

CMT-71-P01 EPROM Programmer

Instruction Manual

4/22/86

Corvallis Microtechnology, Inc. has made every attempt to ensure that the information in this document is accurate and complete. However, CMT assumes no liability for errors, or for any damages that result from use of this document or the equipment which it accompanies.

Corvallis Microtechnology, Inc. reserves the right to make changes to this document without notice at any time.

Copyright 1986, Corvallis Microtechnology, Inc.
All rights reserved



CORVALLIS MICROTECHNOLOGY, INC.

Grant Plaza
895 NW Grant Ave.
CORVALLIS, OREGON 97330
Phone: (503) 752-5456

SAFETY SUMMARY

General safety information for operating personnel is contained in this summary. In addition, specific WARNINGS and CAUTIONS appear throughout the manual where they apply and are not included in this summary.

Definitions:

WARNING statements identify conditions or practices that could result in damage to equipment or property.

CAUTION statements identify conditions or practices that could result in unpredictable behavior or a "memory lost" condition on the HP-71.

Electrostatic Discharge:

The programmer, HP-71, and memory modules are susceptible to large values of electrostatic discharge (ESD). Always follow proper ESD procedures when using the programmer. General handling procedures are listed below:

GENERAL HANDLING RECOMMENDATIONS FOR CMOS DEVICES

	Should be conductive	Should be grounded to common point
Handling Equipment	x	
Metal parts of fixtures and tools		x
Handling trays	x	
Soldering irons		x
Table tops	x	x
Transport Carts	x	
Operating personnel and general handling of devices		Use grounded, metal or conductive plas- tic wrist straps with 1MOhm series resistor.

Power Source:

Verify that the programmer AC adapter is correct for the available line voltage. The adapter shipped from CMT is for 120V - 60Hz operation.

Servicing:

To reduce the risk of electrical shock, do not perform any servicing other than that described in this manual.

MANUAL CONTENTS

A brief description of this manual is provided below.

INTRODUCTION

This section provides a general description of the programmer, its specifications, warranty and service information, and ordering information.

CMT-71-P01 OPERATIONS

This section provides detailed instructions on how to set up the CMT-71-P01 programmer for operation and how to program devices using the front panel switches.

ENTERING AND EXITING PROGRAM MODE

This section describes the interaction between the HP-71 and the CMT-71 P01.

SOFTWARE INTERACTION

This section provides detailed descriptions of the commands used to build ROM image files using the HP-71.

Introduction

The CMT 71-P01 EPROM Programmer is a completely self-contained device that allows you to program CMT-71-32KE EPROMs or CMT-71-64KE EPROMs. There is no requirement for special socket adapters or personality modules. The programmer contains 64K bytes of CMOS RAM that is used to store the image that will be programmed into the EPROM modules.

The programmer connects to the HP-71 through a front port. The programmer's RAM is accessible through the commands in the HP-71. To use the programmer in the 64K mode, a FORTH/ASSM ROM needs to be inserted into another HP-71 port.

Additional software to help build the EPROM image is supplied on the CMT-71 EPROMs along with self-test programs. The EPROMs will need to be erased with ultraviolet light before programming. It is therefore highly recommended that you store these programs and functions on another mass storage medium so they will be available for use when needed.

This manual contains the instructions necessary for operating the CMT-71-P01 programmer.

Generally, you will not have to understand the internal operation of the HP-71 operating system to create custom EPROMs. If you would like to know more about the HP-71 operating system and its file structures, consult the HP-71 software IDS published by HP.

This manual assumes you have some working knowledge of the general use of the HP-71 and the hexadecimal number system. Most references to addresses or data will be made in hex.

The programmer can be inserted into ports 2, 3, or 4. The programmer will not function properly in Port (1) owing to different system control lines available. For simplicity, this manual assumes that the programmer is plugged into Port(4).

An intelligent programming algorithm is used to program both the 32K and 64K EPROMs. Typically the length of time to burn a 32K EPROM is less than 5 minutes. The 64K EPROM will require generally less than 15 minutes.

If you require further assistance in using your CMT-71-P01, contact our office at 33815 Eastgate Circle, Corvallis, OR 97333. Tel: (503)752-5456

Specifications

Specifications for the CMT-71-P01 are listed below.

