HERO-1 BASIC

ET-18-9

Installation/Operation

595-3268
HEATH COMPANY
BENTON HARBOR, MICHIGAN 49022

TABLE OF CONTENTS

INTRODUCTION

This Manual tells you how to install and program HERO-1 BASIC in your Robot. The "Operation" section contains references to and examples of statements, functions, and commands that HERO-1 BASIC recognizes to help you write your own Robot programs. These examples are separate individual illustrations, and are not specific parts of any overall program.

HERO-1 BASIC is an adaptation by Wintek Corporation of Micro Basic Plus by Technical Systems Consultants. The program is a simple integer BASIC with modifications that allow you to use the Robot's Voice Synthesizer and its various motors and sensors within its programs.

HERO-1 BASIC is designed to work with a terminal. However, you can use any computer running a Terminal Emulation program, such as CPS.

You will need the following hardware for operation of HERO-1 BASIC:

ET/ETW-18 Robot.
ET-18-6 Memory Expansion Accessory.
6264 CMOS RAM (ETA-18-6 or equivalent).
ETW-18-10 Serial Interface Accessory with 1.3 or 1.U firmware.
Terminal (or a computer with a terminal emulation program).
Serial interface cable (HCA-10, HCA-11 or equivalent).

Only one 6264 CMOS RAM is required. However, you may use up to three so you can have enough memory for long, sophisticated programs, or large dimensioned variables.

PARTS LIST

To order a replacement part, always include the PART NUMBER. Use the Parts Order Form furnished with this kit. If a Parts Order Form is not available, refer to "Replacement Parts" on the last page of this Manual. For prices, refer to the separate "Heath Parts Price List."

HEATH Part No.	QTY.	DESCRIPTION
444-319	1	HERO 1 BASIC ROM
597-260	1	Parts Order Form
	1	Operation Manual (see Page 1 for part number.)

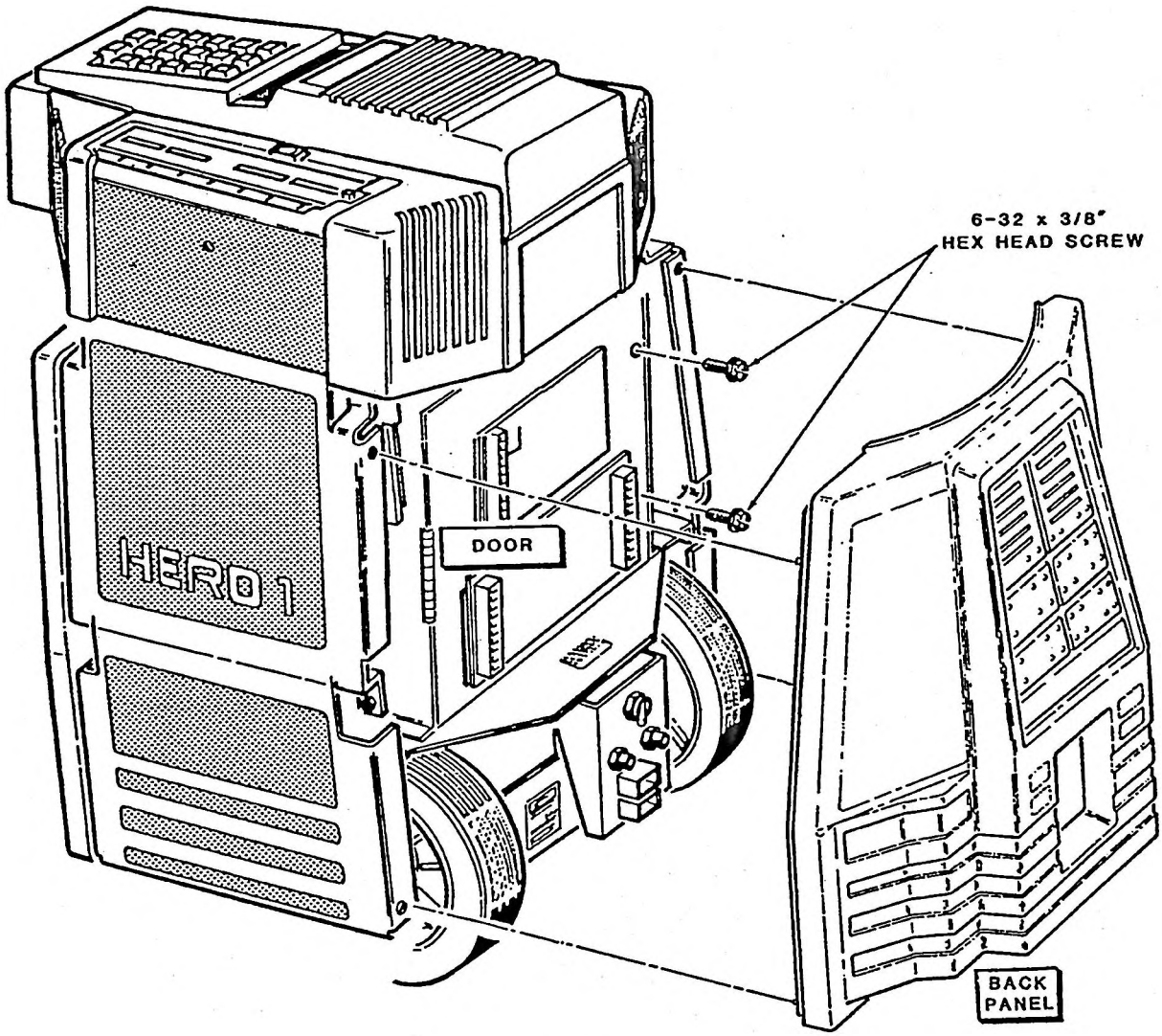
INSTALLATION

Refer to Pictorial 1 for the following steps.

( ) Turn the Robot's POWER switch to the OFF position.

( ) Remove the Robot's rear panel and set it aside.

( ) Remove the two screws that hold the door closed and set them aside.



