

## FOREWORD

The EPSON HX-20 Portable Computer is the ultimate in personal computers – a complete desktop personal computer system miniaturized to fit in a briefcase. To take full advantage of this portability, the hardware and software features listed below have been incorporated in its design.

This manual provides a detailed explanation of one of those features, the unique EPSON developed BASIC, and it is hoped that the user will profitably use this volume as a reference for programming as well as for further application software development. To those ends, we have edited and prepared this manual with ease of reference and use in mind.

### Features

1. The RAMs are backed up with batteries so that programmes and data stored in the RAM are protected against loss even when the power switch is turned OFF. Programme execution is possible upon power application.
2. The memory space is divided into five programme areas, each capable of storing a separate BASIC programme. Each programme can be selected from the menu for immediate execution.
3. The 5 programmes in the menu are managed independently, so that the creation of a new programme or the editing of the existing programme does not affect other programmes in the memory.
4. The HX-20 features an 80-character (20 char. by 4 lines) liquid crystal display (LCD). A virtual screen for larger than the LCD screen can be specified by a WIDTH command and the capacity to scroll freely in any of the four directions enables the user to perform screen editing of the large internal screen beyond the physical limitations of the LCD.
5. All operations of the optional microcassette drive can be controlled under BASIC.
6. In addition to the programme areas, the HX-20 is provided with a RAM file area to facilitate data storage, as well as data transfer between programmes.
7. By connecting an optional TF-20X terminal floppy unit to the HX-20, DISK BASIC can be run.

### NOTICE:

- \* All rights reserved. Reproduction of any part of this manual in any form whatsoever without EPSON's express written permission is forbidden.
- \* The contents of this manual are subject to change without notice.
- \* All efforts have been made to ensure the accuracy of the contents of this manual. However, should any errors be detected, EPSON would greatly appreciate being informed of them.
- \* The above notwithstanding, EPSON can assume no responsibility for any errors in this manual or their consequences.

# TABLE OF CONTENTS

## Chapter 1 Introduction to HX-20

1.1	Initialisation and BASIC .....	1-1
1.2	Operation of keyboard .....	1-6
1.3	Text and graphic screens .....	1-10
1.4	Text screen (virtual screen) .....	1-12
1.5	Screen edit .....	1-14
1.6	File operations .....	1-16
1.7	Peripheral equipment .....	1-18

## Chapter 2 Outline of EPSON BASIC

2.1	Operation modes .....	2-1
2.2	Statements .....	2-2
2.3	Lines .....	2-2
2.4	Character set .....	2-2
2.5	Constants .....	2-3
2.6	Variables .....	2-4
2.7	Type conversion .....	2-5
2.8	Expressions and operations .....	2-6
2.9	Error messages .....	2-11
2.10	DIP switch setting .....	2-12

	How to use Chapters 3 and 4 .....	2-13
--	-----------------------------------	------

## Chapter 3 Commands and Statements

## Chapter 4 Functions

## Chapter 5 Additional Information

5.1	RAM files .....	5-1
5.2	Sequential files .....	5-5
5.3	Machine language programmes .....	5-9
5.4	How to use the RS-232C port .....	5-14

## APPENDIXES

A.	Error Messages
B.	Device Names
C.	Correspondence Table between Device Names and EPSON BASIC Commands
D.	Formatting Characters
E.	Keyboard Layouts and Key assignments
F.	Character Code Tables
G.	Control Codes
H.	Memory Map
I.	Table of Reserved Words
J.	List of Commands and Statements

## Commands and Statements

AUTO .....	3-1	MEMSET .....	3-37
CLEAR .....	3-2	MERGE .....	3-38
CLOSE .....	3-3	MERGE"COM0:" .....	3-38
CLS .....	3-3	MID\$ .....	3-39
COLOR .....	3-4	MON .....	3-40
CONT .....	3-5	MOTOR .....	3-41
COPY .....	3-5	NEW .....	3-42
DATA .....	3-7	ON ERROR GOTO .....	3-42
DEFFIL .....	3-8	ON...GOSUB/ON...GOTO .....	3-44
DEF FN .....	3-9	OPEN .....	3-45
DEFINT/SNG/DBL/STR .....	3-10	OPEN"COM0:" .....	3-46
DEF USR .....	3-12	OPTION BASE .....	3-48
DELETE .....	3-13	PCOPY .....	3-49
DIM .....	3-14	POKE .....	3-50
END .....	3-14	PRESET .....	3-52
ERASE .....	3-15	PRINT/LPRINT .....	3-53
ERROR .....	3-16	PRINT USING/LPRINT USING .....	3-54
EXEC .....	3-17	PRINT# .....	3-58
FILES .....	3-18	PRINT# USING .....	3-58
FOR...TO...STEP - NEXT .....	3-19	PSET .....	3-59
GCLS .....	3-21	PUT% .....	3-60
GET% .....	3-21	RANDOMIZE .....	3-61
GOSUB - RETURN .....	3-22	READ .....	3-61
GO TO/GOTO .....	3-23	REM .....	3-62
IF...THEN...ELSE/ IF...GOTO...ELSE .....	3-24	RENUM .....	3-62
INPUT .....	3-25	RESTORE .....	3-63
INPUT# .....	3-26	RESUME .....	3-64
KEY .....	3-26	RUN .....	3-65
KEY LIST/KEY LLIST .....	3-27	RUN"COM0:" .....	3-65
LET .....	3-27	SAVE .....	3-66
LINE .....	3-28	SAVE"COM0:" .....	3-67
LINE INPUT .....	3-29	SAVEM .....	3-67
LINE INPUT# .....	3-30	SCREEN .....	3-68
LIST/LLIST .....	3-30	SCROLL .....	3-69
LIST<file descriptor> .....	3-31	SOUND .....	3-70
LIST"COM0:" .....	3-31	STAT .....	3-71
LOAD .....	3-32	STOP .....	3-72
LOAD"COM0:" .....	3-32	SWAP .....	3-72
LOADM .....	3-33	TITLE .....	3-73
LOAD? .....	3-34	TRON/TROFF .....	3-74
LOCATE .....	3-35	WIDTH .....	3-75
LOCATES .....	3-36	WIDTH<device name> .....	3-76
LOGIN .....	3-36	WIND .....	3-77

## Functions

ABS .....	4-1	LOF .....	4-15
ASC .....	4-1	LOG .....	4-15
ATN .....	4-2	MID\$ .....	4-16
CDBL .....	4-2	OCT\$ .....	4-17
CHR\$ .....	4-3	PEEK .....	4-17
CINT .....	4-3	POINT .....	4-18
COS .....	4-4	POS .....	4-19
CSNG .....	4-4	RIGHT\$ .....	4-19
CSRLIN .....	4-5	RND .....	4-20
DATE\$ .....	4-5	SGN .....	4-21
DAY .....	4-6	SIN .....	4-21
EOF .....	4-6	SPACE\$ .....	4-22
ERL/ERR .....	4-7	SPC .....	4-22
EXP .....	4-7	SQR .....	4-23
FIX .....	4-8	STR\$ .....	4-23
FRE .....	4-8	STRING\$ .....	4-24
HEX\$ .....	4-9	TAB .....	4-25
INKEY\$ .....	4-10	TAN .....	4-25
INPUT\$ .....	4-11	TAPCNT .....	4-26
INSTR .....	4-12	TIME\$ .....	4-27
INT .....	4-13	USR .....	4-27
LEFT\$ .....	4-14	VAL .....	4-28
LEN .....	4-14	VARPTR .....	4-28

## **NOTICE**

In those countries where no display controller is available, this manual should be used making the following changes.

### **CHAPTER 1 Introduction to HX-20**

The following sections should be read disregarding any reference to the external display or display controller.

- 1.3 Text and graphic screen
- 1.4 Text screen
- 1.5.2 Moving the physical screen
- 1.7.7 Screen

### **CHAPTER 3 Commands and Statements**

#### **COLOR**

This command cannot be used without the display controller being connected to the HX-20.

