

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER  
***E0C6001 DEVELOPMENT TOOL MANUAL***

---



#### ***NOTICE***

---

*No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Control Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency. Please note that "E0C" is the new name for the old product "SMC". If "SMC" appears in other manuals understand that it now reads "E0C".*

# E0C6001 Development Tool Manual

## ***PREFACE***

---

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Micro-computer E0C6001.

Refer to the "E0C62 Family Development Tool Reference Manual" for the details (common to all models) of each development support tool. Manuals for hardware development tools are separate, so you should also refer to the below manuals.

<i>Development tools</i>	☞ E0C62 Family Development Tool Reference Manual EVA6001 Manual ICE6200 Hardware Manual
<i>Development procedure</i>	☞ E0C62 Family Technical Guide
<i>Device (E0C6001)</i>	☞ E0C6001 Technical Manual
<i>Instructions</i>	☞ E0C6200/6200A Core CPU Manual

## ***CONTENTS***

---

<b>1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL</b>	<b>1</b>
1.1 Configuration of DEV6001 .....	1
1.2 Developmental Environment .....	1
1.3 Development Flow .....	2
1.4 Production of Execution Disk .....	3
<b>2 CROSS ASSEMBLER ASM6001</b>	<b>4</b>
2.1 ASM6001 Outline .....	4
2.2 E0C6001 Restrictions .....	4
2.3 ASM6001 Quick Reference .....	5
<b>3 FUNCTION OPTION GENERATOR FOG6001</b>	<b>7</b>
3.1 FOG6001 Outline .....	7
3.2 E0C6001 Option List .....	7
3.3 Option Specifications and Selection Message .....	9
3.4 FOG6001 Quick Reference .....	15
3.5 Sample File .....	17
<b>4 SEGMENT OPTION GENERATOR SOG6001</b>	<b>19</b>
4.1 SOG6001 Outline .....	19
4.2 Option List .....	19
4.3 Segment Ports Output Specifications .....	20
4.4 SOG6001 Quick Reference .....	21
4.5 Sample Files .....	23
<b>5 ICE CONTROL SOFTWARE ICS6001</b>	<b>24</b>
5.1 ICS6001 Outline .....	24
5.2 ICS6001 Restrictions .....	24
5.3 ICS6001 Quick Reference .....	25

<b>6 MASK DATA CHECKER MDC6001</b>	<b>28</b>
6.1 MDC6001 Outline .....	28
6.2 MDC6001 Quick Reference .....	28
<b>APPENDIX A. E0C6001 INSTRUCTION SET</b>	<b>31</b>
<b>B. E0C6001 RAM MAP</b>	<b>35</b>
<b>C. E0C6001 I/O MEMORY MAP</b>	<b>36</b>
<b>D. TROUBLESHOOTING</b>	<b>37</b>
<b>E. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6001</b>	<b>39</b>

# 1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL

Here we will explain the composition of the software for the development support tools, developmental environment and how to generate the execution disk.

## 1.1 Configuration of DEV6001

The below software are included in the product of the E0C6001 development support tool DEV6001.

1. Cross Assembler ASM6001 ..... Cross assembler for program preparation
2. Function Option Generator FOG6001 ..... Function option data preparation program
3. Segment Option Generator SOG6001 ..... Segment option data preparation program
4. ICE Control Software ICS6001 ..... ICE control program
5. Mask Data Checker MDC6001 ..... Mask data preparation program

## 1.2 Developmental Environment

The software product of the development support tool DEV6001 operates on the following host systems:

- IBM PC/AT (at least PC-DOS Ver. 2.0)
- NEC PC-9801 Series (at least MS-DOS Ver. 3.1)

When developing the E0C6001, the above-mentioned host computer, editor, P-ROM writer, printer, etc. must be prepared by the user in addition to the development tool which is normally supported by Seiko Epson.

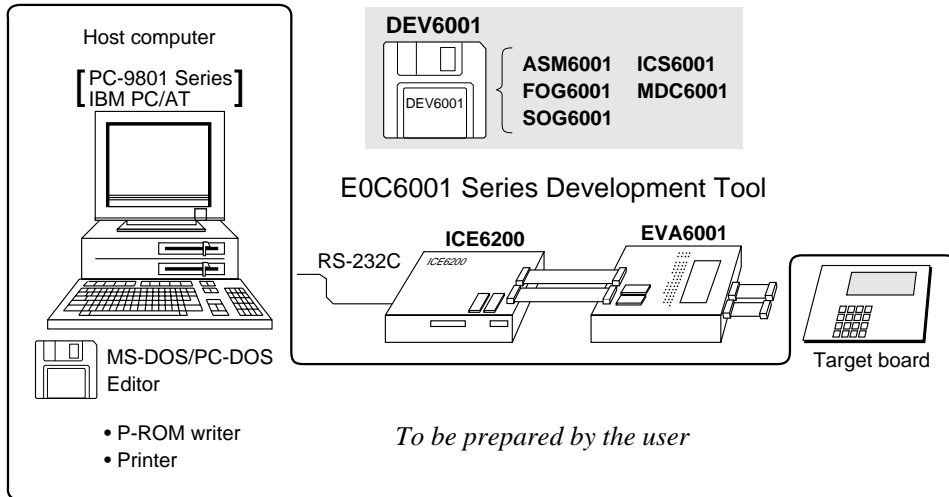


Fig. 1.2.1 System configuration

**Note** The DEV6001 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 is connected to the host computer with a RS-232C serial interface, adapter board for asynchronous communication will be required depending on the host computer used.

### 1.3 Development Flow

Figure 1.3.1 shows the development flow through the DEV6001.

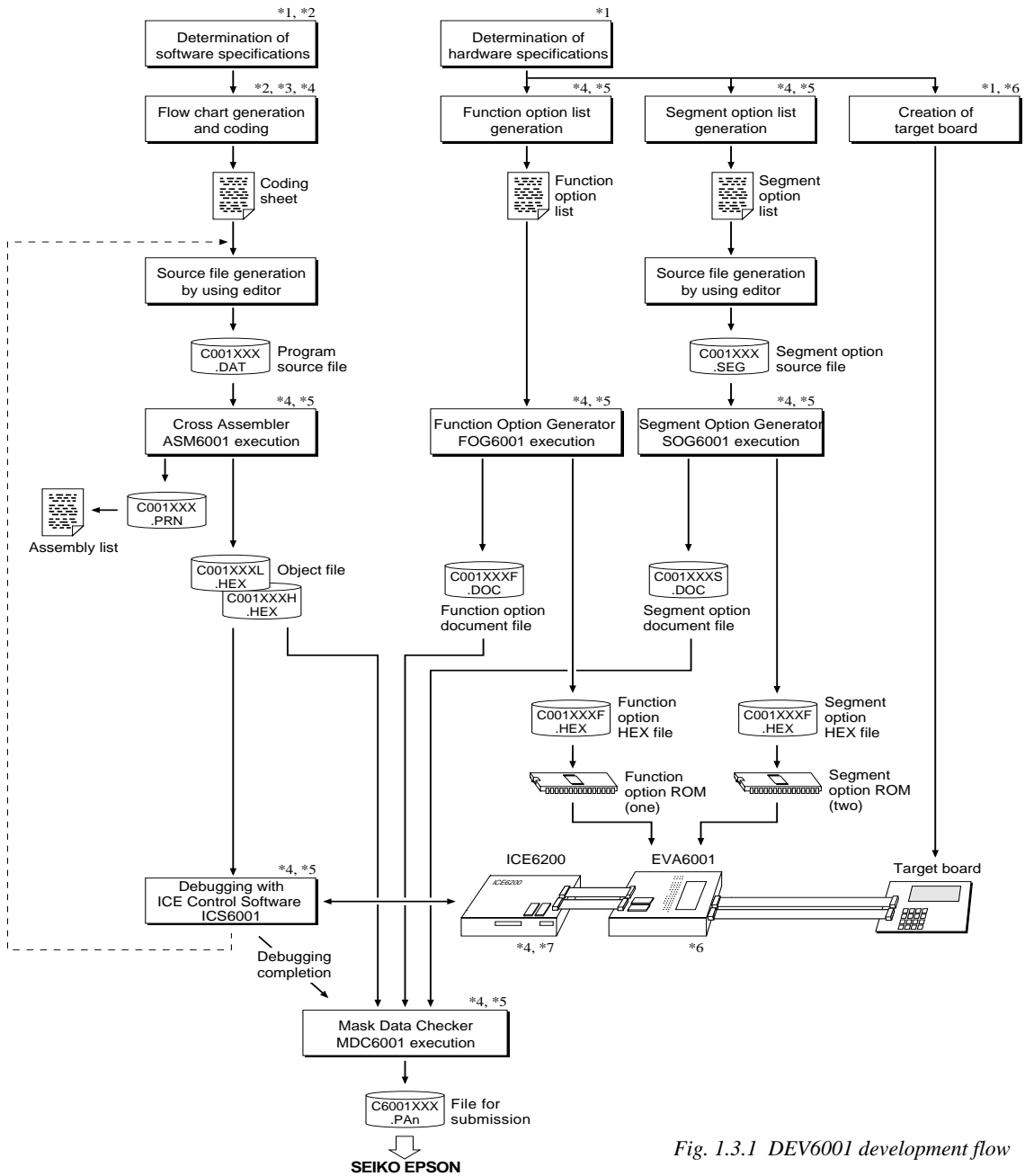


Fig. 1.3.1 DEV6001 development flow

#### Concerning file names

All the input-output file name for the each development support tool commonly use "C001XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

#### Reference Manual

- \*1 E0C6001 Technical Hardware Manual
- \*2 E0C6001 Technical Software Manual
- \*3 E0C6200/6200A Core CPU Manual
- \*4 E0C62 Family Development Tool Reference Manual
- \*5 E0C6001 Development Tool Manual (this manual)
- \*6 EVA6001 Manual
- \*7 ICE6200 Hardware Manual

## 1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 are recorded in the DEV6001 floppy disk.

The content of the files contained in the DEV6001 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM6001.EXE	ASM6001.EXE	Cross Assembler execution file
FOG6001.EXE	FOG6001.EXE	Function Option Generator execution file
ICS6001B.BAT	ICS6001.BAT	ICE Control Software batch file
ICS6001W.EXE	ICS6001J.EXE	ICE Control Software execution file
ICS6001P.PAR	ICS6001P.PAR	ICE Control Software parameter file
MDC6001.EXE	MDC6001.EXE	Mask Data Checker execution file
SOG6001.EXE	SOG6001.EXE	Segment Option Generator execution file

- First copy the entire content of this disk using commands such as DISKCOPY then make the execution disk. Carefully conserve the original floppy disk for storage purposes. When copying into a hard disk, make a sub-directory with an appropriate name (DEV6001, etc.) then copy the content of the floppy disk into that subdirectory using the COPY command.

- Next make a CONFIG.SYS file using Editor or the like. When a CONFIG.SYS has previously been made using a hard disk system, check the setting of the FILES within it. (If there is none add it.) Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC6001 will handle many files.

*Note* The driver for the RS-232C must be included in CONFIG.SYS by the host computer.

- In "ICS6001(B).BAT" the batch process is indicated such that the ICS6001J(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters. The SPEED (MS-DOS) or MODE (PC-DOS) command is used for setting the RS-232C, so you should copy these commands into the disk to be executed or specify a disk or directory with the command using the PATH command.

Example:

### **Copying into a floppy disk**

Insert the original disk into the A drive and the formatted disk to be copied into B drive, then execute the DISKCOPY command.

```
A>DISKCOPY A: B: [↵]
```

### **Copying into a hard disk (C drive)**

Make a subdirectory (DEV6001), then insert the original disk into the A drive and execute the COPY command.

```
C>MD DEV6001 [↵]
```

```
C>CD DEV6001 [↵]
```

```
C\DEV6001\>COPY A:*. * [↵]
```

Example:

### **Setting of FILES (CONFIG.SYS)**

```
C\>TYPE CONFIG.SYS [↵]
```

```
:
```

```
FILES=20
```

```
:
```

### **RS-232C Setting (PC-DOS version)**

```
MODE COM1: 4800, n, 8, 1, p
```

### **RS-232C Setting (MS-DOS version)**

```
SPEED R0 9600 B8 PN S1
```

# 2 CROSS ASSEMBLER ASM6001

## 2.1 ASM6001 Outline

The ASM6001 cross assembler is an assembler program for generating the machine code used by the E0C6001 4-bit, single-chip microcomputers. The Cross Assembler ASM6001 will assemble the program source files which have been input by the user's editor and will generate an object file in Intel-Hex format and assembly list file.

In this assembler, program modularization has been made possible through macro definition functions and programming independent of the ROM page structure has been made possible through the auto page set function. In addition, consideration has also been given to precise error checks for program capacity (ROM capacity) overflows, undefined codes and the like, and for debugging of such things as label tables for assembly list files and cross reference table supplements.

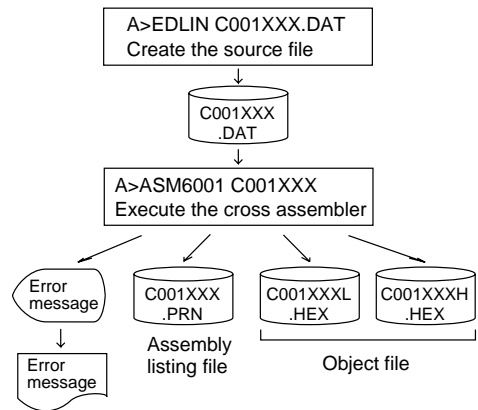


Fig. 2.1.1 ASM6001 execution flow

The format of the source file and its operating method are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

## 2.2 E0C6001 Restrictions

Note the following when generating a program by the E0C6001:

### ROM area

The capacity of the E0C6001 ROM is 1K steps (0000H to 03FFH).

Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

### Memory configuration:

Bank: Only bank 0, Page: 4 pages (0 to 3H), each 256 steps

### Significant specification range:

ORG pseudo-instruction: 0000H to 03FFH  
 PAGE pseudo-instruction: 00H to 03H  
 BANK pseudo-instruction: Only 0H  
 PSET instruction: 00H to 03H

### RAM area

The capacity of the E0C6001 RAM is 113 words (000H to 04FH, 090H to 0AFH, and 0E0H to 0FDH, 4 bits/word).

Memory access is invalid when the unused area of the index register is specified.

**Example:** LD X, 050H 50H is loaded into the IX register, but an unused area has been specified so that the memory accessible with the IX register (MX) is invalid.  
 LD Y, 0C7H C7H is loaded into the IY register, but an unused area has been specified so that the memory accessible with the IY register (MY) is invalid.

### Undefined codes

The following instructions have not been defined in the E0C6001 instruction sets.

SLP			
PUSH	XP	PUSH	YP
POP	XP	POP	YP
LD	XP,r	LD	YP,r
LD	r,XP	LD	r,YP



## 2.3 ASM6001 Quick Reference

### Starting command and input/output files

\_ indicates a blank.

indicates the Return key.

A parameter enclosed by [ ] can be omitted.

**Execution file:** ASM6001.EXE

**Starting command:** **ASM6001\_** [drive-name:] source-file-name [.shp]\_ [-N]

- Option:**
- .shp Specifies the file I/O drives.
  - s Specifies the drive from which the source file is to be input. (A-P, @)
  - h Specifies the drive to which the object file is to be output. (A-P, @, Z)
  - p Specifies the drive to which the assembly listing file is to be output. (A-P, @, Z)
  - @: Current drive, Z: File is not generated
  - N The code (FFH) in the undefined area of program memory is not created.

**Input file:** C001XXX.DAT (Source file)

**Output file:** C001XXXL.HEX (Object file, low-order)  
 C001XXXH.HEX (Object file, high-order)  
 C001XXX.PRN (Assembly listing file)

### Display example

```

*** E0C6001 CROSS ASSEMBLER. --- Ver 1.00 ***

EEEEEEEEEE PPPPPPPP SSSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPPPP SSS SSSS OOO OOO NNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNNN NNN
EEEEEEEEEE PPPPPPPPPP SSSSSS OOO OOO NNN NNN
EEEEEEEEEE PPPPPPPPPP SSSSSS OOO OOO NNN NNNNNN
EEE PPP SSS SSS OOO OOO NNN NNNNN
EEE PPP SSS SSS OOO OOO NNN NNNNN
EEEEEEEEEE PPP SSSS SSS OOO OOO NNN NNN
EEEEEEEEEE PPP SSSSSS OOOOOOOO NNN NN

(C) COPYRIGHT 1996 SEIKO EPSON CORP.

SOURCE FILE NAME IS " C001XXX.DAT "

THIS SOFTWARE MAKES NEXT FILES.

C001XXXH.HEX ... HIGH BYTE OBJECT FILE.
C001XXXL.HEX ... LOW BYTE OBJECT FILE.
C001XXX.PRN ... ASSEMBLY LIST FILE.

DO YOU NEED AUTO PAGE SET? (Y/N) Y ... (1)
DO YOU NEED CROSS REFERENCE TABLE? (Y/N) Y ... (2)
    
```

When ASM6001 is started, the start-up message is displayed.

At (1), select whether or not the auto-page-set function will be used.

Use ..... Y

Not use ..... N

If the assembly listing file output is specified, message (2) is displayed. At this stage, cross-reference table generation may be selected.

Generating ..... Y

Not generating ..... N

When the above operation is completed, ASM6001 assembles the source file.

To suspend execution, press the "CTRL" and "C" keys together at stage (1) or (2).

### Operators

Arithmetic operators		Logical operators	
+a	Monadic positive	a_AND_b	Logical product
-a	Monadic negative	a_OR_b	Logical sum
a+b	Addition	a_XOR_b	Exclusive logical sum
a-b	Subtraction	NOT_a	Logical negation
a*b	Multiplication	Relational operators	
a/b	Division	a_EQ_b	True when a is equal to b
a_MOD_b	Remainder of a/b	a_NE_b	True when a is not equal to b
a_SHL_b	Shifts a b bits to the left	a_LT_b	True when a is less than b
a_SHR_b	Shifts a b bits to the right	a_LE_b	True when a is less than or equal to b
HIGH_a	Separates the high-order eight bits from a	a_GT_b	True when a is greater than b
LOW_a	Separates the low-order eight bits from a	a_GE_b	True when a is greater than or equal to b

## ■ Pseudo-instructions

Pseudo-instruction	Meaning	Example of use
EQU (Equation)	To allocate data to label	ABC EQU 9 BCD EQU ABC+1
SET (Set)	To allocate data to label (data can be changed)	ABC SET 0001H ABC SET 0002H
DW (Define Word)	To define ROM data	ABC DW 'AB' BCD DW 0FFBH
ORG (Origin)	To define location counter	ORG 100H ORG 256
PAGE (Page)	To define boundary of page	PAGE 1H PAGE 3
SECTION (Section)	To define boundary of section	SECTION
END (End)	To terminate assembly	END
MACRO (Macro)	To define macro	CHECK MACRO DATA LOCAL LOOP
LOCAL (Local)	To make local specification of label during macro definition	LOOP CP MX, DATA JP NZ, LOOP
ENDM (End Macro)	To end macro definition	ENDM CHECK 1

## ■ Error messages

Error message	Explanation
S (Syntax Error)	An unrecoverable syntax error was encountered.
U (Undefined Error)	The label or symbol of the operand has not been defined.
M (Missing Label)	The label field has been omitted.
O (Operand Error)	A syntax error was encountered in the operand, or the operand could not be evaluated.
P (Phase Error)	The same label or symbol was defined more than once.
R (Range Error)	<ul style="list-style-type: none"> <li>The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified.</li> <li>A value greater than that which the number of significant digits of the operand will accommodate was specified.</li> </ul>
! (Warning)	<ul style="list-style-type: none"> <li>Memory areas overlapped because of a "PAGE" or "ORG" pseudo-instruction or both.</li> </ul>
FILE NAME ERROR	The source file name was longer than 8 characters.
FILE NOT PRESENT	The specified source file was not found.
DIRECTORY FULL	No space was left in the directory of the specified disk.
FATAL DISK WRITE ERROR	The file could not be written to the disk.
LABEL TABLE OVERFLOW	The number of defined labels and symbols exceeded the label table capacity (4000).
CROSS REFERENCE TABLE OVERFLOW	The label/symbol reference count exceeded the cross-reference table capacity (only when the cross-reference table is generated).


# 3 FUNCTION OPTION GENERATOR FOG6001

## 3.1 FOG6001 Outline

With the 4-bit single-chip E0C6001 microcomputers, the customer may select 9 hardware options. By modifying the mask patterns of the E0C6001 according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG6001 is a software tool for generating data files used to generate mask patterns. It enables the customer to interactively select and specify pertinent items for each hardware option. From the data file created with FOG6001, the E0C6001 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA6001) hardware option ROM is simultaneously generated with the data file.

 The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

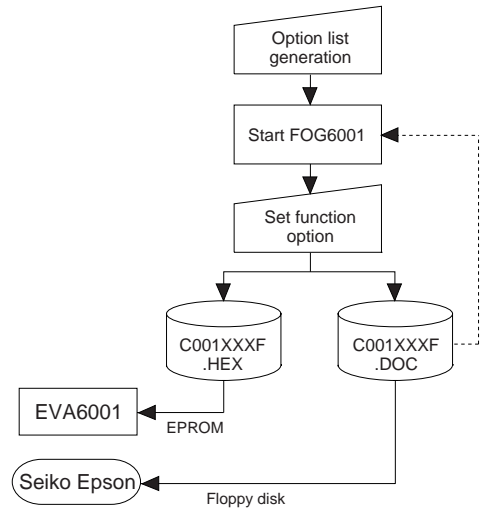


Fig. 3.1.1 FOG6001 execution flow

## 3.2 E0C6001 Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "3.3 Option Specifications and Selection Message" as reference, select the specifications that meet the target system. Be sure to record the specifications for unused ports too, according to the instructions provided.

### 1. DEVICE TYPE AND LCD VOLTAGE

- |                                      |                  |           |
|--------------------------------------|------------------|-----------|
| <input type="checkbox"/> 1. E0C6001  | (Normal Type)    | LCD 3 V   |
| <input type="checkbox"/> 2. E0C6001  | (Normal Type)    | LCD 4.5 V |
| <input type="checkbox"/> 3. E0C60L01 | (Low Power Type) | LCD 3 V   |
| <input type="checkbox"/> 4. E0C60L01 | (Low Power Type) | LCD 4.5 V |

### 2. MULTIPLE KEY ENTRY RESET

- COMBINATION .....
- |  |
|--|
| <input type="checkbox"/> 1. Not Use                |
| <input type="checkbox"/> 2. Use K00, K01           |
| <input type="checkbox"/> 3. Use K00, K01, K02      |
| <input type="checkbox"/> 4. Use K00, K01, K02, K03 |

### 3. INTERRUPT NOISE REJECTOR

- K00-K03 .....
- |                                 |                                     |
|---------------------------------|-------------------------------------|
| <input type="checkbox"/> 1. Use | <input type="checkbox"/> 2. Not Use |
|---------------------------------|-------------------------------------|

### 4. INPUT PORT PULL DOWN RESISTOR

- |             |   |   |
|-------------|---|---|
| • K00 ..... | <input type="checkbox"/> 1. With Resistor | <input type="checkbox"/> 2. Gate Direct |
| • K01 ..... | <input type="checkbox"/> 1. With Resistor | <input type="checkbox"/> 2. Gate Direct |
| • K02 ..... | <input type="checkbox"/> 1. With Resistor | <input type="checkbox"/> 2. Gate Direct |
| • K03 ..... | <input type="checkbox"/> 1. With Resistor | <input type="checkbox"/> 2. Gate Direct |

**5. R00 SPECIFICATION**

- OUTPUT TYPE .....  1. DC Output  
 2. Buzzer Inverted Output (R00 Control)  
 3. Buzzer Inverted Output (R01 Control)  
 4. FOUT Output
  
- FOUT OUTPUT SPACIFICATION

  - F1 .....  1. 256[Hz]  
 2. 512[Hz]  
 3. 1,024[Hz]  
 4. 2,048[Hz]  
 5. 4,096[Hz]
  - F2 .....  1. 512[Hz]  
 2. 1,024[Hz]  
 3. 2,048[Hz]  
 4. 4,096[Hz]  
 5. 8,192[Hz]
  - F3 .....  1. 1,024[Hz]  
 2. 2,048[Hz]  
 3. 4,096[Hz]  
 4. 8,192[Hz]  
 5. 16,384[Hz]
  - F4 .....  1. 2,048[Hz]  
 2. 4,096[Hz]  
 3. 8,192[Hz]  
 4. 16,384[Hz]  
 5. 32,768[Hz]

  
- OUTPUT SPECIFICATION .....  1. Complementary  2. Pch-Open Drain

**6. R01 SPECIFICATION**

- OUTPUT TYPE .....  1. DC Output  2. Buzzer Output
- OUTPUT SPECIFICATION .....  1. Complementary  2. Pch-Open Drain

**7. I/O PORT SPECIFICATION**

- P00 .....  1. Complementary  2. Pch-Open Drain
- P01 .....  1. Complementary  2. Pch-Open Drain
- P02 .....  1. Complementary  2. Pch-Open Drain
- P03 .....  1. Complementary  2. Pch-Open Drain

**8. LCD COMMON DUTY AND BIAS**

- 1. 1/4 Duty 1/3 Bias
- 2. 1/3 Duty 1/3 Bias
- 3. 1/2 Duty 1/3 Bias
- 4. 1/4 Duty 1/2 Bias
- 5. 1/3 Duty 1/2 Bias
- 6. 1/2 Duty 1/2 Bias

**9. OSC1 SYSTEM CLOCK**

- 1. Crystal
- 2. CR

### 3.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6001 are shown below, and their specifications are also described.

#### 1 Device type and LCD voltage

```

*** OPTION NO.1 ***
--- DEVICE TYPE & LCD POWER VREG ---

      1. E0C6001 LCD 3V
      2. E0C6001 LCD 4.5V
      3. E0C60L01 LCD 3V
      4. E0C60L01 LCD 4.5V

PLEASE SELECT NO.(1) ? 4
      4. E0C60L01 LCD 4.5V  SELECTED
    
```

Select the chip specification.

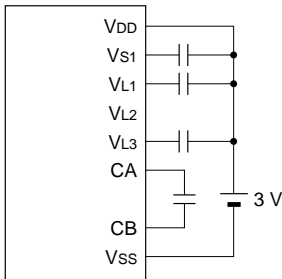
There are two models: E0C6001 (3 V supply voltage) and E0C60L01 (1.5 V supply voltage, low-power specification).

Select the LCD drive voltage (3 V or 4.5 V) according to the LCD panel to be used.

Figure 3.3.1 shows the external elements.

#### E0C6001

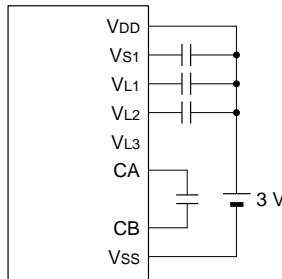
4.5 V LCD panel  
1/4, 1/3, 1/2 duty, 1/3 bias



Note: VL2 is shorted to VSS inside the IC.

