Sohwa & Sophia Technologies

Universal Probe

Software Users Manual

Memory Command Builder

No. E090962-03

Table of Contents

NOTES

Thank you for purchasing the "Universal Probe" from Sohwa & Sophia Technologies. Correctly understand notes and other information in this document before using this product.

1. A permission from the Japanese government is required when exporting the product and technologies described in this document to which the "Foreign Exchange and Foreign Trade Act" applies or bringing them out of Japan.
2. The product described in this manual is intended to be used for general electronic equipment, such as office equipment, communication equipment, measuring instruments and home electric appliances. Do not use this product for automobile, railway, marine, aeronautical or space equipment, traffic equipment, combustion equipment, safety devices, medical devices, infrastructure equipment, nuclear power equipment or other types of equipment designed for a specific application that requires special quality and reliability and can directly threaten human life or cause damage to human body when it gets out of order or malfunctions. The customer should be responsible for using this product in such equipment.
3. It is strictly prohibited to reprint this document in whole or in part without our written permission.
4. The contents of this document are subject to change without notice.
5. The specifications in this document may lead to different results depending on your environment or measurement conditions.
6. Note that we bear no responsibility for consequence of operating this product.
7. "Operating Precautions" in this document are important notes to prevent damage on users and third parties as well as assets and to use the product safely and correctly. Be sure to read these precautions before using this product.
8. The product names and trade names in this document are trademarks or registered trademarks of respective owners.

Contact information can be obtained on the website of **Sohwa & Sophia Technologies Inc.**
URL > **http://www.ss-technologies.co.jp/en/index.html**

Operating Precautions

Failure to observe the following precautions may lead to human death or severe injury.

 **Do**	Avoid supplying voltage out of the range specified in the specifications of this product. Supplying the voltage out of the range may cause damage or fire.
 **Do**	When using the target equipped with the ground terminal, ensure that the ground terminals of the target and peripheral equipment are connected. Failure to do so may cause an equipment failure or electric shock. Avoid connecting the ground terminal to the gas pipe. This causes a fire or explosion.
 **Don't**	Do not transport this product with equipment connected. In particular, hold the plug when removing or inserting the cable. Failure to do so may damage the cable, causing a fire or electric shock.
 **Don't**	Observe the following points when handling the cable. Do not damage, process, forcibly bend, twist, pull, putting any object on or heat the cable, moving the cable close to the heating device, or touch the cable with a wet hand. Failure to observe these precautions may cause a fire or electric shock. If the cable is damaged, stop using it.
 **Don't**	When you hear thunders, do not touch the power plug. This causes an electric shock. If the product seems to be damaged by lightning strike, stop using it.
 **Don't**	Do not let a staple, clip or other metal items enter into the product. This may cause a fire or failure.
 **Don't**	Do not use or leave the product in direct sunlight, near heating devices, in an extremely hot or cold environment, under hard vibrations, in dusty area with a large amount of metal dust or oily dust, or noisy area full of spike noise. Do not apply a strong shock to the product.
 **Do not disassemble**	Do not disassemble, alter or repair the product. This may cause a fire or electric shock.
 **No wetting**	Do not use the product at a place where there is liquid or at a humid place such as bathroom or in vicinity to glasses. This may cause an electric shock. If liquid enters into this product, immediately turn it off and stop using it.
 **Caution**	Touching the energized product for a long time may cause low-temperature burns. Do not use this product covering with comforter or other cloth.
 **Pull out the plug.**	Immediately turn the power off if unusual smell, noise, smoke or fire is detected or if the product is or may be damaged due to a fall or strong shock. Continuing to use it may lead to a serious accident. Stop using the product.

Abbreviations, Terms and Conventions

This section describes the abbreviations, terms and conventions used in this document.

- About numeric values ... All the numeric values are positive unless otherwise specified.
- K (capital letter) ... Represents 2^{10}=1024. (Example: 16K=16384)
- k (small letter) ... Represents 1000. (Example: 1kHz=1000Hz)
- [xxxxx] ... Represents the window title, xxxxx.
- <xxxxx> ... Represents the item named xxxxx in the window.

The annotations and notes used in this document are as shown in Figure 1.

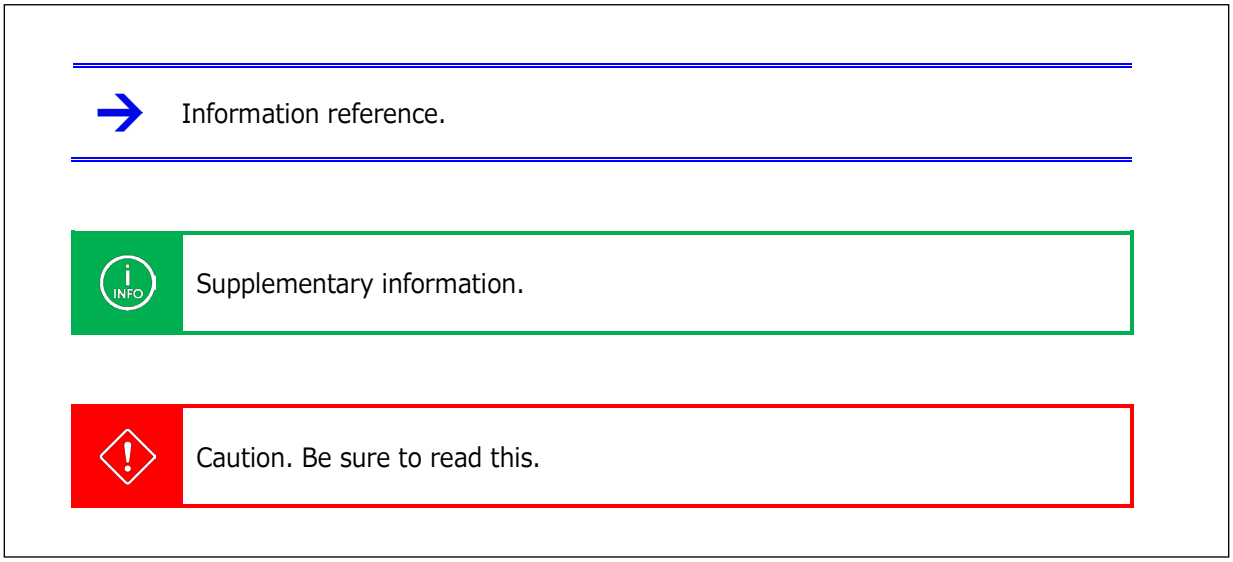


Figure 1

Abbreviations and terms are listed in Table 1.

Table 1

Abbreviations and terms	Description
This software	Memory Command Builder.
Flash memory	Generic term for flash memory, EEPROM, and other memories.
Probe	Universal Probe itself.
Target	Object to be controlled or measured with the Universal Probe.

1. Overview of This Software

1.1. About This Software

The debugger (WATCHPOINT) and Flash Memory Writer from Sohwa & Sophia Technologies Inc. support access to several flash memories and SPI serial memories as default.
This software allows the user to prepare the configuration file for flash memory or SPI serial memory not supported by default and access such memories.

This software is a tool to create a **flash memory configuration file (*.fsh)** required to operate the flash memory and SPI serial memory.

There are four types of memory access processes: “Read,” “Write,” “Block Erase,” and “Chip Erase.” Although it is necessary to assemble what commands are executed with what procedure for these processes, operation is very easy. All you have to do is to drag and drop the commands to be executed from the command list and arrange them in the processing order.

1.2. System Requirements

The following system is required to run this software.

- PC running Microsoft Windows 7 or later
- CPU: 1GHz or faster (depending on the requirements of the used OS)
- Memory: 1GB or larger (depending on the requirements of the used OS)
- HDD: 500MB or larger free hard disk space
- OS: Windows 7 or later (32bit or 64bit versions are supported)
- .NET Framework 4.5

ARM Writer, SPI Writer or WATCHPOINT are required to use the flash memory configuration file created with this software.

1.3. Mechanism of Writing Data into the Flash Memory

The flash memory configuration file (*.fsh) created by using the Memory Command Builder is used to add settings for reading or writing data from/into flash memories not supported by default by the ARM Writer and SPI Writer from Sohwa & Sophia Technologies Inc.
It is necessary to copy user data (i.e. object data) to be written and the memory control program (i.e. read/write program) to the built-in RAM in the CPU and have the CPU itself write the data into the flash memory in order to write data in the built-in flash memory in the ARM CPU or the flash memory connected to the CPU (i.e. external flash memory) via the JTAG/SWD interface.

- The SPI Writer, which directly writes data into the flash memory via the SPI interface, does not have the mechanism where the CPU writes data by itself.
- The ARM Writer, which writes data into the flash memory via CPU, does not have the mechanism where writes data by SPI Interface.

1.3.1. Overview of Writing Data into the Built-in Flash Memory in the ARM CPU

This section provides an overview of how to write object data into the built-in flash memory in the ARM CPU.

(1) The read/write program is copied to the built-in RAM via the JTAG/SWD interface.

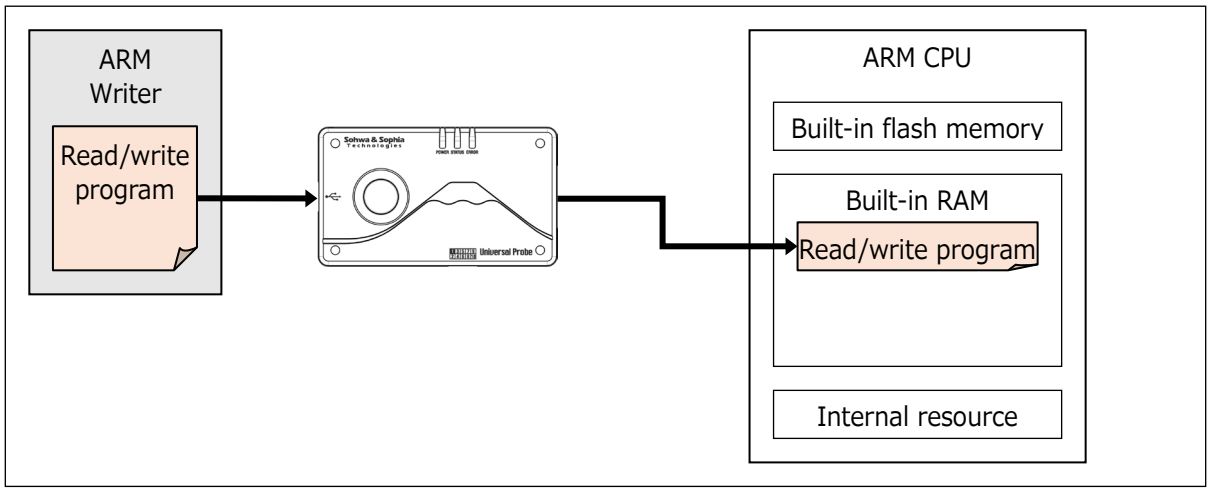


Figure 2

(2) When the <Erase all data from flash memory before download> option checkbox is selected in the ARM Writer, the data in the chip is erased at first. Initialization is performed to access the flash memory in the CPU at the beginning of chip data erasing processing.

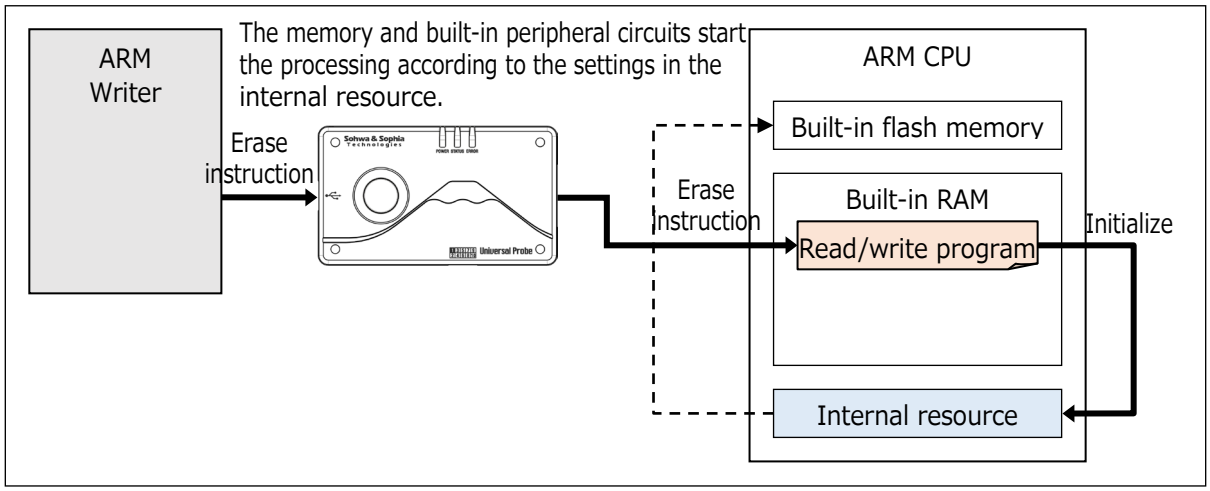


Figure 3

(3) The read/write program erases data from the built-in flash memory after the CPU is initialized.

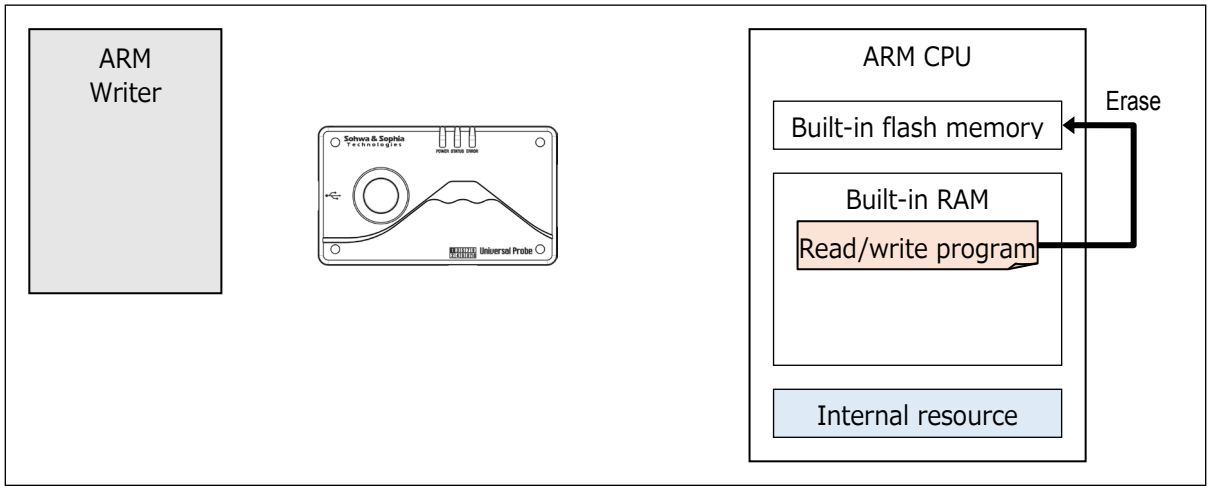


Figure 4

(4) After the data is erased, 4KByte object data is copied to the built-in RAM via the JTAG/SWD interface.

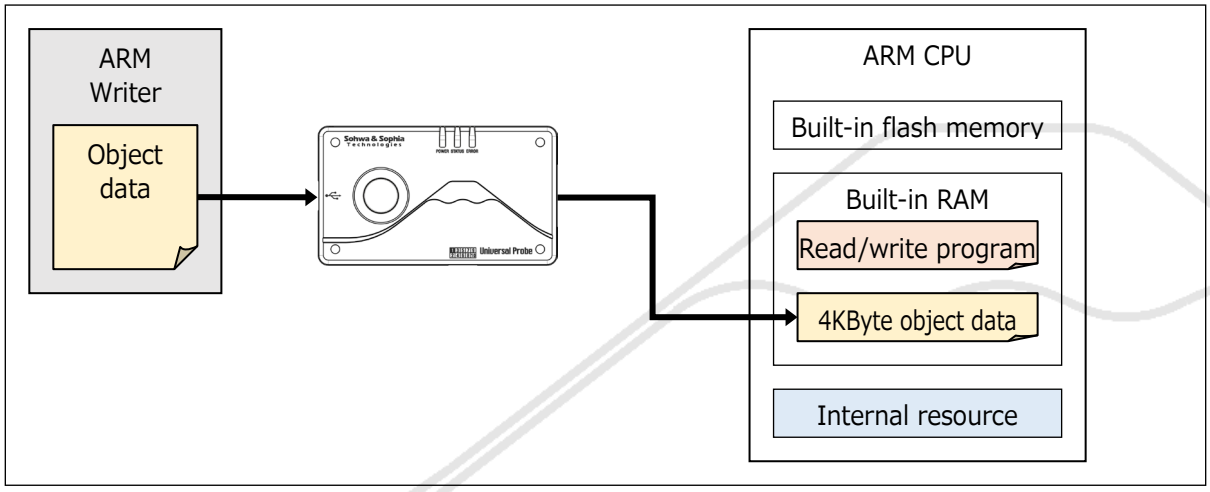


Figure 5

(5) The ARM Writer writes the copied read/write program.