Functional Specifications:

Data RAM: 64K bytes (Usable as System or FREEPORTed RAM)

32K mode: 2 blocks of 32K bytes

First block contains EPROM image

64K mode: 1 block of 64K bytes

Requires Forth ROM inserted to

FREEPORT correctly.

ROM Image Format: Standard HP-71B file chain with the
optional addition of a checksum byte

Device Connection: Extender board from HP-71B port

Power Requirements:

Operating Voltage into DC input: 9-12 VDC (tip +)

Power Consumption: < 250 mA

Physical and Environmental Specifications:

Dimensions: 5.75" x 3.6" x 1.2"

Weight: less than 1 pound

Operating Temperature: 25 C +/- 5 C (EPROM programming spec)

Warranty and Service

This CMT product is warranted against defects in material and workmanship for a period of 90 days from the date of purchase. During this period, Corvallis Microtechnology, Inc. will, at its option, either repair or replace products which prove to be defective.

Buyer shall prepay shipping charges to return the product to CMT for warranty service, repair or replacement. Buyer shall pay all shipping charges, duties, and taxes for products returned from any country outside the U.S.A.

CMT does not warrant that the operation of this device and the firmware and software installed in it will be error-free.

Limitation of Warranty

The foregoing warranty shall not apply to defects or malfunctions resulting from improper or inadequate maintenance by buyer, buyer-supplied interfacing, unauthorized modification, misuse, or operation outside the environmental specifications for the product.

No other warranty is expressed or implied. Corvallis Microtechnology specifically disclaims the implied warranties of fitness for a particular purpose.

Service Information

The CMT-71-P01 is shipped from the factory completely calibrated, and should not require calibration during normal use. If your programmer is not operating properly, read the shipping instructions below. Be sure to read this manual completely before shipping the product back to CMT. Many times, errors are caused by misunderstanding.

Shipping Instructions

Do not ship any product back to CMT without first contacting us about your problem. We do not accept any unauthorized returns.

After obtaining a return authorization number, ship the product in a protective package to avoid damage. (Use the original shipping container and cushioning material if available.) In-transit damage is not covered by the warranty.

When returning the unit for service, repair or replacement, be sure to include your name and address. Also enclose a description of the problem and the system setup when the problem occurred.

For warranty service or repair, return a copy of the sales receipt with the unit.

Ordering

You may order CMT products directly from CMT or from any CMT dealer. When ordering from CMT, please include the following:

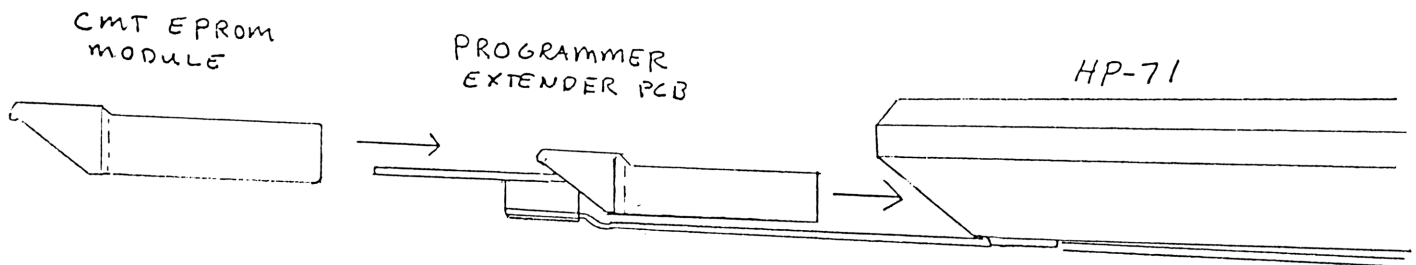
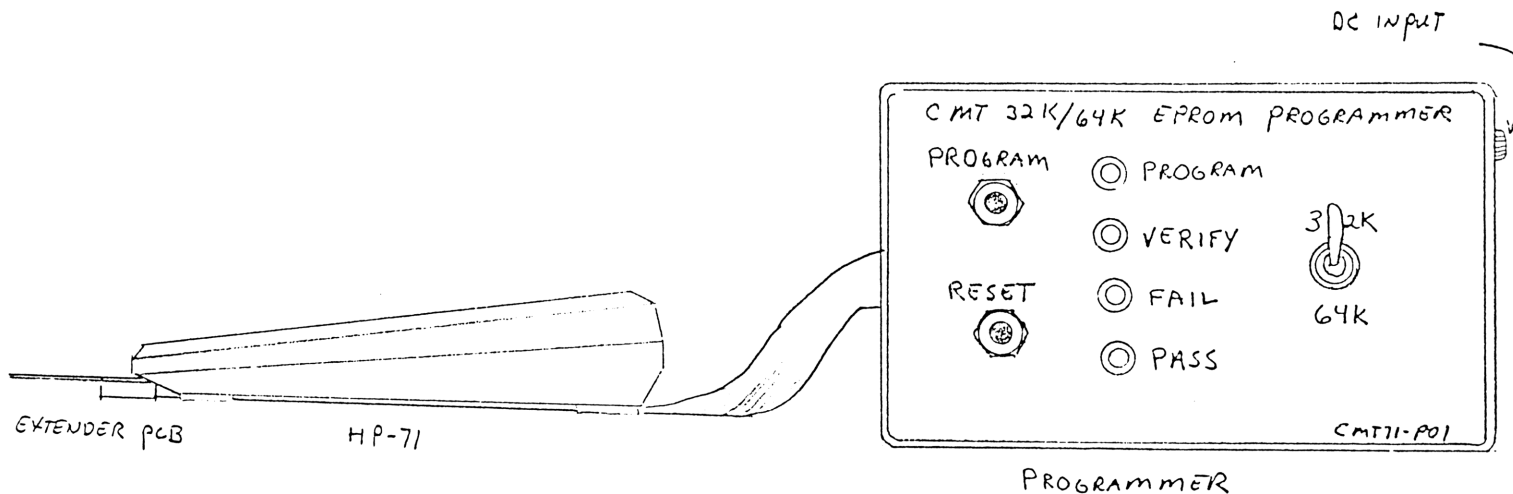
- * Description of the equipment.
- * Quantity of each item ordered.
- * Individual - Your name and address, plus a check in U.S. funds and drawn on a U.S. bank.
Approved corporations - Signed purchase order.
- * Desired method of shipment.

CMT-71-P01 Operations

The CMT-71-P01 allows you to build in the RAM of the programmer an EPROM image of programs residing in the HP-71. (See Downloading Data below.) This image is then burned (programmed) onto a CMT-71-32KE or CMT-71-64KE EPROM module. You could also make copies of a programmed EPROM by first moving its contents into the programmer's RAM. This is described in Copying EPROMs below.)

Many of the commands used have been supplied with your CMT-71 EPROM module. Be sure to copy these to another mass storage medium so you have them handy. A list of these programs and functions is provided in the next section, Software Interaction.

As shown in the figure below, the front panel of the CMT-71-P01 features 2 buttons (PROGRAM and RESET), 4 LED indicators and a switch for selecting 32K or 64K operation. The PROGRAM and RESET buttons allow you to transfer control between the HP-71 and the CMT-71-P01 Programmer.



The ribbon cable of the programmer has been designed to go under the HP-71, so that the EPROM module can be conveniently connected to the extender PCB. Do not connect the programmer to the HP-71 yet. Read Connecting Power (given later in this section) before you do so.

The four LEDs are used to indicate the status of the programmer. These are labeled:

- * PROGRAM - Programmer in Program mode
- * VERIFY - Programmer checking EPROM contents
- * FAIL - Program or Verify operation failed
- * PASS - EPROM programmed correctly

If all LEDs are off, then the programmer is in a reset state. This state returns control of the RAM to the HP-71. The HP-71 also will have control of the HP-71 system bus. When any of the LEDs are lit, the programmer has control of the HP-71 bus and the RAM. The HP-71 will not operate while the programmer has control. Control can be returned to the HP-71 by pressing the RESET button on the programmer.

The PROGRAM LED will blink when the programmer is programming an EPROM. This blink occurs every 512 bytes. If a device fails the LED may be lit or not as determined by the address of the failure.

