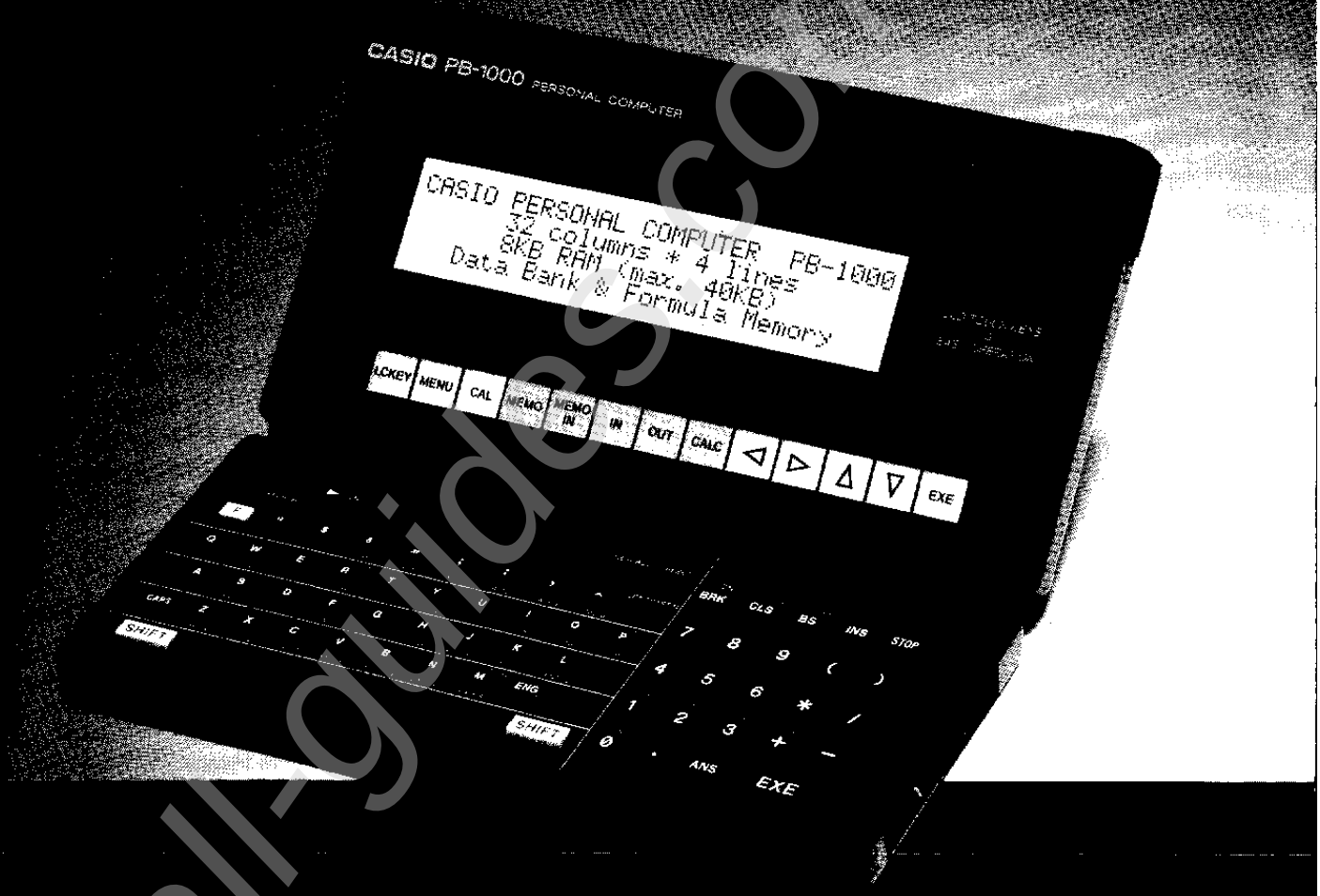
CASIO

OWNER'S MANUAL



PERSONAL COMPUTER

PB-1000

FC 英

PERSONAL COMPUTER
PB-1000
OWNER'S MANUAL

CASIO®

FOREWORD

Congratulations on your selection of a CASIO PB-1000 personal computer.
The PB-1000 features an innovative touch screen system that allows easy operation by simply lightly touching the screen where key simulations appear. Other processing procedures are menu driven, so program creation, editing, and execution are all accomplished by selecting the appropriate menu items. There are 16 touch keys on the screen and they can be easily controlled using BASIC commands. Of course, all input and processing results are produced on the 32-column × 4-line screen, which is part of a 32-column × 8-line virtual screen.
Besides a number of high-level scientific functions and BASIC programming, the PB-1000 is also capable of machine language programming using a built-in assembler function. Addition of an optional 3.5-inch floppy disk drive unit allows reliable high-volume storage of data to a degree never before thought possible on a computer of such compact size. Optional expansion units are also available with RS-232C interfaces for data communications.
In fact, the features, functions and options of the PB-1000 make it the perfect data processing tool for home, office, laboratory or classroom.
Be sure to read this manual carefully to become familiar with the operation of your computer to use its functions to their full potential.

PRECAUTIONS

This computer is a product of CASIO's high level of electronics engineering, testing, and quality control. The following points should be carefully noted to allow this unit to provide the years of trouble free operation for which it is designed.

- This unit is constructed of precision electronic components and should never be disassembled, dropped, or otherwise subjected to strong impact. Strong shocks can cause termination of program execution or alteration of the unit's memory.
- Do not use or store this unit in areas subjected to high temperatures, humidity or dust.
- Display response may become slow or fail completely at extremely low temperatures. Normal operation should resume after the unit reaches normal temperature.
- The connectors of this unit are designed exclusively for connection of the specified FA-7 or MD-100 expansion units only.
- The display may become dim when the buzzer sounds, but this does not indicate malfunction and is no cause for worry.
- Batteries should be replaced as soon as possible after weakened batteries are indicated by a dim display during normal operation.
- Replace batteries at least once every two years even if the unit is not used during this period. Dead batteries left in the unit may cause serious damage due to fluid leakage and should be removed as soon as possible.
- Keep the connector of the unit covered with the connector cap whenever the unit is not connected to an expansion unit, and avoid touching the connector.
- Strong static electrical charge may cause alteration of memory contents or key operation failure. If this situation should occur, press the RESET button followed by the NEW ALL button.
- Always ensure that the power supply of this unit is switched OFF before connecting peripheral devices.
- Never use thinner, benzine, or other volatile agents for cleaning the exterior of the unit. Use a soft cloth dipped into a mild solution of water and a neutral detergent, and wring the cloth out completely.
- Do not switch the power of the unit OFF during program execution or during calculations.
- When a malfunction occurs, contact the store where the computer was purchased or a nearby dealer.
- Before seeking service, please read this manual again, check the power supply, check the program for logic errors, etc.

CONTENTS

PART 1

FUNDAMENTAL OPERATION

The procedures outlined in this part of the manual will provide you with a basic knowledge of the fundamental operations of this computer. More detailed explanations can be found in PART 3 and after of this manual.
The following procedure should always be performed as the first step whenever using this unit:

1. Switch power ON and confirm that the cursor (blinking " – " symbol) appears on the screen.
2. Press the CONTRAST key and then adjust the contrast of the display using the ▲ and ▼ cursor keys.

1-1 MENU KEY

The MENU key is the second key from the left in the bank of keys below the display. Pressing the MENU key causes the display to appear as illustrated below. This is called the MENU screen.

KEY INPUT

MENU

```
[basic ][data  ][edit  ][disk  ]
```

The four touch keys of this menu are used for data and program writing and editing. These are called touch keys because any of the functions named on the screen can be executed by simply pressing the screen where the function name is located.
Now press the LCKEY key located to the left of the MENU key.

LCKEY

```
[name  ][kill  ][load  ][save  ]
```

The functions included on this menu are mainly used for file handling operations. This menu is also composed of touch keys, so functions can be called by simply pressing the appropriate location on the screen. Pressing the LCKEY key again advances to the expanded function menu.

LCKEY

```
[asmbl ][llist ][c.boot][preset]
```

Now pressing the [OKEY] key once again returns to the original MENU screen. The [OKEY] key is used as we have just seen to switch between the various menu screens.
Return to the original MENU screen and lightly press the LCD where the word [basic] is displayed. At this time, the word "Ready" will appear on the display to indicate that the unit is ready and standing by for input of a BASIC program. Further details concerning the MENU function can be found in PART 7 of this manual.

1-2 CAL MODE

Pressing the [CAL] key, which is located to the right of the [MENU] key, causes the cursor to appear in the upper left corner of the screen. This is the CAL mode in which calculations can be performed by directly entering values and operators. It should be noted here that the CAL mode is automatically entered whenever the power of the calculator is switched ON.

KEY INPUT

Press the keys noted below to obtain the result of 2.5 + 3.5 − 2:

[2][.][5][+][3][.][5][−][2][EXE]

```
2.5+3.5-2
 4
```

Note that the [EXE] (execute) key is pressed as the final step. It is best to think of the [EXE] key as taking the place of " = " for calculations using direct input. Besides addition, subtraction, multiplication and division, a variety of other functions can be used in the CAL mode to perform complex calculations (see PART 3).

1-3 DATA BANK FUNCTION

The data bank function makes it possible to store large volumes of information for instant recall when required. Pressing the [MEMO] key in the bank of keys below the screen enters the MEMO IN mode, and the cursor appears in the upper left corner of the screen ready for data input. Either try entering the data noted below or enter your own name and telephone number.

KEY INPUT

[A]SMITH
[SPC]347−237
−4811[EXE]

```
SMITH 347-237-4811
                  ( 1)
```

Record No.

* [SPC] indicates the space bar located at the bottom of the keyboard.

Note that the [EXE] key is also pressed here as the final step to store the entered data into the data bank. Each entry of a data item causes the record number on the lower right of the screen to increase by one. Details concerning the data bank functions can be found in PART 5 of this manual.

1-4 FORMULA STORAGE FUNCTION

The formula storage function makes it possible to record a calculation formula in memory and then obtain calculation results by simply entering values for the variables in the formulas. This is accomplished using the IN , OUT , and CALC keys in the CAL mode.
The formula used in the example here will determine the selling price of goods after input of the purchase price and percentage of profit.

EXAMPLE

SELLING PRICE − (SELLING PRICE X PROFIT%) = PURCHASE PRICE
SELLING PRICE × (1 − PROFIT%) = PURCHASE PRICE
SELLING PRICE = PURCHASE PRICE ÷ (1 − PROFIT%)

KEY INPUT

CAL S E L L = P U R C H A S E / (1 − P R O F I T)

IN

Required to store formula in memory.

Now that the formula is stored in memory, compute the selling price for ITEM A which costs $100 and will be sold for a 30% (.3) profit, and ITEM B which costs $96 and will be sold for a 25% (.25) profit.

OPERATION

CALC
1 0 0 EXE
0 . 3 EXE

```
PURCHASE?100
PROFIT?0.3
SELL= 142.8571429
```

CALC
9 6 EXE
0 . 2 5 EXE

```
PURCHASE?96
PROFIT?0.25
SELL= 128
```

Once a formula is stored in memory, it can be executed as many times as required to calculate the result for any number of data. The formula is retained in memory even when the power of the computer is switched OFF. Details on the formula storage functions can be found in PART 4 of this manual.

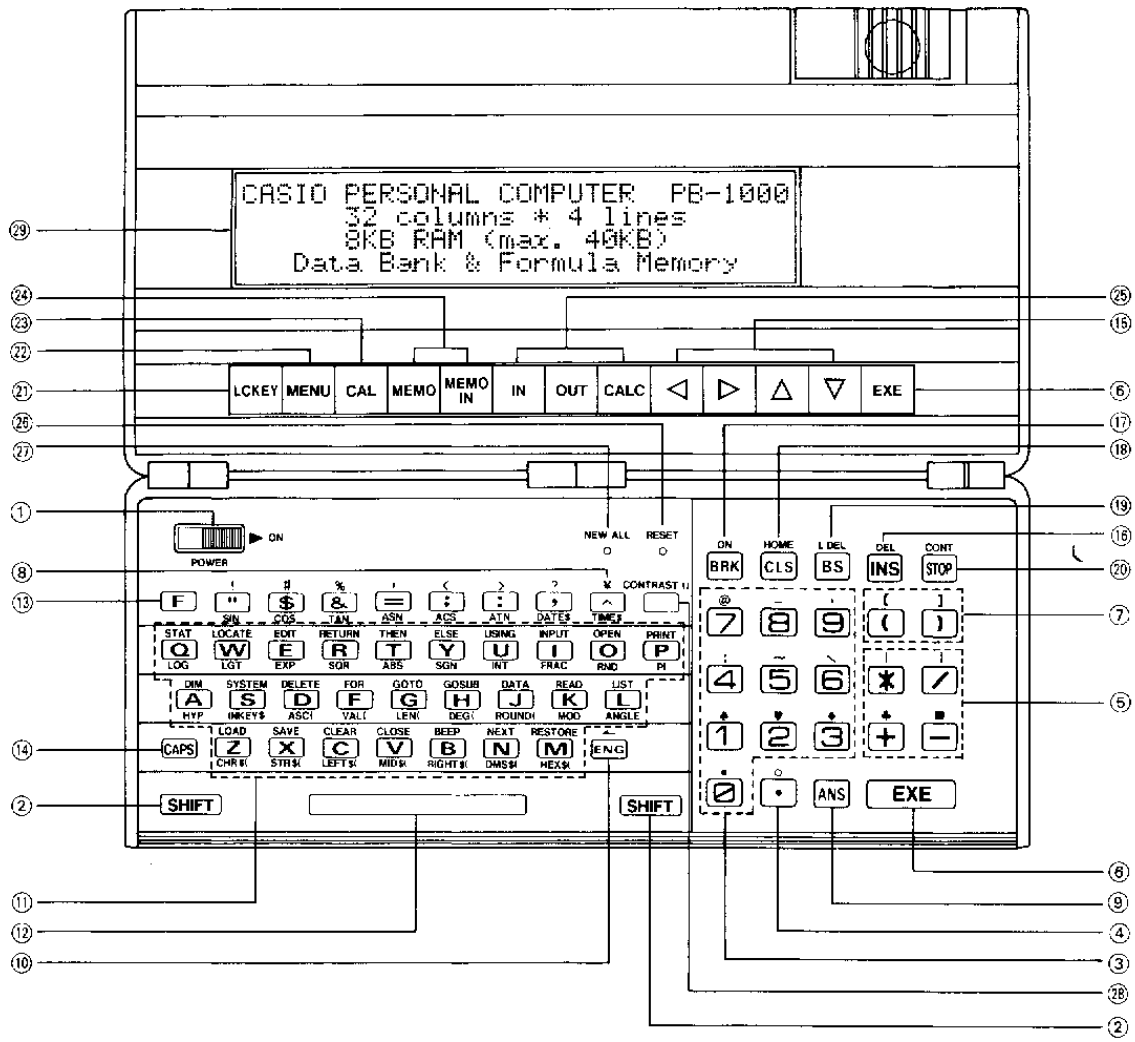
1-5 MAIN FUNCTIONS AND FEATURES

The following table lists the main functions of the unit and includes the features of each of the functions.

Function	Features	Page
Virtual screen	8-line virtual screen	PART 2 page 12
Touch screen keys	Menu selection by directly pressing LCD	PART 2 page 13
Manual calculations, function/statistical calculations	Arithmetic, scientific and statistical calculations	PART 3 page 17
Formula storage	Repeat calculation of formulas stored in memory without programming	PART 4 page 35
Data bank	Electronic notebook capable of storing various types of data	PART 5 page 39
Operation mode	Classification of often used operations	PART 6 page 49
Memory file	Storage of programs and data in individual files	PART 6 page 54
MENU	Easy selection of functions using touch keys	PART 7 page 57
BASIC	BASIC programming	PART 8 page 73
Power ON boot	Automatic execution of specified program when power switched ON	PART 9 page 82
Clock boot	Automatic execution of specified program at a specified date and time	PART 9 page 83
One-touch file	Assignment of file to touch key for one-touch execution	PART 9 page 85
Assembler	Writing of machine language programs using mnemonics	PART 10 page 87
Peripherals	System expansion using FA-7 and MD-100 expansion units	PART 12 page 115

PART 2
UNIT CONFIGURATION

2-1 GENERAL GUIDE



① Power Switch
② Shift Key
③ Numeric keys
④ Decimal Key
⑤ Arithmetic Operator Keys
⑥ Execute Key
⑦ Parentheses Keys
⑧ Power Key
⑨ Answer Key
⑩ Engineering Key
⑪ Alphabet Keys
⑫ Space Bar
⑬ Function Key
⑭ CAPS Key
⑮ Cursor Keys
⑯ Insert/Delete Key
⑰ Break Key
⑱ Clear Screen Key
⑲ Backspace Key
⑳ Stop Key
㉑ LC Display Key
㉒ Menu Key
㉓ CAL Mode Key
㉔ Memo Key/Memo In Key
㉕ Formula Storage Keys
㉖ Reset Button
㉗ New ALL Button
㉘ Contrast Key
㉙ Screen

2-2 OPERATIONAL FUNCTIONS

① Power Switch (ON)
Slides to the right to switch power ON and to the left to switch power OFF.

② Shift Key (SHIFT)
Switches the numeric keys and alphabet keys to the one-key commands or symbols noted above the keys. The two shift keys located to the left and right of the space bar have identical functions.

③ Numeric keys (0 ~ 9)
Enter the numeric values noted on each key.

④ Decimal Key (·)
Enters a decimal point.

⑤ Arithmetic Operator Keys (+ , − , × , ÷)
Enter the arithmetic operators noted on the keys.
+ : Addition
− : Subtraction
× : Multiplication
÷ : Division

⑥ Execute Key (EXE)
Finalizes entry of a calculation and produces the result. The function of this key is equivalent to a "=" key on a standard calculator.
This key is also used to enter lines of a program and for actual execution of programs. Note that there are two EXE keys, one at the lower right of the ten-key pad, and one at the far right of the key bank under the LCD. This is simply to facilitate input, and both keys perform identical functions.

⑦ Parentheses Keys (( ))
Enter parentheses in such parenthetical calculations as: 5 × (10 + 20).

⑧ Power Key (^)
Raises a value to a specified power.

⑨ Answer Key (ANS)
Recalls the result of the most recent manual calculation.

⑩ Engineering Key (ENG)
Converts a calculation result to an exponential display. Each successive press shifts the decimal three places to the right and decreases the exponent by three. Each press while the ENG key is held down shifts the decimal three places to the left and increases the exponent by three. This function is useful when converting metric units.

⑪ Alphabet Keys
Enter the alphabetic characters noted on each key.

⑫ Space Bar
Enters a space. (The space bar is indicated as SPC hereafter in this manual.)

⑬ Function Key (F)
Changes the functions of the alphabet keys to one-key function keys. The alphabet keys enter the one-key functions noted below the keys if pressed while the function key is held down. In this manual, the function key will be indicated by the symbol F to differentiate it from the alphabetic F key.

⑭ **CAPS Key (CAPS)**
Switches the alphabet keys between upper case and lower case characters.

⑮ **Cursor Keys (◁, ▷, △, ▽)**
Move the cursor on the screen. Each press moves the cursor in the direction noted on the keys pressed, while holding down the keys causes continuous, high speed movement. Each cursor key also takes on a different function when pressed in combination with the SHIFT key.

KEY	FUNCTION	SHIFT +
◁	Cursor left	Moves to beginning of logical line
▷	Cursor right	Moves to end of logical line
△	Cursor up	Scrolls screen up without cursor movement
▽	Cursor down	Scrolls screen down without cursor movement

See page 12 for details on physical lines and logical lines.

⑯ **Insert/Delete Key (DEL INS)**
Inserts a space at the current cursor position by shifting everything from the cursor position right one space to the right. In combination with the SHIFT key, deletes the character at the current cursor position and automatically fills in the space created by shifting everything to the right of the cursor one space to the left. Holding down this key for either function causes continuous high speed operation of the respective function.

⑰ **Break Key (ON BRK)**
Terminates manual operations, program execution, peripheral device input/output, printer output and LIST output. Also reactivates the power supply when it has been interrupted by the Auto Power OFF function (see page 9).

⑱ **Clear Screen Key (HOME CLS)**
Clears the contents of the screen and locates the cursor at the upper left corner of the screen. In combination with the SHIFT key, locates the cursor at the home position (of the virtual screen) without clearing the contents of the screen.

⑲ **Backspace Key (BS)**
Deletes the character located immediately to the left of the cursor and automatically fills in the space created by shifting everything from the cursor position right one space to the left. In combination with the SHIFT key, deletes everything from the current cursor position right on the current logical line.

⑳ **Stop Key (CONT STOP)**
Suspends program execution. In combination with the SHIFT key, resumes program execution from the point at which it was originally suspended.

㉑ **LC Display Key (LCKEY)**
Switches the contents of the touch keys (see page 13) appearing on the fourth line of the menu screen. See PART 7 for details.

㉒ **Menu Key (MENU)**
Switches to the menu screen for one-touch selection of a variety of functions. See PART 7 for details.

㉓ **CAL Mode Key (CAL)**
Switches to the CAL mode for manual calculations. See PART 3 for details.

㉔ **Memo Key/Memo In Key (MEMO, MEMO IN)**
Used to input and search for memo data. See PART 5 for details.

㉕ Formula Storage Keys (IN , OUT , CALC)
Used when working with the formula storage function. See PART 4 for details.

㉖ Reset Button (RESET)
Resets the internal hardware of the unit. This button is pressed with a thin pointed object while the power of the computer is ON to correct abnormal operation caused by an erroneous machine language program or strong static electricity. Programs and memory contents are retained when this operation is performed, and can be cleared when necessary by pressing the NEW ALL button.

* Do not hold down the RESET button for an excessively long period. Just press and release immediately to complete the reset operation. Holding down this button may cause mistiming by the internal clock.

㉗ New All Button (NEW ALL)
Erases programs and memory contents and sets the computer to the CAL mode. This button is pressed with a thin pointed object while the power of the computer is ON. This operation should only be used when the current programs and memory contents are no longer required. If pressing the NEW ALL button does not succeed in erasing programs and memory contents, press the RESET button and then the NEW ALL button.

㉘ Contrast Key (CONTRAST)
Changes the function of the cursor ◢ / ◥ keys to display contrast control keys (see page 9).

㉙ Screen
A 32-column × 4-line (192 × 32-dot) liquid crystal display upon which 6 × 8-dot characters appear.

2-3 POWER SUPPLY

This unit is powered by three AA size batteries.
Switch power ON, press the RESET button, and then press the NEW ALL button to initialize memory in the following cases:

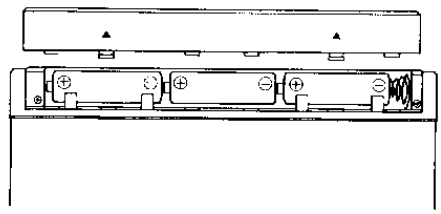
1. After loading batteries for the first time (after purchase).
2. After the computer has been left for longer than 10 minutes without batteries installed (which alters programs and memory contents).

Actual battery life depends upon the way the unit is used (i.e. buzzer operation shortens battery life), but batteries should be replaced as soon as possible after the display begins to become noticeably dim.

* *Batteries should be replaced at least every two years, even if the unit is not used during this period. After two years, batteries tend to leak, which can cause serious damage to the interior of the unit.*

Battery Replacement

1. Switch the power of the unit OFF and open the battery compartment cover (located on the back of the screen panel) by sliding in the direction indicated by the arrows.
2. Remove the three old batteries.
3. Load three new batteries ensuring that their polarity (⊕ / ⊖) is correct.
4. Replace the battery compartment cover.



IMPORTANT

* Memory contents are still retained (approximately one month when SUM-3 type batteries are used) even if batteries become so weak that the display becomes unreadable.
* Frequent use of the buzzer may shorten battery life.
* Always replace all three batteries.

* Never dispose of batteries by incinerating them. Exposing batteries to extremely high temperatures can cause them to EXPLODE.
* Never reverse the polarity (⊕ / ⊖) of batteries when they are loaded in the unit.
* The clock may stop when batteries are replaced, so be sure to check the time when replacement is complete.
* Complete battery replacement as quickly as possible. Programs and memory contents may be altered if batteries are removed form the unit for longer than 10 minutes.
* BATTERY REPLACEMENT PROCEDURES MAY CAUSE LOSS OF MEMORY DATA. Therefore, important programs and data should be saved to a floppy disk or cassette tape before performing battery replacement.

Auto Power OFF

The power of the unit is automatically switched OFF approximately seven minutes after the last key operation (except during program execution), or the last input for an INPUT statement or INPUT$ statement. Power can be resumed by either switching the power switch OFF and then ON again, or by pressing the [BRK] key.

* The clock as well as variable, program and data contents are not affected when the power of the unit is switched OFF.

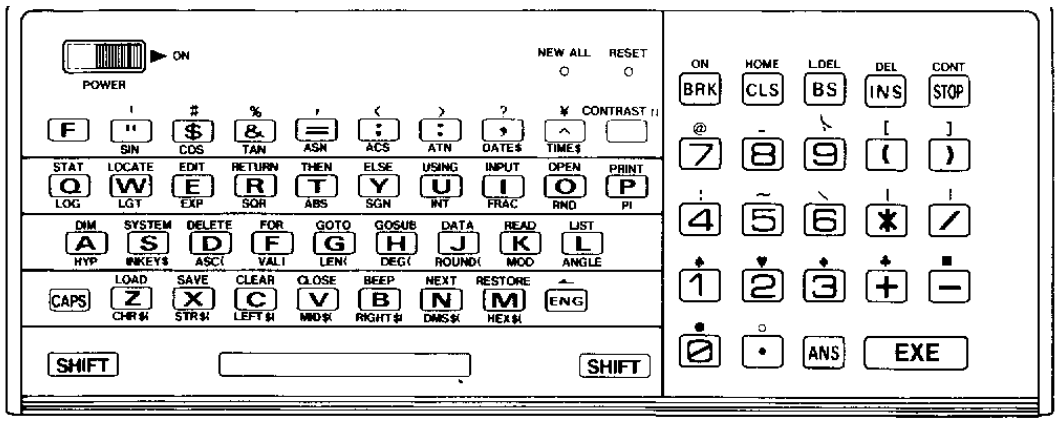
2-4 DISPLAY CONTRAST

The display may appear dark or dim depending upon the strength of the batteries or the viewing angle. The contrast of the display can be adjusted to the desired level by performing the following procedure:

1. Press the [CONTRAST] key.
2. Press the cursor [▲] key to increase contrast, and the cursor [▼] key to decrease contrast.

A weak display when contrast is set to a high level indicates weakened batteries, and batteries should be replaced as soon as possible (see page 8).

2-5 KEYBOARD



Functions are noted on the keytops, as well as above and below the keys on the keyboard. The actual function activated by each key is controlled by the [CAPS], [SHIFT] and [F] keys.

1. Keytop Functions

Normal Mode

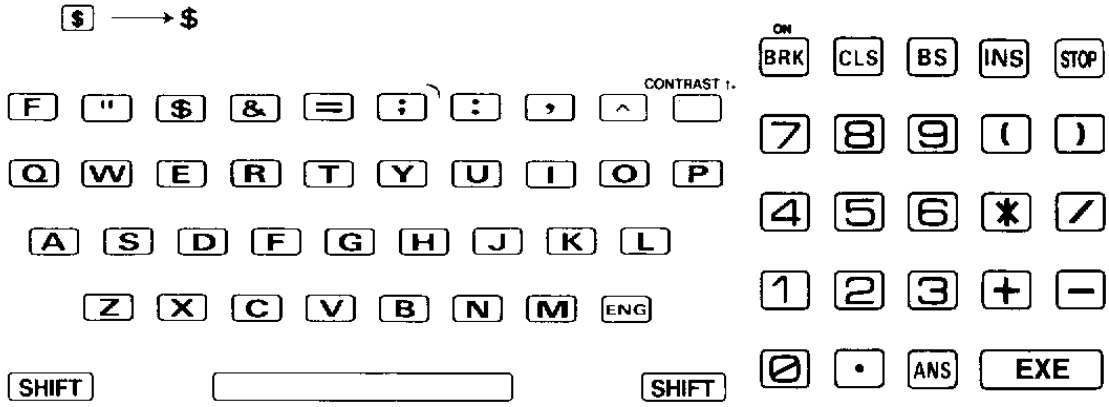
In this mode, each key inputs the characters, symbols, or commands noted on the keys themselves.

EXAMPLE

Operation Display

[A] ⟶ A

[$] ⟶ $



Lower Case Mode

Pressing the [CAPS] key changes the alphabet keys to lower case. Each press of the [CAPS] key switches between upper case and lower case alphabetic characters.

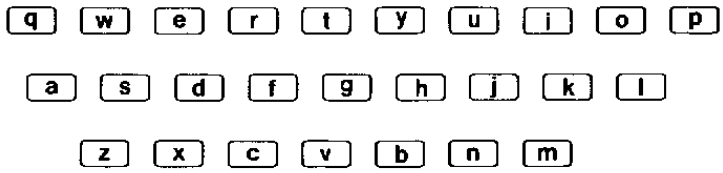
EXAMPLE

Operation Display

[CAPS] [A] ⟶ a

[B] ⟶ b

[D] ⟶ d



2. Functions Noted Above the Keys

The functions, BASIC commands, and symbols noted above the keys can be input when the keys are pressed while the [SHIFT] key is being held down.

EXAMPLE

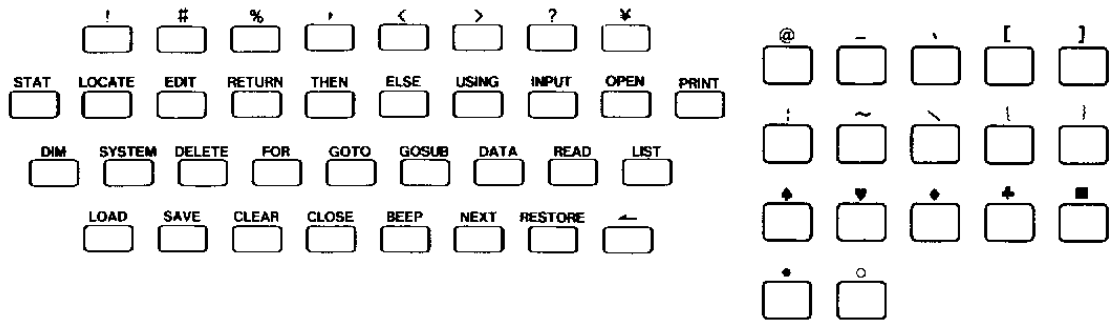
Operation LOAD Display

[SHIFT] [Z] ⟶ LOAD

Held down

[SHIFT] [*] ⟶ {

Held down



* Key operations in combination with the SHIFT key are indicated as SHIFT + key in this manual.
* In the lower case mode, pressing an alphabetic character key while holding down the SHIFT key causes the upper case character for the key to be input instead of the function noted above the key.

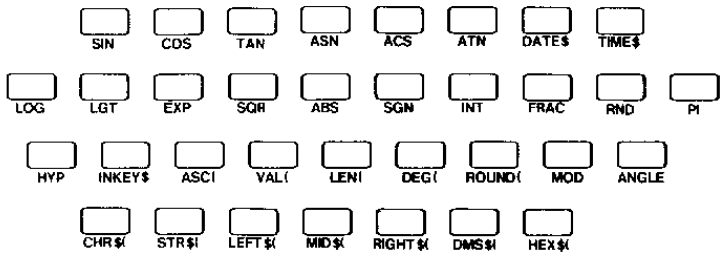
3. Functions Noted Below the Keys

While the F key is held down, pressing a key in either the normal or lower case mode inputs the function noted below the keys.

EXAMPLE

Operation Display

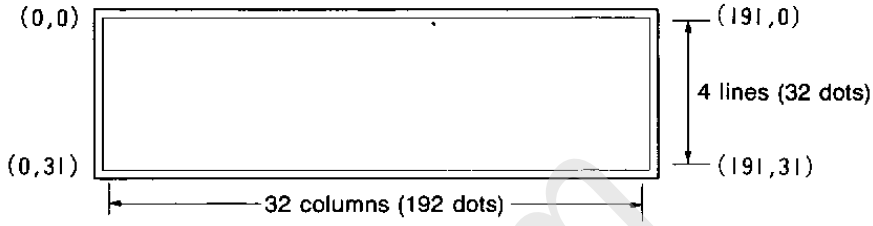
F Z (CHR$( ) → CHR$(



* Key operations in combination with the F key are indicated as F + key in this manual.

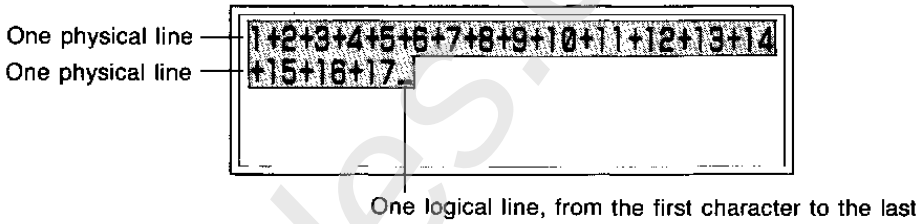
2-6 SCREEN

The screen is a 32-column × 4-line (192 × 32-dot) liquid crystal display. Characters are formed by a 6 × 8 dot matrix.



