

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C6003**

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 Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency.

MS-DOS and EDLIN are registered trademarks of Microsoft Corporation, U.S.A.

PC-DOS, PC/AT, PS/2, VGA, EGA and IBM are registered trademarks of International Business Machines Corporation, U.S.A.

NEC PC-9801 Series and NEC are registered trademarks of NEC Corporation.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective owners.

E0C6003 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Micro-computer E0C6003.

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 EVA6237 Manual ICE6200 (ICE62R) Hardware Manual KIT6003 Manual
<i>Development procedure</i>	☞ E0C62 Family Technical Guide
<i>Device (E0C6003)</i>	☞ E0C6003 Technical Manual
<i>Instructions</i>	☞ E0C6200/6200A Core CPU Manual

CONTENTS

1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL	1
1.1 Configuration of DEV6003	1
1.2 Developmental Environment	1
1.3 Development Flow	2
1.4 Production of Execution Disk	3
2 CROSS ASSEMBLER ASM6003	4
2.1 ASM6003 Outline	4
2.2 E0C6003 Restrictions	4
2.3 ASM6003 Quick Reference	5
3 FUNCTION OPTION GENERATOR FOG6003	7
3.1 FOG6003 Outline	7
3.2 E0C6003 Option List	7
3.3 Option Specifications and Selection Message	9
3.4 FOG6003 Quick Reference	15
3.5 Sample File	17
4 SEGMENT OPTION GENERATOR SOG6003	19
4.1 SOG6003 Outline	19
4.2 Option List	19
4.3 Segment Ports Output Specifications	20
4.4 SOG6003 Quick Reference	21
4.5 Sample Files	23
5 ICE CONTROL SOFTWARE ICS6003	24
5.1 ICS6003 Outline	24
5.2 ICS6003 Restrictions	24
5.3 ICS6003 Quick Reference	25

6 MASK DATA CHECKER MDC6003 _____ **28**
 6.1 MDC6003 Outline 28
 6.2 MDC6003 Quick Reference..... 28

APPENDIX A. E0C6003 INSTRUCTION SET _____ **31**

B. E0C6003 RAM MAP _____ **35**

C. E0C6003 I/O MEMORY MAP _____ **36**

D. TROUBLESHOOTING _____ **37**

E. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6237 _____ **39**

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 DEV6003

The below software are included in the product of the E0C6003 development support tool DEV6003.

1. Cross Assembler ASM6003 Cross assembler for program preparation
2. Function Option Generator FOG6003 Function option data preparation program
3. Segment Option Generator SOG6003 Segment option data preparation program
4. ICE Control Software ICS6003 ICE control program
5. Mask Data Checker MDC6003 Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV6003 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 E0C6003, 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.

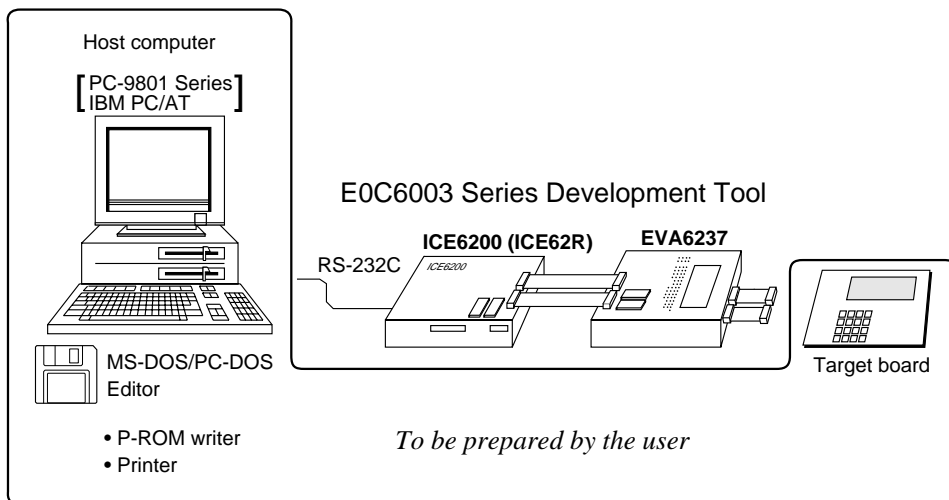


Fig. 1.2.1 System configuration

Note The DEV6003 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 (ICE62R) 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 DEV6003.

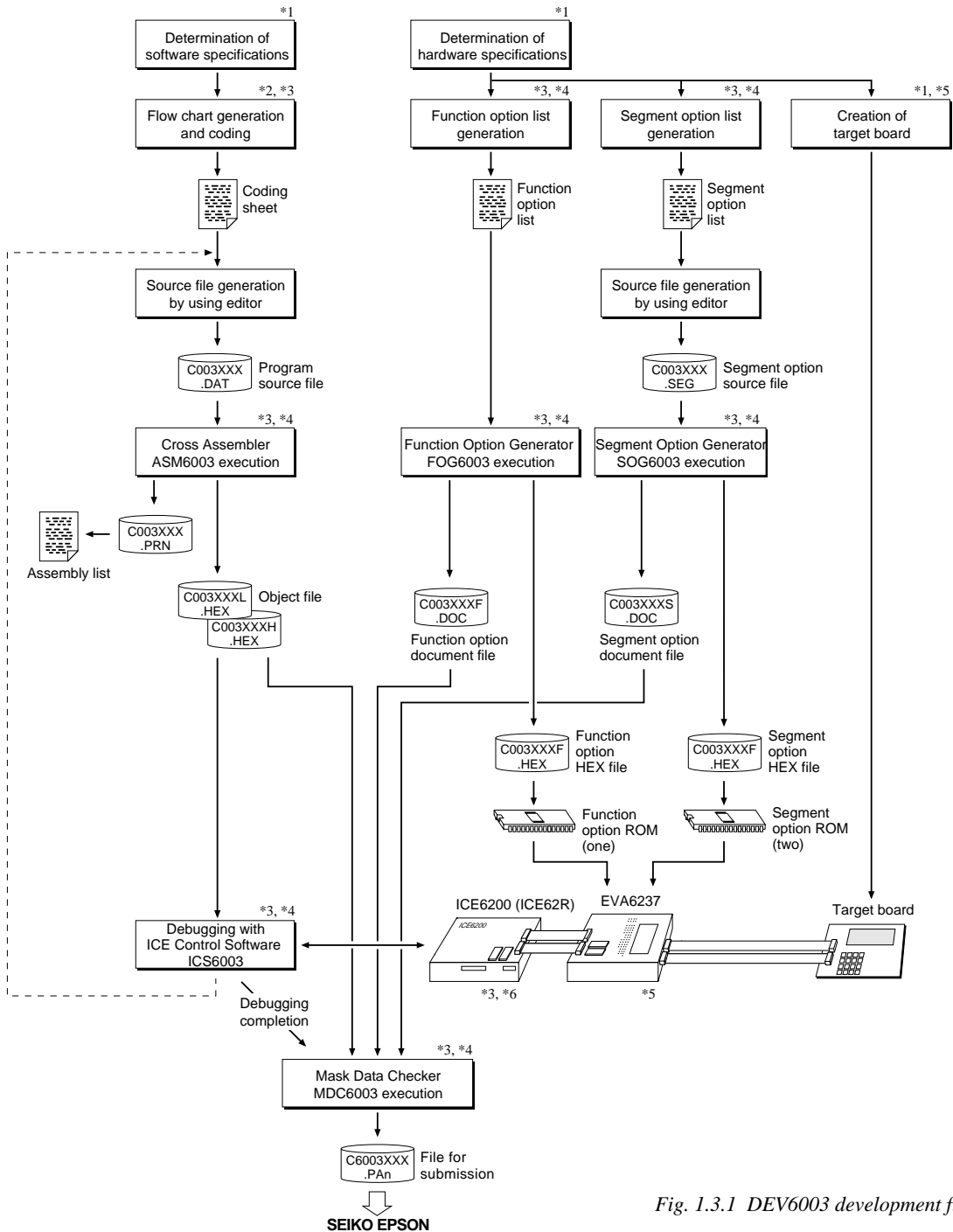


Fig. 1.3.1 DEV6003 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C003XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

Reference Manual

- *1 E0C6003 Technical Hardware Manual
- *2 E0C6200/6200A Core CPU Manual
- *3 E0C62 Family Development Tool Reference Manual
- *4 E0C6003 Development Tool Manual (this manual)
- *5 EVA6237 Manual
- *6 ICE6200 (ICE62R) Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 (ICE62R) are recorded in the DEV6003 floppy disk.

