MF838-03

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

E0C62M2 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 Control Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from another government agency. Please note that "E0C" is the new name for the old product "SMC". If "SMC" appears in other manuals understand that it now reads "E0C".

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

© **SEIKO EPSON CORPORATION** 1997 All rights reserved.

E0C62M2 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C62M2.

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 EVA62M2 Manual ICE6200 Hardware Manual
Development procedure	☞ E0C62 Family Technical Guide
Device (E0C62M2)	☞ E0C62M2 Technical Manual
Instructions	☞ E0C6200/6200A Core CPU Manual

CONTENTS

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 DEV62M2

The below software are included in the product of the E0C62M2 development support tool DEV62M2.

1. Development Tool Management System DMS6200 ..... Menu selection for each software / start-up software
2. Cross Assembler ASM62M2 ..... Cross assembler for program preparation
3. Function Option Generator FOG62M2 ..... Function option data preparation program
4. Segment Option Generator SOG62M2 ..... Segment option data preparation program
5. ICE Control Software ICS62M2 ..... ICE control program
6. Mask Data Checker MDC62M2 ..... Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV62M2 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 E0C62M2, 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.

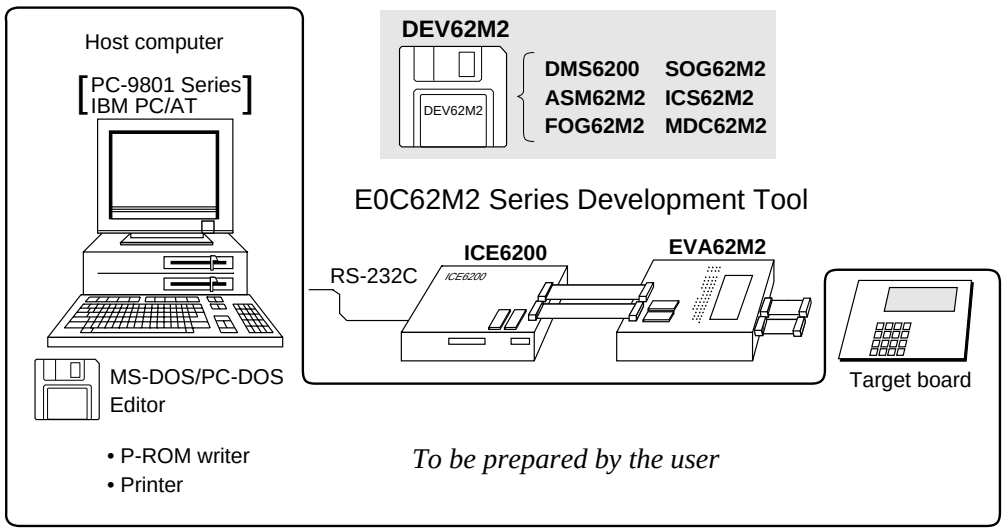


Fig. 1.2.1 System configuration

Note The DEV62M2 system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 is connected to the host computer with a RS-232C serial interface, adapter board for asynchronous communication will be required depending on the host computer used.

1.3 Development Flow

Figure 1.3.1 shows the development flow through the DEV62M2.

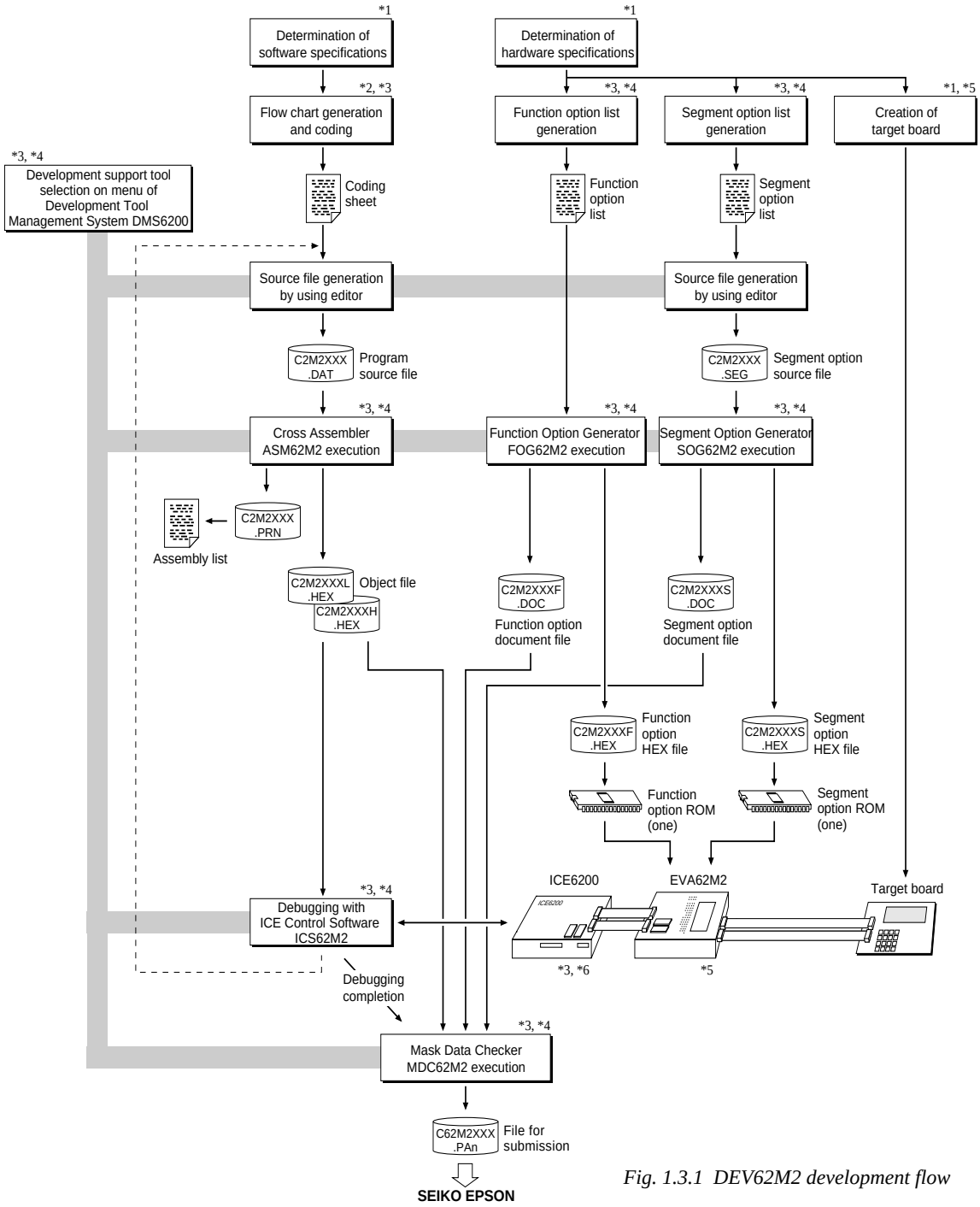


Fig. 1.3.1 DEV62M2 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C2M2XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1 E0C62M2 Technical Manual
*2 E0C6200/6200A Core CPU Manual
*3 E0C62 Family Development Tool Reference Manual
*4 E0C62M2 Development Tool Manual (this manual)
*5 EVA62M2 Manual
*6 ICE6200 Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 are recorded in the DEV62M2 floppy disk.
The content of the files contained in the DEV62M2 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM62M2.EXE	ASM62M2.EXE	Cross Assembler execution file
DMS6200.EXE	DMS6200.EXE	Development Tool Management System execution file
FOG62M2.EXE	FOG62M2.EXE	Function Option Generator execution file
ICS62M2B.BAT	ICS62M2.BAT	ICE Control Software batch file
ICS62M2W.EXE	ICS62M2J.EXE	ICE Control Software execution file
ICS62M2P.PAR	ICS62M2P.PAR	ICE Control Software parameter file
MDC62M2.EXE	MDC62M2.EXE	Mask Data Checker execution file
SOG62M2.EXE	SOG62M2.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 (DEV62M2, 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 (DEV62M2), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV62M2↵
C\>CD DEV62M2↵
C\DEV62M2\>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 MDC62M2 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
    :
```

- It is a good idea to copy the editor into the disk to be copied and the subdirectory so you can also select the editor from the DMS6200 menu.

- In "ICS62M2(B).BAT" the batch process is indicated such that the ICS62M2J(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters. When first executing the ICE Control Software after resetting the host computer, select then activate this batch file from the DMS6200 menu.
  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  COM2:  4800, n, 8, 1, p
```

RS-232C Setting (MS-DOS version)

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

Note The DMS6200 prepares a menu from files that are in the current directory. Consequently, be sure to arrange the above mentioned files in the same disk or the same directory.