Physical Lines and Logical Lines

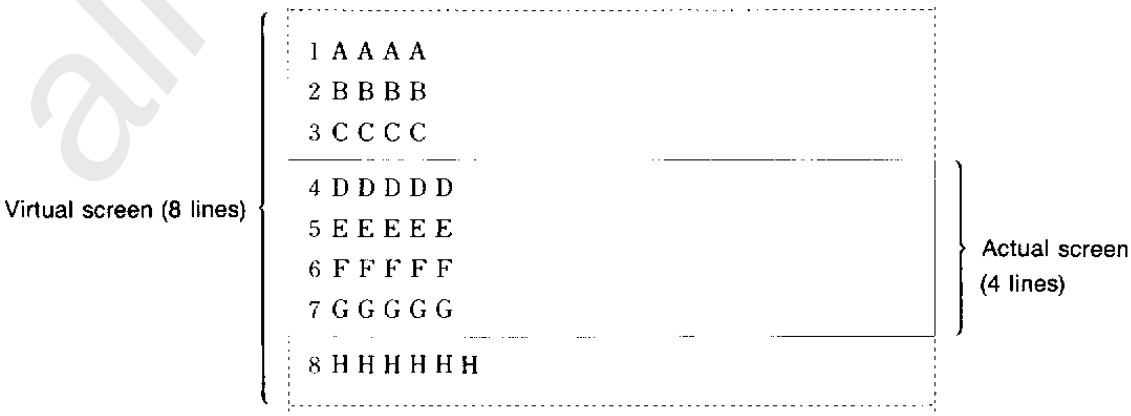
The maximum display capacity of one line is 32 columns, but internally the unit is capable of handling lines up to 255 characters long. The display capacity line (32 character) is referred to as the physical line, while the internal capacity line is called a logical line. A logical line is a continuous line of characters in which any column on the extreme right of the screen is not a null.



Pressing SHIFT ◄ moves the cursor to the beginning of the logical line, while SHIFT ► moves the cursor to the end of the logical line. These operations are useful in determining the extent of logical lines.

Virtual Screen

The screen can display four lines at one time, and as the fifth line is input, the first line scrolls off the top of the screen. Lines that scroll off of the screen can, however, be brought back into view using the cursor (▲ / ▼) keys, because the unit is able to store up to eight lines internally. These eight lines make up the virtual screen, while the four lines actually displayed are called the actual screen.



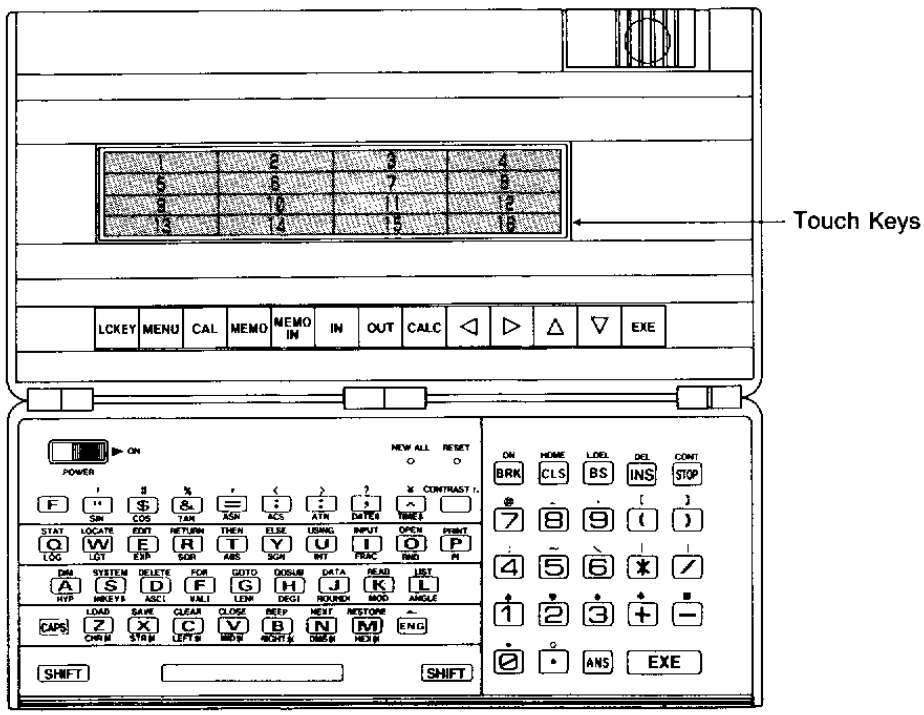
Any four lines of the virtual screen can be displayed on the actual screen at any time.

Screen Editor

Any program lines or data included on the virtual screen can be edited. First the portion of the program or data is brought onto the actual screen, and then the cursor is located at the position to be edited.

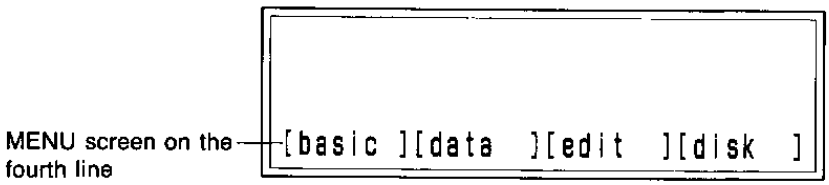
2-7 TOUCH KEYS

The screen displays touch keys in addition to characters and symbols. The fourth line of the MENU screen is used for display of the menu, while a total of 16 touch keys are available during programming.



Menu Screen

The fourth line of the MENU screen is used for the display of touch keys for menu selection. The touch keys on the screen are used for selection of various BASIC program writing and editing functions. Pressing the LCKEY changes the screen to the next menu (see page 59).



CAL Mode

The touch keys on the fourth line of the screen in this mode represent user-generated programs and data files (see page 85).

BASIC Programs

Up to 16 touch keys can be set over the entire screen for use with the BASIC INKEY$ function (see COMMAND REFERENCE, page 52). The use of such touch keys reduces the chance of input errors and greatly facilitates operation.

2-8 DISPLAY CHARACTERS

The relationship between characters and character codes is illustrated in the following table.

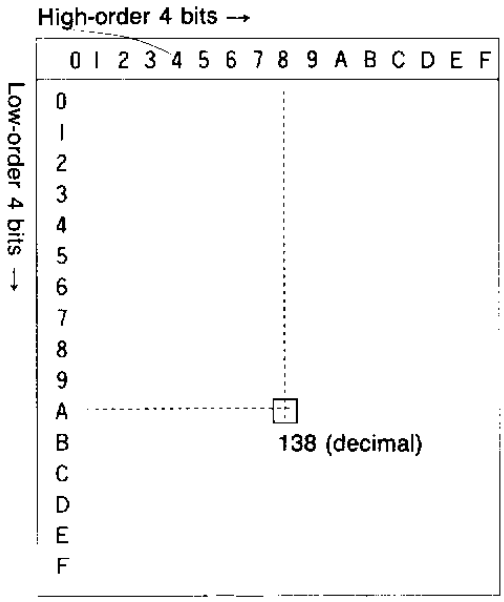
Character Code Table

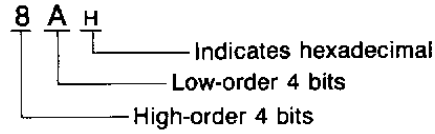
High-order 4 bits →

↓ Low-order 4 bits

Hex.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	(ROLL DOWN) 16	Space 32	0 48	@ 64	P 80	` 96	p 112	― 128	┴ 144	Space 160	ー 176	タ 192	ミ 208	三 224	× 240
1	(ROLL UP) 1	(DEL) 17	! 33	1 49	A 65	Q 81	a 97	q 113	▬ 129	┬ 145	｡ 161	ア 177	チ 193	ム 209	�i 225	円 241
2	(LINE TOP) 2	(INS) 18	" 34	2 50	B 66	R 82	b 98	r 114	▬ 130	┤ 146	「 162	イ 178	ツ 194	メ 210	ㇻ 226	年 242
3	3	19	# 35	3 51	C 67	S 83	c 99	s 115	▬ 131	├ 147	」 163	ウ 179	テ 195	モ 211	ㇼ 227	月 243
4	4	20	$ 36	4 52	D 68	T 84	d 100	t 116	▬ 132	─ 148	、 164	エ 180	ト 196	ヤ 212	◢ 228	日 244
5	(LINE DEL) 5	21	% 37	5 53	E 69	U 85	e 101	u 117	▬ 133	┼ 149	・ 165	オ 181	ナ 197	ユ 213	◣ 229	時 245
6	(LINE END) 6	22	& 38	6 54	F 70	V 86	f 102	v 118	■ 134	│ 150	ヲ 166	カ 182	ニ 198	ヨ 214	◥ 230	分 246
7	7	23	' 39	7 55	G 71	W 87	g 103	w 119	■ 135	▏ 151	ァ 167	キ 183	ヌ 199	ラ 215	◤ 231	秒 247
8	(BS) 8	(LINE C.) 24	(40	8 56	H 72	X 88	h 104	x 120	▌ 136	┌ 152	ィ 168	ク 184	ネ 200	リ 216	♠ 232	千 248
9	9	25	) 41	9 57	I 73	Y 89	i 105	y 121	▌ 137	┐ 153	ゥ 169	ケ 185	ノ 201	ル 217	♥ 233	市 249
A	10	26	* 42	: 58	J 74	Z 90	j 106	z 122	▌ 138	└ 154	ェ 170	コ 186	ハ 202	レ 218	◆ 234	区 250
B	(HOME) 11	27	+ 43	; 59	K 75	[91	k 107	{ 123	▌ 139	┘ 155	ォ 171	サ 187	ヒ 203	ロ 219	♣ 235	町 251
C	(CLS) 12	(→) 28	, 44	< 60	L 76	¥ 92	l 108	¦ 124	■ 140	┌ 156	ャ 172	シ 188	フ 204	ワ 220	● 236	村 252
D	(CR) (LF) 13	(←) 29	- 45	= 61	M 77	] 93	m 109	} 125	■ 141	┐ 157	ュ 173	ス 189	ヘ 205	ン 221	○ 237	人 253
E	14	(↑) 30	. 46	> 62	N 78	^ 94	n 110	~ 126	■ 142	└ 158	ョ 174	セ 190	ホ 206	゛ 222	／ 238	▩ 254
F	15	(↓) 31	／ 47	? 63	O 79	_ 95	o 111	127	＋ 143	┘ 159	ッ 175	ソ 191	マ 207	° 223	＼ 239	255

* Blank segments are not output.
* Notations in parentheses are control codes and are not displayed.
* Characters which cannot be displayed using keyboard input can be displayed using the CHR$ function.
* The values at the lower right of each segment in the table are decimal values.
* Control codes 88H, 89H and 8AH appear to be indentical on the screen, but appear as shown in the character code table when printed.
  (8AH indicates high-order 4 bits = 8 and low-order 4 bits = A (decimal 138), which is the graphic character "■". "H" is added to indicate hexadecimal.)





The CHR$ function is used for display as follows:

PRINT CHR$ (138)
Decimal value

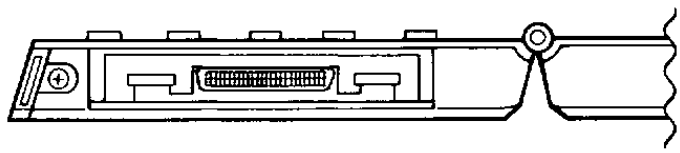
Decimal values are included within the parentheses.

PRINT CHR$ (&H8A)
Hexadecimal value

Hexadecimal are prefixed with "&H".

2-9 CONNECTOR

Optional peripheral devices (i.e. FA-7, MD-100) can be connected to unit via the connector located on the right side of the unit.



Be sure to cover the connector when not in use with the accessory connector cover.

PART 3
CALCULATION FUNCTION

3-1 MANUAL CALCULATIONS AND INPUT CORRECTION

Arithmetic Operators

The following arithmetic operators are used in formulas:

Signs	(+, −)
Addition	(+)
Subtraction	(−)
Multiplication	(*)
Division	(/)
Power	(^)
Integer division	(¥)
Remainder of integer division	(MOD)

The values used with the ¥ and MOD operators are limited to the range of −32768 through 32767, and the fractional part of non-integer values is truncated.

EXAMPLE

```
5¥2.9=2          ( 5÷2.9=2.9 )
7.3 MOD 2.7=1*   ( 7.3÷2.7=3…1 )
```

(The fractional parts crossed out with "×" are truncated before the calculation is performed.)

* Note that a space is required between the value on the left and the MOD operator.

With both ¥ and MOD, the values are converted to their absolute values before division is performed. The sign assigned to the result of the ¥ operation follows the rules of normal division, while the sign assigned to the result of the MOD operation is the sign of the dividend.

EXAMPLE

```
−15¥7=−2
−15 MOD 7=−1    ( −15÷7 = −2 …… −1 )
                          −15¥7  −15MOD7
```

Relational Operators

The following operators can only be used within programs, and they compare two values or strings.

Equal to	=
Not equal to	<>, ><
Less than	<
Greater than	>
Less than or equal to	=<, <=
Greater than or equal to	=>, >=

With character string comparisons, each character in the string to the left of the operator is compared with each character at the corresponding position in the string to the right of the operator. Comparisons are made using the character code for each character. If two strings are of different length and the shorter string is identical to the leading characters of the longer string, the shorter string is judged to be the lesser of the two.

EXAMPLE

```
10 PRINT 125>12          (-1)
20 PRINT "DEF"<"ABCD"    (0)
30 PRINT "ABCD"="ABC"    (0)
```

Logical Operators

The application of logical operators is similar to that of relational operators. The fractional parts of the data are truncated and the specified logical operation is performed bit-by-bit (each bit of the result is obtained by examining the bit in the same position for each argument). There are four different logical operators available with the unit.

NOT	Makes an expression not true.
AND	Expression is true if both parts are true, otherwise expression is false.
OR	Expression is true if either part is true, otherwise expression is false.
XOR	Expression is false if either part is true or either part is false, expression is true if one part is true and one part is false.

Negation

X	NOT X
0	1
1	0

Logical Product

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

Logical Sum

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

Exclusive OR

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

Character Operator

The only string operator available is the plus (+) operator. The length of the result is limited to 255 characters.

EXAMPLE

"A" + "B" ⟶ "AB"

Priority Sequence

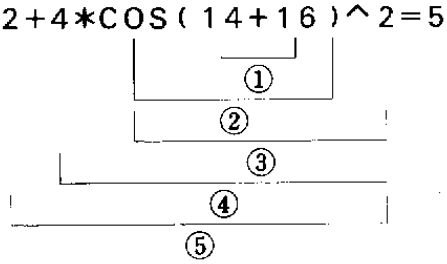
Arithmetic, relational and logical operations are performed in the following priority sequence:

1. (,)
2. Functions
3. Power
4. Signs (+, −)
5. ＊, /, ¥, MOD
6. +, −
7. Relational operators
8. NOT
9. AND
10. OR, XOR

NOTE:

a. Calculations are performed from left to right when the priority sequence is identical.
b. Complex functions (sin cos 60) are performed from right to left.
c. Consecutive powers (5^4^3) is performed from left to right.

EXAMPLE



Number of Digits

- Internal calculations are performed with a 13-digit mantissa + 2-digit exponent. PI, however, is expressed in 11 digits (3.1415926536).
- Calculation results are displayed rounded off to a 10-digit mantissa + 2-digit exponent.
- The maximum input capacity for a single line is 255 characters.

Internal Rounding

Internal calculations are performed with a 13-digit mantissa, but the 11th, 12th, and 13th digits are truncated if they equal 049 or less, and are rounded up if they equal 950 or greater.

EXAMPLE

123456.7890049 ⟶ 123456.7890000
123456.7890236 ⟶ 123456.7890236
123456.7890950 ⟶ 123456.7891000

Calculation Result Display

Calculation results are displayed in the following manner.

1. Integer less than 1 × 10^{10} ——————→ Integer
2. 10 digits or less in fractional part ——————→ Decimal
3. Other ——————→ Exponential

Variables

The following rules apply to variable names for all types of variables used with the unit.

Variable names:

1. Are character strings with an upper case alphabetic character (A ~ Z, internal decimal code 65 ~ 90) or lower case alphabetic character (a ~ z, internal decimal code 97 ~ 122) in the leading (first) position. (See the character code table on page 127 for internal codes.)
2. Are composed of upper or lower case alphabetic characters or numbers (0 ~ 9, internal decimal code 48 ~ 57) following the leading alphabetic character.
3. Cannot use reserved words (see page 133) as the leading characters.
4. Can be up to 255 characters long.

Manual Calculations

EXAMPLE 1

(Formula) 9+7.8÷6−3.5×2=3.3

(Operation) 9 [+] 7 . 8 [/] 6 [−] 3 . 5 [*] 2 [EXE]

```
9+7.8/6-3.5*2
 3.3
```

As can be seen, the [*] key is used for " × ", the [/] key is used for ÷, and the [EXE] key is used for " = ".

EXAMPLE 2

(Formula) 56×(−12)÷(−2.5)=268.8

(Operation) 56 [*] [−] 12 [/] [−] 2 . 5 [EXE]

```
56*-12/-2.5
 268.8
```

The [−] key is used to enter the minus sign in front of the appropriate value.

EXAMPLE 3

(Formula) $(4.5\times10^{75})\times(-2.3\times10^{-78})=-0.01035$

(Operation) 4 . 5 [E] 75 [*] [−] 2 . 3 [E] [−] 78 [EXE]

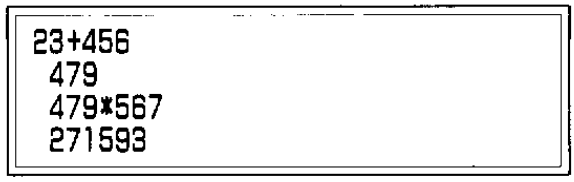
```
4.5E75*-2.3E-78
-0.01035
```

Once the mantissa is entered, press the [E] key prior to entering the exponent.

EXAMPLE 4

(Formula) (2 3 + 4 5 6) × 5 6 7 = 2 7 1 5 9 3

(Operation) 2 3 [+] 4 5 6 [EXE] [*] 5 6 7 [EXE]



The [ANS] key can be used at any time during a calculation to enter the recently obtained calculation result.

EXAMPLE 5

(Formula) 8 1 . 2 ÷ (5 . 6 + 8 . 9) = 5 . 6 — Parentheses calculation takes priority.

(Operation) 5 . 6 [+] 8 . 9 [EXE] 8 1 . 2 [/] [ANS] [EXE]

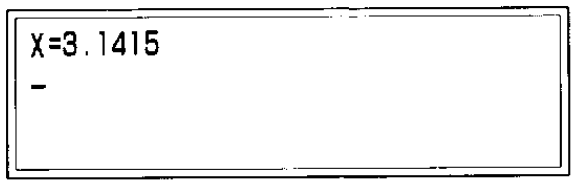
Calculations Using Variables

Algebraic calculations can also be performed using variables. The following list of calculations, for example, becomes much easier to perform if a variable is assigned for the common term.

2 × 3.1415 + 5 =
3 × 3.1415 + 6 =
4 × 3.1415 + 7 =
5 × 3.1415 + 8 =

1. First, assign the value 3.1415 to the variable X.

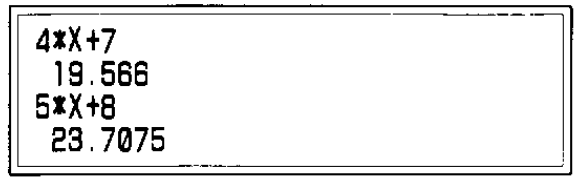
X [=] 3 . 1 4 1 5 [EXE]



2. Then use the variable in place of the value for each of the calculations.

2 [*] X [+] 5 [EXE]
3 [*] X [+] 6 [EXE]

4 [✱] X [+] 7 [EXE]
5 [✱] X [+] 8 [EXE]



As can be seen here, the use of variable is very helpful when performing repeat calculations.

Corrections Using Character Insertion

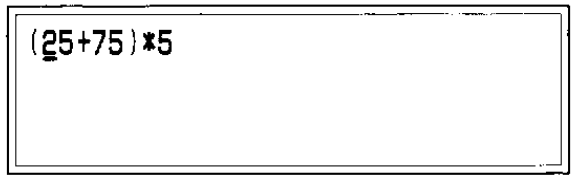
Characters can be inserted by positioning the cursor immediately to the right of the location at which the insertion is to be made, and then creating spaces using the [INS] key. Then characters are entered at the desired location.

EXAMPLE

Correct (25 + 75) ✱ 5 to (125 + 75) ✱ 5

(Operation)

Use the [▶] key to position the cursor under the 2 of 25.



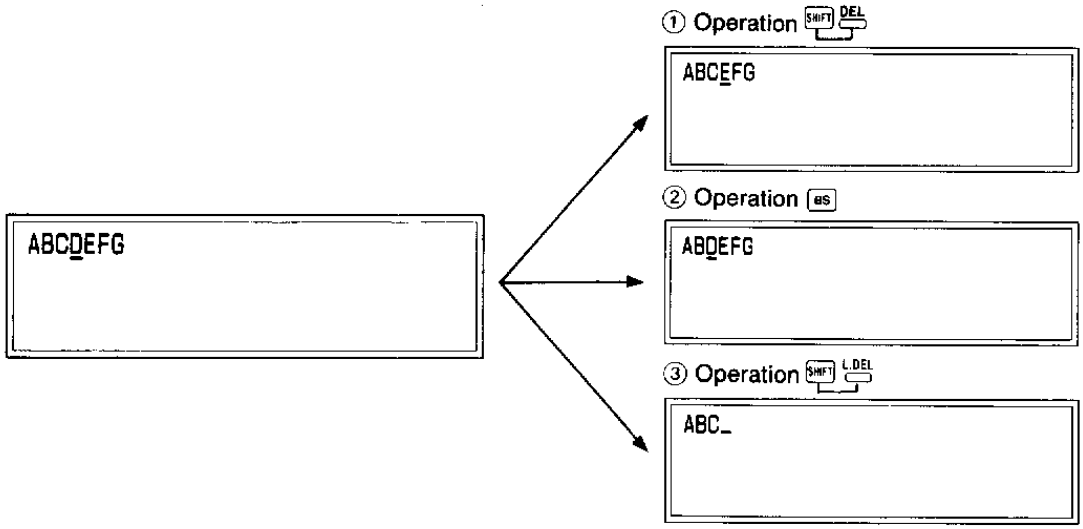
Press [INS] [1].

Corrections Using Character Deletion

One of three different methods can be used to delete characters from a line.

1. Move the cursor to the character to be deleted and then press the [SHIFT] [DEL] keys to delete the character.
2. Move the cursor to the character immediately to the right of the character to be deleted and then press the [BS] (backspace) key to delete the character.
3. Press the [SHIFT] [L.DEL] keys to delete all characters from the current cursor position all the way to the end of the current logical line.

EXAMPLE



3-2 SCIENTIFIC CALCULATIONS

The scientific functions (see the scientific function table on page 29) can be used either within programs or for manual calculations. For the sake of explanation, all of the examples here will cover only manual calculations.

Trigonometric and Inverse Trigonometric Functions

sin:	sine	$\sin^{-1}$:	arc sine
cos:	cosine	$\cos^{-1}$:	arc cosine
tan:	tangent	$\tan^{-1}$:	arc tangent

These functions return a trigonometric function value for a given angle, or an angle value of a given trigonometric function value. The ANGLE command should be used to specify the unit for the angle value when these functions are used. Angle unit specification is only required once for all subsequent trigonometric/inverse trigonometric functions.

ANGLE 0 ..... DEG (degrees)
ANGLE 1 ..... RAD (radians)
ANGLE 2 ..... GRAD (grads)

The relationship among these three specifications is:

90 degrees $= \frac{\pi}{2}$ radians = 100 grads

The current angle unit can be confirmed by performing the operation: [SHIFT] [SYSTEM] [EXE] . The current angle unit is retained when the power of the unit is switched OFF, and the angle unit becomes ANGLE 0 when the NEW ALL button is pressed.

The value for π can be directly entered into a formula using "PI" (3.1415926536).

EXAMPLE 1

(Formula) $\sin 30^\circ = 0.5$

(Operation) [F] [ANGLE] 0 [EXE]
[F] [SIN] 30 [EXE]

```
ANGLE0
SIN30
 0.5
```

EXAMPLE 2

(Formula) $\cos \frac{\pi}{3} = 0.5$

(Operation) [F] [ANGLE] 1 [EXE]
[F] [COS] ([F] [PI] / 3) [EXE]

```
ANGLE1
COS(PI/3)
 0.5
```

EXAMPLE 3

(Formula) $2\sin \frac{\pi}{3} + \cos \frac{\pi}{3} = 2.232050808$

(Operation) [F] [ANGLE] 1 [EXE]
2 [*] [F] [SIN] ([F] [PI] / 3)
[+] [F] [COS] ([F] [PI] / 3) [EXE]

```
ANGLE1
2*SIN(PI/3)+COS(PI/3)
 2.232050808
```

EXAMPLE 4

(Formula) $\tan 60^\circ = 1.732050808$

(Operation) [F] [ANGLE] 0 [EXE]
[F] [TAN] 60 [EXE]

```
ANGLE0
TAN60
 1.732050808
```

EXAMPLE 5

(Formula) $\sin^{-1} 0.5 = 30^\circ$

(Operation) [F] [ASN] 0.5 [EXE] (ANGLE 0)

```
ASN0.5
 30
```

EXAMPLE 6

(Formula) $\cos^{-1} \frac{2^{0.5}}{2} = 45^\circ$

(Operation) [F] [ACS] [(] 2 [^] 0.5 [/] 2 [)] [EXE]

```
ACS(2^0.5/2)
 45
```

EXAMPLE 7

(Formula) $\tan^{-1} \sqrt{3} = 60^\circ = 1.047197551 \left(\frac{\pi}{3}\right)$

(Operation) [F] [ATN] [F] [SQR] 3 [EXE] (ANGLE0)
[F] [ANGLE] 1 [EXE]
[F] [ATN] [F] [SQR] 3 [EXE]

```
ATNSQR3
 60
ANGLE1
ATNSQR3
```

⋮

```
 60
ANGLE1
ATNSQR3
 1.047197551
```

Hyperbolic and Inverse Hyperbolic Functions

sinh:	hyperbolic sine	$\sinh^{-1}$:	hyperbolic arc sine
cosh:	hyperbolic cosine	$\cosh^{-1}$:	hyperbolic arc cosine
tanh:	hyperbolic tangent	$\tanh^{-1}$:	hyperbolic arc tangent

EXAMPLE 1

(Formula) $\sinh 5 = 74.20321058$

(Operation) [F] [HYP] [F] [SIN] 5 [EXE]

The HYPSIN function is used for sinh.

```
HYPSIN5
 74.20321058
```

EXAMPLE 2

(Formula) $\cosh^{-1} 1.5 = 0.962423650 1$

(Operation) [F] [HYP] [F] [ACS] 1 . 5 [EXE]

The HYPACS is used for $\cosh^{-1}$.

```
HYPACS1.5
 0.9624236501
```

Logarithmic Functions, Exponential Functions

$\log_{10}$: common logarithm e^x: exponent
$\log_e$: natural logarithm

EXAMPLE 1

(Formula) $\log_e 123 = 4.812184355$

(Operation) [F] [LOG] 1 2 3 [EXE]

The LOG function is used for $\log_e$.

```
LOG123
 4.812184355
```

EXAMPLE 2

(Formula) $\log_{10} 100 = 2$

(Operation) [F] [LGT] 1 0 0 [EXE]

The LGT function is used for $\log_{10}$.

```
LGT100
 2
```

EXAMPLE 3

(Formula) $e^5 = 148.4131591$

(Operation) [F] [EXP] 5 [EXE]

The EXP function is used for e^x.

```
EXP5
 148.4131591
```

Other Functions

SGN: Sign
ABS: Absolute value
FIX: Integer part
RND: Random number
INT: Integer value
FRAC: Fraction

- **SGN**

For SGN (x), returns a 1 when $x>0$, a -1 when $x<0$, and a 0 when $x=0$)

(Operation) [F] [SGN] 6 [EXE]
[F] [SGN] − 2 [EXE]

```
SGN6
 1
SGN-2
-1
```

• RND

Generates random numbers between 0 and 1.

(Operation) [F] [RND] 5 [EXE]

```
RND5
 0.1394789353
```

* The above is only a sample value.

• ABS

Returns the absolute value of x for ABS (x).

(Formula) | 7 8 . 9 ÷ − 5 . 6 | = 1 4 . 0 8 9 2 8 5 7 1

(Operation) [F] [ABS] (7 8 . 9
[/] [−] 5 . 6) [EXE]

```
ABS(78.9/-5.6)
 14.08928571
```

• INT

For INT (x), returns the largest integer which does not exceed the value of x.

(Operation) [F] [INT] [−] 6 4 . 5 [EXE]

```
INT-64.5
-65
```

• FIX

Returns the integer part of x for FIX (x).

(Operation) [F] [I] [X] (8 0 0 0 [/] 9 6) [EXE]

```
FIX(8000/96)
 83
```

• FRAC

Returns the fractional part of x for FRAC (x).

(Operation) [F] [FRAC] (8 0 0 0 [/] 9 6) [EXE]

```
FRAC(8000/96)
 0.3333333333
```

Decimal⟷Sexagesimal Conversions

DEG: Sexagesimal⟶Decimal
DMS$: Decimal⟶Sexagesimal

EXAMPLE 1

(Formula) $12^\circ\,34'\,56'' = 12.58222222^\circ$

(Operation) [F] [DEG( ] 12 [,] 34 [,] 56 [)] [EXE]

```
DEG(12.34.56)
 12.58222222
```

EXAMPLE 2

(Formula) $12.3456^\circ = 12^\circ\,20'\,44.16''$

(Operation) [F] [DMS$( ] 12.3456 [)] [EXE]

```
DMS$(12.3456)
 12°20'44.16
```

Decimal⟷Hexadecimal Conversions

&H: Hexadecimal⟶Decimal
HEX$: Decimal⟶Hexadecimal

EXAMPLE 1

(Formula) $10_{(16)} = 16_{(10)}$

(Operation) [&] [H] 10 [EXE]

```
&H10
 16
```

EXAMPLE 2

(Formula) $1000_{(10)} = 3E8_{(16)}$

(Operation) [F] [HEX$( ] 1000 [)] [EXE]

```
HEX$(1000)
03E8
```

* Hexadecimal A, B, C, D, E, F corresponds to decimal 10, 11, 12, 13, 14, 15.

Scientific Function Table

Function Name	Formula	Format	Details
Trigonometric	sin	SIN (numeric expression)	$-1440° <$ numeric expression $< 1440°$ (8πrad, 1600grad)
	cos	COS (numeric expression)	$-1440° <$ numeric expression $< 1440°$ (8πrad, 1600grad)
	tan	TAN (numeric expression)	$-1440° <$ numeric expression $< 1440°$ (8πrad, 1600grad) * Except when numeric expression is add multiple of 90° (π/2rad, 100grad)
Inverse Trigonometric	$\sin^{-1}$	ASN (numeric expression)	|numeric expression| $\leqq 1$, $-90° \leqq$ ASN (numeric expression) $\leqq 90°$
	$\cos^{-1}$	ACS (numeric expression)	|numeric expression| $\leqq 1$, $0° \leqq$ ACS (numeric expression) $\leqq 180°$
	$\tan^{-1}$	ATN (numeric expression)	|numeric expression| $< 10^{100}$, $-90° \leqq$ ATN (numeric expression) $\leqq 90°$
Hyperbolic	sinh	HYPSIN (numeric expression)	|numeric expression| $\leqq 230.2585092$
	cosh	HYPCOS (numeric expression)	|numeric expression| $\leqq 230.2585092$
	tanh	HYPTAN (numeric expression)	|numeric expression| $\leqq 10^{100}$, $-1 \leqq$ HYPTAN (numeric expression) $\leqq 1$
Inverse Hyperbolic	$\sinh^{-1}$	HYPASN (numeric expression)	|numeric expression| $< 5 \times 10^{99}$
	$\cosh^{-1}$	HYPACS (numeric expression)	$1 \leqq$ numeric expression $< 5 \times 10^{99}$
	$\tanh^{-1}$	HYPATN (numeric expression)	|numeric expression| < 1
Exponential	e^x	EXP (numeric expression)	$-227 \leqq$ numeric expression $\leqq 230.2585092$
Natural logarithm	$\log_e x$	LOG (numeric expression)	numeric expression > 0
Common logarithm	$\log_{10} x$	LGT (numeric expression)	numeric expression > 0
Square root	$\sqrt{x}$	SQR (numeric expression)	numeric expression $\geqq 0$
Absolute value	$\|x\|$	ABS (numeric expression)	Returns absolute value of numeric expression
Sign		SGN (numeric expression)	numeric expression < 0 : -1 numeric expression $= 0$: 0 numeric expression > 0 : 1
Integer		INT (numeric expression)	Gauss function: Returns maximum integer value that does not exceed numeric expression value.
Fraction		FRAC (numeric expression)	Returns fractional part of numeric expression.

Function Name	Formula	Format	Details
Rounding		ROUND (*x*, *y*) *x*, *y* : numeric expression	Rounds *x* at position specified by *y*.
Fix		FIX (numeric expression)	Returns integer part of *x*.
Degree	Sexagesimal → Decimal	DEG (d [,m[,s]]) d, m, s : numeric expression	Converts sexagesimal to decimal.
PI	π	PI	3.1415926536
Random number		RND (numeric expression)	Returns a random number with 10 decimal places. 0 < RND < 1

* Except for ROUND and DEG, any values used with these functions need not be included in parentheses.

3-3 STATISTICAL CALCULATIONS

The statistical functions allow the analysis of collected data as well as estimates of new data.

Statistical Data Input

A special statistical data memory area is cleared and data is input before statistical calculations. The command STAT CLEAR is used to clear the statistical data memory area.

Single-variable data { Individual data.....STAT data
Repeated identical data.....STAT data ; frequency

Paired-variable data { Individual data.....STAT *x*-data, *y*-data
Repeated identical data.....STAT *x*-data, *y*-data ; frequency

1. The default value for the x-data is the most recent previous x-data.
2. The default value for the y-data is the most recent previous y-data.
3. Either x-data or y-data or both x and y-data must be specified.

