

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

S5U1C17000Y23

Multi-Programmer Ver. 3.0
System Manual

Evaluation board/kit and Development tool important notice

1. This evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. It is not intended to meet the requirements of design for finished products.
2. This evaluation board/kit or development tool is intended for use by an electronics engineer and is not a consumer product. The user should use it properly and in a safe manner. Seiko Epson does not assume any responsibility or liability of any kind of damage and/or fire caused by the use of it. The user should cease to use it when any abnormal issue occurs even during proper and safe use.
3. The part used for this evaluation board/kit or development tool may be changed without any notice.

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. When exporting the products or technology described in this material, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You are requested not to use, to resell, to export and/or to otherwise dispose of the products (and any technical information furnished, if any) for the development and/or manufacture of weapon of mass destruction or for other military purposes.

All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

Configuration of product number

Devices

S1 **C** **17xxx** **F** **00E1** **00**

■ Packing specifications

00: Besides tape & reel	
0A: TCP BL	2 directions
0B: Tape & reel	BACK
0C: TCP BR	2 directions
0D: TCP BT	2 directions
0E: TCP BD	2 directions
0F: Tape & reel	FRONT
0G: TCP BT	4 directions
0H: TCP BD	4 directions
0J: TCP SL	2 directions
0K: TCP SR	2 directions
0L: Tape & reel	LEFT
0M: TCP ST	2 directions
0N: TCP SD	2 directions
0P: TCP ST	4 directions
0Q: TCP SD	4 directions
0R: Tape & reel	RIGHT
99: Specs not fixed	

■ Specification

■ Package

D: Bare chip
F: QFP
B: BGA, WCSP

■ Model number

■ Model name

[C: Microcontroller, digital products]

■ Product classification

[S1: Semiconductor]

Development tools

S5U1 **C** **17000** **Y2** **1** **00**

■ Packing specifications

[00: Standard packing]

■ Version

[1: Version 1]

■ Tool type

Hx: ICE
Tx: Evaluation board
Cx: Compiler package
Yx: Programmer software

■ Corresponding model number

[17xxx: For S1C17xxx]

■ Tool classification

[C: Microcontroller use]

■ Product classification

[S5U1: Development tool
for semiconductor products]

Table of Contents

1. Overview	1
2. Hardware Requirements	2
3. ICDmini Supported Versions.....	3
4. Entire Workflow.....	4
4.1 Advance Preparation Flowchart.....	4
4.2 Gang Writer Configuration Flowchart.....	5
4.3 Multi-Programming Flowchart.....	5
5. Advance Preparation	6
5.1 Preparation of Gang Writer	6
5.2 Preparation of Model-Specific Information File.....	6
5.3 Confirming and Updating ICDmini Firmware.....	7
5.4 Confirming ICDmini Hardware Version and Programming Serial Number.....	9
5.5 ICDmini Configuration.....	10
5.6 Connecting ICDmini to PC	11
6. Gang Writer Configuration	12
6.1 Launching Gang Writer	12
6.2 Setting Gang Writer Parameters	13
6.3 Detecting ICDmini Connected	16
6.4 Loading Program Data to ICDmini	16
7. Multi-Programming	20
7.1 Entering ID Number	20
7.2 Connecting with Target System	20
7.3 Establishing Connection with ICDmini.....	22
7.4 Starting Multi-Programming	22
7.5 Disconnecting Target System.....	23
8. Other Functions	24
8.1 Individual Execution	24
8.2 Selecting a Process	24
8.3 Counter Function	24
8.4 Configuration.....	25
9. Troubleshooting.....	26
9.1 When Detecting ICDmini	26
9.2 When Loading Program Data to ICDmini	26
9.3 When Programming Target System	26
Appendix A Gang Writer Dynamic Link Library	27
A.1 Overview.....	27
A.1.1 Example of Function Call.....	27
A.2 Function Details.....	28
A.2.1 OpenIcdConnection	28
A.2.2 CloseIcdConnection	28

A.2.3	ResetTarget	29
A.2.4	CheckTargetConnection	29
A.2.5	StartOperation	30
A.2.6	GetStatus	31
A.2.7	GetString	32
A.2.8	GetConnectedLCD	33
A.2.9	LoadLcdFisProgram	35
A.2.10	LoadLcdUserProgram	36
A.2.11	GetLcdStatus	37
A.3	Restrictions	39
A.4	Measurement Results	39
Appendix B	List of Configurations by Model	40
Revision History	44

1. Overview

The Multi-Programmer system consists of a PC with the Gang Writer software and required files installed, a USB hub, and the necessary number of ICDmini software development emulators (S5U1C17001H11/12/21**) as shown in the figure below, and it is capable of being used to program one or more target MCUs.

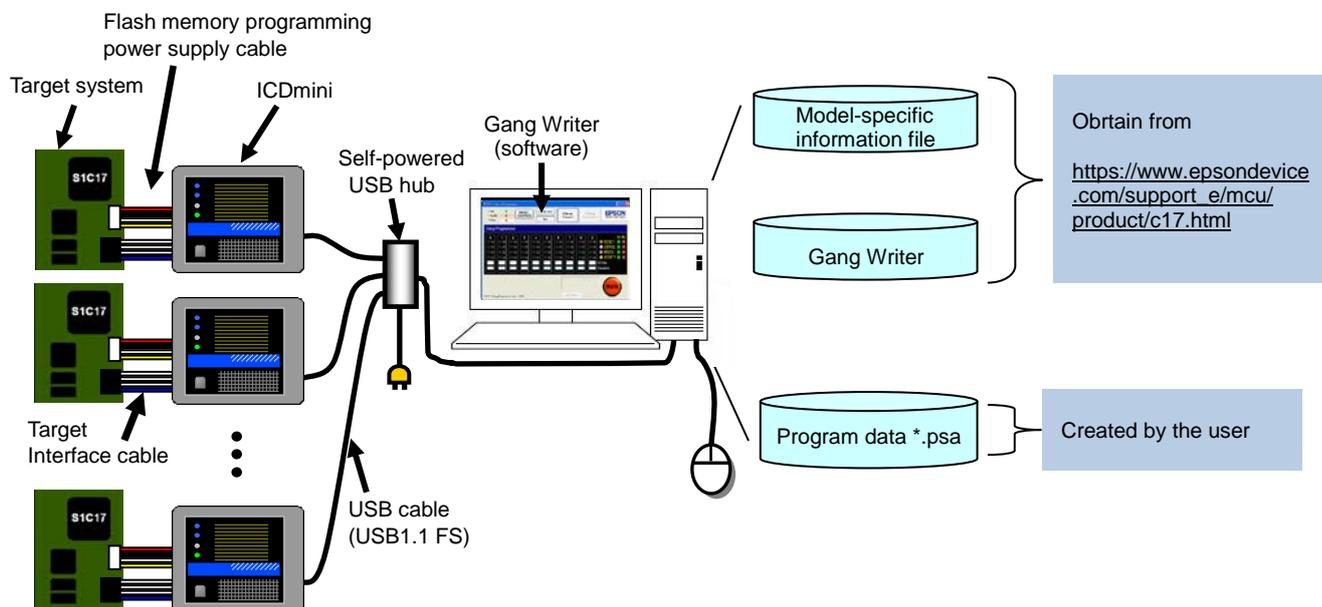


Figure 1.1 Overview of Multi-Programmer System

- The entire system including ICDminis, a USB hub, and a PC referred to as Multi-Programmer.
- Up to 10 targets can be programmed simultaneously.
- A self-powered USB hub should be used. (with external power supply).

2. Hardware Requirements

2. Hardware Requirements

The components shown below are required to configure the Multi-Programmer system. To be provided by the user.

Table 2.1 List of Components Required

Component	Spec Requirements	Remarks
PC	A PC with an Intel Pentium 4 1.50 GHz or higher CPU and 512 MB or more RAM included is recommended. Supported OS: Microsoft Windows	To be provided by the user.
USB hub	A self-powered hub that is able to supply a maximum of 500 mA per port (each ICDmini) is required.	To be provided by the user.
ICDmini	Prepare the necessary number of ICDminis for programming the target system.	Please contact the Seiko Epson sales representative.

3. ICDmini Supported Versions

The table below shows the ICDmini hardware and firmware versions supported by the Multi-Programmer.

Table 3.1 Versions Supported by Multi-Programmer

Hardware/software	Versions supported
ICDmini hardware	The version supported depends on the target MCU model. Refer to “ Available ICDmini hardware version ” under “ Appendix B List of Configurations by Model ” for more information by model. If you do not have an ICDmini of the version listed in that table, contact our sales representative. Note that a serial number must be programmed to the ICDmini if its version is 1.0 or 1.1, or if the firmware has been updated by the user.
ICDmini firmware	The version supported depends on the target MCU model. Refer to “ Available ICDmini firmware version ” under “ Appendix B List of Configurations by Model ” for more information by model. The ICDmini allows the user to update the firmware.

4. Entire Workflow

4. Entire Workflow

The Multi-Programmer workflow consists of three steps, advance preparation, Gang Writer configuration, and multi-programming. The flowcharts of these steps are shown below.

