Heathkit®

MICROPROCESSOR TRAINER

Model ETW-3800
with ETC-6811 CPU MODULE

User's Manual

595-4170-04

Heathkit®
Heathkit®
Heathkit®
Heathkit®
Heathkit®
Heathkit®

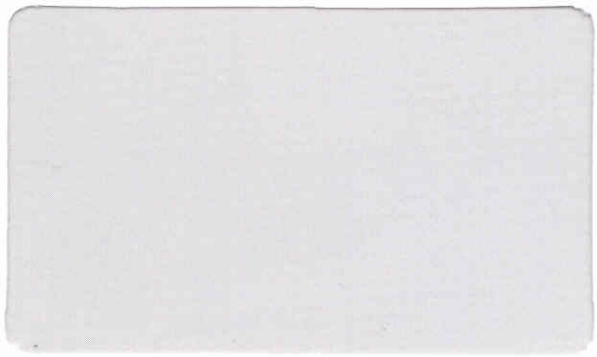
HEATH COMPANY PHONE DIRECTORY

The following telephone numbers are direct lines to the departments listed:

Kit orders and delivery information **616-982-3411**
Credit **616-982-3561**
Replacement Parts **616-982-3571**

Technical Assistance Phone Numbers
(8:00 A.M. to 4:30 P.M. Eastern Time, Weekdays Only)

Education Products **616-982-3980**
Amateur Radio **616-982-3296**
Test Equipment, Weather Instruments,
Clocks **616-982-3315**
Television **616-982-3307**
Home Products, Stereo, Security, Telephone,
Marine, Automotive **616-982-3496**
Computer — Hardware **616-982-3309**



YOUR HEATHKIT 1 YEAR LIMITED WARRANTY

Consumer Protection Plan for Heathkit Consumer Products

Welcome to the Heath family. We believe you will enjoy assembling your kit and will be pleased with its performance. Please read this Consumer Protection Plan carefully. It is a "LIMITED WARRANTY" as defined in the U.S. Consumer Product Warranty and Federal Trade Commission Improvement Act. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Heath's Responsibility

PARTS — Replacements for factory defective parts will be supplied free for 1 year from date of purchase. Replacement parts are warranted for the remaining portion of the original warranty period. You can obtain warranty parts direct from Heath Company by writing or telephoning us at (616) 982-3571. And we will pay shipping charges to get those parts to you . . . anywhere in the world.

SERVICE LABOR — For a period of 1 year from the date of purchase, any malfunction caused by defective parts or materials will be corrected at no charge to you. You must deliver the unit at your expense to the Heath factory, any Heath/Zenith Computers and Electronics center (units of Veritechnology Electronics Corporation), or any of our authorized overseas distributors.

TECHNICAL CONSULTATION — You will receive free consultation on any problem you might encounter in the assembly or use of our Heathkit product. Just drop us a line or give us a call. Sorry, we cannot accept collect calls.

NOT COVERED — The correction of assembly errors, adjustments, calibration, and damage due to misuse, abuse, or negligence are not covered by the warranty. Use of corrosive solder and/or the unauthorized modification of the product or of any furnished component will void this warranty in its entirely. This warranty does not include reimbursement for inconvenience, loss of use, customer assembly, set-up time, or unauthorized service.

This warranty covers only Heath products and is not extended to other equipment or components that a customer uses in conjunction with our products.

SUCH REPAIR AND REPLACEMENT SHALL BE THE SOLE REMEDY OF THE CUSTOMER AND THERE SHALL BE NO LIABILITY ON THE PART OF HEATH FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOSS OF BUSINESS OR PROFITS, WHETHER OR NOT FORESEEABLE.

Some states do not allow the exlusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Owner's Responsibility

EFFECTIVE WARRANTY DATE — Warranty begins on the date of first consumer purchase. You must supply a copy of your proof of purchase when you request warranty service or parts.

ASSEMBLY — Before seeking warranty service, you should complete the assembly by carefully following the manual instructions. Heathkit service agencies cannot complete assembly and adjustments that are customer's resonsibility.

ACCESSORY EQUIPMENT — Performance malfunctions involving other non-Heath accessory equipment (antennas, audio components, computer peripherals and software, etc.) are not covered by this warranty and are the owner's responsibility.

SHIPPING UNITS — Follow the packing instructions published in the assembly manuals. Damage due to inadequate packing cannot be repaired under warranty.

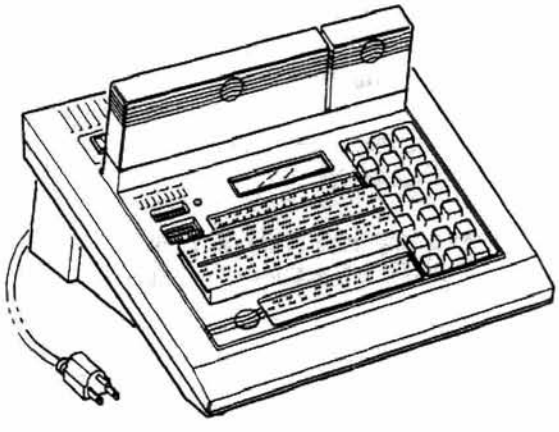
If you are not satisfied with our service (warranty or otherwise) or our products, write to our Director of Customer Service, Heath Company, Benton Harbor MI 49022. He will make certain your problems receive immediate, personal attention.

10-1-87

MICROPROCESSOR TRAINER

Model ETW-3800
with ETC-6811 CPU MODULE

User's Manual 595-4170-04



HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only computers certified to comply with the Class B limits may be attached to this equipment. Operation with non-certified computers is likely to result in interference to radio and TV reception.

This equipment uses radio frequency energy for its operation; and if it is not installed and used properly, that is, in strict accordance with the instruction manual, it may cause interference to radio and television reception. It has been type tested and found to comply with the RF emission limits for a Class B computing device which is intended to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio and television reception, which you can determine by turning the equipment off and on, try to correct the interference by one or more of the following measures:

- Move the computing device away from the receiver being interfered with.
- Relocate the computing device with respect to the receiver.
- Reorient the receiving antenna.
- Plug the computing device into a different AC outlet so that the computing device and receiver are on different branch circuits.
- Disconnect and remove any experimental connecting leads and/or I/O cables that are not being used. (Unterminated leads and/or I/O cables are a potential source of high RF emission levels.)
- Unplug and remove any experimental circuits that are not being used.
- Obtain results of the experiment as quickly as possible, then turn the computing device off.
- Be certain that the computing devices are plugged into grounded outlet receptacles. (Avoid using AC cheater plugs. Lifting of the power cord ground may increase RF emission levels and may also present a lethal shock hazard to the user.)

NOTE: In order to meet Class B emission limits, the user must comply with the following requirements:

- The I/O cables that interconnect between this computer and any peripheral (such as a printer, modem, etc.) must be shielded.
- The line (power) cord shipped with the computer is shielded. If you replace the line cord, be sure to use only a shielded line cord.

If you need additional help, consult your dealer or ask for assistance from the manufacturer. Customer service information is on the inside back cover of this Manual or on an insert sheet supplied with this equipment. You may also find the following booklet helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the US Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

Table of Contents

INTRODUCTION

The Model ETW-3800 Microprocessor Trainer is a practical learning tool specially designed to help you understand 8-bit microprocessor operation, programming, and applications. A unique feature of this Microprocessor Trainer is separate CPU and memory cartridges. By simply replacing a cartridge you can inexpensively study and experiment with different microprocessors using just one educational trainer.

The included ETC-6811 CPU Module cartridge contains the Motorola 68HC11 microprocessor, RAM, ROM and buffer circuitry. The 68HC11 is a highly integrated 8-bit processor in the 6800 family that can also be used in an expanded mode to look like a 6800 microprocessor. For increased flexibility and expanded Trainer operation, the cartridge can be easily replaced with our 8-bit microprocessor cartridges.

A replaceable ETC-128 Memory Module cartridge is also available as a separate program storage option. Use this cartridge to save programs for downloading at a later time. You will save time over manually re-entering programs and you will eliminate the possibility of entry errors. The Trainer maintains a full 16-bit data path to the memory cartridge so that it can be configured for 16K by 8, 8K by 16, or up to 64K by 16 bits of data. This also enables the memory cartridge to be shared with the Heathkit/Zenith 16-bit microprocessor trainer.

Other hardware features include:

- 20-character by 2-line liquid crystal display (LCD).
- Hexadecimal keypad with added function keys.
- Display and status light emitting diodes (LEDs).
- Binary logic switches.
- Input and output data bus ports.
- Logic probe provides visual and audible indications.
- 1 Hz and 60 Hz square wave outputs.
- Large breadboard for building experimental circuits.
- Connector blocks for solderless connections between parts and wires.
- Power supply outputs allow access to +5 VDC, +12 VDC, and −12 VDC voltages.
- Standby power to CPU RAM for saving programs in memory while modifying circuitry with the Trainer turned off.
- RS-232 interface for use with computers and terminals.
- Expansion connector for the ET/ETW-3567 Heathkit/Zenith Accessory Backpack.

The ETW-3800 Trainer provides a help key, help menus, and easy-to-understand prompts to guide you through all operations. The Trainer can be operated from its hexadecimal keyboard or from a terminal or a computer using terminal emulation software. Keyboard commands let you examine and modify memory contents, as well as

insert data; examine and modify registers; copy blocks of memory to another memory location or to a Memory Module cartridge; and initialize blocks of memory. Other keyboard commands let you access 16 software break points, 16 watch locations and seven watch registers. More commands enable you to load and save files, receive Motorola SHEX files, single step through programs, and run programs in real time with or without breakpoints.

The ETW-3800 Microprocessor Trainer is recommended for conducting experiments contained in Heathkit/Zenith Educational Systems learning programs. Associated courses include separate Memory Modules preprogrammed with experiments.

SPECIFICATIONS

CPU Module	Separate plug-in cartridge containing an 8-bit Motorola 68HC11 microprocessor, 16K × 8 Monitor EPROM and 28K × 8 RAM.
Display	20 × 2 line LCD with definable characters.
Keyboard	21-key hexadecimal keypad including RESET and NMI.
Output Port	8-bit output latch in the CPU address space.
Input Port	8-bit input latch in the CPU address space.
Logic Switches	Eight miniature on/off binary switches in a dual in-line package with separate input terminals.
Logic Indicators	Eight green LEDs with separate input terminals.

LOGIC PROBE

Logic High	≥2.0 volts.
Logic Low	≤0.8 volts.
Minimum Glitch Recognition	50 MHz.
Visible Output	Red and green LEDs.
Audible Output with Disable	1.8 kHz, 2.2 kHz.

POWER SUPPLIES

Output Voltages	+5 volts DC at 0.5 amp. +12 volts DC at 0.1 amp. –12 volts DC at 0.1 amp.

Load Regulation	5%.
Standby Supply (to CPU RAM only)	4.7 volts from 0.1F capacitor for RAM locations 0000H - 67FFH.

GENERAL

Power Requirements	105 to 130 volts AC, 60 Hz, 20 watts maximum.
Fuse	1/4-ampere slow-blow.
Dimensions	4-1/4" H × 12" W × 11-1/4" D.
Weight	5 lbs.

The Heath Company reserves the right to discontinue products and to change specifications at any time without incurring any obligation to incorporate new features in products previously sold.

STARTUP

This section of the Manual tells you how to set up your Microprocessor Trainer, shows you how to power up the Trainer and check out basic functions, and lists operating precautions.

SETUP

Your ETW-3800 Microprocessor Trainer requires the installation of the CPU Module to be operational. When the Memory Module is installed, your Trainer becomes completely operational.

1. Remove the static protection clip from the ETC-6811 CPU Module and install the Module into the two left connectors. Make sure the lettering on the cartridge is facing you. This Module is keyed to go into the connectors of the Trainer one way only. Do not force the Module in backwards.

   To permanently install the Module, first remove the eight screws that secure the cabinet top to the cabinet bottom. Then insert the CPU Module into the connectors and use the included #6BT × 1.5" screw to anchor the Module to the Trainer through the circuit board. Do not overtighten the screw. When finished, reassemble the cabinet.

2. The ETC-128 Memory Module occupies the right connector of the Trainer. Install this optional cartridge only when you save or download programs. Make sure the lettering on the cartridge is facing you. This Module is also keyed so it fits the connector one way only.

Refer to the Appendix for the proper procedure when installing, removing, and storing cartridges.

POWERING UP

1. Make sure the POWER switch is off (push down on the left side of the switch).

2. Plug the line cord of the Trainer into the proper AC outlet.

NOTE: If you do not obtain the proper results in the following steps, push the POWER switch to off and unplug the line cord. Then refer to the "In Case of Difficulty" section to correct the problem before you continue.

3. Push the POWER switch to on. The POWER LED should light. Also, the Logic Probe transducer beeps and the two status LEDs, CPU and memory, should light. The LCD display reads:

   ```
   Heath/Zenith  68HC11
   Educational MPU V1.0
   ```

4. Adjust the CONTRAST control for best display viewing. This screwdriver adjustment is accessible through the cabinet at the left of the LCD display.

OPERATING PRECAUTIONS

DO	DO NOT
DO turn the power off when inserting or removing the CPU cartridge. DO turn the power off before inserting or removing backpack boards from the ET-3567 Accessory Backpack when used. DO unplug the program cartridge when you are not loading or saving a program. This will prevent accidental writes to the program address range by your programs. DO place the static protection clip on the CPU cartridge when not plugged into the Trainer.	DO NOT remove the ground lead from the line cord. DO NOT use an ungrounded AC power line adapter. DO NOT replace the fuse with any type or value other than the one specified on the fuse label. DO NOT connect any external power sources to any of the Trainer breadboard points. DO NOT connect voltages above +5 volts or any negative voltages to the inputs or outputs of the Trainer, except the analog inputs. The inputs and outputs of the Trainer are protected with series resistors, but prolonged overvoltage conditions will shorten the life of the ICs in the Trainer. DO NOT insert larger than #20 (0.032") solid wire or component leads in the connectors of the Trainer. DO NOT expose your Trainer to moisture. DO NOT plug any Backpack Boards designed for the ETW-3600, ETW-3700, or ETW-5000 Trainers into an ETW-3567 Accessory Backpack attached to the ETW-3800 Trainer.

OPERATION

This section of the Manual describes the operation of your Trainer, explains the keyboard commands, describes how to enter programs, contains sample programs, and explains about memory I/O and the use of the jump and interrupt tables. Alternate line voltage wiring is also shown.

TRAINER

Pictorial 1 gives a brief description of switches, LEDs and connectors. Also refer to the Pictorial while reading the following paragraphs.

STATUS LEDS

These four LEDs provide a visual indication of the Trainer's operation. From right to left, the first LED is not used. The second LED lights when the external Memory Module is enabled. The third LED lights when the lower 4K of on-board RAM is disabled. The fourth LED lights when the microprocessor is operating properly. This LED flashes off when the Reset key is pressed.

LOGIC PROBE

Two LEDs and a transducer make up the Logic Probe. The red LED lights for a logic high and the green LED lights for a logic low. The Logic Probe is accessible through the lower signal connector block. High and low logic levels also produce an audible response from the nearby transducer. The audible output can be disabled with a jumper to ground from the LPAUDC line on the lower signal connector block.

LOWER SIGNAL CONNECTOR BLOCK

This signal block allows you access to input and output ports, microprocessor control lines, +5 VDC, ±12 VDC, logic probe, and 1 Hz and 60 Hz square waves through connectors. The lines are labeled at their terminal points and are color coded. Red areas are for output only, blue areas are input only, and green areas are bi-directional. The signal block consists of two rows of fifty separate connectors with each top connector internally connected to the bottom connector. The connectors will accept up to #20 (0.032") solid wire and most common component leads. All the signal lines are protected by series resistors. The following lines are available at the signal block.

LPIN	Logic Probe Input.
LPAUDC	Logic Probe Audio Disable. Disable by connecting to ground.
GND	Digital Ground.
+5V	+5 Volt DC Output.
AGND	Analog Ground.

12V	–12 volt DC Output.
+12V	+12 volt DC Output.
AI7-AI0	Analog Inputs. Lines to CPU A/D, 8 bit ±1 volt full range.
60Hz	Output Compare 2. 60 Hz square wave output.
1Hz	Output Compare 3. 1 Hz square wave output.
OC4	Output Compare 4.
OC5	Output Compare 5.
PAI	Pulse Accumulator Input.
IC1-IC3	Input Capture 1-3.
MOUT	SPI Data Output.
MIN	SPI Data Input.
SCLK	SPI Clock.
$\overline{SS}$	SPI Slave Select.
IPL	Input Port Latch. Latches data on falling edge.
IP7-IP0	Input Data Port. Located at B280H.
OP7-OP0	Output Data Port. Located at B240H.
GND	Digital Ground.

UPPER SIGNAL CONNECTOR BLOCK

This signal block enables you to access most of the CPU Module's microprocessor lines through connectors. The lines are labeled at their terminal points and are color coded. Red areas are for output only, blue areas are input only, and green areas are bi-directional. The signal block consists of two rows of fifty separate connectors with each top connector internally connected to the bottom connector. The connectors will accept up to #20 (0.032") solid wire and most common component leads. All the signal lines are protected by series resistors. The following lines are available at the signal block.

D0-D7	CPU Data Bus lines.
32KEN	Lower 20K RAM Disable. When pulled low this line disables lower 20K of on-board RAM and makes EMS memory select line active for 0000H-4FFFH.
$\overline{READ}$	Memory, I/O active-low read line.
$\overline{WRITE}$	Memory, I/O active-low write line.
$\overline{XIRQ}$	Non-Maskable Interrupt.
$\overline{RESET}$	Reset input.
AS	Address Strobe. Low byte of address bus latched on falling edge of this line.
LIR	Inverted CPU LIR line. Indicates an opcode fetch.
A15-A0	CPU Address Bus. Low byte is already latched.
$\overline{IRQ}$	CPU Interrupt Line.
E	CPU E Clock.
R/W	CPU R/W Line.
I/O 0	Decoded I/O Line. Address is B300H-B33FH.
I/O 1	Decoded I/O Line. Address is B340H-B37FH.
I/O 2	Decoded I/O Line. Address is B380H-B3BFH.
I/O 3	Decoded I/O Line. Address is B3C0H-B3FFH.
COP	CPU Computer Operating Properly Output.
EMS	External Memory Select. Active when 32KEN is pulled low.

MEMORY MODULE CONNECTOR

The Memory Module connector accepts optional Memory Module cartridges designed for this 40-pin connector. A full 16-bit data path is implemented to the Memory Module so that it can be configured for 16K × 8, 8K × 16, or up to 64K × 16 bits of data. In addition, other EPROM Memory Modules, preprogrammed with experiments, accompany selected Heathkit/Zenith Educational microprocessor courses.

CPU MODULE CONNECTORS

The CPU Module connectors accept CPU Module cartridges designed for these two dual 62-pin connectors. The interface for the separate, removable ETC-6811 CPU Module is designed so that other 8-bit microprocessor modules can be used with the Trainer. (See Appendix for the proper cartridge removal/insertion procedure.) All microprocessor-specific control lines are routed to two signal connector blocks located above and below the large Breadboard Block. The connector blocks are labeled according to the corresponding microprocessor line. Overlays are included with other CPU Modules that define the microprocessor lines terminated at the connector blocks.

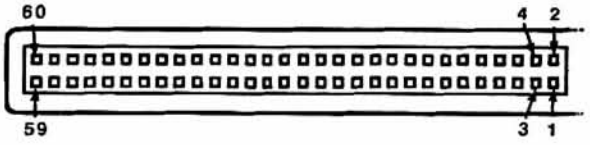
RS-232 INTERFACE

The RS-232 port is located on the right side of the Trainer. It allows the Trainer to be controlled from a terminal or a computer emulating a terminal. All the Trainer keyboard commands are available through the RS-232 port. You can also use the port to download programs written on a computer. The following signals are available at the connectors.

1- Chassis Ground.
2 - RXD (Receive Data).
3 - TXD (Transmit Data).