The PROGRAM LED will blink at different rates for different EPROMs. This is normal as some EPROMs require more pulses than others.

The VERIFY LED will be lit when the programmer has finished programming the EPROM and is verifying that the contents of the EPROM are the same as the RAM.

The FAIL LED will be lit whenever an error is detected during the programming or verifying process. The FAIL LED means that the EPROM has not been programmed as intended and should be removed from the programmer.

The PASS LED will be lit when an EPROM has been programmed and verified successfully.

Loading the CMT-71 Software

You will need to use some of the programs and functions supplied with your CMT-71 EPROM module when you program the module. Before proceeding with the following, copy these programs and functions into the HP-71 RAM by means of the HP-71 COPY command.

Connecting Power

Before connecting power to the programmer, perform the following checks:

- * Verify that the line voltage is appropriate for the wall transformer being used.
- * Verify that the programmer extender PCB and cable are in good condition and are not damaged. The programmer should not be connected to the HP-71.

Powering Up the Programmer

To power up the programmer, proceed as follows:

1. Make sure that the programmer is not connected to the HP-71.
2. Insert the wall transformer DC plug into the jack on the programmer. Plug the wall transformer into the AC outlet.
3. Press the RESET button on the programmer. Verify that there is no module on the programmer PCB.
4. Press the PROGRAM button on the programmer. The PROGRAM and FAIL LEDs should be lit. This ensures that the programmer is functional.
5. Press the RESET button on the programmer.
6. Remove the A.C. recharger from the HP-71 (if one is used). The programmer will supply power to the HP-71 while connected. The batteries of the HP-71 can be left in place without harm.

WARNING: Not removing the AC recharger from the HP-71 could cause the EPROM to be programmed incorrectly.

7. Turn off the HP-71.
8. Remove the module cap from the right-most port (Port (4)) and save it. Insert the programmer's connector into Port(4).
9. Select 32K or 64K, as appropriate, with the switch on the programmer.
10. Turn on the HP-71.

Downloading Data

To load data into the programmer RAM, do as follows:

1. Turn off the HP-71 computer.
2. Press RESET on the programmer.
3. Insert the programmer's connector into the HP-71, Port (4).
4. Select 32K or 64K mode, as appropriate.
5. Turn on the HP-71.
6. FREEPORT the programmer RAM with the following command:

FREEPORT(4)

7. Copy your programs to the programmer RAM using the COPY command.

Example: COPY 'MYFILE' to 'FILENAME:PORT(4)'

You may also wish to copy some of the CMT-71 software functions into the programmer RAM so they will be programmed into the EPROM along with your software.

8. If you want to protect or private a file, execute the appropriate command.

Examples: PRIVATE 'FILENAME:PORT(4)'
 SECURE 'FILENAME:PORT(4)'

See the HP-71 manual for an explanation of what the PRIVATE and SECURE keywords do.

9. Continue steps 7 and 8 until all files are loaded into the programmer. If there is not enough room available the COPY command will return 'ERR: not enough memory' and the file will not be transferred. To determine the amount of memory available in the RAM, execute MEM(4).
10. After all programs have been loaded into the RAM, execute the function PTGENC\$(':port(4)') to generate the proper checksum for the EPROM. If there is not enough room in the port to generate the checksum then the function will display 'ERR: not enough mem'. The function requires the use of the last 2 bytes in the port to generate the Checksum. PTGENC\$ will chain all files that need to be chained.
11. Turn off the HP-71, connect a blank CMT EPROM module to the programmer PCB.

CAUTION: Before connecting the EPROM to the programmer PCB, turn off the HP-71. This ensures that the device will be stored in the configuration table of the HP-71 when you turn on the HP-71.

Checking for Blank Devices

To determine if a device (module) is blank, connect the device to the programmer PCB. Turn on the HP-71 and execute:

```
PTBLANK$(':port(4.02)') if in 32K mode or
PTBLANK$(':port(4.01)') if in 64K mode
```

If the device is blank, the string "00000" will be returned. If the device is not blank, then the first non-blank address will be returned as a hex (hexadecimal) string. This address corresponds to the actual HP-71 address map and not the absolute address of the device. Turn off the HP-71 before programming.