4.1 Advance Preparation Flowchart

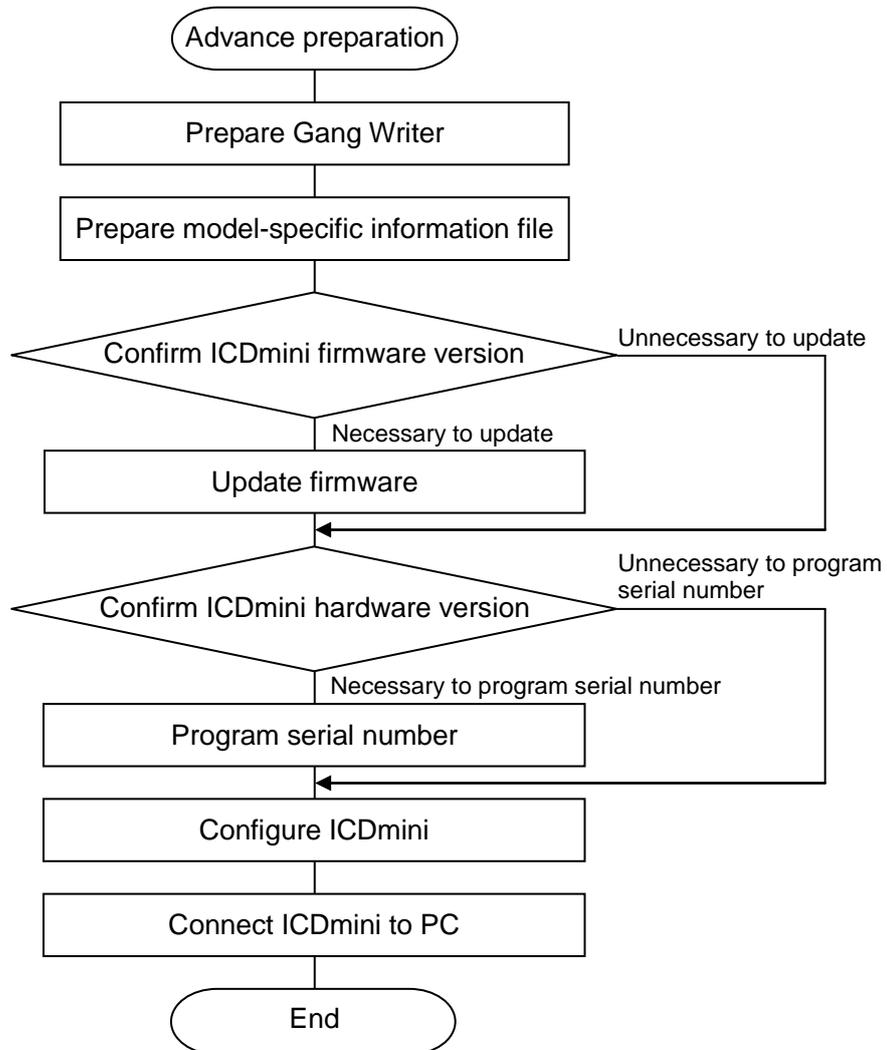


Figure 4.1.1 Advance Preparation Flowchart

4.2 Gang Writer Configuration Flowchart

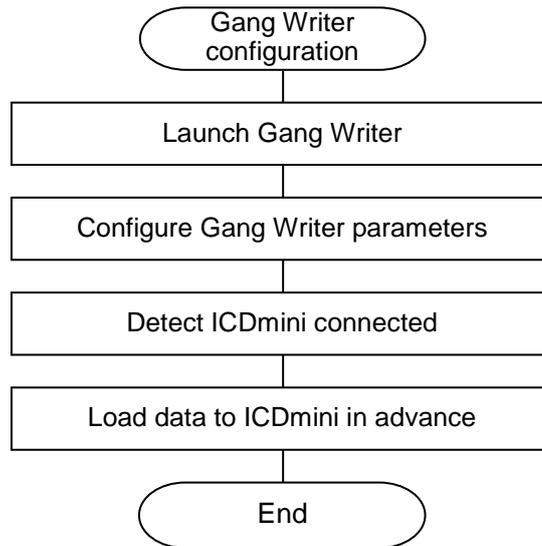


Figure 4.2.1 Gang Writer Configuration Flowchart

4.3 Multi-Programming Flowchart

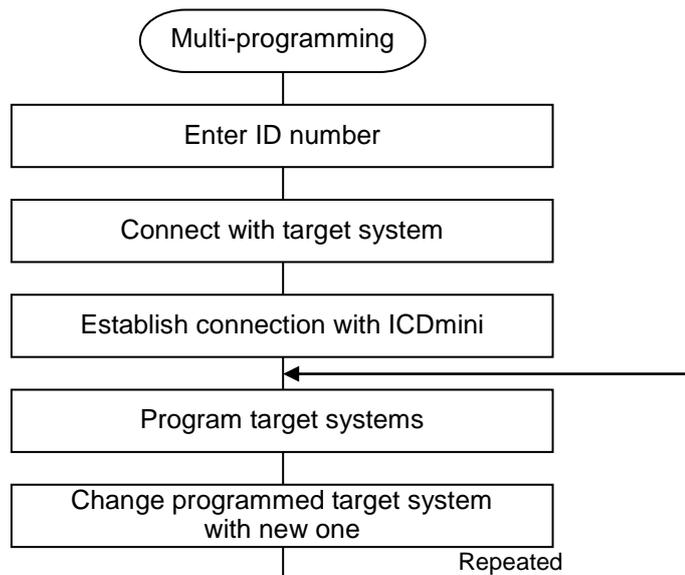


Figure 4.3.1 Multi-Programming Flowchart

5. Advance Preparation

5. Advance Preparation

It is necessary to prepare a file and to confirm the ICDmini version before multi-programming can be started.

5.1 Preparation of Gang Writer

Download the Gang Writer software from our website shown below, then unarchive and install it to the PC.

<MP Support Tool> section on https://www.epsondevice.com/support_e/mcu/product/page04.html#01

5.2 Preparation of Model-Specific Information File

Download the model-specific information file if available from our website shown below, then unarchive and install it to the PC.

<GNU17 IDE> section on https://www.epsondevice.com/support_e/mcu/product/page04.html#02

The same model-specific information files as GNU17 (S1C17 Family software development tool) are included in the S5U1C17000Y23 package. When the model-specific information file of a model is updated or the file for a new model is added before being included to GNU17, the file name appears on the model list of the website independently. If the model-specific information file name for the model to be used appears, be sure to download it.

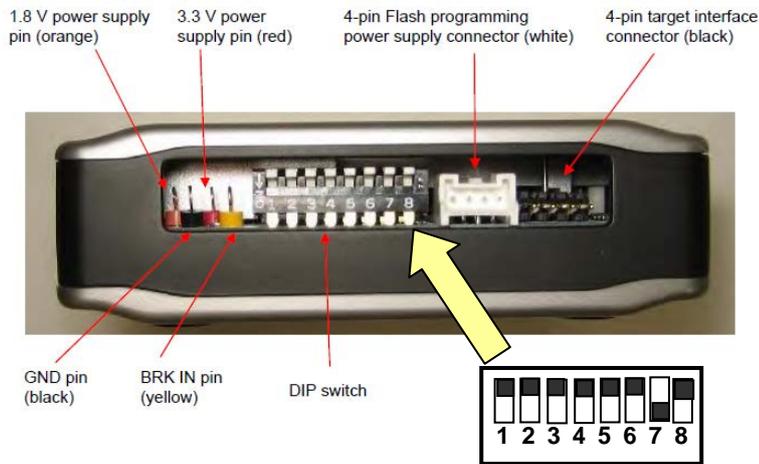
The downloaded model-specific information file should be unarchived into the folder shown below. If an older version of the model-specific information file exists in the folder, overwrite with the newly downloaded file.

<u>C:\EPSON\C17GangWriter\mcu_model</u>

The underlined part shown above is the default path when Gang Writer is installed to the default folder. This path changes to other drives and folders if another location other than the default folder is specified at the installation.

5.3 Confirming and Updating ICDmini Firmware

Set the DIP switch of the ICDmini as shown in the figure below. This makes it possible to establish communication between the PC and the ICDmini without a target system.



* 1.8 V and 3.3 V power supply pins are available only in the ICDmini Ver. 2.0.

Figure 5.3.1 ICDmini Left Side View and DIP Switch Setting

Connect the ICDmini and the PC directly using a USB cable without a USB hub.

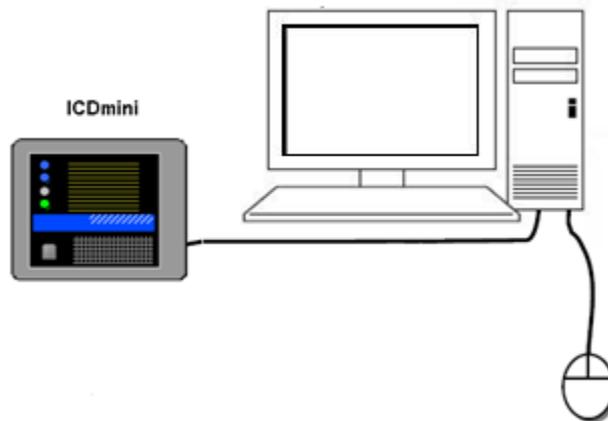


Figure 5.3.2 Connection Between PC and ICDmini

5. Advance Preparation

The “New hardware search wizard” starts when the ICDmini is connected to the PC for the first time. In this case, specify the folder shown below to install the USB driver.

```
C:\EPSON\C17GangWriter\utility\drv_usb\OS_32bit  
C:\EPSON\C17GangWriter\utility\drv_usb\OS_64bit
```

Note: Specify the folder in which Gang Writer is installed for the underlined path above. Moreover, the USB driver must use the one attached to this Gang Writer.

• Confirming ICDmini firmware version

Select [EPSON MCU] → [S1C17GangWriter] → [icdver] from the Start menu. When Windows 7 or Vista is used, right-click on [icdver] and select “Run as administrator.”

A window opens and displays the ICDmini hardware and firmware version numbers. Check to see if the firmware version is equal to or higher than that described in “Available ICDmini firmware version” under “Appendix B List of Configurations by Model.”

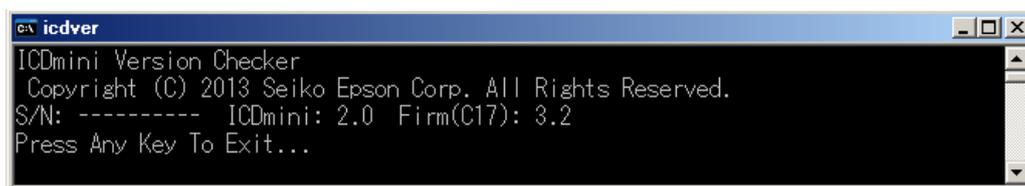


Figure 5.3.3 ICDmini Version Checker Execution Screen

The example above shows the version number as follows:

- Firmware: Ver. 3.2

S/N is the serial number set to the ICDmini, and hyphens are displayed if no serial number is programmed to the ICDmini. If the serial number is not displayed correctly, execute icdver again after pressing the reset button on the ICDmini.

• Updating ICDmini firmware

If the ICDmini firmware version is lower than that described in “Available ICDmini firmware version” under “Appendix B List of Configurations by Model,” update the firmware to the latest version. For how to update the firmware, refer to the website shown below.

<Necessary Tool> section on https://www.epsondevice.com/support_e/mcu/product/page04.html#01

5.4 Confirming ICDmini Hardware Version and Programming Serial Number

Confirm the ICDmini hardware version printed on the bottom. It is necessary to program a serial number to the ICDmini if the hardware version is lower than 2.0 or no serial number is printed. Furthermore, the serial number programmed in the ICDmini is cleared when the firmware is updated, therefore, serial number programming is also required in this case.

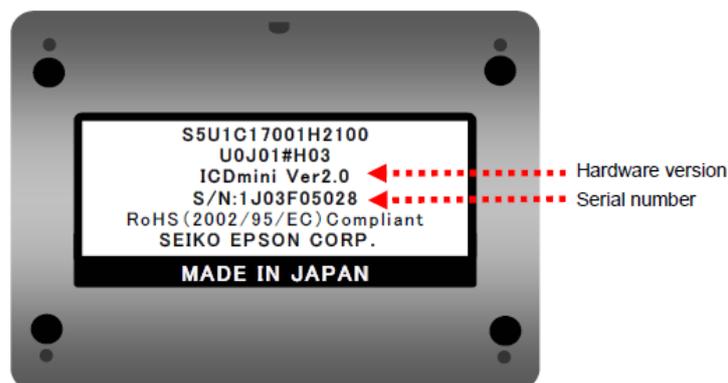


Figure 5.4.1 ICDmini Bottom View

The ICDmini Ver. 1.0 or Ver. 1.1 may not be used depending on the target MCU. For more information, refer to “Available ICDmini hardware version” under “Appendix B List of Configurations by Model.”

• Programming serial number

After connecting the ICDmini to the PC, select [EPSON MCU] → [S1C17GangWriter] → [C17SNwrite] from the Start menu. When Windows 7 or Vista is used, right-click on [C17SNwrite] and select “Run as administrator.” When the window shown below opens, enter the ICDmini serial number to the [Serial No.] field and then click on the [WRITE] button. The [READ] button can be used to read out the serial number of the ICDmini currently connected.

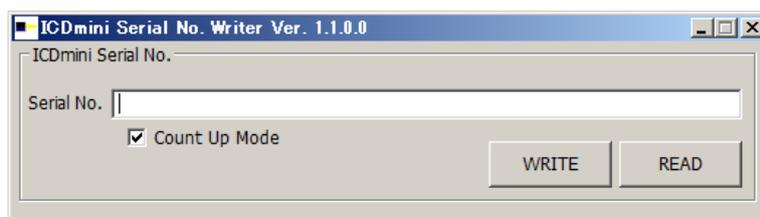


Figure 5.4.2 [ICDmini Serial Number] Writer Window

Although any number, except an overlapping number, can be entered as the serial number, as a general rule enter the one that is printed on the bottom of the ICDmini. If the ICDmini is not recognized, press the reset button on the ICDmini.

Note: The C17SNwrite must be executed with the ICDmini DIP switch No. 7 set to ON.

5. Advance Preparation

5.5 ICDmini Configuration

Set the ICDmini DIP switch as shown in the table below according to the interface voltage level with the target system.



Figure 5.5.1 ICDmini Left Side View

Table 5.5.1 ICDmini DIP Switch Setting

Target system interface voltage level	Switch settings
3.3 V	 1 2 3 4 5 6 7 8
1.8 V	 1 2 3 4 5 6 7 8
Voltage input from the target system	 1 2 3 4 5 6 7 8

For the DIP SW8 setting, refer to “ICDmini DIP switch (SW8) & FLASH VCC OUT connection” under “Appendix B List of Configurations by Model.”

5.6 Connecting ICDmini to PC

Connect the ICDminis and the PC via a USB hub after the preparation described above has been completed.

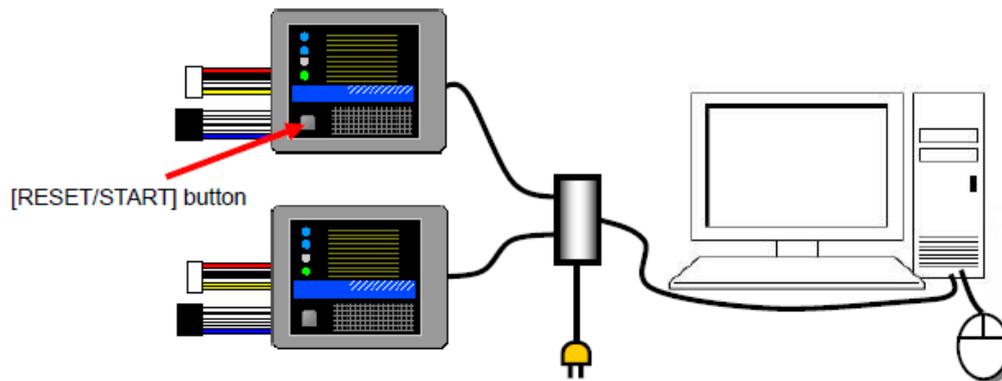


Figure 5.6.1 Connection Diagram Between PC and ICDminis

6. Gang Writer Configuration

6. Gang Writer Configuration

6.1 Launching Gang Writer

The Gang Writer starts up by selecting [EPSON MCU] → [S1C17GangWriter] → [C17GangWriter] from the Start menu of the PC. When Windows 7 or Vista is used, right-click on [C17GangWriter] and select “Run as administrator.”

