MF800-03

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C621C**

DEVELOPMENT TOOL MANUAL

ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

NOTICE

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

MS-DOS and EDLIN are registered trademarks of Microsoft Corporation, U.S.A.
PC-DOS, PC/AT, VGA, EGA and IBM are registered trademarks of International Business Machines Corporation, U.S.A.
NEC PC-9801 Series and NEC are registered trademarks of NEC Corporation.
All other product names mentioned herein are trademarks and/or registered trademarks of their respective owners.

E0C621C Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C621C.

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 EVA621C Manual ICE6200 Hardware Manual
Development procedure	☞ E0C62 Family Technical Guide
Device (E0C621C)	☞ E0C621C 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 DEV621C

The below software are included in the product of the E0C621C development support tool DEV621C.

1. Development Tool Management System DMS6200 ..... Menu selection for each software / start-up software
2. Cross Assembler ASM621C .... Cross assembler for program preparation
3. Function Option Generator FOG621C .... Function option data preparation program
4. Segment Option Generator SOG621C .... Segment option data preparation program
5. ICE Control Software ICS621C .... ICE control program
6. Mask Data Checker MDC621C .... Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV621C 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 E0C621C, 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.

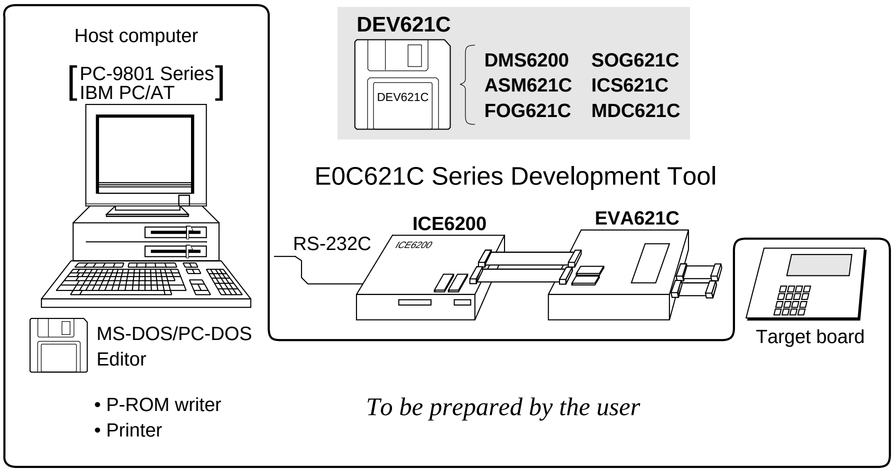


Fig. 1.2.1 System configuration

The EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data generated from the FOG621C. Therefore, if correct function option HEX data for the E0C621C is not loaded in the EVA621C, the EVA621C does not identify as E0C621C functions and does not operate correctly. Especially when the EVA621C is used for both E0C621C and E0C6292, be aware of the function option HEX data to be loaded.

Note *The DEV621C 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 DEV621C.

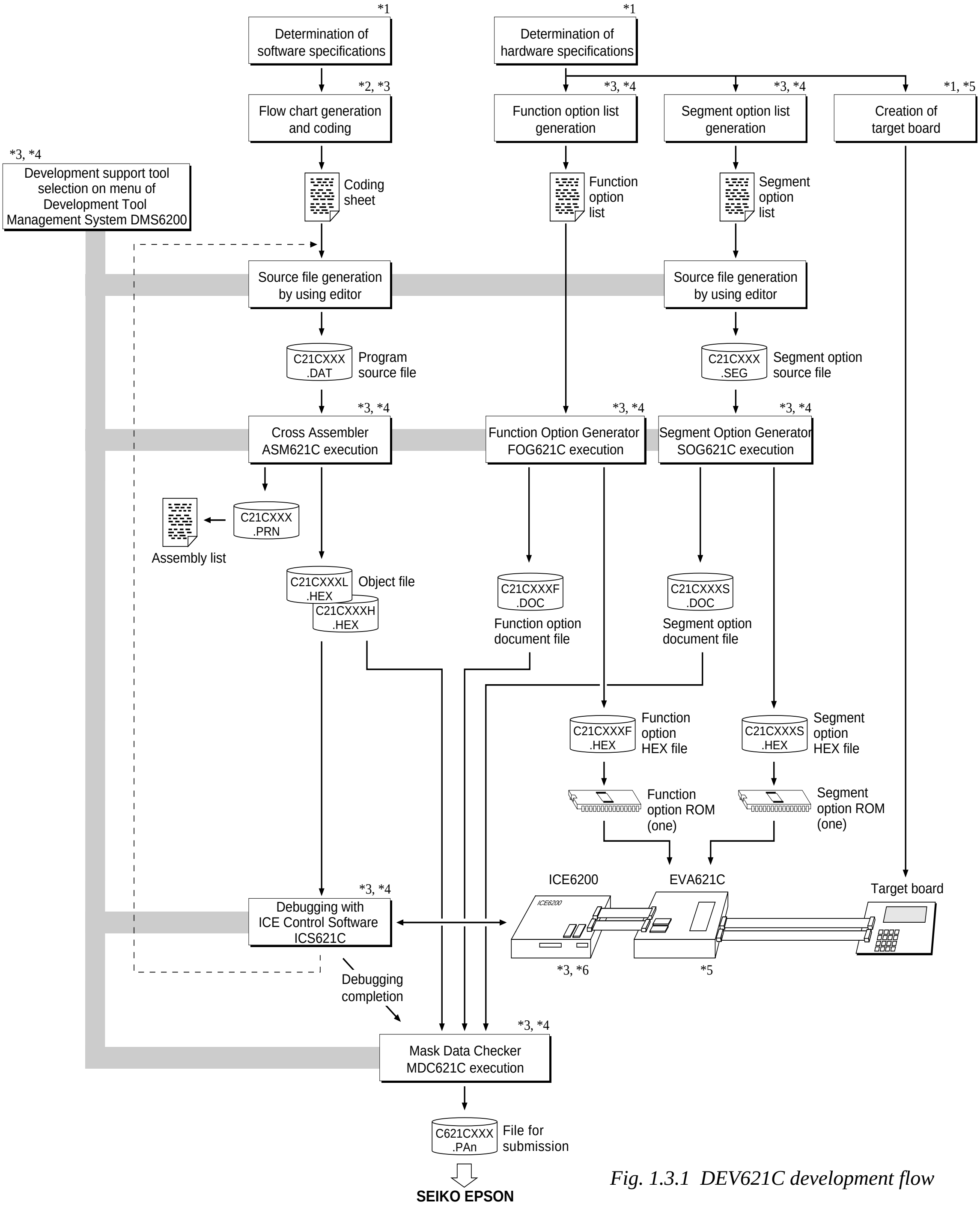

*1
Determination of software specifications
*1
Determination of hardware specifications
*2, *3
Flow chart generation and coding
*3, *4
Function option list generation
*3, *4
Segment option list generation
*1, *5
Creation of target board
*3, *4
Development support tool selection on menu of Development Tool Management System DMS6200
Coding sheet
Function option list
Segment option list
Source file generation by using editor
Source file generation by using editor
C21CXXX .DAT
Program source file
C21CXXX .SEG
Segment option source file
*3, *4
Cross Assembler ASM621C execution
*3, *4
Function Option Generator FOG621C execution
*3, *4
Segment Option Generator SOG621C execution
C21CXXX .PRN
Assembly list
C21CXXXL .HEX
C21CXXXH .HEX
Object file
C21CXXXF .DOC
Function option document file
C21CXXXS .DOC
Segment option document file
C21CXXXF .HEX
Function option HEX file
C21CXXXS .HEX
Segment option HEX file
Function option ROM (one)
Segment option ROM (one)
ICE6200
EVA621C
Target board
ICE6200
*3, *6
*5
*3, *4
Debugging with ICE Control Software ICS621C
Debugging completion
*3, *4
Mask Data Checker MDC621C execution
C621CXXX .PAn
File for submission
SEIKO EPSON

