indicating more files are to be purged), go to Step 2 and continue.
- 4. When completed, the HP-41 will beep, stop, and display "DONE".

# **Examples**

1. Purge the extended memory data file "X2-2A".

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2XMD to create extended memory file "X2-2A".

 1.
 [XEQ]"XMX"

 2a.
 XM FILE(S)?
 X2-2A [R/S]

 3.
 \*WORKING\*

2. Purge the extended memory data files "X2-2B", "X2-2C", and "X2-2D".

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2XMD to create extended memory files "X2-2B", "X2-2C", and "X2-2D".

 1.
 [XE0]"XMX"

 22
 YM EUE(S)?
 Y2-2P Y2-20 Y2-2D [D(S])

- 2a. XM FILE(S)? X2-2B,X2-2C,X2-2D [R/S] 3. \*WORKING\* 4. DONE
- 3. Purge the extended memory data files "X2-2E", "X2-2F", "X2-2G", "X2-2H", and "X2-2I", in two steps.

Step Prompt User Input Display
preliminary - Use X2XMD to create extended memory
 files "X2-2E", "X2-2F", "X2-2G", "X2-2H",
 and "X2-2I".

1.		
2a.	XM FILE(S)?	X2-2E,X2-2F,X2-2G,X2-2H, [R/S]
3.		*WORKING*
2a.	XM FILE(S)?	X2-2I [R/S]
3.		*WORKING*
4.		DONE

Comments

1. If XMX cannot find an extended memory file in the list entered into the ALPHA register, all files to the left of that file in the list will be purged, then XMX will stop and display "FL NOT FOUND". MSZ (<u>Mass Storage Z</u>ero)

Program No.: 41Z3-1 Other Programs in This File: none

# Purpose

To clear (set to zero) registers in an unsecured mass storage data file with all parameters prompted for.

<u>Minimum SIZE</u>: (ending main memory temporary storage register) + 1; 001 if clearing all registers

#### Required Peripherals

HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

# Instructions

- 1. Insure that the minimum SIZE is set.
- 2. Execute MSZ and follow the prompts:
  - a. MS FILE?

Key in the name of the mass storage file (six characters or less) to be cleared and press [R/S].

b. MS bbb.eee=?

Key in the beginning (bbb) and ending (eee) mass storage registers to be cleared and press [R/S]. If eee is less than bbb, only register bbb will be cleared. To clear <u>all</u> registers, just press [R/S]; the HP-41 will display "CLEAR ALL" as the file is cleared, then end normally.

c. TMP bbb.eee=?

Key in the beginning (bbb>2) and ending (eee) main memory temporary storage registers to transfer data through and press [R/S]. If eee is less than bbb, only register bbb will be used.

 The HP-41 will display "\*WORKING\*" as the specified mass storage registers are cleared. When completed, the HP-41 will beep, stop, and display "DONE".

# <u>Examples</u>

 Clear the contents of mass storage data file "Z3-1A" registers 03 through 07. Use main memory registers 10 through 12 for temporary storage.

Step Prompt User Input Display preliminary - Use X2MSD to create and load mass storage file "Z3-1A" with size 8 or more. 1. [XEQ]"SIZE"013 [XEQ]"MSZ" 2. MS FILE? Z3-1A [R/S] a. 3.007 [R/S] MS bbb.eee=? b. TMP bbb.eee=? 10.012 [R/S] с. 3. \*WORKING\* DONE

2. Clear all registers in mass storage data file "Z3-1B".

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2MSD to create and load mass storage file "Z3-1B".
 Image: Complete Comp

2. a.	MS FILE?	[XEQ]"MSZ" 73-18 [R/S]		
ь. З.	MS bbb.eee=?	[R/S]	CLEAR DONE	ALL

### Comments

1. It is not necessary to specify a number of main memory temporary storage registers less than or equal to the number of mass storage registers to clear. If fewer temporary storage registers are needed by MSZ, only those needed will be used. MSX (<u>Mass S</u>torage e<u>X</u>punge)

Program No.: 41X3-2 Other Programs in This File: XMX

# <u>Purpose</u>

To simultaneously purge (expunge) one or more secured or unsecured mass storage files with all parameters prompted for.

Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive

#### Instructions

- 1. Execute MSX.
- 2. Follow the prompt:
  - a. MS FILE(S)?

Key in the names of one or more mass storage files (each <u>seven</u> characters or less) to be purged, separated by commas, and press [R/S]. Up to 24 characters total (signalled by the ALPHA-full tone) can be entered. To indicate that more files are to be purged, put a comma at the end of the file list. Any spaces before or after a comma are interpreted as part of a file name. MSX will purge secured files.

- 3. The HP-41 will display "\*WORKING\*" as the specified mass storage file or files are purged. If a comma was put at the end of the file list (indicating more files are to be purged), go to Step 2 and continue.
- 4. When completed, the HP-41 will beep, stop, and display "DONE".

# <u>Examples</u>

1. Purge the mass storage data file "X3-2A".

 

 Step
 Prompt
 User Input
 Display

 preliminary - Use X2MSD to create mass storage file "X3-2A".
 1.
 [XEQ]"MSX"

 2a.
 MS FILE(S)?
 X3-2A [R/S]
 3.

 3.
 \*WORKING\*

 4.
 DONE

 2. Purge the mass storage data files "X3-2B", "X3-2C", and "X3-2D".

<u>Step Prompt User Input Display</u>

preliminary - Use X2MSD to create mass storage files "X3-2B", "X3-2C", and "X3-2D".

1.		[XEQ]"MSX"
2a.	MS FILE(S)?	X3-2B,X3-2C,X3-2D [R/S]
3.		*WORKING*
4.		DONE

3. Purge the mass storage data files "X3-2E", "X3-2F", "X3-2G", "X3-2H", and "X3-2I", in two steps.

Step \_\_\_\_Prompt \_\_\_\_User Input \_\_\_\_Display\_
preliminary - Use X2MSD to create mass storage
 files "X3-2E", "X3-2F", "X3-2G", "X3-2H",
 and "X3-2I".
1. [XEQ]"MSX"
2a. MS FILE(S)? X3-2E,X3-2F,X3-2G,X3-2H, [R/S]
 \*WOPKING\*

3.			*WORKING*
2a.	MS FILE(S)?	X3-2I [R/S]	
3.			*WORKING*
4.			DONE

Comments

1. If MSX cannot find a mass storage file in the list entered into the ALPHA register, all files to the left of that file in the list will be purged, then MSX will stop and display "FL NOT FOUND".

# D. <u>Program Listings</u>

The program listings on the following pages were printed by an HP 82162A Thermal Printer set to MAN mode. Use the table below to locate the page number of any OS-41 Clear and Purge program listing or barcode.

0S-41 F	rogra	m							Page Nu Listing	mber for Barcode
MZ	· ·	•	•	•	•	•	•	•	357	459
MX		•	•	•	•	•	•	•	357	459
XM	IZ.	•	•	•	•	•	•	•	358	460
XM	IX.	•	•	•	•	•	•	•	359	461
MS	SZ .	•	•	•	•	•	•	•	360	462
MS	sx.	•	•	•	•	•	•	•	359	461

Table 5-3. Clear and Purge Program Listing and Barcode Locations.
01+LBL "MZ" 02 \*4121-1\* 03 FS? 50 04 GTO 00 05 CF 21 06 "MMbbb.eeeII=?" 07 CF 22 **08 PROMPT** 09 "CLEAR ALL" 10 FC? 22 11 AVIEW 12 FC? 22 13 CLRG 14 FC?C 22 15 GTO 02 16 \* \*WORKING\*\* 17 AVIEW 18+LBL 00 19 0 28+LBL 01 21 STO IND Y 22 ISG Y 23 GTO 01 24 GTO 02 25+LBL "MX" 26 \*41X1-2\* 27 \*START PRGM?\* 28 AON 29 PROMPT 30 AOFF 31 PCLPS 32+LBL 02 33 FS? 55 34 SF 21 35 •DONE• 36 ASTO X 37 BEEP 38 CLD 39 END

Program Registers: 18

Bytes: 126

## OS-41 Clear and Purge Program Listing: XMZ

Program Registers: 26 Bytes: 181

	F4 4 57
01+LBL "XMZ"	51 I E3
02 *41Z2-1*	52 /
AZ (F 85	53 +
	54 R*
04 CF 21	554IRI 01
05 FS? 50	534CDC 01
06 GTO 00	36 KUN
07 "XM FILE?"	57 INI
98 90N	58 SEEKPT
DO DDAMPT	59 LASTX
	60 0
IN HUFF	21 CC7 05
11 ASTO WK	01 10: 00 40 DELDEC
12 <b>*XMbbb</b> .eeeII=?*	62 DELKEU
13 CF 22	63 FC? 05
14 PROMPT	64 SAVEX
15 -01 500 011 -	65 DSE Y
10 OLLAN ALL	66 GTO 01
16 FL? 22	4741 RI 02
17 AVIEN	01 VLDL 02
18 CLA	68 KUN
19 ARCL 00	69 INI
20 FC2 22	70 SEEKPT
21 CI FI	71 8
10 COC 20	72 FS? 05
22 FU(U 22	73 DEL REC
23 GIU <b>W</b> 3	74 ECOC 05
24 STO 01	(4 FU/C 65
25+LBL 00	75 SHVEX
26 CLA	76♦LBL 03
27 OPCI 00	77 FS? 55
21 HAGE 00	78 SF 21
28 0	79 - NONE -
29 SEEKPTH	00 0CTO V
30 SF 25	00 H310 A
31 GETX	81 BEEP
32 FC?C 25	82 CLD
77 SE 85	83 END
74 - +UADVINC+*	
24 THORAINGT 25 ONIEU	
30 HYIEM	
36 RUL 01	
37 ENTER†	
38 FRC	
39 X=0?	
40 GTO 02	
A1 1 E7	
49 +	
96 T	
43 INI	
44 LASTX	
45 FRC	
46 1 E3	
47 /	
AQ +	
40 DCI 01	
49 KUL 01	
50 INT	

Program Registers: 22 Bytes: 152

AIAIRI "YMY"	51 ALENG
92 •4192-2•	52 X≠0?
02 THE 2	53 GTO 01
00 01 00 04 CTO 00	54 CLD
OT GIU DO OFAIDI #NCV#	55 GTO AA
0JVLDL NOA 0/ #4197 0#	56+1 BL 03
00 41A372 A7 AF AF	57 CE <b>8</b> 5
07 UF 03 0041 Di 00	58 ES2 55
00*L0L 00	59 SE 21
07 F3/ 30	60 -DONE-
10 GIU 01	61 OSTO Y
	62 REEP
12 FU2 00	67 CLD
	64 FND
14 TF FILE(5)/T	or chu
15 HUN	
16 PRUMPI	
17 AOFF	
18 ASTO X	
19 ASHF	
20 ASTO Y	
21 ASHF	
22 ASTO Z	
23 ASHF	
24 ASTO T	
25 * *WORKING**	
26 CF 21	
27 AVIEW	
28 CLA	
29 ARCL X	
30 ARCL Y	
31 ARCL Z	
32 ARCL T	
33+LBL 01	
34 FC? 05	
35 UNSEC	
36 FS? 85	
37 PURFL	
38 FC? 05	
39 PURGE	
40 44	
41 POSR	
42 X<0?	
43 GTO 03	
44 1	
45 +	
46+LBL 02	
47 ATOX	
48 RDN	
49 DSE X	
50 GTO 02	

OS-41 Clear and Purge Program Listing: MSZ

Program Registers: 34 Bytes: 234

01+LBL "MSZ"
02 •41Z3-1•
03 FS2 50
04 GTO 01
<b>85</b> CF 21
06 "MS FILE?"
07 AON
AS PROMPT
A9 AUEE
10 ASTO 00
11 "MS hhh.eee=?"
12 CE 22
13 PROMPT
14 •CLEAR ALL •
15 FC2 22
19 FC2 22
20 7ED0
21 FC2C 22
22 CTO A4
27 STO 81
23 310 01 2441 Di 90
25 •THD bbb coc=?"
20 TH DDD.EEE-:
20 FRUNFI 97 CTA 89
27 310 02
20 J DO V\VD
27 A/1: 70 CTO 85
30 GIU 0J 71 = +UADVINC+=
31 THURNINGT 73 AUTEU
JE HYIEM 77ai Di Gi
33¥LDL 01 74 €10
34 ULH 75 ADC1 00
33 HKUL 00 74 Dri 41
30 KUL 01 77 CEEVD
07 DEENK 70 DEENK
30 KUL 02 70 D
37 0 4041 Di - 03
40VLDL 02 A1 CTO IND V
41 310 IND 1 42 TCC V
42 136 1 A7 010 02
40 GIU 02 AA DCi Gi
77 KUL DI A5 INT
47 EDC
1 FNU 10 1 F7
19 x
50 X/3Y

51 -52 1 53 X<=Y? 54 + 55 RCL 02 56 INT 57 LASTX 58 FRC 59 1 E3 60 \* 61 X<>Y 62 -63-1 64 + 65 X<=0? 66 SIGN 67 ABS 68 STO 01 69 / 70 FRC 71 ST\* 01 72 LASTX 73 INT 74 STO 00 75 X=0? 76 GTO 04 77+LBL 03 78 RCL 02 79 WRTRX 80 X(0? 81 GTO 06 82 DSE 00 83 GTO 03 84+LBL 04 85 RCL 02 86 INT 87 STO Y 88 RCL 01 89 X=0? 90 GTO 06 91 + 92 1 93 -94 1 E3 95 7 96 + 97 CHS 98 STO 02 99 GTO 03 100+LBL 05

101 -R00-R02 USED-102 AVIEW 103 TONE 0 104 PSE 105 GTO 00 106+LBL 06 107 VERIFY 108 FS? 55 109 SF 21 110 -DONE-111 ASTO X 112 BEEP 113 CLD 114 END

# USE THIS PAGE FOR NOTES

Chapter 6: THE CATALOGUING OF FUNCTIONS, PROGRAMS, AND FILES

## A. <u>Introduction</u>

Serious work on the HP-41 system requires the manipulation of the various functions, programs, and files available. Efficient use of these requires a knowledge of where they reside among the modules and peripherals in the HP-41 system. A catalog is a list of functions, programs, or files in some medium, along with any other pertinent information, such as file type or program size. Functions, programs, and files can be catalogued using a multitude of functions or OS-41 programs. Additionally, alarms can be catalogued using a Time Module function, and key assignments can be catalogued using an HP-41CX function. Table 6-1 shows the list of possibilities.

