EPSON

CMOS 16-BIT SINGLE CHIP MICROCONTROLLER

Multi Programmer Ver.4.0 (S5U1C17000Y24) User Manual

SEIKO EPSON CORPORATION

Rev.2.03

NOTICE: PLEASE READ THE FOLLOWING NOTICE CAREFULLY BEFORE USING THIS DOCUMENT

The contents of this document are subject to change without notice.

1. This document may not be copied, reproduced, or used for any other purpose, in whole or in part, without the consent of the Seiko Epson Corporation ("Epson").
2. Before purchasing or using Epson products, please contact our sales representative for the latest information and always be sure to check the latest information published on Epson's official web sites and other sources.
3. Information provided in this document such as application circuits, programs, usage, etc., are for reference purposes only. Using the application circuits, programs, usage, etc. in the design of your equipment or systems is your own responsibility. Epson makes no guarantees against any infringements or damages to any third parties' intellectual property rights or any other rights resulting from the information. This document does not grant you any licenses, intellectual property rights or any other rights with respect to Epson products owned by Epson or any third parties.
4. Epson is committed to constantly improving quality and reliability, but semiconductor products in general are subject to malfunction and failure. By using Epson products, you shall be responsible for your hardware. Software and systems must be designed well enough to prevent death or injury as well as any property damage even if any of the malfunctions or failures might be caused by Epson products. When designing your products using Epson products, please be sure to check and comply with the latest information regarding Epson products (this document, specifications, data sheets, manuals, Epson's web site, etc.). When using the information included above materials such as product data, charts, technical contents, programs, algorithms and application circuit examples, you shall evaluate your products both on a stand-alone basis as well as within your overall systems. You shall be solely responsible for deciding whether or not to adopt and use Epson products.
5. Epson has prepared this document and programs provided in this document carefully to be accurate and dependable, but Epson does not guarantee that the information and the programs are always accurate and complete. Epson assumes no responsibility for any damages which you incur due to misinformation in this document and the programs.
6. No dismantling, analysis, reverse engineering, modification, alteration, adaptation, reproduction, etc., of Epson products is allowed.
7. Epson products have been designed, developed and manufactured to be used in general electronic applications (office equipment, communications equipment, measuring instruments, home electronics, etc.) ("General Purpose") and applications which is individually listed in this document or designated by Epson ("Designated Purpose"). Epson products are NOT intended for any use beyond the General Purpose and Designated Purpose uses that requires particular/higher quality or reliability in order to refrain from causing any malfunction or failure leading to death, injury, serious property damage or severe impact on society, including, but not limited to those listed below ("Particular Purpose"). Therefore, you are advised to use Epson products only for General Purpose and Designated Purpose uses. Should you desire to buy and use Epson products for a Particular Purpose, Epson makes no warranty and disclaims with respect to Epson products, whether express or implied, including without limitation any implied warranty of merchantability or fitness for any Particular Purpose. Please be sure to contact our sales representative and obtain approval in advance.

   [Examples of Particular Purpose]

   Space equipment (artificial satellites, rockets, etc.) /
   Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.) /
   Medical equipment / Relay equipment to be placed on ocean floor /
   Power station control equipment / Disaster or crime prevention equipment / Traffic control equipment / Financial equipment

   Other applications requiring similar levels of reliability as those listed above. Please be sure to contact our sales representative for details of the other applications.
8. Epson products listed in this document and our associated technologies shall not be used in any equipment or systems that laws and regulations in Japan or any other countries prohibit to manufacture, use or sell. Furthermore, Epson products and our associated technologies shall not be used for developing weapons of mass destruction, or any other military purposes or applications. If exporting Epson products or our associated technologies, you shall comply with the Foreign Exchange and Foreign Trade Control Act in Japan, Export Administration Regulations in the U.S.A. (EAR) and other export-related laws and regulations in Japan and any other countries and follow the required procedures as provided by the relevant laws and regulations.
9. Epson assumes no responsibility for any damages (whether direct or indirect) caused by or in relation with your non-compliance with the terms and conditions in this document.
10. Epson assumes no responsibility for any damages (whether direct or indirect) incurred by any third party that you assign, transfer, loan, etc., Epson products to.
11. For more details or other concerns about this document, please contact our sales representative.
12. Company names and product names listed in this document are trademarks or registered trademarks of their respective companies.

Evaluation board/kit and Development tool important notice

1. Epson evaluation board/kit or development tool is designed for use for engineering evaluation, demonstration, or development purposes only. Do not use it for other purposes. It is not intended to meet the requirements of design for finished products.
2. Epson evaluation board/kit or development tool is intended for use by an electronic engineer and is not a consumer product. The user should use it properly and in a safe manner. Epson dose not assume any responsibility or liability of any kind of damage and/or fire coursed 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 Epson evaluation board/kit or development tool may be changed without any notice.

Rev. e1.4, 2023. 4

Table of Contents

1. Overview

Multi Programmer is software for writing simultaneously to multiple target MCUs (multi-programming) using a PC and S5U1C17001H (ICDmini) units. Multi Programmer consists of a PC application and a DLL for creating the customer's own applications. User program data provided by the customer can be multi-programmed for up to ten target MCUs.

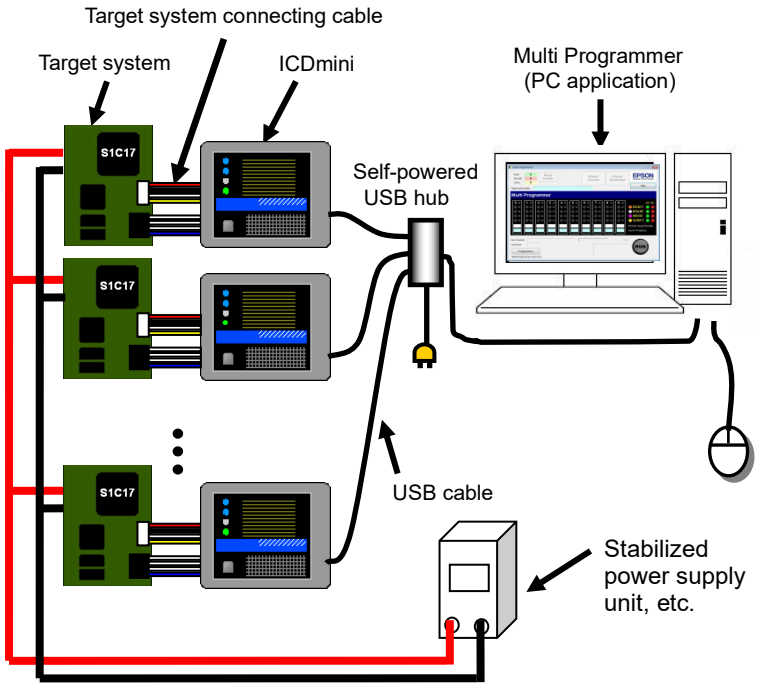


Figure 1.1 Multi-programming configuration

The items required for multi-programming can be obtained from the sources listed below. Obtain these items beforehand.

Table 1.1 List of required components

Item	Source
PC Memory: The Multi Programmer package requires at least 40 MB of free hard disk space. Confirm that sufficient disk space is available. Operating system: Compatible with English and Japanese versions of Windows 7 or later. Also requires the .NET Framework 3.5 application operating environment.	To be provided by the customer
USB hub (Requires a self-powered hub capable of supplying up to 500 mA for each ICDmini.)	To be provided by the customer
User program data (A .psa file debugged using GNU17)	To be provided by the customer
Power supply unit	To be provided by the customer
Required number of ICDminis (Firmware version 3.0 or later for ICDmini Ver. 1.0/1.1/2.0)	Please contact your Seiko Epson sales representative.

2. Overall Flow

The multi-programming process involves three main steps: initial preparations, parameter file creation, and multi-programming. The flowcharts for these three steps are shown below. The items within the flowcharts correspond to the respective sections of this manual.

2.1 Flowchart for initial preparations

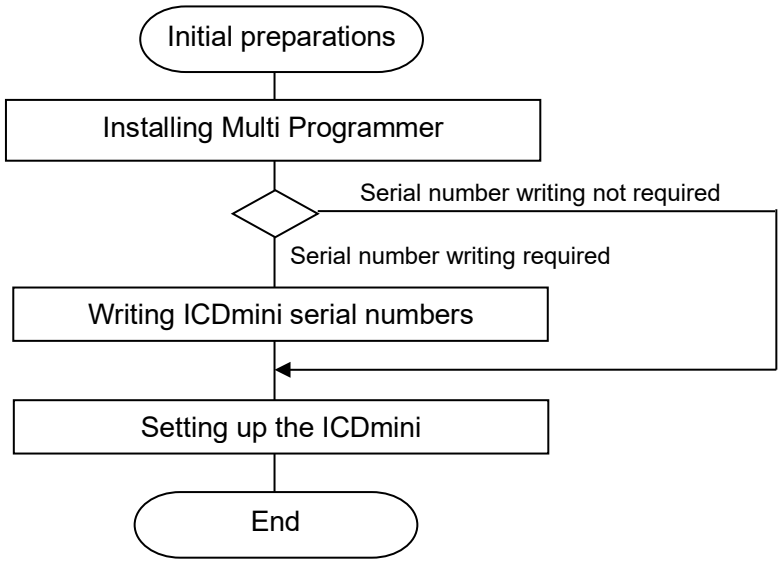


Figure 2.1.1 Flowchart for initial preparations

2.2 Flowchart for parameter file creation

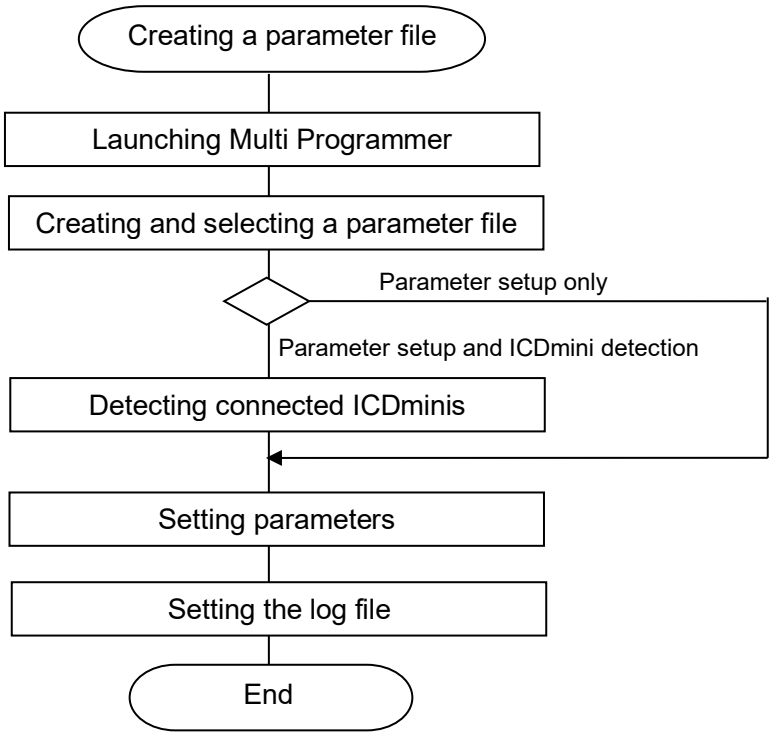


Figure 2.2.1 Flowchart for parameter file creation

2.3 Multi-programming flowchart

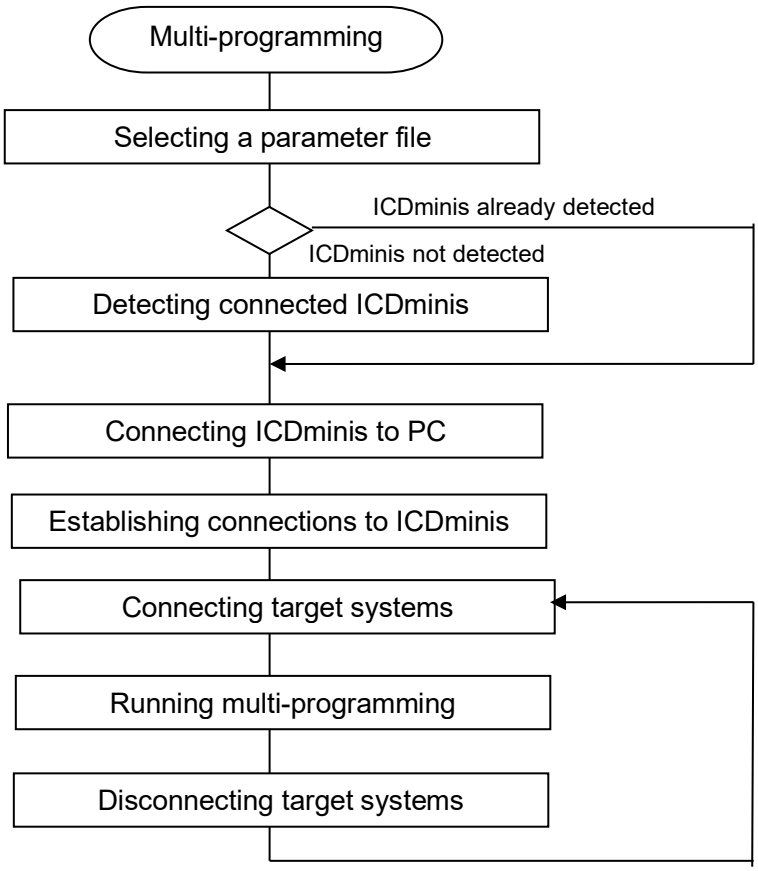


Figure 2.3.1 Multi-programming flowchart

3. Initial Preparations

3.1 Installing Multi Programmer

Prepare and verify files before multi-programming. Download the following software from the Seiko Epson website and install to a folder.