Fig. 1.3.1 DEV621C development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C21CXXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1	E0C621C Technical Manual	*4	E0C621C Development Tool Manual (this manual)
*2	E0C6200/6200A Core CPU Manual	*5	EVA621C Manual
*3	E0C62 Family Development Tool Reference Manual	*6	ICE6200 Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 are recorded in the DEV621C floppy disk.
The content of the files contained in the DEV621C floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM621C.EXE	ASM621C.EXE	Cross Assembler execution file
DMS6200.EXE	DMS6200.EXE	Development Tool Management System execution file
FOG621C.EXE	FOG621C.EXE	Function Option Generator execution file
ICS621CB.BAT	ICS621C.BAT	ICE Control Software batch file
ICS621CW.EXE	ICS621CJ.EXE	ICE Control Software execution file
ICS621CP.PAR	ICS621CP.PAR	ICE Control Software parameter file
MDC621C.EXE	MDC621C.EXE	Mask Data Checker execution file
SOG621C.EXE	SOG621C.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 (DEV621C, 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 (DEV621C), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV621C↵
C\>CD DEV621C↵
C\DEV621C\>COPY A:*.*↵
```

■ Next make a CONFIG.SYS file using Editor or the like.
When a CONFIG.SYS has previously been made using a hard disk system, check the setting of the FILES within it. (If there is none add it.)
Set the number of files to be described in CONFIG.SYS at 10 or more, so that the Mask Data Checker MDC621C will handle many files.

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

Example:

Setting of FILES (CONFIG.SYS)

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

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

■ In "ICS621C(B).BAT" the batch process is indicated such that the ICS621CJ(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters. When first executing the ICE Control Software after resetting the host computer, select then activate this batch file from the DMS6200 menu.
The SPEED (MS-DOS) or MODE (PC-DOS) command is used for setting the RS-232C, so you should copy these commands into the disk to be executed or specify a disk or directory with the command using the PATH command.

RS-232C Setting (PC-DOS version)

```
MODE  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 DEV621C software development support tool and the program such as an editor in menu form and starts it.
In this way the various software frequently executed during debugging can be effectively activated.

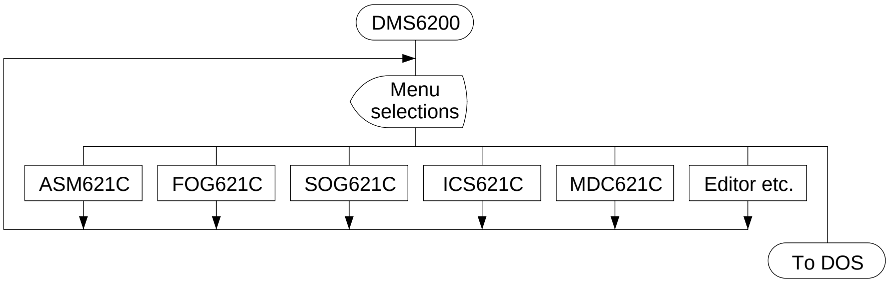


Fig. 2.1.1 DMS6200 execution flow

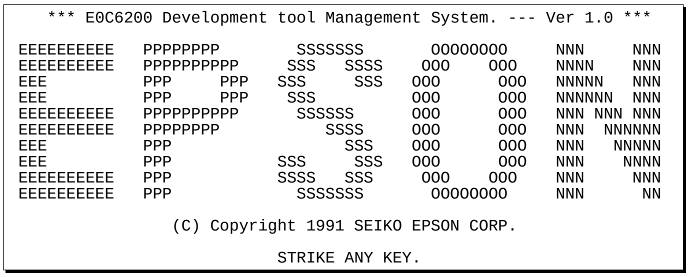
Refer to the "E0C62 Family Development Tool Reference Manual" for detailes of the operation.

2.2 DMS6200 Quick Reference

■ Starting command

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

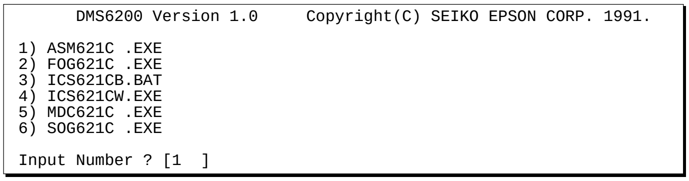
■ Display examples



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

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

                (C) Copyright 1991 SEIKO EPSON CORP.

                           STRIKE ANY KEY.
```

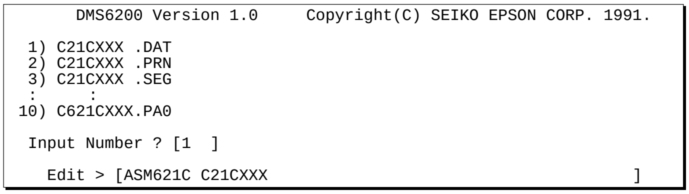
Start message
When DMS6200 is started, the following message is displayed. For "STRIKE ANY KEY.", press any key to advance the program execution.

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) ASM621C .EXE
2) FOG621C .EXE
3) ICS621CB.BAT
4) ICS621CW.EXE
5) MDC621C .EXE
6) SOG621C .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) C21CXXX .DAT
 2) C21CXXX .PRN
 3) C21CXXX .SEG
 :     :
10) C621CXXX.PA0

 Input Number ? [1  ]

   Edit > [ASM621C C21CXXX                                      ]
```

Source file selection screen
To starting ASM621C, 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 ASM621C

3.1 ASM621C Outline

The ASM621C cross assembler is an assembler program for generating the machine code used by the E0C621C 4-bit, single-chip microcomputers. The Cross Assembler ASM621C will assemble the program source files which have been input by the user's editor and will generate an object file in Intel-Hex format and assembly list file.
In this assembler, program modularization has been made possible through macro definition functions and programming independent of the ROM page structure has been made possible through the auto page set function. In addition, consideration has also been given to precise error checks for program capacity (ROM capacity) overflows, undefined codes and the like, and for debugging of such things as label tables for assembly list files and cross reference table supplements.

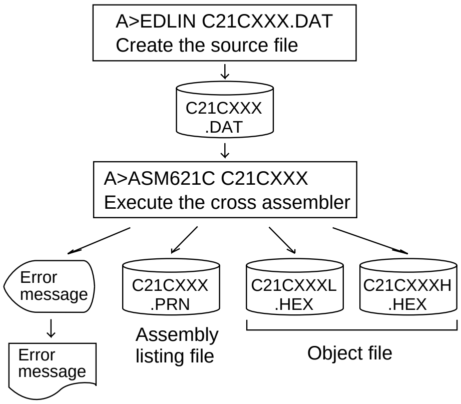


Fig. 3.1.1 ASM621C execution flow

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

3.2 E0C621C Restrictions

Note the following when generating a program by the E0C621C:

■ ROM area

The capacity of the E0C621C ROM is 4K steps (0000H to 0FFFH).
Therefore, the specification range of the memory setting pseudo-instructions and PSET instruction is restricted.

