MF999-03

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C623B**

DEVELOPMENT TOOL MANUAL

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

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. 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".

E0C623B Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C623B.

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.

Development tools	☞ E0C62 Family Development Tool Reference Manual EVA623B Manual ICE62R (ICE6200) Hardware Manual
Development procedure	☞ E0C62 Family Technical Guide
Device (E0C623B)	☞ E0C623B Technical Manual
Instructions	☞ E0C6200/6200A Core CPU Manual

CONTENTS

1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL 1
1.1 Configuration of DEV623B 1
1.2 Developmental Environment 1
1.3 Development Flow 2
1.4 Production of Execution Disk 3

2 CROSS ASSEMBLER ASM623B 4
2.1 ASM623B Outline 4
2.2 E0C623B Restrictions 4
2.3 ASM623B Quick Reference 5

3 FUNCTION OPTION GENERATOR FOG623B 7
3.1 FOG623B Outline 7
3.2 E0C623B Option List 7
3.3 FOG623B Quick Reference 10
3.4 Sample File 12

4 SEGMENT OPTION GENERATOR SOG623B 14
4.1 SOG623B Outline 14
4.2 Option List 14
4.3 Segment Ports Output Specifications 15
4.4 SOG623B Quick Reference 16
4.5 Sample Files 18

5 ICE CONTROL SOFTWARE ICS623B 19
5.1 ICS623B Outline 19
5.2 ICS623B Restrictions 19
5.3 ICS623B Quick Reference 21

6 MASK DATA CHECKER MDC623B ______ 24

6.1 MDC623B Outline 24

6.2 MDC623B Quick Reference 24

APPENDIX A. E0C623B INSTRUCTION SET ______ 27

B. TROUBLESHOOTING ______ 31

1 COMPOSITION OF DEVELOPMENT SUPPORT TOOL

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

1.1 Configuration of DEV623B

The below software are included in the product of the E0C623B development support tool DEV623B.

1. Cross Assembler ASM623B .... Cross assembler for program preparation
2. Function Option Generator FOG623B .... Function option data preparation program
3. Segment Option Generator SOG623B .... Segment option data preparation program
4. ICE Control Software ICS623B .... ICE control program
5. Mask Data Checker MDC623B .... Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV623B 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 E0C623B, 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.

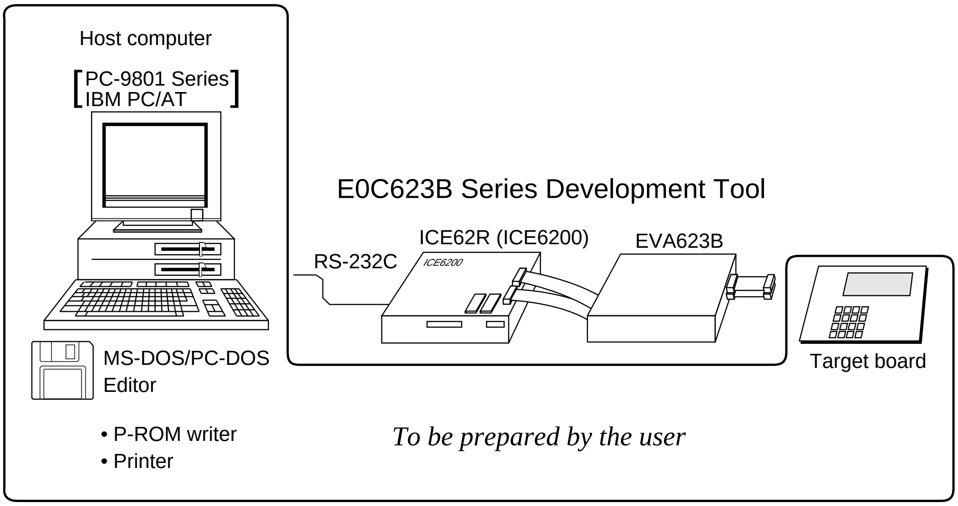


Fig. 1.2.1 System configuration

Note The DEV623B system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE62R (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 DEV623B.