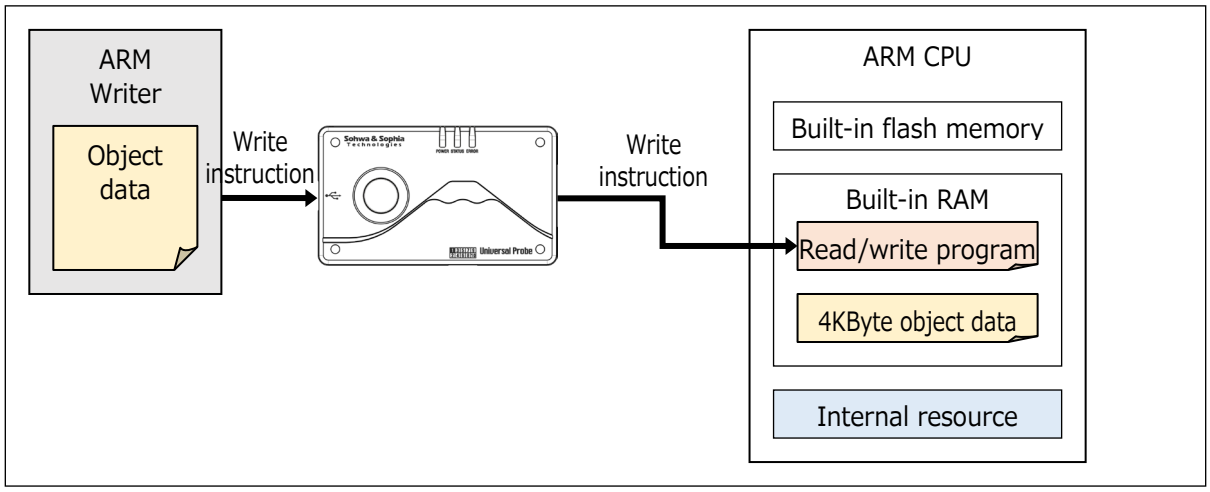


Figure 6

(6) The read/write program initializes the CPU to write data into the flash memory before writing.

INFO The CPU must be initialized both at the chip data erasing process and before writing.

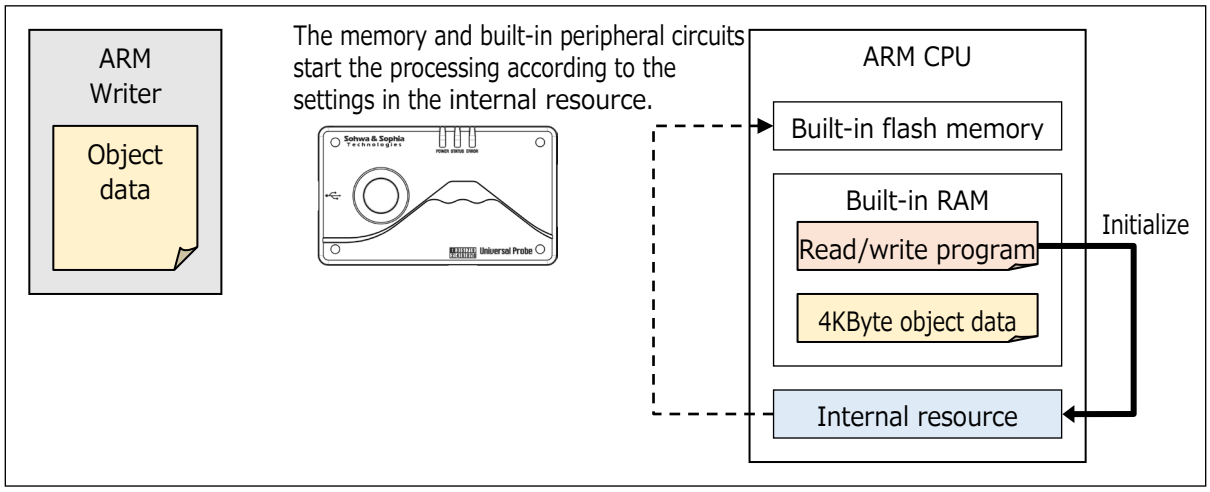


Figure 7

(7) Writing starts after the CPU is initialized.

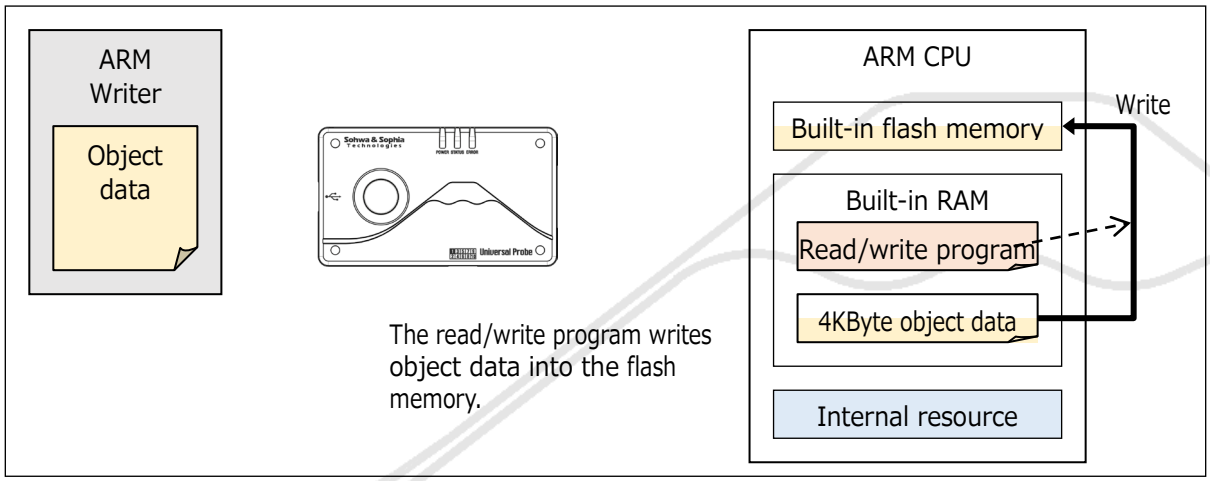


Figure 8

(8) After 4KByte data is written, the next 4KByte data is requested to the ARM Writer.

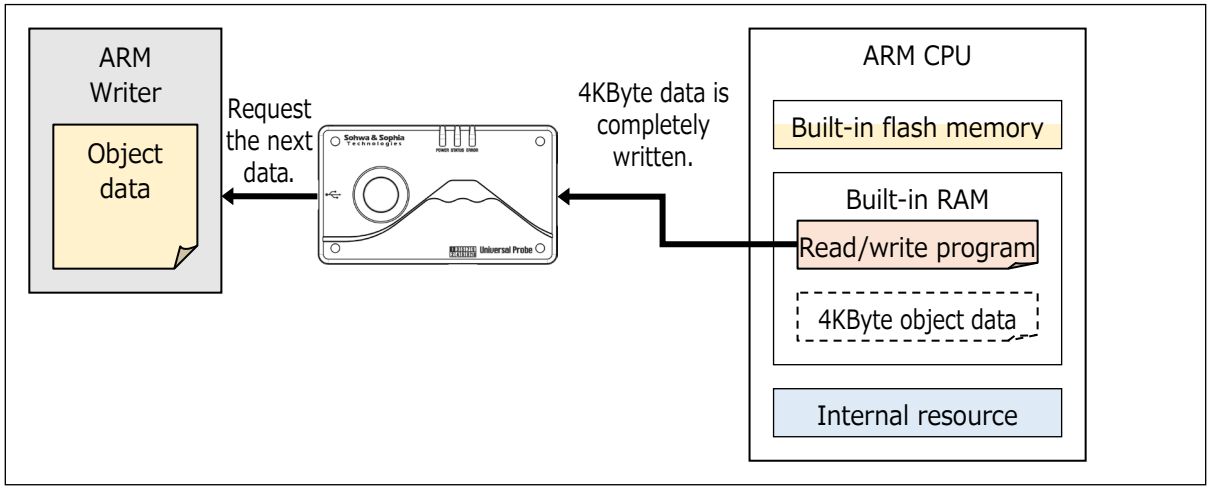


Figure 9

(9) The next 4KByte data is copied to the built-in RAM via the JTAG/SWD interface.

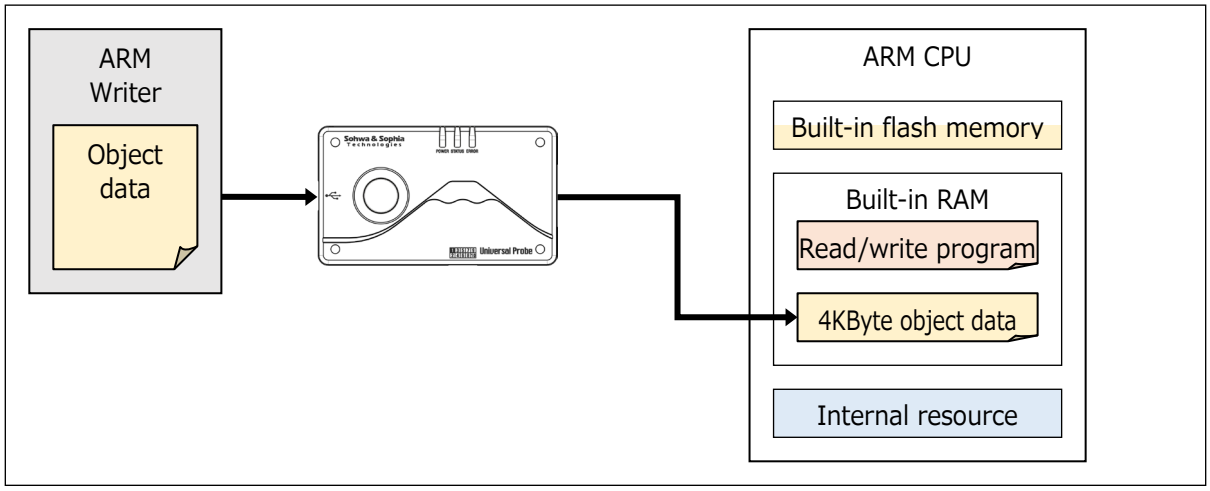


Figure 10

(10) The new 4KByte data is written after the address at which the last data was written.

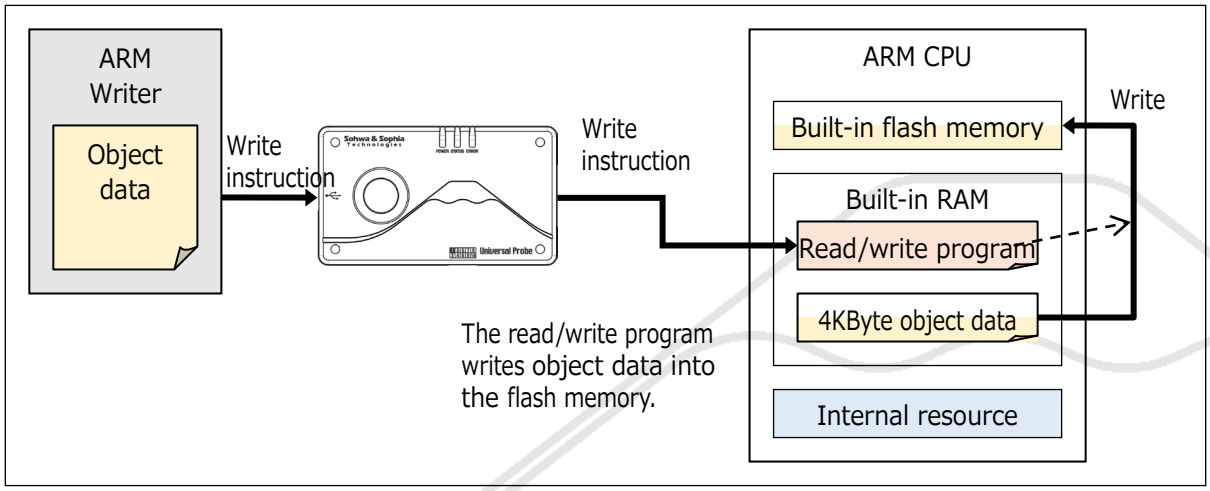


Figure 11

(11) Steps (8) through (10) are repeated until all the object data is processed.

1.3.2. Overview of Writing Data into the Flash Memory External to the ARM CPU

This section provides an overview of how to write object data into the flash memory external to the ARM CPU.

(1) The read/write program is copied to the built-in RAM via the JTAG/SWD interface.

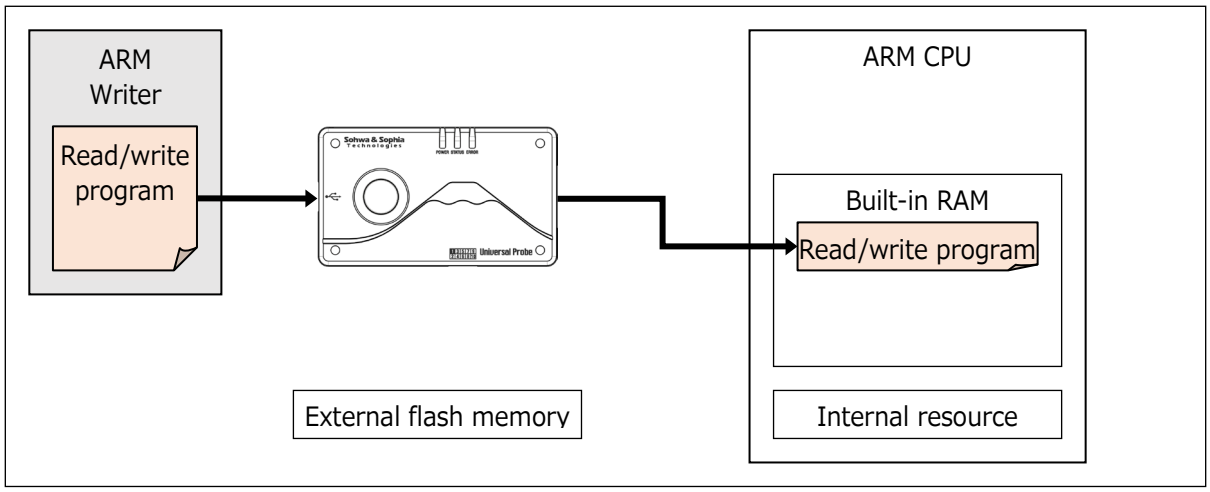


Figure 12

(2) When the <Erase all data from flash memory before download> option checkbox is selected in the ARM Writer, the data in the chip is erased at first. Initialization is performed to access the external flash memory at the beginning of chip data erasing processing.

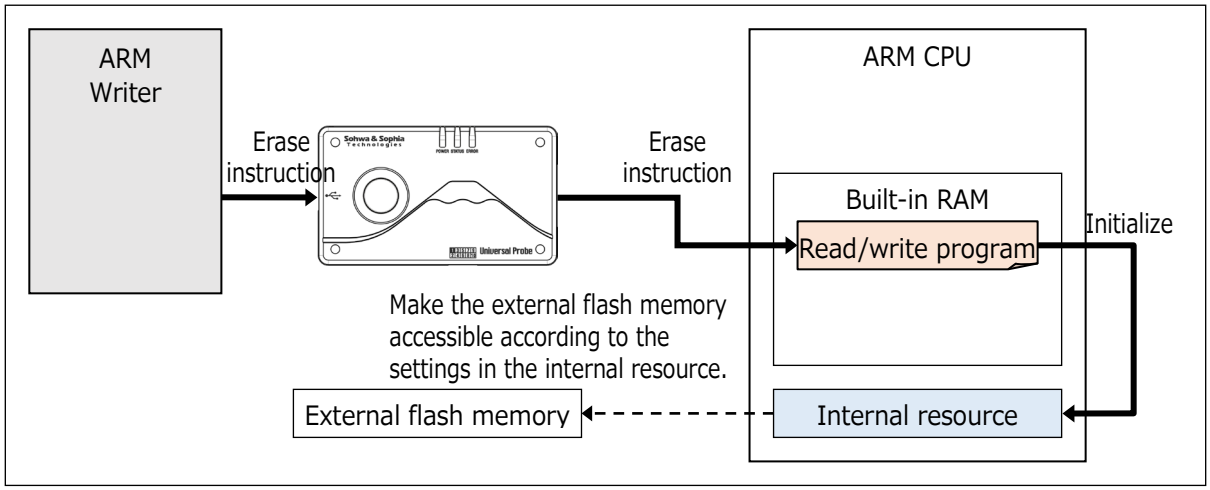


Figure 13

(3) The read/write program erases data from the built-in flash memory by operating the internal resource after the CPU is initialized.