The only OS-41 programs in this chapter are used to catalog extended memory and mass storage files. Both OS-41 programs greatly expand the usefulness of hard copy listings of media files by providing extra information in a formatted manner. The lack of OS-41 programs to streamline or expand the other functions described in this chapter is indicative of the considerable attention Hewlett-Packard has given this important aspect of the HP-41 system.

Table 6-1 shows the common media that can be catalogued along with functions and/or programs to obtain catalogs. In Section B, additional details are given so that the user can avoid most references to the HP-41 or peripheral manuals. <u>OS-41 program instructions</u> can be found in Section C.

The cells in Table 6-1 contain one or two numbers; the one in the upper left corner indicates the page in the text where this cell is discussed while the number in parentheses shows the page where the indicated <u>OS-41 program instructions</u> can be found.

-362-

TEXT SUBSECTION NUMBER	MEDIUM	CATALOG OPERATION
1	MAIN MEMORY	364
	programs	CAT 1
	modules/peripherals	CAT 2
	HP-41 functions	CAT 3
	alarms	T: ALMCAT
		CX: CAT 5
	key assignments	CX: CAT 6
2	EXTENDED MEMORY	365 XF: EMDIR
		CX: CAT 4
		CX: EMDIRX
		"XMCAT"(368)
3	MASS STORAGE (MS)	366 IL: DIR
		XIO: DIRX
		"MSCAT"(372)

Table 6-1. Functions and Programs to Catalog Media in the HP-41 System.

### B. Functions, Programs, and Their Use

The CAT functions in the HP-41 have several basic operations, which are detailed below. Press [shift][CATALOG] followed by the appropriate number key to execute a CAT function.

While the catalog is being listed:

- Pressing any key <u>except</u> [R/S] or [ON] either slows down the listing if an HP-41C or HP-41CV is used, or speeds up the listing if an HP-41CX is used;
- (2) Pressing [R/S] stops the catalog listing.

While the catalog listing is stopped:

- (1) Pressing [SST] displays the next item in the catalog;
- (2) Pressing [shift][BST] displays the previous item in the catalog;
- (3) Pressing [R/S] restarts the catalog listing;
- (4) Pressing [<-] exits the catalog.

The catalog listings are printed if the printer is connected and in TRACE mode.

## 1.1/ MAIN MEMORY CATALOG

a. Use CAT 1

to catalog all programs in main memory in the order they are stored. The end of a program <u>file</u> (containing all lines from 01 to END) is indicated by the display of the file's END statement. If the printer is connected in TRACE mode, the number of bytes used by each program file is printed along with its END statement. Additionally, the HP-41CX will <u>display</u> the number of bytes used along with the END statement. When completed, the HP-41 will display the number of main memory program registers available in the form

.END. REG rrr,

where

<u>rrr</u> = number of unused main memory program registers available.

b. Use CAT 2

to catalog all functions or programs in inserted modules or peripherals. Using an HP-41C or HP-41CV, the functions and programs are listed after a header describing where they reside. Using an HP-41CX, the headers are listed sequentially, and a function and program listing is obtained by pressing [R/S] then [ENTER†] when the desired header is displayed. Programs (which can be copied to main memory with COPY) are distinguished from functions (which can only reside in the module or peripheral) by the presence of the " $\tau$ " symbol before the program name.

c. Use CAT 3

to catalog all HP-41 functions alphabetically. All "illegal" catalogs (such as CAT 9) also perform this function.

- d. To catalog all alarms in main memory, use
  - (1) T: ALMCAT, or
  - (2) CX: CAT 5

CAT 5 acts identically to ALMCAT, listing each alarm time and date chronologically along with its message (if any). Many options are available to reset, inspect, or delete alarms; refer to the Time Module Owner's Manual or the HP-41CX Owner's Manual for more details. If the printer is connected in TRACE mode, ALMCAT prints the catalog listing.

e. HP-41CX users can use the function

CX: CAT 6

to catalog all assigned keys and the functions or programs assigned to them.

- 2. EXTENDED MEMORY CATALOG
  - a. To catalog files in extended memory, use
    - (1) XF: EMDIR,
    - (2) CX: CAT 4, or
    - (3) CX: EMDIRX (X:file position number) $\frac{27}{2}$

 $<sup>^{2/}</sup>$ Throughout this text, the parameters needed in the stack registers are indicated in parentheses.

CAT 4 acts identically to EMDIR, listing each file in extended memory in the order that it's stored, along with its file type (A = ASCII, D = data, or P = program) and number of extended memory registers used. If the printer is connected in TRACE mode (or NORM mode if an HP-41C or HP-41CV is used), EMDIR prints the catalog listing. When completed, the number of unused extended memory registers available is placed in the X register.

EMDIRX places in ALPHA the name of the file whose position in the directory is specified in X, and replaces X with its file type (AS = ASCII, DA = data, or PR = program).

b. To catalog the contents of extended memory, including: the time and date catalogued (if printer connected); the name, type, and size of all files; and the amount of unused extended memory, use the OS-41 program

"XMCAT" (eXtended Memory CATalog).

## 3. MASS STORAGE CATALOG

- a. To catalog files in mass storage, use
  - (1) IL: DIR, or
  - (2) XIO: DIRX (X:file position number)

DIR lists each file in mass storage in the order that it is stored in the directory, along with its file type (AS = ASCII, DA = data, KE = key assignment, PR = program, ST = status, WA = write all, or ?? = unknown type), special options (if any) (P = private program, A = automatic execution, or S = secure), and number of mass storage registers used. If the printer is connected, DIR prints the catalog listing.

DIRX places in ALPHA the name of the file whose position in the directory is specified in X.

b. To catalog the contents of a mass storage medium, including: the medium name, the time and date catalogued, and the date the medium was last backed-up (if printer connected); and the name, type, and size of all files, use the OS-41 program

"MSCAT" (<u>Mass Storage CATalog</u>).

## C. <u>Catalog Program Instructions</u>

The program instructions found on the following pages should enable the user to apply the desired program. An example is provided to enable practice on a problem with "known results." As an organizational convenience, note that the terminal part of the "program number" in all cases is the same as the "cell number" in Table 6-1 and the text "subsection number" in Section B.

Refer to Section D for OS-41 program listings.

XMCAT (eXtended Memory CATalog)

Program No.: 41C2 Other Programs in This File: MSCAT

### Purpose

To catalog the contents of extended memory, including: the time and date catalogued (if printer connected); the name, type, and size of all files; and the amount of unused extended memory.

Minimum SIZE: 000

Required Peripherals

Extended Functions/Memory Module (HP 82180A)

OPTIONAL: Time Module (HP 82182A), used to print time and date catalogued Peripheral Printer (HP 82143A) OR HP-IL Interface Loop Module (HP 82160A) and HP-IL printer, such as the HP 82162A Thermal Printer

#### Instructions

- 1. Execute XMCAT.
- The HP-41 will display "\*WORKING\*". If the printer is not connected, go to Step 5.
- 3. Follow the prompt:
  - a. SET NORM

Set the printer to NORM mode 1/ (set MODE switch to NORM or clear flag 15 and set flag 16), then press [R/S] to continue.

4. The HP-41 will print

EXTENDED MEMORY

If the Time Module is inserted, the HP-41 will print the current time and date in the form

TIME: time, date

where

time = current time in the format (H)H:MM AM, (H)H:MM PM, or HH:MM,

and <u>date</u> = current date in the format MM/DD/YY or DD.MM.YY

The time and date formats are determined by the current CLK12/CLK24 and MDY/DMY settings.

1/HP-41CX users: see Comment 1.

The HP-41 will print the header

NAME TREGS

which indicates the column headings for the catalog: file name, type of file, and number of registers used.

5. The HP-41 will display (and print, if the printer is connected) a catalog of each file in extended memory in the form

file t regs

where

 $\frac{file}{t} = \text{extended memory file name,}$   $\frac{t}{t} = \text{type of file (A = ASCII, D = data, or P = program),}$ and  $\frac{regs}{regs} = \text{number of extended memory registers used by file.}$ 

If extended memory contains no files, the HP-41 will display (and print, if the printer is connected) "DIR EMPTY".

When all extended memory files have been listed, the HP-41 will display (or print, if the printer is connected) the number of unused registers remaining in the form

REGS: <u>uuu</u> (if printer not connected) OR REGS REMAINING: <u>uuu</u> (if printer connected)

where

<u>uuu</u> = number of unused registers in extended memory.

6. When completed, the HP-41 will beep, stop, and display "DONE".

 Print a catalog of extended memory, with the Time Module inserted. (Output will differ depending on extended memory contents and time catalogued.)

<u>Step</u>	Prompt	User Inpu	tDis	play_	Print (HP	er Outpu <sup>.</sup> 82162A)	t 
1. 2. 3a.	SET NORM	[XEQ]"XMCAT (Set printer to N( [R/S]	" *WOR DRM mode)	KING*			
5.			ABC DEF GHI	A005 D010 P015	EXTENI Time: Name Abc Def Ghi	ED MEMORY 12:00 PM, 1 TRECC 4005 D010 P015	12/25/84
6.			DONE		REGS F	EMAINING: S	564

## <u>Comments</u>

 XMCAT cannot be used effectively on the HP-41CX because it will only print the catalog if the printer is set to TRACE mode, which causes messy output.

## USE THIS PAGE FOR NOTES

MSCAT (<u>Mass S</u>torage <u>CAT</u>alog)

Program No.: 41C3 Other Programs in This File: XMCAT

### Purpose

To catalog the contents of a mass storage medium, including: the medium name, the time and date catalogued, and the date the medium was last backed-up (if printer connected); and the name, type, and size of all files.

## Minimum SIZE: 000

#### Required Peripherals

Extended Functions/Memory Module (HP 82180A) HP-IL Interface Loop Module (HP 82160A) HP-IL mass storage device, such as the HP 82161A Digital Cassette Drive OPTIONAL: Time Module (HP 82182A), used to print time and date catalogued Peripheral Printer (HP 82143A) OR HP-IL printer, such as the HP 82162A Thermal Printer

#### Instructions

- 1. If the printer is connected, set it to MAN mode (set MODE switch to MAN or clear flags 15 and 16).
- 2. Execute MSCAT.
- 3. The HP-41 will display "\*WORKING\*". If the printer is not connected, go to Step 6.
- 4. If the mass storage ASCII file "NaMe" does not exist, follow the prompt:
  - a. MS NAME?

Key in the desired name of the mass storage medium (<u>ten</u> <u>characters</u> or less) and press [R/S]. The HP-41 will display "\*WORKING\*" as it creates the "NaMe" file and stores the medium name in it.

5. The HP-41 will print

MASS STORAGE: name

where

<u>name</u> = mass storage medium name.

If the Time Module is inserted, the HP-41 will print the current time and date in the form

TIME: time, date

where

time = current time in the format (H)H:MM AM, (H)H:MM PM, or HH:MM,

and <u>date</u> = current date in the format MM/DD/YY or DD.MM.YY

The time and date formats are determined by the current CLK12/CLK24 and MDY/DMY settings.

The HP-41 will print the date the medium was last backed-up with  $\mbox{MSDUP}^{\perp\prime}$  or  $\mbox{MSDUP}^{\perp\prime}$  in the form

LAST BACKUP: <u>date</u>

where

date = date medium was last backed-up in the format MM/DD/YY, DD.MM.YY, or free-form.2/

The date format is determined by the MDY/DMY setting when MSDUP or MSDUPA was last used, and is free-form if the Time Module wasn't inserted. If the medium has never been backed-up, the HP-41 will print

LAST BACKUP: NONE

The HP-41 will print the header

NAME TYPE REGS

which indicates the column headings for the catalog: file name, type of file, and number of registers used.

6. The HP-41 will display (and print, if the printer is connected) a catalog of each file in the mass storage medium in the form

<u>file</u> OR	<u>type</u>	(if pi	rinter	not c	onnected)	
<u>file</u>	type	reas	(if pr	inter	connected)	,

where

- - -

file = mass storage medium file name, type = type of file (described below), and regs = number of mass storage registers used by file.

 $<sup>^{1/}</sup>$ These OS-41 duplication programs are described in Chapter 4.

<sup>2&#</sup>x27;"free-form" is meant to imply the form used when either MSDUP or MSDUPA was last employed.

The type of file can be one of these seven: AS (ASCII), DA (data), KE (key assignment), PR (program), ST (status), WA (write all), and ?? (unknown type). Furthermore, the file type can be appended with one or more of these special options, separated by commas: P (private program), A (automatic execution), and S (secure).

7. When completed, the HP-41 will beep, stop, and display "DONE".

### Examples

 Print a catalog of the new mass storage medium "OS-41 TEST", which doesn't yet have the "NaMe" file, with the Time Module inserted. (Output will differ depending on medium contents and time catalogued.)