#### E0C6001

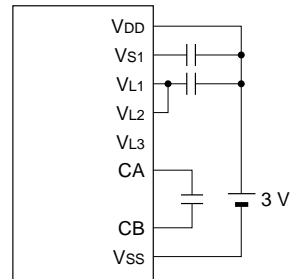
3 V LCD panel  
1/4, 1/3, 1/2 duty, 1/3 bias



Note: VL3 is shorted to VSS inside the IC.

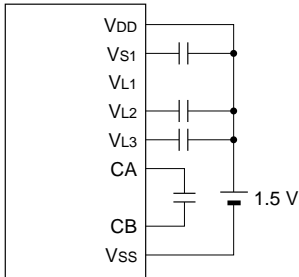
#### E0C6001

3 V LCD panel  
1/4, 1/3, 1/2 duty, 1/2 bias



#### E0C60L01

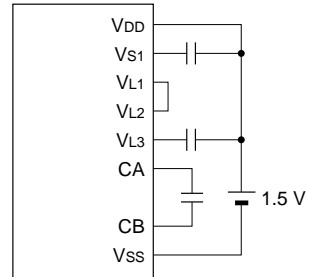
4.5 V LCD panel  
1/4, 1/3, 1/2 duty, 1/3 bias



Note: VL1 is shorted to VSS inside the IC.

#### E0C60L01

3 V LCD panel  
1/4, 1/3, 1/2 duty, 1/2 bias



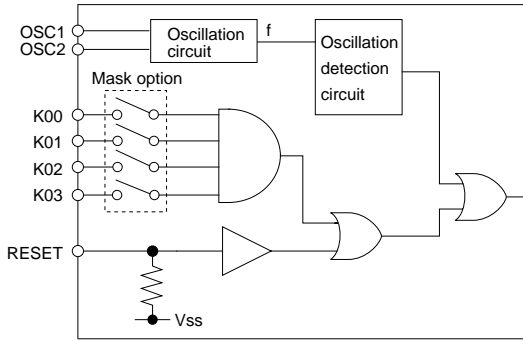
Note: VL1 is shorted to VSS inside the IC.

Fig. 3.3.1 External elements

## 2 Multiple key entry reset

```

*** OPTION NO.2 ***
--- MULTIPLE KEY ENTRY RESET ---
COMBINATION      1. NOT USE
                  2. USE K00,K01
                  3. USE K00,K01,K02
                  4. USE K00,K01,K02,K03
PLEASE SELECT NO.(1) ? 2 [ ]
COMBINATION      2. USE K00,K01  SELECTED
    
```



The reset function is set when K00 through K03 are entered. When "NOT USE" is selected, the reset function is not activated even if K00 through K03 are entered. When "USE K00, K01" is selected, the system is reset immediately the K00 and K01 inputs go high at the same time. Similarly, the system is reset as soon as the K00 through K02 inputs or the K00 through K03 inputs go high. However, the system is reset when a high signal is input for more than a rule time (1–3 sec). The system reset circuit is shown in Figure 3.3.2.

Fig. 3.3.2 System reset circuit

## 3 Interrupt noise rejector

```

*** OPTION NO.3 ***
--- INTERRUPT NOISE REJECTOR ---
K00-K03         1. USE
                  2. NOT USE
PLEASE SELECT NO.(1) ? 1 [ ]
K00-K03         1. USE  SELECTED
    
```

Select whether noise rejector will be supplemented to the input interrupter of K00–K03. When "USE" is selected, the entry signal will pass the noise rejector, and occurrence of interrupt errors due to noise or chattering can be avoided. Note, however, that because the noise rejector performs entry signal sampling at 4 kHz, "NOT USE" should be selected when high speed response is required.

## 4 Input ports pull down resistor

```

*** OPTION NO.4 ***
--- INPUT PORT PULL DOWN RESISTOR ---
K00             1. WITH RESISTOR
                  2. GATE DIRECT
PLEASE SELECT NO.(1) ? 1 [ ]
K01             1. WITH RESISTOR
                  2. GATE DIRECT
PLEASE SELECT NO.(1) ? 1 [ ]
K02             1. WITH RESISTOR
                  2. GATE DIRECT
PLEASE SELECT NO.(1) ? 1 [ ]
K03             1. WITH RESISTOR
                  2. GATE DIRECT
PLEASE SELECT NO.(1) ? 1 [ ]
K00             1. WITH RESISTOR  SELECTED
K01             1. WITH RESISTOR  SELECTED
K02             1. WITH RESISTOR  SELECTED
K03             1. WITH RESISTOR  SELECTED
    
```

Select whether input ports (K00–K03) will each be supplemented with pull down resistors or not. When "GATE DIRECT" is selected, see to it that entry floating state does not occur. Select "WITH RESISTOR" pull down resistor for unused ports. Moreover, the input port status is changed from "H" level (VDD) to "L" level (Vss) with pull down resistors, a delay of approximately 1 msec in waveform fall time will occur depending on the pull down resistor and entry load time constant. Because of this, when input reading is to be conducted, ensure the appropriate wait time with the program. The configuration of the pull down resistor circuit is shown in Figure 3.3.3.

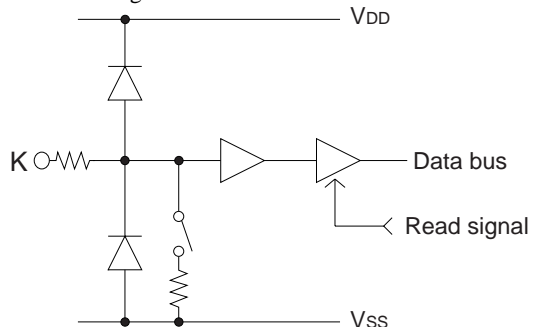


Fig. 3.3.3 Configuration of pull down resistor

## 5 R00 specification

```

*** OPTION NO.5 ***
--- R00 SPECIFICATION ---
      OUTPUT TYPE          1. D.C.
                          2. /BZ R00
                          3. /BZ R01
                          4. FOUT
PLEASE SELECT NO.(4) ? 4 

      F1                   1. 256 [HZ]
                          2. 512 [HZ]
                          3. 1024 [HZ]
                          4. 2048 [HZ]
                          5. 4096 [HZ]
PLEASE SELECT NO.(4) ? 4 

      F2                   1. 512 [HZ]
                          2. 1024 [HZ]
                          3. 2048 [HZ]
                          4. 4096 [HZ]
                          5. 8192 [HZ]
PLEASE SELECT NO.(4) ? 4 

      F3                   1. 1024 [HZ]
                          2. 2048 [HZ]
                          3. 4096 [HZ]
                          4. 8192 [HZ]
                          5. 16384 [HZ]
PLEASE SELECT NO.(4) ? 4 

      F4                   1. 2048 [HZ]
                          2. 4096 [HZ]
                          3. 8192 [HZ]
                          4. 16384 [HZ]
                          5. 32768 [HZ]
PLEASE SELECT NO.(4) ? 4 

      OUTPUT SPECIFICATION 1. COMPLEMENTARY
                          2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 

      OUTPUT TYPE          4. FOUT      SELECTED
      F1                   4. 2048 [HZ]  SELECTED
      F2                   4. 4096 [HZ]  SELECTED
      F3                   4. 8192 [HZ]  SELECTED
      F4                   4. 16384 [HZ]  SELECTED
      OUTPUT SPECIFICATION 1. COMPLEMENTARY  SELECTED
    
```



Fig. 3.3.4 Output waveform at DC output selection

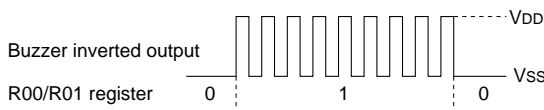


Fig. 3.3.5 Output waveform at buzzer inverted output selection

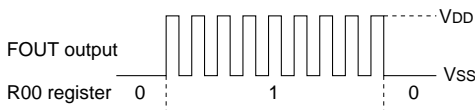


Fig. 3.3.6 Output waveform at FOUT output selection

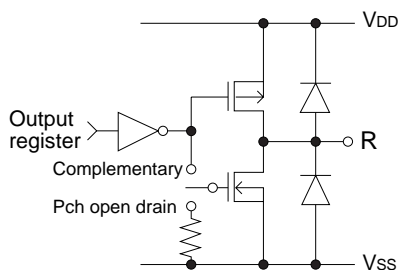


Fig. 3.3.7 Configuration of output circuit

Select the output specification for the R00 terminal. Either complementary output or Pch open drain output may be selected.

When "D.C." (DC output) is selected, R00 becomes a regular output port.

When "/BZ R00" (buzzer inverted output, R00 control) is selected, by writing "1" to the R00 register, clock with frequency specified through the software is generated from the R00 terminal. When "/BZ R01" (buzzer inverted output, R01 control) is selected, by writing "1" to the R01 register, clock with frequency specified through the software is generated from the R00 terminal.

When FOUT is selected, clock with frequency selected from the R00 terminal is generated by writing "1" to the R00 register.

When the DC output or buzzer inverted output is selected as the output type, the FOUT frequencies cannot be selected.

- When DC output is selected
  - When the R00 register is set to "1", the R00 terminal output goes high (VDD), and goes low (VSS) when set to "0".
  - Output waveform is shown in Figure 3.3.4.
- When buzzer inverted output (R00 control) is selected
  - When the R00 register is set to "1", 50% duty and VDD–VSS amplitude square wave is generated at the specified frequency by the software. When set to "0", the R00 terminal goes low (VSS). The clock phase when buzzer drive signal is output from R00 terminal is antiphase to that of the R01 terminal.
  - Output waveform is shown in Figure 3.3.5.
- When buzzer inverted output (R01 control) is selected
  - When the R01 register is set to "1", 50% duty and VDD–VSS amplitude square wave is generated at the specified frequency by the software. When set to "0", the R00 terminal goes low (VSS). The clock phase when buzzer drive signal is output from the R00 terminal is antiphase to that of the R01 terminal.
  - Output waveform is shown in Figure 3.3.5.
- When FOUT output is selected
  - When the R00 register is set to "1", 50% duty and VDD–VSS amplitude square wave is generated at the specified frequency. When set to "0", the FOUT terminal goes low (VSS).
  - The F1 to F4 FOUT frequencies are set by mask option. One of them is used by the software.
  - FOUT output is normally utilized to provide clock to other devices but since hazard occurs at the square wave breaks, great caution must be observed when using it.
  - Output waveform is shown in Figure 3.3.6.

The output circuit configuration is shown in Figure 3.3.7.

### 6 R01 specification

```

*** OPTION NO.6 ***
--- R01 SPECIFICATION ---
      OUTPUT TYPE          1. D.C.
                          2. BZ OUTPUT
PLEASE SELECT NO.(1) ? 2
      OUTPUT SPECIFICATION 1. COMPLEMENTARY
                          2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
      OUTPUT TYPE          2. BZ OUTPUT  SELECTED
      OUTPUT SPECIFICATION 1. COMPLEMENTARY SELECTED
    
```



Fig. 3.3.8 Output waveform at DC output selection

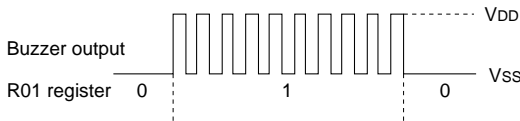


Fig. 3.3.9 Output waveform at buzzer output selection

Select the output specification for the R01 terminal. Either complementary output or Pch open drain output may be selected.

When "D.C." (DC output) is selected, R01 becomes a regular output port.

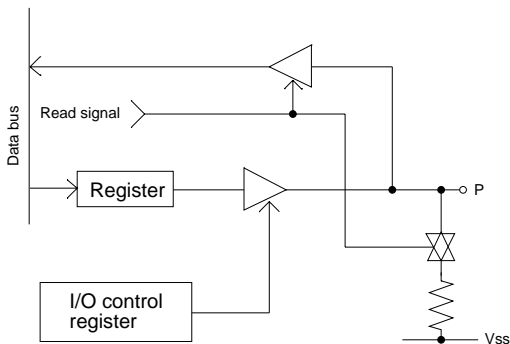
When "BZ OUTPUT" (buzzer output) is selected, by writing "1" to the R01 register, clock with frequency specified through the software is generated from the R01 terminal.

- When DC output is selected  
When the R01 register is set to "1", the R01 terminal output goes high (VDD), and goes low (VSS) when set to "0".  
Output waveform is shown in Figure 3.3.8.
- When buzzer output is selected  
When the R01 register is set to "1", 50% duty and VDD-VSS amplitude square wave is generated at the specified frequency by the software. When set to "0", the R01 terminal goes low (VSS). The clock phase when buzzer drive signal is output from the R01 terminal is antiphase to that of the R00 terminal.  
Output waveform is shown in Figure 3.3.9.

### 7 I/O port specification

```

*** OPTION NO.8 ***
--- I/O PORT OUTPUT SPECIFICATION ---
      P00          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
      P01          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
      P02          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
      P03          1. COMPLEMENTARY
                  2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1
      P00          1. COMPLEMENTARY  SELECTED
      P01          1. COMPLEMENTARY  SELECTED
      P02          1. COMPLEMENTARY  SELECTED
      P03          1. COMPLEMENTARY  SELECTED
    
```



Select the output specification to be used during I/O ports (P00-P03) output mode selection.

Either complementary output or Pch open drain output may be selected.

The circuit configuration of the output driver is the same as that of output ports (R00 shown in Figure 3.3.7).

Select complementary output for unused ports.

The I/O ports can control the input/output direction according to the IOC bit (FC address, D0 bit); at "1" and "0" settings, it is set to output port and input port, respectively.