PICTORIAL 1

Refer to Pictorial 2 (fold-out from this Page) for the following steps.

( ) If there is an IC installed at location U107 on the memory expansion board, carefully remove it and set it aside.

NOTE: Use the "IC Installation Procedure" on Page 5 to install the protected HERO-1 BASIC ROM (#444-319) in its socket at U107 on the memory expansion board. Save the conductive foam pad.

CAUTION: The IC you will install in the following step is an MOS (Metal Oxide Semiconductor) device. Be sure it does not get damaged by static electricity. Once you remove the foam pad from the IC, DO NOT let go of the IC. Install it as follows. Read the entire step before you pick up the IC.

( ) U107: Install the protected HERO-1 BASIC ROM (#444-319) in its socket at U107 on the memory expansion board. Make sure you install pins 1 and 24 ROM into socket holes 3 and 26. Save the conductive foam pad for use in the next step.

( ) If you removed an IC from U107 previously, push the pins of the IC into the conductive foam pad from the previous step. Set this IC aside in a safe place.

( ) Position jumper sockets J121, J122, J123, and J124 as shown in Pictorial 2.

( ) Turn the Robot's POWER switch to the ON position. The Robot's display should read "HEro1.X" (X indicates the revision level of the Monitor). The Robot should also say "READY" if you have the Voice Accessory installed.

( ) Turn the POWER switch to the OFF position.

NOTE: If you have purchased a 6264 RAM to install with the ET-18-9, install it at U104 on the memory expansion board in the following six numbered steps. If you have already installed a 6264 RAM at this location, skip the numbered steps and proceed to the step after step #6.

( ) 1. If one is installed, remove the IC at location U104 of the memory expansion board and set it aside.

NOTE: Use the same procedure for installing the protected 6264 RAM as you did for the previous protected IC.

( ) 2. Remove the 6264 RAM from its protective foam pad. Straighten any bent pins and insert it in the socket at location U104 of the memory expansion board. Make sure the pin 1 end of the IC is positioned correctly.

( ) 3. Position the jumper sockets J109, J110, J111, and J112 for U104 as shown in Pictorial 2.

NOTE: If you have additional 6246 RAM, perform the next two steps. Otherwise, proceed to step #6.

( ) 4. If you have any more 6246 RAM, install them at U105 and U106 at this time. Make sure you position the pin 1 end properly.

( ) 5. If you are installing ICs at U105 and U106, install the jumper sockets at J113, J114, J115, and J116 for U105, and J117, J118, J119, and J120 for U106 as shown in Pictorial 2.

( ) 6. Turn the Robot's POWER switch to the ON position. The Robot's display should read "HEro1.X". The Robot should also say "READY" if the Voice Accessory is installed. Turn the POWER switch to the OFF position.

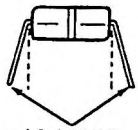
( ) Close the door and secure it with the screws that you set aside earlier.

Heathkit®

IC INSTALLATION PROCEDURE

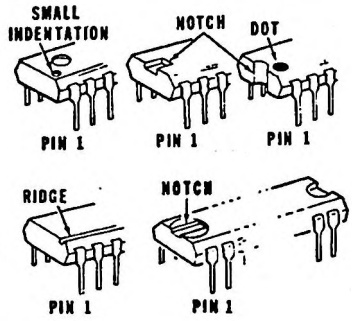
Once you remove a protected IC from its protective foam packing, DO NOT lay the IC down or let go of it until it is installed in its socket. When you bend the leads of a protected IC, hold the IC in one hand and place your other hand on your work surface before you touch the IC to your work surface. This will equalize the static electricity between the work surface and the IC.

The pins on the IC's may be bent out at an angle, so they do not line up with the holes in the IC socket. DO NOT try to install an IC without first bending the pins as described below. To do so may damage the IC pins or the socket, causing intermittent contact.

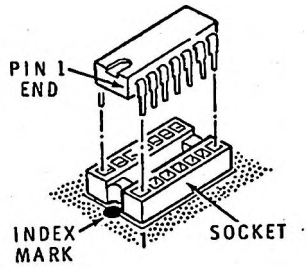


Before you install an IC, lay it down on its side as shown below and very carefully roll it toward the pins to bend the lower pins into line. Then turn the IC over and bend the pins on the other side in the same manner.

Compare the IC to the drawing shown below. Then determine which end of the IC is the pin 1 end.



Position the pin 1 end of the IC over the index mark on the circuit board. Then start the IC pins into the socket. Make sure that all of the pins are started into the socket. Then push the IC firmly into the socket. NOTE: An IC pin can become bent under the IC and it will appear as though it is correctly installed in the socket.



Refer to Pictorial 3 for the following steps.

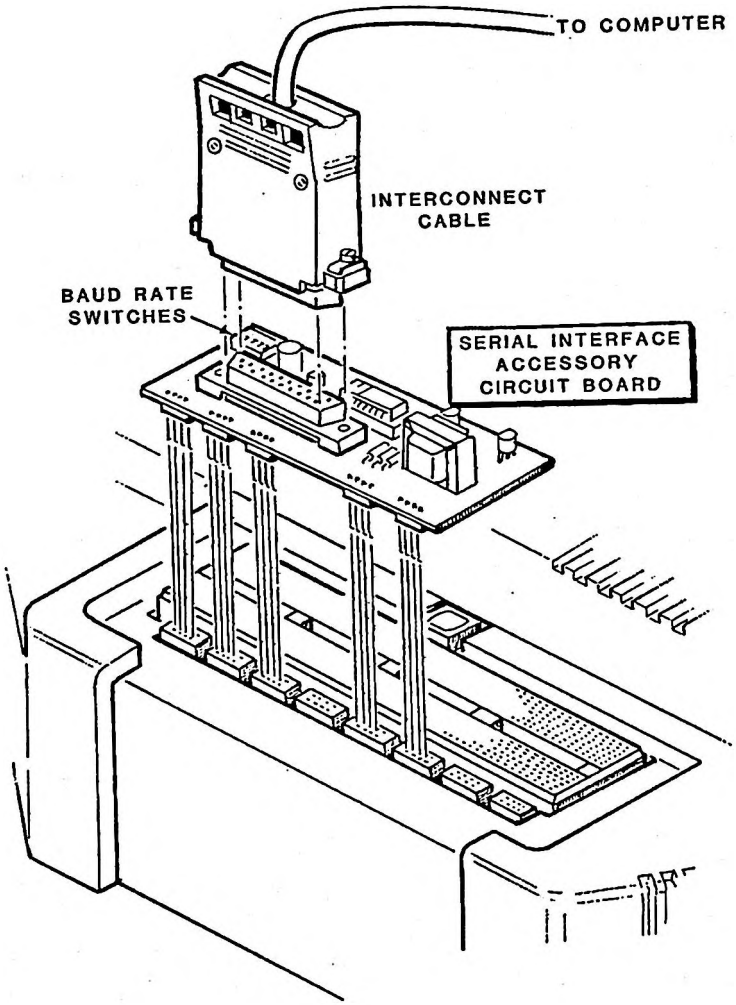
( ) Install the Serial Interface Accessory circuit board plugs into the indicated breadboard holes.