2 DEVELOPMENT TOOL MANAGEMENT SYSTEM DMS6200

2.1 DMS6200 Outline

The DMS6200 (Development Tool Management System) is a software which selects the DEV62M2 software development support tool and the program such as an editor in menu form and starts it.
In this way the various software frequently executed during debugging can be effectively activated.

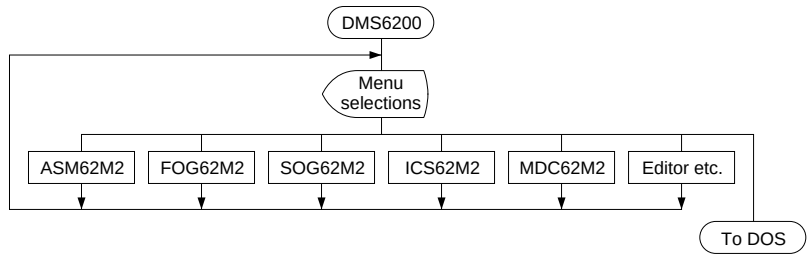


Fig. 2.1.1 DMS6200 execution flow

Refer to the "E0C62 Family Development Tool Reference Manual" for detailes of the operation.

2.2 DMS6200 Quick Reference

■ Starting command

Execution file: **DMS6200.EXE**

Starting command: **DMS6200**↵ ↵ *indicates the Return key.*

■ Display examples

```
   *** E0C6200 Development tool Management System. --- Ver 1.0 ***

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 1988 SEIKO EPSON CORP.

                          STRIKE ANY KEY.
```

Start message
When DMS6200 is started, the following 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.

```
       E0C6200 Development tool Management System. --- Ver 1.0 ***

1) ASM62M2 .EXE
2) FOG62M2 .EXE
3) ICS62M2B.BAT
4) ICS62M2W.EXE
5) MDC62M2 .EXE
6) SOG62M2 .EXE

Input Number ? [1  ]
```

Menu screen (PC-DOS Version)
A list of all executable files will appear on this menu screen.
Input the number of the development support tool you wish to start and then press the "RETURN" key. To return to DOS at this point, press the "ESC" key.

```
       E0C6200 Development tool Management System. --- Ver 1.0 ***

1) ASM62M2 .H
2) C2M20A0 .DAT
3) FOG62M2 .FDT
4) ICS62M2P.PAR
5) MDC62M2 .H
6) SOG62M2 .SDT

Input Number ? [1  ]

Edit > [ASM62M2 C2M2XXX                                          ]
```

Source file selection screen
To starting ASM62M2, select the source file on this screen. When the source file is selected by number, the edit line enclosed in [ ] will appear; enter the option parameter if necessary. Press the "RETURN" key when input is completed. When starting, press the "RETURN" key twice particularly for the support tools which do not require source files. To return to DOS at this point, press the "ESC" key.

3 CROSS ASSEMBLER ASM62M2

3.1 ASM62M2 Outline

The ASM62M2 cross assembler is an assembler program for generating the machine code used by the E0C62M2 4-bit, single-chip microcomputers. The Cross Assembler ASM62M2 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.

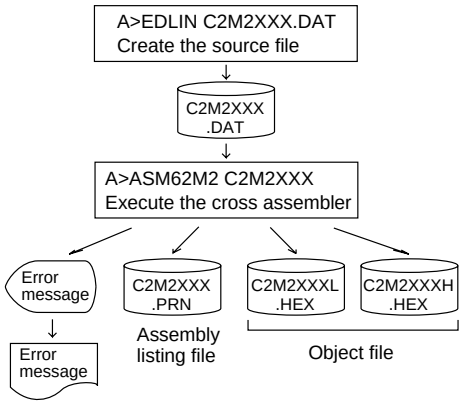


Fig. 3.1.1 ASM62M2 execution flow

☞ *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.*

3.2 E0C62M2 Restrictions

Note the following when generating a program by the E0C62M2:

■ ROM area

The capacity of the E0C62M2 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 05H), 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 E0C62M2 RAM is 184 words (000H to 0FFH, 4 bits/word).
However, note the following points when programming.

(1) The following addresses become unused area. Memory access is invalid when the unused area is specified.
080H–09FH, 0B0H–0BFH, 0C5H–0C7H, 0CBH, 0D2H–0D3H, 0D7H–0DFH, 0ECH–0EFH, 0F7H–0FFH

(2) Since RAM is set only 1 page, the page section (XP, YP) of the index register which specifies address is not effective.

■ Undefined codes

The following instructions have not been defined in the E0C62M2 instruction sets.

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

3.3 ASM62M2 Quick Reference

■ Starting command and input/output files

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

Execution file: ASM62M2.EXE

Starting command: **ASM62M2_ [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: C2M2XXX.DAT (Source file)

Output file: C2M2XXXL.HEX (Object file, low-order)
C2M2XXXH.HEX (Object file, high-order)
C2M2XXX.PRN (Assembly listing file)

■ Display example

```
          *** ASM62M2 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 1988 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C2M2XXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C2M2XXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C2M2XXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C2M2XXX .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 ASM62M2 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, ASM62M2 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)	• A statement exceeded a page boundary although its location was not specified. • 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).

4 FUNCTION OPTION GENERATOR FOG62M2

4.1 FOG62M2 Outline

With the 4-bit single-chip E0C62M2 microcomputers, the customer may select three hardware options. By modifying the mask patterns of the E0C62M2 according to the selected options, the system can be customized to meet the specifications of the target system.
The Function Option Generator FOG62M2 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 FOG62M2, the E0C62M2 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA62M2) 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.*

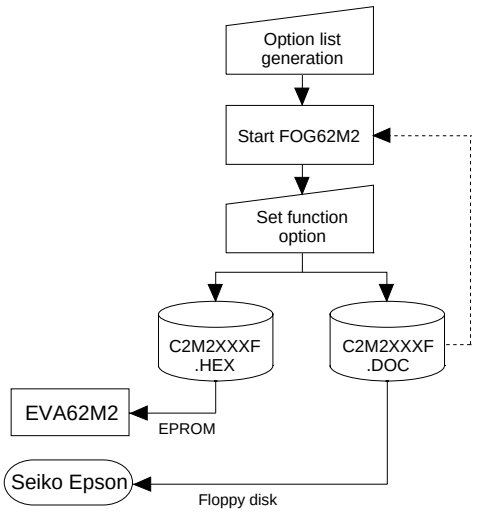


Fig. 4.1.1 FOG62M2 execution flow

4.2 E0C62M2 Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "4.3 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. 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
• K10	☐ 1. With Resistor	☐ 2. Gate Direct
• K11	☐ 1. With Resistor	☐ 2. Gate Direct
• K12	☐ 1. With Resistor	☐ 2. Gate Direct
• K13	☐ 1. With Resistor	☐ 2. Gate Direct

2. OUTPUT PORT OUTPUT SPECIFICATION

• R00	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• R01	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• R02	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• R03	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• BZ	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• /BZ	☐ 1. Complementary	☐ 2. Pch-OpenDrain

3. I/O PORT OUTPUT SPECIFICATION

• P00	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• P01	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• P02	☐ 1. Complementary	☐ 2. Pch-OpenDrain
• P03	☐ 1. Complementary	☐ 2. Pch-OpenDrain

4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C62M2 are shown below, and their specifications are also described.

1 Input port pull down resistor

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

--- 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↵
                           :
                    (Selection for K02 to K12)
                           :
PLEASE SELECT NO.(1) ? 1↵

        K13
                          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
        K10               1. WITH RESISTOR  SELECTED
        K11               1. WITH RESISTOR  SELECTED
        K12               1. WITH RESISTOR  SELECTED
        K13               1. WITH RESISTOR  SELECTED
```

Select whether input ports (K00–K03 and K10–K13) 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 high level (V_{DD}) to low (V_{SS}) with pull down resistors, a delay 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 4.3.1.