The content of the files contained in the DEV6003 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM6003.EXE	ASM6003.EXE	Cross Assembler execution file
FOG6003.EXE	FOG6003.EXE	Function Option Generator execution file
ICS6003B.BAT	ICS6003.BAT	ICE Control Software batch file
ICS6003W.EXE	ICS6003J.EXE	ICE Control Software execution file
ICS6003P.PAR	ICS6003P.PAR	ICE Control Software parameter file
MDC6003.EXE	MDC6003.EXE	Mask Data Checker execution file
SOG6003.EXE	SOG6003.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 (DEV6003, 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 MDC6003 will handle many files.

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

- In "ICS6003(B).BAT" the batch process is indicated such that the ICS6003J(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 (DEV6003), then insert the original disk into the A drive and execute the COPY command.

```
C>\MD DEV6003 [↵]
```

```
C>\CD DEV6003 [↵]
```

```
C\DEV6003>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 ASM6003

2.1 ASM6003 Outline

The ASM6003 cross assembler is an assembler program for generating the machine code used by the E0C6003 4-bit, single-chip microcomputers. The Cross Assembler ASM6003 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.

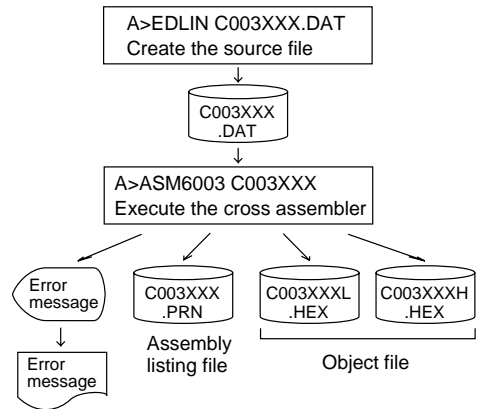


Fig. 2.1.1 ASM6003 execution flow

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 E0C6003 Restrictions

Note the following when generating a program by the E0C6003:

ROM area

The capacity of the E0C6003 ROM is 768 steps (0000H to 02FFH).

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

Memory configuration:

Bank: Only bank 0, Page: 3 pages (0 to 2H), each 256 steps

Significant specification range:

ORG pseudo-instruction: 0000H to 02FFH
 PAGE pseudo-instruction: 00H to 02H
 BANK pseudo-instruction: Only 0H
 PSET instruction: 00H to 02H

RAM area

The capacity of the E0C6003 RAM is 64 words (000H to 03FH, and 0E0H to 0EFH, 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 E0C6003 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 ASM6003 Quick Reference

Starting command and input/output files

Execution file: ASM6003.EXE

_ indicates a blank.

indicates the Return key.

A parameter enclosed by [] can be omitted.

Starting command: **ASM6003_ [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: C003XXX.DAT (Source file)

Output file:

- C003XXXL.HEX (Object file, low-order)
- C003XXXH.HEX (Object file, high-order)
- C003XXX.PRN (Assembly listing file)

Display example

```

*** ASM6003 CROSS ASSEMBLER. --- 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 NNNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSS 00000000 NNN NN

(C) COPYRIGHT 1997 SEIKO EPSON CORP.

SOURCE FILE NAME IS " C003XXX.DAT "

THIS SOFTWARE MAKES NEXT FILES.

C003XXXH.HEX ... HIGH BYTE OBJECT FILE.
C003XXXL.HEX ... LOW BYTE OBJECT FILE.
C003XXX.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 ASM6003 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, ASM6003 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 2
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"> The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified. A value greater than that which the number of significant digits of the operand will accommodate was specified.
! (Warning)	<ul style="list-style-type: none"> Memory areas overlapped because of a "PAGE" or "ORG" pseudo-instruction or both.
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 FOG6003

3.1 FOG6003 Outline

With the 4-bit single-chip E0C6003 microcomputers, the customer may select 10 hardware options. By modifying the mask patterns of the E0C6003 according to the selected options, the system can be customized to meet the specifications of the target system. The Function Option Generator FOG6003 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 FOG6003, the E0C6003 mask pattern is automatically generated by a general purpose computer. The HEX file for the evaluation board (EVA6237) 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.

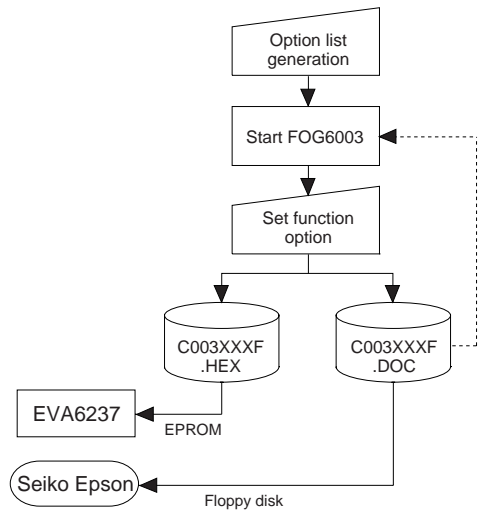


Fig. 3.1.1 FOG6003 execution flow

3.2 E0C6003 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

- 1. E0C6003 (Normal Type)
- 2. E0C60L03 (Low Power Type)

2. LCD SPECIFICATION

- BIAS SELECTION 1. 1/3 Bias By Voltage Divider
 - 2. 1/2 Bias By Voltage Divider
 - 3. 1/2 Bias By Doubler/Halver
- DUTY SELECTION 1. 1/4 Duty
 - 2. 1/3 Duty
 - 3. 1/2 Duty

3. OSC1 SYSTEM CLOCK

- 1. Crystal
- 2. CR

4. MULTIPLE KEY ENTRY RESET

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

5. INTERRUPT NOISE REJECTOR

- K00-K03 1. Use
 - 2. Not Use

6. TIMER INTERRUPT FREQUENCY

- INTERRUPT FREQUENCY 1. 32/16/2 Hz Interrupt
 2. 64/16/2 Hz Interrupt

7. INPUT PORT PULL DOWN RESISTOR

- K00 1. With Resistor 2. Gate Direct
- K01 1. With Resistor 2. Gate Direct
- K02 1. With Resistor 2. Gate Direct
- K03 1. With Resistor 2. Gate Direct

8. R00 SPECIFICATION

- OUTPUT TYPE 1. DC Output
 2. Buzzer Inverted Output (R00 Control)
 3. 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

9. R01 SPECIFICATION

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

10.R02, R03 SPECIFICATION

- R02 OUTPUT SPECIFICATION 1. Complementary 2. Pch-Open Drain
- R03 OUTPUT SPECIFICATION 1. Complementary 2. Pch-Open Drain

3.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C6003 are shown below, and their specifications are also described.

1 Device type

```

*** OPTION NO.1 ***
--- DEVICE TYPE ---
                1. E0C6003
                2. E0C60L03
PLEASE SELECT NO.(1) ? 2 
                2. E0C60L03  SELECTED

```

Select the chip specification.