( ) Connect a suitable cable between the Serial Interface Accessory circuit board connector and your terminal or computer.

( ) Set the baud rate switches on the Serial Interface Accessory circuit board for a baud rate to match that of your terminal or computer.

( ) Refer back to Pictorial 1 and reinstall the Robot's rear panel.

This completes the "Installation."



PICTORIAL 3

OPERATION

OPERATING MODES

The HERO 1 Robot is a computer on wheels. HERO-1 BASIC is a program that uses the keyboard of your terminal to send information, in the form of characters, to the Robot. HERO-1 BASIC also uses the terminal screen to display computed information. If you are using a computer with terminal emulation software, your computer acts as the terminal. It is important to remember that it is the Robot that makes all of the computations.

HERO-1 BASIC allows you two modes of operation; Immediate and Running which are described below. When you are in the Immediate Mode, instructions are interpreted and executed as you enter them from the computer or terminal keyboard. Whenever you are in the Immediate Mode, you will see a prompt character (>) displayed on the screen. This means that the computer or terminal is ready for a command. When you enter the Running Mode, all instructions to the computer or terminal are interpreted and executed in the ascending numerical order of their line numbers.

Turn the Robot and terminal's power on. If you are using a computer instead of a terminal, turn the computer's power on and run your Terminal Emulation program.

There are three entry points to HERO-1 BASIC: "Cold Start," "Warm Start," and "Running Start."

The "COLD START" is used when you first turn the computer or terminal on. It clears the work area (the portion of the Robot's memory where user programs are stored), resets all variables, and displays the Immediate Mode prompt (>).

To perform a "COLD START", press the following keys on the Robot's keyboard:

1	(To select Machine Mode)
D	(To "Do")
A000	(Address of COLD START)

The following title banner is displayed on the screen:

```
Wintek HERO-1 BASIC – VERSION X.X
```

NOTE: The X.X after the "version" indicates the revision level of HERO-1 BASIC.

A "WARM START" will preserve the program in the work area, clear all the variables, and display the Immediate Mode prompt. Use this entry point to enter the Immediate Mode when you already have a program in the work area. To perform a "WARM START", press the following keys on the Robot's keyboard:

1	(To select Machine Mode)
D	(To "Do")
A003	(Address of WARM START)

The screen should show the Immediate Mode prompt:

```
>
```

HERO 1 BASIC is now ready for a command. For example, if you type LIST, the program in the work area will be displayed on the screen.

When you use the "RUNNING START", the program already in the work area is executed. For example, a program may be written in BASIC and saved on cassette. Later, you can load it from the cassette and execute it from the Robot's keyboard without having to use the terminal or computer to enter and execute it.

To perform a "RUNNING START", press the following keys on the Robot's keyboard:

1	(To select Machine Mode)
D	(To "Do")
A006	(Address of RUNNING START)

Keywords

Keywords are the instructions (GOTO, PRINT, etc.) that are recognized and interpreted by HERO-1 BASIC. You need only to enter the first three letters of a keyword to be recognized (REM, DIM, PRI, etc.).

Statements

A statement in HERO-1 BASIC consists of a keyword and any associated expressions or operations. If a statement is entered without a line number, BASIC automatically executes the statement at once. If the statement is entered with a line number, it will be saved in the work area for execution later.

You can place multiple statements on numbered lines by separating them with a colon (:). You cannot make multiple statements in the Immediate Mode.

Statements may be up to 71 characters long (including line numbers and spaces). If you enter a 72nd character, it is echoed, but the entire line is discarded and the Immediate Mode prompt is issued.

Numbered Lines

Line numbers must be within the range of 0 to 9999. You may enter them in any order. HERO-1 BASIC will automatically put them in ascending numerical order. When you are writing a program in BASIC, you should number your lines in multiples of ten or so, in case you find it necessary to insert an instruction later. Format HERO-1 BASIC lines as follows:

```
nnnn <stat>[:<stat>[:etc...]]

10 LET A = 5:PRINT A
```

The line number is specified by nnnn. You must begin line numbers in the first column of the line and follow them by one or more spaces. HERO-1 BASIC "statements" are indicated by <stat>. Additional statements, as shown inside the brackets [ ], are optional.

If you wish to erase the line presently being typed, press the DELETE or RUBOUT key. If you wish to erase characters on the line presently being typed, press the BACKSPACE key or CTRL-H. (NOTE: Even though the characters are still displayed on the screen, they have been erased from the Robot's memory). If you wish to erase a line previously entered, type the line number and press RETURN.

