MF754-02

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

E0C62T3 DEVELOPMENT TOOL MANUAL

ENERGY SAVING
EPSON

SEIKO EPSON CORPORATION

E0C62T3 Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C62T3.

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 EVA62T3 Manual ICE6200 Hardware Manual
Development procedure	☞	E0C62 Family Technical Guide
Device (E0C62T3)	☞	E0C62T3 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 DEV62T3

The below software are included in the product of the E0C62T3 development support tool DEV62T3.

1.	Development Tool Management System DMS6200	Menu selection for each software / start-up software
2.	Cross Assembler ASM62T3	Cross assembler for program preparation
3.	Function Option Generator FOG62T3	Function option data preparation program
4.	Segment Option Generator SOG62T3	Segment option data preparation program
5.	ICE Control Software ICS62T3	ICE control program
6.	Mask Data Checker MDC62T3	Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV62T3 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 E0C62T3, 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.

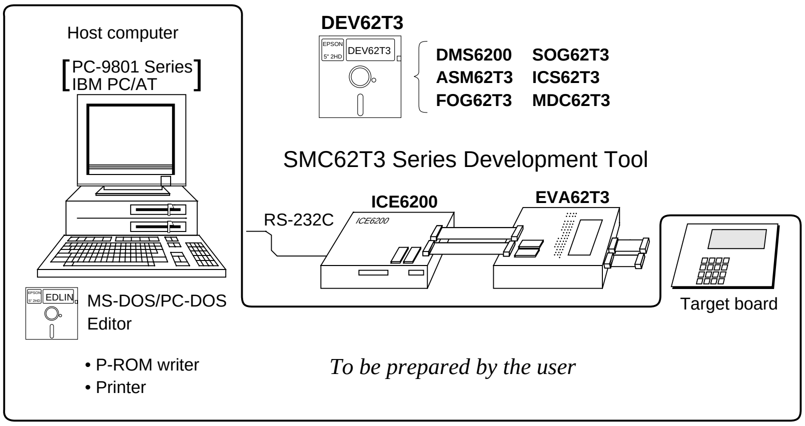


Fig. 1.2.1 System configuration

Note The DEV62T3 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 DEV62T3.

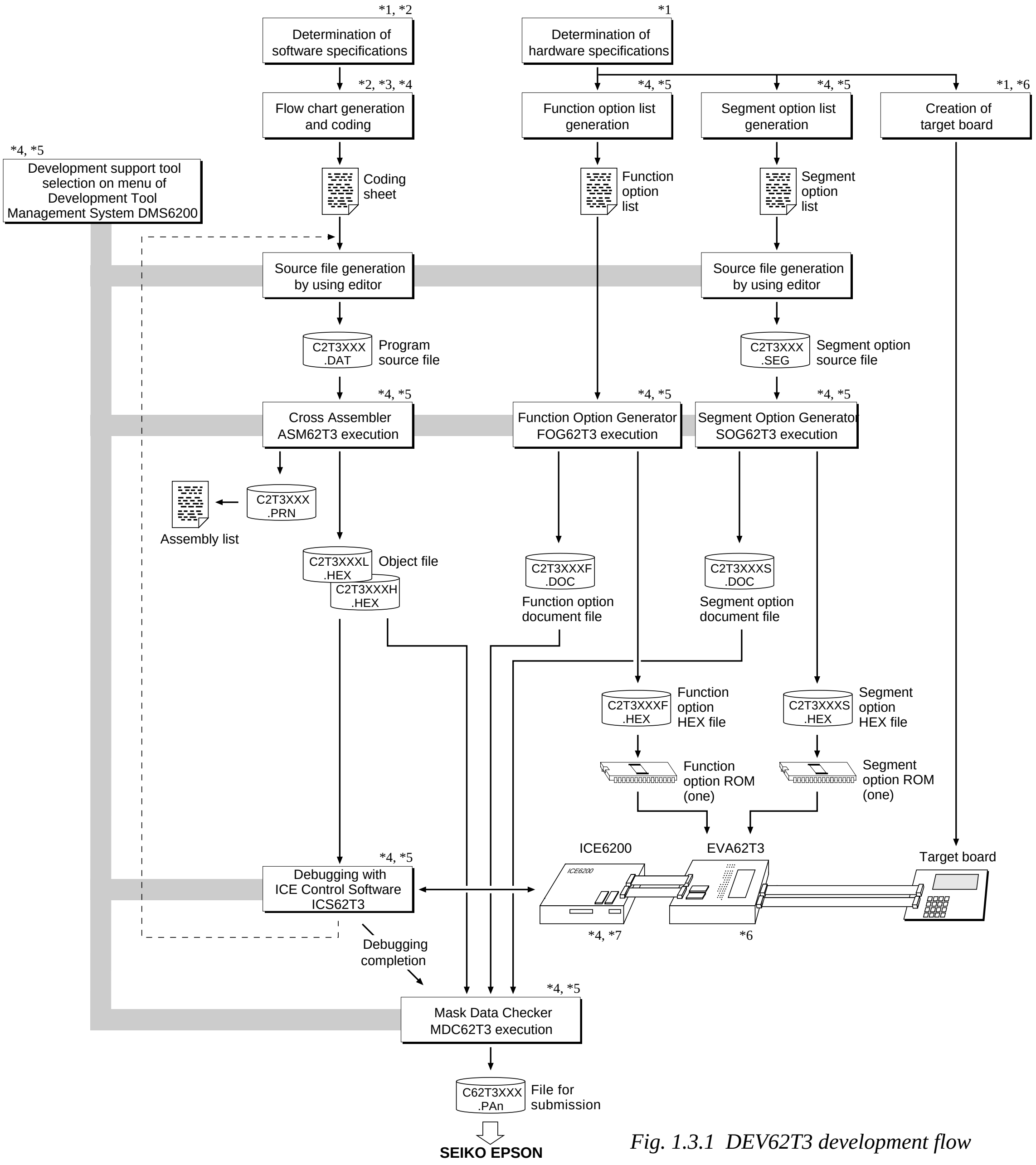

*1, *2
Determination of software specifications
*1
Determination of hardware specifications
*2, *3, *4
Flow chart generation and coding
*4, *5
Function option list generation
*4, *5
Segment option list generation
*1, *6
Creation of target board
*4, *5
Development support tool selection on menu of Development Tool Management System DMS6200
Coding sheet
Function option list
Segment option list
Source file generation by using editor
Source file generation by using editor
C2T3XXX .DAT
Program source file
C2T3XXX .SEG
Segment option source file
*4, *5
Cross Assembler ASM62T3 execution
*4, *5
Function Option Generator FOG62T3 execution
*4, *5
Segment Option Generator SOG62T3 execution
C2T3XXX .PRN
Assembly list
C2T3XXXL .HEX
Object file
C2T3XXXH .HEX
C2T3XXXF .DOC
Function option document file
C2T3XXXS .DOC
Segment option document file
C2T3XXXF .HEX
Function option HEX file
C2T3XXXS .HEX
Segment option HEX file
Function option ROM (one)
Segment option ROM (one)
ICE6200
EVA62T3
Target board
ICE6200
*4, *7
*6
*4, *5
Debugging with ICE Control Software ICS62T3
Debugging completion
*4, *5
Mask Data Checker MDC62T3 execution
C62T3XXX .PAn
File for submission
SEIKO EPSON

Fig. 1.3.1 DEV62T3 development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C2T3XXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1	E0C62T3 Technical Hardware Manual	*5	E0C62T3 Development Tool Manual (this manual)
*2	E0C62T3 Technical Software Manual	*6	EVA62T3 Manual
*3	E0C6200/6200A Core CPU Manual	*7	ICE6200 Hardware Manual
*4	E0C62 Family Development Tool Reference 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 DEV62T3 floppy disk.
The content of the files contained in the DEV62T3 floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM62T3.EXE	ASM62T3.EXE	Cross Assembler execution file
DMS6200.EXE	DMS6200.EXE	Development Tool Management System execution file
FOG62T3.EXE	FOG62T3.EXE	Function Option Generator execution file
ICS62T3B.BAT	ICS62T3.BAT	ICE Control Software batch file
ICS62T3W.EXE	ICS62T3J.EXE	ICE Control Software execution file
ICS62T3P.PAR	ICS62T3P.PAR	ICE Control Software parameter file
MDC62T3.EXE	MDC62T3.EXE	Mask Data Checker execution file
SOG62T3.EXE	SOG62T3.EXE	Segment Option Generator execution file

■ First copy the entire content of this disk using commands such as DISKCOPY then make the execution disk. Carefully conserve the original floppy disk for storage purposes.
When copying into a hard disk, make a subdirectory with an appropriate name (DEV62T3, 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 (DEV62T3), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV62T3↵
C\>CD DEV62T3↵
C\DEV62T3\>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.)
<u>Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC62T3 will handle many files.</u>

Example:

Setting of FILES (CONFIG.SYS)

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

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

■ 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 "ICS62T3(B).BAT" the batch process is indicated such that the ICS62T3J(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.
<u>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.</u>

RS-232C Setting (PC-DOS version)

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

RS-232C Setting (MS-DOS version)

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

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 DEV62T3 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.