<u>Step</u>	Prompt	User Inp	out	_Display_	Print <u>(HP</u>	er Outpu 82162A)	t -
1. 2.	( Se	et printer to [XEQ]"MSCA	MAN m	*WORKING*			
4a. 5.	MS NAME?	OS-41 TEST	[R/S]	*WORKING*			
					MASS S Time: Last B	TORAGE: OS- 12:00 PM, 0 Ackup: None	41 TEST 18/26/83
					NAME	TYPE	REGS
6.		ABC	AS	5	ABC	AS DO C	5 1 a
			DA,S	S 15	UEF CHI	рнур Рр.р.с	15
7.		NaMe DONE	AS,S	3 15	NaMe	AS,S	

#### Comments

- 1. If the printer is connected, MSCAT temporarily creates the extended memory ASCII file "NaMe". Therefore, three extended memory registers must be available and "NaMe" must <u>not</u> already exist in extended memory or an error message will be displayed.
- 2. Because it is frequently accessed, the "NaMe" file should be at the beginning of a mass storage medium. Therefore, MSCAT should be used immediately after a medium is initialized to establish the "NaMe" file as the first file. Alternatively, the "NaMe" file can be placed at the beginning of a medium with several files on it by temporarily copying the first file to another medium (for example, to extended memory using MS2XMD), purging the file with MSX, using MSCAT to place the "NaMe" file at the beginning of the mass storage medium, and copying the original first file back to mass storage (for example, from extended memory using XM2MSD).
- 3. Refer to Chapter 4, Section D for an explanation of the "NaMe" file format.

# D. <u>Program Listings</u>

The program listing for XMCAT and MSCAT on the following page was printed by an HP 82162A Thermal Printer set to MAN mode. Barcode for XMCAT and MSCAT can be found on page 464.

### OS-41 Catalog Program Listing: XMCAT, MSCAT

Program Registers: 51 Bytes: 356

01+LBL "XMCAT" 02 •4102• 03 SF 05 84 GTO 88 05+LBL "MSCAT" **86 \*41C3\*** 07 CF 05 08+LBL 00 09 XEQ 07 10 CF 29 11 FC? 55 12 GTO 04 13 FS? 05 14 GTO 01 15 "NaMe" 16 3 17 CRFLAS 18 SF 25 19 GETAS 20 FC?C 25 21 XEQ 06 22+LBL 01 23 "SET NORM" 24 FS? 05 25 PROMPT 26 ADV 27 "EXTENDED MEMORY" 28 FC2 05 29 MASS STORAGE: " 30 FC? 05 31 ARCLREC 32 PRA 33 \*TIME: \* 34 SE 25 35 TIME 36 FC? 25 37 XEQ 07 38 FC?C 25 39 GTO 02 40 RCLFLAG 41 FIX 2 42 X(>Y 43 ATIME 44 •⊢, • 45 DATE 46 FIX 4 47 ADATE 48 RCL Z 49 STOFLAG 50 PRA

51+LBL 02 52 FS? 05 53 GTO 03 54 \*LAST BACKUP: \* 55 ARCLREC 56 PRA 57 "NaMe" 58 PURFL 59+LBL 03 60 ADV 61 "NAME TREGS\* 62 FS? 05 63 PRA 64+LBL 04 65 FS? 05 66 EMDIR 67 FC? 05 68 DIR 69 ADV 70 FC?C 05 71 GTO 05 72 "REGS" 73 FS? 55 74 "F REMAINING" 75 •F: • 76 RCLFLAG 77 FIX 0 78 ARCL Y 79 STOFLAG 80 FS? 55 81 PRA 82 FC? 55 83 AVIEW 84 ADV 85+LBL 05 86 "DONE" 87 ASTO X 88 BEEP 89 CLD 90 RTN 91+LBL 06 92 TONE 0 93 "MS NAME?" 94 AON 95 PROMPT 96 AOFF 97 ASTO Z 98 ASHF 99 ASTO Y 100 XEQ 07

101 "NaMe" 102 CREATE 103 CLA 104 ARCL Z 105 ARCL Y 106 APPREC 107 "NONE" 108 APPREC 109 0 110 SEEKPT 111 "NaMe" 112 SAVEAS 113 VERIFY 114 SEC 115 RTN 116+LBL 07 117 \* \*WORKING\*\* 118 CF 21 119 AVIEW 120 FS? 55 121 SF 21 122 END

#### LITERATURE CITED

- Beers, T. W. 1980. <u>Prose Storage and Recall on the HP-41C</u>. Proceedings, PPC Midwest Conference, Melrose Park, IL. June 14, 1980.
- Erickson, C. A. 1984a. <u>More 82905B/HP-41 Bar Code Printing</u>. PPC Calculator Journal. V11N2p23. PPC, Box 9599, Fountain Valley, CA. 92728-9599.

Journal. V11N4p5. PPC, Box 9599, Fountain Valley, CA. 92728-9599.

- Hill, S. M. 1983. <u>"STEX"</u>. HP Key Notes. V7N1p6b. Hewlett-Packard Co., Corvallis, OR. 97330.
- Jarett, K. 1982. <u>HP-41 Synthetic Programming Made Easy</u>. Synthetix, 1540 Mathews Ave., Manhattan Beach, CA. 90266.
- . 1983. <u>HP-41 Extended Functions Made Easy</u>. Synthetix, P.O. Box 113, Manhattan Beach, CA. 90266.
- Krabach, M. 1984. <u>Direct Tape Save of EFM and EM</u>. PPC Calculator Journal. V11N3p20b. PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- Lafoiet, C. 1982. <u>"STOL"</u>. HP Key Notes. V6N3p12a. Hewlett-Packard Co., Corvallis, OR. 97330
- Meyer, R. 1980. <u>Psalm 23 on the HP-41C</u>. PPC Calculator Journal. V7N4p14. PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- Moak, D. S. 1983. <u>FEEDBACK Letter Regarding "SAVEM"</u>. PPC Calculator Journal. V10N7p8 and V10N10p37. PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- PPC. 1981. <u>PPC ROM User's Manual</u>. Personal Programming Center, PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- Reinstein, C. E. 1982. <u>HP-41/HP-IL System Dictionary</u>. Corvallis Software, Inc., Box 1412, Corvallis, OR. 97330.
- \_\_\_\_\_\_. 1983. <u>HP-41/HP-IL System Dictionary</u>. PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- Ressel, W. 1983. <u>Automatic Synthetic Text Line Decoder</u>. PPC Calculator Journal. V10N6p25. PPC, Box 9599, Fountain Valley, CA. 92728-9599.
- Wickes, W. C. 1980. <u>Synthetic Programming on the HP-41C</u>. Larken Publications, P.O. Box 987, College Park, MD. 20740.

Appendix A: OS-41 FLAG AND DATA REGISTER USAGE

## 1. Flag Usage

All OS-41 programs that require flags to maintain some program-defined status employ HP-41 flags starting with user flag 05 (the lowest-numbered flag that isn't indicated by an annunciator) and proceeding upward if necessary. OS-41 programs MSDUP and MSDUPA also use flags 00-04 as required by the IL: INSTAT function. Additionally, many OS-41 programs manipulate some control flags (flags 11-29) and system flags (flags 30-55) as required for proper operation.

Table A-1 describes how each OS-41 program manipulates user flags 00-10 and system flags 36-41 (the six flags which together define the display mode of the HP-41). A small x in a flag column indicates that flag is used by the program. Any user flag altered by an OS-41 program is deliberately cleared before program completion. The Other Flags Used column contains additional flag information that need be considered to execute the program as a subroutine or otherwise alter the usual program procedure.

Table A-2 shows the general conditions under which OS-41 programs manipulate the various control flags. With the exception of flags 17 and 21, any control flag altered by an OS-41 program is deliberately cleared before program completion. Flag 17 is left as it was last set; aand flag 21 is set or cleared to match flag 55, the flag which indicates if a printer is connected.

-378-

Table A-1. OS-41 User Flag Usage.

		Flags	5: x =	used						
Program	05	06	Displ	ay Mode			Other	Flags	Used	
M2MP										
M2XMP	x	×								
M2MSP	x	×			06 se	t i	f secured	file	to be	replaced
XM2MP	×									
XM2XMP	×	×								
XM2MSP	×	×			06 se	t i	f secured	file	to be	replaced
MS2MP	×									
MS2XMP	×	×								
MS2MSP	x	x			06 se	t i	f secured	file	to be	replaced
B2XMP	x	×								
B2MSP	x	x			06 se	t i	f secured	file	to be	replaced
M2MD										
M2XMD	x									
M2MSD	x									
M2CD	×									
M2XD										
M2PD			FIX (#	DECIMALS)						
M2DD			FIX (#	DECIMALS)	00 se	t ho	olds disp	lay		
XM2MD	×									
XM2XMD	×	×			06 se	t if	f changing	g file	e size	
XM2MSD	×	×								
XM2CD	×	x								
XM2XD	×									
XM2PD	×	×	FIX (#	DECIMALS)	06 se	t if	f II>l			
XM2DD	×	×	FIX (#	DECIMALS)	00 se	t ho	olds disp	lay; O	6 set	if II>1

Program	05	F1ag 06	gs: x = used Display Mode	Other Flags Used
MS2MD	×			
MS2XMD	×	x		
MS2MSD	×	x		06 set if changing file size
MS2CD	x	x		
MS2XD	x			
MS2PD	x	x	FIX (# DECIMALS)	06 set if II>1
MS2DD	×	x	FIX (# DECIMALS)	00 set holds display; 06 set if II>1
C2MD	×			
C2XMD	x	x		
C2MSD	×	×		
C 2CD				
C2PD			FIX (# DECIMALS)	
C2DD			FIX (# DECIMALS)	00 set holds display
X2MD			FIX 0	
X2XMD	×	x	FIX 0	06 set if II>1
X 2MSD	×	x	FIX O	06 set if II>1
B2MD			FIX O	
B2XMD	x	×	FIX 0	06 set if II>1
B2MSD	×	×	FIX O	06 set if II>1
B2PD			FIX (# DECIMALS)	
B2DD			<pre>FIX (# DECIMALS)</pre>	
M2XMA	×			
XM2MA				
XM2XMA	×	x		05 set if changing file size
XM2MSA	×			

Table A-1. OS-41 User Flag Usage (continued).

		Flags:	x = used	
Program	05	06	Display Mode	Other Flags Used
XM2CA				
XM2AA				
XM2PA		FI	X 0	
XM2DA		FI	X 0	00 set holds display
MS2XMA	x			
C2XMA	×			
A2XMA	×	FI	X 0	
B2XMA	x	FI	X 0	
M2XME	x	FI	X 0	
M2MSE	x	FI	X 0	
XM2ME	×			
XM2MSE	x	×		06 set if # XMS=1; 07 set if # XMS=2
MS2ME	x			
MS2XME	×	×		07 used
MSDUP	x	×		00-08 used; 09 set if backup selected
MSDUPA	x	×		00-08,10 used; 09 set if backup selected
MZ				
MX				
XMZ	x			
XMX	x			
MSZ				
MSX	×			
XMCAT	×			
MSCAT	x			

Table A-1. OS-41 User Flag Usage (continued).

Tab	le	A-2.	0S-41	Control	Flag	Usage.
-----	----	------	-------	---------	------	--------

Flag	Control Purpose	OS-41 Usage
17	not end-of-record	Used by programs that move ASCII data from extended memory.
21	printer enable	Used by programs that have display-only messages.
22	numeric data input	Used by programs that detect lack of numeric data input.
23	alphanumeric data input	Used by programs that detect lack of alpha- numeric data input.
25	error ignore	Used by programs that detect error conditions without halting.
29	separator mark	Used by programs that set the display to FIX 0.

## 2. <u>Register Usage</u>

All OS-41 programs that require main memory temporary data registers to store parameters use registers starting with OO and proceeding upward if necessary. Some OS-41 programs require an <u>additional</u> block of temporary registers defined by the user. These registers must start after the last register already used by a program. Almost without exception, OS-41 programs use most of the stack registers and the ALPHA register.

Table A-3 describes how each OS-41 program uses main memory data registers for temporary storage. A semicolon separates the major purposes of a register that has multiple uses. The primary purpose indicated in each register reflects its contents after the required parameters have been entered. Thus, Table A-3 is useful for setting up a program to be executed as a subroutine. The Comments column contains additional data register usage details. Note that no mention is made of the main memory data

-382-

registers that are known to be used only by the user at the time he is specifically copying data to or from them.

Table A-3. OS-41 Data Register Usage.