Website URL: http://global.epson.com/products_and_drivers/semicon/products/micro_controller/16bit/mp_tool.html

- Multi Programmer package
- Model-specific information file

The Multi Programmer package includes model-specific information files. If the model-specific information files are updated or if new model files are added, the model-specific information files are registered individually. Download the model-specific file for the machine to be used, if registered.

Decompress the downloaded model-specific information files to the following folder. If an older model-specific information file version has already been installed, overwrite this with the newly downloaded model-specific information file.

C:¥EPSON¥C17Multi Programmer¥mcu_model

The portions of the path underlined above assume that Multi Programmer has been installed in the default folder. If you installed Multi Programmer to a different drive and folder, that specified drive and folder will be used.

3.1.2 Configuration of folders installed

The folders are configured as follows after installing the Multi Programmer package:

```
+ EPSON
    + C17MultiProgrammer
        C17MultiProgrammer.exe      : Multi Programmer
        C17SNWriter.exe             : ICDmini Serial No. Write
        MultiProgrammer.dll         : Multi Programmer Dynamic Link Library
        icdmini2.dll                : ICDmini Ver. 1.0/1.1/2.0 control library
        icdmini3.dll                : ICDmini Ver. 3.x control library
        License.txt                 : User license
        uninstall.exe               : Uninstaller
        + doc                       : Manuals and instructions
        + mcu_model                 : Model-specific information files
        + utility
          + drv_usb                 : USB driver
```

3.1.3 Installing the USB driver

The Multi Programmer package includes the S5U1C17001H (ICDmini) driver. Install the ICDmini driver as required, following the procedure given below.

(1) Connect the ICDmini to the host PC using a USB cable.。
The new device is detected by Windows, and a message appears.

(2) Select as shown below, as instructed by the message.
→ "Locate and install driver software"
→ "Browse my computer for driver software"
→ C:¥EPSON¥C17MultiProgrammer¥utility¥drv_usb

(3) Install the driver selected.
If the message "Windows can't verify the publisher of this software" appears, select "Install this driver software anyway".

Once the USB driver is successfully installed, Windows will recognize the ICDmini and display it as follows in Device Manager:

- For ICDmini Ver. 3:
  ICDmini3 Device
  + ICDmini3

- For ICDmini Ver. 1.0/1.1/2:
  ICD mini Device
  + ICD mini

3.2 Writing ICDmini serial numbers

Multi Programmer reads in the serial numbers of ICDminis connected to identify them. Thus, the serial numbers must be written to the ICDminis beforehand.

Check the hardware version written on the rear of the ICDmini to be used. If the hardware version is 1.1 or earlier or 3.0 or later, or if no serial number has been specified, the serial number must be written using ICDmini Serial No. Writer.

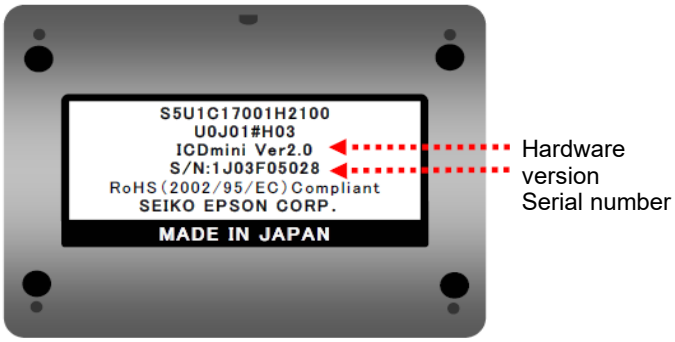


Figure 3.2.1 Rear of ICDmini

With one ICDmini connected to the PC, select [EPSON MCU] → [S1C17 Multi Programmer] → [C17SNwrite] from the Start menu. Enter the ICDmini serial number for "Serial No." in the window that appears, then click the "WRITE" button. Up to 10 characters can be entered. Click the "READ" button to read the serial number for the ICDmini currently connected.

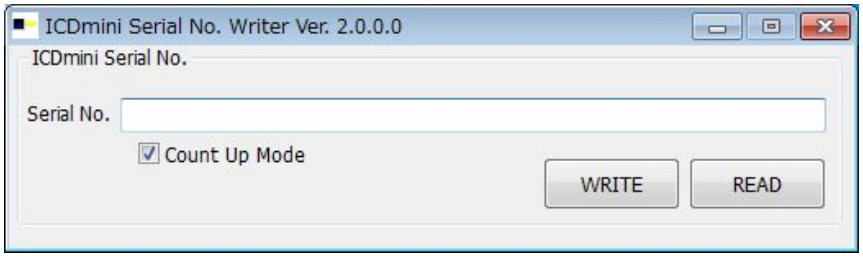


Figure 3.2.2 ICDmini Serial No. Writer startup window

Any serial number can be entered, provided there is no duplication. We recommend entering the serial number indicated on the rear of the ICDmini. Depending on the ICDmini, the serial number indicated on the rear of the ICDmini may be 10 characters or more. In that case, please omit it to 10 characters.

If the ICDmini Ver. number is 1.0/1.1/2.0 and the ICDmini is not recognized, press the reset button on the ICDmini.

Caution! Make sure DIP switch No. 7 on ICDmini Ver. 1.0/1.1/2.0 is set to "On", and return to "Open" once the serial number has been written.

Listed below are error messages you may encounter using the ICDmini Serial No. Writer.

Table 3.2.1 Error messages

Error message	Meaning
It failed in the connection with USB.	The ICDmini is not connected to the PC. Connect the ICDmini to the PC.
Unable to find WINUSB.DLL	The USB driver has not been installed. Install the USB driver.

3.3 Setting up the ICDmini

If using ICDmini Ver. 3.x, no setup is required. If using ICDmini Ver. 1.0/1.1/2.0, the DIP switches need to be set as shown below. Set the DIP switches according to the interface voltage level of the target system.

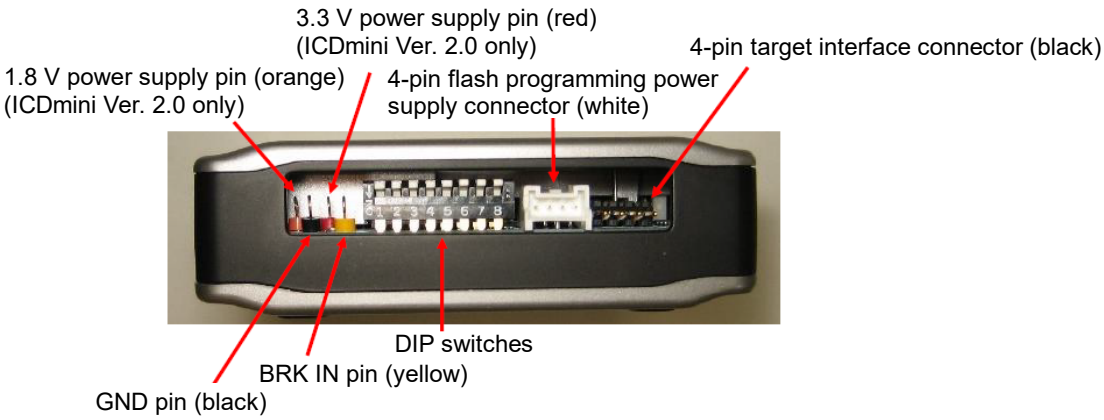
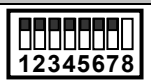
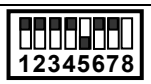
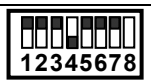


Figure 3.3.1 ICDmini left-hand side panel

Table 3.3.1 ICDmini DIP switch settings

Target system interface voltage level	Switch positions
3.3 V	
1.8 V	
Voltage input from target	

Set DIP switch No. 8 to "On" only when using ICDmini Ver. 2.0 and with a model that requires a flash programming power supply (VPP).

4. Creating a Parameter File

4.1 Launching Multi Programmer

Launch Multi Programmer from the PC Start menu by selecting [EPSON MCU] → [S1C17 Multi Programmer] → [C17 Multi Programmer]. Click the [Configurations] button to detect the connected ICDmini and to set the parameters required for multi-programming.

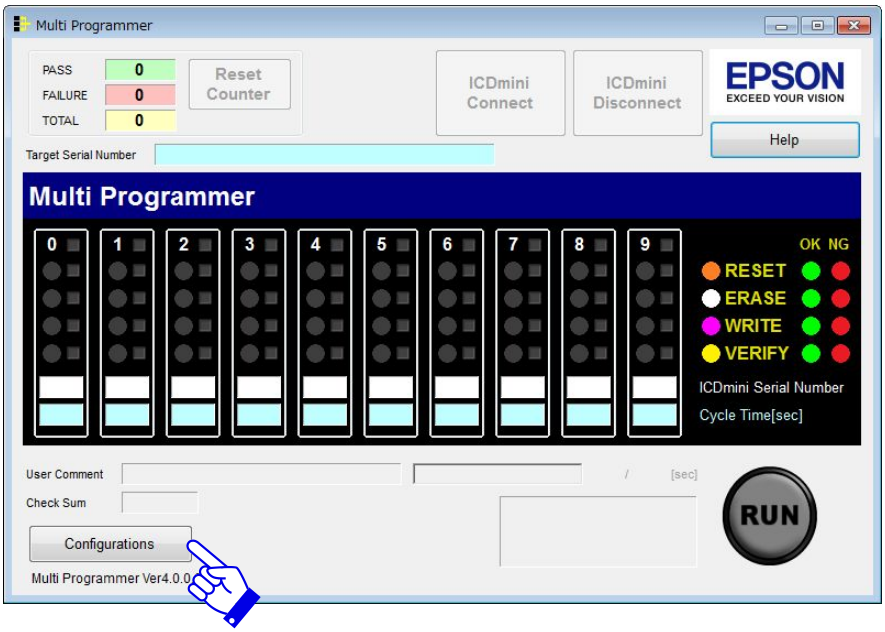


Figure 4.1.1 Multi Programmer startup window

When you start up for the first time, all buttons other than the Parameter File selection button will be disabled. For subsequent startups, the parameter file previously used will be selected and the corresponding details displayed.

