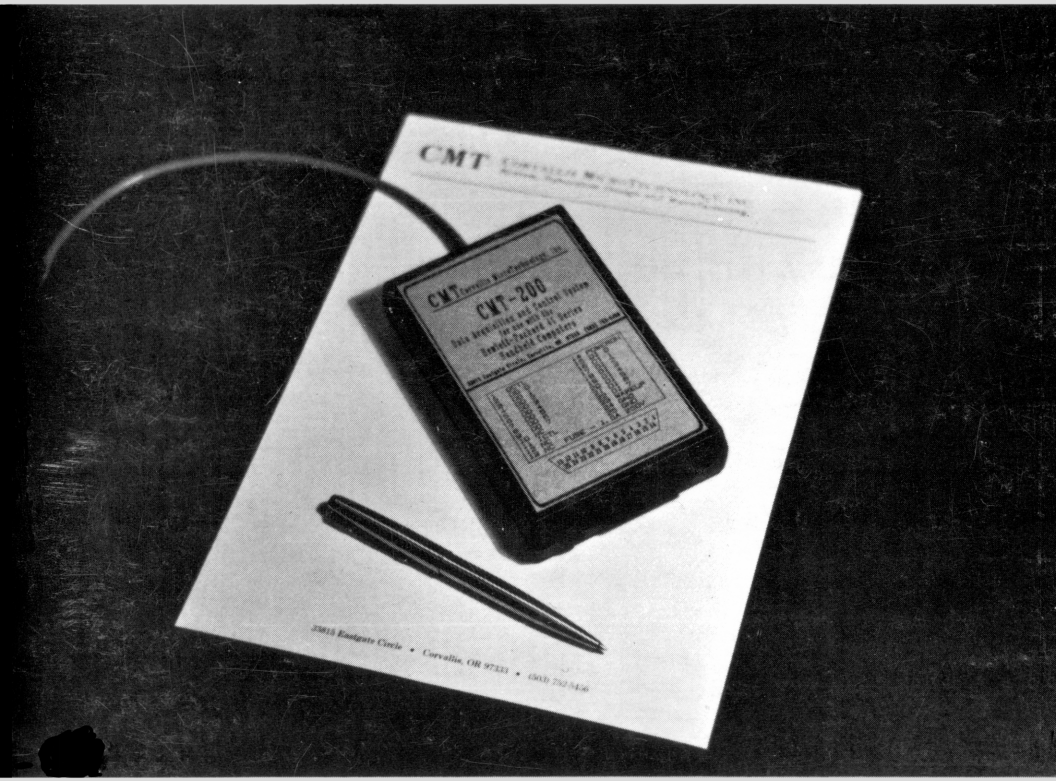
OWNER'S MANUAL

CMT-200

Data Acquisition and Control System



CMT Corvallis MicroTechnology, Inc.

NOTICE

Corvallis Microtechnology, Inc. assumes no liability resulting from any errors or omissions in this manual, from the use of the informaton obtained herein, or from the use of the CMT-200 Data Acquisition and Control System.

CMT-200

Data Acquisition and Control System

OWNER'S MANUAL

CMT Corvallis MicroTechnology, Inc.

CONTENTS

INTRODUCTION

The CMT-200 Data Acquisition and Control System (DACS) is designed to work in conjunction with the HP-41 series calculators as a programmable low-cost hand-held input/output interface and control unit.

The CMT DACS module provides the capability to interface the HP-41 calculator with an external device for the following purposes:

1. To convert an external device with general-purpose input output (GPIO) capabilities into a peripheral of the calculator.

2. To achieve immediate or timed control of external devices via relays and switches, based on commands or measurement results.

3. To allow external devices to trigger control programs residing in the calculator.

4. To provide a means for low-cost, low-power data acquisition even in remote applications.

5. To interface the user's circuitry to the HP-41.

The CMT DACS draws little power from the batteries of the HP-41, and will not appreciably reduce the battery life. While plugged into the HP-41, the DACS can continue control of an external process even after the calculator has been turned off. This is very useful in program-controlled operations over long periods of time.

Another outstanding feature of the DACS is its ease of operation. With the DACS module plugged into one of its ports, the calculator is ready to interact with the external device(s) connected to the DACS. The DACS provides a set of programmable

functions to effect data transfer and external device control. All the standard calculator functions as well as the "external ROM" (XROM) functions of other plugged-in extensions will also be active.

We have included in this package a few examples to familiarize you with the use of the DACS commands. These examples also serve to illustrate some applications of the DACS module. Since the DACS is a general-purpose interfacing and control device, it provides solutions to a great number of application problems.

We encourage you to read or review the HP-41C/41CV/41CX Owner's Handbook and Programming Guide to familiarize yourself with the use of the calculator. If you plan to use any HP-IL devices (such as the 82161A Digital Cassette Drive and the 82162A Thermal Printer), read also the 82160A HP-IL Module Owner's Manual.

Before connecting your DACS to the calculator, be sure to read and follow the instructions given in Section 2, **Connecting the CMT-200 to the HP-41.**

We welcome your comments and suggestions. Also, if you need further assistance, contact our office at:

Corvallis Microtechnology, Inc.
Dept. 200-m
33815 Eastgate Circle
Corvallis, OR 97333
Tel.: (503)752-5456

Section 1

USER SPECIFICATIONS

Hardware Configuration

The DACS has 8 general-purpose output lines, 8 general-purpose input lines, 2 handshake lines, 1 wakeup line, 1 POWERON line and 2 ground lines.

Output Lines (DO0-DO7) - These 8 general-purpose output lines maintain their states even with the calculator shut off. This allows the DACS to continue control of an external process even when the HP-41 is in a power-off condition.

Output Control Line (OCTL) - This is a handshake line controlled by the DACS. This bit is cleared (reset to 0) whenever the HP-41 is in the idle or OFF mode.

Input Lines (DI0-DI7) - These 8 general-purpose input lines are read into the HP-41 in parallel. These are high-impedance lines. Unused input data lines should be grounded, or otherwise taken care of, to avoid picking up spurious signals.

Input Control Line(ICTL) - This is a handshake line controlled by the external circuitry. This is also a high-impedance line.

Wakeup Line - This input line wakes up the HP-41 when it goes high. This provides a means for the external circuitry to activate the HP-41.

POWERON Line - This output line is active whenever the HP-41 is in execution mode. This line can be used to synchronize an external device with the DACS, to clear external logic, or to turn on user's circuitry.

GND Lines - These two lines provide electrical ground. They are connected together inside the DACS.

Electrical Specifications

Output Circuit:

Drive:	Open Drain N-Channel MOSFET
Logic 0:	Non-conductive state with impedance > 1 Megohm
Logic 1:	Conductive state with impedance < 12.5 ohms

Output Technical Data:

Voltage:	30 V maximum
Current:	150 mA maximum @ 25 deg. C per output line
ON Resistance:	10 ohm maximum @ 25 deg. C to 12.5 ohm @ 50 deg. C
OFF Resistance:	1 Megohm minimum
Protection:	SCR crowbar circuit for overcurrent protection, Zener diode for ESD protection, and a 1.5 A fuse

Input Circuit:

Electrical:	N-Channel MOSFET gate with resistive pullup driving a Schmitt trigger
Logic 0:	Input < 0.5V above ground
Logic 1:	Input > 2V

Input Technical Data:

Voltage:	30 V maximum input votage
Impedance:	> 1 Megohm for 0 < Vin < 10V > 100 Kohm for 10 < Vin < 30V
Protection:	Zener for ESD protection
Connector:	DB-25 pin female contact Rating: 1A

Section 2

CONNECTING THE CMT-200 TO THE HP-41

CAUTION: Turn the HP-41 calculator off before inserting or removing the CMT-200 DACS module or any other plug-in extensions or accessories. Failure to do so could damage both the calculator and the plug-in module.