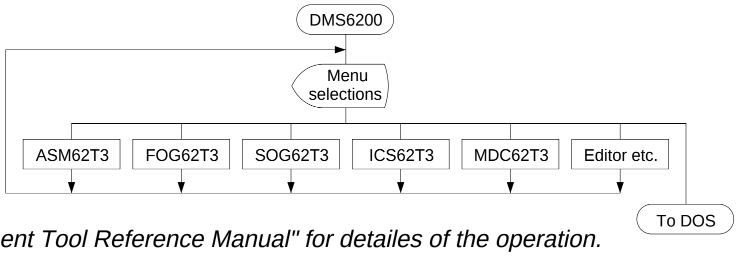


Fig. 2.1.1 DMS6200 execution flow

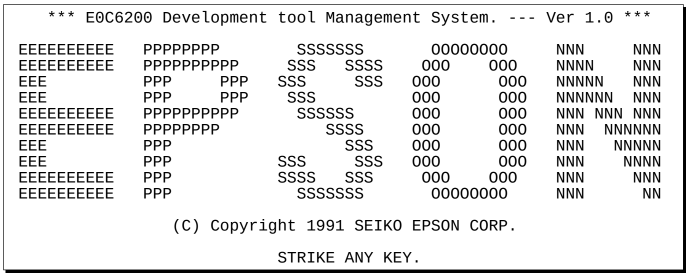
Refer to the "E0C62 Family Development Tool Reference Manual" for detailes of the operation.

2.2 DMS6200 Quick Reference

■ Starting command

Execution file: **DMS6200.EXE** *Starting command:* **DMS6200↵** ↵ *indicates the Return key.*

■ Display examples

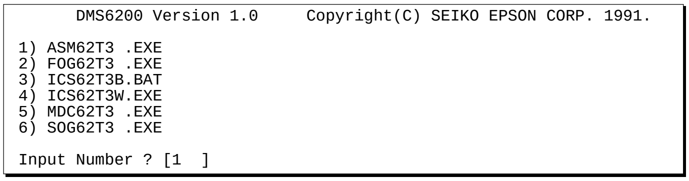


```
   *** E0C6200 Development tool Management System. --- Ver 1.0 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO     NNN     NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    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       OOOOOOOO     NNN      NN

                (C) Copyright 1991 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.

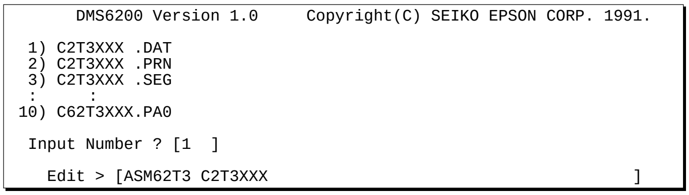


```
      DMS6200 Version 1.0     Copyright(C) SEIKO EPSON CORP. 1991.

1) ASM62T3 .EXE
2) FOG62T3 .EXE
3) ICS62T3B.BAT
4) ICS62T3W.EXE
5) MDC62T3 .EXE
6) SOG62T3 .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.



```
      DMS6200 Version 1.0     Copyright(C) SEIKO EPSON CORP. 1991.

 1) C2T3XXX .DAT
 2) C2T3XXX .PRN
 3) C2T3XXX .SEG
 :      :
10) C62T3XXX.PA0

 Input Number ? [1  ]

   Edit > [ASM62T3 C2T3XXX                                       ]
```

Source file selection screen

To starting ASM62T3, 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 ASM62T3

3.1 ASM62T3 Outline

The ASM62T3 cross assembler is an assembler program for generating the machine code used by the E0C62T3 4-bit, single-chip microcomputers. The Cross Assembler ASM62T3 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.

☞ *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.*

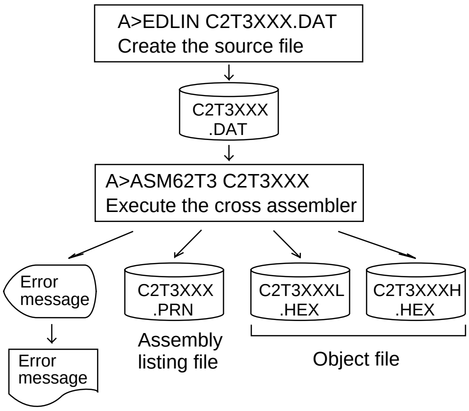


Fig. 3.1.1 ASM62T3 execution flow

3.2 E0C62T3 Restrictions

Note the following when generating a program by the E0C62T3:

■ ROM area

The capacity of the E0C62T3 ROM is 3K steps (0000H to 0BFFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank: Only bank 0, Page: 12 pages (0 to 0BH), each 256 steps

Significant specification range:

ORG	pseudo-instruction:	0000H to 0BFFH
PAGE	pseudo-instruction:	00H to 0BH
BANK	pseudo-instruction:	Only 0H
PSET	instruction:	00H to 0BH

■ RAM area

The capacity of the E0C62T3 RAM is 731 words (000H to 4EBH, 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.
0 to 4 page: B0H–BFH, CFH, ECH–FFH

(2) Since RAM is set for up to 4 page, only the subordinate 3 bits of the page section of the index register which specifies address is effective. (The 1 superordinate bit is ignored.)

Example:

```
LD    A,5
LD    XP,A
LD    X,00H
```

00H is loaded into the IX register, but an unused area has been specified so that the memory accessible with the IX register (MX, 500H) is invalid.

■ Undefined codes

The SLP instruction has not been defined in the E0C62T3 instruction sets.

3.3 ASM62T3 Quick Reference

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

■ Starting command and input/output files

Execution file: ASM62T3.EXE

Starting command: **ASM62T3_ [drive-name:] source-file-name [.shp] _ [-N]↵**

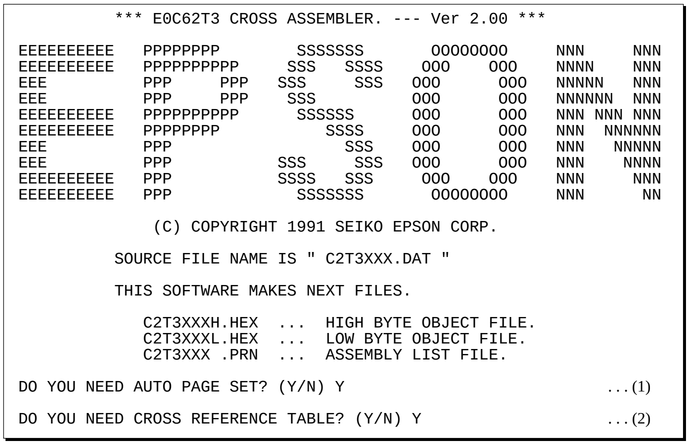
Option:

.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: C2T3XXX.DAT (Source file)

Output file:
C2T3XXXL.HEX (Object file, low-order)
C2T3XXXH.HEX (Object file, high-order)
C2T3XXX.PRN (Assembly listing file)

■ Display example



```
          *** E0C62T3 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 1991 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C2T3XXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C2T3XXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C2T3XXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C2T3XXX .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 ASM62T3 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, ASM62T3 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 FOG62T3*

4.1 FOG62T3 Outline

With the 4-bit single-chip E0C62T3 microcomputers, the customer may select 7 hardware options. By modifying the mask patterns of the E0C62T3 according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG62T3 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 FOG62T3, the E0C62T3 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA62T3) 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.*

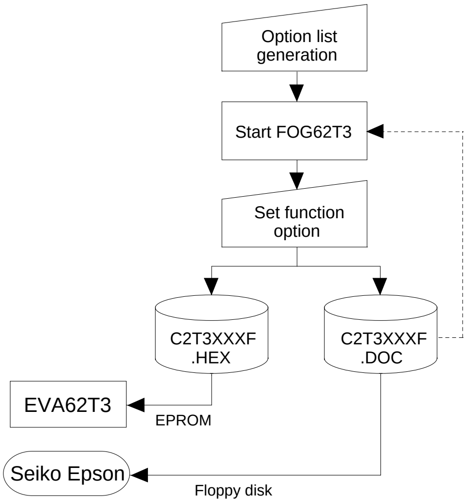


Fig. 4.1.1 FOG62T3 execution flow

4.2 E0C62T3 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. OSC3 SYSTEM CLOCK

□ 1. Not Use
□ 2. Use <Crystal>
□ 3. Use <Ceramic>

2. MULTIPLE KEY ENTRY RESET

- COMBINATION .... □ 1. Not Use
  □ 2. Use <K00, K01>
  □ 3. Use <K00, K01, K02>
  □ 4. Use <K00, K01, K02, K03>
- TIME AUTHORIZE .... □ 1. Not Use □ 2. Use

3. INPUT PORT PULL UP RESISTOR

- K00 .... □ 1. With Resistor □ 2. Gate Direct
- K01 .... □ 1. With Resistor □ 2. Gate Direct
- K02 .... □ 1. With Resistor □ 2. Gate Direct
- K03 .... □ 1. With Resistor □ 2. Gate Direct
- 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
- K20 .... □ 1. With Resistor □ 2. Gate Direct
- K21 .... □ 1. With Resistor □ 2. Gate Direct
- K22 .... □ 1. With Resistor □ 2. Gate Direct

4. I/O PORT OUTPUT SPECIFICATION

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

5. OUTPUT PORT OUTPUT SPECIFICATION

- R00 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R01 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R02 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R03 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R10 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R11 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R12 .... □ 1. Complementary □ 2. Nch-OpenDrain
- R13 .... □ 1. Complementary □ 2. Nch-OpenDrain

6. MUTE PORT OUTPUT SPECIFICATION

- XRMUTE .... □ 1. Complementary □ 2. Nch-OpenDrain
- XTMUTE .... □ 1. Complementary □ 2. Nch-OpenDrain

7. PULSE PORT OUTPUT SPECIFICATION

- XDP .... □ 1. Complementary □ 2. Nch-OpenDrain

4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C62T3 are shown below, and their specifications are also described.

1 OSC3 system clock

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

--- OSC3 SYSTEM CLOCK ---
                         1. NOT USE
                         2. USE <CR>
                         3. USE <CERAMIC>

PLEASE SELECT NO.(1) ? 2↵

                         2. USE <CR>  SELECTED
```