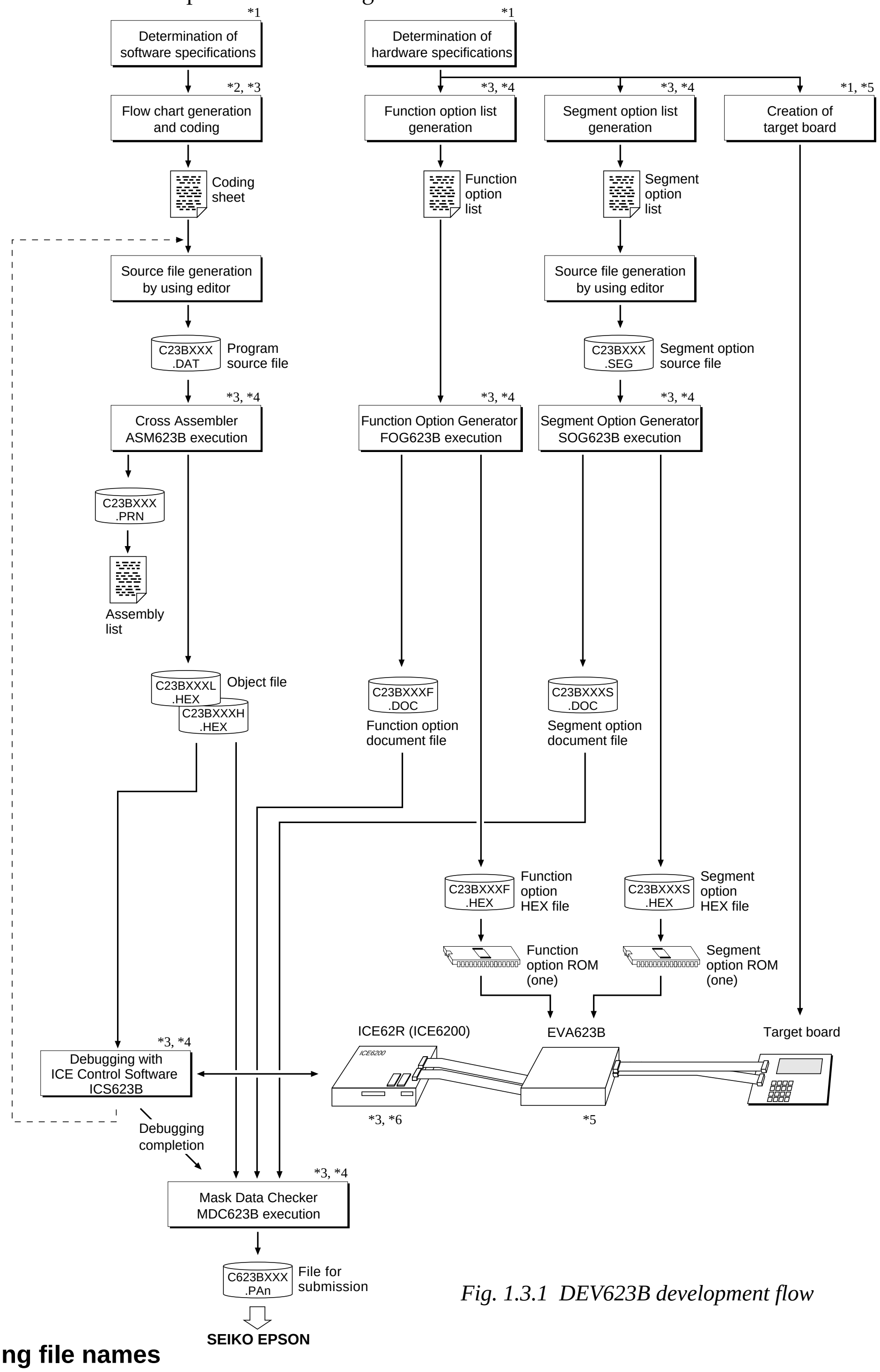

*1
Determination of software specifications
*1
Determination of hardware specifications
*2, *3
Flow chart generation and coding
*3, *4
Function option list generation
*3, *4
Segment option list generation
*1, *5
Creation of target board
Coding sheet
Function option list
Segment option list
Source file generation by using editor
Source file generation by using editor
C23BXXX .DAT
Program source file
C23BXXX .SEG
Segment option source file
*3, *4
Cross Assembler ASM623B execution
*3, *4
Function Option Generator FOG623B execution
*3, *4
Segment Option Generator SOG623B execution
C23BXXX .PRN
Assembly list
C23BXXXL .HEX
C23BXXXH .HEX
Object file
C23BXXXF .DOC
Function option document file
C23BXXXS .DOC
Segment option document file
C23BXXXF .HEX
Function option HEX file
C23BXXXS .HEX
Segment option HEX file
Function option ROM (one)
Segment option ROM (one)
ICE62R (ICE6200)
EVA623B
Target board
ICE6200
*3, *4
Debugging with ICE Control Software ICS623B
*3, *6
*5
Debugging completion
*3, *4
Mask Data Checker MDC623B execution
C623BXXX .PAn
File for submission
SEIKO EPSON

Fig. 1.3.1 DEV623B development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C23BXXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1	E0C623B Technical Hardware Manual	*4	E0C623B Development Tool Manual (this manual)
*2	E0C6200/6200A Core CPU Manual	*5	EVA623B Manual
*3	E0C62 Family Development Tool Reference Manual	*6	ICE62R (ICE6200) Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE62R (ICE6200) are recorded in the DEV623B floppy disk.
The content of the files contained in the DEV623B floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM623B.EXE	ASM623B.EXE	Cross Assembler execution file
FOG623B.EXE	FOG623B.EXE	Function Option Generator execution file
ICS623BB.BAT	ICS623B.BAT	ICE Control Software batch file
ICS623BW.EXE	ICS623BJ.EXE	ICE Control Software execution file
ICS623BP.PAR	ICS623BP.PAR	ICE Control Software parameter file
MDC623B.EXE	MDC623B.EXE	Mask Data Checker execution file
SOG623B.EXE	SOG623B.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 subdirectory with an appropriate name (DEV623B, etc.) then copy the content of the floppy disk into that subdirectory using the COPY 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 (DEV623B), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV623B↵
C\>CD DEV623B↵
C\DEV623B\>COPY A:*.*↵
```

■ 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 MDC623B will handle many files.

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

Example:

Setting of FILES (CONFIG.SYS)

```
C\>TYPE  CONFIG.SYS↵
      :
FILES=20
      :
```

■ In "ICS623BB.BAT" the batch process is indicated such that the ICS623BJ(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.

RS-232C Setting (PC-DOS version)

```
MODE  COM1:  9600, n, 8, 1, p
```

RS-232C Setting (MS-DOS version)

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

2 CROSS ASSEMBLER ASM623B

2.1 ASM623B Outline

The ASM623B cross assembler is an assembler program for generating the machine code used by the E0C623B 4-bit, single-chip microcomputers. The Cross Assembler ASM623B 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.

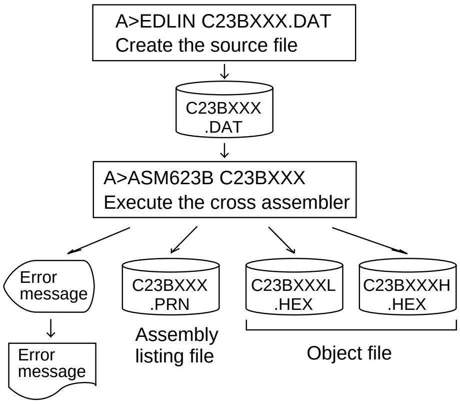


Fig. 2.1.1 ASM623B execution flow

☞ *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 E0C623B Restrictions

Note the following when generating a program by the E0C623B:

■ ROM area

The capacity of the E0C623B ROM is 1,536 steps (0000H to 05FFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank: Only bank 0, Page: 6 pages (0 to 5H), each 256 steps

Significant specification range:

ORG	pseudo-instruction:	0000H to 05FFH
PAGE	pseudo-instruction:	00H to 05H
BANK	pseudo-instruction:	Only 0H
PSET	instruction:	00H to 05H

■ RAM area

The capacity of the E0C623B RAM is 96 words (000H to 05FH, 4 bits/word). Memory access is invalid when the unused area of the index register is specified.

Example: `LD X,080H` 80H 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,0B7H` B7H 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 E0C623B instruction sets.

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

In addition to these instructions, the SLP instruction cannot be used when the E0C623B crystal oscillation circuit is selected by mask option.

2.3 ASM623B Quick Reference

_ indicates a blank.
↵ indicates the Return key.
A parameter enclosed by [ ] can be omitted.

■ Starting command and input/output files

Execution file: ASM623B.EXE

Starting command: **ASM623B_ [drive-name:] source-file-name [.shp]_ [-N]↵**

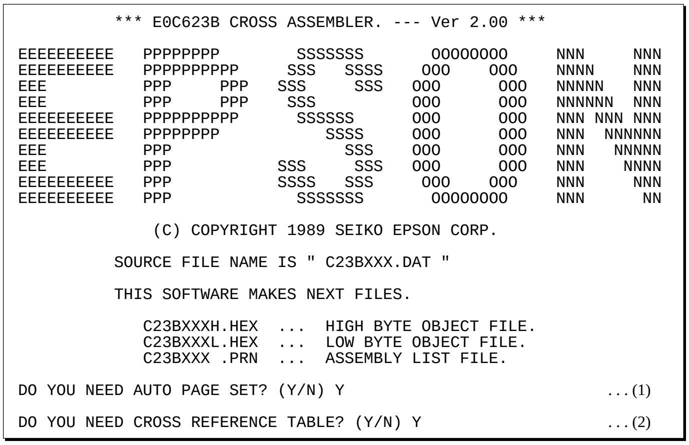
Option:

.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: C23BXXX.DAT (Source file)

Output file:
C23BXXXL.HEX (Object file, low-order)
C23BXXXH.HEX (Object file, high-order)
C23BXXX.PRN (Assembly listing file)

■ Display example