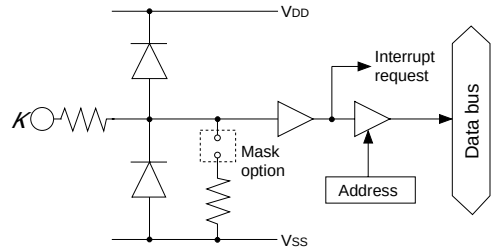


Fig. 4.3.1 Configuration of pull down resistor circuit

2 Output port output specification

```
*** OPTION  NO.2 ***

--- OUTPUT PORT OUTPUT SPECIFICATION ---
        R00
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵
                           :
                    (Selection for R01 to R03)
                           :
PLEASE SELECT NO.(1) ? 1↵

        BZ
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        /BZ
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R00               1. COMPLEMENTARY  SELECTED
        R01               1. COMPLEMENTARY  SELECTED
        R02               1. COMPLEMENTARY  SELECTED
        R03               1. COMPLEMENTARY  SELECTED
        BZ                1. COMPLEMENTARY  SELECTED
        /BZ               1. COMPLEMENTARY  SELECTED
```

Select the output specification for the output ports (R00–R03, BZ and $\overline{BZ}$).
Either complementary output or Pch open drain output may be selected.
When output port is to be used on key matrix configuration, select Pch open drain output.
For unused output ports, select complementary output.
The output circuit configuration is shown in Figure 4.3.2.

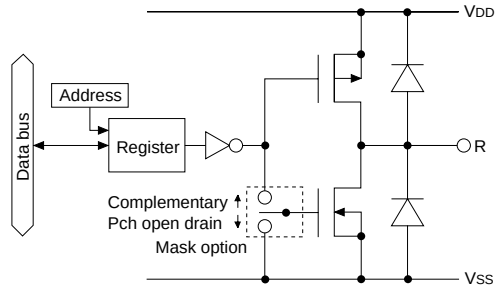


Fig. 4.3.2 Configuration of output circuit

3 I/O port output specification

```
*** OPTION  NO.3 ***

--- I/O PORT OUTPUT SPECIFICATION ---
        P00
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↲

        P01
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↲

        P02
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↲

        P03
                          1. COMPLEMENTARY
                          2. PCH-OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↲

        P00               1. COMPLEMENTARY  SELECTED
        P01               1. COMPLEMENTARY  SELECTED
        P02               1. COMPLEMENTARY  SELECTED
        P03               1. COMPLEMENTARY  SELECTED
```

Select the output specification to be used during I/O ports (P00–P03) output mode selection.
Either complementary output or Pch open drain output may be selected.
The circuit configuration of the output driver is the same as that of output ports (Rxx shown in Figure 4.3.2).
Select complementary output for unused ports.

The I/O ports can control the input/output direction according to the IOC00–IOC03 register (C8 address, D0 bit–D3 bit); at "1" and "0" settings, it is set to output port and input port, respectively.

When the serial interface function is selected, the output specification of the terminals SOUT, SCLK (during the master mode) and SRDY (during the slave mode) that is used as output in the input/ output port of the serial interface is respectively selected by the mask options of P01, P02 and P03. Selects complementary output for the SIN (P00) output specification.

The I/O port circuit configuration is shown in Figure 4.3.3.

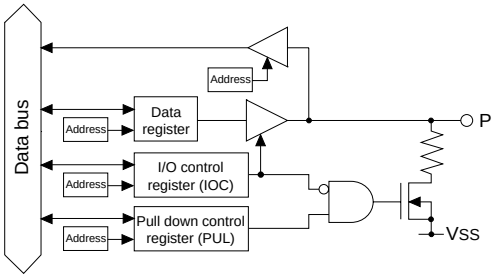


Fig. 4.3.3 Circuit configuration of I/O Port

4.4 FOG62M2 Quick Reference

■ Starting command and input/output files

Execution file: FOG62M2.EXE

Starting command: **FOG62M2**↵ ↵ *indicates the Return key.*

Input file: C2M2XXXF.DOC (Function option document file, when modifying)

Output file: C2M2XXXF.DOC (Function option document file)
C2M2XXXF.HEX (Function option HEX file)

■ Display example

```
     ***  E0C62M2 FUNCTION OPTION GENERATOR. --- Ver 3.14  ***

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 1994  SEIKO EPSON CORP.

        THIS SOFTWARE MAKES NEXT FILES.

           C2M2XXXF.HEX  ...  FUNCTION OPTION HEX FILE.
           C2M2XXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                      STRIKE ANY KEY.
```

Start-up message
When FOG62M2 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.

```
*** E0C62M2 USER'S OPTION SETTING. --- Ver 3.14 ***

CURRENT DATE IS  1994/12/13
PLEASE INPUT NEW DATE : 1994/12/31↵
```

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 FOG62M2.

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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME ? C2M20A0↵                              ..(1)
PLEASE INPUT USER'S NAME ? SEIKO EPSON CORP.↵                  ..(2)
PLEASE INPUT ANY COMMENT
 ( ONE LINE IS 50 CHRS ) ? TOKYO DESIGN CENTER↵                ..(3)
                         ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↵
                         ? ↵
```

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 ? C2M20A0↵
EXISTS  OVERWRITE (Y/N)? N↵
PLEASE INPUT FILE NAME ? C2M20B0↵
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 overwrition 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) ***

C2M20A0          C2M20B0          C2M20C0                  ..(1)

PLEASE INPUT FILE NAME ? C2M20A0↵                          ..(2)
PLEASE INPUT USER'S NAME ? ↵                               ..(3)
PLEASE INPUT ANY COMMENT
 ( ONE LINE IS 50 CHRS ) ? ↵                               ..(4)
PLEASE INPUT EDIT NO.? 1↵                                  ..(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 ? C2M20N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME ?
```

In addition, if specified file format is different (such as document file for the other model), the following message is displayed and FOG62M2 is terminated.

```
BAD FUNCTION OPTION DOCUMENT FILE.
```

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  NO.1 ***

--- INPUT PORT PULL DOWN RESISTOR ---
        K00
                        1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT  NO.(1) ? 2↵

        K01
                        1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT  NO.(1) ? B↵

*** OPTION  NO.1 ***

--- INPUT PORT PULL DOWN RESISTOR ---
        K00
                        1. WITH RESISTOR
                        2. GATE DIRECT

PLEASE SELECT  NO.(2) ?
```

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.

```
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(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
```

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 EVA62M2, 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 EVA62M2 options.

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

4.5 Sample File

■ Example of function option document file

```
* E0C62M2 FUNCTION OPTION DOCUMENT  V 3.14
*
* FILE NAME    C2M20A0F.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   1994/12/31
*
* COMMENT      TOKYO DESIGN CENTER
*              421-8 HINO HINO-SHI TOKYO 191 JAPAN
*
*
* OPTION NO.1
* < INPUT PORT PULL DOWN RESISTOR >
*     K00                    WITH RESISTOR  -----------------  SELECTED
*     K01                    WITH RESISTOR  -----------------  SELECTED
*     K02                    WITH RESISTOR  -----------------  SELECTED
*     K03                    WITH RESISTOR  -----------------  SELECTED
*     K10                    WITH RESISTOR  -----------------  SELECTED
*     K11                    WITH RESISTOR  -----------------  SELECTED
*     K12                    WITH RESISTOR  -----------------  SELECTED
*     K13                    WITH RESISTOR  -----------------  SELECTED
 OPT0101 01
 OPT0102 01
 OPT0103 01
 OPT0104 01
 OPT0105 01
 OPT0106 01
 OPT0107 01
 OPT0108 01
*
* OPTION NO.2
* < OUTPUT PORT OUTPUT SPECIFICATION >
*     R00                    COMPLEMENTARY  -----------------  SELECTED
*     R01                    COMPLEMENTARY  -----------------  SELECTED
*     R02                    COMPLEMENTARY  -----------------  SELECTED
*     R03                    COMPLEMENTARY  -----------------  SELECTED
*     BZ                     COMPLEMENTARY  -----------------  SELECTED
*     /BZ                    COMPLEMENTARY  -----------------  SELECTED
 OPT0201 01
 OPT0202 01
 OPT0203 01
 OPT0204 01
 OPT0205 01
 OPT0206 01
*
* OPTION NO.3
* < I/O PORT OUTPUT SPECIFICATION >
*     P00                    COMPLEMENTARY  -----------------  SELECTED
*     P01                    COMPLEMENTARY  -----------------  SELECTED
*     P02                    COMPLEMENTARY  -----------------  SELECTED
*     P03                    COMPLEMENTARY  -----------------  SELECTED
 OPT0301 01
 OPT0302 01
 OPT0303 01
 OPT0304 01
*
*
```