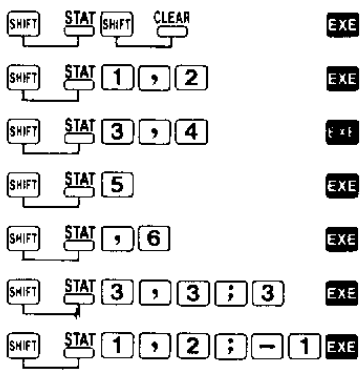
The following shows an example of statistical data input:

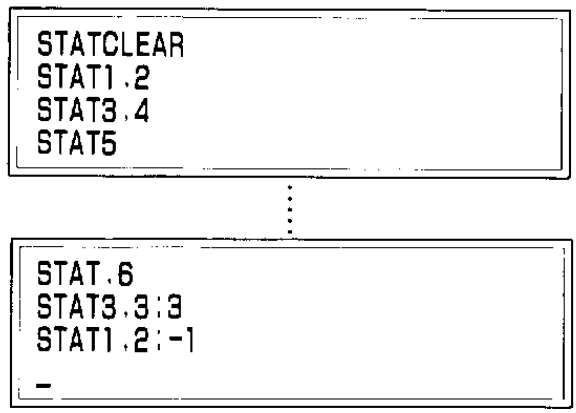
Input data

x	1	3	5	5	3	3	3
y	2	4	4	6	3	3	3

In this example, all of the data are input, and then the first data item ($x = 1, y = 2$) is deleted.

(Operation)





NOTE: Data is deleted by entering the data to be deleted, followed by a negative frequency value (*x*-data, *y*-data; −1).

Statistical Values

The statistical values listed below can be output using the statistical functions or system variables.

	Value	Formula
CNT	Number of data processed	n
SUMX	Cumulative total of X-data	Σx
SUMY	Cumulative total of Y-data	Σy
SUMX2	Sum of squares of X-data	Σx^2
SUMY2	Sum of squares of Y-data	Σy^2
SUMXY	Sum of products of X-data and Y-data	Σxy
MEANX	Mean of X-data	$\frac{\Sigma x}{n}$
MEANY	Mean of Y-data	$\frac{\Sigma y}{n}$
SDX	Sample standard deviation of X-data	$\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n(n-1)}}$
SDY	Sample standard deviation of Y-data	$\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n(n-1)}}$
SDXN	Population standard deviation of X-data	$\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n^2}}$
SDYN	Population standard deviation of Y-data	$\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n^2}}$
LRA	Linear regression constant term	$\frac{\Sigma y-\mathrm{b}\cdot\Sigma x}{n}$
LRB	Linear regression coefficient	$\frac{n\Sigma xy-\Sigma x\cdot\Sigma y}{n\Sigma x^2-(\Sigma x)^2}$
COR	Correlation coefficient	$\frac{n\Sigma xy-\Sigma x\cdot\Sigma y}{\sqrt{\{n\Sigma x^2-(\Sigma x)^2\}\{n\Sigma y^2-(\Sigma y)^2\}}}$
EOX(y)	Estimated value of X in relation to Y	$\mathrm{EOX}(y)=\frac{y-a^*}{\mathrm{b}}$
EOY(x)	Estimated value of Y in relation to X	$\mathrm{EOY}(x)=\mathrm{a}+x\cdot\mathrm{b}^*$

* a and b represent LRA and LRB respectively.

See the COMMAND REFERENCE for details on all of these functions.
The following program uses these functions as well as the touch key features to demonstrate some useful applications of statistical calculations.

Program list

```
100 CLS
110 DEFCHR$(240)="635549416300":DEFCHR$(241)="0E11312E2000"
120 LOCATE 0,0:PRINT "CLEAR STATISTICAL DATA?"
130 LOCATE 1,1:PRINT "[ YES ]"
140 LOCATE 1,3:PRINT "[ N O ]";
150 IN=ASC(INKEY$)
160 IF IN=244 THEN STAT CLEAR:C=1 ELSE IF IN<>252 THEN 150
170 CLS:LOCATE 1,3:PRINT "[STAT] [SLIST1][SLIST2][ END ]";
180 IN=ASC(INKEY$)
190 IF IN<252 THEN 180
200 ON IN-251 GOTO ,300,400,460
210 LOCATE 8,3:PRINT " PRESS [E] TO END INPUT";
220 LOCATE 0,0:PRINT USING"DATA No.&    &";STR$(C)
230 LOCATE 9,1:PRINT TAB(20):LOCATE 0,1:INPUT "x DATA =";X$
240 IF X$="E" THEN 170
250 IF X$<>"" THEN X=VAL(X$)
260 LOCATE 9,2:PRINT TAB(20):LOCATE 0,2:INPUT "y DATA =";Y$
270 IF Y$<>"" THEN Y=VAL(Y$)
280 STAT X,Y
290 C=C+1:GOTO 220
300 LOCATE 2,3:PRINT "    ";:LOCATE 17,3:PRINT "      ][    ";
310 LOCATE 0,0:PRINT " n =";CNT:LOCATE 16,0:PRINT CHR$(240);
    "xy=";SUMXY;
320 LOCATE 0,1:PRINT CHR$(240);"x =";SUMX;:LOCATE 16,1:PRINT
    CHR$(240);"y =";SUMY;
330 LOCATE 0,2:PRINT CHR$(240);"x2=";SUMX2;:LOCATE 16,2:PRINT
    CHR$(240);"y2=";SUMY2;
340 LOCATE 9,3:PRINT " NEXT ";
350 IF ASC(INKEY$)<>253 THEN 350
360 LOCATE 0,0:PRINT TAB(96);
370 LOCATE 0,0:PRINT "x MEAN=";MEANX;
380 LOCATE 0,1:PRINT "y MEAN=";MEANY;
390 IF ASC(INKEY$)<>253 THEN 390 ELSE FOR I=1 TO 15:NEXT:GOTO
    170
400 LOCATE 0,0:PRINT TAB(96);
410 LOCATE 2,3:PRINT "    ] [    ][ NEXT ][    ";
420 LOCATE 0,0:PRINT "x";CHR$(241);"=";SDXN;:LOCATE 16,0:PRINT
    "y";CHR$(241);"=";SDYN;
430 LOCATE 0,1:PRINT " a=";LRA;:LOCATE 16,1:PRINT " b=";LRB;
440 LOCATE 0,2:PRINT " r=";COR;
450 IF ASC(INKEY$)<>254 THEN 450 ELSE FOR I=1 TO 15:NEXT:GOTO
    170
460 CLS :END
```

EXAMPLE

The following table shows the shipments of article x and article y from a certain company. Enter the x and y-data, and produce the sum of products, cumulative total, population standard deviation, and correlation coefficient.

Article \ Date	4	5	6	7	8
x	2	2	5	8	8
y	1	5	5	5	9

Once program execution begins, the first display to appear confirms whether or not statistical data is to be cleared from memory. Pressing the YES touch key clears the area, while the NO touch key proceeds with the program using data already stored in memory. In this example, new data will be entered, so press YES.

```
CLEAR STATISTICAL DATA?
 [ YES ]

 [ N O ]
```

The display will change to appear as illustrated below. Pressing [STAT] enters the data input mode.

[YES]

```
[STAT] [SLIST1][SLIST2][ END ]
```

[STAT]

```
DATA No. 1
x DATA =?_

 [STAT]  PRESS [E] TO END INPUT
```

Here, input the x and y-data given in the table above.

2 EXE 1 EXE
EXE 5 EXE
5 EXE EXE
8 EXE EXE
EXE 9 EXE
E EXE

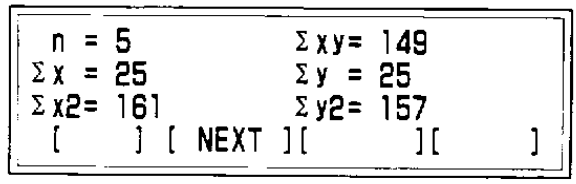
```
DATA No. 1
x DATA =?2
y DATA =?1
 [STAT]  PRESS [E] TO END INPUT
```

⋮

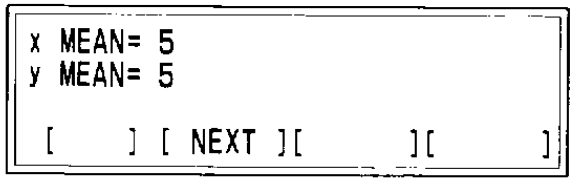
```
[STAT] [SLIST1][SLIST2][ END ]
```

When consecutive x or y-data are identical, simply press EXE key without entering a data value.

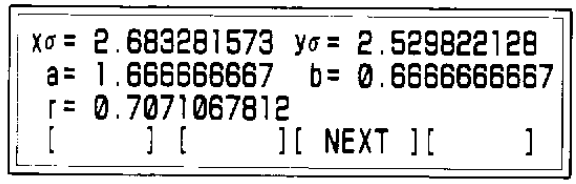
Press [SLIST1] once all data input is complete. The number of data (n), the sum of products for x and y (Σxy), the cumulative total of x (Σx), the cumulative total of y (Σy), the sum of squares of x (Σx^2), and the sum of squares of y (Σy^2) will appear on the screen.



Now press [NEXT] and the screen changes to the means of x and y.



Press [NEXT] again, followed by [SLIST2]. This screen shows the population standard deviation for x (σx), the population standard deviation for y (σy), the linear regression constant term (a), the linear regression coefficient (b), and the correlation coefficient (r).



Additional data can be appended to that already present in memory by pressing the [NEXT] key followed by [STAT]. Pressing [END] ends program execution.

PART 4
FORMULA STORAGE FUNCTION

The formula storage function is very useful when performing repeat calculations, when writing a program, or for successive display of identical strings. Three different keys are used when working with the formula storage function.

- IN key ..... Stores presently displayed formula.
- OUT key ..... Displays formula stored in memory.
- CALC key ..... Assigns values to variables in formula, and displays formula calculation result.

Sample Application

EXAMPLE 1

Determine Y for each X where Y = 200sinX.

X	8°	12°	15°	20°	25°	27°	30°	35°
Y								

Unit angle: [F] [ANGLE] 0 [EXE]

Formula storage: [Y] [=] 2 0 0 [×] [F] [SIN] [X] [IN]

```
ANGLE0
Y=200*SINX
_
```

Note that the final step in the formula storage is pressing the IN key, and NOT the EXE key.
Now clear the screen using the CLS key, and then press OUT to display the newly input formula.
Confirm that the formula has been entered correctly.
Now press the CALC key to begin calculation.

CALC
8 EXE (for 8°)

```
X?8
Y= 27.83462019
```

CALC
12 EXE (for 12°)

```
X?12
Y= 41.58233816
```

[CALC]
15 [EXE] (for 15°)

```
X?15
Y= 51.76380902
```

This procedure is repeated for all of the x-values.
In the example above, the [CALC] key had to be pressed before entry of each value. Adding a semicolon to the end of the formula makes input easier by making it possible to advance to the next execution by pressing the [EXE] key in place of [CALC].

EXAMPLE 2
Complete the following table:

X	Y	P = X . Y	Q = X/Y
4.27	1.17		
8.17	6.48		
6.07	9.47		
2.71	4.36		
1.98	3.62		

1. Formula storage

[P][=][X][*][Y][:][Q][=][X][/][Y][;][IN]

```
P=X*Y:Q=X/Y;
-
```

Note here that multiple formulas can be stored by connecting them with a colon.

2. Calculation

[CALC]
4.27 [EXE]
1.17 [EXE]
[EXE]

```
X?4.27
Y?1.17
P= 4.9959
Q= 3.64957265
```

[EXE]
⋮

```
X?_
```

In this example, execution progresses to the next x and y-value input when the EXE key is pressed.
Variable names up to 255 characters long (see page 20 for restrictions placed on variables) can be used with the formula storage function. This makes it possible to produce prompts on the screen describing the type of entry required.

EXAMPLE 3

(Formula) Total = price x quantity

S U M = P R I C E * Q U A N T I T Y IN

```
SUM=PRICE*QUANTITY
_
```

```
PRICE?100
QUANTITY?4500
SUM= 450000
```

```
PRICE?_
```

```
 100 EXE
4500 EXE
```

CALC

Formula Storage Specification and Clear

- **Specification**

Each press of the IN key stores the current contents of the display (up to 256 characters). Storing a new string deletes the string previously stored in memory.

- **Clear**

Pressing CLS IN stores a blank screen in the formula storage area, so in effect, the area is cleared.

IMPORTANT

1. Up to 256 characters can be stored using the IN key, but calculations using the CALC key are limited to 255 characters.
2. Memory contents are retained even when power is switched OFF, either manually or by the auto power OFF function.
3. An error is generated when an entry stored by the IN key is not a numeric expression.
4. Strings and arrays are simply displayed as stored when recalled.

5. The same limitations that apply to BASIC variables apply to formula storage function variables (see page 20).
6. Results obtained can be used directly in subsequent calculations in combination with the [+], [−], [*], [/], and [^] keys.
7. Calculations are terminated under the the following conditions:
   - Pressing the [BRK] key
   - When an error is generated.
8. Character strings stored using the [IN] key are erased when an ASCII program file is loaded or merged using BASIC.

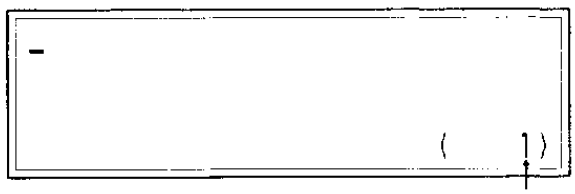
PART 5
DATA BANK FUNCTION

5-1 DATA BANK OPERATION

The data bank function makes it possible to easily store and recall large volumes of memo data without special programming. Memo data are stored in a MEMO file under (MEMO .) filenames.
Data is input into the data bank in the MEMO IN mode.

* Up to 255 characters (8 display lines) can be input into a single memo data record.
* See pages 52 and 53 for details on the MEMO mode and MEMO IN mode.

Press the MEMO IN key.



Record number (logical data line number)

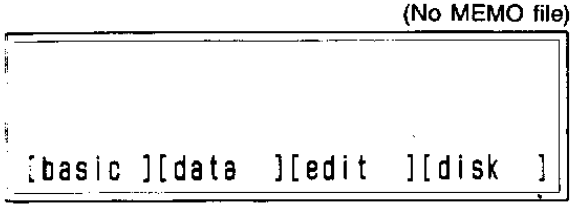
Pressing MEMO IN enters the MEMO IN mode. The following operations are then performed, depending upon whether or not a MEMO file already exists in memory.

1. When a MEMO file does not exist, create a MEMO file and the unit stands by for input of the first record of the memo file.
2. When a MEMO file already exists, access the MEMO file and the unit waits for input of the next record of the memo file.

The MENU mode can be used to determine whether or not a MEMO file currently exists in memory.

Press MENU key.

(No MEMO file)



Press MENU key.

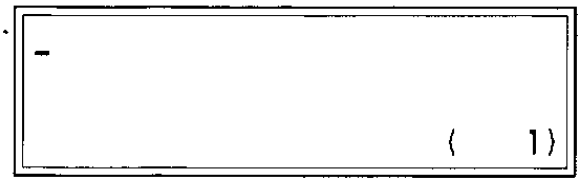
(MEMO file exists)

The MEMO key is used to search the memo data.

* The MEMO key is used to access a MEMO file (data file named MEMO), and is inoperative when no MEMO file exists.

5-2 MEMO DATA INPUT

Press the MEMO key to enter memo input stand by.

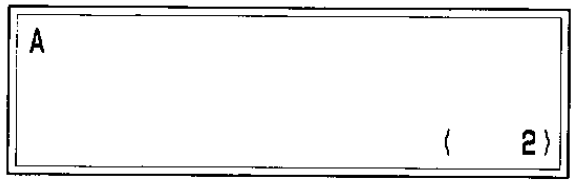


Memo data is input by entering the data and then pressing the EXE key.

[B][A][S][I][C][EXE]

The cursor disappears from the screen, and the input memo data is displayed on the screen. The record number is incremented to indicate that the next record is available for memo data input.

Press [A] [EXE].



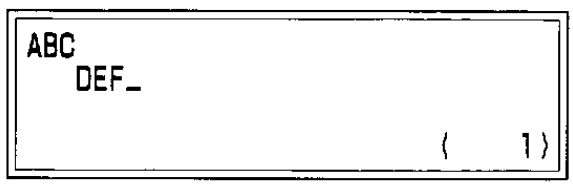
The procedures outlined above can be repeated as many times as required to input various data items in the MEMO file.

Logical lines

The screen editor is used during writing of MEMO data and records are counted in units of logical lines on the 8-line virtual screen. However, the method used for discrimination of logical lines by the screen editor is different from the method used in the CAL mode or in the BASIC programming mode (see page 12 for details on logical lines).
In the MEMO IN mode (as well as the DATA editing and BASIC editing modes), all the data on the virtual screen is taken as part of the same logical line and input into the same record only when the EXE key is pressed. In the CAL/BASIC programming mode, a null in the far right (32nd) column of the screen results in a cut in the logical line.
* Null is the term used for absence of any input at all. This should not be confused with a space which is input using the SPC key.

EXAMPLE MEMO IN mode

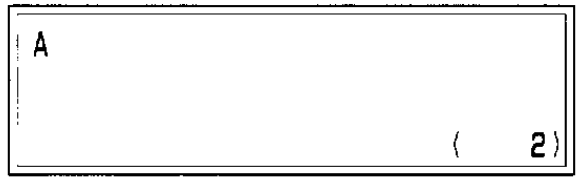
[A][B][C]
[D][E][F]



Considered to be the same logical line.

EXE

A



In the MEMO IN mode, the two physical lines on the screen are considered to be part of the same logical line, while in the CAL mode or BASIC programming mode they are considered to be two different logical lines. The input capacity for each logical line is 255 characters.

5-3 MEMO DATA CORRECTION

Line Insertion

The following procedure is used to insert lines between two previously input lines.

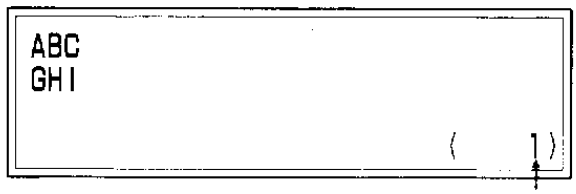
1. Enter the MEMO IN mode.
2. Recall the stored data by pressing MEMO or SHIFT ▲ (SHIFT ▲).
3. Use ▲ (▼) (physical line units) or SHIFT ◀ (SHIFT ▶) (logical line units) to move to the record immediately preceding the location of the insertion.
4. Input the memo data to be inserted and press the EXE key.

* The ▲, ▼ and SHIFT keys only can be used when inserting lines. ◀ and ▶ are not used.

EXAMPLE

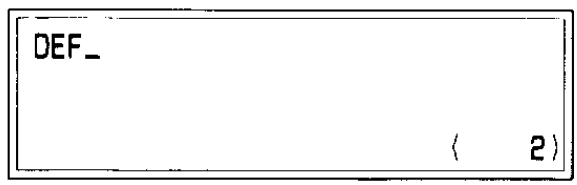
ABC is stored in RECORD #1 and GHI in RECORD #2. Insert DEF between these two records.

MEMO SHIFT ▲ ▲

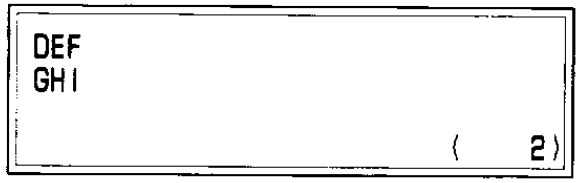


RECORD #1

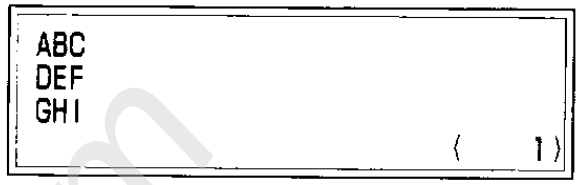
Press D E F.



Press EXE .



Press ▼ .



To insert a new RECORD #1, first display the existing RECORD #1 and press SHIFT ▼ . The unit will stand by for data input, so input the desired data followed by the EXE key.

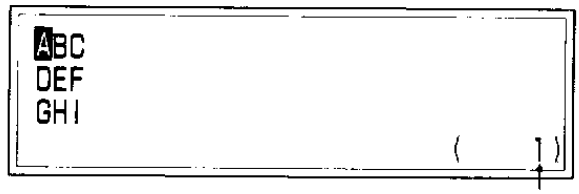
Memo Data Modification

Data already existing in memory can be modified as follows.

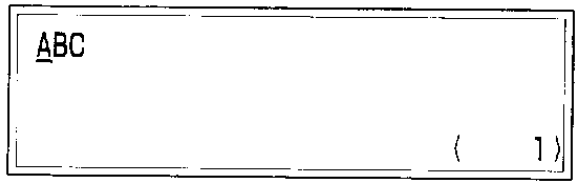
1. Enter the MEMO IN mode.
2. Recall the stored data by pressing MEMO or SHIFT ▼ (SHIFT ▼).
3. Use ▼ (▼) (physical line units) or SHIFT ▼ (SHIFT ▼) (logical line units) to move to the record to be modified.
4. Press either the ◀ or ▶ cursor key to display the cursor.
5. Make the desired corrections and press the EXE key.

EXAMPLE
Change the ABC in RECORD #1 to ABE.

MEMO MEMO

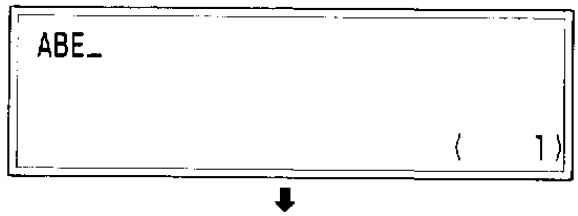


Press ▼ .



The cursor appears under the first character of the record number displayed, and the other lines of the memory are cleared.

Press ◀ ▶ E .



Press the EXE key.

```
ABE
DEF
GHI
                    ( 1)
```

Line Deletion

Full lines can be deleted when they are no longer required within memo data.

1. Enter the MEMO IN mode.
2. Recall the stored data by pressing MEMO or SHIFT ▲ (SHIFT ▼).
3. Use ▲ (▼) (physical line units) or SHIFT ▲ (SHIFT ▼) (logical line units) to move to the record to be deleted.
4. Press either the ◁ or ▷ cursor key to display the cursor.
5. Press CLS EXE . The BS key, SHIFT DEL or space bar can also be used in place of CLS to clear the line before pressing EXE .

EXAMPLE

Delete the ABCDE FGHIJ in RECORD #9.

▲ ▼

```
ABCDE
    FGHIJ
12345
                    ( 9)
```

Press the ▷ key.

```
ABCDE
    FGHIJ
                    ( 9)
```

Press CLS .

```
-
                    ( 9)
```

Press the EXE key.

```
12345
                    ( 9)
```

12345 is data from previous RECORD #10 which is shifted up into RECORD #9.

* *See page 63 for deleting MEMO file.*

5-4 MEMO DATA SEARCH

Once memo data is input and corrected, the following procedures are used to recall specific items.

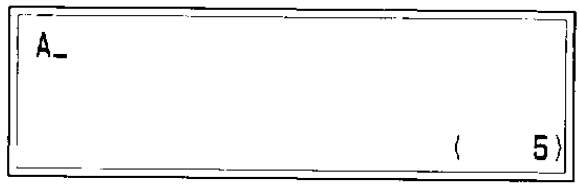
MEMO IN Mode Search

Data is searched for in the MEMO IN mode by inputting the string to be located (up to eight characters long), and then pressing the MEMO key.

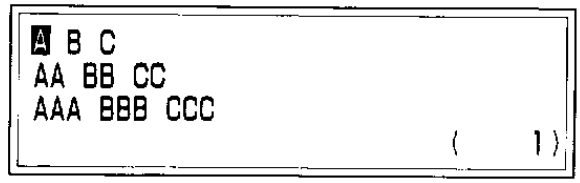
1. When the specified string is located, the first data item found which contains the string is displayed with the specified string in reverse field characters.

(Search object string input)

MEMO [A]



Press the MEMO key.

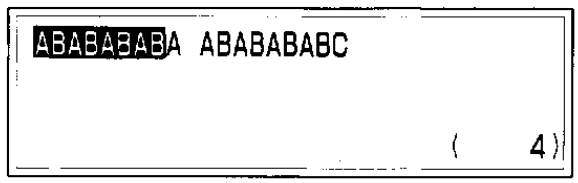


2. Pressing the MEMO key again will continue the search until the next occurrence of the specified string is found. The next data item containing the specified string is displayed with the specified string in reverse field.

Press the MEMO key again.

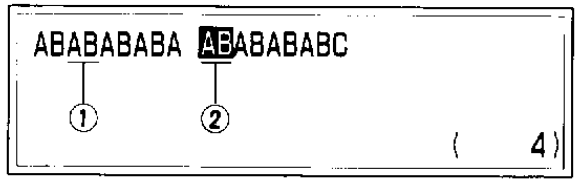
3. The input capacity for the object string is eight characters, and any input exceeding the eighth character is disregarded.

[A][B][A][B][A][B][A][B][C] MEMO



4. Search is performed from the beginning of each data element preceded by a space. Therefore, 1 will not be detected as a match for the specified AB, but 2 will.

AB MEMO MEMO



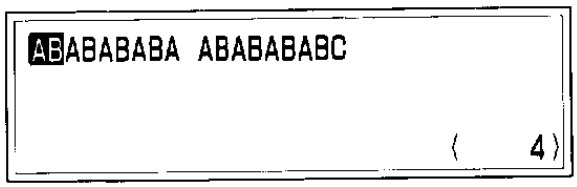
5. The unit stands by for data input at the end of the file if the specified string cannot be found.

A B C MEMO

6. A space cannot be used as a leading character of a search object string.

A B MEMO

A B MEMO



Search Outside of the MEMO IN Mode

MEMO files can also be searched by entering a string followed by the MEMO key while in the BASIC programming mode or CAL mode.

* The MEMO mode is entered as soon as the MEMO key is pressed, but the following situations cause the CAL mode to be entered after string search.
  1. The specified string is not found.
  2. Pressing the BRK, CLS, or OUT key, or the IN key while the final line is being displayed.
  3. Inputting a character (character or symbol).
  4. Pressing the CAL key.
- All other points are identical to those for the MEMO IN mode.
- See PART 6 "OPERATION MODES AND FILES" for details on modes.

5-5 DATA BANK APPLICATIONS

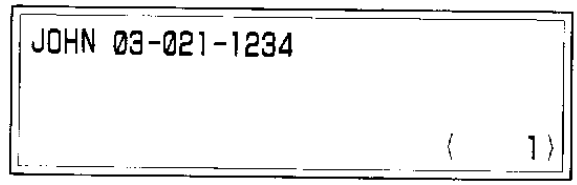
The data bank function can be used in a wide variety of data storage applications.

Electronic Telephone Directory

Here the data bank function will be used to create an electronic telephone directory. First enter the following list of names and telephone numbers into the data bank.

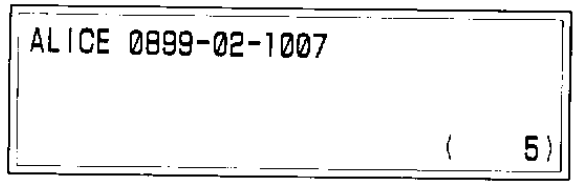
```
JOHN␣03-021-1234
MARY␣011-041-7386
JIM␣06-021-6602
PAUL␣052-031-6221
ALICE␣0899-02-1007
```

[ALPHA] [J] [O] [H] [N] [SPC]
[0] [3] [−] [0] [2] [1] [−] [1] [2] [3] [4] [EXE]



⋮

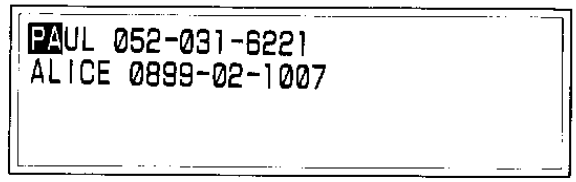
[A] [L] [I] [C] [E] [SPC]
[0] [8] [9] [9] [−] [0] [2] [−] [1] [0] [0] [7] [EXE]



Once all of the telephone directory data is input correctly, Paul's number, for example, can be recalled by the following procedure.

1. Power ON.
2. [P] [A] ([P] [A] [U] [L]) [MEMO]

Paul's name (in reverse field) and telephone number immediately appear on the screen.



Data Bank/Formula Memory Function Combinations

One very powerful application of the data bank is its use in combination with the formula storage function. Various business and technical formulas can be stored in the data bank for individual recall using the MEMO key, storage in the formula memory using the IN key, and execution using the CALC key. This means that a wide selection of important formulas can be accessed with the touch of a key, without programming.

The formula for interest applied to savings, the formula would be:
INTEREST = PRINCIPAL × ANNUAL RATE × NUMBER OF MONTHS/12

First, the formula is input for storage in the data bank.
(Input ITRT as INTEREST, PRIN as PRINCIPAL, RATE as ANNUAL RATE and MO as NUMBER OF MONTHS.)

SHIFT I T R T = P R I N * R A T E /
1 0 0 * M O / 1 2 EXE

```
ITRT=PRIN*RATE/100*MO/12

                          (    1)
```

Now the interest formula is available for recall when it is required.

1. Power ON.
2. I T MEMO (I T R T MEMO)

```
ITRT=PRIN*RATE/100*MO/12
```

Now simply press the IN key to store the formula in the formula memory. Here, calculate the interest for 6 months on a principal of $1000 at an annual rate of 5.5%.

IN CALC 1 0 0 0 EXE
5 . 5 EXE
6 EXE

```
PRIN?1000
RATE?5.5
MO?6
ITRT=27.5
```

Memo Data Handling Precautions

Data can be written to a MEMO file using BASIC commands. In the MEMO and MEMO IN modes, however, control codes (&H00 ~ &H1F) are displayed as spaces. Therefore, modifying a MEMO file line created using BASIC commands causes all control codes to be converted to spaces. An &H1A code within a file is defined as the end of file code and all data past that code cannot be recalled. Also, logical lines longer than 255 characters result in a BV error and cannot be recalled. All of this is also true in the DATA editing mode. Therefore, it is recommended that such data (control codes and lines longer than 255 characters) be handled on only in BASIC.

PART 6
OPERATION MODES AND FILES

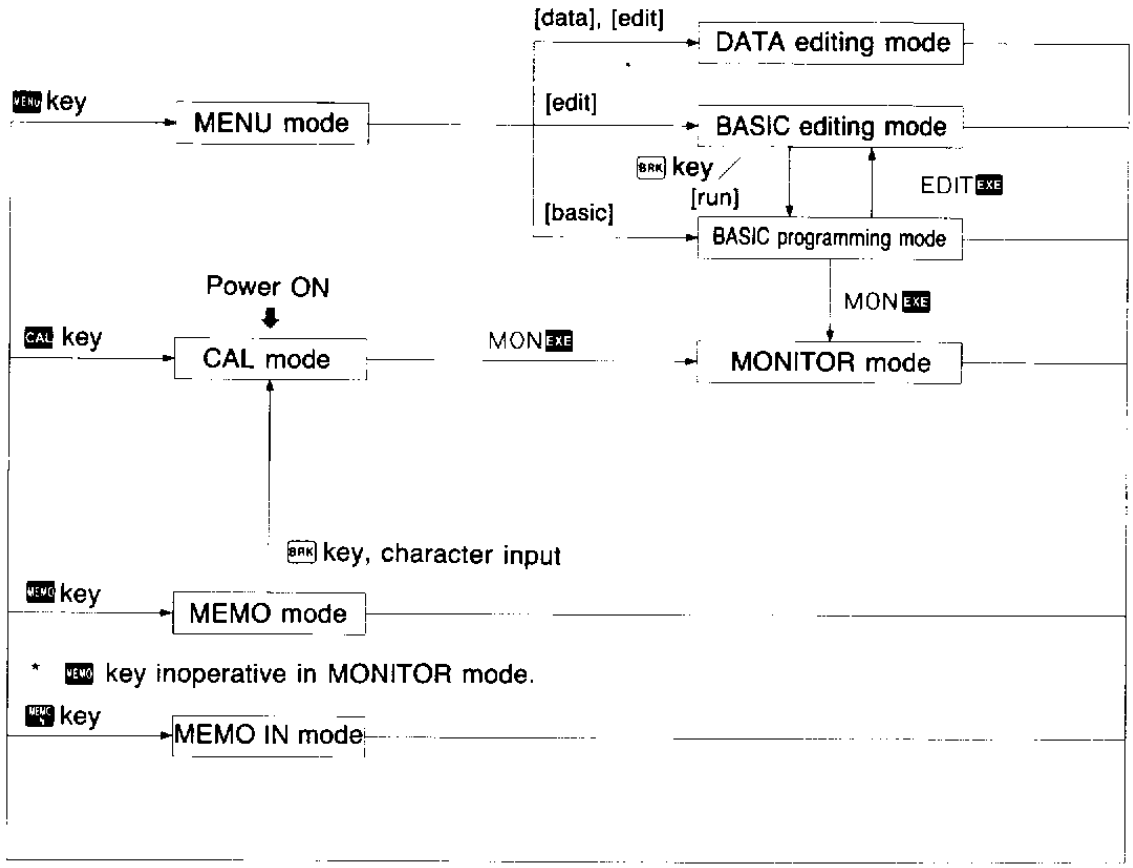
6-1 OPERATION MODE FUNDAMENTALS

This unit has a number of operation modes that are selected according to the application. The following is a list of the various modes available.

a) CAL mode.....calculator functions
b) MENU mode.....menu selection
c) MEMO mode.....memo data display and search
d) MEMO IN mode.....memo data input
e) BASIC programming mode.....BASIC program input and execution
f) BASIC editing mode.....BASIC program editing
g) DATA editing mode.....sequential file (ASCII format files, see page 91 for details) input and editing
h) MONITOR mode.....machine language program editing

* The BASIC programming, BASIC editing, and DATA editing modes are selected from the MENU mode.