ACCESSORY BACKPACK CONNECTOR

A rear panel connector is provided for the ET/W-3567 Accessory Backpack. Available at the 60-pin connector are the microprocessor's data bus lines, the lower six address lines, control lines, four I/O select lines, and several other I/O lines. All the signal lines are protected by series resistors. By using the Accessory Backpack, you can connect prewired and experimental circuit boards to the Trainer. The following lines are available at the connector.



1 - Analog Input 0.
2 - Analog Input 1.
3 - Output Compare 3 (1 Hz).
4 - Output Compare 2 (60 Hz).
5 - Output Compare 5.
6 - Output Compare 4.
7 - Input Capture 1.
8 - Pulse Accumulator Input.
9 - Ground.
10 - Input Capture 2.
11 - Ground.
12 - SPI Data Out.
13 - Ground.
14 - SPI Data In.
15 - Ground.
16 - SPI Clock.
17 - Ground.
18 - SPI Slave Select.
19 - Ground.
20 - CPU Address 5.
21 - Ground.
22 - CPU Address 4.
23 - Ground.
24 - CPU Address 3.
25 - Ground.
26 - CPU Address 2.
27 - Ground.
28 - CPU Address 1.
29 - Ground.
30 - CPU Address 0.
31 - Ground.
32 - IRQ.
33 - Ground.
34 - E Clock.
35 - Ground.
36 - CPU R/W Line.
37 - Ground.
38 - General I/O 0.
39 - Ground.
40 - General I/O 1.
41 - Ground.
42 - General I/O 0.
43 - Ground.
44 - General I/O 3.
45 - Ground.
46 - CPU Data 0.
47 - Ground.
48 - CPU Data 1.
49 - Ground.
50 - CPU Data 2.
51 - Ground.
52 - CPU Data 3.
53 - Ground.
54 - CPU Data 4.
55 - Ground.
56 - CPU Data 5.
57 - Ground.
58 - CPU Data 6.
59 - Ground.
60 - CPU Data 7.

KEYBOARD

The keyboard allows you to quickly enter commands and data to the microprocessor (see Pictorial 2). After you press a command key, a brief prompt helps you complete the entry. All of the keyboard commands are also available through the RS-232 port. Some commands can be accessed with subroutine calls from a program.

After you press the POWER switch, the Trainer turns on in the Select Operation mode and the LCD display shows:

```
Heath/Zenith  68HC11
Educational MPU V1.0
```

The Trainer will interpret the next key entry as a command. After you enter a command, enter data using the 0 to F keys. The following paragraphs discuss the various commands.

PICTORIAL 2

Return RPO Help — Press the RETURN key to exit a function (such as Examine Memory, Set Breakpoint, or Single Step) and to save all data contained in the CPU's registers. If you use the NMI or RESET key to exit a function, the register values will be replaced by the Monitor ROM's default values. In the Examine Memory, Examine Register, and Insert modes, use the RETURN key to move to the previous memory or register location and use the RPO-HELP key to exit the function and still keep all the data in the registers.

List + — Press the LIST key in Break to list all the breakpoints. Likewise, you can list all the Watch Registers and the Watch Locations (addresses) by pressing the LIST key while in each mode.

RPO Help — Press the HELP key to display a help message. Each time the key is pressed more help messages appear. These short, concise messages show you the appropriate keys to type and their meaning. Help messages are available for all functions except Examine Memory, Examine Register, and Insert. The RS-232 equivalent for the HELP key is "?".

RS-232 D — Press the RS-232 key to transfer control from the keyboard of the Trainer to the RS-232 serial port. When you press this key, all keyboard commands can be executed from a connected terminal or a computer using terminal emulation software. Select commands by entering the single key representing that command. All lower case letters are converted to upper case, except for print strings.

PRESS	DISPLAY READS
RS-232	RS-232 control Baud=9600

On your terminal or computer screen a ready prompt will appear:

```
>
```

At this point, press a key that corresponds to the following Trainer commands:

PRESS	FOR COMMAND
0	Single Step
1	Examine Memory
2	Examine Registers
3	Go
4	Move Block
5	Initialize Block
6	Receive SHEX File
7	Set Breakpoint
8	Set Watch Register
9	Set Watch Point
A	Load File
B	Save File
C	Duplicate File
D	Select RS232
E	Select Baud
F	Insert
L	Look Addresses
X	Display Addresses

Any other key entry will show a "?" on the screen. When you enter a "?" from the terminal or computer keyboard, a help menu will appear.

Baud E

Press the BAUD key to enter a new data transfer rate for the RS-232 port. When you first turn on the Trainer, the RS-232 port is automatically set for 9600 baud and the Baud Rate Control Register (SCI) is set to 1 start bit, 8 data bits, and 1 stop bit.

PRESS	DISPLAY READS
Baud	Type in new Baud: 0000

Insert F

Press the INSERT key to change the contents of a memory location. This mode works almost the same as Examine Memory. The difference occurs when you enter a new value from the keyboard into RAM. The values from the current address to 5FFFH are shifted up one byte in memory. The byte previously at 5FFFH is lost. Data in memory locations 6000H - FFFFH is unaffected. Do not use the Insert Memory mode above memory location 6000H; this can produce unpredictable results.

PRESS	DISPLAY READS
Insert	F Look Address: 0000H
0000	Look Address: 0000H 0000: 40 @

Load A

Press the LOAD key to transfer a file stored in the Memory Module to RAM located in the CPU Module. When you select this command, the Trainer asks for a file digit (0-F) that corresponds to the file digit the program was saved under. After you enter a file digit, the display shows the address it will be loaded at. The file is then copied, checksums are compared, and the resulting checksum is displayed.

PRESS	DISPLAY READS
Load	A File digit?
Help	File Digit? type hex digit 0-F

If the file cannot be read, the display will read:

File digit 0-F	File digit ? Error: Load Error

If the file is found, the display will read:

File digit 0-F	Copying to 0000H Verifying Checksum

Save B

Press the SAVE file key to transfer a file to the Memory Module from the CPU Module RAM. The Trainer asks for a start address, number of bytes in the file, and a file digit (0-F). The file is then copied from RAM to the cartridge and a checksum is formed and stored on the cartridge with the start address and number of bytes.

PRESS	DISPLAY READS
Save	B Source: 0000 H
Enter start address	Source: 0000 H # Bytes: 0000 H
Enter number of bytes	# Bytes: 0010 H File Digit ?
Help	File digit ? type hex digit 0-F
Enter file digit	Copying... Verifing Checksum
	Checksum= 07F8

Dup C

Press the DUP (duplicate) key to copy a file from one Memory Module to another. Help prompts take you through each step. First insert the cartridge to copy from and then copy your file into RAM. Next, remove the cartridge and insert the Memory Module you want to copy to. Repeat these steps three more times. In the last step, you copy your file from RAM to the new cartridge.

PRESS	DISPLAY READS
Dup	Insert source press a key.
Press any key.	Press a key. Copying to RAM...
	Insert destination press a key.
	press a key. Copying to Cartridge!
Repeat above steps 3 times.	
	Dup complete: 491F

Break 7

Press the BREAK key to set a breakpoint in memory. A breakpoint stops program execution and transfers control back to the keypad or terminal. At this point you can examine memory, examine registers, set another breakpoint, or GO again. The Trainer asks for an address where you want to set a breakpoint. If you press the LIST key the currently set breakpoints are listed. After you specify a hexadecimal address, the breakpoint is added to the breakpoint table. If the address is already in the table, the previous one will be removed. You can set up to 16 different breakpoints at a time. If you add a breakpoint that is already in the table, the old one will be deleted. Refer to Page 27 for a more detailed discussion of breakpoints.

W Reg 8

Press the W REG key to break a program when a register reaches a desired value. Press the 0 key to scroll through a list of registers. Press the 8 key to select the register to watch. Press the LIST key to list the currently set watch registers. After selecting a register to watch, select a value to watch for.

PRESS	DISPLAY READS
Brk	Watch Register: CCR?
Help	CCR? list,8=select,0=next

W Loc 9

Press the W LOC key to break a program when an address location reaches a desired value. This mode is similar to Break.

PRESS	DISPLAY READS
W Loc	Watch Address: 0100H Watch Value 00 H

M Blk 4

Press the M BLK key to move a block of data in memory to another location in memory. Following the prompts, enter the starting address, number of bytes in the block to move and the destination address. Before the data is moved, the Trainer checks the destination range to make sure that a write does not destroy a source byte. If a source byte already resides in that range, the block of data will copy from the high memory address back down to the low memory address. Keys 0-F are shifted in from the right for each field. RETURN moves to the next field and NMI quits the operation.

PRESS	DISPLAY READS
M Blk	4 Source 0000H
	Source ____H # Bytes ___H Dest ___H
	Copy complete >

I Blk 5

Press the I BLK key to set a block of memory to a desired value. First enter a start address and then an end address. Next, set the block of memory equal to a value. The hexadecimal value of the block is displayed along with the ASCII version. Use the 0 to F keys to enter all values. Press the NMI key to exit this mode.

PRESS	DISPLAY READS
I Brk	Start Address ____H End Address ____H
	End Address Byte = 42 H B

Down 6

Press the DOWN key to download a Motorola SHEX file for storing in RAM. You may specify an offset for easier relocating of the file to another location in RAM or when copying a program into a Memory Module after it has been downloaded into RAM.

Exm Mem 1 Pressing the EXM MEM key in the select operation mode lets you examine the contents of any memory location. The Trainer first asks for the starting address. Type in the address in hexadecimal form. The display then shows the value of the byte at that address in hex (and ASCII if displayable). To change the current byte, simply type any key 0 to F and each value will be shifted in on the low byte value. After you press two keys, the address is automatically stepped to the next address. To move to the next address, press the plus (+) key. To move to the previous address, press the minus (–) key. To quit, press the RPO key.

PRESS	DISPLAY READS
Exm Mem	1 Look Address: 0000H
0000	Look Address: 0000H 0000: 41 A

Exm Reg 2 Pressing the EXM REG key in the select operation mode lets you examine and modify the CPU registers. Enter a desired register value by pressing the 0 to F keys, which are shifted into the low order value of that register. Press the plus (+) key to display the contents of the next register. Press the minus (–) key to display the contents of the previous register. Press the RPO key to end register examination.

PRESS	DISPLAY READS
Exm Reg	2 CCR= 00__ H
+	CCR= 00 H ACCB= 00__ H
+	ACCB= 00 H ACCA= 00__ H
+	ACCA= 00 H IX= 0000H
+	IX= 0000H IY= 0000H
+	IY= 0000H PC= 0000H
+	PC= 0000H SP= 0000H
RPO	PC= 0000H SP= 0000>

Go 3 Press the GO key to run a program stored in memory. Initially, the default start address (0000H) is displayed. You then select a start address. The Trainer jumps to that address and executes the program. If it encounters a breakpoint, the program halts and automatically enters the Single Step mode. From here you can either press GO again or quit by pressing the NMI key. When GO is run from the single step mode, the program will run slower due to the extra time required to check watch points and watch registers.

PRESS	DISPLAY READS
Go	3 Go Address: 0000H

If a breakpoint is encountered the next address is displayed.

	0002 7E 00 00 JMP 0000

If a watch point is encountered the watch address or register is displayed.

	JMP 0000 Address 0100H= 55 H

SS 0 Press the SS key to single step through a program stored in memory. Enter a start address and the Trainer executes the instruction at that address. You can then single step to the next step, press GO, examine registers, examine or insert memory, or set break or watch points. Press the RETURN key to quit at any time. All registers and RAM values are saved at their present values.

PRESS	DISPLAY READS
SS	0 Step Address: 0000H
0000	Look Address: 0000H 0000: 41 A

After the instruction is executed:

	010B 86 54 LDDA #54
Help	SS ,GO, Exm Rg,Exm M Brk, Watch, NMI quit

NMI Press the NMI key to interrupt any program or operation being performed on the Trainer. This function does not alter the contents of RAM and saves all register values. If the Trainer was executing a program in real time (i.e. you press GO from the menu and

not from Single Step), then these values can be checked by the Examine Register function. If you press NMI any other time, the Monitor's register values will overwrite the current register values. In addition, the Interrupt Vector Table is not recopied to RAM , allowing you to interrupt your program and examine this table.

PRESS	DISPLAY READS
NMI	Heath/Zenith 68HC11 Educational MPU V1.0

Reset Press the RESET key to reinitialize the Trainer to its power-up state. The Interrupt Vector Table is initialized to its default values. All breakpoints, watch points, and watch registers are erased.

ENTERING PROGRAMS

Pictorial 3 shows the first two instructions of Sample Program 1 (on Page 18) and indicates the various information they contain. This information is further described in the following paragraphs.

Instruction Address: This is the location of the Op code to be executed. In order to perform an instruction, the Program Counter must contain the address that is in this column. Breakpoints are not recognized except at instruction addresses.

Instruction: This is one, two, or three bytes of data as required by the addressing mode used.

Op code: This is an information byte referred to as machine code. It indicates in hexadecimal form the operation to be performed.

Operand: This is additional hexadecimal information needed to perform the operation. It may be zero, one, or two bytes as determined by the addressing mode.

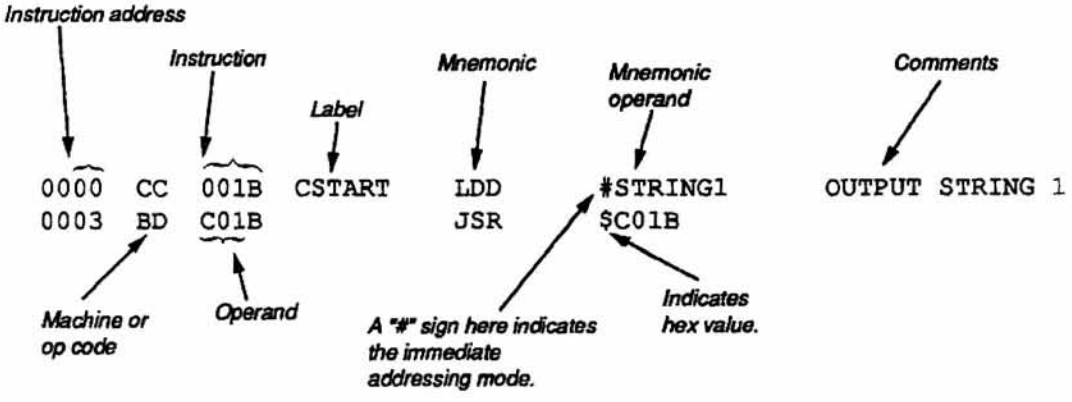
Label: This is usually a name applied to a subroutine in the program used more than once. In the sample programs, the address to be entered to begin execution is labeled "Start."

Mnemonic: This is a three-letter indication of the source instruction. A fourth letter, A or B, is added to indicate which of two accumulators is used if the instruction applies to either one.

Mnemonic operand: Again, this is additional information that is required for the operation. It may be a label, address, or data. The "$" sign indicates that the information is a hexadecimal value. The "#" sign indicates the immediate addressing op code is to be used.

Comments: This is a brief description of what is happening in the program. It makes the program easier to read but has no effect on the program.

When you load a program into the Trainer, only the one, two, or three bytes of each instruction are entered. If you make an entry error, press the – key. Pressing the – key backs you up one address at a time. Use the INSERT key to correct any entry. Pressing the RPO key returns control back to the Trainer. Or, remember where the error was made and continue entering the program. After you're finished entering the program, correct your error by examining that memory location and changing the entry.



PICTORIAL 3

Sample Programs

These sample programs will give you practice entering programs and show the use of several Monitor subroutines.

Sample Program 1

The following program demonstrates how you can use two simple subroutine calls to create and display a message on the Trainer's LCD. The subroutine located at $C01B outputs a string of characters starting at the location stored in the "D" register of the CPU until it encounters a zero. Subroutine $C027 delays the program for 1 second every time it is called. The last jump to subroutine instruction (JSR $C000) returns control back to the Monitor ROM.

```
0000  CC  0018  CSTART:  LDD  #STRING1      OUTPUT STRING 1
0003  BD  C01B           JSR  $C01B
0006  BD  C027           JSR  $C027         PAUSE
0009  CC  0027           LDD  #STRING2      OUTPUT STRING 2
000C  BD  C01B           JSR  $C01B
000F  BD  C027           JSR  $C027         PAUSE
0012  BD  C027           JSR  $C027         PAUSE
0015  BD  C000           JSR  $C000
0018  50726F67 STRING1   FCC                'Programming is'
      72616D6D
      696E6720
      6973
0026  00                 FCB  0
0027  0D         STRING2  FCB  $0D
0028  45617379           FCC                'Easy and Fun!!!'
      20616E64
      2046756E
      212121
0037  00                 FCB  $0
                         END
```