There are two models: E0C6003 (3 V supply voltage) and E0C60L03 (1.5 V supply voltage, low-power specification).

2 LCD specification

```

*** OPTION NO.2 ***
--- LCD SPECIFICATION ---
BIAS SELECTION  1. 1/3 BIAS BY VOLTAGE DIVIDER
                2. 1/2 BIAS BY VOLTAGE DIVIDER
                3. 1/2 BIAS BY DOUBLER/HALVER
PLEASE SELECT NO.(1) ? 1 
DUTY SELECTION  1. 1/4 DUTY
                2. 1/3 DUTY
                3. 1/2 DUTY
PLEASE SELECT NO.(1) ? 1 
BIAS SELECTION  1. 1/3 BIAS BY VOLTAGE DIVIDER  SELECTED
DUTY SELECTION  1. 1/4 DUTY  SELECTED

```

Select the common (drive) duty and bias.

When 1/2 duty is selected, up to 30 segments of LCD panel can be driven with 2 COM terminals and 15 SEG terminals. When 1/3 duty is selected, up to 45 segments can be driven with 3 COM terminals, and when 1/4 duty is selected, up to 60 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. Figures 3.3.1 and 3.3.2 show the drive waveforms of 1/3 bias driving and 1/2 bias driving, respectively.

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.

See Figure 3.3.3 for the external elements according to the bias to be selected.

Table 3.3.1 Common duty selection standard

Number of segments	Common duty
1–30	1/2
31–45	1/3
46–60	1/4

3 FUNCTION OPTION GENERATOR FOG6003

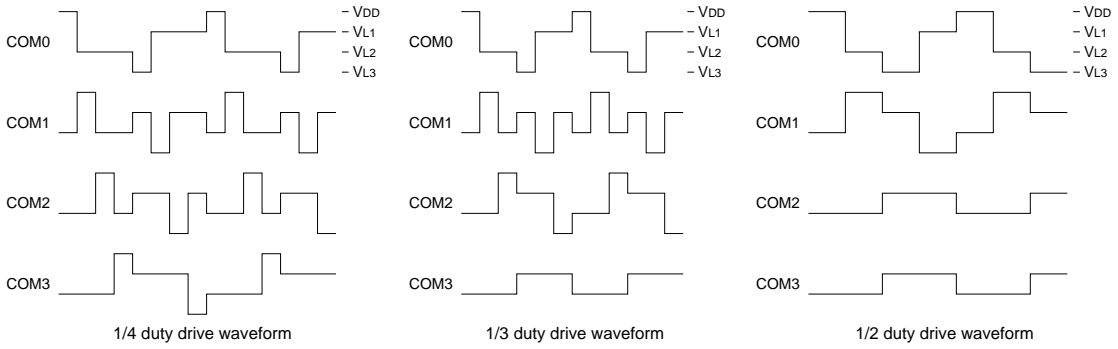


Fig. 3.3.1 Drive waveform from COM terminals (1/3 bias)

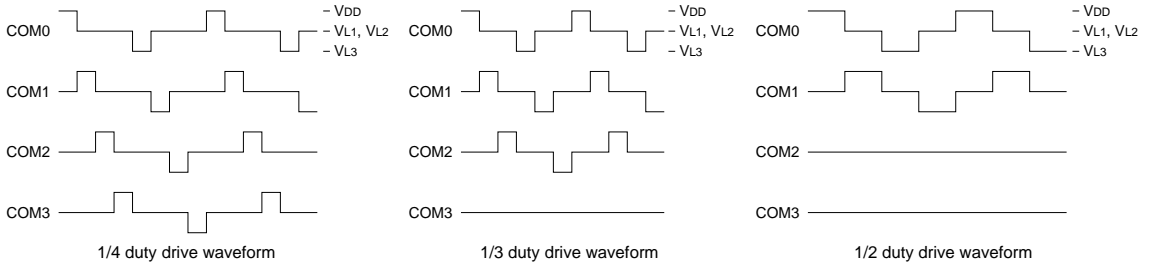
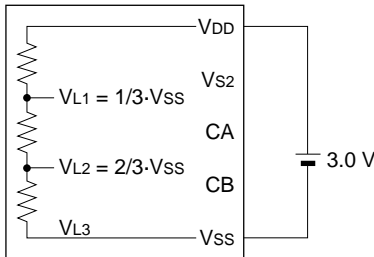


Fig. 3.3.2 Drive waveform from COM terminals (1/2 bias)

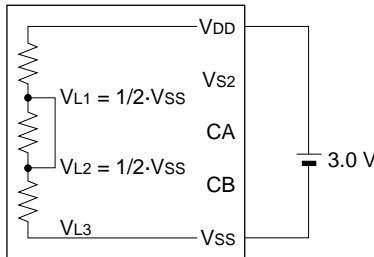
• **E0C6003** (3 V LCD panel)

1/4, 1/3, 1/2 duty
1/3 bias



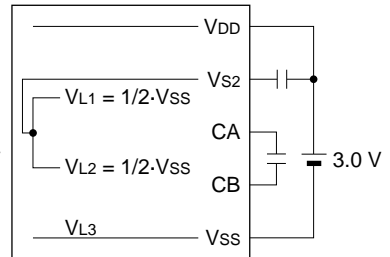
Note: VL3 is shorted to VSS internally.

1/4, 1/3, 1/2 duty
1/2 bias (A)



Note: VL3-VSS and VL1-VL2 are shorted internally.

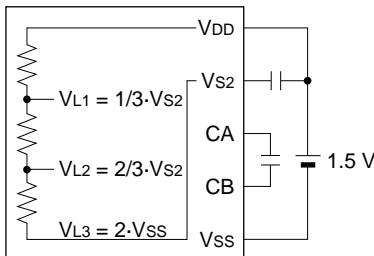
1/4, 1/3, 1/2 duty
1/2 bias (B)



Note: VL3-VSS and VL1-VL2 are shorted internally.

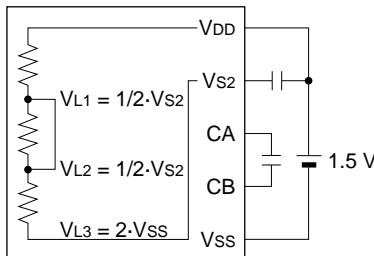
• **E0C60L03** (3 V LCD panel)

1/4, 1/3, 1/2 duty
1/3 bias



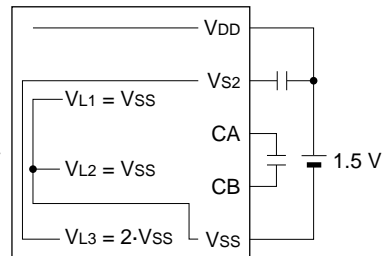
Note: VL3 is shorted to VSS internally.

1/4, 1/3, 1/2 duty
1/2 bias (A)



Note: VL3-VS2 and VL1-VL2 are shorted internally.

1/4, 1/3, 1/2 duty
1/2 bias (B)



Note: VL3-VS2 and VL1-VL2 are shorted internally.

Fig. 3.3.3 External elements

3 OSC1 system clock

```

*** OPTION NO.3 ***
--- 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.

4 Multiple key entry reset

```

*** OPTION NO.4 ***
--- MULTIPLE KEY ENTRY RESET ---
COMBINATION      1. NOT USE
                  2. USE K00,K01
                  3. USE K00,K01,K02
                  4. USE ALL K00-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 3.3.4.