```
          *** E0C623B CROSS ASSEMBLER. --- Ver 2.00 ***

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

              (C) COPYRIGHT 1989 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C23BXXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C23BXXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C23BXXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C23BXXX .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 ASM623B 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, ASM623B 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 LOOP CP MX,DATA JP NZ,LOOP ENDM CHECK 1
LOCAL (Local)	To make local specification of label during macro definition	
ENDM (End Macro)	To end macro definition	

■ 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)	• 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)	• 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 FOG623B

3.1 FOG623B Outline

With the 4-bit single-chip E0C623B microcomputers, the customer may select 15 hardware options. By modifying the mask patterns of the E0C623B according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG623B 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 FOG623B, the E0C623B mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA623B) 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.*

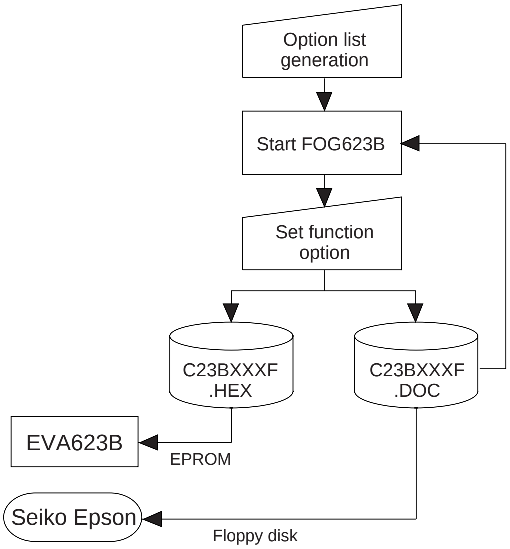


Fig. 3.1.1 FOG623B execution flow

3.2 E0C623B Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "Technical Manual" 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. E0C623B (Low speed model: 30 kHz–90 kHz)
□ 2. E0C62A3B (High speed model: 250 kHz–1 MHz)

2. SYSTEM CLK & SLEEP

When "1. (E0C623B)" is selected in Option 1.

□ 1. Crystal oscillation (No SLEEP function)
□ 2. CR oscillation without SLEEP function
□ 3. CR oscillation with SLEEP function

When "2. (E0C62A3B)" is selected in Option 1.

□ 1. CR oscillation without SLEEP function
□ 2. CR oscillation with SLEEP function
□ 3. Ceramic oscillation without SLEEP function
□ 4. Ceramic oscillation with SLEEP function

3. MULTIPLE KEY ENTRY RESET

• KEY Combination ...... □ 1. Not Use
□ 2. Use K00 + K01
□ 3. Use K00 + K01 + K02
□ 4. Use K00 + K01 + K02 + K03

4. RESET PULL-UP RESISTOR

□ 1. With Resistor □ 2. Gate Direct

5. INTERRUPT NOISE REJECTOR

- K00–K03 .... □ 1. Use □ 2. Not Use
- P01, P03 .... □ 1. Use □ 2. Not Use

(The interrupt function of the P01 and P03 ports is available only when the port is set as an input only port by selecting with Option 13, "P Port Specification". When they are set to I/O ports, the interrupt function cannot be used.)

6. INPUT PORT PULL-UP 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

7. R PORT OUTPUT SPECIFICATION

- R00 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R01 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R02 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R03 .... □ 1. Complementary □ 2. Nch-OpenDrain

8. R00 SPECIFICATION

□ 1. DC output
□ 2. /BZ output
□ 3. /FOUT output

9. FOUT FREQUENCY

(FOSC is the OSC1 oscillation frequency.)

When "1. (E0C623B)" is selected in Option 1, and "3. /FOUT output" is selected in Option 8.

- F1 □ 1. FOSC/128
  □ 2. FOSC/64
  □ 3. FOSC/32
  □ 4. FOSC/16
  □ 5. FOSC/8
- F2 □ 1. FOSC/64
  □ 2. FOSC/32
  □ 3. FOSC/16
  □ 4. FOSC/8
  □ 5. FOSC/4
- F3 □ 1. FOSC/32
  □ 2. FOSC/16
  □ 3. FOSC/8
  □ 4. FOSC/4
  □ 5. FOSC/2
- F4 □ 1. FOSC/16
  □ 2. FOSC/8
  □ 3. FOSC/4
  □ 4. FOSC/1
  □ 5. FOSC/1

When "2. (E0C62A3B)" is selected in Option 1, "1 or 2. (CR oscillation)" is selected in Option 2, and "3. /FOUT output" is selected in Option 8.

- F1 □ 1. FOSC/768
  □ 2. FOSC/384
  □ 3. FOSC/192
  □ 4. FOSC/96
  □ 5. FOSC/48
- F2 □ 1. FOSC/384
  □ 2. FOSC/192
  □ 3. FOSC/96
  □ 4. FOSC/48
  □ 5. FOSC/24
- F3 □ 1. FOSC/192
  □ 2. FOSC/96
  □ 3. FOSC/48
  □ 4. FOSC/24
  □ 5. FOSC/12
- F4 □ 1. FOSC/96
  □ 2. FOSC/48
  □ 3. FOSC/24
  □ 4. FOSC/6
  □ 5. FOSC/1

When "2. (E0C62A3B)" is selected in Option 1, "3 or 4. (Ceramic oscillation)" is selected in Option 2, and "3. /FOUT output" is selected in Option 8.