Any reference to the external display or display controller in the following commands should be disregarded.

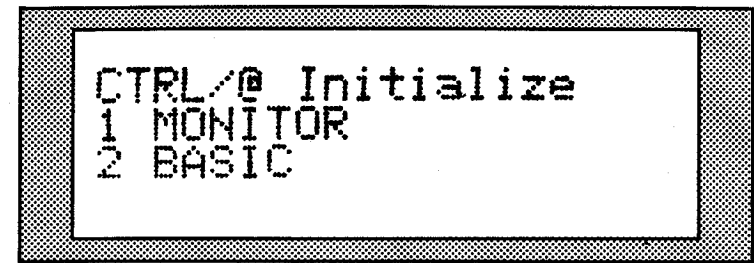
**CLS**  
**GCLS**  
**LINE**  
**PRESET**  
**PSET**  
**SCREEN**  
**SCROLL**  
**WIDTH**



**CHAPTER 1**  
**Introduction to HX-20**

## 1.1 Initialisation and BASIC

Unlike other personal computers, the memory space of your HX-20 Portable Computer is divided into 5 programme areas where 5 BASIC programmes can be stored independently. These programmes will be retained in the memory without any change even if you turn OFF the power switch of the HX-20. The HX-20 incorporates a menu function to enable any of the stored programmes to be executed with a single-touch key operation. Turn ON the power switch, and the menu will appear on the LCD screen as follows.



The menu displays numbers and the functions (i.e., programme names) which can be executed when you press the corresponding numeric keys. When power is applied to the HX-20 for the first time, only two functions will appear on the LCD screen as shown above. "CTRL/@" at the uppermost line of the display indicates that you must press the **@** key while holding down the **CTRL** key.

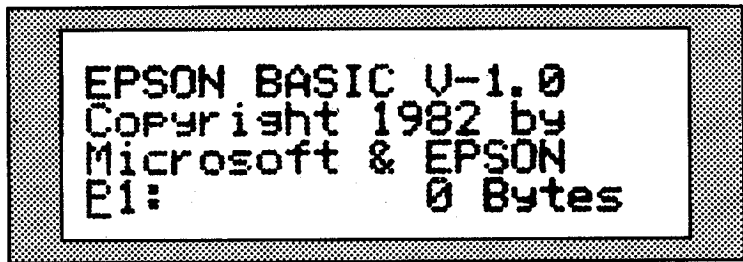
**NOTE:**

*If you select a character set other than that for USA, England, Italy or Spain, the following symbols will appear in lieu of "@".*

USA	France	Germany	Sweden
@	à	§	É

*The character code and function of each of these keys is the same as "@" and the corresponding character for each character set should be pressed while holding down **CTRL** key. (See Section 2.4, Character Sets, for details.)*

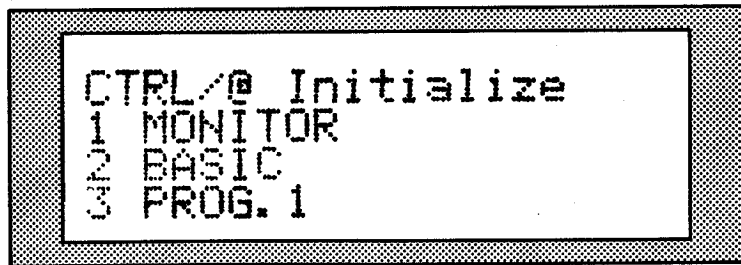
If you do this followed by the date and time setting described later, your HX-20 will be initialised. This state is called "cold start". (You must initialise your HX-20 by cold start when you turn ON the power switch for the first time after purchase.) To execute EPSON BASIC, press numeric key "2".



When BASIC is executed, the programme area No. 1 is always selected. The message "0 Bytes" following "P1:" indicates the length of the programme stored in the current programme area. In this case, as no programme has been stored in program area No. 1, the value displayed is "0" bytes.

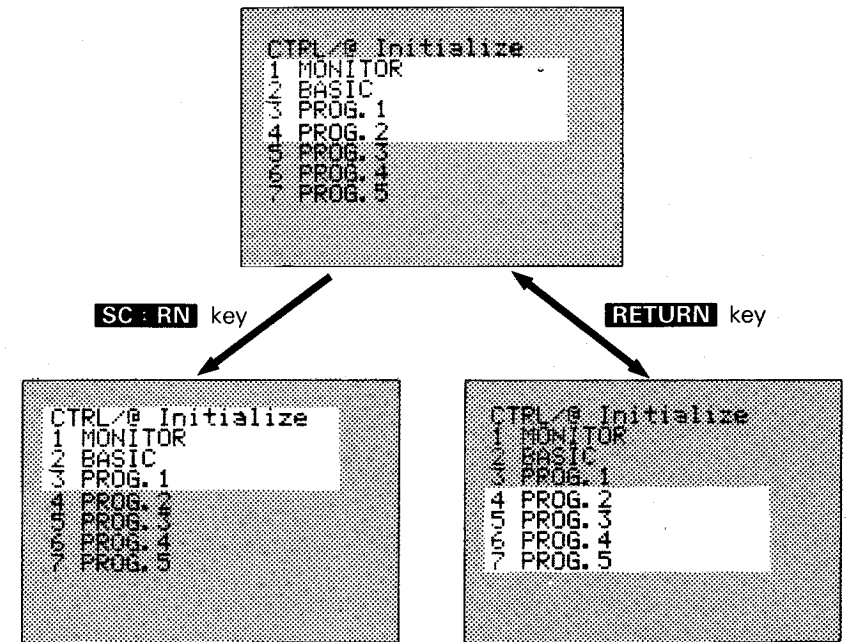
### 1.1.1 Menu

If you name a programme that you have written in BASIC with a TITLE command, the programme name will appear in the menu on the display. This means that the programme name has been registered in the menu.



5 programmes registered in the menu are numbered 3 through 7, respectively. Any of these programmes will be executed directly (without using LOGIN and RUN commands) whenever you press the numeric key corresponding to the programme number.

When you call the menu after all the programme names have been registered, the screen will scroll up to show you the programme names gradually as all the programme names cannot be displayed at one time in the LCD display window. After the display of all the programme names, it will then return to the first name "1 MONITOR", indicating that the HX-20 system has returned to command level. If you press **RETURN** key at this moment, the current display is switched with the display outside the physical screen to show you the remaining programme names. Pressing **SC:RN** key will return the display to the initial screen. If you press **SHIFT** and **SC:RN** keys, the screen will move backwards.

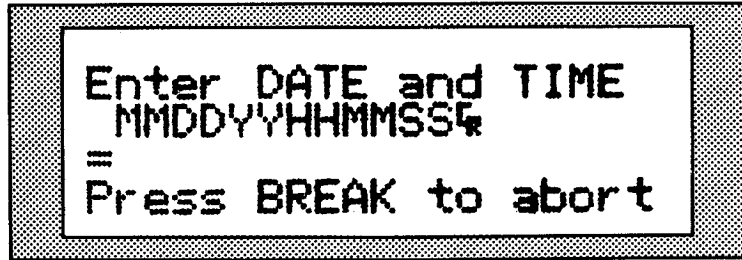


**NOTE:**  
 The numbers assigned to the programmes are used only for menu display and the K command of the MONITOR, and have no direct connection with the programme area numbers used by a LOGIN command. Also, these numbers may change upon registration of other programmes in the menu.

The K command of the MONITOR makes the HX-20 ready for direct programme execution without requiring any other operation immediately after the power switch is turned ON.

### 1.1.2 Cold start

When **CTRL** key and "@" key are pressed while the menu is being displayed, the HX-20 will ask you to input the date and time.



