MF582-04b

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER **E0C621A**

DEVELOPMENT TOOL MANUAL
(EVA621AR Support)

ENERGY SAVING
EPSON

SEIKO EPSON CORPORATION

NOTICE

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

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.

E0C621A Development Tool Manual

PREFACE

This manual mainly explains the outline of the development support tool for the 4-bit Single Chip Microcomputer E0C621A.

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 EVA621AR Manual ICE6200 (ICE62R) Hardware Manual
Development procedure	☞	E0C62 Family Technical Guide
Device (E0C621A)	☞	E0C621A 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 DEV621A

The below software are included in the product of the E0C621A development support tool DEV621A.

1. Cross Assembler ASM621A	Cross assembler for program preparation
2. Function Option Generator FOG621A	Function option data preparation program
3. Segment Option Generator SOG621A	Segment option data preparation program
4. ICE Control Software ICS621A	ICE control program
5. Mask Data Checker MDC621A	Mask data preparation program

1.2 Developmental Environment

The software product of the development support tool DEV621A 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 E0C621A, the above-mentioned host computer, editor, P-ROM writer, printer, etc. must be prepared by the user in addition to the development tool which is normally supported by Seiko Epson.

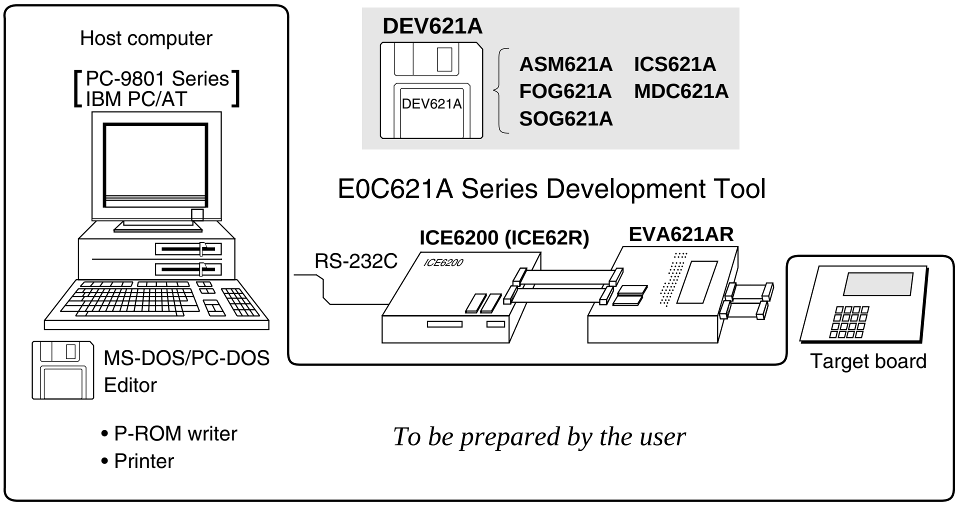


Fig. 1.2.1 System configuration

Note *The DEV621A system requires a host computer with a RAM capacity of about 140K bytes. Since the ICE6200 (ICE62R) is connected to the host computer with a RS-232C serial interface, adapter board for asynchronous communication will be required depending on the host computer used.*

1.3 Development Flow

Figure 1.3.1 shows the development flow through the DEV621A.

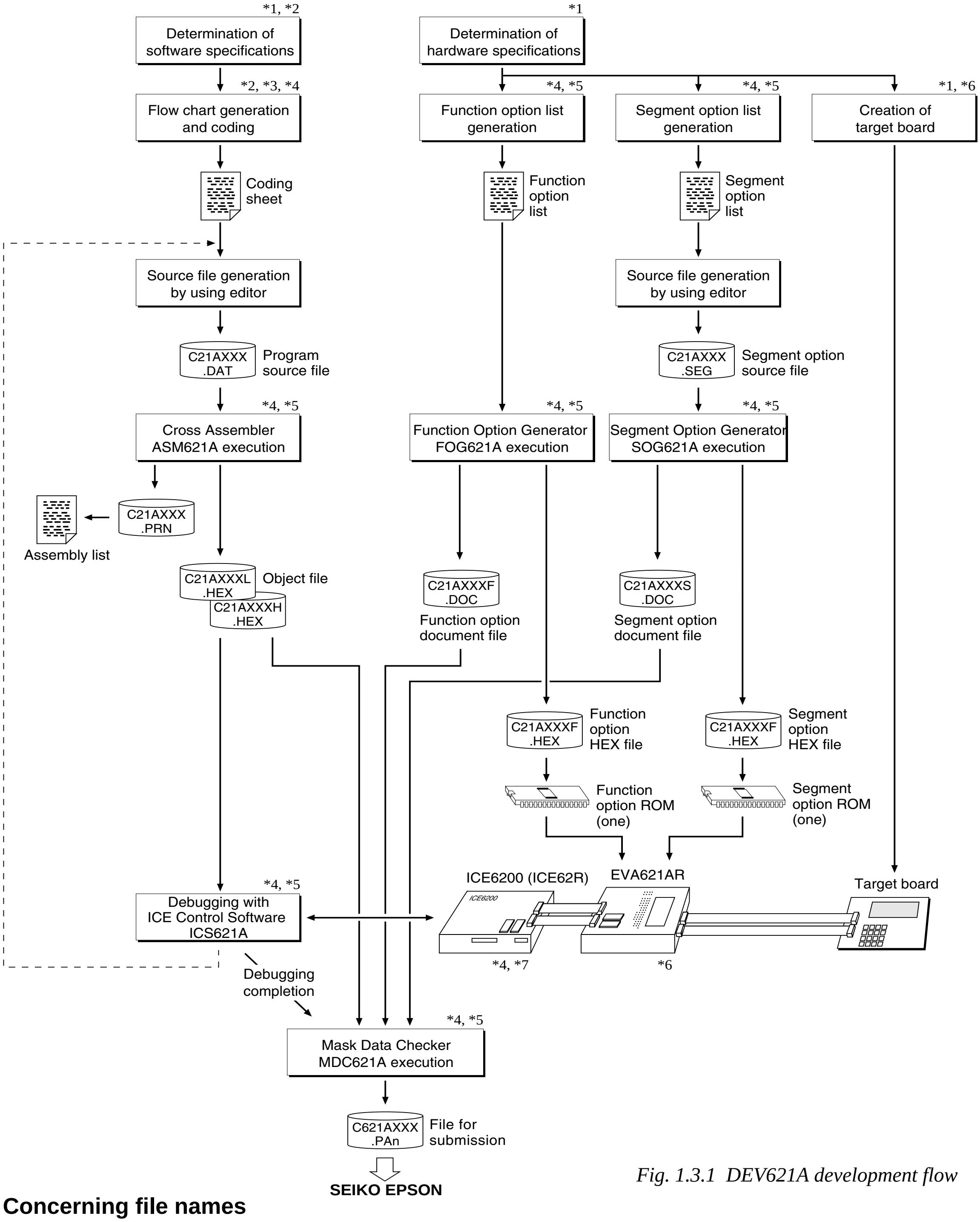

*1, *2
Determination of software specifications
*1
Determination of hardware specifications
*2, *3, *4
Flow chart generation and coding
*4, *5
Function option list generation
*4, *5
Segment option list generation
*1, *6
Creation of target board
Coding sheet
Function option list
Segment option list
Source file generation by using editor
Source file generation by using editor
C21AXXX .DAT
Program source file
C21AXXX .SEG
Segment option source file
*4, *5
Cross Assembler ASM621A execution
*4, *5
Function Option Generator FOG621A execution
*4, *5
Segment Option Generator SOG621A execution
C21AXXX .PRN
Assembly list
C21AXXXL .HEX
C21AXXXH .HEX
Object file
C21AXXXF .DOC
Function option document file
C21AXXXS .DOC
Segment option document file
C21AXXXF .HEX
Function option HEX file
C21AXXXF .HEX
Segment option HEX file
Function option ROM (one)
Segment option ROM (one)
ICE6200 (ICE62R)
EVA621AR
Target board
ICE6200
*4, *5
Debugging with ICE Control Software ICS621A
*4, *7
*6
Debugging completion
*4, *5
Mask Data Checker MDC621A execution
C621AXXX .PAn
File for submission
SEIKO EPSON

Fig. 1.3.1 DEV621A development flow

Concerning file names

All the input-output file name for the each development support tool commonly use "C21AXXX". In principle each file should be produced in this manner. Seiko Epson will designate the "XXX" for each customer.

☞ Reference Manual

*1 E0C621A Technical Hardware Manual
*2 E0C621A Technical Software Manual
*3 E0C6200/6200A Core CPU Manual
*4 E0C62 Family Development Tool Reference Manual
*5 E0C621A Development Tool Manual (this manual)
*6 EVA621AR Manual
*7 ICE6200 (ICE62R) Hardware Manual

1.4 Production of Execution Disk

Execution files for each software development support tool and batch and parameter files for the ICE6200 (ICE62R) are recorded in the DEV621A floppy disk.
The content of the files contained in the DEV621A floppy disk are shown below.

PC-DOS version	MS-DOS version	Contents
ASM621A.EXE	ASM621A.EXE	Cross Assembler execution file
FOG621A.EXE	FOG621A.EXE	Function Option Generator execution file
ICS621AB.BAT	ICS621A.BAT	ICE Control Software batch file
ICS621AW.EXE	ICS621AJ.EXE	ICE Control Software execution file
ICS621AP.PAR	ICS621AP.PAR	ICE Control Software parameter file
MDC621A.EXE	MDC621A.EXE	Mask Data Checker execution file
SOG621A.EXE	SOG621A.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 (DEV621A, 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 (DEV621A), then insert the original disk into the A drive and execute the COPY command.

```
C\>MD DEV621A↵
C\>CD DEV621A↵
C\DEV621A\>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 MDC621A will handle many files.

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

Example:

Setting of FILES (CONFIG.SYS)

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

- ■ In "ICS621A(B).BAT" the batch process is indicated such that the ICS621AJ(W).EXE is executed after the execution of the command for the setting of the RS-232C communication parameters.
  The SPEED (MS-DOS) or MODE (PC-DOS) command is used for setting the RS-232C, so you should copy these commands into the disk to be executed or specify a disk or directory with the command using the PATH command.

RS-232C Setting (PC-DOS version)

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

RS-232C Setting (MS-DOS version)

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

2 CROSS ASSEMBLER ASM621A

2.1 ASM621A Outline

The ASM621A cross assembler is an assembler program for generating the machine code used by the E0C621A 4-bit, single-chip microcomputers. The Cross Assembler ASM621A will assemble the program source files which have been input by the user's editor and will generate an object file in Intel-Hex format and assembly list file.
In this assembler, program modularization has been made possible through macro definition functions and programming independent of the ROM page structure has been made possible through the auto page set function. In addition, consideration has also been given to precise error checks for program capacity (ROM capacity) overflows, undefined codes and the like, and for debugging of such things as label tables for assembly list files and cross reference table supplements.

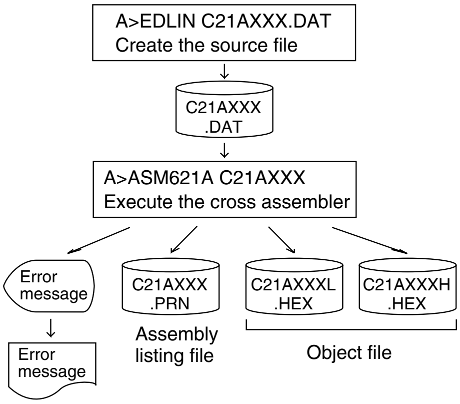


Fig. 2.1.1 ASM621A execution flow

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

2.2 E0C621A Restrictions

Note the following when generating a program by the E0C621A:

■ ROM area

The capacity of the E0C621A 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, I/O area

The capacity of the E0C621A RAM is 208 words (000H to 07FH) and I/O area is 48 words (0D0H to 0EFH, 0F0H to 0FFH, 4 bits/word). Memory access is invalid when the unused area of the index register is specified.

■ Undefined codes

The following instructions have not been defined in the E0C621A instruction sets.

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

2.3 ASM621A Quick Reference

■ Starting command and input/output files

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

Execution file: ASM621A.EXE

Starting command: **ASM621A_ [drive-name:] source-file-name [.shp]_ [-N]↵**

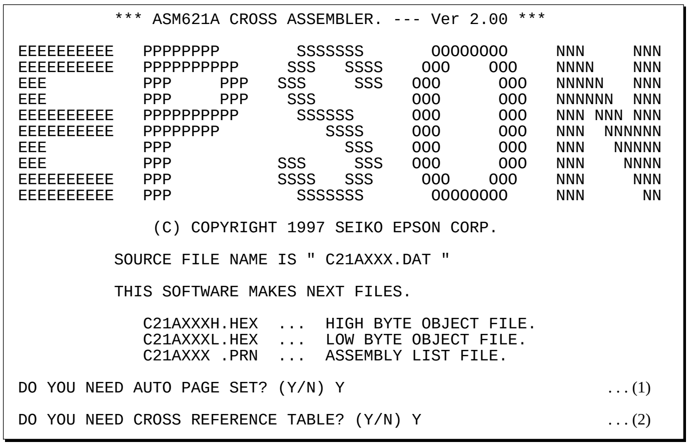
Option:

.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: C21AXXX.DAT (Source file)

Output file:
C21AXXXL.HEX (Object file, low-order)
C21AXXXH.HEX (Object file, high-order)
C21AXXX.PRN (Assembly listing file)

■ Display example



```
          *** ASM621A 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 1997 SEIKO EPSON CORP.