- F1 □ 1. FOSC/1536
  □ 2. FOSC/768
  □ 3. FOSC/384
  □ 4. FOSC/192
  □ 5. FOSC/96
- F2 □ 1. FOSC/768
  □ 2. FOSC/384
  □ 3. FOSC/192
  □ 4. FOSC/96
  □ 5. FOSC/48
- F3 □ 1. FOSC/384
  □ 2. FOSC/192
  □ 3. FOSC/96
  □ 4. FOSC/48
  □ 5. FOSC/24
- F4 □ 1. FOSC/192
  □ 2. FOSC/96
  □ 3. FOSC/48
  □ 4. FOSC/12
  □ 5. FOSC/1

10. R01 SPECIFICATION

□ 1. DC output
□ 2. BZ output (off Level High)
□ 3. BZ output (off Level Low)

11. P PORT PULL-UP RESISTOR

- P00 .... □ 1. With Resistor □ 2. Gate Direct
- P01 .... □ 1. With Resistor □ 2. Gate Direct
- P02 .... □ 1. With Resistor □ 2. Gate Direct
- P03 .... □ 1. With Resistor □ 2. Gate Direct

12. P PORT OUTPUT SPECIFICATION

- P00 .... □ 1. Complementary □ 2. Nch-OpenDrain
- P01 .... □ 1. Complementary □ 2. Nch-OpenDrain
- P02 .... □ 1. Complementary □ 2. Nch-OpenDrain
- P03 .... □ 1. Complementary □ 2. Nch-OpenDrain

13. P PORT SPECIFICATION

(Select a combination of serial interface ports, I/O ports and input only ports.)

	P00	P01	P02	P03
□ 1.	I/O	I/O	I/O	I/O
□ 2.	I/O	I/O	I/O	I
□ 3.	SIN	SOUT	/SCLK	/SRDY
□ 4.	SIN	SOUT	/SCLK	I/O
□ 5.	SIN	SOUT	/SCLK	I
□ 6.	SIN	I/O	/SCLK	I/O
□ 7.	SIN	I/O	/SCLK	I
□ 8.	SIO	MODE	/SCLK	/SRDY
□ 9.	SIO	MODE	/SCLK	I/O
□ 10.	SIO	MODE	/SCLK	I
□ 11.	SIO	I/O	/SCLK	/SRDY
□ 12.	SIO	I	/SCLK	/SRDY
□ 13.	SIO	I/O	/SCLK	I/O
□ 14.	SIO	I/O	/SCLK	I

I/O: I/O port (Input interrupt noise rejector (Option 5) not available)
SIN: Serial data input port
SIO: Serial data input/output port
SOUT: Serial data output port
MODE: SIO direction monitor port
/SCLK: Serial clock input/output port
I: Input only port (Input interrupt noise rejector available)

14. SERIAL INTERFACE SPECIFICATION

- Serial data .... □ 1. LSB first □ 2. MSB first
- SOUT initial level...... □ 1. HIGH level □ 2. LOW level

15. LCD DRIVER OUTPUT SPECIFICATION

- Duty .... □ 1. 1/4 duty
  □ 2. 1/3 duty
  □ 3. 1/2 duty
- LCD power source .... □ 1. Internal regulator (V_{C1} = 1.0 V)
  □ 2. External (V_{C1} = V_{DD} = 1.5 V)
  □ 3. External (V_{C2} = V_{DD} = 3.0 V)
  □ 4. External (V_{C3} = V_{DD} = 3.0 V)

3.3 FOG623B Quick Reference

■ Starting command and input/output files

Execution file: FOG623B.EXE

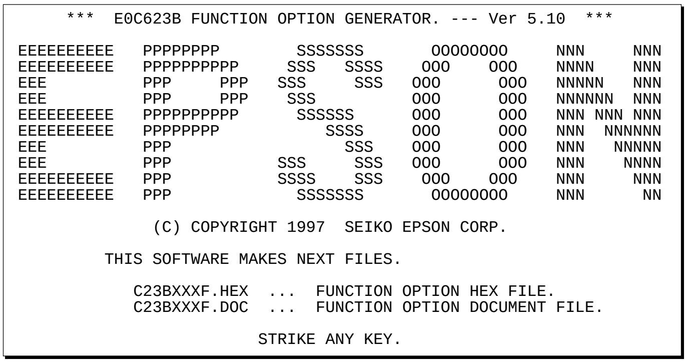
Starting command: **FOG623B↵**

↵ indicates the Return key.

Input file: C23BXXXF.DOC (Function option document file, when modifying)

Output file: C23BXXXF.DOC (Function option document file)
C23BXXXF.HEX (Function option HEX file)

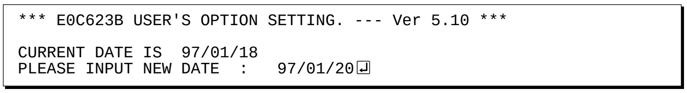
■ Display example



```
     ***  E0C623B FUNCTION OPTION GENERATOR. --- Ver 5.10  ***

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

              (C) COPYRIGHT 1997  SEIKO EPSON CORP.

         THIS SOFTWARE MAKES NEXT FILES.

            C23BXXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C23BXXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

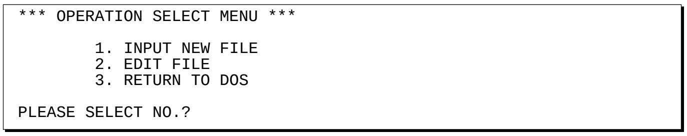
Start-up message
When FOG623B is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

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



```
*** E0C623B USER'S OPTION SETTING. --- Ver 5.10 ***

CURRENT DATE IS  97/01/18
PLEASE INPUT NEW DATE  :   97/01/20↵
```

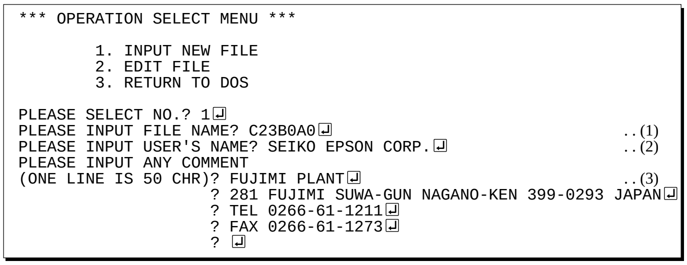
Date input
Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/").
When not modifying the date, press the RETURN key "↵" to continue.



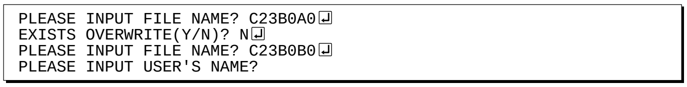
```
*** OPERATION SELECT MENU ***

        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 FOG623B.



```
*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C23B0A0↵                               ..(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↵
                    ? ↵
```



```
PLEASE INPUT FILE NAME? C23B0A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C23B0B0↵
PLEASE INPUT USER'S NAME?
```

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 overwrition is desired. Enter "Y" or "N" accordingly.

