

S1C17 Family EEPROM Emulation Library Manual

Evaluation board/kit and Development tool important notice

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

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, 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.) and applications individually listed in this document ("General Purpose"). Epson products are NOT intended for any use beyond the General 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. Therefore, you are advised to use Epson products only for General Purpose uses. Should you desire to buy and use Epson products for a particular purpose other than a General Purpose uses, 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.
 - 【Particular purpose】
 - Space equipment (artificial satellites, rockets, etc.)
 - Transportation vehicles and their control equipment (automobiles, aircraft, trains, ships, etc.)
 - Medical equipment (other than applications individually listed in this document) / 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
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.

(Rev.e1.0, 2021.9)

©Seiko Epson Corporation 2022, All rights reserved.

Summary

This reference material describes the S1C17 Family EEPROM emulation library that provides an EEPROM emulation function using the embedded flash memory of the S1C17 microcontrollers.

Operating Environment

- PC
The GNU17 (S5U1C17001C) development tool and the ICDmini USB driver must be installed.
- ICDmini (S5U1C17001H2 or S5U1C17001H3)
A USB cable is required for connecting with the PC.
- Target system (user target board or our company's evaluation board)
- S1C17xxx EEPROM emulation library package (this package)

Precautions for Usage

The library included in this package is provided as a sample. 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 material is common to S1C17 Family microcontrollers.

In this material, "xxx" represents an S1C17 model name.

The EEPROM emulation library is provided for each model. For the models that support the EEPROM emulation library, please visit our website.

Table of Contents

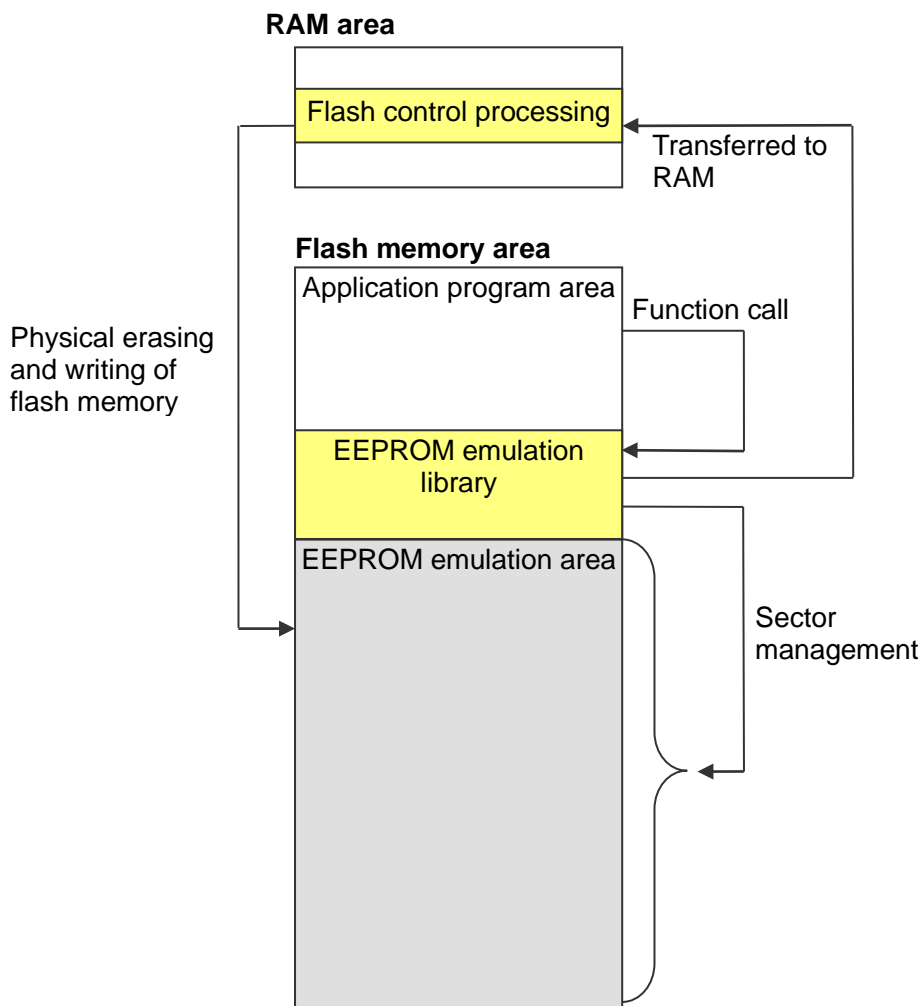
Summary.....	i
Operating Environment	i
Precautions for Usage	i
1. Overview	1
1.1 Features.....	1
1.2 Folder Configuration.....	2
1.3 File Configuration.....	2
2. How to Use Library	3
2.1 Settings for Using Library in Application Program	3
2.2 Internal RAM and Flash Memory Usage	6
2.3 Write Time	6
2.4 Precautions on Use of Library	6
2.5 Sample Program	7
3. Library Specifications	8
3.1 Library Function Details	8
3.2 Error Code Definitions	10
Appendix.....	11
A. How to Incorporate Library into Project (GNU17 Ver. 2.x)	11
B. How to Incorporate Library into Project (GNU17 Ver. 3.x).....	18
Revision History	21