          SOURCE FILE NAME IS " C21AXXX.DAT "

          THIS SOFTWARE MAKES NEXT FILES.

             C21AXXXH.HEX  ...  HIGH BYTE OBJECT FILE.
             C21AXXXL.HEX  ...  LOW BYTE OBJECT FILE.
             C21AXXX .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 ASM621A 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, ASM621A assembles the source file.
To suspend execution, press the "CTRL" and "C" keys together at stage (1) or (2).

■ Operators

Arithmetic operators		Logical operators	
+a	Monadic positive	a_AND_b	Logical product
-a	Monadic negative	a_OR_b	Logical sum
a+b	Addition	a_XOR_b	Exclusive logical sum
a-b	Subtraction	NOT_a	Logical negation
a*b	Multiplication	Relational operators	
a/b	Division	a_EQ_b	True when a is equal to b
a_MOD_b	Remainder of a/b	a_NE_b	True when a is not equal to b
a_SHL_b	Shifts a b bits to the left	a_LT_b	True when a is less than b
a_SHR_b	Shifts a b bits to the right	a_LE_b	True when a is less than or equal to b
HIGH_a	Separates the high-order eight bits from a	a_GT_b	True when a is greater than b
LOW_a	Separates the low-order eight bits from a	a_GE_b	True when a is greater than or equal to b

■ Pseudo-instructions

Pseudo-instruction	Meaning	Example of use

EQU (Equation)	To allocate data to label	`ABC EQU 9`
		`BCD EQU ABC+1`

SET (Set)	To allocate data to label (data can be changed)	`ABC SET 0001H`
		`ABC SET 0002H`

DW (Define Word)	To define ROM data	`ABC DW 'AB'`
		`BCD DW 0FFBH`

ORG (Origin)	To define location counter	`ORG 100H`
		`ORG 256`

PAGE (Page)	To define boundary of page	`PAGE 1H`
		`PAGE 3`

SECTION (Section)	To define boundary of section	`SECTION`
END (End)	To terminate assembly	`END`

MACRO (Macro)	To define macro	`CHECK MACRO DATA`
		`LOCAL LOOP`
		`LOOP CP MX,DATA`
		`JP NZ,LOOP`
		`ENDM`
		
		`CHECK 1`

LOCAL (Local)	To make local specification of label during macro definition	
ENDM (End Macro)	To end macro definition	

■ Error messages

Error message	Explanation
S (Syntax Error)	An unrecoverable syntax error was encountered.
U (Undefined Error)	The label or symbol of the operand has not been defined.
M (Missing Label)	The label field has been omitted.
O (Operand Error)	A syntax error was encountered in the operand, or the operand could not be evaluated.
P (Phase Error)	The same label or symbol was defined more than once.
R (Range Error)	• The location counter value exceeded the upper limit of the program memory, or a location exceeding the upper limit was specified. • A value greater than that which the number of significant digits of the operand will accommodate was specified.
! (Warning)	• Memory areas overlapped because of a "PAGE" or "ORG" pseudo-instruction or both. • A statement exceeded a page boundary although its location was not specified.
FILE NAME ERROR	The source file name was longer than 8 characters.
FILE NOT PRESENT	The specified source file was not found.
DIRECTORY FULL	No space was left in the directory of the specified disk.
FATAL DISK WRITE ERROR	The file could not be written to the disk.
LABEL TABLE OVERFLOW	The number of defined labels and symbols exceeded the label table capacity (4000).
CROSS REFERENCE TABLE OVERFLOW	The label/symbol reference count exceeded the cross-reference table capacity (only when the cross-reference table is generated).

3 FUNCTION OPTION GENERATOR FOG621A

3.1 FOG621A Outline

With the 4-bit single-chip E0C621A microcomputers, the customer may select 15 hardware options. By modifying the mask patterns of the E0C621A according to the selected options, the system can be customized to meet the specifications of the target system.

The Function Option Generator FOG621A 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 FOG621A, the E0C621A mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA621AR) hardware option ROM is simultaneously generated with the data file.

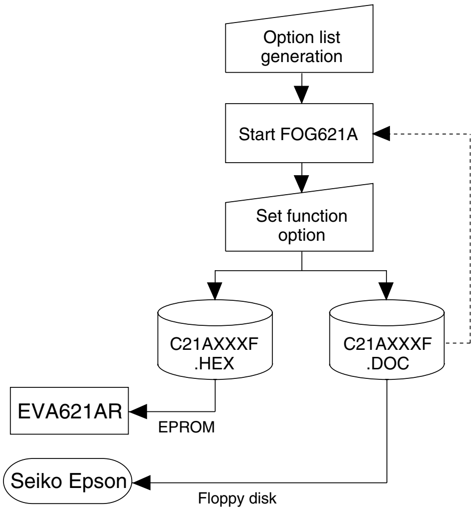


Fig. 3.1.1 FOG621A execution flow

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

3.2 E0C621A Option List

Multiple specifications are available in each option item as indicated in the Option List. Using "3.3 Option Specifications and Selection Message" as reference, select the specifications that meet the target system. Be sure to record the specifications for unused ports too, according to the instructions provided.

1. OSC3 OSCILLATION CIRCUIT

□ 1. Ceramic Oscillation Circuit
□ 2. CR Oscillation Circuit
□ 3. Not Use

2. WATCHDOG TIMER

□ 1. Use
□ 2. Not Use

3. INPUT INTERRUPT (K00–K03)

□ 1. K00
□ 2. K00, K01
□ 3. K00, K01, K02
□ 4. K00, K01, K02, K03

4. INPUT INTERRUPT (K10–K13)

□ 1. K10
□ 2. K10, K11
□ 3. K10, K11, K12
□ 4. K10, K11, K12, K13

5. INPUT INTERRUPT NOISE REJECTOR (K10–K13)

□ 1. Use
□ 2. Not Use

6. INPUT PORT PULL-UP RESISTOR

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

7. OUTPUT PORT 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
- R10 □ 1. Complementary □ 2. Nch Open Drain
- R11 □ 1. Complementary □ 2. Nch Open Drain
- R20 □ 1. Complementary □ 2. Nch Open Drain

8. R12 OUTPUT PORT SPECIFICATION

- R12 □ 1. Complementary □ 2. Nch Open Drain

9. R12 OUTPUT PORT TYPE

- DC OUTPUT □ 1. DC
- FOUT OUTPUT □ 2. 256 [Hz]
  □ 3. 512 [Hz]
  □ 4. 1024 [Hz]
  □ 5. 2048 [Hz]
  □ 6. 4096 [Hz]
  □ 7. 8192 [Hz]
  □ 8. 16384 [Hz]
  □ 9. 32768 [Hz]
- BUZZER OUTPUT □ 10. R13 BUZZER Reverse Output

10. R13 OUTPUT PORT SPECIFICATION

- R13 □ 1. Complementary □ 2. Nch Open Drain

11. R13 OUTPUT PORT TYPE

- BUZZER OUTPUT □ 1. 2048 [Hz]
  □ 2. 4096 [Hz]
- DC OUTPUT □ 3. DC

12. I/O PORT FUNCTION

- P00 □ 1. Input & Output □ 2. Output Only
- P01 □ 1. Input & Output □ 2. Output Only
- P02 □ 1. Input & Output □ 2. Output Only
- P03 □ 1. Input & Output □ 2. Output Only

13. I/O PORT SPECIFICATION

- P00 □ 1. Complementary □ 2. Nch Open Drain
- P01 □ 1. Complementary □ 2. Nch Open Drain
- P02 □ 1. Complementary □ 2. Nch Open Drain
- P03 □ 1. Complementary □ 2. Nch Open Drain

14. R33 (REM) OUTPUT PORT TYPE

□ 1. REM Output
□ 2. DC Output

15. LCD COMMON DUTY

□ 1. 1/3 Duty
□ 2. 1/4 Duty

3.3 Option Specifications and Selection Message

Screen that can be selected as function options set on the E0C621A are shown below, and their specifications are also described.

1 OSC3 oscillation circuit

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

--- OSC 3 SYSTEM CLOCK ---

        1. Ceramic
        2. CR
        3. Not Use

PLEASE SELECT NO.(1) ? 1↵

        1. Ceramic     SELECTED
```

Select oscillation circuit which uses OSC3 or OSC4 terminal. Ceramic oscillation circuit is suitable for obtaining stable oscillation frequency, while CR oscillation circuit is recommended when you wish to reduce external components.

If you select ceramic oscillation circuit, you need a ceramic oscillator, a gate capacity, and a drain capacity as external components. If you select CR oscillation circuit, these capacities are mounted internally, so all you need is a resistor.

If you select "Not Use", the system starts on OSC1 oscillation circuit.

2 Watchdog timer

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

--- WATCH DOG TIMER ---

        1. Use
        2. Not Use

PLEASE SELECT NO.(1) ? 1↵

        1. Use   SELECTED
```

The system has the internal watchdog timer in order to detect CPU going wildcat.
CPU will be re-initialized if the program does not reset the watchdog timer for 3–4 seconds. Select whether you use this watchdog timer.

3 Input interrupt (K00–K03)

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

--- INTERRUPT (K00-K03) ---

        1. K00
        2. K00,K01
        3. K00,K01,K02
        4. K00,K01,K02,K03

PLEASE SELECT NO.(4) ? 4↵

        4. K00,K01,K02,K03   SELECTED
```

Select among terminals K00–K03 the set that generates the input interrupt factor.
You can select one of four sets of input interrupt generating terminals, from K00 terminal only to all four terminals K00–K03.

You must select at least one terminal (K00) from the terminals K00–K03 as an input interrupt factor generating terminal. If you do not need any of the four terminals to generate interrupt, mask the interrupt mask register EIK0 by the program not to generate any interrupt request to CPU.

4 Input interrupt (K10–K13)

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

--- INTERRUPT (K10-K13) ---

        1. K10
        2. K10,K11
        3. K10,K11,K12
        4. K10,K11,K12,K13

PLEASE SELECT NO.(4) ? 1↵

        1. K10    SELECTED
```

Select among terminals K10–K13 the set that generates the input interrupt factor.
You can select one of four sets of input interrupt generating terminals, from K10 terminal only to all four terminals K10–K13.

You must select at least one terminal (K10) from the terminals K10–K13 as an input interrupt factor generating terminal. If you do not need any of the four terminals to generate interrupt, mask the interrupt mask register EIK1 by the program not to generate any interrupt request to CPU.