At this point, if you type the current date in the format "MMDDYY" (month, day and year) and the current time in the format "HHMMSS" (hours, minutes and seconds) in 12 digits (24-hour system), and then press **RETURN** key, the HX-20 will be initialised. This state is called "cold start". You can correct the time setting by pressing **INS DEL** key. When the HX-20 is initialised by cold start, all the memory contents of the HX-20 are cleared and the respective constants in BASIC are reset as follows. (For details, refer to the applicable chapters.)

```
CLEAR 200,256
MEMSET &H0A3F
KEY 1, "AUTO"
KEY 2, "LIST" + CHR$(13)
KEY 3, "LLIST" + CHR$(13)
KEY 4, "STAT"
KEY 5, "RUN" + CHR$(13)
KEY 6, "?DATES?:?TIMES$" + CHR$(13)
KEY 7, "LOAD"
KEY 8, "SAVE"
KEY 9, "TITLE"
KEY 10, "LOGIN"
```

As previously described, when the HX-20 is cold started, all the data currently stored in the HX-20 memory are cleared. So please be careful not to cold start your HX-20 unnecessarily. If you accidentally press **CTRL** and "@" keys, do not set the date and time but press **BREAK** or **MENU** key. This will cause the HX-20 to return to the menu display again.

### 1.1.3 Warm start

All the execution of BASIC by other than the cold start is called "warm start". When the HX-20 is initialised by warm start, the respective constants in BASIC are reset as follows. (These are called "default values after warm start".)

```
CLEAR 200
SCREEN 0,0
SCROLL 9,0,10,4 } LCD
WIDTH 40,8,3
SCROLL 9,0,16,16 } External display
WIDTH 40,37,5
```

If you press **MENU** key even during the execution of a BASIC programme, BASIC will enter the warm start state. Therefore, please note that while you are operating the HX-20, for example, to change the screen size, the default values will be assumed after the menu has been displayed by pressing **MENU** key. (The size of the external display, however, will not be affected by warm start.)

### 1.1.4 Reset

There is a RESET switch, recessed at the rear of the right-hand side of your HX-20. This switch need not be operated during normal BASIC operation. Never use the RESET switch except in the circumstances as described below. The only times when you must operate the RESET switch are those cases where the HX-20 does not respond to **BREAK** key and the POWER switch (as a result of a programme overrun). In the HX-20, a programme overrun will never occur unless the HX-20 is operated incorrectly as follows.

- Data rewrite in the system area (memory addresses &H004D to &H0A3F)
- Attempt to read the data in the I/O area (memory addresses &H0000 to &H004D).
- Execution of an incompletely written machine language programme.

If a programme overrun should occur as a result of one of these operations, turn the POWER switch OFF immediately and then press the RESET switch.

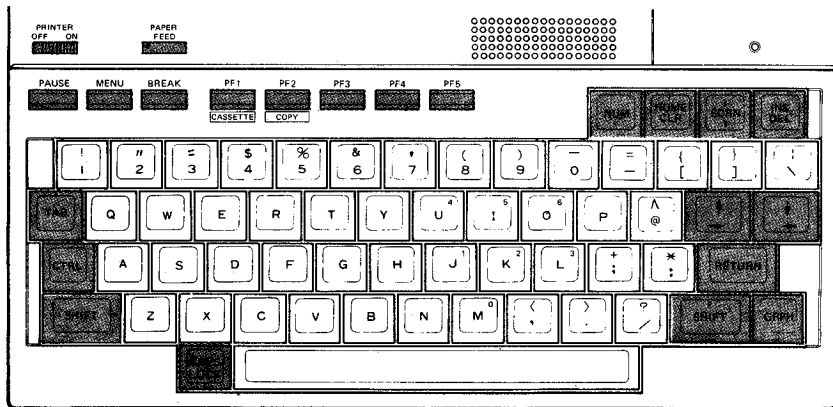
When you attempt to execute BASIC after depressing the RESET switch, there may be a case where part of the memory has been rewritten due to overrun. In such a case, the menu display is disrupted, and your attempts to execute BASIC are not successful. Should this happen, you must initialise the HX-20 by cold start to clear all the memory contents of the HX-20. Before executing an incompletely written machine language programme, always save the data and programmes in the memory to a cassette tape or other external memory so that even if the memory should be erased, no problem will occur.



**NOTE:**  
 While you are operating the HX-20, there may be a case where the message "CHARGE BATTERY!" will suddenly appear on the LCD screen. If this happens, all the HX-20 operations will be suspended and the HX-20 will not accept any inputs. This is a warning message to tell you that the battery voltage is low. If the battery voltage of the HX-20 falls below a certain level, the HX-20 will stop its operation to protect the programmes and data in the memory. If this message appears, turn the power switch OFF and recharge the batteries as soon as possible.

## 1.2 Operation of keyboard

The keyboard arrangement of the HX-20 is as shown below.



**NOTE:**  
 The keyboard arrangement shown above applies to that for U.S.A. Refer to APPENDIX E for England Keyboard.

### 1.2.1 Keyboard modes

The HX-20 has three modes for the input of characters. The characters that can be input in each of these modes is different.

#### (1) Uppercase Mode

In this mode, the normal keyboard input is uppercase characters. For example, by pressing the key marked "A" on the keyboard, the character "A" will be input. If this key is pressed while holding down **SHIFT** key, lowercase "a" will be input. For the numeric and symbolic keys, the number or symbol appearing on the lower half of the key will normally be input and the number or symbol on the upper half of the key will be input when the key is pressed while holding down **SHIFT** key.

For example, when **:** key is pressed in the uppercase mode, ":" is normally input and "\*" is input if this key is pressed while holding down **SHIFT** key. The uppercase mode is the default mode and the HX-20 is set in this mode every time BASIC is executed.

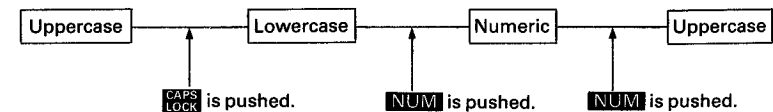
#### (2) Lowercase Mode


The HX-20 enters this mode when the key marked **CAPS LOCK** is pressed. In this mode, lowercase letters are normally input and uppercase letters are input by pressing a letter key while holding down **SHIFT** key. The function of the numeric and the symbolic keys is the same as in the uppercase mode.

#### (3) Numeric Mode

The HX-20 enters the numeric mode when the **NUM** key is pressed. The keys which can be operated in this mode are the numeric keys at the top of the keyboard and the letter keys which have the numbers 0 to 6 written in their upper right-hand corner as well as the symbolic keys (+), (-), (\*), (/), (.), (,) and (?). All other keys will be ineffective even if you press them. For the characters assigned for each mode, refer to APPENDIX E, "Key Assignments for Each Keyboard Mode".

You can change the mode by pressing one of the keyboard mode keys. However, if you press the key for the mode which the HX-20 is currently in, the keyboard mode will return to the uppercase mode. An example of changing key modes is shown below.



Keys marked  in the Figure on page 1-6 are special function keys and can be input in any mode.

### 1.2.2 Input of graphic characters

Graphic characters can be input in the uppercase mode by pressing the respective keys while holding down the **GRPH** key. For details, refer to APPENDIX E, "Key Assignment for Each Keyboard Mode".

### 1.2.3 Special functions of CONTROL key

**CTRL** key, when used in conjunction with other keys, inputs control codes that perform the special functions such as the movement of the cursor on the screen, etc. (For details of these control codes, see APPENDIX G.)

In addition, **CTRL** key has two special functions.

**CTRL** + **PF2** Copies the data displayed on the LCD on the built-in microprinter. This function is the same as a COPY command execution.

**CTRL** + **PF1** Sets the optional microcassette drive in manual operation mode. This mode will not be entered if the microcassette is not connected. When the manual operation mode is entered, the display is extinguished and the tape counter value is displayed in the upper right-hand corner of the physical screen. In the manual operation mode, the microcassette operations can be controlled by the programmable function keys **PF1** to **PF6** as follows.

- PF1** Fast forward.
- PF2** Slow forward.
- PF3** Stops the tape rewind, fast forward or slow forward.
- PF4** Rewinds the tape.
- PF5** Causes exit from manual operation mode.
- PF6** Resets the tape counter value.  
(PF6 is input by pressing **PF1** while holding down **SHIFT** key.)