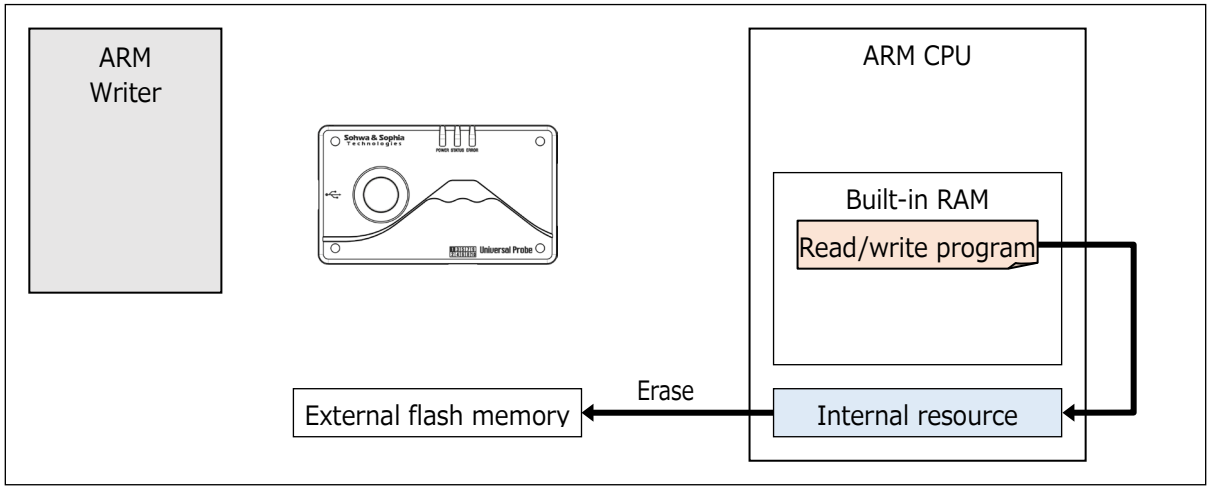


Figure 14

(4) After the data is erased, 4KByte user data is copied to the built-in RAM via the JTAG/SWD interface.
Object data is then copied to the built-in RAM in the ARM CPU in 4KByte blocks.

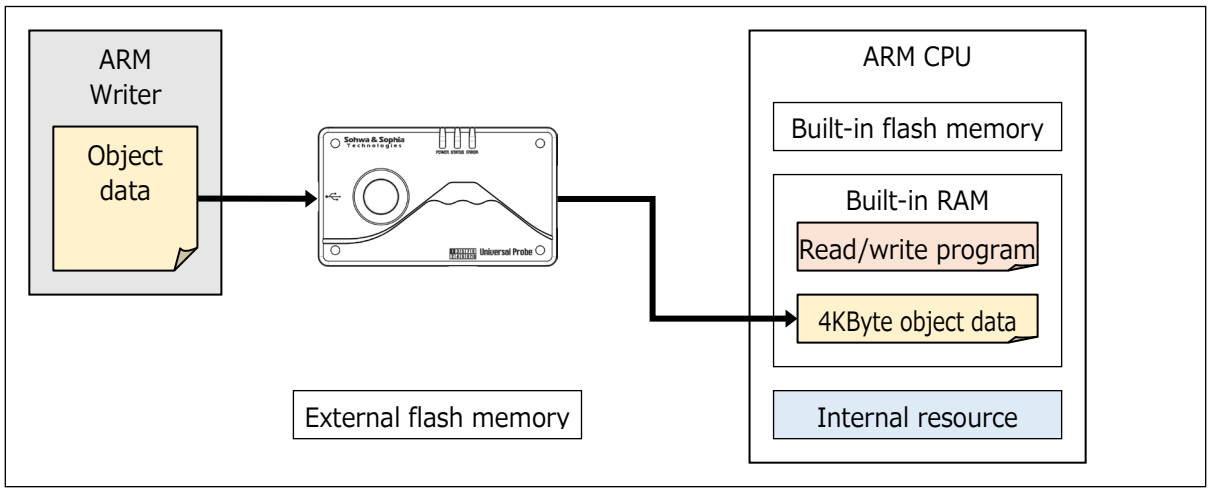


Figure 15

(5) The ARM Writer instructs to write the copied data to the read/write program.

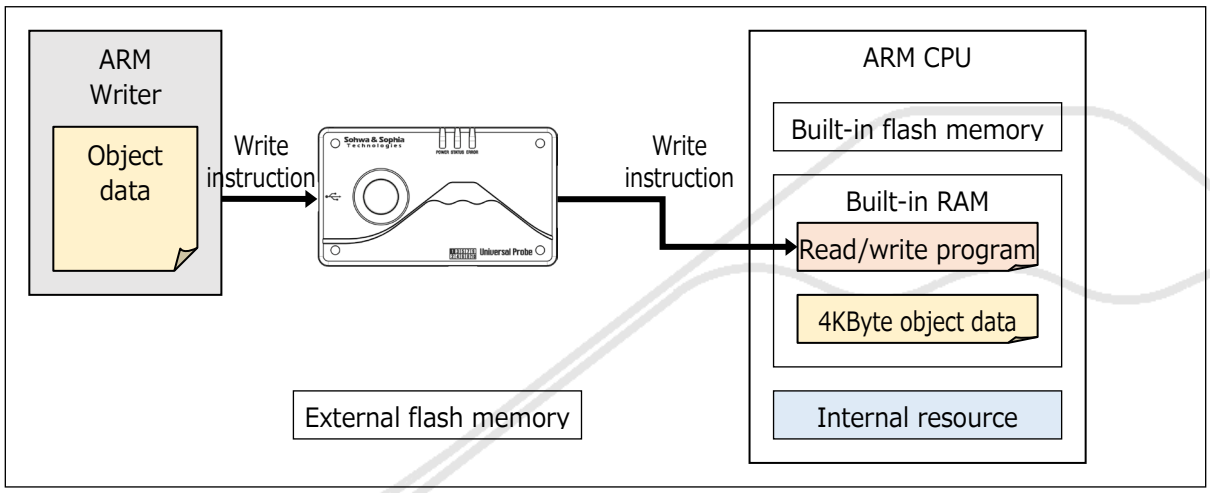


Figure 16

(6) The read/write program initializes the CPU to write data into the flash memory before writing.

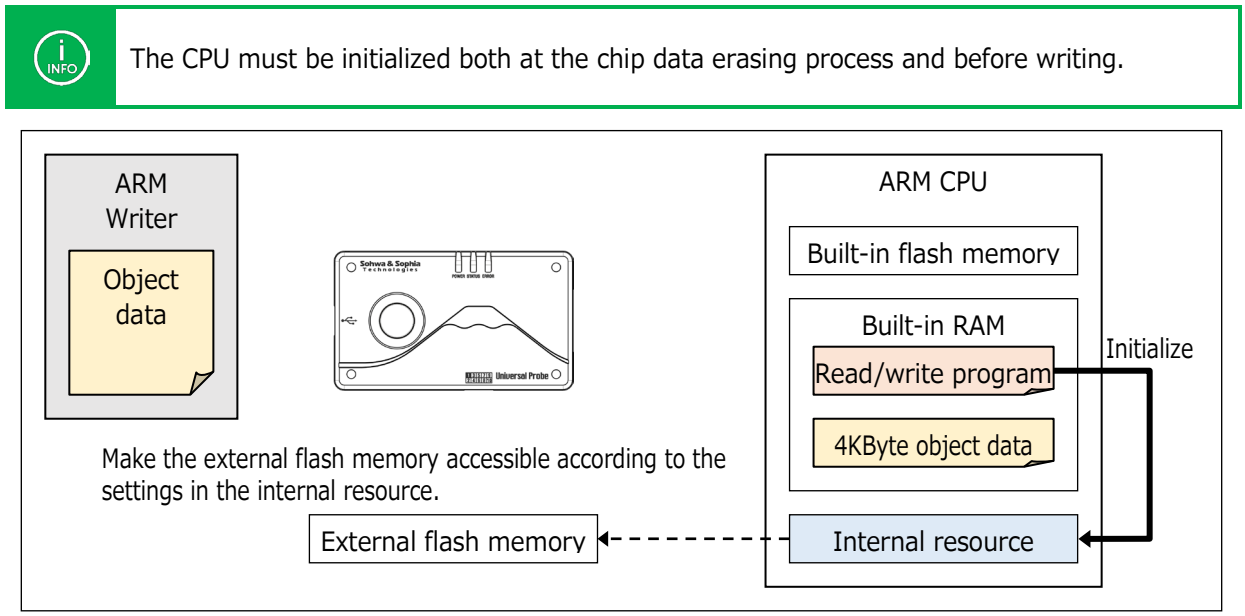


Figure 17

(7) Writing starts after the CPU is initialized.

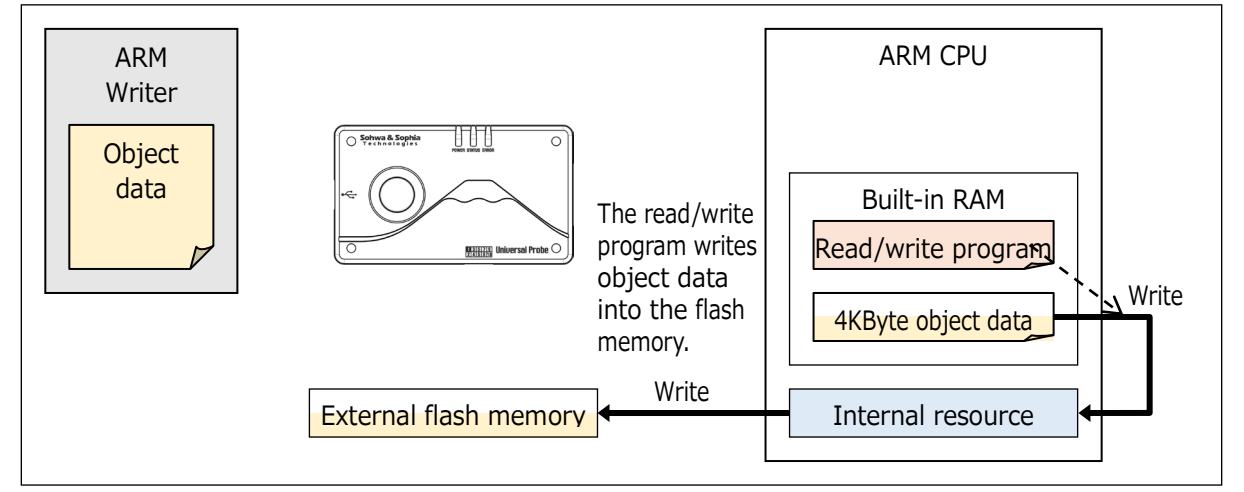


Figure 18

(8) After 4KByte data is written, the next 4 KB data is requested to the ARM Writer.

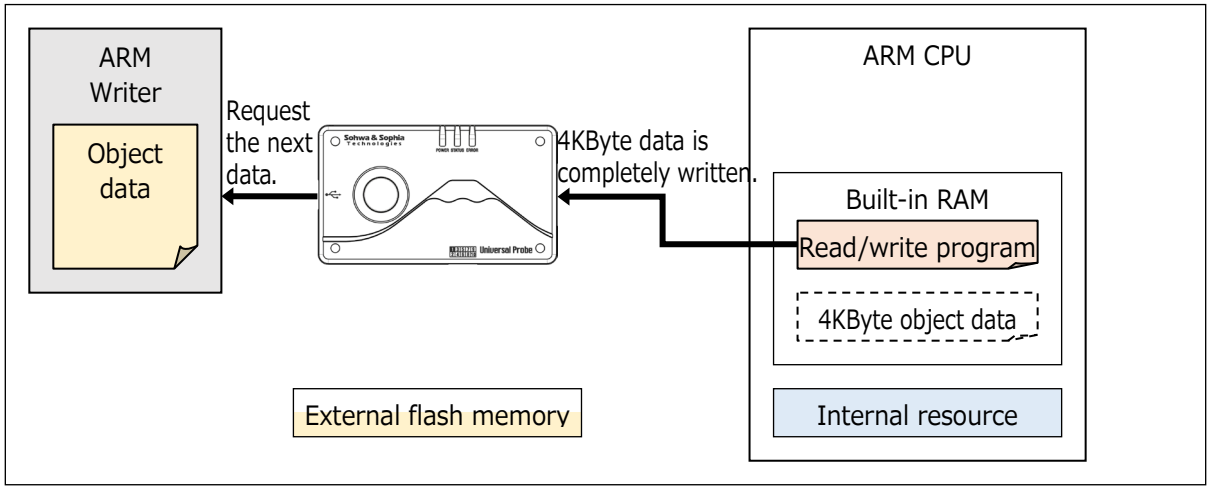


Figure 19

(9) The next 4KByte data is copied to the built-in RAM via the JTAG/SWD interface.

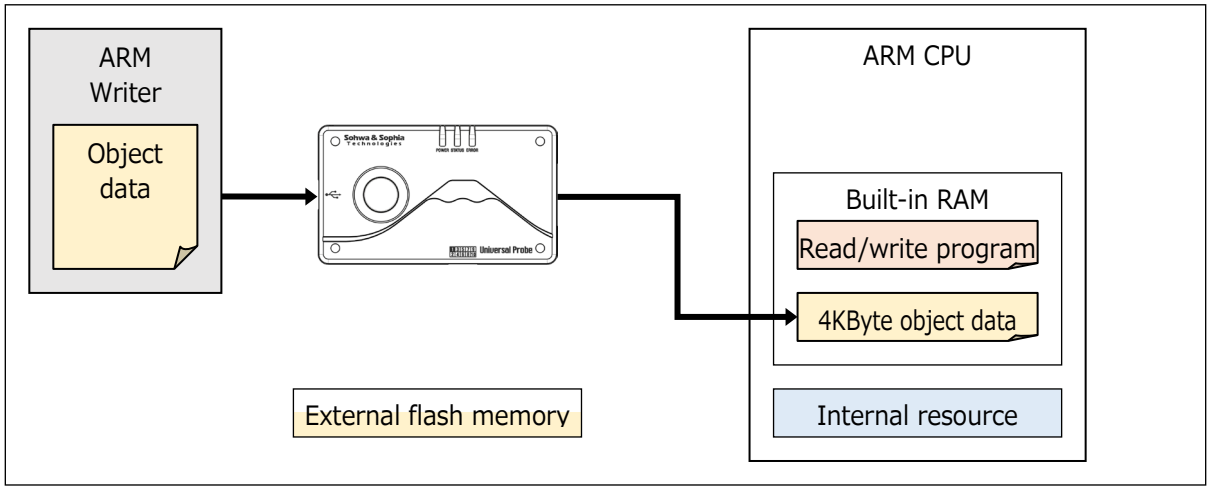


Figure 20

(10) The new 4KByte data is written after the address at which the last data was written.