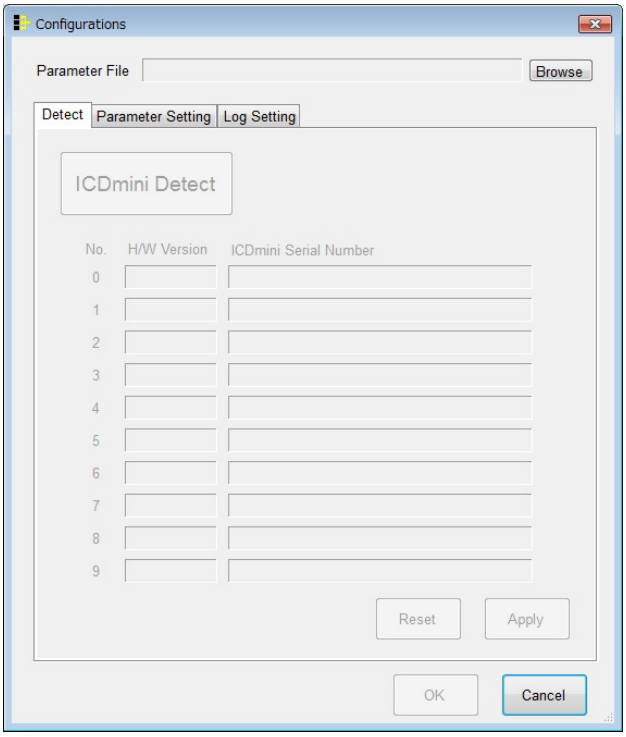


Figure 4.1.2 Configurations window when starting up for the first time

4.2 Creating and selecting a parameter file

Click the Parameter File [Browse] button and create a new parameter file or select an existing parameter file.

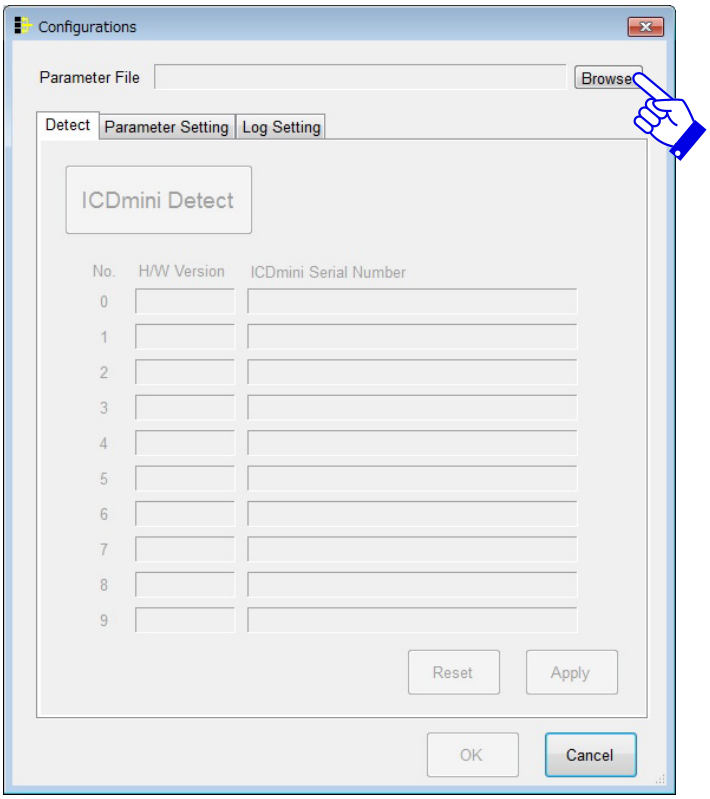


Figure 4.2.1 Parameter file selection window

The contents of the following folder are opened by default. The parameter file name is set as MultiProgrammer.ini. This can be modified, if needed.

C:¥Users¥*Username*¥MultiProgrammer.ini

The following message dialog appears if the specified file does not exist:

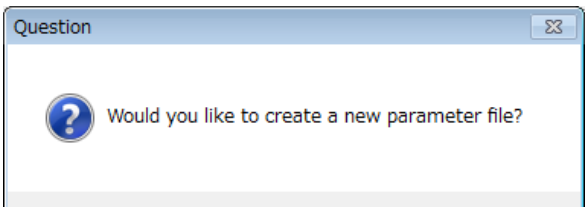


Figure 4.2.2 Message display window

Click the [Yes] button to create a new file with the specified name.

If an existing parameter file is selected and no changes to the settings are required, click the [OK] button.

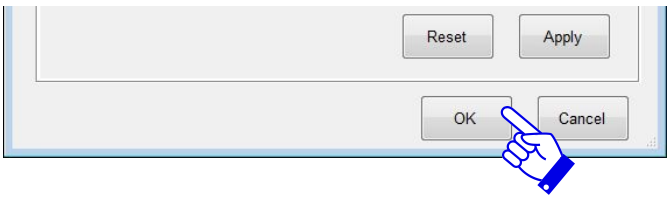


Figure 4.2.3 Importing parameter file

4.3 Detecting connected ICDminis

Selecting a file for Parameter File enables the Detect tab. The ICDminis must be detected before performing the procedure in "5.2 Establishing connection with ICDminis". To detect the ICDminis currently connected to the PC, click the [ICDmini Detect] button.

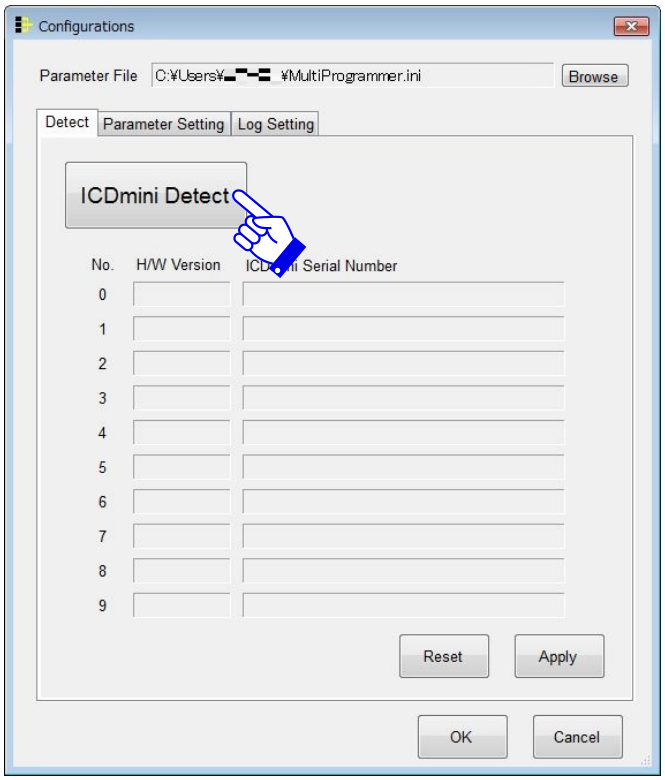


Figure 4.3.1 Detect tab setup window (1)

The following message dialog appears if any ICDminis are newly detected. Click the [OK] button.

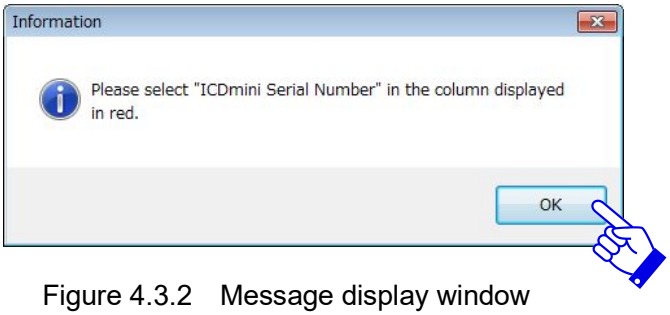


Figure 4.3.2 Message display window

When the ICDminis are detected, the corresponding hardware versions and serial numbers are listed in red for the connection numbers in ascending order from 0 to 9. The connection numbers link the ICDminis detected to the display in the main window. In this example, two ICDminis of Ver. 2.0 have been detected.

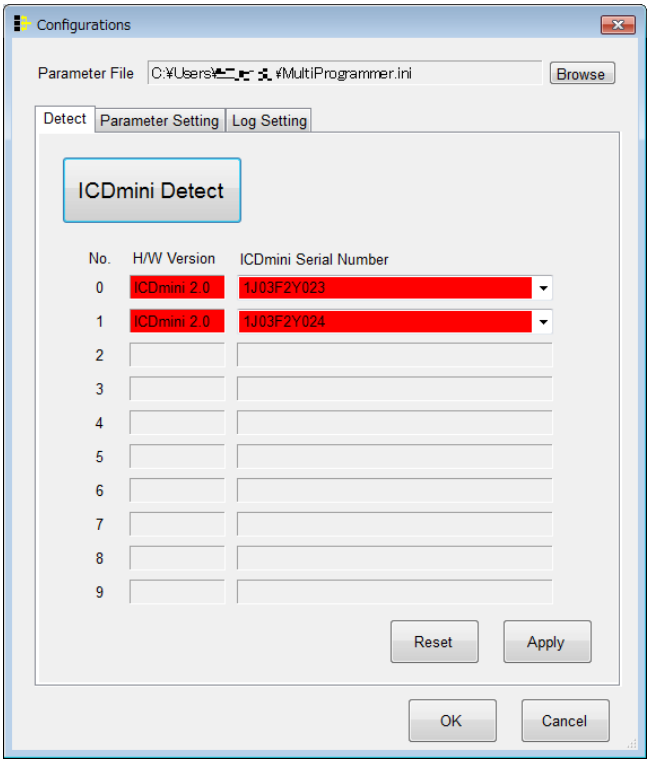


Figure 4.3.3 Detect tab setup window (2)

To alter connection numbers, select the ICDmini to be assigned to that number from the pull-down menu.

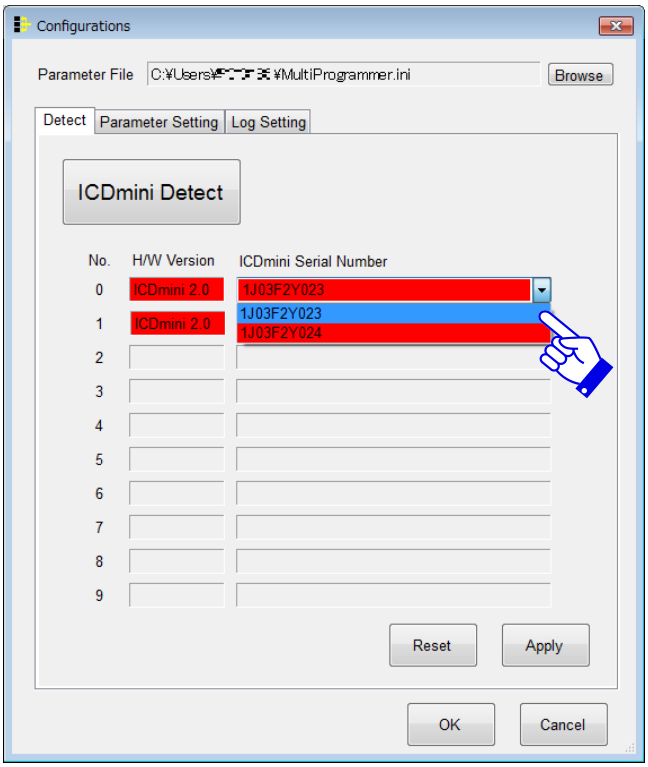


Figure 4.3.4 Detect tab setup window (3)

Click the [Apply] button to update the selected parameter file to reflect the detection results.

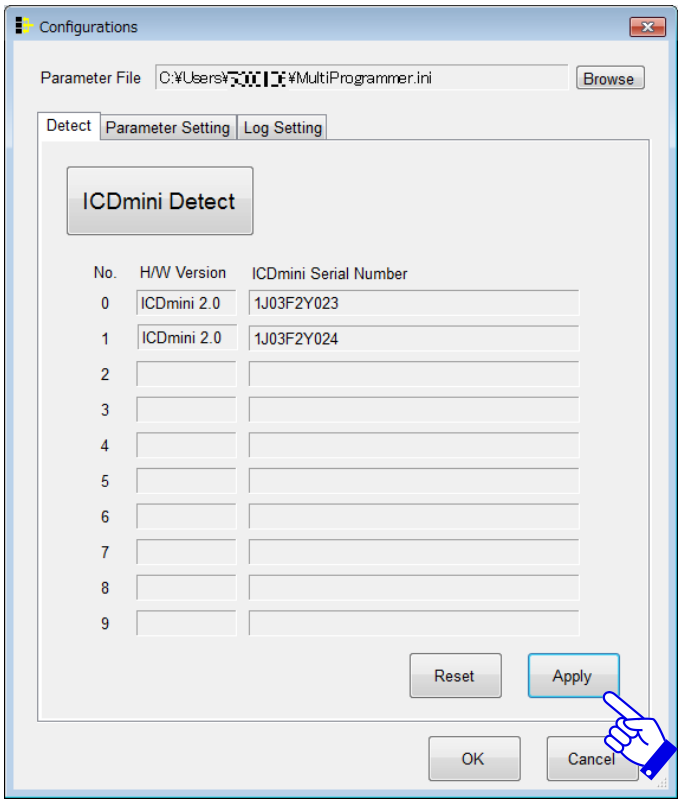


Figure 4.3.5 Detect tab setup window (4)

4.3.1 Redetecting connected ICDminis

If ICDminis connected to the PC have been added or altered, the ICDminis connected to the PC must be redetected. Click the [ICDmini Detect] button to redetect the ICDminis.