### 1.2.4 Functions of special keys



- PAPER FEED** Used to feed paper into the the built-in microprinter.
- PAUSE** Used to temporarily stop the programme execution and listing. The interrupted operation is resumed upon pressing any other key on the keyboard. In this case, if one of the numeric keys 0 through 9 is pressed to resume execution, the scrolling speed can be specified.
- MENU** Used to return the HX-20 to the state prior to BASIC execution. The menu is displayed and the HX-20 waits for input of your selected function.

**BREAK** Used to stop programme execution or listing. Programme execution can be resumed by input of a CONT command.

**PF1 - PF5**  
**PF6 - PF10** You can define the special functions by software in these keys. (For details, see KEY command in Chapter 3.)

**RETURN** Used to signal BASIC that input of data in the required units has been completed. When this key is pressed, the cursor moves to the beginning of the next line.

**NUM GRPH** These keys are used to select the keyboard mode.

**TAB**    
**SC : RN** These keys are used for screen editing. For details of each key, refer to Section 1.5.

For the built-in microprinter, there is a **PRINTER ON/OFF** switch and a **PAPER FEED** key. The microprinter can only be operated when the **PRINTER ON/OFF** switch is in the ON position. For example, even if you execute an LLIST statement or press the **PAPER FEED** key, if the **PRINTER ON/OFF** switch is OFF, the microprinter will not function. The **PAPER FEED** key will feed the roll paper into the feed only while it is being pressed.

### 1.2.5 Auto-repeat function

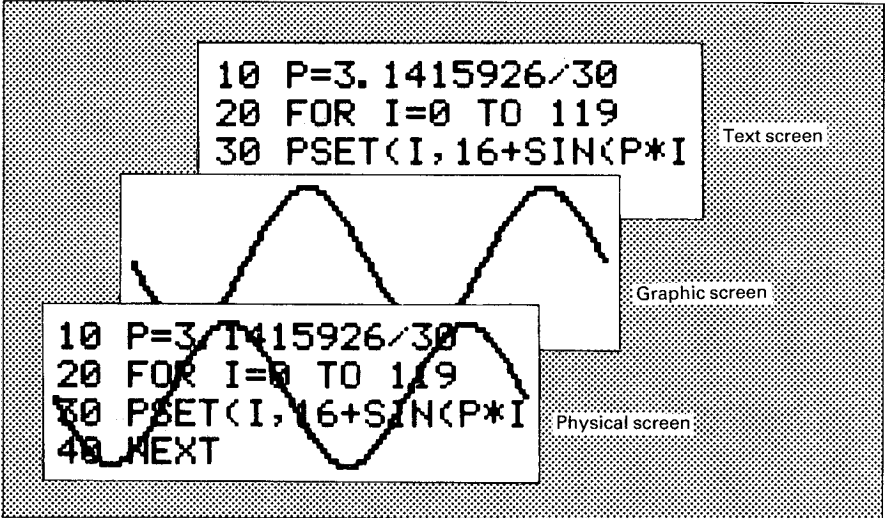
If the alphanumeric keys are pressed continuously, they have a function to automatically input continuously. In addition, the following special keys have the same function.

**HOME CLR** **SC : RN** **INS DEL**   **TAB** **RETURN**

### 1.3 Text and graphic screens

The HX-20 has two distinct screens that you may use for the entirely different purposes. The first screen called a "text screen" is to display characters and the second called a "graphic screen" is used to draw graphics using such statements as LINE and PSET. These two screens can be output through two different output devices; one is the built-in LCD and the other is an optional external display.

In EPSON BASIC, with a SCREEN statement, you can specify three different methods of display by combining the two screen modes and the two output devices. When BASIC is executed, both the text screen and graphic screen are specified for display on the LCD.

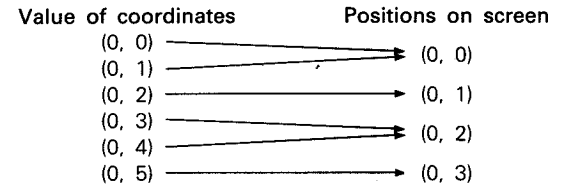


When you want to draw lines and dots on the graphic screen, you must use rectangular coordinates with the upper left-hand corner of the display as the origin. The value of a horizontal coordinate increases as a point on the X-axis moves towards the right, while the value of a vertical coordinate increases as a point on the Y-axis moves downwards. The range of coordinates for the LCD is different from that for the external display. Also, for the external display, the range of coordinates varies depending on the selected display mode. In a LINE or PSET statement, the HX-20 does not check to see whether or not the specified coordinates are actually within the screen. Please pay special attention to the range of coordinates.

	Mode	Horizontal (dots)	Vertical (dots)
LCD		120	32
External display	4-colour	128	64
	High resolution	128	96

**NOTE:**  
The above table indicates the resolution of each display and does not indicate the range of coordinates. For example, the LCD has a resolution of 120x32 dots. But the specified values of coordinates must be in the range of 0 to 119 horizontally and 0 to 31 vertically.

When the external display is set in 4-colour graphic mode, the resolution of the display is 64 dots vertically. But the range of coordinates that you can specify is between 0 and 96, which is the same as that in high resolution mode. In other words, the values of your specified coordinates do not correspond to the dots on the screen.



Assume that the value of the vertical coordinate to be specified is N and the vertical position on the screen is V. The relationship between these two can be expressed by the following formula.

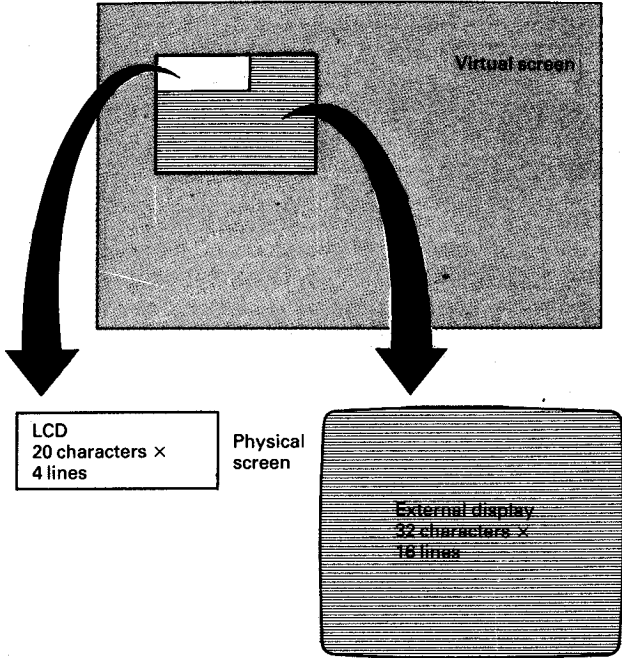
$$V = 2N \setminus 3 \text{ (" \setminus " indicates integer division.)}$$

### 1.4 Text screen (virtual screen)

The LCD screen of the HX-20 is capable of displaying only 80 characters (20 characters x 4 lines) at one time. However, by moving the cursor, a new screen may appear one after another. You can operate the HX-20 as if you are using a large screen.

This is because the HX-20 has adopted the concept of a "virtual screen". Let's suppose there is a very large screen inside the HX-20. Commands such as LIST, PRINT are all output to this internal screen. Now, assume the LCD display as a "viewing window" that allows you to see only a part of the internal screen. This "viewing window" can be moved anywhere on the large internal screen. It cannot, however, leave the bounds of the internal screen. The same concept of a "virtual screen" also applies to the external display. The only difference is that the size of the "window" through which you can see is larger than that of the LCD's window. As compared with the virtual screen, this "window" on both the LCD and external display is called the "physical screen".

Thanks to this virtual screen, you can see statements which were previously input or output by moving the physical screen. However, if the virtual screen becomes full of data, the screen will be scrolled up by one line for the next display and thus you cannot see the line overflowed from the virtual screen by moving the physical screen.