5 Input interrupt noise rejector (K10–K13)

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

--- INTERRUPT NOISE REJECTOR (K10-K13) ---

        1. Use
        2. Not Use

PLEASE SELECT NO.(1) ? 1↵

        1. Use   SELECTED
```

In order to avoid misgeneration of interrupt by noise or chattering to input terminals, a noise rejector circuit which operates at 4 kHz sampling is prepared in each input interrupt circuit for terminals K00–K03 and for K10–K13. Since the circuit for terminals K10–K13 responds to input interrupt at a higher speed, you can select not to use the noise rejector circuit.

6 Input port pull-up resistor

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

--- IN 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) ? 2↵

        K11             1. With Resistor
                        2. Gate Direct

PLEASE SELECT NO.(1) ? 2↵

        K12             1. With Resistor
                        2. Gate Direct

PLEASE SELECT NO.(1) ? 2↵

        K13             1. With Resistor
                        2. Gate Direct

PLEASE SELECT NO.(1) ? 2↵

        K00             1. With Resistor   SELECTED
        K01             1. With Resistor   SELECTED
        K02             1. With Resistor   SELECTED
        K03             1. With Resistor   SELECTED
        K10             2. Gate Direct   SELECTED
        K11             2. Gate Direct   SELECTED
        K12             2. Gate Direct   SELECTED
        K13             2. Gate Direct   SELECTED
```

Select "With Resistor" if you wish to use pull-up resistor for each of input ports K00–K03 and K10–K13, and "Gate Direct" if not. The pull-up circuit contains the regular pull-up and the feed back pull-up as shown in Figure 3.3.1.

The time constant of input gate capacity and pull-up resistor may generate a time lag in wave rise, when you shift the state of input port from low level (VSS) to high level (VDD) by using pull-up resistor. Therefore, write proper waiting time in program before taking input port.

If you select "Gate Direct", be sure not to create input floating state.

Select "With Resistor" in the option table above, as to input ports to be not used.

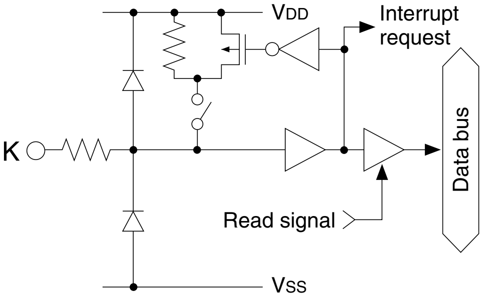


Fig. 3.3.1 Configuration of pull-up resistor

7 Output port specification (R00–R03, R10, R11, R20)

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

--- OUT PORT SPECIFICATION ---

        R00            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        R01            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        R02            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        R03            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        R10            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 1↵

        R11            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 1↵

        R20            1. C-MOS
                       2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 1↵

        R00            2. Nch-Open Drain   SELECTED
        R01            2. Nch-Open Drain   SELECTED
        R02            2. Nch-Open Drain   SELECTED
        R03            2. Nch-Open Drain   SELECTED
        R10            1. C-MOS   SELECTED
        R11            1. C-MOS   SELECTED
        R20            1. C-MOS   SELECTED
```

Select output specification of output ports R00–R03, R10, R11 and R20.
Either "Complementary (C-MOS)" output or "Nch Open Drain" output may be selected for each output port.
If you are using output for key matrix structure, select "Nch Open Drain".
In this case, input port pull-up resistors should be "With Resistor".
Select "Complementary (C-MOS)" output for all unused output ports.

Figure 3.3.2 shows the configuration of output circuit.

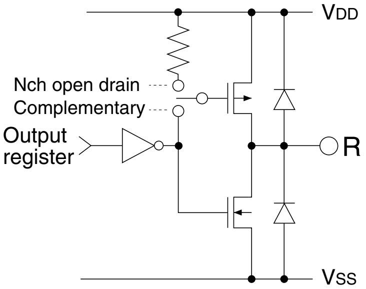


Fig. 3.3.2 Configuration of output circuit

8 R12 output port specification

```
*** OPTION  NO.8 ***

--- R12 OUT PORT SPECIFICATION ---

        1. C-MOS
        2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        2. Nch-Open Drain   SELECTED
```

Select output specification of R12 output port. Either "Complementary (C-MOS)" output or "Nch Open Drain" output may be selected.

The circuit configuration is the same with that of output ports R00–R03, R10, R11 and R20.

Select "Complementary (C-MOS)" output if you are not using R12 output port.

9 R12 output port type

```
*** OPTION  NO.9 ***

--- R12 OUT PORT TYPE ---

         1. D.C.
         2. Fout    256 [Hz]
         3. Fout    512 [Hz]
         4. Fout   1024 [Hz]
         5. Fout   2048 [Hz]
         6. Fout   4096 [Hz]
         7. Fout   8192 [Hz]
         8. Fout  16384 [Hz]
         9. Fout  32768 [Hz]
        10. BUZZER

PLEASE SELECT NO.(1) ? 1↵

         1. D.C.   SELECTED
```

Select output type of R12 output port. You have three choices. Select one output type from DC output, FOUT output, or BUZZER output. However, you must select BUZZER output for R13 output port type if you select BUZZER output here.

If you do not use R12 output port, select DC output.

- In case of selecting DC output
  The port R12 is set in high level (VDD) output by setting "1" on the R12 register, and low level (VSS) by setting "0".

  This output type is the same as that of R00–R03, R10, R11 and R20 output ports.

  Figure 3.3.3 shows the waveform.

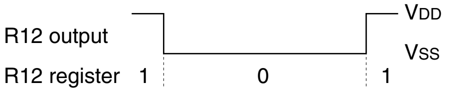


Fig. 3.3.3 Waveform of R12 DC output

- In case of selecting FOUT output
  The port R12 is set in low level (VSS) output by setting "0" on the R12 register, and by setting "1" a square wave at specified frequency with 50% duty and amplitude VDD–VSS. You have eight choices for frequency, from 256 Hz up to 32,768 Hz. Select one in the list.

  FOUT output is normally used to supply clock to other devices, but be careful because it creates hazard at the break of square waves.

  Figure 3.3.4 shows the waveform.

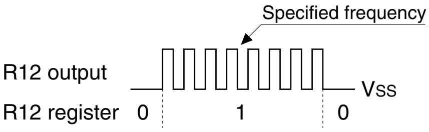


Fig. 3.3.4 Waveform of R12 FOUT output

- In case of selecting BUZZER output
  The port R12 is set in low level (VSS) output by setting "0" on the R12 register, and by setting "1" a square wave at specified frequency with 50% duty and amplitude VDD–VSS.

  BUZZER output from port R12 is the reverse wave of R13 BUZZER output, you must select BUZZER output for R13 output port type if you select BUZZER output here. This doubles the effective voltage applied to the piezo-electric buzzer, and enables direct driving only by the protective circuit against generation of counter electromotive force by the piezo-electric buzzer. Frequency is specified by R13 output port specification.

  Figure 3.3.5 shows the waveform.

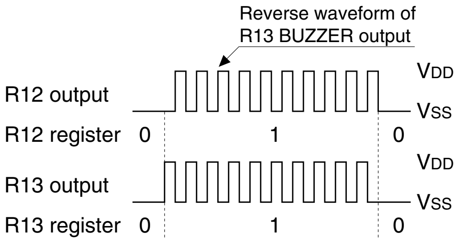


Fig. 3.3.5 Waveform of R12 BUZZER output

10 R13 output port specification

```
*** OPTION  NO.10 ***

--- R13 OUT PORT SPECIFICATION ---

        1. C-MOS
        2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        2. Nch-Open Drain   SELECTED
```

Select output specification of R13 output port. Either "Complementary (C-MOS)" output or "Nch Open Drain" output may be selected.

The circuit configuration is the same as that of output ports R00–R03, R10, R11 and R20.

Select "Complementary (C-MOS)" output if you are not using R13 output port.

11 R13 output port type

```
*** OPTION  NO.11 ***

--- R13 OUT PORT TYPE ---

        1. BUZZER   2048 [Hz]
        2. BUZZER   4096 [Hz]
        3. D.C.

 PLEASE SELECT NO.(3) ? 3↵

        3. D.C.   SELECTED
```

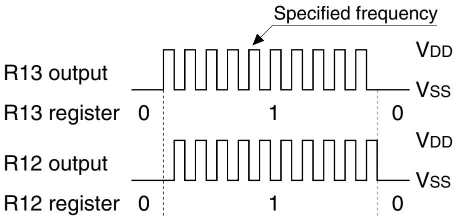


Fig. 3.3.6 Waveform of R13 BUZZER output

Select output type of R13 output port. You have two choices. Select either DC output or BUZZER output.

If you do not use R13 output port, select DC output.

- In case of selecting DC output
  The port R13 is set in high level (VDD) output by setting "1" on the R13 register, and low level (VSS) by setting "0".

  This output type is the same with R00–R03, R10, R11 and R20 output port.
- In case of selecting BUZZER output
  The port R13 is set in low level (VSS) output by setting "0" on the R13 register, and by setting "1" a square wave at specified frequency with 50% duty and amplitude VDD–VSS.

  If you select BUZZER output for R13 output port type, you may also select BUZZER output for R12. In this case, you can achieve direct driving of piezo-electric buzzer between output ports R12 and R13 only with the protective circuit. Specify frequency either 2,048 Hz or 4,096 Hz.
  Figure 3.3.6 shows the waveform.

12 I/O port function

```
*** OPTION  NO.12 ***

--- I/O PORT FUNCTION ---

        P00            1. I/O PORT
                       2. OUT PORT

PLEASE SELECT NO.(1) ? 1↵

        P01            1. I/O PORT
                       2. OUT PORT

PLEASE SELECT NO.(1) ? 1↵

        P02            1. I/O PORT
                       2. OUT PORT

PLEASE SELECT NO.(1) ? 2↵

        P03            1. I/O PORT
                       2. OUT PORT

PLEASE SELECT NO.(1) ? 2↵

        P00            1. I/O PORT   SELECTED
        P01            1. I/O PORT   SELECTED
        P02            2. OUT PORT   SELECTED
        P03            2. OUT PORT   SELECTED
```

Select port function of each I/O port (P00–P03). Each of port P00–P03 can be used for both input and output or for output only.

If you select "Input & Output", the direction of port will be controlled by setting "1" for output and "0" for input on the I/O control register. The I/O control register is only valid for ports specified with I/O function.

The pull-up resistor of an I/O port is switched on by the read signal. It is usually switched off to reduce the leak current. So be careful not to create floating state when you set the port direction as input.

If you select "Output Only", the port will function only as an output port and the I/O control register will become invalid.

Select "Output Only" for the unused I/O ports.

Circuit configuration of I/O port is shown in Figure 3.3.7.

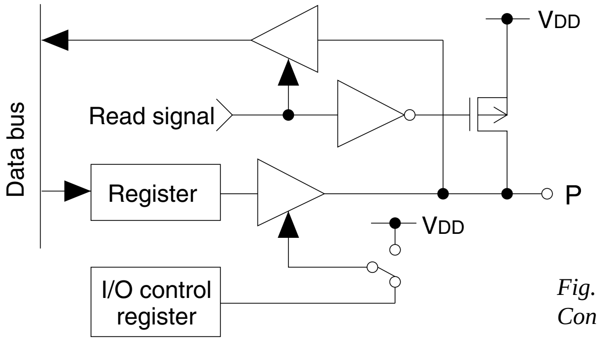


Fig. 3.3.7
Configuration of I/O port circuit

13 I/O port specification

```
*** OPTION  NO.13 ***

--- I/O PORT SPECIFICATION ---

        P00           1. C-MOS
                      2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 1↵

        P01           1. C-MOS
                      2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 1↵

        P02           1. C-MOS
                      2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        P03           1. C-MOS
                      2. Nch-Open Drain

PLEASE SELECT NO.(1) ? 2↵

        P00           1. C-MOS   SELECTED
        P01           1. C-MOS   SELECTED
        P02           2. Nch-Open Drain   SELECTED
        P03           2. Nch-Open Drain   SELECTED
```

Select output specification of any I/O port which is used as an output port. Either "Complementary (C-MOS)" output or "Nch Open Drain" output may be selected.
The circuit configuration of output driver is the same as that of output ports R00–R03, R10, R11 and R20.

Select "Complementary (C-MOS)" output for an unused I/O port.

14 R33 (REM) output port type