```
*
* SEIKO EPSON'S AREA
*
*
*
* OPTION NO.4
 OPT0401 01
*
* OPTION NO.5
 OPT0501 01
*
* OPTION NO.6
 OPT0601 01
*
* OPTION NO.7
 OPT0701 01
*
* OPTION NO.8
 OPT0801 01
*
* OPTION NO.9
 OPT0901 01
*
* OPTION NO.10
 OPT1001 01
\\END
```

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

5 SEGMENT OPTION GENERATOR SOG62M2

5.1 SOG62M2 Outline

With the 4-bit single-chip E0C62M2 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C62M2 according to the selected options, the system can be customized to meet the specifications of the target system.
The Segment Option Generator SOG62M2 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG62M2, the E0C62M2 mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA62M2) 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.*

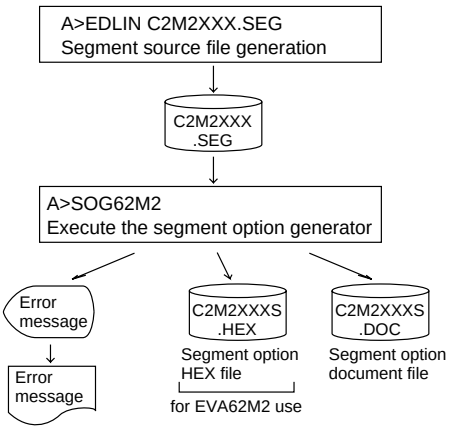


Fig. 5.1.1 SOG62M2 execution flow

5.2 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 ☐ P
SEG2													SEG output
SEG3													DC output ☐ C ☐ P
SEG4													SEG output
SEG5													DC output ☐ C ☐ P
SEG6													SEG output
SEG7													DC output ☐ C ☐ P
SEG8													SEG output
SEG9													DC output ☐ C ☐ P
SEG10													SEG output
SEG11													DC output ☐ C ☐ P
SEG12													SEG output
SEG13													DC output ☐ C ☐ P
SEG14													SEG output
SEG15													DC output ☐ C ☐ 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.*

5.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG15, 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 display memory area (A0H–AFH) can be allocated to the optional segment. With this, up to 64 segments (48 segments when 1/3 duty is selected, respectively) of liquid crystal panel could be driven.
The display memory may be allocated only one segment and multiple setting is not possible.
The allocated segment displays when the bit for this display memory is set to "1", and goes out when bit is set to "0".
Segment allocation is set to H for high address (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  A00  A01  A02  A03  S
1  A10  A11  A12  A13  S
```

• When 1/3 duty is selected

```
0  A00  A01  A02  ---  S
1  A10  A11  A12  ---  S
```

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 16 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 display memory is set to "1", the segment output port goes high (V_{DD}), and goes low (V_{SS}) when set to "0". Segment allocation is the same as when segment output is selected but for the while the display 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 SEG12 and SEG13, and Pch open drain output is set to SEG14 and SEG15.

```
12  AC0  ---  ---  ---  C
13  AD0  ---  ---  ---  C
14  AE0  ---  ---  ---  P
15  AF0  ---  ---  ---  P
```

Note *Only complementary output is enabled as the DC output of the SEG ports of EVA62M2. Therefore, complementary output is enabled even if Pch open drain output is selected. Respond to it by adding external circuits as required.*

5.4 SOG62M2 Quick Reference

■ Starting command and input/output files

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

Execution file: SOG62M2.EXE

Starting command: **SOG62M2_ [-H]↵**

Option: -H: Specifies the segment option document file for input file of SOG62M2.

Input file: C2M2XXX.SEG (Segment option source file)
C2M2XXXS.DOC(Segment option document file, when -H option use)

Output file: C2M2XXXS.DOC(Segment option document file)
C2M2XXXS.HEX (Segment option HEX file)

■ Display example

```
     ***  E0C62M2 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 1994  SEIKO EPSON CORP.

         SEGMENT OPTION SOURCE FILE NAME IS " C2M2XXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C2M2XXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C2M2XXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

Start-up message
When SOG62M2 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.

```
*** E0C62M2 USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  94/12/13
PLEASE INPUT NEW DATE : 94/12/31↵
```

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.

```
*** E0C62M2 SOURCE FILE(S) ***

C2M20A0         C2M20B0         C2M20C0                          ..(1)

PLEASE INPUT SEGMENT SOURCE FILE NAME ? C2M20A0↵                 ..(2)
PLEASE INPUT USER'S NAME ? SEIKO EPSON CORP.↵                    ..(3)
PLEASE INPUT ANY COMMENT
 ( ONE LINE IS 50 CHR ) ? TOKYO DESIGN CENTER↵                   ..(4)
                        ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↵
                        ? ↵
```

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.

```
*** E0C62M2 SOURCE FILE(S) ***

SEGMENT OPTION FILE IS NOT FOUND.                                ..(5)
```

In step (1), if no modifiable source exists, an error message (5) will be displayed and the program will be terminated.

```
*** E0C62M2 SOURCE FILE(S) ***

C2M20A0         C2M20B0         C2M20C0

PLEASE INPUT SEGMENT SOURCE FILE NAME ? C2M20N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.                         ..(6)

*** E0C62M2 SOURCE FILE(S) ***

C2M20A0         C2M20B0         C2M20C0

PLEASE INPUT SEGMENT SOURCE FILE NAME ?
```

In step (2), if the specified file name is not found in the current drive, an error message (6) 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.? 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 EVA62M2, 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 EVA62M2 options.

When a series of operations are complete, the SOG62M2 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG62M2 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. Though DC output has been selected for output specification, data are described in COM1–COM3.

5.5 Sample Files

■ Example of segment option source file

```
; C2M20A0.SEG, VER.3.21
; EVA62M2 LCD SEGMENT DECODE TABLE
;
; SEG COM0 COM1 COM2 COM3 SPEC
;
   0  A00  A01  A02  A03  S
   1  A13  A12  A11  A10  S
   2  A20  A21  A22  A23  S
   3  A33  A32  A31  A30  S
   4  A40  A41  A42  A43  S
   5  A53  A52  A51  A50  S
   6  A60  A61  A62  A63  S
   7  A73  A72  A71  A70  S
   8  A80  A81  A82  A83  S
   9  A93  A92  A91  A90  S
  10  ---  ---  ---  ---  S
  11  ---  ---  ---  ---  S
  12  AC0  ---  ---  ---  P
  13  AD3  ---  ---  ---  P
  14  AE0  ---  ---  ---  C
  15  AF3  ---  ---  ---  C
```

■ Example of segment option document file