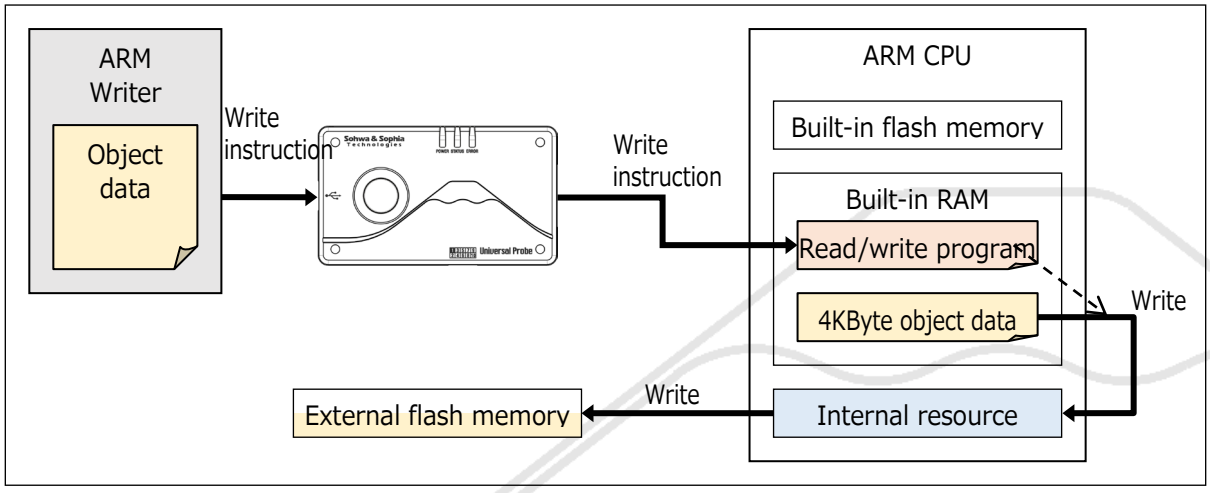


Figure 21

(11) Steps (8) through (10) are repeated until all the object data is processed.

1.3.3. Overview of Writing Data into the SPI Serial Memory

This section provides an overview of how to write object data into the SPI serial memory.

(1) The read/write program is copied to the built-in memory in the probe.

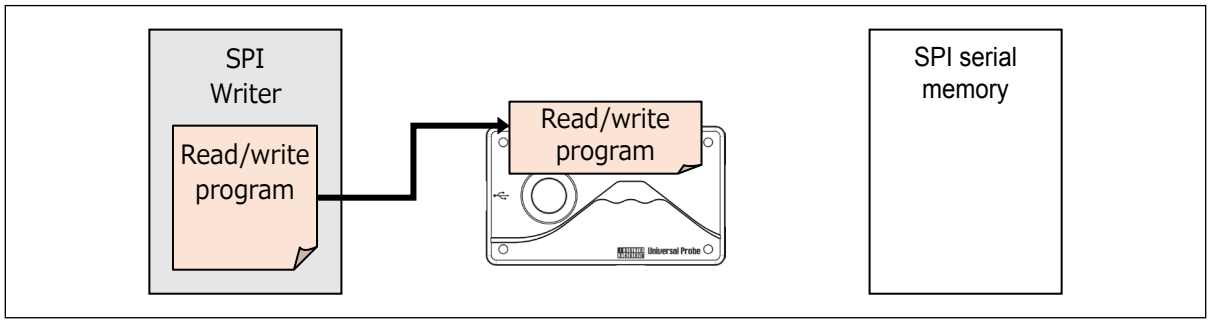


Figure 22

(2) When the <Automatically erase data from chip before download> option checkbox is selected in the SPI Writer, the data in the chip is erased at first.

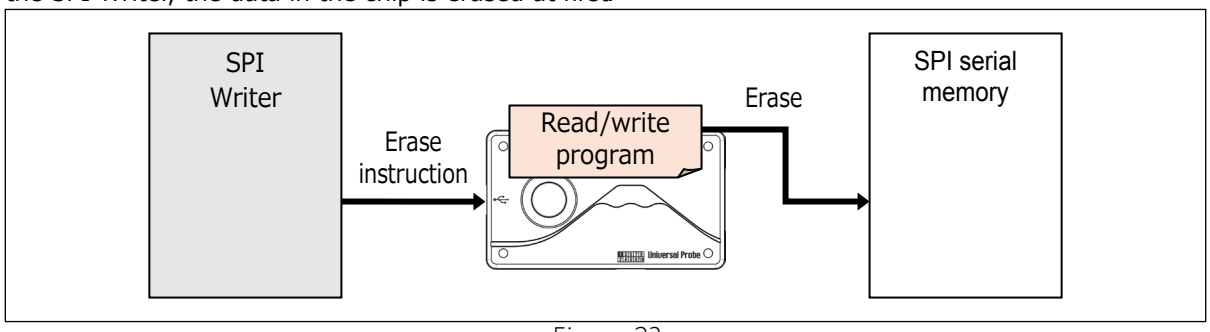


Figure 23

(3) Object data is copied to the built-in memory in the probe. The data is copied in 4KByte blocks.

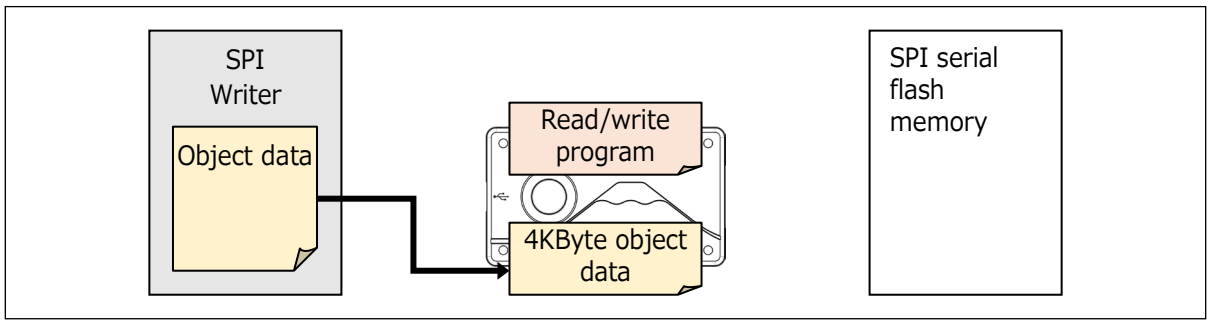


Figure 24

(4) The SPI Writer instructs to write the copied data to read/write program.

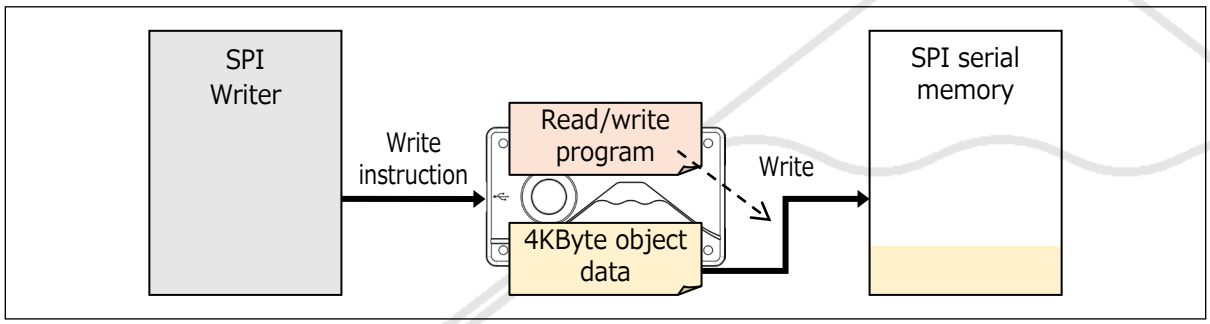


Figure 25

(5) After 4KByte data is written, the next 4 KB data is requested to the SPI Writer.

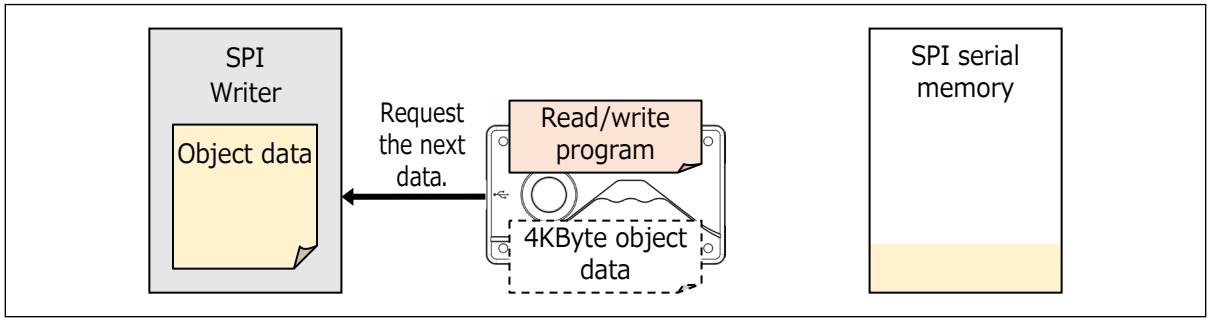


Figure 26

(6) Steps (3) through (5) are repeated until all the object data is processed.

1.3.4. Usage of Built-in RAM

This section describes the usage of the built-in RAM mentioned in “1.3.1. Overview of Writing Data into the Built-in Flash Memory in the ARM CPU” and “1.3.2. Overview of Writing Data into the Flash Memory External to the ARM CPU.”

The ARM Writer assigns a memory area called "work memory” to the built-in RAM.
Usually, 8 KByte is assigned to the work memory.
If the RAM capacity in your CPU is less than 8 KByte, specify a value within that RAM capacity.

In this example, it is assumed that the capacity of used work memory is 8 KByte.

The 8 KByte work memory contains the parameter area, program area, and data area. Their standard addresses are allocated with the offsets shown in Figure 27.

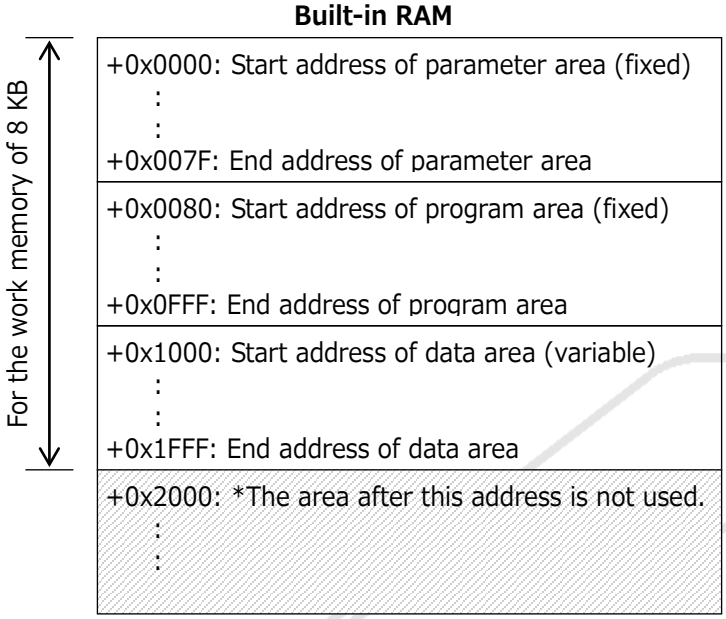


Figure 27

(1) Start address of the work memory

Any address can be allocated as the start address of the work memory as long as that address is within the range of the built-in RAM. However, note that addresses must be in four-byte blocks. In addition, specify the address so that the whole work memory fits into the built-in RAM area.
Usually, specify the start address of the built-in RAM.

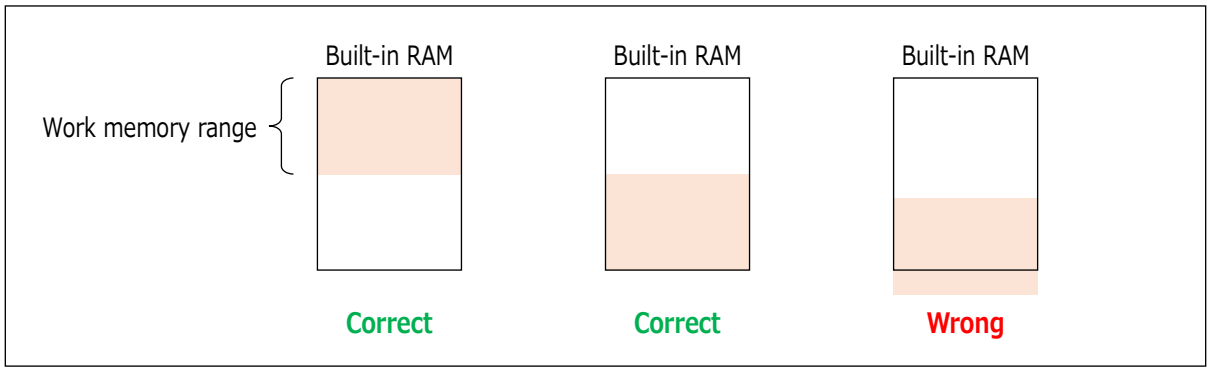


Figure 28

(2) Work memory range length

Specify the size of the work memory.
Usually, specify 8 KByte.
If the size of built-in RAM is less than 8 KByte, specify a value within the size of that built-in RAM.

(3) Parameter area in the work memory

The first 128 bytes in the work memory (with offsets of 0x0000 to 0x007F) are the parameter area for using the read/write program. This area is used to communicate information between the read/write program and the probe, for example.
User operations are not required.

Table 2

Offset	Description
+0x0000	**[At beginning of processing] Sets the operation mode of the read/write program** (from the probe to the read/write program.) 1: Read 2: Write 3: Block Erase 4: Chip Erase **[At end of processing] Displays the execution result of the read/write program** (from the read/write program to the probe.) 0: Finished normally Other than 0: Finished with error
+0x0004	**Address of data to be processed** Indicates the start address of data to be read when the operation mode is Read. Indicates the start address of data to be written when the operation mode is Write. Indicates the address of the block or device from which data is erased when the operation mode is Block Erase or Chip Erase.

+0x0008	**Start address of the data storage area** Read data is stored from the address specified here when the operation mode is Read. Written data is stored from the address specified here when the operation mode is Write.
+0x000C	**Data length** Specify the length of data to be read when the operation mode is Read. Specify the length of data to be written when the operation mode is Write.
+0x0010	**End address of data to be processed** Processing branches to the address specified here after finished.
+0x0014	**(Reserved)**
+0x0018	**(Reserved)**
+0x001C	**(Reserved)**
+0x0020 : +0x003C	**Area used by the probe** Do not access this area.
+0x0040 : +0x007F	**(Reserved)**

(4) Program area in the work memory

The read/write program is stored in this area.
The start address of this area is always (1) start address of the work memory + 0x0080.
User operations are not required.

(5) Data area in the work memory

The start address of the data area is determined as described below.

- When the work memory range length is less than 8 KByte: **the address at the center of the work memory is automatically set as the start address.**
- When the work memory range length is 8 KByte or more: **the address 4 KByte after the beginning of the work memory is set as the start address.**

When the work memory of 8 KByte is set, the area after the 4 KByte position is used as the data area.

(6) Other information

The target CPU registers have the value shown in the table below, immediately after the ARM Writer calls each processing in the read/write program.
User operations are not required.

For ARM

Table 3

Register	Description
PC	Start address of read/write program (start address of the work memory + 0x080 in standard configuration)
R14	End address of data to be processed
R13	Start address of the data area (The area immediately before the data area can be used as the stack for the read/write program.)
R0	Start address of the parameter area
CPSR	0xD3 (IRQ, FIQ prohibited, ARM State, Supervisor Mode)

For Thumb2

Table 4

Register	Description
PC	Start address of read/write program (start address of the work memory + 0x080 in standard configuration)
LR	End address of data to be processed
SP_main, SP_process	Start address of the data area (The area immediately before the data area can be used as the stack for the read/write program.)
R0	Start address of the parameter area
xPSR	0x01000000

The registers not described in

and
Table are indefinite.

1.3.5. Internal Structure of Read/Write Program

The Memory Command Builder is a tool to create **flash memory configuration files (*.fsh)**, which include the read/write program.

The read/write program contains the procedures on the “Read,” “Write,” “Block Erase,” and “Chip Erase” tabs that are created by using the Memory Command Builder and converted into a single binary code. This binary code is copied to the program area in the work memory.

The ARM or SPI Writer gives instructions to execute these procedures via the probe when writing data or erasing data from the chip.