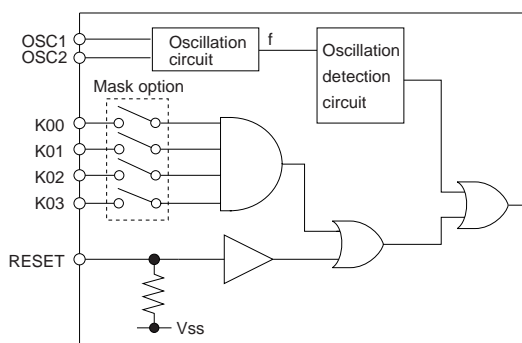


Fig. 3.3.4
System reset circuit

5 Interrupt noise rejector

```

*** OPTION NO.5 ***
--- 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.

6 Timer interrupt frequency

```

*** OPTION NO.6 ***
--- TIMER INTERRUPT FREQUENCY ---
  INTERRUPT FREQUENCY  1. 32/16/2 HZ INTERRUPT
                       2. 64/16/2 HZ INTERRUPT
PLEASE SELECT NO.(1) ? 1 [ ]
  INTERRUPT FREQUENCY  1. 32/16/2 HZ INTERRUPT  SELECTED
    
```

Select timer interrupt frequencies. Either 32 Hz or 64 Hz can be selected for one of among three interrupt frequencies.

The EIT32 interrupt mask register (0F4H•D0) is used for controlling the 32 Hz or 64 Hz interrupt. When a 32 Hz or 64 Hz interrupt occurs, the IT32 interrupt factor flag (0F6H•D0) is set.

7 Input ports pull down resistor

```

*** OPTION NO.7 ***
--- 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 3.3.5.

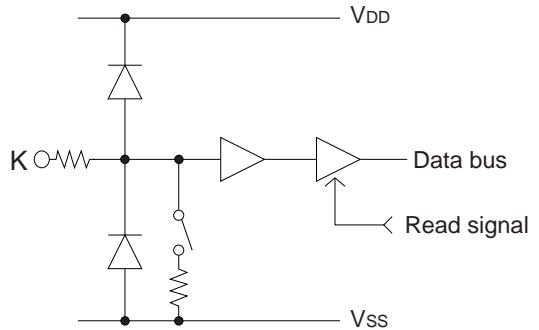


Fig. 3.3.5 Configuration of pull down resistor

8 R00 specification

```

*** OPTION NO.8 ***
--- R00 SPECIFICATION ---
      OUTPUT TYPE          1. D.C.
                          2. /BZ R00
                          3. FOUT
PLEASE SELECT NO.(3) ? 3 

      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          3. 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. 3.3.6 Output waveform at DC output selection

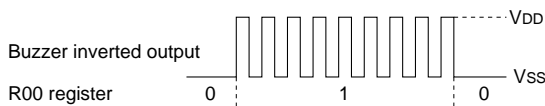


Fig. 3.3.7 Output waveform at buzzer inverted output selection

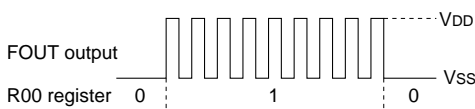


Fig. 3.3.8 Output waveform at FOUT output selection

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 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 3.3.6.
- 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 3.3.7.
- 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 3.3.8.

The output circuit configuration is shown in Figure 3.3.9.

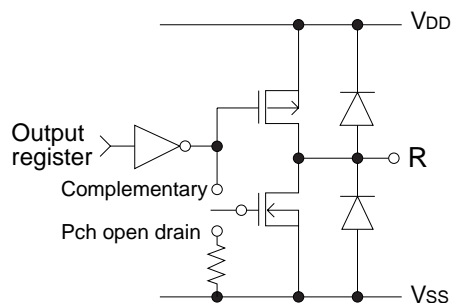


Fig. 3.3.9 Configuration of output circuit

9 R01 specification

```

*** OPTION NO.9 ***
--- 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. 3.3.10 Output waveform at DC output selection

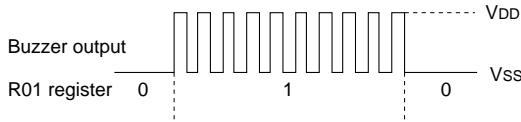


Fig. 3.3.11 Output waveform at buzzer output selection

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 3.3.10.
- 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 3.3.11.

10 R02, R03 specification

```

*** OPTION NO.10 ***
--- R02, R03 OUTPUT SPECIFICATION ---
      R02 OUTPUT SPECIFICATION 1. COMPLEMENTARY
                              2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]
      R03 OUTPUT SPECIFICATION 1. COMPLEMENTARY
                              2. PCH-OPENDRAIN
PLEASE SELECT NO.(1) ? 1 [ ]
      R02 OUTPUT SPECIFICATION 1. COMPLEMENTARY   SELECTED
      R03 OUTPUT SPECIFICATION 1. COMPLEMENTARY   SELECTED
    
```

Select the output specification for the R02 and R03 terminals. Either complementary output or Pch open drain output may be selected.

3.4 FOG6003 Quick Reference

■ Starting command and input/output files

Execution file: FOG6003.EXE

Starting command: FOG6003

indicates the Return key.

Input file: C003XXXF.DOC (Function option document file, when modifying)

Output file: C003XXXF.DOC (Function option document file)
C003XXXF.HEX (Function option HEX file)

■ Display example

```

*** E0C6003 FUNCTION OPTION GENERATOR. --- Ver 1.00 ***
EEEEEEEEEE PPPPPPPP SSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPP SSS SSSS OOO OOO NNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNN NNN
EEE PPP PPP SSS SSS OOO OOO NNNNNN NNN
EEEEEEEEEE PPPPPPPP SSSSSS OOO OOO NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS OOO OOO NNN NNNNN
EEE PPP SSS OOO OOO NNN NNNNN
EEE PPP SSS SSS OOO OOO NNN NNNN
EEEEEEEEEE PPP SSS SSS OOO OOO NNN NNN
EEEEEEEEEE PPP SSSSSS OOOOOOO NNN NN

(C) COPYRIGHT 1997 SEIKO EPSON CORP.

THIS SOFTWARE MAKES NEXT FILES.

C003XXXF.HEX ... FUNCTION OPTION HEX FILE.
C003XXXF.DOC ... FUNCTION OPTION DOCUMENT FILE.

STRIKE ANY KEY.

```

Start-up message

When FOG6003 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.

```

*** E0C6003 USER'S OPTION SETTING. --- Ver 1.00 ***
CURRENT DATE IS 1997/03/03
PLEASE INPUT NEW DATE : 1997/03/03

```

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

PLEASE SELECT NO.?

```

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 FOG6003.

```

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

PLEASE SELECT NO.? 1
PLEASE INPUT FILE NAME? C0030A0 ..(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 399-0293 JAPAN
? TEL 0266-61-1211
? FAX 0266-61-1273
? 

```

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.