6-2 FUNDAMENTAL MODE SELECTIONS



- Pressing the CALC key in any mode, or pressing the BRK key enters the CAL mode.
- Characters enclosed in brackets represent MENU screen touch keys.

6-3 OUTLINE OF EACH MODE

The following table shows the pages where detailed a explanation of each mode appears.

<MODE>			<PAGE>
LCKEY			58
MENU	basic		59
	data		61
	edit		61
	disk	name	62
		kill	62
		load	62
	name		62
	kill		63
	load	RS232C	65
		disk	65
		MT	64
	save	RAM	68
		RS232C	68
		disk	67
		MT	67
	asmbl		69
	llist		70
	c.boot		70
	preset		70
CAL			52
MEMO			52
MEMO IN			53
IN			35
OUT			35
CALC			35

* The MENU mode is explained in PART 7.

CAL Mode

Purpose

1. Manual calculations (calculator functions)
2. BASIC command manual execution

Specification

This mode is entered when:

1. Power is switched ON.
2. CAL is pressed.
3. BRK, OUT, CLS, ANS or character keys are pressed in the MEMO mode, or when memo data does not exist.
4. Formula storage function calculations are complete, BRK is pressed during formula storage function calculation, or when an error is generated during formula storage function execution.
5. Program execution begun in the CAL or MENU mode is complete or when an error is generated during execution.

MENU Mode

Purpose

1. Display and selection of filenames stored in internal memory
2. Call of BASIC programming and BASIC editing modes
3. Call of the DATA editing mode
4. Disk file processing (disk MENU)
5. Internal memory file processing
6. Source program assembly
7. Clock boot setting
8. One-touch file setting
9. Program (BASIC, machine language) execution
10. File printout

Specification

This mode is entered when the MENU key is pressed in any mode.

MEMO Mode

Purpose

Memo data display and search

Specification

This mode is entered when the MEMO key is pressed in any mode besides the MEMO IN mode or MONITOR mode. The MEMO key is inoperative unless data exists in a MEMO file.

MEMO IN Mode

Purpose
Memo data input and editing

Specification
This mode is selected when the MEMO IN key is pressed.

*** IMPORTANT**
When this key is pressed, any BASIC or machine language file present as a MEMO is deleted and a new sequential file named MEMO (data bank file) is created.

BASIC Programming Mode

Purpose
1. BASIC program execution
2. BASIC command manual execution (excluding multi-statements)
3. BASIC programming

Specification
This mode is entered when:
1. [basic] is pressed on the MENU screen.
2. BRK is pressed in the BASIC editing mode.
3. An error is generated after a program is executed by pressing [run] in the BASIC editing mode, or when program execution is terminated or suspended.

BASIC Editing Mode

Purpose
1. BASIC program editing
2. BASIC program execution (BASIC editing mode entered when execution is complete.)

Specification
This mode is entered when:
1. [edit] is pressed while the BASIC program name to be edited is indicated by the menu pointer on the MENU screen.
2. The EDIT command is executed in the BASIC programming mode.

DATA Editing Mode

Purpose
1. Sequential file (i.e. data bank file) creation
2. Assembler source program writing and editing

Specification
This mode is entered when:
1. [data] is pressed on the MENU screen.
2. [edit] is pressed while the sequential file to be edited is indicated by the menu pointer on the MENU screen.
3. A touch screen key which is preset as a one-touch file is pressed in the CAL mode.

MONITOR Mode

Purpose
Machine language program modification.

Specification
This mode is entered when the MON command is executed in the CAL or BASIC programming mode.

6-4 FILES

The word "file" has already appeared a number of times in this manual. Here, the computer will be used to create a few files to provide an idea of what the term actually means.

What Is a File?

A file is data saved under a filename in main memory, on a floppy disk, or on cassette tape. This unit automatically creates a file whenever programs or data are input.

Types of Files

The following table shows the four types of files available and their identifiers.

Identifier	File
B	BASIC program file
R	Random file (disk only)
S	Sequential file or source program file in ASCII format
M	Machine language file

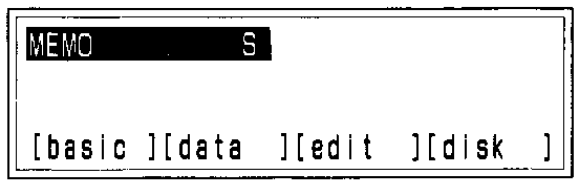
File Creation

This unit automatically creates a file whenever a program or data are input.

1. A work file appended with the B identifier is created when in the BASIC programming or BASIC editing mode.

2. A work file appended with the S identifier is created when in the DATA editing mode.

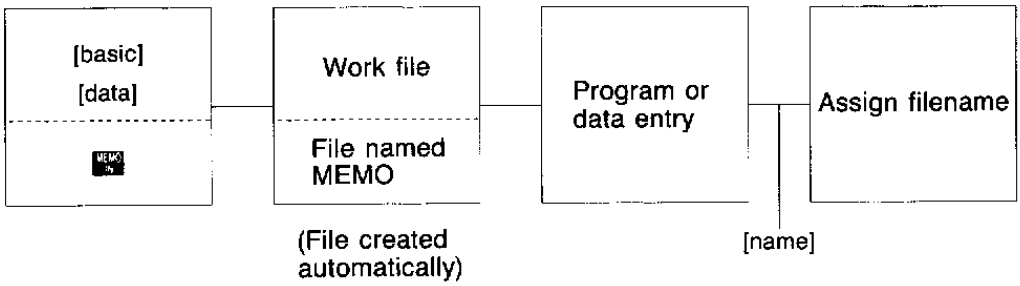
3. A file with the name MEMO appended with the S identifier is created when in the MEMO IN mode.



Work Files

Storage area is required for programs and data entered into the computer. When writing a BASIC program, for example, the [basic] touch screen key is pressed and the BASIC programming mode is entered. A work file (without a name) appended with the B identifier is created at this time to store the program. Even though this file does not have a name, it can be used for such file handling functions as program execution and editing, and is recalled whenever its mode is entered. When the second and subsequent files are created, however, it is necessary to assign names to the files to distinguish them from one another. Assigning a filename to a program moves it out of the work file and stores it in memory under its filename, freeing the work file for new input. Therefore, it is recommended to assign filenames to work files as soon as data are entered.

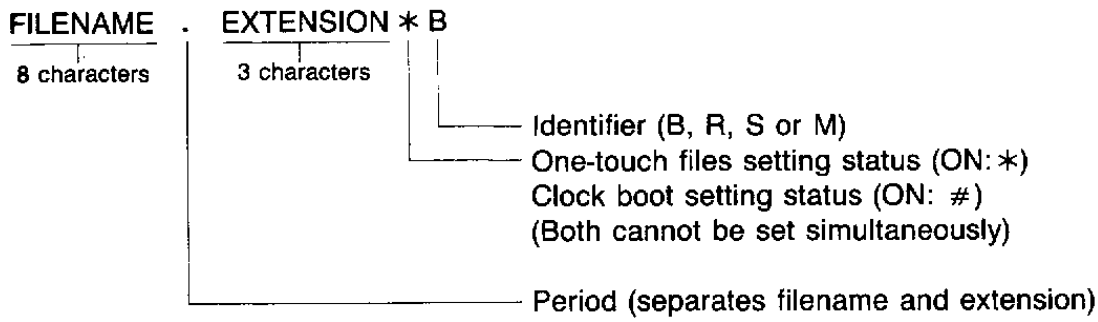
• File Creation Procedure



All newly entered data are stored under the work file. It is recommended that this file always be assigned a name as soon as data entry is complete. After a name is assigned to this file, subsequent new entries are stored in another work file.
Sequential files can also be created using the BASIC OPEN command.

Filenames

The filenames of this unit can be up to eight characters long, followed by up to three characters as an extension. Each filename is displayed in 14 characters on the MENU screen as illustrated below.



NOTE: Periods and colons cannot be used within the eight characters of the filename. Blank filenames cannot be used except with [basic] and [data].

Device Name

Are classified and communicated according to the device used for storage as noted below.

Device Name	File Destination
0 :	Disk
CAS0 :	Cassette tape
COM0 :	Communications circuit
None	Main memory

EXAMPLE: LOAD "CAS0:TEST"
Loads a program named "TEST" from cassette tape.

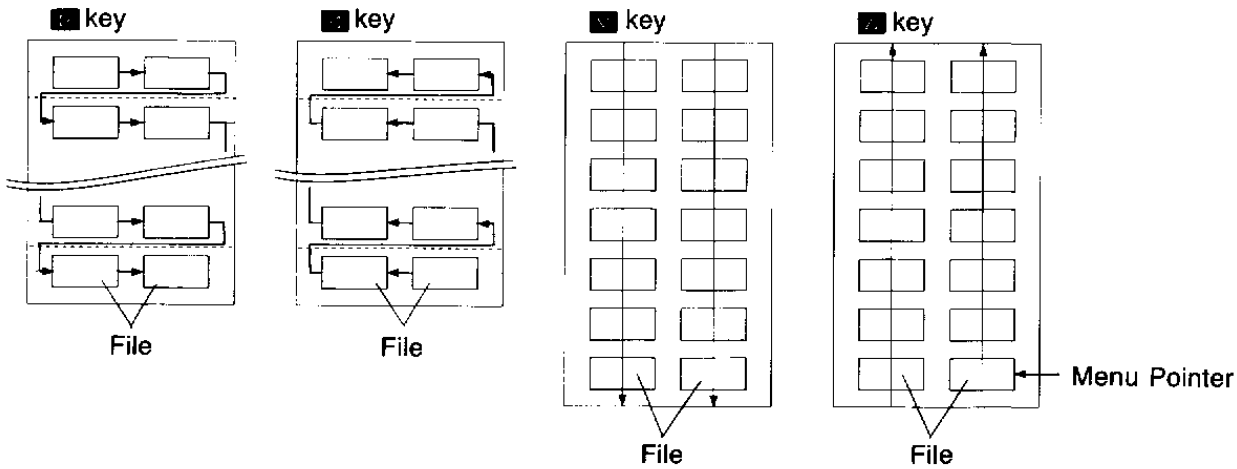
AUTO. EXE File

The AUTO.EXE allows automatic execution of an identifier B (BASIC) file or M (machine language) file whenever the power of the unit is switched ON. An S (sequential file) identifier causes the unit to enter the DATA editing mode when power is switched ON. Note, however, that this function is only valid for files already existing in memory only.

File Specification

A file must first be specified before programs and data can be executed or edited. Filenames are displayed on the screen in two columns of three lines each per screen. The cursor keys are used to move the menu pointer to the desired filename, and also to scroll the screen to view any additional filenames not included on the screen.

- **Menu Pointer Movement**



* The following keys become inoperative when the menu pointer is located at the positions noted.

Menu pointer location
key Upper left (first) filename
key Last filename
key Top filename line
key Bottom filename line

PART 7
MENU FUNCTION

7-1 MENUS

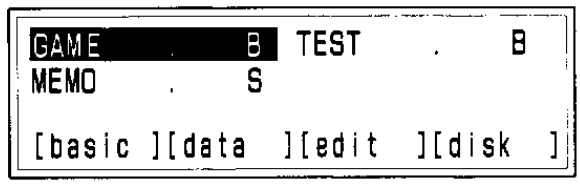
The MENU screen is displayed whenever the MENU key is pressed. The fourth line of the MENU screen includes a set of touch keys for easy selection of a wide variety of features. Lines one through three are used for the display of user generated filenames for files currently stored in memory.

7-2 MENU SCREEN

The MENU screen differs according to whether or not files are stored in memory.

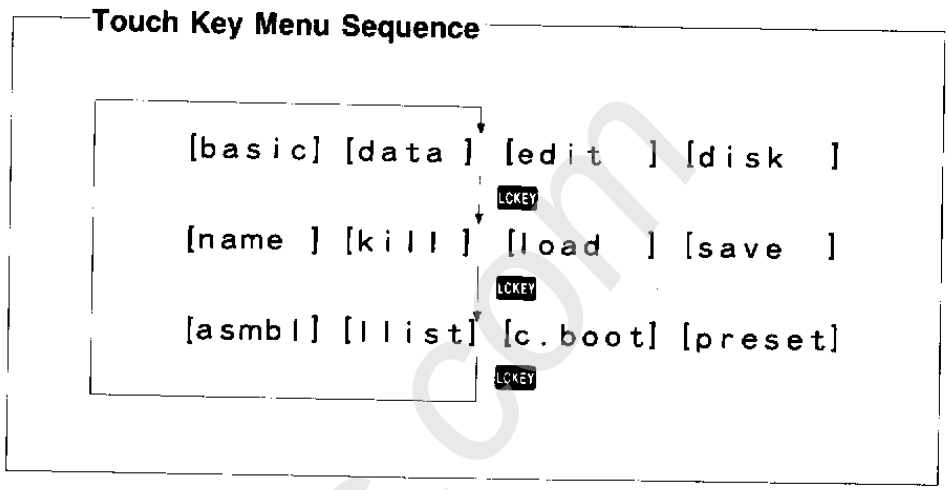
- **Files not stored in memory**

- **Files stored in memory**



7-3 CHANGING TOUCH KEY FUNCTIONS

The MENU screen actually has three sets of touch screens, though only one set can be displayed at any one time. The LCKEY key is used to change the touch keys that appear on the MENU screen in the following order.



EXE key

Pressing the EXE key executes the BASIC or machine language program file at which the menu pointer is currently located. The unit enters the DATA editing mode when the menu pointer is located at a sequential file.

* EXE will execute a machine language program, but an AM (access mode) error is generated when the file is a binary data file.
* The CAL mode is automatically entered after the execution of a BASIC or machine language program is complete.

7-4 MENU SELECTION

A total of 12 different modes can be selected using the touch keys in the fourth line of the screen.

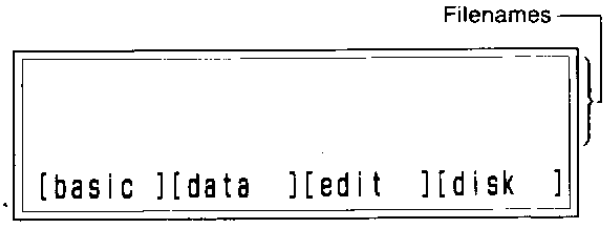
Mode	Operation
basic	BASIC programming.
data	DATA editing mode for input, modification and editing of a sequential file.
edit	BASIC editing mode for modification and editing of a previously input BASIC program. Similar to the DATA editing mode.
disk	Used when an MD-100 FDD is connected for disk file operations (name, kill, load).
name	Assignment or change of filename for file in main memory.
save	Transmission of files in main memory to floppy disk, cassette tape, main memory or through RS-232C.
load	Input of files from floppy disk, cassette tape or through RS-232C to main memory.
kill	Deletion of file in main memory.
asmbl	Assembly of source program.
llist	Output of data file or program file to printer.
c.boot	Specification of a program for execution on a particular date at a particular time.
preset	Setting and resetting of one-touch files (direct execution of programs using the touch keys in the CAL mode).

basic

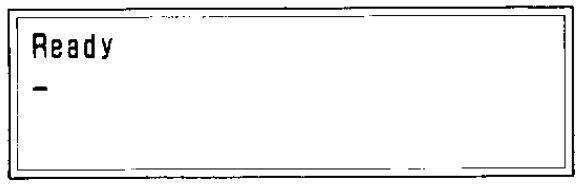
Selecting [basic] on the MENU screen enters the BASIC programming mode, and causes the "Ready" prompt to appear on the screen.

```
Ready
-
```

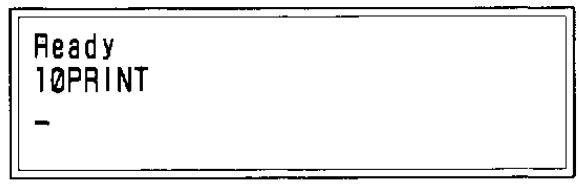
a) Press MENU.
The screen illustrated here appears when there are no files stored in memory. Filenames would appear in lines one through three on the screen if files were present.



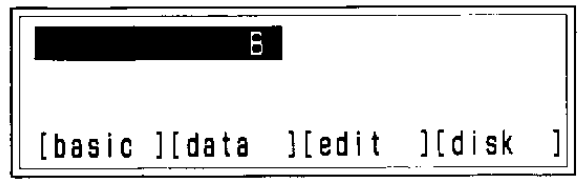
b) Press [basic].
BASIC programming mode is entered and the "Ready" prompt appears on the screen.



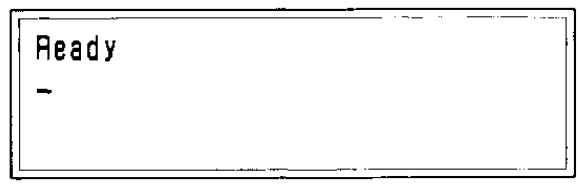
c) Press 10 SHIFT PRINT EXE.



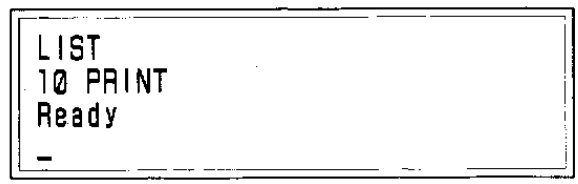
d) Press MENU.
A work file is registered in reverse field.



e) Press [basic].
Unlike step b), this operation selects the previously registered work file.



f) Press SHIFT LIST EXE.
The previously entered program is displayed.



See page 73 for details on BASIC programming.

data

Pressing [data] on the MENU screen enters the DATA editing mode for sequential files. Assembler source programs are created in this mode. As with the [basic] selection, a previously created sequential work file is selected if one exists in memory and is displayed from the first line. If a file does not already exist, a sequential work file is created at this time and the unit stands by for input of the first line of the file.

edit

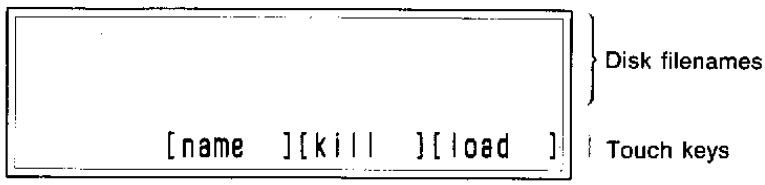
Pressing [edit] on the MENU screen enters either the BASIC editing mode or DATA editing mode, depending upon the location of the menu pointer when [edit] touch key is pressed. This key become inoperative when no file exists for editing.

* A PR error is generated and the BASIC programming mode is entered when a password is registered for the selected file. Passwords can only be registered for BASIC files.
* The "Ready" prompt appears and the BASIC programming mode is entered when the selected BASIC file is empty.

See page 74 for details on the BASIC editing mode, and page 53 for the DATA editing mode.

disk

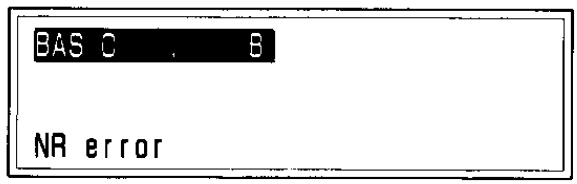
An optional 3.5" floppy disk drive unit can be connected to this unit for storage of large volumes of data and long programs. Pressing [disk] on the MENU screen changes to a Disk menu screen which shows the files stored on the floppy disk and the disk touch key display.



* [disk] is inoperative under the following conditions:

a) When the floppy disk drive is improperly connected.

b) When a disk is not loaded in a floppy disk drive. An NR error is generated when [disk] is pressed.



* The MENU screen is returned to from the Disk menu screen by pressing the MENU key (not the BRK key).

a) Changing a Disk Filename
Press [name]. The disk filename at which the reverse field is currently located can be changed after [name] is pressed.

b) Deleting a Disk File
Pressing [kill] followed by [Y] key deletes the disk file under the filename at which the menu pointer is currently located.

c) Disk File Load
Pressing [load] loads the disk file under the filename at which the menu pointer is currently located into main memory. The display returns to the MENU screen after program load is complete.

* The file loaded from the disk receives priority in memory, so any existing file in internal memory with the identical filename is erased.
* When the size of a sequential disk file exceeds the amount of available internal memory capacity, the file is loaded until the memory is full, and then an OM error is generated. For BASIC and machine language program files, the OM error is generated immediately after the filename is loaded (see page 128). The filename cannot be loaded unless at least 32 bytes of internal memory capacity is available.

d) Disk File Load and Execute
Pressing [EXE] loads the disk file under the filename at which the menu pointer is currently located and automatically executes the file.

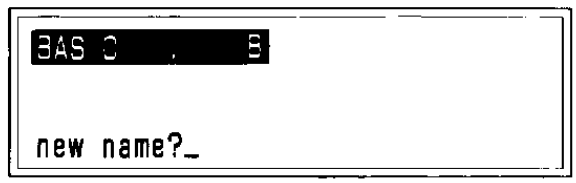
* An AM error is generated when an attempt is made to load a random data file using this procedure.
* The DATA editing mode is entered when a sequential file is specified by the menu pointer.
* The CAL mode is entered after program execution is complete.

IMPORTANT:
Never change floppy disks while the Disk menu screen is displayed. First return to the MENU screen, change disks, and return to the Disk menu.

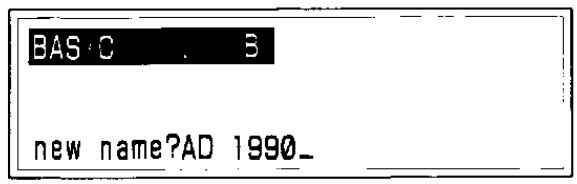
name

Pressing [name] on the MENU screen makes it possible to assign or change the filename at which the menu pointer is currently displayed.

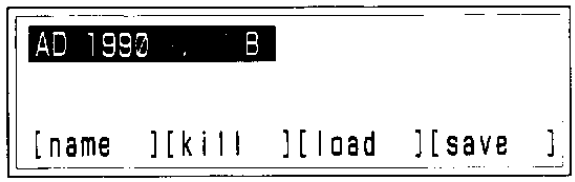
a) Use the cursor keys to move the menu pointer to the filename to be changed.
b) Press [name] and the prompt "new name?" appears on the screen.



c) Enter the desired filename.



d) Press EXE .

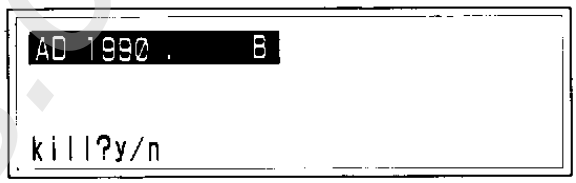


* Pressing EXE without entering a filename returns to the MENU screen.
* A filename + extension identical to a filename + extension already existing in main memory cannot be used.
* [name] is inoperative then no files exist in memory.
* Assigning a name to a work file using the [name] operation makes it possible to create another file when [basic] is pressed. Multiple files can be simultaneously stored in memory using this procedure.

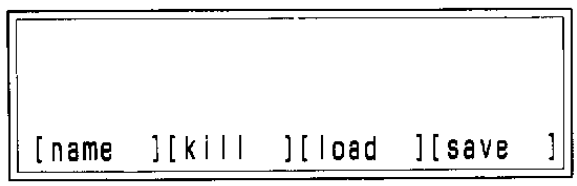
kill

Pressing [kill] on the MENU screen deletes the file under filename at which the menu pointer is currently located.

a) Use the cursor keys to move the menu pointer to the filename to be deleted.
b) Press [kill] and the prompt "kill?y/n" appears on the screen.



c) Press the Y key on the keyboard to delete the file, or the N key to return to the MENU screen without deleting the file.



* [kill] is inoperative then no files exist in memory.
* [kill] will delete a file even if it has a password.

load

Pressing [load] on the MENU screen loads a program or data file from cassette tape or floppy disk into main memory. [load] is also used to load a program or data from an external device via the communications circuit (RS-232C interface).

* [load] is inoperative unless an optional interface unit or floppy disk drive unit is connected.
* Subsequent menus and operations after [load] is pressed differ according to the type of external device being used.

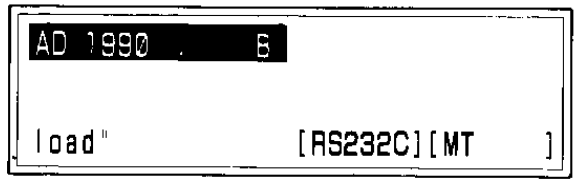
<MT>

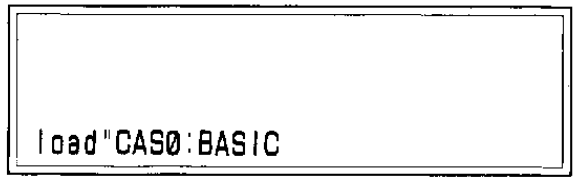
a) Connect a cassette recorder to the interface unit, load the cassette tape which contains the file to be loaded, and press the recorder's PLAY button.

b) Press [load], and the following prompt will appear on the fourth line of the screen.
load" [RS232C] [MT]

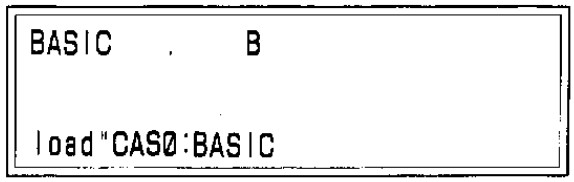
c) Press [MT] (magnetic tape).

d) Enter the name of the file to be read and press EXE .

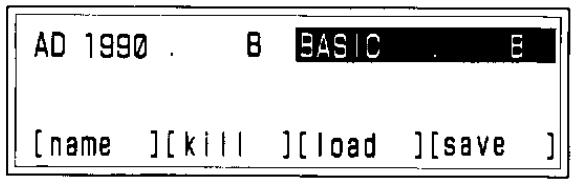




e) The first three lines of the screen are cleared and the specified filename appears when file loading begins. If other files exist on the tape, their names are also displayed as the programs are skipped to find the specified file.



f) The MENU screen appears with the filename of the newly loaded file indicated by the menu pointer when load operations are complete.



* The MT baud rate (transmission speed) is set using the baud rate switch on the back of the interface unit (see page 120).
* [load] is inoperative when the interface unit is not connected.
* Files loaded from tape receive priority in memory, so any existing file in internal memory with the identical filename (filename + extension) is erased.
* The first file found on the tape is loaded when [load] is executed for MT without specifying a filename.
* Reading a work file from tape erases the work file currently in main memory.
* Loading a file from tape at a baud rate which is different from that used when the program was saved generates an error or skips the specified program. Always load programs at the same baud rate as that used to save the program.
* If a file cannot be loaded successfully from tape, try setting the PHASE switch on the right of the interface unit to REV, and then attempt load procedures again.

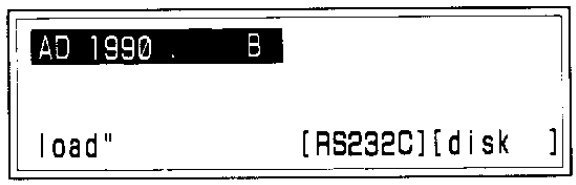
<DISK>

a) Connect a floppy disk drive unit, and set the floppy disk which contains the file to be loaded.

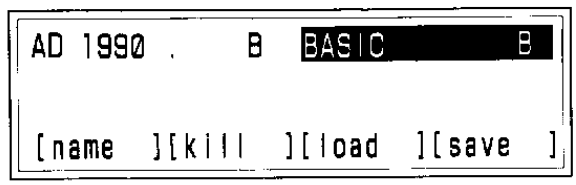
b) Press [load], and the following prompt will appear on the fourth line of the screen.
load " [RS232C] [disk]

c) Press [disk].

d) Enter the name of the file to be read and press EXE.



e) The MENU screen appears with the filename of the newly loaded file indicated by the menu pointer when load operations are complete.

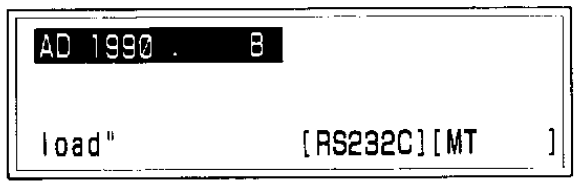


* [load] is inoperative when the floppy disk drive unit is not connected.
* Files loaded from disk receive priority in memory, so any existing file in internal memory with the identical filename (filename + extension) is erased.

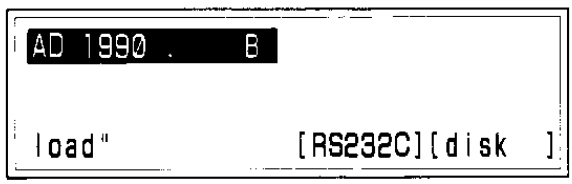
<RS-232C>

a) Connect an interface unit or floppy disk drive unit, and set the RS-232C switch of the connected unit to ON.

Connection with FA-7



Connection with MD-100



b) Press [load], and one of the following prompts will appear on the fourth line of the screen, in accordance with the type of unit connected.
load " [RS232C] [MT]
load " [RS232C] [disk]

c) Press [RS232C].

d) Enter the communications parameters (see below) for the file transfer as follows.

load "COM0 : 2 , E , 8 , 1 , C , D , C , N , N
a b c d e f g h i

* The parameters shown here are only intended to act as an example. Actual parameters would differ according to each specific communications link.

* Confirm the parameters on the screen and make any required changes by moving the cursor to the appropriate position using the cursor ◄ ► keys. Finally, press EXE.

(RS-232C Paremeters)

a) Baud Rate

The following values indicate bits per second (BPS).

[0 = 75, 1 = 150, 2 = 300, 3 = 600, 4 = 1200, 5 = 2400, 6 = 4800, 7 = 9600]

* The default value is the baud rate switch setting on the back of the connected unit.

b) Parity Bit

N = None, E = Even, O = Odd

c) Character Bit Length

7 = JIS 7 bits, 8 = JIS 8 bits

d) Stop Bit Length

1 = 1 stop bit, 2 = 2 stop bits

e) CTS Signal Control

C = CTS control, N = No CTS control

f) DSR Signal Control

D = DSR control, N = No DSR control

g) CD Signal Control

C = CD signal control, N = No CD signal control

h) Buffer Busy Control

B = Buffer busy control, N = No buffer busy control

i) SI/SO Control

S = SI/SO control, N = No SI/SO control

The initialized default values for the RS-232C parameters (after NEW ALL is pressed) are the following:

2 , E , 8 , 1 , N , N , N , B , N

* Parameter settings only are entered for the "load" prompt of RS-232C loading. Filename input is not necessary.
* Data transmitted through the RS-232C are loaded to the sequential work file.
* A parameter setting of no parity, data length 7, 1 stop bit cannot be used with a baud rate of 9600.
* The menu pointer can be located anywhere during RS-232C [load] operations.
* Press BRK to terminate RS-232C [load] operations.
* Always ensure that the RS-232C switch of the connected unit is set to ON when using the RS-232C interface.
* BASIC and machine language files may be entered in memory even though an error (excluding OM error) is generated during load to the main memory. Though the file is present, it should not be assumed that the file was loaded correctly. Anytime an error is generated during loading, delete the file (using [kill]) and reload. This is also true when the BRK key is pressed during BASIC or machine language file loading from tape.
* Clock boot becomes inoperative during save/load operations and reactivates after save/load is complete.
* BRK is inoperative during disk reading.

save

Pressing [save] on the MENU screen saves a program or data file from main memory onto a cassette tape, floppy disk, or main memory. [save] is also used to transmit a program or data to an external device via the communications circuit (RS-232C interface).

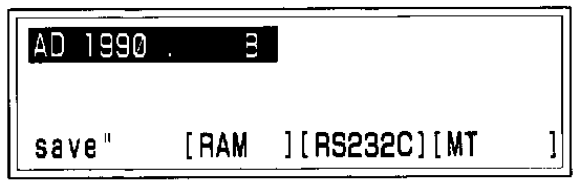
* Save operations using tape, disk or RS-232C are impossible unless an optional interface unit or floppy disk drive unit is connected. Pressing [save] in this situation specifies the main memory.
* Subsequent menus and operations after [save] is pressed differ according to the type of external device being used.

<MT>

a) Connect a cassette recorder to the interface unit and load a cassette tape.
b) Align the menu pointer with the name of the file to be saved.

c) Press the REC and PLAY buttons on the recorder, and press [save]. The following prompt will appear on the fourth line of the screen.

save " [RAM] [RS232C] [MT]

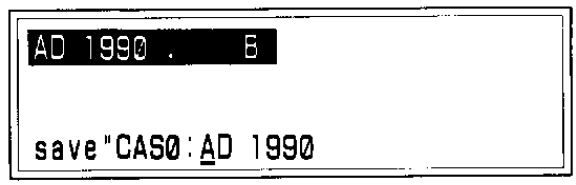


d) Press [MT] (magnetic tape).

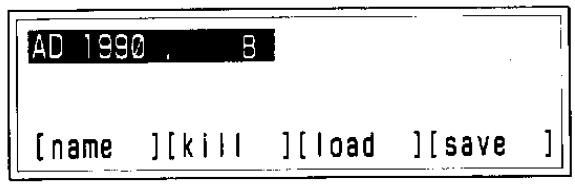
e) The following message will appear on the fourth line of the screen.

save "CAS0 : (menu pointer filename)

Make any desired changes (or leave the filename as it is) and then press EXE .



f) The unit returns to the MENU screen when save operations are complete.