Inserting the DACS Module:

1. Turn the HP-41 calculator off!

2. Remove the port cover of the port you intend to insert the DACS into. Note, however, that you should **never insert the DACS or any other application module into a lower-numbered port than a memory module.** (Turn your calculator over to see a map of the ports.) Save the port cap.

3. Insert the DACS connector into the selected port. Gently push it in all the way.

4. If you are also using the HP-IL module or other application modules, plug them into any port after the last memory module plugged in. You may leave gaps in the port sequence.

5. Turn the calculator on. If the interface loop is used, be sure all the peripheral devices are properly connected and turned on before turning the calculator on. This will ensure proper auto-addressing.

Removing the DACS module:

1. Turn the HP-41 calculator off!

2. Grasp the DACS connector and pull it out.

3. Insert a port cap into the empty port.

Section 3

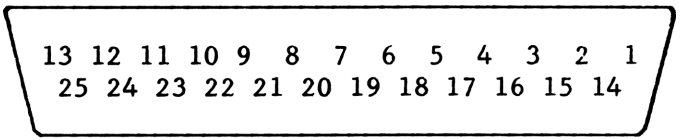
CONNECTING THE CMT-200 TO AN EXTERNAL DEVICE

A standard DB-25 (D-subminiature) connector provides connections for the external device. It is located at one end of the CMT-200, opposite to the plug-in connector. A 25-pin male connector and a cable is required to complete the electrical connection to the external device.

Pin Assignments

Following is a diagram of the pin assignments. Pins 10, 12 and 13 are not used by the CMT-200.

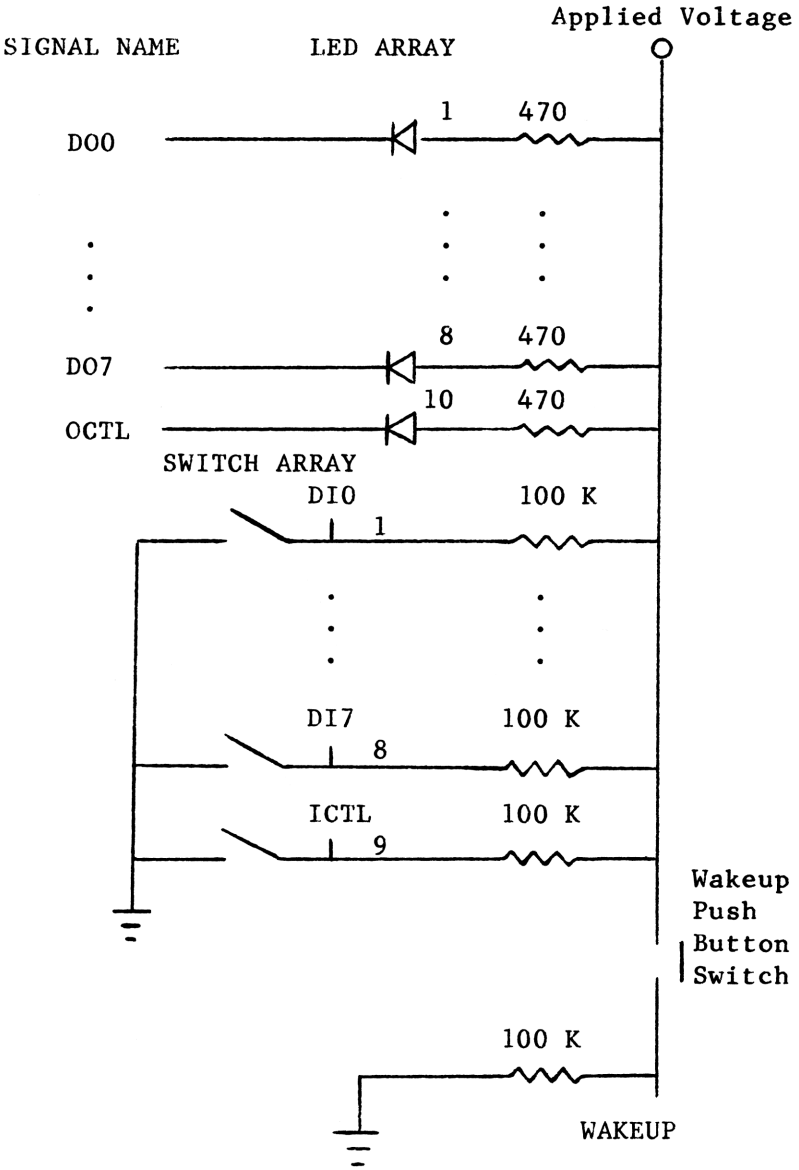
1	OCTL	14	POWERON
2	DO0	15	DI0
3	DO1	16	DI1
4	DO2	17	DI2
5	DO3	18	DI3
6	DO4	19	DI4
7	DO5	20	DI5
8	DO6	21	DI6
9	DO7	22	DI7
10	NC	23	WAKEUP
11	ICTL	24	GND
12	NC	25	GND
13	NC		



Refer to **User Specifications** in Section 1 for a description of these lines.

I/O Test Board

Following is the wiring diagram for an I/O test board that you can readily wire up and connect to the CMT-200. This test board offers a simple way for you to familiarize yourself with the operation of the CMT-200 DACS.



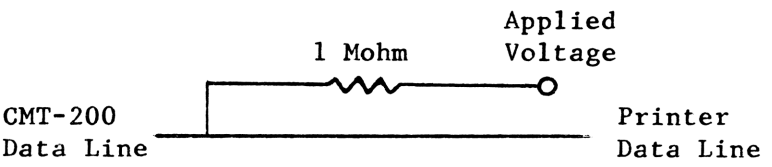
Connecting the CMT-200 to a Parallel Printer

The CMT-200 DACS can output text from the HP-41 calculator to any standard printer that uses the "parallel printer" interface.

Parallel printers with TTL inputs may be connected directly to the CMT-200. For instance, you may connect the MX-80 EPSON DOT MATRIX PRINTER to the CMT-200 DACS as follows:

DB-25 Connector	DACS Signal	Printer Connector	Printer Signal
1	OCTL	1	$\overline{\text{STROBE}}$
2	DO0	2	DATA1
3	DO1	3	DATA2
4	DO2	4	DATA3
5	DO3	5	DATA4
6	DO4	6	DATA5
7	DO5	7	DATA6
8	DO6	8	DATA7
9	DO7	9	DATA8
10	NC	10	$\overline{\text{ACKNLG}}$
11	ICTL	11	BUSY
12	NC	12	PE
13	NC	13	SLCT
24	GND	29	GND
25	GND	30	GND

Pullup resistors are required when connecting parallel printers without TTL inputs to the CMT-200:



Section 4

ABOUT THE CMT-200 DACS FUNCTIONS

The operation of the CMT-200 DACS is controlled by the functions stored in the DACS module. These functions become active in the HP-41 system whenever the DACS module is plugged into the HP-41.

All of the DACS functions may be executed from the display or entered into user programs for remote execution. To execute a function from the HP-41 display, key in the sequence [XEQ] [ALPHA] function name [ALPHA]. To enter the function into a program, use the same key stroke sequence in the PRGM mode.