The pull down resistor of this port is turned on by the read signal and is normally turned off to minimize leak current. Because of this, when the port is set for input, take care that a floating state does not occur in the terminal.

The circuit configuration of the I/O port is shown in Figure 3.3.10.

Fig. 3.3.10 Configuration of I/O port



## 8 LCD common duty and bias

```

*** OPTION NO.9 ***
--- LCD COMMON DUTY AND BIAS ---
      1. 1/4 DUTY 1/3 BIAS
      2. 1/3 DUTY 1/3 BIAS
      3. 1/2 DUTY 1/3 BIAS
      4. 1/4 DUTY 1/2 BIAS
      5. 1/3 DUTY 1/2 BIAS
      6. 1/2 DUTY 1/2 BIAS

PLEASE SELECT NO.(1) ? 1[ ]

      1. 1/4 DUTY 1/3 BIAS   SELECTED
    
```

Table 3.3.1 Common duty selection standard

Number of segments	Common duty
1-40	1/2
41-60	1/3
61-80	1/4

Select the common (drive) duty and bias.

When 1/2 duty is selected, up to 40 segments of LCD panel can be driven with 2 COM terminals and 20 SEG terminals. When 1/3 duty is selected, up to 60 segments can be driven with 3 COM terminals, and when 1/4 duty is selected, up to 80 segments with 4 COM terminals.

When 1/2 duty is selected, the COM0 and COM1 terminals are effective for COM output and the COM2 and COM3 terminals always output an off signal. When 1/3 duty is selected, the COM0 to COM2 terminals are effective and the COM3 terminal always outputs an off signal. Refer to Table 3.3.1 for common duty selection.

For the LCD drive bias, either 1/3 bias (drives LCD with 4 levels, VDD, VL1, VL2 and VL3) or 1/2 bias (drives LCD with 3 levels, VDD, VL1=VL2 and VL3) can be selected.

By selecting 1/2 bias, external elements can be minimized (see Figure 3.3.1).

Furthermore, when 1/2 bias is selected, be sure to short between the VL1 terminal and the VL2 terminal outside the IC.

Figures 3.3.11 and 3.3.12 show the drive waveforms of 1/3 bias driving and 1/2 bias driving, respectively.

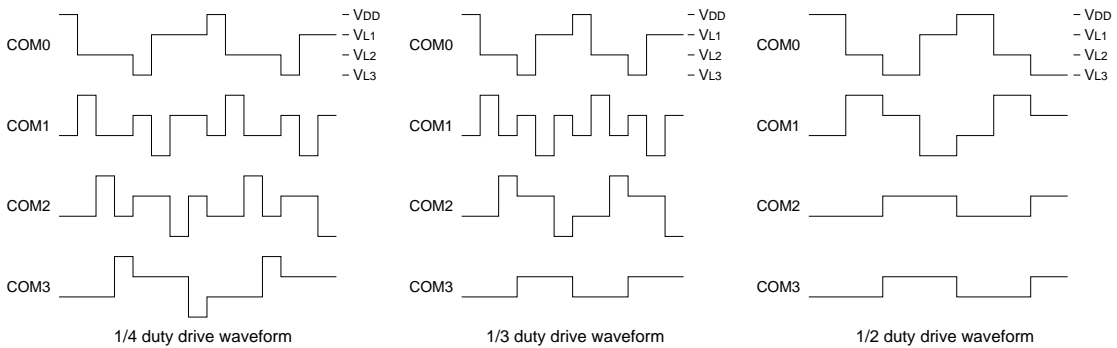


Fig. 3.3.11 Drive waveform from COM terminals (1/3 bias)

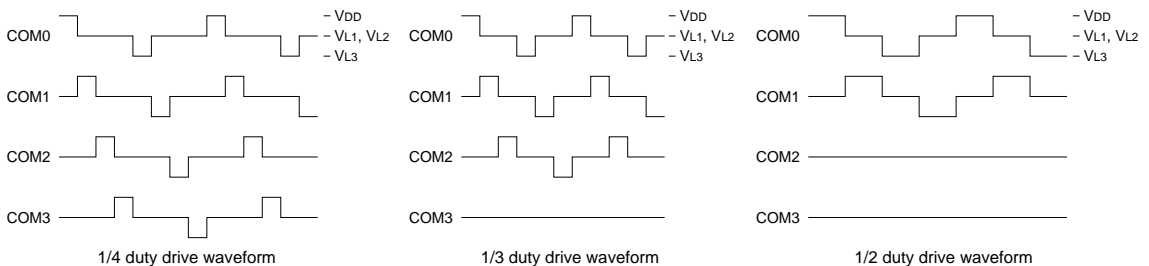


Fig. 3.3.12 Drive waveform from COM terminals (1/2 bias)

## 9 OSC1 system clock

```
*** OPTION NO.10 ***
--- OSC1 SYSTEM CLOCK ---
                        1. CRYSTAL
                        2. CR
PLEASE SELECT NO.(1) ? 1 
                        1. CRYSTAL  SELECTED
```

Select oscillation circuit that uses OSC1 and OSC2. To minimize external components, CR oscillation circuit would be suitable; to obtain a stable oscillation frequency, crystal oscillation circuit would be suitable.

When CR oscillation circuit is selected, only resistors are needed as external components since capacities are built-in.

On the other hand, when crystal oscillation circuit is selected, crystal oscillator and trimmer capacitor are needed as external components. Although when crystal oscillation circuit is selected, it is fixed at 32.768 kHz, when CR oscillation circuit is selected, frequency may be modified to a certain extent depending on the resistance of external components.

## 3.4 FOG6001 Quick Reference

### ■ Starting command and input/output files

**Execution file:** FOG6001.EXE

**Starting command:** **FOG6001**   indicates the Return key.

**Input file:** C001XXXF.DOC (Function option document file, when modifying)

**Output file:** C001XXXF.DOC (Function option document file)  
C001XXXF.HEX (Function option HEX file)

### ■ Display example

```

*** EOC6001 FUNCTION OPTION GENERATOR. --- Ver 1.00 ***
EEEEEEEEEE Pppppppp SSSSSSS 0000000 NNN NNN
EEEEEEEEEE Ppppppppp SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEEEEEEEEE Ppppppppp SSSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE Ppppppppp SSSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEEEEEEEEE PPP SSSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSSS 0000000 NNN NN

(C) COPYRIGHT 1996 SEIKO EPSON CORP.

THIS SOFTWARE MAKES NEXT FILES.

C001XXXF.HEX ... FUNCTION OPTION HEX FILE.
C001XXXF.DOC ... FUNCTION OPTION DOCUMENT FILE.

STRIKE ANY KEY.

```

```

*** EOC6001 USER'S OPTION SETTING. --- Ver 1.00 ***
CURRENT DATE IS 1996/07/26
PLEASE INPUT NEW DATE : 1996/07/28

```

```

*** OPERATION SELECT MENU ***
1. INPUT NEW FILE
2. EDIT FILE
3. RETURN TO DOS

PLEASE SELECT NO.?  


```

```

*** OPERATION SELECT MENU ***
1. INPUT NEW FILE
2. EDIT FILE
3. RETURN TO DOS

PLEASE SELECT NO.? 1
PLEASE INPUT FILE NAME? C0010A0 ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP. ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? FUJIMI PLANT ..(3)
? 281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN
? TEL 0266-61-1211
? FAX 0266-61-1273
? 

```

```

PLEASE INPUT FILE NAME? C0010A0
EXISTS OVERWRITE(Y/N)? N
PLEASE INPUT FILE NAME? C0010B0
PLEASE INPUT USER'S NAME?

```

#### Start-up message

When FOG6001 is started, the start-up message is displayed.

For "STRIKE ANY KEY.", press any key to advance the program execution.

To suspend execution, press the "CTRL" and "C" keys together: the sequence returns to the DOS command level.

#### Date input

Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/").

When not modifying the date, press the RETURN key "" to continue.

#### Operation selection menu

Enter a number from 1 to 3 to select a subsequent operation.

1. To set new function options.
2. To modify the document file.
3. To terminate FOG6001.

#### Setting new function options

Select "1" on the operation selection menu.

- (1) Enter the file name.
- (2) Enter the customer's company name.
- (3) Enter any comment.

(Within 50 characters x 10 lines)

Next, start function option setting from option No. 1.

In case a function option document file with the same name as the file name specified in the current drive exists, the user is asked whether overwriting is desired. Enter "Y" or "N" accordingly.

```

*** OPERATION SELECT MENU ***

    1. INPUT NEW FILE
    2. EDIT FILE
    3. RETURN TO DOS

PLEASE SELECT NO.? 2

*** SOURCE FILE(S) ***

C0010A0          C0010B0          C0010C0          ..(1)

PLEASE INPUT FILE NAME? C0010A0 ..(2)
PLEASE INPUT USER'S NAME?      ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)?          ..(4)
PLEASE INPUT EDIT NO.? 4      ..(5)
:
(Modifying function option settings)
:
PLEASE INPUT EDIT NO.? E
    
```

In step (1), if no modifiable source exists, the following message is displayed and the sequence returns to the operation selection menu.

```

*** SOURCE FILE(S) ***

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
    
```

In step (2), if the function option document file is not in the current drive, the following message is displayed, prompting entry of other file name.

```

PLEASE INPUT FILE NAME? C0010N0
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
    
```

```

*** OPTION NO.2 ***

--- MULTIPLE KEY ENTRY RESET ---

COMBINATION      1. Not Use
                 2. Use K00,K01
                 3. Use K00,K01,K02
                 4. Use K00,K01,K02,K03

PLEASE SELECT NO.(1) ? 2

COMBINATION      2. Use K00,K01  SELECTED
    
```

```

END OF OPTION SETTING.
DO YOU MAKE HEX FILE (Y/N) ? Y
*** OPTION EPROM SELECT MENU ***

    1. 27C64
    2. 27C128
    3. 27C256
    4. 27C512

PLEASE SELECT NO.? 2

    2. 27C128  SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

    1. INPUT NEW FILE
    2. EDIT FILE
    3. RETURN TO DOS

PLEASE SELECT NO.?
    
```

**Modifying function option settings**

Select "2" on the operation selection menu.

- (1) Will display the files on the current drive.
- (2) Enter the file name.
- (3) Enter the customer's company name.
- (4) Enter any comment.

Previously entered data can be used by pressing the RETURN key "" at (3) and (4).

- (5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected.

Enter "E" to end option setting. Then, move to the confirmation procedure for HEX file generation.

**Option selection**

The selections for each option correspond one to one to the option list. Enter the selection number. The value in parentheses ( ) indicates the default value, and is set when only the RETURN key "" is pressed.

In return, the confirmation is displayed. When you wish to modify previously set function options in the new setting process, enter "B" to return 1 step back to the previous function option setting operation.

**EPROM selection**

When setting function options setting is completed, the following message is output to ask the operator whether to generate the HEX file.

- (1) When debugging the program with EVA6001, HEX file is needed, so enter "Y". If "N" is entered, no HEX file is generated and only document file is generated.
- (2) For the option ROM selection menu displayed when "Y" is entered in Step (1), select the EPROM to be used for setting EVA6001 options.

When a series of operations are complete, the sequence returns to the operation selection menu.

## 3.5 Sample File

### ■ Example of function option document file

```

* E0C6001 FUNCTION OPTION DOCUMENT V 1.00
*
* FILE NAME      C0010A0F.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    1996/07/26
*
* COMMENT       FUJIMI PLANT
*               281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN
*               TEL 0266-61-1211
*               FAX 0266-61-1273
*
*
* OPTION NO.1
* < DEVICE TYPE & LCD POWER VREG >
*               E0C60L01 LCD 4.5V ----- SELECTED
OPT0101 02
OPT0103 03
*
* OPTION NO.2
* < MULTIPLE KEY ENTRY RESET >
*   KEY COMBINATION   USE  K00,K01, K02, K03 ----- SELECTED
OPT0201 04
*
* OPTION NO.3
* < INTERRUPT NOISE REJECTOR >
*   K00-K03          USE  ----- SELECTED
OPT0301 01
*
* OPTION NO.4
* < INPUT PORT PULL DOWN RESISTOR >
*   K00              WITH RESISTOR ----- SELECTED
*   K01              WITH RESISTOR ----- SELECTED
*   K02              WITH RESISTOR ----- SELECTED
*   K03              WITH RESISTOR ----- SELECTED
OPT0401 01
OPT0402 01
OPT0403 01
OPT0404 01
*
* OPTION NO.5
* < R00 SPECIFICATION >
*   OUTPUT TYPE     FOUT ----- SELECTED
*   F1              2048 (HZ) ----- SELECTED
*   F2              4096 (HZ) ----- SELECTED
*   F3              8192 (HZ) ----- SELECTED
*   F4              16384 (HZ) ----- SELECTED
*   OUTPUT SPECIFICATION  COMPLEMENTARY ----- SELECTED
OPT0501 04
OPT0502 04
OPT0503 04
OPT0504 04
OPT0505 04
OPT0506 01
*
* OPTION NO.6
* < R01 PORT OUTPUT SPECIFICATION >
*   OUTPUT TYPE     BZ OUTPUT ----- SELECTED
*   OUTPUT SPECIFICATION  COMPLEMENTARY ----- SELECTED
OPT0601 02
OPT0602 01
*

```

```

* OPTION NO.8
* < I/O PORT OUTPUT SPECIFICATION >
*   P00          COMPLEMENTARY ----- SELECTED
*   P01          COMPLEMENTARY ----- SELECTED
*   P02          COMPLEMENTARY ----- SELECTED
*   P03          COMPLEMENTARY ----- SELECTED
OPT0801 01
OPT0802 01
OPT0803 01
OPT0804 01
*
* OPTION NO.9
* < LCD COMMON DUTY AND BIAS >
*           1/4 DUTY 1/3 BIAS ----- SELECTED
OPT0901 01
*
* OPTION NO.10
* < OSC 1 SYSTEM CLOCK >
*           CRYSTAL ----- SELECTED
OPT1001 01
*
*
* SEIKO EPSON'S AREA
*
*
* OPTION NO.11
OPT1101 01
OPT1102 01
OPT1103 01
OPT1104 01
*
* OPTION NO.12
OPT1201 01
OPT1202 01
OPT1203 01
OPT1204 01
*
* OPTION NO.14
OPT1401 01
OPT1402 01
*
* OPTION NO.15
OPT1501 01
*
* OPTION NO.16
OPT1601 01
*
* OPTION NO.17
OPT1701 01
\\END

```

*Note* End mark "~~¥~~END" may be used instead of "\\END" depending on the PC used.  
(The code of \ and ¥ is 5CH.)

