

S1C31 Family Application Note
S1C31 Family
EEPROM Emulation Library
Manual

arm



Evaluation board/kit and Development tool important notice

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

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. When exporting the products or technology described in this material, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You are requested not to use, to resell, to export and/or to otherwise dispose of the products (and any technical information furnished, if any) for the development and/or manufacture of weapon of mass destruction or for other military purposes.

Arm, Cortex, Keil and μ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. IAR Systems, IAR Embedded Workbench, C-SPY, I-jet, IAR and the logotype of IAR Systems are trademarks or registered trademarks owned by IAR Systems AB. SEGGER and J-Link are trademarks or registered trademarks of SEGGER Microcontroller GmbH & Co. KG. All rights reserved. All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

“Reproduced with permission from Arm Limited. Copyright © Arm Limited”

©SEIKO EPSON CORPORATION 2020, All rights reserved.



Table of Contents

1. Overview	1
1.1 Features	1
1.2 Working Environment	2
1.3 Precautions for Usage	2
2. Library configuration	3
2.1 Folder Configuration	3
2.2 Library function	4
3. How to Use Library	5
3.1 Settings for Using Library in Application Program	5
3.2 Internal RAM and Flash Memory Usage	6
3.3 Write Time	6
3.4 Precautions on Use of Library	6
3.5 Sample Program	7
4. Library Specifications	8
4.1 EEPROM Read/Write Function Details	8
4.2 Error Code Definitions	9
Appendix	10
A. How to Incorporate Library into Project (IAR EAWRM)	10
B. How to Incorporate Library into Project (MDK-ARM)	13
Revision History	16





1. Overview

The S1C31 EEPROM emulation library includes a library to emulate an EEPROM using the flash memory embedded in the target model. Application programs with this library linked can use the embedded flash memory as an EEPROM by calling the library functions.

This library and sample software are included in the S1C31xxx peripheral circuit sample software package. The S1C31xxx peripheral circuit sample software package is available on Seiko Epson's website.

In addition to this manual, please also refer to the "S1C31xxx Technical Manual".

1.1 Features

This library implements the emulation function allowing the S1C31 Family microcontrollers to use the embedded flash memory as an EEPROM. It uses a flash memory area of 4K to 64K bytes as an EEPROM emulation area to emulate a 32- to 512-byte EEPROM. The number of rewriting guarantee times of the embedded flash memory is about 1,000 times. The emulation function allocates one flash memory sector for one EEPROM address, this makes it possible to increase the number of rewriting times of each emulated EEPROM address to 100,000 times or more theoretically.

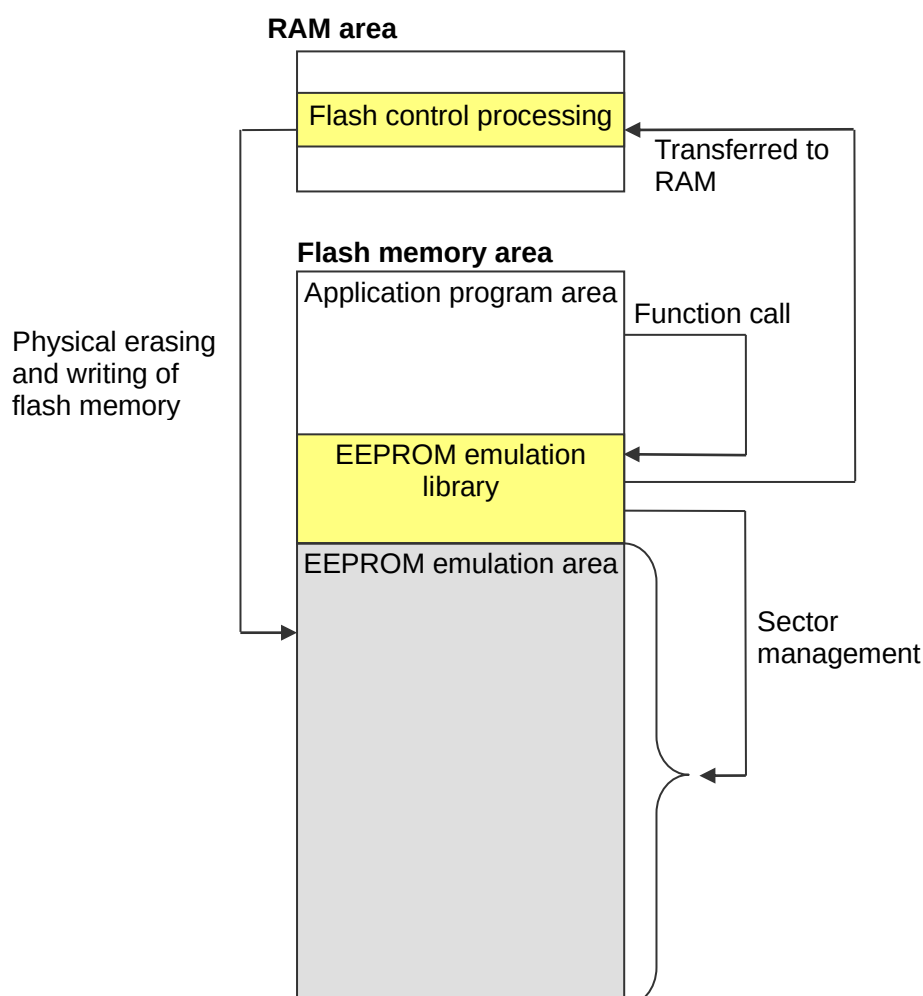


Figure 1.1.1 EEPROM emulation library features

1. Overview

1.2 Working Environment

The following is required when writing and debugging the sample software.