Select whether the OSC3 system clock will be used or not. If you use OSC3 system clock, either crystal oscillation circuit or ceramic oscillation circuit can be selected.

To minimize current consumption and to obtain an accurate oscillation frequency, crystal oscillation circuit would be suitable. When crystal oscillation circuit is selected, crystal oscillator and gate capacity are needed as external components.

On the other hand, when ceramic oscillation circuit is selected, ceramic oscillator, gate capacity and drain capacity are needed as external components.

2 Multiple key entry reset

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

--- MULTIPLE KEY ENTRY RESET ---
       COMBINATION       1. NOT USE
                         2. USE <K00,K01>
                         3. USE <K00,K01,K02>
                         4. USE <K00,K01,K02,K03>

PLEASE SELECT NO.(1) ? 2↵

       TIME AUTHORIZE    1. NOT USE
                         2. USE

PLEASE SELECT NO.(1) ? 2↵

       COMBINATION       2. USE <K00,K01>  SELECTED
       TIME AUTHORIZE    2. USE  SELECTED
```

* If "Not Use" is set for the combination, the time authorize selection is not required.

The reset function and time authorize circuit are set when K00 through K03 are entered.

When "Not Use" is set for the combination, the reset function is not activated even if K00 through K03 are entered. When "Use <K00, K01>" is set, the system is reset immediately the K00 and K01 inputs go low at the same time. Similarly, the system is reset as soon as the K00 through K02 inputs or the K00 through K03 inputs go low.

When "Use" is set for the time authorize circuit, a simultaneous low input time is authorized. The system is reset when a signal is input for more than 1 to 2 sec.

If the time authorize circuit is not used, the system is reset when a low signal is input for more than 1.5 msec.

The multiple key entry reset circuit is shown in Figure 4.3.1.

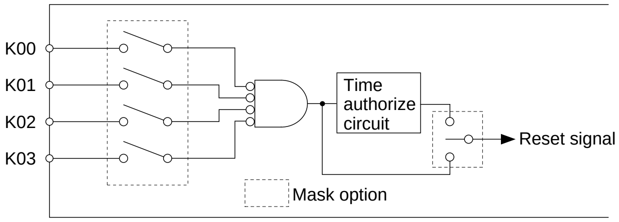


Fig. 4.3.1
Multiple key entry reset circuit

3 Input port pull up resistor

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

--- INPUT PORT PULL UP 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 K21)
                                   :

PLEASE SELECT NO.(1) ? 1↵

        K22              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
        K20              1. WITH RESISTOR  SELECTED
        K21              1. WITH RESISTOR  SELECTED
        K22              1. WITH RESISTOR  SELECTED
```

Select whether input ports (K00–K03, K10–K13 and K20–K22) will each be supplemented with pull up resistors or not. When "Gate Direct" is selected, see to it that entry floating state does not occur. Moreover, the input port status is changed from low level (VSS) to high (VDD) with pull up resistors, a delay in waveform rise time will occur depending on the pull up resistor and entry load time constant. Because of this, when input reading is to be conducted, ensure the appropriate wait time with the program.
Select "With Resistor" pull up resistor for unused ports.
The configuration of the pull up resistor circuit is shown in Figure 4.3.2.

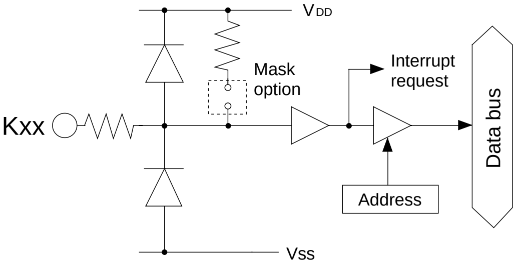


Fig. 4.3.2 Configuration of pull up resistor circuit

4 I/O port output specification

```
*** OPTION  NO.4 ***

--- I/O PORT OUTPUT SPECIFICATION ---
        P00             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P01             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P02             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        P03             1. COMPLEMENTARY
                        2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

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

Select the output specification to be used during I/O port (P00–P03) output mode selection.
Either complementary output or Nch open drain output may be selected.
Select complementary output for unused port.

The I/O port can control the input/output direction according to the IOC0–IOC3 registers (address D5H); at "1" and "0" settings, it is set to output port and input port, respectively.

When I/O port is set to input mode, it can control the PUP0–PUP3 registers (address D6H); at "1" and "0" settings, it is set to pull up and no pull up resistor, respectively.

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

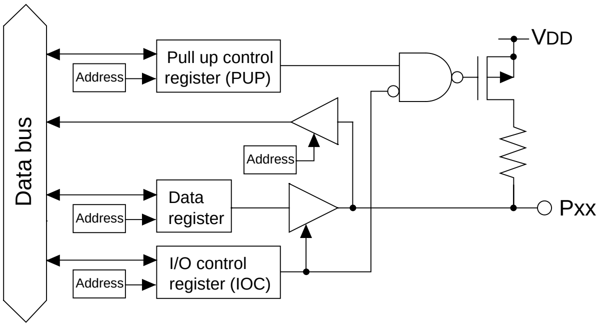


Fig. 4.3.3
Circuit configuration of I/O port

5 Output port output specification

```
*** OPTION  NO.5 ***

--- OUTPUT PORT OUTPUT SPECIFICATION ---
        R00            1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R01            1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵
                                       :
                        (Selection for R02 to R12)
                                       :

PLEASE SELECT NO.(1) ? 1↵

        R13            1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        R00            1. COMPLEMENTARY  SELECTED
        R01            1. COMPLEMENTARY  SELECTED
        R02            1. COMPLEMENTARY  SELECTED
        R03            1. COMPLEMENTARY  SELECTED
        R10            1. COMPLEMENTARY  SELECTED
        R11            1. COMPLEMENTARY  SELECTED
        R12            1. COMPLEMENTARY  SELECTED
        R13            1. COMPLEMENTARY  SELECTED
```

Select the output specification for the output ports (R00–R03 and R10–R13).
Either complementary output or Nch open drain output may be selected.
When output port is to be used on key matrix configuration, select Nch open drain output.
For unused output ports, select complementary output.
The output circuit configuration is shown in Figure 4.3.4.

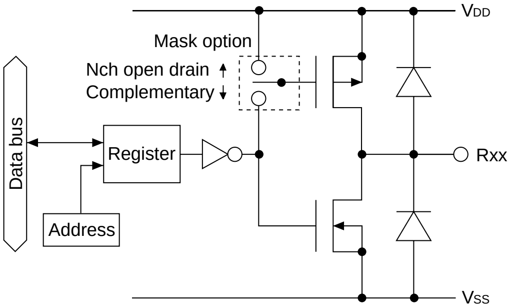


Fig. 4.3.4 Circuit configuration of output port

6 Mute port output specification

```
*** OPTION  NO.6 ***

--- MUTE PORT OUTPUT SPECIFICATION ---
        XRMUTE         1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        XTMUTE         1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        XRMUTE         1. COMPLEMENTARY  SELECTED
        XTMUTE         1. COMPLEMENTARY  SELECTED
```

Select the output specification for $\overline{\text{RMUTE}}$ and $\overline{\text{TMUTE}}$ terminals.
Either complementary output or Nch open drain output may be selected.
The circuit configuration is shown in Figure 4.3.5.

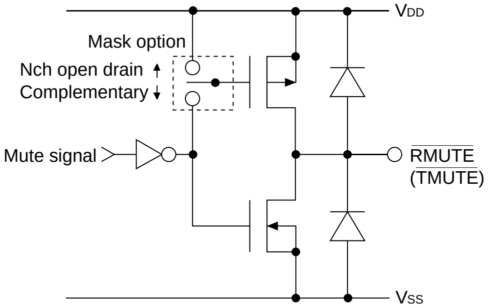


Fig. 4.3.5 Circuit configuration of mute port

7 Pulse port output specification