1. Overview

The S1C17 EEPROM emulation library package 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.

1.1 Features

This library implements the emulation function allowing the S1C17 Family microcontrollers to use the embedded flash memory as an EEPROM. It uses a flash memory area of 2K to 64K bytes as an EEPROM emulation area to emulate a 16- 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 25,000-100,000 times (For details, refer to the supplementary manual “s1c17 (xxx) eeprom_notes_e.txt” attached to each library.) theoretically.



1. Overview

1.2 Folder Configuration

The folders of this package are configured as shown below.

+ s1c17(xxx)eeprom	
+ eeprom	: EEPROM emulation library
+ s1c17(xxx)eeprom_gnu17v2	: Sample program for GNU17 Ver. 2.x
+ s1c17(xxx)eeprom_gnu17v3	: Sample program for GNU17 Ver. 3.x
- s1c17(xxx)eeprom_notes_e.txt	: Supplementary document (English)
- s1c17(xxx)eeprom_notes_j.txt	: Supplementary document (Japanese)
- License_e.txt	: Software license agreement (English)

1.3 File Configuration

The table below lists the library file configuration.

Table 1.1 s1c17(xxx)eeprom/eeprom

Filename	Function
dataFlash17(xxx).a	S1C17(xxx) EEPROM emulation library (for running on flash memory)
dataFlash17(xxx)ram.a	S1C17(xxx) EEPROM emulation library (for running on RAM)
FlashControlErase.o	Flash memory erasing function
FlashControlWrite.o	Flash memory writing function
DataFlashConfig.h	EEPROM configuration header file
DataFlashConfig.c	EEPROM configuration source file
DataFlashCommand.h	Function declaration header file
OscControl.h	Clock source control header file
OscControl.c	Clock source control source file
FlashArea.s	EEPROM emulation area allocation file

The table below lists the file configuration of sample program.

Table 1.2 s1c17(xxx)eeprom/s1c17(xxx)eeprom_gnu17vx

File/folder name	Function
eeprom	EEPROM emulation library (folder)
boot.c	Boot program
main.c	Main program

2. How to Use Library

Describes how to use the S1C17 EEPROM emulation library and sample software.