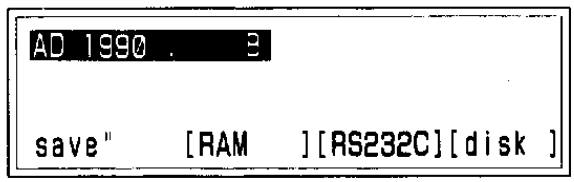


<disk>

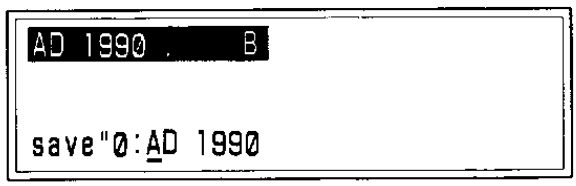
a) Connect a floppy disk drive unit, and load a floppy disk.
b) Align the menu pointer on the screen with the name of the file to be saved.

c) Press [save], and the following prompt will appear on the fourth line of the screen.

save " [RAM] [RS232C] [disk]



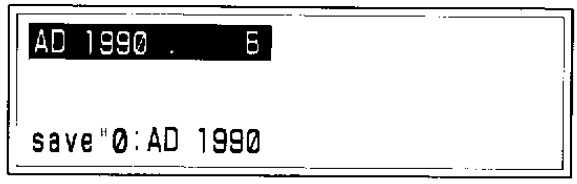
d) Press [disk].



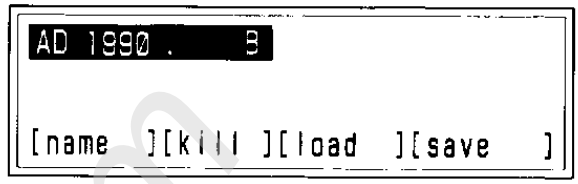
e) The following message will appear on the fourth line of the screen.

save "0 : (menu pointer filename)

Make any desired changes (or leave the filename as it is) and then press EXE .



f) The unit returns to the MENU screen when save operations are complete.



<RS-232C>

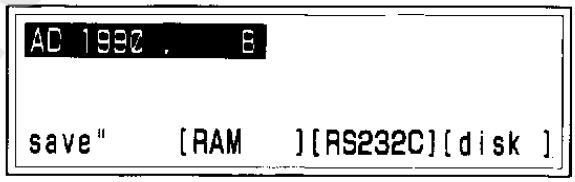
a) Connect an interface unit or floppy disk drive unit, and set the RS-232C switch of the connected unit to ON.

b) Align the menu pointer on the screen with the name of the file to be saved.

c) Press [save], and one of the following prompts will appear on the fourth line of the screen, in accordance with the type of unit connected.

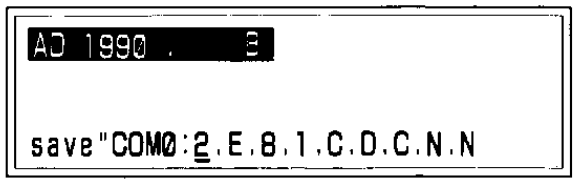
save "[RAM] [RS232C] [MT]
save "[RAM] [RS232C] [disk]

Connection with MD-100



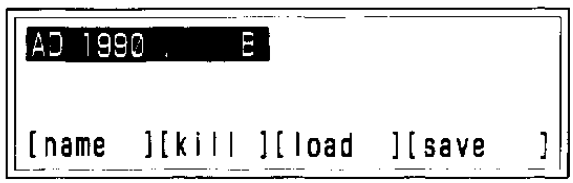
d) Enter the communications parameters (see page 65) for the file transfer as follows.

save "COM0 : 2 , E , 8 , 1 , C , D , C , N, N



* The parameters shown here are only intended to act as an example. Actual parameters would differ according to each specific communications link.

e) The unit returns to the MENU screen when save operations are complete.



* Save operations to the RS-232C interface automatically converts the file to ASCII format.
* Machine language files cannot be saved.

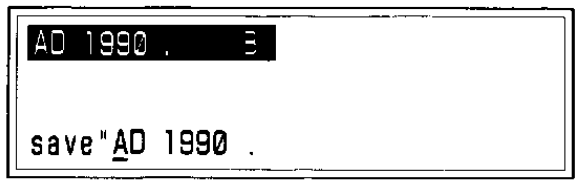
<Main Memory>

a) Align the menu pointer on the screen with the name of the file to be saved.

b) Press [save], and one of the following prompts will appear on the fourth line of the screen, in accordance with the type of unit connected.

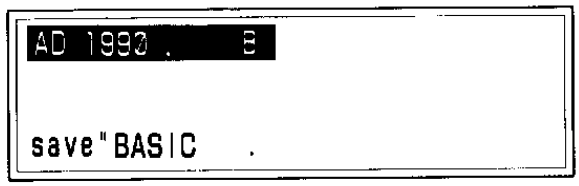
save "[RAM] [RS232C] [MT]

save "[RAM] [RS232C] [disk]



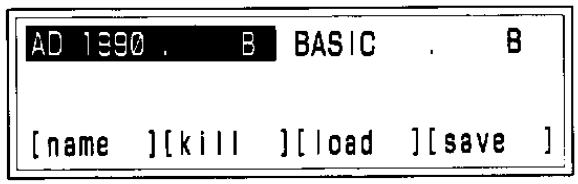
c) Press [RAM].
d) The following message will appear on the fourth line of the screen.

save "menu pointer filename"



Since two files with identical filenames cannot exist in memory, a filename different from that assigned to the file currently in memory must be assigned.

e) The unit returns to the MENU screen with name of the newly saved file indicated by the menu pointer when save operations are complete.



IMPORTANT

* Saving a file to main memory in effect makes a duplicate copy of an existing file.
* Since two files in main memory cannot have identical filenames, attempting to save a file to main memory without changing the filename results in a BF error.
* Affix " , A" following the filename to save a binary file in ASCII format.
* [BRK] is inoperative during save operations to disk or RAM.

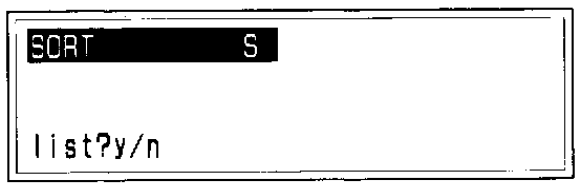
asmbl

This functions is used to convert (assemble) a program written in assembly language into a machine language program. Pressing [asmbl] assembles the source file (created in the DATA editing mode) indicated by reverse field. The prompt [list?y/n] will appear when the interface unit or floppy disk drive unit is connected, and pressing the [Y] key at this time will produce a hardcopy of the program list on the printer.

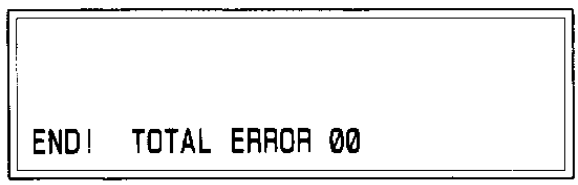
* Press either the [N] key or any key other than the [Y] key to cancel the print command.
* Assembly of the source program begins immediately after [asmbl] is pressed when an interface or floppy disk drive unit is not connected.

The following message appears once assembly operations are complete (when no errors are found).

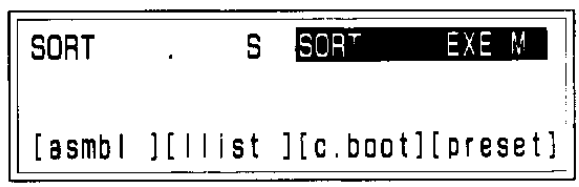
[END! TOTAL ERROR 00]



At this time, pressing [EXE] or [MENU] returns to the MENU screen, with the newly assembled machine language program indicated by the menu pointer, ready for execution.



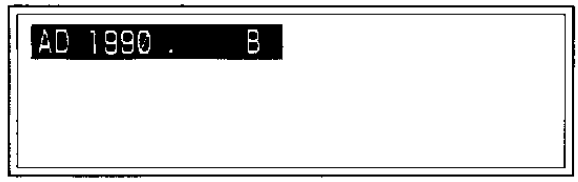
* See PART 10 for details on using the assembler function.
* [asmbl] is inoperative except for sequential files.



llist

Pressing [llist] outputs to the printer the contents of the file under the filename specified by the menu pointer. The menu on the fourth line of the the screen is cleared during printing operations, and the unit returns to the MENU screen after printing is complete.

* [llist] is inoperative when a password is registered.
* [llist] is inoperative when no files are present or when an interface unit or floppy disk drive unit is not connected.
* [llist] cannot be used for machine language files.
* The unit stands by for input when a printer error is generated or when the printer is off line.



4th line menu cleared during print operation.

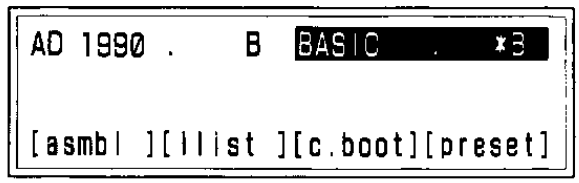
c.boot

Pressing [c.boot] makes it possible to enter a specific time and date for automatic execution of a BASIC or machine language program. File contents are displayed at the specified date and time when [c.boot] is set for a sequential file (DATA editing mode).

* Automatic execution takes place regardless of whether the power of the unit is switched ON or OFF.
* Automatic execution takes place even if a BASIC program is being executed.
* See part 9, section 3 on page 83 for details on the clock boot.

preset

Pressing [preset] creates a one-touch file using the file under the filename specified by the menu pointer. The one-touch file can then be directly executed by pressing the appropriate touch screen key while in the CAL mode. At this time, an asterisk will be inserted between the file extension and identifier on the MENU screen for the file assigned as a one-touch file. Up to four files can be assigned as one-touch files, and the specified one-touch filenames are displayed on the fourth line of the CAL mode display. Execution of a one-touch program is a simple as pressing the corresponding touch screen key.



* See PART 9, section 4 on page 85 for details on one-touch files.
* Executing a sequential file (while in the CAL mode) specified as a one-touch file automatically enters the DATA editing mode.

7-5 TYPICAL MENU MODE ERRORS

Operation	Result
a) [basic] or [data] pressed while memory full	OM error
b) [edit] pressed for machine language file	No operation
c) [disk] pressed while FDD drive not connected	No operation
d) [disk] pressed while disk not loaded	NR error
e) [disk] pressed for unformatted floppy disk	FM error
f) Attempt to load random file while in Disk menu	AM error
g) Change of disk while in the Disk menu and attempt to use new disk without pressing MENU [disk]	NF error for [name], [kill], [load] operations for new disk
h) Attempt to assign existing name using [name]	Return to input stand by
i) Attempt to assign illegal name using [name]	BF error
j) [load] pressed without connection of interface or FDD unit	No operation
k) [save] [disk] EXE pressed while disk is not loaded	NR error
l) [load] [disk] EXE pressed while unformatted disk is loaded	FM error
m) [asmbl] pressed for non-sequential file	No operation
n) [llist] pressed without connection of interface or FDD unit	No operation
o) [llist] pressed without connection of printer	Computer waits until printer ready. BRK returns to MENU screen.
p) Attempt to specify more than one [c.boot]	No operation
q) Attempt to specify more than four [preset] keys	No operation
r) Erroneous entry of [c.boot] time or date	Returns to stand by
s) [edit] pressed for password protected file	Unit enters BASIC programming mode and displays: PR error Ready — (PASS can only be assigned to BASIC file.)
t) [edit], [name], [kill], [save], [asmbl], [llist], [c.boot], [preset] pressed when file not present	No operation
u) [load] pressed while memory full	OM error
v) [edit] or EXE key pressed while more than 255 characters exist in first line of sequential file	BV error Unit enters DATA editing mode
w) Erroneous filename entry for [load] or [save]	BF error
x) [load] filename cannot be found	NF error
y) Attempt made to save file to memory without changing filename	BF error Return to MENU screen

PART 8
BASIC PROGRAMS

This unit can be programmed using an enhanced version of BASIC. Since BASIC is a commonly used personal computer language, a large amount of literature devoted to detailed programming and application procedures is available on the market. Details concerning the specific BASIC commands used by this unit are provided in a separate COMMAND REFERENCE, so this manual will simply cover precautions to be aware of during BASIC programming.

8-1 FUNDAMENTALS OF BASIC

The BASIC programming mode and BASIC editing mode are provided for BASIC language programming.

* See page 49 for details concerning modes.

• BASIC Programming Mode

This mode is used for the creation of new BASIC programs and is entered by pressing the [basic] touch screen key on the MENU screen.

• BASIC Editing Mode

This mode is used to correct or edit previously written BASIC programs, and is entered by pressing the [edit] touch screen key on the MENU screen. BASIC programs can be executed while in either the BASIC programming or BASIC editing mode.

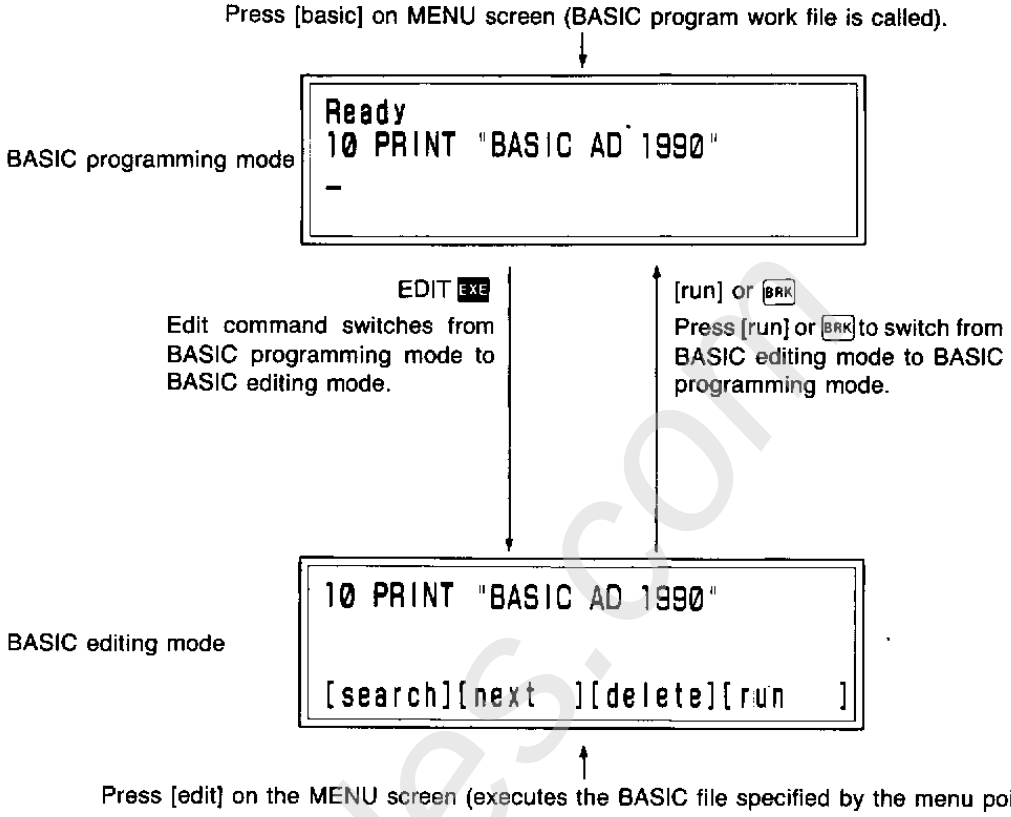
• Files

BASIC programs, as well as the data which are handled during execution of the programs, are stored in memory as files.

* See section 6-4 FILES for further information on files.

1. Program file.....A file which stores a program
2. Data file.....A file which stores data

• BASIC Programming Mode v.s. BASIC Editing Mode



* See section 6-2 FUNDAMENTAL MODE SELECTIONS on page 50.

8-2 BASIC PROGRAM INPUT

Input of a BASIC program will be illustrated by writing a formula which performs the calculation represented by the formula: $y = 2x^2 + 91x + 125$.

```
10 INPUT"X=",X
20 Y=2*X^2+91*X+125
30 PRINT"Y=";Y
40 END
```

(Program) The value of y is calculated for a value input for x and the result is displayed.

As can be seen, a BASIC program is formed from a number of different "lines" each with its own unique line number. The program is then executed starting from the smallest line number through the highest.

Programming Mode Specification

The BASIC programming mode is entered by first pressing the MENU key, followed by the [basic] touch key. At this time, the BASIC work file without a name is prepared and the unit stands by for input.

* See PART 7, section 7-4 for details on [basic].

Program Deletion

The NEW command is used to delete any program that may already be in the BASIC work file, to erase the file for new input.

[N][E][W][EXE]

```
Ready
NEW
Ready
_
```

Program Input

Programs are input line-by-line, with the [EXE] key being pressed when input of each line is complete. The [EXE] key must always be the final step whenever a program line is input or modified, to enter the line or modification into memory. Certain commands that are often used in programming are available for one-key input to make programming quicker and easier. The PRINT command, for example, can be input by pressing [SHIFT] [P].

[1][0][SHIFT][INPUT]["][X]["][,][X][EXE]
[2][0][Y][=][2][*][X][^][2][+][9][1][*][X][+][1][2][5][EXE]
[3][0][SHIFT][PRINT]["][Y][=]["][;][Y][EXE]
[4][0][E][N][D]

Program Modification

Mistakes in a line which has not yet been input (using [EXE]) can be modified by moving the cursor to the location of the modification (using [◄] and [►]). The [INS], [SHIFT] [DEL] and [BS] keys can also be used to add or delete characters. The final step is pressing [EXE] to input the line into memory. The entire line is input regardless of the cursor position (as long as it is within the line to be input) when [EXE] is pressed.
Lines which have already been input (using [EXE]) can be corrected by first moving to the line to be corrected (using [▲] and [▼]) and then making the desired change. When a line has already been scrolled off on the 8-line virtual screen, it can be recalled to the actual display by pressing [SHIFT] [LIST] line number [EXE]. Once again, the final step is pressing [EXE] to input the line into memory, and the entire line is input regardless of the cursor position (as long as it is within the line to be input) when [EXE] is pressed. Programs can also be edited in the EDIT mode.

8-3 PROGRAM EXECUTION

Once the program has been input, it can be executed using the RUN command in the BASIC programming mode.

[R][U][N][EXE]

```
Ready
RUN
```

For the sample program, execution causes the program to ask for an input for the value of x.

```
Ready
RUN
X=
```

A value for x is input using EXE, the result of the calculation is displayed, and the "Ready" prompt appears.

10 EXE

```
X=10
Y=1235
Ready
-
```

- Program execution can also be started from a specified line number using the format noted below.

  [R][U][N] line number EXE

* Programs can also be executed by pressing the [run] touch key while in the BASIC editing mode.

8-4 PROGRAM STORAGE

Programs can be stored in main memory, on floppy disks, or on a cassette tape.

- Main Memory Storage

  [S][A][V][E] "filename" EXE

- Disk Storage

  [S][A][V][E] "0 : filename" EXE

- Cassette Storage

  [S][A][V][E] "CAS0 : filename" EXE

Program storage can be performed either by using the SAVE command or the MENU screen [save] key (see page 67). A newly entered work file without a name is automatically stored in main memory, even if the SAVE command is not executed.

* The "0 : " and "CAS0 : " are put in front of the filename to specify the device to be used for storage.

8-5 PROGRAM LOADING

The LOAD command or the MENU screen [load] key is used to load into memory programs stored in the memory or saved in the external memory.

LOAD Command

As with the SAVE command, programs are loaded by specifying a device name and filename.

- Main memory: LOAD "filename" EXE
- Disk: LOAD "0 : filename" EXE
- Cassette: LOAD "CAS0 : filename" EXE

8-6 VARIABLES

Variable Types

The three following types of variables are available for use with this unit.

1. Numeric variables (up to 13-digit mantissa)		A, a, NUMBER, POINTS
2. String variables (up to 255 characters)		A$, CHR$
3. Array variables	Numeric array variables	A(10), XX(3, 3, 3)
	String array variables	A$(10), ARRAY$(2, 2)

Variable Names

- Variable names can consist of upper, lower case or numeric characters, but a numeric character cannot be used in the first position of the variable name (i.e. 1AE, 3BC$ are illegal).
- Reserved words (see page 133) cannot be used as the leading characters of a variable name (i.e. RUNON, LIST1$ are illegal).
- The maximum length of a variable name is 255 characters.

Counting Bytes Used by Variables

The following outlines the number of bytes reserved when a variable appears the first time within a program.

- Numeric Variables
  (variable name length + 12) bytes in system work area
- String Variables
  (variable name length + 4) bytes in system work area and (string length + 1) bytes in string area

Areas are reserved for array variables when the array is declared by the DIM statement.

- Numeric Array Variables
  (variable name length + 4) + (array size × 8) + (dimension x 2 + 1) bytes in system work area

Example: DIM XYZ(3 , 3 , 5 , 2)
Variable name length = 3
Array size = 4 × 4 × 6 × 3 = 288
(NOTE: 3 = 0, 1, 2, 3, so the size of DIM (3) = 4)
Dimension = 4
Therefore, (3 + 4) + (288 × 8) + (4 × 2 + 1) = 2320 bytes are required for this array.

- String Array Variables
  (variable name length + 4) bytes in the system work area and (array size) + (dimension × 2 + 1) bytes in the string area
  In addition, the number of bytes which correspond to the string length is used in the string area when a string is assigned.

Example: 10 DIM AB$(3, 3)
20 AB$(0, 0) = "＊＊＊＊＊"
Variable name length = 2
Array size = 4 × 4 = 16
Dimension = 2
The array declaration in line 10 uses 6 bytes (2 + 4) in the system work area, and 21 bytes (16 + 2 × 2 + 1) bytes in the string area. The characters assigned to the array in line 20 use 5 bytes in the string area.

8-7 COUNTING BYTES USED IN PROGRAMS

The number of bytes used during program input includes two bytes for the line number, one byte for the space immediately following the line number, two bytes per command, one byte per character not part of a command, plus two additional bytes per line.

Example: **200␣CLS : PRINT␣RIGHT$ (A$, 3) ;**
2 1 2 1 2 1 2 1 1 1 1 1 1 1

These 18 bytes plus two additional bytes means that the total amount of memory space used by this program line is 20 bytes.

* The space following the line number automatically added internally if it is omitted during program input.
* The line number following GOTO and other branching statements requires three bytes.
* ELSE requires three bytes.

8-8 CONVENIENT EDITING FUNCTIONS

Using the [search], [next], and [delete] touch keys in the BASIC editing mode makes program editing much easier.

Example: Input the following program in the BASIC programming mode.

```
10 CLS
20 PRINT "ABC" ;
30 PRINT "DEF" , : PRINT "GHI"
40 END
```

Now perform the following operation to enter the BASIC editing mode.

[E][D][I][T][EXE]
(or [SHIFT][EDIT] [EXE])

```
10 CLS
20 PRINT "ABC";
30 PRINT "DEF",:PRINT "GHI"
[search][next  ][delete][run   ]
```

Now [search] and [next] can be used to instantly locate strings included within the program. In this example, the string "PRINT" will be searched for.

Touch key
[search] [P][R][I][N][T]
(or [SHIFT][PRINT])

```
10 CLS
20 PRINT "ABC";
30 PRINT "DFF",:PRINT "GHI"
search?PRINT_
```

[EXE]

```
20 PRINT "ABC"
30 PRINT "DEF",:PRINT "GHI"
40 END
[search][next  ][delete][run   ]
```

The first occurrence of "PRINT" is located and displayed in reverse field. The pressing [next] locates and displays the next occurrence in reverse field. The unit returns to input stand by when the search is complete.

[delete] can be used to delete single or multiple program lines by specifying the line numbers as with the DELETE statement.

PART 9
OTHER CONVENIENT FUNCTIONS

This part of the manual will cover the clock function of the unit, how to set a program for execution at a certain time on a certain date, and other convenient special features.

9-1 CLOCK FUNCTION

The clock function continues to keep track of the current time and date even when the power of the unit is switched OFF. The time and date are set in the CAL mode, and requires resetting only when batteries are replaced or after the NEW ALL button is pressed.

Date Setting

The date is set by sequentially entering the month, date, and year for the DATE$ function. To set June 20, 1986 for example, input:

DATE$ = "06-20-1986"

[F] [DATE$] [=] ["] [0] [6] [-] [2] [0] [-] [1] [9] [8] [6] ["] [EXE]

```
DATE$="06-20-1986"
-
```

* The current DATE$ setting can be confirmed by [F] [DATE$] [EXE] in the CAL mode.

Time Setting

The date is set in 24-hour format by sequentially entering the hours and minutes for the TIME$ function. To set 7:00 PM, for example, input:

TIME$ = "19:00"

[F] [TIME$] [=] ["] [1] [9] [:] [0] [0] ["] [EXE]

```
TIME$="19:00"
-
```

* The current TIME$ setting can be confirmed by [F] [TIME$] [EXE] in the CAL mode.
* See the COMMAND REFERENCE for details on DATE$ and TIME$.

9-2 POWER ON BOOT

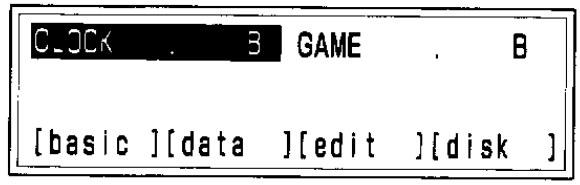
The power on boot makes it possible to specify a program for automatic execution each time the power of the unit is switched ON. The program used for the power on boot must be in a file stored in main memory, and specifying a sequential file causes the unit to enter the DATA editing mode when power is switched ON.

Power ON Boot Set and Cancel

The power on boot is set in the MENU screen.

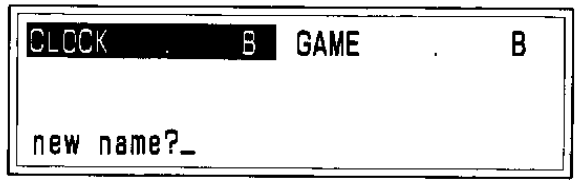
1. Press the MENU key to display the MENU screen.

MENU

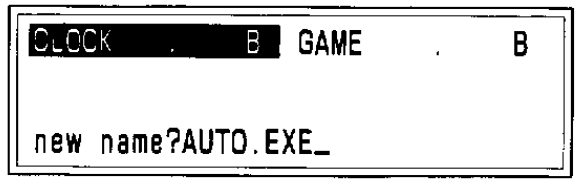


2. Use the cursor keys to specify the file to be executed by the power on boot with the menu pointer.
3. Press LCKEY to advance to the next display, and press [name].
4. Input "AUTO.EXE" for the "new name?_" prompt on the fourth line of the screen.
5. Press EXE .

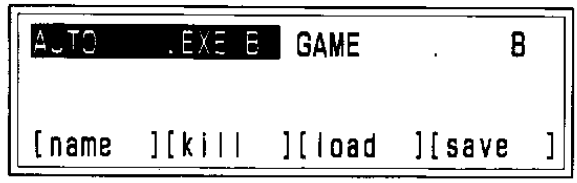
LCKEY [name]



A U T O · E X E



EXE



* The specified program will be executed each time the power of the unit is switched ON.
* Only one AUTO.EXE specification can exist in internal memory, and [name] will not operate to change the name of a second file to AUTO.EXE.
* A different AUTO.EXE file can be specified by first changing the filename of the original AUTO.EXE file to another filename (using [name]).
* The power ON boot is canceled by changing the name of the AUTO.EXE file to another name (using [name]).
* See section 7-4 for details on [name].

9-3 CLOCK BOOT

Clock Boot Function Outline

The clock boot function makes it possible to specify a particular time on a particular date for execution of a BASIC or machine language program, or for display of a sequential file. The clock boot is performed regardless of whether the power of the unit is ON or OFF, and even if a BASIC program is being executed. Only one clock boot specification can be in effect at any one time, and the one-touch files cannot be specified for execution by the clock boot function.

* Be sure to correctly set the current date and time to ensure proper operation of the clock boot function.

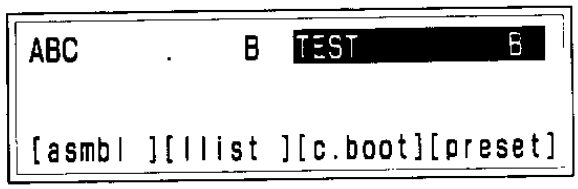
Clock Boot Set

1. Use the cursor keys to move the menu pointer on the MENU screen to the program to be set for clock boot.
2. Press LCKEY twice and then [c.boot].
3. Enter the hour and minute of the clock boot (hour [:] minute EXE).
4. Enter the month and date of the clock boot (month [−] date EXE).

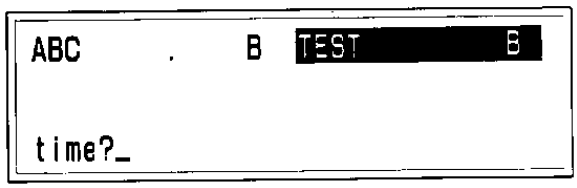
* Note that time is entered in 24-hour format with the hours and minutes separated by a colon, and date is entered with the month and date separated by a minus sign.
* The specified clock boot program is executed everyday at the specified time when the date specification is omitted.

Example: Specify a previously written program under the filename "TEST" for a clock boot at 11:10 on February 1.

1. Press MENU [▷] LCKEY LCKEY .

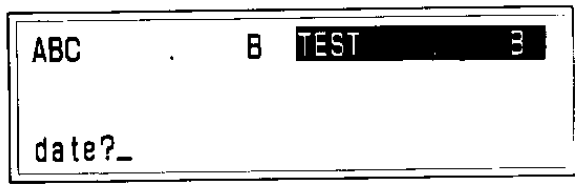


2. Press the [c.boot] touch key.



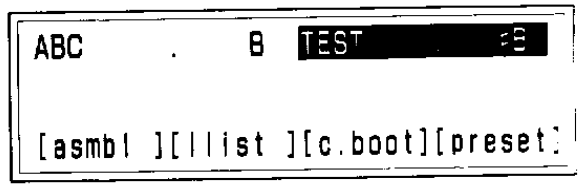
3. Enter the time.

[1][1][:][1][0] EXE



4. Enter the month and date.

[2][−][1] EXE



* Note that # is added in front of the identifier (B) of the file to indicate that the file is being set for clock boot.
* The same procedure as that described above can be used to confirm or change the current time/date settings of the clock boot. Pressing [c.boot] displays the current time setting, and the date setting is then displayed when EXE is pressed.

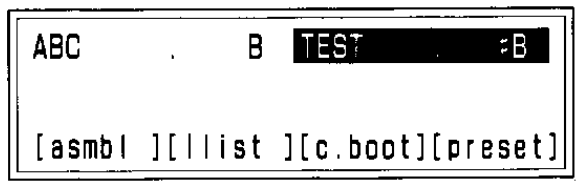
Clock Boot Cancel

The following procedure can be used to cancel a previously specified clock boot.

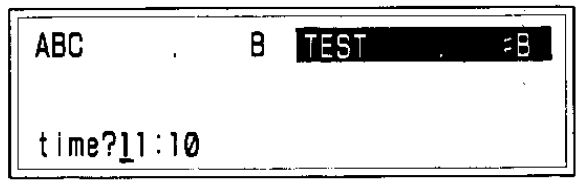
a) Use the cursor keys to move the menu pointer on the MENU screen to the program currently set for clock boot.
b) Press LCKEY twice and then [c.boot].
c) Delete the time setting (using SPC, SHIFT L.DEL, or SHIFT DEL) followed by EXE.

Example: Cancel the clock boot set in the previous example.

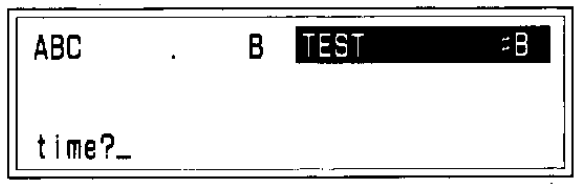
a) Press MENU ▷ LCKEY LCKEY.



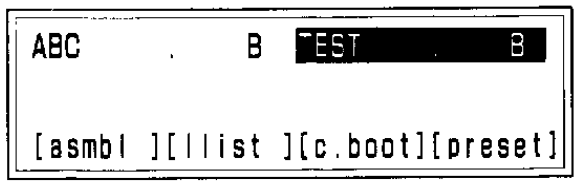
b) Press the [c.boot] touch screen key.



c) Press SHIFT L.DEL (or SHIFT DEL) to delete the time setting.



d) Press EXE.



The # mark in front of the file's identifier (B) is no longer displayed, indicating that clock boot was canceled.

IMPORTANT
Clock boots are not performed any time a device (including memory) is being accessed (SAVE, LOAD, FORMAT). The clock boot is delayed and performed after the access operations are complete.

9-4 ONE-TOUCH FILES

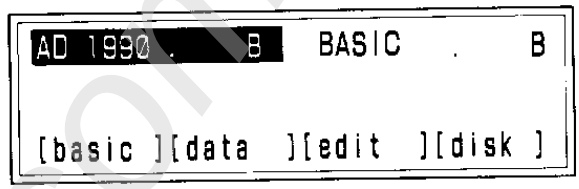
As their name implies, one-touch files allow instant execution of files in the CAL mode by assigning them to touch keys. Up to four files can be assigned as one-touch files, and the DATA editing mode is entered when a sequential one-touch file is executed.

* The [preset] key becomes inoperative once four one-touch files are created.

One-touch File Set and Cancel

One-touch file setting is performed on the MENU screen. In the following example, the file under the filename "BASIC" will be set as a one-touch file.

a) Press MENU to display the MENU screen.

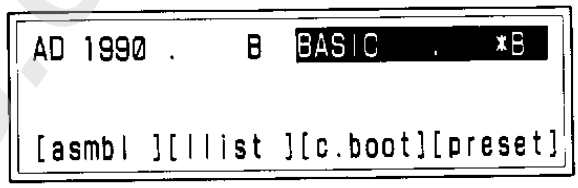


b) Use the cursor keys to move the menu pointer to the filename "BASIC".

c) Press LCKEY twice and then [preset].