An EPROM is blank if it has been erased and all locations contain 'FF' hex. An EPROM module can be erased by exposing the window of the EPROM to a high intensity ultraviolet (UV) light source. (Remove the label covering the EPROM window.) A dosage of 15W-second/cm² is required to completely erase a CMT module. The UV lamp should have a wavelength of 2537 Angstroms, with an intensity of 12000 uW/cm². Recommended exposure time is 15-20 minutes. The module should be about one inch from the source, and all filters should be removed from the UV light source.

Copying EPROMs

To make copies of EPROMs, first load the EPROM image into the programmer's RAM as follows. Insert the master EPROM into a port other than the one the programmer is connected to. (Port 5 OK for HHP module). Make sure that the CLONE program provided with the EPROM module is loaded into the HP-71. Type RUN 'CLONE'. The program will prompt for the source port. This is the port you will copy from. Just type the port number (i.e. 1 or 5 or 4.02). The program will then prompt for the destination port. This is where the source port will be copied to. Just type the port number. You can copy from a smaller memory to a larger one; the program will issue a warning as a caution. You can not copy from a larger memory to a smaller one (i.e. 64K bytes from EPROM to the programmer RAM when the programmer is set for 32K operation). **A FORTH module is required for the CLONE program to operate.** The program will display "DONE..." when it is finished.

CAUTION: Before running the CLONE program, clear the programmer RAM by executing CLAIMPORT(4) (if already FREEPORTed) and then FREEPORT(4). Failure to do so may cause the HP-71 operating system to crash. This crash is due to the operating system getting confused with the old and new LEX files.

After loading the EPROM image into the programmer's RAM, program a blank EPROM as described below.

Programming a Device

To program a device, connect the device to the programmer PCB. Press the RESET button on the programmer and then press the PROGRAM button on the programmer.

If the device is connected incorrectly or if there is a programming error, the programmer FAIL light will turn on. If the programmer is programming the device correctly, the program LED will blink on and off every 512 bytes. This assures you that the programmer is proceeding correctly. When the programmer has programmed the entire EPROM, the programmer will read the entire EPROM and compare its contents to the programmer RAM. If there is an error the FAIL LED will be lit. During the verifying operation the VERIFY LED will light. If the device is programmed properly, the VERIFY and PASS LEDs will be lit.

You must press RESET in order to turn on your HP-71.

WARNING: If a device fails during programming, remove it from the programmer PCB before turning on the HP-71. Failure to do so could cause the HP-71 to behave unpredictably.

Aborting an Operation

To abort a programming operation press the RESET button on the programmer. Remove the EPROM from the programmer PCB before turning the HP-71 on.

Verifying a Programmed Device

To verify that a device has been programmed correctly after the programmer has finished programming it, execute the keyword `PTTEST$(':PORT(P.DD)').` If the device returns a proper checksum then the function will return the string '00000'. If the device has an improper checksum, the function will return a string consisting of the hex value of the checksum read minus one.

Disconnecting the Programmer

Before disconnecting the programmer from the HP-71, FREEPORT the programmer RAM as follows:

In the 64K mode, `FREEPORT(4).`

In the 32K mode, `FREEPORT(4)` and `FREEPORT(4.01).`

CAUTION: Failure to FREEPORT the programmer RAM as above may cause a 'memory lost' condition.

Entering and Exiting Program Mode

This section explains the interaction between the HP-71 and the CMT71-P01 programmer.

When the programmer is inserted into the HP-71 and the RESET button pushed, the HP-71 has control of the system bus. The RAM memory in the programmer is available to the HP-71 for use as would any other RAM inserted into the port. If the programmer is in 32K mode then there are 2 separate 32K byte RAM blocks that the HP-71 operating system can use. Each block can be FREEPORTed individually or used as system RAM. Only the first block in the chain is used to copy its contents into the EPROM. If the programmer is inserted into Port(4) then the first device is (4.00). The second device is (4.01). If the EPROM is inserted, it is (4.02).

If 64K mode is selected, the RAM in the programmer is configured as a single 64K block. The programmer's RAM can only be FREEPORTed as one block. The 64K RAM block is device (4.00) and the EPROM, if inserted is device (4.01).