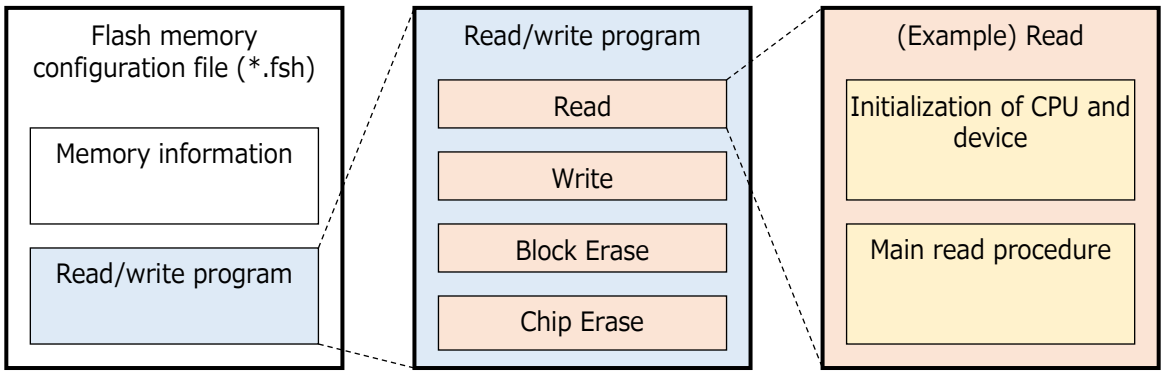


Figure 29

The Memory Command Builder does not automatically generate the initialization program for the ARM CPU. It is, therefore, necessary to incorporate the CPU initialization procedure at the beginning of each procedure. (*This is why the CPU is repeatedly initialized in “1.3.1. Overview of Writing Data into the Built-in Flash Memory in the ARM CPU” and “1.3.2. Overview of Writing Data into the Flash Memory External to the ARM CPU1.3.2. Overview of Writing Data into the Flash Memory External to the ARM CPU
.”)
If the device in which data is written requests the initialization of that device, it is also necessary to include this procedure in the initialization processing.

For the SPI Writer, there is only one initialization processing, which is requested on the device.

2. How to Start

2.1. Starting Operations

Select “Start” → “All Programs” → “Memory Command Builder” → “Memory Command Builder” to start the Memory Command Builder.

2.2. Screen at Startup

Figure 30 shows the screen at startup.

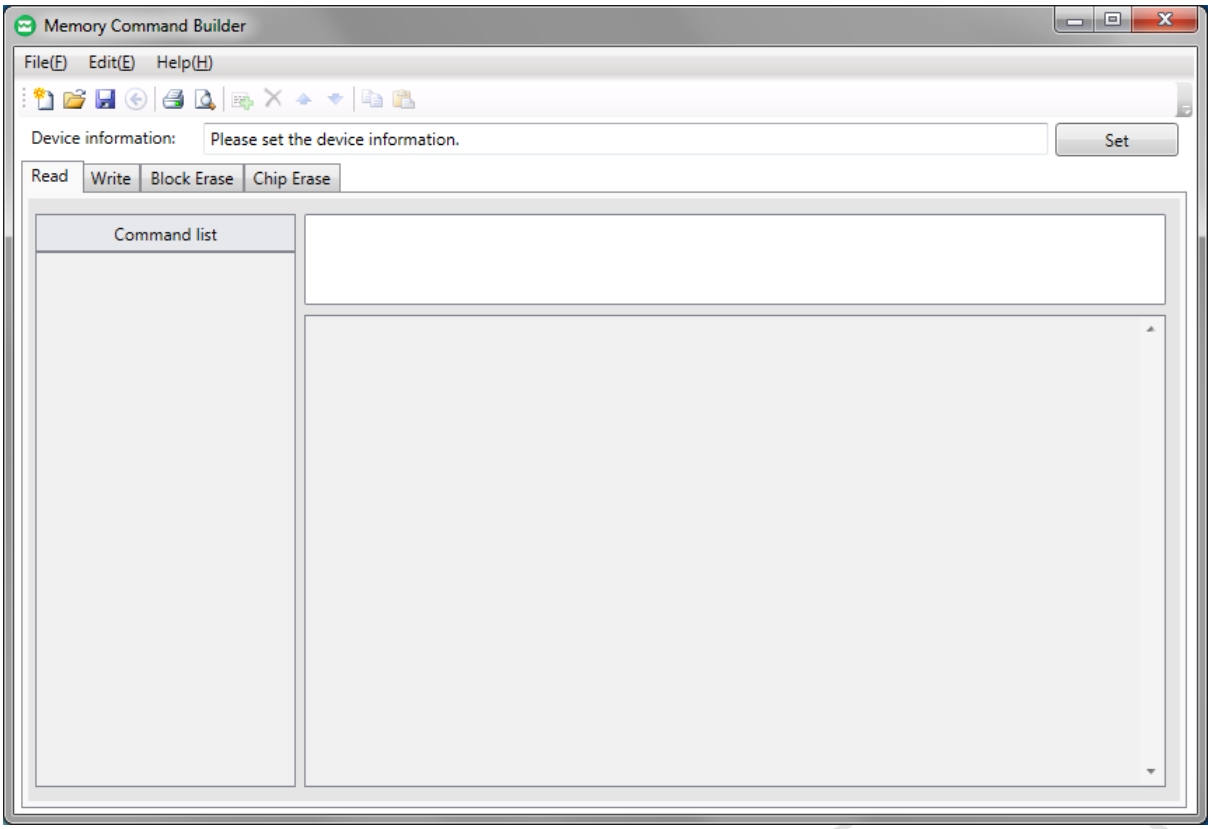


Figure 30

2.3. Operation after Startup

Immediately after starting the software, set the device information.
Press Set button or double-click the text box that displays “Enter device information.” to display the [Device information] window.
Specifying the connection type between the probe and the target in the [Device information] window displays the available commands in the command list.

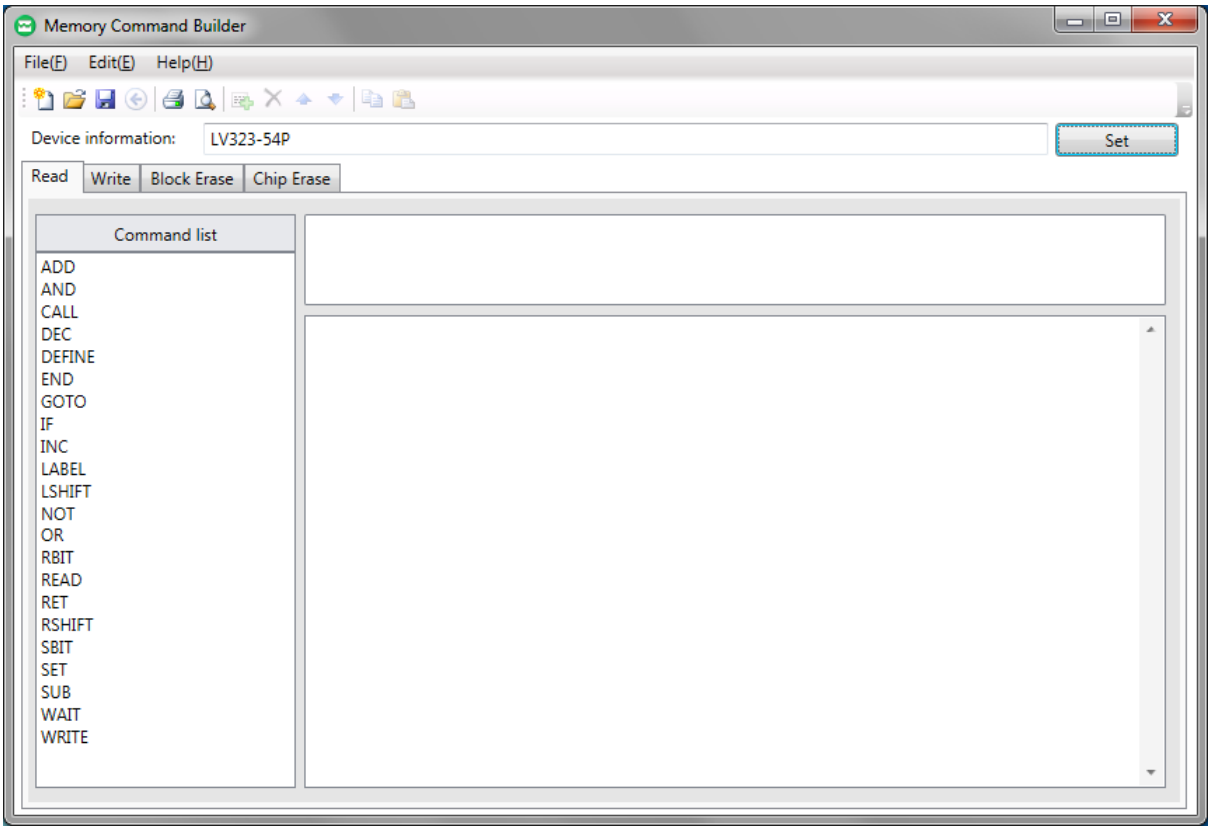


Figure 31

→ For how to set the connection type, refer to “**3.2. Device information window**.”

3. Screen Description

3.1. Memory Command Builder Window

The Memory Command Builder window is shown below.

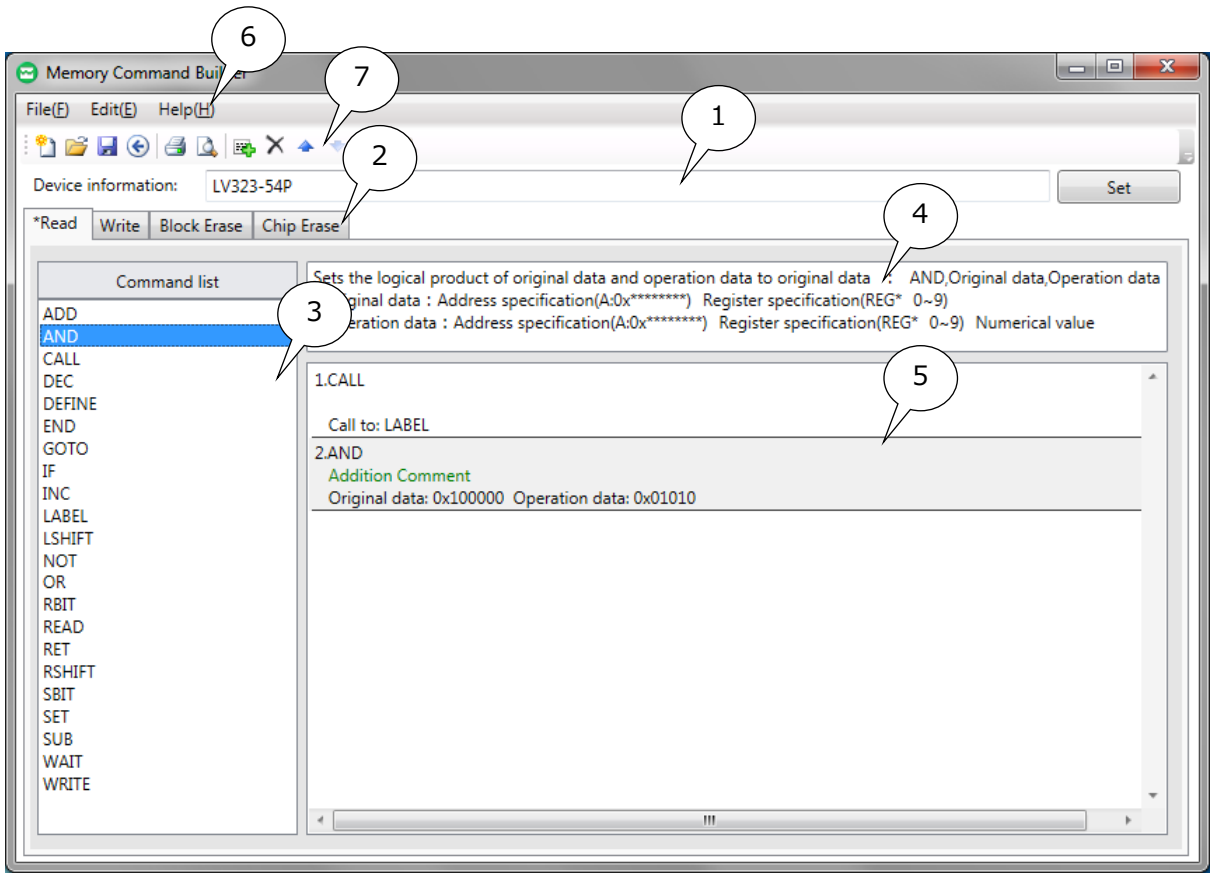


Figure 32

(1) Device information

Displays the set device name.
Double-click the text box or press the Set button to display the window to set device information.

→ For details, refer to "**3.2. Device information window**."

(2) Switch of processing

To switch the assembly processing, click each processing tab.
An asterisk (*) is attached to edited processing tabs.

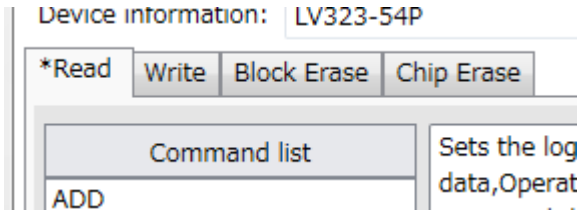


Figure 33

Table 5

Read	Processing to read data from the flash memory
Write	Processing to write data to the flash memory
Block Erase	Processing to delete a block *Delete some data from the flash memory.
Chip Erase	Processing to delete data from the chip *Delete all data from the flash memory.
Status Read	Processing to read status data *This is shown when the connection type is SPI in the device information. This is executed in Waiting for status command.

(3) Command list

Displays the commands that can be added to the assembly list. Displayed commands differ depending on the set connection type.
To add a command, select “Edit” → “Add” from the menu bar, press button on the toolbar, select “Add” from the context menu, drag and drop the command or double-click the command.
Right-click within the command list to display “Add” in the context menu.
Adding a command displays the [Parameter Settings] window. Set parameters.
If you select Cancel while setting parameters, the command is not added.

Dragging a command from the command list to the assembly list displays the position where the command is inserted with a dotted line.
Dropping the command inserts it above the command displayed with the dotted line.

Figure 34 shows the window displayed when the command is dragged and dropped.

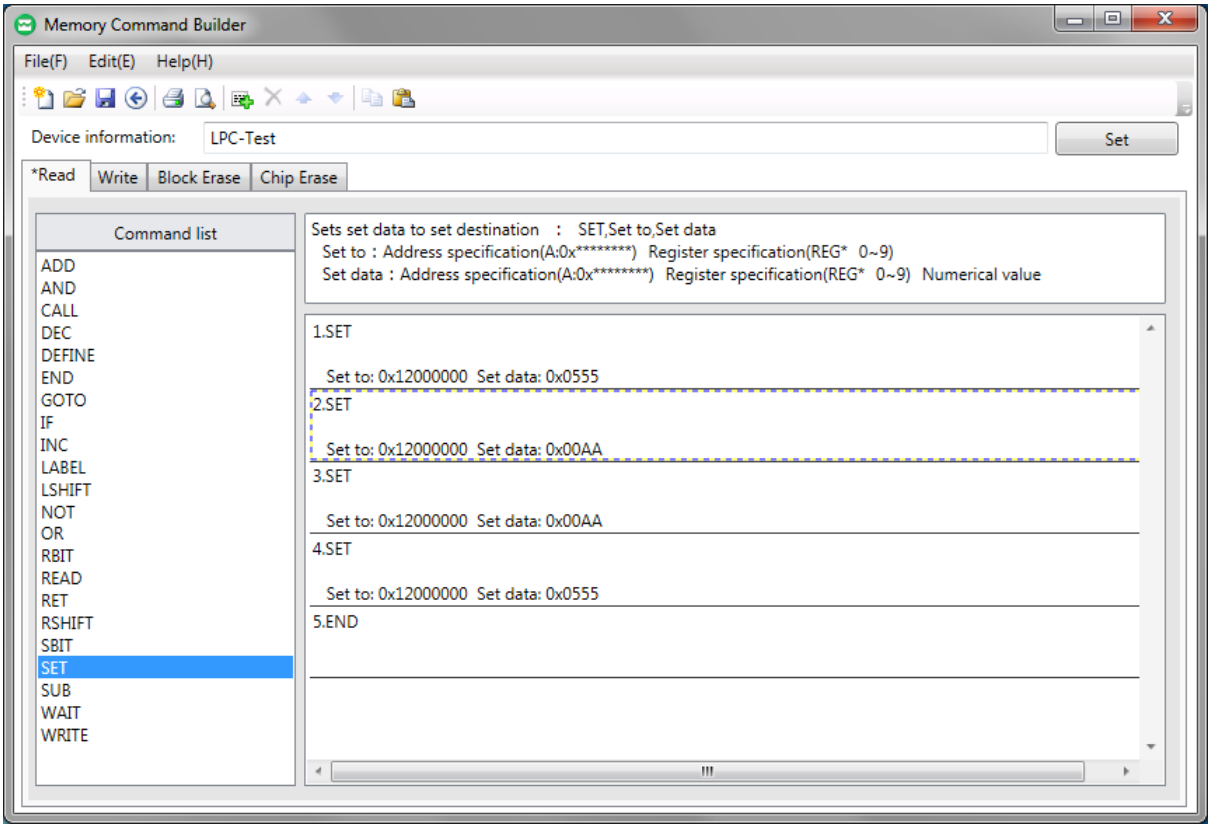


Figure 34

(4) Command information output text

Displays the information on the command selected in the command list.