```

PLEASE INPUT FILE NAME? C0030A0
EXISTS OVERWRITE(Y/N)? N
PLEASE INPUT FILE NAME? C0030B0
PLEASE INPUT USER'S NAME?

```

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) ***

C0030A0          C0030B0          C0030C0          ..(1)

PLEASE INPUT FILE NAME? C0030A0
PLEASE INPUT USER'S NAME?
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)?
PLEASE INPUT EDIT NO.? 4
:
(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? C0030N0
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
    
```

```

*** OPTION NO.4 ***

--- MULTIPLE KEY ENTRY RESET ---

COMBINATION      1. Not Use
                 2. Use  K00,K01
                 3. Use  K00,K01,K02
                 4. Use  All K00-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.? 3
3. 27C256  SELECTED ..(2)

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 EVA6237, 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 EVA6237 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

```

* E0C6003 FUNCTION OPTION DOCUMENT V 1.00
*
* FILE NAME      C0030A0F.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    1997/03/03
*
* COMMENT       FUJIMI PLANT
*               281 FUJIMI SUWA-GUN NAGANO-KEN 399-0293 JAPAN
*               TEL 0266-61-1211
*               FAX 0266-61-1273
*
*
* OPTION NO.1
* < DEVICE TYPE >
*
*               E0C60L03 ----- SELECTED
OPT0101 02
*
* OPTION NO.2
* < LCD SPECIFICATION >
*   BIAS SELECTION      1/3 BIAS BY VOLTAGE DIVIDER - SELECTED
*   DUTY SELECTION      1/4 DUTY ----- SELECTED
OPT0201 01
OPT0202 01
*
* OPTION NO.3
* < OSC1 SYSTEM CLOCK >
*
*               CRYSTAL ----- SELECTED
OPT0301 01
*
* OPTION NO.4
* < MULTIPLE KEY ENTRY RESET >
*   KEY COMBINATION     USE ALL K00-K03 ----- SELECTED
OPT0401 04
*
* OPTION NO.5
* < INTERRUPT NOISE REJECTOR >
*   K00-K03             USE ----- SELECTED
OPT0501 01
*
* OPTION NO.6
* < TIMER INTERRUPT FREQUENCY >
*   INTERRUPT FREQUENCY 32/16/2 HZ INTERRUPT ----- SELECTED
OPT0601 01
*
* OPTION NO.7
* < INPUT PORT PULL DOWN RESISTOR >
*   K00                 WITH RESISTOR ----- SELECTED
*   K01                 WITH RESISTOR ----- SELECTED
*   K02                 WITH RESISTOR ----- SELECTED
*   K03                 WITH RESISTOR ----- SELECTED
OPT0701 01
OPT0702 01
OPT0703 01
OPT0704 01
*
* OPTION NO.8
* < R00 SPECIFICATION >
*   OUTPUT TYPE         FOUT ----- SELECTED
*   F1 FREQUENCY SELECTION 512 HZ ----- SELECTED
*   F2 FREQUENCY SELECTION 1024 HZ ----- SELECTED
*   F3 FREQUENCY SELECTION 2048 HZ ----- SELECTED
*   F4 FREQUENCY SELECTION 4096 HZ ----- SELECTED
*   OUTPUT SPECIFICATION COMPLEMENTARY ----- SELECTED
OPT0801 03
OPT0802 02
OPT0803 02
OPT0804 02

```

3 FUNCTION OPTION GENERATOR FOG6003

```
OPT0805 02
OPT0806 01
*
* OPTION NO.9
* < R01 SPECIFICATION >
*   OUTPUT TYPE           BZ OUTPUT ----- SELECTED
*   OUTPUT SPECIFICATION COMPLEMENTARY ----- SELECTED
OPT0901 02
OPT0902 01
*
* OPTION NO.10
* < R02, R03 OUTPUT SPECIFICATION >
*   R02 OUTPUT SPECIFICATION COMPLEMENTARY ----- SELECTED
*   R03 OUTPUT SPECIFICATION COMPLEMENTARY ----- SELECTED
OPT1001 01
OPT1002 01
*
*
* SEIKO EPSON'S AREA
*
*
* OPTION NO.11
OPT1101 01
OPT1102 01
*
* OPTION NO.12
OPT1201 01
*
* OPTION NO.13
OPT1301 01
*
* OPTION NO.14
OPT1401 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 SOG6003

4.1 SOG6003 Outline

With the 4-bit single-chip E0C6003 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C6003 according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG6003 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG6003, the E0C6003 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA6237) 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.

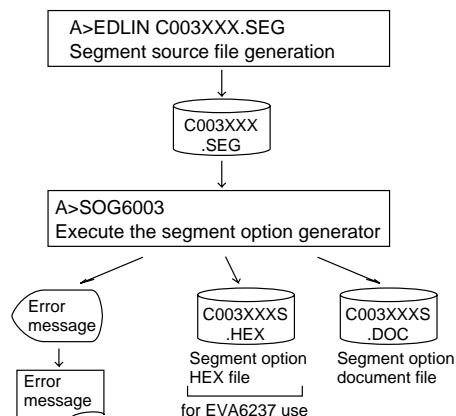


Fig. 4.1.1 SOG6003 execution flow

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
														DC output <input type="checkbox"/> C <input type="checkbox"/> P
Legend:	<ADDRESS> H: High order address, L: Low order address D: Data bit												<OUTPUT SPECIFICATION> C: Complementary output 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–SEG14, 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 (0E0H–0EFH) can be allocated to the optional segment. With this, up to 60 segments (45 segments when 1/3 duty is selected or 30 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 (E), 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  E00  E01  E02  E03  S
1  E10  E11  E12  E13  S
```

- When 1/3 duty is selected

```
0  E00  E01  E02  ---  S
1  E10  E11  E12  ---  S
```

- When 1/2 duty is selected

```
0  E00  E01  ---  ---  S
1  E10  E11  ---  ---  S
```

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 15 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 SEG10 and SEG11, and Pch open drain output is set to SEG12 and SEG13.

```
10  E00  ---  ---  ---  C
11  E10  ---  ---  ---  C
12  E21  ---  ---  ---  P
13  E31  ---  ---  ---  P
```

Note Only complementary output is enabled as the DC output of the SEG ports of EVA6237. Therefore, complementary output is enabled even if Pch open drain output is selected. Respond to it by adding external circuits as required.

4.4 SOG6003 Quick Reference

■ Starting command and input/output files

Execution file: SOG6003.EXE

_ indicates a blank.

Starting command: SOG6003_ [-H]

indicates the Return key.

A parameter enclosed by [] can be omitted.

Option: -H: Specifies the segment option document file for input file of SOG6003.

Input file: C003XXX.SEG (Segment option source file)
C003XXXS.DOC (Segment option document file, when -H option use)

Output file: C003XXXS.DOC (Segment option document file)
C003XXXS.HEX (Segment option HEX file)

■ Display example

```

*** E0C6003 SEGMENT OPTION GENERATOR. --- 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 SSS 000 000 NNNNNN NNN
EEEEEEEEEE PPPPPPPP SSSSSSS 000 000 NNN NNN NNN
EEEEEEEEEE PPPPPPPP SSSS 000 000 NNN NNNNNN
EEE PPP SSS SSS 000 000 NNN NNNN
EEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSS SSS 000 000 NNN NNN
EEEEEEEEEE PPP SSSSSSS 00000000 NNN NN

(C) COPYRIGHT 1997 SEIKO EPSON CORP.

SEGMENT OPTION SOURCE FILE NAME IS " C003XXX.SEG "

THIS SOFTWARE MAKES NEXT FILES.

C003XXXS.HEX ... SEGMENT OPTION HEX FILE.
C003XXXS.DOC ... SEGMENT OPTION DOCUMENT FILE.

STRIKE ANY KEY.

```

Start-up message

When SOG6003 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.

```

*** E0C6003 USER'S OPTION SETTING. --- Ver 1.00 ***
CURRENT DATE IS 97/03/26
PLEASE INPUT NEW DATE : 97/03/28 

```

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.

```

*** SOURCE FILE(S) ***
C0030A0 C0030B0 C0030C0 ..(1)
PLEASE INPUT SEGMENT OPTION FILE NAME? C0030A0  ..(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 399-0293 JAPAN 
? TEL 0266-61-1211 
? FAX 0266-61-1273 
? 

```

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.

```

*** 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

```

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.

```

PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C0030N0 
SEGMENT OPTION SOURCE FILE IS NOT FOUND. ..(7) -H option not use
PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C0030N0 
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND. ..(8) -H option use

```

4 SEGMENT OPTION GENERATOR SOG6003

```

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.? 3  .. (2)

    3. 27C256   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 EVA6237, 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 EVA6237 options.