```
*** OPTION  NO.7 ***

--- PULSE PORT OUTPUT SPECIFICATION ---
        XDP            1. COMPLEMENTARY
                       2. NCH-OPENDRAIN

PLEASE SELECT NO.(1) ? 1↵

        XDP            1. COMPLEMENTARY  SELECTED
```

Select the output specification for $\overline{\text{DP}}$ terminal.
Either complementary output or Nch open drain output may be selected.
The circuit configuration is shown in Figure 4.3.6.

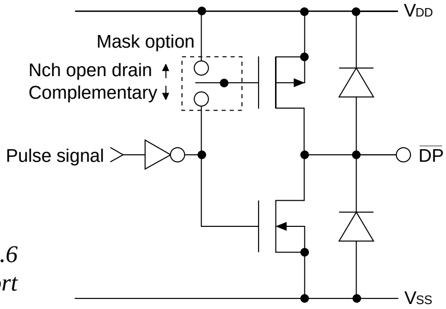


Fig. 4.3.6
Circuit configuration of pulse port

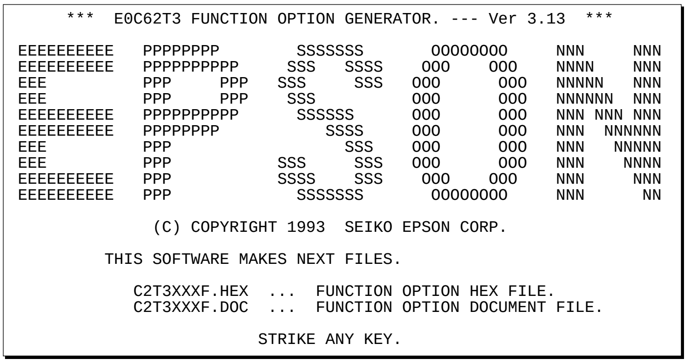
4.4 *FOG62T3 Quick Reference*

■ Starting command and input/output files

Execution file:	FOG62T3.EXE
Starting command:	**FOG62T3↵**
Input file:	C2T3XXXF.DOC (Function option document file, when modifying)
Output file:	C2T3XXXF.DOC (Function option document file) C2T3XXXF.HEX (Function option HEX file)

↵ *indicates the Return key.*

■ Display example



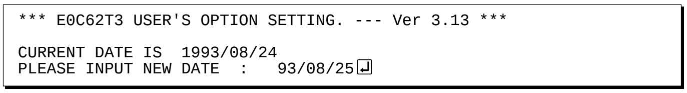
```
     ***  E0C62T3 FUNCTION OPTION GENERATOR. --- Ver 3.13  ***

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.

         THIS SOFTWARE MAKES NEXT FILES.

            C2T3XXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C2T3XXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

Start-up message

When FOG62T3 is started, the start-up message is displayed.

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

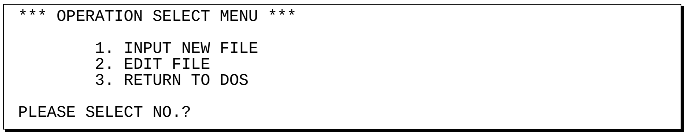
To suspend execution, press the "CTRL" and "C" keys together: the sequence returns to the DOS command level.



```
*** E0C62T3 USER'S OPTION SETTING. --- Ver 3.13 ***

CURRENT DATE IS  1993/08/24
PLEASE INPUT NEW DATE  :   93/08/25↵
```

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.

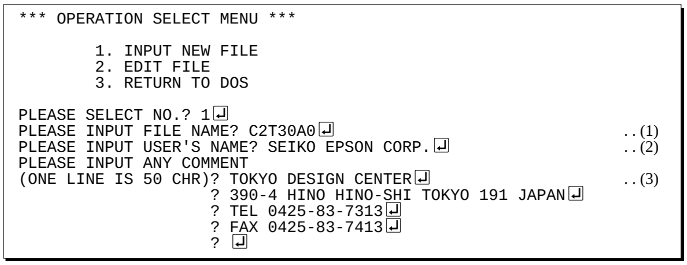


```
*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
```

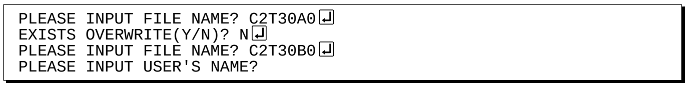
Operation selection menu

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 FOG62T3.



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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C2T30A0↵                                ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                    ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                      ..(3)
                    ? 390-4 HINO HINO-SHI TOKYO 191 JAPAN↵
                    ? TEL 0425-83-7313↵
                    ? FAX 0425-83-7413↵
                    ? ↵
```



```
PLEASE INPUT FILE NAME? C2T30A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C2T30B0↵
PLEASE INPUT USER'S NAME?
```

Setting new function options

Select "1" on the operation selection menu.

(1) Enter the file name.
(2) Enter the customer's company name.
(3) Enter any comment.
    (Within 50 characters x 10 lines)

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

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

Modifying function option settings

Select "2" on the operation selection menu.

(1) Will display the files on the current drive.

(2) Enter the file name.

(3) Enter the customer's company name.

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

(5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected.
Enter "E↵" to end option setting. Then, move to the confirmation procedure for HEX file generation.

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

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

PLEASE SELECT NO.? 2↵

*** SOURCE FILE(S) ***

C2T30A0         C2T30B0         C2T30C0                 ..(1)

PLEASE INPUT FILE NAME? C2T30A0↵                        ..(2)
PLEASE INPUT USER'S NAME? ↵                             ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ↵                                 ..(4)
PLEASE INPUT EDIT NO.? 4↵                               ..(5)
                :
(Modifying function option settings)
                :
PLEASE INPUT EDIT NO.? E↵
```

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

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

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

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

```
PLEASE INPUT FILE NAME? C2T30N0↵
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 FOG62T3 is terminated.

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

Option selection

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

In return, the confirmation is displayed.

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

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

--- OSC2 SYSTEM CLOCK ---
                              1. Not Use
                              2. Use <CRYSTAL>
                              3. Use <CERAMIC>

PLEASE SELECT  NO.(1) ? 2↵

                              2. Use <CRYSTAL>  SELECTED
```

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 EVA62T3, 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 EVA62T3 options.

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

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

*** OPTION EPROM SELECT MENU ***

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

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

        2. 27C128   SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
```

4.5 Sample File

■ Example of function option document file

```
* E0C62T3 FUNCTION OPTION DOCUMENT  V 3.13
*
* FILE NAME    C2T30A0F.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   93/08/24
*
* COMMENT      TOKYO DESIGN CENTER
*              390-4 HINO HINO-SHI TOKYO 191 JAPAN
*              TEL 0425-83-7313
*              FAX 0425-83-7413
*
*
* OPTION NO.1
* < OSC3 SYSTEM CLOCK >
*                           USE <CRYSTAL>  -----------------  SELECTED
 OPT0101 02
*
* OPTION NO.2
* < MULTIPLE KEY ENTRY RESET >
*     COMBINATION           USE <K00, K01, K02, K03  -------  SELECTED
*     TIME AUTHORIZE        USE  ---------------------------  SELECTED
 OPT0201 04
 OPT0202 01
*
* OPTION NO.3
* < INPUT PORT PULL UP 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
*     K20                   WITH RESISTOR  -----------------  SELECTED
*     K21                   WITH RESISTOR  -----------------  SELECTED
*     K22                   WITH RESISTOR  -----------------  SELECTED
 OPT0301 01
 OPT0302 01
 OPT0303 01
 OPT0304 01
 OPT0305 01
 OPT0306 01
 OPT0307 01
 OPT0308 01
 OPT0309 01
 OPT0310 01
 OPT0311 01
*
* OPTION NO.4
* < I/O PORT OUTPUT SPECIFICATION >
*     P00                   COMPLEMENTARY  -----------------  SELECTED
*     P01                   COMPLEMENTARY  -----------------  SELECTED
*     P02                   COMPLEMENTARY  -----------------  SELECTED
*     P03                   COMPLEMENTARY  -----------------  SELECTED
 OPT0401 01
 OPT0402 01
 OPT0403 01
 OPT0404 01
*
```

```
* OPTION NO.5
* < OUTPUT PORT OUTPUT SPECIFICATION >
*     R00                    COMPLEMENTARY  -----------------  SELECTED
*     R01                    COMPLEMENTARY  -----------------  SELECTED
*     R02                    COMPLEMENTARY  -----------------  SELECTED
*     R03                    COMPLEMENTARY  -----------------  SELECTED
*     R10                    COMPLEMENTARY  -----------------  SELECTED
*     R11                    COMPLEMENTARY  -----------------  SELECTED
*     R12                    COMPLEMENTARY  -----------------  SELECTED
*     R13                    COMPLEMENTARY  -----------------  SELECTED
 OPT0501 01
 OPT0502 01
 OPT0503 01
 OPT0504 01
 OPT0505 01
 OPT0506 01
 OPT0507 01
 OPT0508 01
*
* OPTION NO.6
* < MUTE PORT OUTPUT SPECIFICATION >
*     XRMUTE                 COMPLEMENTARY  -----------------  SELECTED
*     XTMUTE                 COMPLEMENTARY  -----------------  SELECTED
 OPT0601 01
 OPT0602 01
*
* OPTION NO.7
* < PULSE PORT OUTPUT SPECIFICATION >
*     XDP                    COMPLEMENTARY  -----------------  SELECTED
 OPT0701 01
*
*
*
* SEIKO EPSON'S AREA
*
*
*
* OPTION NO.8
 OPT0801 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 SOG62T3

5.1 SOG62T3 Outline

With the 4-bit single-chip E0C62T3 microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C62T3 according to the selected options, the system can be customized to meet the specifications of the target system.