▷ LCKEY LCKEY

[preset]

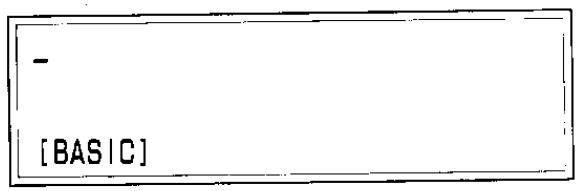


* Note that an asterisk is added in front of the identifier (B) of the file to indicate that the file is specified as a one-touch file.
* The one-touch file specification can be canceled simply by locating the menu pointer at the file to be canceled and pressing [preset] again.
* Files specified as one-touch files can be deleted without canceling the one-touch file specification.
* Canceling a one-touch file specification also eliminates the corresponding touch key on the CAL mode screen.

One-touch File Confirmation and Execution

Press CAL to confirm the one-touch file specification.

CAL



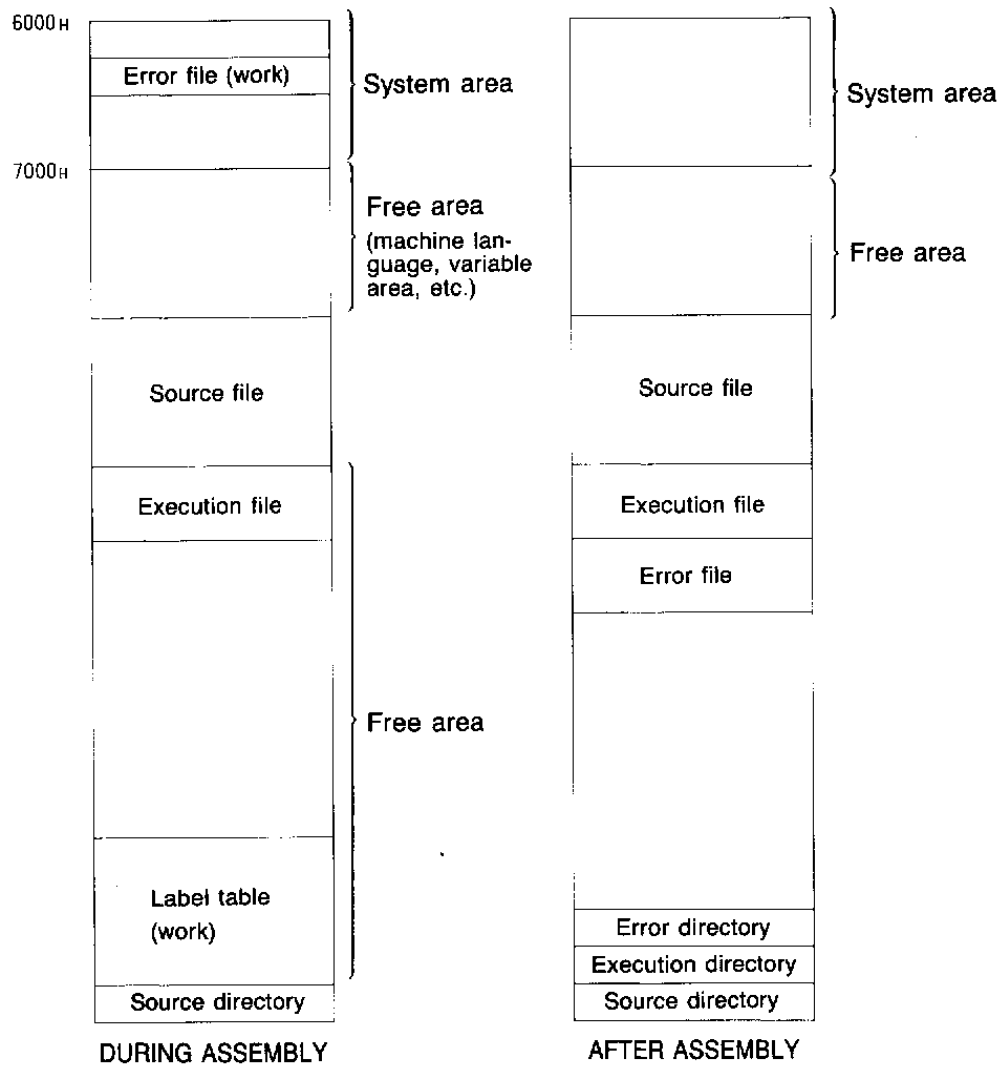
* The specified filenames are displayed as touch screen keys on the fourth line of the CAL mode screen.
* Pressing LCKEY in the CAL mode clears the one-touch file keys from the fourth line of the screen.
  Pressing LCKEY again returns the one-touch keys to the CAL mode screen.
* The one-touch keys are cleared from the screen when a one-touch file program is executed. The one-touch file key display returns when execution is complete, an error is generated, STOP is pressed, or BRK is pressed.
* The first six characters only of the specified program are used for the one-touch file key name.
* A file already specified for clock boot cannot be specified as a one-touch file.

PART 10 ASSEMBLER

10-1 ASSEMBLER CHARACTERISTICS

The assembler function of this unit makes it possible to use mnemonics to develop programs in the machine language of the HD61700 custom LSI. A source program is created by pressing [data] on the MENU screen. The source program is input using mnemonics, and then the assembler function is used to convert the file into a machine language program. An error file is created when an error is found within the source file.

10-2 ASSEMBLER FUNCTION MEMORY MAP



When an error is found during assembler execution an error file in the system area, while a execution file and label table are created in the free area. Once assembly is complete, an error file is created beneath the execution table, unless no errors were found during the execution.
The execution file, label table, and source program are all present in memory during execution, taking up memory space. This may cause generation of an OM (out of memory) error, making assembly of the program impossible.

10-3 ASSEMBLER FORMAT

The lines in the assemble source program are written in the following format.

△ label : △ mnemonic ␣ 1 operand △ , △ 2 operand △ ; comment

△ As many spaces as desired can be used, or spaces can be omitted completely.
␣ indicates that at least one space is required.
, is the operand delimiter symbol.
: following the label is always necessary.
; begins a non-executed comment.

Labels

- The number of labels used is limited only by the amount of memory available.
- Labels must begin with an alphabetic character, and can include letters, underline (SHIFT 8), or @(SHIFT 7).
- Labels can be up to five characters long.
- Labels can also be registered by the pseudo-instructions (EQU).

Mnemonics

A space should be included following a mnemonic. See the COMMAND REFERENCE for details on mnemonics.

Operands

Operands are sometimes required for certain types of mnemonics. Multiple operands should always be delimited by commas.
The following symbols are used in operands.

$0 ~ $31 or $&H0 ~ $&H1F.....main register (32-byte) address
&H.....hexadecimal prefix

Comments

Anything following the semicolon is treated as a comment and so is ignored during assembly of the program.

10-4 PSEUDO-INSTRUCTIONS

The following five types of pseudo-instructions are used by the assembler of this unit.

ORG

FORMAT: ORG address

PURPOSE: Start address specification of an object program
A single program can have multiple ORG instructions. The address specified a subsequent ORG instruction must be greater than the previously specified address. An OR error is generated and assembly becomes impossible when an ORG address is incorrectly specified.

EQU

FORMAT: label : EQU address or numeric value

PURPOSE: Gives a numeric value to the label

DS

FORMAT: label: DS numeric value

PURPOSE: Reserves memory area of the size specified by the numeric value, starting from the address immediately following the address at which this command is located. The label can be omitted.

DB

FORMAT: label: DB numeric value or label: DB "string"

PURPOSE: Generates an object code as byte length data when a numeric value operand is specified.
Outputs an object code as the ASCII code of the string when a string operand is specified.
Multiple values and strings can be specified if they are delimited by commas.
The label can be omitted.

START

FORMAT: START address or label name

PURPOSE: Specifies execution start address of the program.

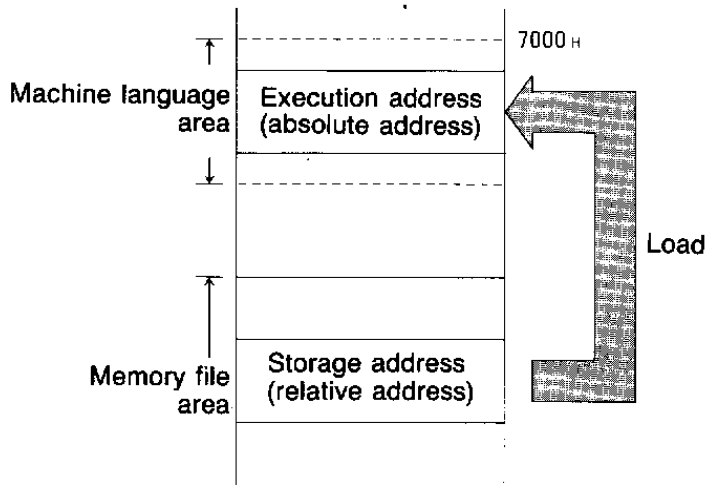
10-5 EXECUTION FILES

The machine language program execution area for this unit is 6FFAH ~ 7FFEH, so source file addressing must be within this area. Addressing outside of this area may result in abnormal execution of the program.
The execution file creates and stores a relative file in the memory file storage area of the unit's main memory. At execution, it is loaded at the absolute address and executed from the START address. Therefore, the contents of the address at which it is loaded are changed when no control is received. An OM error is generated when the CLEAR statement is not used to reserve a machine language area large enough to hold the execution file, and load/execution of the program becomes impossible.
Always use the CLEAR statement to reserve sufficient machine language area before executing a machine language program. The CLEAR statement reserves a machine language area from address 7000H, and 6FFAH through 6FFFH is reserved as a machine language area.

EXAMPLE: CLEAR 1 2 3 4 , 4 0 0 , 2 0 0 0

Machine language area

* Must be greater than the number of bytes in the execution file



10-6 ERRORS

General

Generation of syntax errors in the source program file creates a file with the extension ".ERR" while displaying the number of errors on the screen. The error count on the screen is cumulative, the maximum value displayed is 99, even when 100 or more errors are generated. The error file can include up to 32 errors, but when a program list is produced on the printer, the addresses of lines containing errors are indicated by "?" to assist in error location.

Error Files and Error Codes

The error file is composed of the record numbers where errors where generated, addresses and error codes.

LOO15 : 751A ERR5

- LOO15 → Record number
- 751A → Address
- ERR5 → Error code

Error codes have the following meanings.

ERR1 : Multiple definition of the same label
ERR2 : Assembler system error
ERR3 : Operand or label notation error
ERR4 : Mnemonic notation error
ERR5 : Label on command which cannot take a label

Errors During Assembly

The following errors may be generated during execution of the assembler. Assembly is terminated when either of these errors is generated.

- OM error: Insufficient memory (Out of Memory)
- OR error: ORG specification error

10-7 MACHINE LANGUAGE PROGRAM

The following source program which displays character codes &H20 ~ &HFF will be used for explanation of the machine language program.

```
OUTAC:  EQU    &HFF9E
        ORG    &H7000      ← Execution address of the machine language file
                             (Assembly cannot be performed if not specified).
        START  ABC         ← Execution start address of the machine language file
                             (Assembly cannot be performed if not specified.)
ABC:    LD     $16,&H20
LOOP:   CAL    OUTAC
        AD     $16,1
        JR     NC,LOOP
        RTN                ← Machine language execution is performed by calling
                             the execution start address. The RTN command is al-
                             ways included at the end of the program to return con-
                             trol to the system.
```

Assembler Source File Creation Precautions

- Keep 0 as the value of the high-order address (bank) of the program counter (PC) and system stack pointer (SSP).
- The contents of main registers $30 and $31 are 0 and 1 respectively. These values are used by the system and should not be changed.
- Changing the setting of the closed interrupt may result in abnormal operation.
- Machine language programs are executed by a call from the system, and so should be terminated using the RTN command.

10-8 SOURCE PROGRAM CREATION

Assembler source programs are created in the DATA editing mode.

DATA Editing Mode

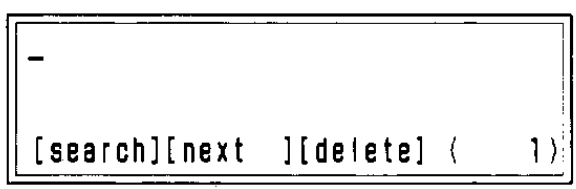
The DATA editing mode is used to edit sequential data files (source program files). There are three different procedures which can be used to enter the DATA editing mode.

1. Press [data] in the MENU mode.
2. Move the menu pointer in the MENU mode to a sequential data file and press EXE.
3. Move the menu pointer in the MENU mode to a sequential data file and press [edit].

Operation

• Sequential Data File (Source Program File) Creation

1. Press [data] in the MENU mode to enter the DATA editing mode.



2. The unit stands by for input when data is not present. Data input is essentially the same as that for the MEMO IN mode.
3. Return to the MENU mode after data input is complete.
4. A sequential work file has been created and is indicated by the menu pointer.
5. The new file can be assigned a name by pressing [name] while the file is specified by the menu pointer.

• Sequential File Editing

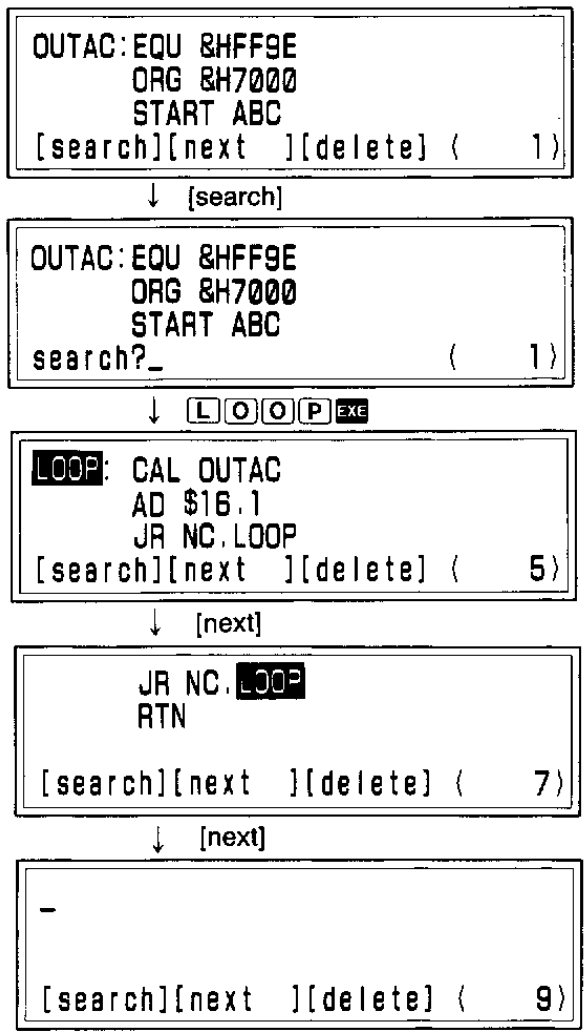
1. Use the cursor keys to move the menu pointer to the sequential file to be edited.
2. Press EXE or [edit] to display the file data.
3. Data input, insert, and corrections are performed using the same procedures outlined for the MEMO IN mode. Pressing BRK causes the unit to stand by for input at the last line of the file.

Search and Delete

The following procedures can be used to locate and delete data from a sequential file.

1. Search

a) Press [search] in the DATA editing mode.
b) Enter the string to be located and press EXE .
c) The beginning of each string is searched for the specified string and displayed in reverse field when located.
d) Pressing [next] continues the search for the next occurrence of the specified string.



2. Delete

a) Press [delete] in the DATA editing mode.
b) Enter the records to be deleted in response to the prompt and press EXE (i.e. [delete] 2 − 5 EXE deletes records 2 through 5).
c) The records following the deleted records are displayed when deletion is complete, or the unit stands by for input when data is not present.

10-9 SOURCE PROGRAM SAVE AND LOAD

Once a source program is created, it can be save to cassette tape when an optional interface unit is connected, or to floppy disk when an optional FDD unit is connected.

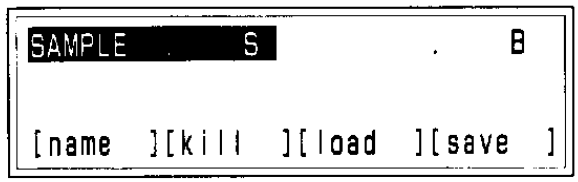
Save

<MT>

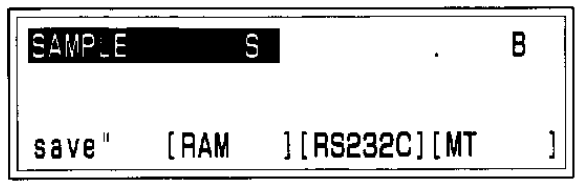
1. Connect a cassette recorder to the interface unit and load a cassette tape.

2. Use the cursor keys to locate the menu pointer at the file to be saved.

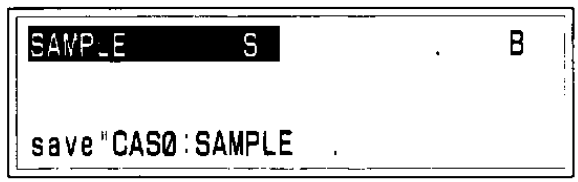
3. Press the REC and PLAY buttons on the recorder, and press [save]. The following prompt will appear on the fourth line of the screen.
save " [RAM] [RS232C] [MT]



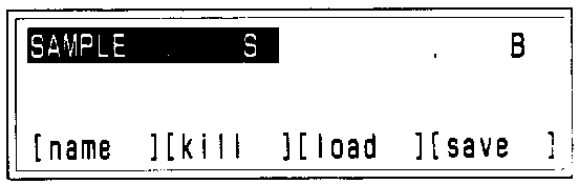
4. Press [MT] (magnetic tape).



5. The following message will appear on the fourth line of the screen.
save "CAS0 : (menu pointer filename)
Make any desired changes (or leave the filename as it is) and then press EXE.



6. The unit returns to the MENU screen when save operations are complete.



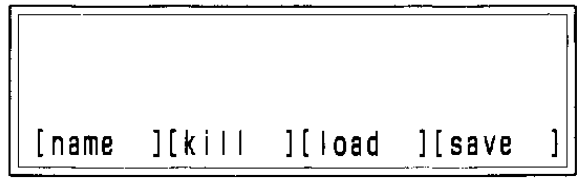
<DISK>

* The procedures for saving programs to disk are identical to those used for cassette tape, except that the prompt on the fourth line of the screen after [save] is pressed appears as follows.
  save " [RAM] [RS232C] [disk]
  In this case, [disk] is pressed.

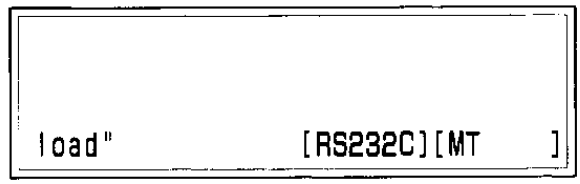
Load

<MT>

1. Connect a cassette recorder to the interface unit, load the cassette tape which contains the file to be loaded, and press the recorder's PLAY button.



2. Press [load] on the MENU screen, and the following prompt will appear on the fourth line of the screen.
   load " [RS232C] [MT]

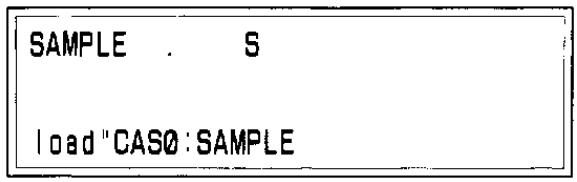


3. Press [MT].

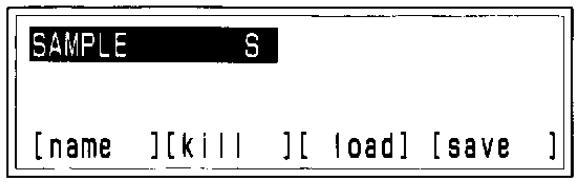
4. Enter the name of the file to be read and press EXE.

   SAMPLE EXE

5. The first three lines of the screen are cleared and the specified filename appears when file loading begins. If other files exist on the tape, their names are also displayed as the programs are skipped to find the specified file.



6. The MENU screen appears with the filename of the newly loaded file indicated by the menu pointer when load operations are complete.



* The MT baud rate (transmission speed) is set using the baud rate switch on the back of the interface unit (see page 118).
* Loading a file from tape at a baud rate which is different from that used when the program was saved generates an error or skips the specified program. Always load programs at the same baud rate at that used to save the program.

<DISK>

* The procedures for loading programs from disk are identical to those used for cassette tape, except that the prompt on the fourth line of the screen after [load] is pressed appears as follows.
  load " [RS232C] [disk]
  In this case, [disk] is pressed.
* The filename cannot be omitted when loading from a disk.

10-10 ASSEMBLY

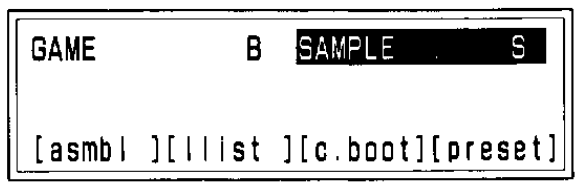
This section contains a step-by-step explanation of assembling a source program contained in main memory.

1. Press the MENU key to display the MENU screen.
2. Use the cursor keys to move the menu pointer to the source program file to be assembled.
3. Use the LCKEY to select the menu which contains the [asmbl] touch key.
4. Press [asmbl] to begin assembly.
5. The following prompt will appear on the screen when an optional interface unit or floppy disk drive unit is connected.
   list ? y/n
   Press [Y] to produce a printout of the assembly list or [N] if printout is not required. At this time "START!" will appear on the screen and assembly will begin.
6. The following display will appear on the fourth line of the screen when assembly is complete.
   END! TOTAL ERROR XX
   The assembler assigns the source program filename to an execution file and an error file. The execution filename is followed by the extension ".EXE", while the error file extension is ".ERR"
   Execution File: SAMPLE.EXE
   Error File: SAMPLE.ERR

* An error file is only created when errors are generated.
* An execution file or error file with the same name as that created during assembly previously existed in memory is erased by the creation of the new file with the same name.
* Execution files and error files for an assembly are deleted when assembly is interrupted by an error or by pressing the BRK key.
* Attempting to execute a machine language program which still contains errors will result in improper operation and may alter important memory contents.
* An execution file can be executed directly from the MENU screen using EXE, or by using a CALL statement in BASIC. The CLEAR statement must also be used, however, to prepare a machine language area for the program (see page 89). In the previously mentioned example, enter CLEAR, 100 EXE.
* Files under filenames containing the ERR or EXE extension cannot be assembled.

Sample Assembly

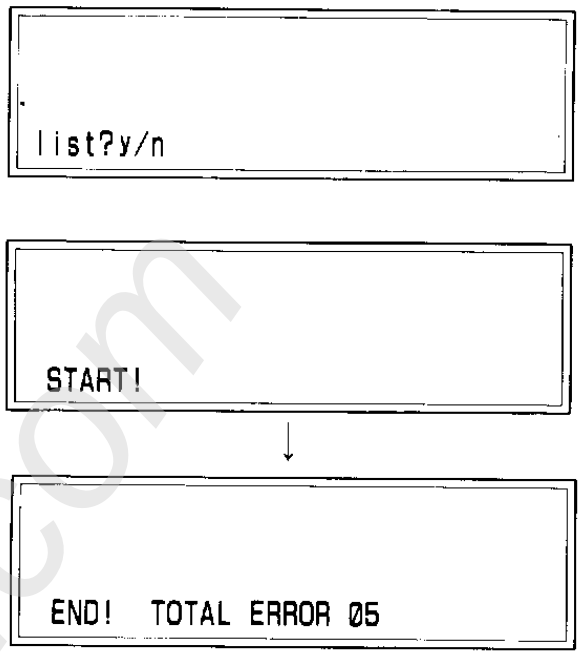
Assembly the source file stored under the filename "SAMPLE".



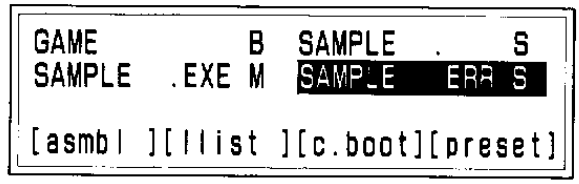
1. Locate the menu pointer at the filename "SAMPLE" and use LCKEY to display [asmbl] on the MENU screen.

2. Press [asmbl].

3. If the optional interface unit or floppy disk drive unit are connected, press [Y] (or [EXE]) to produce a printout, or press [N] is a printout is not required.



4. Press [MENU] or [EXE].



* If no errors have been generated, a "SAMPLE.ERR" file is not created, and MENU screen appears with the menu pointer located at the newly created "SAMPLE.EXE" filename.

10-11 MACHINE LANGUAGE PROGRAM EXECUTION

A machine language program created using the assembler can be executed using any one of the three following methods.

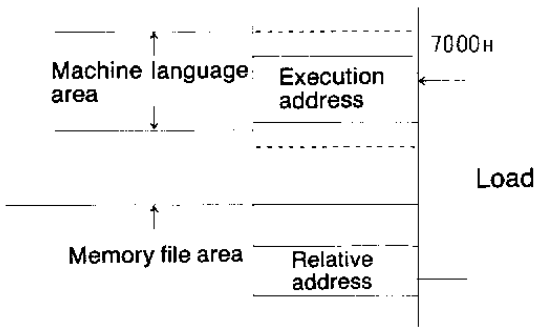
1. Direct execution from the MENU screen
2. [preset] key execution
3. CALL statement execution from BASIC

Whichever method is used, a machine language area larger than the total number of bytes contained in the execution file must first be reserved using the CLEAR statement. Otherwise an OM error will be generated and execution will be impossible.

EXAMPLE: CLEAR 50, 512, 1024 [EXE]

Machine language area: Must be larger than the number of bytes contained in the execution file.

The machine language program is created by the assembler at a relative address, so the program must be loaded at an appropriate address before execution.



Actual execution of the assembled machine language program is performed using on of the three following procedures.

- **MENU Screen**

1. Press the MENU key to enter the MENU mode.
2. Specify the file to be executed with the menu pointer.
3. Press EXE to execute the program. In this case, the program is automatically loaded at the execution address.

- **BASIC**

1. Executing CALL "machine language filename" automatically loads the program at the execution address and begins execution.
2. Executing BLOAD "machine language filename" [, execution start address] [, R] loads the program at the specified address. Then the CALL statement is used to execute the program from the execution start address. Including the [, R] option causes immediate execution of the program (without the CALL statement) as soon as loading is complete.

EXAMPLE: 10 BLOAD "ABC.EXE", &H7000
20 CALL &H7000

- **[preset] Key**

1. Press the MENU key to enter the MENU mode.
2. Specify the file to be executed with the menu pointer.
3. Press [preset] to assign the program as a one-touch file. See 7-4 for details on this function.
4. Press the one-touch file key.

10-12 MONITOR

The monitor function provides valuable assistance in the creation of machine language programs.

Monitor Mode

The MONITOR mode can be entered from either the CAL mode or BASIC mode by executing the MON command. The prompt while in the monitor mode is ">", and the cursor can only be moved within its present line (and are inoperative). The monitor mode can be exited by pressing either the MENU key or CAL key.

Monitor Commands

Three commands are used in the monitor mode to switch banks, dump memory contents, or edit memory contents.

Bank Switch Command (B)

The B command is used to switch the memory bank for monitor operations. Entering B EXE displays the current bank number, and the unit stands by for input of the bank number to be switched to. Either BANK 0 or BANK 1 can be specified using this command, and the initial value for this setting is 1. Pressing EXE without entering a value returns to monitor command stand by without changing the bank number.

FORMAT: B

Sample Execution

EXAMPLE:
The following illustrates a change from BANK 0 to BANK 1.

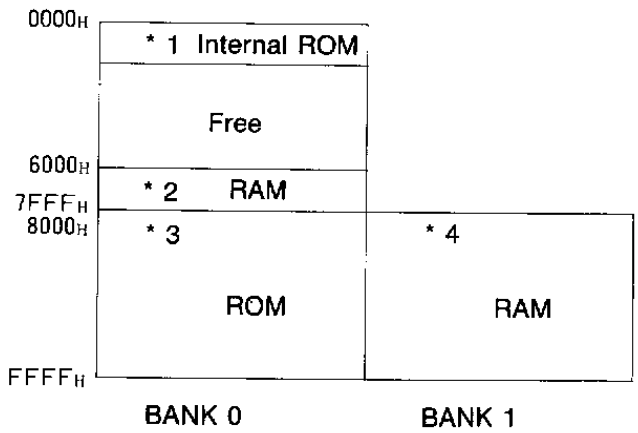
B EXE
1

```
)B
0-1
)_
```

Key input: B EXE bank switch command
1 BANK 1

• Banks

The following memory map illustrates the two banks available in the memory of the unit.



* 1 : HD61700 internal ROM
* 2 : Standard 8KB RAM
* 3 : System ROM
* 4 : Expansion 32KB RAM

BANK 1 is set when the MONITOR mode is entered.

Dump Memory Command (D)

This command is used to dump the memory contents onto the screen.

FORMAT: D [display start address]

The display start address and end address can be omitted from the dump command. If both addresses are omitted, 8 bytes of data starting from the last address displayed are dumped. Specifying only the start address displays 8 bytes of data starting from the start address. A memory dump can be suspended at any point by pressing the [STOP] key, and resumed by pressing any other key on the keyboard.

Sample Execution
Dump the memory contents of address 7000H.

```
)D7000
7000  30 31 30 00 20 31 32 39
-
```

The actual data displayed may differ from that illustrated here because of a difference in memory contents.

Edit Command (E)

This command allows direct alteration of memory contents.

FORMAT: E [start address]

This command displays the contents of the specified start address which can then be altered by key input of hexadecimal data. Omitting the start address displays the contents at the address following the address last specified by this command.
Input for the E command is performed using the following keys.

0 ~ 9, A ~ F, a ~ f : Data input
[SPC] : Advance to next address without changing contents of current address.
[BS] : Return to previous address without changing contents of current address.
[EXE] , [BRK] : Cancel E command

Sample Execution
Change the contents of addresses 7000H through 7007H to a value of 22H.

```
)E7000
7000  30-22 37-22 38-22 39-22
7004  30-22 39-22 36-22 38-22
7008  31-_
```

The actual data displayed may differ from that illustrated here because of a difference in memory contents.

Key input	Meaning	
E7000 EXE	7000H specification	8 bytes
22	Change to 22H	
22	Change to 22H	
22	Change to 22H	
22	Change to 22H	
≀		
22	Change to 22H	
EXE	End	

PART 11
PROGRAM LIBRARY

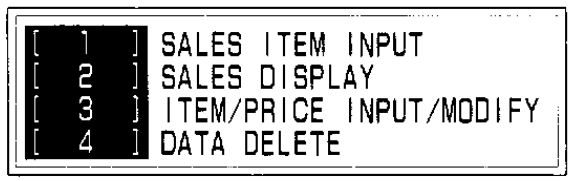
1. Touch Register

This program applies the convenient function of the touch keys to allow the computer to act as a cash register. Sales for up to 10 items are entered directly from the screen, and totals are calculated automatically. For the example here, the program will be set up to handle the sales for the 10 items listed below.

No.	Item	Unit Price	No.	Item	Unit Price
1	Tomato	1.20	6	Radish	0.80
2	Parsley	0.60	7	Carrot	0.30
3	Lettuce	1.50	8	Potato	1.50
4	Celery	1.30	9	Cabbage	1.70
5	G. Pepper	0.70	10	Squash	0.90

Execution

Once the program is executed, the following WORK menu appears on the screen.



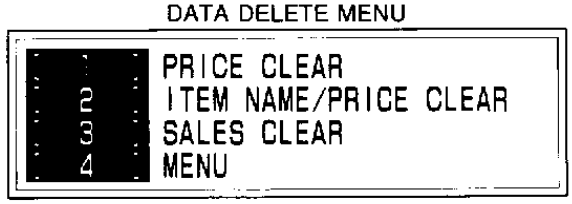
The meaning of the menu items are as follows.
[1] Input of sales data into the register
[2] Display of sales totals
[3] Input and modification of item names and unit prices
[4] Deletion of previous day's or other data no longer required

Simply press the numbered touch key on the scren to select the desired function.

Operation

The first time this program is used, it is a good idea to clear the variables.

Press [4].

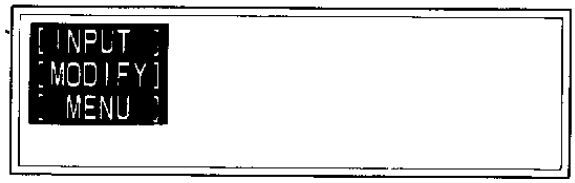


Select [2] on this menu to clear all data and return to the WORK menu.

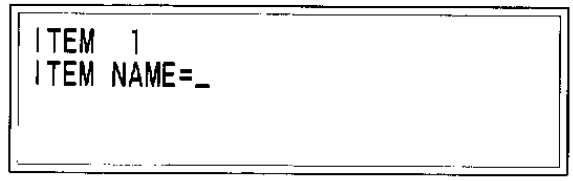
Next, select [3] on the WORK menu to input item names and unit prices.

[3]

INPUT MENU

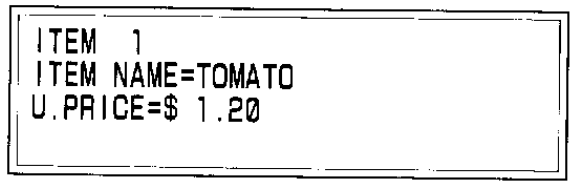


[INPUT]



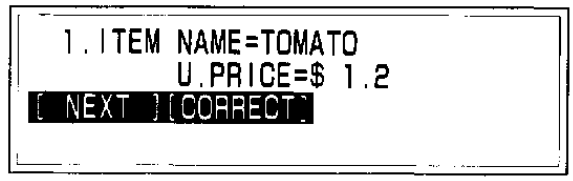
Input the item names and unit prices listed in the table.