The size of the virtual screen can be specified by a WIDTH command. You can specify the size of the virtual screen freely within the range of 255 characters per line or 255 lines, subject to the limitations imposed by the capacity of the memory. (On the external display, the maximum screen size is 40 characters by 37 lines (i.e., 1,480 characters in total). If too large a virtual screen is specified, the available programme area will be restricted. With this in mind, specify the size of the virtual screen as required.

The virtual screen is valid for only the text screen and the size of the graphic screen is fixed.

## 1.5 Screen edit

EPSON BASIC has a screen edit function to facilitate programme editing. Using this function, a text or programme line being displayed on the screen can be corrected. The corrected text or programme line can be entered into the programme by pressing **RETURN** key. In other words, you can edit a programme that has already been input and stored in the memory according to the following procedure.

- (1) Display the programme to be edited using a LIST command.
- (2) Make necessary corrections by moving the cursor.
- (3) Press **RETURN** key.

To enter a text into BASIC, you must first press **RETURN** key to inform BASIC of the input of the text. BASIC will ignore any and all characters or symbols entered until you press **RETURN** key. When **RETURN** key is pressed, BASIC pays attention to only the text being displayed on the line where the cursor is located. For this reason, BASIC makes no distinction between the text called by a LIST statement and the previously entered text or the one displayed by just typing.

The "line where the cursor is located" is not limited to one line currently being displayed on the screen but refers to a text which is continuously typed from the keyboard and does not exceed a maximum of 255 characters in length. BASIC regards a text written over plural lines as a "logical single line". Wherever the cursor is positioned on the "logical single line" when **RETURN** key is pressed, that line will be input up to its end. When the line on the screen is shorter than the logical single line, you sometimes cannot judge by merely looking at the screen whether the text displayed is part of the logical single line or not. Please pay attention to this point.

### 1.5.1 Moving the cursor

The following keys are used to freely move the cursor alone during programme editing.

- TAB** Moves the cursor to the next TAB position. (Every 8 columns.)
- ←** Moves the cursor one position to the left.
- Moves the cursor one position to the right.
- SHIFT** + **↑** Moves the cursor up one line.
- SHIFT** + **↓** Moves the cursor down one line.

If the cursor at either end of the physical screen moves out of the physical screen as a result of any cursor movement key operation, the physical screen automatically moves within the virtual screen. In this way, the cursor will be prevented from leaving the physical screen. Also, when warm started, BASIC automatically sets the "scroll margin" to 3. This means that when the cursor reaches the 3rd position from either end of the physical screen, the physical screen will move with the margin left on either end. The scroll margin can be specified with a WIDTH statement.

If the cursor is at either end of a line on the virtual screen, any attempt to move the cursor in that direction will cause it to move to the next (or the preceding) line. The display will, of course, follow any such cursor movement. However, the text advanced to a new line in this way by one of the cursor movement keys will not be considered to be a logical single line. The cursor cannot be moved beyond the upper left- or lower right-hand corner of the virtual screen.

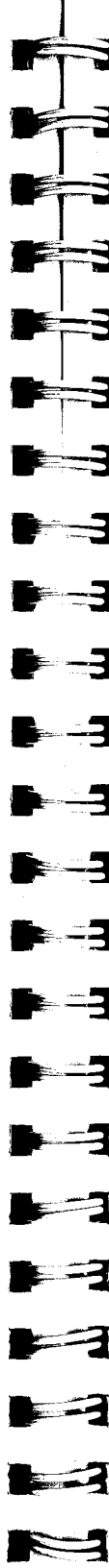
### 1.5.2 Moving the physical screen

- CTRL** + **A** Moves the physical screen to the left corner of the virtual screen.
- CTRL** + **F** Moves the physical screen to the right corner of the virtual screen.
- CTRL** + **←** or **CTRL** + **S** Moves the physical screen to the left by the specified number of columns.
- CTRL** + **→** or **CTRL** + **D** Moves the physical screen to the right by the specified number of columns.
- SC : RN** or **CTRL** + **P** Moves the physical screen up by the specified number of lines.
- SHIFT** + **SC : RN** or **CTRL** + **Q** Moves the physical screen down by the specified number of lines.

These scroll steps are set at the following values when BASIC is warm started: for the LCD, four lines vertically and 10 columns horizontally and for the external display, 16 lines vertically and 16 columns vertically.

The scroll steps setting can be changed using a SCROLL command. The movement of the physical screen differs from that of the cursor. Namely, when the physical screen reaches either end of the virtual screen, it stops there and will move no further.

- SHIFT** + **HOME** or **CTRL** + **K** Moves the cursor to its home position which is the upper left-hand corner of the virtual screen.



### 1.5.3 Insertion and deletion

When correcting a text being displayed on the screen, the character at the cursor position can be changed by typing another character or symbol over it. However, for major corrections, the following key operations are used.

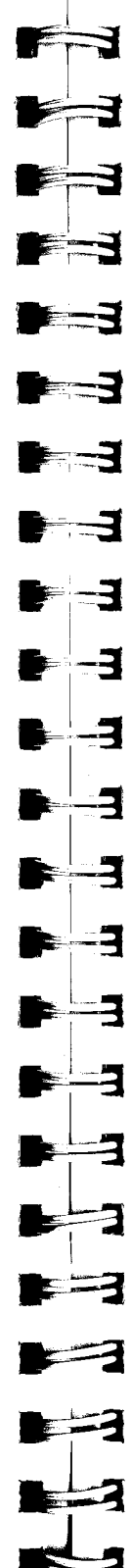
**SHIFT** + **INS DEL** or **CTRL** + **R** Upon pressing these keys, the HX-20 enters the insert mode. Any characters typed after this operation will be inserted to the left of the cursor position and the characters to the right of the cursor will move as the cursor moves. This mode continues until you press **RETURN** key, any of the cursor movement keys, or **INS DEL** key.

**INS DEL** or **CTRL** + **H** The characters immediately to the left of the cursor position are deleted and the cursor moves to the left. Mistypes are normally corrected by this operation.

**CTRL** + **E** All the characters to the right of the cursor position on the line where the cursor is currently positioned are deleted. In this case, the cursor does not move.

**CTRL** + **Z** All the characters from the line where the cursor is currently positioned to the bottom line of the virtual screen are deleted. In this case, the cursor does not move.

**HOME CLR** or **CTRL** + **L** Deletes all the characters on the virtual screen and returns the cursor to its home position.



## 1.6 File operations

A collection of information is referred to as a "Record" and a collection of records, as a "File". In EPSON BASIC, I/O transfer to and from peripheral devices is performed in units of one record and I/O data is managed in units of one file. In EPSON BASIC, a special RAM file area is provided. GET% and PUT% statements can read and write RAM files for efficient data storage and I/O transfer.

### 1.6.1 File numbers

To enhance the efficiency of I/O, the HX-20 has a special area called "Buffer" in the memory in which records are temporarily stored. The individual numbers assigned to the locations in the buffer area are called "File Numbers". These file numbers are used to access the files.

Each file number must be an integer in the range of 1 to 16.

### 1.6.2 File descriptors

As the concept of "File" has been adopted for all I/O devices, all I/O operations can be effected by standardized commands. Distinction of one I/O device from another is made by a "file descriptor" which is a string consisting of the following elements.

"<device name>:[<filename>]"

A file descriptor must always be enclosed in double quotation marks.

### 1.6.3 Device names

<device name> indicates the name of an I/O device and is represented as a rule by a string of either 4 alphabetic characters or of 3 alphabetic characters and 1 numeric character, followed by a colon. The following device names are defined in EPSON BASIC.