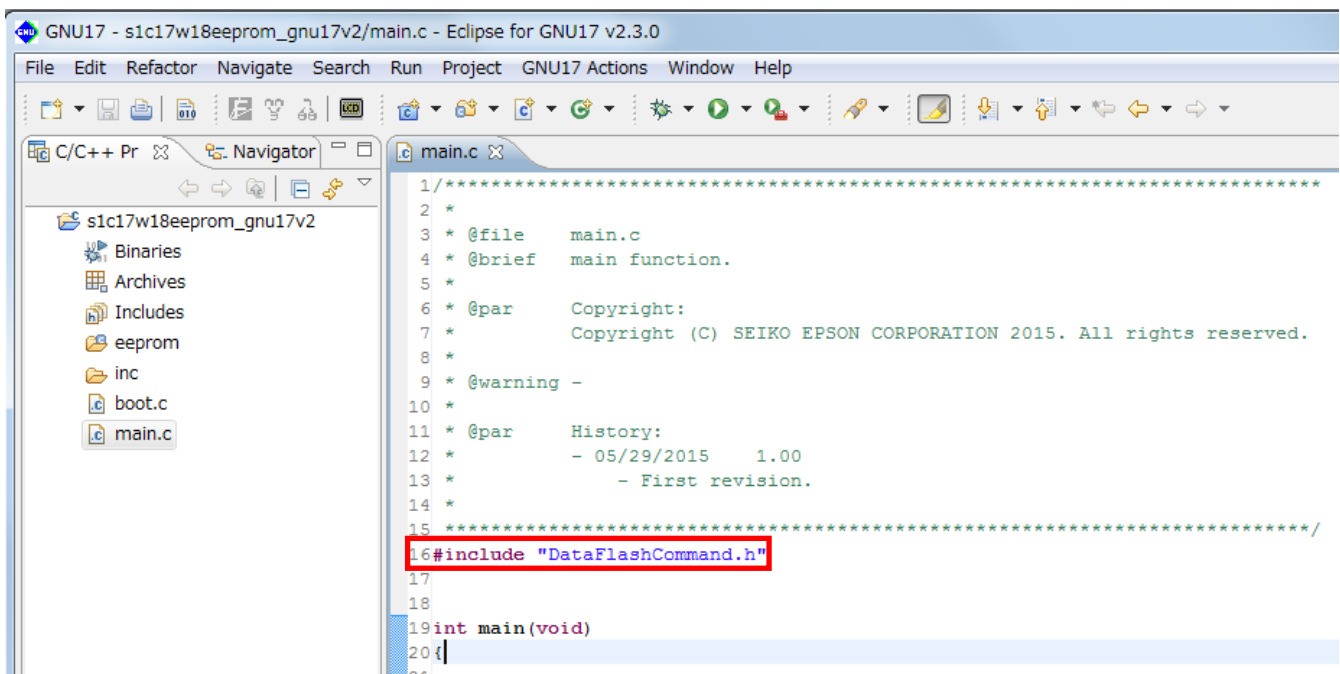
2.1 Settings for Using Library in Application Program

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. Header file declaration

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

Note: Add the path to the header file if no include path has been defined.



2. Setting EEPROM size and number of write retries

Edit “DataFlashConfig.h” to redefine the constants shown below.

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

- CONFIG_EEPROM_SIZE_MAX

Rewrite the “CONFIG_EEPROM_SIZE_MAX” value with the size (byte) of EEPROM to be emulated. For the sizes that can be specified, refer to the supplementary document “s1c17(xxx)eeprom_notes_e.txt.”

- CONFIG_RETRY_COUNT

Rewrite the “CONFIG_RETRY_COUNT” value with 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.

2. How to Use Library

3. Setting clock source

In order to use this library, it is necessary to set the system clock and T16 ch.0 to the optimum speed. The system clock and T16 ch.0 are set in OscControl.c. The contents of each function are as follows.

Table 2.1.1 Functions of "OscControl.c"

Function name	Contents
OscClockSourceInitialize()	Set the system clock and T16 ch.0 to the appropriate speeds to use this library. ◇ Change the settings of the user program.
OscClockSourceFinalize()	Set the system clock and T16 ch.0 back to the user program settings.

NOTE: This function is used in this library. Therefore, the user must check the source code of this function and modify it if necessary.

To use this library, use the above function and set the following appropriate operating speed.

Table 2.1.2 Operating speed of system clock and T16ch.0

Model name	This library version	System clock	T16 ch.0*1
S1C17M23/24/25	Before Ver1.3x	OSC3=16MHz Division ratio 1/4	OSC3=16MHz
	After Ver1.4x	OSC3=16MHz Division ratio 1/1	OSC3=16MHz
S1C17M30/31/32/34	Before Ver2.2	OSC3=16MHz Division ratio 1/4	OSC3=16MHz
	After Ver2.3	OSC3=16MHz Division ratio 1/1	OSC3=16MHz
S1C17M33	Before Ver2.2	OSC3=16MHz Division ratio 1/4	OSC3=16MHz
	After Ver2.3	OSC3=16MHz Division ratio 1/1	OSC3=16MHz
S1C17W18	All	OSC3=4MHz Division ratio 1/1	OSC3=4MHz
S1C17W36	All	OSC3=4MHz Division ratio 1/1	OSC3=4MHz

*1 The division ratio of T16 ch.0 is set in this library.