Memory configuration:

Bank: Only bank 0, Page: 16 pages (0 to 0FH), each 256 steps

Significant specification range:

ORG	pseudo-instruction:	0000H to 0FFFH
PAGE	pseudo-instruction:	00H to 0FH
BANK	pseudo-instruction:	Only 0H
PSET	instruction:	00H to 0FH

■ RAM area

The capacity of the E0C621C RAM is 297 words (000H to 4FFH, 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.
- 82H–8FH, 93H, 97H–9FH, A1H–AFH, B3H–BFH, C2H–C3H, C7H, CBH–CFH, D7H–DFH, E4H–EFH, F5H–F7H and FDH–FFH in page 0 to 4
- 150H–17FH, 200H–27FH, 300H–37FH, 400H–44FH

(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,1
LD   XP,A
LD   X,50H
```

150H is loaded into the IX register, but an unused area has been specified so that the memory accessible with the IX register (MX) is invalid.

■ Undefined codes

The SLP instruction has not been defined in the E0C621C instruction sets.

3.3 ASM621C Quick Reference

■ Starting command and input/output files

_ indicates a blank.

↵ indicates the Return key.

A parameter enclosed by [ ] can be omitted.

Execution file: ASM621C.EXE

Starting command: **ASM621C_ [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)

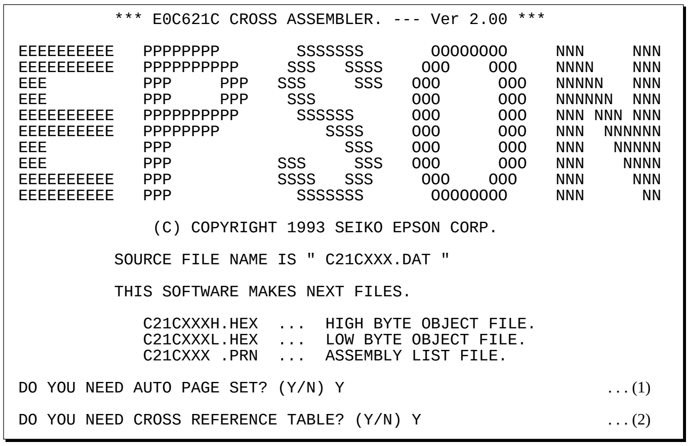
p Specifies the drive to which the assembly listing file is to be output. (A–P, @, Z)
@: Current drive, Z: File is not generated

-N The code (FFH) in the undefined area of program memory is not created.

Input file: C21CXXX.DAT (Source file)

Output file:
C21CXXXL.HEX (Object file, low-order)
C21CXXXH.HEX (Object file, high-order)
C21CXXX.PRN (Assembly listing file)

■ Display example



```
          *** E0C621C 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 1993 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C21CXXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C21CXXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C21CXXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C21CXXX .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 ASM621C 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, ASM621C 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 FOG621C

4.1 FOG621C Outline

With the 4-bit single-chip E0C621C microcomputers, the customer may select five hardware options. By modifying the mask patterns of the E0C621C according to the selected options, the system can be customized to meet the specifications of the target system.
The Function Option Generator FOG621C 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 FOG621C, the E0C621C mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA621C) hardware option ROM is simultaneously generated with the data file.

(Be aware that the EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data.)

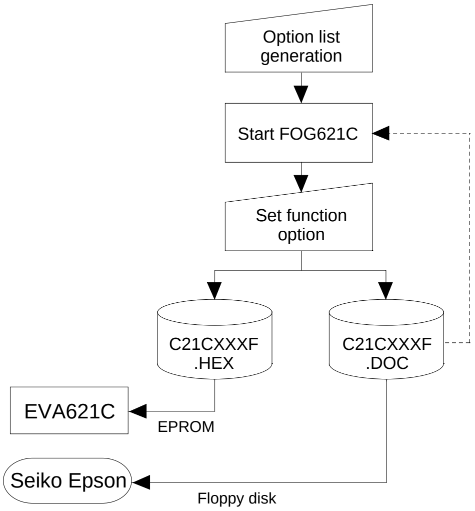


Fig. 4.1.1 FOG621C execution flow

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

4.2 E0C621C 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 OSCILLATOR

□ 1. Ceramic
□ 2. CR

2. 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

3. OUTPUT PORT OUTPUT SPECIFICATION

- R00 .... □ 1. Complementary □ 2. Nch Open Drain
- R01 .... □ 1. Complementary □ 2. Nch Open Drain
- R02 .... □ 1. Complementary □ 2. Nch Open Drain
- R03 .... □ 1. Complementary □ 2. Nch Open Drain

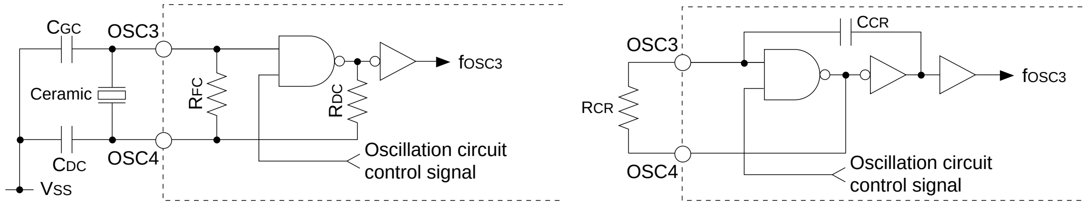
4.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C621C are shown below, and their specifications are also described.

1 OSC3 oscillator

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

--- OSC3 OSCILLATOR ---
                       1. CERAMIC
                       2. CR

PLEASE SELECT NO.(1) ? 1↵

                       1. CERAMIC  SELECTED
```

Select the OSC3 system clock to be used for the E0C621C. The OSC3 oscillation circuit generates the system clock when operating the CPU in high-speed. Either ceramic oscillation or CR oscillation can be selected for the oscillation circuit that uses the OSC3 and OSC4 terminals. When ceramic oscillation is selected, ceramic oscillator, gate capacitor and drain capacitor should be connected between the OSC3 and the OSC4 terminals. When CR oscillation is selected, resistor only need be connected between the terminals.
The configuration of the OSC3 oscillation circuit is shown in Figure 4.3.1.


CGC
OSC3
Ceramic
RFC
RDC
fOSC3
CDC
OSC4
VSS
Oscillation circuit control signal
CCR
OSC3
RCR
fOSC3
OSC4
Oscillation circuit control signal

(1) Ceramic oscillation circuit

(2) CR oscillation circuit

Fig. 4.3.1 OSC3 oscillation circuit

2 Input port pull-up resistor

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

--- 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↵

        K02            1. WITH RESISTOR
                       2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        K03            1. WITH RESISTOR
                       2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        K10            1. WITH RESISTOR
                       2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        K11            1. WITH RESISTOR
                       2. GATE DIRECT
```

Select whether each input port (K00–K03 and K10–K13) will be supplemented with a pull-up resistor or not.
When "GATE DIRECT" is selected, see to it that an entry floating state does not occur.
Moreover, when "WITH RESISTOR" has been selected and when input ports are changed from low level (V_{SS}) to high level (V_{DD}) with pull-up resistor, the rise of the waveform is delayed on account of the time constant of the pull-up resistor and input gate capacitance. Hence, when fetching input ports, be sure to wait an appropriate amount of time with the program.
For unused ports, select "WITH RESISTOR".

```
PLEASE SELECT NO.(1) ? 1↵

        K12            1. WITH RESISTOR
                       2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        K13            1. WITH RESISTOR
                       2. GATE DIRECT

PLEASE SELECT NO.(1) ? 1↵

        K00            1. WITH RESISTOR  SELECTED
        K01            1. WITH RESISTOR  SELECTED
        K02            1. WITH RESISTOR  SELECTED
        K03            1. WITH RESISTOR  SELECTED
        K10            1. WITH RESISTOR  SELECTED
        K11            1. WITH RESISTOR  SELECTED
        K12            1. WITH RESISTOR  SELECTED
        K13            1. WITH RESISTOR  SELECTED
```

The configuration of the pull-up resistor circuit is shown in Figure 4.3.2.