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.

```
*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.? 2↵

*** SOURCE FILE(S) ***

C23B0A0         C23B0B0         C23B0C0                ..(1)

PLEASE INPUT FILE NAME? C23B0A0↵                       ..(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? C23B0N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
```

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.

```
*** OPTION  NO.1 ***

--- DEVICE TYPE ---

                           1. E0C623B
                           2. E0C62A3B

PLEASE SELECT  NO.(1) ? 1↵

       KEY COMBINATION      1. E0C623B
```

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 EVA623B, 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 EVA623B options.

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

```
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(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
```

3.4 Sample File

■ Example of function option document file

```
* E0C623B FUNCTION OPTION DOCUMENT  V 5.10
*
* FILE NAME    C23B0A0.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   1997/02/13
*
*
*
* OPTION NO.1
* < DEVICE TYPE >
*                            E0C623B  ---------------------  SELECTED
 OPT0101 01
*
* OPTION NO.2
* < SYSTEM CLK & SLEEP >
*                            CRYSTAL  ---------------------  SELECTED
 OPT0201 01
 OPT0202 01
 OPT0203 01
*
* OPTION NO.3
* < MULTIPLE KEY ENTRY RESET >
*                            USE K00 & K01 & K02 & K03  ---  SELECTED
 OPT0301 04
*
* OPTION NO.4
* < RESET PULL-UP RESISTOR >
*                            USE  -------------------------  SELECTED
 OPT0401 01
*
* OPTION NO.5
* < INTERRUPT NOISE REJECTOR >
* < K PORT>
*                            USE  -------------------------  SELECTED
 OPT0501 01
* < P PORT>
*                            USE  -------------------------  SELECTED
 OPT0502 01
*
* OPTION NO.6
* < INPUT PORT PULL-UP RESISTOR >
*     K00                    WITH RESISTOR  ---------------  SELECTED
*     K01                    WITH RESISTOR  ---------------  SELECTED
*     K02                    WITH RESISTOR  ---------------  SELECTED
*     K03                    WITH RESISTOR  ---------------  SELECTED
 OPT0601 01
 OPT0602 01
 OPT0603 01
 OPT0604 01
*
* OPTION NO.7
* < R PORT OUTPUT SPECIFICATION >
*     R00                    COMPLEMENTARY  ---------------  SELECTED
*     R01                    COMPLEMENTARY  ---------------  SELECTED
*     R02                    NCH-OPENDRAIN  ---------------  SELECTED
*     R03                    NCH-OPENDRAIN  ---------------  SELECTED
 OPT0701 01
 OPT0702 01
 OPT0703 02
 OPT0704 02
*
* OPTION NO.8
* < R00 SPECIFICATION >
*                            /FOUT  -----------------------  SELECTED
 OPT0801 03
*
* OPTION NO.9
* < FOUT FREQUENCY >
* <F1>
*                            FOSC/128  --------------------  SELECTED
* <F2>
*                            FOSC/32  ---------------------  SELECTED
* <F3>
*                            FOSC/8  ----------------------  SELECTED
* <F4>
*                            FOSC/1  ----------------------  SELECTED
 OPT0901 01
 OPT0902 02
 OPT0903 03
 OPT0904 05
*
* OPTION NO.10
* < R01 SPECIFICATION >
*                            BZ <OFF LEVEL LOW >   ---------  SELECTED
 OPT1001 03
*
* OPTION NO.11
* < P PORT PULL-UP RESISTOR >
*     P00                    WITH RESISTOR  ---------------  SELECTED
*     P01                    WITH RESISTOR  ---------------  SELECTED
*     P02                    WITH RESISTOR  ---------------  SELECTED
*     P03                    WITH RESISTOR  ---------------  SELECTED
 OPT1101 01
 OPT1102 01
 OPT1103 01
 OPT1104 01
*
* OPTION NO.12
* < P PORT OUTPUT SPECIFICATION >
*     P00                    COMPLEMENTARY  ---------------  SELECTED
*     P01                    COMPLEMENTARY  ---------------  SELECTED
*     P02                    COMPLEMENTARY  ---------------  SELECTED
*     P03                    COMPLEMENTARY  ---------------  SELECTED
 OPT1201 01
 OPT1202 01
 OPT1203 02
 OPT1204 02
*
* OPTION NO.13
* < P PORT SPECIFICATION >
*     P00                    SIN  -------------------------  SELECTED
*     P01                    SOUT  ------------------------  SELECTED
*     P02                    /SCLK  -----------------------  SELECTED
*     P03                    I  ---------------------------  SELECTED
 OPT1301 02
 OPT1302 02
 OPT1303 02
 OPT1304 03
 OPT1305 04
*
* OPTION NO.14
* < SIO SPECIFICATION >
* < SERIAL DATA ORDER >
*                            LSB FIRST  -------------------  SELECTED
* < SOUT INITIAL LEVEL >
*                            HIGH LEVEL  ------------------  SELECTED

 OPT1401 01
 OPT1402 01
*
* OPTION NO.15
* < LCD COMMON DUTY >
*                            1/4 DUTY  --------------------  SELECTED
* < LCD POWER >
*                            INTERNAL  --------------------  SELECTED
 OPT1501 01
 OPT1502 01
 OPT1503 01
*
* SEIKO EPSON’S AREA
*
\\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 SOG623B

4.1 SOG623B Outline

With the 4-bit single-chip E0C623B microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C623B according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG623B is a software tool for generating data file used to generate mask patterns. From the data file created with SOG623B, the E0C623B mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA623B) 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.*

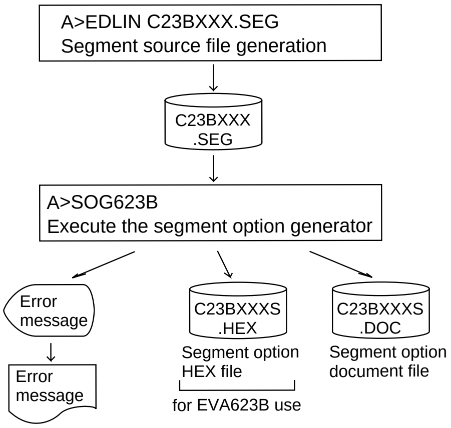


Fig. 4.1.1 SOG623B execution flow

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 □ C □ N
SEG2													SEG output
SEG3													DC output □ C □ N
SEG4													SEG output
SEG5													DC output □ C □ N
SEG6													SEG output
SEG7													DC output □ C □ N
SEG8													SEG output
SEG9													DC output □ C □ N
SEG10													SEG output
SEG11													DC output □ C □ N
SEG12													SEG output
SEG13													DC output □ C □ N
SEG14													SEG output
SEG15													DC output □ C □ N
SEG16													SEG output
SEG17													DC output □ C □ N
SEG18													SEG output
SEG19													DC output □ C □ N
SEG20													SEG output
SEG21													DC output □ C □ N
SEG22													SEG output
SEG23													DC output □ C □ N
SEG24													SEG output
SEG25													DC output □ C □ N

Legend: <ADDRESS>
H: High order address, L: Low order address
D: Data bit

<OUTPUT SPECIFICATION>
C: Complementary output
N: Nch 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–SEG25, segment output and DC output can be selected in units of two terminals (SEG0 & SEG1, SEG2 & SEG3...). 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 Nch open drain output may be selected for the output specification.
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 104 segments (78 segments when 1/3 duty is selected or 52 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 26 terminals may be allocated for DC output. 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 SEG20 and SEG21.