CAUTION: When in the 64K mode, always have a FORTH ROM inserted into another HP-71 port. This prevents a bug in the HP-71 operating system from configuring the 64K RAM block incorrectly. The FORTH ROM must be inserted to FREEPORT the RAM correctly.

After the FREEPORTed RAM has been loaded with the appropriate files, you may want to make a back-up of your image onto a HPIL mass storage device. Use the EPRMCPY command to do this.

To connect a device to the programmer PCB, carefully insert the PCB's edge into the EPROM module. There should be little force required for this action. The EPROM module should be in the same orientation as it would be if inserted directly into the HP-71.

CAUTION: Turn off the HP-71 before connecting the EPROM module to the programmer PCB. Also turn off the HP-71 before inserting the programmer into the port. All programming operations should be done while the HP-71 is in the off state. This procedure prevents unpredictable behavior when control is returned to the HP-71. When the programmer has finished programming the EPROM, pass or fail, the programmer RAM and the EPROM will be unconfigured from the HP-71 memory map. When the HP-71 is turned on, it will reconfigure these devices into its memory address map.

Selecting a Device Type

The device type to program is selected by setting the 32K/64K switch on the front panel of the programmer.

WARNING: Never connect any other module to the programmer PCB. When the programmer takes control, the programming voltages will be present at the port. These voltages may damage and/or destroy any modules inserted that are not the correct type.

CAUTION: Changing the programmer's 32K/64K switch from 32K to 64K may cause a 'memory lost' condition. To prevent this from occurring, FREEPORT 4.01 as well as 4 when building EPROM images.

Editing Memory

In most cases, the memory in the programmer will not need to be examined or modified. In those cases where this is required, the memory can be examined and modified with the functions provided by the FORTH/ASSEMBLER ROM. Please refer to its manual and the HP-71 software IDS for more information and instructions.

Estimating EPROM Size

An approximation to the amount of memory required for an EPROM module is done by multiplying the number of files by 25 bytes (for the file headers), adding the data lengths (from CAT) of all the files, and adding an additional 11 bytes for EPROM module overhead.

Generating a Checksum for the Data

Actually, there is no requirement that a checksum exist in the ROM. The checksum technique from HP was devised to allow the user to verify quickly that the ROM image is correct. The actual selection that the sum add up to "1" is arbitrary for this method. The sum of "1" was picked to be a standard for use by all parties. It is good practice to use this checksum when generating EPROMs so that users of your software product can be ensured that all is functional. The checksum is generated with the PTGENC\$ function.

A LEX file is included on each EPROM shipped from CMT that can be loaded into your ROM image if you desire and if there is room available. TEST32KE or TEST64KE executes by typing ROMTST and pressing RETURN. ROMTST returns ROM OK if the ROM is OK or ROM BAD if the checksum does not add up to "1". If more than one ROM is inserted into the HP-71 with this file loaded, only the first ROM searched with this function will be tested.

Chaining Files

For the most part, file chains in the HP-71 operating system are the same for FREEPORTed RAM and ROM. The difference is that the RAM BASIC files can be edited. This means that after a file has been loaded into the RAM it may be edited and the line number references changed. Chaining is a process which ensures that the GOTOs and GOSUBs are jumping to the proper locations. Chaining is accomplished by any of the following methods:

- * Execute PTGENC\$ command.
- * Do not alter the file after it has been copied into the FREEPORTed RAM.
- * Run the file in the FREEPORTed RAM.
- * Transform the file into TEXT and back to BASIC.
- * Renumber the current file by executing RENUMBER 1,1,1,1.

Refer to the HP-71 Software IDS for more information on chaining files.

CAUTION: Running a BASIC file that is not chained properly will invoke a system error.