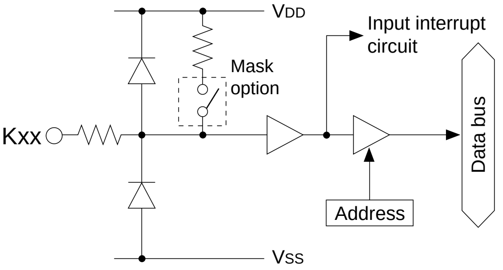


Fig. 4.3.2 Configuration of pull up resistor circuit

3 Output port output specification

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

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

PLEASE SELECT NO.(1) ? 1↵

        R01            1. COMPLEMENTARY
                       2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R02            1. COMPLEMENTARY
                       2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

        R03            1. COMPLEMENTARY
                       2. NCH OPEN DRAIN

PLEASE SELECT NO.(1) ? 1↵

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

Select the output specification for the output ports (R00–R03).

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

When the output port is used as a common output for key matrix, select Nch open drain output.

For unused output ports, select complementary output.

The output circuit configuration is shown in Figure 4.3.3.

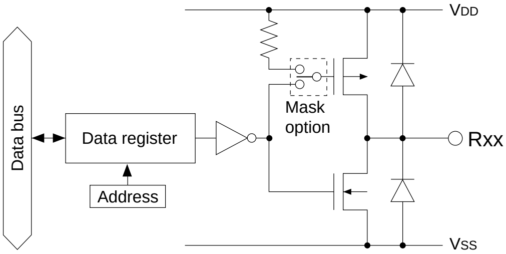


Fig. 4.3.3 Configuration of output circuit

4.4 FOG621C Quick Reference

■ Starting command and input/output files

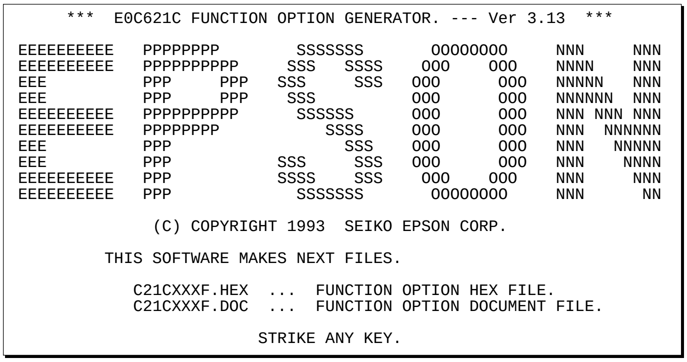
Execution file: FOG621C.EXE

Starting command: **FOG621C↵** ↵ *indicates the Return key.*

Input file: C21CXXXF.DOC (Function option document file, when modifying)

Output file: C21CXXXF.DOC (Function option document file)
C21CXXXF.HEX (Function option HEX file)

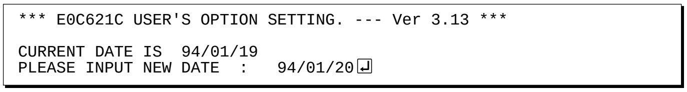
■ Display example



```
     ***  E0C621C 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.

            C21CXXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C21CXXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

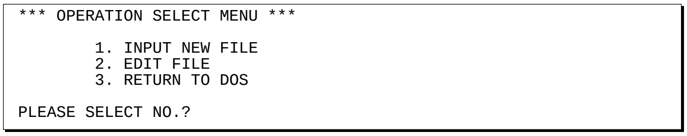
Start-up message
When FOG621C is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

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



```
*** E0C621C USER'S OPTION SETTING. --- Ver 3.13 ***

CURRENT DATE IS  94/01/19
PLEASE INPUT NEW DATE  :   94/01/20↵
```

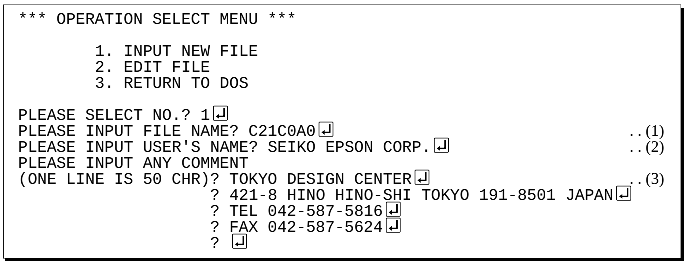
Date input
Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/"). When not modifying the date, press the RETURN key "↵" to continue.



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

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

PLEASE SELECT NO.?
```

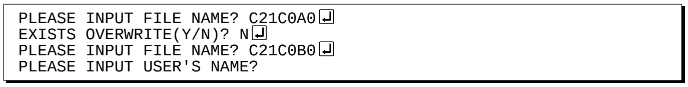
Operation selection menu
Enter a number from 1 to 3 to select a subsequent operation.

1. To set new function options.
2. To modify the document file.
3. To terminate FOG621C.



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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C21C0A0↵                                ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                    ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                      ..(3)
                    ? 421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN↵
                    ? TEL 042-587-5816↵
                    ? FAX 042-587-5624↵
                    ? ↵
```

Setting new function options
Select "1" on the operation selection menu.
(1) Enter the file name.
(2) Enter the customer's company name.
(3) Enter any comment.
(Within 50 characters x 10 lines)
Next, start function option setting from option No. 1.



```
PLEASE INPUT FILE NAME? C21C0A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C21C0B0↵
PLEASE INPUT USER'S NAME?
```

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

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) ***

C21C0A0         C21C0B0         C21C0C0               ..(1)

PLEASE INPUT FILE NAME? C21C0A0↵                      ..(2)
PLEASE INPUT USER'S NAME? ↵                           ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ↵                               ..(4)
PLEASE INPUT EDIT NO.? 1↵                             ..(5)
            :
(Modifying function option settings)
            :
PLEASE INPUT EDIT NO.? E↵
```

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

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

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

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

```
PLEASE INPUT FILE NAME? C21C0N0↵
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 FOG621C 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 ***

--- OSC3 OSCILLATOR ---
                         1. CERAMIC
                         2. CR

PLEASE SELECT  NO.(1) ? 2↵

                         2. CR  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 EVA621C, 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 EVA621C 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

```
* E0C621C FUNCTION OPTION DOCUMENT  V 3.13
*
* FILE NAME    C21CTS1F.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   1994/01/19
*
*
*
* OPTION NO.1
* < OSC3 OSCILLATOR >
*                            CERAMIC  --------------  SELECTED
 OPT0101 01
*
* OPTION NO.2
* < 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
 OPT0201 01
 OPT0202 01
 OPT0203 01
 OPT0204 01
 OPT0205 01
 OPT0206 01
 OPT0207 01
 OPT0208 01
*
* OPTION NO.3
* < OUTPUT PORT OUTPUT SPECIFICATION >
*     R00                    COMPLEMENTARY  --------  SELECTED
*     R01                    COMPLEMENTARY  --------  SELECTED
*     R02                    COMPLEMENTARY  --------  SELECTED
*     R03                    COMPLEMENTARY  --------  SELECTED
 OPT0301 01
 OPT0302 01
 OPT0303 01
 OPT0304 01
*
*
*
* SEIKO EPSON’S AREA
*
*
*
* OPTION NO.4
 OPT0401 01
*
* OPTION NO.5
 OPT0501 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 SOG621C

5.1 SOG621C Outline

With the 4-bit single-chip E0C621C microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C621C according to the selected options, the system can be customized to meet the specifications of the target system.
The Segment Option Generator SOG621C is a software tool for generating data file used to generate mask patterns. From the data file created with SOG621C, the E0C621C mask pattern is automatically generated by a general purpose computer.
The HEX file for the evaluation board (EVA621C) segment option ROM is simultaneously generated with the data file.

(Be aware that the EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data.)