```
*** OPTION  NO.14 ***

--- R33 OUT PORT TYPE ---

        1. REM
        2. D.C.

PLEASE SELECT NO.(1) ? 2↵

        2. D.C.   SELECTED
```

Select output type of R33 (REM) output port. You have two choices, REM output or DC output. Either type may be selected.

Output specification is complementary output only. Select DC output if you are not using R33 (REM) output port.

- In case of selecting REM output
  Select this output type if you are using the port R33 to output transmission wave for remote controlling. You have two modes of control, soft timer and hard timer. Either mode can be selected by software.

  When using the soft timer mode, the port R33 (REM) is set in low level (VSS) output by setting "1" on REMC register and "0" on REMSO register. By setting "1" on REMSO register, you will have a carrier wave with amplitude VDD–VSS. Frequency division ratio from original oscillation (OSC3) and duty of the output wave are decided by the values set on RCDIV register and RCDUTY register respectively.
  If you are using hard timer mode, set "1" on REMC register as in soft timer mode and a value "1"–"3" on 2-bit ROUT register, and the carrier wave is output for a period of t-value. After expiration of the period, output wave will automatically return to low level (VSS).
  In this mode, the system outputs the carrier wave for a unit time t. The value is decided by setting "0" or "3" on 2-bit RT register. Frequency division ratio and duty are set on RCDIV register and RCDUTY register as in soft timer mode.

  Figures 3.3.8 and 3.3.9 show the output waveforms of two modes.

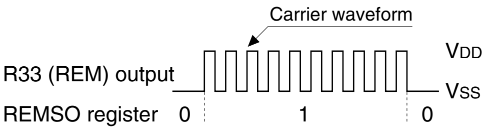


Fig. 3.3.8 Waveform of R33 REM output (soft timer mode)

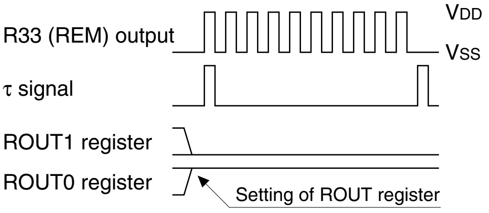


Fig. 3.3.9 Waveform of R33 REM output (hard timer mode)

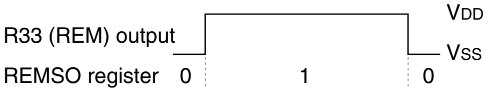


Fig. 3.3.10 Waveform of R33 DC output

- In case of selecting DC output

  If you set "0" on REMC register and "1" on REMSO register, the output from R33 (REM) port will be high level (VDD). If you set "0" on REMSO, R33 output will be low level (VSS). This output type is the same as that of R00–R03, R10, R11 and R20 output ports.

  Figure 3.3.10 shows the output waveform.

15 LCD common duty

```
*** OPTION  NO.15 ***

-- LCD COMMON DUTY ---

        1. 1/3 Duty
        2. 1/4 Duty

PLEASE SELECT NO.(1) ? 1↵

        1. 1/3 Duty   SELECTED
```

Table 3.3.1 Common duty selection standard

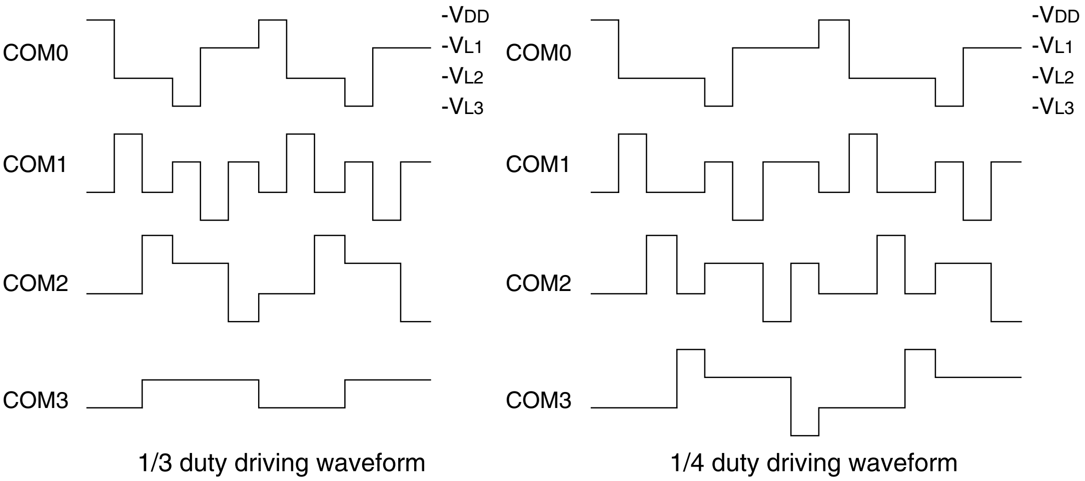
Number of LCD segment drives	Common duty
1–96	1/3
97–128	1/4

Select the liquid crystal segment driver duty. If you select 1/3 duty, 96 segments at the most can be driven with 3 COM terminals and 32 SEG terminals. If you select 1/4 duty, as many as 128 segments can be driven with 4 COM terminals and 32 SEG terminals.

In case of 1/3 duty, COM0–COM2 of COM terminals are valid, and COM3 always outputs OFF signal.

Figure 3.3.11 shows the driver waveform.

The table below gives the recommended criterion for driver duty selection.



1/3 duty driving waveform

1/4 duty driving waveform

Fig. 3.3.11 Drive waveform of COM terminals

3.4 FOG621A Quick Reference

■ Starting command and input/output files

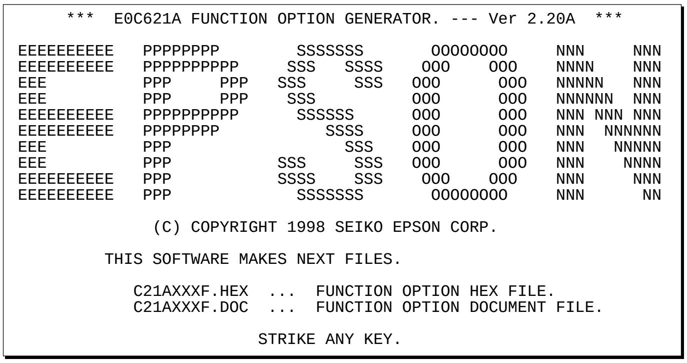
Execution file: FOG621A.EXE

Starting command: **FOG621A↵** ↵ *indicates the Return key.*

Input file: C21AXXXF.DOC (Function option document file, when modifying)

Output file: C21AXXXF.DOC (Function option document file)
C21AXXXF.HEX (Function option HEX file)

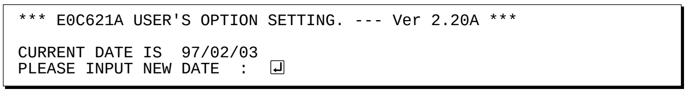
■ Display example



```
     ***  E0C621A FUNCTION OPTION GENERATOR. --- Ver 2.20A  ***

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 1998 SEIKO EPSON CORP.

         THIS SOFTWARE MAKES NEXT FILES.

            C21AXXXF.HEX  ...  FUNCTION OPTION HEX FILE.
            C21AXXXF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

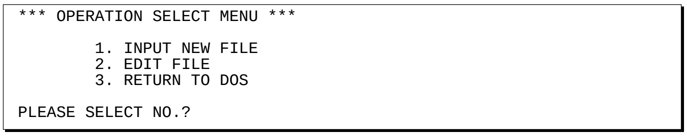
Start-up message
When FOG621A is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

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



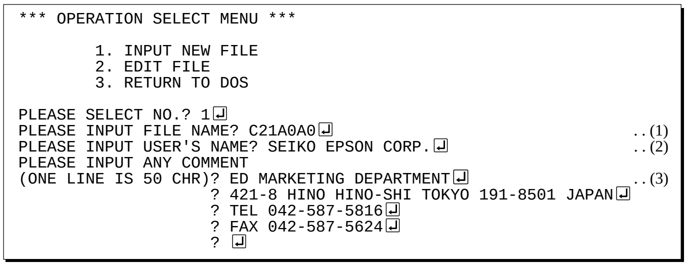
```
*** E0C621A USER'S OPTION SETTING. --- Ver 2.20A ***

CURRENT DATE IS  97/02/03
PLEASE INPUT NEW DATE  :  ↵
```

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.

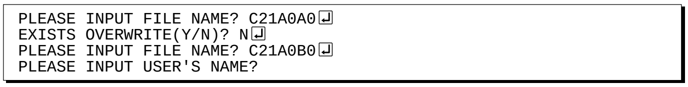


```
*** OPERATION SELECT MENU ***

        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 FOG621A.



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

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

PLEASE SELECT NO.? 1↵
PLEASE INPUT FILE NAME? C21A0A0↵                                    ..(1)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                        ..(2)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ED MARKETING DEPARTMENT↵                      ..(3)
                    ? 421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN↵
                    ? TEL 042-587-5816↵
                    ? FAX 042-587-5624↵
                    ? ↵
```



```
PLEASE INPUT FILE NAME? C21A0A0↵
EXISTS OVERWRITE(Y/N)? N↵
PLEASE INPUT FILE NAME? C21A0B0↵
PLEASE INPUT USER'S NAME?
```

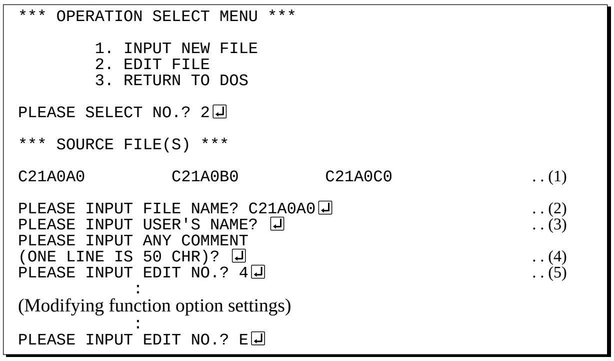
Setting new function options
Select "1" on the operation selection menu.
(1) Enter the file name.
(2) Enter the customer's company name.
(3) Enter any comment.
(Within 50 characters x 10 lines)
Next, start function option setting from option No. 1.

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

Modifying function option settings

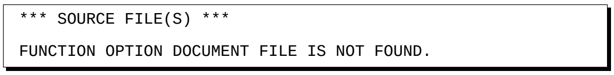
Select "2" on the operation selection menu.

(1) Will display the files on the current drive.
(2) Enter the file name.
(3) Enter the customer's company name.
(4) Enter any comment.
Previously entered data can be used by pressing the RETURN key "↵" at (3) and (4).
(5) Enter the number of the function option to be modified. When selection of one option is complete, the system prompts entry of another function option number. Repeat selection until all options to be modified are selected.
Enter "E↵" to end option setting. Then, move to the confirmation procedure for HEX file generation.



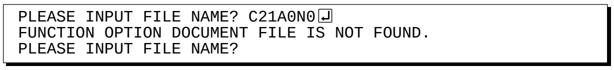
```
*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.? 2↵

*** SOURCE FILE(S) ***

C21A0A0         C21A0B0         C21A0C0                ..(1)

PLEASE INPUT FILE NAME? C21A0A0↵                       ..(2)
PLEASE INPUT USER'S NAME? ↵                            ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ↵                                ..(4)
PLEASE INPUT EDIT NO.? 4↵                              ..(5)
            :
(Modifying function option settings)
            :
PLEASE INPUT EDIT NO.? E↵
```

In step (1), if no modifiable source exists, the following message is displayed and the FOG621A program will be terminated.



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

FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

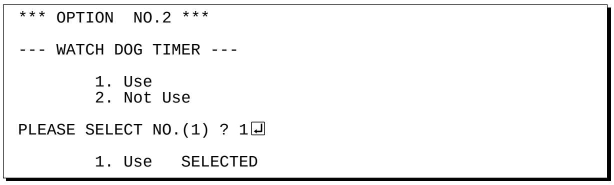
In step (2), if the function option document file is not in the current drive, the following message is displayed, prompting entry of other file name.



```
PLEASE INPUT FILE NAME? C21A0N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
PLEASE INPUT FILE NAME?
```

Option selection

The selections for each option correspond one to one to the option list. Enter the selection number. The value in parentheses ( ) indicates the default value, and is set when only the RETURN key "↵" is pressed.
In return, the confirmation is displayed.
When you wish to modify previously set function options in the new setting process, enter "B↵" to return 1 step back to the previous function option setting operation.