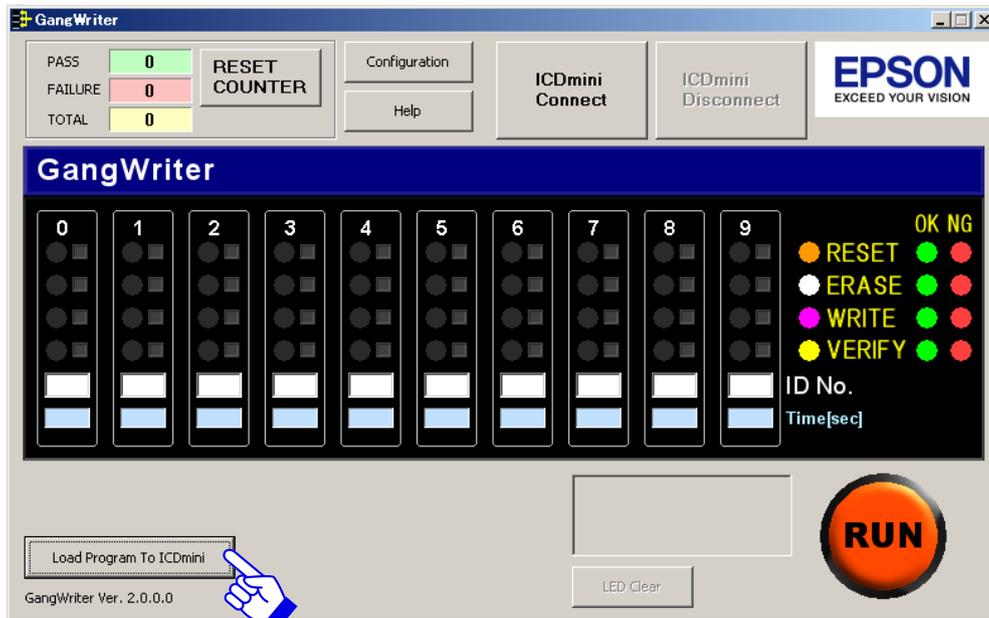


Figure 6.1.1 [GangWriter] Window

Before loading the target program data to the ICDmini, click on the [Load Program To ICDmini] button. Then click on the [Advanced] button in the window opened to set parameters.

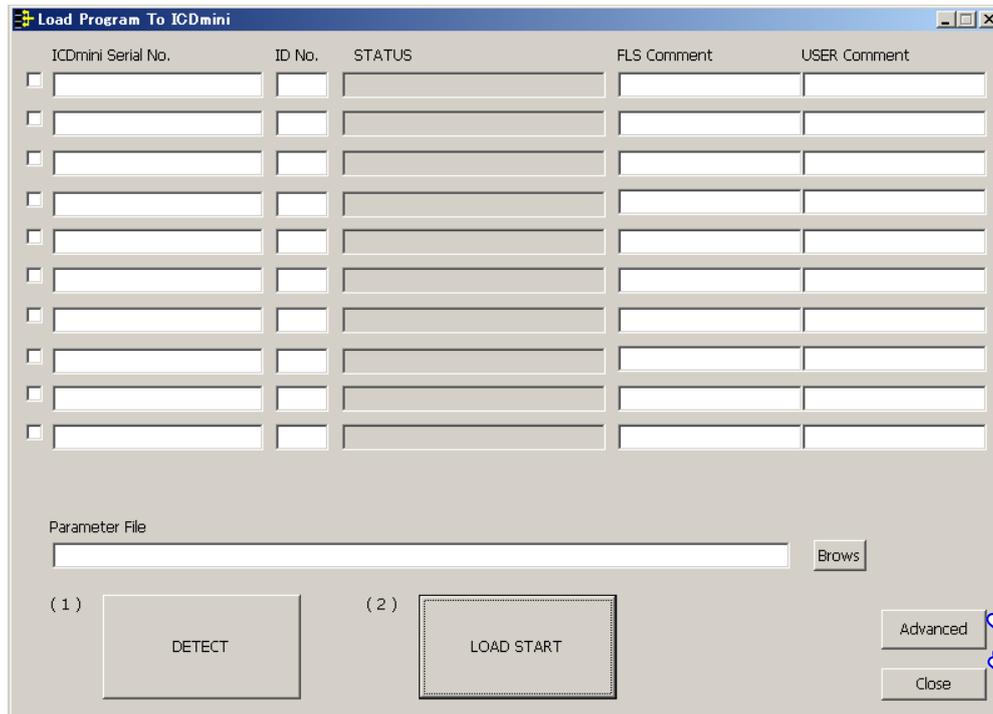


Figure 6.1.2 [Load Program To ICDmini] Window

6.2 Setting Gang Writer Parameters

Enter parameters in the dialog box shown below. For the parameter setting values that depend on the model, refer to “Appendix B List of Configurations by Model.” Parameters (1) and (6) that specify file names are required to be entered in a full-path. Use the [Browse] button to select them from a file select dialog box.

The dialog box is titled "Set Load Parameter" and contains the following fields and controls:

- Erase/Write Program**
 - File: (1) [Brows]
 - Erase Routine Address: 0x (2) (0-ffffffe)
 - Write Routine Address: 0x (3) (0-ffffffe)
 - Time out: (4) [0-60 sec]
 - Comment: (5)
- User Program**
 - File: (6) [Brows]
 - Start Block No. of Erase: (7) (0-16777215)
 - End Block No. of Erase: (8) (0-16777215)
 - Flash Memory Top Address: 0x (9) (0-ffffffe)
 - Comment: (10)

Buttons: OK, Cancel

Figure 6.2.1 [Set Load Parameter] Dialog Box

6. Gang Writer Configuration

Table 6.2.1(a) Details of Parameters Set in [Set Load Parameter] Dialog Box (1)

Erase/Write Program																	
(1) File	Enter the FLS file name.																
(2) Erase Routine Address	Enter the flash memory erase routine start address in a hexadecimal number. (Effective range: 0x0 to 0xfffffe) The prefix "0x" or "0X" is not necessary to enter.																
(3) Write Routine Address	Enter the flash memory program routine start address in a hexadecimal number. (Effective range: 0x0 to 0xfffffe) The prefix "0x" or "0X" is not necessary to enter.																
(4) Time out	Enter the time used to determine a time-out error during erasing/programming the flash memory. (Effective range: 0 to 60 seconds) Setting to "0" disables monitoring of time-out. The time-out value should be set to a time required to erase and program the flash memory + a margin in order to avoid a system freeze. If the time required for erasing and programming the flash memory is unknown, first set the maximum value and adjust it by confirming the actual programming time.																
(5) Comment	<p>Enter any comments and/or the commands shown below. Up to 100 characters can be entered.</p> <table border="1"> <tr> <td colspan="2">-v</td> </tr> <tr> <td>Function</td> <td>Flash programming voltage control option Specifying the "-v" option within the comment field allows the flash programming voltage to be set.</td> </tr> <tr> <td>Format</td> <td>-vEraseVoltage-WriteVoltage EraseVoltage: Erasing voltage WriteVoltage: Programming voltage</td> </tr> <tr> <td>Condition</td> <td>EraseVoltage: 6.0 V ≤ EraseVoltage ≤ 8.0 V WriteVoltage: 6.0 V ≤ WriteVoltage ≤ 8.0 V</td> </tr> <tr> <td colspan="2">-s</td> </tr> <tr> <td>Function</td> <td>Communication packet size specification option Specifying the "-s" option within the comment section allows the communication packet size to be set for transferring user programs.</td> </tr> <tr> <td>Format</td> <td>-sSendSize SendSize: Communication packet size</td> </tr> <tr> <td>Condition</td> <td>SendSize: 1 byte ≤ SendSize ≤ 1,010 bytes (decimal number)</td> </tr> </table>	-v		Function	Flash programming voltage control option Specifying the "-v" option within the comment field allows the flash programming voltage to be set.	Format	-vEraseVoltage-WriteVoltage EraseVoltage: Erasing voltage WriteVoltage: Programming voltage	Condition	EraseVoltage: 6.0 V ≤ EraseVoltage ≤ 8.0 V WriteVoltage: 6.0 V ≤ WriteVoltage ≤ 8.0 V	-s		Function	Communication packet size specification option Specifying the "-s" option within the comment section allows the communication packet size to be set for transferring user programs.	Format	-sSendSize SendSize: Communication packet size	Condition	SendSize: 1 byte ≤ SendSize ≤ 1,010 bytes (decimal number)
-v																	
Function	Flash programming voltage control option Specifying the "-v" option within the comment field allows the flash programming voltage to be set.																
Format	-vEraseVoltage-WriteVoltage EraseVoltage: Erasing voltage WriteVoltage: Programming voltage																
Condition	EraseVoltage: 6.0 V ≤ EraseVoltage ≤ 8.0 V WriteVoltage: 6.0 V ≤ WriteVoltage ≤ 8.0 V																
-s																	
Function	Communication packet size specification option Specifying the "-s" option within the comment section allows the communication packet size to be set for transferring user programs.																
Format	-sSendSize SendSize: Communication packet size																
Condition	SendSize: 1 byte ≤ SendSize ≤ 1,010 bytes (decimal number)																

Table 6.2.1(b) Details of Parameters Set in [Set Load Parameter] Dialog Box (2)

User Program	
(6) File	Enter the name of the user program data file in psa or saf format that is built under the S1C17 development environment. (A psa file is recommended.)
(7) Start Block No. of Erase	Enter the erasing start block number of the flash memory in a decimal number. (Effective range: 0 to 16,777,215) Enter "0" when erasing the entire flash memory.
(8) End Block No. of Erase	Enter the erasing end block number of the flash memory in a decimal number. (Effective range: 0 to 16,777,215) Enter "0" when erasing the entire flash memory.
(9) Flash Memory Top Address	Enter the flash memory top address in a hexadecimal number. (Effective range: 0x0 to 0xfffffe) The prefix "0x" or "0X" is not necessary to enter. Specify the address with an even number.
(10) Comment	Enter any comments (optional). Up to 100 characters can be entered.

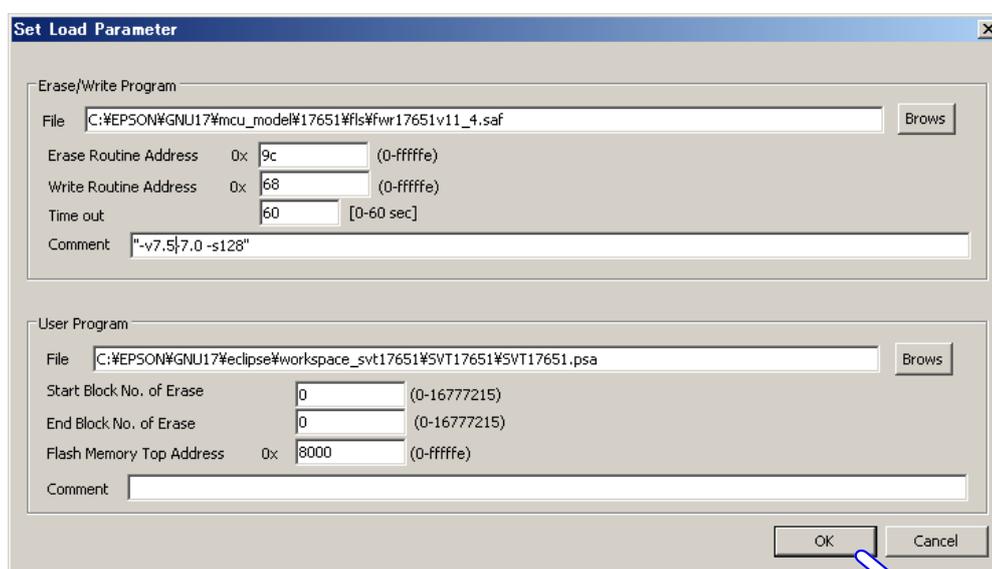


Figure 6.2.2 [Set Load Parameter] Setting Example

After the necessary parameters have been entered, click [OK].

A dialog box appears to prompt the user to save the parameter file. Click [Yes] to save the parameter file into an arbitrary folder.