- Evaluation Board
 - S5U1C31xxxTx evaluation board with S1C31 series.
- Debug Probes *1*2
 - IAR Systems I-jet or SEGGER J-Link
- Integrated Development Environment
 - IAR Embedded Workbench for ARM® (IAR EWARM) or MDK-ARM® (uVision)
- S1C31SetupTool package
 - Includes Flash loader and Configuration files (.svd etc).
- S1C31xxx Peripheral circuit sample software package

*1: Debug probes are not required for library function calls from the sample software.

*2: I-jet is available only with IAR EWARM. J-Link is available for both IAR EWARM and MDK-ARM.

For details on the above, refer to the attached manual.

1.3 Precautions for Usage

The S1C31 EEPROM emulation library and sample software are for reference only. Our company will not take any responsibility for any problems caused by this library. Please thoroughly verify the operation when using this library for your product.

This manual is common to the EEPROM emulation library provided for each model of the S1C31 series. About the specifications (RAM usage, etc.) that differ depending on the model, refer to the readme included in the S1C31xxx peripheral circuit sample software package.



2. Library configuration

2.1 Folder Configuration

The configuration of the S1C31 EEPROM emulation library, sample software, and related programs included in the S1C31xxx peripheral circuit sample software package is as follows.

```

S1C31xxxSamplePKG_ver_yy.zip
[S1C31xxxSamplePKG_ver_yy]
|- [Licenses]
|- [Drivers] : Drivers
|   |- [board] : Evaluation board related driver
|   |- [CMSIS] : CMSIS driver
|       |- [Device]
|           |- [S1C31xxx]
|               |- [Include]
|                   |- S1C31xxx.h : CMSIS peripheral circuit access layer header file
|                   |- ...
|               |- [Source]
|                   |- [ARM]
|                   |- [IAR]
|                       |- startup_S1C31xxx.s : CMSIS sartup program
|                       |- system_S1C31xxx.c : CMSIS peripheral circuit access layer header program
|           |- [Driver]
|               |- [Include]
|                   |- Driver_Common.h : Common driver definition
|                   |- Driver_EEPROM.h : CMSIS EEPROM emulation library driver definition
|                   |- ...
|               |- [Source]
|                   |- Driver_EEPROM.c : CMSIS EEPROM emulation library driver
|           |- [SVD]
|- [sePeripheralLibrary] : Peripheral circuit library

|- [Middlewares] : Middlewares
|   |- [seEepromLibrary] : EEPROM emulation library
|       |- [Device]
|           |- [S1C31xxx]
|               |- seEepromLibraryS1C31xxx.a : Library for IAR EAWRM
|               |- seEepromLibraryS1C31xxx.lib : Library for MDK-ARM
|       |- [Include]
|           |- DataFlashCommand.h : EEPROM emulation library driver definition
|           |- DataFlashConfig.h : Setting header file
|       |- eeepromLibraryForS1c31xxx_readme_e.txt : readme
|       |- eeepromLibraryForS1c31xxx_readme_j.txt
|   |- ...
|- [Projects] : Sample softwares
|   |- [Applications] : Various application software
|       |- [EEPROM] : Sample software for EEPROM emulation library
|           |- [ARM] : Project for MDK-ARM
|           |- [IAR] : Project for IAR EWARM
|           |- main.c
|       |- ...
|   |- ...

README_e.txt
README_j.txt

```

Figure 2.1.1 S1C31xxx Sample software package configuration

2. Library configuration

2.2 Library function

The functions provided by this library are defined in Drivers\CMSIS\Driver\Include\Driver_EEPROM.h. The functions provided by this library are as follows.

Table 2.2.1 Functions provided by the S1C31 EEPROM emulation library Function name	Functional overview
int32_t ProgramData (uint32_t addr, unsigned char *data, uint32_t cnt)	Write internal flash memory
int32_t ReadData (uint32_t addr, unsigned char *data, int32_t cnt)	Read internal flash memory
ARM_DRIVER_VERSION GetVersion (void)	Get this library version
ARM_FLASH_INFO * GetInfo (void)	Get information on built-in flash memory



3. How to Use Library

Describes how to use the S1C31 EEPROM emulation library and sample software.

3.1 Settings for Using Library in Application Program

This section describes how to use this library on the application program. For how to incorporate the library into the project of an application program, refer to “Appendix x. How to Incorporate Library into Project.”