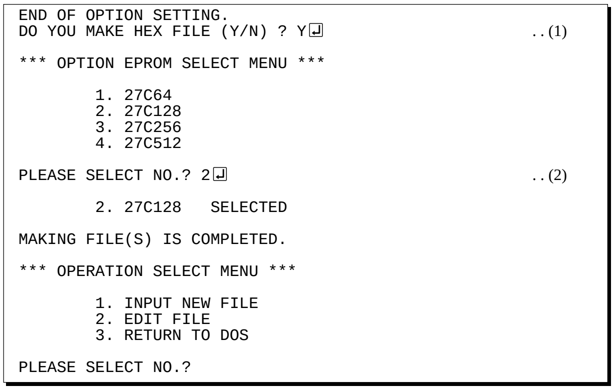
```
*** OPTION  NO.2 ***

--- WATCH DOG TIMER ---

        1. Use
        2. Not Use

PLEASE SELECT NO.(1) ? 1↵

        1. Use   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 EVA621AR, 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 EVA621AR options.

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



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

*** OPTION EPROM SELECT MENU ***

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

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

        2. 27C128   SELECTED

MAKING FILE(S) IS COMPLETED.

*** OPERATION SELECT MENU ***

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

PLEASE SELECT NO.?
```

3.5 Sample Files

■ Example of function option document file

```
* E0C621A FUNCTION OPTION DOCUMENT V 2.20A
*
* FILE NAME    C21A0A0F.DOC
* USER'S NAME  SEIKO EPSON CORP.
* INPUT DATE   1998/10/02
* COMMENT      ED MARKETING DEPARTMENT
*              421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN
*              TEL 042-587-5816
*              FAX 042-587-5624
*
*
* OPTION NO.1
* < OSC 3 SYSTEM CLOCK >
*     CERAMIC  ----------------   SELECTED
 OPT01 01
*
*
* OPTION NO.2
* < WATCH DOG TIMER >
*     USE ---------------------   SELECTED
 OPT02 01
*
*
* OPTION NO.3
* < INTERRUPT (K00-K03) >
*     K00,K01,K02,K03  --------   SELECTED
 OPT03 04
*
*
* OPTION NO.4
* < INTERRUPT (K10-K13) >
*     K10,K11,K12,K13  --------   SELECTED
 OPT04 04
*
*
* OPTION NO.5
* < INTERRUPT NOISE REJECTOR (K10-K13) >
*     USE  --------------------   SELECTED
 OPT05 01
*
*
* OPTION NO.6
* < IN 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
 OPT06 01 02 03 04 05 06 07 08
*
*
* OPTION NO.7
* < OUT PORT SPECIFICATION >
*     R00  C-MOS  -------------   SELECTED
*     R01  C-MOS  -------------   SELECTED
*     R02  C-MOS  -------------   SELECTED
*     R03  C-MOS  -------------   SELECTED
*     R10  C-MOS  -------------   SELECTED
*     R11  C-MOS  -------------   SELECTED
*     R20  C-MOS  -------------   SELECTED
 OPT07 01 03 05 07 09 11 13
*
*
```

```
* OPTION NO.8
* < R12 OUT PORT SPECIFICATION >
*     C-MOS  ------------------   SELECTED
 OPT08 01
*
*
* OPTION NO.9
* < R12 OUT PORT TYPE >
*     D.C.  -------------------   SELECTED
 OPT09 01
*
*
* OPTION NO.10
* < R13 OUT PORT SPECIFICATION >
*   C-MOS  --------------------   SELECTED
 OPT10 01
*
*
* OPTION NO.11
* < R13 OUT PORT TYPE >
*     D.C.  -------------------   SELECTED
 OPT11 01
*
*
* OPTION NO.12
* < I/O PORT FUNCTION >
*     P00  I/O PORT  ----------   SELECTED
*     P01  I/O PORT  ----------   SELECTED
*     P02  I/O PORT  ----------   SELECTED
*     P03  I/O PORT  ----------   SELECTED
 OPT12 01 03 05 07
*
*
* OPTION NO.13
* < I/O PORT SPECIFICATION >
*     P00  C-MOS  -------------   SELECTED
*     P01  C-MOS  -------------   SELECTED
*     P02  C-MOS  -------------   SELECTED
*     P03  C-MOS  -------------   SELECTED
 OPT13 01 03 05 07
*
*
* OPTION NO.14
* < R33 OUT PORT TYPE >
*     REM  --------------------   SELECTED
 OPT14 01
*
*
* OPTION NO.15
* < LCD COMMON DUTY >
*     1/3 DUTY  ---------------   SELECTED
 OPT15 01
*
*
* SEIKO EPSON’S AREA
*
 OPT16
*
*
 OPT17
*
*
 OPT18
*
*
 OPT19
*
*
 OPT20
*
*
```

```
 OPT21
*
*
 OPT22
*
*
 OPT23
*
*
 OPT24
*
*
 OPT25 01 02 03 04
*
*
 OPT26
*
*
 OPT27
*
*
 OPT28
*
*
 OPT29
\\END
```

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

4 SEGMENT OPTION GENERATOR SOG621A

4.1 SOG621A Outline

With the 4-bit single-chip E0C621A microcomputers, the customer may select the LCD segment options. By modifying the mask patterns of the E0C621A according to the selected options, the system can be customized to meet the specifications of the target system.
The Segment Option Generator SOG621A is a software tool for generating data file used to generate mask patterns. From the data file created with SOG621A, the E0C621A mask pattern is automatically generated by a general purpose computer.

The HEX file for the evaluation board (EVA621AR) segment option ROM is simultaneously generated with the data file.

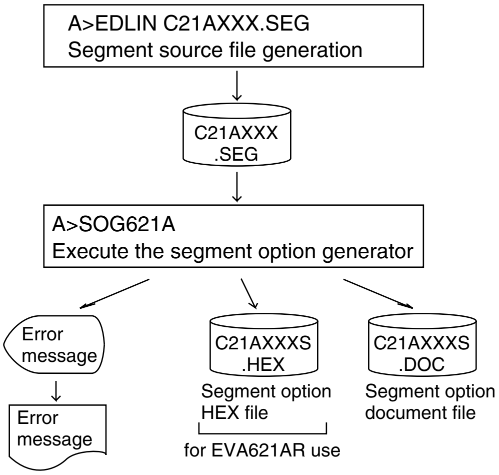


Fig. 4.1.1 SOG621A execution flow

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

4.2 Option List

TERMINAL NAME	ADDRESS												OUTPUT SPECIFICATION
	COM0			COM1			COM2			COM3			
	H	L	D	H	L	D	H	L	D	H	L	D	
SEG0													SEG output
SEG1													DC output □ C □ N
SEG2													SEG output
SEG3													DC output □ C □ N
SEG4													SEG output
SEG5													DC output □ C □ N
SEG6													SEG output
SEG7													DC output □ C □ N
SEG8													SEG output
SEG9													DC output □ C □ N
SEG10													SEG output
SEG11													DC output □ C □ N
SEG12													SEG output
SEG13													DC output □ C □ N
SEG14													SEG output
SEG15													DC output □ C □ N
SEG16													SEG output
SEG17													DC output □ C □ N
SEG18													SEG output
SEG19													DC output □ C □ N
SEG20													SEG output
SEG21													DC output □ C □ N
SEG22													SEG output
SEG23													DC output □ C □ N
SEG24													SEG output
SEG25													DC output □ C □ N
SEG26													SEG output
SEG27													DC output □ C □ N
SEG28													SEG output
SEG29													DC output □ C □ N
SEG30													SEG output
SEG31													DC output □ C □ N

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

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

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

4.3 Segment Ports Output Specifications

For the output specification of the segment output ports SEG0–SEG31, segment output and DC output can be selected in units of two terminals. When used for liquid crystal panel drives, select segment output; when used as regular output port, select DC output. When DC output is selected, either complementary output or Nch open drain output may further be selected.
However, for segment output ports that will not be used, select segment output.
Refer to the "E0C62 Family Development Tool Reference Manual" for the segment option source file creation.

■ When segment output is selected

The segment output port has a segment decoder built-in, and the data bit of the optional address in the segment memory area (0D0H–0EFH) can be allocated to the optional segment.
The segment memory may be allocated only one segment and multiple setting is not possible.
The allocated segment displays when the bit for this segment memory is set to "1", and goes out when bit is set to "0".
Segment allocation is set to H for high address (D, E), to L for low address (0–F), and to D for data bit (0–3) and are recorded in their respective column in the option list. For segment ports that will not be used, write "---" (hyphen) in the H, L, and D columns of COM0–COM3.

Examples

- When 1/4 duty is selected

```
0  D00  D01  D02  D03  S
1  D10  D11  D12  D13  S
```

- When 1/3 duty is selected

```
0  D00  D01  D02  ---  S
1  D10  D11  D12  ---  S
```

■ When DC output is selected

The DC output can be selected in units of two terminals and up to 32 terminals may be allocated for DC output. Also, either complementary output or Nch open drain output is likewise selected in units of two terminals. When the bit for the selected segment memory is set to "1", the segment output port goes high (VDD), and goes low (VSS) when set to "0". Segment allocation is the same as when segment output is selected but for the while the segment memory allocated to COM1–COM3 becomes ineffective. Write three hyphens ("---") in the COM1–COM3 columns in the option list.

Example

- When complementary output is set to SEG16 and SEG17, and Nch open drain output is set to SEG18 and SEG19.

```
16  D00  ---  ---  ---  C
17  D10  ---  ---  ---  C
18  D20  ---  ---  ---  N
19  D30  ---  ---  ---  N
```

4.4 SOG621A Quick Reference

■ Starting command and input/output files

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

Execution file: SOG621A.EXE

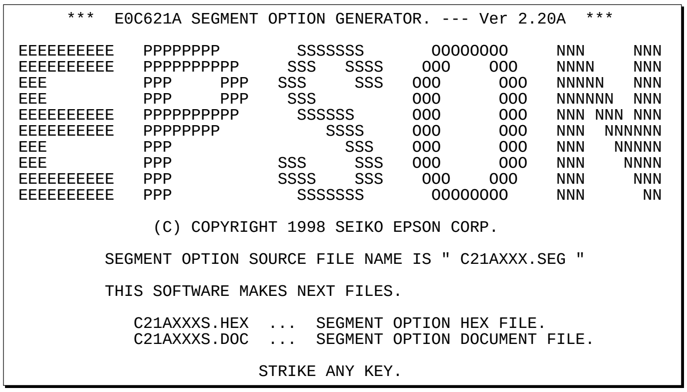
Starting command: **SOG621A_ [-H]↵**

Option: -H: Specifies the segment option document file for input file of SOG621A.

Input file: C21AXXX.SEG (Segment option source file)
C21AXXXS.DOC (Segment option document file, when -H option use)

Output file: C21AXXXS.DOC (Segment option document file)
C21AXXXS.HEX (Segment option HEX file)

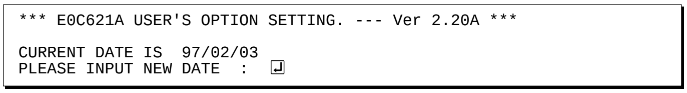
■ Display example



```
     ***  E0C621A SEGMENT OPTION GENERATOR. --- Ver 2.20A  ***

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 1998 SEIKO EPSON CORP.

         SEGMENT OPTION SOURCE FILE NAME IS " C21AXXX.SEG "

         THIS SOFTWARE MAKES NEXT FILES.

            C21AXXXS.HEX  ...  SEGMENT OPTION HEX FILE.
            C21AXXXS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                         STRIKE ANY KEY.
```

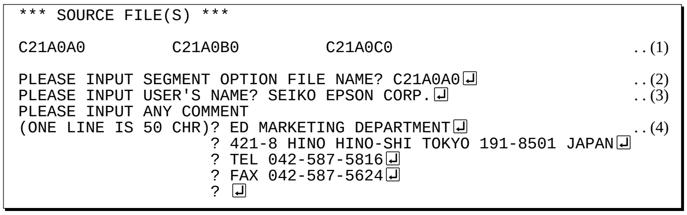
Start-up message
When SOG621A is started, the start-up message is displayed.
For "STRIKE ANY KEY.", press any key to advance the program execution.