The Segment Option Generator SOG62T3 is a software tool for generating data file used to generate mask patterns. From the data file created with SOG62T3, the E0C62T3 mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA62T3) 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.*

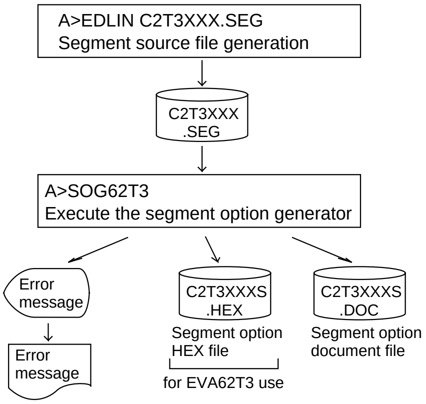


Fig. 5.1.1 SOG62T3 execution flow

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 □ N
SEG2													SEG output
SEG3													DC output □ C □ N
SEG4													SEG output
SEG5													DC output □ C □ N
SEG6													SEG output
SEG7													DC output □ C □ N
SEG8													SEG output
SEG9													DC output □ C □ N
SEG10													SEG output
SEG11													DC output □ C □ N
SEG12													SEG output
SEG13													DC output □ C □ N
SEG14													SEG output
SEG15													DC output □ C □ N
SEG16													SEG output
SEG17													DC output □ C □ N
SEG18													SEG output
SEG19													DC output □ C □ N
SEG20													SEG output
SEG21													DC output □ C □ N
SEG22													SEG output
SEG23													DC output □ C □ N
SEG24													SEG output
SEG25													DC output □ C □ N
SEG26													SEG output
SEG27													DC output □ C □ N
SEG28													SEG output
SEG29													DC output □ C □ N
SEG30													SEG output
SEG31													DC output □ C □ N

Legend: <ADDRESS> <OUTPUT SPECIFICATION>
H: High order address (8–A), L: Low order address (0–F) C: Complementary output
D: Data bit (0–3) N: Nch open drain output

Note:

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

5.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG31, 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 Nch 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 (80H–AFH) can be allocated to the optional segment. With this, up to 128 segments (96, 64 or 32 segments when 1/3, 1/2 or 1/1 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 (8–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  801  800  832  803  S
1  812  811  810  823  S
```

- When 1/3 duty is selected

```
0  801  800  832  ---  S
1  812  811  810  ---  S
```

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 32 terminals may be allocated for DC output. Also, either complementary output or Nch 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 (VDD), and goes low (VSS) when set to "0". Segment allocation is the same as when segment output is selected but for the while the 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 SEG28 and SEG29, and Nch open drain output is set to SEG30 and SEG31.

```
28  900  ---  ---  ---  C
29  910  ---  ---  ---  C
30  920  ---  ---  ---  N
31  930  ---  ---  ---  N
```

5.4 SOG62T3 Quick Reference

■ Starting command and input/output files

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

Execution file: SOG62T3.EXE

Starting command: **SOG62T3_ [-H]↵**

Option: -H: Specifies the segment option document file for input file of SOG62T3.

Input file: C2T3XXX.SEG (Segment option source file)
C2T3XXXS.DOC (Segment option document file, when -H option use)

Output file: C2T3XXXS.DOC (Segment option document file)
C2T3XXXS.HEX (Segment option HEX file)

■ Display example

```
     ***  E0C62T3 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 1993  SEIKO EPSON CORP.

         SEGMENT OPTION SOURCE FILE NAME IS " C2T3XXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C2T3XXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C2T3XXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

Start-up message
When SOG62T3 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.

```
*** E0C62T3 USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  93/09/20
PLEASE INPUT NEW DATE  :   93/09/20↵
```

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

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

C2T30A0         C2T30B0         C2T30C0                       ..(1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C2T30A0↵               ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                  ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                    ..(4)
                    ? 390-4 HINO HINO-SHI TOKYO 191 JAPAN↵
                    ? TEL 0425-83-7313↵
                    ? FAX 0425-83-7413↵
                    ? ↵
```

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

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

*** SOURCE FILE(S) ***

SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.    ..(6)  -H option use
```

```
PLEASE INPUT SEGMENT OPTION SOURCE FILE NAME? C2T30N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.      ..(7)  -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C2T30N0↵
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.    ..(8)  -H option use
```

Input file selection
(1) Will display the files on the current drive.
(2) Enter the file name.
(3) Enter the customer's company name.
(4) Enter any comment.
(Within 50 characters x 10 lines)
Then, move to the confirmation procedure for HEX file generation.

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

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

*** OPTION EPROM SELECT MENU ***

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

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

        2. 27C128   SELECTED

MAKING FILE IS COMPLETED.
```

EPROM selection

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

(1) When debugging the program with EVA62T3, 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 EVA62T3 options.

When a series of operations are complete, the SOG62T3 generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG62T3 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

```
; C2T30A0.SEG, VER.3.21
; EVA62T3 LCD SEGMENT DECODE TABLE
;
 0    800  801  802  803 S       ;1st DIGIT
 1    810  811  812  813 S
 2    820  821  822  823 S
 3    830  831  832  833 S
 4    840  841  842  843 S       ;2nd DIGIT
 5    850  851  852  853 S
 6    860  861  862  863 S
 7    870  871  872  873 S
 8    880  881  882  883 S       ;3rd DIGIT
 9    890  891  892  893 S
10    8A0  8A1  8A2  8A3 S
11    8B0  8B1  8B2  8B3 S
12    8C0  8C1  8C2  8C3 S       ;4th DIGIT
13    8D0  8D1  8D2  8D3 S
14    8E0  8E1  8E2  8E3 S
15    8F0  8F1  8F2  8F3 S
16    900  901  902  903 S       ;5th DIGIT
17    910  911  912  913 S
18    920  921  922  923 S
19    930  931  932  933 S
20    940  941  942  943 S       ;6th DIGIT
21    950  951  952  953 S
22    960  961  962  963 S
23    970  971  972  973 S
24    980  981  982  983 S       ;7th DIGIT
25    990  991  992  993 S
26    9A0  9A1  9A2  9A3 S
27    9B0  9B1  9B2  9B3 S
28    9C0  ---  ---  --- C       ;DC OUTPUT
29    9D0  ---  ---  --- C
30    9E0  ---  ---  --- C
31    9F0  ---  ---  --- C
```

■ Example of segment option source file

```
* E0C62T3 SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C2T30A0S.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   93/09/20
* COMMENT      TOKYO DESIGN CENTER
*              390-4 HINO HINO-SHI TOKYO 191 JAPAN
*              TEL 0425-83-7313
*              FAX 0425-83-7413
*
*
* OPTION NO.9
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
   0  800  801  802  803  S
   1  810  811  812  813  S
   2  820  821  822  823  S
   3  830  831  832  833  S
   4  840  841  842  843  S
   5  850  851  852  853  S
   6  860  861  862  863  S
   7  870  871  872  873  S
   8  880  881  882  883  S
   9  890  891  892  893  S
  10  8A0  8A1  8A2  8A3  S
  11  8B0  8B1  8B2  8B3  S
  12  8C0  8C1  8C2  8C3  S
  13  8D0  8D1  8D2  8D3  S
  14  8E0  8D1  8E2  8E3  S
  15  8F0  8D1  8F2  8F3  S
  16  900  901  902  903  S
  17  910  911  912  913  S
  18  920  921  922  923  S
  19  930  931  932  933  S
  20  940  941  942  943  S
  21  950  951  952  953  S
  22  960  961  962  963  S
  23  970  971  972  973  S
  24  980  981  982  983  S
  25  990  991  992  993  S
  26  9A0  9A1  9A2  9A3  S
  27  9B0  9B1  9B2  9B3  S
  28  9C0  9C1  9C2  9C3  C
  29  9D0  9D1  9D2  9D3  C
  30  9E0  9E1  9E2  9E3  C
  31  9F0  9F1  9F2  9F3  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 ICS62T3

6.1 ICS62T3 Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA62T3 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 ICS62T3.
The ICS62T3 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.