Software Interaction

Software Commands

For most applications, the software functions required to program EPROMs are included in the HP-71 mainframe code. Some useful commands are:

- * COPY filename TO :port() - copies files to port.
- * PRIVATE filename:port() - restricts file access.
- * SECURE filename:port() - prevents file purge or alteration.
- * CAT:PORT() - catalogs files in port.
- * MEM() - shows available memory left in port.
- * SHOWPORT - shows size and type of memory in port.
- * FREEPORT() - converts port to Iindependent RAM.
- * CLAIMPORT() - converts Iindependent RAM to Main RAM.
- * ADDR\$() - shows hex address of start of file.
- * PEEK\$('hexaddr',#nibs) - displays memory contents.
- * POKE 'hexaddr','hexdata' - modifies memory contents.
- * ENTIRE FORTH Library

For more detailed information on the above commands, please refer to your HP-71 reference manual.

Some helpful functions used to program the EPROM are stored on each EPROM when it is shipped from CMT's factory. These functions allow you to determine if an EPROM is blank, determine if an EPROM has a proper checksum, generate a proper checksum, determine the size of an EPROM, and determine the starting address of the EPROM in the HP-71 address map.

Also included is a function which allows you to store the EPROM image on an HPIL mass storage device. The same function also allows you to retrieve the image from the HPIL mass storage medium.

These functions may be copied from the EPROM and stored on another mass storage medium for use later when the EPROM has been erased and your programs burned into it. They are also available from CMT on cassette or disk for a small fee.

The CMT-71 EPROM software functions are listed below.

- * PTADDR\$(':port()') - shows hex address of start of port.
- * PTBLANK\$(':port()') - determines if port is blank.
- * PTGENC\$(':port()') - generates proper checksum.
- * PTSIZE\$(':port()') - returns hex size in nibbles of port.
- * PTTEST\$(':port()') - determines if port has proper checksum.
- * EPRMCOPY ':port()' TO 'newfile:hpildev' - copies EPROM image.
in FREEPORTed RAM to mass storage device file.
- * EPRMCOPY 'newfile:hpildev' TO ':port()' - copies EPROM image
file on mass storage to FREEPORTed RAM.
- * ROMTST - executes checksum test of device where located.
- * CLONE - BASIC program, used to COPY master EPROMs.

The following descriptions follow the standard used in the HP-71 reference manual.

PTADDR\$ - Returns the hex address where the requested port is configured in the HP-71 memory map.

Function, Keyboard Execution, IF...THEN...ELSE

-----> PTADDR\$ --> (-----> port specifier ---->) --->

EXAMPLES

```
A$=PTADDR$(':port(4)')    D$=PTADDR$(C$)
DISP PTADDR$(':PORT(2)')
```

Port specifier - String expression

COMMENTS

Returned address is related to the nibble address space of the HP-71.

PTSIZE\$ - Returns the hex size of the requested port in #nibbles.

Function, Keyboard Execution, IF...THEN...ELSE

-----> PTSIZE\$ --> (-----> port specifier ---->) --->

EXAMPLES

```
A$=PTSIZE$(':port(4)')    D$=PTSIZE$(C$)
DISP PTSIZE$(':PORT(2)')
```

Port specifier - String expression

COMMENTS

Returns 10000 hex if 32K byte EPROM, 20000 hex if 64K byte EPROM. This function returns the port's total block size regardless of the number of ICs used to create it.

PTGENC\$ - Generates the proper checksum for the port and places it in the last byte in the port's address space. The hex value returned is the byte written to that location. This function also chains all files that require chaining, before generating the checksum.

Function, Keyboard Execution, IF...THEN...ELSE

-----> PTGENC\$ --> (-----> port specifier ---->) --->

EXAMPLES

```
A$=PTGENC$(':port(4)')    D$=PTGENC$(C$)
DISP PTGENC$(':PORT(2)')
```

Port specifier - String expression

COMMENTS

The proper checksum is that defined by HP as being an 8 bit ADD with wrap-around CARRY summed to equal 01 hex. The string returned always has a length of 5. For a ROM or EPROM, the function returns the checksum but the checksum will not be written to the device. If a BASIC program has a GOTO or GOSUB to a non-existent line number, the function will return an error message: "Statement not found". Depending on the number of files, this function may require a few seconds to complete.

PTTEST\$ - Performs a checksum test on the requested port. The hex string value returned is '00000' if test is OK. If the checksum is incorrect the hex string returned will be the checksum read minus one.