(5) Assembly list

Displays the information on each command for which parameters are set.
Command parameters can be edited by double-clicking the command in the assembly list.
Right-click within the assembly list to display the context menu.
Multiple commands can be selected in the list while holding down the Shift key.
Figure 35 shows the state when multiple commands are selected.

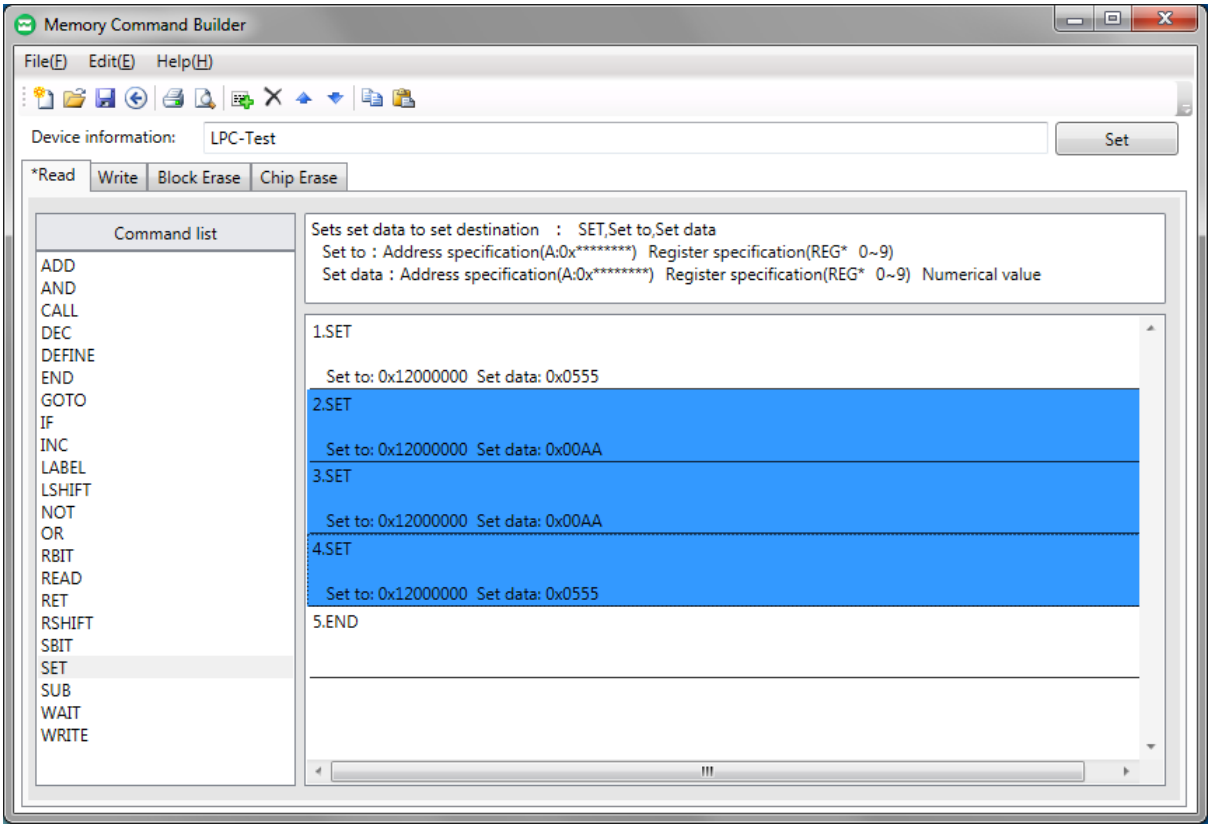


Figure 35

→ For how to set parameters, refer to "**3.3. Parameter Settings Window_4.3_パラメータ設定ウィンドウ**."

(6) Menu bar

●File(F)

Table 6

New	Creates a new assembly list.
Open	Reads an assembly list file.
Save	Overwrites an assembly list file.
Save As	Saves the current assembly list under another name.
Output Flash Memory Configuration File	Displays the [Flash Memory Configuration File Creation Information] window.
Print	Displays the Print dialog box.
Print Preview	Displays a print preview.
Page Settings	Displays the print page settings screen.
Exit	Exits the Memory Command Builder.

● Edit(E)

Table 7

Add	When a command is selected in the assembly list: The command is inserted above the selected command. When no command is selected in the assembly list: The command is added at the bottom of the assembly list.
Delete	Deletes the command selected in the assembly list.
Move Upward	Moves the command selected in the assembly list upward by one.
Move Downward	Moves the command selected in the assembly list downward by one.
Copy	Copies the command selected in the assembly list.
Paste	Pastes the copied command above the command selected in the assembly list.

● Help(H)

Table 8

Help	Displays help information.
About	Displays version information.

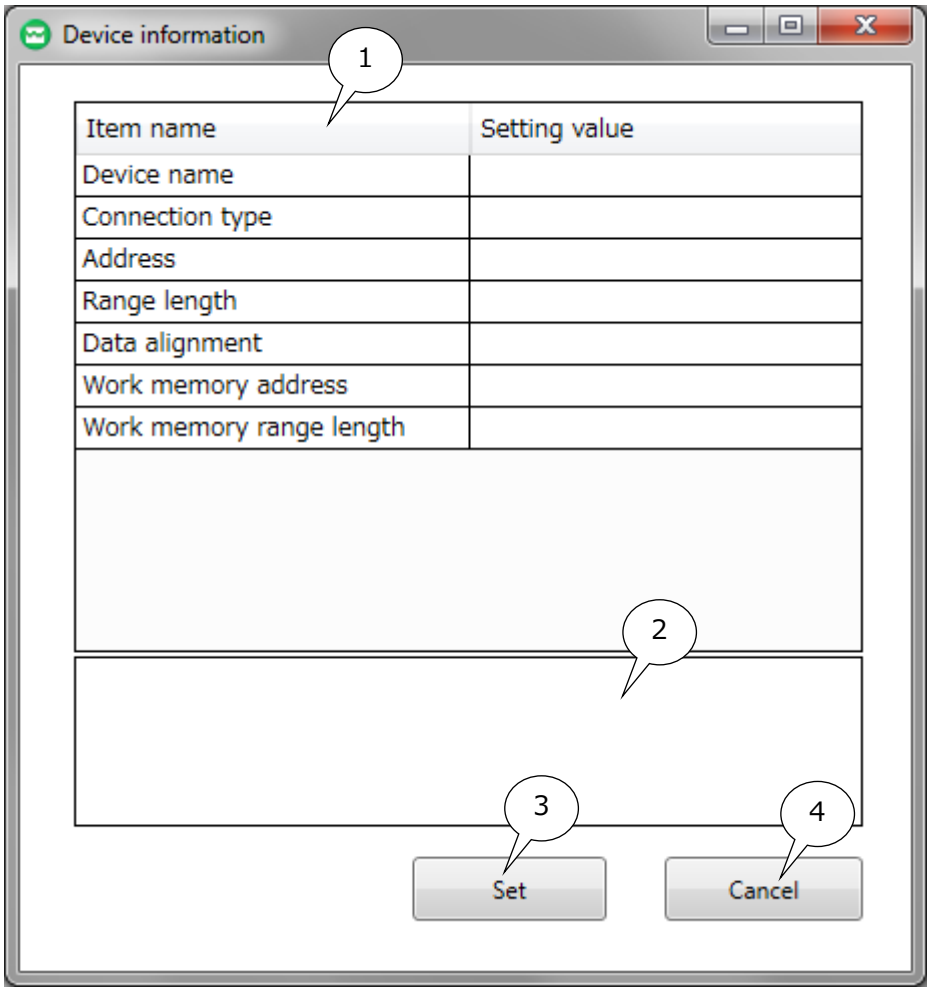
(7) Toolbar

Table 9

New		Deletes all the commands from the assembly list and create the new file.
Open		Reads an assembly list file.
Save		Overwrites an assembly list file.
Output Flash Memory Configuration File		Displays the [Flash Memory Configuration File Creation Information] window.
Print		Displays the Print dialog box.
Print Preview		Displays a print preview.
Add		When a command is selected in the assembly list: The command is inserted above the selected command. When no command is selected in the assembly list: The command is added at the bottom of the assembly list.
Delete		Deletes the command selected in the assembly list.
Move Upward		Moves the command selected in the assembly list upward by one.
Move Downward		Moves the command selected in the assembly list downward by one.
Copy		Copies the command selected in the assembly list.
Paste		Pastes the copied command above the command selected in the assembly list.

3.2. Device information window

The [Device information] window is used to set device information.
Double-click the device information text displayed in the [Memory Command Builder] window or press Set button to display the [Device information] window.



Double-clicking each item other than the connection type in Figure 36 displays the Enter Set Values dialog box.

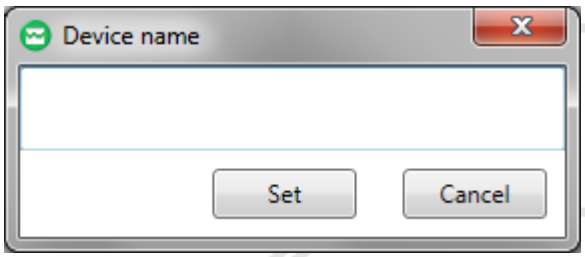


Figure 37

Press Set button to set entered items.

Double-clicking the connection type item in Figure 36 displays the Select Connection Type dialog box. Select the interface between the probe and the target.

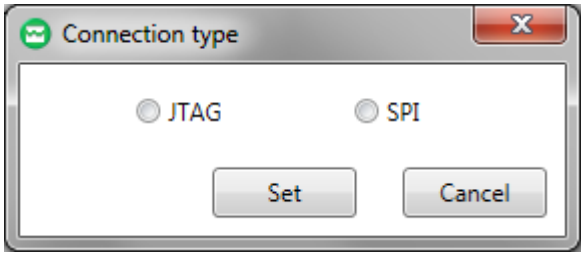


Figure 38

(1) Item name

Table 10

Device name	Set the memory device name.
Connection type	Set the connection interface between the probe and the target. - Select "JTAG" when it will connect by JTAG or SWD. - Select "SPI" when it will connect by SPI.
Address	Set the start address of custom program processing.
Range length	Set the range length of custom program.
Data alignment	Set the data alignment processed by the custom program. - When 1 is set, alignment is not adjusted. - Any other value must be 2^n.
Work memory address	Set the start address of the work memory.
Work memory range length	Set the range length of the work memory.

For work memory, refer to "**1.3.4. Usage of Built-in RAM**".

Although the flash memory configuration file can be generated with both connection types, it is necessary to purchase the corresponding tool.

(2) Item information output text

Text on the button displays the information on the item selected in the list.

(3) Set button

Updates device information and closes the [Device information] window.

(4) Cancel button

Closes the window.

3.3. Parameter Settings Window

The Parameter Settings window is used to set command parameters.

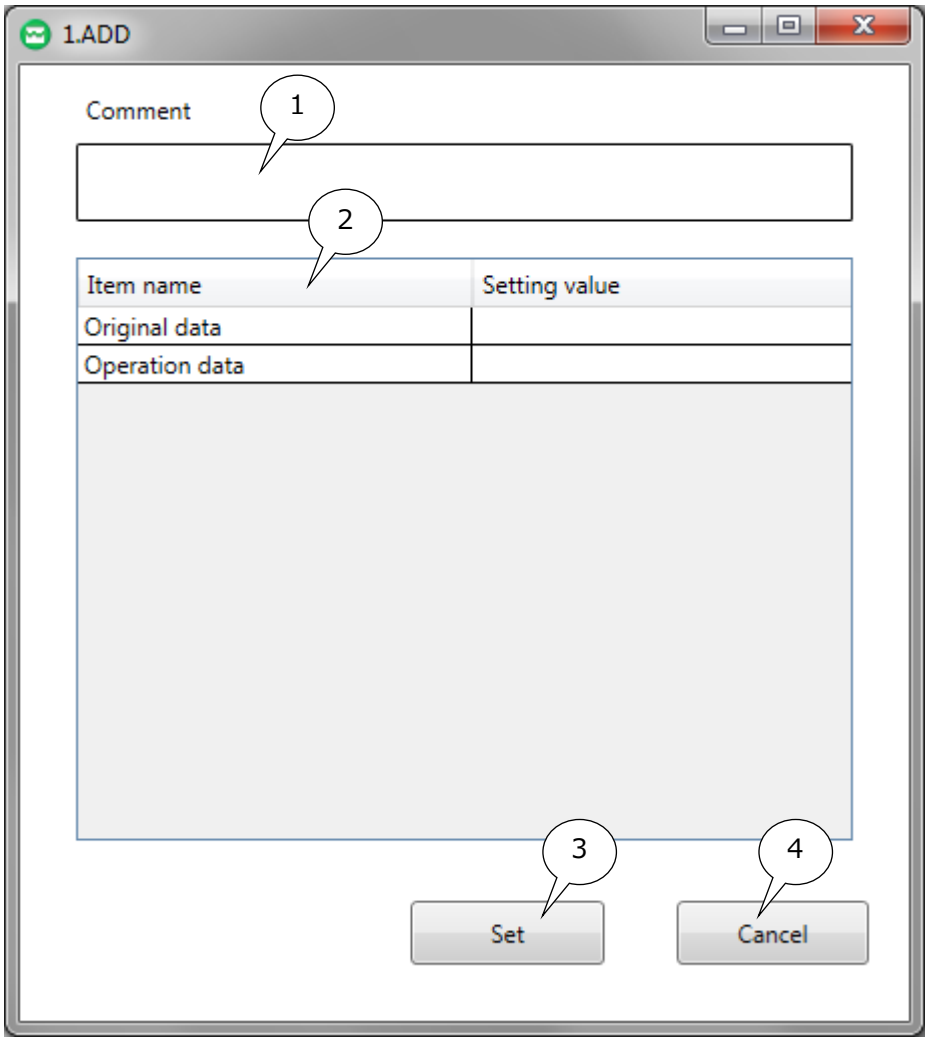


Figure 39

Double-clicking each item in Figure 39 displays the Enter Set Values dialog box.

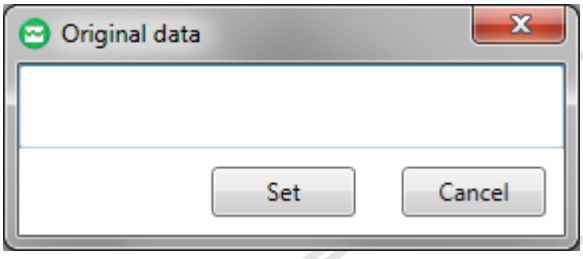


Figure 40

(1) Comment

Comments about the command can be entered.

(2) Item name

The parameters for the selected command are displayed.

(3) Set button

Updates parameter information and closes the Set Parameter window.

(4) Cancel button

Closes the window.

Figure 41 shows an example where multiple commands for which parameters are set are added to the assembly list.