The DACS functions may also be assigned to HP-41 keyboard locations of your choice (except [ON], [USER], [PRGM], [ALPHA] and the yellow SHIFT key). To assign a function to a keyboard location, key in [SHIFT] [ASN] [ALPHA] function name [ALPHA] [destination key]. To execute a function that has been assigned to a keyboard location, simply press the reassigned key in the USER mode.

The DACS functions are divided into five groups. Each group is designed to carry out a specific set of tasks. These functions are listed and explained in Sections 5 through 9. A reference table is given on the back cover of this manual.

Following are some general remarks related to the DACS functions.

Handshaking

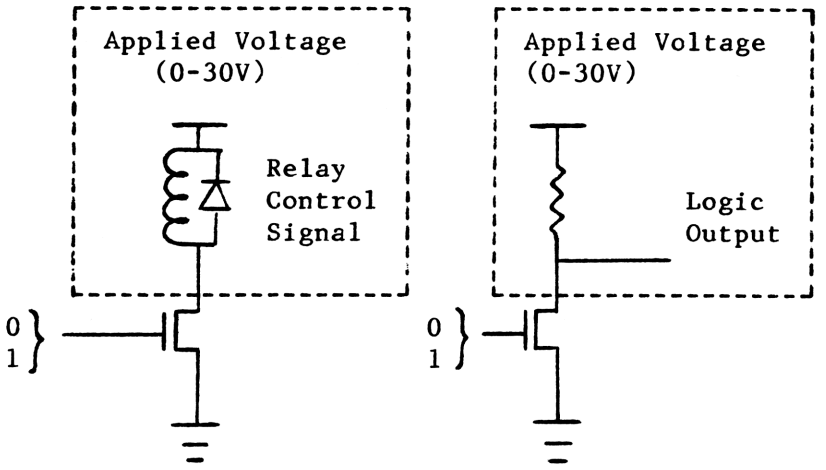
In general, data transfers between two devices need to be synchronized for proper operation. Typically, one device will signal the other device when it has placed valid data on the data lines.

As soon as the data has been accepted, the second device will acknowledge receipt of the data by asserting the approriate control line. This process is often referred to as "handshaking".

The DACS uses the control lines ICTL and OCTL to synchronize data transfer between an external device and the HP-41. Section 8 contains functions that allow the user to choose the handshake mode and delay parameter appropriate for his application. The default handshake mode is HS with the delay parameter 0. (Refer to Section 8.)

Changing the Output Polarity

The output lines of the DACS are used for the purpose of control or data transfer. In the illustration below, those parts of the circuitry enclosed in boxes are external to the DACS.



(1) Control (2) Data Transfer

Default polarity: 1 = closed circuit
0 = opened circuit
Inverted polarity: 1 = opened circuit
0 = closed circuit

There are two possible polarities for the digital output. In the default condition, a 1 bit transfered to one of the output lines DO0 - DO7 means ON when the output line is being used to control an external device. When the output line is used for data transfer with resistor pullups, a 1 bit means 0 volt in the output circuit.

The output polarity can be changed by executing the function **INVON** (Section 7). In this inverted mode, a 1 bit transfered to one of the output lines DO0 - DO7 means OFF when the output line is used for control. When the ouput line is used for data transfer with resistor pullups, a 1 bit means the externally applied voltage will appear across the output circuit.

Interrupts

The functions presented in Section 8 allow a properly connected external device to trigger the execution of a program stored in the HP-41. In this manner, changes in the external process can serve as criteria for executing control functions. The interrupt operation is inactive under default conditions.

The I/O Buffer

The DACS uses a set of program registers for temporary storage of data. This I/O buffer is used in many of the CMT-200 DACS operations. It has an internal pointer whose value can range from zero to the number of bytes in the buffer less one.

Functions are provided for manipulating the buffer, such as creating a buffer of a given size or moving the buffer pointer to a desired point in the buffer. Some other functions move data between the buffer and the external device, or between the buffer and the X-register or the ALPHA register of the HP-41.

The CMT automatically creates a buffer header of 7 bytes to store information pertaining to the current buffer size, pointer position, handshake mode, output polarity and interrupt conditions. These 7 memory bytes are inaccessible to the user. They remain intact as long as the I/O buffer remains intact. However, they are reset to default whenever a buffer is created or resized by the user.

Turning the HP-41 on or off will not affect the I/O buffer if the DACS module remains plugged in. The I/O buffer will be deleted the first time the calculator is turned on without the DACS in place. (This is true if the HP00041-15043 Development Module is not plugged in.)

The Byte

A byte comprises 8 binary digits. 10001011 and 00111000 are examples of a byte. The right-most (least significant) bit is referred to as bit 0; the left-most bit is bit 7.

The byte is the basic data unit used by the DACS. A byte can represent numeric or text data, the states of the data lines, or the status of the HP-41 User Flags 00 - 07. The various ways in which the DACS can interpret a byte are described in **Appendix B.**

Time Module Requirement

Some Pulse Functions supplied by the DACS module require the HP 82182A Time Module to be plugged in for proper operation. (This is not necessary if you are using an HP-41CX calculator.) These functions enable the users to time an external process, control an external device according to pre-specified time data, or count the frequency of occurrence of an external event.

Error Messages

If you attempt to execute any DACS function when the DACS module is not plugged in, the error message **NONEXISTENT** will be displayed. Other error messages and their meanings are listed at the end of the group of functions they are associated with. A summary of error messages is given in Appendix C.

Notes on Programming

All the DACS functions are programmable functions. You can design programs to carry out many operations with or without user intervention. (Note, however, that the VIEWIN function can only be terminated by pressing the [R/S] or [ON] key. The VIEWIN function is described in Section 6.)

Besides the DACS functions and the standard HP-41 functions, you may also use the functions supplied by another plugged-in module. (Note: Do not use the HP-41C Finance Pac simultaneously with the DACS. These two modules share the same XROM number 04.)

For example, you may use the functions provided in the HP 82182A Time Module to achieve long-term timed control or monitoring operations.

Also, the various functions provided by the 82160A HP-IL Module are useful for print or mass storage operations.

To speed up execution of functions or program entry, you may wish to assign the DACS functions to various keyboard locations. Thus the functions can be executed in the USER mode by simply pressing the reassigned keys.

You should not assign functions to keys [A] -[J] and the shifted keys [A] - [E] if you expect to use local ALPHA labels in your programs.

Section 5

SUPPORT FUNCTIONS

The purpose of the following functions is to facilitate the input/output operations. Consult Appendix B for a description of the relationship between a byte, its decimal equivalent, HP-41 user flag status, and ASCII codes.

BUFX - Create an I/O BUFfer of X bytes.

This function creates an I/O buffer for use by the DACS. Its size in bytes is specified in the X-register. Immediately after a buffer is created, the buffer pointer points to the first byte location in the buffer. The initial value of the pointer is 0.

The I/O buffer is created in the program memory of the HP-41. Each program register uses 7 bytes of calculator memory. Therefore, the buffer is always created in units of 7 bytes. If the number specified in the X-register is not a multiple of 7, the buffer size will be rounded up to the next larger increment of 7 bytes.

The maximum buffer size that can be created is 1771 bytes. Make sure you have a sufficient number of free program registers for the buffer before executing this function.

To clear the I/O buffer, place 0 in the X-register and execute this function.

Information pertaining to the current status of output polarity, handshake mode and interrupt operation is stored in the buffer header. Every time the buffer is resized, the previous buffer contents are lost, and the DACS is reset to the default output polarity and handshake mode. Also, any interrupt operation in progress will be turned off automatically. (See Sections 8 and 9.)