1. Declaration of Header File

Include “Driver_EEPROM.h” in the source file that uses this library.

```
#include <stdio.h>
#include <string.h>
#include "Driver_EEPROM.h"
```

2. Setting EEPROM size and number of write retries

Set the following values in "Driver_EEPROM.h".

```
#define CONFIG_EEPROM_SIZE_MAX          (512)
#define CONFIG_RETRY_COUNT              (4)
```

- CONFIG_EEPROM_SIZE_MAX

Sets the size of the EEPROM to emulate. Please refer to the readme for the size that can be set by the user.

- CONFIG_RETRY_COUNT

Sets the number of write retries when a writing has failed. Increasing the number of write retries causes the processing time of the writing routine to increase and performance to decrease. So it should only be set to several times.

3. Add function

Add the functions provided by the library to the source file that uses this library. About the function specifications, refer to "Chapter 4 Library Specifications".

extern ARM_DRIVER_FLASH Driver_Eeprom;

...

```
int main(void) {
    unsigned char compbuf[CONFIG_EEPROM_SIZE_MAX];
    ARM_FLASH_INFO *Info = Driver_Eeprom.GetInfo();
    Driver_Eeprom.GetVersion();

    if (Driver_Eeprom.ProgramData(0, updateLineBit, CONFIG_EEPROM_SIZE_MAX) == ARM_DRIVER_OK)
    {
        printf("Program: OK\n");
        Driver_Eeprom.ReadData(0, compbuf, CONFIG_EEPROM_SIZE_MAX);
        if (memcmp(updateLineBit, compbuf, CONFIG_EEPROM_SIZE_MAX) == 0) {
            printf("Verify: OK\n");
        } else {
            printf("Verify: NG\n");
        }
    }
    } else {
```

Write EEPROM

Read EEPROM



3. How to Use Library

```
    printf("Program: NG\n");  
}  
...
```

3.2 Internal RAM and Flash Memory Usage

This library uses an internal RAM area and a flash memory area. About the RAM usage of the EEPROM emulation library of each model, refer to the readme included in the S1C31xxx peripheral circuit sample software package.

3.3 Write Time

The write time using this library depends on the EEPROM size configured, number of flash memory rewriting times, the clock source to be used, and other conditions.

For reference, the following shows a write time example when data is written to the same EEPROM address 100,000 times using the EEPROM emulation library of S1C31W74:

Typ. value	4.6 ms
Max.value	27.4 ms

Check the actual writing time with the application program that uses this library.

3.4 Precautions on Use of Library

When using this library, be sure to note the following points:

- The S1C31D01 / S1C31D5x / S1C31W74 uses ch.0 of the 16-bit timer (T16). Therefore, the contents of the register of 16-bit timer ch.0 are changed. Please note when using this library with an application program that uses a 16-bit timer.
- In this library, the system clock is changed to the high-speed clock (OSC3 or IOSC). Therefore, please note that the contents of the control register of the clock generator (CLG) will be changed.
- The EEPROM emulation area in the flash memory must be erased before this library can be used and when the EEPROM area location and/or size are changed.
- The EEPROM emulation area occupies a flash memory space of $\text{<CONFIG_EEPROM_SIZE_MAX * DATAFLASH_SECTOR_SIZE>}$ bytes. CONFIG_EEPROM_SIZE_MAX should be set so that the EEPROM emulation area will not exceed the flash memory capacity.
- When using this library, connect a capacitor to the Vpp pin as shown in the basic external connection diagram in the “S1C31xxx Technical Manual”, and disconnect the connection between the Vpp pin and other pin.
- Be aware of the number of flash memory rewriting guarantee times when using this library. For the flash memory specifications, refer to the “S1C17(xxx) Technical Manual”.
- While executing the write function, supply the VDD operating voltage for Flash programming (When VPP is generated internally) specified in the data sheet of each model. If the voltage falls outside the range, the written value is not guaranteed.



3.5 Sample Program

1. Sample program specification

The sample software uses the S1C31 EEPROM emulation library to write the increment data from address 0 to (CONFIG_EEPROM_SIZE_MAX -1).

2. Preparation

About details on how to execute this sample software project in an integrated development environment, refer to the "S1C31xxx Peripheral Circuit Sample Software Manual".

3. Operation overview

- (1) Get information on the built-in flash memory. (optional)
- (2) Get the version of the S1C31 EEPROM emulation library. (optional)
- (3) Write updateLineBit [] (512 bytes) of update data to EEPROM.
- (4) Read 512 bytes of data from the EEPROM.
- (5) Compares cmpbuf [] of the read data with updateLineBit [] of the update data and displays the result. (Verify)



4. Library Specifications

4. Library Specifications

4.1 EEPROM Read/Write Function Details

Details of the functions provided by the S1C31 EEPROM emulation library are described below.