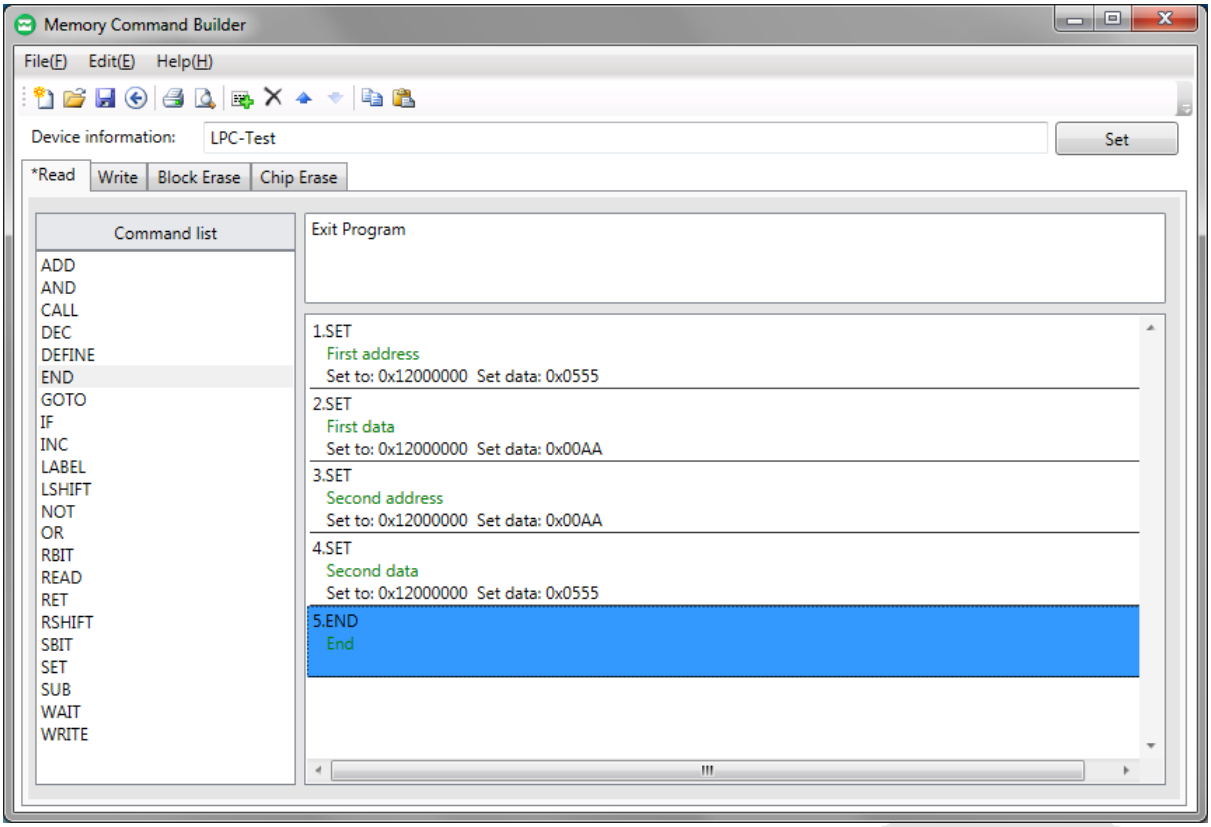


Figure 41

Display format of command in the assembly list

(Line number).command name
  Comment (displayed in green)
  Parameter

Figure 42

3.4. Flash Memory Configuration File Creation Information Window

Select "File" → "Output Flash Memory Configuration File" from the menu bar to display the [Flash Memory Configuration File Creation Information] window. Set (1) through (4) and press Start Creation button to create a flash memory configuration file.

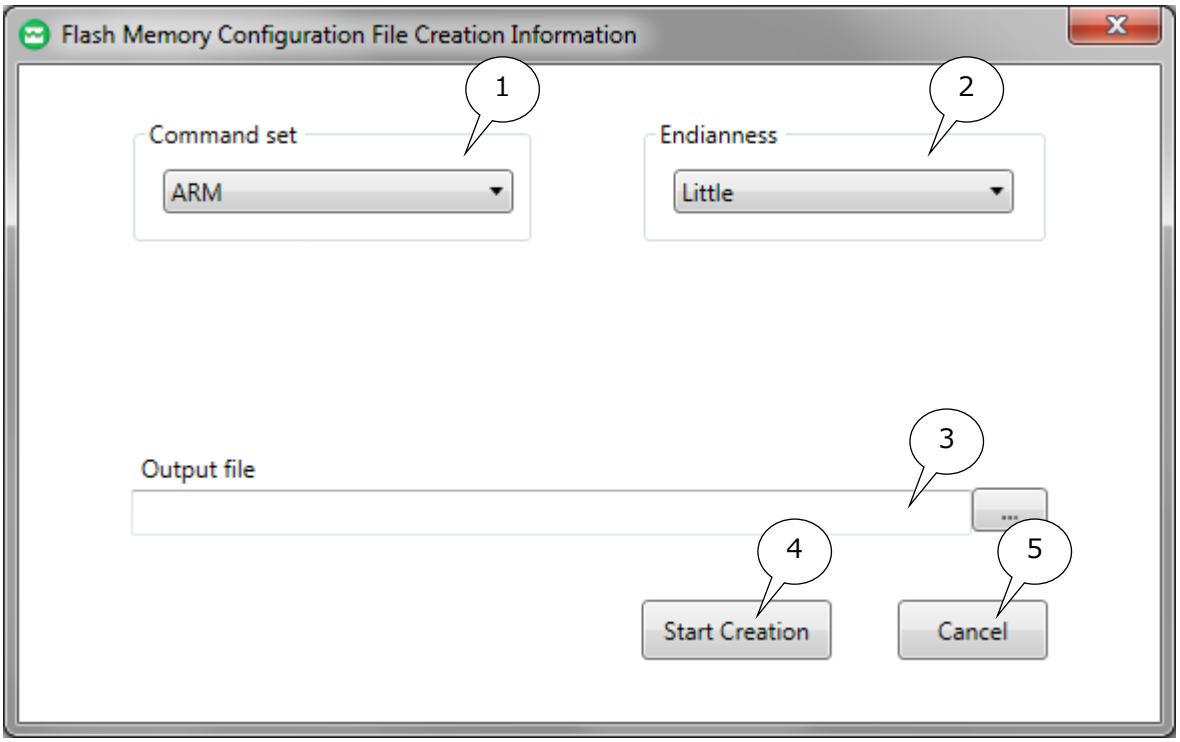


Figure 43

(1) Command set

Select a command set. When SPI is selected as the connection type, this setting is ignored.

Table 11

ARM	Select this when the core is not Cortex-M.
Thumb-2	Select this for the Cortex-M series core.

(3) Endianness

Select endianness. (This supports only little endian.)

Table 12

Little	Creates a flash memory configuration file with little endian.

(4) Output file

Select the flash memory configuration file to be output. The extension is *.fsh.

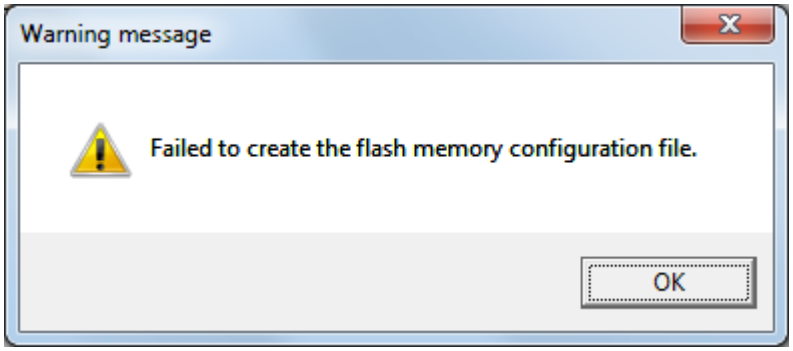
(5) Start Creation button

Creates the flash memory configuration file and closes the window.

(6) Cancel button

Closes the window.

When the following dialogbox is appeared after you click Start Creation button, you probably mistook your command description.



Please check your commands, and retry to create the Flash Memory Configuration file.

4. Saving and Loading a File

4.1. Saving a File

To save a file, select “File” → “Save” or “Save As” from the menu bar or press button on the toolbar.

To save a new file, select “Save As.”

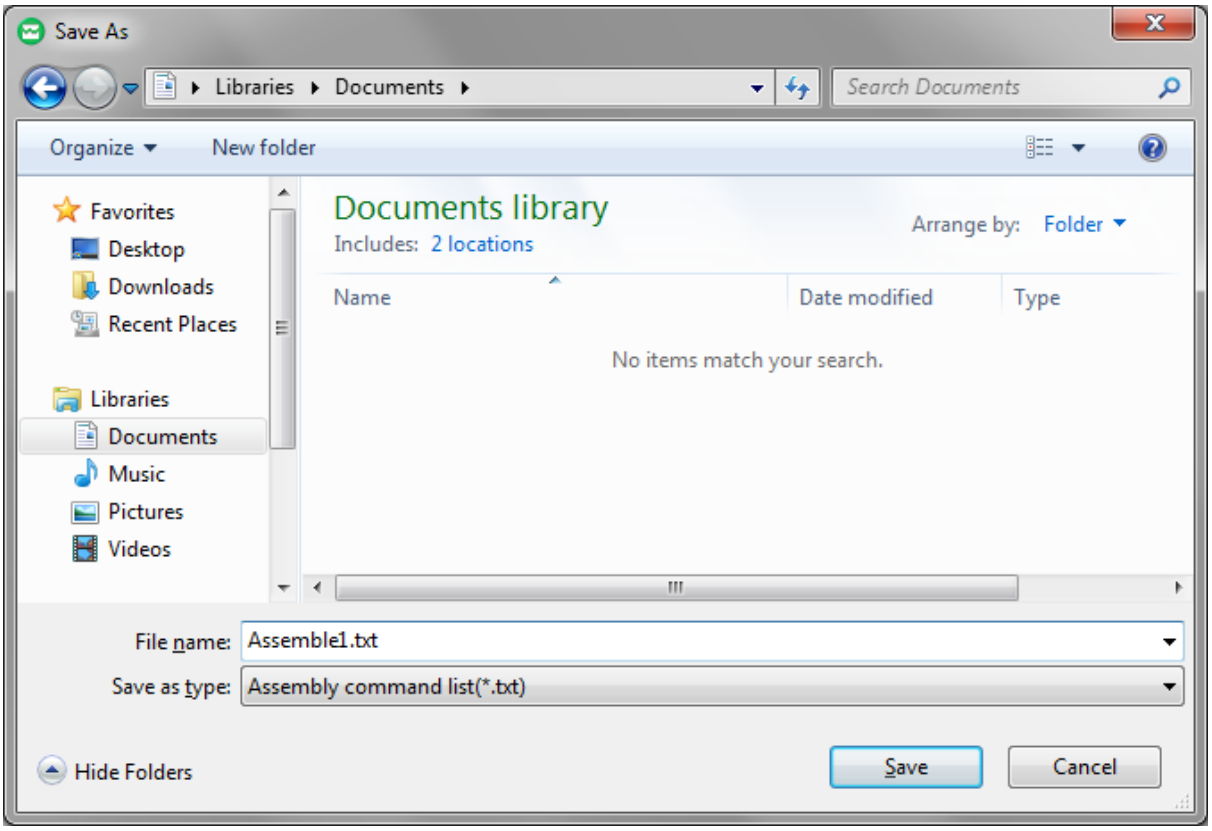


Figure 44

4.2. Loading a File

To load a file, select “File” → “Open” from the menu bar or press on the toolbar.
Select the file to be loaded in the dialog box shown in Figure 45.

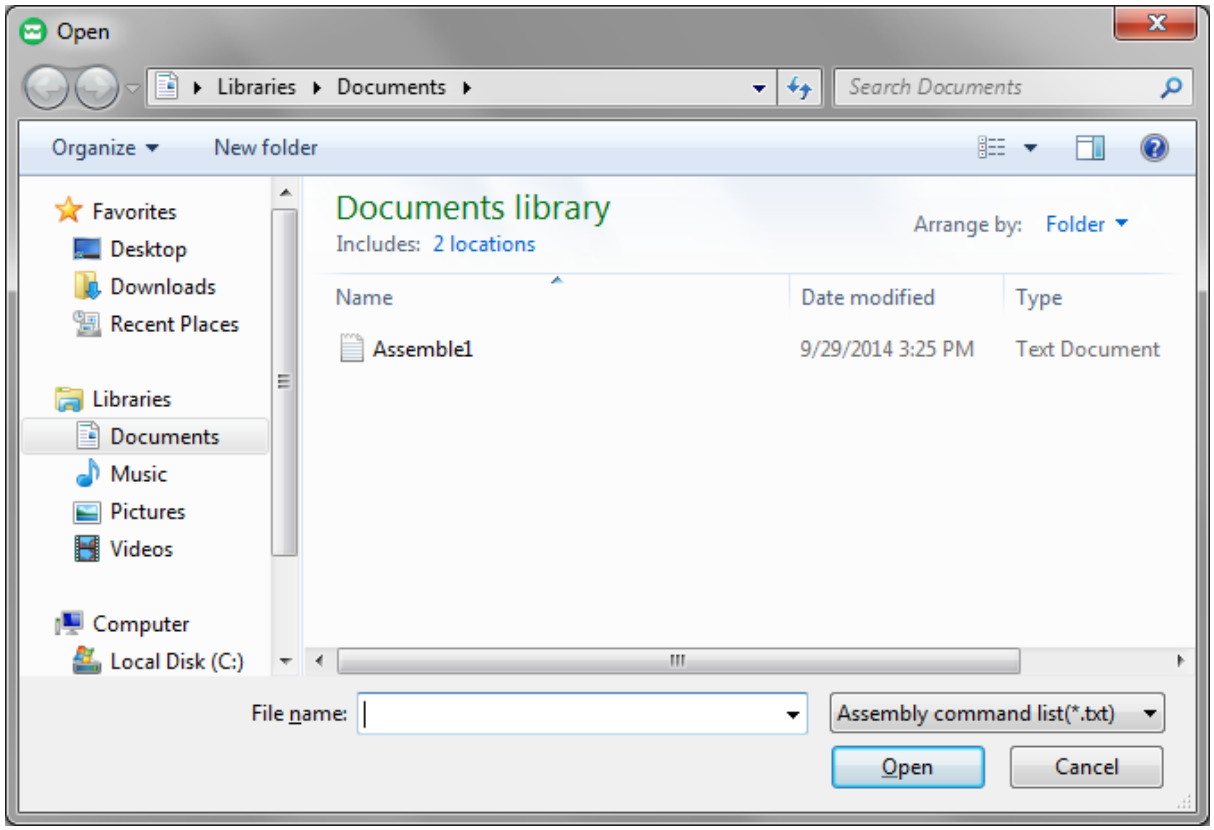


Figure 45

5. Print

5.1. Print Dialog Box

Select “File” → “Print” from the menu bar or press button on the toolbar to display the [Print] dialog box.
Press OK button to start printing.

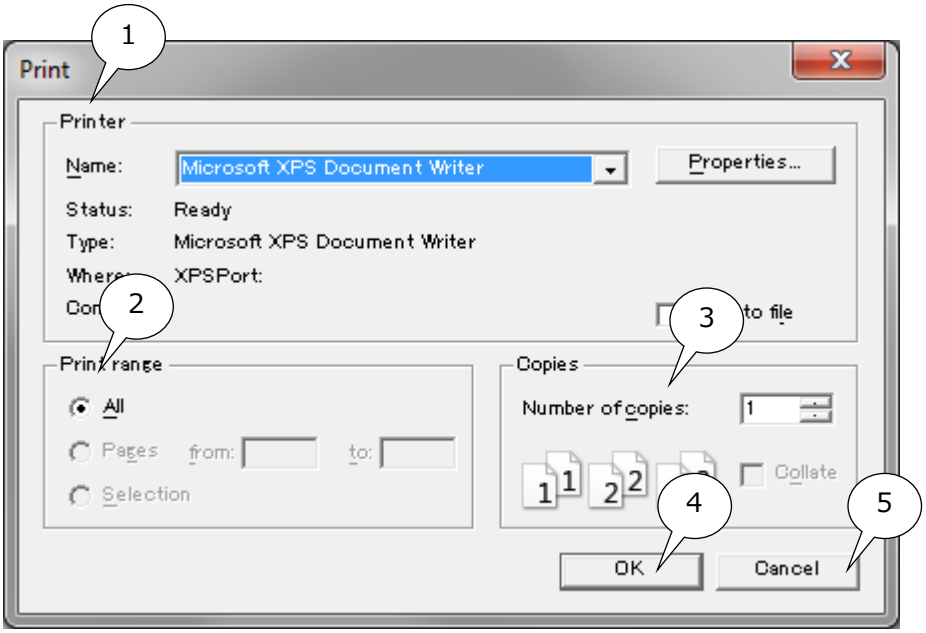


Figure 46

(1) Printer

Set the printer used for printing.

(2) Print range

Set the range to be printed.

(3) Number of copies

Set the number of copies to be printed.

(4) OK button

Starts printing with the set conditions.

(5) Cancel button

Closes the window.

5.2. Page Settings Dialog Box

Select "File" → "Page Settings" from the menu bar to display the [Page Setup] window.