X>PT - Use X to set the buffer PoinTer.

The pointer value specified in the X-register may range from 0 to the buffer size in bytes less one. All data transfers into and out of the buffer start at the current pointer position. Use this function to place the pointer at the desired position before performing these operations.

PT>X - Return the buffer PoinTer to X.

The current value of the buffer pointer is returned to the X-register.

X>BUF - Write X to BUFfer at pointer.

The integer (modulo 256) in the X-register is stored in the buffer as a byte. The pointer is advanced afterwards.

BUF>X - Read the next byte in the I/O BUFfer to X.

The next byte in the buffer is placed in the X-register as an integer (0-255). The pointer is advanced afterwards.

A>BUF - Copy ALPHA to the I/O BUFfer.

This function copies the entire ALPHA register contents to the I/O buffer. Leading null characters are ignored. This operation starts at the buffer pointer. The buffer pointer is advanced past the last character copied.

The ALPHA register contents are stored as binary ASCII codes in the buffer. Each character takes up one byte of buffer space.

BUF>AX - From I/O BUFfer copy into ALPHA X bytes.

This function appends a number of bytes to the ALPHA register from the buffer, starting from the buffer pointer.

The number of bytes to be copied to the ALPHA register is specified in the X-register. The buffer pointer is advanced after the operation.

X>A - Shift X into ALPHA from the right.

This function appends the X-register contents to the existing ALPHA register contents.

The number placed in the X-register is the decimal ASCII code of the character to be copied into the ALPHA register.

Example:

(Press the [ALPHA] key before and after keying in any function name. To access the > symbol, press shifted [TAN] in the ALPHA mode.)

Keystrokes	Comments
7 [XEQ] **BUFX**	Create an I/O buffer.
66 [XEQ] **X>BUF**	The binary equivalent of 66 will be stored at the first byte location in the buffer.
[ALPHA] CDEFG [ALPHA]	Key in ALPHA data.
XEQ **A>BUF**	The binary ASCII codes of the characters in the string "CDEFG" will be stored at the 2nd through the 6th byte locations in the buffer.
[ALPHA] A [ALPHA]	The ALPHA register now contains "A".
0 [XEQ] **X>PT**	The pointer is reset to zero.
6 [XEQ] **BUF>AX**	The first 6 bytes of the buffer are copied into the ALPHA register.

[ALPHA]	Turn ALPHA display on. The characters "ABCDEFG" are displayed.
[ALPHA]	Turn ALPHA display off.
72 [XEQ]**X>A**	Append the character whose decimal ASCII code is 72 to the existing ALPHA register contents.
[ALPHA]	The characters "ABCDEFGH" are displayed.
[ALPHA]	Turn the ALPHA display off.

ANDXY - Do a logical AND of X and Y.

The binary equivalents of the integers (modulo 256) in the X- and Y-registers are ANDed. The result is placed in the X-register as an integer.

ORXY - Do an inclusive OR of X and Y.

An inclusive OR is performed on the binary equivalents of the integers (modulo 256) in the X- and Y-registers. The result is placed in the X-register as an integer.

NOTX - Do a 1's complement (NOT) of X.

This function inverts all the bits of the binary equivalent of the integer (modulo 256) in the X-register. The resulting byte is placed in the X-register as an integer.

X>FLAG - Copy X to FLAGs 00 through 07.

This function sets the user flags 00 through 07 according to the binary equivalent of the integer part of the X-register contents. Any number from 0 through 999 in the X-register is valid, but only the low eight bits will be used to set the flags.

FLAG>X - Copy FLAGs 00 through 07 to X.

This function places into the X-register the decimal equivalent of the binary number representing the status of the user flags 00 through 07.

This provides a convenient means for setting up masks used by other functions. (See Sections 7 and 9.)

FLAG<>X - Exchange FLAGs 00 through 07 with X.

This function has the combined effect of the above two functions. It places the status of flags 00 through 07 into the X-register as an integer. In additions, it sets flags 00 through 07 according to the binary equivalent of the integer part of the original X-register contents.

ERROR MESSAGES:

BSIZE>1771 Size specified for the buffer is too large.

NO BUFFER This message will be displayed if you attempt to execute a function involving the buffer or the buffer pointer when a buffer has not been previously created.

NONEXISTENT The contents of the X- or Y-register exceed 999.

PACKING Program memory is being packed to provide more room for the buffer.

REACH EOB Reached the end of the I/O buffer.

TRY AGAIN Execute BUFX again. If there still is not enough room for the buffer, specify a smaller buffer size, or use the HP-41 function SIZE to reduce the number of data storage registers.

Remark: The functions BUF-RGX and RG-BUFX in the HP-IL Development Module (HP 00041-15043) can be used to transfer data between the buffer created by the DACS and a block of data registers. This allows you to print or store bolcks of buffer data via the printer and mass storage operations available in the HP 82160A HP-IL Module.

While the Development Module is in one of the HP-41 ports, the I/O buffer remains intact even with the CMT-200 DACS removed.

Section 6

DIRECT I/O FUNCTIONS

The direct I/O Functions allow direct transfer of information between the HP-41C (the controller) and the external device. These functions can be used in applications that do not require synchronization for proper information exchange. Device control through the activation of relays and switches is one such application.

Refer to Appendix B for a description of the correspondence between a byte and a decimal number.

The I/O Test Board described in Section 3 can be very helpful in demonstrating the operation of the direct I/O functions.

IDX - Input Data to X.

This function reads the 8 input data lines and places the byte as an integer (0 - 255) in the X-register of the calculator. The integer is the decimal equivalent of the 8-bits representing the states of the input data lines. The right-most bit is the least significant bit.

Example:

The integer 70 is placed in the X-register if the status of the input lines is as shown below.

Line	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
State	0	1	0	0	0	1	1	0

If you have the I/O Test Board (described in Section 3) wired up and connected to the DACS, execute IDX to display the integer corresponding to a given switch setting.

ICX - Input Control line (ICTL) to X.

This function reads the input control line and places the result (0 or 1) in the X-register. When the ICTL line is high, 1 is placed in the X-register.

ODX - Output Data from X.

This function sets the 8 output lines according to the binary equivalent of the integer (modulo 256) in the X-register. When the DACS is first plugged into the calculator, these output lines may be in any state (0 or 1).

Example:

If the number placed in the X-register is 130, the status of the output lines is as shown below. The default output polarity is assumed to be effective here, ie. 1 = closed circuit, 0 = opened circuit.

Line	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
State	1	0	0	0	0	0	1	0

If you have the I/O Test Board (described in Section 3) wired up and connected to the DACS, place a number in the X-register, and execute ODX to observe the corresponding LED(s) light up.

OCX - Output Control from X.

This function sets the output control line (OCTL) according to the contents of the X-register. It sets the output control line to 1 if the number in the X-register is nonzero. The output control line is set to 0 if the X-register contains zero.

The OCTL line is reset to 0 whenever the HP-41 is idle (not executing any functions) or turned off. If this function is executed as part of a program, the OCTL line maintains its state as long as the

program is running. To prevent this bit from being reset, you may execute the ON function to keep the HP-41 in execution mode. This, however, consumes a great deal of calculator power.

VIEWIN - VIEW the input lines.

This function continuously reads and displays the 8 input data lines and the input control line until the [R/S] key is pressed. The states of these input lines are displayed as either 0 or 1.