Function name	int32_t ProgramData (uint32_t addr, unsigned char *data, uint32_t cnt)	
Arguments	uint32_t addr	Write address. Scope: 0 - (CONFIG_EEPROM_SIZE_MAX - 1)
	const void * data	Write data. Indicates a pointer to the write data. The pointer must point to the RAM area.
	uint32_t cnt	Write data size.
Return value	int	Write result. (error code)
Description	Write EEPROM. (1) Check if the argument is within the EEPROM address. (2) Writes the write data to the specified write address. (3) Check if the write address is write data. (Verify) (4) Returns the writing result.	
Remarks	None.	

Function name	int32_t ReadData (uint32_t addr, unsigned char *data, int32_t cnt)	
Arguments	uint32_t addr	Read address. Scope: 0 - (CONFIG_EEPROM_SIZE_MAX - 1)
	const void * data	Read data. Indicates a pointer to the write data. The pointer must point to the RAM area.
	uint32_t cnt	Read data size.
Return value	uint32_t	ARM_DRIVER_OK (0)
Description	Read EEPROM. (1) Check if the argument is within the EEPROM address. (2) Reads to the specified read address. (3) Returns the reading result.	
Remarks	None.	

Function name	ARM_DRIVER_VERSION GetVersion (void)	
Return value	ARM_DRIVER_VERSION	This library version.
Description	Get this library version.	
Remarks	None.	

Function name	ARM_FLASH_INFO * GetInfo (void)	
Return value	ARM_FLASH_INFO *	Built-in flash memory information.
Description	The following is acquired as the information of the built-in flash memory. • Number of sectors • Size of EEPROM	
Remarks	None.	



4.2 Error Code Definitions

The error code used in the return value of each function is as follows.

Table 4.2.1 Error Codes

Definition Name	Value	Description
ARM_DRIVER_OK	0	Successfully completed.
ARM_DRIVER_ERROR_TIMEOUT	-3	Timeout / Verify error
ARM_DRIVER_ERROR_UNSUPPORTED	-4	Unsupported operation
ARM_DRIVER_ERROR_PARAMETER	-5	Argument error

These definitions are defined in Drivers\CMSIS\Driver\Include\Driver_Common.h.



Appendix

A. How to Incorporate Library into Project (IAR EWARM)

The following describes how to incorporate the S1C31 EEPROM emulation library into the project of application program created by IAR EWARM. About IAR EWARM, please refer to the attached manual.

1. Add Library

- (1) Select [Project]> [Options] from the IAR EWARM menu.
- (2) Select [Linker] from the [Category] list in the displayed dialog.
- (3) From the [Library] tab, add this library included in the S1C31xxx peripheral circuit sample software package to "Additional libraries".

Middlewares\seEepromLibrary\Device\S1C31xxx\seEepromLibraryS1C31xxx.a

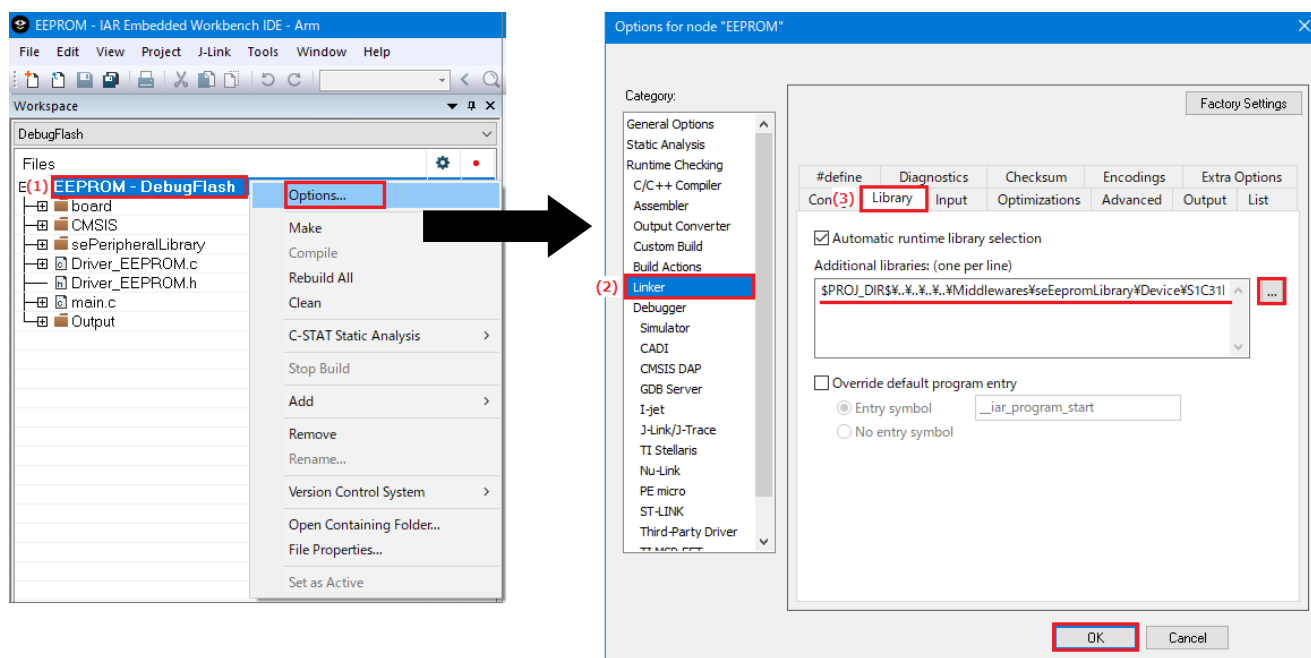


Figure A.1 Add library

- (4) Select [Project]> [Add]> [Add File] from the IAR EWARM menu.
- (5) Add the following driver and driver definition.
 - Drivers\CMSIS\Driver\Source\Driver_EEPROM.c
 - Drivers\CMSIS\Driver\Include\Driver_EEPROM.h