	Comments											
	Other											
	02											
Registers	01											
	00											
	Stack	X: MM PROGRAM	Y: MM PROGRAM X: XM FILE	Y: MM PROGRAM X: MS FILE	X: XM FILE	Y: <u>file</u> PRGM X: XM DST FILE L: XM SRC FILE	Y: <u>file</u> PRGM X: MS FILE L: XM FILE	X: MS FILE	Y: <u>file</u> PRGM X: XM FILE L: MS FILE	Y: <u>file</u> PRGM X: MS DST FILE L: MS SRC FILE	Y: XM FILE X: <u>file</u> PRGM	Y: MS FILE X: <u>file</u> PRGM
	Program	M2MP	M2XMP	MZMSP	XM2MP	XM2XMP	XMZMSP	MS 2MP	MS2XMP	MS 2MSP	B2XMP	B2MSP

	Comments									TMP bbb through eee used 08 used only if changing file size	TMP bbb through eee used
	Other									03: SRC bbb.eee 04: DST bbb 05: # tmp registers 06: # iterations 07: remainder 08: NEW SIZE	03: XM bbb.eee; # iterations 04: MS bbb; remainder
	02									TMP bbb.eee	TMP bbb.eee
Registers	01									XM DST FILE	MS FILE
	00									XM SRC FILE	XM FILE
	Stack	Y: SRC bbb.eee X: DST bbb	Z: XM FILE Y: MM bbb.eee X: XM bbb	Z: MS FILE Y: MM bbb.eee X: MS bbb	X: MM bbb.eee	X: MM RRR	Y: MM bbb.eeeII X: # DECIMALS	Y: MM bbb.eeeII X: # DECIMALS	Z: XM FILE Y: XM bbb.eee X: MM bbb		
	Program	M2MD	MZXMD	MZMSD	M2CD	MZXD	M2PD	M2DD	QMZMX	XM2XMD	XM2MSD

Program XM2CD XM2ZD XM2PD	Stack	00 XM FILE XM FILE XM FILE	Registers 01 XM bbb.eee XM RRR XM bbb.eeeII	02 TMP bbb; tmp bbb.eae # DECIMALS	Other	Comments TMP bbb through bbb + (XM eee - bbb) used
DM22MX	Z: MS FILE Y: MS bbb.eee X: MM bbb	XM FILE	XM bbb.eeeII	# DECIMALS		
<b>DWXZ SW</b>		MS FILE	XM FILE	TMP bbb.eee	03: MS bbb.eee; # iterations 04: XM bbb.eee; remainder	TMP bbb through eee used
MSZMSD		MS SRC FILE	MS DST FILE	TMP bbb.eee	03: SRC bbb.eee 04: DST bbb 05: # tmp registers 06: # iterations 07: remainder 08: NEW SIZE	TMP bbb through eee used 08 used only if changing file size
MS 2CD MS 2XD	X: TMP RRR	MS FILE MS FILE	MS bbb.eee MS RRR	TMP bbb; tmp bbb.eee		TMP bbb through bbb + (MS eee - bbb) used TMP RRR used
MS 2PD		MS FILE	MS bbb.eeeII	# DECIMALS	03: TMP RRR	TMP RRR used
MS 2DD C 2MD	X: MM bbb.eee	MS FILE	MS bbb.eeeII	# DECIMALS	03: TMP RRR	TMP RRR used

Program	Stack	00	Registers 01	02	Other	Comments
CZXMD		XM FILE	XM bbb.eee	TMP bbb; tmp bbb.eee		TMP bbb through bbb + (XM eee - bbb) used
CZMSD		MS FILE	MS bbb.eee	TMP bbb; tmp bbb.eee		TMP bbb through bbb + (MS eee - bbb) used
czcD		TMP bbb.4				TMP bbb through SIZE-1 used
C2PD		# DECIMALS	TMP bbb; tmp bbb.4			TMP bbb through SIZE-l used
CZDD		# DECIMALS	TMP bbb; tmp bbb.4			TMP bbb through SIZE-l used
X 2MD	X: MM bbb.eeeII					
X 2XMD		XM FILE	XM bbb.eeeII			
X 2MSD		MS FILE	MS bbb.eeeII	TMP RRR		TMP RRR used
B2MD	X: MM bbb.eeeII					
BZXMD		XM FILE	XM bbb.eeeII			
BZMSD		WS FILE	MS bbb.eeeII	TMP RRR		TMP RRR used
B2PD	X: # DECIMALS					
B2DD	X: # DECIMALS					
MZXMA	Y: XM FILE X: MM bbb					
XMZMA	Y: XM FILE X: MM bbb					

Table A-	3. OS-41 Data Re	gister Usage (c	continued).			
Program	Stack	00	Registers 01	02	Other	Comments
XM2XMA		XM SRC FILE	XM DST FILE	src pointer	03: dst pointer 04: NEW SIZE	04 used only if changing file size
XMZMSA		XM FILE	MS FILE			
XM2CA		XM FILE	TMP bbb			TMP bbb through bbb + (regs <sup>1/</sup> -1) used
XM2AA		XM FILE	XM RRR.CCC			
XM2PA	X: XM bbb.eee	XM FILE				
XM2DA	X: XM bbb.eee	XM FILE				
<b>MS2XMA</b>		MS FILE	XM FILE			
CZXMA		XM FILE	TMP bbb			TMP bbb through bbb + (regs <sup>1/</sup> -1) used
AZXMA		XM FILE				
BZXMA		XM FILE				
M2XME	Х: ТМР ЬЬЬ	XM FILE; xm file.				TMP bbb through bbb + (# prgms - 1) used
M2MSE	Х: ТМР ЬЬЬ	MS FILE; ms file.				TMP bbb through bbb + (# prgms - 1) used
XM2ME	X: XM FILE					
XM2MSE	X: # XMS	MS FILE				
MS 2ME	X: MS FILE					

 $1/regs \leq INT$  ((7 \* (file size) + 5 \* (# recs) + 4)/6) (suggested by Jarett, 1983).

Program	Stack	00	Registers 01	02	Other	Comments
MS 2XME		MS FILE				
MSDUP		# devices	src address	lst ms address	03: DATE 04: DATE (cont'd)	03-04 used only if no Time Module
MSDUPA		# devices; l.nnnnn; l.nnn	src address; see comments	lst ms address; see comments		0l through (# files - 1) used nnn = # of files
ZW	X: MM bbb.eeeII					
WX						
ZMX		XM FILE	XM bbb.eeeII			
XMX						ALPHA: XM FILE(S)
ZSW		MS FILE; # iterations	MS bbb.eee; remainder	TMP bbb.eee		TMP bbb through eee used
MSX						ALPHA: MS FILE(S)
XMCAT						
MSCAT						

Appendix B: THE USE OF OS-41 PROGRAMS AS SUBROUTINES

A primary purpose of the OS-41 programs is to free the user from the responsibility of setting arbitrary parameters required to perform HP-41 system operations. To this end, all OS-41 programs prompt the user in plain English for all required parameters in (hopefully) a sensible and consistent fashion. However, the usefulness of OS-41 programs can be extended by the ability to call them as <u>subroutines</u>, which are non-prompting and hence act effectively as functions to the calling program. The calling program can set input parameters in any manner, then continue after calling an OS-41 subroutine.

All OS-41 programs distinguish whether they are being used as programs or subroutines by the status of flag 50 (message flag). If flag 50 is clear, as is always the case when a program is executed from the keyboard, an OS-41 program prompts for input parameters as detailed in its Instructions section. If, however, flag 50 is set, which means data are being VIEWed or AVIEWed, then an OS-41 program skips its input prompts and immediately begins operation. Of course, the input parameters must be stored and any relevant flags set or cleared before using an OS-41 program as a subroutine. When completed, an OS-41 program operating as a subroutine returns to the calling program.

The general procedure that a calling program must follow before executing an OS-41 program as a subroutine is as follows:

- 1. Insure that the minimum SIZE is set (described in each program's Instructions section).
- Set the status of all user flags required by the subroutine (described in Appendix A, Section 1).
- 3. Store all input parameters in the proper main memory data registers (described in Appendix A, Section 2).

4. VIEW or AVIEW data to signal the subroutine not to prompt.

-390-

For example, consider a program XM2PS (e<u>X</u>tended <u>Memory TO</u> <u>Printer</u>, <u>Special</u>) designed to use the OS-41 program XM2PD, but without prompting for the required parameters. Instead, XM2PS <u>automatically</u> prints the contents of extended memory file "TEST" registers 05, 08, and 11 shown to three decimal places. The entries from Tables A-1 and A-3 in Appendix A are shown below, indicating that main memory data registers 00-02 and flags 05-06 are used. Since II>1, flag 06 must be set.

Program	05	F1ag 06	Flags: x = used O6 Display Mode		ə O	ther F	lags Used	l
XM2PD	×	×	FIX (# DECIMALS		_S)	06 set	if II>1	
			F	ərs				
00		0	01		02		Other	Comments
XM2PD	XM I	FILE	XM bbb.e	eeII	# DEC	IMALS		

The program below follows the general setup procedure described above:

LBL"XM2PS" SIZE? 3 Step 1: insures SIZE  $\geq$  003 X>Y? PSIZE SF 06 Step 2: since II > 1 "TEST" ASTO 00 5.00803 Step 3: stores needed parameters STO 01 3 STO 02 " \*WORKING\*" Step 4: sets flag 50, signalling XM2PD use as a CF 21 subroutine AVIEW XEQ"XM2PD" END

Note that most OS-41 programs clear the display before returning control to the calling program.

Appendix C: OS-41 SUPPORT PROGRAMS

Several programs were helpful in the preparation of this text. They aren't really OS-41 programs; rather, they "support" the OS-41 programs since they aid in making the Operating System understandable and useful.

The program PRPR (<u>PR</u>int <u>PR</u>ogram) was used to print the header for the program listings in this text. PRPR prints the number of main memory program registers required by a program and its size in bytes. PRPR is listed below.

Program Listing for PRPR:

Program Registers: 15 Bytes: 103

01+LBL "PRPR"	17 INT
A2 FIX A	18 ADV
<b>R</b> 3 CE 29	19 "Program Registe"
04 "NM PROGRAM?"	20 "Frs: "
95 ANN	21 ARCL X
AG PROMPT	22 PRA
07 90FF	23 <b>"</b> Bytes: "
AR OSTA X	24 ARCL Y
00 noto n 09 *L. *	25 PRA
10 COVEP	26 ADV
	27 •*
10 DOI V	28 PURFL
12 RUL A 17 Z	29 <b>•</b> PRP •
10 0	30 ARCI 7
14 +	31 PROMPT
15 (	72 END
16 /	JZ LNP

Data barcode provided in some Example sections was produced with the program PRBCD (<u>PRint BarCode</u>, <u>Data</u>). PRBCD uses the Plotter Module (HP 82184A) to print HP-41 numeric or alphanumeric data barcode on the HP 82162A Thermal Printer. PRBCD is listed below.
Program Listing for PRBCD:

Program Registers: 19 Bytes: 128 01+LBL "PRBCD" 21 SCI IND X **R**2\_CE\_12\_ 22+LBL 01 83 CF 13 23 **DATA** 04 "BC TYPE? A/D" 24 FC? 05 **0**5 CF 21 25 "+=" 06 AVIEW 26 • + ? • 07 SF 21 27 FS? 05 08+LBL 00 28 AON **09 GETKEY** 29 PROMPT 10 X=0? 30 AOFF 11 GTO 00 31 RND 12 CF 05 32 14 13-11 33 ALENG 14 X=Y? 34 X>Y2 15 SF 05 35 GTO 01 16 \*PRECISION=?\* 36 ADV 17.9 37 34 18 FC? 05 38 FS? 85 19 PROMPT 39 ACCHR 20 FC? 05 40 RCL T

41 FS? 05 42 ACA 43 FC? 05 44 ACX 45 X<>Y 46 FS? 05 47 ACCHR 48 X<>Y 49 FS? 05 50 BCA 51 FC?C 05 52 BCX 53 CHS 54 BC0 55 END

The program PRSYNT (<u>PRint SYN</u>thetic <u>Text lines</u>) was used to print the hexadecimal equivalents of synthetic text lines in those OS-41 programs which have them. PRSYNT also decodes synthetic TONE instructions, thus providing the means to uniquely describe all synthetic lines in a program, since all other synthetic lines are readable directly from a program listing. PRSYNT is based on a program written by Ressel (1983).

All OS-41 program barcode was produced by the program PRBCOS (<u>PR</u>int HP 82905B <u>BarCode</u>, <u>OS</u>-41). PRBCOS, which was inspired by the BAR program written by Erickson (1984a and 1984b), uses the Plotter Module (HP 82184A) to convert program bytes to barcode bit strings, then formats the program barcode output for the HP 82905B Printer. Users interested in generating their own barcode on this or other printers are referred to the summary article by Erickson (1984a). Appendix D: BARCODE FOR PROGRAMS

The OS-41 program barcode listings on the following pages are arranged according to the order of the program listings in Section D of Chapters 1-6. Refer to the index of program listings in Section D of the chapter describing the desired program to locate the page number of any OS-41 program barcode listing, or locate the program name in the master INDEX following the Appendices.

As a further convenience, the groups of barcode begin on pages indicated below.

Chapter	Subject	Page
1	Programs	395
2	Data: numeric and alphanumeric	405
3	Data: ASCII	432
4	Entire Media	449
5	Purge and Clear	459
6	Cataloguing	464

\* \* \* REMINDER \* \* \*

Protect barcode with a transparent plastic sheet while scanning with the Optical Wand.

Unauthorized duplication of the barcode included in this publication represents a violation of the Copyright Laws.