Device name	Device	Input	Output	Remarks
KYBD:	Keyboard	○	×	
SCRN:	Screen	×	○	
LPT0:	Built-in microprinter	×	○	
COM0:	RS-232C port	○	○	
CAS0:	Microcassette	○	○	Option
CAS1:	Audio cassette	○	○	
PAC0:	ROM cartridge	○	×	Option
A:	Flexible disk drive A	○	○	Device names used in DISK BASIC
B:	Flexible disk drive B	○	○	
C:	Flexible disk drive C	○	○	
D:	Flexible disk drive D	○	○	

○: Applicable      ×: Not applicable

If device name is omitted, EPSON BASIC automatically checks the peripheral equipment connected to the HX-20. If a microcassette drive or a ROM cartridge is connected, BASIC gives precedence to these devices over the audio cassette. If none of these devices is connected, the audio cassette is set. The device set when device name is omitted is called the "default device".

## 1.6.4 Filenames

<filename> is the name given to each file by the user. No filename can be omitted if I/O data transfer is to be performed between the HX-20 and another auxiliary memory unit such as a cassette or ROM cartridge. In other cases, it can be omitted.

The filename must be used in the following format.

<filename>[.<filetype>]

The first <filename> consists of an 8-character string and <filetype> following a period consists of a string of 3 characters. Any characters other than colons, periods, brackets, and character codes 0 and 255 can be used as <filename> and <filetype>.

Usually, <filename> indicates the name of a file and <filetype>, its attribute. This, however, is left to the discretion of the user. <filetype> is normally included in any reference to <filename>.

If you specify a filename exceeding 8 characters or a filetype exceeding 3 characters, an "FD" error occurs.

## 1.7 Peripheral equipment

The concept of "File" has been adopted to handle the I/O operation between the HX-20 and all peripheral devices and for this reason, use of standardized commands is effective for almost all the devices.

The following table shows the basic commands and statements for handling files and the devices for which the respective commands and statements are valid.

Command \ Device	KYBD:	SCRN:	LPT0:	COM0:	CAS0:	CAS1:	PAC0:
LOAD	x	x	x	o	o	o	o
LOADM	x	x	x	x	o	o	o
LOAD?	x	x	x	x	o	o	x
RUN "<file descriptor>"	x	x	x	o	o	o	o
MERGE	x	x	x	o	o	o	o
FILES	x	x	x	x	o	o	o
INPUT#	o	x	x	o	o	o	o
INPUT\$	o	x	x	o	o	o	o
EOF	-	x	x	o	o	o	o
LOF	-	x	x	o	-	-	o
SAVE	x	o	o	o	o	o	x
SAVEM	x	x	x	x	o	o	x
LIST	x	o	o	o	o	o	x
PRINT# (USING)	x	o	o	o	o	o	x
POS	x	o	o	o	-	-	x
OPEN mode	I	O	O	I/O	I/O	I/O	I

### NOTE:

o or x in this table indicates that when a device is specified for a command or statement, the device

- o: Can be used.
- x: Cannot be used. An FC error occurs.
- : Causes no error but the command is invalid.

Refer to Chapter 3 for each command or statement.

### 1.7.1 Screen

If a file output device is specified as "SCRN:" (screen), it refers to either the LCD or external display which has been specified by a SCREEN command for text screen display. Characters to be output are actually written on the virtual screen and the screen that you can see at a time is limited by the display capacity of the LCD (or external display). Pay adequate attention to the output allocation and output speed.

The screen can also be used for programme debugging. When data is output on cassette or disk file, you will need a separate programme to check the data for correct output. In such a case, by changing the device name in the OPEN statement to "SCRN:", you can confirm visually the output data.

### 1.7.2 Printer

With the HX-20, you can use two types of printers. The first is the built-in microprinter which you must specify as "LPT0:". The other is an external printer connected to the HX-20 through the RS-232C interface, which you must specify as "COM0:".

As you may use the built-in microprinter more frequently, the following special output commands are provided in addition to the general output commands.

LPRINT, LPRINT USING  
LLIST  
KEY LLIST  
COPY

By using a WIDTH<device name> command, you can also specify the print width. In this way, your design of output format can be greatly simplified.

### 1.7.3 Cassette

With BASIC, you can use two types of cassettes as auxiliary memory units.

"CAS1:"

Using a commercially available audio cassette, you can SAVE and LOAD programmes and data freely. When loading programmes with a LOAD command, or when a file is opened for input, specify the filename and the tape will be automatically searched for that file.

While executing a LOAD or OPEN command in the direct mode, the following message is displayed on the LCD screen each time a file other than the specified file is searched.

Skip: <filename>

When the specified file is found, the following message appears on the LCD screen.

Found: <filename>

Also, a LOAD? command may be used to skip the specified file. For further details, see LOAD? command in Chapter 3.

“CAS0:”

This device name indicates the optional microcassette drive. “CAS0” can be used in the same manner as “CAS1:”. You can SAVE, LOAD, VERIFY or SEARCH the files using the built-in counter and executing a WIND command.

#### **1.7.4 RS-232C port**

Communications with external devices are regarded as most important in EPSON BASIC. For this reason, the RS-232C port is designed to allow programming in BASIC to set the conditions for communication not only with the external printer, but also with all other external devices. These conditions can be specified not only by an OPEN command, but also by the following commands.

LOAD  
LIST  
SAVE  
RUN  
MERGE

For further details, see Chapters 3 and 5.

#### **1.7.5 ROM cartridge (option)**

You can use the optional ROM cartridge by specifying the device name as “PAC0:”. Except that the ROM cartridge is used exclusively for input and that file loading can be accomplished in a much shorter time, it can be used in essentially the same manner as the microcassette drive. For these reasons, this device is extremely useful when you handle special programmes and/or data.

## **CHAPTER 2**

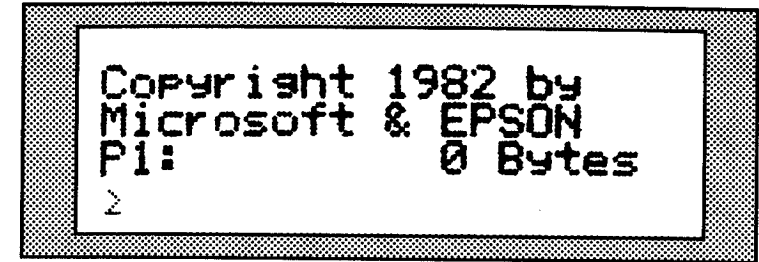
# **Outline of EPSON BASIC**





## 2.1 Operation modes

When BASIC is started up, your HX-20 displays the prompt sign ">" following the opening message. The prompt sign tells you that the HX-20 is now in command level, that is, an await state for command input.



If you input a programme statement written in accordance with the syntax established for BASIC but without line number, the statement is executed immediately upon pressing **RETURN** key. This type of programme execution is called "execution in the direct mode". Almost all BASIC commands and statements can be executed in the direct mode. The exceptions are those statements listed below.

- INPUT
- LINE INPUT
- INPUT#
- LINE INPUT#
- RANDOMIZE

If you input a statement preceded by a line number (ranging from 0 to 63999), that statement and the line number are stored together in the memory as a program. The stored programme can then be executed by either a RUN command or a GOTO or GOSUB statement. This type of execution is called "execution in the program mode". When a programme is executed by a RUN command, all variables (numeric, string and array variables and other defined statements) are cleared prior to execution. In contrast, a GOTO statement executes a programme without changing the current values of all variables.

## 2.2 Statements

A programme statement is a description of any of expressions, commands, statements, functions, etc., which is executed by BASIC. More than one statement can be specified by connecting them with a colon. These connected statements are called "multiple statements".

## 2.3 Lines

A programme line of BASIC always begins with a line number (represented by an integer in the range of 0 to 63999), followed by one or more statements and ends with a carriage return. A programme line can contain a maximum of 255 characters. Line numbers show the order in which the programme lines are stored in memory. Programme execution likewise follows this order starting from the lowest-numbered line. Line numbers are also used to access programmes for branching and editing. A full stop may be used instead of the line number after commands such as LIST and AUTO to instruct BASIC to operate on the last line, e.g., the line in which an error has occurred during programme execution or the last line input in the programme.