Constants

Constants are actual numbers specified by the programmer. They may be in decimal (base 10) or hexadecimal (base 16) format. Decimal format numbers use the digits 0 through 9, while hexadecimal format numbers also use the letters A through F to represent values 10 through 15. HERO-1 BASIC will accept numbers as decimal unless you preceed them with a "$" symbol as in the example below.

```
64075     (Decimal)
$FA4B     (Hexadecimal)
```

Constants must be integers within the range of +99999 (+$1869F) to − 99999 (-$1869F). They cannot have decimal points, commas, or imbedded spaces. If the result of a calculation exceeds these limits, an error will be reported. Since HERO-1 BASIC deals only with integers, any fractional portion of a number is disregarded.

```
99  / 100 =0
101 / 100 =1
```

Variables

Variables are represented by the letters A-Z and may have up to two dimensioned subscripts (see "DIMENSION" on Pages 10 and 11). The value may be assigned by a constant or with the value of an expression.

Expressions

Expressions consist of constants or variables to be evaluated. They may be accompanied by arithmetic operators, (+) for addition, (−) for subtraction, (*) for multiplication, (/) for division, (↑) for exponentiation, or special functions (refer to "SGN" and "ABS" on Page 13) NOTE: Division by 0 is not permitted. If HERO-1 BASIC encounters such an operation, it will report an error.

Expressions are evaluated from left to right with all operators having equal precedence. Remember: When in doubt, group subexpressions with parentheses.

The following are examples of expressions.

```
15/3
A^B
3+3*2       (evaluates to 12)
3+(3*2)     (evaluates to 9)
```

Relations

A relation compares the value of two expressions. The result of the comparison will be either TRUE or FALSE. The following table defines the relational operators.

SYMBOL	EXAMPLE	MEANING
=	A = B	A is equal to B
<	A < B	A is less than B
>	A > B	A is greater than B
<=	A <= B	A is less than or equal to B
>=	A >= B	A is greater than or equal to B
<>	A <> B	A is not equal to B

BASIC STATEMENTS AND FUNCTIONS

SCRATCH is used to clear the current user's program from the work area and reset all variables. It is normally used in the Immediate Mode but it can also appear on a numbered line.

LIST is used to display all or some of the lines of the program in the work area. List has several different forms:

LIST	Displays the entire program
LIST nnnn	Displays line number "nnnn"
LIST nnnn,	Displays all lines starting at "nnnn"
LIST nnnn, X	Displays X lines starting at "nnnn"

Although it is used primarily in the Immediate Mode, LIST may appear on a numbered line. However, after execution of LIST, the program will be suspended and control will return to the Immediate Mode.

RUN is used to execute the program in the work area. Execution always begins with the lowest numbered line.

The **END** statement suspends program execution whenever it is encountered. Program control returns to the Immediate Mode.

NOTE: The BREAK key on most terminals normally sends the ASCII NUL character (see "ASCII To Decimal and Hexidecimal Conversion Chart on Page 16). Refer to your terminal manual for the procedure for sending a NUL.

Use the BREAK key to suspend program execution or LIST at any time. This causes an ATTENTION INTERRUPT (Error number 99), which returns control to the Immediate Mode and causes the prompt to be issued.

"BREAK" is checked between statements and at INPUT prompts. Since some statements which involve Robot functions take a while to execute, it may be necessary to hold the "BREAK" key for a time. Also, on some terminals or computers, it may be necessary to hold the SHIFT key down along with the BREAK key.

MONITOR is used to return to the Executive ("ready") Mode of the HEro1.X Monitor.

Type **PUNCH** to save the program in the work area. The program is converted to Motorola's S1/S9 format and sent to the terminal or computer for storage. Similarly, **CPUNCH** sends the data to the cassette.

Type **LOAD** to receive a program saved in Motorola's S1/S9 format by PUNCH from the terminal or computer. Similarly, **CLOAD** receives a program from the cassette.

Use the **DIMENSION** statement to establish the size of an array. All subscripted variables must appear in a DIMENSION statement before they are referred to by any other statement.

The format of a DIMENSION statement is shown in the following example. The variable to be dimensioned is specified by <var>. The size is specified by <constant1> and the optional <constant2>. Each of these constants must be in decimal form within the range of 0 to 98. Variables may be as large as available memory will permit. Dimensioned variables cannot be used as the index variable specified in a FOR...NEXT statement.

```
DIMENSION <var>(<constant1>[,<constant2>])
```

When more than one variable is specified in a DIMENSION statement, they must be separated by commas as shown in the following statement.

```
DIMENSION A(2,3), B(18), C(0)
```

LET is used to assign a value to a variable. The word "LET" is optional. The first line of the example below shows the format of the statement. The variable specified by <var> is assigned the value of the expression specified by <expr>. The second assigns the variable A the value 15. The third assigns the variable B the value 64.

```
LET <var> = <expr>
LET A = 3*5
B = $40
```

The **REMARK** statement is used to put comments in your program. These statements are ignored by HERO-1 BASIC.

```
10 REM - THIS IS AN EXAMPLE OF A REMARK STATEMENT
```

The **IF...THEN** statement is used to perform a test and then conditionally execute a statement or transfer execution to a specified line according to the result. The form is shown in the example below where <expr1> and <expr2> are expressions, <rel op> is a relational operator, <stat> is a statement, and <nnnn> specifies a line number. For example:

```
10 IF A <10 THEN 30:ELSE A=10
IF<expr1><rel op><expr2> THEN <stat or nnnn> [:ELSE<stat>]
```

The relational operator compares the values and returns the condition. If the comparison is true, the statement following THEN is executed. If a line number was specified, execution is transfered to that line. If the comparison returns a false condition, then the statement or line number is ignored and the optional **ELSE** statement is executed. ELSE is executed if, and only if, the comparison is false and it appears directly behind the IF...THEN statement on the same line separated by a colon. When found anywhere else, it is ignored. If the ELSE statement is omitted, control transfers to the next highest numbered line.

With **FOR...NEXT** statements, a program loop may be executed a specified number of times. In the example that follows, <var> is any variable A through Z without a subscript. <expr1>, <expr2> and <expr3> are expressions. <expr2> must not contain any reference to the variable specified by <var>.