The left-most digit in the display shows the state of the input control line. The second digit from the left is bit 7 of the byte representing the states of the input data lines. The right-most digit is bit 0 of the byte representing the states of the input data lines.

Pressing any key but [ON] or [R/S] will freeze the display until the key is released.

Example:

With the DACS connected to an external device, the display might show 100111100. This corresponds to the following status of the input lines:

Line	ICTL	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
State	1	0	0	1	1	1	1	0	0

ERROR MESSAGE:

NONEXISTENT The number placed in the X-register is greater than 999.

Section 7

PULSE FUNCTIONS

The following functions incorporate timing capabilities into the input/output operations. Time intervals are specified in centiseconds. (A centisecond is a hundredth of a second, or 10 milliseconds.)

Time data are interpreted as BCD numbers before being stored into the I/O buffer. Each time interval is allotted 2 bytes of buffer space. Therefore, the maximum time that can be stored in the I/O buffer is 99.99 seconds.

Refer to Appendix B for a description of the relationship between a byte and a BCD number or a decimal number. Section 5 contains functions for manipulating the I/O buffer.

TX>BUF - Store the Time in X into the I/O BUFfer.

The time is specified as an integer in the X-register. The time data are stored in the buffer starting at the current position of the buffer pointer. The buffer pointer is moved past the data afterwards. If the contents of the X-register exceed 9999, 9999 will be stored instead of the specified number.

BUF>TX - Copy the next two bytes in the I/O BUFfer as Time data to X.

The time data is returned to the X-register as an integer in hundredths of a second. The two bytes are taken from the buffer starting at the current buffer pointer. The pointer is moved past the data afterwards.

TIMEI - Time the Input signal.

This function times a number of state changes of certain input lines. The number of state changes to be timed is specified in the X-register. The input lines of interest are specified by a mask in the Y-register.

A "mask" is an integer whose binary equivalent will be ANDed with the input data lines. This provides a means for selective monitoring of the input data lines. For instance, place 31 in the Y-register if you are only interested in monitoring lines DI0 through DI4.

A change in state (from 0 to 1, or vice versa) in any of the lines specified by the mask constitutes a "state change".

The time intervals between successive state changes are stored as BCD numbers in the I/O buffer. The time data are stored into the buffer starting from the current buffer pointer. The buffer pointer is moved past the data afterwards.

An appropriate I/O buffer should have been created prior to executing this function. (See **BUFX** in Section 5.) Remember that each time data requires 2 bytes of storage space in the buffer.

Example:

(Press the [ALPHA] key before and after entering the function names.)

Keystrokes	Comments
14 [XEQ] **BUFX**	Create a buffer.
3 [ENTER ↑]	Place the mask in the Y-register. Only DI0 and DI1 will be monitored.
7 [XEQ] **TIMEI**	Record the time for 7 state changes.

0 [XEQ] X>PT	Reset pointer to zero.
[XEQ] **BUF>TX**	Do this step 7 times to view the time data stored in the buffer.

TIMEO - Toggle at predetermined TIMEs the Output signal.

This function alters the states of the output lines D01 through D07 according to the time data stored in the I/O buffer, starting from the current buffer pointer. The number of times you want the output lines to be toggled is specified in the X-register. The initial status of the 8 output lines is specified in the Y-register.

Upon execution of this function, the output lines will first be set according to the integer (modulo 256) in the Y-register. They are then inverted at the end of each specified time interval.

Example:

Keystrokes	Comments
7 [XEQ] **BUFX**	Create a buffer.
1000 [XEQ] **TX>BUF**	Store the time data into the buffer at the first 2 byte locations.
800 [XEQ] **TX>BUF**	Store the time data into the buffer at the next 2 byte locations.
0 [XEQ] **X>PT**	Reset the buffer pointer to zero.
9 [ENTER ↑]	Place the initial status of the output lines in the Y-register.
2 [XEQ] **TIMEO**	The output lines will be toggled as follows. Initial status: 00001001 1000 cs later: 11110110 800 cs later: 00001001

MATCHI - MATCH the Input lines with a given bit pattern.

This function compares the status of the input lines specified by a mask in the Y-register with the bit pattern represented by the integer in the X-register.

When a match is obtained, the function places in the X-register the length of time (in cs) it waited for the match. If the [R/S] key was used to terminate the wait, the wait time is displayed as a negative number.

Example:

Keystrokes	Comments
14 [ENTER ↑]	Place the mask (14) in the Y-register. Lines DI1, DI2 and DI3 will be monitored.
6 [XEQ] **MATCHI**	Wait time will be displayed when DI1 and DI2 both read 1, and DI3 reads 0, irrespective of the status of the other lines.

RATE - Count events in the time interval X.

The number of changes in the lines specified by the mask in the Y-register is tallied during the interval (in cs) specified in the X-register. At the end of the interval the number of events counted is placed in the X-register.

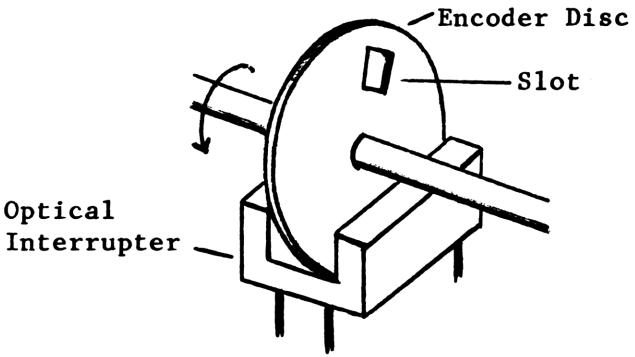
Pressing [R/S] will cut the interval short, and the number of events tallied to that time will be displayed as a negative number.

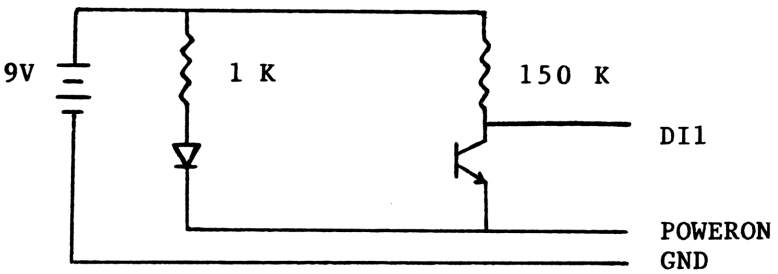
Example: Motion Sensing

Motion sensing can be easily achieved by connecting an optical interrupter to the CMT-200 DACS. The optical interrupter normally consists of an infrared emitting diode and a phototransistor separated by an air gap. The interruption of the infrared beam causes an on/off signal from the phototransistor. A slotted circular disc or a straight comb with rectangular openings may be used to cause the light path to open or close.

The circuit below shows how DI1 might be wired with an optical interrupter with these output current characteristics:

100 nanoamp	Dark
500 microamp	ON (min.@ 10 V)





The program listed below illustrates the use of the RATE function for measuring windspeed. For this application, a circular disc with a single slot is mounted so that it rotates in the air gap of the optical interrupter at the same rate as the windcups. Thus every revolution of the disc produces 2 state changes in DI1.

$$\text{Windspeed} = \frac{2 \times 3.1412 \times r}{2 \times t} \times \frac{60}{5280} \times n \quad \text{(mph)}$$

where r = distance of windcups from the center of rotation (ft)
n = number of state changes in DI1 counted during time interval t
t = time interval (min.)