When a series of operations are complete, the SOG6003 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG6003 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 (Duprication 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

```
; C0030A0.SEG, VER.1.00
; EVA6237 LCD SEGMENT DECODE TABLE
;
0   E00 E01 E02 E03 S
1   E10 E11 E12 E13 S
2   E20 E21 E22 E23 S
3   E30 E31 E32 E33 S
4   E40 E41 E42 E43 S
5   E50 E51 E52 E53 S
6   E60 E61 E62 E63 S
7   E70 E71 E72 E73 S
8   E80 E81 E82 E83 S
9   E90 E91 E92 E93 S
10  EA0 EA1 EA2 EA3 S
11  EB0 EB1 EB2 EB3 S
12  EC0 EC1 EC2 EC3 S
13  ED0 ED1 ED2 ED3 S
14  EE3 --- --- --- C
```

■ Example of segment option document file

```
* E0C6003 SEGMENT OPTION DOCUMENT V 1.00
*
* FILE NAME      C0030A0S.DOC
* USER'S NAME   SEIKO EPSON CORP.
* INPUT DATE    97/03/26
* COMMENT       FUJIMI PLANT
*               281 FUJIMI SUWA-GUN NAGANO-KEN 399-0293 JAPAN
*               TEL 0266-61-1211
*               FAX 0266-61-1273
*
*
* OPTION NO.11
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
0   E00 E01 E02 E03 S
1   E10 E11 E12 E13 S
2   E20 E21 E22 E23 S
3   E30 E31 E32 E33 S
4   E40 E41 E42 E43 S
5   E50 E51 E52 E53 S
6   E60 E61 E62 E63 S
7   E70 E71 E72 E73 S
8   E80 E81 E82 E83 S
9   E90 E91 E92 E93 S
10  EA0 EA1 EA2 EA3 S
11  EB0 EB1 EB2 EB3 S
12  EC0 EC1 EC2 EC3 S
13  ED0 ED1 ED2 ED3 S
14  EE3 EE0 EE1 EE2 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 ICS6003

5.1 ICS6003 Outline

The In-circuit Emulator ICE6200 (ICE62R) connects the target board produced by the user via the EVA6237 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 (ICE62R) control is done through the ICE Control Software ICS6003.

The ICS6003 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.

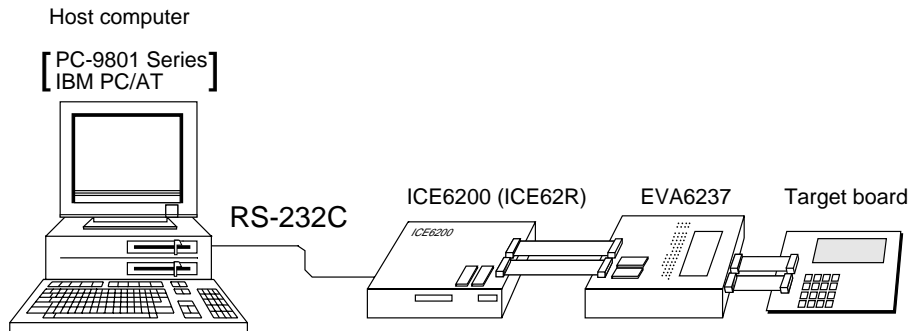


Fig. 5.1.1 Debugging system using ICE6200 (ICE62R)

☞ The functions of the ICE6200 (ICE62R) and commands are same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.

5.2 ICS6003 Restrictions

Take the following precautions when using the ICS6003.

■ ROM Area

The ROM area is limited to a maximum address of 2FFH. Assigning data above the 2FFH address causes an error.

■ RAM Area

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

Unused area: 040H to 0DFH

Memory 0E0H to 0EFH is display memory; 0F0H to 0FFH is I/O memory.
(Refer to the "E0C6003 Technical Manual" for details.)

■ Undefined Code

The instructions below are not specified for the E0C6003 and so cannot be used.

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

■ OPTLD Command

In the ICS6003, OPTLD command cannot be used.

5.3 ICS6003 Quick Reference

Starting command and input/output files

☐ indicates the Return key.

Execution file: ICS6003.BAT (ICS6003J.EXE) . . . for MS-DOS
ICS6003B.BAT (ICS6003W.EXE) . . . for PC-DOS

Starting command: ICS6003 (ICS6003J)☐ . . . for MS-DOS
ICS6003B (ICS6003W)☐ . . . for PC-DOS

Input file: C003XXXL.HEX (Object file, low-order)
C003XXXH.HEX (Object file, high-order)
C003XXXD.HEX (Data RAM file)
C003XXXC.HEX (Control file)

Output file: C003XXXL.HEX (Object file, low-order)
C003XXXH.HEX (Object file, high-order)
C003XXXD.HEX (Data RAM file)
C003XXXC.HEX (Control file)

Display example

```

*** E0C6003 ICE CONTROL SOFTWARE. --- Ver 1.00 ***
EEEEEEEEEE P P P P P P P P S S S S S S 0 0 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P P P S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP PPP S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP PPP S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE P P P P P P P P S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP PPP S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEE PPP S S S S S S 0 0 0 0 0 0 N N N N N N N N
EEEEEEEEEE PPP S S S S S S 0 0 0 0 0 0 N N N N N N
EEEEEEEEEE PPP S S S S S S 0 0 0 0 0 0 N N N N N N

(C) COPYRIGHT 1997 SEIKO EPSON CORP.

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

```

Start-up message

When ICS6003 is started, the start-up message is displayed, and a self-test is automatically performed. ICS6003 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 ICS6003 program is terminated by entering the Q (Quit) command.

Note Confirm that the cables connected properly, then operate the ICS6003.

Error messages

Error message	Meaning	Recover procedure
* COMMUNICATION ERROR OR ICE NOT READY *	ICE6200 (ICE62R) is disconnected or power is OFF.	Switch OFF the host power supply, connect cable, and reapply power. Or switch ON power to ICE6200 (ICE62R).
* 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 (ICE62R).
* 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 (ICE62R).
* 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 (ICE62R).
* 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.

■ Command list

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 EVA6237 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 EVA6237 CPU internal registers
		#SR [↵]	Set EVA6237 CPU internal registers
		#I [↵]	Reset EVA6237 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 (ICE62R) set condition from file
		#CS,file <input type="checkbox"/>	Save ICE6200 (ICE62R) 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 (ICE62R) instruction
17	Self Diagnosis	#CHK <input type="checkbox"/>	Report results of ICE6200 (ICE62R) self diagnostic test

means press the RETURN key.

6 MASK DATA CHECKER MDC6003

6.1 MDC6003 Outline

The Mask Data Checker MDC6003 is a software tool which checks the program data (C003XXXH.HEX and C003XXXL.HEX) and option data (C003XXXF.DOC and C003XXXS.DOC) created by the user and creates the data file (C6003XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC6003 has the capability to restore the generated data file (C6003XXX.PAn) to the original file format.

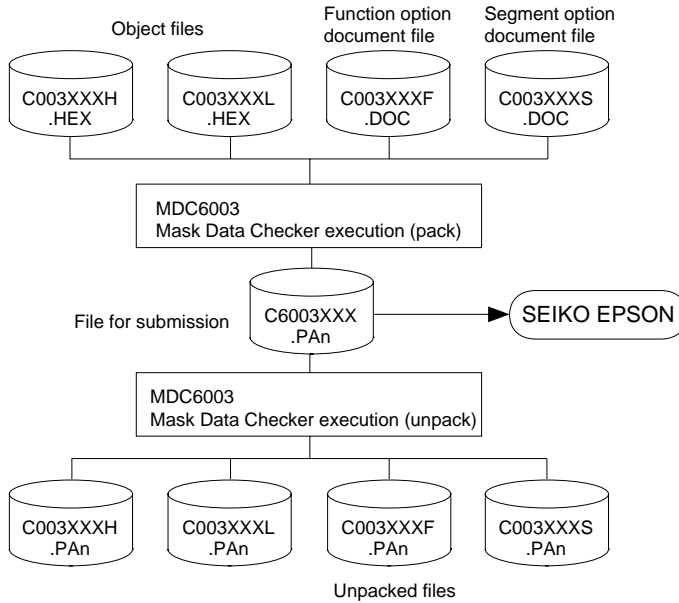


Fig. 6.1.1
MDC6003 execution flow

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

6.2 MDC6003 Quick Reference

■ Starting command and input/output files

Execution file: MDC6003.EXE

Starting command: **MDC6003** indicates the Return key.

Input file:	C003XXXL.HEX (Object file, low-order)] When packing
	C003XXXH.HEX (Object file, high-order)	
	C003XXXF.DOC (Function option document file)	
	C003XXXS.DOC (Segment option document file)	
	C6003XXX.PAn (Packed file)	

Output file:	C6003XXX.PAn (Packed file)] When packing
	C003XXXL.PAn (Object file, low-order)] When unpacking
	C003XXXH.PAn (Object file, high-order)	
	C003XXXF.PAn (Function option document file)	
	C003XXXS.PAn (Segment option document file)	

■ Display examples