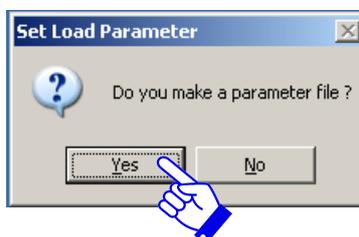


Figure 6.2.3 Saving Parameter File

6. Gang Writer Configuration

6.3 Detecting ICDmini Connected

Click on the [DETECT] button to detect the ICDminis currently connected to the PC.

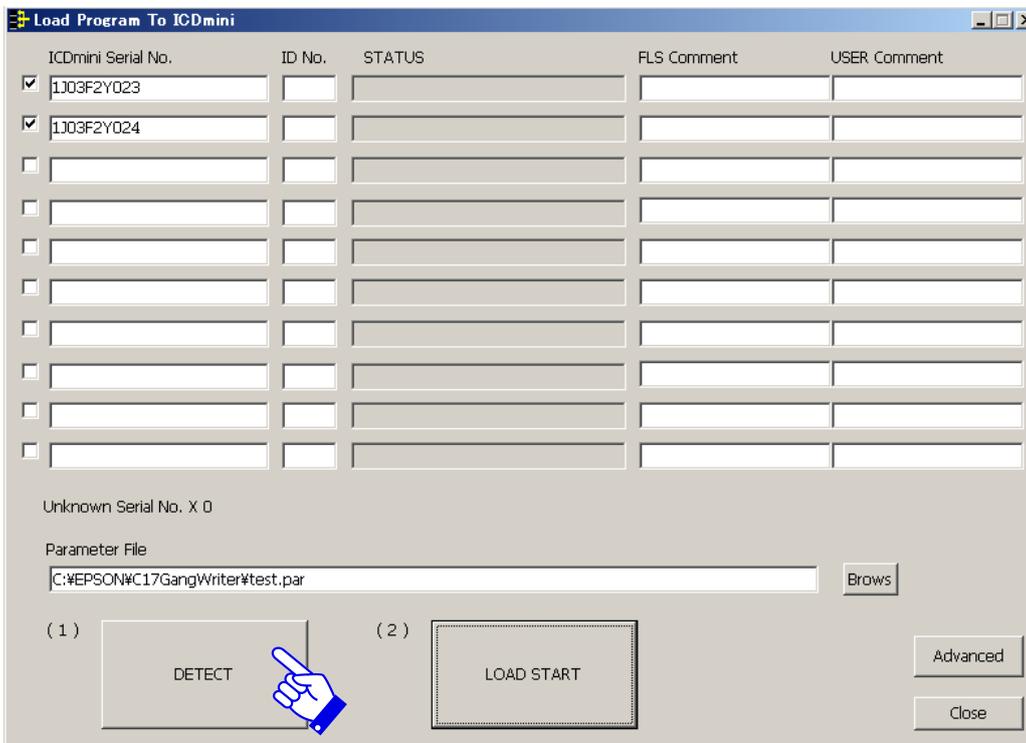


Figure 6.3.1 Detecting ICDmini Connected

6.4 Loading Program Data to ICDmini

When Gang Writer detects the ICDminis connected to the PC, their serial numbers are displayed in the [ICDmini Serial No.] fields. The ID No. is a convenience number so that the programming status information displayed on the PC will correspond to each ICDmini. Enter a unique value within 0 to 999 to each [ID No.] field located to the right of the serial number displayed.

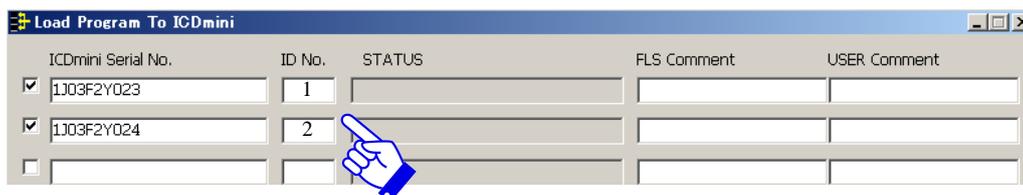


Figure 6.4.1 ID No. Setting

Then select the parameter file that has been saved in advance by clicking on the [Browse] button.

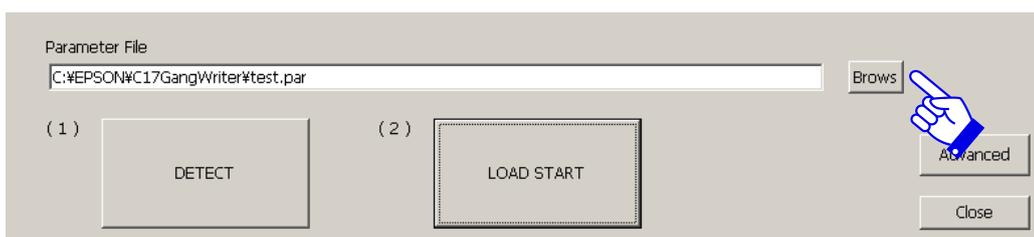


Figure 6.4.2 Parameter File Selection

6. Gang Writer Configuration

When all the preparations are completed, click on the [LOAD START] button to start loading program data into the ICDminis. When loading to each ICDmini has completed without an error, the [STATUS] field for that ICDmini displays “(G00) COMPLETE” with the background highlighted in green. If the loading operation has failed, the [STATUS] field for that ICDmini displays an error status message with the background highlighted in red.

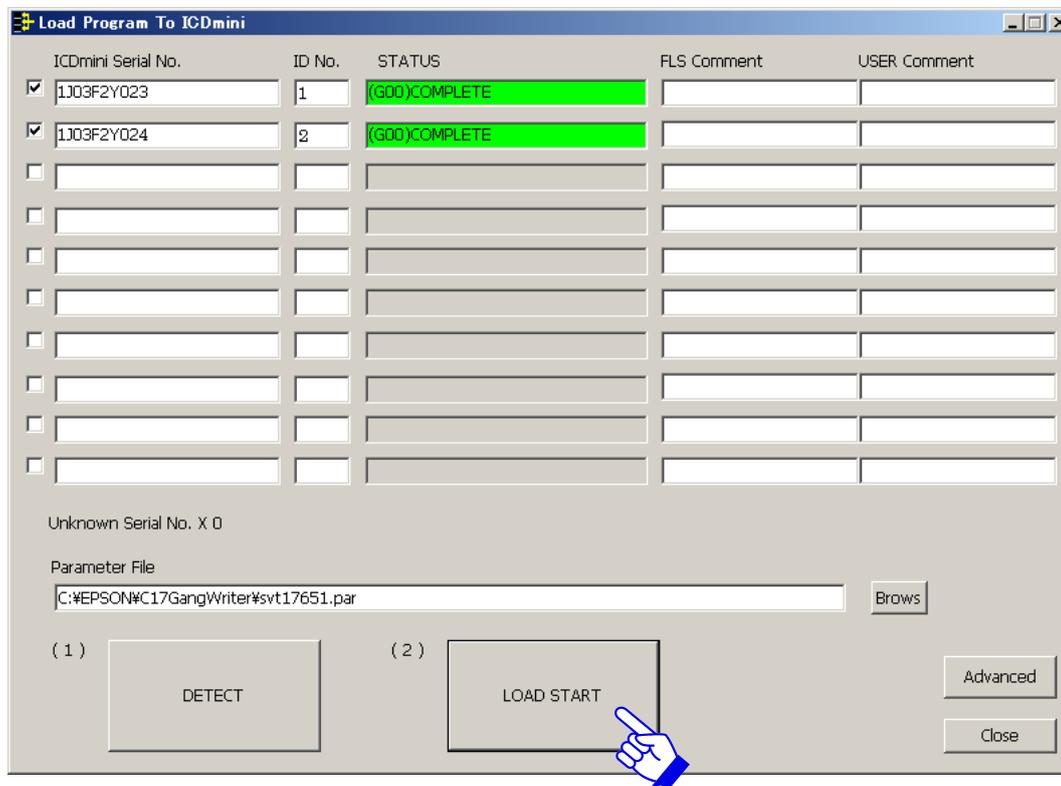


Figure 6.4.3 Loading Program Data to ICDmini

If the check is removed from the check box located to the left of the serial number, Gang Writer does not load the program data to the ICDmini. After all the loading operations have completed, click on the [Close] button to close this window.

6. Gang Writer Configuration

Table 6.4.1 List of Error Messages

Error Message	Description
Please check all items.	No target system to be programmed is found. Make sure that the ICDmini is connected to the PC correctly and then click on the [DETECT] button to get the connected ICDmini recognized before clicking the [LOAD START] button.
Over useful range. Check ID No.	One or more values entered in the [ID No.] fields exceed the effective range. The ID No. must be within the range from 0 to 999.
There are the same as "ICDmini Serial No." item two or more.	Two or more ICDminis have the same serial number. Refer to "5.4 Confirming ICDmini Hardware Version and Programming Serial Number" and reconfigure the ICDminis with a unique serial number.
There are the same as "ID No." item two or more.	Two or more [ID No.] fields have the same number entered. Change them to a unique number.
Please set load parameter information.	The [Parameter File] field is empty and the parameters have not been configured in the [Set Load Parameter] dialog box yet. Refer to "6.2 Setting Gang Writer Parameters" and fill in the parameter fields.
Can not found Parameter file.	The parameter file specified in the [Parameter File] field cannot be found. Use the [Browse] button to specify the parameter file. Refer to "6.4 Loading Program to ICDmini."
Parameter file error.	There is an error in the contents of the parameter file. Refer to "6.2 Setting Gang Writer Parameters" and correct the illegal parameter value.
Following characters of parameter file are not allowed to used. / ; * ? < > "	One of the characters shown below is included in the parameter file name. / ; , * ? < > " Do not use these characters for the parameter file name.

Table 6.4.2 List of STATUS Messages

STATUS message	Description
(G00) COMPLETE	Loading FLS and the user program data to the ICDmini has completed successfully.
(G10) FLS ERASE	FLS storage area is being erased.
(G11) FLS LOAD xxx%	FLS is being loaded (0% to 100%).
(G12) FLS VERIFY	FLS storage area is being verified.
(G13) USER ERASE	User program data storage area is being erased.
(G14) USER LOAD xxx%	User program data is being loaded (0% to 100%)
(G15) USER VERIFY	User program data storage area is being verified.
(G21) FLS TIMEOUT ERROR	A communication time-out has occurred during loading FLS to the ICDmini.
(G22) FLS FILE OPEN ERROR	The FLS file cannot be opened.
(G23) FLS FILE FORMAT ERROR	The FLS file format is invalid. (It is not a Motorola format.)
(G24) FLS FILE SIZE OVER	The FLS file size exceeds 8KB.
(G25) FLS FILE ADDRESS OVER	The address in the FLS file exceeds the 24-bit range.
(G26) FLS LOAD VERIFY ERROR	An error has occurred during verify check after loading FLS to the ICDmini.
(G27) FLS UNEXPECTED ERROR	Unexpected error has occurred during loading FLS to the ICDmini.
(G31) USER TIMEOUT ERROR	A communication time-out has occurred during loading the user program data to the ICDmini.
(G32) USER FILE OPEN ERROR	The user program data file cannot be opened.
(G33) USER FILE FORMAT ERROR	The user program data file format is invalid. (It is not a Motorola format.)
(G34) USER FILE SIZE OVER	The user program data file size exceeds 4MB.
(G35) USER FILE ADDRESS OVER	The address in the user program data file exceeds the 24-bit range.
(G36) USER LOAD VERIFY ERROR	An error has occurred during verify check after loading the user program data to the ICDmini.
(G37) USER UNEXPECTED ERROR	Unexpected error has occurred during loading the user program data to the ICDmini.
(G99) ICD COMMUNICATION ERROR	Communication with the ICDmini cannot be established.

Display message number classification

(G00): Loading has completed successfully.

(G1x): Data is being loading.

(G2x): An error has occurred during loading FLS.

(G3x): An error has occurred during loading the user program data.

(G99): Communication with the ICDmini has not been established.

7. Multi-Programming

7. Multi-Programming

7.1 Entering ID Number

Enter the same ID number as that set in the [Load Program To ICDmini] window to the [ID No.] fields in the [GangWriter] window to associate the target numbers in the [GangWriter] window and the ICDminis connected to the PC.