Function, Keyboard Execution, IF...THEN...ELSE

-----> PTTEST\$ --> (-----> port specifier ---->) --->

EXAMPLES

```
A$=PTTEST$(':port(4)')    D$=PTTEST$(C$)
DISP PTTEST$(':PORT(2)')
```

Port specifier - String expression

COMMENTS

The returned string always has a length of 5.

PTBLANK\$ - Returns '00000' if port blank, else returns hex address of first non-blank location.

Function, Keyboard Execution, IF...THEN...ELSE

-----> PTBLANK\$ --> (-----> port specifier ---->) --->

EXAMPLES

```
A$=PTBLANK$(':port(4)')    D$=PTBLANK$(C$)
DISP PTBLANK$(':port(2)')
```

Port specifier - String expression

COMMENTS

Returned address is related to the nibble address space of the HP-71.

EPRMCPY - Copies EPROM images from a mass storage device to the programmer and vice versa.

Statement, Keyboard Execution

```
-> EPRMCPY --> port specifier --> TO --> HPIL Image File
or
-> EPRMCPY --> HPIL Image File --> TO --> port specifier
```

EXAMPLES

```
EPRMCPY ':port(3)' to 'ROMIMAGE:tape'
```

```
EPRMCPY 'ROMIMAGE:tape' to ':port(4)'
```

COMMENTS

ROM image file is compatible with ROMCOPY-generated files. Always test the ROM image after it has been loaded into the FREEPORTed RAM to verify that no errors have occurred. Execute PTGENC\$ command before burning EPROMS to make sure that you have the proper checksum.

HP Resource Allocation Procedure

If your custom software includes LEX files, reserved sytem RAM, or other special requirements, it is good practice to register your uses of rare system resources. This procedure helps to prevent conflicts with other software programmers and ensures compatibility with the HP-71 system operation. Please contact HP Technical Support for more information. Ask for information on HP-71 Resource Allocation -- Assembly Language Support request forms. Additional information is provided in the HP-71 software IDS manual.

Copyright Procedures

It is highly recommended to place a copyright statement within your code to protect your interests. If you have copyrighted your code, include the following comments at the front of your BASIC programs:

```
!(name of the pac) Version x  
!Copyright (c) (company name) (date)
```

The company name should uniquely identify the company. The date should reflect the year the software will be sold.

CAUTION: Federal law prohibits the unauthorized copying of copyrighted material.

CMT-71-P01 EPROM PROGRAMMER MANUAL ADDENDUM
5/8/86

A brief discussion on the use of EPRMCPY and ROMCOPY lexfiles is presented here.

EPRMCPY is a variant of ROMCOPY: same lex ID's, different defaults.

Syntax for both is the same:

```
(Out-going copy)
(ROMCOPY)
EPRMCPY <port specifier> TO [<filename>]<device specifier>
    [<CHECKSUM=<csum1>,<csum2>,... >]
    [<CODESIZE=<size of code in bytes>]
    [<ROMSIZE=<size of EPROM in bytes>]
```

```
(In-coming copy)
(ROMCOPY)
EPRMCPY [<filename>]<device spec> TO <port specifier>
    [<ROMSIZE=<Size of one ROM in bytes>]
```

Use:

To copy a 32K byte ROMIMAGE to tape:

```
PTGENC$(':port(4)')
and
EPRMCPY ':port(4)' to :tape
or
ROMCOPY ':port(4)' to :tape;romsize=32768
```

To copy a 64K byte ROMIMAGE to tape:

```
PTGENC$(':port(4)')
and
EPRMCPY ':port(4)' to :tape;ROMSIZE=65536;CODESIZE=65536
or
ROMCOPY ':port(4)' to :tape;ROMSIZE=65536;CODESIZE=65536
```

To copy a 32K byte ROMIMAGE on tape to a freeported port:

```
EPRMCPY ROMIMAGE:tape to :port(4)
or
ROMCOPY romimage:tape to :port(4);ROMSIZE=32768
and
PTGENC$(':port(4)')
```

To copy a 64K byte ROMIMAGE on tape to a freeported port:

```
EPRMCPY ROMIMAGE:tape to :port(4);ROMSIZE=65536
or
ROMCOPY romimage:tape to :port(4);ROMSIZE=65536
and
PTGENC$(':port(4)')
```