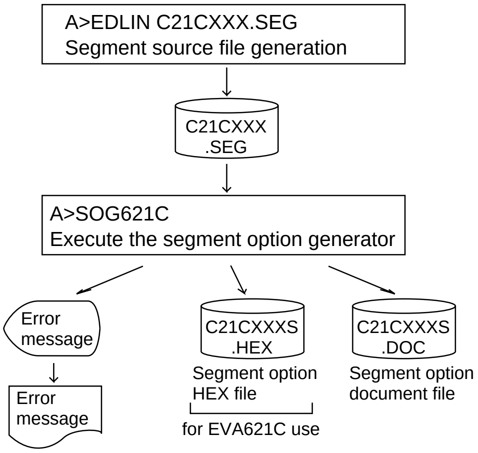


Fig. 5.1.1 SOG621C execution flow

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

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
SEG32													SEG output
SEG33													DC output □ C □ N

Legend: <ADDRESS>
H: High order address (5–7)
L: Low order address (0–F)
D: Data bit (0–3)

<OUTPUT SPECIFICATION>
C: Complementary output
N: Nch open drain output

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

5.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG33, 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 (50H–7FH) can be allocated to the optional segment. With this, up to 136 segments (102, 68 segments when 1/3 or 1/2 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 (5–7), 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  500  501  502  503  S
1  510  511  512  513  S
```

- When 1/3 duty is selected

```
0  500  501  502  ---  S
1  510  511  512  ---  S
```

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 34 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  6C0  ---  ---  ---  C
29  6D0  ---  ---  ---  C
30  6E0  ---  ---  ---  N
31  6F0  ---  ---  ---  N
```

5.4 SOG621C Quick Reference

■ Starting command and input/output files

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

Execution file: SOG621C.EXE

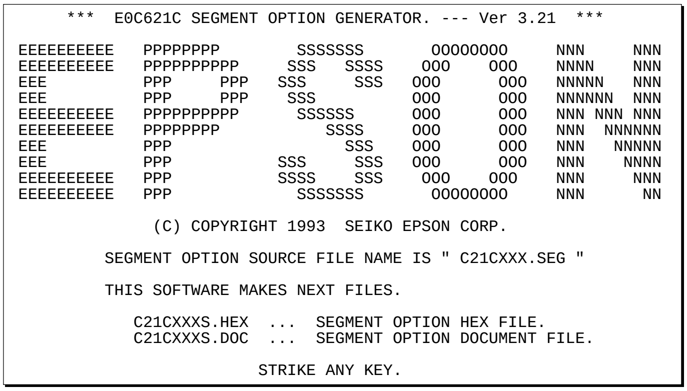
Starting command: **SOG621C_ [-H]↵**

Option: -H: Specifies the segment option document file for input file of SOG621C.

Input file: C21CXXX.SEG (Segment option source file)
C21CXXXS.DOC (Segment option document file, when -H option use)

Output file: C21CXXXS.DOC (Segment option document file)
C21CXXXS.HEX (Segment option HEX file)

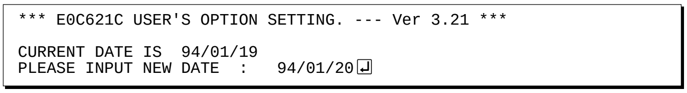
■ Display example



```
     ***  E0C621C 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 " C21CXXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C21CXXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C21CXXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

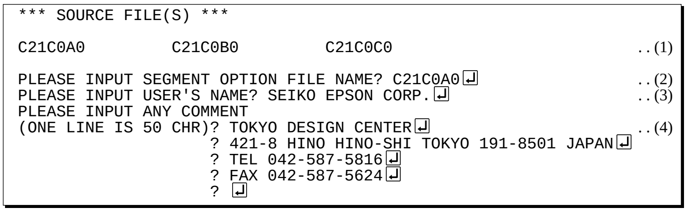
Start-up message
When SOG621C is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

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

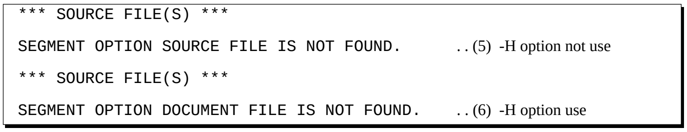


```
*** E0C621C USER'S OPTION SETTING. --- Ver 3.21 ***

CURRENT DATE IS  94/01/19
PLEASE INPUT NEW DATE  :   94/01/20↵
```

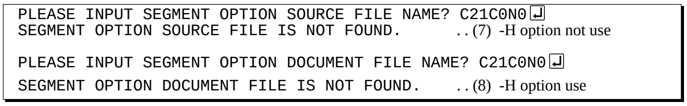
Date input
Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/"). When not modifying the date, press the RETURN key "↵" to continue.



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

C21C0A0         C21C0B0         C21C0C0                         ..(1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C21C0A0↵                 ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                    ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                      ..(4)
                    ? 421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN↵
                    ? TEL 042-587-5816↵
                    ? FAX 042-587-5624↵
                    ? ↵
```



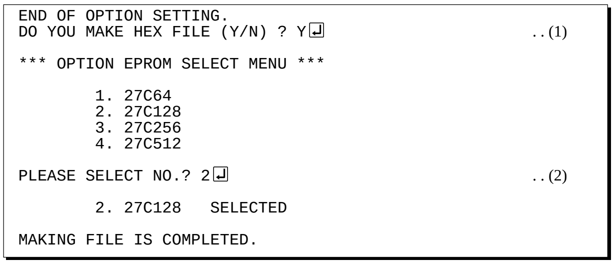
```
*** SOURCE FILE(S) ***

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? C21C0N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.      ..(7) -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C21C0N0↵
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 EVA621C, 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 EVA621C options.

When a series of operations are complete, the SOG621C generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG621C 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

```
;C21CTS1.SEG, VER3.21
;EVA621C LCD SEGMENT DECODE TABLE
;
   0  500  501  502  503  S
   1  510  511  512  513  S
   2  520  521  522  523  S
   3  530  531  532  533  S
   4  540  541  542  543  S
   5  550  551  552  553  S
   6  560  561  562  563  S
   7  570  571  572  573  S
   8  580  581  582  583  S
   9  590  591  592  593  S
  10  5A0  5A1  5A2  5A3  S
  11  5B0  5B1  5B2  5B3  S
  12  5C0  5C1  5C2  5C3  S
  13  5D0  5D1  5D2  5D3  S
  14  5E0  5E1  5E2  5E3  S
  15  5F0  5F1  5F2  5F3  S
  16  600  601  602  603  S
  17  610  611  612  613  S
  18  620  621  622  623  S
  19  630  631  632  633  S
  20  640  641  642  643  S
  21  650  651  652  653  S
  22  660  661  662  663  S
  23  670  671  672  673  S
  24  680  681  682  683  S
  25  690  691  692  693  S
  26  6A0  6A1  6A2  6A3  S
  27  6B0  6B1  6B2  6B3  S
  28  6C0  6C1  6C2  6C3  S
  29  6D0  6D1  6D2  6D3  S
  30  6E0  6E1  6E2  6E3  S
  31  6F0  6F1  6F2  6F3  S
  32  700  701  702  703  S
  33  710  711  712  713  S
```

■ Example of segment option document file

```
* E0C621C SEGMENT OPTION DOCUMENT  V 3.21
*
* FILE NAME    C21CTS1S.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   94/01/19
*
*
* OPTION NO.6
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
    0  500  501  502  503  S
    1  510  511  512  513  S
    2  520  521  522  523  S
    3  530  531  532  533  S
    4  540  541  542  543  S
    5  550  551  552  553  S
    6  560  561  562  563  S
    7  570  571  572  573  S
    8  580  581  582  583  S
    9  590  591  592  593  S
   10  5A0  5A1  5A2  5A3  S
   11  5B0  5B1  5B2  5B3  S
   12  5C0  5C1  5C2  5C3  S
   13  5D0  5D1  5D2  5D3  S
   14  5E0  5E1  5E2  5E3  S
   15  5F0  5F1  5F2  5F3  S
   16  600  601  602  603  S
   17  610  611  612  613  S
   18  620  621  622  623  S
   19  630  631  632  633  S
   20  640  641  642  643  S
   21  650  651  652  653  S
   22  660  661  662  663  S
   23  670  671  672  673  S
   24  680  681  682  683  S
   25  690  691  692  693  S
   26  6A0  6A1  6A2  6A3  S
   27  6B0  6B1  6B2  6B3  S
   28  6C0  6C1  6C2  6C3  S
   29  6D0  6D1  6D2  6D3  S
   30  6E0  6E1  6E2  6E3  S
   31  6F0  6F1  6F2  6F3  S
   32  700  701  702  703  S
   33  710  711  712  713  S
\\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 ICS621C

6.1 ICS621C Outline

The In-circuit Emulator ICE6200 connects the target board produced by the user via the EVA621C 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 ICS621C.
The ICS621C 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.