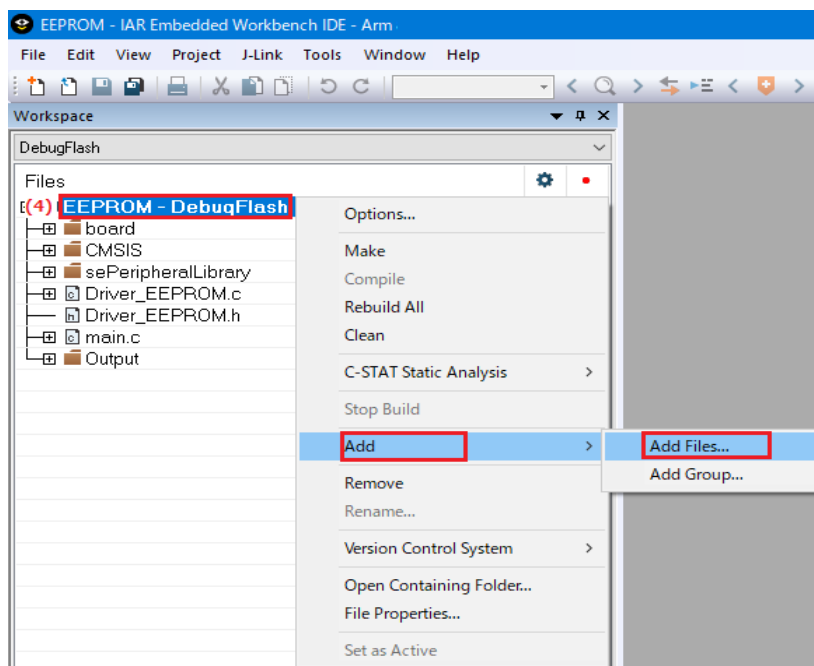


Figure A.2 Add CMSIS driver

2. Include Path Settings

- (1) Select [Project] > [Options] from the IAR EWARM menu.
- (2) Select [C/C++ Compiler] from the [Category] list in the displayed dialog.
- (3) From the [Preprocessor] tab, add the following include path of the driver definition included in the S1C31xxx peripheral circuit sample software package to the "Additional include directories".
 - Drivers\CMSIS\Driver\Include
 - Middlewares\seEepromLibrary\Include