Sample Program 2

The following program is for a real time clock. It uses the 1 Hz real time interrupt available from the CPU Module's programmable timer. The program first disables the interrupt so that you can enter the current time by using the plus (+) and minus (–) keys on the Trainer. As soon as you enter the last digit of the time, the interrupt will be enabled and the clock will start. If you press the NMI key, the clock will continue to run while allowing you to use other Trainer functions.

```
0000  86  00    CSTART:  LDAA #$00          DISABLE 1HZ AND 60HZ
0002  B7  B022           STAA $B022         INTERRUPTS
0005  CC  0123           LDD  #STRING1      OUTPUT STRING 1
0008  BD  C01B           JSR  $C01B
000B  86  01             LDAA #1            GET HOURS TENS DIGIT,
                                            SET A TO MAXIMUM VALUE
000D  BD  00DA           JSR  GETTIM
0010  48                 LSLA               SHIFT TO HIGHER NIBBLE
0011  48                 LSLA
0012  48                 LSLA
0013  48                 LSLA
```

```
0014  B7  011E            STAA HOURS          STORE A HOURS
0017  81  01              CMPA #1             CHECK FOR 10,11,12
0019  26  04              BNE  HNOTTEN
001B  86  02              LDAA #2             SINCE FIRST DIGIT WAS ONE
                                              MAX DIGIT IS 2
001D  20  02              BRA  GETHOUR1
001F  86  09    HNOTTEN   LDAA #9             SINCE FIRST DIGIT WAS NOT 1
0021  BD  00DA  GETHOUR1  JSR  GETTIM
0024  BB  011E            ADDA HOURS          ADD IN HIGH DIGIT
0027  B7  011E            STAA HOURS
002A  C6  3A              LDAB #$3A           MOVE TO NEXT POSITION
002C  BD  C006            JSR  $C006
002F  86  05              LDAA #5             GET MINUTES
0031  BD  00DA            JSR  GETTIM
0034  48                  LSLA
0035  48                  LSLA
0036  48                  LSLA
0037  48                  LSLA
0038  B7  011F            STAA MINUTES        SAVE IT
003B  86  09              LDAA #9             GET LOW NIBBLE
003D  BD  00DA            JSR GETTIM
0040  BB  011F            ADDA MINUTES
0043  B7  011F            STAA MINUTES
0046  C6  3A              LDAB #$3A           MOVE TO NEXT POSITION
0048  BD  C006            JSR  $C006
004B  86  05              LDAA #5             GET SECONDS
004D  BD  00DA            JSR GETTIM
0050  48                  LSLA
0051  48                  LSLA
0052  48                  LSLA
0053  48                  LSLA
0054  B7  0120            STAA SECONDS        SAVE IT
0057  86  09              LDAA #9             GET LOW NIBBLE
0059  BD  00DA            JSR  GETTIM
005C  BB  0120            ADDA SECONDS
005F  B7  0120            STAA SECONDS

                        * CURRENT TIME IS SAVED, SET INTERRUPT VECTOR

0062  CC  007B            LDD  #T1HZ          GET VECTOR LOCATION
0065  FD  6916            STD  $6916          STORE AT OC#3 VECTOR LOCATION
0068  86  20              LDAA #$20           SET TCO3 FLAG
006A  B7  B023            STAA $B023
006D  CC  0148            LDD  #STRING3       OUTPUT STRING 3
0070  BD  C01B            JSR  $C01B

                        * START CLOCK

0073  86  20              LDAA #$20           ENABLE 1HZ INTERRUPT ONLY
0075  B7  B022            STAA $B022
0078  3E        WAIT      WAI                 WAIT FOR INTERRUPT
0079  20  FD              BRA  WAIT
```

```
                        * TIMER INTERRUPT ROUTINE

007B                    T1HZ
007B  FC  B00E          LDD  $B00E          GET CURRENT TIMER VALUE
007E  C3  E0E4          ADDD #57572         SET FOR 2 HZ
0081  FD  B01A          STD  $B01A          SET TO TIMER 3
0084  86  20            LDAA #$20           CLEAR TCO3 FLAG
0086  B7  B023          STAA $B023
0089  B6  0120          LDAA SECONDS        GET SECONDS
008C  8B  01            ADDA #1             INCREMENT IT
008E  19                DAA                 DECIMAL ADJUST IT
008F  81  60            CMPA #$60           CHECK FOR OVERFLOW
0091  2D  1E            BLT  NSECOV         IF NO OVERFLOW SAVE SECONDS ONLY
0093  B6  011F          LDAA MINUTES        GET MINUTES
0096  8B  01            ADDA #1             INCREMENT IT
0098  19                DAA                 DECIMAL ADJUST IT
0099  81  60            CMPA #$60           CHECK FOR OVERFLOW
009B  2D  10            BLT  NMINOV         IF NO OVERFLOW SAVE MINUTES AND
                                            SECONDS ONLY
009D  B6  011E          LDAA HOURS          GET HOURS
00A0  8B  01            ADDA #1             INCREMENT IT
00A2  19                DAA                 DECIMAL ADJUST IT
00A3  81  12            CMPA #$12           CHECK FOR OVERFLOW
00A5  23  02            BLS  NHOROV         IF NO OVERFLOW SAVE MINUTES ONLY
00A7  86  01            LDAA #1             SET TO 1 O'CLOCK
00A9  B7  011E  NHOROV  STAA HOURS
00AC  4F                CLRA                CLEAR MINUTES AND FALL THRU
00AD  B7  011F  NMINOV  STAA MINUTES        SAVE MINUTES AND FALL THRU
00B0  4F                CLRA                CLEAR SECONDS
00B1  B7  0120  NSECOV  STAA SECONDS        SAVE SECONDS
00B4  CC  013F          LDD  #STRING2       BACKUP TO START OF DISPLAY
00B7  BD  C01B          JSR  $C01B
00BA  F6  011E          LDAB  HOURS         GET HOURS AND OUTPUT
00BD  BD  C015          JSR $C015
00C0  C6  3A            LDAB #$3A           OUTPUT COLON
00C2  BD  C006          JSR $C006
00C5  F6  011F          LDAB MINUTES        GET MINUTES AND OUTPUT
00C8  BD  C015          JSR $C015
00CB  C6  3A            LDAB #$3A           OUTPUT COLON
00CD  BD  C006          JSR $C006
00D0  F6  0120          LDAB SECONDS        GET SECONDS AND OUTPUT
00D3  BD  C015          JSR $C015
00D6  3B                RTI                 RETURN FROM INTERRUPT
00D7  BD  C000          JSR $C000

                        * PROGRAM GETS A SINGLE DIGIT OF MAXIMUM
                        * VALUE FROM REGISTER "A" AND PASSES IT TO
                        * "GETTIM" SUBROUTINE WHEN THE MINUS (-) KEY
                        * IS PRESSED, INCREMENTS AND DISPLAYS DIGIT
                        * WHEN PLUS (+) KEY IS PRESSED.
```

```
00DA  B7  0121  GETTIM   STAA MDIGIT        SAVE MAXIMUM VALUE
00DD  5F                 CLRB               SET INITIAL TO ZERO
00DE  F7  0122           STAB CURTIM
00E1  CB  30             ADDB #$30          CONVERT TO ASCII
00E3  BD  C006           JSR  $C006
00E6  C6  08             LDAB #$08          BACK UP
00E8  BD  C006           JSR  $C006
00EB  BD  C003  REPTIM   JSR  $C003         GET A CHARACTER
00EE  C1  2D             CMPB #$2D          IF MINUS RETURN
00F0  26  OC             BNE GTNOR
00F2  F6  0122           LDAB CURTIM        WRITE OUT DIGIT
00F5  CB  30             ADDB #$30          CONVERT TO ASCII
00F7  BD  C006           JSR $C006
00FA  B6  0122           LDAA CURTIM        GET SELECTED DIGIT AND RETURN
00FD  39                 RTS
00FE  C1  2B    GTNOR    CMPB #$2B          IF PLUS INCREMENT TIME
0100  26  E9             BNE REPTIM
0102  B6  0122           LDAA CURTIM        GET CURRENT TIME
0105  4C                 INCA               INCREMENT IT
0106  B1  0121           CMPA MDIGIT        COMPARE TO MAX
0109  23  01             BLS VALOK          IF LOWER OR SAME TAKE IT
010B  4F                 CLRA               SET TO ZERO
010C  B7  0122  VALOK    STAA CURTIM        SAVE THE TIME
010F  F6  0122           LDAB CURTIM        GET IT AGAIN
0112  CB  30             ADDB #$30          CONVERT TO A DIGIT
0114  BD  C006           JSR  $C006         PRINT IT
0117  C6  08             LDAB #$08          BACK UP
0119  BD  C006           JSR  $C006
011C  20  CD             BRA  REPTIM        REPEAT UNTIL '-' TYPED
011E  00        HOURS    FCB  $0
011F  00        MINUTES  FCB  $0
0120  00        SECONDS  FCB  $0
0121  00        MDIGIT   FCB  $0
0122  00        CURTIM   FCB  $0
0123  456E7465  STRING1  FCC                'Enter Current Time:'
      72204375
      7272656E
      74205469
      6D653A
0136  0D                 FCB  $0D
0137  5F5F3A5F           FCC                '__:__:__'
      5F3A5F5F
      3A
013F  08080808  STRING2  FCB  $08,$08,$08,
      08080808           $08,$08,$08,
      0800               $08,$08,$00
```

```
0148  OD         STRING3   FCB   $0D
0149  02020202             FCC                 '   The Time is:'
      54686521
      54696D65
      2069733A
0159  OD                   FCB   $0D
015A  20202020             FCC                 '     __:__:__'
      20205F5F
      3A5F5F3A
      5F5F3A
0168  00                   FCB   00
                           END
```

The following are labels for subroutines found in this program.

T1HZ 0000007B
CSTART 00000000 E
CURTIM 00000122
GETHOUR1 00000021
GETTIM 000000DA
GTNOR 000000FE
HNOTTEN 0000001F
HOURS 0000011E
MDIGIT 00000121
MINUTES 0000011F
NHOROV 000000A9
NMINOV 000000AD
NSECOV 000000B1
REPTIM 000000EB
SECONDS 00000120
STRING1 00000123
STRING2 0000013F
STRING3 00000148
VALIK 0000010C
WAIT 00000078

Sample Program 3

This program tests the user memory and stack area. It is a simple two pass test in which a continuous count is stored in consecutive memory locations and then verified. One incrementing count and one decrementing count is used.

```
0000  CC    008C  CSTART:   LDD   #STRING1   MTEST, OUTPUT STRING 1
0003  BD    C01B            JSR   $C01B
0006  CE    0000            LDX   #0000      INITIALIZE FIRST TEST
0009  18CE  00E5            LDY   #BEGIN
000D  08          TEST1A:   INX              LOOP THROUGH STORING A
000E  8F                    XGDX
000F  18E7  00              STAB  0,Y        CONTINUOUS COUNT IN MEMORY
0012  8F                    XGDX
0013  1808                  INY
0015  188C  68FF            CPY   #$68FF
0019  26    F2              BNE   TEST1A
001B  CE    0000            LDX   #0000
001E  18CE  00E5            LDY   #BEGIN
0022  08          TEST1B    INX              VERIFY CORRECT CONTENTS OF MEMORY
0023  8F                    XGDX
0024  18E1  00              CMPB  0,Y
0027  26   56               BNE   ERROR
0029  8F                    XGDX
002A  1808                  INY
002C  188C  68FF            CPY   #$68FF
0030  26   F0               BNE   TEST1B
```

```
0032  CC    009B          LDD   #STRING2
0035  BD    C01B          JSR   $C01B
0038  CE    0000          LDX   #$0000     SAME TEST AS ABOVE EXCEPT
                                            OPPOSITE DIRECTION
003B  18CE  68FF          LDY   #$68FF
003F  08          TESTE2A INX
0040  8F                  XGDX
0041  18E7  00            STAB  0,Y
0044  8F                  XGDX
0045  1809               DEY
0047  188C  00E5          CPY   #BEGIN
004B  26    F2            BNE   TEST2A
004D  CE    00            LDX   #0000
0050  18CE  68FF          LDY   #$68FF
0054  08          TEST2B  INX
0055  8F                  XGDX
0056  18E1  00            CMPB  0,Y
0059  26   24             BNE   ERROR
005B  8F                  XGDX
005C  1809                DEY
005E  188C  00E5          CPY   #BEGIN
0062  26   F0             BNE   TEST2B
0064  CC    009B          LDD   #STRING2
0067  BD    C01B          JSR   $C01B
006A  CC    00AE          LDD   $STRING3
006D  BD    C01B          JSR   $C01B
0070  CC    00C1          LDD   #STRING4
0073  BD    C01B          JSR   $C01B
0076  BD    C027          JSR   $C027
0079  BD    C027          JSR   $C027
007C  BD    C000          JSR   $C000
007F  CC    00D4  ERROR:  LDD   #ERRSTR    MEMORY ERROR RESULTED AT
0082  BD    C01B          JSR   $C01B      THE LOCATION REGISTER Y
                                            IS POINTING TO.
0085  188F               XGDY
0087  BD    C018          JSR   $C018
008A  20    FE    LOOP:   BRA   LOOP
008C  4D656D6F    STRING1 FCC             'Memory Test...'
      72792054
      6573742E
      2E2E
009A  00                  FCB   0
009B  OD          STRING2 FCB   $0D
009C  50617373            ACC             'Pass one complete'
      206F6E65
      20636F6D
      706C6574
      65
00AD  00                  FCB   $0
00AE  0D          STRING3 FCB   $0D
```

```
00AF  50617373            FCC          "Pass two complete"
      2074776F
      20636F6D
      706C6574
      65
00C0  00                  FCB   $0
00C1  0D            STRING4 FCB   $0D
00C2  4D656D6F            FCC          "Memory check O,K.'
      72792063
      5865636B
      204F2E4B
      2E
00D3  00                  FCB   $0

00D4  4D454D4F      ERRSTR FCC         'MEMORY ERROR AT:'
      52592045
      52524F52
      2041543A
00E4  00                  FCB   $0
00E5  00            BEGIN FCB   $0
                          END
```

The following are labels for subroutines found in this program.

BEGIN 000000E5
ERROR 0000007F E
ER\STR 000000D4
LOOP 0000008A E
STRING2 0000009 E
STRING3 000000AE
STRING4 000000C1
TEST1A 0000000D E
TEST1B 00000022 E
TEST2A 0000003F E
TEST2B 00000054 E
CSTART 00000000 E
MTEST 00000000 E
STRING1 0000008C

Sample Program 4

The following program lets you convert your Trainer into a digital DC voltmeter. This program uses the A/D Converter of the CPU Module's 68HC11 microprocessor and other CPU Module circuitry to monitor and display a range of positive and negative input voltages. A 1.1 Megohm resistor in series with the voltmeter's input (the analog AI0 line) reduces the voltage input to a level acceptable by the A/D Converter.

This program is written so that the voltmeter can measure voltages ranging from +12.0 to -12.0 VDC full scale. The displayed voltage is determined by multiplying the input to the A/D Converter by the fraction 30/32. This is done by multiplying the input by 30 at address 0052 (1E hex value) and using 5 LSRDs (logical shift right instructions) at addresses 0054 to 0058 to divide by 32.

To change the voltage range of this voltmeter, you must change the series resistor and alter the program. Use the following table for the proper series resistor and program alterations. Enter the hex value of the numerator at address 0052. To get a 64 in the denominator of the fraction, change address 0059 from "NOP" to another "LSRD."

VOLTAGE FULL SCALE READING (V_{fs})	SERIES RESISTOR R_s	MULTIPLYING FRACTION
−1 to +1 VDC	NO RESISTOR	5/64
−2 to +2 VDC	100 kΩ	5/32
−5 to +5 VDC	400 kΩ	25/64
−6 to +6 VDC	500 kΩ	15/32
−9 to +9 VDC	800 kΩ	45/64
−12 to +12 VDC	1.1 MΩ	30/32
−15 to +15 VDC	1.4 MΩ	75/64

To determine other series resistors and fractions when the full scale voltage is known, use these two equations:

$R_s = (50 \text{ kohm} \times V_{fs}) - 100 \text{ kohm}$

$\text{FRACTION} = V_{fs} \times 10/256$

```
0000  86  00    CSTART:   LDAA #$00          DISABLE 1HZ AND 60HZ INTERRUPTS
0002  B7  B022            STAA $B022
0005  CC  0064            LDD  #STRING1      OUTPUT STRING 1
0008  BD  C01B            JSR  $C01B

                        * CURRENT TIME IS SAVED, SET INTERRUPT VECTOR

000B  CC  002E            LDD  #T01HZ        GET VECTOR LOCATION
000E  FD  6913            STD  $6913         STORE AT OC# VECTOR LOCATION
0011  86  10              LDAA #$10          SET TCO4 FLAG
0013  B7  B023            STAA $B023
0016  86  20              LDAA #$20          SET CONTINUOUS ON CHANNEL 1
0018  B7  B030            STAA $B030
001B  B6  B039            LDAA $B039         GET CONFIG REGISTER
001E  8A  C0              ORAA #$C0          TURN A/D ON, R-C CLOCK
0020  B7  B039            STAA $B039
```

```
                        * START CLOCK

0023  B6  B022          LDAA $B022          GET CURRENT INTERRUPT MASK
0026  8A  10            ORAA #$10           ENABLE TIMER 4
0028  B7  B022          STAA $B022
002B  3E        WAIT    WAI                 WAIT FOR INTERRUPT
002C  20  FD            BRA  WAIT

                        * TIMER INTERRUPT ROUTINE

002E  FC  B00E  T01HZ   LDD  $B00E          GET CURRENT TIMER VALUE
0031  C3  7064          ADDD #28772         SET FOR 0.5 HZ
0034  FD  B01C          STD  $B01C          SET TIMER TO 4
0037  86  10            LDAA #$10           CLEAR TCO4 FLAG
0039  B7  B023          STAA $B023
003C  CC  0097          LDD  #STRING2       BACKUP TO START OF DISPLAY
003F  BD  C01B          JSR  $C01B
0042  B6  B031          LDAA $B031          GET A/D VALUE
0045  80  80            SUBA #$80
0047  4D                TSTA
0048  2D  14            BLT SIGNPOS         JUMP IF VOLTAGE IS POSITIVE
004A  C6  2D            LDAB #$2D           OUTPUT MINUS SIGN
004C  36        OUTVAL  PSHA                SAVE VALUE
004D  BD  C006          JSR $C006           OUTPUT PLUS OR MINUS SIGN
0050  32                PULA                GET VALUE
0051  C6  1E            LDAB #30            MULTIPLY BY 30
0053  3D                MUL
0054  04                LSRD                DIVIDE BY 32
0055  04                LSRD
0056  04                LSRD
0057  04                LSRD
0058  04                LSRD
0059  01                NOP                 REPLACE WITH "LSRD" IF
                                            NEED TO DIVIDE BY 64
005A  BD  C02D          JSR $C02D           PRINT IN DECIMAL
005D  3B                RTI                 RETURN FOR INTERRUPT

005E  43        SIGNPOS COMA                SINCE POSITIVE VOLTAGE COMPLIMENT IT
005F  4C                INCA
0060  C6  20            LDAB #$20           OUTPUT SPACE
0062  20  E8            BRA  OUTVAL
0064  4554572D STRING1  FCC                 'ETW-3800 Voltmeter '
      33383030
      20566F6C
      746D6574
      6572
0077  0D                FCB  $0D
0078  205F5F5F          FCC                 ' _____ tenthsVolts'
      5F5F5F20
      73656E74
      6873566F
      6C7473
```

```
008A  08080808            FCB  $08,$08,$08,
      08080808                 $08,$08,$08,
      08080808                 $08,$08,$08,
      00                       $08,$08,$08,
                               $00
0097  08080808  STRING2   FCB  $08,$08,$08,
      080800                   $08,$08,$08,
                               $00
009E  00                 FCB  00
```

The following are labels for subroutines found in this program.

T01HZ 0000002E
CSTART 00000000 E
OUTVAL 0000004C
SIGNPOS 0000005E
STRING1 00000064
STRING2 00000097
WAIT 0000002B

MEMORY ORGANIZATION

A memory map of the CPU Module is located in the Appendix. The major sections shown in the memory map are:

User code area — Available for user purposes.

User data and stack area — Used to hold the stack of the user program and Monitor. The top of the stack is initialized to the top of user RAM and builds downward. Both Monitor and user programs use the stack, so it is advisable not to move the stack. It is best to maintain at least 400H (1024 decimal) bytes between the top of any data area and the top of the stack, to allow plenty of room for user programs. This will enable you to make use of calls/returns and pushes/pops without being concerned about stack space.

Monitor data and stack area — Used by the Monitor to hold important system variables. Altering these variables may cause unpredictable results.

Memory module area — When the cartridge is installed, the addresses 7000H to AFFFH are available for storing user programs.

Internal I/O area — This is the location of the internal I/O registers for the built-in timer and A/D converter peripherals.

Keyboard area — In this area are three addresses corresponding to separate columns of the keyboard, one address per column.

Output port — Locations B240H to B27FH are used to latch values in the output port.

Input port — Locations B280H to B2BFH are used to read values from the input port.

LCD registers — Used to hold the liquid crystal display commands and addresses of display registers.

General I/O areas — Available for user purposes.

Unused — Available for user purposes.

Monitor ROM area — Contains the Monitor routine and several general purpose routines. The ROM is addressed beginning at C000H and ends at FFFFH.

INTERRUPTS

In order to maintain full control of your CPU, you must have a way to halt the execution of a program to service an internal CPU function or allow an external peripheral access to the CPU. Generally speaking, an interrupt is a temporary break in the normal execution of a program. When the CPU encounters an interrupt, it jumps to the area in memory that holds the subroutine to service the interrupt.

The starting addresses for all interrupt routines are stored in the CPU Module's Monitor ROM at addresses FFD6H to FFFDH. This dedicated area of memory is called the Interrupt Pointer Table (see Appendix). This table points to interrupt service routines stored in RAM addresses starting at 6900H. By changing the address that the interrupt vector uses, you can program your own interrupt service routines. Interrupts are not used by the Monitor to jump to a stop instruction in the ROM and you cannot modify the Reset Vector.

To use your interrupt service routine instead of the Monitor's, place the starting address of your service routine at the appropriate address, as specified in the table located in the Appendix.

BREAKPOINTS

Breakpoints (also known as a breakpoint instructions) are interrupts that you place in a program when you want to execute part of a program and then stop. Breakpoints are usually inserted into programs during the debugging process as a way of displaying registers, memory locations, etc., at critical points in a program. To set a breakpoint, use the breakpoint key (key number 7). Up to 16 different breakpoints can be set at one time.

Breakpoints must be placed at instruction addresses. If you set a breakpoint at an improper location, it could inadvertently change or even "crash" your program. Properly inserted, your program will run until it encounters the breakpoint. When the program stops, the address of the break instruction (contained in the program counter) and the instruction will be displayed. You may examine and make changes to any register or memory location. The instruction displayed when the program stopped will be the next one executed when you press the SS or the GO key.

JUMP TABLE

At the beginning of the CPU Module's EPROM (address C000H) is a jump table containing the addresses of each subroutine. For jump instructions to some of the system-level subroutines, see the jump table located in the Appendix. When using the table, remember that all data in the registers, except the stack pointer, is deleted. So push any register values that you want to preserve. Make sure the stack has plenty of room (400H bytes), since the Monitor ROM routines are stack intensive.