Before the loop is executed, all expressions in the statement are evaluated. <var> is assigned the value of <expr1>. The program then executes until the NEXT statement is encountered. The value of <expr3> is added to <var>. If <var> does not exceed <expr2>, the loop is repeated. Otherwise, execution continues on the line following the NEXT statement. The variable in the NEXT statement must be the same variable specified by <var> in the FOR statement.

```
10 FOR <var> = <expr1> TO <expr2> [STEP <expr3>]
20 .
30 .
40 .
50 NEXT <var>
```

The **STEP** keyword and <expr3> are optional. If they are omitted, the value of <expr3> is assumed to be +1. If the STEP value is the wrong sign (for example, stepping from 1 to 10 in increments of −2), or if it is 0, then the loop is executed only once.

FOR...NEXT loops may be nested as deep as the stack will permit. Nesting means that a FOR...NEXT is completely contained within another FOR...NEXT loop. Each nested loop must have a unique variable. In the example below, note that the entire loop specified by <var2> is within the boundries of the loop specified by <var1>.

```
10 FOR <var1> = <expr1> to <expr2>
20 FOR <var2> = <expr3> to <expr4>
30 .
40 .
50 NEXT <var2>
60 NEXT <var1>
```

GOTO unconditionally transfers execution of the program to the line specified by <expr>.

```
GOTO <expr>
```

GOSUB is used to call a subroutine. Execution is transfered to the line number specified by <expr> and continues until the RETURN statement is encountered. The RETURN statement transfers execution to the statement immediatly following the GOSUB statement whether on the same line separated by a colon, or on the next line in the program. GOSUB must appear on a numbered line.

```
nnnn GOSUB <expr>
10 GOSUB 100
20
.
.
.
100 REMARK THIS IS A SUBROUTINE
110 PRINT "THIS IS A SUBROUTINE"
120 RETURN
```

Subroutines may call other subroutines and may be nested as deep as the stack will allow.

ON...GOTO and **ON...GOSUB** are used to transfer execution to one of up to nine specified line numbers, according to the value of an expression. This value should be between 1 and 9.

In the following example, <expr> is an expression and nnnn1 through nnnn9 are the line numbers in their assigned order. For the ON...GOSUB statement, the line numbers must be subroutines.

```
ON <expr> GOTO nnnn1, nnnn2,...,nnnn9
ON <expr> GOSUB nnnn1, nnnn2,...,nnnn9
```

INPUT assigns values to variables with data from the terminal or computer. The prompt string is printed on the screen along with a question mark that indicates that HERO-1 BASIC is ready for data. NOTE: Whenever HERO-1 BASIC prints letters to the screen, they will appear in uppercase.

```
INPUT ["PROMPT STRING"] <var>
```

You may specify more than one variable in an INPUT statement. In this case, the prompt string is printed once along with the question mark. Enter the value of each variable in the order specified, separated by commas. Values entered with an INPUT statement must be in decimal format. The first line of the example below shows the format of the INPUT statement. The second is a sample.

```
INPUT ["PROMPT STRING"] <var1>[, <var2>[, etc.]]
10 INPUT "ENTER THE VALUE OF X, Y, AND Z" X,Y,Z
```

If you place a comma between the prompt string and <var1>, the question mark will be suppressed.

```
INPUT ["PROMPT STRING"], <var1>[, <var2>[, etc.]]
```

DATA statements contain a list of values entered on the same line and separated by commas. These values may be expressions or constants and are assigned to variables in **READ** statements. Any expressions are evaluated when the READ is encountered. Whenever a DATA statement is encountered, it becomes the current DATA statement. This allows you to change the values to be assigned while the program is running.

NOTE: A DATA statement must be encountered before the first READ statement is encountered.

When the computer encounters line 10 of the example below, it becomes the current DATA statement. Then, on line 20, the variable W is assigned the value of 1. Similarly, the variables X, Y, and Z are assigned the values 2, 3, and 4 repectively. When line 30 is encountered, it becomes the current DATA statement. Line 40 assigns the value 5 to variable A. On line 50 the variables W, X, and Y are assigned the values 5, 6, and 7. Variable Z is assigned the value of variable A (5) + 4, or 9.

```
10 DATA 1, 2, 3, 4
20 READ W, X, Y, Z
30 DATA 5, 6, 7, A+4
40 LET A=5
50 READ W, X, Y, Z
```

The **RESTORE** statement causes the "data pointer", which points to the next value to be assigned, to point to the first value on the current DATA statement. On line 20 of the example below, variables X and Y are assigned the values 2 and 4. The RESTORE statement on line 30 resets the "data pointer" so that, on line 40, the variable Z is assigned the value 2.

```
10 DATA 2, 4, 6, 8
20 READ X, Y
30 RESTORE
40 READ Z
```

Use the **PRINT** statement to display a list of values and/or text on the screen. Strings of text must be enclosed in quotation marks. Punctuation used in the PRINT statement determines the actual position on the screen to which each item in the print list will be printed. The following functions are also available to format the output of PRINT statements: TAB, SPC, and CHR.

There are 10 print zones per printed line. These zones are eight columns wide and start in columns 1, 9, 17, 25, 33, 41, 49, 57, 65, and 73 of the printed line. In the example below, the items in the print list are separated by commas. <item1> is displayed in the first print zone (column 1 of the printed line). <item2> is printed in the next available print zone. If a string of text extends beyond its print zone, the next item in the list will be displayed in the next available print zone.

```
PRINT <item1>, <item2>
```

To skip print zones, insert one comma for each zone to be skipped. In the example below, <item1> is printed in the second print zone starting in column 9. <item2> is printed in the fifth print zone starting is column 33.

```
PRINT ,<item1>,,,<item2>
```

You can print items with no spacing between them by using semicolons (;) instead of commas. If a semicolon is placed at the end of the print list, the list will be printed, but no line feed will be issued at the end of the line. This allows items from PRINT statements on different lines of the program to be displayed on the same physical line on the screen. In the example below, the message "HERO-1 BASIC" is printed on one line of the screen.

```
10 PRINT "HERO-1";
20 PRINT "BASIC"
```

You can print an item starting in a specified column by using the **TAB** function. The TAB function cannot specify a column number lower than or equal to one that has already been printed. If this occurs, the TAB will be ignored.