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

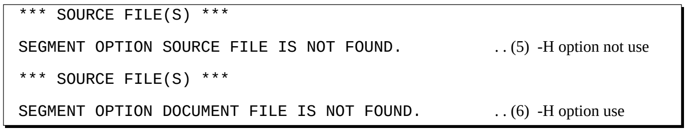


```
*** E0C621A USER'S OPTION SETTING. --- Ver 2.20A ***

CURRENT DATE IS  97/02/03
PLEASE INPUT NEW DATE  :  ↵
```

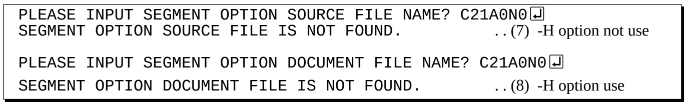
Date input
Enter the 2-digit year, month, and day of the month by delimiting them with a slash ("/"). When not modifying the date, press the RETURN key "↵" to continue.



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

C21A0A0         C21A0B0         C21A0C0                          ..(1)

PLEASE INPUT SEGMENT OPTION FILE NAME? C21A0A0↵                  ..(2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                     ..(3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? ED MARKETING DEPARTMENT↵                   ..(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? C21A0N0↵
SEGMENT OPTION SOURCE FILE IS NOT FOUND.          ..(7)  -H option not use

PLEASE INPUT SEGMENT OPTION DOCUMENT FILE NAME? C21A0N0↵
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 EVA621AR, 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 EVA621AR options.

When a series of operations are complete, the SOG621A generates files. If no error is committed while setting segment options, "MAKING FILE IS COMPLETED" will be displayed and the SOG621A program will be terminated.

■ Error messages

Error message	Explanation
S (Syntax Error)	The data was written in an invalid format.
N (Segment No. Select Error)	The segment number outside the specificable range was specified.
R (RAM Address Select Error)	The segment memory address or data bit outside the specificable range was specified.
D (Duprication Error)	The same data (SEG port No., segment memory address, or data bit) was specified more then once.
Out Port Set Error	The output specifications were not set in units of two ports.

4.5 Sample Files

■ Example of input file

```
0       D02     E80     E81     E82     S
1       D01     E90     E91     E92     S
2       D03     EA0     EA1     EA2     S
3       D41     EB0     EB1     EB2     S
4       D40     EC0     EC1     EC2     S
5       DA1     DA2     DA0     D93     S
6       D90     D91     D92     EB3     S
7       EE0     DB1     DB0     DB2     S
8       ED2     ED3     DE0     DD0     S
9       E71     E72     E70     E63     S
10      E60     E61     E62     EA3     S
11      E51     E52     E50     E43     S
12      E40     E41     E42     DC0     S
13      D33     D32     D30     D31     S
14      D13     D12     D10     D11     S
15      D00     E83     EF0     D20     S
16      D81     D82     D80     D73     S
17      D70     D71     D72     D42     S
18      D61     D62     D60     D53     S
19      D50     D51     D52     E93     S
20      E00     E01     E02     ED1     S
21      E11     E12     E10     E03     S
22      E20     E21     E22     ED0     S
23      E31     E32     E30     E23     S
24      E13     E33     E53     EC3     S
25      DD1     DD2     DE1     DE2     S
26      EF2     ---     ---     ---     C
27      EF3     ---     ---     ---     C
28      DF0     ---     ---     ---     N
29      DF1     ---     ---     ---     N
30      DF2     ---     ---     ---     N
31      DF3     ---     ---     ---     N
```

■ Example of output file

```
* E0C621A SEGMENT OPTION DOCUMENT  V 2.20A
*
* FILE NAME    C21A0A0S.DOC
* USER’S NAME  SEIKO EPSON CORP.
* INPUT DATE   98/10/05
* COMMENT      ED MARKETING DEPARTMENT
*              421-8 HINO HINO-SHI TOKYO 191-8501 JAPAN
*              TEL 042-587-5816
*              FAX 042-587-5624
*
*
* OPTION NO.16
*
* < LCD SEGMENT DECODE TABLE >
*
* SEG COM0 COM1 COM2 COM3 SPEC
*
   0  D02  E80  E81  E82  S
   1  D01  E90  E91  E92  S
   2  D03  EA0  EA1  EA2  S
   3  D41  EB0  EB1  EB2  S
   4  D40  EC0  EC1  EC   S
   5  DA1  DA2  DA0  D93  S
   6  D90  D91  D92  EB3  S
   7  EE0  DB1  DB0  DB2  S
   8  ED2  ED3  DE0  DD0  S
   9  E71  E72  E70  E63  S
  10  E60  E61  E62  EA3  S
  11  E51  E52  E50  E43  S
  12  E40  E41  E42  DC0  S
  13  D33  D32  D30  D31  S
  14  D13  D12  D10  D11  S
  15  D00  E83  EF0  D20  S
  16  D81  D82  D80  D73  S
  17  D70  D71  D72  D42  S
  18  D61  D62  D60  D53  S
  19  D50  D51  D52  E93  S
  20  E00  E01  E02  ED1  S
  21  E11  E12  E10  E03  S
  22  E20  E21  E22  ED0  S
  23  E31  E32  E30  E23  S
  24  E13  E33  E53  EC3  S
  25  DD1  DD2  DE1  DE2  S
  26  EF2  D21  D22  D23  C
  27  EF3  D43  D63  D83  C
  28  DF0  DA3  DB3  DC1  N
  29  DF1  DC2  DC3  DD3  N
  30  DF2  DE3  E73  EE1  N
  31  DF3  EE2  EE3  EF1  N
\\END
```

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

5 ICE CONTROL SOFTWARE ICS621A

5.1 ICS621A Outline

The In-circuit Emulator ICE6200 (ICE62R) connects the target board produced by the user via the EVA621AR and performs real time target system evaluation and debugging by passing through the RS-232C from the host computer and controlling it. The operation on the host computer side and ICE6200 (ICE62R) control is done through the ICE Control Software ICS621A.
The ICS621A has a set of numerous and highly functional emulation commands which provide sophisticated break function, on-the-fly data display, history display, etc., and so perform a higher level of debugging.

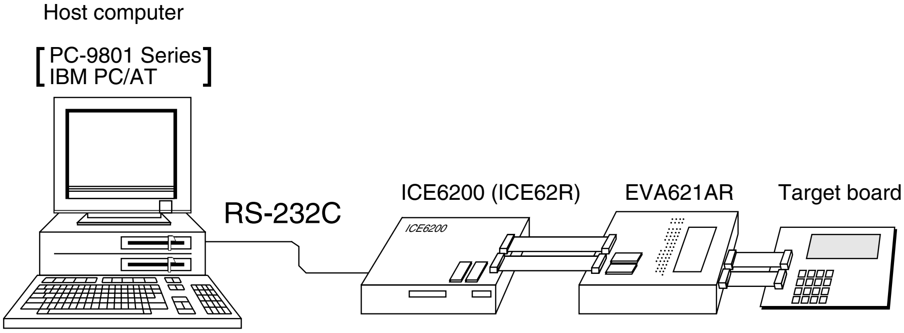


Fig. 5.1.1 Debugging system using ICE6200 (ICE62R)

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

5.2 ICS621A Restrictions

Take the following precautions when using the ICS621A.

■ ROM Area

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

■ RAM Area

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

Memory 0D0H to 0EFH is display memory; 0F0H to 0FFH is I/O memory.
(Refer to the "E0C621A Technical Manual" for details.)

■ Undefined Code

The instructions below are not specified for the E0C621A and so cannot be used.

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

■ OPTLD Command

In the ICS621A, OPTLD command can be used.
This command loads HEX files (function option and LCD segment option data) into the EVA621AR memory via the ICE6200 (ICE62R).

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 EVA621AR 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 EVA621AR segment option data memory. It is HEX file output by the segment option generator and has intel HEX format.

Examples

`#OPTLD,1,C21AXXX↵` ..... C21AXXXF.HEX file is loaded in the function option data memory.

`#OPTLD,2,C21AXXX↵` ..... C21AXXXS.HEX file is loaded in the segment option data memory.

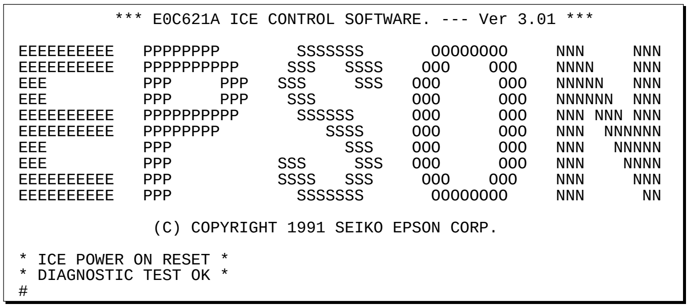
5.3 ICS621A Quick Reference

■ Starting command and input/output files

↵ *indicates the Return key.*

Execution file:	ICS621A.BAT	(ICS621AJ.EXE)	. . . for MS-DOS
	ICS621AB.BAT	(ICS621AW.EXE)	. . . for PC-DOS
Starting command:	**ICS621A (ICS621AJ)↵**		. . . for MS-DOS
	ICS621AB (ICS621AW)↵		. . . for PC-DOS
Input file:	C21AXXXL.HEX	(Object file, low-order)	
	C21AXXXH.HEX	(Object file, high-order)	
	C21AXXXD.HEX	(Data RAM file)	
	C21AXXXC.HEX	(Control file)	
Output file:	C21AXXXL.HEX	(Object file, low-order)	
	C21AXXXH.HEX	(Object file, high-order)	
	C21AXXXD.HEX	(Data RAM file)	
	C21AXXXC.HEX	(Control file)	