When in the Single Step mode, the Monitor ROM will set the next step at the instruction after your call, since the Monitor cannot modify the code in the ROM to add a software interrupt.

FURTHER INFORMATION

The preceding sections are a very brief overview of the CPU's instruction set and its use. For more information about microprocessors and programming, refer to the appropriate Heathkit/Zenith Educational Systems courses and their related experiments using the versatile ETW-3800 Microprocessor Trainer.

CROSS ASSEMBLERS

Motorola currently offers a cross assembler that will run on the IBM PC and compatibles and the Apple Macintosh computer. These assemblers are free and may be copied for your own use. The assemblers are available on the Motorola bulletin board:

(512) 440-3733

You may download the assemblers at either 600 or 1200 baud.

ALTERNATE LINE VOLTAGE WIRING

Your Microprocessor Trainer has been factory-wired for 120 VAC line voltage, the most often used voltage in the United States. However, in other countries, 220/240 VAC is the most common line voltage. To change the operating line voltage for your Trainer, complete the following steps. CAUTION: Completing these steps allows operation of your Trainer ONLY ON 220/240 VAC.

IMPORTANT: The plug on the line cord furnished with your Microprocessor Trainer does NOT meet the U.S. National Electrical Code requirements for use on line voltages above 120 VAC. Therefore, you MUST cut the plug from the line cord and install an appropriate plug that matches your 220/240 VAC outlet and meets the electrical code requirements. In addition, you must perform the following steps:

Refer to Pictorial 4 for the following steps.

NOTE: These steps should be completed by qualified service personnel only.

( ) Disconnect the Trainer from the AC outlet, if this has not already been done.

( ) Remove the eight #6 × 3/8" self-tapping screws that secure the cabinet top to the cabinet bottom, and lift the top off.

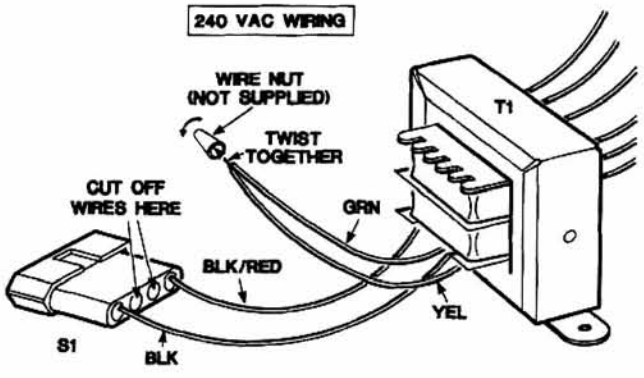
( ) Cut the yellow and green power transformer leads from pins 2 and 3 of the 4-pin socket. Cut the leads as close to the socket as possible. Then remove 3/8" of insulation from the end of each wire.

( ) Twist the ends of the yellow and green power transformer leads together. Then twist a wire nut (not supplied) clockwise over the wires as shown. A wire nut can be obtained locally.

( ) Replace the 1/4-ampere line fuse with a 1/8-ampere, slow-blow fuse (not supplied).

( ) Reposition the cabinet top over the cabinet bottom. Then reinstall the eight #6 × 3/8" self-tapping screws you removed earlier.

Your ETW-3800 Microprocessor Trainer is now wired for operation on 220/240 VAC only.



PICTORIAL 4

IN CASE OF DIFFICULTY

This section of the manual will help you locate and correct minor difficulties which may occur in your Trainer. Any difficulty you might experience can probably be traced to improper setup or interconnections. Use the following Troubleshooting Chart to help narrow down the cause of a problem.

If you cannot resolve the problem, refer to the "Customer Service" information and Warranty supplied with this Manual.

TROUBLESHOOTING CHART

DIFFICULTY	POSSIBLE CAUSE
Power LED does not light.	1. Power cord not plugged into AC outlet. 2. Power switch not turned on. 3. Fuse blown or missing.
No display.	1. CPU Module not installed or not properly inserted. (See Appendix for correct insertion procedure.)
Load errors.	1. Memory Module not properly inserted.
Bad checksum.	1. Program in the Memory Module has been modified since it was last saved. Check program for incorrect values.

CIRCUIT DESCRIPTION

As you read this section, refer to the Schematic Diagrams (fold-in).

CPU MODULE

The replaceable CPU Module is a computer system within a cartridge and its operation is very complex. It contains not only a CPU (control processing unit) but also memory, I/O interfaces, A/D converters, timer, and interrupt logic.

The CPU within the cartridge is a highly integrated 8-bit 68HC11 microprocessor that is object-code compatible with the 6800 microprocessor. For detailed information concerning the CPU, refer to Motorola's MC68HC11A8 data manual.

Memory within the cartridge consists of 16K by 8 EPROM and 28K by 8 Static RAM. The EPROM contains all firmware needed to operate the Trainer from both the keyboard and the RS-232 port. The 28K by 8 Static RAM supplies plenty of memory for complex experiments and demonstrations.

Two communications interfaces are accessible in this cartridge. One is a serial communications interface that provides RS-232 communications on all the popular baud rates from 150 to 19.2k baud. With this interface, only software handshaking is available. The other is a high-speed synchronous serial peripheral interface. It allows several microprocessors and/or peripherals to communicate over a 5-wire interface. With the plug-in ET/ETW-3567 Accessory Backpack, a simple but powerful multiple processor system can be constructed.

The A/D Converters consist of eight multiplexed input channels. You can use them to digitize a sine wave up to 10 kHz with a four-sample resolution. Each of the channels is buffered by an operational amplifier (op amp). The op amp level shifts the input so that a full scale range of ±1 volt can be converted with a resolution of 256 bits. You can adjust the input range anywhere in the range of ±1 volt to ±15 volts with the addition of the appropriate series resistor (R_s) using this equation:

$$R_s = 50k \times V_r - 100k$$

where V_r is the full scale input voltage.

For ±1 volt, R_s= 0 ohms. For ±12 volt, R_s= 1.1M ohms.

A timer system contains three input capture channels, five output capture channels, and a pulse accumulator input. The input and output channels can time events from 1.1 seconds down to 17.36 microseconds.

Seventeen hardware interrupts and one software interrupt are available, of which fifteen are maskable, including an external interrupt. One of the two non-maskable interrupts is external; the other is the RESET line. To interrupt a program and save the register values, use NMI. To get absolute control of the processor and to reinitialize it, use RESET. There is also an illegal opcode trap interrupt to prevent the processor from following a runaway program.

MEMORY MODULE

The optional Memory Module plug-in cartridge contains two 8K × 8 EEPROMs. $\overline{\text{WRITE}}$ and $\overline{\text{READ}}$ signals from the CPU Module determine the direction of data flow into or out of memory.

LOGIC PROBE

The Logic Probe is accessible through the lower signal connection block. Logic pulses are fed to a high-speed comparator (U31) which accepts only the correct logic levels. The output of U31 is then applied to multivibrator U35 which captures high-speed pulses and generates a 0.1 second pulse. The outputs of U35 and U31 are combined at U36 so that if either a DC state or a high-speed pulse occurs, the appropriate LED will light and a high or low audible tone will be generated. The audible tone can be disabled by grounding the LPAUDC terminal point located at the lower signal connector block. The ground resets U32, which disables the audible tone.

RS-232 INTERFACE

The RS-232 port is a serial I/O port. All RS-232 lines pass through the driver/receiver U33 to the CPU Module cartridge socket. From there the transmit/receive lines go to the serial communications interface buffer (U23) located inside the CPU Module. These two lines are then routed to the CPU.

I/O PORTS

All 8-bit input and output lines are connected to the CPU Module through latches U22 and U21. Output data is latched by an OPPRT (output port) signal from the I/O decoding section of the CPU Module. Input data is sampled when the INPRT (input port) line goes high. This occurs with a CPU read from location B280H. Input data may also be latched with a falling edge signal from the IPL input on the lower signal connector block.

LOGIC SWITCHES

One side of the 8-section logic switch is connected to ground. The other side of each section is connected through a 1000 ohm pull-up resistor in resistor pack R215 to the +5 volts DC power supply. The connectors below the switches provide convenient connection for two wires to each section. With a switch in the lower (closed) position, that terminal provides a logic 0 level (ground); in the up (open) position the level will be a logic 1.

LOGIC LEDS

Eight LEDs are accessible through an 8-section connector block located below the LEDs. Each section can accept two wires, which connect to an LED driver. A 10K pull-down resistor at each terminal holds the input to a logic 0 when no connection is made at the connector block. The driver output passes through an LED and a current-limiting resistor. With a logic 0 input the LED is off. When the input rises to a logic 1 the LED lights.

STATUS LEDS

With no input to the Status LEDs, they are off. When a logic low is applied, the LED lights and a path is completed through a 150 ohm current-limiting resistor to the +5 VDC power supply.

KEYBOARD

Address lines A_3, A_4, and A_5 from the CPU connect to keyboard decoder U29. When U29 is enabled, a logic 0 is applied to one of the keyboard key columns and a logic 1 to the other columns. If a key is closed in the column with a logic 0 on it, a logic 0 is placed on the data line for that row of keys. The CPU determines which key is closed by knowing the address that is on the line and which data line is 0. The diodes in series with the three address lines serve as buffers to prevent two adjacent keys from shorting the column select lines together.

+5 VOLT SUPPLIES

Diodes D101 and D102 rectify the voltage from one secondary winding of transformer T1. Capacitors C101 and C102 filter the resulting voltage before it is applied to regulators U101 and U105. Diodes D109 and D110 provide reverse bias protection.

+12 VOLT SUPPLY

Diodes D104 and D106 rectify the voltage from the remaining secondary winding of transformer T1. Capacitors C104 and C105 filter the resulting voltage before it is applied to current regulator U103. Diode D107 provides transient protection for the regulator. Capacitor C107 further filters the output voltage. Diode D111 provides reverse bias protection.

–12 VOLT SUPPLY

The negative 12-volt supply consists of D103, D105, C108, C109, U104, D108, D112, and C112. This circuit operates similarly to the positive 12-volt supply, but produces a regulated –12 volts.

REPLACEMENT PARTS LIST

Component values and circuit component numbers are those referred to on the Schematics and on the Circuit Board X-Ray Views. If a circuit component number is not listed in the Parts List, that component is not used in the circuit.

To order a replacement part, always include the PART NUMBER. Use the Parts Order Form furnished with this unit. If a Parts Order Form is not available, refer to the "Customer Service" information supplied with this Manual.

A replacement part may look slightly different than the original part, or may have different printing on it. In any case, the performance of the replacement part will meet or exceed the requirements of the original part. For example: A 15-volt capacitor (10 µF, 15 V) may be replaced with a 25-volt capacitor (10 µF, 25 V).

CABINET

ELECTRONIC PARTS

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
C52, C53	21-71	.001 µF (1000 pF) ceramic capacitor (may be marked 102)
C51	27-127	.047 µF (474) Mylar capacitor
L51	45-615	RF choke
T1	54-1054	Power transformer
SW1	61-58	Power switch
D1	412-634	Red LED (light emitting diode)
F1	421-33	Fuse, 1/4-ampere slow-blow

CONNECTORS

	432-1279	Flat connector lug
	230-6329	2-pin plug with wires
	230-6330	2-pin socket with wires
	230-6344	Green wire with connectors

CABLES-LINE CORD

	230-6334	Line cord
	134-1693	LED (2-wire) cable
	134-1692	8-wire power cable
	134-2046	8-wire RS-232 cable

PLASTIC AND METAL PARTS

	92-929	Cabinet bottom
	92-930	Cabinet top
	94-691	Cabinet insert
	204-3001	AC chassis
	204-3066	AC shield
	205-2005	Backpack mounting plate

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

HARDWARE

NOTE: Metric equivalent hardware may be used. Do not mix standard and metric hardware.

810-13	3 x 8mm screw
810-17	3.5 x 10mm flat head screw
250-1412	4-40 x 3/8" black screw
254-9	#4 lockwasher
252-2	#4 nut
255-757	4-40 spacer
250-1434	6-BT x 3/8" self-tapping screw
250-1436	8-32 x 3/8" screw
254-2	#8 lockwasher
252-4	#8 nut

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

MISCELLANEOUS

230-6353	Foam cushion
75-736	Strain relief
230-6355	Paper insulator
230-6356	Filter circuit board (may be marked 85-2789-1)
261-49	Foot
230-6358	Fuseholder
230-6345	Small sleeving
230-6346	Large sleeving
331-7	Solder
354-5	Cable tie
75-918	Top insert insulator
	Blue and white label
	Caution label
	Caution marking label
	Warning label

POWER SUPPLY CIRCUIT BOARD

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

RESISTORS

R101	6-271-12	270 Ω, 1/4-watt, 5% (red-vio-brn)

CAPACITORS

C101	25-903	6800 µF electrolytic
C102-C103	27-145	.22 µF (224) Mylar
C104	25-875	1000 µF electrolytic
C105	21-786	.1 µF (104) axial-lead ceramic
C107	25-885	100 µF electrolytic
C108	25-875	1000 µF electrolytic
C109	21-786	.1 µF (104) axial-lead ceramic
C112	25-885	100 µF electrolytic
C113	27-145	.22 µF (224) Mylar

DIODES-INTEGRATED CIRCUITS

D101-D106	57-42	1N5401 or 3A1 diode
D107-D112	57-65	1N4002 diode

NOTE: Integrated circuits may be marked for identification in any of the following four ways.

1. Part number.
2. Type number. (For integrated circuits, this refers only to the numbers and letters shown in BOLD print. Disregard any other numbers or letters shown on the IC.)
3. Part number and type number.
4. Part number with a type number other than the one shown.

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

U101	442-30	LM309K
U103	442-664	79M12
U104	442-663	78M12
U105	442-30	LM309K

CONNECTORS-SOCKET

P101	432-876	8-pin plug
P102	432-943	2-pin plug
P103	432-877	10-pin plug
	432-1279	Flat connector lug
	434-189	Transistor socket

HARDWARE

250-1425	6-32 x 1/2" screw

MISCELLANEOUS

230-6271	Power supply circuit board (may be marked 85-3191-1)
215-698	Flat heat sink
352-31	Thermal compound

MAIN CIRCUIT BOARD

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

RESISTORS

NOTE: The following resistors have a tolerance of 5% unless otherwise listed. A 5% tolerance is indicated by a gold fourth color band.

1/4-Watt Resistors

R11-R19	6-271-12	270 Ω (red-viol-brn)
R34-R35	6-271-12	270 Ω (red-viol-brn)
R36	6-4322-12	43.2 kΩ (red-blk-blk-red)
R37	6-2002-12	20 kΩ (yel-org-red-red)
R110-R156	6-271-12	270 Ω (red-viol-brn)
R157-R164	6-101-12	100 Ω (brn-blk-brn)
R165-R172	6-271-12	270 Ω (red-viol-brn)
R177-R178	6-271-12	270 Ω (red-viol-brn)
R181-R188	6-151-12	150 Ω (brn-grn-brn)
R189-R196	6-271-12	270 Ω (red-viol-brn)
R216-R223	6-271-12	270 Ω (red-viol-brn)
R311	6-271-12	270 Ω (red-viol-brn)
R330	6-471-12	470 Ω (yel-viol-brn)
R435-R466	6-271-12	270 Ω (red-viol-brn)

1/8-Watt Resistors

R23-R29	6-181-11	180 Ω (brn-gry-brn)
R31	6-5230-11	523 Ω, 1% (grn-red-org-blk)
R32	6-1000-11	100 Ω, 1% (brn-blk-blk-blk)
R33	6-103-11	10 kΩ (brn-blk-org)
R38-R39	6-181-11	180 Ω (brn-gry-brn)
R41-R49	6-101-11	100 Ω (brn-blk-brn)
R210	6-181-11	180 Ω (brn-gry-brn)
R224-R231	6-122-11	1200 Ω (brn-red-red)
R240-R247	6-122-11	1200 Ω (brn-red-red)
R310	6-181-11	180 Ω (brn-gry-brn)
R312-R313	6-104-11	100 kΩ (brn-blk-yel)
R314	6-130-11	10 kΩ (brn-blk-org)
R315	6-8250-11	825 Ω, 1% (gry-red-grn-blk)
R316	6-130-11	10 kΩ (brn-blk-org)
R318	6-5490-11	549 Ω, 1% (grn-yel-wht-blk)
R319	6-130-11	10 kΩ (brn-blk-org)
R320-R322	6-181-11	180 Ω (brn-gry-brn)
R323	6-222-11	2200 Ω (red-red-red)
R324	6-223-11	22 kΩ (red-red-org)
R325	6-273-11	27 kΩ (red-viol-org)
R326	6-272-11	2700 Ω (red-viol-red)
R327	6-130-11	10 kΩ (brn-blk-org)
R328	6-130-11	10 kΩ (brn-blk-org)
R410-R432	6-101-11	100 Ω (brn-blk-brn)

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION

Other Resistors

R21	10-1137	2 kΩ potentiometer
RN173-RN176	9-128	10 kΩ resistor pack
RN179-RN180	9-128	10 kΩ resistor pack
RN211-RN213	9-128	10 kΩ resistor pack
RN215	9-118	1 kΩ resistor pack

CAPACITORS

C11-C14	25-866	22 µF electrolytic
C15	25-978	.1 F
C19	25-927	22 µF electrolytic
C21-C23	21-786	.1 µF ceramic
C24	21-761	.01 µF (103) glass
C25-C28	21-786	.1 µF ceramic
C31	25-863	4.7 µF electrolytic
C32	25-863	4.7 µF electrolytic
C33	25-866	22 µF electrolytic
C34	21-786	.1 µF ceramic
C35-C36	25-866	22 µF electrolytic
C37	25-927	22 µF electrolytic
C38	21-786	.1 µF ceramic
C39	27-161	.01 µF Mylar
C310	21-786	.1 µF ceramic
C311	21-761	.01 µF (103) glass
C312	21-761	.01 µF (103) glass
C313	27-161	.01 µF Mylar
C314	21-811	.33 µF (334) axial-lead ceramic
C315	21-786	.1 µF ceramic
C316	21-761	.01 µF

DIODES-TRANSISTORS

D11	57-607	1N5817
D29	56-655	1N6263
D37, D39	57-607	1N5817
D210-D211	56-655	1N6263
D310-D313	57-607	1N5817
J31	417-902	NPD 5566

INTEGRATED CIRCUITS

U21	443-1673	74ACT374
U22	443-1590	74ACT373
U24	443-1591	74ACT04
U25	443-1443	74ACT244
U27	443-1443	74ACT244
U31	442-820	NE521
U32	442-665	79L05
U33	443-1467	MC145406
U35	443-1592	74HCT423
U36	443-1593	74HCT32
U326	442-740	LM556

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
CONNECTORS-SOCKETS		
P11	432-1816	60-pin male connector
P12	432-1727	40-pin female memory cartridge connector
P13-P14	432-1720	100-pin signal connector block
P15	432-1064	10-pin male connector
P21-P22	432-1719	16-pin connector block
P25	432-1811	14-pin LCD module connector
P32	432-1268	8-pin male RS-232 cable connector
P212-P215	432-1656	62-pin female CPU cartridge connector
	432-1610	Breadboard block
MISCELLANEOUS		
D21-D28	412-657	Green LED
D31-D34	412-657	Green LED
D35	412-634	Red LED
D36	412-657	Green LED
L31-L32	475-39	Inductor
S101-S121	64-955	Pushbutton switch
S225	60-656	8-section DIP switch
T101	473-29	Transducer
V301	411-902	LCD display module
	85-3399-1	Circuit board
	462-1238	1 set of keycaps

CPU MODULE*

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
RESISTORS		
NOTE: All resistors are rated at 1/8-watt and have a tolerance of 5% unless otherwise stated.		
R11	627-106-18	10 MΩ
R12	627-103-18	10 kΩ
R13	627-104-18	100 kΩ
R14	627-102-18	1 kΩ
R15-R19	627-103-18	10 kΩ
R21-R25	627-223-18	22 kΩ
R31	627-1503-18	150 kΩ, 1%
R32	627-1003-18	100 kΩ, 1%
R33	627-3003-18	300 kΩ, 1%
R34-R35	627-753-18	75 kΩ
R36	627-3003-18	300 kΩ, 1%
R37	627-1003-18	100 kΩ, 1%
R38-R39	627-1503-18	150 kΩ, 1%
R110-R113	627-103-18	10 kΩ
R141	627-472-18	4700 Ω
R142-R143	627-103-18	10 kΩ
R144	627-471-18	470 Ω
R145	627-2001-18	2 kΩ, 1%
R146	627-1002-18	10 kΩ, 1%
R310	627-1003-18	100 kΩ, 1%
R311	627-3003-18	300 kΩ, 1%
R312-R313	627-753-18	75 kΩ
R314	627-3003-18	300 kΩ, 1%
R315	627-1003-18	100 kΩ, 1%
R316	627-1503-18	150 kΩ, 1%
R317	627-3003-18	300 kΩ, 1%
R318	627-1003-18	100 kΩ, 1%
R319	627-753-18	75 kΩ
R320	627-3003-18	300 kΩ, 1%
R321	627-1003-18	100 kΩ, 1%
R322	627-753-18	75 kΩ
R323	627-3003-18	300 kΩ, 1%
R324	627-1003-18	100 kΩ, 1%
R325	627-3003-18	300 kΩ, 1%
R326	627-1003-18	100 kΩ, 1%
R327	627-753-18	75 kΩ
R328	627-1503-18	150 kΩ, 1%
R329	627-753-18	75 kΩ
R330-R332	627-1503-18	150 kΩ, 1%
Other Resistors		
AN11	9-169	22 kΩ resistor pack
AN12	9-143	22 kΩ resistor pack
CAPACITORS		
C11-C12	633-5	25 pF
C13-C19	21-786	0.1 µF
C21-C26	21-786	0.1 µF
C27	25-212	22 µF
C31-C36	21-786	0.1 µF
C37	631-26-79	1000 pF
C38	21-786	0.1 µF
C110-C111	21-786	0.1 µF
C113-C114	21-786	0.1 µF
DIODES		
D11	57-607	1N5817
D31-D39	640-21	1N4149
D310-D316	640-21	1N4149

* Parts are for reference only. CPU Module to be serviced by Heath Company only.