In the example below, <expr2> must be greater than <expr1>.

```
PRINT TAB(<expr1>);<item1>;TAB(<expr2>); <item2> etc..
```

The **SPC** function prints a specified number of spaces between or in front of items in the print list.

```
PRINT SPC(<expr>);<item>
```

The **CHR** function prints a specified ASCII character. The character is determined by the absolute value modulo 256 of an expression. If the value is greater than or equal to 128, the character is printed but the column counter is not advanced. (Refer to the "ASCII to Decimal and Hexidecimal Conversion Chart" on Page 16.)

```
PRINT CHR(<expr>)
```

The **PEEK** function reads values from the robot's memory or I/O ports. The address is specified by an expression. Remember: PEEK only returns a value. If you want the value to be output to the screen, the PEEK function must appear in a PRINT statement: PRINT PEEK(<expr>).

```
PEEK (<expr>)
```

The **POKE** statement writes values to the robot's memory or I/O ports. The values may be within the range of 0 to 255 (0 to $FF hexadecimal). In the example below, <expr1> specifies the address and <expr2> specifies the value to be written.

```
POKE <expr1>, <expr2>
```

USER calls an assembly language subroutine already residing in the Robot's memory. When called, this subroutine will be executed in the Machine Mode. The Robot's operating mode may be switched between Machine and Robot Modes as desired to take full advantage of the Robot interpreter's special instructions. However, it must end with an RTS opcode and be running in Machine Mode when it is encountered.

You must place the address of the subroutine at the user vector with the high order byte of the address at $003F and the low order byte at $0040. The user subroutine must not reside between addresses $0000 to $01FF or $4000 to $DFFF. The following example shows how to set up this vector. In this example, the address of the subroutine is assumed to be $206C. This is the address of a subroutine in the ET-18-4 Utility ROM.

```
10 POKE $3F,$20 : REM LOAD VECTOR WITH HIGH ORDER ADDRESS BYTE
20 POKE $40,$6C : REM LOAD VECTOR WITH LOW ORDER ADDRESS BYTE
30 USER : REM CALL USER SUBROUTINE
```

The **SGN** function returns the sign of the value of an expression. For expressions that have a value greater than zero, the value +1 is returned. If the expression is less than zero, the value −1 is returned. If the value of the expression equals zero, a 0 is returned.

```
SGN(<expr>)
```

The **ABS** function returns the absolute value of an expression.

```
ABS(<expr>)
```

RND generates a random number in the range of 0 to 99.

ROBOT STATEMENTS AND FUNCTIONS

Speech Synthesis

With the **SPEAK** statement, the Robot can use its Voice Synthesizer to simulate human speech. This statement has two forms. With the first form, the Robot will speak a preprogrammed phrase at the address specified by the value of the expression. The first line of the example below shows the format. The second and third lines are samples.

```
SPEAK <expr>
SPEAK $FA4B
SPEAK 64709
```

With the second form, the Robot translates the phoneme mnemonics contained in <phoneme string>. These mnemonics are illustrated in depth in the "Robot Voice Dictionary" that comes with the Voice Synthesizer. The "Alphanumeric Listing of Phoneme Mnemonics" reference table on Page 15 lists the mnemonics in alphanumerical order.

```
SPEAK <phoneme string>
SPEAK "THV I S PA0 I1 Z UH1 F O W N EH1 M PA0 S T R I1 NG PA1"
```

Remember: For proper operation, always put a PA0, PA1, or STOP phoneme at the end of a phoneme string.

You can select four levels of inflection by placing 1, 2, 3, or 4 in front of a phoneme in the string, with 1 being the lowest level and 4 the highest. These may be placed anywhere within the string and will remain in effect until the next level is encountered, or the end of the string. If no level is specified, the default value is level 1.

```
10 SPEAK "1W UH N PA1"
20 SPEAK "2T IU U W PA1"
30 SPEAK "3TH R E1 Y PA1"
40 SPEAK "4F O1 ER PA1"
50 SPEAK "4F O1 ER PA1 3TH R E1 Y PA1 2T IU U W PA1 1W UH N PA1"
```

Motor Movement

You can control the positions of each of the Robot's axes of motion with an assignment command similar to a LET statement. As in the example below, the axis to be moved is specified by name and set equal to the value of an expression. The names of the axis variables appear in the table below. These variables cannot be set with LET, READ, INPUT or FOR...NEXT statements.

```
<axis> = <expr>
```

You can read the position of any of the robot's axes as a variable. When encountered, the current position of the specified axis is read and converted to a standard unit of measure.

VARIABLE	RANGE	UNITS	COMMENTS
EXTEND	0 to 51	.1 Inches	0 = Retracted
SHOULDER	0 to 159	Degrees	0 = Down
ROTATE	−183 to 166	Degrees	0 = Centered
PIVOT	0 to 179	Degrees	
GRIPPER	0 to 100	Percent	0 = Closed
HEAD	−165 to 162	Degrees	0 = Centered

Four other commands govern the movement of the Robot's base. Movement is specified in inches **FWD** (forward) or **BWD** (backward) or in degrees **LEFT** or **RIGHT**. FWD and BWD move the robot along a straight line. LEFT and RIGHT spin the robot on its axis.

```
<direction> <expr>
   LEFT      90
```

NOTE: Due to the nature of integer BASIC, the position to which the Robot moves may differ slightly from the actual value of the expression. In the example below, line 10 directs the Robot to turn its HEAD 10 degrees clockwise. The actual position is printed on the screen by line 20. Note that it is only 9 degrees clockwise.

```
10 HEAD = 10
20 PRINT HEAD
```

To compensate for variations in the Robot's environment, two special variables have been provided to calibrate the Robot's base movements. **LCF** (Linear Calibration Factor) is used to adjust the length the Robot moves during FWD and BWD commands. **TCF** (Turn Calibration Factor) adjusts how far the Robot spins during LEFT and RIGHT commands.

```
10 TCF = 96
20 LCF = 120
```

Both of these variables default to a value of 100%, but may be set to values from 0 to 233%. For consistency of movement, they should appear before any base movement command is encountered.