4. Adding EEPROM read/write functions

Add the EEPROM read/write functions included in this library to the application program source. For detailed information on the functions, refer to Chapter 3, "Library Specifications."

```

for(i = 0 ; i < CONFIG_EEPROM_SIZE_MAX ; i++)
{
    if(DataFlashWrite(i, i) != DATAFLASH_SUCCESS)
    {
        asm("nop");
    }
    testdata[i] = DataFlashRead(i);
}
//compare
for(i = 0 ; i < CONFIG_EEPROM_SIZE_MAX ; i++)
{
    if(testdata[i] != (i & 0xff))
    {
        asm("nop");
    }
}

```

5. About the data read unit and write unit

The data read unit and write unit differ depending on the target model and the version of this library. Please confirm that these units depend on the type of the following function definition of "DataFlashCommand.h" in this library.

- function definition

```

extern int DataFlashWrite(unsigned short address, unsigned char data);
extern unsigned char DataFlashRead(unsigned short address);
extern unsigned char DataFlashReadCurrent(void);

```

- Data read unit and write unit

Table 2.1.3 Type and unit

Type	Data read unit and write unit
unsigned char	1byte
unsigned short	2byte
unsigned long	4byte

2. How to Use Library

2.2 Internal RAM and Flash Memory Usage

This library uses an internal RAM area and a flash memory area. For the memory usage in each model, refer to the supplementary document “s1c17(xxx)eeprom_notes_e.txt.”

2.3 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 S1C17W18 and S1C17M33.

- S1C17W18 (System clock : Built-in OSC3 4MHz)

Typ. value	7 ms
Max.value	43 ms

- S1C17M33 (System clock : Built-in OSC3 16MHz)

Typ. value	6 ms
Max.value	38 ms

The actual write time should be determined using the target system with this library implemented.

2.4 Precautions on Use of Library

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

- Ch.0 of the 16-bit timer (T16) is exclusively used for this library to control the flash write timing.
- The system clock and T16 Ch.0 configurations are changed at the beginning of the EEPROM write function when being executed. For details, refer to “3. Setting clock source” in Section 2.1.
- 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 $\langle \text{CONFIG_EEPROM_SIZE_MAX} * 128 \rangle$ 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 “S1C17(xxx) Technical Manual”, and disconnect the connection between the FLASH_VCC_OUT pin of ICDmini and the Vpp pin of the MCU.
- 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.

2.5 Sample Program

1. Sample program specification

The sample program performs the operation shown below using this library.

- Writes data, which starts from 0 and is incremented by 1 in each address, to the address range from 0 to <CONFIG_EEPROM_SIZE_MAX – 1> and then verifies the data written.

2. Preparation

Follow the procedure shown below to run the sample program on IDE. Also keep the descriptions under Sections 2.1 to 2.4 in mind when using the library.

- (1) Importing project
Launch IDE and import the sample program.
- (2) Building
Build the sample program using IDE.
- (3) Connecting
Connect ICDmini and the target system to the PC.
- (4) Unprotecting flash
When debugging the sample program in an IC with protected flash, it must be changed to unprotected.
- (5) Loading program
Load the program into IDE.
- (6) Executing
Run the program by resetting the target system or other method.

For more information, refer to “S1C17(xxx) Technical Manual,” “S5U1C17001C Manual,” and “S5U1C17001H User Manual (ICDmini).”

3. Operation overview

- (1) Initializes the EEPROM address to 0 and the write data to 0.
- (2) Writes data by calling the EEPROM write function (DataFlashWrite() in main.c).
- (3) Reads data from the address to which data is written in Step (2) (Data Flash Read() in main.c).
- (4) Increments the address and write data by 1 and returns to Step (2) if the current address is smaller than CONFIG_EEPROM_SIZE_MAX.
- (5) Compares the read data and the write data.

For the DataFlashRead() and DataFlashWrite() functions, refer to Section 3, “Library Specifications”.

3. Library Specifications

3. Library Specifications

3.1 Library Function Details

This section describes the functions defined in this library.

EEPROM write function

Format	DataFlashWrite(unsigned short <i>address</i> , unsigned xxx <i>data</i>);	
Arguments	unsigned short <i>address</i>	EEPROM address
	unsigned char <i>data</i> unsigned short unsigned long	Write data
Return value	int	Writing result (error code)
Description	This function writes data according to the conditions specified via the arguments. (1) Checks whether the arguments are correct or not. (2) Writes data to the specified address. (3) Returns the error code as the return value.	
Remarks	- The effective range of the first argument is 0 to <CONFIG_EEPROM_SIZE_MAX - 1>. - The type of write data differs depending on the model and version.	