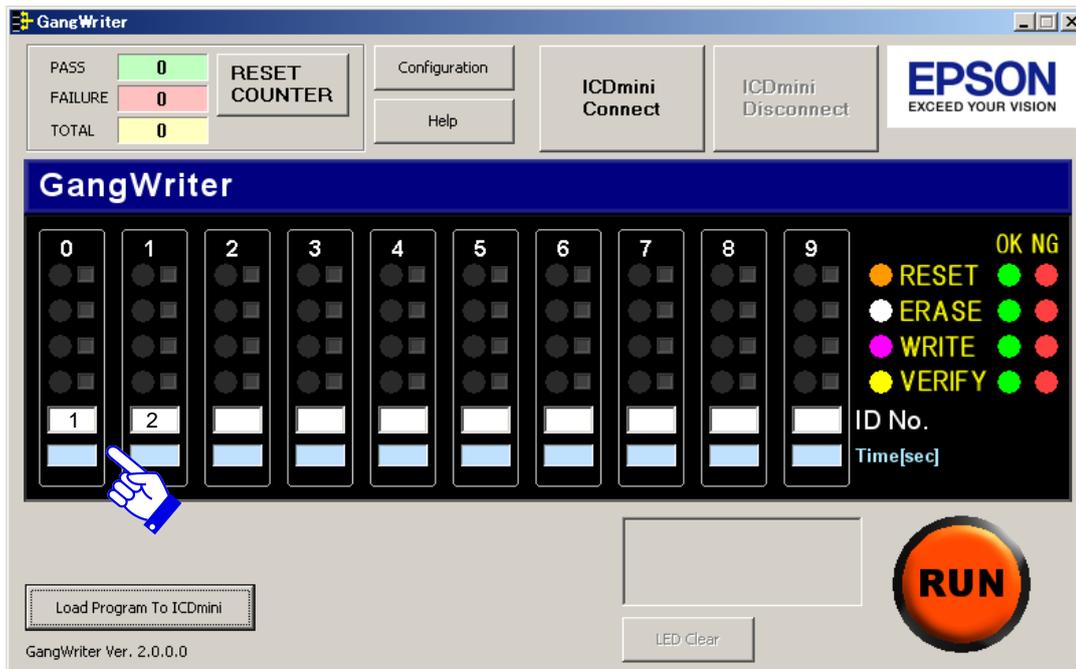


Figure 7.1.1 ID Number Setting in Gang Writer Main Window

7.2 Connecting with Target System

Each target system is connected to the 4-pin target interface connector and the 4-pin flash memory programming power supply connector of each ICDmini.

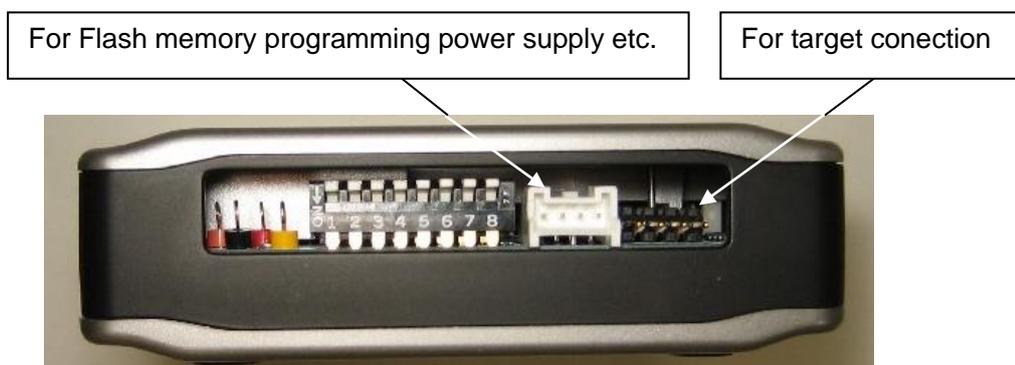


Figure 7.2.1 Connectors on ICDmini (Left Side View)

- Target interface connector (black)

Table 7.2.1 Target Interface Connector Pin Assignment

4 3 2 1	No.	Pin name	I/O	Pin function
1	1	DCLK	I	Clock signal input for debugging
2	2	GND	–	Power supply ground
3	3	DSIO	I/O	Serial communication signal input/output for debugging
4	4	DST2	I	Debug status signal input

Note: Be sure to connect all pins of this connector to the target MCU.

- Flash memory programming power supply connector (white)

Table 7.2.2 Flash Memory Programming Power Supply Connector Pin Assignment

1 2 3 4	No.	Pin name	I/O	Pin function
1	1	FLASH VCC OUT	O	Flash memory programming power supply voltage output (available only for ICDmini Ver. 2.0) Refer to "ICDmini DIP switch (SW8) & FLASH VCC OUT Connection" under "Appendix B List of Configurations by Model" to determine whether this pin is connected to the target MCU or not.
2	2	GND	–	Power supply ground Connect this pin to the ground of the target system.
3	3	TARGET RST OUT	O	Target reset signal output Be sure to connect this pin to the reset pin on the target system.
4	4	TARGET VCC IN	I	Target voltage input When an external power supply voltage (1.0 V to 5.5 V) is used for interfacing between an ICDmini and a target system, it should be input from the target system to this pin. When using the interface voltage generated in the ICDmini, leave this pin open and select the voltage level from 3.3 V and 1.8 V using the DIP switch. For the DIP switch setting, refer to Table 5.5.1, "ICDmini DIP Switch Setting."

Connect each ICDmini with a target system and supply power to the target systems.

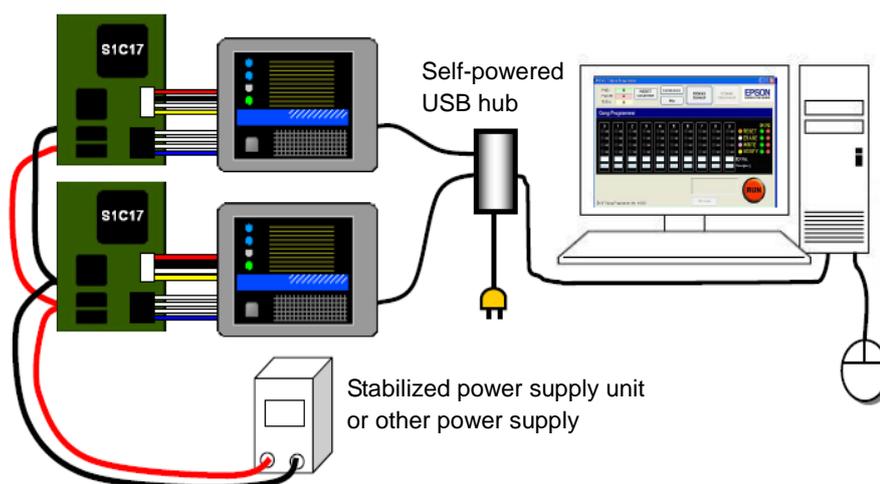


Figure 7.2.2 Multi-Programming System Configuration Diagram

When connections are completed, press the reset buttons on the ICDminis. If the "New hardware search wizard" starts on the PC after that, install the USB driver. For the location of the USB driver, refer to "5.3 Confirming and Updating ICDmini Firmware."

7. Multi-Programming

7.3 Establishing Connection with ICDmini

Click on the [ICDmini Connect] button to establish connection with the ICDminis.

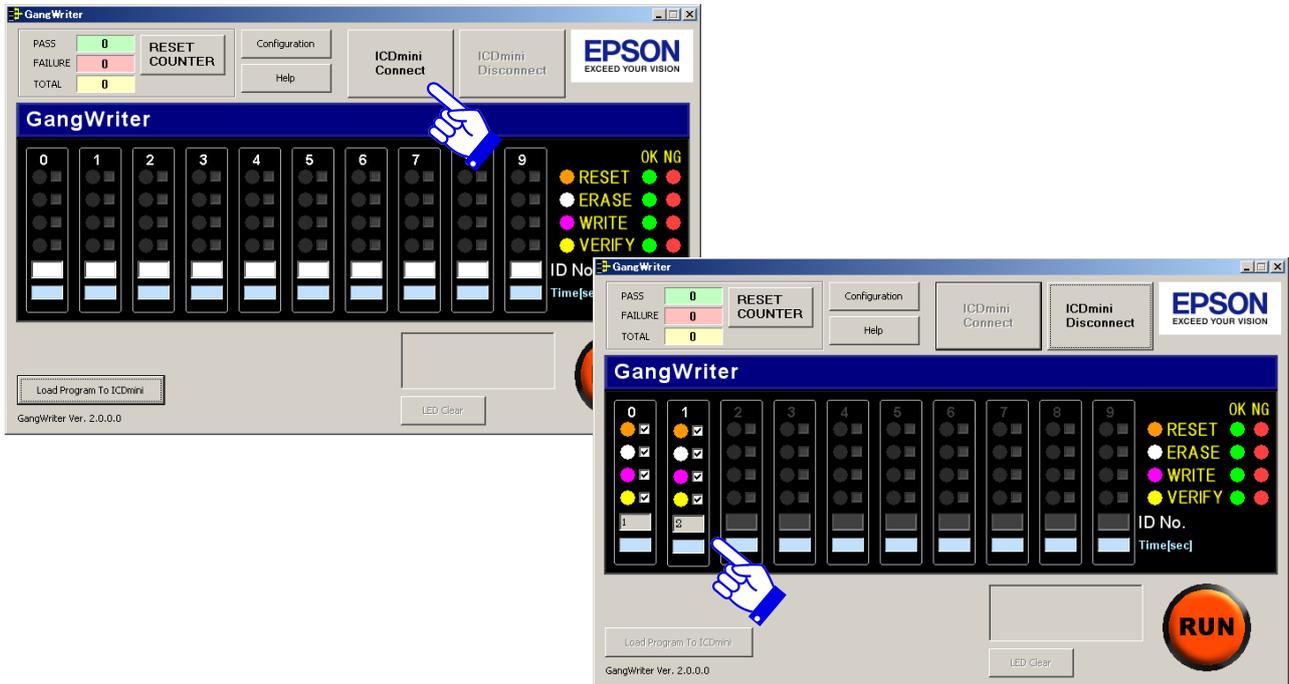


Figure 7.3.1 Connection Establishment with ICDminis

7.4 Starting Multi-Programming

To start multi-programming, click on the [RUN] button or press [Enter] on the keyboard. The multi-programming is executed in four steps, RESET, ERASE, WRITE, and VERIFY processes in order, and the process being currently executed is indicated by the indicators (LED symbols) located at each target field that is blinking. When each process has completed normally, the corresponding indicator lights in green.

When multi-programming for all the target systems has completed, “ALL PASS” is displayed in the execution result display field located to the left of the [RUN] button.

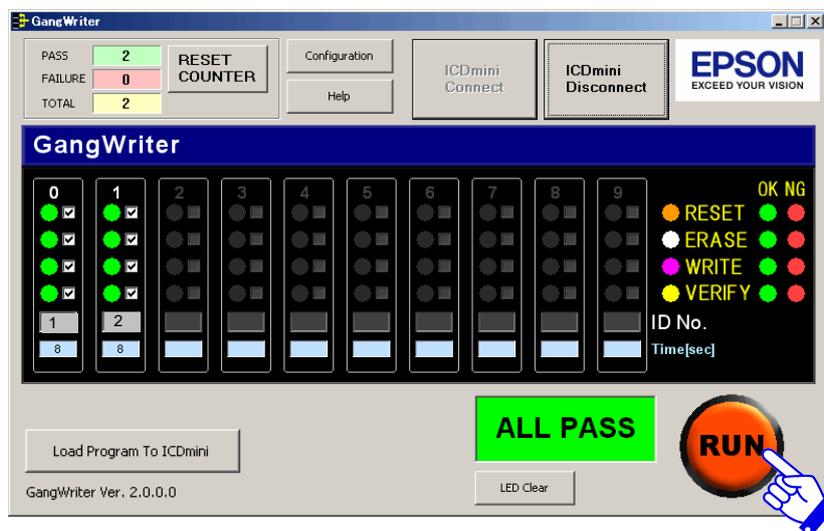


Figure 7.4.1 [GangWriter] Window with “ALL PASS” Displayed

When an error has occurred in a process for a target system, the indicator corresponding to that lights in red, and multi-programming is terminated abnormally at that point. In this case, “FAILURE” is displayed in the execution result display field.