```
20  A40  ---  ---  ---  C
21  A50  ---  ---  ---  C
```

4.4 SOG623B Quick Reference

■ Starting command and input/output files

_ indicates a blank.
↵ indicates the Return key.
A parameter enclosed by [ ] can be omitted.

Execution file: SOG623B.EXE

Starting command: **SOG623B_ [-H]↵**

Option: -H: Specifies the segment option document file for input file of SOG623B.

Input file: C23BXXX.SEG (Segment option source file)
C23BXXXS.DOC (Segment option document file, when -H option use)

Output file: C23BXXXS.DOC (Segment option document file)
C23BXXXS.HEX (Segment option HEX file)

■ Display example

```
     ***  E0C623B SEGMENT OPTION GENERATOR. --- Ver 3.21  ***

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

              (C) COPYRIGHT 1997  SEIKO EPSON CORP.

         SEGMENT OPTION SOURCE FILE NAME IS " C23BXXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C23BXXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C23BXXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

Start-up message
When SOG623B 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.

```
*** E0C623B USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  97/01/18
PLEASE INPUT NEW DATE  :   97/01/20↵
```

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) ***

C23B0A0         C23B0B0         C23B0C0                       ..(1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C23B0A0↵               ..(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↵
                    ? ↵
```

```
*** 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? C23B0N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.      ..(7) -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C23B0N0↵
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.    ..(8) -H option use
```

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 EVA623B, 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 EVA623B options.

When a series of operations are complete, the SOG623B generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG623B 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

```
; C23B0A2.SEG, VER.3.21
; SOG623B LCD SEGMENT DECODE TABLE
;
 0    900  901  912  902 S
 1    913  911  910  903 S
 2    920  921  932  922 S
 3    933  931  930  923 S
 4    940  941  952  942 S
 5    953  951  950  943 S
 6    960  961  972  962 S
 7    973  971  970  963 S
 8    980  981  992  982 S
 9    993  991  990  983 S
10    9A0  9A1  9B2  9A2 S
11    9B3  9B1  9B0  9A3 S
12    9C0  9C1  9D2  9C2 S
13    9D3  9D1  9D0  9C3 S
14    9E0  9E1  9F2  9E2 S
15    9F3  9F1  9F0  9E3 S
16    A00  A01  A12  A02 S
17    A13  A11  A10  A03 S
18    A20  A21  A32  A22 S
19    A33  A31  A30  A23 S
20    A40  A41  A52  A42 S
21    A53  A51  A50  A43 S
22    A60  A61  A72  A62 S
23    A73  A71  A70  A63 S
24    A80  ---  ---  --- C
25    A90  ---  ---  --- C
```

■ Example of segment option document file

```
* E0C623B SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C23B0A2S.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   97/01/18
*
*
* OPTION NO.16
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
   0  900  901  912  902  S
   1  913  911  910  903  S
   2  920  921  932  922  S
   3  933  931  930  923  S
   4  940  941  952  942  S
   5  953  951  950  943  S
   6  960  961  972  962  S
   7  973  971  970  963  S
   8  980  981  992  982  S
   9  993  991  990  983  S
  10  9A0  9A1  9B2  9A2  S
  11  9B3  9B1  9B0  9A3  S
  12  9C0  9C1  9D2  9C2  S
  13  9D3  9D1  9D0  9C3  S
  14  9E0  9E1  9F2  9E2  S
  15  9F3  9F1  9F0  9E3  S
  16  A00  A01  A12  A02  S
  17  A13  A11  A10  A03  S
  18  A20  A21  A32  A22  S
  19  A33  A31  A30  A23  S
  20  A40  A41  A52  A42  S
  21  A53  A51  A50  A43  S
  22  A60  A61  A72  A62  S
  23  A73  A71  A70  A63  S
  24  A80  A81  A82  A83  C
  25  A90  A91  A92  A93  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 ICS623B

5.1 ICS623B Outline

The In-circuit Emulator ICE62R (ICE6200) connects the target board produced by the user via the EVA623B 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 ICE62R (ICE6200) control is done through the ICE Control Software ICS623B.

The ICS623B 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.

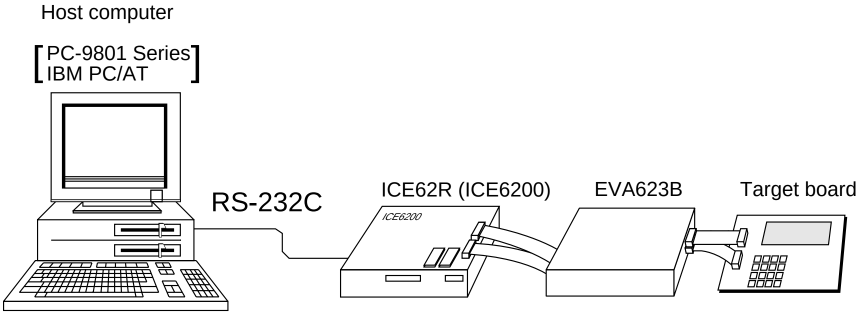


Fig. 5.1.1 Debugging system using ICE62R (ICE6200)

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

5.2 ICS623B Restrictions

Take the following precautions when using the ICS623B.