■ Display example



```
          *** E0C621A 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 ICS621A is started, the start-up message is displayed, and a self-test is automatically performed. ICS621A 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 ICS621A program is terminated by entering the Q (Quit) command.

Note *Confirm that the cables connected properly, then operate the ICS621A.*

■ Error messages

Error message	Meaning	Recover procedure
* COMMUNICATION ERROR OR ICE NOT READY *	ICE6200 (ICE62R) is disconnected or power is OFF.	Switch OFF the host power supply, connect cable, and reapply power. Or switch ON power to ICE6200 (ICE62R).
* TARGET DOWN (1) *	Evaluation board is disconnected. (Check at power ON)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200 (ICE62R).
* TARGET DOWN (2) *	Evaluation board is disconnected. (Check at command execution)	Switch OFF power to ICE, and connect the evaluation board. Then, apply power to ICE6200 (ICE62R).
* UNDEFINED PROGRAM CODE EXIST *	Undefined code is detected in the program loaded from ROM or FD.	Convert ROM and FD data with the cross assembler, then restart the ICE6200 (ICE62R).
* COMMAND ERROR *	A miss occurs by command input.	Reenter the proper command.
(No response after power on)	The ICE-to-HOST cable is disconnected on the host side.	Switch OFF the host power supply, connect cable, and reapply power.

■ Command list

Item No.	Function	Command Format	Outline of Operation
1	Assemble	#A,a↵	Assemble command mnemonic code and store at address "a"
2	Disassemble	#L,a1,a2 ↵	Contents of addresses a1 to a2 are disassembled and displayed
3	Dump	#DP,a1,a2 ↵	Contents of program area a1 to a2 are displayed
		#DD,a1,a2 ↵	Content of data area a1 to a2 are displayed
4	Fill	#FP,a1,a2,d ↵	Data d is set in addresses a1 to a2 (program area)
		#FD,a1,a2,d↵	Data d is set in addresses a1 to a2 (data area)
5	Set Run Mode	#G,a↵	Program is executed from the "a" address
		#TIM↵	Execution time and step counter selection
		#OTF↵	On-the-fly display selection
6	Trace	#T,a,n ↵	Executes program while displaying results of step instruction from "a" address
		#U,a,n ↵	Displays only the final step of #T,a,n
7	Break	#BA,a↵ #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 EVA621AR 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 EVA621AR CPU internal registers
		#SR↵	Set EVA621AR CPU internal registers
		#I ↵	Reset EVA621AR 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 (ICE62R) set condition from file
		#CS,file↵	Save ICE6200 (ICE62R) 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 (ICE62R) instruction
17	Self Diagnosis	#CHK↵	Report results of ICE6200 (ICE62R) self diagnostic test

↵ means press the RETURN key.

6 MASK DATA CHECKER MDC621A

6.1 MDC621A Outline

The Mask Data Checker MDC621A is a software tool which checks the program data (C21AXXXH.HEX and C21AXXXL.HEX) and option data (C21AXXXF.DOC and C21AXXXS.DOC) created by the user and creates the data file (C621AXXX.PAn) for generating mask patterns. The user must send the file generated through this software tool to Seiko Epson.

Moreover, MDC621A has the capability to restore the generated data file (C621AXXX.PA0) to the original file format.

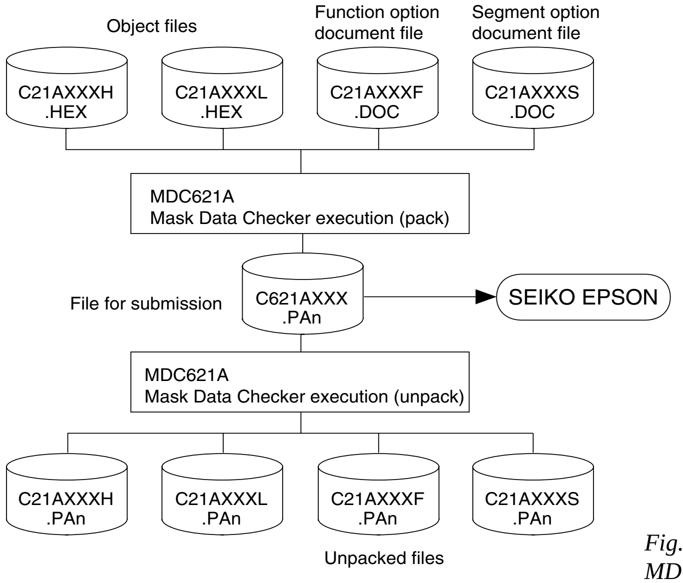


Fig. 6.1.1
MDC621A execution flow

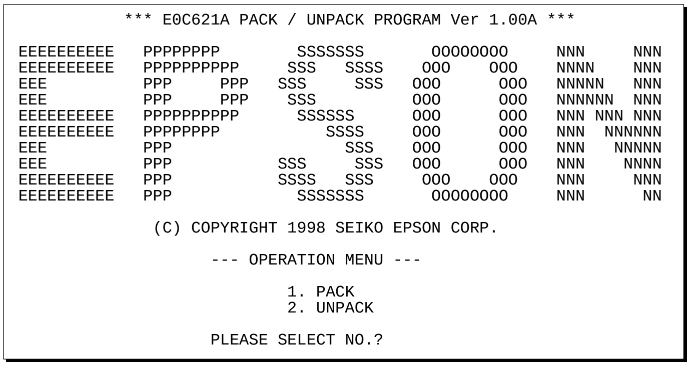
☞ *The operating method is same as for the E0C62 Family. Refer to the "E0C62 Family Development Tool Reference Manual" for details.*

6.2 MDC621A Quick Reference

■ Starting command and input/output files

Execution file:	MDC621A.EXE		
Starting command:	**MDC621A↵**		*↵ indicates the Return key.*
Input file:	C21AXXXL.HEX	(Object file, low-order)	When packing
	C21AXXXH.HEX	(Object file, high-order)	
	C21AXXXF.DOC	(Function option document file)	
	C21AXXXS.DOC	(Segment option document file)	
	C621AXXX.PAn	(Packed file)	When unpacking
Output file:	C621AXXX.PAn	(Packed file)	When packing
	C21AXXXL.PAn	(Object file, low-order)	When unpacking
	C21AXXXH.PAn	(Object file, high-order)	
	C21AXXXF.PAn	(Function option document file)	
	C21AXXXS.PAn	(Segment option document file)	

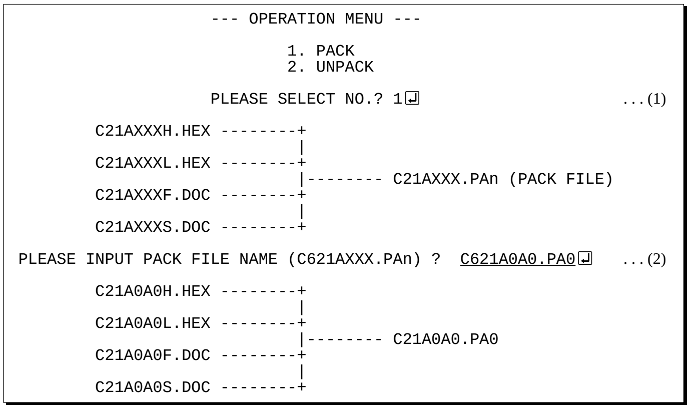
■ Display examples



```
           *** E0C621A PACK / UNPACK PROGRAM Ver 1.00A ***

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 1998 SEIKO EPSON CORP.

                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

                    PLEASE SELECT NO.?
```

Start-up message

When MDC621A 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)

        C21AXXXH.HEX --------+
                             |
        C21AXXXL.HEX --------+
                             |-------- C21AXXX.PAn (PACK FILE)
        C21AXXXF.DOC --------+
                             |
        C21AXXXS.DOC --------+

PLEASE INPUT PACK FILE NAME (C621AXXX.PAn) ?  C621A0A0.PA0↵    ...(2)

        C21A0A0H.HEX --------+
                             |
        C21A0A0L.HEX --------+
                             |-------- C21A0A0.PA0
        C21A0A0F.DOC --------+
                             |
        C21A0A0S.DOC --------+
```

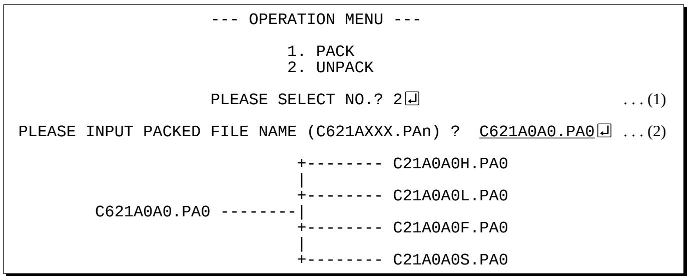
Packing of data

(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 "C621AXXX.PA0" has been submitted, the pack file name should be entered as "C621AXXX.PA1".)

With this, the mask file (C621AXXX.PAn) is generated, and the MDC621A 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 (ASM621A) as program data.



```
                    --- OPERATION MENU ---

                            1. PACK
                            2. UNPACK

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

PLEASE INPUT PACKED FILE NAME (C621AXXX.PAn) ?  C621A0A0.PA0↵ ...(2)

                             +-------- C21A0A0H.PA0
                             |
                             +-------- C21A0A0L.PA0
        C621A0A0.PA0 --------|
                             +-------- C21A0A0F.PA0
                             |
                             +-------- C21A0A0S.PA0
```

Unpacking of data

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

(2) Enter the packed file name.

With this, the mask data file (C621AXXX.PAn) is restored to the original file format, and the MDC621A program will be terminated.

Since the extension of the file name remains as "PAn", it must be renamed back to its original form ("HEX" and "DOC") in order to re-debug or modify the restored file.

■ Error messages

Program data error

Error Message	Explanation
1. HEX DATA ERROR : NOT COLON.	There is no colon.
2. HEX DATA ERROR : DATA LENGTH. (NOT 00-20h)	The data length of 1 line is not in the 00–20H range.
3. HEX DATA ERROR : ADDRESS.	The address is beyond the valid range of the program ROM.
4. HEX DATA ERROR : RECORD TYPE. (NOT 00)	The record type of 1 line is not 00.
5. HEX DATA ERROR : DATA. (NOT 00-FFh)	The data is not in the range between 00H and 0FFH.
6. HEX DATA ERROR : TOO MANY DATA IN ONE LINE.	There are too many data in 1 line.
7. HEX DATA ERROR : CHECK SUM.	The checksum is not correct.
8. HEX DATA ERROR : END MARK.	The end mark is not : 00000001FF.
9. HEX DATA ERROR : DUPLICATE.	There is duplicate definition of data in the same address.

Function option data error

Error Message	Explanation
1. OPTION DATA ERROR : START MARK.	The start mark is not "\OPTION". (during unpacking) *
2. OPTION DATA ERROR : OPTION NUMBER.	The option number is not correct.
3. OPTION DATA ERROR : SELECT NUMBER.	The option selection number is not correct.
4. OPTION DATA ERROR : END MARK.	The end mark is not "\\END" (packing) or "\END" (unpacking).*

Segment option data error

Error Message	Explanation
1. SEGMENT DATA ERROR : START MARK.	The start mark is not "\SEGMENT". (during unpacking) *
2. SEGMENT DATA ERROR : DATA.	The segment data is not correct.
3. SEGMENT DATA ERROR : SEGMENT NUMBER.	The SEG No. is not correct.
4. SEGMENT DATA ERROR : SPEC.	The output specification of the SEG terminal is not correct.
5. SEGMENT DATA ERROR : END MARK.	The end mark is not "\\END" (packing) or "\END" (unpacking).*

File error

Error Message	Explanation
1. <File_name> FILE IS NOT FOUND.	The file is not found or the file number set in CONFIG.SYS is less than 10.
2. PACK FILE NAME (File_name) ERROR.	The packed input format for the file name is wrong.
3. PACKED FILE NAME (File_name) ERROR.	The unpacked input format for the file name is wrong.

System error

Error Message	Explanation
1. DIRECTORY FULL.	The directory is full.
2. DISK WRITE ERROR.	Writing on the disk is failed.

** \ sometimes appears as ¥, depending on the personal computer being used.*

APPENDIX A. E0C621A 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
		XH, r	1	1	1	0	1	0	0	0	0	1	r1	r0					5	XH←r
		XL, r	1	1	1	0	1	0	0	0	1	0	r1	r0					5	XL←r
		YH, r	1	1	1	0	1	0	0	1	0	1	r1	r0					5	YH←r
		YL, r	1	1	1	0	1	0	0	1	1	0	r1	r0					5	YL←r
		r, XH	1	1	1	0	1	0	1	0	0	1	r1	r0					5	r←XH
		r, XL	1	1	1	0	1	0	1	0	1	0	r1	r0					5	r←XL
		r, YH	1	1	1	0	1	0	1	1	0	1	r1	r0					5	r←YH
		r, YL	1	1	1	0	1	0	1	1	1	0	r1	r0					5	r←YL
	ADC	XH, i	1	0	1	0	0	0	0	0	i3	i2	i1	i0			↕	↕	7	XH←XH+i3~i0+C
		XL, i	1	0	1	0	0	0	0	1	i3	i2	i1	i0			↕	↕	7	XL←XL+i3~i0+C
		YH, i	1	0	1	0	0	0	1	0	i3	i2	i1	i0			↕	↕	7	YH←YH+i3~i0+C
		YL, i	1	0	1	0	0	0	1	1	i3	i2	i1	i0			↕	↕	7	YL←YL+i3~i0+C