Sensors

Data from each of the Robot's sensors may also be addressed as a variable.

VARIABLE	RANGE	UNITS	COMMENTS
EYE	0 to 255		
EAR	0 to 255		
SONAR	0 to 99	Inches	
MOTION	0 or 1		1 = Motion

Other Robot Statements

The **KEYIN** function returns the value of a key on the Robot's keyboard. An underscore prompt (_) is printed on the Robot's display to indicate that HERO-1 BASIC is waiting for a keystroke.

```
<var> = KEYIN
```

NOTE: The "BREAK" key will not respond during a KEYIN function.

The **DPRINT** statement prints expressions or strings of text on the Robot's displays. Values of expressions are printed in decimal format. Strings of text must be enclosed in quotation marks.

Printing begins with the left-most display and continues to the right. If the number of characters to be printed exceeds the number of available digits, the display will scroll left as additional characters are printed. If the special character $ is imbedded in a DPRINT text string, it will cause the displays to be cleared. Spaces will appear as one blank display.

```
10 A = -12345
20 DPRINT "$THIS$IS A$TEST!"
30 DPRINT "$A= "
40 DPRINT A
```

The following characters may be output to the Robot's displays:

```
!'()-.0123456789=ABCDEFGHIJKLMNOPQRSTUVWXYZ[]_
```

NOTE: Due to the limitations of seven-segment displays, the characters K, M, V, W, X, and Z may take a little getting used to. These characters are shown below.

CHARACTER	DISPLAY
K	
M	
V	
W	
X	
Z	

ERROR CODES

NUMBER	MEANING
10	Unrecognized keyword
12	Error while loading program
14	Illegal variable
16	No line number referenced by GOTO or GOSUB
18	Equal sign missing in assignment statement
20	Expression syntax error or unbalanced parenthesis
21	Expression expected but not found
22	Division by zero
23	Arithmetic overflow
24	Expression too complex
31	Syntax error in PRINT statement
32	Missing closing quote in printed string
40	Error in DIM statement
45	Syntax error in INPUT statement
51	Syntax error in READ statement
62	Syntax error in IF statement
73	RETURN without GOSUB
81	FOR/NEXT error
90	Memory overflow
99	"BREAK" (attention interrupt)

ALPHANUMERIC LISTING OF PHONEME MNEMONICS

MNEMONIC	DURATION	EXAMPLE	MNEMONIC	DURATION	EXAMPLE
A	185 ms	dAY	K	80 ms	triCK
A1	103 ms	mAde	L	103 ms	Land
A2	71 ms	mAde	M	103 ms	Mat
AE	185 ms	dAd	N	80 ms	suN
AE1	103 ms	After	NG	121 ms	riNG
AH	250 ms	mOp	O	185 ms	cOld
AH1	146 ms	fAther	O1	121 ms	abOArd
AH2	71 ms	hOnest	O2	80 ms	fOr
AW	250 ms	cAll	OO	185 ms	bOOk
AW1	146 ms	lAWful	OO1	103 ms	lOOking
AW2	30 ms	sAlty	P	103 ms	Past
AY	21 ms	dAY	R	90 ms	Red
B	71 ms	Bag	S	90 ms	paSS
CH	71 ms	CHop	SH	121 ms	SHop
D	55 ms	faDe	T	71 ms	Tap
DT	47 ms	buTTer	TH	71 ms	THin
E	185 ms	mEEt	THV	80 ms	THe
E1	121 ms	bE	U	185 ms	mOve
EH	185 ms	gEt	U1	90 ms	yOU
EH1	121 ms	hEAvy	UH	185 ms	cUp
EH2	71 ms	Enlist	UH1	103 ms	Uncle
EH3	59 ms	jackEt	UH2	71 ms	About
ER	146 ms	bIRd	UH3	47 ms	missIOn
F	103 ms	Fast	V	71 ms	Van
G	71 ms	Get	W	80 ms	Win
H	71 ms	Hello	Y	103 ms	anY
I	185 ms	pIn	Y1	80 ms	Yard
I1	121 ms	inhIbit	Z	71 ms	Zoo
I2	80 ms	Inhibit	ZH	90 ms	aZure
I3	55 ms	inhibIt	PA0	47 ms	silent
IU	59 ms	yOU	PA1	185 ms	silent
J	47 ms	JuDGe	STOP	47 ms	silent

ASCII TO DECIMAL AND HEXADECIMAL CONVERSION CHART

ASCII	DECIMAL	HEX	ASCII	DECIMAL	HEX	ASCII	DECIMAL	HEX
NUL	0	$00	+	43	$2B	V	86	$56
SOH	1	$01	,	44	$2C	W	87	$57
STX	2	$02	–	45	$2D	X	88	$58
ETX	3	$03	.	46	$2E	Y	89	$59
EOT	4	$04	/	47	$2F	Z	90	$5A
ENQ	5	$05	0	48	$30	[	91	$5B
ACK	6	$06	1	49	$31	/	92	$5C
BEL	7	$07	2	50	$32	]	93	$5D
BS	8	$08	3	51	$33	^	94	$5E
HT	9	$09	4	52	$34	←	95	$5F
LF	10	$0A	5	53	$35		96	$60
VT	11	$0B	6	54	$36	a	97	$61
FF	12	$0C	7	55	$37	b	98	$62
CR	13	$0D	8	56	$38	c	99	$63
SO	14	$0E	9	57	$39	d	100	$64
SI	15	$0F	:	58	$3A	e	101	$65
DLE	16	$10	;	59	$3B	f	102	$66
DC1	17	$11	<	60	$3C	g	103	$67
DC2	18	$12	=	61	$3D	h	104	$68
DC3	19	$13	–	62	$3E	i	105	$69
DC4	20	$14	?	63	$3F	j	106	$6A
NAK	21	$15	@	64	$40	k	107	$6B
SYN	22	$16	A	65	$41	l	108	$6C
ETB	23	$17	B	66	$42	m	109	$6D
CAN	24	$18	C	67	$43	n	110	$6E
EM	25	$19	D	68	$44	o	111	$6F
SUB	26	$1A	E	69	$45	p	112	$70
ESC	27	$1B	F	70	$46	q	113	$71
FS	28	$1C	G	71	$47	r	114	$72
GS	29	$1D	H	72	$48	s	115	$73
RS	30	$1E	I	73	$49	t	116	$74
US	31	$1F	J	74	$4A	u	117	$75
SP	32	$20	K	75	$4B	v	118	$76
!	33	$21	L	76	$4C	w	119	$77
"	34	$22	M	77	$4D	x	120	$78
#	35	$23	N	78	$4E	y	121	$79
$	36	$24	O	79	$4F	z	122	$7A
%	37	$25	P	80	$50	{	123	$7B
&	38	$26	Q	81	$51	|	124	$7C
'	39	$27	R	82	$52	}	125	$7D
(	40	$28	S	83	$53	~	126	$7E
)	41	$29	T	84	$54	DEL	127	$7F
*	42	$2A	U	85	$55			