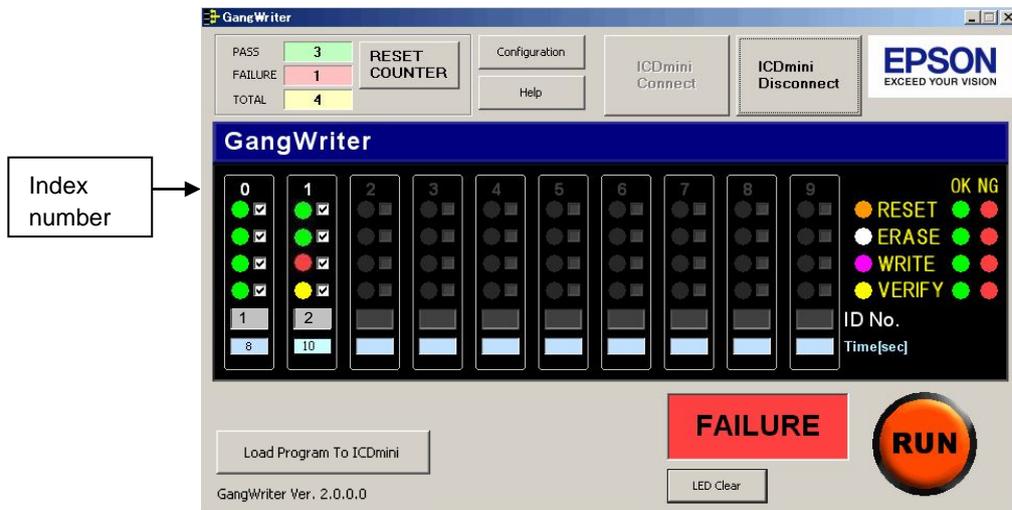


Figure 7.4.2 Display when an Error Occurred

After an error has occurred, entering the index number of the ICDmini in which an error occurred from the keyboard, retries the process for the ICDmini. For example, to retry the process for index number 1, press the “1” key.

7.5 Disconnecting Target System

After turning power to the target systems off, disconnect the programmed target systems and connect the new target systems to be programmed next. Then repeat the multi-programming operations described in the previous section.

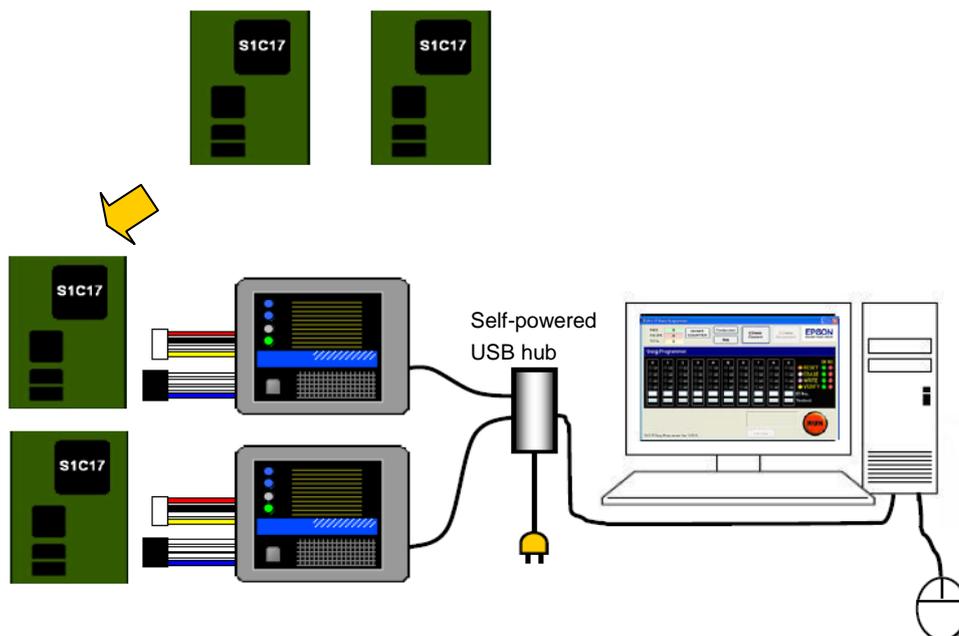


Figure 7.5.1 Changing Target Systems

8. Other Functions

8. Other Functions

8.1 Individual Execution

After connections with the ICDminis are established by clicking on the [ICDmini Connect] button, entering an index number from the keyboard (ten keyboard can also be used) executes programming of only the target system specified by the index number.

8.2 Selecting a Process

The check boxes in the each target system field, that correspond to the RESET, ERASE, WRITE, and VERIFY processes from the top, can be used to select the processes to be executed. Selecting a check box will execute the process and clearing a check box skips the process.

Example: Setting to execute the RESET and VERIFY processes in the target system index number 0

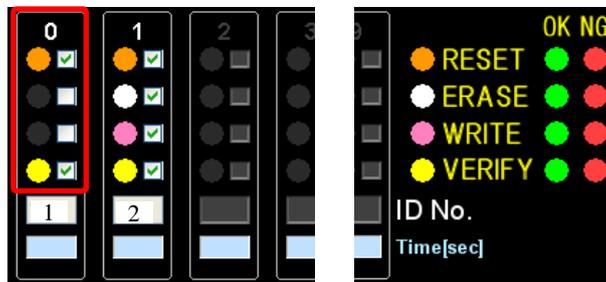


Figure 8.2.1 Selecting Execution Processes Individually

The check box settings are saved and will also be effective at the next time Gang Writer starts up.

8.3 Counter Function

The PASS, FAILURE, and TOTAL fields in the Gang Writer window show the counts of programming succeeded, failed, and the total of them, respectively. Clicking on the [RESET COUNTER] button clears these counters to 0.



Figure 8.3.1 Counter Display

8.4 Configuration

Clicking on the [Configuration] button opens the [Configurations] dialog box allowing the user to specify the saving location and name of log file, and a time-out condition.



Figure 8.4.1 [Configuration] Button

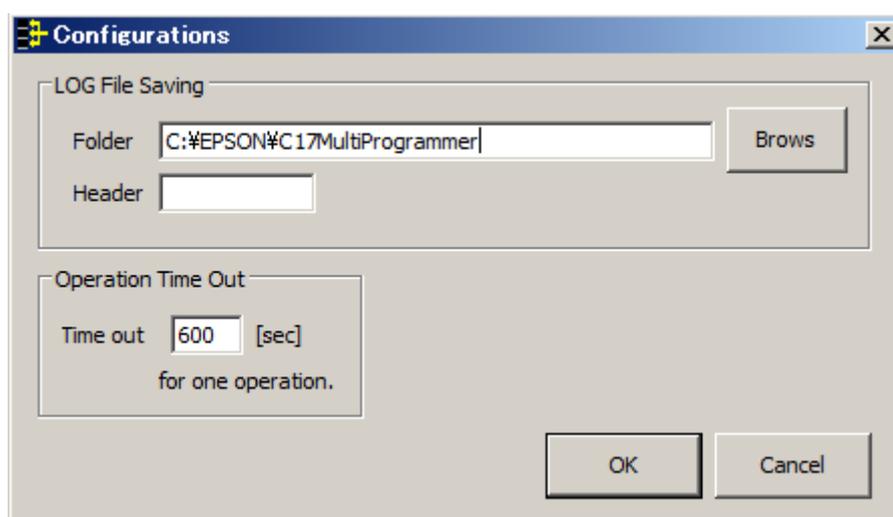


Figure 8.4.2 [Configurations] Dialog Box

• Folder

Specify the folder where log files are to be saved in the [Folder] field. When entering a path to the folder, it must begin with the drive name. It can also be specified using a folder select dialog box that appears by clicking on the [Browse] button.

• Header

Enter the leading characters of log files in the [Header] field. This field can be left blank. With the leading characters added, the log file appears as follows:

[Header]YYYYMMDD.txt

If “LOG” is specified in [Header] and the current date is June 30, 2014, the file name is as follows:

LOG20140630.txt

The Folder and Header settings will take effect from the next time Gang Writer starts up.

• Time out

Enter the time-out time from the beginning of programming (from erasing) in the [Time out] field. (The default value is 600 seconds.) If data to be programmed is large, a long time-out should be set. Enter a value that allows a sufficient margin. (Setting to 3 seconds or less will always cause an error.)

9. Troubleshooting

9. Troubleshooting

9.1 When Detecting ICDmini

- **“Detected “Serial No” is 0.” is displayed and the ICDmini is not recognized.**

- (1) The PC and the ICDmini may be disconnected.
Try to connect the ICDmini to the PC once again. (Refer to “7.2 Connecting with Target System.”)
- (2) The ICDmini that is connected has no serial number programmed.
Program the serial number to the ICDmini. (Refer to “5.4 Confirming ICDmini Hardware Version and Programming Serial Number.”)

9.2 When Loading Program Data to ICDmini

- **A file error has occurred.**

The file name may not be specified in a full-path in the [Set Load Parameter] dialog box.
Use the [Browse] button to select the file. (Refer to “6.2 Setting Gang Writer Parameters.”)

9.3 When Programming Target System

- **A write error has occurred. (FAILURE)**

- (1) The debug pins (DCLK, DSIO, DST2) of the target MCU may not be correctly connected to the target interface connector on the ICDmini.
Check the connection between the ICDmini and the target system. (Refer to “7.2 Connecting with Target System.”)
- (2) Noise may cause an error. Make sure that the wiring length between the target MCU and the ICDmini is as short as possible (roughly 15 cm or shorter).
Provide a countermeasure against noise for the wiring pattern on the target board.
- (3) There may be a difference in the interface voltage between the target system and the ICDmini.
Set the DIP switch on the ICDmini, which allows selection of the interface voltage from 3.3 V, 1.8 V, or target voltage input, appropriately. (Refer to “5.5 ICDmini Configuration.”)
- (4) The flash programming voltage may not be supplied to the target MCU that needs an external flash programming power supply.
Make sure that the target system is connected to the flash memory programming power supply connector correctly. (Refer to “7.2 Connecting with Target System.”)
- (5) The Gang Writer parameters may not be set to appropriate values.
Configure them with an appropriate value. (Refer to “6.2 Setting Gang Writer Parameters.”)
- (6) The reset button on the ICDmini may be pressed after connection with the ICDmini is established by clicking on the [ICDmini Connect] button.
If the reset button has been pressed, click on the [ICDmini Connect] button again after clicking on the [ICDmini Disconnect] button once.

Appendix A Gang Writer Dynamic Link Library

A.1 Overview

This chapter describes details of the ICD_GANG.DLL functions used for communicating with the ICDmini. Refer to the following descriptions when using the Dynamic Link Library directly.

A.1.1 Example of Function Call

Example:

```
OpenIcdConnection // Establishes a connection with the ICDmini
                  (Execute this function the number of times for the number of ICDminis to
                  be used.)
```

* Repeated part

(Replace the target systems)

```
ResetTarget // Issues a target reset.
GetStatus // Monitors the end of target reset.
CheckTargetConnection // Issues confirmation of connection with the target IC.
GetStatus // Monitors the end of confirmation of connection with the target IC.
StartOperation // Executes the specified operations.
GetStatus // Monitors the end of all specified operations.
```

```
CloseIcdConnection // Closes the connection with the ICDmini
                   (Execute this function the number of times for the number of ICDminis
                   used.)
```

A.2 Function Details

A.2.1 OpenIcdConnection

- **Function**

Opens the USB communication port and establishes communication with the ICDmini. Control is not returned until this function either succeeds or fails.

- **Format**

long OpenIcdConnection (long IcdDeviceNumber);

- **Argument**

IcdDeviceNumber: ICDmini ID number (0 to 999)

- **Return values**

OK
ERROR_PARAMETER
ERROR_ICD_OPEN_CONNECTION

A.2.2 CloseIcdConnection

- **Function**

Corresponds to OpenIcdConnection, and closes communication with the ICDmini. Control is not returned until this function either succeeds or fails.

- **Format**

long CloseIcdConnection (long IcdDeviceNumber);

- **Argument**

IcdDeviceNumber: ICDmini ID number (0 to 999)

- **Return values**

OK
ERROR_PARAMETER
ERROR_ICD_CLOSE_CONNECTION

A.2.3 ResetTarget

- **Function**

Issues a target reset to the target system.
Time-out is 1 second.
Control is returned immediately after this function is called.
Monitor the end of the operation with the GetStatus function.

- **Format**

long ResetTarget (long IcdDeviceNumber);

- **Argument**

IcdDeviceNumber: ICDmini ID number (0 to 999)

- **Return values**

OK
NG
ERROR_PARAMETER
ERROR_ICD_CONNECTION

A.2.4 CheckTargetConnection

- **Function**

Confirms connection with the target system.
Time-out is 1 second.
Control is returned immediately after this function is called.
Monitor the end of the operation with the GetStatus function.

- **Format**

long CheckTargetConnection (long IcdDeviceNumber);

- **Argument**

IcdDeviceNumber: ICDmini ID number (0 to 999)

- **Return values**

OK
NG
ERROR_PARAMETER
ERROR_ICD_CONNECTION

A.2.5 StartOperation

• Function

Executes the specified operation (target reset, erasure, programming, or verification).

Confirms connection with the target system within this function.

Control is returned immediately after this function is called.

Monitor the end of the operation with the GetStatus function.

When performing multiple operations, the GetStatus function returns OK when all operations are finished.