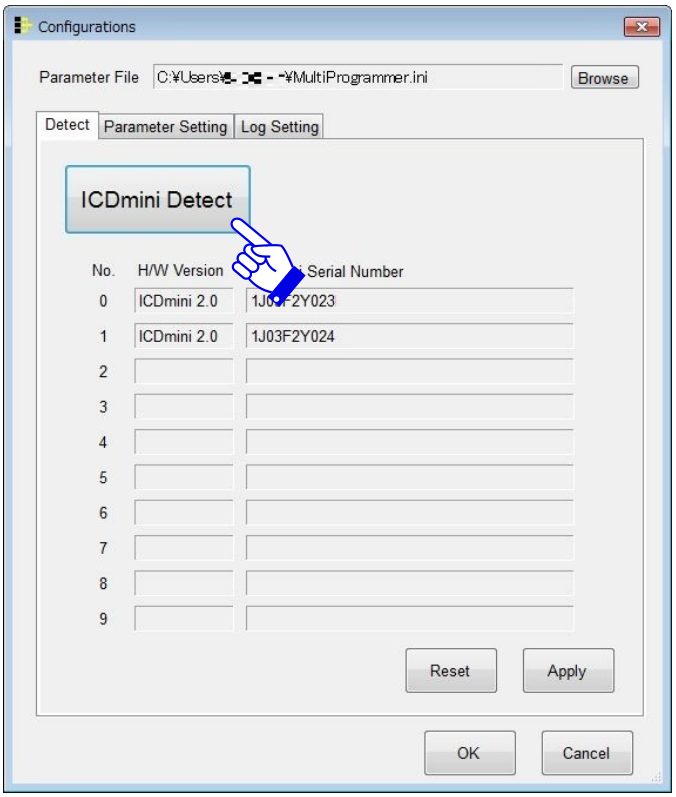


Figure 4.3.6 Detect tab setup window (5)

The messages displayed for the respective statuses are as follows:

Table 4.3.1 Messages displayed when redetecting ICDminis

Status	Message displayed
When there is no change in the ICDminis	There is no change from the previous detection result.
When ICDminis have been added or changed	Please select “ICDmini Serial Number” in the colum displayed in red.
When ICDminis have been removed	The number of connected “ICDmini” has decreased.

In the example shown below, steps ① to ③ describes the applicable procedure when the ICDmini (serial number 1J03F2Y023) connected to connection number 1 has been replaced with a different ICDmini (serial number 1J03F2Y025).

Step ①
The following message dialog appears when a new ICDmini is detected. Click the [OK] button:

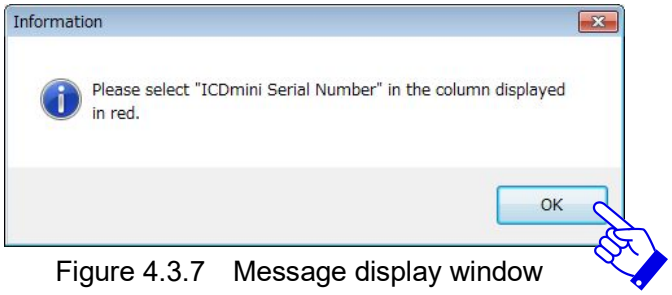


Figure 4.3.7 Message display window

Step ②
Locations where the redetected results differ from the previously detected results are displayed in red. Select the ICDmini newly assigned to the corresponding connection number.

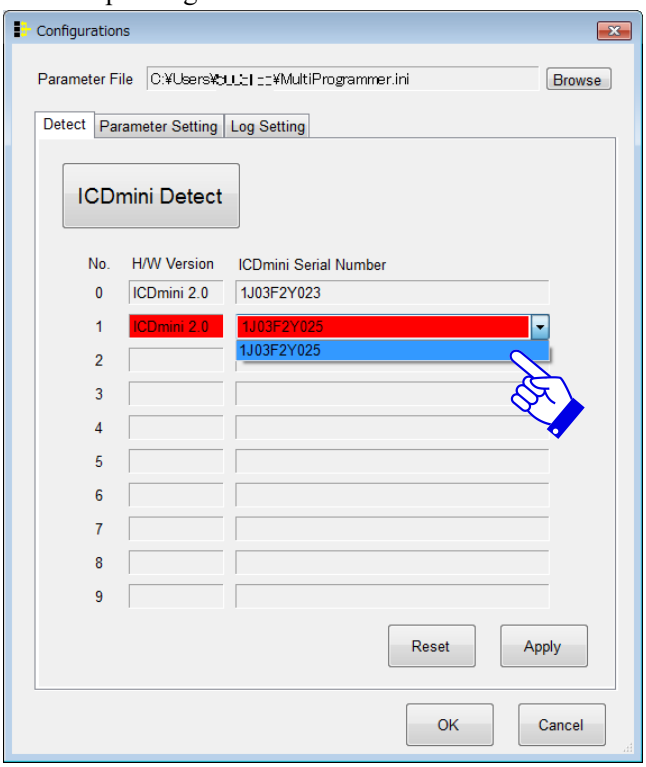


Figure 4.3.8 Detect tab setup window (6)

Step ③
Click the [Apply] button to update the selected parameter file to reflect the redetection results.

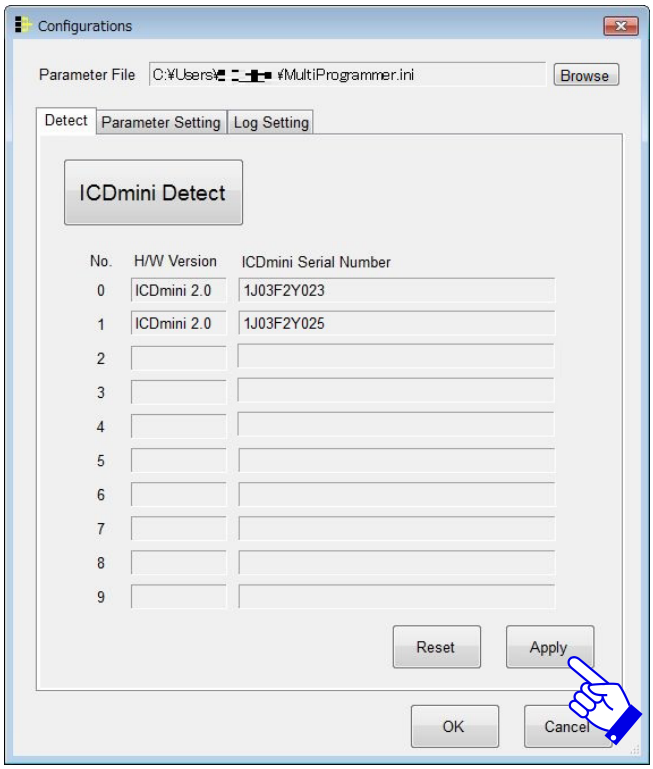


Figure 4.3.9 Detect tab setup window (7)

4.3.2 Resetting the detection results

Resetting the ICDmini detection results reflected in the parameter file will reset the detection results, regardless of the ICDminis connected. Click the [Reset] button to reset the detection results.

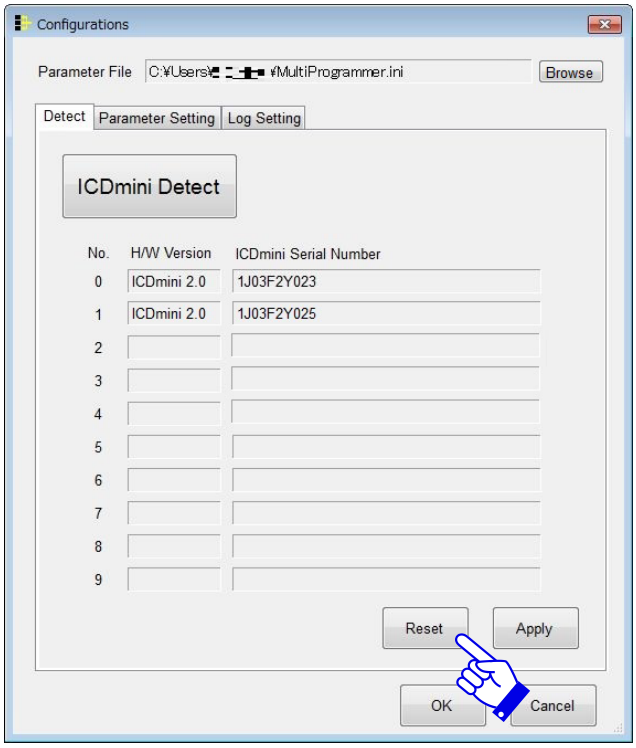


Figure 4.3.10 Detect tab setup window (8)

Immediately after resetting, no ICDminis will appear as detected. To redetect the ICDminis currently connected to the PC, click the [ICDmini Detect] button.

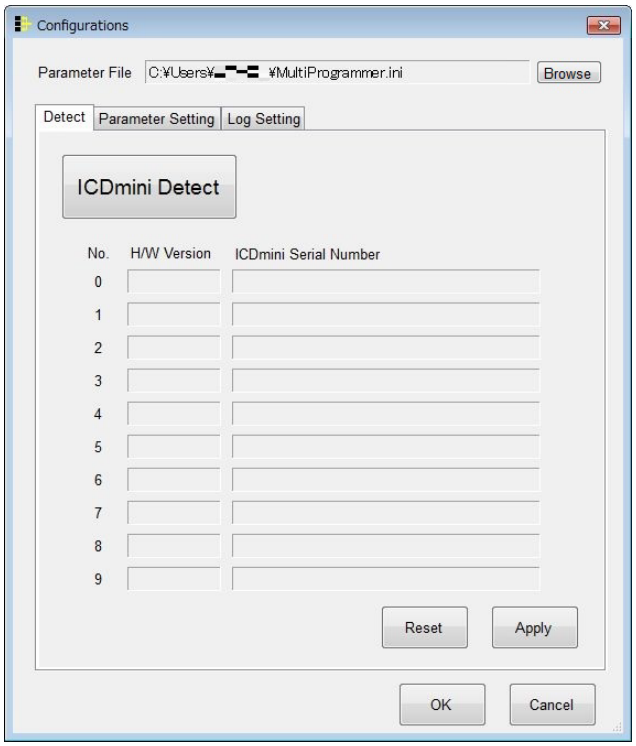


Figure 4.3.11 Detect tab setup window (9)

4.4 Setting parameters

Selecting a file for Parameter File enables the Parameter Setting tab. This tab lets you set the various parameters required for multi-programming.

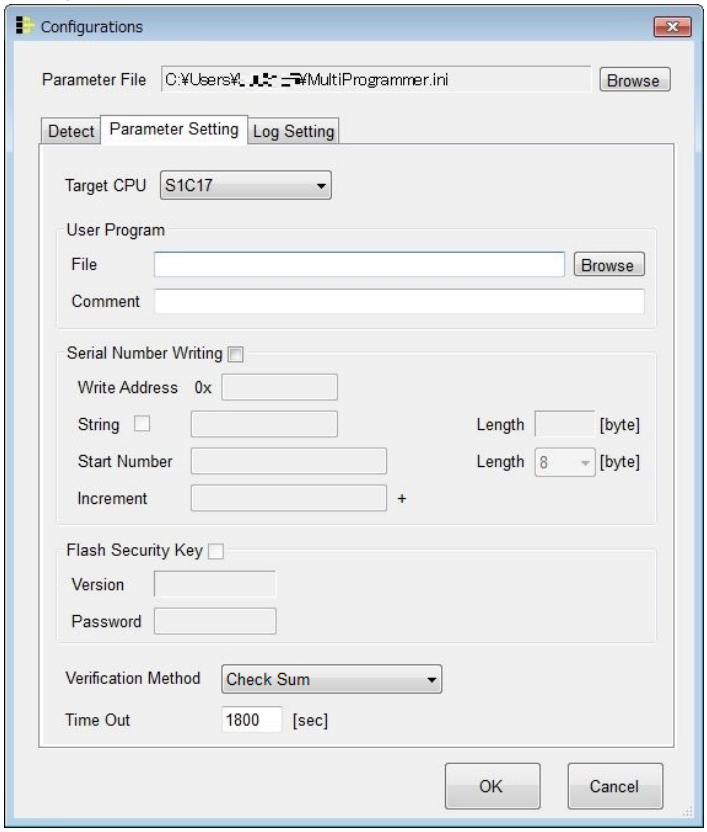


Figure 4.4.1 Parameter Setting tab setup window

(1) Target CPU

Select the target MCU model.