CUSTOMER SERVICE

REPLACEMENT PARTS

Please provide complete information when you request replacements from either the factory or Heath Electronic 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 ELECTRONIC CENTERS

For your convenience, "over the counter" replacement parts are available from the Heath Electronic 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 Electronic 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.

Heathkit Electronic 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 Heathkit Electronic Center. For warranty parts replacement, supply a copy of the invoice or sales slip.

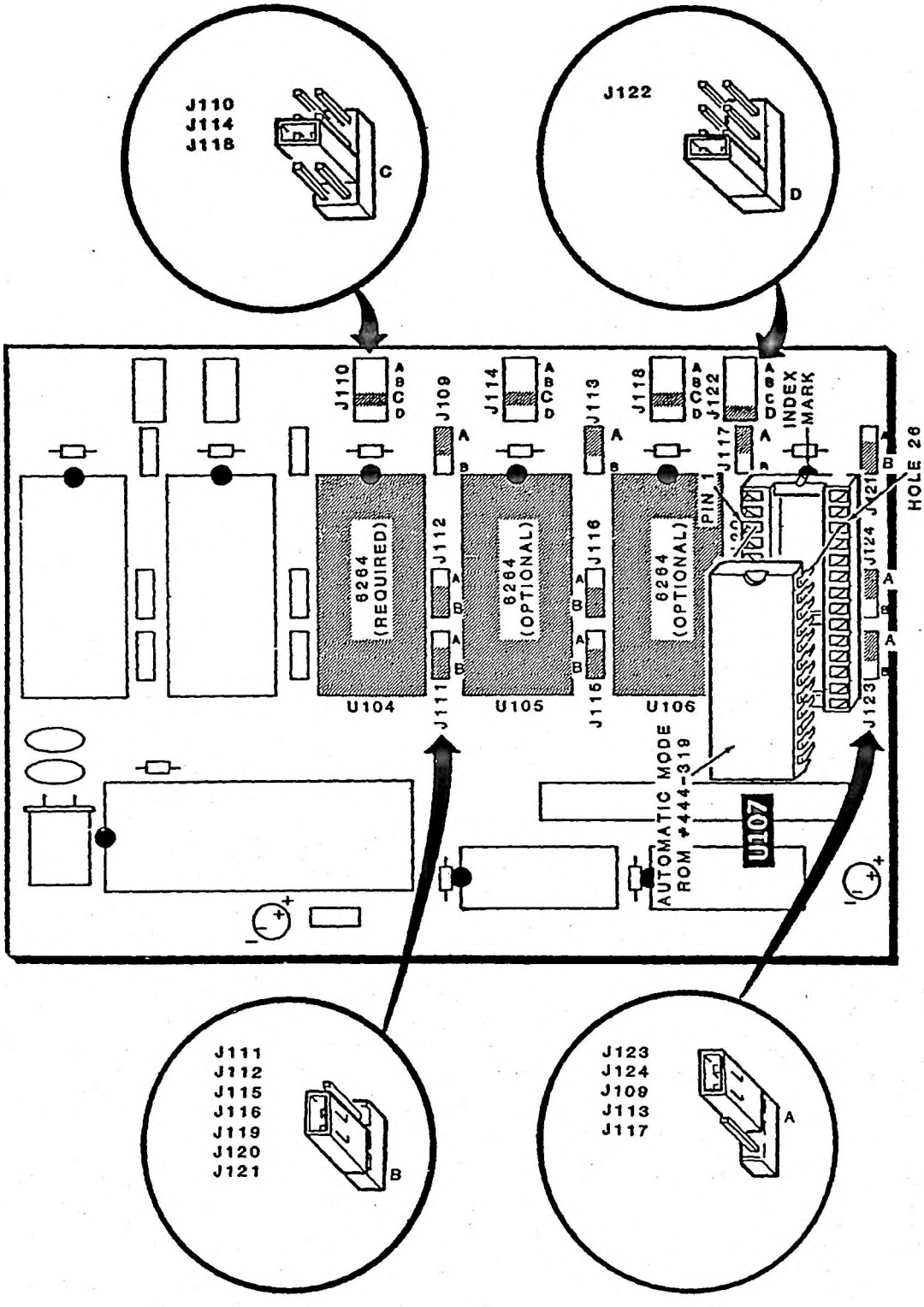
If you prefer to ship your kit to the factory, attach a letter containing the following information directly to the unit:

- 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


J110
J114
J118
C
J122
D
J111
J112
J115
J116
J119
J120
J121
B
J123
J124
J109
J113
J117
A
8264
(REQUIRED)
8264
(OPTIONAL)
8264
(OPTIONAL)
U104
U105
U106
U107
INDEX
MARK
PIN 1
HOLE 26
AUTOMATIC MODE
ROM #444-319

PICTORIAL 2