• Format

long StartOperation (long IcdDeviceNumber, long IcdOperation, long TimeOut);

• Arguments

IcdDeviceNumber: ICDmini ID number (0 to 999)

IcdOperation: Sets the operation to execute.

bit 0: Target Reset (1: Executed 0: Not executed)

bit 1: Erasing flash memory (1: Executed 0: Not executed)

bit 2: Programming flash memory (1: Executed 0: Not executed)

bit 3: Verifying flash memory (1: Executed 0: Not executed)

TimeOut: Time-out (1 = 0.1 seconds)

The allowable range is 0 to 72,000 (maximum 120 minutes).

If 0 is specified, there is no time-out.

This setting is used for all flash memory operations (erasing, programming, and verification).

• Return values

OK

NG

ERROR_PARAMETER

ERROR_ICD_CONNECTION

A.2.6 GetStatus

- **Function**

Returns the status of the current operation.

- **Format**

long GetStatus (long IcdDeviceNumber, long* ErrorAddress);

- **Arguments**

IcdDeviceNumber: ICDmini ID number (0 to 999)

ErrorAddress: The error occurrence address is stored.

ResetTarget: Fixed at 0.

CheckTargetConnection: Fixed at 0.

StartErase: Erase start address

StartWrite: Error occurrence address

StartVerify: Error occurrence address

- **Return values**

OK

OPERATION_TARGET_CONNECTION

OPERATION_TARGET_RESET

OPERATION_ERASE

OPERATION_WRITE

OPERATION_VERIFY

ERROR_PARAMETER

ERROR_TIMEOUT_TARGET_CONNECTION

ERROR_TIMEOUT_TARGET_RESET

ERROR_TIMEOUT_ERASE

ERROR_TIMEOUT_WRITE

ERROR_TIMEOUT_VERIFY

ERROR_ICD_CONNECTION

ERROR_TARGET_CONNECTION

ERROR_TARGET_RESET

ERROR_ERASE

ERROR_WRITE

ERROR_VERIFY

A.2.7 GetString

- **Function**

Converts the return code to a character string.

- **Format**

long GetString (long ReturnedCode, char* ReturnedString);

- **Arguments**

ReturnedCode: Return code.

ReturnedString: The character string after converting the return code is stored

The caller must allocate 256 bytes for this area.

“Invalid returned code” is returned if an invalid return code is specified.

- **Return values**

OK

NG

A.2.8 GetConnectedICD

• **Function**

Returns the lists of the serial numbers and the corresponding ID numbers of the ICDminis connected to the PC. When NULL is specified to the IcdSerialNumberList, this function returns the necessary number of buffers. Control is not returned until this function succeeds or fails.

• **Format**

```
long GetConnectedICD ( short* IcdSerialNumberList, long* IcdSerialNumberBufferCount,
                    long* IDNumber, long* IDNumberBufferCount, char* FlsComment,
                    long* FlsCommentBufferCount, char* UserComment,
                    long* UserCommentBufferCount, long* TakenCount, long* ConnectedCount );
```

• **Arguments**

When the function is called with IcdSerialNumberList set to NULL

short* IcdSerialNumberList:	NULL
long* IcdSerialNumberBufferCount:	Returns the number of buffers for the serial number list that must be allocated by the caller.
long* IDNumber:	Ignored
long* IDNumberBufferCount:	Returns the number of buffers for the ID number list that must be allocated by the caller.
char* FlsComment:	Ignored
long* FlsCommentBufferCount:	Returns the number of buffers for FLS program comments that must be allocated by the caller.
char* UserComment:	Ignored
long* UserCommentBufferCount:	Returns the number of buffers for the user program comments that must be allocated by the caller.
long* TakenCount:	Returns the maximum number of serial numbers that can be acquired.
long* ConnectedCount:	Returns the number of ICDminis currently connected via USB.

When the function is called with IcdSerialNumberList set to other than NULL

short* IcdSerialNumberList:	Buffer address for the serial number list. The serial numbers are stored separated by a delimiter 0x0000 (Unicode).
long* IcdSerialNumberBufferCount:	Number of buffers for the serial number list.
long* IDNumber:	Buffer address for the ID number list. The ID numbers are stored in IDNumber corresponding to the serial number list. If an ID number cannot be acquired, -1 is stored in the buffer for that ICDmini.
long* IDNumberBufferCount:	Number of buffers for the ID number list.
char* FlsComment:	Buffer address for FLS program comments.
long* FlsCommentBufferCount:	Number of buffers for FLS program comments.
char* UserComment:	Buffer address for the user program comments. Comments are stored in UserComment separated by a delimiter 0x00. However, if the retrieved data is terminated with 0x00, DLL does not add 0x00.
long* UserCommentBufferCount:	Number of buffers for the user program comments.
long* TakenCount:	Returns the number of the serial numbers that have been stored in the buffer.
long* ConnectedCount:	Returns the number of ICDminis currently connected via USB.

- **Return values**

OK
ERROR_EXECUTING
ERROR_PARAMETER
ERROR_INSUFFICIENT_ICDSERIALNUMBUF
ERROR_INSUFFICIENT_IDNUMBUF
ERROR_INSUFFICIENT_FLSCOMMENTBUF
ERROR_INSUFFICIENT_USERCOMMENTBUF

- **Explanation**

This function executes the following processes internally.

- Opens the USB port.
- Reads the firmware version.
Checks if it is a C17 firmware and the version is Ver. 2.2 or later.
- Acquires the serial number.
- Acquires the ID number.
- Closes the USB port.

- * **Comments for FLS**

FLS comments (commands to be transferred to ICDmini) are stored in FlsComment separated by a delimiter 0x00. However, if the retrieved data is terminated with 0x00, DLL does not add 0x00.

ID number

Numeric characters up to 3 bytes from the beginning of comments (0x30 to 0x39) are regarded as an ID. If there is another character other than numeric within 3 bytes from the head, the numeric characters preceding that character are defined as the ID.

Examples:

- (1) If the comment begins with "12ABC," the ID is "12."
- (2) If the comment begins with "5 43," the ID is "5" as the characters preceding the space are effective.
- (3) If the comment begins with " 123," it is regarded as that this comment has no ID included, as there is no character preceding a space.

Time Out

The comment may set the time-out value for erasing and programming of the flash memory. Note that an ID number must be placed at the beginning.

Example: "123-T60"

The ID number is "123" and the time-out value is 60 seconds.

Others

Refer to readme_e.txt/readme_j.txt attached to the FLS for each S1C17 MCU model as there is model-specific information described.

- **Precautions**

When using this function, the necessary buffers must be allocated by the caller. Acquire the number of buffers taking the precautions shown below into consideration.

- Calling this function with IcdSerialNumberList set to NULL, the necessary number of buffers can be acquired.
- Note that the number of buffers is counted based on the type of the buffer, not the number of bytes of the buffer.
- Note that the necessary number of buffers is not one per one ICDmini. That is, the necessary number of buffers for FLS/user program comments and the necessary number of buffers for the serial number list are the number of characters respectively described in one ICDmini.
- If no ICDmini is connected, nothing is written in each buffer. Then, the necessary number of buffers is 0.

A.2.9 LoadIcdFlsProgram

- **Function**

Loads FLS to the ICDmini with the specified serial number programmed.
Control is returned immediately after this function is called.
The progress status is monitored with the GetIcdStatus function.

- **Format**

long LoadIcdFlsProgram (short* IcdSerialNumber, short* FileName, long EraseAddress, long WriteAddress, long IcdDeviceNumber, long TimeOut, char* Comment);

- **Arguments**

short* IcdSerialNumber: Serial number (Unicode)
It should be within 20 characters and terminated with NULL.

short* FileName: FLS file name in a full-path (Unicode)
It should be terminated with NULL.

long EraseAddress: Erase routine address

long WriteAddress: Write routine address

long IcdDeviceNumber: ID number (0 to 999)

long TimeOut: Time-out value [seconds]

char* Comment: Comments
It should be terminated with NULL.
The comment length including ID number and time-out value must be within 127 bytes excluding NULL.

- **Return values**

OK
ERROR_EXECUTING
ERROR_PARAMETER
ERROR_ICD_OPEN_CONNECTION

- **Explanation**

This function executes the following processes internally.

- Opens the USB port.
- Searches ICDmini with the specified serial number programmed.
- Erases the flash memory.
- Sets entry information.
The ID number, time-out value, and any comments in the format “ID number _ -T Time out value _ any comments” can be set to the comment field of the packet.
- Programs the flash memory.
- Verifies data programmed in the flash memory.
- Closes the USB port.

A.2.10 LoadIcdUserProgram

- **Function**

Loads the user program to the ICDmini with the specified serial number programmed.
Control is returned immediately after this function is called.
The progress status is monitored with the GetIcdStatus function.

- **Format**

```
long LoadIcdUserProgram ( short* IcdSerialNumber, short* FileName, long EraseBlockStart,  
                        long EraseBlockEnd, long FlashAddress, char* Comment );
```

- **Arguments**

short* IcdSerialNumber: Serial number (Unicode)
It should be within 20 characters and terminated with NULL.

short* FileName: User program file name in a full-path (Unicode)
It should be terminated with NULL.

long EraseBlockStart: Erasing start block number

long EraseBlockEnd: Erasing end block number

long FlashAddress: Flash memory top address

char* Comment: Comments
It should be terminated with NULL.
The comment length must be within 127 bytes excluding NULL.

- **Return values**

OK
ERROR_EXECUTING
ERROR_PARAMETER
ERROR_ICD_OPEN_CONNECTION

- **Explanation**

This function executes the following processes internally.

- Opens the USB port.
- Searches the ICDmini with specified serial number programmed.
- Erases the flash memory.
- Sets address information.
Any comments can be set to the comment field in the packet.
- Programs the flash memory.
- Verifies data programmed in the flash memory.
- Closes the USB port.

A.2.11 GetIcdStatus

- **Function**

Returns the progress status of the ICDmini with the specified serial number programmed.

- **Format**

long GetIcdStatus (short* IcdSerialNumber, long* TotalBytes, long* ProgBytes);

- **Arguments**

short* IcdSerialNumber: Serial number (Unicode)

It should be within 20 characters and terminated with NULL.

long* TotalBytes: Returns the number of all bytes.

long* ProgBytes: Returns the number of bytes of which programming has completed.

- **Return values**

OK

OPERATION_FLS_ICD_ERASE
 OPERATION_FLS_ICD_WRITE
 OPERATION_FLS_ICD_VERIFY
 OPERATION_USER_ICD_ERASE
 OPERATION_USER_ICD_WRITE
 OPERATION_USER_ICD_VERIFY

ERROR_PARAMETER

ERROR_TIMEOUT_FLS_ICD
 ERROR_TIMEOUT_USER_ICD

ERROR_MOT_OPEN_FLS
 ERROR_MOT_FORMAT_FLS
 ERROR_MOT_SIZE_OVER_FLS
 ERROR_MOT_ADDR_FLS
 ERROR_MOT_OPEN_USER_PRG
 ERROR_MOT_FORMAT_USER_PRG
 ERROR_MOT_SIZE_OVER_USER_PRG
 ERROR_MOT_ADDR_USER_PRG

ERROR_ICD_OPEN_CONNECTION
 ERROR_ICD_CONNECTION
 ERROR_VERIFY_FLS_ICD
 ERROR_VERIFY_USER_ICD

ERROR_LOAD_FLS_ICD
 ERROR_LOAD_USER_ICD

- **Explanation**

By referring TotalBytes and ProgBytes, the caller can determine how much programming has been progressed.