T O M A T O EXE
1 . 2 0 EXE



Return to the WORK menu after data for all ten items are input. To correct mistakes or make changes in the input data, press [MODIFY] in the INPUT menu.

[MODIFY]



Each press of [NEXT] advances to the next item, and pressing [CORRECT] allows corrections in the currently displayed data item. Once all data items are input correctly, return to the WORK menu by pressing [MENU] in the INPUT menu.

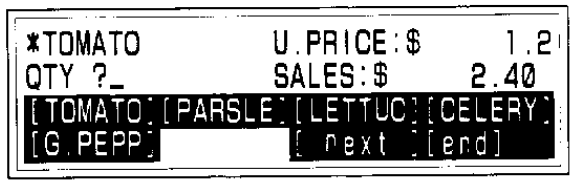
Now press [1] in the WORK menu for input of sales data.

[1]

Entering sales amounts is also performed using touch keys. For an input of two tomatoes, for example, press the [TOMATO] key and enter the number of tomatoes sold.

[TOMATO]

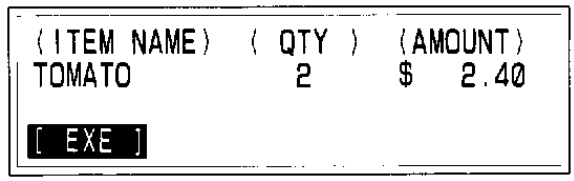
[2] [EXE]

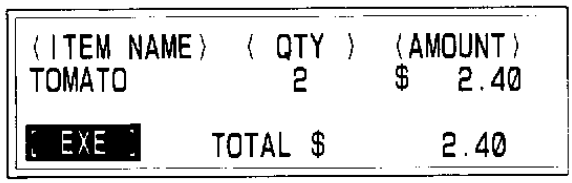


Only five items are shown on the screen at one time, and [next] is used to advance to the next screen for the other five items. [end] is pressed after sales input is complete to display the sales just input. Each press of [EXE] displays the sales data for an item, and the final display shows the total amount sold. Pressing [EXE] at this time returns to the WORK menu.

[EXE]

[EXE]

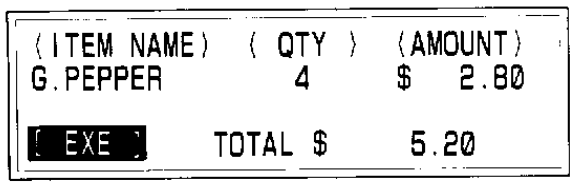




The register is closed (sales amount up to that point determined) by selecting [2] on the WORK menu.

[2] [EXE]

[EXE]



Each press of [EXE] displays the sales data for an item, and the final display shows the total amount sold. Pressing [EXE] at this time returns to the WORK menu.

Data are retained even when the power of the unit is switched OFF, and program execution is terminated by pressing the BRK key. Remember to clear all previous sales data before using the program again for input of new data.

PROGRAM LIST

```
10 DIM NA$(9),NO(9),SEL(9),SUM(9)
20 CLS :PRINT REV;"[  1  ]";NORM;" SALES ITEM INPUT"
30 PRINT REV;"[  2  ]";NORM;" SALES DISPLAY"
40 PRINT REV;"[  3  ]";NORM;" ITEM/PRICE INPUT/MODIFY"
50 PRINT REV;"[  4  ]";NORM;" DATA DELETE";
60 IN=ASC(INKEY$):IN=IN/4-59
70 IF IN<1 OR 4<IN OR FRAC(IN)<>0 THEN 60
80 ON IN GOTO ,500,1000,2000
100 CLS
110 SW=0:FOR I=0 TO 9:SEL(I)=0:NEXT I
120 LOCATE 0,2:PRINT REV;"[      ][      ][      ][      ]
    [      ]";NORM;"        ";REV;"[ next ][end]";
130 FOR I=0 TO 4
140 LOCATE I MOD 4*8+1,2+INT(I/4)
150 PRINT USING"&    &";NA$(I+5*SW);
160 NEXT I:LOCATE 0,0:PRINT NORM
170 IN=ASC(INKEY$)
180 IF IN<248 OR IN=253 THEN 170
190 IF IN=254 THEN SW=(SW+1) MOD 2:PRINT REV:GOTO 130
200 IF IN=255 THEN 300
210 LOCATE 0,0:PRINT TAB(64)
220 PA=IN-248+5*SW
230 LOCATE 0,0:PRINT USING"*&          &";NA$(PA)
240 LOCATE 15,0:PRINT USING"U.PRICE:$##,###.##";NO(PA)
250 LOCATE 0,1:INPUT "QTY ?",NO$:LOCATE 0,0:QTY=VAL(NO$)
260 SEL(PA)=NO(PA)*QTY:BEEP
270 LOCATE 15,1:PRINT USING"SALES:$##,###.##";SEL(PA)
280 GOTO 170
300 GOSUB 600:S=0
310 FOR I=0 TO 9
320 IF SEL(I)=0 THEN 350
330 A=SEL(I):GOSUB 700
340 S=S+SEL(I):SUM(I)=SUM(I)+SEL(I)
350 NEXT I
360 LOCATE 1,3:PRINT REV;"[ EXE ]";NORM;:LOCATE 12,3:PRINT
    USING"AMOUNT $##,###.##";S;
370 IN=ASC(INKEY$)
380 IF IN=13 OR IN=252 THEN 20 ELSE 370
500 GOSUB 600:S=0
510 FOR I=0 TO 9
520 IF SUM(I)=0 THEN 550
530 A=SUM(I):GOSUB 700
540 S=S+SUM(I)
550 NEXT I
560 LOCATE 12,3:PRINT USING"AMOUNT$##,###.##";S;
570 IN=ASC(INKEY$)
580 IF IN=13 OR IN=252 THEN 20 ELSE 570
600 CLS :LOCATE 1,0:PRINT "<ITEM NAME>  < QTY >   < AMOUNT >"
```

```
610 RETURN .
700 LOCATE 1,1:PRINT USING"&          &#####  $###,###.##";
    NA$(I);A/NO(I);A
710 LOCATE 1,3:PRINT REV;"[ EXE ]";NORM;
720 IN=ASC(INKEY$)
730 IF IN=13 OR IN=252 THEN RETURN ELSE 720
1000 CLS
1010 PRINT REV;"[INPUT ]","[MODIFY]","[ MENU ]";NORM
1020 IN=ASC(INKEY$)
1030 IF IN=248 THEN 20
1040 IF IN=244 THEN 1200
1050 IF IN<>240 THEN 1020
1060 FOR I=0 TO 9
1070 GOSUB 1300
1080 NEXT I
1090 GOTO 1000
1200 FOR I=0 TO 9
1210 CLS :PRINT USING"###.";I+1;
1220 PRINT USING"ITEM NAME=&          &";NA$(I):LOCATE 9,1:
     PRINT "U.PRICE=$";NO(I)
1230 LOCATE 0,2:PRINT REV;"[ NEXT ][CORRECT]";NORM
1240 IN=ASC(INKEY$)
1250 IF IN=248 THEN 1280
1260 IF IN<>249 THEN 1240
1270 GOSUB 1300
1280 NEXT I
1290 GOTO 1000
1300 CLS :PRINT "ITEM ";I+1
1310 INPUT "ITEM NAME=",N$:IF N$<>"" THEN NA$(I)=N$
1320 INPUT "U.PRICE=$",NO(I)
1330 RETURN
2000 CLS
2010 PRINT REV;"[  1  ]";NORM;" PRICE CLEAR"
2020 PRINT REV;"[  2  ]";NORM;" ITEM NAME/PRICE CLEAR"
2030 PRINT REV;"[  3  ]";NORM;" SALES CLEAR"
2040 PRINT REV;"[  4  ]";NORM;" MENU";
2050 IN=ASC(INKEY$):IF IN=252 THEN 20
2060 IF IN<>240 AND IN<>244 AND IN<>248 THEN 2050 ELSE CLS
2070 FOR I=0 TO 9
2080 IF IN=240 THEN NO(I)=0:GOTO 2110
2090 IF IN=244 THEN NA$(I)="":NO(I)=0
2100 SUM(I)=0
2110 NEXT I
2120 GOTO 20
```

2. High Speed Sort Program

Sorting random data into ascending or descending order usually takes quite a bit of time, except when a high speed sort program is used. This program combines and BASIC program and machine language program, with the BASIC portion handling input and display, while the machine language portion doing the actual high speed sort. Up to 200 data items can be input with data input being terminated by input of a negative number.

a) First reserve a machine language area in memory.
In the CAL mode, enter:
[SHIFT] [CLEAR] , 1 0 0 [EXE]

```
CLEAR,100
-
```

b) Enter the mnemonic source program in the DATA editing mode under the filename "SORT" (see pages 53 and 55 for details on the DATA editing mode and assigning filenames).

c) Assemble the SORT file by aligning the menu pointer and pressing [asmbl].

```
SORT          S
[asmbl ][list ][c.boot][preset]
```

The display illustrated here appears on the screen when an optional interface unit or FDD unit is attached. Press [Y] to produce a printout of the list, and [N] when a list is not required.

```
SORT          S
list? y/n
```

d) Assembly starts at this time.

```
START!
```

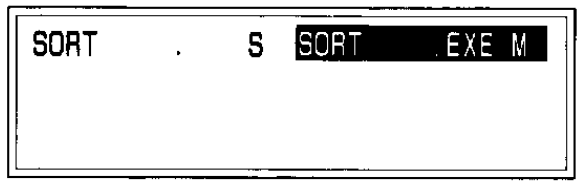
e) Assembly complete.
TOTAL ERROR 00 indicates that no errors were generated during the assembly.

```
END!  TOTAL ERROR 00
```

NOTE: An display as that illustrated here indicates that errors were generated in three lines of the program. Should this occur, check the error file (in this example SORT.ERR) to determine the lines in which the errors were generated, and make the necessary corrections. This procedure must be repeated and the program reassembled until the TOTAL ERR 00 message is displayed.

```
END!  TOTAL ERROR 03
```

f) An execution file (SORT.EXE in this example) is created when the source program is successfully converted into a machine language program. Press MENU to confirm the presence of the execution file filename on the MENU screen.



g) Input the BASIC program included in list 2, ensuring that input is complete and correct. Assign the filename "SORTB" to the BASIC program file.

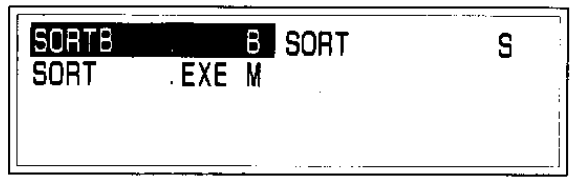
h) Enter the following series of 10 values to be sorted.

36, 30, 182, 152, 29, 172, 196, 81, 180, 118

i) Now execute the BASIC program.

MENU

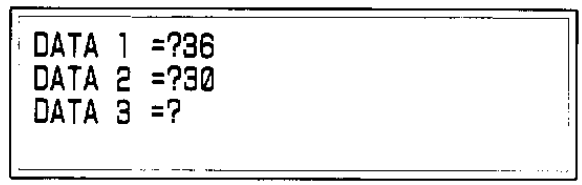
Use the cursor keys to move the menu pointer to the SORTB filename and press EXE.



j) Enter the values.

36 EXE

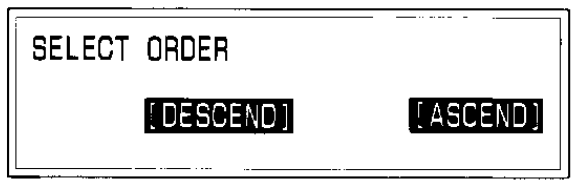
30 EXE



After input of all of the desired values is complete, input −1 to end input procedures.

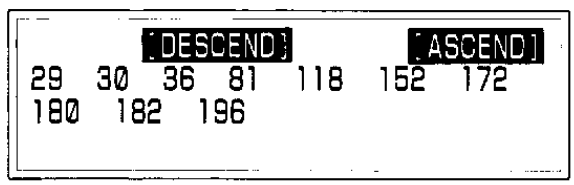
−1 EXE

k) Select whether the values should be sorted into ascending order or descending order.



Here, ascending order will be selected, so press:

[ASCEND]



The sorted list of values is produced almost immediately.

Sorted lists of up to 200 values can be sorted in a matter of a few seconds.

PROGRAM LIST 1

```
        ORG   &H7000
        START &H7000
        LDW   $1,&H70FF
        LD    $0,($1)
        SB    $0,00
        JR    Z,PRO
        LDW   $1,&H7100
        LD    $0,($1)
        PRE   IX,&H7100
        LD    $1,1
LOOP:   LD    $2,$1
        AD    $2,1
LOOP2:  LD    $3,(IX+$1)
        LD    $4,(IX+$2)
        SBC   $3,$4
        JR    C,NON
        LD    $5,$3
        LD    $3,$4
        LD    $4,$5
        ST    $3,(IX+$1)
        ST    $4,(IX+$2)
NON:    AD    $2,1
        SBC   $0,$2
        JR    NC,LOOP2
        AD    $1,1
        SBC   $0,$1
        JR    NZ,LOOP
        RTN
PRO:    LDW   $1,&H7100
        LD    $0,($1)
        PRE   IX,&H7100
        LD    $1,1
LOOP3:  LD    $2,$1
        AD    $2,1
LOOP4:  LD    $3,(IX+$1)
        LD    $4,(IX+$2)
        SBC   $3,$4
        JR    NC,NON1
        LD    $5,$3
        LD    $3,$4
        LD    $4,$5
        ST    $3,(IX+$1)
        ST    $4,(IX+$2)
NON1:   AD    $2,1
        SBC   $0,$2
        JR    NC,LOOP4
        AD    $1,1
        SBC   $0,$1
        JR    NZ,LOOP3
        RTN
```

PROGRAM LIST 2

```
10 'INITIALIZATION
20 N=0
30 'DATA READ
40 PRINT "DATA";N+1;"=";:INPUT D
50 IF D>200 THEN PRINT "OVER FLOW:";:GOTO 40
60 IF D<0 THEN 110
70 N=N+1:POKE &H7100,N
80 POKE &H7100+N,D
90 IF N>200 THEN 110
100 GOTO 40
110 'SORT SPECIFICATION
120 CLS:PRINT "SELECT ORDER"
130 LOCATE 7,2:PRINT REV;"[DESCEND]
140 LOCATE 23,2:PRINT "[ASCEND]";NORM;
150 IN=ASC(INKEY$)
160 IF IN=249 THEN H=0 ELSE IF IN=251 THEN H=1 ELSE 150
170 POKE &H70FF,H
180 'SORT ROUTINE
190 BLOAD"SORT.EXE"
200 CALL &H7000
210 'DATA DISPLAY
220 FOR I=1 TO N
230 PRINT PEEK(&H7100+I);
240 NEXT I
250 END
```

3. RENUMBER

This program rearranges the line numbers of a BASIC program so the program is numbered at 10 line increments starting from line 10. Jump destinations (in such statements as GOTO) are also adjusted to match the renumbered program. This program helps to keep BASIC programs easy to follow for quicker debugging procedures. Since a machine language program is used, very little time is required for the renumbering procedure. See PART 10 of this manual for details on machine language.

a) First reserve a machine language area.

CAL CLEAR , 200 EXE

```
CLEAR,200
-
```

b) Enter the data editing mode for source input.

MENU [data]

```
-
[search][next ][delete] (    1)
```

c) Input the program starting with ORG␣ &H7000.

ORG␣&H7000

```
ORG &H7000_
[search][next ][delete] (    1)
```

EXE

```
ORG &H7000
[seahch][next ][delete] (    1)
```

d) Assign a name to the program once input is complete. In this example, the name "RENUM" is used.

MENU LCKEY [name] RENUM EXE

```
RENUM        S
[name ][kill ][load ][save ]
```

e) Next, assemble the source program.

LCKEY [asmbl]

```
START!
```

f) Assembly is completed successfully when the total error display reads 00.

```
END!  TOTAL ERROR 00
```

* Machine language programs will not run properly even if a single error is present. Correct and assemble the source program until the total error display shows 00.

g) Confirms that an execution file (RENUM.EXE) has been created for the assembled machine language program.
MENU

```
RENUM    .    S  RENUM    .EXE M

[basic ][data  ][edit  ][disk  ]
```

h) Input the following BASIC program (which displays the character code and character for a key input) and execute RENUM to renumber the program lines.

```
1 REM KEY TEST
2 CLS
3 A$=INKEY$:IF A$="" THEN 3
4 B=ASC(A$)
5 PRINT B;CHR$(B)
7 GOTO 3
```

i) Assign a filename to the BASIC program (KEYTEST) once input is complete.

MENU LCKEY [name] KEYTEST EXE

```
KEYTEST   .    B  RENUM    .    S
RENUM    .EXE M

[name  ][kill  ][load  ][save  ]
```

j) Move the menu pointer to the file to be renumbered (KEYTEST) and enter the BASIC editing mode.

MENU [edit]

```
1 REM KEY TEST
2 CLS
3 A$=INKEY$:IF A$"" THEN 3
[search][next  ][delete][run   ]
```

k) Perform the following input after pressing the BRK key.

CALL "RENUM.EXE" EXE

```
Ready
CALL"RENUM.EXE"
_
```

Completion of the RENUM program execution will be indicated by the the cursor returning to the screen.
Use the LIST or EDIT command to confirm that the BASIC program was properly renumbered.

```
10 REM KEY TEST
20 CLS
30 A$=INKEY$:IF A$="" THEN 30
40 B=ASC(A$)
50 PRINT B;CHR$(B)
60 GOTO 30
```

As seen here, the process to renumber a BASIC program is:
1. Move the menu pointer to filename of the program to be renumbered.
2. Press [edit].
3. Press BRK.
4. CALL "RENUM.EXE"
5. Press EXE.

Until this program is assembled into an execution file, 1542 bytes are required for the source program and 223 bytes for the execution file, for a total of 1765 bytes. Once the source program is assembled, however, only the execution file is required in memory.

PROGRAM LIST

```
0001:7000                       ORG   &H7000
0002:7000                       START &H7000
0003:7000                       ;
0004:7000                       FCERR: EQU   &H8BE9
0005:7000                       ;
0006:7000  D640516F             RENUM: PRE   IZ,&H6F51
0007:7004  690000               LD    $0,(IZ+0)
0008:7007  410001               SBC   $0,1
0009:700A  34E98B               JP    NZ,FCERR
0010:700D  D640546F             PRE   IZ,&H6F54
0011:7011  A9611F               LDW   $1,(IZ+$31)
0012:7014  9601                 PRE   IX,$1
0013:7016  A8791E               LDW   $25,(IX+$30)
0014:7019  896F0F               SBW   $15,$15
0015:701C  777970               CAL   LNSCH
0016:701F  D1009A19             LDW   $0,6554
0017:7023  816200               SBCW  $2,$0
0018:7026  31E98B               JP    NC,FCERR
0019:7029  9659                 PRE   IZ,$25
0020:702B  7B1F00               RENM1: SBC   (IZ+0),$31
0021:702E  B01C                 JR    Z,RENM3
0022:7030  6B0002               LDI   $0,(IZ+2)
0023:7033  6B0000               RENM2: LDI   $0,(IZ+0)
0024:7036  410000               SBC   $0,0
0025:7039  B08F                 JR    Z,RENM1
0026:703B  410003               SBC   $0,3
0027:703E  B48C                 JR    NZ,RENM2
0028:7040  AB6F1F               LDIW  $15,(IZ+$31)
0029:7043  777970               CAL   LNSCH
0030:7046  B594                 JR    C,RENM2
0031:7048  37E98B               JP    FCERR
0032:704B  9659                 RENM3: PRE   IZ,$25
```

```
0033:704D  7B1F00    RENM4: SBC   (IZ+0),$31
0034:7050  B046             JR    Z,LNNEW
0035:7052  6B0002           LDI   $0,(IZ+2)
0036:7055  6B0000    RENM5: LDI   $0,(IZ+0)
0037:7058  410000           SBC   $0,0
0038:705B  B08F             JR    Z,RENM4
0039:705D  410003           SBC   $0,3
0040:7060  B48C             JR    NZ,RENM5
0041:7062  A96F1F           LDW   $15,(IZ+$31)
0042:7065  777970           CAL   LNSCH
0043:7068  9862             BIUW  $2
0044:706A  826002           LDW   $0,$2
0045:706D  9862             BIUW  $2
0046:706F  9862             BIUW  $2
0047:7071  886200           ADW   $2,$0
0048:7074  A3621F           STIW  $2,(IZ+$31)
0049:7077  B7A3             JR    RENM5
0050:7079                   ;
0051:7079  9619      LNSCH: PRE   IX,$25
0052:707B  D1020100         LDW   $2,1
0053:707F  7A1F00    LNSC1: SBC   (IX+0),$31
0054:7082  F0               RTN   Z
0055:7083  BA6F1E           SBCW  (IX+$30),$15
0056:7086  B00C             JR    Z,SEC
0057:7088  680000           LD    $0,(IX+0)
0058:708B  2A6000           LDI   $0,(IX+$0)
0059:708E  88621E           ADW   $2,$30
0060:7091  B793             JR    LNSC1
0061:7093  017F1E    SEC:   SBC   $31,$30
0062:7096  F7               RTN
0063:7097                   ;
0064:7097  9619      LNNEW: PRE   IX,$25
0065:7099  D1000A00         LDW   $0,10
0066:709D  826200           LDW   $2,$0
0067:70A0  7A1F00    LNNE1: SBC   (IX+0),$31
0068:70A3  F0               RTN   Z
0069:70A4  A0621E           STW   $2,(IX+$30)
0070:70A7  680400           LD    $4,(IX+0)
0071:70AA  2A6404           LDI   $4,(IX+$4)
0072:70AD  886200           ADW   $2,$0
0073:70B0  B791             JR    LNNE1
```

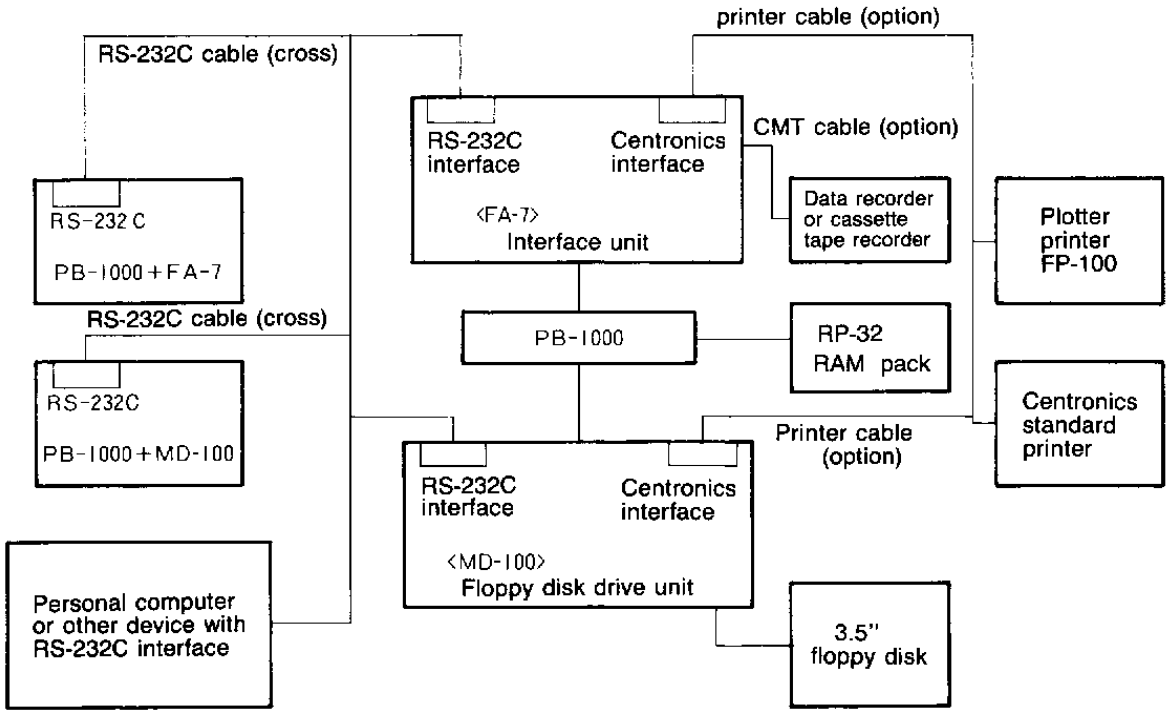
* The values to the left of the list show memory addresses and values which can be confirmed and corrected in the monitor mode if program execution is abnormal.

* Input the source program enclosed in the dotted line.

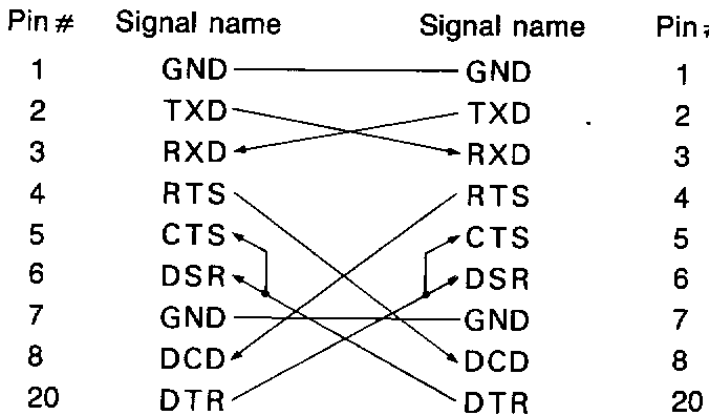
PART 12
PERIPHERAL DEVICE EXPANSION

Besides an optional cassette interface, this unit can also be used in combination with an optional 3.5" floppy disk drive unit. The RS-232C interfaces built into these units also provide capabilities for wide variety of data communications applications. This part of the manual outlines the peripheral devices that can be connected to this computer.

System Configuration



- **RS-232C cable (cross wiring diagram)**

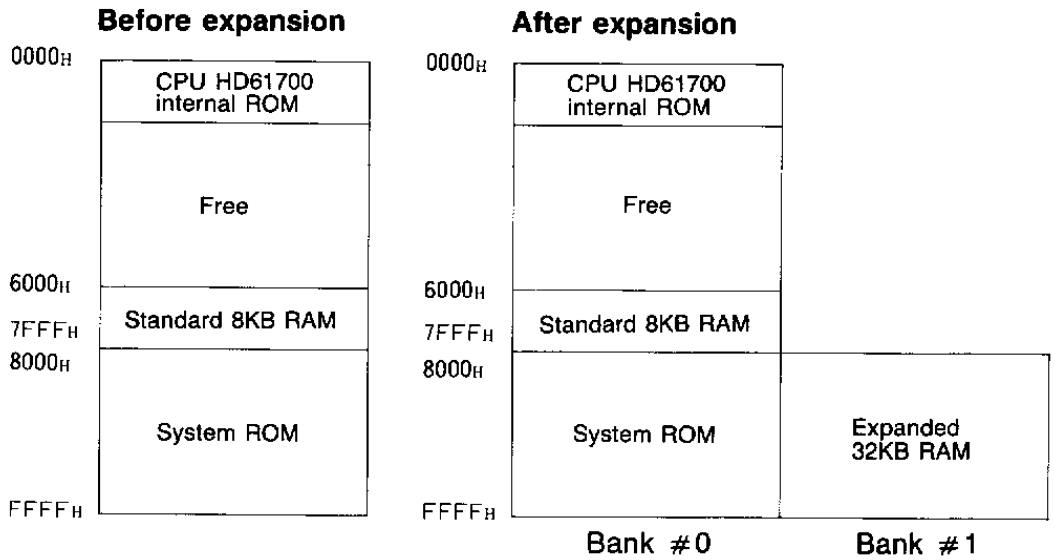


NOTE: Cross wiring is required when connecting to another identical computer or to a personal computer via an expansion unit.

12-1 RAM EXPANSION PACK

The 8KB of RAM (4KB user area) can be expanded up to 40KB (36KB user area) using an optional RAM expansion packs.

Expansion Memory Map



Free area	TEXT	3072 bytes	(FREE)	Free area	TEXT	27648 bytes	
	String variables	256 bytes	($)		String variables	1024 bytes	
	Numeric variables	768 bytes	(V)		Numeric variables	8191 bytes	

RAM Pack Loading Procedure

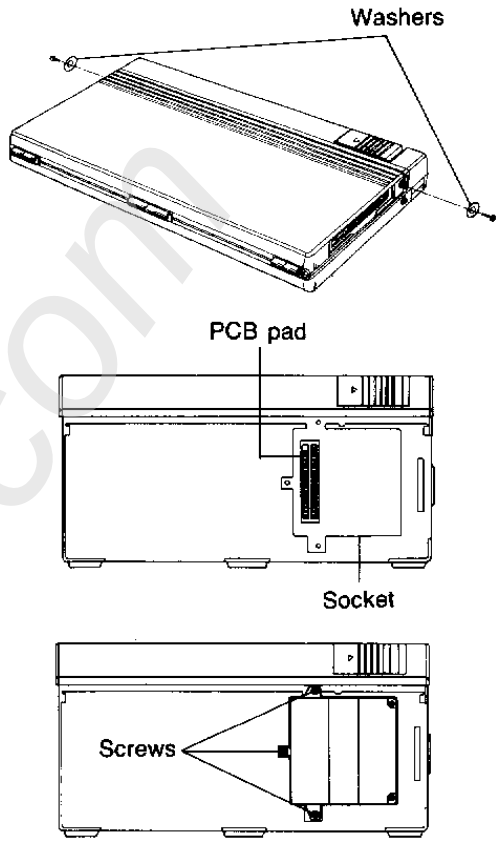
Static electricity may potentially damage the internal circuitry of RAM packs. Therefore, be sure to touch a doorknob or other metal fixture to discharge any static electricity present before handling RAM pack.

1. Switch the power of the unit OFF.
2. Remove the two screws on the sides of the keyboard panel of the unit and remove the back cover.
3. Load a RAM pack into the socket inside of the unit and fix the RAM pack in place using the three screws provided.

* Do not touch the connectors of the RAM pack or the PCB pad.

4. Replace the back cover and secure it in place using the two screws.
5. Switch the power of the unit ON and press the NEW ALL button with a thin, pointed object.

If NEW ALL does not operate properly, press NEW ALL again while holding down the RESET button. See the precautions concerning this operation on page 8.



* Loading or removing a RAM pack without pressing the NEW ALL button may result in altered memory contents or abnormal display.
* Dirt, dust, or fingerprints on the RAM pack connectors or PCB pad may make proper connection impossible.
* Store RAM packs not loaded in the unit in their original boxes to protect them from dust and dirt.

12-2 INTERFACE UNIT

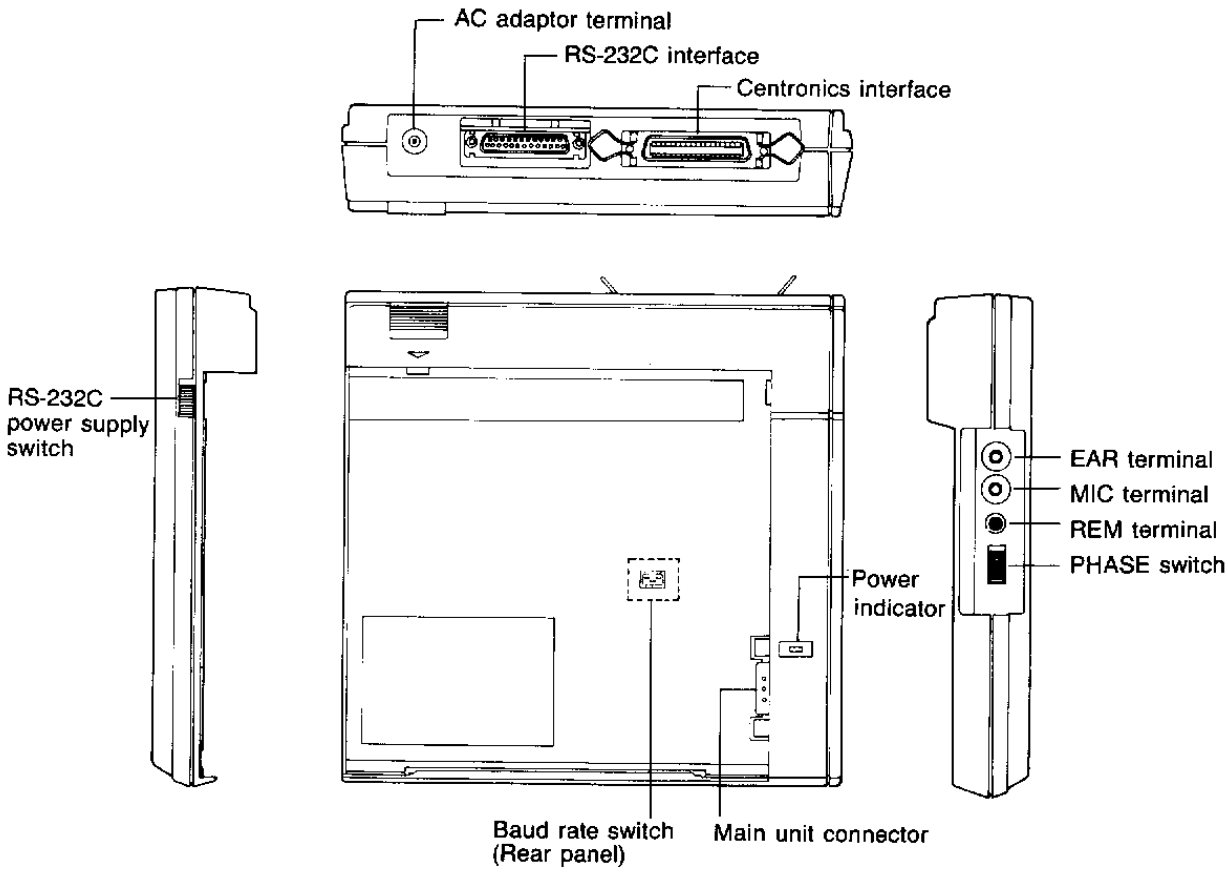
Features

The interface unit makes it possible to connect a cassette tape recorder for use as an external storage device. An RS-232C interface and Centronics standard printer interface are also built-in.