Classification	Mne-monic	Operand	Operation Code												Flag				Clock	Operation
			B	A	9	8	7	6	5	4	3	2	1	0	I	D	Z	C		
Index operation instructions	CP	XH, i	1	0	1	0	0	1	0	0	i3	i2	i1	i0			↕	↕	7	XH-i3~i0
		XL, i	1	0	1	0	0	1	0	1	i3	i2	i1	i0			↕	↕	7	XL-i3~i0
		YH, i	1	0	1	0	0	1	1	0	i3	i2	i1	i0			↕	↕	7	YH-i3~i0
		YL, i	1	0	1	0	0	1	1	1	i3	i2	i1	i0			↕	↕	7	YL-i3~i0
Data transfer instructions	LD	r, i	1	1	1	0	0	0	r1	r0	i3	i2	i1	i0					5	r←i3~i0
		r, q	1	1	1	0	1	1	0	0	r1	r0	q1	q0					5	r←q
		A, Mn	1	1	1	1	1	0	1	0	n3	n2	n1	n0					5	A←M(n3~n0)
		B, Mn	1	1	1	1	1	0	1	1	n3	n2	n1	n0					5	B←M(n3~n0)
		Mn, A	1	1	1	1	1	0	0	0	n3	n2	n1	n0					5	M(n3~n0)←A
		Mn, B	1	1	1	1	1	0	0	1	n3	n2	n1	n0					5	M(n3~n0)←B
	LDPX	MX, i	1	1	1	0	0	1	1	0	i3	i2	i1	i0					5	M(X)←i3~i0, X←X+1
		r, q	1	1	1	0	1	1	1	0	r1	r0	q1	q0					5	r←q, X←X+1
	LDPY	MY, i	1	1	1	0	0	1	1	1	i3	i2	i1	i0					5	M(Y)←i3~i0, Y←Y+1
		r, q	1	1	1	0	1	1	1	1	r1	r0	q1	q0					5	r←q, Y←Y+1
	LBPX	MX, *l*	1	0	0	1	*l*7	*l*6	*l*5	*l*4	*l*3	*l*2	*l*1	*l*0					5	M(X)←*l*3~*l*0, M(X+1)←*l*7~*l*4, X←X+2
Flag operation instructions	SET	F, i	1	1	1	1	0	1	0	0	i3	i2	i1	i0	↑	↑	↑	↑	7	F←F∨i3~i0
	RST	F, i	1	1	1	1	0	1	0	1	i3	i2	i1	i0	↓	↓	↓	↓	7	F←F∧i3~i0
	SCF		1	1	1	1	0	1	0	0	0	0	0	1				↑	7	C←1
	RCF		1	1	1	1	0	1	0	1	1	1	1	0				↓	7	C←0
	SZF		1	1	1	1	0	1	0	0	0	0	1	0			↑		7	Z←1
	RZF		1	1	1	1	0	1	0	1	1	1	0	1			↓		7	Z←0
	SDF		1	1	1	1	0	1	0	0	0	1	0	0		↑			7	D←1 (Decimal Adjuster ON)
	RDF		1	1	1	1	0	1	0	1	1	0	1	1		↓			7	D←0 (Decimal Adjuster OFF)
	EI		1	1	1	1	0	1	0	0	1	0	0	0	↑				7	I ←1 (Enables Interrupt)
	DI		1	1	1	1	0	1	0	1	0	1	1	1	↓				7	I ←0 (Disables Interrupt)
Stack operation instructions	INC	SP	1	1	1	1	1	1	0	1	1	0	1	1					5	SP← SP+1
	DEC	SP	1	1	1	1	1	1	0	0	1	0	1	1					5	SP← SP-1
	PUSH	r	1	1	1	1	1	1	0	0	0	0	r1	r0					5	SP← SP-1, M(SP)←r
		XH	1	1	1	1	1	1	0	0	0	1	0	1					5	SP← SP-1, M(SP)←XH
		XL	1	1	1	1	1	1	0	0	0	1	1	0					5	SP← SP-1, M(SP)←XL
		YH	1	1	1	1	1	1	0	0	1	0	0	0					5	SP← SP-1, M(SP)←YH
		YL	1	1	1	1	1	1	0	0	1	0	0	1					5	SP← SP-1, M(SP)←YL
		F	1	1	1	1	1	1	0	0	1	0	1	0					5	SP← SP-1, M(SP)←F
	POP	r	1	1	1	1	1	1	0	1	0	0	r1	r0					5	r←M(SP), SP←SP+1
		XH	1	1	1	1	1	1	0	1	0	1	0	1					5	XH← M(SP), SP←SP+1
		XL	1	1	1	1	1	1	0	1	0	1	1	0					5	XL← M(SP), SP←SP+1

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

Abbreviations used in the explanations have the following meanings.

Symbols associated with registers and memory

A	A register
B	B register
X	XHL register (low order eight bits of index register IX)
Y	YHL register (low order eight bits of index register IY)
XH	XH register (high order four bits of XHL register)
XL	XL register (low order four bits of XHL register)
YH	YH register (high order four bits of YHL register)
YL	YL register (low order four bits of YHL register)
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. E0C621A RAM MAP

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																

PROGRAM NAME:

P	H	L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	8	NAME																
		MSB																
		LSB																
	9	NAME																
		MSB																
		LSB																
	A	NAME																
		MSB																
		LSB																
	B	NAME																
		MSB																
		LSB																
	C	NAME																
		MSB																
		LSB																
	D	NAME																
		MSB																
		LSB																
	E	NAME																
		MSB																
		LSB																
	F	NAME																
		MSB	REMSO	WDRST	REMC	TMRUN	TM03	TM13	SVDDT	RCDIV	RIC3	ROUT1	K03	K13	R03	R13	P03	OPDT
			IREM	TI2	EIREM	ETI2	TM02	TM12	SVDON	RCDUTY	RIC2	ROUT0	K02	K12	R02	R12	P02	OPON
			IK1	TI8	EIK1	ETI8	TM01	TM11	CLKCHG	RT1	RIC1	MF91	K01	K11	R01	R11	P01	IOC
		LSB	IK0	TI32	EIK0	ETI32	TM00	TM10	OSCC	RT0	RIC0	MF90	K00	K10	R00	R10	P00	R20

APPENDIX C. E0C621A I/O MEMORY MAP

Address	Register								Comment
	D3	D2	D1	D0	Name	Init	1	0	
0F0H	REMSO	IREM	IK1	IK0	REMSO	0	On	Off	Forced REM output (on/off)
					IREM	–	Yes	No	Interrupt factor flag (REM) Clear to 0 after read
	R/W	R			IK1	0	Yes	No	Interrupt factor flag (K10–K13)
					IK0	0	Yes	No	Interrupt factor flag (K00–K03)
0F1H	WDRST	TI2	TI8	TI32	WDRST	Clear	Clear	–	Watchdog timer reset
					TI2	0	Yes	No	Interrupt factor flag (Timer 2 Hz) Clear to 0 after read
	W	R			TI8	0	Yes	No	Interrupt factor flag (Timer 8 Hz)
					TI32	0	Yes	No	Interrupt factor flag (Timer 32 Hz)
0F2H	REMC	EIREM	EIK1	EIK0	REMC	1	On	Off	REM carrier generation on/off
					EIREM	0	Enable	Mask	Interrupt mask register (REM)
	R/W				EIK1	0	Enable	Mask	Interrupt mask register (K10–K13)
					EIK0	0	Enable	Mask	Interrupt mask register (K00–K03)
0F3H	TMRUN	ETI2	ETI8	ETI32	TMRUN	0	Run	Clear&Stop	Timer Run/Clear & Stop
					ETI2	0	Enable	Mask	Interrupt mask register (Timer 2 Hz)
	R/W				ETI8	0	Enable	Mask	Interrupt mask register (Timer 8 Hz)
					ETI32	0	Enable	Mask	Interrupt mask register (Timer 32 Hz)
0F4H	TM03	TM02	TM01	TM00	TM03	0			Timer data (low-order) 16 Hz
					TM02	0			Timer data (low-order) 32 Hz
	R				TM01	0			Timer data (low-order) 64 Hz
					TM00	0			Timer data (low-order) 128 Hz
0F5H	TM13	TM12	TM11	TM10	TM13	0			Timer data (high-order) 1 Hz
					TM12	0			Timer data (high-order) 2 Hz
	R				TM11	0			Timer data (high-order) 4 Hz
					TM10	0			Timer data (high-order) 8 Hz
0F6H	SVDDT	SVDON	CLKCHG	OSCC	SVDDT	–	Normal	Low	SVD data (1 when SVDON = 0)
					SVDON	0	On	Off	SVD circuit on/off
	R	R/W			CLKCHG	0	OSC1	OSC3	CPU CLK change OSC1/OSC3
					OSCC	1	On	Off	OSC3 oscillation on/off
0F7H	RCDIV	RCDUTY	RT1	RT0	RCDIV	–			REM carrier cycle set
					RCDUTY	–			REM carrier duty set
	R/W				RT1	–			REM τ-cycle set
					RT0	–			REM τ-cycle set
0F8H	RIC3	RIC2	RIC1	RIC0	RIC3	–			REM interrupt counter set (τ clock)
					RIC2	–			(all: 1 after count completed)
	W				RIC1	–			
					RIC0	–			
0F9H	ROUT1	ROUT0	MF91	MF90	ROUT1	0			REM output-on timer set (0τ–3τ)
					ROUT0	0			REM output-on timer set (0τ–3τ)
	R/W				MF91	–			General-purpose register (bit)
					MF90	–			General-purpose register (bit)
0FAH	K03	K02	K01	K00	K03	–	High	Low	Input port (fall: interrupt factor)
					K02	–	High	Low	Input port (fall: interrupt factor)
	R				K01	–	High	Low	Input port (fall: interrupt factor)
					K00	–	High	Low	Input port (fall: interrupt factor)
0FBH	K13	K12	K11	K10	K13	–	High	Low	Input port (fall: interrupt factor)
					K12	–	High	Low	Input port (fall: interrupt factor)
	R				K11	–	High	Low	Input port (fall: interrupt factor)
					K10	–	High	Low	Input port (fall: interrupt factor)
0FCH	R03	R02	R01	R00	R03	1	High	Low	Output port
					R02	1	High	Low	Output port
	R/W				R01	1	High	Low	Output port
					R00	1	High	Low	Output port
0FDH	R13	R12	R11	R10	R13	1	High	Low	Output port
					R12	1	High	Low	Output port
	R/W				R11	1	High	Low	Output port
					R10	1	High	Low	Output port
0FEH	P03	P02	P01	P00	P03	1	High	Low	I/O port (used as input port after initial reset)
					P02	1	High	Low	I/O port (used as input port after initial reset)
	R/W				P01	1	High	Low	I/O port (used as input port after initial reset)
					P00	1	High	Low	I/O port (used as input port after initial reset)
0FFH	OPDT	OPON	IOC	R20	OPDT	–	High	Low	Analog comparator data (0 or 1)
					OPON	0	On	Off	Analog comparator on/off
	R	R/W			IOC	0	Out	In	I/O port control out/in
					R20	0	High	Low	Output port

APPENDIX D. TROUBLESHOOTING

Tool	Problem	Remedy measures
ICE6200 (ICE62R)	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 ICS621AJ.EXE PC-DOS ICS621AW.EXE • Is the DOS version correct? MS-DOS Ver. 3.1 or later PC-DOS Ver. 2.1 or later • Is the DIP switches that set the baud rate of the main ICE6200 (ICE62R) unit set correctly? • Is the breaker of the ICE6200 (ICE62R) set to ON?
	The ICE6200 breaker tripped or the ICE62R fuse cut immediately after activation.	Check the following and remedy if necessary: • Are connectors F1 and F5 connected to the EVA621AR correctly? • Is the target board power short-circuiting?
	<ILLEGAL VERSION ICE6200> appears on the screen immediately after activation.	The wrong version of ICE is being used. Use the latest version.
	<ILLEGAL VERSION PARAMETER FILE> appears on the screen immediately after activation.	The wrong version of ICS621AP.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.
SOG621A	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
ASM621A	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.
MDC621A	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?
EVA621AR	The EVA621AR 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: • Has the VADJ VR inside the EVA621AR top cover been turned to a lower setting?

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

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

- SALES OFFICES -

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

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

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

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

EUROPE

EPSON EUROPE ELECTRONICS GmbH

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

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

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

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

ASIA

- CHINA -
EPSON (CHINA) CO., LTD.
28F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 Fax: 64107319

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

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

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

HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300, TAIWAN
Phone: 03-573-9900 Fax: 03-573-9169

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

- KOREA -
SEIKO EPSON CORPORATION KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: 02-784-6027 Fax: 02-767-3677

- JAPAN -
SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

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

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

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

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

EPSON

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

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

First issue FEBRUARY 1991, Printed FEBRUARY 1999 in Japan Ⓜ A