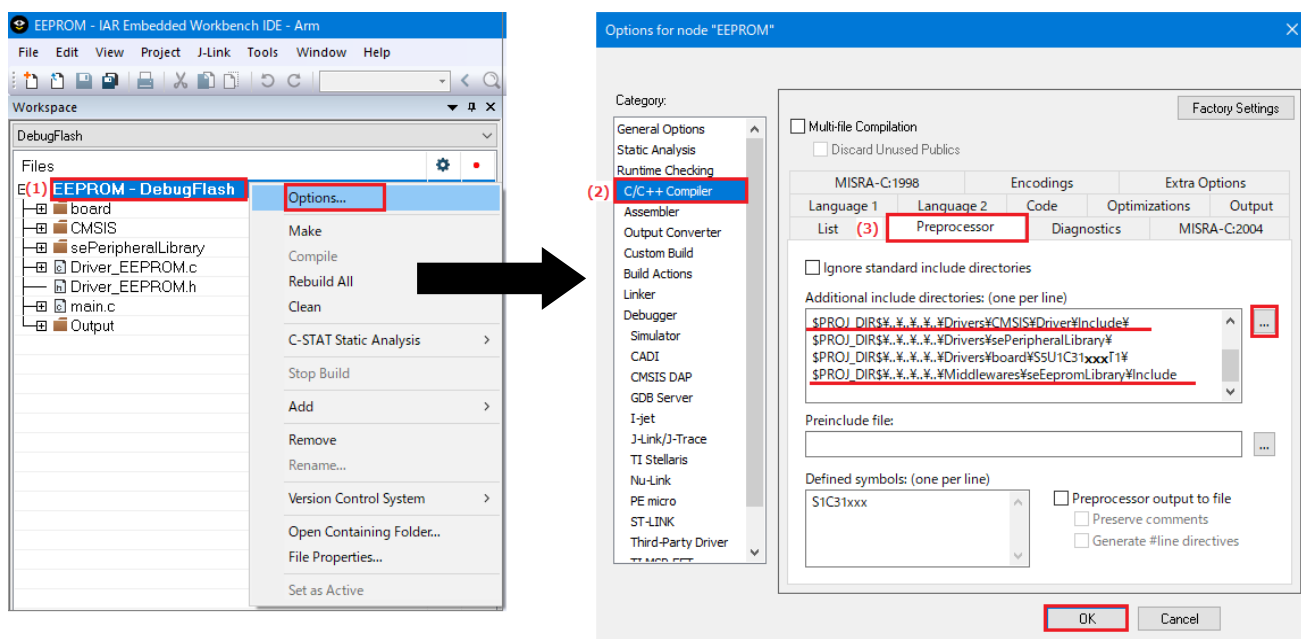


Figure A.3 Set include path

3. Linker Script Settings

- (1) Edit the linker script file (.icf) included in the project.
- (2) Refer to the linker script file (S1C31xxx_flash.icf) of the sample software included in the S1C31xxx peripheral circuit sample software package, and add the following sections.

...

```
initialize by copy { readwrite };
```

```
initialize manually with packing = none { section .flash_common_text};
```

```
//initialize by copy with packing = none { section __DLIB_PERTHREAD }; // Required in a multi-threaded application
```

```
do not initialize { section .noinit };
```

```
place at address mem: __ICFEDIT_intvec_start__ { readonly section .intvec };
```

```
place in ROM_region { readonly };
```

```
place in RAM_region { readwrite, block CSTACK, block HEA
```

```
place in ROM_region { section .flash_common_text_init};
```

```
place in RAM_region { section .flash_common_text};
```

```
/"ROM2":place in ROM2_region { section EEPROM1 };
```

```
place at address mem: 0x20000 { readonly section EEPROM1 };
```

Creating a flash common text section

Specifying the copy source section of the ROM area

Specifying the copy destination section of the RAM area

Specify the start address of the internal flash memory used as the EEPROM emulation area

When the above is added, the code of this library is placed in the RAM area in this library. Also, specify the internal flash memory to be used as the EEPROM emulation area. Specify the start address in the unit described in the readme.

- (3) Select [Project] > [Options] from the IAR EWARM menu.
- (4) Select [Linker] from the [Category] list in the displayed dialog.
- (5) From the [Config] tab, check "Default Override" and specify the edited linker script file.

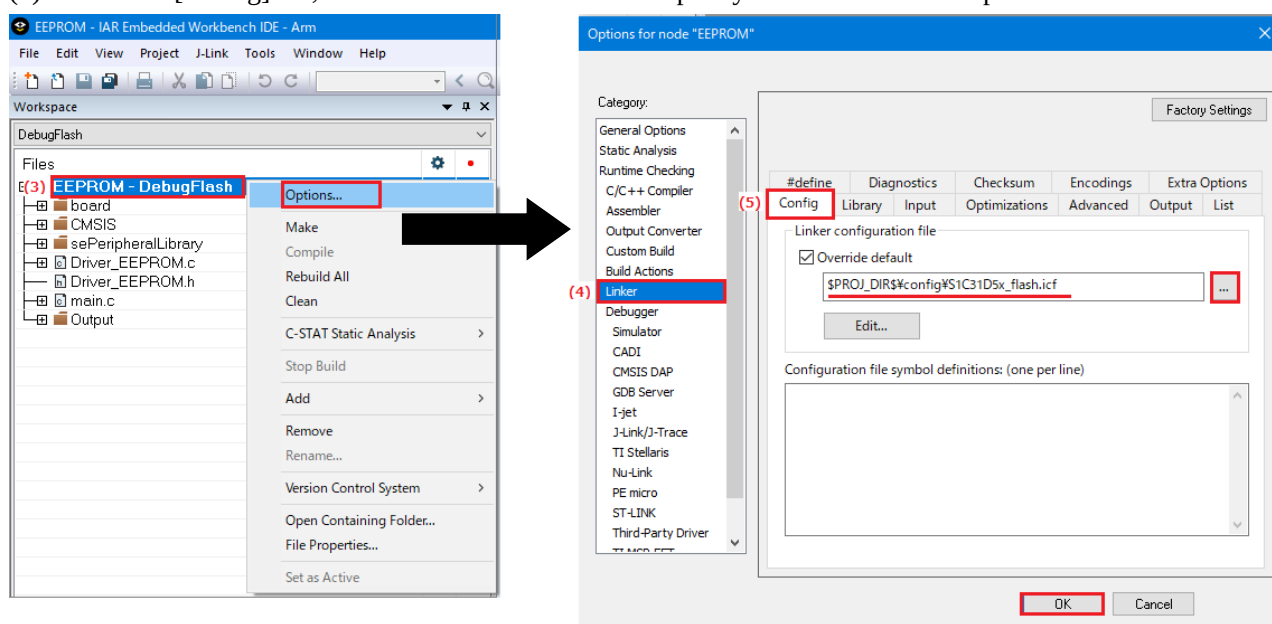


Figure A.4 Set linker script file

B. How to Incorporate Library into Project (MDK-ARM)

The following describes how to incorporate the S1C31 EEPROM emulation library into the project of application program created by MDK-ARM (uVision). About MDK-ARM, please refer to the attached manual.