```
* E0C62M2 SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C2M20A0S.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   94/12/13
* COMMENT      TOKYO DESIGN CENTER
*              421-8 HINO HINO-SHI TOKYO 191 JAPAN
*
*
* OPTION NO.4
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
   0  A00  A01  A02  A03  S
   1  A13  A12  A11  A10  S
   2  A20  A21  A22  A23  S
   3  A33  A32  A31  A30  S
   4  A40  A41  A42  A43  S
   5  A53  A52  A51  A50  S
   6  A60  A61  A62  A63  S
   7  A73  A72  A71  A70  S
   8  A80  A81  A82  A83  S
   9  A93  A92  A91  A90  S
  10  AA0  AA1  AA2  AA3  S
  11  AB0  AB1  AB2  AB3  S
  12  AC0  AC1  AC2  AC3  P
  13  AD3  AD0  AD1  AD2  P
  14  AE0  AE1  AE2  AE3  C
  15  AF3  AF0  AF1  AF2  C
\\END
```

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

6 ICE CONTROL SOFTWARE ICS62M2

6.1 ICS62M2 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA62M2 and performs real time target system evaluation and debugging by passing through the RS-232C from the host computer and controlling it. The operation on the host computer side and ICE6200 control is done through the ICE Control Software ICS62M2.
The ICS62M2 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.

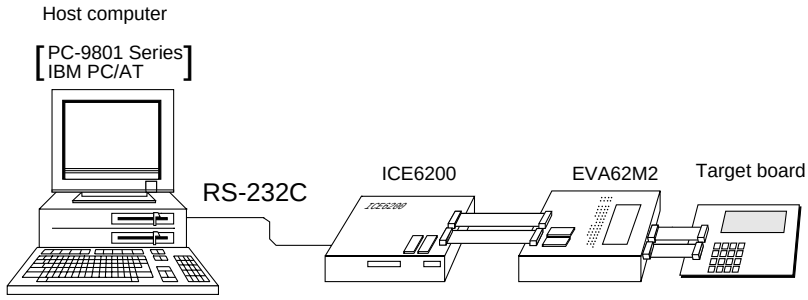


Fig. 6.1.1 Debugging system using ICE6200

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

6.2 ICS62M2 Restrictions

Take the following precautions when using the ICS62M2.

■ ROM Area

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

■ RAM Area

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

Unused area: 080H–09FH, 0B0H–0BFH, 0C5H–0C7H, 0CBH
0D2H–0D3H, 0D7H–0DFH, 0ECH–0EEH, 0F7H–0FFH

(Refer to the "E0C62M2 Technical Manual" for details.)

■ OPTLD Command

In the ICS62M2, OPTLD command can be used.
This command is used to load HEX files (function option and segment option data for LCD) in the EVA62M2 memory with the ICE6200.

Load of function option data: `#OPTLD,1,C2M2XXX↵`
Load of segment option data: `#OPTLD,2,C2M2XXX↵`

■ Undefined Code

The following instructions are not specified for the E0C62M2 and so cannot be used.

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

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 EVA62M2 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 EVA62M2 segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.

Examples

#OPTLD,1,C2M2XXX↲ ..... C2M2XXXF.HEX file is loaded in the function option data memory.

#OPTLD,2,C2M2XXX↲ ..... C2M2XXXS.HEX file is loaded in the segment option data memory.

6.3 ICS62M2 Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:
ICS62M2.BAT (ICS62M2J.EXE) . . . for MS-DOS
ICS62M2B.BAT (ICS62M2W.EXE) . . . for PC-DOS

Starting command:
ICS62M2 (ICS62M2J)↵ . . . for MS-DOS
ICS62M2B (ICS62M2W)↵ . . . for PC-DOS

Input file:
C2M2XXXL.HEX (Object file, low-order)
C2M2XXXH.HEX (Object file, high-order)
C2M2XXXD.HEX (Data RAM file)
C2M2XXXC.HEX (Control file)
C2M2XXXF.HEX (Function option HEX file)
C2M2XXXS.HEX (Segment option HEX file)

Output file:
C2M2XXXL.HEX (Object file, low-order)
C2M2XXXH.HEX (Object file, high-order)
C2M2XXXD.HEX (Data RAM file)
C2M2XXXC.HEX (Control file)

■ Display example

```
          *** E0C62M2 IN-CIRCUIT EMULATOR. --- Ver 3.01 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       00000000      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    000     000    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   000       000   NNNNN    NNN
EEE          PPP     PPP    SSS          000       000   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      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           SSSS   SSS     000     000    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       00000000      NNN      NN

             (C) COPYRIGHT 1989 SEIKO EPSON CORP.

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

Start-up message
When ICS62M2 is started, the start-up message is displayed, and a self-test is automatically performed. ICS62M2 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 ICS62M2 program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS62M2.*

■ Error messages

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

■ ICE6200 commands

Item No.	Function	Command Format	Outline of Operation
1	Assemble	#A,a ↵	Assemble command mnemonic code and store at address "a"
2	Disassemble	#L,a1,a2 ↵	Contents of addresses a1 to a2 are disassembled and displayed
3	Dump	#DP,a1,a2 ↵	Contents of program area a1 to a2 are displayed
		#DD,a1,a2 ↵	Content of data area a1 to a2 are displayed
4	Fill	#FP,a1,a2,d ↵	Data d is set in addresses a1 to a2 (program area)
		#FD,a1,a2,d ↵	Data d is set in addresses a1 to a2 (data area)
5	Set Run Mode	#G,a ↵	Program is executed from the "a" address
		#TIM ↵	Execution time and step counter selection
		#OTF ↵	On-the-fly display selection
6	Trace	#T,a,n ↵	Executes program while displaying results of step instruction from "a" address
		#U,a,n ↵	Displays only the final step of #T,a,n
7	Break	#BA,a ↵	Sets Break at program address "a"
		#BAR,a ↵	Breakpoint is canceled
		#BD ↵	Break condition is set for data RAM
		#BDR ↵	Breakpoint is canceled
		#BR ↵	Break condition is set for EVA62M2 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 EVA62M2 CPU internal registers
		#SR ↵	Set EVA62M2 CPU internal registers
		#I ↵	Reset EVA62M2 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 ICE6200 set condition from file
		#CS,file ↵	Save ICE6200 set condition to file
		#OPTLD,1,file ↵ #OPTLD,2,file ↵	Load function option data from 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 ICE6200 instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE6200 self diagnostic test

↵ means press the RETURN key.

7 MASK DATA CHECKER MDC62M2

7.1 MDC62M2 Outline

The Mask Data Checker MDC62M2 is a software tool which checks the program data (C2M2XXXH.HEX and C2M2XXXL.HEX) and option data (C2M2XXXF.DOC and C2M2XXXS.DOC) created by the user and creates the data file (C62M2XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC62M2 has the capability to restore the generated data file (C62M2XXX.PA0) to the original file format.

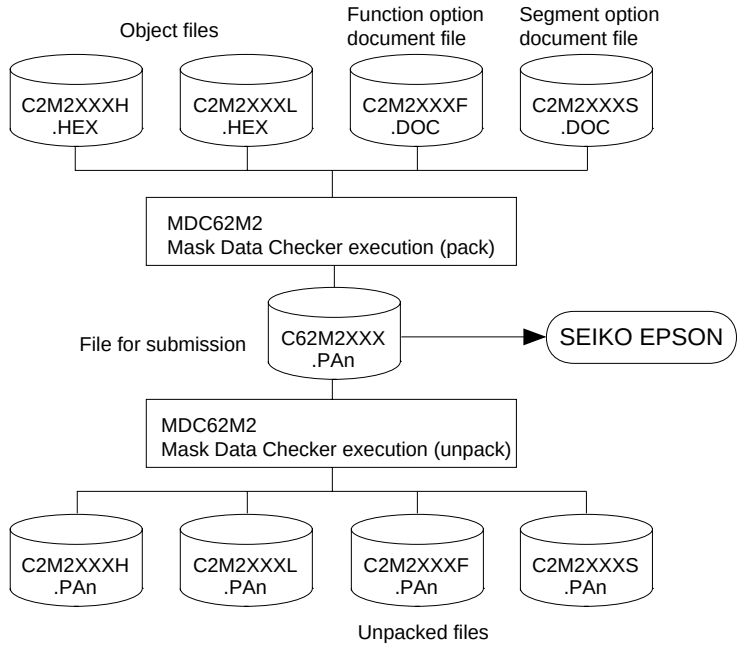


Fig. 7.1.1 MDC62M2 execution flow

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

7.2 MDC62M2 Quick Reference

■ Starting command and input/output files

Execution file: MDC62M2.EXE

Starting command: **MDC62M2**↵ ↵ *indicates the Return key.*

Input file:

C2M2XXXL.HEX (Object file, low-order)	When packing
C2M2XXXH.HEX (Object file, high-order)	When packing
C2M2XXXF.DOC (Function option document file)	When packing
C2M2XXXS.DOC (Segment option document file)	When packing
C62M2XXX.PAn (Packed file)	When unpacking

Output file:

C62M2XXX.PAn (Packed file)	When packing
C2M2XXXL.PAn (Object file, low-order)	When unpacking
C2M2XXXH.PAn (Object file, high-order)	When unpacking
C2M2XXXF.PAn (Function option document file)	When unpacking
C2M2XXXS.PAn (Segment option document file)	When unpacking

■ Display examples

```
          *** E0C62M2 PACK / UNPACK PROGRAM Ver 2.001 ***

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 1993 SEIKO EPSON CORP.

                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

                   PLEASE SELECT NO.?
```

Start-up message

When MDC62M2 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)

            C2M2XXXH.HEX   ------+
                                 |
            C2M2XXXL.HEX   ------+
                                 |------   C62M2XXX.PAn (PACK FILE)
            C2M2XXXF.DOC   ------+
                                 |
            C2M2XXXS.DOC   ------+

 PLEASE INPUT PACK FILE NAME (C62M2XXX.PAn) ?  C62M20A0.PA0↵  ...(2)

            C2M20A0H.HEX   ------+
                                 |
            C2M20A0L.HEX   ------+
                                 |------   C62M20A0.PA0
            C2M20A0F.DOC   ------+
                                 |
            C2M20A0S.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 "C62M2XXX.PA0" has been submitted, the pack file name should be entered as "C62M2XXX.PA1".)