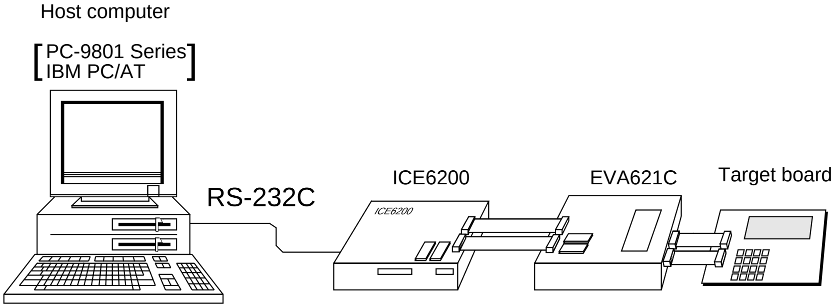


Fig. 6.1.1 Debugging system using ICE6200

(Be aware that the EVA621C can be used as the EVA board for the E0C6292 by modifying the function option HEX data.)

☞ *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 ICS621C Restrictions

Take the following precautions when using the ICS621C.

■ ROM Area

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

■ RAM Area

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

- 82H–8FH, 93H, 97H–9FH, A1H–AFH, B3H–BFH, C2H–C3H, C7H, CBH–CFH, D7H–DFH, E4H–EFH, F5H–F7H and FDH–FFH in page 0 to 4
- 150H–17FH, 200H–27FH, 300H–37FH, 400H–44FH

(Refer to the "E0C621C Technical Manual" for details.)

■ Undefined Code

The SLP instruction is not specified for the E0C621C and so cannot be used.

■ OPTLD Command

In the ICS621C, OPTLD command can be used.
This command is used to load HEX files (function option and segment option data for LCD) in the EVA621C memory with the ICE6200.

```
Load of function option data:   #OPTLD,1,C21CXXX↵
Load of segment option data:    #OPTLD,2,C21CXXX↵
```

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 EVA621C 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 EVA621C segment option data memory.
It is HEX file output by the segment option generator and has intel HEX format.
Loading time of a segment option HEX data is dependent on the OSC3 system clock. It takes about 5 to 10 minutes.

Examples

`#OPTLD,1,C21CXXX`↵ .... C21CXXXF.HEX file is loaded in the function option data memory.

`#OPTLD,2,C21CXXX`↵ .... C21CXXXS.HEX file is loaded in the segment option data memory.

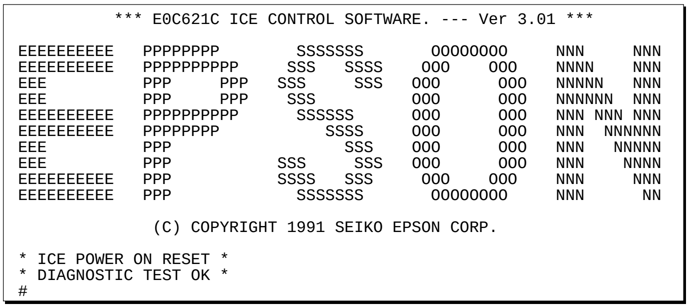
6.3 ICS621C Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:	ICS621C.BAT	(ICS621CJ.EXE)	. . . for MS-DOS
	ICS621CB.BAT	(ICS621CW.EXE)	. . . for PC-DOS
Starting command:	**ICS621C (ICS621CJ)↵**		. . . for MS-DOS
	ICS621CB (ICS621CW)↵		. . . for PC-DOS
Input file:	C21CXXXL.HEX	(Object file, low-order)	
	C21CXXXH.HEX	(Object file, high-order)	
	C21CXXXD.HEX	(Data RAM file)	
	C21CXXXC.HEX	(Control file)	
	C21CXXXF.HEX	(Function option HEX file)	
	C21CXXXS.HEX	(Segment option HEX file)	
Output file:	C21CXXXL.HEX	(Object file, low-order)	
	C21CXXXH.HEX	(Object file, high-order)	
	C21CXXXD.HEX	(Data RAM file)	
	C21CXXXC.HEX	(Control file)	

■ Display example



```
          *** E0C621C 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 ICS621C is started, the start-up message is displayed, and a self-test is automatically performed. ICS621C 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 ICS621C program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS621C.*

■ 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 EVA621C 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 EVA621C CPU internal registers
		#SR↵	Set EVA621C CPU internal registers
		#I ↵	Reset EVA621C CPU
		#DXY↵	Display X, Y, MX and MY
		#SXY↵	Set data for X and Y display and MX, MY

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

↵ means press the RETURN key.

7 MASK DATA CHECKER MDC621C

7.1 MDC621C Outline

The Mask Data Checker MDC621C is a software tool which checks the program data (C21CXXXH.HEX and C21CXXXL.HEX) and option data (C21CXXXF.DOC and C21CXXXS.DOC) created by the user and creates the data file (C621CXXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC621C has the capability to restore the generated data file (C621CXXX.PA0) to the original file format.

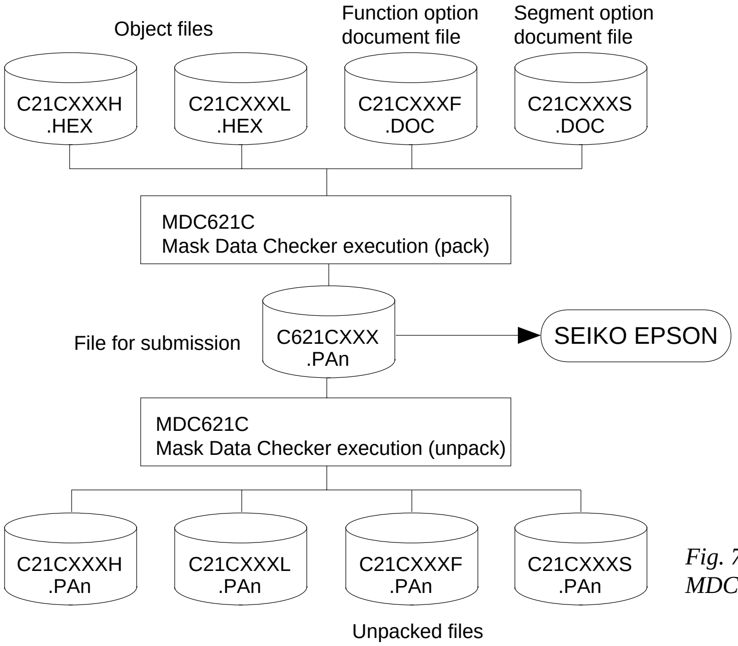


Fig. 7.1.1
MDC621C execution flow

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

7.2 MDC621C Quick Reference

■ Starting command and input/output files

Execution file:	MDC621C.EXE		
Starting command:	**MDC621C**↵		↵ *indicates the Return key.*
Input file:	C21CXXXL.HEX	(Object file, low-order)	When packing
	C21CXXXH.HEX	(Object file, high-order)	
	C21CXXXF.DOC	(Function option document file)	
	C21CXXXS.DOC	(Segment option document file)	
	C621CXXX.PAn	(Packed file)	When unpacking
Output file:	C621CXXX.PAn	(Packed file)	When packing
	C21CXXXL.PAn	(Object file, low-order)	When unpacking
	C21CXXXH.PAn	(Object file, high-order)	
	C21CXXXF.PAn	(Function option document file)	
	C21CXXXS.PAn	(Segment option document file)	

■ Display examples

```
           *** E0C621C 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 1993 SEIKO EPSON CORP.