Examples: LIST.  
          AUTO.

## 2.4 Character set

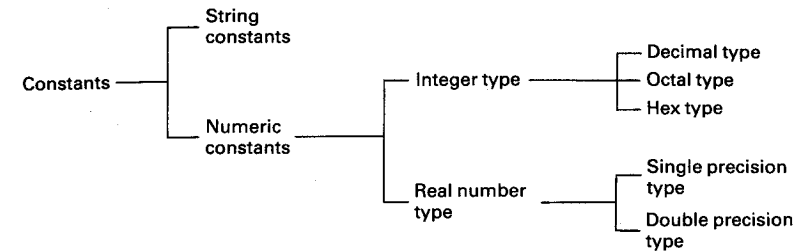
The character set which can be used in BASIC consists of alphabetic (uppercase and lowercase) characters, numeric characters (0 to 9), special symbols and graphic characters. There are also a set of special control characters that perform the special functions as defined by their codes.

For details of these characters, see APPENDIX F, "Character Code Table" and APPENDIX G, "Control codes".

Also see section 2.10, DIP switch setting, and POKE command in Chapter 3 for character set selection.

## 2.5 Constants

Constants are values which are used by EPSON BASIC without any change for programme execution. The following types of constants are used in EPSON BASIC.



### 2.5.1 String constants

A string constant is a sequence of up to 255 alphanumeric characters and symbols enclosed in double quotation marks ("").

Examples: "HX-20"  
          "Program-1"

Double quotation marks and control codes cannot be included in string constants. Use the CHR\$ function to handle these marks and codes as characters.

### 2.5.2 Integer type numeric constants

#### (1) Decimal type

Whole numbers between -32768 to 32767 or numbers followed by the "%" sign. A decimal point cannot be suffixed to these numbers.

Examples: 12345  
          -9999  
          45678%

#### (2) Octal type

Octal numbers 0 to 7, prefixed by "&O" or "&". Octal numbers are in the range of &0 to &177777.

Examples: &O123  
          &77777

#### (3) Hex type

Hexadecimal numbers 0 to F, prefixed by "&H". Hex numbers are in the range of &H0 to &HFFFF.

Examples: &H1F  
          &HABCD

### 2.5.3 Single precision type real numbers

Single precision constants are stored in 7 significant digits. Six of the 7 significant digits are displayed after rounding off the 7th digit.

A single precision constant is any numeric constant that has:

- (1) seven or fewer digits, or
- (2) exponential form using E, or
- (3) a trailing exclamation mark (!).

Examples: 12345.6  
-5E-38  
234.5!

### 2.5.4 Double precision type real numbers

Double precision constants are stored in precision of 16 significant digits. Up to 16 significant digits are displayed.

A double precision constant is any numeric constant that has:

- (1) eight or more digits, or
- (2) exponential form using D, or
- (3) a trailing number sign (#).

Examples: 3.141592653  
-1.23 D 22  
888.8#

## 2.6 Variables

Variables are a kind of location to take the values used in BASIC programmes. Each variable corresponds to a variable name consisting of alphanumeric characters.

If a numeric variable is referenced before a value is assigned to it, 0 is assigned as a numeric value. For a string variable, a null string, that is, a string of length 0, is assigned as a string value.

### 2.6.1 Variable names and declaration characters

- (1) A variable name is represented by a sequence of up to 255 characters beginning with an alphabetic character. Only the first 16 characters of a name are identified.

A variable name can contain a reserved word but cannot begin with a reserved word. (Reserved words are keywords such as commands, statements and functions.)

Variable names starting with alphabetic characters "FN" are not allowed.

In variable names, capitals and lowercase letters are regarded as the same.

- (2) Variable types are determined by type declaration characters. A declaration character written as the last character of a variable name declares the type of the variable. If a declaration character is omitted, the variable type is assumed as a single precision type real number.

- (3) BASIC distinguishes different variable types under the same variable name.

Type declaration characters	{	% Integer variable
	!	Single precision variable
	#	Double precision variable
	\$	String variable

### 2.6.2 Array variables

An array refers to a variable whose several elements can be referenced by one variable name. Each element in the array is referenced by an "array variable" that is subscripted by integer notation. The dimensions of an array variable and the maximum value of a subscript depend on the capacity of the memory. (See DIM in Chapter 3.)

## 2.7 Type conversion

When necessary, EPSON BASIC automatically converts a numeric constant from one type to another.

- (1) If you assign a numeric data of one type to a numeric variable of another type, the numeric constant is converted to and stored as the type declared in the variable name. If you attempt to assign a numeric variable to a string variable or vice versa, a "Type mismatch" error occurs.

Example: A%=32.34

In this example, 32 will be stored in A%.

- (2) During expression evaluation, all of the operands in an arithmetic or relational operation are converted, and their results are returned to the same degree of precision of the most precise operand.

Examples: 10#/3  
10#/3#

The above two expressions are evaluated at the same degree of precision. However, if you neglect to pay attention to the number of significant digits in the variable and to the order of executing operations, an error may result.

A#=10/3\*3#  
B#=10#/3\*3

In the above example, 9.99999761581421 will be stored in variable A# and 1.0E+01 will be stored in B#.

- (3) In logical operations, all the logical operands are converted to integers and an integer result is returned.

An "Overflow" error will occur if the operands are not in the range -32768 to 32767 when converted to integers.

Example: A=NOT 123.456

In this example, -124 will be stored in A.

- (4) When a real number is converted to an integer, the fractional portion is truncated. If the real number when converted to an integer is not in the range -32768 to 32767, an error also occurs.

Example: A%=777.7

778 will be stored in A.

- (5) If a double precision variable is assigned to a single precision variable, only the first seven digits rounded down will be stored.

Example: P!=3.141592653589793

3.141593 will be stored in P! and displayed as 3.14159

## 2.8 Expressions and operations

An expression refers to ① a string or numeric constant, ② a string or numeric variable, and ③ a combination of string or numeric constants and variables connected by operators.

Examples: "ABCD"

1.41421356

2\*3

A+B/C

BASIC employs the following five types of operators in performing mathematical or logical operations.

- (1) Arithmetic operators
- (2) Relational operators
- (3) Logical operators
- (4) Functional operators
- (5) String operators

### 2.8.1 Arithmetic operators

- (1) EPSON BASIC uses the following arithmetic operators

Operator	Operation	Sample expression
^	Exponentiation	A^B
-	Negation	-A
*, /	Multiplication and division of real numbers	A*B, A/B
+, -	Addition and subtraction	A+B, A-B

Parentheses are used to change the order of operations. Operations within parentheses are performed first.

Shown below are the representations of the algebraic expressions in BASIC.

EPSON BASIC	Algebraic expression
3*X+Y	3X+Y
X/Y-Z	(X÷Y)-Z
X^2+Y*3+4	X <sup>2</sup> +3Y+4
Y^Y^Z	(X <sup>Y</sup> ) <sup>Z</sup>
X*(-Y)	X(-Y)

- (2) Integer division and modulus arithmetic

Integer division is denoted by the backslash (\ ). If operands are real numbers, they are converted to integers before execution. Quotients are truncated to integers.

Example: A%=10\3

B%=9.5\3.3

The value stored in both A% and B% will be 3.

Modulus arithmetic, denoted by the operator MOD, returns the integer value of the remainder of an integer division.

Example: A%=10 MOD 3

1 will be stored in A%.

(3) Division by zero and overflow

If a division by zero occurs during the evaluation of an expression, a "Division by zero" error occurs and execution of the operation is terminated. If the result of an evaluation or a value assignment exceeds the number of digits that can be handled by the variable, an "Overflow" occurs and execution stops.

**2.8.2 Relational operators**

Relational operators are used to compare two values. The result of this comparison is either "true" (-1) or "false" (0). This result is then used to make a decision regarding programme flow.