With this, the mask file (C62M2XXX.PAn) is generated, and the MDC62M2 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 (ASM62M2) as program data. If the program data generated with the -N option of the Cross Assembler is packed, following message is displayed.*

```
HEX DATA ERROR : DATA (NO FFh)
```

```
                   --- OPERATION MENU ---

                          1. PACK
                          2. UNPACK

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

 PLEASE INPUT PACKED FILE NAME (C62M2XXX.PAn) ?  C62M20A0.PA0↵...(2)

                                 +------   C2M20A0H.PA0
                                 |
                                 +------   C2M20A0L.PA0
            C62M20A0.PA0   ------|
                                 +------   C2M20A0F.PA0
                                 |
                                 +------   C2M20A0S.PA0
```

Unpacking of data

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

With this, the mask data file (C62M2XXX.PAn) is restored to the original file format, and the MDC62M2 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.
10. HEX DATA ERROR : DATA (NO FFh)	There is an undefined field in the HEX data.

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.
4. VERSION NUMBER ERROR : X.DOC	FOG62M2, SOG62M2 different from the version No. has been used.

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. E0C62M2 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	*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)
Index operation instructions	INC	X	1	1	1	0	1	1	1	0	0	0	0	0					5	X ← X+1
		Y	1	1	1	0	1	1	1	1	0	0	0	0					5	Y ← Y+1
	LD	X, x	1	0	1	1	x7	x6	x5	x4	x3	x2	x1	x0					5	XH ← x7~x4, XL ← x3~x0
		Y, y	1	0	0	0	y7	y6	y5	y4	y3	y2	y1	y0					5	YH ← y7~y4, YL ← y3~y0
		XH, r	1	1	1	0	1	0	0	0	0	1	r1	r0					5	XH ← r
		XL, r	1	1	1	0	1	0	0	0	1	0	r1	r0					5	XL ← r
		YH, r	1	1	1	0	1	0	0	1	0	1	r1	r0					5	YH ← r
		YL, r	1	1	1	0	1	0	0	1	1	0	r1	r0					5	YL ← r
		r, XH	1	1	1	0	1	0	1	0	0	1	r1	r0					5	r ← XH
		r, XL	1	1	1	0	1	0	1	0	1	0	r1	r0					5	r ← XL
		r, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r ← YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r ← YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0			↕	↕	7	XH ← XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0			↕	↕	7	XL ← XL+i3~i0+C
		YH, i	1	0	1	0	0	0	1	0	i3	i2	i1	i0			↕	↕	7	YH ← YH+i3~i0+C
		YL, i	1	0	1	0	0	0	1	1	i3	i2	i1	i0			↕	↕	7	YL ← YL+i3~i0+C

Classification	Mnemonic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	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	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
r1	r0	q1	q0	specified
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

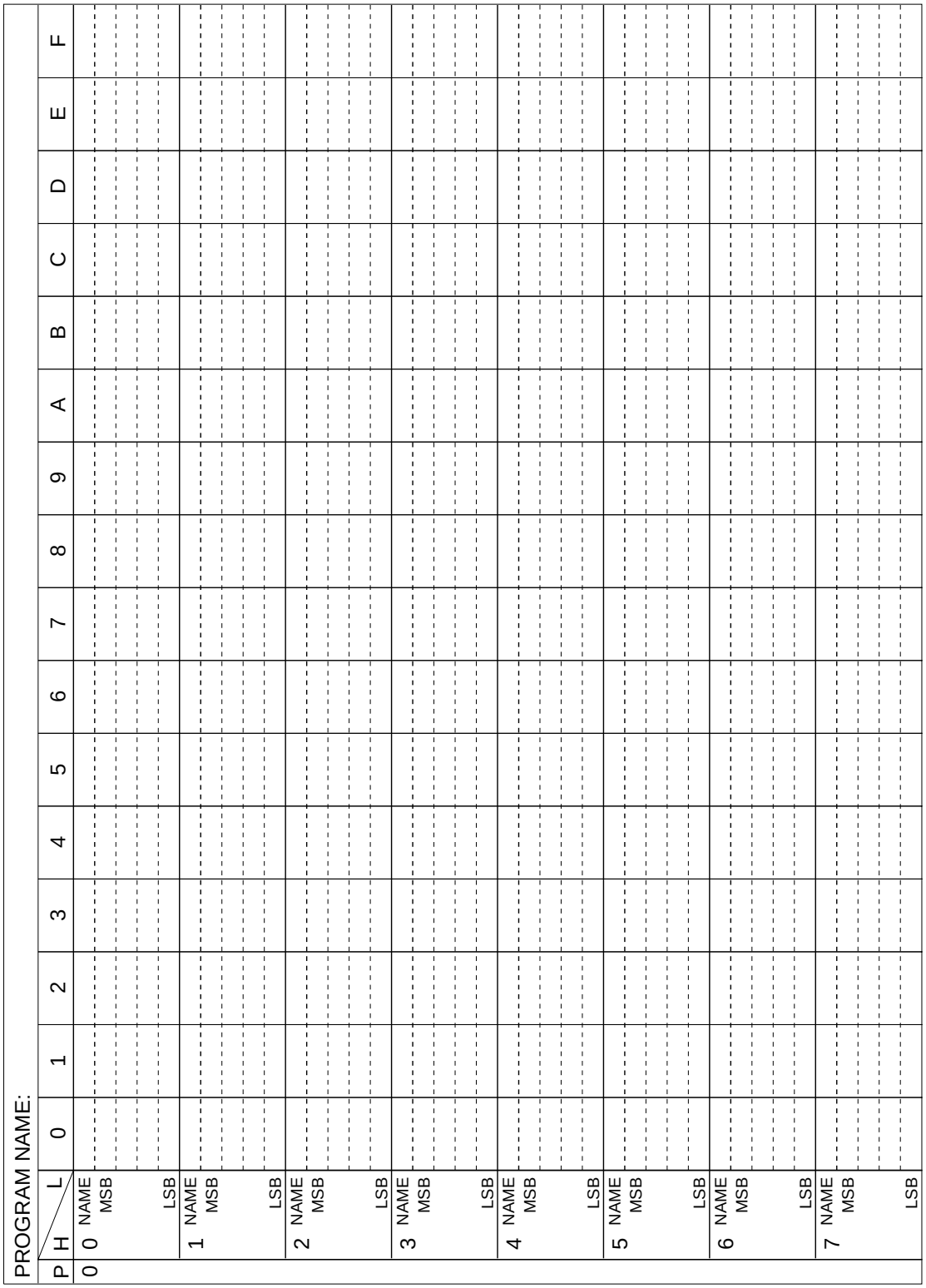
p Five-bit immediate data or label 00H–1FH
s Eight-bit immediate data or label 00H–0FFH
l Eight-bit immediate data 00H–0FFH
i Four-bit immediate data 00H–0FH

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. E0C62M2 RAM MAP

RAM map - 1 (000H–07FH)