```

*** E0C6003 PACK / UNPACK PROGRAM Ver 1.000 ***

EEEEEEEEEE PPPPPPPP SSSSSSS OOOOOOOO NNN NNN
EEEEEEEEEE PPPPPPPPP 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 PPPPPPPPP SSSSSSS OOO OOO NNN NNN NNN
EEEEEEEEEE PPPPPPPPP SSSSSSS OOO OOO NNN NNN NNN
EEE PPP SSS SSS OOO OOO NNN NNN NNN
EEE PPP SSS SSS OOO OOO NNN NNN NNN
EEEEEEEEEE PPP SSSS SSS OOO OOO NNN NNN
EEEEEEEEEE PPP SSSSSS OOOOOOOO NNN NN

(C) COPYRIGHT 1997 SEIKO EPSON CORP.

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

PLEASE SELECT NO.?
```

```

--- OPERATION MENU ---

      1. PACK
      2. UNPACK

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

C003XXXH.HEX -----+
C003XXXL.HEX -----+
C003XXXF.DOC -----+----- C6003XXX.PAn (PACK FILE)
C003XXS.DOC -----+

PLEASE INPUT PACK FILE NAME (C6003XXX.PAn) ? C60030A0.PA0 [ ] ... (2)

C0030A0H.HEX -----+
C0030A0L.HEX -----+
C0030A0F.DOC -----+----- C60030A0.PA0
C0030A0S.DOC -----+
```

Start-up message

When MDC6003 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.PACK" 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 "C6003XXX.PA0" has been submitted, the pack file name should be entered as "C6003XXX.PA1".)

With this, the mask file (C6003XXX.PAn) is generated, and the MDC6003 program will be terminated. Submit this file to Seiko Epson.

Note Don't use the data generated with the -N option of the Cross Assembler (ASM6003) 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 (C6003XXX.PAn) ? C60030A0.PA0 [ ] ... (2)

C60030A0.PA0 -----+----- C0030A0H.PA0
|-----+----- C0030A0L.PA0
|-----+----- C0030A0F.PA0
|-----+----- C0030A0S.PA0
```

Unpacking of data

- (1) Select "2.UNPACK" in the operation menu.
- (2) Enter the packed file name.

With this, the mask data file (C6003XXX.PAn) is restored to the original file format, and the MDC6003 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. E0C6003 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	

APPENDIX A. E0C6003 INSTRUCTION SET

Classification	Mne-monic	Operand	Operation Code								Flag			Clock	Operation					
			B	A	9	8	7	6	5	4	3	2	1			0	I	D	Z	C
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 ← M(SP), SP ← SP+1
		YL	1	1	1	1	1	1	0	1	1	0	0	1					5	YL ← M(SP), SP ← SP+1
		F	1	1	1	1	1	1	0	1	1	0	1	0	↑	↓	↑	↓	5	F ← M(SP), SP ← SP+1
	LD	SPH, r	1	1	1	1	1	1	1	0	0	0	r1	r0					5	SPH ← r
		SPL, r	1	1	1	1	1	1	1	1	0	0	r1	r0					5	SPL ← r
		r, SPH	1	1	1	1	1	1	1	0	0	1	r1	r0					5	r ← SPH
		r, SPL	1	1	1	1	1	1	1	1	0	1	r1	r0					5	r ← SPL
Arithmetic instructions	ADD	r, i	1	1	0	0	0	0	r1	r0	i3	i2	i1	i0	★	↑	↓		7	r ← r+i3~i0
		r, q	1	0	1	0	1	0	0	0	r1	r0	q1	q0	★	↑	↓		7	r ← r+q
	ADC	r, i	1	1	0	0	0	1	r1	r0	i3	i2	i1	i0	★	↑	↓		7	r ← r+i3~i0+C
		r, q	1	0	1	0	1	0	0	1	r1	r0	q1	q0	★	↑	↓		7	r ← r+q+C
	SUB	r, q	1	0	1	0	1	0	1	0	r1	r0	q1	q0	★	↑	↓		7	r ← r-q
	SBC	r, i	1	1	0	1	0	1	r1	r0	i3	i2	i1	i0	★	↑	↓		7	r ← r-i3~i0-C
		r, q	1	0	1	0	1	0	1	1	r1	r0	q1	q0	★	↑	↓		7	r ← r-q-C
	AND	r, i	1	1	0	0	1	0	r1	r0	i3	i2	i1	i0		↑	↓		7	r ← r∧i3~i0
		r, q	1	0	1	0	1	1	0	0	r1	r0	q1	q0		↑	↓		7	r ← r∧q
	OR	r, i	1	1	0	0	1	1	r1	r0	i3	i2	i1	i0		↑	↓		7	r ← r∨i3~i0
		r, q	1	0	1	0	1	1	0	1	r1	r0	q1	q0		↑	↓		7	r ← r∨q
	XOR	r, i	1	1	0	1	0	0	r1	r0	i3	i2	i1	i0		↑	↓		7	r ← r∨i3~i0
		r, q	1	0	1	0	1	1	1	0	r1	r0	q1	q0		↑	↓		7	r ← r∨q
	CP	r, i	1	1	0	1	1	1	r1	r0	i3	i2	i1	i0		↑	↓		7	r-i3~i0
		r, q	1	1	1	1	0	0	0	0	r1	r0	q1	q0		↑	↓		7	r-q
	FAN	r, i	1	1	0	1	1	0	r1	r0	i3	i2	i1	i0		↑	↓		7	r∧i3~i0
		r, q	1	1	1	1	0	0	0	1	r1	r0	q1	q0		↑	↓		7	r∧q
	RLC	r	1	0	1	0	1	1	1	1	r1	r0	r1	r0		↑	↓		7	d3 ← d2, d2 ← d1, d1 ← d0, d0 ← C, C ← d3
	RRC	r	1	1	1	0	1	0	0	0	1	1	r1	r0		↑	↓		5	d3 ← C, d2 ← d3, d1 ← d2, d0 ← d1, C ← d0
	INC	Mn	1	1	1	1	0	1	1	0	n3	n2	n1	n0		↑	↓		7	M(n3~n0) ← M(n3~n0)+1
	DEC	Mn	1	1	1	1	0	1	1	1	n3	n2	n1	n0		↑	↓		7	M(n3~n0) ← M(n3~n0)-1
	ACPX	MX, r	1	1	1	1	0	0	1	0	1	0	r1	r0	★	↑	↓		7	M(X) ← M(X)+r+C, X ← X+1
	ACPY	MY, r	1	1	1	1	0	0	1	0	1	1	r1	r0	★	↑	↓		7	M(Y) ← M(Y)+r+C, Y ← Y+1
	SCPX	MX, r	1	1	1	1	0	0	1	1	1	0	r1	r0	★	↑	↓		7	M(X) ← M(X)-r-C, X ← X+1
	SCPY	MY, r	1	1	1	1	0	0	1	1	1	1	r1	r0	★	↑	↓		7	M(Y) ← M(Y)-r-C, Y ← Y+1
	NOT	r	1	1	0	1	0	0	r1	r0	1	1	1	1		↑	↓		7	r ← \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. E0C6003 I/O MEMORY MAP