We assume that r = 1 and t = 1/2. Therefore,

Windspeed = 3 x 3.1412 x n / 132 (mph)

Program	Comments
01 LBL^{T}DWS	Program name.
02 LBL 01	
03 CLA	Clear the ALPHA register.
04 2	Mask selecting line DI1.
05 ENTER ↑	Place mask in the Y-register.
06 3000	30 sec = 3000 cs.
07 RATE	Count number of state changes during the 30 sec interval.
08 ENTER ↑	
09 3	
10 *	
11 PI	
12 *	Windspeed calculation.
13 132	
14 /	
15 ARCL X	Place result in ALPHA register.
16 ⊢ MPH	Append units to the result.
17 AVIEW	Display windspeed.
18 GTO 01	Do next sampling.
19 END	

Note that this same program may be used to measure the flow rate of a river. It can be modified so that the HP-41 is waked up at specified intervals to take the data. Most probably you would want to include the HP 82161A Digital Cassette Drive in the system setup so as to secure a permanent record of the data.

WAITIX - WAIT for the Input lines to change states X times.

This function waits for the states of the lines specified by the mask in the Y-register to change a given number of times. The desired number of state changes is specified in the X-register.

After the specified number of state changes have occurred, the length of the wait time is placed in the X-register.

If the [R/S] key is used to terminate the wait, the curtailed wait time will be displayed as a negative number.

Example:

Keystrokes	Comments
6 [ENTER ↑]	Place the mask in the Y-register. Only DI1 and DI2 will be monitored.
4 [XEQ] WAITIX	Toggle DI1 or DI2 4 times. The wait time will be displayed right after the 4th state change has taken place.

ERROR MESSAGES:

NONEXISTENT	The contents of the X- or Y-register exceed 999.
NO BUFFER	A buffer has not been created.
NO TIMER	The Timer module is not present in the HP-41C or HP-41CV.
REACH EOB	Reached the end of the I/O buffer.

Caution:

The pulse functions draw considerably more battery power than the other functions. The functions TIMEI, MATCHI and WAITIX will wait indefinitely (or until the HP-41 batteries run out of power) for the specified state change to occur unless the [R/S] key is pressed or the calculator is turned off.

Section 8

DATA OUTPUT AND INPUT WITH HANDSHAKE

With one input control line and one output control line, a parallel handshake is implemented for text data input and output. In the HandShake mode, both control lines are used. In the STROBE mode, only one handshake line is used. These modes will only affect the functions given in this section. The handshake protocol is given below following the function descriptions.

Refer to Section 5 for a description of the I/O buffer. See also Appendix B for a description of the correspondence between a byte and an ASCII character.

HS - HandShake mode active.

In this mode, both control lines ICTL and OCTL are used for handshake for input and output.

This function calls for a delay count parameter to be placed by the user in the X-register. Each count lasts about .75 msec.

For a device that is slower than the DACS, the delay parameter should be specified as 0. In this case, the DACS will wait indefinitely for a change to occur in the status of the ICTL as an indication that the external circuitry is busy or ready for data. (Note: Here the delay parameter 0 is used to denote an indefinite delay interval.)

For other devices, specify a non-zero delay interval. This takes care of situations where the DACS is too slow to respond to the rapid changes in the ICTL line. In this case, the delay parameter is used as a basis for timing the data transfer. For example, the handshake mode to use with a standard parallel printer is HS with a delay of 1 count.

The default handshake mode is this handshake mode with an indefinite delay (delay parameter 0).

STROBE - STROBE mode active.

In this mode, only the OCTL is used for handshake for input and output.

The duration between successive character inputs or outputs is specified as a delay count parameter in the X-register. Each count lasts about .75 msec. The maximum delay parameter allowed is 255, so the longest possible wait is .19 seconds. If the delay parameter exceeds 255, it will be taken as 255.

INVON - INVert mode ON.

This function inverts the bits of every byte sent to the output lines DO0-DO7. This mode remains in effect as long as the I/O buffer remains intact.

Execute this function before attempting to send data to a parallel printer.

INVOFF - INVert mode OFF.

The polarity of the output lines is restored to that in the default conditions.

OA - Output ALPHA to the output lines.

This function outputs the contents of the ALPHA register as ASCII codes to the output lines DO0-DO7. The end line sequence, CR (carriage return) and LF (line feed), is automatically appended to the output string unless the user flag 17 is set.

Example:

The following program will allow you to print text to a parallel printer from the ALPHA register. (See "Connecting the CMT-200 to a Parallel Printer" in Section 3.)

Program		Comments
01	LBL^{T}PT	Program name (Print Text).
02	1	Delay parameter (1 count).
03	HS	HandShake mode selected.
04	INVON	Output polarity inverted.
05	AON	Turn ALPHA mode on.
06	LBL 01	
07	PROMPT	Key in up to 24 characters and press [R/S] key.
08	OA	Text will be printed in one line.
09	GTO 01	Repeat the process.
10	END	

IA - Input data to the ALPHA register.

This function inputs text to the ALPHA register. The data input is terminated by a LF or the condition of a full ALPHA register. The CR is ignored.

The ALPHA register is cleared at the beginning of the function. If the ALPHA register is full before the LF is detected, input will be terminated, and user flag 17 will be set to indicate that the record is incomplete.

IAX - Input data to ALPHA by X.

This function inputs a given number of characters into the ALPHA register. The number of characters to be input is specified in the X-register. No terminating character is used.

The ALPHA register is not cleared at the beginning of this function. Therefore, the characters input by this function will be appended to the existing ALPHA register contents.

Input will not stop when the ALPHA register becomes full. Any character input after the ALPHA register is full will push the left-most character out of the ALPHA register.

OBUFX - Output from the I/O BUFfer by X.

This function outputs a number of bytes from the I/O buffer, starting from the buffer pointer.

The number of bytes to be output is specified in the X-register. After the operation, the pointer is moved past the last data output.

IBUFX - Input to the I/O BUFfer by X.

This function inputs a number of bytes and stores them in the I/O buffer.

This operation starts from the buffer pointer. The pointer is moved past each byte as it is entered. The number of bytes to be input to the buffer is specified in the X-register.

ERROR MESSAGES:

NONEXISTENT	The number specified in the X-register exceeds 999.
REACH EOB	Reached the end of the I/O buffer.

How the Handshake Works

The following descriptions refer to polarities measured at the DO pins assuming you are using resistive pullups to a positive supply. The high voltage is indicated by a 1 or "true"; the low voltage is indicated by a 0 or "false".

The states of the ICTL and OCTL lines are interpreted as follows during data I/O operations.

Output Data:

ICTL	1	Peripheral is busy.
	0	Peripheral is ready to receive data.

	1	Data is not available.
OCTL		
	0	Data is available.

Input Data:

	1	Data is not available.
ICTL		
	0	Data is available.
	1	The DACS is busy.
OCTL		
	0	The DACS is ready to receive data.

HandShake Mode

There are two versions of this handshake mode. They are distinguished by the delay parameter specified for the HS function.

A. HandShake mode with an indefinite delay.

This is the default handshake mode. It can also be set with the HS command by specifying a delay parameter of 0 in the X-register.

This handshake method requires the DACS to be able to respond to every transition in the ICTL line.

B. HandShake mode with a non-zero delay parameter

This mode is set with the HS command by specifying a number from 1 through 255 in the X-register.

This method of handshake is to be used when data is transfered between the DACS and a device faster than the DACS software.

The steps for data transfer are similar for these two variations of the HandShake mode, except that when a non-zero delay parameter is specified, the delay time is used for timing the data transfer.