■ ROM Area

The ROM area is limited to a maximum address of 5FFH. Assigning data above the 5FFH address causes an error.

■ RAM Area

The RAM area is limited to a maximum address of 05FH. Assigning data above the 05FH address causes an error.

■ Undefined Code

The instructions below are not specified for the E0C623B and so cannot be used.

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

In addition to these instructions, the SLP instruction cannot be used when the E0C623B crystal oscillation circuit is selected by mask option.

■ OPTLD Command

In the ICS623B, OPTLD command can be used.

This command is used to load HEX files (function option data and segment option data for LCD) in the EVA623B memory with the ICE62R (ICE6200).

Load of function option data: `#OPTLD,1,C23BXXX`↵
Load of segment option data: `#OPTLD,2,C23BXXX`↵

OPTLD *READ HEXA DATA FILE*

Format

#OPTLD,1,<*file name*>↵ ...(1)
#OPTLD,2,<*file name*>↵ ...(2)

Function

(1) Load function option HEX file in the EVA623B function option data memory.
It is HEX file output by the function option generator and has intel HEX format.

(2) Load segment option HEX file in the EVA623B segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.

Examples

#OPTLD,1,C23BXXX↵ ..... C23BXXXF.HEX file is loaded in the function option data memory.

#OPTLD,2,C23BXXX↵ ..... C23BXXXS.HEX file is loaded in the segment option data memory.

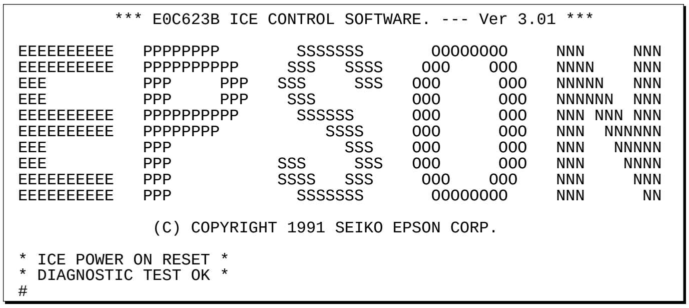
5.3 ICS623B Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:	ICS623B.BAT	(ICS623BJ.EXE)	. . . for MS-DOS
	ICS623BB.BAT	(ICS623BW.EXE)	. . . for PC-DOS
Starting command:	**ICS623B (ICS623BJ)↵**		. . . for MS-DOS
	ICS623BB (ICS623BW)↵		. . . for PC-DOS
Input file:	C23BXXXL.HEX	(Object file, low-order)	
	C23BXXXH.HEX	(Object file, high-order)	
	C23BXXXD.HEX	(Data RAM file)	
	C23BXXXC.HEX	(Control file)	
Output file:	C23BXXXL.HEX	(Object file, low-order)	
	C23BXXXH.HEX	(Object file, high-order)	
	C23BXXXD.HEX	(Data RAM file)	
	C23BXXXC.HEX	(Control file)	

■ Display example



```
          *** E0C623B ICE CONTROL SOFTWARE. --- Ver 3.01 ***

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

              (C) COPYRIGHT 1991 SEIKO EPSON CORP.

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

Start-up message

When ICS623B is started, the start-up message is displayed, and a self-test is automatically performed. ICS623B 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 ICS623B program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS623B.*

■ Error messages

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

■ 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↵ #BAR,a ↵	Sets Break at program address "a" Breakpoint is canceled
		#BD↵ #BDR↵	Break condition is set for data RAM Breakpoint is canceled
		#BR ↵ #BRR↵	Break condition is set for EVA623B CPU internal registers Breakpoint is canceled
		#BM↵ #BMR↵	Combined break conditions set for program data RAM address and registers 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 EVA623B CPU internal registers
		#SR↵	Set EVA623B CPU internal registers
		#I ↵	Reset EVA623B 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 ↵	Display history data for pointer 1 and pointer 2
		#HB ↵	Display upstream history data
		#HG ↵	Display 21 line history data
		#HP ↵	Display history pointer
		#HPS,a ↵	Set history pointer
		#HC,S/C/E ↵	Sets up the history information acquisition before (S), before/after (C) and after (E)
		#HA,a1,a2 ↵	Sets up the history information acquisition from program area a1 to a2
		#HAR,a1,a2 ↵	Sets up the prohibition of the history information acquisition from program area a1 to a2
		#HAD ↵	Indicates history acquisition program area
		#HS,a ↵	Retrieves and indicates the history information which executed a program address "a"
		#HSW,a ↵ #HSR,a ↵	Retrieves and indicates the history information which wrote or read the data area address "a"
12	File	#RF,file ↵	Move program file to memory
		#RFD,file ↵	Move data file to memory
		#VF,file ↵	Compare program file and contents of memory
		#VFD,file ↵	Compare data file and contents of memory
		#WF,file ↵	Save contents of memory to program file
		#WFD,file ↵	Save contents of memory to data file
		#CL,file ↵	Load ICE62R (ICE6200) set condition from file
		#CS,file ↵	Save ICE62R (ICE6200) set condition to file
		#OPTLD,1,file ↵	Load function option data from file
		#OPTLD,2,file ↵	Load segment option data from file
13	Coverage	#CVD ↵	Indicates coverage information
		#CVR ↵	Clears coverage information
14	ROM Access	#RP ↵	Move contents of ROM to program memory
		#VP ↵	Compare contents of ROM with contents of program memory
		#ROM ↵	Set ROM type
15	Terminate ICE	#Q ↵	Terminate ICE and return to operating system control
16	Command Display	#HELP ↵	Display ICE62R (ICE6200) instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE62R (ICE6200) self diagnostic test

↵ means press the RETURN key.

Note The system reset sequence of the E0C623B is different from other E0C62 Family models, therefore the "#G, R" command cannot be used. First reset the CPU using the "I" command, then execute the "G" command after confirming that the HALT LED on the ICE is lit.

6 MASK DATA CHECKER MDC623B

6.1 MDC623B Outline

The Mask Data Checker MDC623B is a software tool which checks the program data (C23BXXXH.HEX and C23BXXXL.HEX) and option data (C23BXXXF.DOC and C23BXXXS.DOC) created by the user and creates the data file (C623BXXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC623B has the capability to restore the generated data file (C623BXXX.PA0) to the original file format.

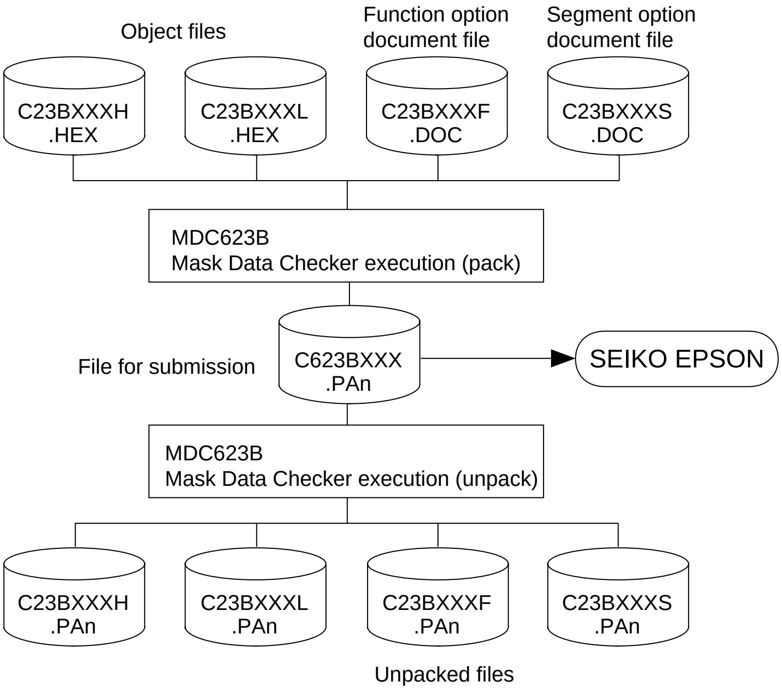


Fig. 6.1.1 MDC623B execution flow

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

6.2 MDC623B Quick Reference

■ Starting command and input/output files

Execution file: MDC623B.EXE

Starting command: **MDC623B**↵ ↵ *indicates the Return key.*

Input file:	C23BXXXL.HEX	(Object file, low-order)	When packing
	C23BXXXH.HEX	(Object file, high-order)	
	C23BXXXF.DOC	(Function option document file)	
	C23BXXXS.DOC	(Segment option document file)	
	C623BXXX.PAn	(Packed file)	When unpacking
Output file:	C623BXXX.PAn	(Packed file)	When packing
	C23BXXXL.PAn	(Object file, low-order)	When unpacking
	C23BXXXH.PAn	(Object file, high-order)	
	C23BXXXF.PAn	(Function option document file)	
	C23BXXXS.PAn	(Segment option document file)	

■ Display examples