Address	Register				Name	Init *1	1	0	Comment
	D3	D2	D1	D0					
0F0H	K03	K02	K01	K00	K03	-*2	High	Low	K0 input port data
					K02	-*2	High	Low	
	R				K01	-*2	High	Low	
					K00	-*2	High	Low	
0F1H	R03	R02	R01	R00	R03	0	High	Low	R03 output port data
				FOUT	R02	0	High	Low	R02 output port data
			BUZZER	BUZZER	R01	0	High	Low	R01 output port data
	R/W				BUZZER	0	On	Off	Buzzer output On/Off control
					R00	0	High	Low	R00 output port data
					FOUT	0	On	Off	FOUT output On/Off control
					BUZZER	0	On	Off	Buzzer inverted output On/Off control
0F2H	TM3	TM2	TM1	TM0	TM3	-*2			Clock timer data (2 Hz)
					TM2	-*2			Clock timer data (4 Hz)
	R				TM1	-*2			Clock timer data (8 Hz)
					TM0	-*2			Clock timer data (16 Hz)
0F3H	EIK03	EIK02	EIK01	EIK00	EIK03	0	Enable	Mask	Interrupt mask register (K03)
	R/W				EIK02	0	Enable	Mask	Interrupt mask register (K02)
					EIK01	0	Enable	Mask	Interrupt mask register (K01)
					EIK00	0	Enable	Mask	Interrupt mask register (K00)
0F4H	TMRST	EIT2	EIT16	EIT32	TMRST*3	Reset	Reset	-	Clock timer reset
	W	R/W			EIT2	0	Enable	Mask	Interrupt mask register (clock timer 2 Hz)
					EIT16	0	Enable	Mask	Interrupt mask register (clock timer 16 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (clock timer 32 Hz)
0F5H	0	0	0	IK0	0*3	-*2	-	-	Unused
	R				0*3	-*2	-	-	Unused
					0*3	-*2	-	-	Unused
					IK0*4	0	Yes	No	Interrupt factor flag (K00-K03)
0F6H	0	IT2	IT16	IT32	0*3	-*2	-	-	Unused
	R				IT2*4	0	Yes	No	Interrupt factor flag (clock timer 2 Hz)
					IT16*4	0	Yes	No	Interrupt factor flag (clock timer 16 Hz)
					IT32*4	0	Yes	No	Interrupt factor flag (clock timer 32 Hz)
0F7H	XBZR	0	XFOUT1	XFOUT0	XBZR	0	2 kHz	4 kHz	Buzzer frequency control
	R/W				0*3	-*2	-	-	Unused
	R	R	R/W		XFOUT1	0			FOUT frequency control
					XFOUT0	0			0: F1, 1: F2, 2: F3, 3: F4
0F8H	LON	0	0	CSDC	LON	0	On	Off	LCD power and display On/Off control
	R/W				0*3	-*2	-	-	Unused
	R				0*3	-*2	-	-	Unused
					CSDC	0	Static	Dynamic	LCD drive switch

*1 Initial value at initial reset

*3 Always "0" being read

*2 Not set in the circuit

*4 Reset (0) immediately after being read

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE6200 (ICE62R)	Nothing appears on the screen, or nothing works, after activation.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Is the RS-232C cable connected correctly? • Is the RS-232C driver installed? • Is SPEED.COM or MODE.COM on the disk? • Is the execution file correct? MS-DOS ICS6003J.EXE PC-DOS ICS6003W.EXE • Is the DOS version correct? MS-DOS Ver. 3.1 or later PC-DOS Ver. 2.1 or later • Is the DIP switches that set the baud rate of the main ICE6200 (ICE62R) unit set correctly? • Is the breaker of the ICE6200 (ICE62R) set to ON?
	The ICE6200 breaker tripped or the ICE62R fuse cut immediately after activation.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Are connectors F1 and F5 connected to the EVA6237 correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS6003P.PAR is being used. Use the latest version.
	Immediate values A (10) and B (11) cannot be entered correctly with the A command.	<p>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).</p> <p><i>Example:</i> 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.</p>
	<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 a 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.	
SOG6003	An R error occurs although the address is correctly set in the segment source file.	<p>Check the following and remedy if necessary:</p> <ul style="list-style-type: none"> • Does the address symbol use capital letters? • Are the output ports set for every two terminals?

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ASM6003	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.
MDC6003	Activation is impossible.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Is the number of files set at ten or more in OS environment file CONFIG.SYS?
EVA6237	The EVA6237 does not work when it is used independently.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target? • Is the EPROM for F.HEX and S.HEX installed correctly? • Is the appropriate voltage being supplied? (5V DC, 3A, or more) • Are the program ROMs (H and L) installed correctly? • Is data written from address 4000H? (When the 27C256 is used as the program ROM) • Is the EN/DIS switch on the EVA6237 set to EN?
	Target segment does not light.	Check the following and remedy if necessary: <ul style="list-style-type: none"> • Is an EPROM with an access time of 250 ns or less being used for S.HEX. • Has the VADJ VR inside the EVA6237 top cover been turned to a lower setting?

APPENDIX E. DEBUGGING FOR 1/2 DUTY DRIVE ON EVA6237

The EVA6237 is used to debug E0C6003 systems. However, the EVA6237 cannot output 1/2 duty signal to drive LCD. When debugging a E0C6003 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 FOG6003.
2. Assign the segments using the SOG6003.
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    E00   E01   ---   ---   S
  1    E02   E03   ---   ---   S
  2    E10   E11   ---   ---   S
  3    E12   E13   ---   ---   S
  4    E20   E21   ---   ---   S
  :
  :
  :
 13    EA0   EB0   ---   ---   S
 14    EC0   EC1   ---   ---   S
```

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

3. Debug with the EVA6237 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. 2) using the FOG6003 to change the 1/3 duty setting to the 1/2 duty setting.
2. It is unnecessary to change the segment assignment by the SOG6003.

Note: The EVA6237 does not support the 1/2 bias LCD drive method. The LCD drive waveform is a 1/3 bias waveform.

EPSON International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

- HEADQUARTERS -

1960 E. Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 Fax: +1-310-955-5400

- SALES OFFICES -

West

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

Central

101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 Fax: +1-815-455-7633

Northeast

301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 Fax: +1-781-246-5443

Southeast

3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 Fax: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH

- HEADQUARTERS -

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

- GERMANY -

SALES OFFICE

Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)2171-5045-0 Fax: +49-(0)2171-5045-10

- UNITED KINGDOM -

UK BRANCH OFFICE

Unit 2.4, Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

- FRANCE -

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

- CHINA -

EPSON (CHINA) CO., LTD.

28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 Fax: 64107319

SHANGHAI BRANCH

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

- 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

- TAIWAN -

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

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

HsinCHU OFFICE

13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300, TAIWAN
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

50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, 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 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 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

■ EPSON Electronic Devices Website
<http://www.epson.co.jp/device/>

First issue DECEMBER 1997, Printed JULY 1999 in Japan ® A