Operator	Relation tested	Sample expression
=	Equality	X=Y
< > >>	Inequality	X<>Y, X><Y
<	Less than	X<Y
>	Greater than	X>Y
<=, =<	Less than or equal to	X<=Y, X=>Y
>=, =>	Greater than or equal to	X<=Y, X=>Y

Example: X=A=0  
 If A=0, -1 will be stored in X, and  
 if A<>0, 0 will be stored in X.

**2.8.3 Logical operators**

Logical operators are used to perform tests on multiple relations, bit manipulation, or Boolean operations. Logical operators return a result for each bit that is either a true (1) or false (0).

NOT (Negation)		
X		NOT X
1		0
0		1
AND (Logical product)		
X	Y	X AND Y
1	1	1
1	0	0
0	1	0
0	0	0
OR (Logical sum)		
X	Y	X OR Y
1	1	1
1	0	1
0	1	1
0	0	0

XOR (Exclusive-OR)		
X	Y	X XOR Y
1	1	0
1	0	1
0	1	1
0	0	0
IMP (Implication)		
X	Y	X IMP Y
1	1	1
1	0	0
0	1	1
0	0	1
EQV (Equivalence)		
X	Y	X EQV Y
1	1	1
1	0	0
0	1	0
0	0	1

Logical operators can be used to connect two or more relational operators to make decisions on complex conditions with a single statement. If the value given by the logical operator is -1 (true) or 0 (false), the result obtained must be either -1 or 0.

Example: IF A<0 AND B=0 then 100

In this example, if A is negative and B is 0, program control branches to line 100.

Logical operators convert their operands into two's complement integers in the range -32768 to 32767 before any operations. If the operands are not in this range, an error occurs. The given operation is performed bit by bit on these integers.

Example: A=21 AND 13

21=(10101)<sub>2</sub>, 13=(01101)<sub>2</sub>  
 (10101)<sub>2</sub> AND (01101)<sub>2</sub>=(00101)<sub>2</sub>  
 (00101)<sub>2</sub>=5

Thus the value entered in A is 5.

**2.8.4 Functional operators**

A function is used in an expression to call a predetermined operation that is to be performed on a given argument.

BASIC has "Intrinsic Functions" consisting of numeric functions such as SIN, SQR, etc., and string functions such as RIGHT\$, STR\$, etc. For details of these functions, please refer to Chapter 4.

As explained later at DEFFN in Chapter 3, BASIC also allows "User defined" functions. If a real number is assigned to a function which normally takes only an integer as its argument, the fractional portion of the real number is rounded off to the nearest integer and then the functional operation is performed. A double precision real number may be assigned as the argument of a numeric function. In this case, however, the operation is performed in single precision.

### 2.8.5 String operations

Strings may be concatenated using a plus sign (+).

Example: A\$="ABC"+CHR\$(34)  
"ABC" will be stored in A\$.

Comparison of strings can also be made using relational operators.

=, <, >, <>, ><, <=, =<, >=, ==>

Strings are compared by taking one character at a time from each string. If two strings compared are the same, both the strings are judged "equal". If any of the codes differs, the lower character code number precedes the higher. In string comparison, the shorter string is considered smaller. Leading and trailing blanks are significant in string comparison.

Example: "AAA"="AAA"  
"X" < "Y"  
"XYZ" > "XY"  
"ABC" > "A BC"

### 2.8.6 Precedence order of operations

Mathematical and logical operations are performed in the following order.

1. Operations within parentheses
2. Functions
3. Exponentiation (^)
4. Negation (-)
5. Multiplication and division of real numbers (\*, /)
6. Integer division (\)
7. Modulus integer division (MOD)
8. Addition and subtraction (+, -)
9. Relational operands
10. NOT
11. AND
12. OR
13. XOR
14. IMP
15. EQV

## 2.9 Error messages

If EPSON BASIC detects an error which causes the execution of a programme to stop, an error message is printed and the HX-20 returns to command level. The format for error messages in the direct mode is:

XX Error

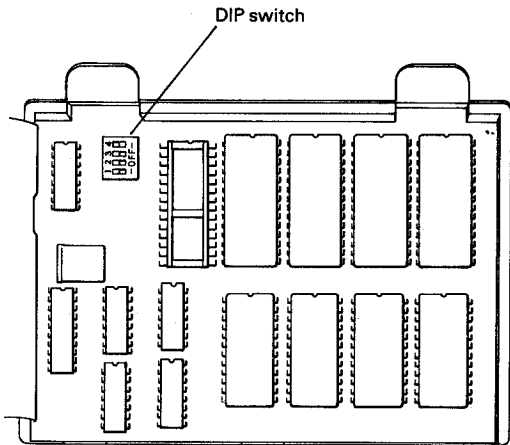
The format in the programme mode is:

XX Error in nnnn

XX is the error code and nnnn is the line number where the error was detected. For details of the BASIC error messages, refer to APPENDIX A.

## 2.10 DIP switch setting

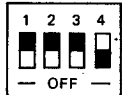
The DIP switch setting can be changed by opening the cover on the back of the HX-20 as shown below.



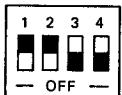
Location of the DIP Switch

The DIP switch settings for the character codes for each country are as shown below.

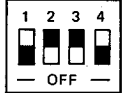
USASCII



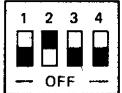
Denmark



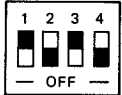
France



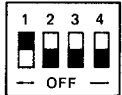
Sweden



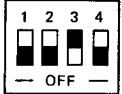
Germany



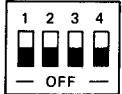
Italy



England



Spain



You can also do this by software. (See POKE in Chapter 3.)  
For the respective countries' character codes, see APPENDIX F.

## How to use Chapters 3 and 4

All the commands and statements of EPSON BASIC are explained in alphabetical order for easy reference in Chapters 3 and 4. All I/O commands related to peripheral devices are explained collectively.

For details of the operations and features peculiar to the respective peripherals, refer to Chapters 1 and 5.

Commands and statements prefixed with an asterisk "\*" are those unique to BASIC. As these commands and statements operate differently from conventional BASIC commands and statements, please read the explanation of each of these commands with great care. The descriptions of BASIC commands and statements are presented in the following format.

**FORMAT** Explains how to write the command or statement. Follow the syntax described below when you input each command or statement.

1. Input items shown in uppercase letters using either uppercase or lowercase letters. Make this distinction, particularly for those items enclosed in double quotation marks (filenames, etc.).
2. Items surrounded by "< >" are those to be specified by the user.
3. Items enclosed in brackets "[ ]" are optional and can be omitted. If you omit these items, the HX-20 will supply the default values or previously specified values.
4. Symbols other than those described above, e.g., parentheses, commas, colons, semicolons, hyphens, equal signs, etc., must be input exactly at the positions as they are shown in the manual.
5. Optional items indicated by ". . ." can be repeated as many times as desired within a maximum of 255 characters.  
Example: <variable>[,<variable>. . .]  
A, B\$, C!, D, etc., may be repeated.
6. When two or more items appear between two vertical lines, this means that you must specify the desired function by selecting one of the items for input.

Example: 

THEN	<statement>
	<line number>
GOTO	<line number>

In this example, the possible choices for input are:

- THEN <statement>
- THEN <line number>
- GOTO <line number>

7. As a rule, BASIC ignores spaces. However, spaces are not permitted within variable names and keywords. If a variable is to be followed by a keyword, a space must be inserted to delimit them for correct operation.

**PURPOSE** Describes briefly how the command or statement functions.

**EXAMPLE** This shows a simple example of actual input.

**REMARKS** Hints on the correct use of each command or statement are given, along with a description of their functions.

**SAMPLE PROGRAMME** Sample programmes using the command or statement are presented for exercise.

In the HX-20, 5 programmes are independently managed in the respective programme areas. Therefore, to avoid the accidental destruction or loss of important data or programmes in the memory, be sure to check the programme area using a STAT or LIST command, before modifying any programme.

## CHAPTER 3

# Commands and Statements