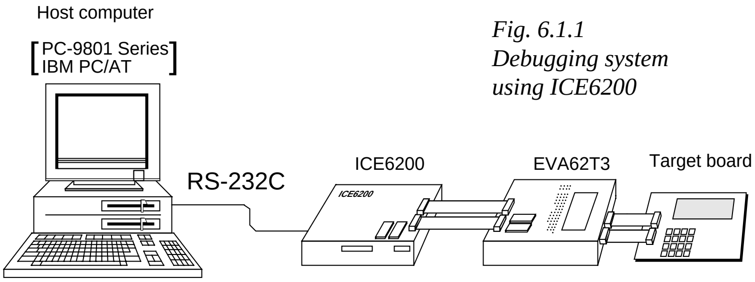


Fig. 6.1.1
Debugging system using ICE6200

☞ *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 ICS62T3 Restrictions

Take the following precautions when using the ICS62T3.

■ ROM Area

The ROM area is limited to a maximum address of 0BFFH. Assigning data above the 0BFFH address causes an error.

■ RAM Area

The RAM area is limited to a maximum address of 4EBH. However, as the following addresses are in the unused area, designation of this area with the ICE commands produces an error.

0 to 4 page: B0H–BFH, CFH, ECH–FFH

(Refer to the "E0C62T3 Technical Manual" for details.)

■ Undefined Code

The SLP instruction is not specified for the E0C62T3 and so cannot be used.

■ OPTLD Command

In the ICS62T3, OPTLD command can be used.
This command is used to load HEX files (function option and segment option data for LCD) in the EVA62T3 memory with the ICE6200.

Load of function option data: `#OPTLD,1,C2T3XXX`↵
Load of segment option data: `#OPTLD,2,C2T3XXX`↵

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 EVA62T3 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 EVA62T3 segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.
Since it takes about 10 minutes (in case of 80386 with 10 MHz clock) to load segment option HEX data, when you want to load at high speed, execute this command by changing the EVA62T3 operation clock from OSC1 to OSC3. (When OSC3 = 3.58 MHz, since it takes about 4 minutes to load segment option HEX data.)

* Since function option HEX file cannot be loaded in OSC3 clock operation, you should not change the operation clock.

Examples

```
#OPTLD,1,C2T3XXX↵ ..... C2T3XXXF.HEX file is loaded in the function option data memory.

#OPTLD,2,C2T3XXX↵ ..... C2T3XXXS.HEX file is loaded in the segment option data memory.

#SD,D8↵
 D8 |0:1↵             ..... The OSC3 oscillation is turned ON.
 D9 |0:/↵

#SD,D8↵
 D8 |1:3↵             ..... Switching from OSC1 to OSC3.
 D9 |0:/↵

#I↵                   ..... The CPU is reset.
                            (Switches CPU clock to OSC1 when OSC3 oscillation is set.)
```

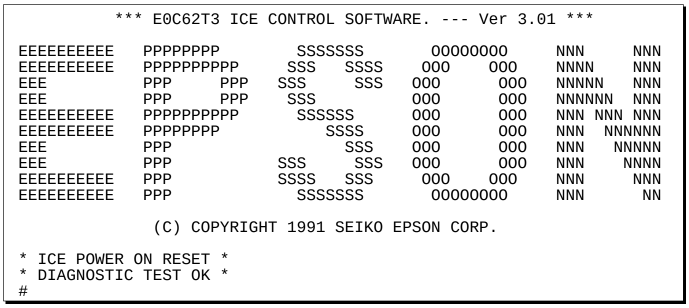
6.3 ICS62T3 Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:	ICS62T3.BAT	(ICS62T3J.EXE)	. . . for MS-DOS
	ICS62T3B.BAT	(ICS62T3W.EXE)	. . . for PC-DOS
Starting command:	**ICS62T3 (ICS62T3J)↵**		. . . for MS-DOS
	ICS62T3B (ICS62T3W)↵		. . . for PC-DOS
Input file:	C2T3XXXL.HEX	(Object file, low-order)	
	C2T3XXXH.HEX	(Object file, high-order)	
	C2T3XXXD.HEX	(Data RAM file)	
	C2T3XXXC.HEX	(Control file)	
	C2T3XXXF.HEX	(Function option HEX file)	
	C2T3XXXS.HEX	(Segment option HEX file)	
Output file:	C2T3XXXL.HEX	(Object file, low-order)	
	C2T3XXXH.HEX	(Object file, high-order)	
	C2T3XXXD.HEX	(Data RAM file)	
	C2T3XXXC.HEX	(Control file)	

■ Display example



```
          *** E0C62T3 ICE CONTROL SOFTWARE. --- Ver 3.01 ***

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

              (C) COPYRIGHT 1991 SEIKO EPSON CORP.

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

Start-up message

When ICS62T3 is started, the start-up message is displayed, and a self-test is automatically performed. ICS62T3 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 ICS62T3 program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS62T3.*

■ 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 ↵ #BAR,a ↵	Sets Break at program address "a" Breakpoint is canceled
		#BD ↵ #BDR ↵	Break condition is set for data RAM Breakpoint is canceled
		#BR ↵ #BRR ↵	Break condition is set for EVA62T3 CPU internal registers Breakpoint is canceled
		#BM ↵ #BMR ↵	Combined break conditions set for program data RAM address and registers Cancel combined break conditions for program data ROM address and registers
		#BRES ↵	All break conditions canceled
		#BC ↵	Break condition displayed
		#BE ↵	Enter break enable mode
		#BSYN ↵	Enter break disable mode
		#BT ↵	Set break stop/trace modes
		#BRKSEL,REM ↵	Set BA condition clear/remain modes
8	Move	#MP,a1,a2,a3 ↵	Contents of program area addresses a1 to a2 are moved to addresses a3 and after
		#MD,a1,a2,a3 ↵	Contents of data area addresses a1 to a2 are moved to addresses a3 and after
9	Data Set	#SP,a ↵	Data from program area address "a" are written to memory
		#SD,a ↵	Data from data area address "a" are written to memory
10	Change CPU Internal Registers	#DR ↵	Display EVA62T3 CPU internal registers
		#SR ↵	Set EVA62T3 CPU internal registers
		#I ↵	Reset EVA62T3 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 ↵	Load function option data from file
		#OPTLD,2,file ↵	Load segment option data from file
13	Coverage	#CVD ↵	Indicates coverage information
		#CVR ↵	Clears coverage information
14	ROM Access	#RP ↵	Move contents of ROM to program memory
		#VP ↵	Compare contents of ROM with contents of program memory
		#ROM ↵	Set ROM type
15	Terminate ICE	#Q ↵	Terminate ICE and return to operating system control
16	Command Display	#HELP ↵	Display ICE6200 instruction
17	Self Diagnosis	#CHK ↵	Report results of ICE6200 self diagnostic test

↵ means press the RETURN key.

7 MASK DATA CHECKER MDC62T3

7.1 MDC62T3 Outline

The Mask Data Checker MDC62T3 is a software tool which checks the program data (C2T3XXXH.HEX and C2T3XXXL.HEX) and option data (C2T3XXXF.DOC and C2T3XXXS.DOC) created by the user and creates the data file (C62T3XXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC62T3 has the capability to restore the generated data file (C62T3XXX.PA0) to the original file format.

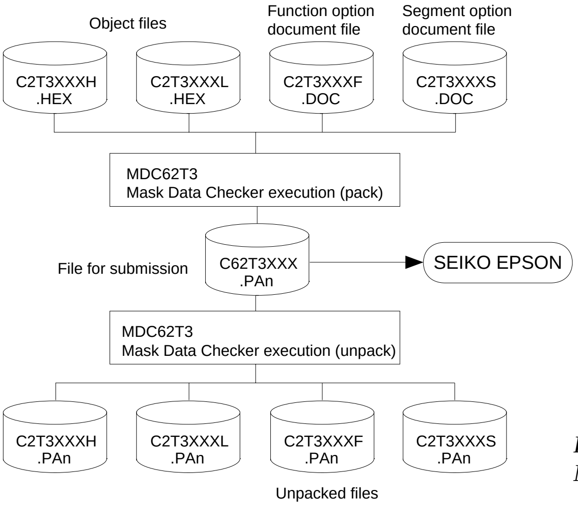


Fig. 7.1.1
MDC62T3 execution flow

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

7.2 MDC62T3 Quick Reference

■ Starting command and input/output files

Execution file:	MDC62T3.EXE		
Starting command:	**MDC62T3**↵		↵ *indicates the Return key.*
Input file:	C2T3XXXL.HEX	(Object file, low-order)	When packing
	C2T3XXXH.HEX	(Object file, high-order)	
	C2T3XXXF.DOC	(Function option document file)	
	C2T3XXXS.DOC	(Segment option document file)	
	C62T3XXX.PAn	(Packed file)	When unpacking
Output file:	C62T3XXX.PAn	(Packed file)	When packing
	C2T3XXXL.PAn	(Object file, low-order)	When unpacking
	C2T3XXXH.PAn	(Object file, high-order)	
	C2T3XXXF.PAn	(Function option document file)	
	C2T3XXXS.PAn	(Segment option document file)	

■ Display examples

```
           *** E0C62T3 PACK / UNPACK PROGRAM Ver 2.000 ***

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

              (C) COPYRIGHT 1991 SEIKO EPSON CORP.