Output data:

1. Set the DO lines according to the next output byte. (With INVON set, the polarity of these lines will correspond to the bits of the byte transfered to the output lines.)
2. Wait for ICTL to become 0, indicating that the external circuitry is ready to receive data.
3. Set OCTL to 0 to indicate that the data is available.
4. Do one of the following.
   a) If the delay parameter is 0:
      Wait for ICTL to become 1, indicating that the external circuitry has received the data and is now busy. (Do not go to the next step if this condition has not been met.)
   b) If the delay parameter is not 0:
      Wait for ICTL to become 1, indicating that the external circuitry has received the data and is now busy. If a 1 state of the ICTL has not been detected by the end of the specified delay interval, go to step 5 anyway.
5. Set OCTL to 1, indicating that the data is no longer available. Go back to step 1.

Input data:

1. Set OCTL to 0, indicating that the DACS is ready to receive data.
2. Wait for ICTL to become 0, indicating that the data is available.
3. The DACS reads the data lines.
4. Set OCTL to 1, indicating that the input lines are being read and the DACS is now busy.

5. Do one of the following.
   a) If the delay parameter is 0:
      Wait for the external circuitry to set ICTL to 1, indicating that it acknowledges that the DACS has received the data. Go to step 1 only when this condition has been met.
   b) If the delay parameter is not 0:
      Wait for the external circuitry to set the ICTL to 1, indicating that it acknowledges that the DACS has received the data. If a 1 state of the ICTL has not been detected by the end of the specified delay interval, go to step 1 anyway.

STROBE Mode

This mode is set with the STROBE command by specifying a number from 0 to 255 as delay parameter in the X-register.

Output data:

1. Set OCTL to 1, indicating that the data is not available. Set the output lines according to the next output byte. (The polarity of the output lines can be set with the INVON and INVOFF commands.)
2. Set OCTL to 0 for the delay interval specified in the STROBE command. This indicates that the data is available. Go to step 1.

Input data:

1. Set OCTL to 1, indicating that the DACS is busy.
2. Set OCTL to 0, indicating that the DACS is ready to receive data. Wait through the delay interval specified in the STROBE command. Read the input lines and go to step 1.

Section 9

WAKING UP THE HP-41

The wakeup line in the CMT-200 DACS is capable of waking up the HP-41 from the off mode or the idle mode. The off mode is the power-off condition obtained by pressing the [ON] key or executing the function **OFF**. In the idle mode, the display is on, but the calculator is not executing any functions.

As long as the wakeup line is held high, the HP-41 will be awake. In this state, the HP-41 will either be executing a function or checking to see why it has been waked up. Therefore, even though the display may not be on, the HP-41 is draining current as long as the wakeup line is high.

The following functions give you flexibility in controlling the interrupt.

INTON - INTerrupt ON.

This function will be executed when both of the following conditions are met.

1. The wakeup line is high.

2. Any one of the input data lines specified by a mask is high. The mask is specified as an integer (0 - 255) in the X-register.

If the ALPHA register contains the name of a program, this program will be executed when the HP-41 is waked up by the DACS.

If the ALPHA register is empty when this function is executed, the HP-41 will just be turned on from the off mode, or go back to the idle mode.

Read the remark given later in this section. Be sure you understand it before executing this function.

INTMASK - Set the INTerrupt MASK by X.

This function allows you to change the interrupt mask without changing the interrupt program name in the ALPHA register. Specify the mask of the input data lines as an integer (0 - 255) in the X-register.

INTOFF - INTerrupt OFF.

This function turns off the interrupt. When the interrupt is turned on again, the interrupt mask and the interrupt program name have to be specified anew.

*** Remark: Be sure to include one of the following steps in the program to be run when the HP-41 is waked up by the DACS. Otherwise, the calculator will get into an endless loop - It executes the specified program, goes back to sleep, and is waked up right away to execute the program over again because the cause of the interrupt is still active.

1. Execute INTMASK to set the interrupt mask to zero or something different from its current pattern.
2. Execute INTON to clear the interrupt mask or the program name. (Clear the X-register or the ALPHA register before executing this function.)
3. Execute INTOFF to turn off the interrupt.

In case you get into a vicious loop, you may get out of it using either of the following methods.

1. Reset the input lines involved to 0. Execute INTOFF to turn off the interrupt.
2. Turn off the HP-41 by simultaneously pressing and holding down the [ON] and [ENTER ↑] keys. Remove all plugged-in modules. Turn on the HP-41. This clears the interrupt information stored in the I/O buffer header. Turn off the HP-41 before reinserting the modules.

Example:

First key in the programs "BYE" and "HI".

Program		Comments
01	LBL^{T}BYE	Main program name.
02	^{T}HI	Name of program to be executed when the HP-41 is waked up by the DACS.
03	2	Interrupt mask. Only DI1 will be monitored.
04	INTON	Turn on the interrupt.
05	OFF	Turn off the HP-41.
06	END	End of program.

Program		Comments
01	LBL^{T}HI	Program name.
02	ASTOX	Copy "HI" from the ALPHA register to the X-register.
03	INTOFF	Turn off the interrupt.
04	END	End of program.

Now, when you execute the program "BYE", the HP-41 will be turned off. When the wakeup line and DI1 are both high, the HP-41 will be waked up to execute the program "HI". As a result the program name "HI" will be displayed. Note that INTOFF is used in the program "HI" to turn off the interrupt.

If you have the I/O Test Board (described in Section 3) wired up and connected to the DACS, the above condition of the wakeup line and DI1 being simultaneously high can be obtained by setting switch 2 on and pressing the wakeup push button switch.

ERROR MESSAGES:

NONEXISTENT The number placed in the X-register exceeds 999, or the program named in the ALPHA register does not exist.

Appendix A

MAINTENANCE INFORMATION

Initial Inspection

When you first receive the device, examine the package for signs of damage. Check to see that the device is in good mechanical condition. Also, check the fuse located at one end of the CMT-200 module.

If there is any mechanical damage, notify our office. If the shipping container is damaged, notify the carrier as well as our office.

Care of the Unit

* Always turn off the calculator before connecting or disconnecting the CMT-200 DACS module or any peripheral.
* Keep the electrical contacts of the module clean. When necessary, carefully brush or blow the dirt out of the contact area. Do not use any liquid for cleaning the contacts.

* Store the module in a clean, dry place.
* Observe the temperature specifications:
  Operating: 0 to 45 deg. C (32 to 113 deg. F)
  Storage: -20 to 55 deg. C (-4 to 131 deg. F)
* Observe the electrical specifications given in Section 1 of this manual.

Replacing the Fuse

In case you need to replace the fuse, use a screw driver to remove the original fuse. Use 5 mm x 20 mm 1.5A 250 V fast-blow fuse for replacement.

Before replacing a blown fuse, verify that the external circuitry operates within the current and voltage ranges specified for the DACS. (See **Electrical Specifications** in Section 1.)

Warranty Information

WARRANTY

This CMT product is warranted against defects in material and workmanship for a period of 90 days from the date of shipment. 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 to CMT from any country outside the U.S.A.

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

LIMITATION OF WARRANTY

The foregoing warranty shall not apply to defects or malfunctioning resulting from improper or inadequate maintenance by buyer, buyer-supplied interfacing, unauthorized modification or misuse, or operation outside of 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.

Shipping Instructions

Should the need arise, ship your DACS in a protective package to avoid damage. (Use the original shipping container and cushioning material.) 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 the sales receipt with the unit.