1. Add Library

- (1) Right-click the target source folder in the uVision [Project] window and select [Add Existing Files to Group 'xxx'...].
- (2) From the displayed dialog, add this library and driver included in the S1C31xxx peripheral circuit sample software package below.
 - Middlewares\seEepromLibrary\Device\S1C31xxx\seEepromLibraryS1C31xxx.lib
 - Drivers\CMSIS\Driver\Source\Driver_EEPROM.c

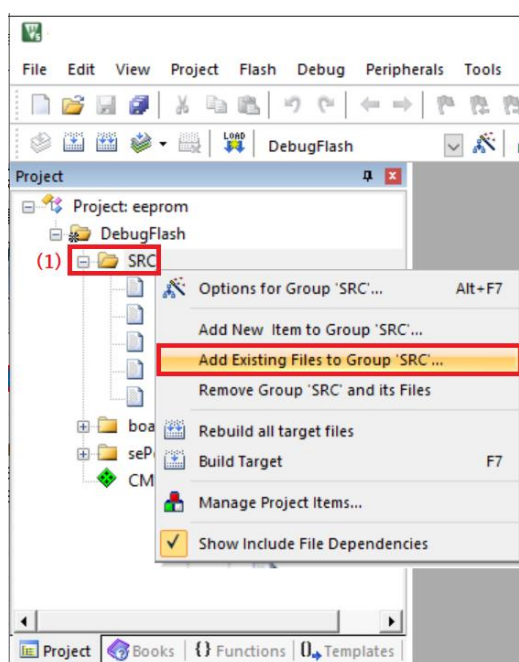


Figure B.1 Add library

2. Include Path Settings

- (1) Select [Project]> [Options for Target 'xxx'...] from the uVision menu.
- (2) Browse to the folder from [C / C ++]> 'Include Paths' in the displayed dialog.
- (3) From [New (Insert)], add the following include path of the driver definition included in the S1C31xxx peripheral circuit sample software package.
 - Drivers\CMSIS\Driver\Include
 - Middlewares\seEepromLibrary\Include