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
INTEGRATED CIRCUITS		
U13	444-675	27128
U14	442-835	S8054HN
U15	643-109	74ACT245
U16-U17	643-117	74ACT244
U18	444-676	16V8
U19	643-182	74ACT04
U21	643-183	74ACT138
U22-U23	643-117	74ACT244
U24	643-266	74HCT32
U26	643-117	74ACT244
U29	643-266	74HCT32
U31	642-4	TL431CD
U32	642-3	MC78L05D
U101	643-180	68HC11
U111	643-265	74HCT00
U110	643-178	62256
U102	643-267	74HCT373
U341	642-7	LM353
U349	642-6	LM324
U355	642-6	LM324

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
MISCELLANEOUS		
Y11	404-724	3.6864 MHz crystal
	85-3521-1	Circuit board
	92-919-1	Case back cover
	92-931	Case front cover with ground clips
	250-1322	#6 x 5/8" BT self-threading screw
	250-1630	#6 x 1.5 BT self-threading screw
	260-735	Insert ground clip
	266-1329	Static protection clip
	390-3212	Label
	434-312	28-pin IC socket
	434-311	20-pin IC socket
	434-469	52-pin IC socket
	595-4170-02	User's Manual
	597-4983	8-bit information card

MEMORY MODULE

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
RESISTORS		
R11-R12	6-472-12	4.7 kΩ, 1/4-watt, 5%
CAPACITORS		
C301-C302	21-786	.1 µF
DIODES		
D11-D12	56-655	1N6263

CIRCUIT Comp. No.	HEATH Part No.	DESCRIPTION
INTEGRATED CIRCUITS		
U301-U302	643-179	58C65
MISCELLANEOUS		
	85-3434-1	Circuit board
	92-932	Case front with ground clips
	92-917	Case back
	250-1322	#6 x 5/8" BT self-threading screw
	266-1330	Memory static protection clip
	390-3211	Label
	434-312	IC socket

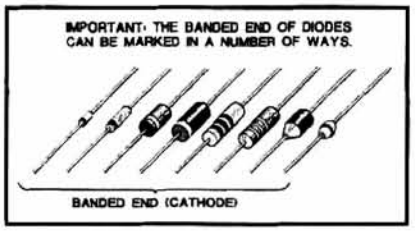
SEMICONDUCTOR IDENTIFICATION

TRAINER

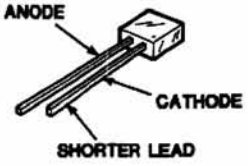
DIODES-LEDS-TRANSISTORS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D101-D106	57-42	1N5401 or 3A1	A1
D107-D112	57-65	1N4002	A1
D29, D210, C211	56-655	1N6263	A1
D11, D37-D39, D310-D311	57-607	1N5817	A1
D21-D28, D31-D34, D36	412-657	Green LED	A2
D1, D35	412-634	Red LED	A2
J31	417-902	NPD 5566	A3

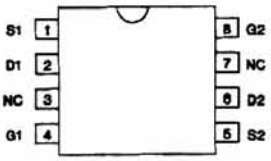
A1



A2



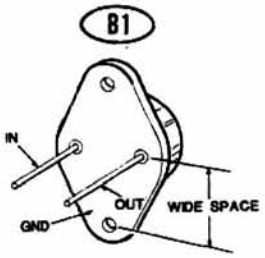
A3



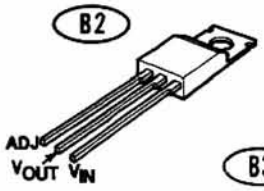
INTEGRATED CIRCUITS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U101, U105	442-30	LM309K	B1
U103	442-708	LM317T	B2
U104	442-709	LM337T	B3

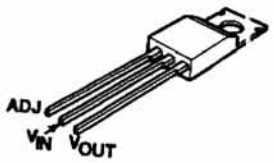
B1



B2

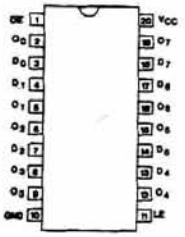


B3

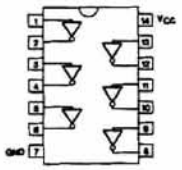


COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U21	443-1673	74ACT374	B4
U22	443-1590	74ACT373	B4
U24	443-1591	74ACT04	B5
U25, U27	443-1443	74ACT244	B6
U31	442-820	NE521	B7
U32	442-665	79L05	B8
U33	443-1467	MC145406	B9
U35	443-1592	74HCT423	B10
U36	443-1593	74HCT32	B11
U326	442-740	LM556	B12

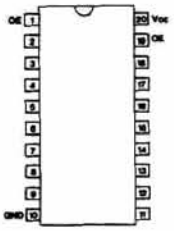
B4



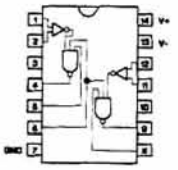
B5



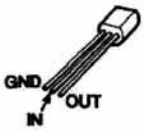
B6



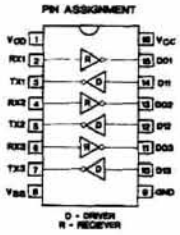
B7



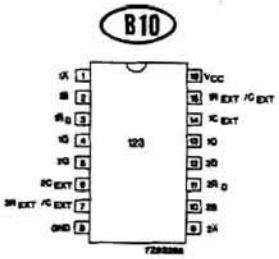
B8



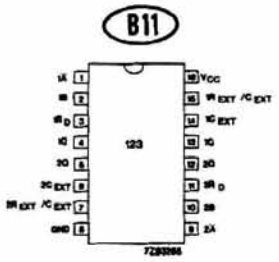
B9



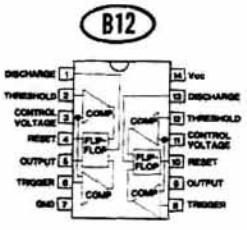
B10



B11



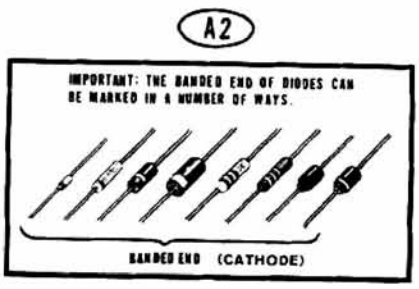
B12



CPU MODULE

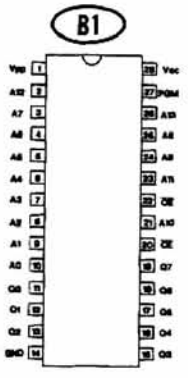
DIODES

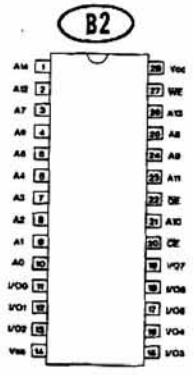
COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D31-D39, D310-D316	640-21	1N4149	A1
D11	57-607	1N5817	A2

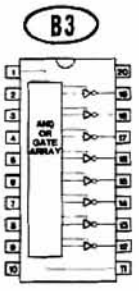


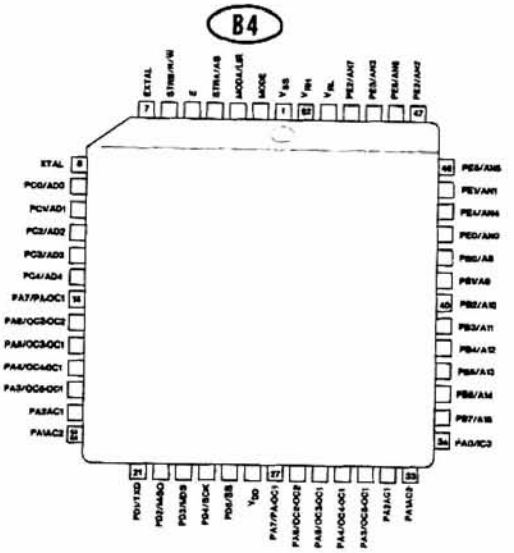
INTEGRATED CIRCUITS

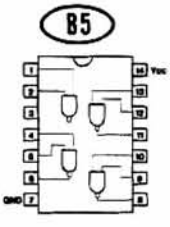
COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U13	444-675	27128	B1
U110	643-178	62256	B2
U18	444-676	16VL8	B3
U101	643-180	68HC11	B4
U111	643-181	74HCT00	B5
U19	643-182	74ACT04	B6
U21	643-183	74ACT138	B7

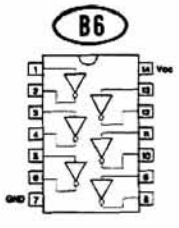


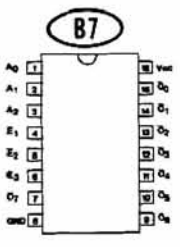






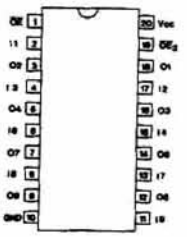




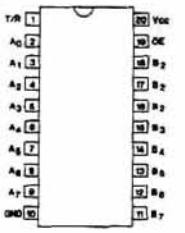


COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U16, U17, U22 U23, U26	643-186	74ACT244	B8
U15	643-109	74ACT245	B9
U24, U29	643-266	74HCT32	B10
U102	643-185	74ACT373	B11
U32	642-3	MC78L05D	B12
U31	642-4	TL431CD	B13
U14	442-835	S8054HN	B14
U349, U355	642-6	LM324	B15
U341	642-7	LM353	B16

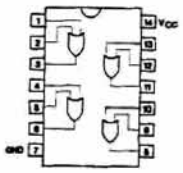
B8



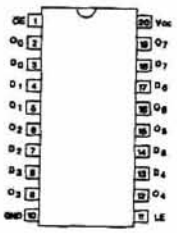
B9



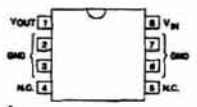
B10



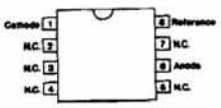
B11



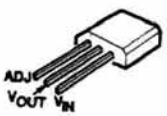
B12



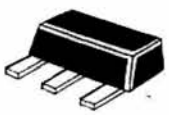
B13



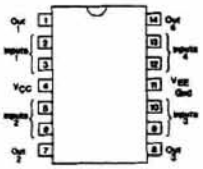
B14



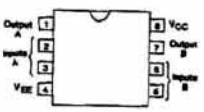
OR



B15



B16

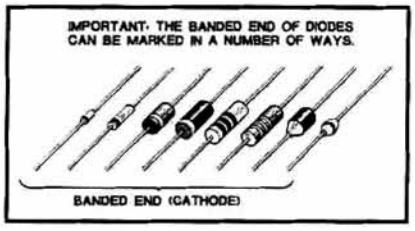


MEMORY MODULE

DIODES

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
D11, D12	56-655	1N6263	A1

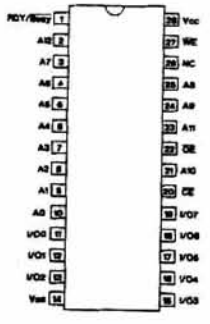
A1



INTEGRATED CIRCUITS

COMPONENT NUMBER	HEATH PART NUMBER	MAY BE REPLACED WITH	KEY NUMBER
U301, U302	643-179	58C65	B1

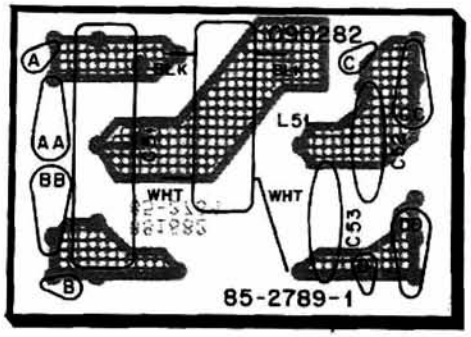
B1



CIRCUIT BOARD X-RAY VIEWS

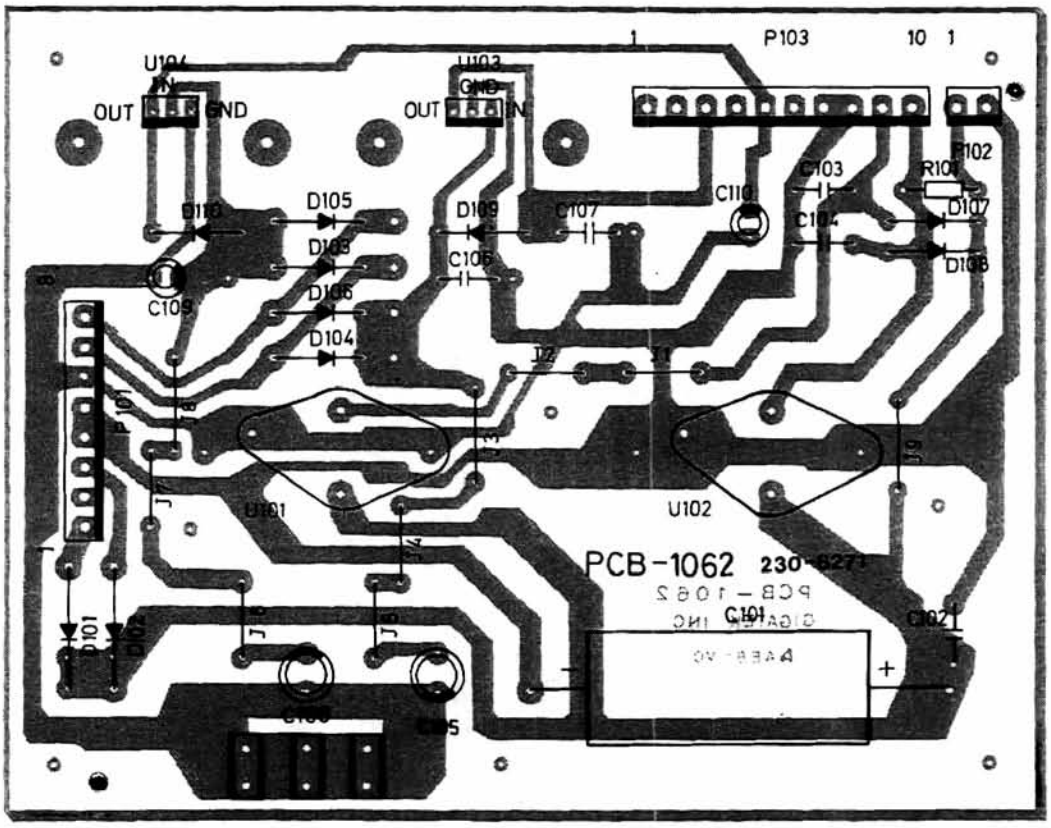
To find the PART NUMBER of a component for the purpose of ordering a replacement part:

1. Find the circuit component part number on the appropriate X-Ray View.
2. Locate the same number in the "Circuit Component Number" column of the corresponding "Parts List".
3. Adjacent to the circuit component number, you will find the PART NUMBER and DESCRIPTION, which you must supply when you order a replacement part.