OS-41 Program Barcode: M2MP Program Registers Required: 13 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 11 ROW 4: LINES 11 - 14 ROW 5: LINES 15 - 18 ROW 6: LINES 18 - 23 ROW 7: LINES 23 - 27 

OS-41 Program Barcode: M2XMP, M2MSP Program Registers Required: 29 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 13 ROW 5: LINES 13 - 18 ROW 6: LINES 18 - 22 ROW 7: LINES 22 - 29 ROW 8: LINES 29 - 36 ROW 9: LINES 36 - 38 ROW 10: LINES 39 - 45 ROW 11: LINES 46 - 53 ROW 12: LINES 53 - 57 ROW 13: LINES 58 - 64 ROW 14: LINES 64 - 71 ROW 15: LINES 71 - 76 

M2XMP, M2MSP (continued) ROW 16: LINES 76 - 80 OS-41 Program Barcode: XM2MP, MS2MP Program Registers Required: 16 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 14 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 21 ROW 7: LINES 22 - 29 ROW 8: LINES 29 - 35 ROW 9: LINES 36 - 38 

OS-41 Program Barcode: XM2XMP, MS2MSP Program Registers Required: 36 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 13 ROW 5: LINES 14 - 16 ROW 6: LINES 16 - 20 ROW 7: LINES 20 - 26 ROW 8: LINES 27 - 32 ROW 9: LINES 33 - 34 ROW 10: LINES 35 - 40 ROW 11: LINES 40 - 44 ROW 12: LINES 45 - 50 ROW 13: LINES 51 - 57 ROW 14: LINES 57 - 63 ROW 15: LINES 63 - 70 

-399-

XM2XMP, MS2MSP (continued)

ROW 16: LINES 70 - 77

ROW 17: LINES 77 - 84

ROW 18: LINES 84 - 91

ROW 19: LINES 91 - 96

OS-41 Program Barcode: XM2MSP, MS2XMP Program Registers Required: 36 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 13 ROW 5: LINES 14 - 16 ROW 6: LINES 17 - 21 ROW 7: LINES 21 - 29 ROW 8: LINES 29 - 33 ROW 9: LINES 33 - 39 ROW 10: LINES 40 - 46 ROW 11: LINES 46 - 49 ROW 12: LINES 50 - 56 ROW 13: LINES 56 - 63 ROW 14: LINES 63 - 68 ROW 15: LINES 68 - 75 

OS-41 Program Barcode: B2XMP, B2MSP Program Registers Required: 34 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 13 ROW 5: LINES 13 - 17 ROW 6: LINES 18 - 22 ROW 7: LINES 22 - 26 ROW 8: LINES 27 - 34 ROW 9: LINES 34 - 36 RDW 10: LINES 36 - 44 ROW 11: LINES 44 - 51 ROW 12: LINES 52 - 53 ROW 13: LINES 53 - 56 ROW 14: LINES 56 - 63 ROW 15: LINES 63 - 70 

B2XMP, B2MSP (continued)

ROW 16: LINES 70 - 77

# ROW 17: LINES 77 - 84

### 

ROW 18: LINES 84 - 88

#### 

OS-41 Program Barcode: M2MD Program Registers Required: 20 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 9 ROW 4: LINES 9 - 12 ROW 5: LINES 13 - 15 ROW 6: LINES 15 - 24 ROW 7: LINES 25 - 35 ROW 8: LINES 36 - 44 ROW 9: LINES 45 - 52 ROW 10: LINES 52 - 57 ROW 11: LINES 58 - 59 

OS-41 Program Barcode: M2XMD, M2MSD Program Registers Required: 31 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 12 ROW 5: LINES 12 - 16 ROW 6: LINES 16 - 20 ROW 7: LINES 21 - 26 ROM 8: LINES 26 - 30 ROW 9: LINES 31 - 37 ROW 10: LINES 38 - 44 ROW 11: LINES 45 - 51 ROW 12: LINES 51 - 58 ROW 13: LINES 58 - 62 ROW 14: LINES 63 - 65 ROW 15: LINES 66 - 73 

M2XMD, M2MSD (continued)

ROW 16: LINES 73 - 75

### 

ROW 17: LINES 76 - 77

OS-41 Program Barcode: M2CD, C2MD Program Registers Required: 18 ROW 1: LINES 1 - 2 **ROW 2: LINES 2 - 5** ROW 3: LINES 5 - 9 ROW 4: LINES 10 - 12 ROW 5: LINES 12 - 15 ROW 6: LINES 15 - 19 ROW 7: LINES 19 - 25 ROW 8: LINES 25 - 27 ROW 9: LINES 27 - 32 ROW 10: LINES 33 - 35 

OS-41 Program Barcode: M2XD

Program Registers Required: 5

ROW 1: LINES 1 - 2

ROW 2: LINES 2 - 5

ROW 3: LINES 5 - 7

OS-41 Program Barcode: M2PD, M2DD Program Registers Required: 21 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 4 ROW 3: LINES 4 - 9 ROW 4: LINES 9 - 9 ROW 5: LINES 10 - 12 ROW 6: LINES 12 - 18 ROW 7: LINES 18 - 21 ROW 8: LINES 22 - 30 ROW 9: LINES 30 - 36 ROW 10: LINES 36 - 43 ROW 11: LINES 44 - 50 ROW 12: LINES 50 - 50 

OS-41 Program Barcode: XM2MD, MS2MD Program Registers Required: 26 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 14 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 23 ROW 7: LINES 23 - 26 ROW 8: LINES 26 - 30 ROW 9: LINES 31 - 32 ROW 10: LINES 33 - 41 ROW 11: LINES 41 - 50 ROW 12: LINES 51 - 61 ROW 13: LINES 61 - 67 ROW 14: LINES 67 - 71 

OS-41 Program Barcode: XM2XMD, MS2MSD Program Registers Required: 84 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 13 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 26 ROW 7: LINES 26 - 28 ROW 8: LINES 28 - 33 ROW 9: LINES 34 - 39 ROW 10: LINES 39 - 41 ROW 11: LINES 41 - 49 ROW 12: LINES 49 - 55 ROW 13: LINES 56 - 62 ROW 14: LINES 63 - 70 ROW 15: LINES 70 - 78 

XM2XMD, MS2MSD (continued) ROW 16: LINES 79 - 89 ROW 17: LINES 90 - 102 ROW 18: LINES 102 - 111 ROW 19: LINES 112 - 122 ROW 20: LINES 123 - 129 ROW 21: LINES 130 - 139 RDW 22: LINES 140 - 146 ROW 23: LINES 147 - 155 ROW 24: LINES 155 - 164 ROW 25: LINES 165 - 173 ROW 26: LINES 173 - 178 ROW 27: LINES 178 - 188 ROW 28: LINES 189 - 196 ROW 29: LINES 196 - 202 ROW 30: LINES 203 - 213 

XM2XMD, MS2MSD (continued) ROW 31: LINES 214 - 223 RDW 32: LINES 224 - 230 ROW 33: LINES 230 - 233 RDW 34: LINES 233 - 237 RDW 35: LINES 238 - 246 ROW 36: LINES 246 - 250 RDW 37: LINES 250 - 252 ROW 38: LINES 252 - 256 ROW 39: LINES 257 - 262 ROW 40: LINES 262 - 266 ROW 41: LINES 267 - 273 ROW 42: LINES 274 - 276 ROW 43: LINES 276 - 278 ROW 44: LINES 278 - 284 ROW 45: LINES 284 - 286 

OS-41 Program Barcode: XM2MSD, MS2XMD Program Registers Required: 55 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 13 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 22 ROW 7: LINES 22 - 28 ROW 8: LINES 29 - 36 ROW 9: LINES 37 - 44 ROW 10: LINES 44 - 52 ROW 11: LINES 53 - 63 ROW 12: LINES 64 - 75 ROW 13: LINES 76 - 84 ROW 14: LINES 84 - 91 ROW 15: LINES 91 - 100 

-415-

ROW 16: LINES 101 - 110 ROW 17: LINES 111 - 116 ROW 18: LINES 116 - 123 ROW 19: LINES 124 - 127 ROW 20: LINES 128 - 136 ROW 21: LINES 137 - 141 RDW 22: LINES 141 - 143 ROW 23: LINES 143 - 148 ROW 24: LINES 148 - 154 ROW 25: LINES 155 - 157 ROW 26: LINES 157 - 163 ROW 27: LINES 164 - 168 ROW 28: LINES 168 - 172 ROW 29: LINES 173 - 175 ROW 30: LINES 176 - 176 

XM2MSD, MS2XMD (continued)

OS-41 Program Barcode: XM2CD, MS2CD, C2XMD, C2MSD Program Registers Required: 54 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 8 ROW 4: LINES 8 - 11 ROW 5: LINES 11 - 15 ROW 6: LINES 15 - 17 ROW 7: LINES 17 - 22 ROW 8: LINES 23 - 27 ROW 9: LINES 27 - 32 ROW 10: LINES 33 - 35 ROW 11: LINES 35 - 39 ROW 12: LINES 39 - 45 ROW 13: LINES 46 - 53 ROW 14: LINES 54 - 60 ROW 15: LINES 61 - 71 

XM2CD, MS2CD, C2XMD, C2MSD (continued) ROW 16: LINES 71 - 77 ROW 17: LINES 77 - 80 ROW 18: LINES 80 - 87 ROW 19: LINES 87 - 91 ROW 20: LINES 91 - 97 ROW 21: LINES 97 - 101 ROW 22: LINES 101 - 103 ROW 23: LINES 104 - 112 ROW 24: LINES 112 - 116 ROW 25: LINES 116 - 120 RDW 26: LINES 120 - 127 ROW 27: LINES 127 - 128 ROW 28: LINES 129 - 133 ROW 29: LINES 133 - 135 

OS-41 Program Barcode: XM2XD, MS2XD Program Registers Required: 25 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 14 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 23 ROW 7: LINES 23 - 29 ROW 8: LINES 29 - 33 ROW 9: LINES 34 - 36 ROW 10: LINES 37 - 44 ROW 11: LINES 45 - 51 ROW 12: LINES 51 - 57 ROW 13: LINES 57 - 61 ROW 14: LINES 61 - 62 

OS-41 Program Barcode: XM2PD, XM2DD, MS2PD, MS2DD Program Registers Required: 44 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 4 ROW 3: LINES 4 - 7 ROW 4: LINES 7 - 8 ROW 5: LINES 9 - 10 ROW 6: LINES 10 - 16 ROW 7: LINES 16 - 21 ROW 8: LINES 21 - 26 ROW 9: LINES 26 - 29 ROW 10: LINES 29 - 34 ROW 11: LINES 35 - 39 ROW 12: LINES 39 - 43 ROW 13: LINES 43 - 48 ROW 14: LINES 48 - 55 ROW 15: LINES 55 - 58 

XM2PD, XM2DD, MS2PD, MS2DD (continued) ROW 16: LINES 59 - 67 ROW 17: LINES 67 - 74 ROW 18: LINES 75 - 82 ROW 19: LINES 82 - 88 ROW 20: LINES 89 - 95 ROW 21: LINES 96 - 101 ROW 22: LINES 101 - 110 ROW 23: LINES 110 - 111 ROW 24: LINES 112 - 115

OS-41 Program Barcode: C2CD Program Registers Required: 25 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 8 ROW 3: LINES 8 - 12 ROW 4: LINES 12 - 20 ROW 5: LINES 21 - 27 ROW 6: LINES 27 - 31 ROW 7: LINES 31 - 37 ROW 8: LINES 38 - 46 ROW 9: LINES 47 - 52 ROW 10: LINES 52 - 55 ROW 11: LINES 56 - 57 ROW 12: LINES 57 - 64 ROW 13: LINES 64 - 67 ROW 14: LINES 67 - 69 

OS-41 Program Barcode: C2PD, C2DD Program Registers Required: 34 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 4 ROW 3: LINES 4 - 9 ROW 4: LINES 9 - 10 ROW 5: LINES 10 - 17 ROW 6: LINES 17 - 23 ROW 7: LINES 23 - 29 ROW 8: LINES 30 - 36 ROW 9: LINES 37 - 39 ROW 10: LINES 39 - 47 ROW 11: LINES 48 - 55 ROW 12: LINES 56 - 65 ROW 13: LINES 66 - 71 ROW 14: LINES 72 - 79 ROW 15: LINES 80 - 85  C2PD, C2DD (continued)

ROW 16: LINES 86 - 89

## 

ROW 17: LINES 89 - 95

ROW 18: LINES 95 - 97

OS-41 Program Barcode: X2MD Program Registers Required: 12 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 9 ROW 4: LINES 10 - 16 ROW 5: LINES 16 - 23 RON 6: LINES 24 - 29 ROW 7: LINES 30 - 31 

OS-41 Program Barcode: X2XMD, X2MSD Program Registers Required: 43 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 15 ROW 5: LINES 15 - 18 ROW 6: LINES 18 - 25 ROW 7: LINES 25 - 26 ROW 8: LINES 26 - 33 ROW 9: LINES 34 - 38 ROW 10: LINES 38 - 44 ROW 11: LINES 45 - 52 ROW 12: LINES 53 - 60 ROW 13: LINES 61 - 66 ROW 14: LINES 67 - 73 ROW 15: LINES 74 - 81 

X2XMD, X2MSD (continued)

 OS-41 Program Barcode: B2MD

 Program Registers Required: 11

 ROW 1: LINES 1 - 2

 ROW 2: LINES 2 - 6

 ROW 3: LINES 4 - 9

 ROW 4: LINES 10 - 14

 ROW 5: LINES 15 - 20

 ROW 6: LINES 15 - 20
OS-41 Program Barcode: B2XMD, B2MSD Program Registers Required: 44 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 15 ROW 5: LINES 15 - 19 ROW 6: LINES 19 - 25 ROW 7: LINES 25 - 27 ROW 8: LINES 27 - 33 ROW 9: LINES 33 - 40 ROW 10: LINES 40 - 45 ROW 11: LINES 45 - 52 ROW 12: LINES 53 - 59 ROW 13: LINES 59 - 65 ROW 14: LINES 66 - 72 ROW 15: LINES 72 - 78 

B2XMD, B2MSD (continued) ROW 16: LINES 79 - 85 ROW 17: LINES 86 - 92 ROW 18: LINES 93 - 98 ROW 19: LINES 99 - 102 ROW 20: LINES 102 - 107 ROW 21: LINES 108 - 113 ROW 22: LINES 113 - 116 ROW 23: LINES 117 - 119 ROW 24: LINES 119 - 121