Configuration

- An RS-232C interface, a Centronics standard printer interface connector, and an AC adaptor jack are all located on the back of the unit.
- An RS-232C power switch to control the power supply to the RS-232C interface is located on the left of the unit.

- EAR, MIC and REM terminals for the cassette interface, and a PHASE switch for reversing the polarity of LOAD operations are located on the right side of the unit.
- A power indicator (LED) on the right side of the top of the unit lights (green) when power supply is normal. Batteries should be replaced as soon as possible when the power indicator fails to light.
- Baud rate switches and battery compartment are located on the bottom of the unit.
- To replace batteries, remove the battery compartment cover, remove the old batteries, and replace with a full new set ensuring that polarity (+/−) is correct.

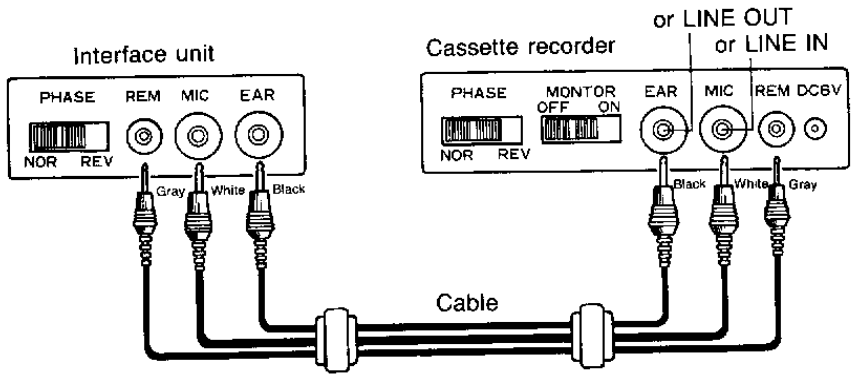


Connection

Always be sure to switch the power of the computer OFF before connecting it to the interface unit. Once connected, the power switch of the computer also controls the power supply of the interface unit.

Cassette Interface

The cassette interface is used to save data/programs to and load data/programs from the computer to a cassette tape. The cassette recorder is connected to the interface unit using the accessory cables as noted below.



* The recorder must be preset for recording before the SAVE command is executed.
* The recorder must be preset for playback before the LOAD command is executed.
* LOAD may be impossible with some types of recorders. In this case, change the position of the PHASE switch and try loading again.
* The baud rate is set using the baud rate switches on the back of the interface unit. Though baud rates of 300, 600, 1200, and 2400 are available, operations with some recorders will only be possible at slower speeds. As with the RS-232C port, setting the baud rate outside the range of 300 ~ 2400 will result in an AM error.

Other Interfaces

See the following section for details on the RS-232C and Centronics interfaces.

12-3 FLOPPY DISK DRIVE

Features

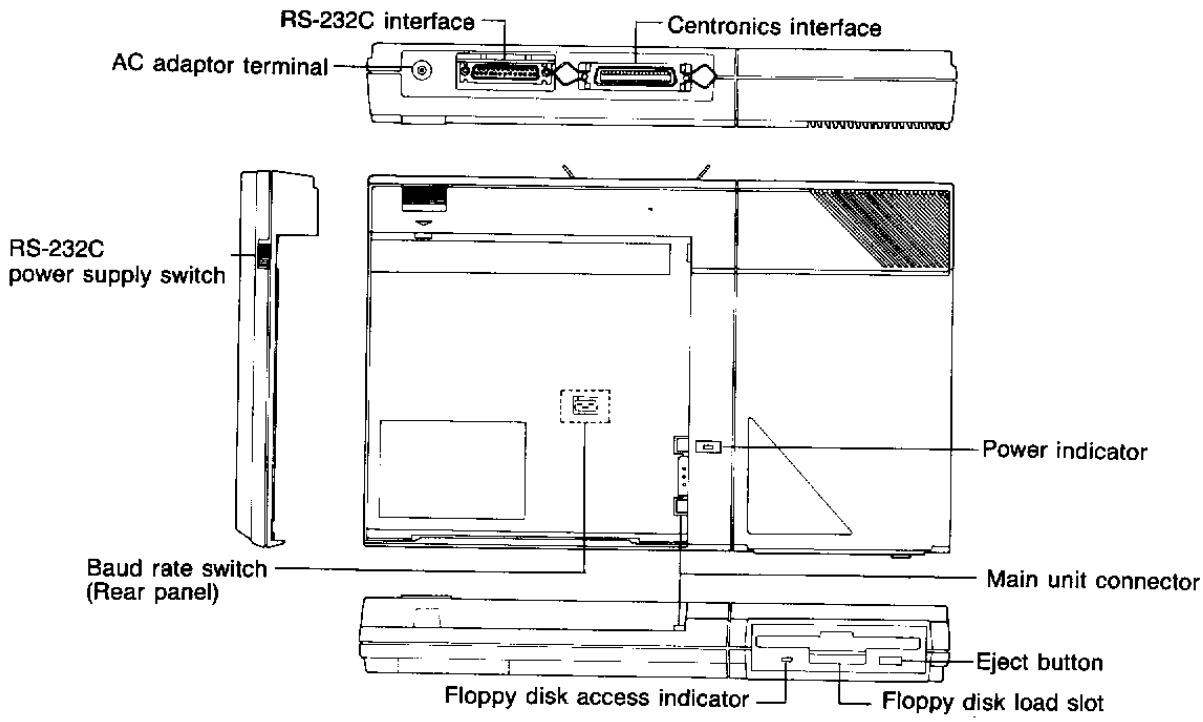
The optional 3.5" floppy disk drive unit is also equipped with an RS-232C interface and a Centronics standard printer interface.

Configuration

- An RS-232C interface, a Centronics standard printer interface connector, and an AC adaptor jack are all located on the back of the unit.
- An RS-232C power switch to control the power supply to the RS-232C interface is located on the left of the unit.
- A power indicator (LED) on the right side of the top of the unit lights (green) when power supply is normal. The AC adaptor (AD-4175) should be used or batteries should be replaced as soon as possible when the power indicator lights red. Continued use under low power may result in an LB error and loss of the contents in the currently accessed file.
- Baud rate switches and battery compartment are located on the bottom of the unit.
- To replace batteries, remove the battery compartment cover, remove the old batteries, and replace with a full new set ensuring that polarity (+/−) is correct.

Connection

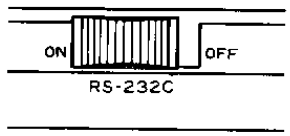
Always be sure to switch the power of the computer OFF before connecting it to the floppy disk drive unit. Once connected, the power switch of the computer also controls the power supply of the floppy disk drive unit.



RS-232C Interface

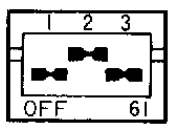
• RS-232C Switch

The RS-232C power switch should be set to OFF when the interface is not being used for data communications.



• Baud rate

The baud rate switches are located on the back of the unit. The baud rate set by these switches is taken as the default option when the baud rate specification is omitted in a software command. The following table shows the baud rate settings for the different combination of baud rate switch settings.



BPS	1	2	3
75	OFF	OFF	OFF
150	ON	OFF	OFF
300	OFF	ON	OFF
600	ON	ON	OFF
1200	OFF	OFF	ON
2400	ON	OFF	ON
4800	OFF	ON	ON
9600	ON	ON	ON

Specifications

Communication method	:	Start-stop (asynchronous) full-duplex mode only
Transmission speed	:	75, 150, 300, 600, 1200, 2400, 4800, 9600 baud
Parity bit	:	Odd, Even, None
Character bit length	:	7 or 8 bits
Stop bits	:	1 or 2 bits
CTS signal control	:	Control/no control
DSR signal control	:	Control/no control
CD signal control	:	Control/no control
Busy control	:	XON/XOFF control/no control
Input/output code system	:	SI/SO control/no control

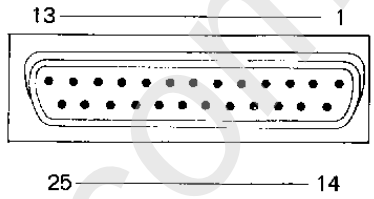
RS-232C Parameters

1. Baud Rate
   The following values indicate bits per second (BPS).
   0 = 75 4 = 1200
   1 = 150 5 = 2400
   2 = 300 6 = 4800
   3 = 600 7 = 9600
2. Parity Bit
   N = None E = Even O = Odd
3. Character Bit Length
   7 = JIS 7 bits 8 = JIS 8 bits
4. Stop Bit Length
   1 = 1 stop bit 2 = 2 stop bits
5. CTS Signal Control
   C = CTS control N = No CTS control
6. DSR Signal Control
   D = DSR control N = No DSR control
7. CD Signal Control
   C = CD signal control N = No CD signal control
8. Buffer Busy Control
   B = Buffer busy control N = No buffer busy control
9. SI/SO Control
   S = SI/SO control N = No SI/SO control

• Pin Configuration

Terminal #	Signal name
1	GND
2	TXD
3	RXD
4	RTS
5	CTS
6	DSR
7	GND
8	DCD
9	NC
10	NC
11	NC
12	NC
13	NC
14	NC
15	NC
16	NC
17	NC
18	NC
19	NC
20	DTR
21	NC
22	NC
23	NC
24	NC
25	NC

Pin connection



• RS-232C BASIC Commands

Command	Purpose
OPEN	Declares use of communications circuit
CLOSE	Closes an open communications circuit
PRINT#	Outputs data to communications circuit
PRINT#USING	Outputs data to communications circuit
INPUT#	Reads data from communications circuit
LINE INPUT#	Reads data from communications circuit
INPUT$	Reads data from communications circuit
EOF	Indicates status of receive buffer
LOF	Indicates the remaining number of bytes in receive buffer
SAVE	Outputs program to communications buffer
LOAD	Reads program from communications buffer
MERGE	Reads and merges program from communications buffer

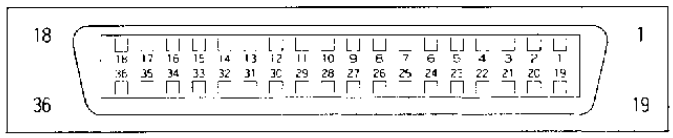
Centronics Interface

• General

The printer interface allows output of processing results and program lists from the computer to a Centronics standard printer. The Centronics standard interface is made available by connection of an interface unit or floppy disk drive unit.

• Pin Configuration

Number		Terminal Name	Number		Terminal Name
1	Output	$\overline{\text{STROBE}}$	19		GND
2	Output	DATA 1	20		GND
3	Output	DATA 2	21		GND
4	Output	DATA 3	22		GND
5	Output	DATA 4	23		GND
6	Output	DATA 5	24		GND
7	Output	DATA 6	25		GND
8	Output	DATA 7	26		GND
9	Output	DATA 8	27		GND
10	Input	$\overline{\text{ACKNLG}}$	28		GND
11	Input	BUSY	29		GND
12		NC	30		GND
13		NC	31	Output	$\overline{\text{INIT}}$
14		NC	32	Input	$\overline{\text{ERROR}}$
15		NC	33		GND
16		NC	34		NC
17		NC	35		NC
18		NC	36		NC



Control Lines

1. $\overline{\text{STROBE}}$
2. $\overline{\text{INIT}}$

} Output control

3. $\overline{\text{ACKNLG}}$
4. BUSY
5. $\overline{\text{ERROR}}$

} Input control

• Printer Commands Used in BASIC

Command	Purpose
LLIST	Outputs program contents to printer
LPRINT	Outputs specified characters to printer
TAB	Outputs spaces up to a specified location to printer
LPRINTUSING	Outputs data to printer for printing in specified format

Floppy Disk Drive Unit

General

The interface unit allows recording and storage of programs and data on a micro floppy disk. The disk must be initialized using the BASIC FORMAT command before it can be used for data storage.

Sepcifications

Type	:	3.5" single-sided, double-density track micro floppy disk unit
Capacity	:	320K bytes when formatted 500K bytes when unformatted
Number of tracks	:	80
Track density	:	135TPI
Transmission speed	:	250K bytes/second
Rotation speed	:	300rpm

• Disk Commands Used in BASIC

Command	Purpose
FORMAT	Initializes floppy disk
OPEN	Declares file use
CLOSE	Closes open file
PRINT#	Outputs data to sequential file
PRINT#USING	Outputs data to sequential file
INPUT#	Reads data from sequential file
LINE INPUT#	Reads data from sequential file up to CR code
INPUT$	Reads data from sequential file up to specified character
GET	Reads data from file to I/O buffer
PUT	Writes data from I/O buffer to file
LOF	Returns number of records in file
EOF	Returns end of file read
SAVE	Saves program to specified file
LOAD	Loads program contents
BSAVE	Saves memory contents to specified file
BLOAD	Loads file to specified memory address
MERGE	Merges program with program in file
CHAIN	Loads program contents and executes

12-4 PLOTTER–PRINTER

The plotter-printer makes it possible to produce hardcopies of calculation results and program lists, as well as graphics in four colors on letter-size paper.

Features

- Four color printing in black, red, blue and green.
- Italics, bold, emphasized and underline printing.
- Print resolution of 0.1mm/step.
- Post card size and letter size paper can be used.
- 256 character sizes from 1.0mm × 1.2mm (S0.0) ~ 16.0mm × 19.2mm (S15, 15)

Connection

The printer is connected to the computer via an interface unit or floppy disk drive unit.

Data Printing

- Programs are output to the printer using the BASIC LLIST command.
- Output to the printer during BASIC execution is accomplished using the LPRINT command.
- Data file contents are output to the printer by pressing [llist] in the MENU mode.
- Answering the prompt "list ? y/n" with [Y] when executing the assembler outputs the assembly list to the printer.

Plotter-printer Commands Used in BASIC

Command	Purpose
LLIST	Outputs program contents to printer
LPRINT	Outputs specified characters to printer
TAB	Outputs spaces up to a specified location to printer
LPRINTUSING	Outputs data to printer in specified format

* See the Operation Manual for further details.

PART 13
APPENDICES

13-1 CHARACTER CODE TABLE

High-order 4 bits →

↓ Low-order 4 bits	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0 16	(ROLL DOWN)	SPACE 32	0 48	@ 64	P 80	` 96	p 112	▬ 128	┴ 144	SPACE 160	ー 176	タ 192	ミ 208	三 224	× 240
1	(ROLL UP) 1	(DEL) 17	! 33	1 49	A 65	Q 81	a 97	q 113	▬ 129	┬ 145	。 161	ア 177	チ 193	ム 209	上 225	円 241
2	(LINE TOP) 2	(INS) 18	" 34	2 50	B 66	R 82	b 98	r 114	▬ 130	┤ 146	「 162	イ 178	ツ 194	メ 210	下 226	年 242
3	3	19	# 35	3 51	C 67	S 83	c 99	s 115	▬ 131	├ 147	」 163	ウ 179	テ 195	モ 211	丰 227	月 243
4	4	20	$ 36	4 52	D 68	T 84	d 100	t 116	▬ 132	─ 148	、 164	エ 180	ト 196	ヤ 212	◢ 228	日 244
5	(LINE DEL) 5	21	% 37	5 53	E 69	U 85	e 101	u 117	■ 133	┼ 149	・ 165	オ 181	ナ 197	ユ 213	◣ 229	時 245
6	(LINE END) 6	22	& 38	6 54	F 70	V 86	f 102	v 118	■ 134	├ 150	ヲ 166	カ 182	ニ 198	ヨ 214	◥ 230	分 246
7	7	23	' 39	7 55	G 71	W 87	g 103	w 119	■ 135	┤ 151	ア 167	キ 183	ヌ 199	ラ 215	◤ 231	秒 247
8	(BS) 8	(LINE C) 24	(40	8 56	H 72	X 88	h 104	x 120	▌ 136	┌ 152	イ 168	ク 184	ネ 200	リ 216	♠ 232	干 248
9	9	25	) 41	9 57	I 73	Y 89	i 105	y 121	▌ 137	┐ 153	ウ 169	ケ 185	ノ 201	ル 217	♥ 232	市 249
A	10	26	* 42	: 58	J 74	Z 90	j 106	z 122	▌ 138	└ 154	エ 170	コ 186	ハ 202	レ 218	◆ 234	区 250
B	(HOME) 11	27	+ 43	; 59	K 75	[91	k 107	{ 123	▌ 139	┘ 155	オ 171	サ 187	ヒ 203	ロ 219	♣ 235	町 251
C	(CLS) 12	(→) 28	, 44	< 60	L 76	¥ 92	l 108	¦ 124	■ 140	┌ 156	ヤ 172	シ 188	フ 204	ワ 220	● 236	村 252
D	(CR) (LF) 13	(←) 29	− 45	= 61	M 77	] 93	m 109	} 125	■ 141	┐ 157	ユ 173	ス 189	ヘ 205	ン 221	○ 237	人 253
E	14	(↑) 30	. 46	> 62	N 78	^ 94	n 110	~ 126	■ 142	└ 158	ヨ 174	セ 190	ホ 206	゛ 222	／ 238	▩ 254
F	15	(↓) 31	／ 47	? 63	O 79	_ 95	o 111	127	+ 143	┘ 159	ッ 175	ソ 191	マ 207	° 223	＼ 239	255

* Nothing is output for character codes for a character or function is not specified (indicated by a blank cell in the table).
* Control codes are indicated by parentheses and are not displayed.
* Characters which cannot be input directly can be displayed using the CHR$ function.
* Because of display limitations, 88H, 89H and 8AH appear to be the same on the screen, but are different when printed out.
* Character codes are hexadecimal values. 8AH should be represented as &H8A where &H indicates hexadecimal notation, 8 is the low-order 4 bits, and A is the high-order 4 bits.
* Values in the lower right corner of each cell indicate the decimal value of the corresponding character code.

13-2 ERROR MESSAGE TABLE

Error code	Error message	Meaning	Correction
1	OM error	a) Insufficient memory or system over flow. b) Erroneous CLEAR statement specification.	a) Shorten program and check array dimensioning. b) Check CLEAR statement value. c) Use expansion RAM pack.
2	SN error	Erroneous command or statement format.	a) Check spelling of commands. b) Check program input.
3	ST error	String length exceeds 255 characters.	Shorten string to 255 characters or less.
4	TC error	Formula too complex.	Divide formula into smaller sub-formulas
5	BV error	a) I/O buffer overflow. b) Line length exceed 255 bytes or 256 characters.	a) Set RS-232C baud rate to lower value or set XON/OFF. b) Keep lines 255 characters or less in length.
6	NR error	I/O device not ready for input/output.	a) Check connection and power switch of I/O device. b) Load a floppy disk into FDD.
7	RW error	a) Error generated in I/O device operation. b) LOAD of file for which FL error was generated at SAVE.	a) Check I/O device. See page 118 for cassette tape and page 119 for disk operations. b) Erase (kill) loaded program. NOTE: Repeated RW errors indicate that object program was not saved properly. Erase and resave, if possible.
8	BF error	Improper filename specification.	Check filename.
9	BN error	Improper file number specification.	Check file number.
10	NF error	Cannot find specified filename.	Recheck filename.
11	LB error	Low batteries in FDD.	a) Replace batteries in FDD. b) Use AC adaptor.
12	FL error	Disk space unavailable for writing.	a) Erase unneeded files. b) Use a new disk.
13	OV error	Value exceeds allowable calculation result or input range.	Check values.
14	MA error	a) Mathematical error such as division by zero. b) Argument exceeds allowable calculation range.	Check expressions and values.

Error code	Error message	Meaning	Correction
15	DD error	Double declaration of identical array.	Either erase previous array or use a different array name.
16	BS error	Subscript or parameter outside of allowable range.	a) Check subscripts. b) Increase size of arrays.
17	FC error	a) Erroneous use of function or statement. b) Illegal command used in direct mode or program mode. c) Illegal command used in CAL mode. d) Attempt to use undeclared array	a) Check argument values and statements. b) Check for statements that can not be used in respective mode. c) Check statements. d) Declare array using DIM statement.
18	UL error	a) Branch destination line number does not exist. b) Input of statement without line number in BASIC editing mode.	a) Check line numbers. b) Always use line numbers in BASIC editing mode. c) Enter BASIC programming mode.
19	TM error	a) Mismatch of variable type and contents. b) Mismatch of READ statement variable and data. c) Mismatch of INPUT # statement variable and data.	Check for illegal numeric assignment to string variables or string assignment to numeric variable.
20	RE error	RESUME statement outside of error handling routine.	Check RESUME statement location.
21	PR error	a) Attempt to write to password or write-protected disk or file. b) Execution of command that cannot be used with password protected files.	Cancel password or write protect status.
22	DA error	READ statement execution when no data present.	Check READ and DATA statements.
23	FO error	No FOR for NEXT statement.	Check for matching of FOR and NEXT statements.
24	NX error	No NEXT for FOR statement.	Check for matching of FOR and NEXT statements.
25	GS error	Mismatch of GOSUB and RETURN statements.	Check for matching of GOSUB and RETURN statements.
26	FM error	Unformatted or damaged disk.	Reformat disk or use new disk.
27	FD error	FIELD statement length exceeds 256 characters.	Ensure data length specified by FIELD statement is 256 characters or less.

Error code	Error message	Meaning	Correction
28	OP error	a) Attempt to access unopened file. b) Attempt to open already open file.	a) Execute OPEN statement. b) CLOSE file and then reopen.
29	AM error	a) Attempt to use random access for file opened for sequential access or vice versa. b) Attempt to use output-related command for device opened for input or vice versa. c) Attempt to load a random file. d) Attempt to use APPEND OPEN for BASIC or machine language file. e) Mismatched recorder baud rate. f) Attempt to execute machine language file without start address.	a) Do not use random access for sequential file and vice versa. b) Ensure proper used of input-related and output-related commands. c) Random files cannot be loaded. d) Do not use APPEND OPEN for BASIC files or machine language files e) Check MT baud rate. f) Include start address.
30	FR error	Framing error detected by RS-232C port.	Check RS-232C connection and data transmission method.
31	PO error	Parity error or over run error detected by RS-232C port.	a) Check RS-232C connection and data transmission method. b) Use slower baud rate.
32	DF error	a) Undefined command sent to FDD. b) Abnormality in FDD.	a) Erroneous machine language program b) Disk contents may not be retained.
0	?? error	Undefined error.	Abnormal operation. Press RESET and check memory contents. If abnormal, press NEW ALL.

13-3 COMMAND/FUNCTION TABLE

COMMANDS

· CLEAR	©	· SYSTEM	©	· LIST	Ⓜ
· VARLIST	©	· EDIT	Ⓜ	· DELETE	Ⓜ
· RUN	Ⓜ	· TRON/TROFF	©	· END	
· STOP		· GOTO		· GOSUB/RETURN	
· ON GOTO		· ON GOSUB		· IF/THEN/ELSE	
· FOR/NEXT		· REM(')		· LET	©
· DATA/READ/RESTORE		· INPUT		· PRINT	©
· PRINT USING	©	· LOCATE	©	· ANGLE	©
· BEEP(ON/OFF)	©	· CLS	©	· DIM	©
· ERASE	©	· DRAW/DRAWC	©	· MON	©
· CALL	©	· ON ERROR GOTO		· RESUME	
· DEFCHR$	©	· PASS	Ⓜ	· NEW	Ⓜ
· STAT	©	· STAT CLEAR	©	· POKE	©

INPUT/OUTPUT COMMANDS

· LLIST	Ⓜ	· LPRINT	©	· LPRINT USING	©
· FORMAT	©	· BSAVE	©	· BLOAD	©
· OPEN		· CLOSE	©	· PRINT #	
· INPUT #		· SAVE	Ⓜ	· LOAD	Ⓜ
· PUT/GET		· FIELD		· RSET/LSET	
· VERIFY	©	· CHAIN	Ⓜ	· MERGE	Ⓜ
· LINEINPUT #		· PRINT # USING			

Ⓜ : manual execution only

© : manual or CAL mode execution

NOTE: Ⓜ and © are not included for commands which would be meaningless in manual execution.

SCIENTIFIC FUNCTIONS

· CHR $	· ASC	· STR $
· VAL	· MID $	· RIGHT $
· LFFT $	· LEN	· HEX $
· &H	· INKEY $	· INPUT $
· INPUT #	· DEG	· DMS $
· POINT		
· SIN	· COS	· TAN
· ASN	· ACS	· ATN
· HYP SIN	· HYP COS	· HYP TAN
· HYP ASN	· HYP ACS	· HYP ATN
· EXP	· LOG	· LGT
· SQR	· ABS	· SGN
· INT	· FRAC	· ROUND
· PI	· RND	· PEEK
· TAB	· FIX	
· CNT	· SUMX	· SUMY
· SUMXY	· SUMX2	· SUMY2
· MEANX	· MEANY	· SDX
· SDY	· SDXN	· SDYN
· LRA	· LRB	· COR
· EOX	· EOY	
· EOF	· ERR	· ERL
· LOF	· REV	· NORM
· TIME $	· DATE $	

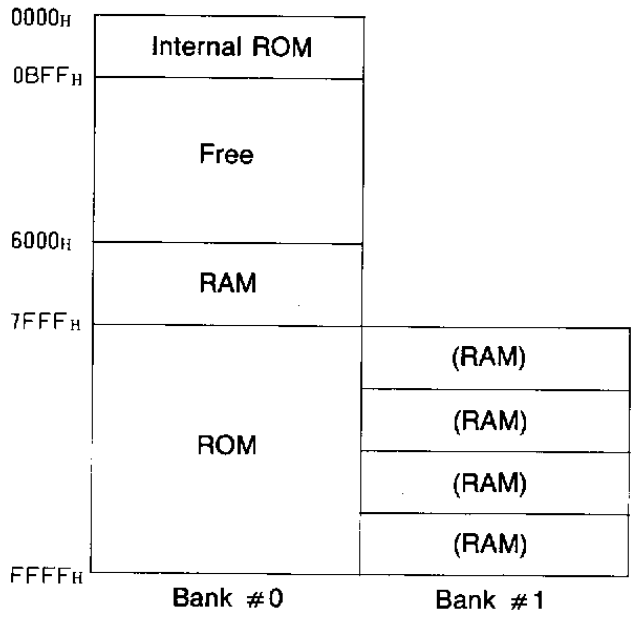
13-4 RESERVED WORD LIST

- A
  - ABS
  - ACS
  - AND
  - ANGLE
  - APPEND
  - AS
  - ASC
  - ASN
  - ATN
- B
  - BEEP
  - BLOAD
  - BSAVE
- C
  - CALL
  - CHAIN
  - CHR$
  - CLEAR
  - CLOSE
  - CLS
  - CNT
  - COR
  - COS
- D
  - DATA
  - DATE$
  - DEF
  - DEG
  - DELETE
  - DIM
  - DMS$
  - DRAW
  - DRAWC
- E
  - EDIT
  - ELSE
  - END
  - EOF
  - EOX
  - EOY
  - ERASE
  - ERL
  - ERR
  - ERROR
  - EXP
- F
  - FIELD
  - FIX
  - FOR
  - FORMAT
  - FRAC
- G
  - GET
  - GOSUB
  - GOTO
- H
  - HEX$
  - HYP
- I
  - IF
  - INKEY$
  - INPUT
  - INT
- L
  - LEFT$
  - LEN
  - LET
  - LGT
  - LINE
  - LIST
  - LLIST
  - LOAD
  - LOCATE
  - LOF
  - LOG
  - LPRINT
  - LRA
  - LRB
  - LSET
- M
  - MEANX
  - MEANY
  - MERGE
  - MID$
  - MOD
  - MON
- N
  - NEW
  - NEXT
  - NORM
  - NOT
- O
  - OFF
  - ON
  - OPEN
  - OR
  - OUT
- P
  - PASS
  - PEEK
  - PI
  - POINT
  - POKE
  - PRINT
  - PUT
- R
  - READ
  - REM
  - RESUME
  - RESTORE
  - RETURN
  - REV
  - RIGHT$
  - RND
  - ROUND
  - RSET
  - RUN
- S
  - SAVE
  - SDX
  - SDXN
  - SDY
  - SDYN
  - SGN
  - SIN
  - SQR
  - STAT
  - STEP
  - STOP
  - STR$
  - SUMX
  - SUMX2
  - SUMXY
  - SUMY
  - SUMY2
  - SYSTEM
- T
  - TAB
  - TAN
  - THEN
  - TIME$
  - TO
  - TROFF
  - TRON
- U
  - USING
- V
  - VAL
  - VARLIST
  - VERIFY
- X
  - XOR

13-5 MEMORY MAP

Memory Free Areas

The HD61700 has free areas from addresses 00000H through 3FFFFH, (18 external addresses and 8 data addresses), while the free area of the computer is located at the addresses illustrated below.



The internal ROM indicated in the memory map is the HD61700 ROM (addresses 0H ~ BFFH, 3072 × 16 bits).

SPECIFICATIONS

Model :
PB-1000

Basic calculation functions :
Negative numbers, exponents, parenthetical arithmetic operations (with priority sequence judgment function – true algebraic logic), integer division, integer division remainders, logical operators

Built-in functions :
Trigonometric/inverse trigonometric functions (angular units: degrees, radians, grads), logarithmic/exponential functions, square roots, powers, hyperbolic/inverse hyperbolic functions, conversion to integer, deletion of integer portion, absolute values, signs, rounding, random numbers, pi, decimal-sexagesimal conversions, decimal-hexadecimal conversion

Statistical calculation functions :
Standard deviation – number of data, sum of x, sum of squares, standard deviation (two types)
Linear regression – number of data, sum of x, sum of y, sum of squares of x, sum of squares of y, sum of products of data, standard deviation of x (two types), standard deviation of y (2 types), constant term, regression coefficient, correlation coefficient, estimated value of x, estimated value of y

Function calculation accuracy :
10th digit ± 1 of the mantissa

Commands :
CLEAR, VARLIST, END, GOTO, ON GOSUB, REM ('), PRINT, ANGLE, INPUT, CLS, ON ERROR GOTO, DEFCHR$, NEW, POKE, SYSTEM, EDIT, TRON/TROFF, GOSUB/RETURN, IF/THEN/ELSE, LET, DRAW/DRAWC, LOCATE, DIM, MON, RESUME, PASS, STAT, STAT CLEAR, LIST, RUN, STOP, ON GOTO, FOR/NEXT, DATA/READ/RESTORE, PRINT USING, BEEP (ON/OFF), ERASE, CALL, DELETE, LLIST, FORMAT, OPEN, INPUT#, PUT/GET, VERIFY, LINE INPUT#, LPRINT, BSAVE, CLOSE, SAVE, FIELD, CHAIN, PRINT# USING, LPRINT USING, BLOAD, PRINT#, LOAD, RSET/LSET, MERGE

Functions :
INPUT$, INPUT#, ERR, ERL, EOF, LOF, REV, NORM, TIME$, DATE$

Program functions :
CHR$, ASC, STR$, VAL, MID$, RIGHT$, LEFT$, LEN, HEX$, INKEY$, DEG, DMS$, POINT, PEEK, TAB, &H

Calculation precision :
± 1 at 10th digit However, errors may be cumulative for internal consecutive calculations using $\wedge$, HYP, and statistical functions, and accuracy is sometime affected.
Accuracy is lessened when the following functions approach the values noted

sinx	$\|x\| = 90° \times 2n$
cosx	$\|x\| = 90° \times (2n + 1)$
tanx	$\|x\| = 90° \times n$

Program language :
C_{61}-BASIC, HD61700 assembler

Memory capacity (user area) :
4,096 bytes (when 8KB)
36,864 bytes (when 40KB)

Number of variables :
Limited by memory capacity only

Number of stacks :

System stack	255 bytes
User stack	249 bytes
FOR ~ NEXT	Limited by memory capacity or 255 levels
GOSUB	Limited by memory capacity

Numeric display :
10-digit mantissa + 2-digit exponent

Display element :
192 × 32 dot LCD (32 × 4 characters)

Main component :
C-MOS LSI

Power supply :
AA-size batteries × 3

Power consumption :
0.14W

Battery life:
1. Continuous program execution: Approx. 55 hours
2. Continuous display of 5555555555 at 20°C (68°F): Approx. 100 hours
   3 months when unit is used 1 hour per day.

* NOTE: 1 hour includes 10 minutes of condition 1 and 50 minutes of condition 2.

Auto power OFF :
Approximately 7 minutes after last key operation.

Ambient temperature range :
0°C ~ 40°C (32°F ~ 104°F)

Dimensions :

Folded:	24(H) × 187(W) × 97(D)mm
	(1"(H) × $7^{3}/_{8}$"(W) × $3^{7}/_{8}$"(D))
Unfolded :	24(H) × 187(W) × 176.5(D)mm
	(1"(H) × $7^{3}/_{8}$"(W) × 7"(D))

Weight :
435g (15.3 oz) including batteries

INDEX

R

S

T

V

W

X

GUIDELINES LAID DOWN BY FCC RULES FOR USE OF THE UNIT IN THE U.S.A. (not applicable to other areas).

WARNING: This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

. . . . reorient the receiving antenna
. . . . relocate the computer with respect to the receiver
. . . . move the computer away from the receiver
. . . . plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

"How to Identify and Resolve Radio-TV Interference Problems"

This booklet is available from the US Government Printing Office, Washington D.C., 20402, Stock No.004-000-00345-4.