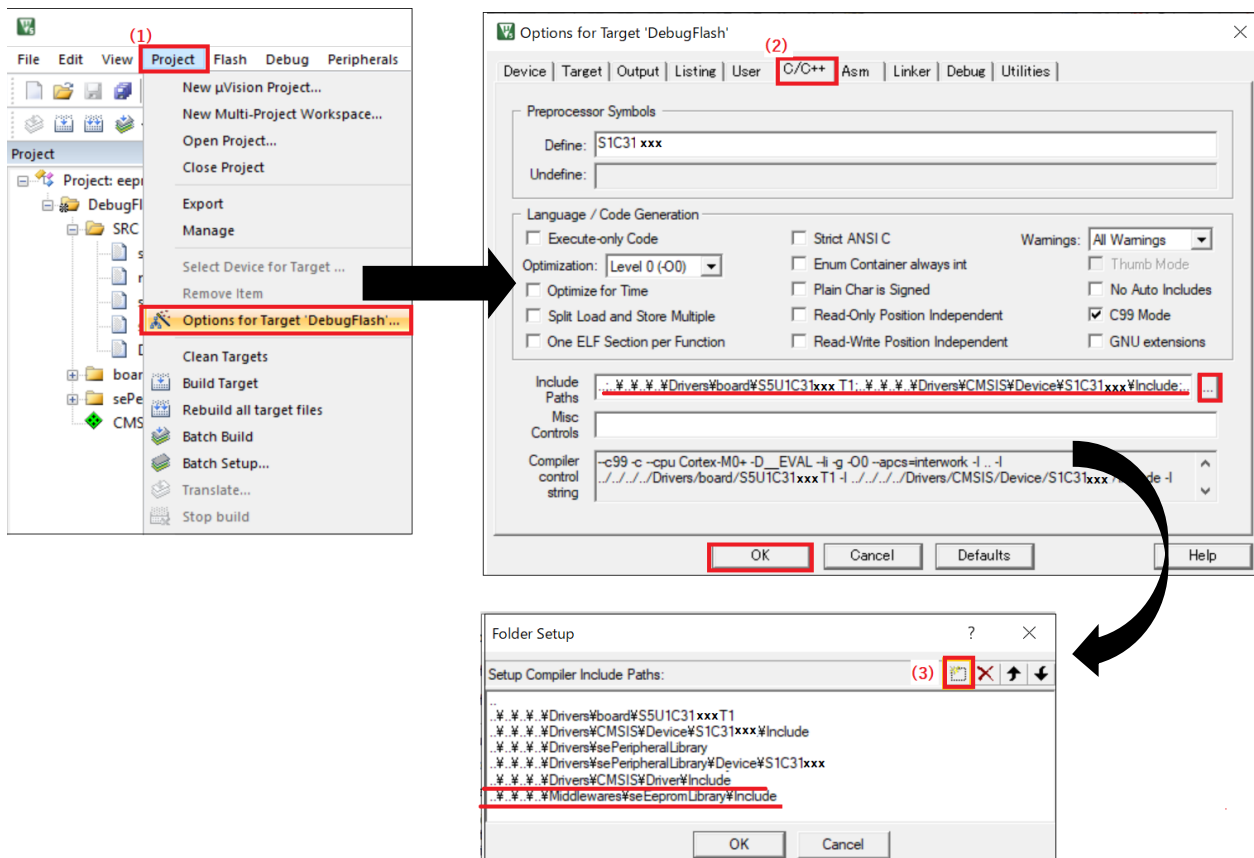


Figure B.2 Set include path

3. Linker Script Settings

- (1) Edit the linker script file (.sct) included in the project.
- (2) Refer to the linker script file (S1C31xxx_flash.icf) of the sample software included in the S1C31xxx peripheral circuit sample software package, and add the following sections.

```

...
LR_IROM1 0x00000000 0x00020000 { ; load region size_region
...
  RW_IRAM2 +0 { ; RW data
    *(.flash_common_text)
  }
  ARM_LIB_HEAP 0x00151800 EMPTY 0x400 ; Heap region growing up
  {}
  ARM_LIB_STACK 0x00152000 EMPTY -0x400 ; Stack region growing down
  {}
}

LR_IROM2 0x00020000 0x00010000 { ; load region size_region
  ER_IROM2 0x00020000 0x00010000 { ; load address = execution address
    *(.eeprom)
  }
}

```

Creating a flash_common_text section in RAM area

Specify the start address of the internal flash memory used as the EEPROM emulation area

When the above is added, the code of this library is placed in the RAM area in this library. Also, specify the internal flash memory to be used as the EEPROM emulation area. Specify the start address in the unit described in the readme.

- (3) Select [Project]> [Options for Target 'xxx'...] from the uVision menu.
- (4) Specify the linker script file edited from [Linker]> 'Scatter File' in the displayed dialog.

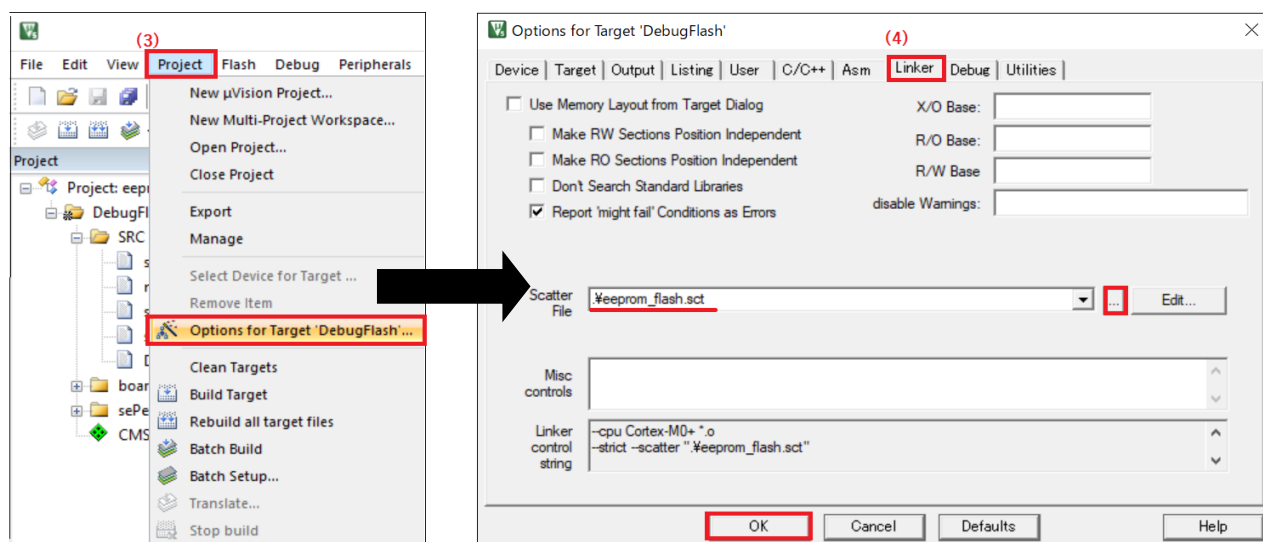


Figure B.3 Set linker script file

Revision History

Revision History

Attachment-1

[illegible]

America

Epson America, Inc.

Headquarter:
3131 Katella Ave., Los Alamitos, CA 90720, USA
Phone: +1-562-290-4677

San Jose Office:
214 Devcon Drive
San Jose, CA 95112 USA
Phone: +1-800-228-3964 or +1-408-922-0200

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 1701 & 1704, 17 Floor, Greenland Center II,
562 Dong An Road, Xu Hui District, Shanghai, 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

29th Floor, JR Shinjuku Miraina Tower, 4-1-6 Shinjuku,
Shinjuku-ku, Tokyo 160-8801, Japan