Appendix B

INTERPRETING A BYTE

The DACS interprets a byte in one of the following ways:

As Data Lines Status

Each binary digit represents the state (0 or 1) of a data line. Bit 0 corresponds to the state of the input line DI0 or that of the output line DO0; bit 7 corresponds to the state of the input line DI7 or that of the output line DO7.

Example:

The byte 00100011 may be interpreted as the status of the output lines. It shows that lines DO0-DO7 assume the following states:

Line	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0
State	0	0	1	0	0	0	1	1

As a Decimal Number

The decimal equivalent of a byte is obtained by finding the sum of powers of 2 given in the following table.

Bit	7	6	5	4	3	2	1	0
Binary	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal	128	64	32	16	8	4	2	1

Example:

The decimal equivalent of 10001010
= 2 + 8 + 128 = 138

As a Binary-Coded Decimal Number

Decimal numbers can be coded using binary digits. Four bits are required to code the decimal digits as follows:

BCD	Binary
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

Therefore, each byte can be interpreted as 2 decimal digits.

Example:

The byte 00010110 can be interpreted as the BCD number 16.

As an ASCII code

In data transfer opertions, a byte represents an ACSII character. ASCII codes are given in binary or decimal form. Consult the printer manual for the character set your printer supports.

Refer to Page 310 of the HP-41CX Owner's Manual (Volume 2) for a listing of the characters that can be displayed on the HP-41. The HP-41 shows the "starburst" display (all display segments lit) for all decimal codes from 128 through 255.

Example:

The byte 01000001 corresponds to the standard ASCII character A. Its decimal equivalent is 65.

As User Flags Status

A byte can be interpereted as the user flags 00 through 07. Each 0 or 1 bit represents the status (cleared or set) of the flag it represents.

The least significant (right-most) bit corresponds to flag 00, while the most significant bit (left-most) bit corresponds to flag 07.

As mentioned above, a byte can also be interpreted as a decimal number. Thus it serves as an intermediary in flag-number conversions. (Refer to X>FLAG, FLAG>X and FLAG<>X described in Section 5 of this manual.)

Example:

The byte 00000101 represents the following flag status:

Flag number	Flag status
00	set
01	clear
02	set
03	clear
04	clear
05	clear
06	clear
07	clear

Appendix C

ERRORS AND PROBLEMS

Error Messages

Following is a summary of error messages related to the DACS functions only. For an explanation of other messages that might appear, consult the manuals for the calculator and any other module you have in your system setup.

Message	Functions	Meaning
BSIZE>1771	BUFX	Buffer size specified is too large.
NO BUFFER	A>BUF BUF>AX BUF>TX BUF>X IBUFX OBUFX PT>X TIMEI TIMEO TX>BUF X>BUF X>PT	No buffer has been created. Use BUFX to create a buffer of appropriate size.
NONEXISTENT	-all-	DACS module is not plugged in.
	ANDXY FLAG<>X IAX INTMASK INTON MATCHI NOTX ODX ORXY RATE TIMEI TIMEO WAITIX X>BUF X>FLAG	Contents of X- or Y-register exceed 999.

	INTON	Program named in the ALPHA register does not exist.
NO TIMER	MATCHI RATE TIMEI TIMEO WAITIX	Time Module is not plugged into HP-41C/CV.
PACKING	BUFX	Program memory is being packed to provide more room for buffer.
REACH EOB	A>BUF BUF>AX BUF>TX BUF>X IBUFX OBUFX PT>X TX>BUF X>PT X>BUF	Reached the end of the buffer. Check to see if buffer is large enough. Was pointer at desired location when function was executed?
TRY AGAIN	BUFX	Execute BUFX again. If there is still not enough room, reduce buffer size or allocate a larger number of registers to program memory.

In Case of Difficulty

Any time you suspect that your calculator or DACS is not operating properly, check the following for a possible cause.

Buffer Altered

The contents of the buffer are erased whenever you resize the buffer or turn on the HP-41 while the DACS is disconnected from it. This includes the information stored in the buffer header. When the buffer is resized or deleted, the DACS assumes the default conditions of HS handshake mode with zero delay, no output polarity inversion and no active interrupt.

Duplicate XROM Numbers

Do not simultaneously use any plug-in ROM modules with duplicate ROM identification numbers. The DACS module has the XROM number 04 which is shared by the HP-41C Finance Pac. Using the Finance Pac simultaneously with the DACS may cause problems.

Endless Loop

When you get into an endless program loop involving the INTON function, recover from it by resetting the input lines involved to 0. Then, execute INTOFF to turn off the interrupt.

Improper Handshaking

Data transfer will not take place if proper handshake has not been established. If you run into difficulties with input/output operations, use the [R/S] key to terminate the data transfer function. Then change the handshake mode and delay to those appropriate for the peripheral used.

Incorrect Wiring of External Circuits

Improper wiring can produce unexpected results. Make sure your system setup meets the electrical specifications for the DACS. Check to see that the design of the external circuit is correct.

OCTL Reset

Remember that the OCTL will be reset to 0 whenever the HP-41 is in a power-off condition.

Peripheral Not Connected

Make sure the peripheral used is properly connected to the DACS. Press the [R/S] key to terminate a function that is waiting for a response from a disconnected peripheral.

Notes:

Addendum
(CMT-200 Manual)

Page 1

All the input lines are high-impedance lines. These include the 8 input lines DI0 through DI7, the Input Control Line (ICTL) and the Wakeup Line. Ground all unused input lines to avoid picking up spurious signals.

*** Important ***

Some HP-41CX owners may experience difficulties while using repeating alarms. For example, the data-logging process may suddenly stop after a number of iterations. If this happens, contact Service Department, Hewlett-Packard Company, 1000 N.W. Circle Blvd.,Corvallis, OR 97330, and request a capacitor update.

ANDXY Do a logical AND of X and Y.

A>BUF Copy ALPHA to the I/O BUFfer.

BUF>AX From I/O BUFfer copy into ALPHA X bytes.

BUF>TX Copy the next two bytes in the BUFfer as Time data to X.

BUF>X Read the next byte in the I/O BUFfer to X.

BUFX Create an I/O BUFfer of X bytes.

FLAG>X Copy FLAGs 00 through 07 to X.

FLAG<>X Exchange FLAGs 00 through 07 with X.

HS HandShake mode active.

IA Input data to the ALPHA register.

IAX Input data to ALPHA by X.

IBUFX Input to the I/O BUFfer by X.

ICX Input Control line (ICTL) to X.

IDX Input Data to X.

INTMASK Set INTerrupt MASK by X.

INTOFF INTerrupt OFF.

INTON INTerrupt ON.

INVON INVert mode ON.

INVOFF INVert mode OFF.

MATCHI MATCH the Input lines with a given bit pattern.

NOTX Do a 1's complement (NOT) of X.

OA Output ALPHA to the output lines.

OBUFX Output from the I/O BUFfer by X.

OCX Output Control from X.

ODX Output Data from X.

ORXY Do an inclusive OR of X and Y.

PT>X Return the buffer PoinTer to X.

RATE Count events in the time interval X.

STROBE STROBE mode active.

TIMEI TIME the Input signal.

TIMEO Toggle at predetermined TIMEs the Output signal.

TX>BUF Store the Time in X into the I/O BUFfer.

VIEWIN VIEW the input lines.

WAITIX WAIT for the Input lines to change state X times.

X>A Shift X into ALPHA from the right.

X>BUF Write X to BUFfer at pointer.

X>FLAG Copy X to FLAGs 00 through 07.

X>PT Use X to set the I/O buffer PoinTer.