                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

                    PLEASE SELECT NO.?
```

Start-up message

When MDC621C 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)

        C21CXXXH.HEX --------+
                             |
        C21CXXXL.HEX --------+
                             |-------- C21CXXX.PAn (PACK FILE)
        C21CXXXF.DOC --------+
                             |
        C21CXXXS.DOC --------+

 PLEASE INPUT PACK FILE NAME (C621CXXX.PAn) ?  C621C0A0.PA0↵  ...(2)

        C21C0A0H.HEX --------+
                             |
        C21C0A0L.HEX --------+
                             |-------- C21C0A0.PA0
        C21C0A0F.DOC --------+
                             |
        C21C0A0S.DOC --------+
```

Packing of data

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

(2) Enter the file name.
After submitting the data to Seiko Epson and there is a need to re-submit the data, increase the numeric value of "n" by one when the input is made.
(Example: When re-submitting data after "C621CXXX.PA0" has been submitted, the pack file name should be entered as "C621CXXX.PA1".)

With this, the mask file (C621CXXX.PAn) is generated, and the MDC621C 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 (ASM621C) 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 (C621CXXX.PAn) ?  C621C0A0.PA0↵...(2)

                             +-------- C21C0A0H.PA0
                             |
                             +-------- C21C0A0L.PA0
        C621C0A0.PA0 --------|
                             +-------- C21C0A0F.PA0
                             |
                             +-------- C21C0A0S.PA0
```

Unpacking of data

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

(2) Enter the packed file name.

With this, the mask data file (C621CXXX.PAn) is restored to the original file format, and the MDC621C 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`	FOG621C, SOG621C 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. E0C621C 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	Mne-monic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0			↕	↕	7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0			↕	↕	7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0			↕	↕	7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0			↕	↕	7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0					5	r←i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0					5	r←q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0					5	A←M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0					5	B←M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0					5	M(n3~n0)←A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0					5	M(n3~n0)←B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0					5	M(X)←i3~i0, X←X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0					5	r←q, X←X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0					5	M(Y)←i3~i0, Y←Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0					5	r←q, Y←Y+1
	LBPX	MX, *l*	1	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					5	M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, X←X+2
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i3	i2	i1	i0	↑	↑	↑	↑	7	F←F∨i3~i0
	RST	F, i	1	1	1	1	0	1	0	1	i3	i2	i1	i0	↓	↓	↓	↓	7	F←F∧i3~i0
	SCF		1	1	1	1	0	1	0	0	0	0	0	1				↑	7	C←1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0				↓	7	C←0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0			↑		7	Z←1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1			↓		7	Z←0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0		↑			7	D←1 (Decimal Adjuster ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1		↓			7	D←0 (Decimal Adjuster OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	↑				7	I ←1 (Enables Interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	↓				7	I ←0 (Disables Interrupt)
Stack operation instructions	INC	SP	1	1	1	1	1	1	0	1	1	0	1	1					5	SP← SP+1
	DEC	SP	1	1	1	1	1	1	0	0	1	0	1	1					5	SP← SP-1
	PUSH	r	1	1	1	1	1	1	0	0	0	0	r1	r0					5	SP← SP-1, M(SP)←r
		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. E0C621C 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–14FH)

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																

Display memory - 1 (050H–07FH, R/W)

PROGRAM NAME:

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

Display memory - 2 (450H–47FH, W only)

PROGRAM NAME:

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

I/O memory (80H–FCH)

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	NAME																
1		MSB	0	0														
			0	0														
2			SVDDT	CLKCHG														
		LSB	SVDON	OSCC														
3	9	NAME																
		MSB	SIK03	K03	KCP03		SIK13	K13	KCP13									
4			SIK02	K02	KCP02		SIK12	K12	KCP12									
			SIK01	K01	KCP01		SIK11	K11	KCP11									
		LSB	SIK00	K00	KCP00		SIK10	K10	KCP10					Unused area				
	A	NAME																
		MSB	R03															
			R02															
			R01															
		LSB	R00															
	B	NAME																
		MSB	IOC03	PUL03	P03													
			IOC02	PUL02	P02													
			IOC01	PUL01	P01													
		LSB	IOC00	PUL00	P00													
	C	NAME																
		MSB	LDMS	0			BZE	FOUTE	0		0	TM3	TM7					
			STCD	0			0	0	0		0	TM2	TM6					
			LDTY1	LOFF			0	FOFQ1	0		TMRUN	TM1	TM5					
		LSB	LDTY0	LPWR			BZFQ	FOFQ0	WDRST		TMRST	TM0	TM4					
	D	NAME																
		MSB	DBGOS	0	OVTBC	MC03	MC07	MC11	MC15									
			RFDBG	0	OVMC	MC02	MC06	MC10	MC14									
			RFCLK	0	RFRUN	MC01	MC05	MC09	MC13									
		LSB	RFPWR	SENS0	0	MC00	MC04	MC08	MC12									
	E	NAME																
		MSB	0	RCDIV	RIC3	ROUT3												
			REMSO	RCDTY	RIC2	ROUT2												
			REMDC	RT1	RIC1	ROUT1												
		LSB	REMCR	RT0	RIC0	ROUT0												
	F	NAME																
		MSB	0	0	0	0	0				0	0	0	0	0			
			0	0	0	0	EIT2				0	0	0	0	IT2			
			0	0	0	0	EIT1				0	0	0	0	IT1			
		LSB	EIREM	EIRF	EIK1	EIK0	EIT0				IREM	IRF	IK1	IK0	IT0			

APPENDIX C. E0C621C I/O MEMORY MAP

I/O memory map (80H–B2H)

Address	Register								Comment
*7	D3	D2	D1	D0	Name	Init *1	1	0	
80H	0	0	SVDDT	SVDON	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	SVDDT	0	Low	Normal	Supply voltage detection data
					SVDON	0	On	Off	SVD circuit On/Off
81H	0	0	CLKCHG	OSCC	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		CLKCHG	0	OSC3	OSC1	CPU clock switch
					OSCC	0	On	Off	OSC3 oscillation On/Off
90H	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	
91H	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	
92H	KCP03	KCP02	KCP01	KCP00	KCP03	1	↓	↑	Input comparison register (K00–K03)
					KCP02	1	↓	↑	
	R/W				KCP01	1	↓	↑	
					KCP00	1	↓	↑	
94H	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	
95H	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	
96H	KCP13	KCP12	KCP11	KCP10	KCP13	1	↓	↑	Input comparison register (K10–K13)
					KCP12	1	↓	↑	
	R/W				KCP11	1	↓	↑	
					KCP10	1	↓	↑	
A0H	R03 (BZ)	R02 (FOUT)	R01 (REMDC)	R00 (REMCR)	R03	0	High	Low	Output port R03 Fix at "0" during BZ, FOUT, REMDC and REMCR output