Display memory (A0H–AFH), I/O memory (C0H–FFH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	A	NAME																
		MSB																
		LSB																
	C	NAME	ZSIK0	ZK0	ZSIK1	ZK1	ZR0	–	–	–	ZIOC0	ZPUL0	ZP0	–	ZSIF1	ZSIF2	ZSDL	ZSDH
		MSB	SIK03	K03	SIK13	K13	R03	–	–	–	IOC03	PUL03	P03	–	0	SDP	SD3	SD7
			SIK02	K02	SIK12	K12	R02	–	–	–	IOC02	PUL02	P02	–	0	SCPS	SD2	SD6
			SIK01	K01	SIK11	K11	R01	–	–	–	IOC01	PUL01	P01	–	SCTRG	SCS1	SD1	SD5
		LSB	SIK00	K00	SIK10	K10	R00	–	–	–	IOC00	PUL00	P00	–	ESIF	SCS0	SD0	SD4
	D	NAME	ZBZ1	ZBZ2	–	–	ZRST	ZTML	ZTMH	–	–	–	–	–	–	–	–	–
		MSB	0	0	–	–	0	TM3	TM7	–	–	–	–	–	–	–	–	–
			0	BSHOT	–	–	0	TM2	TM6	–	–	–	–	–	–	–	–	–
			BZFQ	BZFNC	–	–	WDRST	TM1	TM5	–	–	–	–	–	–	–	–	–
		LSB	REG	BZON	–	–	TMRST	TM0	TM4	–	–	–	–	–	–	–	–	–
	E	NAME	ZVSS2	ZLCD	ZAD	ZSVD	ZFNC	ZRNG	ZADP	ZDSC0	ZDSC1	ZDSC2	ZDSC3	ZSTS	–	–	–	–
		MSB	0	LOFF	BUFF	0	0	0	0	DSC03	DSC13	DSC23	0	0	–	–	–	–
			0	0	IIN	0	FNC2	RNG2	0	DSC02	DSC12	DSC22	DSC32	IDR	–	–	–	–
			0	LDTY	ADSPD	SVDDT	FNC1	RNG1	0	DSC01	DSC11	DSC21	DSC31	STS1	–	–	–	–
		LSB	VSS2	LPWR	ADON	SVDON	FNC0	RNG0	ADP	DSC00	DSC10	DSC20	DSC30	STS0	–	–	–	–
	F	NAME	ZEI	ZEIT	ZIK1	ZIK0	ZISIF	ZIT	ZIAD	–	–	–	–	–	–	–	–	–
		MSB	EIK1	EIT1	0	0	0	IT1	0	–	–	–	–	–	–	–	–	–
			EIK0	EIT2	0	0	0	IT2	0	–	–	–	–	–	–	–	–	–
			EISIF	EIT16	0	0	0	IT16	0	–	–	–	–	–	–	–	–	–
		LSB	EIAD	EIT32	IK1	IK0	ISIF	IT32	IAD	–	–	–	–	–	–	–	–	–

APPENDIX C. E0C62M2 I/O MEMORY MAP

I/O memory map (C0H–CAH)

Address	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
C0H	SIK03	SIK02	SIK01	SIK00	SIK03	0	Enable	Disable	Interrupt selection register (K03)
					SIK02	0	Enable	Disable	Interrupt selection register (K02)
	R/W				SIK01	0	Enable	Disable	Interrupt selection register (K01)
					SIK00	0	Enable	Disable	Interrupt selection register (K00)
C1H	K03	K02	K01	K00	K03	– *2	High	Low	Input port (K00–K03)
					K02	– *2	High	Low	
	R				K01	– *2	High	Low	
					K00	– *2	High	Low	
C2H	SIK13	SIK12	SIK11	SIK10	SIK13	0	Enable	Disable	Interrupt selection register (K13)
					SIK12	0	Enable	Disable	Interrupt selection register (K12)
	R/W				SIK11	0	Enable	Disable	Interrupt selection register (K11)
					SIK10	0	Enable	Disable	Interrupt selection register (K10)
C3H	K13	K12	K11	K10	K13	– *2	High	Low	Input port (K10–K13)
					K12	– *2	High	Low	
	R				K11	– *2	High	Low	
					K10	– *2	High	Low	
C4H	R03	R02	R01	R00	R03	0	High	Low	Output port (R00–R03)
					R02	0	High	Low	
	R/W				R01	0	High	Low	
					R00	0	High	Low	
C8H	IOC03	IOC02	IOC01	IOC00	IOC03	0	Output	Input	I/O control register (P00–P03) (ESIF = 0)
					IOC02	0	Output	Input	
	R/W				IOC01	0	Output	Input	
					IOC00	0	Output	Input	
	When the serial I/F is used (ESIF = 1):				IOC03	0	Output	Input	Master mode: P03 I/O control register
	P00 = SIN (in), P01 = SOUT (out),				IOC03	0	1	0	Slave mode: General-purpose register
	P02 = SCLK (master: out, slave: in),				IOC02	0	1	0	General-purpose register
	P03 = SRDY (slave: out),				IOC01	0	1	0	
	P03 = I/O port (master: in/out)				IOC00	0	1	0	
C9H	PUL03	PUL02	PUL01	PUL00	PUL03	1	On	Off	Pull down control register (P00–P03) (ESIF = 0)
					PUL02	1	On	Off	
	R/W				PUL01	1	On	Off	
					PUL00	1	On	Off	
	When the serial I/F is used (ESIF = 1):				PUL03	1	On	Off	Master mode: P03 pull down control register
	P00 = SIN (in), P01 = SOUT (out),				PUL03	1	1	0	Slave mode: General-purpose register
	P02 = SCLK (master: out, slave: in),				PUL02	1	1	0	Master mode: General-purpose register
	P03 = SRDY (slave: out),				PUL02	1	On	Off	Slave mode: SCKL pull down control register
	P03 = I/O port (master: in/out)				PUL01	1	1	0	General-purpose register
					PUL00	1	On	Off	SIN pull down control register
CAH	P03	P02	P01	P00	P03	– *2	High	Low	I/O port (P00–P03) (ESIF = 0)
					P02	– *2	High	Low	
	R/W				P01	– *2	High	Low	
					P00	– *2	High	Low	
	When the serial I/F is used (ESIF = 1):				P03	– *2	High	Low	Master mode: I/O port P03
	P00 = SIN (in), P01 = SOUT (out),				P03	– *2	1	0	Slave mode: General-purpose register
	P02 = SCLK (master: out, slave: in),				P02	– *2	1	0	General-purpose register
	P03 = SRDY (slave: out),				P01	– *2	1	0	
	P03 = I/O port (master: in/out)				P00	– *2	1	0	

Remarks

*1 Initial value at the time of initial reset
*2 Not set in the circuit
*3 Undefined
*4 Reset (0) immediately after being read
*5 Constantly "0" when being read

I/O memory map (CCH–CFH)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
CCH	0	0	SCTRG	ESIF	0 *5	– *2			Unused
					0 *5	– *2			Unused
					SCTRG(W)	– *2	Trigger	–	Serial interface clock trigger (writing)
	R		R/W		SCTRG(R)	0	Run	Stop	Serial interface clock status (reading)
					ESIF	0	SIF port	I/O port	P0 port function selection
CDH	SDP	SCPS	SCS1	SCS0	SDP	0	LSB first	MSB first	Serial data input/output permutation
					SCPS	0	⌐	¬	Serial interface clock phase selection
	R/W				SCS1	0			Serial interface clock mode selection
					SCS0	0			0: Slave, 1: CLK/2, 2: CLK, 3: CLK
CEH	SD3	SD2	SD1	SD0	SD3	– *2			MSB
					SD2	– *2			Serial interface data (low-order 4 bits)
	R/W				SD1	– *2			
					SD0	– *2			LSB
CFH	SD7	SD6	SD5	SD4	SD7	– *2			MSB
					SD6	– *2			Serial interface data (high-order 4 bits)
	R/W				SD5	– *2			
					SD4	– *2			LSB

I/O memory map (D0H–D6H)

Address	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
D0H	0	0	BZFQ	R	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		BZFQ	0	2 kHz	4 kHz	Buzzer signal frequency selection
					R	0	1	0	1 bit general-purpose register
D1H	0	BSHOT	BZFNC	BZON	0 *5	– *2			Unused
					BSHOT*5	– *2	One-shot	–	One-shot buzzer signal (31 msec) output trigger
	R	W	R/W		BZFNC	0	Intermittent	Continuous	Continuous/intermittent output selection
					BZON	0	On	Off	Buzzer signal output control
D4H	0	0	WDRST	TMRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		W		WDRST*5	– *2	Reset	–	Watchdog timer reset
					TMRST*5	– *2	Reset	–	Clock timer and watchdog timer reset
D5H	TM3	TM2	TM1	TM0	TM3	– *3			Clock timer data (16 Hz)
					TM2	– *3			Clock timer data (32 Hz)
	R				TM1	– *3			Clock timer data (64 Hz)
					TM0	– *3			Clock timer data (128 Hz)
D6H	TM7	TM6	TM5	TM4	TM7	– *3			Clock timer data (1 Hz)
					TM6	– *3			Clock timer data (2 Hz)
	R				TM5	– *3			Clock timer data (4 Hz)
					TM4	– *3			Clock timer data (8 Hz)

I/O memory map (E0H–EBH)