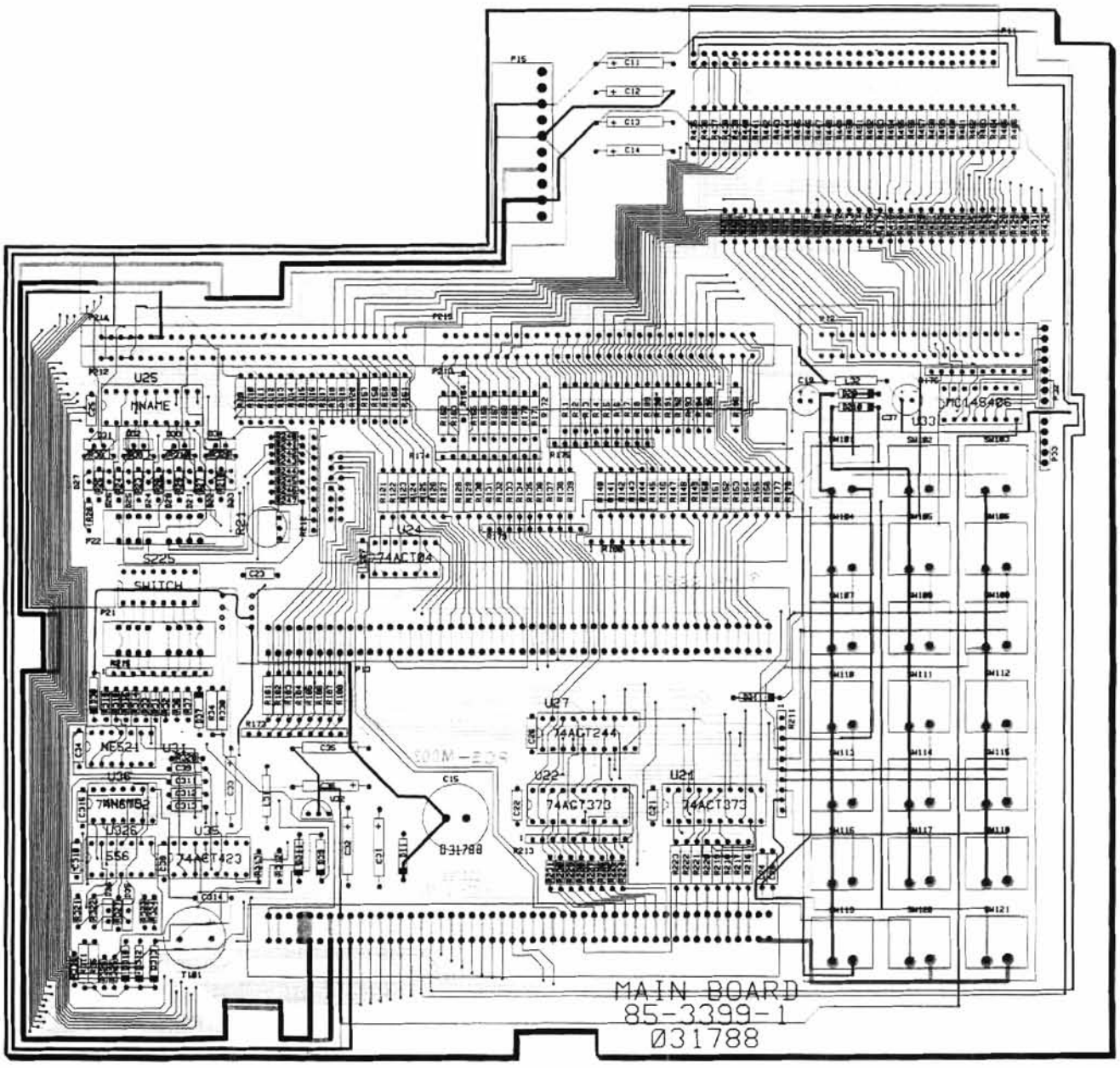
FILTER CIRCUIT BOARD

(Shown from the component side)



POWER SUPPLY CIRCUIT BOARD

(Shown from the component side)

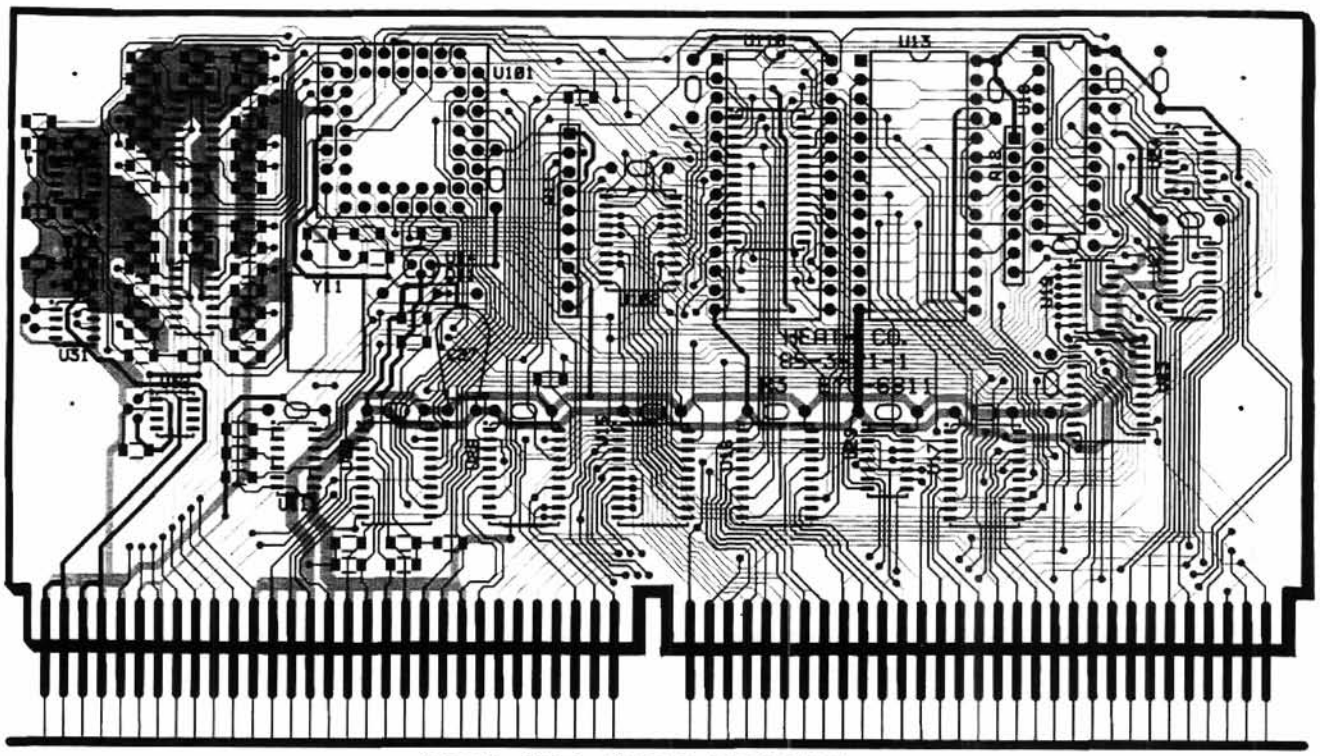


MAIN CIRCUIT BOARD

(Shown from the component side)

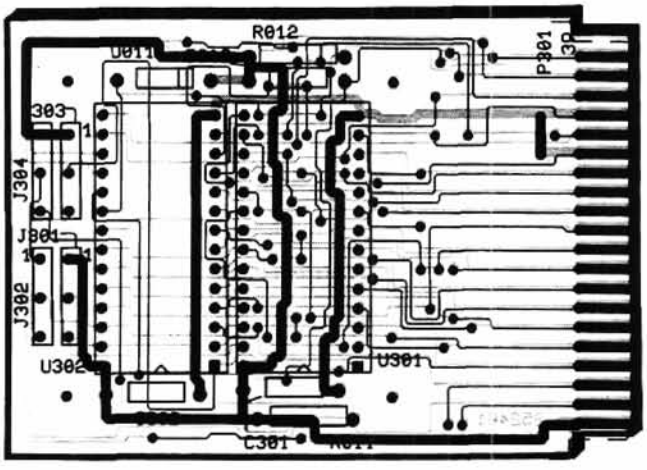
MAIN CIRCUIT BOARD

(Shown from the component side)



ETC-6811 CPU MODULE

(Shown from the component side)



ETC-128 MEMORY MODULE

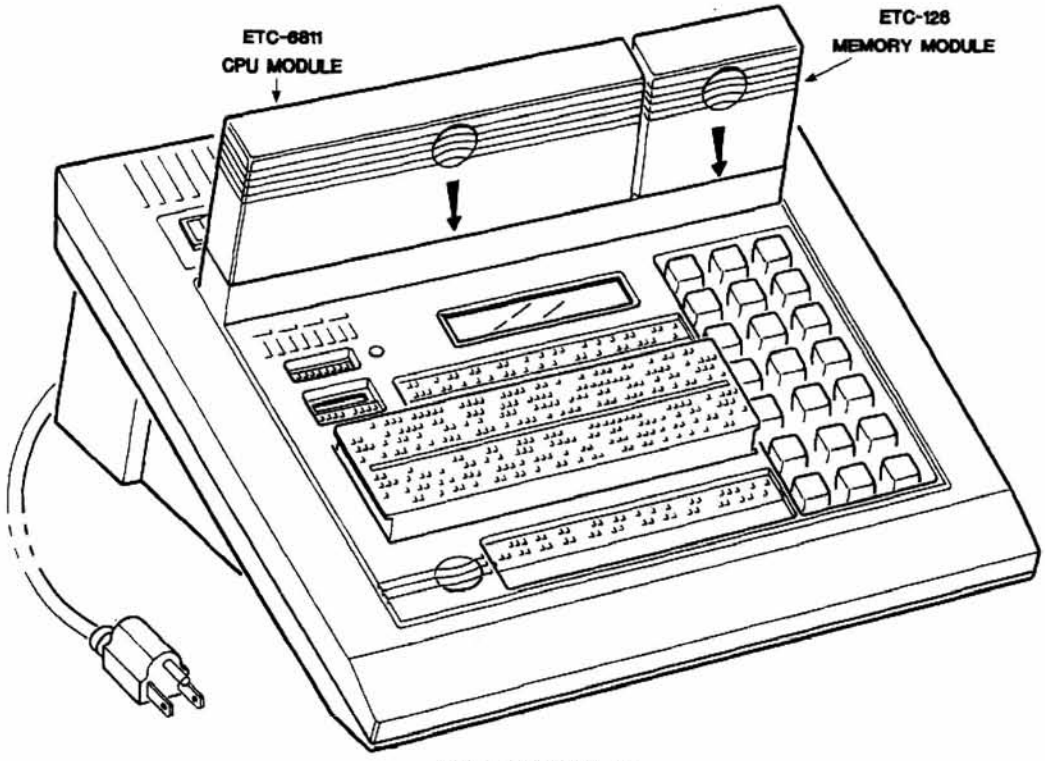
(Shown from the component side)

APPENDIX

CARTRIDGE INSERTION AND REMOVAL

With proper care and handling, your ETW-3800 Microprocessor Trainer will provide you with years of service. Knowing how to properly install and remove the cartridges used with the Trainer will help you to protect your investment. Refer to Pictorial 5 while following these steps to correctly insert and remove cartridges.

1. Make sure the Trainer power switch is off before removing or inserting a cartridge.
2. Remove the static protection clip from the cartridge. A cartridge should always have a static protection clip on whenever it is not in the Trainer.
3. Insert the cartridge into its appropriate connector with the lettering facing you. The CPU Module installs in the two left connectors and the Memory Module installs into the right connector. Both the CPU and the Memory Module are keyed to go into the connectors of the Trainer one way only. Do not force the cartridges in backwards or they will be damaged.



PICTORIAL 5

MEMORY I/O MAP

ADDRESS	MEMORY	
0000H 5FFFH	User Code Area	
6000H 6FFFH	User Data and Stack Area NOTE: This is where the user stack should start.	
6900H 69FFH	Monitor Data and Stack Area	
7000H AFFFH	Memory Module Area	
	NOTE: This is the Internal I/O Area.	
B000H	PORTA	Port A Data Register
B001H	Reserved	
B002H	PIOC	Parallel I/O Control Register
B003H	PORTC	Port C Data Register
B004H	PORTB	Port B Data Register
B005H	PORTCL	Port C Latched Data Register
B006H	Reserved	
B007H	DDRC	Data Direction Register for Port C
B008H	PORTD	Port D Data Register
B009H	DDRD	Data Direction Register for Port D
B00AH	PORTE	Port E Data Register
B00BH	CFORC	Time Compare Force Register
B00CH	OC1M	Output Compare 1 Mask Register
B00DH	OC1D	Output Compare 1 Data Register
B00EH, B00FH	TCNT	Timer Counter Register
B010H, B011H	TIC1	Timer Input Capture Register 1
B012H, B013H	TIC2	Timer Input Capture Register 2
B014H, B015H	TIC3	Timer Input Capture Register 3
B016H, B017H	TOC1	Timer Output Compare Register 1
B018H, B019H	TOC2	Timer Output Compare Register 2
B01AH, B01BH	TOC3	Timer Output Compare Register 3
B01CH, B01DH	TOC4	Timer Output Compare Register 4
B01EH, B01FH	TOC5	Timer Output Compare Register 5
B020H	TCTL1	Timer Control Register 1
B021H	TCTL2	Timer Control Register 2
B022H	TMSK1	Main Timer Interrupt Mask Reg. 1
B023H	TFLG1	Main Timer Interrupt Flag Reg. 1
B024H	TMSK2	Misc. Timer Interrupt Mask Reg. 2
B025H	TFLG2	Misc. Timer Interrupt Flag Reg. 2
B026H	PACTL	Pulse Accumulator Control Register
B027H	PACNT	Pulse Accumulator Count Register
B028H	SPCR	SPI Control Register
B029H	SPSR	SPI Status Register
B02AH	SPDR	SPI Data Register
B02BH	BAUD	SCI Baud Rate Control Register
B02CH	SCCR1	SCI Control Register 1
B02DH	SCCR2	SCI Control Register 2
B02EH	SCSR	SCI Status Register
B02FH	SCDR	SCI Data Register

B030H	ADCTL	A/D Control/Status Register
B031H	ADR1	A/D Result Register 1
B032H	ADR2	A/D Result Register 2
B033H	ADR3	A/D Result Register 3
B034H	ADR4	A/D Result Register 4
B035H	Reserved	
B036H	Reserved	
B037H	Reserved	
B038H	Reserved	
B039H	OPTION	System Configuration Options
B03AH	COPRST	Arm/Reset COP Timer Circuitry
B03BH	PPROG	EEPROM Programming Register
B03CH	HPRIO	Highest Priority Interrupt and Misc.
B03DH	INIT	RAM and I/O Mapping Register
B03EH	TEST1	Factory Test Register
B03FH	CONFIG	Configuration Control Register
B218H	Keyboard Column 3	NOTE: Keyboard Columns are not fully decoded. Use only the address specified.
B228H	Keyboard Column 2	
B230H	Keyboard Column 1	
B240H B27FH	Output Port	
B280H B2BFH	Input Port	
B2C0H B2FFH	LCD Registers	NOTE: B2C0H is a Command Register and B2C1H is a Data Register. The Remaining Registers are undecoded
B300H B33FH	General I/O 0	
B340H B37FH	General I/O 1	
B380H B3BFH	General I/O 2	
B3C0H B3FFH	General I/O 3	
B400H BFFFH	Unused	
C000H FFFFH	Monitor ROM Area	

SUBROUTINE JUMP TABLE

EPROM ADDRESS	LABEL	INTRODUCTION
C000	JMP MAIN	Return Control to Monitor
C003	JMP GETC	Returns ASCII Character in B, A=0
C006	JMP PUTC	Outputs ASCII Character in B
C009	JMP SETRS232	Select RS-232 for I/O
C00C	JMP SETLCD	Select LCD, Keypad for I/O
C00F	JMP GETHBYT	Get a Hex Byte, Return in B
C012	JMP GETHWRD	Get a Hex Word, Return in D
C015	JMP PUTHBYT	Print Byte in B, Out in Hex
C018	JMP PUTHWRD	Print Word in D, Out in Hex
C01B	JMP PUTSTRING	Print a Null Terminated String Pointed to by D
C01E	JMP COPYUP	Copy Block of Memory, B=# Copy, Push Source, Push Destination
C021	JMP COPYDOWN	Copy Block of Memory, B=# Copy, Push Source, Push Destination
C024	JMP COPY	Push Incordec, # of Bytes, Destination, D=Source
C027	JMP PAUSE	Wait Over a Second
C02A	JMP PUTDSWRD	Convert Signed Value in D to Decimal and Print Out
C02D	JMP PUTUDWRD	Convert Unsigned Value in D to Decimal and Print Out

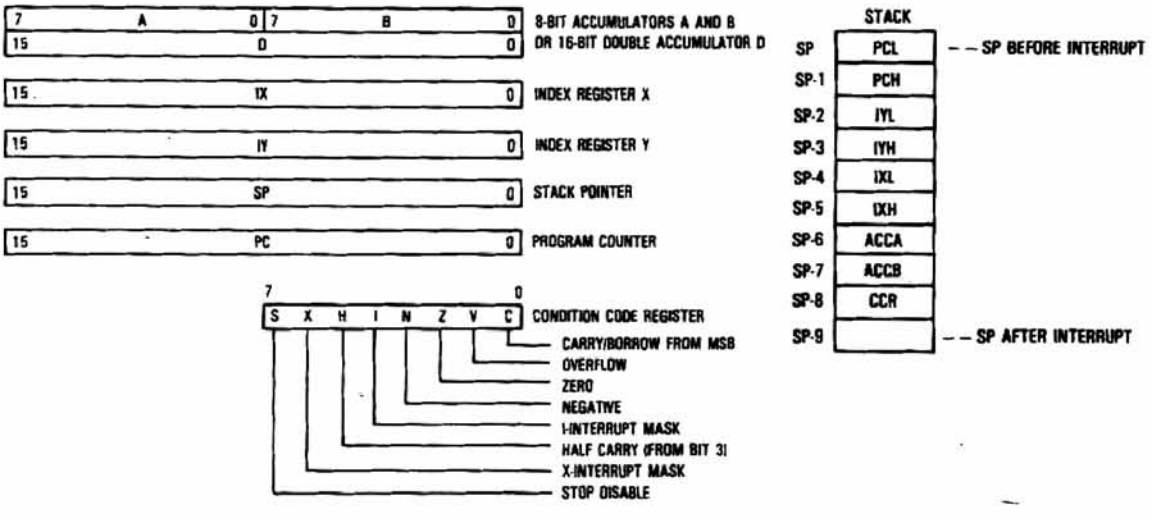
INTERRUPT VECTOR TABLE

ADDRESS	INTERRUPT SOURCE
6901H,6902H	SCI Serial System
6904H,6905H	SPI Serial Transfer Complete
6907H,6908H	Pulse Accumulator Input Edge
690AH,690BH	Pulse Accumulator Overflow
690DH,690EH	Timer Overflow
6910H,6911H	Timer Output Compare 5
6913H,6914H	Timer Output Compare 4
6916H,6917H	Timer Output Compare 3
6919H,691AH	Timer Output Compare 2
691CH,691DH	Timer Output Compare 1
691FH,6920H	Timer Input Capture 3
6922H,6923H	Timer Input Capture 2
6925H,6926H	Timer Input Capture 1
6928H,6929H	Real Timer Interrupt
692BH,692CH	IRQ External Pin Interrupt
692EF,692FH	XIRQ Pin Interrupt
6931H,6932H	SWI Instruction Interrupt
6934H,6935H	Illegal Opcode Trap
6937H,6938H	COP Failure
693AH,693BH	COP Clock Monitor Fail

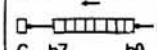
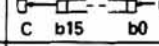
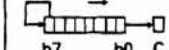
ASCII CHART

ASCII CHARACTER SET (7-Bit Code)								
MS Dig. / LS Dig.	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

PROGRAMMING MODEL

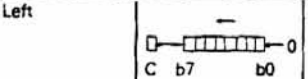
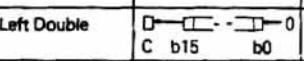
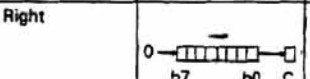
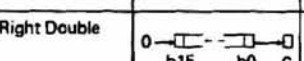


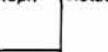
INSTRUCTIONS, ADDRESSING MODES, AND EXECUTION TIMES

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B → A	INH	1B		1	2	-	-	↕	-	↕	↕	↕	↕
ABX	Add B to X	IX + 00:B → IX	INH	3A		1	3	-	-	-	-	-	-	-	-
ABY	Add B to Y	IY + 00:B → IY	INH	18 3A		2	4	-	-	-	-	-	-	-	-
ADCA (opr)	Add with Carry to A	A + M + C → A	A IMM	89	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			A DIR	99	dd	2	3								
			A EXT	B9	hh ll	3	4								
			A IND,X	A9	ff	2	4								
			A IND,Y	18 A9	ff	3	5								
ADCB (opr)	Add with Carry to B	B + M + C → B	B IMM	C9	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			B DIR	D9	dd	2	3								
			B EXT	F9	hh ll	3	4								
			B IND,X	E9	ff	2	4								
			B IND,Y	18 E9	ff	3	5								
ADDA (opr)	Add Memory to A	A + M → A	A IMM	8B	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			A DIR	9B	dd	2	3								
			A EXT	BB	hh ll	3	4								
			A IND,X	AB	ff	2	4								
			A IND,Y	18 AB	ff	3	5								
ADDB (opr)	Add Memory to B	B + M → B	B IMM	CB	ii	2	2	-	-	↕	-	↕	↕	↕	↕
			B DIR	DB	dd	2	3								
			B EXT	FB	hh ll	3	4								
			B IND,X	EB	ff	2	4								
			B IND,Y	18 EB	ff	3	5								
ADDD (opr)	Add 16-Bit to D	D + M:M + 1 → D	IMM	C3	jj kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	D3	dd	2	5								
			EXT	F3	hh ll	3	6								
			IND,X	E3	ff	2	6								
			IND, Y	18 E3	ff	3	7								
ANDA (opr)	AND A with Memory	A•M → A	A IMM	84	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	94	dd	2	3								
			A EXT	B4	hh ll	3	4								
			A IND,X	A4	ff	2	4								
			A IND,Y	18 A4	ff	3	5								
ANDB (opr)	AND B with Memory	B•M → B	B IMM	C4	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D4	dd	2	3								
			B EXT	F4	hh ll	3	4								
			B IND,X	E4	ff	2	4								
			B IND,Y	18 E4	ff	3	5								
ASL (opr)	Arithmetic Shift Left	←  C b7 b0	EXT	78	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	68	ff	2	6								
			IND,Y	18 68	ff	3	7								
ASLA			A INH	48		1	2								
ASLB			B INH	58		1	2								
ASLD	Arithmetic Shift Left Double	 C b15 b0	INH	05		1	3	-	-	-	-	↕	↕	↕	↕
ASR (opr)	Arithmetic Shift Right	→  b7 b0 C	EXT	77	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	67	ff	2	6								
			IND,Y	18 67	ff	3	7								
ASRA			A INH	47		1	2								
ASRB			B INH	57		1	2								
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	2	3	-	-	-	-	-	-	-	-
BCLR (opr) (msk)	Clear Bit(s)	M•(mm) → M	DIR	15	dd mm	3	6	-	-	-	-	↕	↕	0	-
			IND,X	1D	ff mm	3	7								
			IND,Y	18 1D	ff mm	4	8								
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	2	3	-	-	-	-	-	-	-	-
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	2	3	-	-	-	-	-	-	-	-