					R02	0	High	Low	Output port R02
	R/W				R01	0	High	Low	Output port R01
					R00	0	High	Low	Output port R00
B0H	IOC03	IOC02	IOC01	IOC00	IOC03	0	Output	Input	I/O control register (P00–P03)
					IOC02	0	Output	Input	
	R/W				IOC01	0	Output	Input	
					IOC00	0	Output	Input	
B1H	PUL03	PUL02	PUL01	PUL00	PUL03	1	On	Off	Pull up control register (P00–P03)
					PUL02	1	On	Off	
	R/W				PUL01	1	On	Off	
					PUL00	1	On	Off	
B2H	P03	P02	P01	P00	P03	1	High	Low	I/O port P00–P03
					P02	1	High	Low	
	R/W				P01	1	High	Low	
					P00	1	High	Low	

Remarks

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

*5 Constantly "0" when being read
*6 Refer to main manual
*7 Page switching in I/O memory is not necessary

I/O memory map (C0H–D6H)

Address *7	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
C0H	LDMS	STCD	LDTY1	LDTY0	LDMS	0	0p(R/W)	4p(W)	LCD display memory area selection
					STCD	0	Static	Dynamic	LCD drive switch
	R/W				LDTY1	0			LCD drive duty selection
					LDTY0	0			0: 1/2, 1: 1/3, 2&3: 1/4
C1H	0	0	LOFF	LPWR	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W		LOFF	0	All off	Normal	LCD display control
					LPWR	0	On	Off	LCD power supply On/Off
C4H	BZE	0	0	BZFQ	BZE	0	Enable	Disable	Buzzer output enable
					0 *5	– *2			Unused
	R/W	R		R/W	0 *5	– *2			Unused
					BZFQ	0	4,096 Hz	2,048 Hz	Buzzer frequency selection
C5H	FOUTE	0	FOFQ1	FOFQ0	FOUTE	0	Enable	Disable	FOUT output enable
					0 *5	– *2			Unused
	R/W	R	R/W		FOFQ1	0			FOUT frequency selection
					FOFQ0	0			0: 512 Hz, 1: 4,096 Hz, 2: f_{OSC1}, 3: f_{OSC3}
C6H	0	0	0	WDRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			W	0 *5	– *2			Unused
					WDRST*5	Reset	Reset	–	Watchdog timer reset
C8H	0	0	TMRUN	TMRST	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R		R/W	W	TMRUN	0	Run	Stop	Clock timer Run/Stop
					TMRST*5	Reset	Reset	–	Clock timer reset
C9H	TM3	TM2	TM1	TM0	TM3	0			Clock timer data (16 Hz)
					TM2	0			Clock timer data (32 Hz)
	R				TM1	0			Clock timer data (64 Hz)
					TM0	0			Clock timer data (128 Hz)
CAH	TM7	TM6	TM5	TM4	TM7	0			Clock timer data (1 Hz)
					TM6	0			Clock timer data (2 Hz)
	R				TM5	0			Clock timer data (4 Hz)
					TM4	0			Clock timer data (8 Hz)
D0H	DBGOS	RFDBG	RFCLK	RFPWR	DBGOS	0	Sensor	Ref.-R	R/F converter debugging element selection
					RFDBG	0	Debug	Normal	R/F converter debug mode selection
	R/W				RFCLK	0	OSC3	OSC1	R/F converter clock source selection
					RFPWR	0	1	0	Reserved register
D1H	0	0	0	SENS0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					SENS0	0	SEN1	SEN0	Sensor selection
D2H	OVTBC	OVMC	RFRUN	0	OVTBC	0	Overflow	Non-ov	Time base counter overflow flag *6
					OVMC	0	Overflow	Non-ov	Measurement counter overflow flag *6
	R/W			R	RFRUN	0	Run	Stop	R/F converter Run/Stop
					0 *5	– *2			Unused
D3H	MC03	MC02	MC01	MC00	MC03	– *3			Measurement counter MC00–MC03
					MC02	– *3			
	R/W				MC01	– *3			
					MC00	– *3			
D4H	MC07	MC06	MC05	MC04	MC07	– *3			Measurement counter MC04–MC07
					MC06	– *3			
	R/W				MC05	– *3			
					MC04	– *3			
D5H	MC11	MC10	MC09	MC08	MC11	– *3			Measurement counter MC08–MC11
					MC10	– *3			
	R/W				MC09	– *3			
					MC08	– *3			
D6H	MC15	MC14	MC13	MC12	MC15	– *3			Measurement counter MC12–MC15
					MC14	– *3			
	R/W				MC13	– *3			
					MC12	– *3			

I/O memory map (E0H–FCH)

address *7	Register								Comment
	D3	D2	D1	D0	Name	Init *1	1	0	
E0H	0	REMSO	REMDC	REMCR	0 *5	– *2			Unused
					REMSO	0	On	Off	REM soft-timer output control
	R	R/W			REMDC	0	On	Off	REM DC output control
					REMCR	0	On	Off	REM carrier output control
E1H	RCDIV	RCDTY	RT1	RT0	RCDIV	0	f_{OSC3}/12	f_{OSC3}/8	REM carrier cycle selection
					RCDTY	0			REM carrier duty selection *6
	R/W				RT1	0			REM τ cycle selection
					RT0	0			0: 1/12, 1: 1/16, 2:1/20, 3: 1/32 (x fcarrier)
E2H	RIC3	RIC2	RIC1	RIC0	RIC3	1			
					RIC2	1			REM interrupt τ cycle selection
	R/W				RIC1	1			0H = 1τ ... FH = 16τ
					RIC0	1			
E3H	ROUT3	ROUT2	ROUT1	ROUT0	ROUT3	0			
					ROUT2	0			REM carrier output width selection
	R/W				ROUT1	0			0H = 0τ ... FH = 15τ
					ROUT0	0			
F0H	0	0	0	EIREM	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIREM	0	Enable	Mask	Interrupt mask register (Remote controller)
F1H	0	0	0	EIRF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIRF	0	Enable	Mask	Interrupt mask register (R/F converter)
F2H	0	0	0	EIK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
F3H	0	0	0	EIK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R			R/W	0 *5	– *2			Unused
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
F4H	0	EIT2	EIT1	EIT0	0 *5	– *2			Unused
					EIT2	0	Enable	Mask	Interrupt mask register (Clock timer 2 Hz)
	R	R/W			EIT1	0	Enable	Mask	Interrupt mask register (Clock timer 8 Hz)
					EIT0	0	Enable	Mask	Interrupt mask register (Clock timer 32 Hz)
F8H	0	0	0	IREM	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IREM *4	0	Yes	No	Interrupt factor flag (Remote controller)
F9H	0	0	0	IRF	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IRF *4	0	Yes	No	Interrupt factor flag (R/F converter)
FAH	0	0	0	IK1	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK1 *4	0	Yes	No	Interrupt factor flag (K10–K13)
FBH	0	0	0	IK0	0 *5	– *2			Unused
					0 *5	– *2			Unused
	R				0 *5	– *2			Unused
					IK0 *4	0	Yes	No	Interrupt factor flag (K00–K03)
FCH	0	IT2	IT1	IT0	0 *5	– *2			Unused
					IT2 *4	0	Yes	No	Interrupt factor flag (Clock timer 2 Hz)
	R				IT1 *4	0	Yes	No	Interrupt factor flag (Clock timer 8 Hz)
					IT0 *4	0	Yes	No	Interrupt factor flag (Clock timer 32 Hz)

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 ICS621CJ.EXE PC-DOS ICS621CW.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 EVA621C 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 ICS621CP.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.
SOG621C	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
ASM621C	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.
MDC621C	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?
EVA621C	The EVA621C does not work when it is used independently.	Check the following and remedy if necessary: • Has the EPROM for F.HEX and S.HEX been replaced by the EPROM for the target? • Is the EPROM for F.HEX and S.HEX installed correctly? • Is the appropriate voltage being supplied? (5V DC, 3A, or more) • Are the program ROMs (H and L) installed correctly? • Is data written from address 4000H? (When the 27C256 is used as the program ROM)
	Target segment does not light.	Check the following and remedy if necessary: • Is an EPROM with an access time of 250 ns or less being used for S.HEX. • Has the VADJ VR inside the EVA621C 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

ENERGY
SAVING
EPSON

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

EPSON

SEIKO EPSON CORPORATION

ELECTRONIC DEVICES MARKETING DIVISION

■ Electronic devices information on the Epson WWW server

http://www.epson.co.jp

First issue SEPTEMBER 1994, Printed SEPTEMBER 1998 in Japan Ⓜ A