(2) User Program

Set the user program file.

File	Use the [Browse] button to select the user program file (.psa).
Comment	Enter comments as required concerning the user program file.

(3) Serial Number Writing

Set the serial numbers as required.

Serial Number Writing□	Use the checkbox to select/unselect use of the Serial Number Writing area. Check off: Not set Check on: Enter the following items for Serial Number Writing:
Write Address	Enter the destination address for writing the serial number (hexadecimal).
String□	Use the checkbox to select whether to use a text string. Check off: Not set Check on: Enter a text string of up to 8 alphanumeric characters. “Length” indicates the number of bytes used for the text string entered.
Start Number	Enter an initial serial number value of up to 64 bits (decimal).
Length	Select the length of the serial number to be handled. Enter a value of up to 8 bytes.
Increment	The value to be added to the serial number (decimal)

(4) Flash Security Key

Set the unlock password if a flash security password has been set for the target MCU.

Flash Security Key□	Use the checkbox to select whether the Flash Security Key area is used. Check off: Not set Check on: Enter the following items for Flash Security Key:
Version	Display the flash security version.
Password	Set the unlock password. Enter a value of up to 12 characters of 0 - 9, a - z, and A - Z.

(5) Verification Method

Select the verification method.

All Data Comparison	Compare all data.
Check Sum	Compare using the checksum for faster comparisons.

(6) Time Out

Set the execution time timeout value. You can set a value of up to 7,200 seconds.

4.5 Setting the log file

Selecting a file for Parameter File enables the Log Setting tab. This tab let you set the file (.csv) for saving the multi-programming execution log. The date of execution will be appended to the name of the file created.

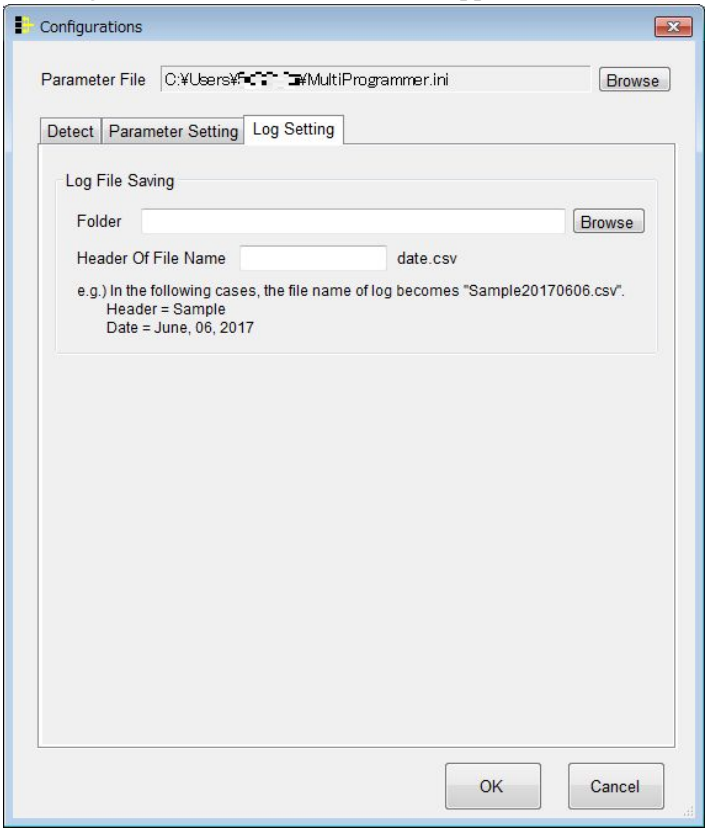


Figure 4.5.1 Log Setting tab setup window

(1) Folder
Enter the name of the folder in which the log file is to be saved. Use an absolute path. Click the [Browse] button to display the folder selection window.

(2) Header of File Name
Enter the text for the initial part (header) of the log file name.

Log file name format
[header]date.csv
header: Text entered here
date: Execution date

For example, if the header is "Sample" and the date is June 6, 2017, the file name will be Sample20170606.csv.

4.6 Error message list

The error messages generated when setting each tab are listed below.

Table 4.6.1 Common error messages

Error message	Meaning
The setting was changed. Is the setting applied?	Another tab was selected after changing the settings. To update the settings, click the [Yes] button; to cancel the settings, click the [No] button.
Initialization of multi programmer DLL failed.	The model-specific information file for this model does not contain cfg17xxx.dll. Go to the Seiko Epson website to obtain the most recent model-specific information file.
The parameter file has been deleted.	The specified parameter file was deleted. Reselect the parameter file.

Table 4.6.2 "Detect" tab error messages

Error message	Meaning
Detected “ICDmini Serial Number” is 0.	No ICDmini serial numbers are detected. Check the ICDmini connections and write in the ICDmini serial numbers.
Not detected "ICDmini Serial Number" is *n.*	No serial numbers are detected for n ICDminis. Write in the serial numbers for the ICDminis that cannot be detected.
Detect is invalid. There are the same as "ICDmini Serial Number." item two or more.	Two or more ICDminis with the same serial number exist. Refer to “3.2 Writing ICDmini serial numbers” and rewrite the duplicated ICDmini serial numbers.
Detect is invalid. Detected "ICDmini Serial Number" is over 10.	Serial numbers are found for 11 or more ICDminis. No more than 10 ICDminis can be connected. Do not connect more than 10.

Table 4.6.3 "Parameter Setting" tab error messages

Error message	Meaning
Target CPU is not set.	No target MCU model is selected. Select the target MCU model.
There are no MCU model files.	There is no model-specific information file. Refer to "Installing Multi Programmer" and save the model-specific information file to the specified location.
File not found.	The path for the user program file is incorrect. Enter the correct path for the user program file.
Please input user program file path.	The path for the user program file has not been entered. Enter the user program file path.
Following characters of user program file are not allowed to used. /;,*?<>|"	The user program file name contains one or more of the following invalid characters: / ; , * ? < > | " Make sure the user program file name includes none of these characters.
User program file is format error.	The user program file format is incorrect. Confirm that the format is Motorola S.
There is a part overlapped with the address in the user program file.	Part of the user program data is overlapping. Correctly configure the user program data.

Please input all "Serial Number Writing" items.	Serial Number Writing is checked as "On", but not all details for "Write Address", "Start Number", and "Increment" are entered. Enter "Write Address", "Start Number", and "Increment".
Write address includes invalid character.	Write Address includes a character other than a hexadecimal value. Enter the Write Address using only hexadecimal values.
Start number includes invalid character.	Start number includes a character other than a decimal value. Enter the Start number using only decimal values.
Increment includes invalid character.	Increment includes a character other than a decimal value. Enter the Increment using only decimal values.
Over useful range(6-12). Flash security password.	The length of the flash security password is outside the valid range. Enter a flash security password of 6 to 12 characters.
User security password includes invalid character.	The flash security password contains characters other than alphanumeric characters. Enter the correct password.
Over useful range(0-7200).	The permitted timeout range is exceeded. Enter a timeout value in the range 0 to 7200.
Time out includes invalid character.	The timeout entered contains a non-numerical character. Enter a timeout value in the range of numbers from 0 to 7200.
Please input time out.	No timeout value is entered. Enter a timeout value in the range 0 to 7200.
Parameter file is format error.	The parameter file format is incorrect. Create a parameter file in the correct format.
The user program of the parameter file is broken.	The user program is corrupted. Create a new parameter file.

Table 4.6.4 "Log Setting" tab error messages

Error message	Meaning
Write folder name in full from drive name.	Only the drive letter is entered. Specify the correct folder name using an absolute path.
Write folder name correctly.	The drive separator ":" is omitted. Specify the correct folder name using an absolute path.
Can not use sequential two ¥.	Two directory separators "¥" are used in succession. Specify the correct folder name using an absolute path.
Following characters are not allowed to used. /:;,*?<>|"	The folder or file name contains one or more of the following invalid characters: / :; , * ? < > | " Make sure the folder or file name contains none of these characters.
Can not create the folder.	Folder creation failed. Confirm that you are authorized to create folders at the destination.

5. Multi-programming

The main window appears as shown below when Multi Programmer is launched.

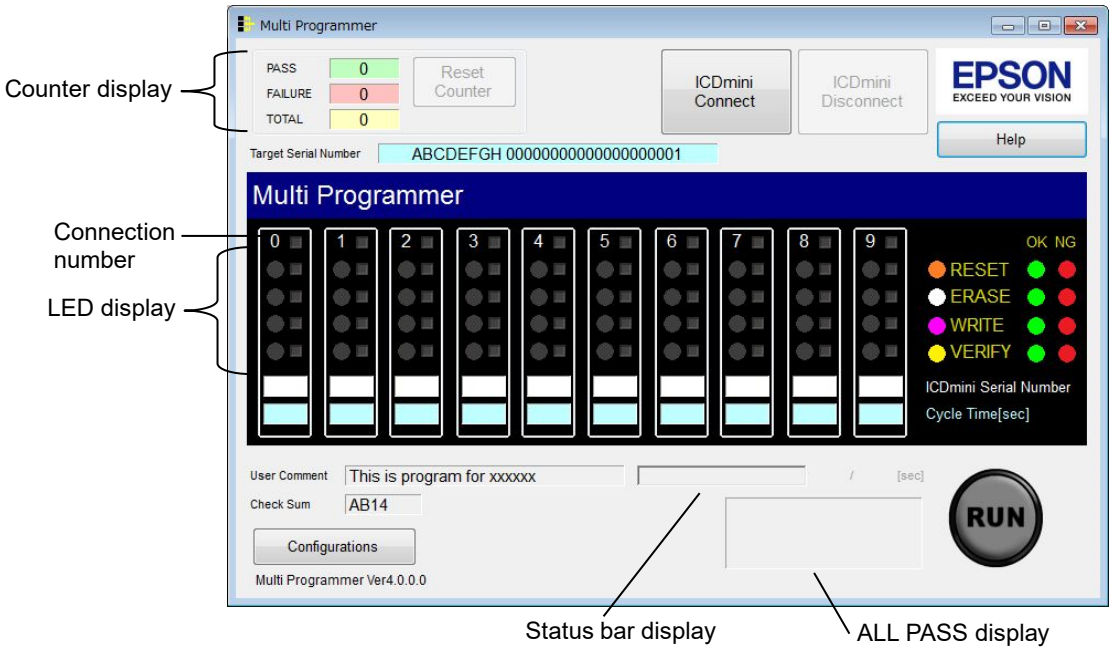


Figure 5.1 Multi Programmer main window

Details of the various buttons are given below.

Table 5.1 Individual button details

Display	Meaning
Configurations	Launches the Multi Programmer setup window.
ICDmini Connect	Connects an ICDmini to Multi Programmer.
ICDmini Disconnect	Disconnects an ICDmini from Multi Programmer.
RUN	Starts target system control.
Reset Counter	Clears the counter display, resetting the counter to zero.
Help	Displays the Help window.

The display details are described below.

Table 5.2 Individual display details

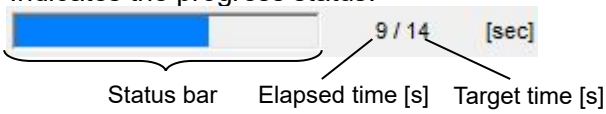
Display	Meaning
Counter display	
PASS	The number of successful target systems
FAILURE	The number of failed target systems
TOTAL	The total of number of "PASS" and "FAILURE" judgments
LED display	
RESET	Indicates the target flash reset status or results.
ERASE	Indicates the target flash erase status or results.