                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

                    PLEASE SELECT NO.?
```

Start-up message

When MDC62T3 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)

        C2T3XXXH.HEX --------+
                             |
        C2T3XXXL.HEX --------+
                             |-------- C2T3XXX.PAn (PACK FILE)
        C2T3XXXF.DOC --------+
                             |
        C2T3XXXS.DOC --------+

 PLEASE INPUT PACK FILE NAME (C62T3XXX.PAn) ?  C62T30A0.PA0↵   ...(2)

        C2T30A0H.HEX --------+
                             |
        C2T30A0L.HEX --------+
                             |-------- C2T30A0.PA0
        C2T30A0F.DOC --------+
                             |
        C2T30A0S.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 "C62T3XXX.PA0" has been submitted, the pack file name should be entered as "C62T3XXX.PA1".)

With this, the mask file (C62T3XXX.PAn) is generated, and the MDC62T3 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 (ASM62T3) 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 (C62T3XXX.PAn) ?  C62T30A0.PA0↵...(2)

                             +-------- C2T30A0H.PA0
                             |
                             +-------- C2T30A0L.PA0
        C62T30A0.PA0 --------|
                             +-------- C2T30A0F.PA0
                             |
                             +-------- C2T30A0S.PA0
```

Unpacking of data

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

(2) Enter the packed file name.

With this, the mask data file (C62T3XXX.PAn) is restored to the original file format, and the MDC62T3 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`	FOG62T3, SOG62T3 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. E0C62T3 INSTRUCTION SET

Classification	Mne-monic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Branch instructions	PSET	p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0					5	NBP←p4, NPP←p3~p0
	JP	s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB←NBP, PCP←NPP, PCS←s7~s0
		C, s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB←NBP, PCP←NPP, PCS←s7~s0 if C=1
		NC, s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB←NBP, PCP←NPP, PCS←s7~s0 if C=0
		Z, s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB←NBP, PCP←NPP, PCS←s7~s0 if Z=1
		NZ, s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0					5	PCB←NBP, PCP←NPP, PCS←s7~s0 if Z=0
	JPBA		1	1	1	1	1	1	1	0	1	0	0	0					5	PCB←NBP, PCP←NPP, PCSH←B, PCSL←A
	CALL	s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1)←PCP, M(SP-2)←PCSH, M(SP-3)←PCSL+1 SP←SP-3, PCP←NPP, PCS←s7~s0
	CALZ	s	0	1	0	1	s7	s6	s5	s4	s3	s2	s1	s0					7	M(SP-1)←PCP, M(SP-2)←PCSH, M(SP-3)←PCSL+1 SP←SP-3, PCP←0, PCS←s7~s0
	RET		1	1	1	1	1	1	0	1	1	1	1	1					7	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2) SP←SP+3
	RETS		1	1	1	1	1	1	0	1	1	1	1	0					12	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2) SP←SP+3, PC←PC+1
	RETD	*l*	0	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					12	PCSL←M(SP), PCSH←M(SP+1), PCP←M(SP+2) SP←SP+3, M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, X←X+2
System control instructions	NOP5		1	1	1	1	1	1	1	1	1	0	1	1					5	No operation (5 clock cycles)
	NOP7		1	1	1	1	1	1	1	1	1	1	1	1					7	No operation (7 clock cycles)
	HALT		1	1	1	1	1	1	1	1	1	0	0	0					5	Halt (stop clock)
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
		XP, r	1	1	1	0	1	0	0	0	0	0	r1	r0					5	XP←r
		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
		YP, r	1	1	1	0	1	0	0	1	0	0	r1	r0					5	YP←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, XP	1	1	1	0	1	0	1	0	0	0	r1	r0					5	r←XP
		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, YP	1	1	1	0	1	0	1	1	0	0	r1	r0					5	r←YP
		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
		XP	1	1	1	1	1	1	0	0	0	1	0	0					5	SP← SP-1, M(SP)←XP
		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
		YP	1	1	1	1	1	1	0	0	0	1	1	1					5	SP← SP-1, M(SP)←YP
		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
		XP	1	1	1	1	1	1	0	1	0	1	0	0					5	XP←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
		YP	1	1	1	1	1	1	0	1	0	1	1	1					5	YP←M(SP), SP←SP+1

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

Abbreviations used in the explanations have the following meanings.

Symbols associated with registers and memory

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

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

Symbols associated with program counter

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

Symbols associated with flags

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

Associated with immediate data

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

Associated with arithmetic and other operations

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

APPENDIX B. E0C62T3 RAM MAP

RAM map - 1 (000H–07FH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM map - 2 (100H–17FH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM map - 3 (200H–27FH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM map - 4 (300H–37FH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
3	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

RAM map - 5 (400H–47FH)

PROGRAM NAME:

P	H L		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	0	NAME																
		MSB																
		LSB																
	1	NAME																
		MSB																
		LSB																
	2	NAME																
		MSB																
		LSB																
	3	NAME																
		MSB																
		LSB																
	4	NAME																
		MSB																
		LSB																
	5	NAME																
		MSB																
		LSB																
	6	NAME																
		MSB																
		LSB																
	7	NAME																
		MSB																
		LSB																

Display memory (80H–AFH), I/O memory (C0H–EBH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	NAME																
		MSB																
1																		
2		LSB																
3	9	NAME																
4		MSB																
		LSB																
	A	NAME																
		MSB																
		LSB																
	C	NAME																–
		MSB	0	0	0	0	IT1	0	0	SIK13	SIK03	0	K13	K03	0	DFK13	DFK03	–
			0	0	0	0	IT2	0	0	SIK12	SIK02	K22	K12	K02	DFK22	DFK12	DFK02	–
			0	0	0	0	IT16	0	SIK21	SIK11	SIK01	K21	K11	K01	DFK21	DFK11	DFK01	–
		LSB	IK22	IK2	IK1	IK0	IT32	ID	SIK20	SIK10	SIK00	K20	K10	K00	DFK20	DFK10	DFK00	–
	D	NAME																
		MSB	EIK22	EIT1	0	R13/HFO	R03	IOC3	PUP3	P03	0	0	TM3	TM7	WDON	BZR11	LDTY1	0
			EIK2	EIT2	0	R12/HDO	R02	IOC2	PUP2	P02	0	0	TM2	TM6	WDRST	BZR10	LDTY0	0
			EIK1	EIT16	0	R11/BZ	R01	IOC1	PUP1	P01	CLKCHG	0	TM1	TM5	WD1	0	0	SVDDT
		LSB	EIK0	EIT32	EID	R10/$\overline{BZ}$	R00	IOC0	PUP0	P00	OSCC	TMRST	TM0	TM4	WD0	BZFQ	LCDON	SVDON
	E	NAME													–	–	–	–
		MSB	TPS	PTS3	FTS3	0	HF	IDP3	0	TCD3	0	0	CHFO	CTO	–	–	–	–
			0	PTS2	FTS2	HOLD	0	IDP2	0	TCD2	0	0	CHDO	0	–	–	–	–
			MB	PTS1	FTS1	PAUSE	0	IDP1	SINR	TCD1	CRMUT	0	0	0	–	–	–	–
		LSB	DRS	PTS0	FTS0	FLASH	0	IDP0	SINC	TCD0	CTMUT	HSON	0	0	–	–	–	–

APPENDIX C. E0C62T3 I/O MEMORY MAP

I/O memory map (C0H–CEH)

Address	Register								Comment
*7	D3	D2	D1	D0	Name	Init *1	1	0	
C0H	0	0	0	IK22	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK22 *6	0	Yes	No	Interrupt factor flag (K22)
C1H	0	0	0	IK2	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK2 *6	0	Yes	No	Interrupt factor flag (K20, K21)
C2H	0	0	0	IK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK1 *6	0	Yes	No	Interrupt factor flag (K10–K13)
C3H	0	0	0	IK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK0 *6	0	Yes	No	Interrupt factor flag (K00–K03)
C4H	IT1	IT2	IT16	IT32	IT1 *6	0	Yes	No	Interrupt factor flag (Clock timer 1 Hz)
					IT2 *6	0	Yes	No	Interrupt factor flag (Clock timer 2 Hz)
	R				IT16 *6	0	Yes	No	Interrupt factor flag (Clock timer 16 Hz)
					IT32 *6	0	Yes	No	Interrupt factor flag (Clock timer 32 Hz)
C5H	0	0	0	ID	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					ID *6	0	Yes	No	Interrupt factor flag (dialing)
C6H	0	0	SIK21	SIK20	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		SIK21	0	Enable	Disable	Interrupt selection register (K21)
					SIK20	0	Enable	Disable	Interrupt selection register (K20)
C7H	SIK13	SIK12	SIK11	SIK10	SIK13	0	Enable	Disable	Interrupt selection register (K10–K13)
					SIK12	0	Enable	Disable	
	R/W				SIK11	0	Enable	Disable	
					SIK10	0	Enable	Disable	
C8H	SIK03	SIK02	SIK01	SIK00	SIK03	0	Enable	Disable	Interrupt selection register (K00–K03)
					SIK02	0	Enable	Disable	
	R/W				SIK01	0	Enable	Disable	
					SIK00	0	Enable	Disable	
C9H	0	K22	K21	K20	0 *5	– *2			Unused
					K22	– *2	High	Low	Input port (K20–K22)
	R				K21	– *2	High	Low	
					K20	– *2	High	Low	
CAH	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	
CBH	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	
CCH	0	DFK22	DFK21	DFK20	0 *5	– *2			Unused
					DFK22	1	↓	↑	Input comparison register (K20–K22)
	R	R/W			DFK21	1	↓	↑	
					DFK20	1	↓	↑	
CDH	DFK13	DFK12	DFK11	DFK10	DFK13	1	↓	↑	Input comparison register (K10–K13)
					DFK12	1	↓	↑	
	R/W				DFK11	1	↓	↑	
					DFK10	1	↓	↑	
CEH	DFK03	DFK02	DFK01	DFK00	DFK03	1	↓	↑	Input comparison register (K00–K03)
					DFK02	1	↓	↑	
	R/W				DFK01	1	↓	↑	
					DFK00	1	↓	↑	