# 4 SEGMENT OPTION GENERATOR SOG6001

## 4.1 SOG6001 Outline

With the 4-bit single-chip E0C6001 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6001 according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG6001 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6001, the E0C6001 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA6001) segment option ROM is simultaneously generated with the data file.

☞ The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

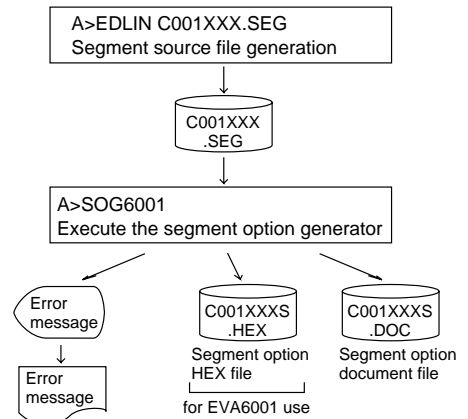


Fig. 4.1.1 SOG6001 execution flow

## 4.2 Option List

TERMINAL NAME	ADDRESS												OUTPUT SPECIFICATION		
	COM0			COM1			COM2			COM3					
	H	L	D	H	L	D	H	L	D	H	L	D			
SEG0														SEG output	
SEG1														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG2														SEG output	
SEG3														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG4														SEG output	
SEG5														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG6														SEG output	
SEG7														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG8														SEG output	
SEG9														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG10														SEG output	
SEG11														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG12														SEG output	
SEG13														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG14														SEG output	
SEG15														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG16														SEG output	
SEG17														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
SEG18														SEG output	
SEG19														DC output <input type="checkbox"/> C <input type="checkbox"/> P	
Legend:	<ADDRESS>												<OUTPUT SPECIFICATION>		
	H: High order address, L: Low order address												C: Complementary output		
	D: Data bit												P: Pch open drain output		

Note:

1. Even if there are unused areas, set "---" (hyphens) such that there are no blank columns.
2. When DC output is selected, the display memory of the COM0 column becomes effective.

### 4.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG19, segment output and DC output can be selected in units of two terminals. When used for liquid crystal panel drives, select segment output; when used as regular output port, select DC output. When DC output is selected, either complementary output or Pch open drain output may further be selected.

However, for segment output ports that will not be used, select segment output.

Refer to the "E0C62 Family Development Tool Reference Manual" for the segment option source file creation.

■ **When segment output is selected**

The segment output port has a segment decoder built-in, and the data bit of the optional address in the segment memory area (090H–0AFH) can be allocated to the optional segment. With this, up to 80 segments (60 segments when 1/3 duty is selected or 40 segments when 1/2 duty is selected) of liquid crystal panel could be driven.

The segment memory may be allocated only one segment and multiple setting is not possible.

The allocated segment displays when the bit for this segment memory is set to "1", and goes out when bit is set to "0".

Segment allocation is set to H for high address (9–A), to L for low address (0–F), and to D for data bit (0–3) and are recorded in their respective column in the option list. For segment ports that will not be used, write "---" (hyphen) in the H, L, and D columns of COM0–COM3.

**Examples**

- When 1/4 duty is selected

```
0  900  901  902  903  S
1  910  911  912  913  S
```

- When 1/3 duty is selected

```
0  900  901  902  ---  S
1  910  911  912  ---  S
```

- When 1/2 duty is selected

```
0  900  901  ---  ---  S
1  910  911  ---  ---  S
```

■ **When DC output is selected**

The DC output can be selected in units of two terminals and up to 20 terminals may be allocated for DC output. Also, either complementary output or Pch open drain output is likewise selected in units of two terminals. When the bit for the selected segment memory is set to "1", the segment output port goes high (VDD), and goes low (VSS) when set to "0". Segment allocation is the same as when segment output is selected but for the while the segment memory allocated to COM1–COM3 becomes ineffective. Write three hyphens ("---") in the COM1–COM3 columns in the option list.

**Example**

- When complementary output is set to SEG16 and SEG17, and Pch open drain output is set to SEG18 and SEG19.

```
16  A00  ---  ---  ---  C
17  A10  ---  ---  ---  C
18  A20  ---  ---  ---  P
19  A30  ---  ---  ---  P
```

*Note Only complementary output is enabled as the DC output of the SEG ports of EVA6001. Therefore, complementary output is enabled even if Pch open drain output is selected. Respond to it by adding external circuits as required.*



## 4.4 SOG6001 Quick Reference

### ■ Starting command and input/output files

**Execution file:** SOG6001.EXE

\_ indicates a blank.

**Starting command:** SOG6001\_ [-H]

indicates the Return key.

A parameter enclosed by [ ] can be omitted.

**Option:** -H: Specifies the segment option document file for input file of SOG6001.

**Input file:** C001XXX.SEG (Segment option source file)  
C001XXXS.DOC (Segment option document file, when -H option use)

**Output file:** C001XXXS.DOC (Segment option document file)  
C001XXXS.HEX (Segment option HEX file)

### ■ Display example

```

*** E0C6001 SEGMENT OPTION GENERATOR. --- Ver 1.00 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PPPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPPP SSSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEEEEEEEEE PPF SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPF SSSSSSS 00000000 NNN NN

(C) COPYRIGHT 1996 SEIKO EPSON CORP.

SEGMENT OPTION SOURCE FILE NAME IS " C001XXX.SEG "

THIS SOFTWARE MAKES NEXT FILES.

C001XXXS.HEX ... SEGMENT OPTION HEX FILE.
C001XXXS.DOC ... SEGMENT OPTION DOCUMENT FILE.

STRIKE ANY KEY.

```

```

*** E0C6001 USER'S OPTION SETTING. --- Ver 1.00 ***
CURRENT DATE IS 96/07/26
PLEASE INPUT NEW DATE : 96/07/28 

```

```

*** SOURCE FILE(S) ***
C0010A0 C0010B0 C0010C0 ..(1)
PLEASE INPUT SEGMENT OPTION FILE NAME? C0010A0  ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.  ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? FUJIMI PLANT  ..(4)
? 281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN 
? TEL 0266-61-1211 
? FAX 0266-61-1273 
? 

```

```

*** SOURCE FILE(S) ***
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(5) -H option not use
*** SOURCE FILE(S) ***
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(6) -H option use

```

```

PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C0010N0 
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(7) -H option not use
PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C0010N0 
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(8) -H option use

```

#### Start-up message

When SOG6001 is started, the start-up message is displayed.

For "STRIKE ANY KEY.", press any key to advance the program execution.

To suspend execution, press the "CTRL" and "C" keys together: the sequence returns to the DOS command level.

#### Date input

Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/"). When not modifying the date, press the RETURN key "" to continue.

#### Input file selection

- (1) Will display the files on the current drive.
- (2) Enter the file name.
- (3) Enter the customer's company name.
- (4) Enter any comment.

(Within 50 characters x 10 lines)

Then, move to the confirmation procedure for HEX file generation.

In step (1), if no modifiable source exists, an error message (5) or (6) will be displayed and the program will be terminated. In step (2), if the specified file name is not found in the current drive, an error message (7) or (8) is displayed, prompting entry of other file name.

```

END OF OPTION SETTING.
DO YOU MAKE HEX FILE (Y/N) ? Y  ..(1)

*** OPTION EPROM SELECT MENU ***

1. 27C64
2. 27C128
3. 27C256
4. 27C512

PLEASE SELECT NO.? 2  ..(2)

2. 27C128 SELECTED

MAKING FILE IS COMPLETED.
    
```

**EPROM selection**

When selecting file is completed, the following message is output to ask the operator whether to generate the HEX file.

- (1) When debugging the program with EVA6001, HEX file is needed, so enter "Y ". If "N " is entered, no HEX file is generated and only document file is generated.
- (2) For the option ROM selection menu displayed when "Y " is entered in Step (1), select the EPROM to be used for setting EVA6001 options.

When a series of operations are complete, the SOG6001 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6001 program will be terminated.

■ **Error messages**

Error message	Explanation
S (Syntax Error)	The data was written in an invalid format.
N (Segment No. Select Error)	The segment number outside the specificable range was specified.
R (RAM Address Select Error)	The segment memory address or data bit outside the specificable range was specified.
D (Duplication Error)	The same data (SEG port No., segment memory address, or data bit) was specified more then once.
Out Port Set Error	The output specifications were not set in units of two ports.

## 4.5 Sample Files

### ■ Example of segment option source file

```
; C0010A0.SEG, VER.1.00
; EVA6001 LCD SEGMENT DECODE TABLE
;
0   900  901  902  903  S
1   910  911  912  913  S
2   920  921  922  923  S
3   930  931  932  933  S
4   940  941  942  943  S
5   950  951  952  953  S
6   960  961  962  963  S
7   970  971  972  973  S
8   980  981  982  983  S
9   990  991  992  993  S
10  9A0  9A1  9A2  9A3  S
11  9B0  9B1  9B2  9B3  S
12  9C0  9C1  9C2  9C3  S
13  9D0  9D1  9D2  9D3  S
14  9E0  9E1  9E2  9E3  S
15  9F0  9F1  9F2  9F3  S
16  A00  A01  A02  A03  S
17  A10  A11  A12  A13  S
18  AE3  ---  ---  ---  C
19  AF3  ---  ---  ---  C
```

### ■ Example of segment option document file

```
* E0C6001 SEGMENT OPTION DOCUMENT V 1.00
*
* FILE NAME      C0010A0S.DOC
* USER'S NAME    SEIKO EPSON CORP.
* INPUT DATE     96/07/26
* COMMENT        FUJIMI PLANT
*                281 FUJIMI SUWA-GUN NAGANO-KEN 399-0200 JAPAN
*                TEL 0266-61-1211
*                FAX 0266-61-1273
*
*
* OPTION NO.11
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
0   900  901  902  903  S
1   910  911  912  913  S
2   920  921  922  923  S
3   930  931  932  933  S
4   940  941  942  943  S
5   950  951  952  953  S
6   960  961  962  963  S
7   970  971  972  973  S
8   980  981  982  983  S
9   990  991  992  993  S
10  9A0  9A1  9A2  9A3  S
11  9B0  9B1  9B2  9B3  S
12  9C0  9C1  9C2  9C3  S
13  9D0  9D1  9D2  9D3  S
14  9E0  9E1  9E2  9E3  S
15  9F0  9F1  9F2  9F3  S
16  A00  A01  A02  A03  S
17  A10  A11  A12  A13  S
18  AE3  A20  A21  A22  C
19  AF3  A23  A30  A31  C
\\END
```

*Note* End mark "~~¥~~END" may be used instead of "\\END" depending on the PC used. (The code of \ and ¥ is 5CH.)

# 5 ICE CONTROL SOFTWARE ICS6001

## 5.1 ICS6001 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA6001 and performs real time target system evaluation and debugging by passing through the RS-232C from the host computer and controlling it. The operation on the host computer side and ICE6200 control is done through the ICE Control Software ICS6001.

The ICS6001 has a set of numerous and highly functional emulation commands which provide sophisticated break function, on-the-fly data display, history display, etc., and so perform a higher level of debugging.

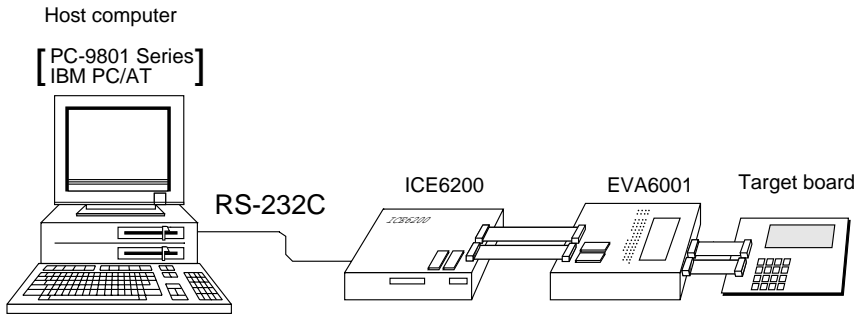


Fig. 5.1.1  
Debugging system  
using ICE6200

The functions of the ICE6200 and commands are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

## 5.2 ICS6001 Restrictions

Take the following precautions when using the ICS6001.

### ■ ROM Area

The ROM area is limited to a maximum address of 3FFH. Assigning data above the 3FFH address causes an error.

### ■ RAM Area

The RAM area is limited to a maximum address of 0FFH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

Unused area: 050H to 08FH, 0B0H to 0DFH

Memory 090H to 0AFH is display memory; 0E0H to 0FFH is I/O memory.  
(Refer to the "E0C6001 Technical Manual" for details.)

### ■ Undefined Code

The instructions below are not specified for the E0C6001 and so cannot be used.