Appendix A Gang Writer Dynamic Link Library

• Return codes

OK	0x00	Terminated normally.
NG	0x01	An error has occurred.
ERROR_EXECUTING	0x03	The request cannot be accepted as the ICDmini with the specified number programmed is busy.
ERROR_TIMEOUT_TARGET_CONNECTION	0x12	A time-out occurred while the function attempted to connect with the target system.
ERROR_TIMEOUT_TARGET_RESET	0x13	A time-out occurred while the function attempted to reset the target system.
ERROR_TIMEOUT_ERASE	0x14	A time-out occurred while the function attempted to erase the flash memory.
ERROR_TIMEOUT_WRITE	0x15	A time-out occurred while the function attempted to program the flash memory.
ERROR_TIMEOUT_VERIFY	0x16	A time-out occurred while the function attempted to verify the flash memory.
ERROR_TIMEOUT_FLS_ICD	0x17	A time-out occurred while the function attempted to load the FLS program to the ICDmini.
ERROR_TIMEOUT_USER_ICD	0x18	A time-out occurred while the function attempted to load the user program to the ICDmini.
ERROR_ICD_OPEN_CONNECTION	0x21	The ICDmini cannot be connected.
ERROR_ICD_CONNECTION	0x22	The ICDmini has already been disconnected.
ERROR_ICD_CLOSE_CONNECTION	0x29	The ICDmini cannot be disconnected.
ERROR_TARGET_CONNECTION	0x32	The target system is disconnected.
ERROR_TARGET_RESET	0x33	The target system does not respond after a target reset is issued.
ERROR_ERASE	0x44	An error occurred while the flash memory was being erased.
ERROR_WRITE	0x45	An error occurred while the flash memory was being programmed.
ERROR_VERIFY	0x46	An error occurred while flash memory was being verified.
ERROR_LOAD_FLS_ICD	0x47	An error occurred while the FLS program was being loaded to the ICDmini.
ERROR_LOAD_USER_ICD	0x48	An error occurred while the user program was being loaded to the ICDmini.
ERROR_PARAMETER	0x50	The specified argument is invalid.
ERROR_INSUFFICIENT_ICDSERIALNUMBUF	0x51	The serial number buffer size is insufficient.
ERROR_INSUFFICIENT_IDNUMBUF	0x52	The ID number buffer size is insufficient.
ERROR_INSUFFICIENT_FLSCOMMENTBUF	0x53	The FLS comment buffer size is insufficient.
ERROR_INSUFFICIENT_USERCOMMENTBUF	0x54	The user program comment buffer size is insufficient.
ERROR_MOT_OPEN_FLS	0x60	The FLS program file in the Motorola format cannot be opened.
ERROR_MOT_FORMAT_FLS	0x61	The FLS program file is not a Motorola format file.
ERROR_MOT_SIZE_OVER_FLS	0x62	The FLS program file in the Motorola format exceeds the allowable size.
ERROR_MOT_ADDR_FLS	0x63	The addresses in the FLS program file exceed the allowable maximum address of a Motorola format file.
ERROR_MOT_OPEN_USER	0x64	The user program file in the Motorola format cannot be opened.
ERROR_MOT_FORMAT_USER	0x65	The user program file is not a Motorola format file.
ERROR_MOT_SIZE_OVER_USER	0x66	The user program file in the Motorola format exceeds the allowable size.
ERROR_MOT_ADDR_USER	0x67	The addresses in the user program file exceed the allowable maximum address of a Motorola format file.
ERROR_VERIFY_FLS_ICD	0x68	An error occurred while the FLS program loaded into the ICDmini was being verified.
ERROR_VERIFY_USER_ICD	0x69	An error occurred while the user program loaded into the ICDmini was being verified.
OPERATION_TARGET_CONNECTION	0x82	Connection with the target system is being established.
OPERATION_TARGET_RESET	0x83	The target system is currently being reset.
OPERATION_ERASE	0x84	The flash memory is currently being erased.
OPERATION_WRITE	0x85	The flash memory is currently being programmed.
OPERATION_VERIFY	0x86	The flash memory is currently being verified.
OPERATION_FLS_ICD_ERASE	0x87	The ICDmini is erasing the area for loading an FLS program.
OPERATION_FLS_ICD_WRITE	0x88	The ICDmini is loading an FLS program.
OPERATION_FLS_ICD_VERIFY	0x89	The ICDmini is verifying the loaded FLS program.
OPERATION_USER_ICD_ERASE	0x8a	The ICDmini is erasing the area for loading a user program.
OPERATION_USER_ICD_WRITE	0x8b	The ICDmini is loading a user program.
OPERATION_USER_ICD_VERIFY	0x8c	The ICDmini is verifying the loaded user program.

A.3 Restrictions

Do not connect two or more ICDminis with the same ID number set at the same time. If connected, only the ICDmini first connected is recognized.

A.4 Measurement Results

For reference, the table below lists the measured times required for loading a 3M bytes of user program (*.sa) from the PC to the ICDmini.

Table A.4.1 ICDmini Loading Time List

Operation	Time
Erasing	34 seconds
Loading	1 minute 36 seconds
Verifying	3 minutes 14 seconds
Total time	5 minutes 24 seconds

* When the user program is loaded using the LoadIcdUserProgram() function in the dynamic link library (ICD_GANG.DLL)

Appendix B List of Configurations by Model

Appendix B List of Configurations by Model

Table B.1 List of Configurations by Model (1)

Model name	Erase/Write Program					User Program			Available ICDmini hardware version	Available ICDmini firmware version	ICDmini DIP switch (SW8) & FLASH VCC OUT connection ^{*3}	Remarks
	File (1) ^{*1}	Erase Routine Address (2)	Write Routine Address (3)	Time out (4) ^{*2}	Comment (5)	Start Block No. of Erase (7)	End Block No. of Erase (8)	Flash Memory Top Address (9)				
S1C17501	fwr17501v11.saf	0x44	0x78	60		0	0	0x20000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17554	fwr17554v11.saf	0xa2	0x6c	60	"-v7.5-7.0"	1	32	0x8000	2.0	3.0 or later	ON Connect VPP	^{*4}
	fwr17554_lowv11.saf	0xa2	0x6c	60	"-v7.5-7.0"	1	32	0x8000	2.0	3.0 or later	ON Connect VPP	^{*5}
	fwr17554_highv11.saf	0xa2	0x6c	60	"-v7.5-7.0"	1	32	0x8000	2.0	3.0 or later	ON Connect VPP	^{*6}
S1C17555	fwr17555_2kbv11.saf	0x42	0x76	60	"-v7.5-7.5 -s128"	1	64	0x8000	2.0	3.3 or later	ON Connect VPP	^{*7}
	fwr17555_128bv11.saf	0x42	0x76	60	"-s128"	1	64	0x8000	1.0, 1.1, 2.0	3.3 or later	OFF No connection	^{*8}
S1C17564	fwr17564v11.saf	0x9c	0x68	60	"-v7.5-7.0"	1	32	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17565	fwr17565_2kbv11.saf	0x42	0x76	60	"-v7.5-7.5 -s128"	1	64	0x8000	2.0	3.3 or later	ON Connect VPP	^{*7}
	fwr17565_128bv11.saf	0x42	0x76	60	"-s128"	1	64	0x8000	1.0, 1.1, 2.0	3.3 or later	OFF No connection	^{*8}
S1C17589	fwr17589v11.saf	0x44c	0x444	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17601	fwr17601v11.saf	0x3f8	0x430	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17602	fwr17602v11.saf	0x48	0x80	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17604	fwr17604v11.saf	0x88	0xc0	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	

Table B.2 List of Configurations by Model (2)

Model name	Erase/Write Program					User Program			Available ICDmini hardware version	Available ICDmini firmware version	ICDmini DIP switch (SW8) & FLASH VCC OUT connection ³	Remarks
	File (1) ¹	Erase Routine Address (2)	Write Routine Address (3)	Time out (4) ²	Comment (5)	Start Block No. of Erase (7)	End Block No. of Erase (8)	Flash Memory Top Address (9)				
S1C17611	fwr17611v11.saf	0x3f8	0x430	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17621	fwr17621v11.saf	0x28	0x60	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17622	fwr17622v11.saf	0x48	0x80	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17624	fwr17624v11.saf	0x88	0xc0	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17651	fwr17651v11_4.saf	0x9c	0x68	60	"-v7.5-7.0 -s128"	1	4	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17653	fwr17653v11_4.saf	0x9c	0x68	60	"-v7.5-7.0 -s128"	1	4	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17656	fwr17656v11.saf	0xcc	0xc4	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17701	fwr17701v11.saf	0x48	0x80	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17702	fwr17702v11.saf	0x88	0xc0	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17703	fwr17703v11.saf	0x108	0x140	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17704	fwr17704v11.saf	0x48	0x80	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17705	fwr17705v11.saf	0x208	0x240	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17706	fwr17706v11.saf	0x408	0x440	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17711	fwr17711v11.saf	0x48	0x80	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	
S1C17801	fwr17801v11.saf	0x40	0x74	60		0	0	0x20000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	

Appendix B List of Configurations by Model

Table B.3 List of Configurations by Model (3)

Model name	Erase/Write Program					User Program			Available ICDmini hardware version	Available ICDmini firmware version	ICDmini DIP switch (SW8) & FLASH VCC OUT connection ^{*3}	Remarks
	File (1) ^{*1}	Erase Routine Address (2)	Write Routine Address (3)	Time out (4) ^{*2}	Comment (5)	Start Block No. of Erase (7)	End Block No. of Erase (8)	Flash Memory Top Address (9)				
S1C17955	fwr17955_2kbv11.saf	0x42	0x76	60	"-v7.5-7.5 -s128"	1	64	0x8000	2.0	3.3 or later	ON Connect to VPP	^{*7}
	fwr17955_128bv11.saf	0x42	0x76	60	"-s128"	1	64	0x8000	1.0, 1.1, 2.0	3.3 or later	OFF No connection	^{*8}
S1C17965	fwr17965_2kbv11.saf	0x42	0x76	60	"-v7.5-7.5 -s128"	1	64	0x8000	2.0	3.3 or later	ON Connect to VPP	^{*7}
	fwr17965_128bv11.saf	0x42	0x76	60	"-s128"	1	64	0x8000	1.0, 1.1, 2.0	3.3 or later	OFF No connection	^{*8}
S1C17F13	fwr17f13_2kbv11.saf	0x50	0x84	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.0 or later	ON Connect to VPP	^{*7}
	fwr17f13_128bv11.saf	0x50	0x84	60		0	0	0x8000	1.0, 1.1, 2.0	3.0 or later	OFF No connection	^{*8}
S1C17F57	fwr17f57v11.saf	0x9c	0x68	60	"-v7.5-7.0 -s128"	1	8	0x8000	2.0	3.3 or later	ON Connect to VPP	
S1C17M01	fwr17m01v11.saf	0x42	0x76	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect to VPP	
S1C17W03	fwr17w03v11.saf	0xcc	0xc4	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17W04	fwr17w04v11.saf	0xcc	0xc4	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17W13	fwr17w13v11.saf	0x48	0x7c	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17W14	fwr17w14v11.saf	0x44c	0x444	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	
S1C17W15	fwr17w15v11.saf	0x48	0x7c	60	"-v7.5-7.5 -s128"	0	0	0x8000	2.0	3.3 or later	ON Connect to VPP	
S1C17W16	fwr17w16v11.saf	0x44c	0x444	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	

Table B.4 List of Configurations by Model (4)

Model name	Erase/Write Program					User Program			Available ICDmini hardware version	Available ICDmini firmware version	ICDmini DIP switch (SW8) & FLASH VCC OUT connection ^{*3}	Remarks
	File (1) ^{*1}	Erase Routine Address (2)	Write Routine Address (3)	Time out (4) ^{*2}	Comment (5)	Start Block No. of Erase (7)	End Block No. of Erase (8)	Flash Memory Top Address (9)				
S1C17W18	fwr17w18_2kbv11.saf	0x48	0x7c	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.3 or later	ON Connect VPP	※ ⁷
	fwr17w18_16bv11.saf	0x48	0x7c	60		0	0	0x8000	1.0, 1.1, 2.0	3.3 or later	OFF No connection	※ ⁸
S1C17W22	fwr17w22v11.saf	0x48	0x7c	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.3 or later	ON Connect to VPP	
S1C17W23	fwr17w23v11.saf	0x48	0x7c	60	"-v7.5-7.5"	0	0	0x8000	2.0	3.3 or later	ON Connect to VPP	

Note:

*1 The files are stored in the C:\EPSON\C17GangWriter\mcu_model folder (default) by model.

*2 Time-out time (Although any time can be set, the table lists the maximum time.)

*3 This column shows the ICDmini DIP switch (SW8) setting and whether the FLASH VCC OUT pin of the flash memory programming power supply connector must be connected to the VPP pin of the target MCU or not.

*4 Settings for the model with 8.1 to 16.0 MHz operating frequency range

*5 Settings for the model with 4.0 to 8.0 MHz operating frequency range

*6 Settings for the model with 16.1 to 24.0 MHz operating frequency range

*7 When the FLASH VCC OUT pin output of the ICDmini Ver. 2.0 is used as the flash memory programming power supply

*8 When the voltage regulator/booster embedded in the target MCU is used as the flash memory programming power supply



International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.

214 Devcon Drive,
San Jose, CA 95112, USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH

Riesstrasse 15, 80992 Munich,
GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.

7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-8522-1199 FAX: +86-10-8522-1125

SHANGHAI BRANCH

7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5577 FAX: +86-21-5423-4677

SHENZHEN BRANCH

12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.

1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORP.**KOREA OFFICE**

19F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

SEIKO EPSON CORP.**MICRODEVICES OPERATIONS DIVISION****IC Sales & Marketing Department**

421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117