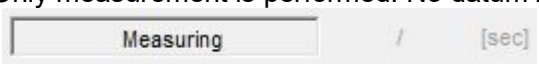
RESET

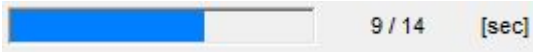
LED status	Meaning
Off	Not active
Orange	Standby
Flashing orange	Running
Green	Successful
Red	Failed

ERASE

LED status	Meaning
Off	Not active
White	Standby

<table>
<tr><td></td><td>
<table>
<tr><td>Flashing white</td><td>Running</td></tr>
<tr><td>Green</td><td>Successful</td></tr>
<tr><td>Red</td><td>Failed</td></tr>
</table>
</td></tr>
<tr><td>WRITE</td><td>Indicates the target flash write status or results.
<table>
<tr><th colspan="2">LED status</th><th>Meaning</th></tr>
<tr><td>Off</td><td>Not active</td></tr>
<tr><td>Purple</td><td>Standby</td></tr>
<tr><td>Flashing purple</td><td>Running</td></tr>
<tr><td>Green</td><td>Successful</td></tr>
<tr><td>Red</td><td>Failed</td></tr>
</table>
</td></tr>
<tr><td>VERIFY</td><td>Indicates the target flash verification status or results.
<table>
<tr><th colspan="2">LED の状態</th><th>Meaning</th></tr>
<tr><td>Off</td><td>Not active</td></tr>
<tr><td>Yellow</td><td>Standby</td></tr>
<tr><td>Flashing yellow</td><td>Running</td></tr>
<tr><td>Green</td><td>Successful</td></tr>
<tr><td>Red</td><td>Failed</td></tr>
</table>
</td></tr>
<tr><td>Status bar display</td><td>Indicates the progress status.
 

<table>
<tr><th>Display</th><th>Meaning</th></tr>
<tr><td>Status bar</td><td>Indicates the progress status.</td></tr>
<tr><td>Elapsed time</td><td>Indicates the elapsed time from execution start (RUN) until finish.</td></tr>
<tr><td>Target time</td><td>Indicates the mean time for the previous execution from start (RUN) until finish. This is recalculated after each execution.</td></tr>
</table>
[Display for first execution]
Only measurement is performed. No datum mean time exists.


[Display for second and subsequent executions]
Indicates the progress based on the previous mean time.


</td></tr>
<tr><td>ALL PASS display</td><td>Displays results for all units.
<table>
<tr><th>Status</th><th>Meaning</th></tr>
<tr><td>ALL PASS</td><td>Control for all of the target systems connected was successful for one execution.</td></tr>
<tr><td>FAILURE</td><td>Control for one or more target systems connected failed for one execution.</td></tr>
</table>
</td></tr>
<tr><td>Connection No.</td><td>The number linking the display in the main window to the ICDmini detected in "ICDmini Detect".</td></tr>
<tr><td>Target Serial Number</td><td>Indicates the first serial number to be written to the next target flash (decimal). The serial number is automatically incremented from the lowest connection number. The value will be a masked number if the maximum serial number length specified in "Configurations" is exceeded.</td></tr>
<tr><td>User Commnet</td><td>A user-defined comment about the user program file entered in the parameter file</td></tr>
<tr><td>Check Sum</td><td>This is the checksum for the user program specified in the parameter file. The checksum is the value incremented for each 2 bytes (16 bits) of data, disregarding values exceeding 16 bits in length.</td></tr>
<tr><td>ICDmini Serial Number</td><td>The last four digits of the ICDmini serial number</td></tr>
<tr><td>Cycle Time</td><td>The time taken from execution start (RUN) until finish</td></tr>
</table>

5.1 Connecting ICDminis to PC

Once all devices are set up and prepared, connect the ICDminis to the PC via a USB hub, as shown below.

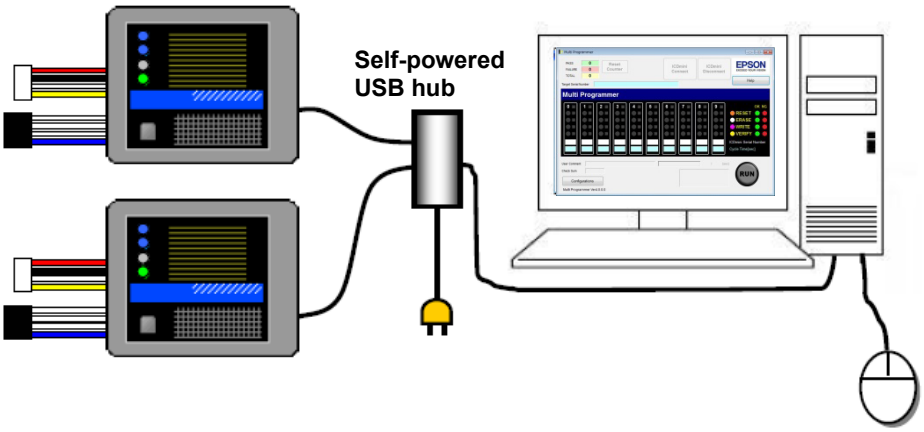


Figure 5.1.1 PC - ICDmini connection diagram

5.2 Establishing connections to ICDminis

The ICDminis must be detected before performing the procedures described here. Click the [ICDmini Connect] button to establish the connection to the ICDminis. Connections are possible only for ICDminis already detected.

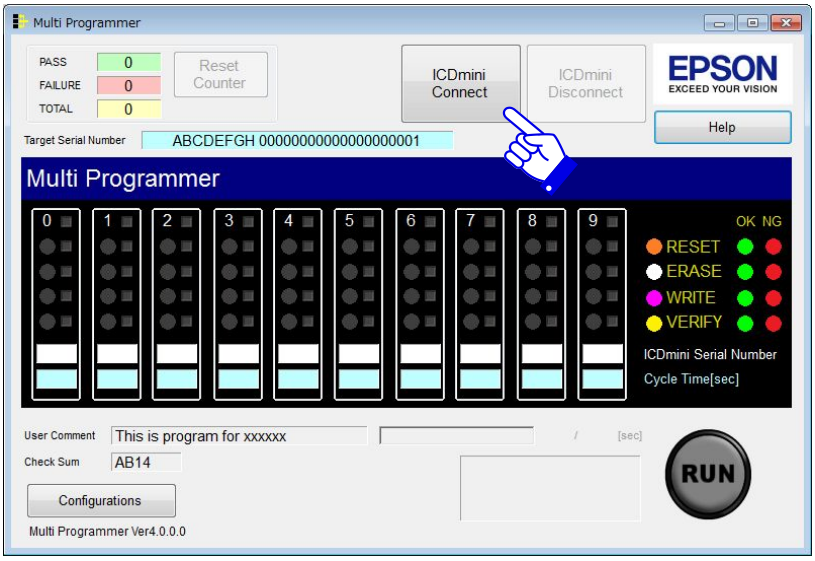


Figure 5.2.1 Establishing connections to ICDminis (1)

The individual numbered LEDs will light up once the corresponding connection is established.

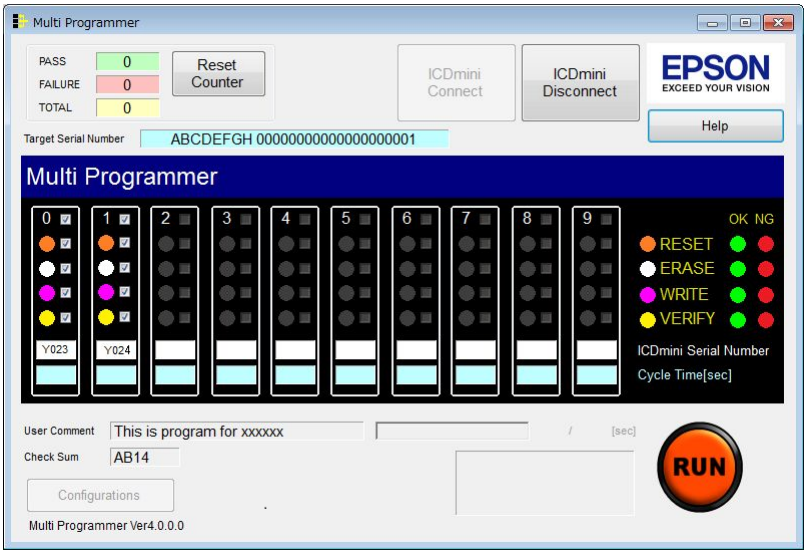


Figure 5.2.2 Establishing connections to ICDminis (2)

5.3 Connecting target systems

Use the target system connecting cables to connect the target systems to the corresponding ICDminis. The connector pins are as shown below. For more information on connecting to the ICDminis, refer to the S5U1C17001H** (ICDmini Ver *.*) User Manual.

Table 5.3.1 Target system connecting pins

Pin name	I/O	Pin function
DCLK	I	Clock signal input for debugging
GND	–	Power supply (GND)
DSIO	I/O	Serial communication signal input/output for debugging
DST2	I	Debug status signal input
FLASH VCC OUT	O	Flash memory programming voltage output (ICDmini Ver.2.0/3.x only)
GND	–	Power supply (GND) Connect to GND on the target system.
TARGET RST OUT	O	Target reset signal output Always connect this to the reset pin on the target system.
TARGET VCC IN	I	Target power supply voltage input Use this pin when supplying the interface power voltage between the ICDmini and target system from the target system. (1.0 to 5.5 V) If this pin is not used, the interface power is fixed at 3.3 V or 1.8 V. Use the DIP switches on the ICDmini to select these voltages. (Refer to “3.3 Setting up the ICDmini.”)

Note that the customer is responsible for configuring power supplies for the target systems. Connect the power supplies to the target systems after connecting the target systems to the ICDminis.

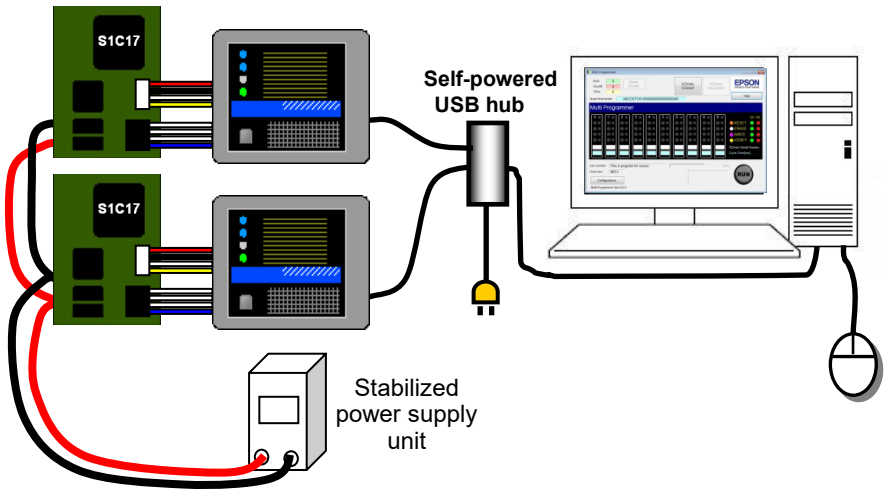


Figure 5.3.1 Overall multi-programming configuration diagram

If you are using ICDmini Ver. 1.0/1.1/2.0, press the reset button on the ICDmini once the systems are connected.

5.4 Selecting the processing to be executed

Select the processing you want to execute by checking/unchecking the corresponding checkboxes. From top to bottom, the checkboxes correspond to RESET, ERASE, WRITE, and VERIFY. The initial default setting is to execute all of the following: RESET, ERASE, WRITE, and VERIFY.

Table 5.4.1 Processing details

Processing	Processing details
RESET	Resets the target system.
ERASE	Erases the target flash.
WRITE	Writes to the target flash.
VERIFY	Verifies target flash details and user program.

In the example shown below, only RESET and VERIFY are set to be executed for connection number 0.

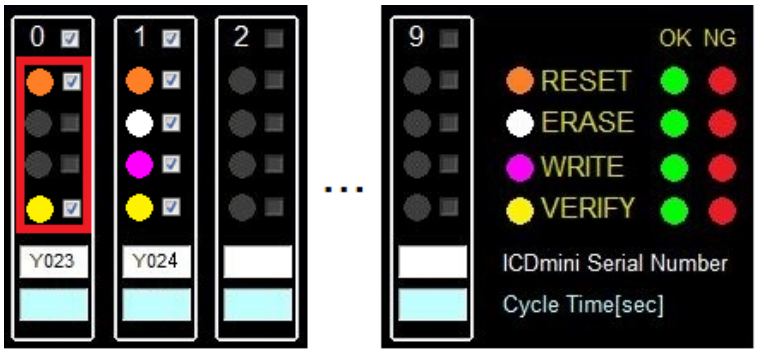


Figure 5.4.1 Individual execution function selection

The individual checkbox selection settings are saved to the parameter file.