```
            *** E0C623B 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   PPPPPPPP          SSSS    OOO        OOO  NNN  NNNNNN
EEE          PPP                 SSS   OOO        OOO  NNN   NNNNN
EEE          PPP          SSS      SSS OOO        OOO  NNN    NNNN
EEEEEEEEEE   PPP          SSSS   SSS    OOO     OOO    NNN     NNN
EEEEEEEEEE   PPP            SSSSSSS       OOOOOOOO     NNN      NN

              (C) COPYRIGHT 1990 SEIKO EPSON CORP.

                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

                    PLEASE SELECT NO.?
```

Start-up message

When MDC623B is started, the start-up message and operation menu are displayed. Here, the user is prompted to select operation options.

```
                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

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

        C23BXXXH.HEX --------+
                             |
        C23BXXXL.HEX --------+
                             |-------- C623BXXX.PAn (PACK FILE)
        C23BXXXF.DOC --------+
                             |
        C23BXXXS.DOC --------+

 PLEASE INPUT PACK FILE NAME (C623BXXX.PAn) ?  C623B0A0.PA0↵    ...(2)

        C23B0A0H.HEX --------+
                             |
        C23B0A0L.HEX --------+
                             |-------- C623B0A0.PA0
        C23B0A0F.DOC --------+
                             |
        C23B0A0S.DOC --------+
```

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 "C623BXXX.PA0" has been submitted, the pack file name should be entered as "C623BXXX.PA1".)

With this, the mask file (C623BXXX.PAn) is generated, and the MDC623B 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 (ASM623B) 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 (C623BXXX.PAn) ?  C623B0A0.PA0↵ ...(2)

                              +-------- C23B0A0H.PA0
                              |
                              +-------- C23B0A0L.PA0
        C623B0A0.PA0 --------|
                              +-------- C23B0A0F.PA0
                              |
                              +-------- C23B0A0S.PA0
```

Unpacking of data

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

(2) Enter the packed file name.

With this, the mask data file (C623BXXX.PAn) is restored to the original file format, and the MDC623B 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. E0C623B 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		
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	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					12	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2) SP←SP+3, M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, 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)
	SLP		1	1	1	1	1	1	1	1	1	0	0	1					5	Sleep (stop oscillation)
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←ry7~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, 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	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	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					5	M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, 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	Mne-monic	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←$\overline{r}$

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 B. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE62R (ICE6200)	Nothing appears on the screen, or nothing works, after activation.	Check the following and remedy if necessary: • 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 ICS623BJ.EXE PC-DOS ICS623BW.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 ICE62R (ICE6200) unit set correctly? • Is the breaker of the ICE62R (ICE6200) set to ON?
	The ICE6200 breaker tripped or the ICE62R fuse cut immediately after activation.	Check the following and remedy if necessary: • Are connectors F1 and F5 connected to the EVA623B 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 ICS623BP.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). *Example:* `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.
	<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 indicates 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.
SOG623B	An R error occurs although the address is correctly set in the segment source file.	Check the following and remedy if necessary: • Does the address symbol use capital letters? • Are the output ports set for every two terminals?

Tool	Problem	Remedy measures
ASM623B	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.
MDC623B	Activation is impossible.	Check the following and remedy if necessary: • Is the number of files set at ten or more in OS environment file CONFIG.SYS?
EVA623B	The EVA623B does not work when it is used independently.	Check the following and remedy if necessary: • 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)
	Target segment does not light.	Check the following and remedy if necessary: • Is an EPROM with an access time of 250 ns or less being used for S.HEX.

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

- GERMANY -
SALES OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)217-15045-0 Fax: +49-(0)217-15045-10

- UNITED KINGDOM -
UK BRANCH OFFICE
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: 64107320

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, 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

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
50F, 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

ENERGY
SAVING
EPSON

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 Epson WWW server
http://www.epson.co.jp/device/

First issue DECEMBER 1997, Printed SEPTEMBER 1999 in Japan Ⓜ A