SLP  
 PUSH XP            POP XP            LD XP,r            LD r,XP  
 PUSH YP            POP YP            LD YP,r            LD r,YP

### ■ OPTLD Command

In the ICS6001, OPTLD command cannot be used.

## 5.3 ICS6001 Quick Reference

### ■ Starting command and input/output files

␣ indicates the Return key.

**Execution file:** ICS6001.BAT (ICS6001J.EXE) ... for MS-DOS  
ICS6001B.BAT (ICS6001W.EXE) ... for PC-DOS

**Starting command:** ICS6001 (ICS6001J)␣ ... for MS-DOS  
ICS6001B (ICS6001W)␣ ... for PC-DOS

**Input file:** C001XXXL.HEX (Object file, low-order)  
C001XXXH.HEX (Object file, high-order)  
C001XXXD.HEX (Data RAM file)  
C001XXXC.HEX (Control file)

**Output file:** C001XXXL.HEX (Object file, low-order)  
C001XXXH.HEX (Object file, high-order)  
C001XXXD.HEX (Data RAM file)  
C001XXXC.HEX (Control file)

### ■ Display example

```

*** E0C6001 ICE CONTROL SOFTWARE. --- Ver 1.00 ***
EEEEEEEEEE PPPPPPPP SSSSSSS 00000000 NNN NNN
EEEEEEEEEE PPPPPPPP SSS SSSS 000 000 NNNN NNN
EEE PPP PPP SSS SSS 000 000 NNNNN NNN
EEE PPP PPP SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPP SSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS 000 000 NNN NNNNN
EEE PPP SSS SSS 000 000 NNN NNNNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

```

(C) COPYRIGHT 1996 SEIKO EPSON CORP.

```

* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *
#

```

#### Start-up message

When ICS6001 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6001 commands are awaited when the program is properly loaded and the # mark is displayed.

Debugging can be done by entering command after the # mark.

The ICS6001 program is terminated by entering the Q (Quit) command.

**Note** Confirm that the cables connected properly, then operate the ICS6001.

### ■ Error messages

Error message	Meaning	Recover procedure
* COMMUNICATION ERROR OR ICE NOT READY *	ICE6200 is disconnected or power is OFF.	Switch OFF the host power supply, connect cable, and reapply power. Or switch ON power to ICE6200.
* TARGET DOWN (1) *	Evaluation board is disconnected. (Check at power ON)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200.
* TARGET DOWN (2) *	Evaluation board is disconnected. (Check at command execution)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200.
* UNDEFINED PROGRAM CODE EXIST *	Undefined code is detected in the program loaded from ROM or FD.	Convert ROM and FD data with the cross assembler, then restart the ICE6200.
* COMMAND ERROR *	A miss occurs by command input.	Reenter the proper command.
(No response after power on)	The ICE-to-HOST cable is disconnected on the host side.	Switch OFF the host power supply, connect cable, and reapply power.

■ ICE6200 commands

Item No.	Function	Command Format	Outline of Operation
1	Assemble	#A,a [↵]	Assemble command mnemonic code and store at address "a"
2	Disassemble	#L,a1,a2 [↵]	Contents of addresses a1 to a2 are disassembled and displayed
3	Dump	#DP,a1,a2 [↵]	Contents of program area a1 to a2 are displayed
		#DD,a1,a2 [↵]	Content of data area a1 to a2 are displayed
4	Fill	#FP,a1,a2,d [↵]	Data d is set in addresses a1 to a2 (program area)
		#FD,a1,a2,d [↵]	Data d is set in addresses a1 to a2 (data area)
5	Set Run Mode	#G,a [↵]	Program is executed from the "a" address
		#TIM [↵]	Execution time and step counter selection
		#OTF [↵]	On-the-fly display selection
6	Trace	#T,a,n [↵]	Executes program while displaying results of step instruction from "a" address
		#U,a,n [↵]	Displays only the final step of #T,a,n
7	Break	#BA,a [↵]	Sets Break at program address "a"
		#BAR,a [↵]	Breakpoint is canceled
		#BD [↵]	Break condition is set for data RAM
		#BDR [↵]	Breakpoint is canceled
		#BR [↵]	Break condition is set for EVA6001 CPU internal registers
		#BRR [↵]	Breakpoint is canceled
		#BM [↵]	Combined break conditions set for program data RAM address and registers
		#BMR [↵]	Cancel combined break conditions for program data ROM address and registers
		#BRES [↵]	All break conditions canceled
		#BC [↵]	Break condition displayed
		#BE [↵]	Enter break enable mode
		#BSYN [↵]	Enter break disable mode
		#BT [↵]	Set break stop/trace modes
#BRKSEL,REM [↵]	Set BA condition clear/remain modes		
8	Move	#MP,a1,a2,a3 [↵]	Contents of program area addresses a1 to a2 are moved to addresses a3 and after
		#MD,a1,a2,a3 [↵]	Contents of data area addresses a1 to a2 are moved to addresses a3 and after
9	Data Set	#SP,a [↵]	Data from program area address "a" are written to memory
		#SD,a [↵]	Data from data area address "a" are written to memory
10	Change CPU Internal Registers	#DR [↵]	Display EVA6001 CPU internal registers
		#SR [↵]	Set EVA6001 CPU internal registers
		#I [↵]	Reset EVA6001 CPU
		#DXY [↵]	Display X, Y, MX and MY
		#SXY [↵]	Set data for X and Y display and MX, MY

Item No.	Function	Command Format	Outline of Operation
11	History	#H,p1,p2 <input type="checkbox"/>	Display history data for pointer 1 and pointer 2
		#HB <input type="checkbox"/>	Display upstream history data
		#HG <input type="checkbox"/>	Display 21 line history data
		#HP <input type="checkbox"/>	Display history pointer
		#HPS,a <input type="checkbox"/>	Set history pointer
		#HC,S/C/E <input type="checkbox"/>	Sets up the history information acquisition before (S), before/after (C) and after (E)
		#HA,a1,a2 <input type="checkbox"/>	Sets up the history information acquisition from program area a1 to a2
		#HAR,a1,a2 <input type="checkbox"/>	Sets up the prohibition of the history information acquisition from program area a1 to a2
		#HAD <input type="checkbox"/>	Indicates history acquisition program area
		#HS,a <input type="checkbox"/>	Retrieves and indicates the history information which executed a program address "a"
		#HSW,a <input type="checkbox"/>	Retrieves and indicates the history information which wrote or read the data area address "a"
12	File	#RF,file <input type="checkbox"/>	Move program file to memory
		#RFD,file <input type="checkbox"/>	Move data file to memory
		#VF,file <input type="checkbox"/>	Compare program file and contents of memory
		#VFD,file <input type="checkbox"/>	Compare data file and contents of memory
		#WF,file <input type="checkbox"/>	Save contents of memory to program file
		#WFD,file <input type="checkbox"/>	Save contents of memory to data file
		#CL,file <input type="checkbox"/>	Load ICE6200 set condition from file
		#CS,file <input type="checkbox"/>	Save ICE6200 set condition to file
13	Coverage	#CVD <input type="checkbox"/>	Indicates coverage information
		#CVR <input type="checkbox"/>	Clears coverage information
14	ROM Access	#RP <input type="checkbox"/>	Move contents of ROM to program memory
		#VP <input type="checkbox"/>	Compare contents of ROM with contents of program memory
		#ROM <input type="checkbox"/>	Set ROM type
15	Terminate ICE	#Q <input type="checkbox"/>	Terminate ICE and return to operating system control
16	Command Display	#HELP <input type="checkbox"/>	Display ICE6200 instruction
17	Self Diagnosis	#CHK <input type="checkbox"/>	Report results of ICE6200 self diagnostic test

means press the RETURN key.

# 6 MASK DATA CHECKER MDC6001

## 6.1 MDC6001 Outline

The Mask Data Checker MDC6001 is a software tool which checks the program data (C001XXXH.HEX and C001XXXL.HEX) and option data (C001XXXF.DOC and C001XXXS.DOC) created by the user and creates the data file (C6001XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6001 has the capability to restore the generated data file (C6001XXX.PAn) to the original file format.

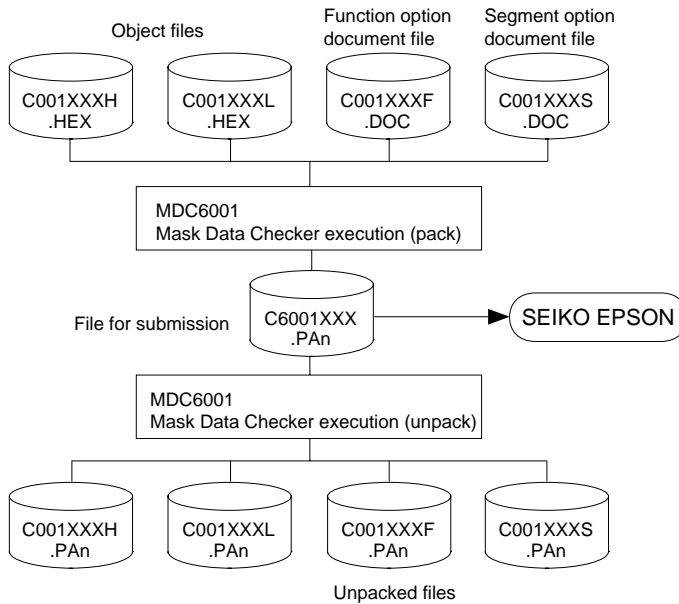


Fig. 6.1.1  
MDC6001 execution flow

☞ The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

## 6.2 MDC6001 Quick Reference

### ■ Starting command and input/output files

**Execution file:** MDC6001.EXE

**Starting command:** **MDC6001**   indicates the Return key.

<b>Input file:</b>	C001XXXL.HEX (Object file, low-order)	] When packing
	C001XXXH.HEX (Object file, high-order)	
	C001XXXF.DOC (Function option document file)	
	C001XXXS.DOC (Segment option document file)	
	C6001XXX.PAn (Packed file)	

<b>Output file:</b>	C6001XXX.PAn (Packed file)	] When packing
	C001XXXL.PAn (Object file, low-order)	] When unpacking
	C001XXXH.PAn (Object file, high-order)	
	C001XXXF.PAn (Function option document file)	
	C001XXXS.PAn (Segment option document file)	



## ■ Display examples

```

*** E0C6001 PACK / UNPACK PROGRAM Ver 1.00 ***

EEEEEEEEEE PPPPPPPP SSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPPPP SSS SSSS OOO OOO NNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNN NNN
EEE PPP PPP SSS OOO OOO NNNNNN NNN
EEEEEEEEEE PPPPPPPPPP SSSSSS OOO OOO NNN NNN NNN
EEEEEEEEEE PPPPPPPPPP SSSS OOO OOO NNN NNNNNN
EEE PPP SSS OOO OOO NNN NNNNNN
EEE PPP SSS SSS OOO OOO NNN NNNN
EEEEEEEEEE PPP SSS SSS OOO OOO NNN NNN
EEEEEEEEEE PPP SSSSSS OOOOOOOO NNN NN

(C) COPYRIGHT 1996 SEIKO EPSON CORP.

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

PLEASE SELECT NO.?
```

```

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

PLEASE SELECT NO.? 1  ... (1)

C001XXXH.HEX -----+
C001XXXL.HEX -----+
C001XXXF.DOC -----+----- C6001XXX.PAn (PACK FILE)
C001XXXS.DOC -----+

PLEASE INPUT PACK FILE NAME (C6001XXX.PAn) ? C60010A0.PA0  ... (2)

C0010A0H.HEX -----+
C0010A0L.HEX -----+
C0010A0F.DOC -----+----- C60010A0.PA0
C0010A0S.DOC -----+
```

**Note** Don't use the data generated with the `-N` option of the Cross Assembler (ASM6001) as program data. If the program data generated with the `-N` option of the Cross Assembler is packed, undefined program area is filled with FFH code. In this case, following message is displayed.

```
WARNING: FILLED <file_name> FILE WITH FFH.
```

```

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

PLEASE SELECT NO.? 2  ... (1)

PLEASE INPUT PACKED FILE NAME (C6001XXX.PAn) ? C60010A0.PA0  ... (2)

C60010A0.PA0 -----+----- C0010A0H.PA0
|-----+----- C0010A0L.PA0
|-----+----- C0010A0F.PA0
|-----+----- C0010A0S.PA0
```

### Start-up message

When MDC6001 is started, the start-up message and operation menu are displayed. Here, the user is prompted to select operation options.

### Packing of data

(1) Select "1" in the operation menu.

(2) Enter the file name.

After submitting the data to Seiko Epson and there is a need to re-submit the data, increase the numeric value of "n" by one when the input is made.

(Example: When re-submitting data after "C6001XXX.PA0" has been submitted, the pack file name should be entered as "C6001XXX.PA1".)

With this, the mask file (C6001XXX.PAn) is generated, and the MDC6001 program will be terminated.

Submit this file to Seiko Epson.

### Unpacking of data

(1) Select "2" in the operation menu.

(2) Enter the packed file name.

With this, the mask data file (C6001XXX.PAn) is restored to the original file format, and the MDC6001 program will be terminated.

Since the extension of the file name remains as "PAn", it must be renamed back to its original form ("HEX" and "DOC") in order to re-debug or modify the restored file.

## ■ Error messages

### *Program data error*

Error Message	Explanation
1. HEX DATA ERROR : NOT COLON.	There is no colon.
2. HEX DATA ERROR : DATA LENGTH. (NOT 00-20h)	The data length of 1 line is not in the 00-20H range.
3. HEX DATA ERROR : ADDRESS.	The address is beyond the valid range of the program ROM.
4. HEX DATA ERROR : RECORD TYPE. (NOT 00)	The record type of 1 line is not 00.
5. HEX DATA ERROR : DATA. (NOT 00-FFh)	The data is not in the range between 00H and 0FFH.
6. HEX DATA ERROR : TOO MANY DATA IN ONE LINE.	There are too many data in 1 line.
7. HEX DATA ERROR : CHECK SUM.	The checksum is not correct.
8. HEX DATA ERROR : END MARK.	The end mark is not : 00000001FF.
9. HEX DATA ERROR : DUPLICATE.	There is duplicate definition of data in the same address.