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
BGE (rel)	Branch if ≥ = Zero	? N ⊕ V = 0	REL	2C	rr	2	3	-	-	-	-	-	-	-	-
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	2	3	-	-	-	-	-	-	-	-
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	2	3	-	-	-	-	-	-	-	-
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	2	3	-	-	-	-	-	-	-	-
BITA (opr)	Bit(s) Test A with Memory	A•M	A IMM	85	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	95	dd	2	3								
			A EXT	B5	hh ll	3	4								
			A IND,X	A5	ff	2	4								
			A IND,Y	18 A5	ff	3	5								
BITB (opr)	Bit(s) Test B with Memory	B•M	B IMM	C5	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D5	dd	2	3								
			B EXT	F5	hh ll	3	4								
			B IND,X	E5	ff	2	4								
			B IND,Y	18 E5	ff	3	5								
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	2	3	-	-	-	-	-	-	-	-
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	2	3	-	-	-	-	-	-	-	-
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	2	3	-	-	-	-	-	-	-	-
BLT (rel)	Branch If < Zero	? N ⊕ V = 1	REL	2D	rr	2	3	-	-	-	-	-	-	-	-
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	2	3	-	-	-	-	-	-	-	-
BNE (rel)	Branch if Not = Zero	? Z = 0	REL	26	rr	2	3	-	-	-	-	-	-	-	-
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	2	3	-	-	-	-	-	-	-	-
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	2	3	-	-	-	-	-	-	-	-
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M• mm = 0	DIR	13	dd mm rr	4	6	-	-	-	-	-	-	-	-
			IND,X	1F	ff mm rr	4	7								
			IND,Y	18 1F	ff mm rr	5	8								
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	-	-	-	-	-	-	-	-
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? (M)•mm = 0	DIR	12	dd mm rr	4	6	-	-	-	-	-	-	-	-
			IND,X	1E	ff mm rr	4	7								
			IND,Y	18 1E	ff mm rr	5	8								
BSET(opr) (msk)	Set Bit(s)	M + mm → M	DIR	14	dd mm	3	6	-	-	-	-	↕	↕	0	-
			IND,X	1C	ff mm	3	7								
			IND,Y	18 1C	ff mm	4	8								
BSR (rel)	Branch to Subroutine	See Special Ops	REL	8D	rr	2	6	-	-	-	-	-	-	-	-
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	2	3	-	-	-	-	-	-	-	-
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	2	3	-	-	-	-	-	-	-	-
CBA	Compare A to B	A − B	INH	11		1	2	-	-	-	-	↕	↕	↕	↕
CLC	Clear Carry Bit	0 → C	INH	0C		1	2	-	-	-	-	-	-	-	0
CLI	Clear Interrupt Mask	0 → I	INH	0E		1	2	-	-	-	0	-	-	-	-
CLR (opr)	Clear Memory Byte	0 → M	EXT	7F	hh ll	3	6	-	-	-	-	0	1	0	0
			IND,X	6F	ff	2	6								
			IND,Y	18 6F	ff	3	7								
CLRA	Clear Accumulator A	0 → A	A INH	4F		1	2	-	-	-	-	0	1	0	0
CLRB	Clear Accumulator B	0 → B	B INH	5F		1	2	-	-	-	-	0	1	0	0
CLV	Clear Overflow Flag	0 → V	INH	0A		1	2	-	-	-	-	-	-	0	-
CMPA (opr)	Compare A to Memory	A − M	A IMM	81	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	91	dd	2	3								
			A EXT	B1	hh ll	3	4								
			A IND,X	A1	ff	2	4								
			A IND,Y	18 A1	ff	3	5								
CMPB (opr)	Compare B to Memory	B − M	B IMM	C1	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D1	dd	2	3								
			B EXT	F1	hh ll	3	4								
			B IND,X	E1	ff	2	4								
			B IND,Y	18 E1	ff	3	5								
COM (opr)	1's Complement Memory Byte	$FF − M → M	EXT	73	hh ll	3	6	-	-	-	-	↕	↕	0	1
			IND,X	63	ff	2	6								
			IND,Y	18 63	ff	3	7								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
COMA	1's Complement A	$FF – A → A	A INH	43		1	2	-	-	-	-	↕	↕	0	1
COMB	1's Complement B	$FF – B → B	B INH	53		1	2	-	-	-	-	↕	↕	0	1
CPD (opr)	Compare D to Memory 16-Bit	D – M:M + 1	IMM	1A 83	jj kk	4	5	-	-	-	-	↕	↕	↕	↕
			DIR	1A 93	dd	3	6								
			EXT	1A B3	hh ll	4	7								
			IND,X	1A A3	ff	3	7								
			IND,Y	CD A3	ff	3	7								
CPX (opr)	Compare X to Memory 16-Bit	IX – M:M + 1	IMM	8C	jj kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	9C	dd	2	5								
			EXT	BC	hh ll	3	6								
			IND,X	AC	ff	2	6								
			IND,Y	CD AC	ff	3	7								
CPY (opr)	Compare Y to Memory 16-Bit	IY – M:M + 1	IMM	18 8C	jj kk	4	5	-	-	-	-	↕	↕	↕	↕
			DIR	18 9C	dd	3	6								
			EXT	18 BC	hh ll	4	7								
			IND,X	1A AC	ff	3	7								
			IND,Y	18 AC	ff	3	7								
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19		1	2	-	-	-	-	↕	↕	↕	↕
DEC (opr)	Decrement Memory Byte	M – 1 → M	EXT	7A	hh ll	3	6	-	-	-	-	↕	↕	↕	-
			IND,X	6A	ff	2	6								
			IND,Y	18 6A	ff	3	7								
DECA	Decrement Accumulator A	A – 1 → A	A INH	4A		1	2	-	-	-	-	↕	↕	↕	-
DECB	Decrement Accumulator B	B – 1 → B	B INH	5A		1	2	-	-	-	-	↕	↕	↕	-
DES	Decrement Stack Pointer	SP – 1 → SP	INH	34		1	3	-	-	-	-	-	-	-	-
DEX	Decrement Index Register X	IX – 1 → IX	INH	09		1	3	-	-	-	-	-	↕	-	-
DEY	Decrement Index Register Y	IY – 1 → IY	INH	18 09		2	4	-	-	-	-	-	↕	-	-
EORA (opr)	Exclusive OR A with Memory	A ⊕ M → A	A IMM	88	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	98	dd	2	3								
			A EXT	B8	hh ll	3	4								
			A IND,X	A8	ff	2	4								
			A IND,Y	18 A8	ff	3	5								
EORB (opr)	Exclusive OR B with Memory	B ⊕ M → B	B IMM	C8	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D8	dd	2	3								
			B EXT	F8	hh ll	3	4								
			B IND,X	E8	ff	2	4								
			B IND,Y	18 E8	ff	3	5								
FDIV	Fractional Divide 16 by 16	D/IX → IX; r → D	INH	03		1	41	-	-	-	-	-	↕	↕	↕
IDIV	Integer Divide 16 by 16	D/IX → IX; r → D	INH	02		1	41	-	-	-	-	-	↕	0	↕
INC (opr)	Increment Memory Byte	M + 1 → M	EXT	7C	hh ll	3	6	-	-	-	-	↕	↕	↕	-
			IND,X	6C	ff	2	6								
			IND,Y	18 6C	ff	3	7								
INCA	Increment Accumulator A	A + 1 → A	A INH	4C		1	2	-	-	-	-	↕	↕	↕	-
INCB	Increment Accumulator B	B + 1 → B	B INH	5C		1	2	-	-	-	-	↕	↕	↕	-
INS	Increment Stack Pointer	SP + 1 → SP	INH	31		1	3	-	-	-	-	-	-	-	-
INX	Increment Index Register X	IX + 1 → IX	INH	08		1	3	-	-	-	-	-	↕	-	-
INY	Increment Index Register Y	IY + 1 → IY	INH	18 08		2	4	-	-	-	-	-	↕	-	-
JMP (opr)	Jump	See Special Ops	EXT	7E	hh ll	3	3	-	-	-	-	-	-	-	-
			IND,X	6E	ff	2	3								
			IND,Y	18 6E	ff	3	4								
JSR (opr)	Jump to Subroutine	See Special Ops	DIR	9D	dd	2	5	-	-	-	-	-	-	-	-
			EXT	BD	hh ll	3	6								
			IND,X	AD	ff	2	6								
			IND,Y	18 AD	ff	3	7								
LDAA (opr)	Load Accumulator A	M → A	A IMM	86	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	96	dd	2	3								
			A EXT	B6	hh ll	3	4								
			A IND,X	A6	ff	2	4								
			A IND,Y	18 A6	ff	3	5								
LDAB (opr)	Load Accumulator B	M → B	B IMM	C6	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	D6	dd	2	3								
			B EXT	F6	hh ll	3	4								
			B IND,X	E6	ff	2	4								
			B IND,Y	18 E6	ff	3	5								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
LDD (opr)	Load Double Accumulator D	M → A, M + 1 → B	IMM	CC	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	DC	dd	2	4								
			EXT	FC	hh ll	3	5								
			IND,X	EC	ff	2	5								
			IND,Y	18 EC	ff	3	6								
LDS (opr)	Load Stack Pointer	M:M + 1 → SP	IMM	8E	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	9E	dd	2	4								
			EXT	BE	hh ll	3	5								
			IND,X	AE	ff	2	5								
			IND,Y	18 AE	ff	3	6								
LDX (opr)	Load Index Register X	M:M + 1 → IX	IMM	CE	jj kk	3	3	-	-	-	-	↕	↕	0	-
			DIR	DE	dd	2	4								
			EXT	FE	hh ll	3	5								
			IND,X	EE	ff	2	5								
			IND,Y	CD EE	ff	3	6								
LDY (opr)	Load Index Register Y	M:M + 1 → IY	IMM	18 CE	jj kk	4	4	-	-	-	-	↕	↕	0	-
			DIR	18 DE	dd	3	5								
			EXT	18 FE	hh ll	4	6								
			IND,X	1A EE	ff	3	6								
			IND,Y	18 EE	ff	3	6								
LSL (opr)	Logical Shift Left	 C b7 b0	EXT	78	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	68	ff	2	6								
			IND,Y	18 68	ff	3	7								
LSLA			A INH	48		1	2								
LSLB			B INH	58		1	2								
LSLD	Logical Shift Left Double	 C b15 b0	INH	05		1	3	-	-	-	-	↕	↕	↕	↕
LSR (opr)	Logical Shift Right	 b7 b0 C	EXT	74	hh ll	3	6	-	-	-	-	0	↕	↕	↕
			IND,X	64	ff	2	6								
			IND,Y	18 64	ff	3	7								
LSRA			A INH	44		1	2								
LSRB			B INH	54		1	2								
LSRD	Logical Shift Right Double	 b15 b0 C	INH	04		1	3	-	-	-	-	0	↕	↕	↕
MUL	Multiply 8 by 8	A x B → D	INH	3D		1	10	-	-	-	-	-	-	-	↕
NEG (opr)	2's Complement Memory Byte	0 − M → M	EXT	70	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	60	ff	2	6								
			IND,Y	18 60	ff	3	7								
NEGA	2's Complement A	0 − A → A	A INH	40		1	2	-	-	-	-	↕	↕	↕	↕
NEGB	2's Complement B	0 − B → B	B INH	50		1	2	-	-	-	-	↕	↕	↕	↕
NOP	No Operation	No Operation	INH	01		1	2	-	-	-	-	-	-	-	-

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
ORAA (opr)	OR Accumulator A (Inclusive)	A + M → A	A IMM	8A	ii	2	2	-	-	-	-	↕	↕	0	-
			A DIR	9A	dd	2	3								
			A EXT	BA	hh ll	3	4								
			A IND,X	AA	ff	2	4								
			A IND,Y	18 AA	ff	3	5								
ORAB (opr)	OR Accumulator B (Inclusive)	B + M → B	B IMM	CA	ii	2	2	-	-	-	-	↕	↕	0	-
			B DIR	DA	dd	2	3								
			B EXT	FA	hh ll	3	4								
			B IND,X	EA	ff	2	4								
			B IND,Y	18 EA	ff	3	5								
PSHA	Push A onto Stack	A → Stk, SP = SP − 1	A INH	36		1	3	-	-	-	-	-	-	-	-
PSHB	Push B onto Stack	B → Stk, SP = SP − 1	B INH	37		1	3	-	-	-	-	-	-	-	-
PSHX	Push X onto Stack (Lo First)	IX → Stk, SP = SP − 2	INH	3C		1	4	-	-	-	-	-	-	-	-
PSHY	Push Y onto Stack (Lo First)	IY → Stk, SP = SP − 2	INH	18 3C		2	5	-	-	-	-	-	-	-	-
PULA	Pull A from Stack	SP = SP + 1, A ← Stk	A INH	32		1	4	-	-	-	-	-	-	-	-
PULB	Pull B from Stack	SP = SP + 1, B ← Stk	B INH	33		1	4	-	-	-	-	-	-	-	-
PULX	Pull X from Stack (Hi First)	SP = SP + 2, IX ← Stk	INH	38		1	5	-	-	-	-	-	-	-	-
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY ← Stk	INH	18 38		2	6	-	-	-	-	-	-	-	-
ROL (opr)	Rotate Left	 C b7 ← b0 C	EXT	79	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	69	ff	2	6								
			IND,Y	18 69	ff	3	7								
ROLA			A INH	49		1	2								
ROLB			B INH	59		1	2								
ROR (opr)	Rotate Right	 C b7 → b0 C	EXT	76	hh ll	3	6	-	-	-	-	↕	↕	↕	↕
			IND,X	66	ff	2	6								
			IND,Y	18 66	ff	3	7								
RORA			A INH	46		1	2								
RORB			B INH	56		1	2								
RTI	Return from Interrupt	See Special Ops	INH	3B		1	12	↕	↕	↕	↕	↕	↕	↕	↕
RTS	Return from Subroutine	See Special Ops	INH	39		1	5	-	-	-	-	-	-	-	-
SBA	Subtract B from A	A − B → A	INH	10		1	2	-	-	-	-	↕	↕	↕	↕
SBCA (opr)	Subtract with Carry from A	A − M − C → A	A IMM	82	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	92	dd	2	3								
			A EXT	B2	hh ll	3	4								
			A IND,X	A2	ff	2	4								
			A IND,Y	18 A2	ff	3	5								
SBCB (opr)	Subtract with Carry from B	B − M − C → B	B IMM	C2	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D2	dd	2	3								
			B EXT	F2	hh ll	3	4								
			B IND,X	E2	ff	2	4								
			B IND,Y	18 E2	ff	3	5								

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal) Opcode	Operand(s)	Bytes	Cycle	Condition Codes S	X	H	I	N	Z	V	C
SEC	Set Carry	1 → C	INH	0D		1	2	-	-	-	-	-	-	-	1
SEI	Set Interrupt Mask	1 → I	INH	0F		1	2	-	-	-	1	-	-	-	-
SEV	Set Overflow Flag	1 → V	INH	0B		1	2	-	-	-	-	-	-	1	-
STAA (opr)	Store Accumulator A	A → M	A DIR	97	dd	2	3	-	-	-	-	↕	↕	0	-
			A EXT	B7	hh ll	3	4								
			A IND,X	A7	ff	2	4								
			A IND,Y	18 A7	ff	3	5								
STAB (opr)	Store Accumulator B	B → M	B DIR	D7	dd	2	3	-	-	-	-	↕	↕	0	-
			B EXT	F7	hh ll	3	4								
			B IND,X	E7	ff	2	4								
			B IND,Y	18 E7	ff	3	5								
STD (opr)	Store Accumulator D	A → M, B → M + 1	DIR	DD	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	FD	hh ll	3	5								
			IND,X	ED	ff	2	5								
			IND,Y	18 ED	ff	3	6								
STOP	Stop Internal Clocks		INH	CF		1	2	-	-	-	-	-	-	-	-
STS (opr)	Store Stack Pointer	SP → M:M + 1	DIR	9F	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	BF	hh ll	3	5								
			IND,X	AF	ff	2	5								
			IND,Y	18 AF	ff	3	6								
STX (opr)	Store Index Register X	IX → M:M + 1	DIR	DF	dd	2	4	-	-	-	-	↕	↕	0	-
			EXT	FF	hh ll	3	5								
			IND,X	EF	ff	2	5								
			IND,Y	CD EF	ff	3	6								
STY (opr)	Store Index Register Y	IY → M:M + 1	DIR	18 DF	dd	3	5	-	-	-	-	↕	↕	0	-
			EXT	18 FF	hh ll	4	6								
			IND,X	1A EF	ff	3	6								
			IND,Y	18 EF	ff	3	6								
SUBA (opr)	Subtract Memory from A	A − M → A	A IMM	80	ii	2	2	-	-	-	-	↕	↕	↕	↕
			A DIR	90	dd	2	3								
			A EXT	B0	hh ll	3	4								
			A IND,X	A0	ff	2	4								
			A IND,Y	18 A0	ff	3	5								
SUBB (opr)	Subtract Memory from B	B − M → B	B IMM	C0	ii	2	2	-	-	-	-	↕	↕	↕	↕
			B DIR	D0	dd	2	3								
			B EXT	F0	hh ll	3	4								
			B IND,X	E0	ff	2	4								
			B IND,Y	18 E0	ff	3	5								
SUBD (opr)	Subtract Memory from D	D − M:M + 1 → D	IMM	83	jj kk	3	4	-	-	-	-	↕	↕	↕	↕
			DIR	93	dd	2	5								
			EXT	B3	hh ll	3	6								
			IND,X	A3	ff	2	6								
			IND,Y	18 A3	ff	3	7								
SWI	Software Interrupt	See Special Ops	INH	3F		1	14	-	-	-	1	-	-	-	-

NOTES:

Cycle:

* = Infinity or until reset occurs

** = 12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycle (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (total = 14 + n).

Operands:

dd = 8-bit direct address $0000-$00FF. (High byte assumed to be $00.)
ff = 8-bit positive offset $00 (0) to $FF (255) added to index.
hh = High order byte of 16-bit extended address.
ii = One byte of immediate data.
jj = High order byte of 16-bit immediate data.
kk = Low order byte of 16-bit immediate data.
ll = Low order byte of 16-bit extended address.
mm = 8-bit mask (set bits to be affected).
rr = Signed relative offset $80 (−128) to $7F (+127). Offset relative to the address following the machine code offset byte.

Condition Codes:

– Bit not changed
0 Always cleared (logic 0).
1 Always set (logic 1).
↕ Bit cleared or set depending on operation.
↓ Bit may be cleared, cannot become set.