OS-41 Program Barcode: B2PD, B2DD Program Registers Required: 20 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 4 ROW 3: LINES 4 - 9 ROW 4: LINES 9 - 10 ROW 5: LINES 10 - 18 ROW 6: LINES 19 - 19 ROW 7: LINES 20 - 27 ROW 8: LINES 27 - 32 ROW 9: LINES 32 - 38 ROW 10: LINES 39 - 45 ROW 11: LINES 46 - 48

OS-41 Program Barcode: M2XMA Program Registers Required: 25 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 11 ROW 4: LINES 12 - 16 ROW 5: LINES 16 - 23 RON 6: LINES 24 - 32 RDW 7: LINES 33 - 40 ROW 8: LINES 41 - 48 ROW 9: LINES 48 - 53 ROW 10: LINES 54 - 61 ROW 11: LINES 61 - 64 ROW 12: LINES 64 - 72 ROW 13: LINES 72 - 74  DS-41 Program Barcode: SCROLL

Program Registers Required: 8

ROW 1: LINES 1 - 4

ROW 2: LINES 5 - 11

ROW 3: LINES 11 - 18

ROW 4: LINES 18 - 26

ROW 4: LINES 18 - 26

ROW 5: LINES 27 - 27

OS-41 Program Barcode: XM2MA Program Registers Required: 19 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 12 ROW 4: LINES 12 - 16 ROW 5: LINES 16 - 18 ROW 6: LINES 18 - 27 ROW 7: LINES 28 - 35 ROW 8: LINES 36 - 43 ROW 9: LINES 44 - 51 ROW 10: LINES 51 - 57 ROW 11: LINES 57 - 57 

OS-41 Program Barcode: XM2XMA 44 Program Registers Required: ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 8 ROW 4: LINES 9 - 11 ROW 5: LINES 11 - 18 ROW 6: LINES 18 - 26 ROW 7: LINES 27 - 29 ROW 8: LINES 30 - 35 ROW 9: LINES 36 - 43 ROW 10: LINES 43 - 51 ROW 11: LINES 52 - 59 ROW 12: LINES 60 - 66 ROW 13: LINES 67 - 75 ROW 14: LINES 76 - 82 ROW 15: LINES 83 - 89 

-435-

XM2XMA (continued) ROW 16: LINES 90 - 96 ROW 17: LINES 97 - 103 ROW 18: LINES 104 - 113 ROW 19: LINES 113 - 120 ROW 20: LINES 120 - 126 ROW 21: LINES 127 - 131 ROW 22: LINES 131 - 135 ROW 23: LINES 136 - 139 RDW 24: LINES 139 - 141

OS-41 Program Barcode: XM2MSA, MS2XMA Program Registers Required: 31 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 13 ROW 5: LINES 14 - 17 ROW 6: LINES 18 - 22 ROW 7: LINES 22 - 28 ROW 8: LINES 28 - 35 ROW 9: LINES 36 - 41 ROW 10: LINES 42 - 48 ROW 11: LINES 49 - 53 ROW 12: LINES 54 - 62 ROW 13: LINES 63 - 67 ROW 14: LINES 67 - 70 ROW 15: LINES 71 - 77 

XM2MSA, MS2XMA (continued)

ROW 16: LINES 77 - 81

## 

ROW 17: LINES 81 - 83

### 

OS-41 Program Barcode: XM2CA Program Registers Required: 27 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 12 ROW 4: LINES 12 - 17 ROW 5: LINES 17 - 19 ROW 6: LINES 20 - 29 ROW 7: LINES 30 - 37 ROW 8: LINES 38 - 45 ROW 9: LINES 45 - 53 ROW 10: LINES 53 - 57 ROW 11: LINES 57 - 61 ROW 12: LINES 62 - 62 ROW 13: LINES 63 - 69 ROW 14: LINES 69 - 73 ROW 15: LINES 73 - 74 

OS-41 Program Barcode: XM2PA, XM2DA Program Registers Required: 21 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 4 ROW 3: LINES 4 - 8 ROW 4: LINES 8 - 12 ROW 5: LINES 13 - 14 ROW 6: LINES 14 - 16 ROW 7: LINES 16 - 24 ROW 8: LINES 25 - 32 ROW 9: LINES 32 - 39 ROW 10: LINES 39 - 46 ROW 11: LINES 46 - 52 ROW 12: LINES 52 - 52 

OS-41 Program Barcode: C2XMA Program Registers Required: 30 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 12 ROW 4: LINES 12 - 17 ROW 5: LINES 17 - 24 ROW 6: LINES 25 - 30 ROW 7: LINES 30 - 34 ROM 8: LINES 35 - 43 ROW 9: LINES 44 - 50 ROW 10: LINES 51 - 58 ROW 11: LINES 58 - 65 ROW 12: LINES 65 - 67 ROW 13: LINES 67 - 75 ROW 14: LINES 75 - 77 ROW 15: LINES 77 - 83 

### C2XMA (continued) ROW 16: LINES 83 - 85

OS-41 Program Barcode: ALSTO Program Registers Required: 17 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 9 ROW 4: LINES 10 - 17 ROW 5: LINES 18 - 24 ROW 6: LINES 24 - 25 ROW 7: LINES 26 - 35 ROW 8: LINES 35 - 38 ROW 9: LINES 38 - 39 

OS-41 Program Barcode: A2XMA Program Registers Required: 29 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 11 ROW 4: LINES 12 - 19 ROW 5: LINES 20 - 23 ROW 6: LINES 23 - 28 ROW 7: LINES 29 - 36 ROW 8: LINES 36 - 42 ROW 9: LINES 43 - 48 ROW 10: LINES 48 - 54 ROW 11: LINES 54 - 61 ROW 12: LINES 61 - 67 ROW 13: LINES 68 - 75 ROW 14: LINES 75 - 81 ROW 15: LINES 82 - 84 

A2XMA (continued) ROW 16: LINES 85 - 85



OS-41 Program Barcode: B2XMA Program Registers Required: 31 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 11 ROW 4: LINES 12 - 19 ROW 5: LINES 20 - 23 ROW 6: LINES 23 - 28 ROW 7: LINES 29 - 36 RON 8: LINES 36 - 42 ROW 9: LINES 43 - 46 ROW 10: LINES 46 - 48 ROW 11: LINES 48 - 53 ROW 12: LINES 53 - 58 ROW 13: LINES 59 - 65 ROW 14: LINES 66 - 72 ROW 15: LINES 72 - 78 

B2XMA (continued)

ROW 16: LINES 78 - 84

#### ROW 17: LINES 84 - 86

OS-41 Program Barcode: M2XME, M2MSE Program Registers Required: 43 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 14 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 22 ROW 7: LINES 23 - 30 ROW 8: LINES 30 - 34 ROW 9: LINES 34 - 40 ROW 10: LINES 41 - 45 ROW 11: LINES 45 - 54 ROW 12: LINES 54 - 61 ROW 13: LINES 62 - 69 ROW 14: LINES 70 - 75 ROW 15: LINES 76 - 82 

M2XME, M2MSE (continued) ROW 16: LINES 83 - 89 ROW 17: LINES 89 - 95 ROW 18: LINES 95 - 102 ROW 19: LINES 103 - 106 ROW 20: LINES 107 - 111 ROW 21: LINES 111 - 115 ROW 22: LINES 115 - 120 ROW 23: LINES 120 - 125 ROW 24: LINES 125 - 125 

#### -450-

OS-41 Program Barcode: XM2ME, MS2ME Program Registers Required: 24 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 8 ROW 4: LINES 9 - 14 ROW 5: LINES 14 - 19 ROW 6: LINES 19 - 21 ROW 7: LINES 22 - 29 ROW 8: LINES 29 - 35 ROW 9: LINES 35 - 42 ROW 10: LINES 42 - 48 ROW 11: LINES 49 - 53 ROW 12: LINES 54 - 59 ROW 13: LINES 59 - 62 

OS-41 Program Barcode: XM2MSE, MS2XME Program Registers Required: 44 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 7 ROW 4: LINES 7 - 14 ROW 5: LINES 14 - 18 ROW 6: LINES 19 - 20 ROW 7: LINES 20 - 27 ROW 8: LINES 27 - 35 ROW 9: LINES 35 - 39 ROW 10: LINES 40 - 48 ROW 11: LINES 49 - 57 ROW 12: LINES 57 - 63 ROW 13: LINES 64 - 68 ROW 14: LINES 69 - 77 ROW 15: LINES 78 - 82 

ROW 17: LINES 88 - 94 ROW 18: LINES 94 - 99 ROW 19: LINES 100 - 105 RDW 20: LINES 105 - 110 ROW 21: LINES 110 - 118 ROW 22: LINES 119 - 123 ROW 23: LINES 124 - 129 ROW 24: LINES 129 - 131 

-453-

XM2MSE, MS2XME (continued)

ROW 16: LINES 83 - 88

OS-41 Program Barcode: MSDUP Program Registers Required: 50 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 9 ROW 4: LINES 9 - 11 ROW 5: LINES 12 - 18 ROW 6: LINES 18 - 25 ROW 7: LINES 26 - 31 ROW 8: LINES 32 - 39 ROW 9: LINES 39 - 40 ROW 10: LINES 41 - 42 ROW 11: LINES 42 - 47 ROW 12: LINES 47 - 49 ROW 13: LINES 49 - 55 ROW 14: LINES 55 - 61 ROW 15: LINES 61 - 64 

ROW 16: LINES 65 - 71 ROW 17: LINES 72 - 78 ROW 18: LINES 79 - 85 ROW 19: LINES 85 - 92 ROW 20: LINES 93 - 97 ROW 21: LINES 98 - 105 ROW 22: LINES 105 - 112 ROW 23: LINES 113 - 122 ROW 24: LINES 122 - 128 ROW 25: LINES 129 - 136 ROW 26: LINES 137 - 143 ROM 27: LINES 143 - 145 

MSDUP (continued)

OS-41 Program Barcode: MSDUPA Program Registers Required: 79 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 7 - 8 ROW 4: LINES 8 - 15 ROW 5: LINES 15 - 17 ROW 6: LINES 17 - 19 ROW 7: LINES 20 - 26 ROW 8: LINES 26 - 34 ROW 9: LINES 34 - 35 ROW 10: LINES 35 - 37 ROW 11: LINES 37 - 42 ROW 12: LINES 42 - 43 ROW 13: LINES 44 - 52 ROW 14: LINES 52 - 54 ROW 15: LINES 54 - 59 

MSDUPA (continued) ROW 16: LINES 59 - 64 ROW 17: LINES 64 - 69 ROW 18: LINES 70 - 73 ROW 19: LINES 74 - 81 ROW 20: LINES 81 - 87 ROW 21: LINES 89 - 96 ROW 22: LINES 97 - 105 ROW 23: LINES 106 - 116 ROW 24: LINES 116 - 123 ROW 25: LINES 124 - 131 ROW 26: LINES 131 - 140 ROW 27: LINES 141 - 149 ROW 28: LINES 149 - 154 ROW 29: LINES 154 - 161 ROW 30: LINES 162 - 168 

MSDUPA (continued) ROW 31: LINES 169 - 175 ROW 32: LINES 176 - 181 ROW 33: LINES 181 - 188 ROW 34: LINES 189 - 194 ROW 35: LINES 195 - 203 ROW 36: LINES 204 - 212 ROW 37: LINES 212 - 219 ROW 38: LINES 220 - 228 ROW 39: LINES 229 - 237 ROW 40: LINES 237 - 245 ROW 41: LINES 246 - 254 ROW 42: LINES 255 - 257 ROW 43: LINES 258 - 258

#### 

OS-41 Program Barcode: MZ, MX Program Registers Required: 18 ROW 1: LINES 1 - 2 ROW 2: LINES 3 - 6 ROW 3: LINES 6 - 9 ROW 4: LINES 9 - 13 ROW 5: LINES 14 - 16 ROW 6: LINES 16 - 24 ROW 7: LINES 24 - 26 ROW 8: LINES 26 - 27 ROW 9: LINES 28 - 35 ROW 10: LINES 35 - 39 

OS-41 Program Barcode: XMZ Program Registers Required: 26 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 7 ROW 3: LINES 7 - 12 ROW 4: LINES 12 - 13 ROW 5: LINES 14 - 16 ROW 6: LINES 17 - 24 ROW 7: LINES 25 - 32 ROW 8: LINES 33 - 34 ROW 9: LINES 35 - 44 ROW 10: LINES 45 - 53 ROW 11: LINES 54 - 63 ROW 12: LINES 63 - 71 ROW 13: LINES 72 - 78 ROW 14: LINES 79 - 83 

OS-41 Program Barcode: XMX, MSX Program Registers Required: 22 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 10 ROW 4: LINES 11 - 14 ROW 5: LINES 14 - 20 ROW 6: LINES 20 - 25 ROW 7: LINES 25 - 30 ROW 8: LINES 31 - 37 ROW 9: LINES 38 - 45 ROW 10: LINES 46 - 53 ROW 11: LINES 54 - 60 ROW 12: LINES 60 - 64 

OS-41 Program Barcode: MSZ Program Registers Required: 34 ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 6 ROW 3: LINES 6 - 11 ROW 4: LINES 11 - 14 ROW 5: LINES 14 - 18 ROW 6: LINES 18 - 25 ROW 7: LINES 25 - 26 ROW 8: LINES 27 - 31 ROW 9: LINES 31 - 39 ROW 10: LINES 40 - 48 ROW 11: LINES 48 - 59 ROW 12: LINES 59 - 71 ROW 13: LINES 71 - 81 ROW 14: LINES 81 - 90 ROW 15: LINES 90 - 99 

MSZ (continued)