5.5 Running multi-programming

Click the [RUN] button to execute multi-programming. (You can also use the Enter key on the keyboard.) Press any of the numerical keys on the keyboard to write to a single ICDmini with the connection number corresponding to that key.

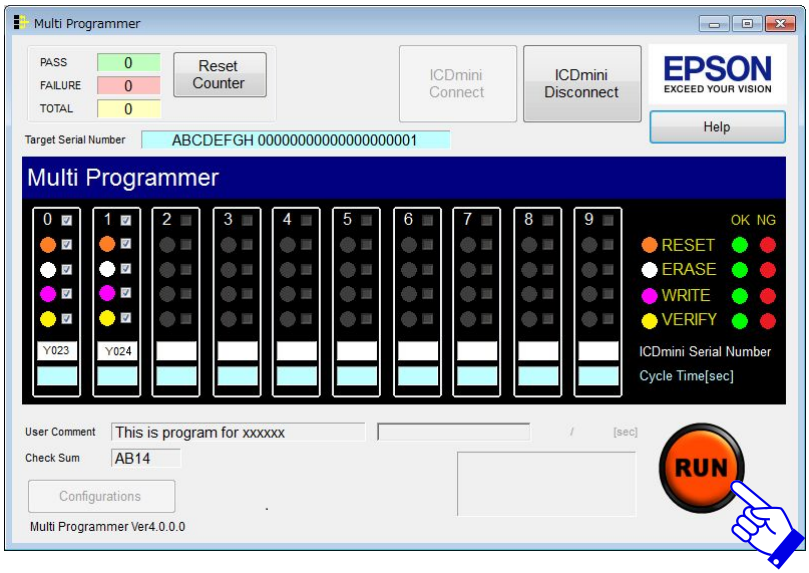


Figure 5.5.1 Running multi-programming

Processing is performed in the following sequence: RESET, ERASE, WRITE, and VERIFY. The corresponding LED flashes while the process in question is underway and turns to steady green once the process is successfully completed. "ALL PASS" will appear at the bottom right once all processes are completed.

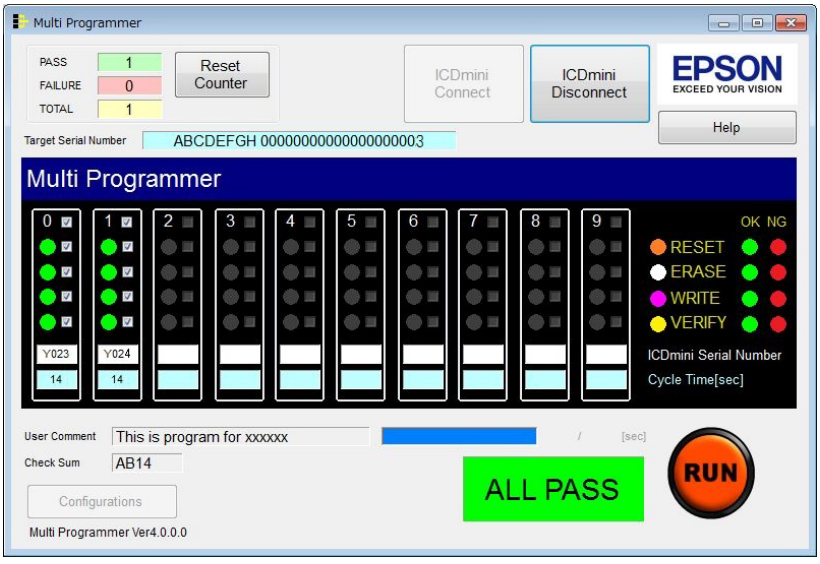


Figure 5.5.2 Display when all processing is successful

If a problem occurs during processing, the LED turns to steady red for that process, and processing is terminated for that target system. If an error occurs in one or more target systems, "FAILURE" will appear as the execution result.

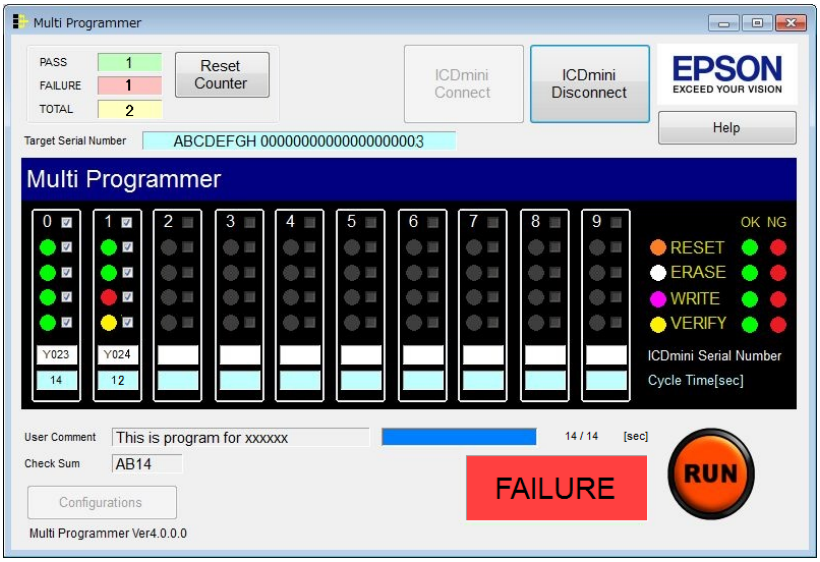


Figure 5.5.3 Display when an error occurs

5.6 Disconnecting target systems

Turn off the target system power supply, disconnect the individual target systems, and replace with the next target systems to be controlled. After replacing the target systems, repeat the procedures from "5.3 Connecting target systems."

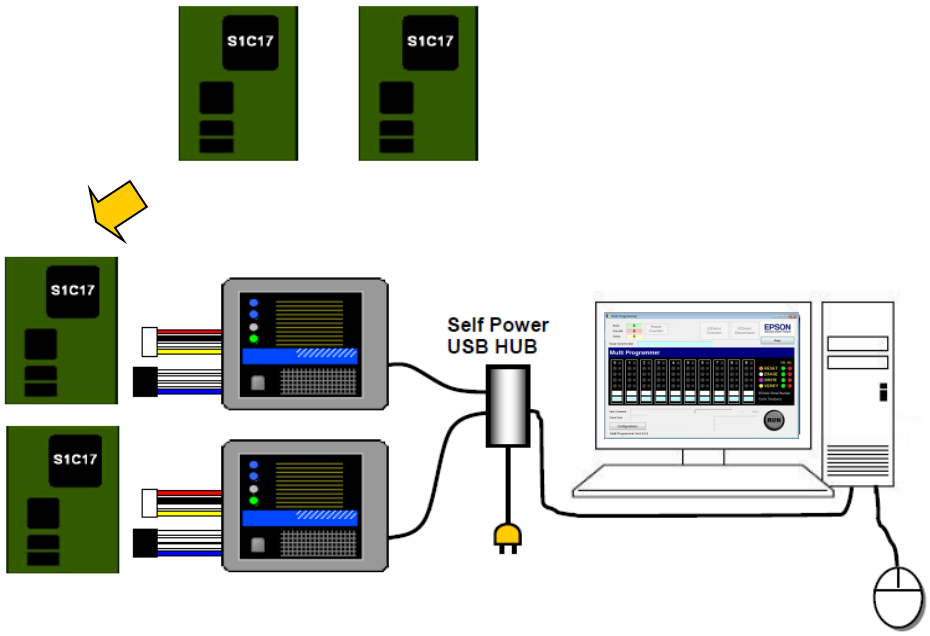


Figure 5.6.1 Target system exchange diagram

5.7 Error message list

The error messages generated when using the main screen are listed below.

Table 5.7.1 Error messages

Error message	Meaning
Not found parameter file. Please set parameter.	No parameter file exists. Reselect a parameter file.
Please select (check) any operation.	The processing selection checkboxes are set to "Off" for all connection numbers. Select the processing and execute (RUN).
Target is disconnected (SN:xxxxxx).	No target systems are connected. Connect a target system.
Can not open the ICDmini (SN:xxxxxx). All process are canceled.	No connection can be established with the target ICDmini. Connect the target ICDmini.
Time out occurred the ICDmini (SN:xxxxxx)	Processing was incomplete when the duration specified for "Time Out" expired. Extend the time setting.
Target serial number exceeded the maximum value.	The serial number specified exceeds the maximum length. Check the serial number to be written to the target flash next time.

6. Troubleshooting

6.1 ICDmini detection

- **The message "Detected "ICDmini Serial Number" is 0." is displayed. No ICDminis are detected.**

  (1) The ICDminis may be disconnected from the PC.

  Reconnect the ICDminis to the PC. (Refer to "5.3 Connecting target systems.")

  (2) The serial numbers have not been written for the ICDminis being used.

  Write the serial number for each ICDmini. (Refer to "3.2 Writing ICDmini serial numbers.")

  (3) The USB driver has not been installed.

  Install the USB driver. (Refer to "3.1 Installing Multi Programmer.")

6.2 Multi-programming

- **Write error (FAILURE) occurs.**

  (1) Check to confirm that the debugging pins (DCLK, DSIO, DST2) on the target MCU are all connected correctly to the debugging pins on the ICDmini.

  (For more information on the ICDmini debugging pins, refer to the S5U1C17001H** (ICDmini Ver*.*) User Manual.)

  (2) Check to confirm that the connection signal wire between the target MCU and ICDmini is the minimum length (not more than approximately 15 cm for ICDmini Ver. 1.0/1.1/2.0 or approximately 30 cm for ICDmini Ver. 3.0).

  Check to confirm that the target system circuit board wiring incorporates noise prevention measures.

  (3) Check to confirm that the target system power supply voltage matches the ICDmini interface power supply voltage.

  With ICDmini Ver. 1.0/1.1/2.0, the interface voltage with the target system can be switched between 3.3 V and 1.8 V for the external input using the ICDmini DIP switches. (For more information on the ICDmini Ver. 1.0/1.1/2.0 DIP switches, refer to the S5U1C17001H** (ICDmini Ver*.*) User Manual.)

  (4) When using target MCUs that require an external flash programming power supply (VPP), check to confirm that the flash programming power supply is being supplied appropriately from the ICDmini.

  (Refer to "5.3 Connecting target systems.")

  (5) Check to confirm that all Multi Programmer parameters are set correctly. Specifically, check for the following problems:

  - The wrong target system model is selected.
  - The user program size exceeds the limits.
  - The address for writing the serial number is incorrect.
  - The address for writing the serial number is the same as the location for installing the user program.
  - The flash security password is incorrect.

- A password has been set for a target MCU for which no flash security password is set.
- No password has been set for a target MCU for which a flash security password is set.

(Refer to "4.4 Setting parameters.")

(6) With ICDmini Ver. 1.0/1.1/2.0, check to confirm that the ICDmini reset button has not been pressed after running [ICDmini Connect].

If the reset button has been pressed, press [ICDmini Disconnect], then repeat [ICDmini Connect].

Appendix A Multi Programmer Dynamic Link Library Manual

A.1 Overview

This appendix describes how to use the MultiProgrammer.dll functions used when interfacing with the ICDmini. Refer to this explanation when using the Dynamic Link Library directly.