### *Function option data error*

Error Message	Explanation
1. OPTION DATA ERROR : START MARK.	The start mark is not "\OPTION". (during unpacking) *
2. OPTION DATA ERROR : OPTION NUMBER.	The option number is not correct.
3. OPTION DATA ERROR : SELECT NUMBER.	The option selection number is not correct.
4. OPTION DATA ERROR : END MARK.	The end mark is not "\\END" (packing) or "\END" (unpacking).*

### *Segment option data error*

Error Message	Explanation
1. SEGMENT DATA ERROR : START MARK.	The start mark is not "\SEGMENT". (during unpacking) *
2. SEGMENT DATA ERROR : DATA.	The segment data is not correct.
3. SEGMENT DATA ERROR : SEGMENT NUMBER.	The SEG No. is not correct.
4. SEGMENT DATA ERROR : SPEC.	The output specification of the SEG terminal is not correct.
5. SEGMENT DATA ERROR : END MARK.	The end mark is not "\\END" (packing) or "\END" (unpacking).*

### *File error*

Error Message	Explanation
1. <File_name> FILE IS NOT FOUND.	The file is not found or the file number set in CONFIG.SYS is less than 10.
2. PACK FILE NAME (File_name) ERROR.	The packed input format for the file name is wrong.
3. PACKED FILE NAME (File_name) ERROR.	The unpacked input format for the file name is wrong.

### *System error*

Error Message	Explanation
1. DIRECTORY FULL.	The directory is full.
2. DISK WRITE ERROR.	Writing on the disk is failed.

\* \ sometimes appears as ¥, depending on the personal computer being used.

# APPENDIX A. E0C6001 INSTRUCTION SET

Classification	Mnemonic	Operand	Operation Code						Flag			Clock	Operation							
			B	A	9	8	7	6	5	4	3			2	1	0	I	D	Z	C
Branch instructions	PSET	p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0					5	NBP ← p4, NPP ← p3~p0
	JP	s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0
		C, s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=1
		NC, s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if C=0
		Z, s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=1
		NZ, s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB ← NBP, PCP ← NPP, PCS ← s7~s0 if Z=0
	JPBA		1	1	1	1	1	1	1	0	1	0	0	0					5	PCB ← NBP, PCP ← NPP, PCSH ← B, PCSL ← A
	CALL	s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← NPP, PCS ← s7~s0
	CALZ	s	0	1	0	1	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1) ← PCP, M(SP-2) ← PCSH, M(SP-3) ← PCSL+1 SP ← SP-3, PCP ← 0, PCS ← s7~s0
	RET		1	1	1	1	1	1	0	1	1	1	1	1					7	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3
RETS		1	1	1	1	1	1	0	1	1	1	1	0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, PC ← PC+1	
RETD	l	0	0	0	1	l7	l6	l5	l4	l3	l2	l1	l0					12	PCSL ← M(SP), PCSH ← M(SP+1), PCP ← M(SP+2) SP ← SP+3, M(X) ← l3~l0, M(X+1) ← l7~l4, X ← X+2	
System control instructions	NOP5		1	1	1	1	1	1	1	1	1	0	1	1					5	No operation (5 clock cycles)
	NOP7		1	1	1	1	1	1	1	1	1	1	1	1					7	No operation (7 clock cycles)
	HALT		1	1	1	1	1	1	1	1	1	0	0	0					5	Halt (stop clock)
Index operation instructions	INC	X	1	1	1	0	1	1	1	0	0	0	0	0					5	X ← X+1
		Y	1	1	1	0	1	1	1	1	0	0	0	0					5	Y ← Y+1
	LD	X, x	1	0	1	1	x7	x6	x5	x4	x3	x2	x1	x0					5	XH ← x7~x4, XL ← x3~x0
		Y, y	1	0	0	0	y7	y6	y5	y4	y3	y2	y1	y0					5	YH ← y7~y4, YL ← y3~y0
		XH, r	1	1	1	0	1	0	0	0	0	1	r1	r0					5	XH ← r
		XL, r	1	1	1	0	1	0	0	0	1	0	r1	r0					5	XL ← r
		YH, r	1	1	1	0	1	0	0	1	0	1	r1	r0					5	YH ← r
		YL, r	1	1	1	0	1	0	0	1	1	0	r1	r0					5	YL ← r
		r, XH	1	1	1	0	1	0	1	0	0	1	r1	r0					5	r ← XH
		r, XL	1	1	1	0	1	0	1	0	1	0	r1	r0					5	r ← XL
		r, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r ← YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r ← YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0		↓	↓		7	XH ← XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0		↑	↑		7	XL ← XL+i3~i0+C
YH, i		1	0	1	0	0	0	1	0	i3	i2	i1	i0		↓	↓		7	YH ← YH+i3~i0+C	
YL, i		1	0	1	0	0	0	1	1	i3	i2	i1	i0		↑	↑		7	YL ← YL+i3~i0+C	

Classification	Mnemonic	Operand	Operation Code								Flag			Clock	Operation				
			B	A	9	8	7	6	5	4	3	2	1			0	I	D	Z
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0	↓	↓		7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0	↓	↓		7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0	↓	↓		7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0	↓	↓		7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0				5	r ← i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0				5	r ← q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0				5	A ← M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0				5	B ← M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0				5	M(n3~n0) ← A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0				5	M(n3~n0) ← B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0				5	M(X) ← i3~i0, X ← X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0				5	r ← q, X ← X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0				5	M(Y) ← i3~i0, Y ← Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0				5	r ← q, Y ← Y+1
LBPX	MX, l	1	0	0	1	l7	l6	l5	l4	l3	l2	l1	l0				5	M(X) ← l3~l0, M(X+1) ← l7~l4, X ← X+2	
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i3	i2	i1	i0	↑	↑	↑	7	F ← F∨i3~i0
	RST	F, i	1	1	1	1	0	1	0	1	i3	i2	i1	i0	↓	↓	↓	7	F ← F∧i3~i0
	SCF		1	1	1	1	0	1	0	0	0	0	0	1	↑			7	C ← 1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0	↓			7	C ← 0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0	↑			7	Z ← 1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1	↓			7	Z ← 0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0	↑			7	D ← 1 (Decimal Adjuster ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1	↓			7	D ← 0 (Decimal Adjuster OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	↑			7	I ← 1 (Enables Interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	↓			7	I ← 0 (Disables Interrupt)
Stack operation instructions	INC	SP	1	1	1	1	1	1	0	1	1	0	1	1				5	SP ← SP+1
	DEC	SP	1	1	1	1	1	1	0	0	1	0	1	1				5	SP ← SP-1
	PUSH	r	1	1	1	1	1	1	0	0	0	0	r1	r0				5	SP ← SP-1, M(SP) ← r
		XH	1	1	1	1	1	1	0	0	0	1	0	1				5	SP ← SP-1, M(SP) ← XH
		XL	1	1	1	1	1	1	0	0	0	1	1	0				5	SP ← SP-1, M(SP) ← XL
		YH	1	1	1	1	1	1	0	0	1	0	0	0				5	SP ← SP-1, M(SP) ← YH
		YL	1	1	1	1	1	1	0	0	1	0	0	1				5	SP ← SP-1, M(SP) ← YL
		F	1	1	1	1	1	1	0	0	1	0	1	0				5	SP ← SP-1, M(SP) ← F
	POP	r	1	1	1	1	1	1	0	1	0	0	r1	r0				5	r ← M(SP), SP ← SP+1
XH		1	1	1	1	1	1	0	1	0	1	0	1				5	XH ← M(SP), SP ← SP+1	
XL		1	1	1	1	1	1	0	1	0	1	1	0				5	XL ← M(SP), SP ← SP+1	

Classification	Mnemonic	Operand	Operation Code							Flag			Clock	Operation						
			B	A	9	8	7	6	5	4	3	2			1	0	I	D	Z	C
Stack operation instructions	POP	YH	1	1	1	1	1	1	0	1	1	0	0	0					5	$YH \leftarrow M(SP), SP \leftarrow SP+1$
		YL	1	1	1	1	1	1	0	1	1	0	0	1					5	$YL \leftarrow M(SP), SP \leftarrow SP+1$
		F	1	1	1	1	1	1	0	1	1	0	1	0	$\uparrow \downarrow \uparrow \downarrow$	$\uparrow \downarrow \uparrow \downarrow$	$\uparrow \downarrow \uparrow \downarrow$	$\uparrow \downarrow \uparrow \downarrow$	5	$F \leftarrow M(SP), SP \leftarrow SP+1$
	LD	SPH, r	1	1	1	1	1	1	1	0	0	0	r1	r0					5	$SPH \leftarrow r$
		SPL, r	1	1	1	1	1	1	1	1	0	0	r1	r0					5	$SPL \leftarrow r$
		r, SPH	1	1	1	1	1	1	1	0	0	1	r1	r0					5	$r \leftarrow SPH$
		r, SPL	1	1	1	1	1	1	1	1	0	1	r1	r0					5	$r \leftarrow SPL$
Arithmetic instructions	ADD	r, i	1	1	0	0	0	0	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r+i3-i0$
		r, q	1	0	1	0	1	0	0	0	r1	r0	q1	q0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r+q$
	ADC	r, i	1	1	0	0	0	1	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r+i3-i0+C$
		r, q	1	0	1	0	1	0	0	1	r1	r0	q1	q0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r+q+C$
	SUB	r, q	1	0	1	0	1	0	1	0	r1	r0	q1	q0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r-q$
		r, i	1	1	0	1	0	1	r1	r0	i3	i2	i1	i0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r-i3-i0-C$
	SBC	r, q	1	0	1	0	1	0	1	1	r1	r0	q1	q0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r-q-C$
		r, i	1	1	0	0	1	0	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \wedge i3-i0$
	AND	r, q	1	0	1	0	1	1	0	0	r1	r0	q1	q0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \wedge q$
		r, i	1	1	0	0	1	1	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \vee i3-i0$
	OR	r, q	1	0	1	0	1	1	0	1	r1	r0	q1	q0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \vee q$
		r, i	1	1	0	1	0	0	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \vee i3-i0$
	XOR	r, q	1	0	1	0	1	1	1	0	r1	r0	q1	q0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \vee q$
		r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow r \vee i3-i0$
	CP	r, q	1	1	1	1	0	0	0	0	r1	r0	q1	q0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r-q$
		r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r-i3-i0$
	FAN	r, q	1	1	1	1	0	0	0	1	r1	r0	q1	q0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \wedge q$
		r, i	1	1	0	1	1	0	r1	r0	i3	i2	i1	i0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \wedge i3-i0$
	RLC	r	1	0	1	0	1	1	1	1	r1	r0	r1	r0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$d3 \leftarrow d2, d2 \leftarrow d1, d1 \leftarrow d0, d0 \leftarrow C, C \leftarrow d3$
	RRC	r	1	1	1	0	1	0	0	0	1	1	r1	r0	$\uparrow \downarrow$	$\uparrow \downarrow$			5	$d3 \leftarrow C, d2 \leftarrow d3, d1 \leftarrow d2, d0 \leftarrow d1, C \leftarrow d0$
	INC	Mn	1	1	1	1	0	1	1	0	n3	n2	n1	n0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$M(n3 \sim n0) \leftarrow M(n3 \sim n0)+1$
	DEC	Mn	1	1	1	1	0	1	1	1	n3	n2	n1	n0	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$M(n3 \sim n0) \leftarrow M(n3 \sim n0)-1$
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r1	r0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$M(X) \leftarrow M(X)+r+C, X \leftarrow X+1$
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r1	r0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$M(Y) \leftarrow M(Y)+r+C, Y \leftarrow Y+1$
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r1	r0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$M(X) \leftarrow M(X)-r-C, X \leftarrow X+1$
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r1	r0	$\star \uparrow \downarrow$	$\uparrow \downarrow$			7	$M(Y) \leftarrow M(Y)-r-C, Y \leftarrow Y+1$
	NOT	r	1	1	0	1	0	0	r1	r0	1	1	1	1	$\uparrow \downarrow$	$\uparrow \downarrow$			7	$r \leftarrow \bar{r}$

Abbreviations used in the explanations have the following meanings.