ROW 16: LINES 100 - 101

# ROW 17: LINES 101 - 109

ROW 18: LINES 109 - 114

OS-41 Program Barcode: XMCAT, MSCAT 51 Program Registers Required: ROW 1: LINES 1 - 2 ROW 2: LINES 2 - 5 ROW 3: LINES 5 - 10 ROW 4: LINES 10 - 15 ROW 5: LINES 15 - 21 ROW 6: LINES 22 - 25 ROW 7: LINES 26 - 27 ROW 8: LINES 27 - 29 ROW 9: LINES 29 - 32 ROW 10: LINES 32 - 36 ROW 11: LINES 36 - 42 ROW 12: LINES 43 - 48 ROW 13: LINES 48 - 54 ROW 14: LINES 54 - 55 ROW 15: LINES 56 - 61
ROW 16: LINES 61 - 62 ROW 17: LINES 62 - 69 ROW 18: LINES 70 - 74 ROW 19: LINES 74 - 75 ROW 20: LINES 75 - 81 ROW 21: LINES 82 - 88 ROW 22: LINES 89 - 93 ROW 23: LINES 93 - 101 ROW 24: LINES 101 - 106 ROW 25: LINES 107 - 111 ROW 26: LINES 111 - 117 ROW 27: LINES 117 - 120 ROW 28: LINES 121 - 122

XMCAT, MSCAT (continued)

All OS-41 programs (and some others) are listed in quotes and have page number

references preceded by one of the following letters to denote the subject matter as indicated:

- t table of options at beginning of chapter
- i detailed program <u>instructions</u>
- p program listing
- b program <u>b</u>arcode

+К	326, 334
+R	12
+REC	12
Σ <b>REG</b>	328
1K	326, 334

- A -

Α	9,10
A/D	8
"A2XMA"	t215, i248,
	p278, b445
ACX	79
ADV	65, 79
alarm catalog	363 . 365
alarm. all types	340
algorithm. data	
movement	187
ALMCAT	326, 340, 341.
	363 . 365
alphabetical file	
sort	308
alphanumeric data	64
"AI STO"	t215, 221,
	i256. p277.
	h444
APPCHR	215, 221
append method	216
APPREC	215, 221
ARCIREC	215, 219, 220
ASCIT	2137 2137 220
ASCII characters	259
ASCII data processing	214, 216
ASCII file minimum	
Size	220
ASCII file record	~~V
insertion	217

ASCII file size ASCII file. registers	216
used by	388
ASN	326, 333
automatic execution	
flag (11)	266
autostart/duplication	
module	8
- B -	
B	Q
	+65, i184,
2200	p212, b431
"B2MD"	t65, i172,
	p210, b428
"B2MSD"	t65, i178,
	p211, b429
"B2MSP"	t21, i54, p63,
	b403
"B2PD"	t05, 1182,
	$\mu 212, \mu 451$
	n279, $h447$
"B2XMD"	t65. i174.
	p211, b429
"B2XMP"	t21, i50, p63,
	b403
backup copy	304, 308, 373
"BAR"	393
bbb	12
	325, 328
	1Z 25
BF	65.66
BI	65, 76, 325,
	328

INDEX

BM	65,	66
BR	65,	66
bug, PURFL	331	
bug, VER	191	
Β٧	65,	70
byte table	334	

- C -

С	9
c, status register	299, 303
"C2CD"	t65, i148,
	p206, b422
	+65. 1156.
0200	$r_{207}$ $r_{130}$
	$\mu_{207}$ 0425
"CZMD"	1130, 1130,
	p195, D408
"C2MSD"	t65, 1144,
	p203, b417
"C2PD"	t65, i152,
	p207, b423
"C2XMA"	t215, i244,
	p276. b442
"C2XMD"	$t_{65}$ , $i_{140}$ .
	$p_{203}$ , $p_{417}$
calculator status	p203, 041/
	207
after master clear .	527
calculator status	
after MEMORY LOSI	327
card reader	8
CAT	10
CAT 1	363, 364
CAT 2	363, 365
CAT 3	363, 365
CAT 4	363, 365, 366
CAT 5	326. 341. 363.
	365
CAT 6	326, 337
	362
	264
catalog options	504
	12
CDS	12
CF	326,338
CK	326, 335
CLA	260, 325, 329
CLALMA	326, 340
CLALMX	326, 340
CLD	329
CLEI	325, 330, 331
UI KEAS	326, 335
0LNL13	325, 320
	226 240
	520; 540 205 200
	525, 520 205, 200
CLRGX	325, 329
CLST	325, 329

CLST, +	325, 329
CLX	78, 325, 329
<b>CL</b> Σ	325, 328
conditional alarm	340
configuration,	
extended memory	
module	317
control alarm	340
control flag	378, 382
conventions	8
copy, backup	304, 308
copy, duplicate	304, 308
COPYFL	21, 27, 65, 73
CR	8
crash recovery	18, 19
CREATE	67, 219
CRFLAS	220
CRFLD	67
custom module	6
СХ	8

- D -

D	9,10	
data format,	052	
pseudo-ASCII	253	
algorithm	187	
data register usage,		
OS-41	382,3	384
data, alphanumeric	64 64	
decimal equivalents,	04	
ASCII characters	259	
DELREC	325, 3	331
device, primary	285	266
DIR	363, 3	366
display term	11	
display, key		
assignment	337	
DSE	66	
DST	12	
dual extended memory		
	192	
DUPA	10	
duplicate copy	304, 3	308

# - E -

Е	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•		10	
ED		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	215,	220
ee	е		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	12	

END, superfluous, removal ..... 20, 22 ENTER † ..... 78, 329 EP ..... 325, 330 error ignore flag (25) ..... 16, 17 error recovery ..... 15 example, OS-41 program ..... 13 extended functions/ memory module ..... 8 extended i/o module ... 8 extended memory module configuration ..... 317 extended memory module, dual ..... 192 external ROM ..... 17 - F file sort, alphabetical ..... 308 file, ASCII, registers used by ..... 388 file, program ..... 14 file, pseudo-write all ..... 282, 289, 299 file, working ..... 67 file, write all ..... 280 flag 11 (automatic

execution) ..... 266 flag 25 (error ignore) ..... 16, 17 flag 50 (message) ..... 390, 391 flag usage, OS-41 ..... 378, 379 flag, control ..... 378, 382 flag, system ..... 378, 379 flag, user ..... 378, 379 format, "NaMe" file ... 312 format, OS-41 program name ..... 9 format, pseudo-ASCII data ..... 253 format, pseudo-write all file ..... 282, 289, 299 FS? ..... 326, 339 function catalog ..... 363, 365 function key assignment ..... 333

function prefix ..... 8

## - G -

GETAS	215, 216,	220
"GETK"	t326, 336	
GETP	21, 25	
GETR	65, 70	
GETREC	215, 219,	220
GETRX	65, 71	
GETSUB	21, 25	
GETX	65,72	

# – H –

HP-41 function
 catalog ..... 363, 365
HP-IL interface loop
 module ..... 8

## - I -

IF	326,	338	
II	12		
IL	8		
information, status	280		
INSCHR	215,	222	
INSREC	215,	221,	222
INSTAT	378		
interface, video	24		
interrupting control			
alarm	340		
ISG	66		

#### – K –

key assignment		
catalog	363,	365
key assignment,		
display	337	
key assignment,		
function	333	
key assignment,		
program	333	
keycode	333	
keys	11	

## - L -

LASTX	78	
LIST	21,	25
"LOKMO"	266	
"LOKOFF"	266	

– M –

M "M2CD"	9 t65, i90,
"M2DD"	t65, i96, p197, b410
"M2MD"	t65, i84, p193, b405
"M2MP"	t21, i32, p58, b395
"M2MSD"	t65, i88, p194, b406
"M2MSE"	t21, t281, i292, p318,
"M2MSP"	b449 t21, i36, p59, b396
"M2MSPL"	13, 14
"M2PD"	t65, i94,
	p197, b410
"M2XD"	t65, i92,
	p196, b409
"M2XMA"	t215, i224,
	p268, b432
"M2XMD"	t65, i86,
	p194, b406
"M2XME"	t21, t281,
	i288, p318,
	b449
"M2XMP"	b449 t21, i34, p59,
"M2XMP"	b449 t21, i34, p59, b396
"M2XMP"	b449 t21, i34, p59, b396 324, 325
"M2XMP" master clear master clear,	b449 t21, i34, p59, b396 324, 325
"M2XMP" master clear master clear, calculator status	b449 t21, i34, p59, b396 324, 325
"M2XMP" master clear master clear, calculator status after	b449 t21, i34, p59, b396 324, 325 327 281 285
"M2XMP" master clear master clear, calculator status after MCOPY	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285
"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285
"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV medium, source MEMORY LOST	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327
"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV medium, source MEMORY LOST	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340
<pre>"M2XMP" master clear master clear,    calculator status    after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50)</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391
<pre>"M2XMP" master clear master clear,    calculator status    after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50) MK</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334
<pre>"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50) MK MM</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12
<pre>"M2XMP" master clear master clear,    calculator status    after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50) MK MM module catalog</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365
<pre>"M2XMP" master clear master clear,     calculator status     after MCOPY MCOPYPV MCOPYPV MEMORY LOST MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6
<pre>"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV MCOPYPV MEMORY LOST MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom monitor, TV</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24
<pre>"M2XMP" master clear master clear,    calculator status    after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom monitor, TV MRG</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24 21, 28
<pre>"M2XMP" master clear master clear,    calculator status    after MCOPY MCOPYPV MCOPYPV Medium, source MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom monitor, TV MRG MS</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24 21, 28 9, 12, 65, 69
<pre>"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV medium, source MEMORY LOST message alarm message flag (50) MK  MM  module catalog module, custom monitor, TV MRG MS  "MS2CD"</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24 21, 28 9, 12, 65, 69 t65, i128, p203, b417
<pre>"M2XMP" master clear calculator status after MCOPY MCOPYPV medium, source MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom monitor, TV MRG MS "MS2CD"</pre>	b449 t21, i34, p59, b396 324, 325 324, 325 321, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24 21, 28 9, 12, 65, 69 t65, i128, p203, b417 t65, i136,
<pre>"M2XMP" master clear master clear, calculator status after MCOPY MCOPYPV medium, source MEMORY LOST message alarm message flag (50) MK MM module catalog module, custom monitor, TV MRG MS2CD" "MS2DD"</pre>	b449 t21, i34, p59, b396 324, 325 327 281, 285 281, 285 285 327 340 390, 391 326, 334 12 363, 365 6 24 21, 28 9, 12, 65, 69 t65, i128, p203, b417 t65, i136, p205, b420 t65, i118,

"MS2ME"	t21, t281,
	i300, p319,
	b451
"MS2MP"	t21, i44, p60,
	b398
"MS2MSD"	t65, i124,
	p199, b412
"MS2MSP"	t21, i48, p61,
	b399
"MS2PD"	t65, i134,
	p205, b420
"MS2XD"	t65, i132,
	p204, b419
"MS2XMA"	t215, i242,
	p272, b437
"MS2XMD"	t65, il20,
	p201, b450
"MS2XME"	t21, t65,
	t215, t281,
	i302, p320,
	b452
"MS2XMP"	t21, i46, p62,
	b401
"MSCAT"	t363, i372,
	p376, b464
MSCOPY	281, 285
MSDUP mass storage	
device	
compatibility	306
"MSDUP"	t281. i304.
	p321. h454
MSDUPA mass storage	
device	
compatibility	311
	+281. $i308.$
	$n_{322}$ , $h_{156}$
IIMCYII	+325, 1354
MGA	n350 h461
"M\$7"	+3.25, 1352
	n360 h162
нмхн	+225 1246
	13239 13409 n257 6450
1111711	4295 1244
······································	いてつ 1344, 257 b450
	459 , U459

# – N –

name format, OS-41
 program ..... 9
"NaMe" file format .... 312
NEWM ...... 309, 325, 332
no-op ...... 78
noninterrupting
 control alarm ..... 340
null string ..... 253

number, XROM ..... 17, 18 numeric data ..... 64 - 0 optical wand ..... 8 OS-41 ..... 4 - P -P ..... 8, 9, 10 package, program ..... 313 package, program, example ..... 315 PASN ..... 326, 334 PCLPS ..... 325, 330 peripheral catalog .... 363, 365 peripheral printer .... 8 plotter module ..... 8 pointer, program ..... 15, 16 port extender ..... 192 power reset ..... 327 PPC ..... 7, 8 PPC ROM module ..... 8 "PRBC" ..... t21, 25 "PRBCD" ..... 392, 393 "PRBCOS" ..... 393 PRBUF ..... 65, 79 prefix, function ..... 8 prefix, XROM ..... 17, 18 PRFLAGS ..... 326, 339 PRGM ..... 12 primary device ..... 285 PRKEYS ..... 326, 337 program catalog ..... 363, 364, 365 program key assignment ..... 333 program name format, OS-41 ..... 9 program number ..... 31 program package ..... 313 program pointer ..... 15, 16 programming, synthetic ..... 7, 78, 393 prompt term ..... 11 PRP ..... 21, 24 "PRPR" ..... t21, 25, 392 PRREG ..... 65, 69 PRREGX ..... 65, 69 PRSTK ..... 65, 79 "PRSYNT" ..... 393 PRX ..... 65, 79 PRΣ ..... 65, 70

pseudo-ASCII ..... 214, 216

pseudo-ASCII data format ..... 253 pseudo-write all file . 282, 289, 299 PURFL ..... 325, 331 PURFL bug ..... 331 PURGE ..... 325, 332 - R -R ..... 12 "RCEX" ..... t65, 72, i113, pll3, bll3 RCL ..... 65, 68, 69 RCL (ST) ..... 78 RCL IND ..... 68, 69 RCL IND (ST) ..... 68, 69 RCLALM ..... 326, 340, 341 "RCLD" ..... t65, 69, i92, p93, b93 "RCLL" ..... 69 RDN ..... 78 RDTA ..... 65, 74 RDTAX ..... 65, 74 READA ..... 21, 27, 65, 73, 281, 284 **READK** ..... 326, 336 READP ..... 21, 26 READR ..... 65, 73 **READRX** ..... 65, 73 READS ..... 326, 339 **READSUB** ..... 21, 27 REC ..... 12 record insertion ..... 217 record, ASCII file .... 214 recovery, crash ..... 18, 19 recovery, error ..... 15 recovery, trauma ..... 18, 19 register, temporary storage ..... 13 registers used by ASCII file ..... 388 **REGMOVE** ..... 65, 66 **REGSWAP** ..... 65, 66 RENAME ..... 21, 27, 65, 73 reset, power ..... 327 RESZFL ..... 65, 71, 215, 218 RF ..... 326, 338 RK ..... 326, 336 RRR ..... 12 RSUB ..... 21, 28