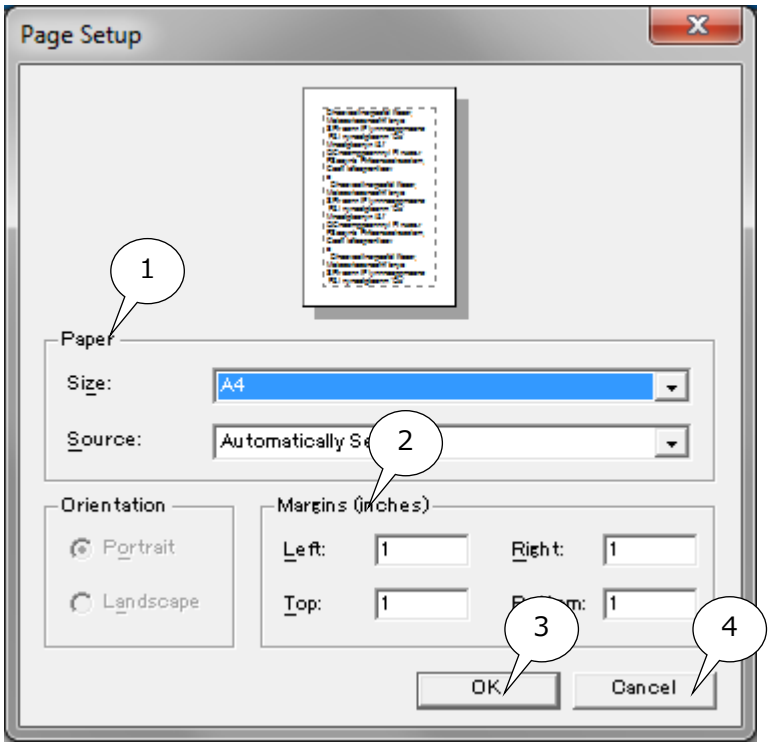


Figure 47

(1) Paper

Set the size of printing paper and the paper feeding method.

(2) Margins

Set the margins of pages to be printed.

(3) OK button

Set according to the conditions displayed on the screen.

(4) Cancel button

Closes the window.

5.3. Print Preview

Select “File” → “Print Preview” from the menu bar or press button on the toolbar to display the [Print Preview] window.
When the print preview screen is generated, the dialog box that shows the number of pages that have been generated (Figure 48) is displayed.
Press Cancel button to abort generation of the preview.

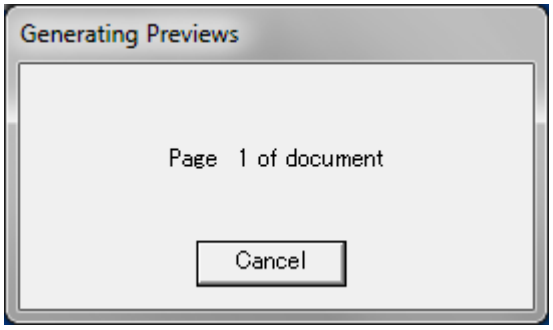


Figure 48

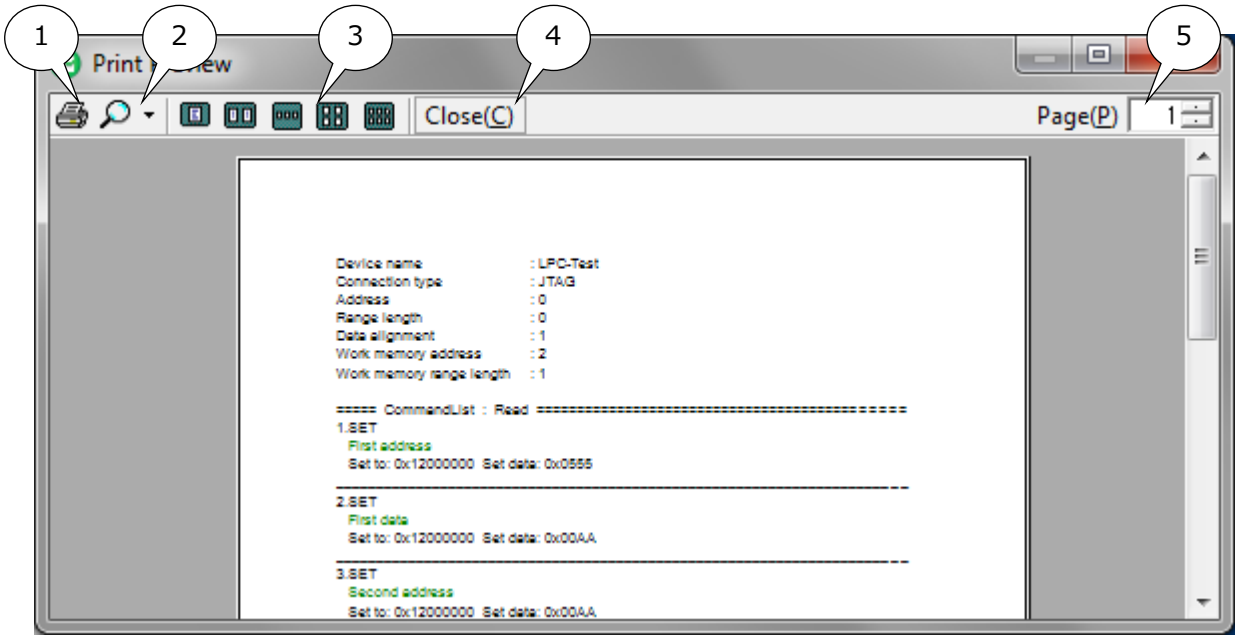


Figure 49

(1) Print button

Starts printing.

(2) Zoom button

Zooms in or out the preview.

(3) Page display button

Changes the number of pages displayed in the window.

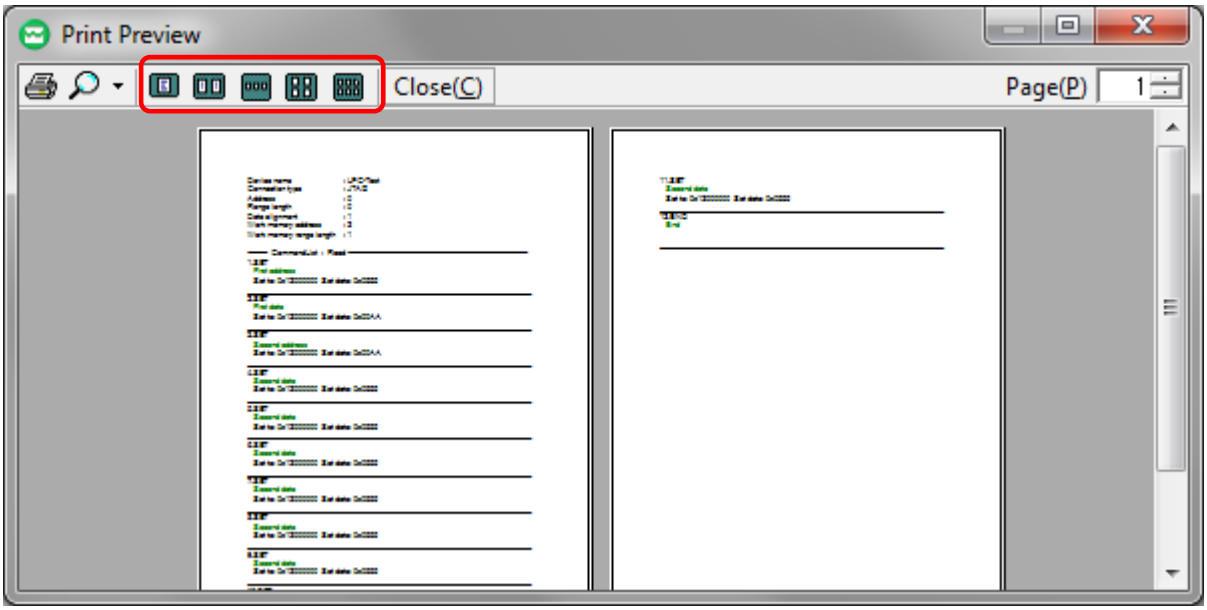


Figure 50

(4) Close button

Closes the window.

(5) Page selection

Moves to the specified page.

6. Command List Table

6.1. JTAG/SWD Connection

Table 13 lists the commands for JTAG/SWD connection.

Table 13

ADD (Addition)	Sets the sum of <original data> and <operation data> to <original data>. Append "A:" at the beginning when specifying an address value. **ADD,original data,operation data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Operation data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification
AND (Logical multiplication)	Sets the logical product of <original data> and <operation data> to <original data>. Append "A:" at the beginning when specifying an address value. **AND,original data,operation data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Operation data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification
CALL (Call)	Calls the specified calling destination. Append "A:" at the beginning when specifying an address value. **CALL,calling destination** Calling destination : Address specification (A:0x00000000) : Label name
DEC (Decrement)	Sets the decremented (1 is subtracted) <original data> to <original data>. Append "A:" at the beginning when specifying an address value. **DEC,original data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9

DEFINE (Define)	Defines string which is exchanged to numeric values in SET command. **DEFINE**,**defined name**,**data1**,**data2**,… defined name : String which exchange to data data : Numerical value : up to six Set multiple data (data1, data2) in the DEFINE, and specify <define name> by the SET command. Then, it will set 'Data1', 'Data2' in sequence to the same address. For example, it is useful to write 0xAA and 0x55 to the specified address sequentially that such as before access to the flash memory or to reflect PLL settings.
END (Exit program)	Exits the program. There are no parameters.
GOTO (Jump)	Jumps to the specified <label name> position. **GOTO**,**label name** label name : The label name of jump destination.
IF (Condition)	Compares <comparison value 1> and <comparison value 2> with the condition indicated by <condition> and jumps to <jump destination> if the result is true and executes the next command if the result is false. Append "A:" at the beginning when specifying an address value. **IF**,**condition**,**comparison value 1**,**comparison value 2**,**jump destination** Condition : Comparison condition "EQ": <comparison value 1> = <comparison value 2> "LT": <comparison value 1> < <comparison value 2> "LE": <comparison value 1> <= <comparison value 2> "GT": <comparison value 1> > <comparison value 2> "GE": <comparison value 1> >= <comparison value 2> "NE": <comparison value 1> not equal to <comparison value 2> comparison value 1,comparison value 2 : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification jump destination : Label name
INC (Increment)	Sets the incremented (1 is added) <original data> to <original data>. Append "A:" at the beginning when specifying an address value. **INC**,**original data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9

LABEL (Label name)	Defines <label name>. **LABEL**,**label name** Label name : Label name
LSHIFT (Shift bits to left)	Shifts <original data> to left by <shift amount> bits and sets the result to <original data>. Append "A:" at the beginning when specifying an address value. **LSHIFT**,**original data**,**shift amount** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Shift amount : Numerical value
NOT (Logical negation)	Sets the result of logically negated <original data> to <original data>. Append "A:" at the beginning when specifying an address value. **NOT**,**original data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9
OR (Logical addition)	Sets the logical sum of <original data> and <operation data> to <original data>. Append "A:" at the beginning when specifying an address value. **OR**,**original data**,**operation data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Operation data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification
RBIT (Clear bits)	Clears (=0) the bit at <bit position> in <original data> and sets the result to <original data>. Append "A:" at the beginning when specifying an address value. **RBIT**,**original data**,**bit position** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Bit position : Numerical value

READ (Reading data from the flash memory)	Reads the data in the parameter storage area in the work memory to read data from the flash memory. The arguments in the program are used. **READ,transfer destination data,transfer source data,data length** Transfer destination/source data: Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification Data length : Register specification (REG*) *=0 to 9 : Numerical value specification
RET (Back)	Goes back to the calling source. There are no parameters.
RSHIFT (Shift bits to right)	Shifts <original data> to right by <shift amount> bits and sets the result to <original data>. Append "A:" at the beginning when specifying an address value. **RSHIFT,original data,shift amount** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Shift amount : Numerical value
SBIT (Bit setting)	Sets the bit at <bit position> in <original data> and sets the result to <original data>. Append "A:" at the beginning when specifying an address value. **SBIT,original data,bit position** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Bit position : Numerical value

SET (Set data)	Sets <set data> to <set destination>. Append "A:" at the beginning when specifying an address value. You can also append suffixes to specify the data type. **SET,set destination,set data** Set destination : Immediate address specification (A:0x00000000) : Register address specification (A:REG*) *=0 to 9 : Register specification (REG*) *=0 to 9 Data type suffix : 8 bit none(A:0x00000000 Immedate address 8 bit) : 16 bit S(A:REG0S Register address 16 bit) : 32 bit W(A:0x00000000W Immedate address 32 bit) Set data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification : Defined name specification (This name is defined by the DEFINE command.) Examples: SET A:0x00000000,0x55 Immediate address 8 bit access SET A:REG0W,0xAAAA Register address 32 bit access SET REG0,REG1 Sets register value to register. SET REG2,FLASH_ADDRESS Sets value to register with define name.
SUB (Subtraction)	Sets the difference of <original data> and <operation data> to <original data>. Append "A:" at the beginning when specifying an address value. **SUB,original data,operation data** Original data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 Operation data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification

WAIT (Wait)	Compares <comparison value 1> and <comparison value 2> with the condition indicated by <condition> and waits until the result becomes true. Append "A:" at the beginning when specifying an address value. **WAIT,condition,comparison value 1,comparison value 2** Condition : Comparison condition "EQ": <comparison value 1> = <comparison value 2> "LT": <comparison value 1> < <comparison value 2> "LE": <comparison value 1> <= <comparison value 2> "GT": <comparison value 1> > <comparison value 2> "GE": <comparison value 1> >= <comparison value 2> "NE": <comparison value 1> not equal to <comparison value 2> Comparison value 1/comparison value 2 : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification
WRITE (Writing data to the flash memory)	Writes the data in the parameter storage area in the work memory to write data to the flash memory. The arguments in the program are used. **WRITE,transfer destination data,transfer source data,data length** Transfer destination/source data : Address specification (A:0x00000000) : Register specification (REG*) *=0 to 9 : Numerical value specification Data length : Register specification (REG*) *=0 to 9 : Numerical value specification

From REG0 to REG9 are not assigned CPU registers.
From REG0 to REG2 are the 32-bit variable, and initialized by followings.

REG0 : Target address. (Used in Read, Wrie, Block Erase or Chip Erase)
REG1 : Start address of data area. (Used in Read or Write)
REG2 : Date bytes. (Used in Read or Wrie)

REG3 to REG9 are the 32-bit variable, and these are not initialized.

6.2. SPI Connection

Table 14 lists the commands for SPI connection.

Table 14

CS signal set	Sets the CS signal to the status in <data>. Data : Numerical value={0(=Low), 1(=High)}
Setting command	Sets <command> with the size specified in <bit size>. Bit size : Numerical value Command : Numerical value
Setting address	Sets <address> with the size specified in <bit size>. Bit size : Numerical value
Writing data	Writes data of the specified size. This is available in “Write” tab. Output data size : Numerical value
Reading data	Reads data of the specified size. This is available in “Read” tab. Input data size : Numerical value
Reading status	Reads status data of the specified size. This is available in “Status Read” tab. Bit size : Numerical value
Waiting for status	Repeatedly executes the status reading script. Specified bit : Numerical value Set value : Numerical value Set the wait time use : Numerical value Waiting time (in milliseconds) : Numerical value. Max. 100000 (100 sec)
Waiting for time (in microseconds)	Waits for the specified time. Waiting time : Numerical value. Max. 10000000 (10 sec).
Waiting for time (in milliseconds)	Waits for the specified time. Waiting time : Numerical value. Max. 10000 (10 sec).

Revision History

Ver. No.	Revision date	Contents of revision
01	09/29/2014	Initial Release.
02	10/23/2014	Modified the followings “1.2. System Requirements”. - Fixed the description about “PC” and “Memory” - Added the description about “HDD” and “OS” - Added ARM Writer and SPI Writer to the software which is required to use the flash memory configuration file. Added the explanation about the warning message which is appeared after you create Flash Memory Configuation file in “3.4. Flash Memory Configuration File Creation Information Window”. Modified the followings in “6.1. JTAG Connectoin” . - Added the explanation about the DEFINE command. - Added the description about the register address specification, the data type suffix, and its examples in the SET command. - Removed “Supplementary explanation about the address specification method”.
03	11/17/2014	Correction of typographical errors.

Manufacturer Information

Sohwa & Sophia Technologies Inc.

[Headquarters]

6-2, Minami-kurokawa, Asao-ku, Kawasaki City, Kanagawa Pref.,
215-8588, JAPAN
Web: http://www.ss-technologies.co.jp/en/index.html

Subsidiary

SOHWA MALAYSIA SND. BHD.

Unit 5-2, Level 5, Tower 6, Avenue 5,
The Horizon, Bangsar South No.8, Jalan Kerinchi
59200, Kuala Lumpur, Malaysia

HomePage : http://www.sohwa-m.com.my/