Address	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
E0H	–	–	–	VSS2	–	– *3			Unused (Undefined when reading)
					–	– *3			Unused (Undefined when reading)
	R			R/W	–	– *3			Unused (Undefined when reading)
					VSS2	0	On	Off	VSS2 booster control
E1H	LOFF	0	LDTY	LPWR	LOFF	0	All off	Normal	LCD display all off / normal switch
					0 *5	– *2			Unused
	R/W	R	R/W		LDTY	0	1/3	1/4	LCD drive duty selection
					LPWR	0	On	Off	LCD driver On/Off
E2H	BUFF	IIN	ADSPD	ADON	BUFF	0	–	–	Fix at 0
					IIN	0	IIH	IIL	Current measurement terminal switching
	R/W				ADSPD	0	100 mS	400 mS	A/D conversion speed switching
					ADON	0	On	Off	A/D converter On/Off
E3H	0	0	SVDDT	SVDON	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	SVDDT	0	Low	Normal	Supply voltage detection data
					SVDON	0	On	Off	SVD circuit On/Off
E4H	0	FNC2	FNC1	FNC0	0 *5	– *2			Unused
					FNC2	0			Measurement function selection (See Table 4.1.2)
	R	R/W			FNC1	0			
					FNC0	0			
E5H	0	RNG2	RNG1	RNG0	0 *5	– *2			Unused
					RNG2	0			Measurement range selection (See Table 4.1.3)
	R	R/W			RNG1	0			
					RNG0	0			
E6H	0	0	0	ADP	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					ADP	1	Positive	Negative	A/D converter polarity judgment
E7H	DSC03	DSC02	DSC01	DSC00	DSC03	0	1	0	A/D conversion data (00–03)
					DSC02	0	1	0	
	R				DSC01	0	1	0	
					DSC00	0	1	0	
E8H	DSC13	DSC12	DSC11	DSC10	DSC13	0	1	0	A/D conversion data (10–13)
					DSC12	0	1	0	
	R				DSC11	0	1	0	
					DSC10	0	1	0	
E9H	DSC23	DSC22	DSC21	DSC20	DSC23	0	1	0	A/D conversion data (20–23)
					DSC22	0	1	0	
	R				DSC21	0	1	0	
					DSC20	0	1	0	
EAH	0	DSC32	DSC31	DSC30	0 *5	– *2			Unused
					DSC32	0	1	0	A/D conversion data (30–32)
	R				DSC31	0	1	0	
					DSC30	0	1	0	
EBH	0	IDR	STS1	STS0	0 *5	– *2			Unused
					IDR	0	Invalid	Effective	Read data status
	R				STS1	0			A/D conversion status 0: auto zero adjustment,
					STS0	0			1: input integral, 3: reverse integral

I/O memory map (F0H–F6H)

Address	Register D3	D2	D1	D0	Name	Init *1	1	0	Comment
F0H	EIK1	EIK0	EISIF	EIAD	EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
	R/W				EISIF	0	Enable	Mask	Interrupt mask register (Serial interface)
					EIAD	0	Enable	Mask	Interrupt mask register (A/D converter)
F1H	EIT1	EIT2	EIT16	EIT32	EIT1	0	Enable	Mask	Interrupt mask register (Clock timer 1 Hz)
					EIT2	0	Enable	Mask	Interrupt mask register (Clock timer 2 Hz)
	R/W				EIT16	0	Enable	Mask	Interrupt mask register (Clock timer 16 Hz)
					EIT32	0	Enable	Mask	Interrupt mask register (Clock timer 32 Hz)
F2H	0	0	0	IK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK1 *4	0	Yes	No	Interrupt factor flag (K10–K13)
F3H	0	0	0	IK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
F4H	0	0	0	ISIF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					ISIF *4	0	Yes	No	Interrupt factor flag (Serial interface)
F5H	IT1	IT2	IT16	IT32	IT1 *4	0	Yes	No	Interrupt factor flag (Clock timer 1 Hz)
					IT2 *4	0	Yes	No	Interrupt factor flag (Clock timer 2 Hz)
	R				IT16 *4	0	Yes	No	Interrupt factor flag (Clock timer 16 Hz)
					IT32 *4	0	Yes	No	Interrupt factor flag (Clock timer 32 Hz)
F6H	0	0	0	IAD	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IAD *4	0	Yes	No	Interrupt factor flag (A/D converter)

A/D converter measurement function list

Address	D3	D2 FNC2	D1 FNC1	D0 FNC0	Measurement function	Integral resistor normal mode (400 ms)	Integral resistor high speed mode (100 ms)	General amplifier	Comparator
E4H	–	0	0	0	DC voltmeter mode	BUF1 terminal	BUF3 terminal	OFF	OFF
	–	0	0	1	AC voltmeter mode			ON	OFF
	–	0	1	0	DC ammeter mode			OFF	OFF
	–	0	1	1	AC ammeter mode			ON	OFF
	–	1	0	0	Resistance measurement mode	Input integral: BUF1 terminal		OFF	OFF
	–	1	0	1	Continuity check mode	Reverse integral: BUF1 and BUF2 terminals parallel		OFF	ON
	–	1	1	0	–	BUF1 terminal	BUF3 terminal	OFF	OFF
	–	1	1	1	ADPT mode			OFF	OFF

∗ In the resistance measurement mode and continuity check mode, switching between input integral (BUF1 terminal) and reverse integral (BUF1 and BUF2 terminals parallel) is automatically done by the hardware.

A/D converter measurement range list

Address	D3	D2 RNG2	D1 RNG1	D0 RNG0	Measurement function: DC voltmeter	AC voltmeter	Resistance	Continuity check	Diode	Current
E5H	–	0	0	0	400 mV	400 mV	400 Ω	50 Ω judgment	Fixed at 4 V range	Switching outside of IC
	–	0	0	1	4 V	4 V	4 kΩ	100 Ω judgment		
	–	0	1	0	40 V	40 V	40 kΩ	500 Ω judgment		
	–	0	1	1	400 V	400 V	400 kΩ	1 kΩ judgment		
	–	1	0	0	1000 V	750 V	4 MΩ	↑		
	–	1	0	1	↑	↑	40 MΩ	↑		
	–	1	1	0	↑	↑	↑	↑		
	–	1	1	1	↑	↑	↑	↑		

∗ In the current measurement mode, the E0C62M2 performs an A/D conversion using a voltage value (within ±437 mV) input from the IIL terminal or IIH terminal. Consequently, it is not necessary to switch the range.

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE6200	Nothing appears on the screen, or nothing works, after activation.	Check the following and remedy if necessary: • 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 ICS62M2J.EXE PC-DOS ICS62M2W.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 unit set correctly? • Is the breaker of the ICE6200 set to ON?
	The ICE6200 breaker tripped immediately after activation.	Check the following and remedy if necessary: • Are connectors F1 and F5 connected to the EVA62M2 correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE6200 is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS62M2P.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.
SOG62M2	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
ASM62M2	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.
MDC62M2	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?
EVA62M2	The EVA62M2 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) • Is the EN/DIS switch on the EVA62M2 set to EN?
	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. • Has the VADJ VR inside the EVA62M2 top cover been turned to a lower setting?

EPSON International Sales Operations

AMERICA

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

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

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

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

EUROPE

- HEADQUARTERS -
EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15
80992 Muenchen, GERMANY
Phone : +49-(0)89-14005-0 Fax : +49-(0)89-14005-110

- GERMANY -
EPSON EUROPE ELECTRONICS GmbH
SALES OFFICE
Breidenbachstrasse 46
D-51373 Leverkusen, GERMANY
Phone : +49-(0)214-83070-0 Fax : +49-(0)214-83070-10

- UNITED KINGDOM -
EPSON EUROPE ELECTRONICS GmbH
UK BRANCH OFFICE
G6 Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 Fax: +44-(0)1344-381701

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

ASIA

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

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

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

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
HSINCHU OFFICE
13F-3,No.295,Kuang-Fu Road,Sec.2
HsinChu 300,TAIWAN,R.O.C.
Phone: 03-573-9900 Fax: 03-573-9169

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

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

- JAPAN -
SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

Electronic Device Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 Fax: +81-(0)42-587-5624

ED International Marketing Department I
(Europe & U.S.A.)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5812 Fax: +81-(0)42-587-5564

ED International Marketing Department II (Asia)
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 Fax: +81-(0)42-587-5110

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

EPSON

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ Electronic devices information on the Epson WWW server

http://www.epson.co.jp

Issue APRIL 1998 Printed in Japan Ⓜ A