- S -

SAVEAS	215, 219
"SAVEK"	t326, 335
SAVEP	21, 23
SAVER	65, 67
SAVERXSAVEX	65, 67 65, 77 t215, i262,
SEC SEEKPT SEEKPTA	p269, b433 68 67, 219 67, 219
SEEKRSF	68 326,337 11
size size, ASCII file size, minimum, ASCII	11 216
SK	220
SM	326,335,336
sort. alphabetical	65,76
file	308
source medium	285
SRC	12
<pre>stack lift</pre>	78
standby mode	304, 308
statistical registers .	70
status information	280
status register c	209, 303
status, calculator,	327
status, calculator,	327
after MEMORY LOST	±65, 77, i166.
STO	pl66, bl13 65, 76, 325, 328
STO (ST)	78
STO (ST) X	65,78
STO IND	76
STO IND (ST)	76 t65, 76, i161, p162, b93
"STOL"	69,70
subroutine, OS-41	390
removal support program, OS-41	20,22 392

synthetic programming . 7, 78, 393
system flag ..... 378, 379 - T -

T "TE" temporary storage register time module TMP tone alarm trauma recovery TV monitor - U -	8 t215, 222 13 8 12 340 18, 19 24
UNSECuser flag	332 378, 379
VER bug VF bug VF video interface VIEW VIEW (ST) VIEW (ST) X VIEW IND VIEW IND VIEW IND (ST) VK	68 191 326, 339 24 65, 69 79 65, 79 69 69 326, 337
- W -	
W WALL WDTA WDTAX	8 21, 24, 65, 68, 281, 283 65, 68 65, 68, 325, 333
WNDDTA WNDDTX WNDLNK WNDSUB working file	65, 80 65, 79 21, 29 21, 29 67 280
WRPV WRTA WRTK WRTP WRTPV WRTR	21, 24 21, 24, 65, 68, 281, 283 326, 335 21, 23 21, 23 65, 67
WRTRX	65,68

WRTS		326,	338
WSTS	•••••	326,	336

- X -

X	9,10
"X 2MD"	t65, il60,
	p208, b425
"X 2MSD"	t65, i168,
	p209, b426
"X 2XMD"	t65. i164.
	p209, b426
$X \leftrightarrow (ST)$	78
X<>F	326, 337
Y<>Y	65.78
X × × × × × × × × × × × × × × × × × × ×	78
VE	70 8
	0
	0 12
	9,12
"XMZAA"	t215, 1236,
	p2/4, b440
"XM2CA"	t215, i232,
	p273, b439
"XM2CD"	t65, i108,
	p203, b417
"XM2DA"	t215, i240,
	p275, b441
"XM2DD"	t65, ill6,
	p205, b420
"XM2MA"	t215, i226,
	p270, b434
"XM2MD"	t65. 198.
	p198, b411
"X M2MF "	+21, +281,
	i296, $n319$ .
	h451
"XM2MP"	+21, i38, p60,
	h308
ͲϒΜΌΜϚΔΙΙ	+215 +220
	12139 $12309$
ихмомери	μζ/ζο υ45/
"XMZM3D"	105; 1104; 201 b415
	p201, 0415
"XM2MSE"	τ21, το5,
	t215, t281,
	1298, p320,
	b452
"XM2MSP"	t21, i42, p62,
	b401
"XM2PA"	t215, i238,
	p275, b441
"XM2PD"	t65, ill4,
	p205, b420
"XM2PS"	391
"XM2XD"	t65, ill2,
	p204, b419

"XM2XMA"	t215, i228,
	p271, b435
"XM2XMD"	t65, i100,
	p199, b412
"XM2XMP"	t21, i40, p61,
	b399
"XMCAT"	t363, i368,
	p376, b464
XMS	12
"XMX"	t325, i350,
	p359, b461
"XMZ"	t325, i348,
	p358, b460
XROM prefix number	17, 18
хтоа	259
XYZALM	326, 340, 341

# - Z -

Ζ	• • •	•	••	• •		•	•	•	•	•	•	•	•	•	•	•	•	10	
ZE	RO	•	••	• •	••	•	•	•	•	•	•	•	•	•	•	•	•	325,	332

#### ORDER FORM FOR ADDITIONAL COPIES

Send form to: THOMAS W. BEERS 1808 Summit Drive West Lafayete, IN 47906

DATE:

Please send \_\_\_\_\_ copy(ies) of OS-41/An Operating System for the HP-41 Handheld Computer and Peripherals by Thomas W. and Ted W. Beers to:

	(Name)
	(Street)
	(City)
(State)	(Zip)
Enclosed copy(i	.es)@\$42.00 \$
plus \$3.0	)0 each copy \$
for sh	ipping
and ha	undling
TOTAL	AMOUNT \$

Please make checks payable to THOMAS W. BEERS. Indiana residents add 5% (\$2.10) sales tax per copy. Overseas orders should double the shipping charge. ALL FOREIGN ORDERS MUST BE PREPAID IN U.S. DOLLARS AND BE DRAWN ON A UNITED STATES BANK.

1-2210	05-41·J=02105-41·J=0	2105-41-1-22105-41-1-22105-41-1-22105-41-1-22105-41-1-22105-41-1-2
) <b>5-4</b> [·	·I=12105-41-1-12105-	4[·]=f2' 05-4[·]=f2' 05-4[·]=f2' 05-4[·]=f2' 05-4[·]=f2' 05-4[·]=f2' 05-
	05-41·1-9:105-41·1-9	2105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=12105-41+1=1
		"\\_#\\05-4\\\_#\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\\_#\\05-4\\
		105-4]./FIL105-4[/FIL105-4[/FIL105-4]./FIL105-4[/FIL105-4[/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4]/FIL105-4[/FIL105-4]/FIL105-4]/FIL105-4]/FIL105-4[/FIL105-4]/FIL1000000000000000000000000000000000000
		['FIL'(US-4!'FIL'(US-4!'FIL'(US-4!'FIL'(US-4!'FIL'(US-4!'FIL'(US-4!'FIL')US-4!'FIL'(US-4!'FIL')US-4!'FIL'(US-4!'FIL')US-4!'FIL'(US-4!'FIL')US-4!'FIL'(US-4!'FIL')US-4!'FIL')US-4!'FIL'(US-4!'FIL')US-5'
		105-41.F1L105-41.F
	INTRODUCTION	['F11'  03"1'F11'05"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"1'F11'03"
		103-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.1-54.05-41.0-54.05-
		inc_41.1-84.0C_41.1-84.0C_41.1-84.0S_41.0S_41
		103-41-20103-41-20103-41-20103-41-20103-41-20103-41-20 11-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-1-84105-01-
		ins.41.LT/ins.41.LT/ins.41.LT/ins.41.LT/ins.41.LT/ins.41.LT/i
		165-41/PE105-41/PE105-41/PE105-41/PE105-41/PE105-41/PE
	PROGRAM	ins.411274105-41127405-41127405-41127405-41127405-41127
	MONEMENT	105-41-541-541-541-541-541-541-541-541-541
	MUVEMENI	i05-41/L-C/05-41/L-C/05-41/L-C/05-41/L-C/05-41/L-C/05-41/L-C
		1.LT'IOS-41.LT'IOS-41.LT'IOS-41.LT'IOS-41.LT'IOS-41.LT'IOS-
		105.41 LT 105.41 LT 105.41 LT 105.41 LT 105.41 LT 105.41 LT
		1.LT4 05-41 LT405-41 LT405-41 LT4 05-41 LT405-41 LT405-41 LT405-
	ΔΤΔ	ins.411LT1105.411LT1105.411LT1105.411LT1105.411LT1105.411LT
	MOVEMENT	[LT":05-41.LT":05-41.LT":05-41.LT":05-41.LT":05-41.LT":05-41.LT
	MOVEMENI	05.41154105.41154105.41154105.41154105.41154105.41154105.41154
		[LT' 05-41 LT' 05-41 LT' 05-41 LT' 05-41 LT' 05-41 LT' 05-
		05-41-1-0105-41-1-0105-41-1-0105-41-1-0105-41-1-0105-41-1-0
		[LT] 05-41 LT'105-41 LT'105-41 LT'105-41 LT'105-41 LT'105-41 LT'105-
	11 <b>0</b> 20	05-411-57105-411-57105-411-57105-411-57105-411-57105-411-57
	ASCII	[1-0105-41-1-0105-41-1-0105-41-1-0105-41-1-0105-41-1-0105-4
	MOVEMENT	05-41-1-04105-41-1-04105-41-1-04105-41-1-04105-41-1-04105-41-1-0
		[1-1-11]05-41-1-11:05-41-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-1-11:05-41-11:05-41-1-11:05-41-1-11:05-41-11:05-11-11-11-11:05-11-1
		05-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2
		[1-1-11] 05-41-1-12:105-41-12:105-41-12:105-41-12:105-41-12:105-
	ENTIDE MEDIA	05-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2
	LINTINE MEDIA	[·]-11105-41-J-12105-41-J-12105-41-J-12105-41-J-12105-41-J-12105-
		05-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2105-41-1-2
		[·]-11105-41-J-1105-41-J-1105-41-J-1105-41-J-1105-4105-41-J-1105-41-J-1105-4105-4105-4105-4105-4105-4105-4105
		05-41:1-2105-41-1-2105-41-1-2105-41:1-2105-41-1-2105-41-1-2
		[+]=#1 05-41-]=#1 05-1
		05-41:1-21105-41:1-21105-41:1-21105-41:1-21105-41:1-21105-41:1-2
	CLEAR, FUNDE	[·]=f['i 05-41·]=f['i05-41·]=f
		05-41:1=2105-41:1=2105-41:1=2105-41:1=2105-41:1=2105-41:1=2
		[+]=12'i 05-41:J=12'i 05-41:J=12'i 05-41:J=12'i 05-41:J=12'i 05-41:J=12'i 05-41:J=12'i 05-41:J=12'i 05-41';J=12'i 05-41';J=12';J=11';J=12'
		05-41:1=21105-41:1=21105-41:1=22105-4100
		[+]=f2'  05-41+]=f2'  05-41+]=f
	CATALOCUINC	05-41:1-21105-41:1-21105-41:1-21105-41:1-21105-41:1-21105-41:1-2
	CATALUGUING	[+]=f2'i 05-41-1=f2'i 05-41:1=f2'i 05-41-1=f2'i 05-41-1=f2'i 05-41-1=f2'i 05-4
		05-41-1-21105-41-1-21105-41-1-22105-41-1-22105-41-1-22105-41-1-2
		[!]=12'i 05-41'J=12'i 05'J=12'II'J=12'I 05'J=12'II'II'J=12'II'II'J=12'II'II'II'J=12'II'II'I'I'I'I'I'I'I'I'I'I'I'I'I'I'I'I
		05-41:1-2105-41:1-2105-41:1-2105-41:1-2105-41:1-2105-41:1-21
		·FE2105-41·FE2105-41·FE2105-41·FE2105-41·FE2105-41·FE2105-4
		05-41:FE(105-41:FE(105-41:FE(105-41:FE(105-41:FE(105-41:FE
	APPENDICES	·注意:105-41:
		05-41:1=02105-41:1=02105-41:1=02105-41:1=02105-41:1=02105-41:1=02
		·]=12105-41·]=12105-41·]=12105-41·]=12105-41·]=12105-41·]=12105-4
		05-41:1=12105-41:1=12105-41:1=12105-41:1=12105-41:1=12105-41:1=12
		·]=12105-41·]=12105-41·]=12105-41·]=12105-41·]=12105-41·]=12105-4
		05-41·1=22105-41·1=22105-41·1=22105-41·1=22105-41·1=22105-41·1=22
	INDEX	·FE:05-41·FE:05-41·FE:05-41·FE:05-41·FE:05-41·FE:05-41·FE:05-4
		US-41.1FE(105-41.1FE(105-41.1FE(105-41.1FE(105-41.1FE))
		······································
-41.L	-T'IDS-41-1-T'IOC-4	₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩₩
<b>T</b> 'i04	5-41-1-21103-41-1-21	175-41/J-7105-41/J-74 06-41/J-74 06-41/J-74 06-41/J-74 06-41/J-74
-41.1	-E'105-41'-1-8''105-4	1:57405-41:57405-41:67406-41:57406-41:57406-41:57406-41:57406-41:57406-41:57406-41:57406-41:57406-41:57406-41:5
1109	5-41-1-8"1 05-41-1-8"	105-41·J-6/105-