**Symbols associated with registers and memory**

- A** A register
- B** B register
- X** XHL register  
(low order eight bits of index register IX)
- Y** YHL register  
(low order eight bits of index register IY)
- XH** XH register  
(high order four bits of XHL register)
- XL** XL register  
(low order four bits of XHL register)
- YH** YH register  
(high order four bits of YHL register)
- YL** YL register  
(low order four bits of YHL register)
- SP** Stack pointer SP
- SPH** High-order four bits of stack pointer SP
- SPL** Low-order four bits of stack pointer SP
- MX, M(X)** Data memory whose address is specified with index register IX
- MY, M(Y)** Data memory whose address is specified with index register IY
- Mn, M(n)** Data memory address 000H–00FH  
(address specified with immediate data n of 00H–0FH)
- M(SP)** Data memory whose address is specified with stack pointer SP
- r, q** Two-bit register code  
r, q is two-bit immediate data; according to the contents of these bits, they indicate registers A, B, and MX and MY (data memory whose addresses are specified with index registers IX and IY)

r		q		Register specified
r1	r0	q1	q0	
0	0	0	0	A
0	1	0	1	B
1	0	1	0	MX
1	1	1	1	MY

**Symbols associated with program counter**

- NBP** New bank pointer
- NPP** New page pointer
- PCB** Program counter bank
- PCP** Program counter page
- PCS** Program counter step
- PCSH** Four high order bits of PCS
- PCSL** Four low order bits of PCS

**Symbols associated with flags**

- F** Flag register (I, D, Z, C)
- C** Carry flag
- Z** Zero flag
- D** Decimal flag
- I** Interrupt flag
- ↓ Flag reset
- ↑ Flag set
- ◇ Flag set or reset

**Associated with immediate data**

- p** Five-bit immediate data or label 00H–1FH
- s** Eight-bit immediate data or label 00H–0FFH
- l** Eight-bit immediate data 00H–0FFH
- i** Four-bit immediate data 00H–0FH

**Associated with arithmetic and other operations**

- + Add
- Subtract
- ^ Logical AND
- ∨ Logical OR
- ⊕ Exclusive-OR
- ★ Add-subtract instruction for decimal operation when the D flag is set



## APPENDIX C. E0C6001 I/O MEMORY MAP

Address	Register				Name	Init	1	0	Comment
	D3	D2	D1	D0					
0E0H	K03	K02	K01	K00	K03	-	High	Low	Input port data K03
	R				K02	-	High	Low	Input port data K02
	R				K01	-	High	Low	Input port data K01
	R				K00	-	High	Low	Input port data K00
0E4H	TM3	TM2	TM1	TM0	TM3	-	High	Low	Clock timer data 2 Hz
	R				TM2	-	High	Low	Clock timer data 4 Hz
	R				TM1	-	High	Low	Clock timer data 8 Hz
	R				TM0	-	High	Low	Clock timer data 16 Hz
0E8H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
	R/W				EIK01	0	Enable	Mask	Interrupt mask register (K01)
	R/W				EIK00	0	Enable	Mask	Interrupt mask register (K00)
0EBH	0	EIT2	EIT8	EIT32	0				
	R	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
	R/W				EIT8	0	Enable	Mask	Interrupt mask register (clock timer 8 Hz)
	R/W				EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0EDH	0	0	0	IK0	0				
	R				0				
	R				0				
	R				IK0	0	Yes	No	Interrupt factor flag (K00-K03)
0EFH	0	IT2	IT8	IT32	0				
	R				IT2	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
	R				IT8	0	Yes	No	Interrupt factor flag (clock timer 8 Hz)
	R				IT32	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)
0F3H	0	0	R01	R00	R01	0	High	Low	Output port data R01
	R		BUZZER	FOUT	BUZZER	0	On	Off	Buzzer On/Off control register
	R		R/W		R00	0	High	Low	Output port data R00
	R		R/W		FOUT	0	On	Off	Frequency output control register
0F6H	P03	P02	P01	P00	P03	-	High	Low	I/O port data P03
	R/W				P02	-	High	Low	I/O port data P02
	R/W				P01	-	High	Low	I/O port data P01
	R/W				P00	-	High	Low	I/O port data P00
0F9H	-	TMRST	-	-	TMRST	Reset	Reset	-	Clock timer reset
	-	W	-	-	-				
0FAH	HLMOD	0	0	0	HLMOD	0	Heavy	Normal	Heavy load protection mode register
	R				0				
	R				0				
0FBH	CSDC	0	0	0	CSDC	0	Static	Dynamic	LCD drive switch
	R				0				
	R				0				
0FCH	0	0	0	IOC	0				
	R			R/W	0				
	R			R/W	IOC	0	Out	In	I/O port I/O control register
0FDH	XBZR	0	XFOUT1	XFOUT0	XBZR	0	2 kHz	4 kHz	Buzzer frequency control
	R/W				0				
	R/W	R	R/W		XFOUT1	0			FOUT frequency control
					XFOUT0	0			FOUT frequency control



## APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures								
ICE6200	Nothing appears on the screen, or nothing works, after activation.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Is the RS-232C cable connected correctly?</li> <li>• Is the RS-232C driver installed?</li> <li>• Is SPEED.COM or MODE.COM on the disk?</li> <li>• Is the execution file correct?               <table style="margin-left: 40px; border: none;"> <tr> <td style="padding-right: 20px;">MS-DOS</td> <td>ICS6001J.EXE</td> </tr> <tr> <td>PC-DOS</td> <td>ICS6001W.EXE</td> </tr> </table> </li> <li>• Is the DOS version correct?               <table style="margin-left: 40px; border: none;"> <tr> <td style="padding-right: 20px;">MS-DOS</td> <td>Ver. 3.1 or later</td> </tr> <tr> <td>PC-DOS</td> <td>Ver. 2.1 or later</td> </tr> </table> </li> <li>• Is the DIP switches that set the baud rate of the main ICE6200 unit set correctly?</li> <li>• Is the breaker of the ICE6200 set to ON?</li> </ul>	MS-DOS	ICS6001J.EXE	PC-DOS	ICS6001W.EXE	MS-DOS	Ver. 3.1 or later	PC-DOS	Ver. 2.1 or later
	MS-DOS	ICS6001J.EXE								
	PC-DOS	ICS6001W.EXE								
	MS-DOS	Ver. 3.1 or later								
	PC-DOS	Ver. 2.1 or later								
	The ICE6200 breaker tripped immediately after activation.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Are connectors F1 and F5 connected to the EVA6001 correctly?</li> <li>• Is the target board power short-circuiting?</li> </ul>								
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE6200 is being used. Use the latest version.								
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS6001P.PAR is being used. Use the latest version.								
Immediate values A (10) and B (11) cannot be entered correctly with the A command.	The A and B registers are reserved for the entry of A and B. Write 0A and 0B when entering A (10) and B (11). <i>Example:</i> <pre>LD  A, B    Data in the B register is               loaded into the A register. LD  B, 0A   Immediate value A is loaded               into the B register.</pre>									
<UNUSED AREA> is displayed by the SD command.	This message is output when the address following one in which data is written is unused. It does not indicate problem. Data is correctly set in areas other than the read-only area.									
You can not do a real-time run in break-trace mode.	Since the CPU stops temporarily when breaking conditions are met, executing in a real-time is not performed.									
Output from the EVA is impossible when data is written to the I/O memory for Buzzer and Fout output with the ICE command.	Output is possible only in the real-time run mode.									
SOG6001	An R error occurs although the address is correctly set in the segment source file.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Does the address symbol use capital letters?</li> <li>• Are the output ports set for every two terminals?</li> </ul>								

Tool	Problem	Remedy measures
ASM6001	An R error occurs although the final page is passed.	The cross assembler is designed to output "R error" every time the page is changed. Use a pseudo-instruction to set the memory, such as ORG or PAGE, to change the page. See "Memory setting pseudo-instructions" in the cross assembler manual.
MDC6001	Activation is impossible.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Is the number of files set at ten or more in OS environment file CONFIG.SYS?</li> </ul>
EVA6001	The EVA6001 does not work when it is used independently.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target?</li> <li>• Is the EPROM for F.HEX and S.HEX installed correctly?</li> <li>• Is the appropriate voltage being supplied? (5V DC, 3A, or more)</li> <li>• Are the program ROMs (H and L) installed correctly?</li> <li>• Is data written from address 4000H? (When the 27C256 is used as the program ROM)</li> <li>• Is the EN/DIS switch on the EVA6001 set to EN?</li> </ul>
	Target segment does not light.	Check the following and remedy if necessary: <ul style="list-style-type: none"> <li>• Is an EPROM with an access time of 250 ns or less being used for S.HEX.</li> <li>• Has the VADJ VR inside the EVA6001 top cover been turned to a lower setting?</li> </ul>

# APPENDIX E. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6001

The EVA6001 is used to debug E0C6001 systems. However, the EVA6001 cannot output 1/2 duty signal to drive LCD. When debugging a E0C6001 system that uses a 1/2 duty drive, it should be done with the following procedure.

1. Set 1/3 duty function option using the FOG6001.
2. Assign the segments using the SOG6001.  
Since only the COM0 and COM1 signals are used for 1/2 duty drive, assign the RAM addresses to be used to COM0 and COM1.

```
;< LCD SEGMENT DECODE TABLE >
;SEG  COM0  COM1  COM2  COM3  SPEC
  0    900   901   ---   ---   S
  1    902   903   ---   ---   S
  2    910   911   ---   ---   S
  3    912   913   ---   ---   S
  4    920   921   ---   ---   S
  :
  :
  :
 18    9A0   9B0   ---   ---   S
 19    9C0   9C1   ---   ---   S
```

↑  
RAM addresses  
that are used for 1/2 duty drive

3. Debug with the EVA6001 and check whether the segment assignment is correct or not.  
Be aware that the LCD drive waveform is different and the LCD corresponding to COM2 goes on.

## Creating Mask Data

1. Edit the function option data (option No. 9) using the FOG6001 to change the 1/3 duty setting to the 1/2 duty setting.
2. It is unnecessary to change the segment assignment by the SOG6001.

*Note: The EVA6001 does not support the 1/2 bias LCD drive method. The LCD drive waveform is a 1/3 bias waveform.*

## AMERICA

---

### S-MOS SYSTEMS, INC.

150 River Oaks Parkway  
San Jose, CA 95134, U.S.A.  
Phone: +1-408-922-0200 Fax: +1-408-922-0238  
Telex: 176079 SMOS SNJUD

### S-MOS SYSTEMS, INC.

**EASTERN AREA SALES AND TECHNOLOGY CENTER**  
301 Edgewater Place, Suite 120  
Wakefield, MA 01880, U.S.A.  
Phone: +1-617-246-3600 Fax: +1-617-246-5443

### S-MOS SYSTEMS, INC.

**SOUTH EASTERN AREA SALES AND TECHNOLOGY CENTER**  
4300 Six Forks Road, Suite 430  
Raleigh, NC 27609, U.S.A.  
Phone: +1-919-781-7667 Fax: +1-919-781-6778

### S-MOS SYSTEMS, INC.

**CENTRAL AREA SALES AND TECHNOLOGY CENTER**  
1450 E.American Lane, Suite 1550  
Schaumburg, IL 60173, U.S.A.  
Phone: +1-847-517-7667 Fax: +1-847-517-7601

## EUROPE

---

- HEADQUARTERS -

### EPSON EUROPE ELECTRONICS GmbH

Riesstrasse 15  
80992 Muenchen, GERMANY  
Phone : +49-(0)89-14005-0 Fax : +49-(0)89-14005-110

- GERMANY -

### EPSON EUROPE ELECTRONICS GmbH SALES OFFICE

Breidenbachstrasse 46  
D-51373 Leverkusen, GERMANY  
Phone : +49-(0)214-83070-0 Fax : +49-(0)214-83070-10

- UNITED KINGDOM -

### EPSON EUROPE ELECTRONICS GmbH UK BRANCH OFFICE

G6 Doncastle House, Doncastle Road  
Bracknell, Berkshire RG12 8PE, ENGLAND  
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -

### EPSON EUROPE ELECTRONICS GmbH FRENCH BRANCH OFFICE

1 Avenue de l'Atlantique, LP 915 Les Conquerants  
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE  
Phone: +33-(0)1-64862350 Fax: +33-(0)1-64862355

## ASIA

---

- HONG KONG, CHINA -

### EPSON HONG KONG LTD.

20/F., Harbour Centre, 25 Harbour Road  
Wanchai, HONG KONG  
Phone: +852-2585-4600 Fax: +852-2827-4346  
Telex: 65542 EPSCO HX

- CHINA -

### SHANGHAI EPSON ELECTRONICS CO., LTD.

4F, Bldg., 27, No. 69, Gui Jing Road  
Caohejing, Shanghai, CHINA  
Phone: 21-6485-5552 Fax: 21-6485-0775

- TAIWAN, R.O.C. -

### EPSON TAIWAN TECHNOLOGY & TRADING LTD.

10F, No. 287, Nanking East Road, Sec. 3  
Taipei, TAIWAN, R.O.C.  
Phone: 02-2717-7360 Fax: 02-2712-9164  
Telex: 24444 EPSONTB

### EPSON TAIWAN TECHNOLOGY & TRADING LTD. HSINCHU OFFICE

13F-3, No. 295, Kuang-Fu Road, Sec. 2  
HsinChu 300, TAIWAN, R.O.C.  
Phone: 03-573-9900 Fax: 03-573-9169

- SINGAPORE -

### EPSON SINGAPORE PTE., LTD.

No. 1 Temasek Avenue, #36-00  
Millenia Tower, SINGAPORE 039192  
Phone: +65-337-7911 Fax: +65-334-2716

- KOREA -

### SEIKO EPSON CORPORATION KOREA OFFICE

10F, KLI 63 Bldg., 60 Yoido-Dong  
Youngdeungpo-Ku, Seoul, 150-010, KOREA  
Phone: 02-784-6027 Fax: 02-767-3677

- JAPAN -

### SEIKO EPSON CORPORATION ELECTRONIC DEVICES MARKETING DIVISION

#### Electronic Device Marketing Department IC Marketing & Engineering Group

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

#### ED International Marketing Department I (Europe & U.S.A.)

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

#### ED International Marketing Department II (Asia)

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN  
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110



In pursuit of **“Saving” Technology**, Epson electronic devices.  
Our lineup of semiconductors, liquid crystal displays and quartz devices  
assists in creating the products of our customers’ dreams.  
**Epson IS energy savings.**

# EPSON

---

**SEIKO EPSON CORPORATION**  
**ELECTRONIC DEVICES MARKETING DIVISION**

■ Electronic devices information on the Epson WWW server

<http://www.epson.co.jp>

Issue APRIL 1998 Printed in Japan  A