When you use the MultiProgrammer.dll functions, the following files are also used, in addition to this DLL:

- icdmini2.dll
- icdmini3.dll
- Model-specific information file
- USB driver

A.1.1 Function Call-up Example

Example:

```
InitializeTargetInfo        //Reads and initializes the model-specific information file.
GetConnectedICD             //Fetches information for the ICDmini connected to the PC.
OpenIcdConnection           //Establishes a connections to the ICDmini.
                              (Execute for the number of units used.)
                                                          * Repeated Part
(Exchanging target systems)

ResetTarget                 //Resets the target system.
GetStatus                   //Monitors completion of the target system reset.
CheckTargetConnection       //Checks connections to the target system.
GetStatus                   //Monitors completion of the target system connection check.
StartOperation              //Executes the specified processing.
GetStatus                   //Monitors completion of all specified processing.

CloseIcdConnection          // Disconnects the ICDmini.
                              (Execute for the number of units used.)
ReleaseTargetInfo           //Releases the model-specific information file.
```

A.2 Function Details

A.2.1 InitializeTargetInfo

Function	Target MCU information and user program initialization Model-specific information file reading and initialization		
Format	int InitializeTargetInfo(const TargetInfo * pTargetInfo, const UserInfo * pUserInfo, unsigned long *userProgramCheckSum);		
Arguments	IN	pTargetInfo	Target MCU information
	IN	pUserInfo	User program information
	OUT	userProgramCheckSum	User program checksum Calculates and stores the user program checksum.
Return value	OK		
	NG		
Remarks	■ TargetInfo definition char mcuName[100] — Model name char mcuPath[2000] — Model-specific information file path char mcuOption[100] — MCU model Detail option text string Set to "ALL 0x00" if not specified. (For more information, refer to the readme file included in the model-specific information file package.) int mcuSecurityRelease — Flash security password unlocking (0: No, 1: Yes) char mcuSecurityVersion[10] — Flash security version (fixed at M03) char mcuSecurityPassword[30] — Flash security password ■ UserInfo definition int userProgramVerify — Verification method (0: Compare all data, 1: Compare checksum) int userParamCount — User program segments (The number of segments if the user program is divided into multiple address areas. Maximum 1,024) struct UserProgramParam *userParam — User program information start pointer (Corresponding information if the user program is divided into multiple address areas) ■ UserProgramParam definition unsigned long userProgramAddr — User program address unsigned long userProgramSize — User program size (Units: bytes; this must be an even value) unsigned char *userProgramPointer — User program start pointer		

A.2.2 ReleaseTargetInfo

Function	Model-specific information file release
Format	int ReleaseTargetInfo(void);
Arguments	None
Return value	OK
	NG
Remarks	None

A.2.1 OpenIcdConnection

Function	Connects an ICDmini to the corresponding specified ICD handle.		
Format	int OpenIcdConnection(long icdHandle);		
Arguments	IN	icdHandle	ICD handle
Return value	OK		
	ERROR_PARAMETER		
	ERROR_ICD_OPEN_CONNECTION		
Remarks	・This function checks to confirm that GetConnectedICD() has been executed. ・This function does not return control until success or failure is confirmed.		

A.2.2 CloseIcdConnection

Function	Disconnects an ICDmini from the corresponding specified ICD handle.		
Format	int CloseIcdConnection(long icdHandle);		
Arguments	IN	icdHandle	ICD handle
Return value	OK		
	ERROR_PARAMETER		
	ERROR_ICD_CLOSE_CONNECTION		
Remarks	・This function checks to confirm that GetConnectedICD() has been executed. ・This function does not return control until success or failure is confirmed.		

A.2.3 ResetTarget

Function	Resets the target for the target system connected to the ICDmini corresponding to the ICD handle specified.		
Format	int ResetTarget(long icdHandle);		
Arguments	IN	icdHandle	ICD handle
Return value	OK		
	NG		
	ERROR_PARAMETER		
	ERROR_ICD_CONNECTION		
Remarks	・This function checks to confirm that GetConnectedICD() has been executed. ・This function returns control immediately after calling. ・This function requires monitoring of processing completion with GetStatus().		

A.2.4 CheckTargetConnection

Function	Checks the connections to the target system connected to the ICDmini corresponding to the ICD handle specified.		
Format	int CheckTargetConnection(long icdHandle);		
Arguments	IN	icdHandle	ICD handle
Return value	OK		
	NG		
	ERROR_PARAMETER		
	ERROR_ICD_CONNECTION		
Remarks	・This function checks to confirm that GetConnectedICD() has been executed. ・This function returns control immediately after calling. ・This function requires monitoring of processing completion with GetStatus()。		

A.2.5 StartOperation

Function	Executes the specified processing using the ICDmini corresponding to the ICD handle specified.		
Format	int StartOperation(long icdHandle, long icdOperation, long timeOut, unsinged long serialWriteAddress, int serialNumberSize, unsigned char *serialNumber);		
Arguments	IN	icdHandle	ICD handle
	IN	icdOperation	Processing performed bit0: Reset target system (1: Yes, 0: No) bit1: Erase target flash (1: Yes, 0: No) bit2: Write target flash (1: Yes, 0: No) bit3: Verify target flash (1: Yes, 0: No) bit4: Write serial number (1: Yes, 0: No)
	IN	timeOut	Execution timeout value (1 = 0.1 s) This can be specified in the range from 0 to 72000 s. If 0 is specified, the timeout is not detected.
	IN	serialWriteAddress	Address for writing serial number (0x0-0xfffffc)
	IN	serialNumberSize	Serial number size (Units: Bytes) If 0 is specified, no serial number is written.
	IN	serialNumber	Serial number
Return value	OK		
	NG		
	ERROR_PARAMETER		
	ERROR_ICD_CONNECTION		
Remarks	・This function checks to confirm that GetConnectedICD() has been executed. ・This function returns control immediately after calling. ・This function requires monitoring of processing completion with GetStatus(). When multiple processing is executed, GetStatus() returns a response once all processing is complete.		

A.2.6 GetStatus

Function	Fetches the processing status for the ICDmini corresponding to the ICD handle specified.		
Format	int GetStatus(long icdHandle, int *serialNumberSize, unsigned char *serialNumber);		
Arguments	IN	icdHandle	ICD handle
	OUT	serialNumberSize	Serial number size (0: Serial number matching) Verifies the target flash with a serial number. Stores the serial number size read from the target system only if the serial number does not match.
	OUT	serialNumber	Serial number read in Verifies the target flash with a serial number. Stores the serial number read from the target system only if the serial number does not match.
Return value	OK		
	OPERATION_TARGET_CONNECTION		
	OPERATION_TARGET_RESET		
	OPERATION_ERASE		
	OPERATION_WRITE		
	OPERATION_VERIFY		
	OPERATION_WRITE_SERIALNO		
	ERROR_PARAMETER		
	ERROR_TIMEOUT_TARGET_CONNECTION		
	ERROR_TIMEOUT_TARGET_RESET		

	ERROR_TIMEOUT_ERASE
	ERROR_TIMEOUT_WRITE
	ERROR_TIMEOUT_VERIFY
	ERROR_TIMEOUT_WRITE_SERIALNO
	ERROR_ICD_CONNECTION
	ERROR_TARGET_CONNECTION
	ERROR_TARGET_RESET
	ERROR_ERASE
	ERROR_WRITE
	ERROR_VERIFY
	ERROR WRITE_SERIALNO
	ERROR VERIFY_SERIALNO
Remarks	This function checks to confirm that GetConnectedICD() has been executed.

A.2.7 GetString

Function	Returns a text string in response to a return code.		
Format	int GetString(int retuenCode, char * retuenedString);		
Arguments	IN	retuenCode	Return code
	OUT	retuenedString	Text string corresponding to return code Stores a text string corresponding to the return code. The call source must have free space of at least 256 bytes. "Invalid returned code" is returned for an invalid return code.
Return value	OK		
	NG		
Remarks	None		

A.2.8 GetConnectedICD

Function	Fetches information for the ICDmini connected to the PC.		
Format	int GetConnectedICD (long maxCount, long *connecttedCount, struct icdInfo *plcdInfo);		
Arguments	IN	maxCount	Maximum number of ICDminis connected (Up to 10)
	OUT	connecttedCount	Number of ICDminis Stores information on the ICDminis connected.
	OUT	plcdInfo	ICDmini information Stores information on the ICDminis connected.
Return value	OK		
	NG		
Remarks	■ icdInfo definition long icdHandle — ICD handle init icdDllVersion — Number of icdminix.dll used(icdmini2.dll=2, icdmini3.dll=3) char icdVersion[8] — ICD version char icdSerialNumbert — Serial number [50]		

A.2.9 Return Codes

Status	Return code	Corresponding text string
OK	0x00	Normally ended.
NG	0x01	Error occurred.
ERROR_TIMEOUT_TARGET_CONNECTION	0x12	Time out occurred while connecting with the target system.
ERROR_TIMEOUT_TARGET_RESET	0x13	Time out occurred while executing target-reset.
ERROR_TIMEOUT_ERASE	0x14	Time out occurred while erasing the FLASH memory.
ERROR_TIMEOUT_WRITE	0x15	Time out occurred while writing to the FLASH memory.
ERROR_TIMEOUT_VERIFY	0x16	Time out occurred while verifying the FLASH memory.
ERROR_ICD_OPEN_CONNECTION	0x21	Can not connect with the ICDmini.
ERROR_ICD_CONNECTION	0x22	Already disconnected with the ICDmini.
ERROR_ICD_CLOSE_CONNECTION	0x29	Can not disconnect with the ICDmini.
ERROR_TARGET_CONNECTION	0x32	Disconnected with the target system.
ERROR_TARGET_RESET	0x33	No response from the target for target-reset.
ERROR_ERASE	0x44	Error occurred while erasing the FLASH memory.
ERROR_WRITE	0x45	Error occurred while writing to the FLASH memory.
ERROR_VERIFY	0x46	Error occurred while verifying the FLASH memory.
ERROR_PARAMETER	0x50	Parameter is invalid.
OPERATION_TARGET_CONNECTION	0x82	Connecting with the target system.
OPERATION_TARGET_RESET	0x83	Executing target-reset.
OPERATION_ERASE	0x84	Erasing the FLASH memory.
OPERATION_WRITE	0x85	Writing to the FLASH memory.
OPERATION_VERIFY	0x86	Verifying the FLASH memory.
ERROR_TIMEOUT_WRITE_SERIALNO	0x90	Time out occurred while writing serial number to the FLASH memory.
ERROR_WRITE_SERIALNO	0x91	Error occurred while writing serial number to the FLASH memory.
OPERATION_WRITE_SERIALNO	0x92	Writing serial number to the FLASH memory.
ERROR_VERIFY_SERIALNO	0x93	Error occurred while verifying serial number to the FLASH memory.

Revision History

Attachment-1

Rev. No.	Date	Page	Category	Contents
Rev.2.0	2017/06/08	All	New	New
Rev.2.01	2017/09/13	Cover, Back cover	Revision	Title and document code were changed
Rev.2.01	2017/09/13	37	Revision	Item is added to the definition of icdInfo
Rev.2.02	2021/02/26	8	Revision	Changed the number of characters in the serial number of ICDmini to a maximum 10 characters.
Rev.2.03	2024/03/08	Cover, 2 Back cover	Revision	Company logo, Notice, Addresses of Epson America, Inc. and Epson (China) Co., Ltd., Documentation Code were updated.

EPSON

International Sales Operations

America

Epson America, Inc.

Headquarter:
3131 Katella Ave.
Los Alamitos, CA 90720, USA
Phone: +1-800-463-7766

San Jose Office:
2860 Zanker Road Suite 204
San Jose, CA 95134, USA
Phone: +1-800-463-7766

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.
4th Floor, Tower 1 of China Central Place, 81 Jianguo Road, Chaoyang District, Beijing 100025 China
Phone: +86-10-8522-1199 FAX: +86-10-8522-1120

Shanghai Branch
Room 601-603, Building A One East, No.325 East Longhua Road, Shanghai 200023, China
Phone: +86-21-5330-4888 FAX: +86-21-5423-4677

Shenzhen Branch
Room 804-805, 8 Floor, Tower 2, Ali Center,No.3331 Keyuan South RD(Shenzhen bay), Nanshan District, Shenzhen 518054, China
Phone: +86-10-3299-0588 FAX: +86-10-3299-0560

Epson Taiwan Technology & Trading Ltd.
15F, No.100, Songren Rd, Sinyi Dist, Taipei City 110. Taiwan
Phone: +886-2-8786-6688

Epson Singapore Pte., Ltd.
438B Alexandra Road,
Block B Alexandra TechnoPark, #04-01/04, Singapore 119968
Phone: +65-6586-5500 FAX: +65-6271-7066

Epson Korea Co.,Ltd
10F Posco Tower Yeoksam, Teheranro 134 Gangnam-gu,
Seoul, 06235, Korea
Phone: +82-2-3420-6695

Seiko Epson Corp.
Sales & Marketing Division

Device Sales & Marketing Department
JR Shinjuku Miraina Tower, 4-1-6 Shinjuku,
Shinjuku-ku, Tokyo 160-8801, Japan

Document Code: 413551502
First Issue June 2017
Revised March 2024