REGISTER AND CONTROL BIT SUMMARY

	Bit 7	6	5	4	3	2	1	Bit 0	
$B000	Bit 7	-	-	-	-	-	-	Bit 0	PORTA
$B001									Reserved
$B002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
$B003	Bit 7	-	-	-	-	-	-	Bit 0	PORTC
$B004	Bit 7	-	-	-	-	-	-	Bit 0	PORTB
$B005	Bit 7	-	-	-	-	-	-	Bit 0	PORTCL
$B006									Reserved
$B007	Bit 7	-	-	-	-	-	-	Bit 0	DDRC
$B008			Bit 5	-	-	-	-	Bit 0	PORTD
$B009			Bit 5	-	-	-	-	Bit 0	DDRD
$B00A	Bit 7	-	-	-	-	-	-	Bit 0	PORTE
$B00B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC
$B00C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M
$B00D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D
$B00E	Bit 15	-	-	-	-	-	-	Bit 8	TCNT
$B00F	Bit 7	-	-	-	-	-	-	Bit 0	
$B010	Bit 15	-	-	-	-	-	-	Bit 8	TIC1
$B011	Bit 7	-	-	-	-	-	-	Bit 0	
$B012	Bit 15	-	-	-	-	-	-	Bit 8	TIC2
$B013	Bit 7	-	-	-	-	-	-	Bit 0	
$B014	Bit 15	-	-	-	-	-	-	Bit 8	TIC3
$B015	Bit 7	-	-	-	-	-	-	Bit 0	
$B016	Bit 15	-	-	-	-	-	-	Bit 8	TOC1
$B017	Bit 7	-	-	-	-	-	-	Bit 0	
$B018	Bit 15	-	-	-	-	-	-	Bit 8	TOC2
$B019	Bit 7	-	-	-	-	-	-	Bit 0	
$B01A	Bit 15	-	-	-	-	-	-	Bit 8	TOC3
$B01B	Bit 7	-	-	-	-	-	-	Bit 0	
$B01C	Bit 15	-	-	-	-	-	-	Bit 8	TOC4
$B01D	Bit 7	-	-	-	-	-	-	Bit 0	
$B01E	Bit 15	-	-	-	-	-	-	Bit 8	TOC5
$B01F	Bit 7	-	-	-	-	-	-	Bit 0	
	Bit 7	6	5	4	3	2	1	Bit 0	

REGISTER AND CONTROL BIT ASSIGNMENTS

	Bit 7	6	5	4	3	2	1	Bit 0	
$B020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$B021			EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$B022	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK1
$B023	OC1F	OC2F	OC3F	OC4F	OC5F	IC1F	IC2F	IC3F	TFLG1
$B024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2
$B025	TOF	RTIF	PAOVF	PAIF					TFLG2
$B026	DORA7	PAEN	PAMOD	PEDGE			RTR1	RTR0	PACTL
$B027	Bit 7	-	-	-	-	-	-	Bit 0	PACNT
$B028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$B029	SPIF	WCOL		MODF					SPSR
$B02A	Bit 7	-	-	-	-	-	-	Bit 0	SPDR
$B02B	TCLR		SCP1	SPC0	RCKB	SCR2	SCR1	SCR0	BAUD
$B02C	R8	T8		M	WAKE				SCCR1
$B02D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$B02E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR
$B02F	Bit 7	-	-	-	-	-	-	Bit 0	SCDR
$B030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL
$B031	Bit 7	-	-	-	-	-	-	Bit 0	ADR1
$B032	Bit 7	-	-	-	-	-	-	Bit 0	ADR2
$B033	Bit 7	-	-	-	-	-	-	Bit 0	ADR3
$B034	Bit 7	-	-	-	-	-	-	Bit 0	ADR4
$B035									Reserved
$B036									Reserved
$B037									Reserved
$B038									Reserved
$B039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION
$B03A	Bit 7	-	-	-	-	-	-	Bit 0	COPRST
$B03B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
$B03C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$B03D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$B03E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
$B03F					NOSEC	NOCOP	ROMON	EEON	CONFIG
	Bit 7	6	5	4	3	2	1	Bit 0	

USER NOTES

1. During a write to the output port latch (address B240-B27F), the latch is enabled before the data bus has stabilized. This will cause some of the bits to vary before the value written is latched in. This effect is not noticeable unless the output port is connected to the counter. Reads of the output port will write an FF hex to the port.

2. Using the EXM MEM (Examine Memory) command generates a write to the current address when it moves to the next address. If the current address was not modified, the old value will be written to the current address. The write is not noticeable when examining ROM or RAM, but when you examine the EEPROM in the ETC-128 Memory Module, the write causes the EEPROM to time out and initiate an internal write cycle. The contents of the EEPROM are not affected, but the display will show incorrect data for the previous byte on the display.

3. Before calling any Monitor ROM subroutine (such as GETC), save any register values that are needed after the call. The Monitor's subroutine may modify some register's value.

QUICK REFERENCE GUIDE

MEMORY I/O MAP

ADDRESS	MEMORY		
0000H 5FFFH	User Code Area		
6000H 6FFFH	User Data and Stack Area NOTE: This is where the user stack should start.		
6900H 69FFH	Monitor Data and Stack Area		
7000H AFFFH	Memory Module Area		
	NOTE: This is the Internal I/O Area.		
B000H	PORTA	Port A Data Register	
B001H	Reserved		
B002H	PIOC	Parallel I/O Control Register	
B003H	PORTC	Port C Data Register	
B004H	PORTB	Port B Data Register	
B005H	PORTCL	Port C Latched Data Register	
B006H	Reserved		
B007H	DDRC	Data Direction Register for Port C	
B008H	PORTD	Port D Data Register	
B009H	DDRD	Data Direction Register for Port D	
B00AH	PORTE	Port E Data Register	
B00BH	CFORC	Time Compare Force Register	
B00CH	OC1M	Output Compare 1 Mask Register	
B00DH	OC1D	Output Compare 1 Data Register	
B00EH, B00FH	TCNT	Timer Counter Register	
B010H, B011H	TIC1	Timer Input Capture Register 1	
B012H, B013H	TIC2	Timer Input Capture Register 2	
B014H, B015H	TIC3	Timer Input Capture Register 3	
B016H, B017H	TOC1	Timer Output Compare Register 1	
B018H, B019H	TOC2	Timer Output Compare Register 2	
B01AH, B01BH	TOC3	Timer Output Compare Register 3	
B01CH, B01DH	TOC4	Timer Output Compare Register 4	
B01EH, B01FH	TOC5	Timer Output Compare Register 5	
B020H	TCTL1	Timer Control Register 1	
B021H	TCTL2	Timer Control Register 2	
B022H	TMSK1	Main Timer Interrupt Mask Reg. 1	
B023H	TFLG1	Main Timer Interrupt Flag Reg. 1	
B024H	TMSK2	Misc. Timer Interrupt Mask Reg. 2	
B025H	TFLG2	Misc. Timer Interrupt Flag Reg. 2	
B026H	PACTL	Pulse Accumulator Control Register	
B027H	PACNT	Pulse Accumulator Count Register	
B028H	SPCR	SPI Control Register	
B029H	SPSR	SPI Status Register	
B02AH	SPDR	SPI Data Register	
B02BH	BAUD	SCI Baud Rate Control Register	
B02CH	SCCR1	SCI Control Register 1	
B02DH	SCCR2	SCI Control Register 2	
B02EH	SCSR	SCI Status Register	
B02FH	SCDR	SCI Data Register	
B030H	ADCTL	A/D Control/Status Register	
B031H	ADR1	A/D Result Register 1	
B032H	ADR2	A/D Result Register 2	
B033H	ADR3	A/D Result Register 3	
B034H	ADR4	A/D Result Register 4	
B035H	Reserved		
B036H	Reserved		
B037H	Reserved		
B038H	Reserved		
B039H	OPTION	System Configuration Options	
B03AH	COPRST	Arm/Reset COP Timer Circuitry	
B03BH	PPROG	EEPROM Programming Register	
B03CH	HPRIO	Highest Priority Interrupt and Misc.	
B03DH	INIT	RAM and I/O Mapping Register	
B03EH	TEST1	Factory Test Register	
B03FH	CONFIG	Configuration Control Register	
B218H	Keyboard Column 3		NOTE: Keyboard Columns are not fully decoded. Use only the address specified.
B228H	Keyboard Column 2		
B230H	Keyboard Column 1		
B240H B27FH	Output Port		
B280H B2BFH	Input Port		
B2C0H B2FFH	LCD Registers		NOTE: B2C0H is a Command Register and B2C1H is a Data Register. The Remaining Registers are uncoded.
B300H B33FH	General I/O 0		
B340H B37FH	General I/O 1		
B380H B3BFH	General I/O 2		
B3C0H B3FFH	General I/O 3		
B400H BFFFH	Unused		
C000H FFFFH	Monitor Rom Area		

SUBROUTINE JUMP TABLE

EPROM ADDRESS	LABEL	INTRODUCTION
C000	JMP MAIN	Return Control to Monitor
C003	JMP GETC	Returns ASCII Character in B, A=0
C006	JMP PUTC	Outputs ASCII Character in B
C009	JMP SETRS232	Select RS-232 for I/O
C00C	JMP SETLCD	Select LCD, Keypad for I/O
C00F	JMP GETHBYT	Get a Hex Byte, Return in B
C012	JMP GETHWRD	Get a Hex Word, Return in D
C015	JMP PUTHBYT	Print Byte in B, Out in Hex
C018	JMP PUTHWRD	Print Word in D, Out in Hex
C01B	JMP PUTSTRING	Print a Null Terminated String Pointed to by D
C01E	JMP COPYUP	Copy Block of Memory, B=# Copy, Push Source, Push Destination
C021	JMP COPYDOWN	Copy Block of Memory, B=# Copy, Push Source, Push Destination
C024	JMP COPY	Push Incordec, # of Bytes, Destination, D=Source
C027	JMP PAUSE	Wait Over a Second
C02A	JMP PUTDSWRD	Convert Signed Value in D to Decimal and Print Out
C02D	JMP PUTUDWRD	Convert Unsigned Value in D to Decimal and Print Out

INTERRUPT VECTOR ASSIGNMENTS

Vector Address	Interrupt Source	CC Register Mask	Local Mask	
6901H, 2H	SCI Serial System	1 Bit	See Table	
6904H, 5H	SPI Serail Transfer Complete	1 Bit	SPIE	LOWEST
6907H, 8H	Pulse Accumulator Input Edge	1 Bit	PAII	
690AH, BH	Pulse Accumulator Overflow	1 Bit	PAOVI	
690DH, EH	Timer Overflow	1 Bit	TOI	
6910H, 1H	Timer Output Compare 5	1 Bit	OC5I	
6913H, 4H	Timer Output Compare 4	1 Bit	OC4I	
6916H, 7H	Timer Output Compare 3	1 Bit	OC3I	
6919H, AH	Timer Output Compare 2	1 Bit	OC2I	
691CH, DH	Timer Output Compare 1	1 Bit	OC1I	RELATIVE PRIORITY
691FH, 0H	Timer Input Capture 3	1 Bit	IC3I	
6922H, 3H	Timer Input Capture 2	1 Bit	IC2I	
6925H, 6H	Timer Input Capture 1	1 Bit	IC1I	
6928H, 9H	Real Time Interrupt	1 Bit	RTII	
692BH, CH	IRQ (External Pin or Parallel I/O)	1 Bit	See Table	
SEE HPRIO REGISTER FOR HIGHEST PRIORITY I-BIT SOURCE				
692EH, FH	XIRQ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None	
6931H, 2H	SWI	None	None	
6934H, 5H	Illegal Opcode Trap	None	None	
6937H, 8H	COP Failure (Reset)	None	NOCOP	
693AH, BH	COP Clock Monitor Fail (Reset)	None	CME	HIGHEST

SCI Serial System Interrupts

Interrupt Cause	Local Mask
Receive Data Register Full	RIE
Receiver Overrun	RIE
Idle Line Detect	ILIE
Transmit Data Register Empty	TIE
Transmit Complete	TCIE

IRQ Vector Interrupts

Interrupt Cause	Local Mask
External Pin	None
Parallel I/O Handshake	STAI

KEY NAMES/FUNCTIONS

RETURN: Exit a function.

E-BAUD: Enter letter "E" or change baud rate.

D-RS232: Enter letter "D" or transfer control to serial port.

A-LOAD: Enter letter "A" or load program at selected address.

7-BRK: Enter numeral "7" or permit entry of break points.

8-W REG: Enter numeral "8" or set register break value.

4-M BLK: Enter numeral "4" or move a block of memory.

1-EXM MEM: Enter numeral "1" or display memory value.

0-SS: Enter numeral "0" or single step a program.

2-EXM REG: Enter numeral "2" or examine and modify CPU registers.

NMI: Interrupt any program or operation.

RESET: Reset system.

3-GO: Enter numeral "3" or execute a program.

6-DOWN: Enter numeral "6" or download a file to RAM.

5-I BLK: Enter numeral "5" or set a block of memory to a value.

9-W LOC: Enter numeral "9" or set address break value.

C-DUP: Enter letter "C" or copy memory cartridge.

B-SAVE: Enter letter "B" or save file to Memory Module.

SIGNAL CONNECTOR BLOCKS

UPPER BLOCK

D0	CPU Data Bus
D1	CPU Data Bus
D2	CPU Data Bus
D3	CPU Data Bus
D4	CPU Data Bus
D5	CPU Data Bus
D6	CPU Data Bus
D7	CPU Data Bus
32KEN	Lower 20K RAM Disable
READ	Memory Read
WRITE	Memory Write
XIRQ	Non-Maskable Interrupt
RESET	Reset Input
AS	Address Strobe
LIR	Inverted CPU LIR
A15	CPU Address Bus
A14	CPU Address Bus
A13	CPU Address Bus
A12	CPU Address Bus
A11	CPU Address Bus
A10	CPU Address Bus
A9	CPU Address Bus
A8	CPU Address Bus
A7	CPU Address Bus
A6	CPU Address Bus
A5	CPU Address Bus
A4	CPU Address Bus
A3	CPU Address Bus
A2	CPU Address Bus
A1	CPU Address Bus
A0	CPU Address Bus
IRQ	CPU Interrupt
E	CPU E Clock
R/W	CPU R/W
I/O 0	Decoded I/O
I/O 1	Decoded I/O
I/O 2	Decoded I/O
I/O 3	Decoded I/O
COP	CPU Computer Operating Properly Output
EMS	External Memory Select

LOWER BLOCK

LPIN	Logic Probe Input
LPAUDC	Logic Probe Audio Disable
GND	Digital Ground
+5V	+5 Volt DC Output
AGND	Analog Ground
-12V	-12 Volt DC Output
+12V	+12 Volt DC Output
AI7	Analog Input
AI6	Analog Input
AI5	Analog Input
AI4	Analog Input
AI3	Analog Input
AI2	Analog Input
AI1	Analog Input
AI0	Analog Input
60Hz	Output Compare 2
1Hz	Output Compare 3
OC4	Output Compare 4
OC5	Output Compare 5
PAI	Pulse Accumulator Input
IC1	Input Capture 1
IC2	Input Capture 2
IC3	Input Capture 3
MOUT	SPI Data Output
MIN	SPI Data Input
SCLK	SPI Clock
SS	SPI Slave Select
IPL	Input Port Latch
IP7	Input Data Port
IP6	Input Data Port
IP5	Input Data Port
IP4	Input Data Port
IP3	Input Data Port
IP2	Input Data Port
IP1	Input Data Port
IP0	Input Data Port
OP7	Output Data Port
OP6	Output Data Port
OP5	Output Data Port
OP4	Output Data Port
OP3	Output Data Port
OP2	Output Data Port
OP1	Output Data Port
OP0	Output Data Port
GND	Digital Ground

NOTE:
Red areas are output only. Blue areas are input only. Green areas are bi-directional.

BACKPACK CONNECTOR

1- Analog Input 0
2- Analog Input 1
3- Output Compare 3 (1Hz)
4- Output Compare 2 (60Hz)
5- Output Compare 5
6- Output Compare 4
7- Input Capture 1
8- Pulse Accumulator Input
9- Ground
10- Input Capture 2
11- Ground
12- SPI Data Out
13- Ground
14- SPI Data In
15- Ground
16- SPI Clock
17- Ground
18- SPI Slave Select
19- Ground
20- CPU Address 5
21- Ground
22- CPU Address 4
23- Ground
24- CPU Address 3
25- Ground
26- CPU Address 2
27- Ground
28- CPU Address 1
29- Ground
30- CPU Address 0
31- Ground
32- IRQ
33- Ground
34- E Clock
35- Ground
36- CPU R/W Line
37- Ground
38- General I/O 0
39- Ground
40- General I/O 1
41- Ground
42- General I/O 2
43- Ground
44- General I/O 3
45- Ground
46- CPU Data 0
47- Ground
48- CPU Data 1
49- Ground
50- CPU Data 2
51- Ground
52- CPU Data 3
53- Ground
54- CPU Data 4
55- Ground
56- CPU Data 5
57- Ground
58- CPU Data 6
59- Ground
60- CPU Data 7

NOTE:
All signal lines are protected by series resistors.

CUSTOMER SERVICE

REPLACEMENT PARTS

Please provide complete information when you request replacements from either the factory or Heath/Zenith Computers and Electronics centers. Be certain to include the **HEATH** part number exactly as it appears in the parts list.

ORDERING FROM THE FACTORY

Print all of the information requested on the parts order form furnished with this product and mail it to Heath. For telephone orders (parts only) dial 616 982-3571. If you are unable to locate an order form, write us a letter or card including:

- Heath part number.
- Model number.
- Date of purchase.
- Location purchased or invoice number.
- Nature of the defect.
- Your payment or authorization for COD shipment of parts not covered by warranty.

Mail letters to: Heath Company
Benton Harbor
MI 49022
Attn: Parts Replacement

Retain original parts until you receive replacements. Parts that should be returned to the factory will be listed on your packing slip.

OBTAINING REPLACEMENTS FROM HEATH/ZENITH COMPUTER AND ELECTRONICS CENTERS

For your convenience, "over the counter" replacement parts are available from the Heath/Zenith Computer and Electronics centers listed in your catalog. Be sure to bring in the original part and purchase invoice when you request a warranty replacement from a Heath/Zenith Computer and Electronics center.

TECHNICAL CONSULTATION

Need help with your kit? — Self-Service? — Construction? – Operation? — Call or write for assistance. You'll find our Technical Consultants eager to help with just about any technical problem except "customizing" for unique applications.

The effectiveness of our consultation service depends on the information you furnish. Be sure to tell us:

- The Model number and Series number from the blue and white label.
- The date of purchase.
- An exact description of the difficulty.
- Everything you have done in attempting to correct the problem.

Also include switch positions, connections to other units, operating procedures, voltage readings, and any other information you think might be helpful.

Please do not send parts for testing, unless this is specifically requested by our Consultants.

Hints: Telephone traffic is lightest at midweek — please be sure your Manual and notes are on hand when you call.

Heath/Zenith Computer and Electronics center facilities are also available for telephone or "walk-in" personal assistance.

REPAIR SERVICE

Service facilities are available, if they are needed, to repair your completed kit. (Kits that have been modified, soldered with paste flux or acid core solder, cannot be accepted for repair.)

If it is convenient, personally deliver your kit to a Heath/Zenith Computers and Electronics center. For warranty parts replacement, supply a copy of the invoice or sales slip.

If you prefer to ship your kit to the factory, attach a letter containing the following information directly to the unit:

- Your name and address.
- Date of purchase and invoice number.
- Copies of all correspondence relevant to the service of the kit.
- A brief description of the difficulty.
- Authorization to return your kit COD for the service and shipping charges. (This will reduce the possibility of delay.)

Check the equipment to see that all screws and parts are secured. (Do not include any wooden cabinets or color television picture tubes, as these are easily damaged in shipment. Do not include the kit Manual.) Place the equipment in a strong carton with at least THREE INCHES of *resilient* packing material (shredded paper, excelsior, etc.) on all sides. Use additional packing material where there are protrusions (control sticks, large knobs, etc.). If the unit weighs over 15 lbs., place this carton in another one with 3/4" of packing material between the two.

Seal the carton with reinforced gummed tape, tie it with a strong cord, and mark it "Fragile" on at least two sides. Remember, the carrier will not accept liability for shipping damage if the unit is insufficiently packed. Ship by prepaid express, United Parcel Service, or insured Parcel Post to:

Heath Company
Service Department
Benton Harbor, Michigan 49022

Heath Company
Benton Harbor, Michigan