Remarks
*1 Initial value at the time of initial reset
*2 Not set in the circuit
*3 Undefined
*4 Inhibit state (output port will be set to "1")
*5 Constantly "0" when being read
*6 Reset (0) immediately after being read
*7 Page switching in I/O memory is not necessary

I/O memory map (D0H–DDH)

Address *7	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
D0H	EIK22	EIK2	EIK1	EIK0	EIK22	0	Enable	Mask	Interrupt mask register (K22)
					EIK2	0	Enable	Mask	Interrupt mask register (K20, K21)
	R/W				EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
D1H	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)
D2H	0	0	0	EID	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EID	0	Enable	Mask	Interrupt mask register (dialing)
D3H	R13	R12	R11	R10	R13	0	High	Low	Output port (R13)
					HFO		– *4	On	Handfree output (HFO)
	HFO	HDO	BZ	$\overline{BZ}$	R12	0	High	Low	Output port (R12)
					HDO		– *4	On	Hold-line output (HDO)
	R/W				R11	0	High	Low	Output port (R11)
					BZ		– *4	On	Buzzer output (BZ)
					R10	0	High	Low	Output port (R10)
					$\overline{BZ}$		– *4	On	Buzzer inverted output ($\overline{BZ}$)
D4H	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	
D5H	IOC3	IOC2	IOC1	IOC0	IOC3	0	Output	Input	I/O control register
					IOC2	0	Output	Input	
	R/W				IOC1	0	Output	Input	
					IOC0	0	Output	Input	
D6H	PUP3	PUP2	PUP1	PUP0	PUP3	0	On	Off	Pull up control register
					PUP2	0	On	Off	
	R/W				PUP1	0	On	Off	
					PUP0	0	On	Off	
D7H	P03	P02	P01	P00	P03	1	High	Low	I/O port
					P02	1	High	Low	
	R/W				P01	1	High	Low	
					P00	1	High	Low	
D8H	0	0	CLKCHG	OSCC	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		CLKCHG	0	OSC3	OSC1	CPU system clock switch
					OSCC	0	On	Off	OSC3 oscillation On/Off
D9H	0	0	0	TMRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			W	0 *5	– *2			Unused
					TMRST *5	– *2	Reset	Invalid	Clock timer reset
DAH	TM3	TM2	TM1	TM0	TM3	– *3			Clock timer low-order data (16 Hz)
					TM2	– *3			Clock timer low-order data (32 Hz)
	R				TM1	– *3			Clock timer low-order data (64 Hz)
					TM0	– *3			Clock timer low-order data (128 Hz)
DBH	TM7	TM6	TM5	TM4	TM7	– *3			Clock timer high-order data (1 Hz)
					TM6	– *3			Clock timer high-order data (2 Hz)
	R				TM5	– *3			Clock timer high-order data (4 Hz)
					TM4	– *3			Clock timer high-order data (8 Hz)
DCH	WDON	WDRST	WD1	WD0	WDON	0	On	Off	Watchdog timer On/Off
					WDRST *5	Reset	Reset	Invalid	Watchdog timer reset
	R/W	W	R		WD1	0			Watchdog timer data (1/4 Hz)
					WD0	0			Watchdog timer data (1/2 Hz)
DDH	BZR11	BZR10	0	BZFQ	BZR11	0	Buzzer	DC	R11 port output selection
					BZR10	0	Buzzer (inverted)	DC	R10 port output selection
	R/W		R	R/W	0 *5	– *2			Unused
					BZFQ	0	2 kHz	4 kHz	Buzzer frequency selection

I/O memory map (DEH–EBH)

Address	Register								Comment
*7	D3	D2	D1	D0	Name	Init *1	1	0	
DEH	LDTY1	LDTY0	0	LCDON	LDTY1	0			LCD drive duty selection
					LDTY0	0			0: 1/4, 1: 1/3, 2: 1/2, 3: 1/1
	R/W		R	R/W	0 *5	– *2			Unused
					LCDON	0	On	Off	LCD display control (LCD display all off)
DFH	0	0	SVDDT	SVDON	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	SVDDT	0	Low	Normal	Supply voltage detector data
					SVDON	0	On	Off	SVD circuit On/Off
E0H	TPS	0	MB	DRS	TPS	0	Pulse	Tone	Tone/pulse mode selection
					0 *5	– *2			Unused
	R/W	R	R/W		MB	0	33.3:66.6	40:60	Make : Break ratio selection
					DRS	0	20 pps	10 pps	Dialing pulse rate selection
E1H	PTS3	PTS2	PTS1	PTS0	PTS3	0			Pulse time selection (sec)
					PTS2	1			0: ×,
	R/W				PTS1	0			1–F: 1–15 (default: 4)
					PTS0	0			
E2H	FTS3	FTS2	FTS1	FTS0	FTS3	0			Flash time selection (msec)
					FTS2	1			0: ×, 1: 94, 2: 188, 3: 281, 4: 375, 5: 469, 6: 563
	R/W				FTS1	1			7: 656, 8: 750, 9: 844, A: 938, B: 1031, C: 1125
					FTS0	0			D: 1219, E: 1313, F: 1406 (default: 563)
E3H	0	HOLD	PAUSE	FLASH	0 *5	– *2			Unused
					HOLD	0	On	Off	Hold-line function
	R	R/W	W		PAUSE	0	Yes	No	Pause function
					FLASH	0	Yes	No	Flash function
E4H	HF	0	0	0	HF	0	Yes	No	Hand free
					0 *5	– *2			Unused
	R/W	R			0 *5	– *2			Unused
					0 *5	– *2			Unused
E5H	IDP3	IDP2	IDP1	IDP0	IDP3	1			Inter-digit pause selection for dial pulse (msec)
					IDP2	0			0: ×, 1: 94, 2: 188, 3: 281, 4: 375, 5: 469, 6: 563
	R/W				IDP1	0			7: 656, 8: 750, 9: 844, A: 938, B: 1031, C: 1125
					IDP0	0			D: 1219, E: 1313, F: 1406 (default: 750)
E6H	0	0	SINR	SINC	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		SINR	1	Enable	Disable	DTMF column frequency output enable
					SINC	1	Enable	Disable	DTMF row frequency output enable
E7H	TCD3	TCD2	TCD1	TCD0	TCD3	0			Telephone code for dialing TCD DTMF DP TCD DTMF DP 0: (R1C4) × 8: (R3C2) 8 1: (R1C1) 1 9: (R3C3) 9 2: (R1C2) 2 A: (R4C2) 10 3: (R1C3) 3 B: (R4C3) 11 4: (R2C1) 4 C: (R4C1) 12 5: (R2C2) 5 D: (R2C4) 13 6: (R2C3) 6 E: (R4C4) 14 7: (R3C1) 7 F: (R3C4) 15
					TCD2	0			
	R/W				TCD1	0			
					TCD0	0			
E8H	0	0	CRMUT	CTMUT	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		CRMUT	1	Output	0	Receive mute control
					CTMUT	1	Output	0	Transmit mute control
E9H	0	0	0	HSON	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					HSON	0	Hook off	Hook on	Hook switch On/Off
EAH	CHFO	CHDO	0	0	CHFO	0	Handfree	DC	R13 output selection (R13 register has to be "0")
					CHDO	0	Hold	DC	R12 output selection (R12 register has to be "0")
	R/W		R		0 *5	– *2			Unused
					0 *5	– *2			Unused
EBH	CTO	0	0	0	CTO	0	On	Off	Tone duration time control
					0 *5	– *2			(continuous tone output On/Off)
	R/W	R			0 *5	– *2			
					0 *5	– *2			

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 ICS62T3J.EXE PC-DOS ICS62T3W.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 EVA62T3 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 ICS62T3P.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.
SOG62T3	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
ASM62T3	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.
MDC62T3	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?
EVA62T3	The EVA62T3 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 EVA62T3 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 EVA62T3 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