EEPROM read function

Format	DataFlashRead(unsigned short <i>address</i>);	
Argument	unsigned short <i>address</i>	EEPROM address
Return value	unsigned char unsigned short unsigned long	Read data
Description	This function reads data from the address specified via the argument. (1) Checks whether the argument is correct or not. (2) Reads data from the specified address. (3) Returns the read data as the return value.	
Remarks	- The effective range of the argument is 0 to <CONFIG_EEPROM_SIZE_MAX - 1>. - 0xff is read from the address in which no data has been written. - The type of read data differs depending on the model and version.	

Sequential EEPROM read function

Format	DataFlashReadCurrent(void);	
Argument	–	None
Return value	unsigned char unsigned short unsigned long	Read data
Description	<p>This function reads data from the current address.</p> <p>(1) Reads data from the current address.</p> <ul style="list-style-type: none"> • The address is incremented by 1 after being read. • The address is reset to 0 after data is read from the end address. • If this function is called after the DataFlashWrite() function is executed, data is read from the address specified in the DataFlashWrite() function. • If this function is called after the DataFlashRead() function is executed, data is read from the next address of the address specified in the DataFlashRead() function. <p>(2) Returns the read data as the return value.</p>	
Remarks	<ul style="list-style-type: none"> - The initial current address is 0. - 0xff is read from the address in which no data has been written. - The type of read data differs depending on the model and version. 	

Change the clock source setting function

Format	OscClockSourceInitialize (void)	
Argument	none	
Return value	none	
Description	<p>Set the system clock and T16Ch.0, to the appropriate speed for the library to operate.</p> <p>(1) Save the operating clock and T16Ch.0 settings.</p> <p>(2) Change the Flash Read Cycle.</p> <p>(3) Set the system clock.</p> <p>(4) Set T16 Ch.0.</p>	
Remarks	<p>It is called from the library function before the write process.</p> <p>Since the source file is open to the public, it can be modified by the user.</p>	

Restoration of operating clock settings

Format	OscClockSourceFinalize (void)	
Argument	none	
Return value	none	
Description	<p>Restores the system clock and T16Ch.0 settings to the settings before calling the library function.</p> <p>(1) Restore the system clock setting.</p> <p>(2) Restore the settings of T16 Ch.0.</p> <p>(3) Restore the Flash Read Cycle.</p>	
Remarks	<p>It is called from the library function before the write process.</p> <p>Since the source file is open to the public, it can be modified by the user.</p>	

3. Library Specifications

3.2 Error Code Definitions

Table 3 Error Codes

Definition Name	Value	Description
DATAFLASH_SUCCESS	0	The writing has successfully completed.
DATAFLASH_ERROR_ERASE	1	An erase error has occurred.
DATAFLASH_ERROR_WRITE	2	A write error has occurred.
DATAFLASH_ERROR_PARAMETER	3	A parameter error has occurred.

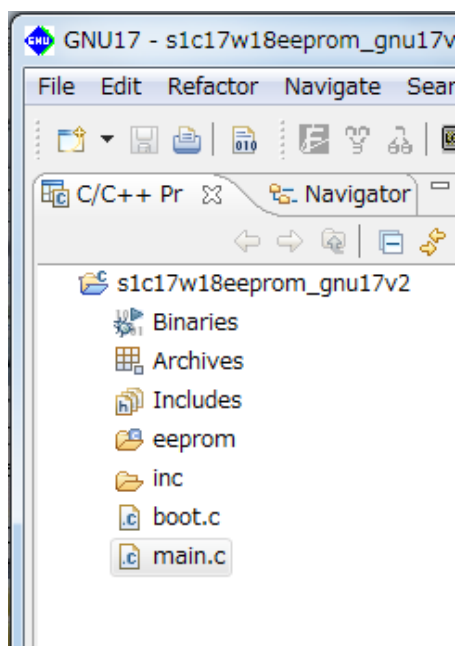
Appendix

A. How to Incorporate Library into Project (GNU17 Ver. 2.x)

The following describes how to handle this library with GNU17 Ver. 2.x using S1C17W18 as an example. For detailed information on usage of GNU17 Ver. 2.x, refer to the “S5U1C17001C Manual (Ver. 2.x.x).”

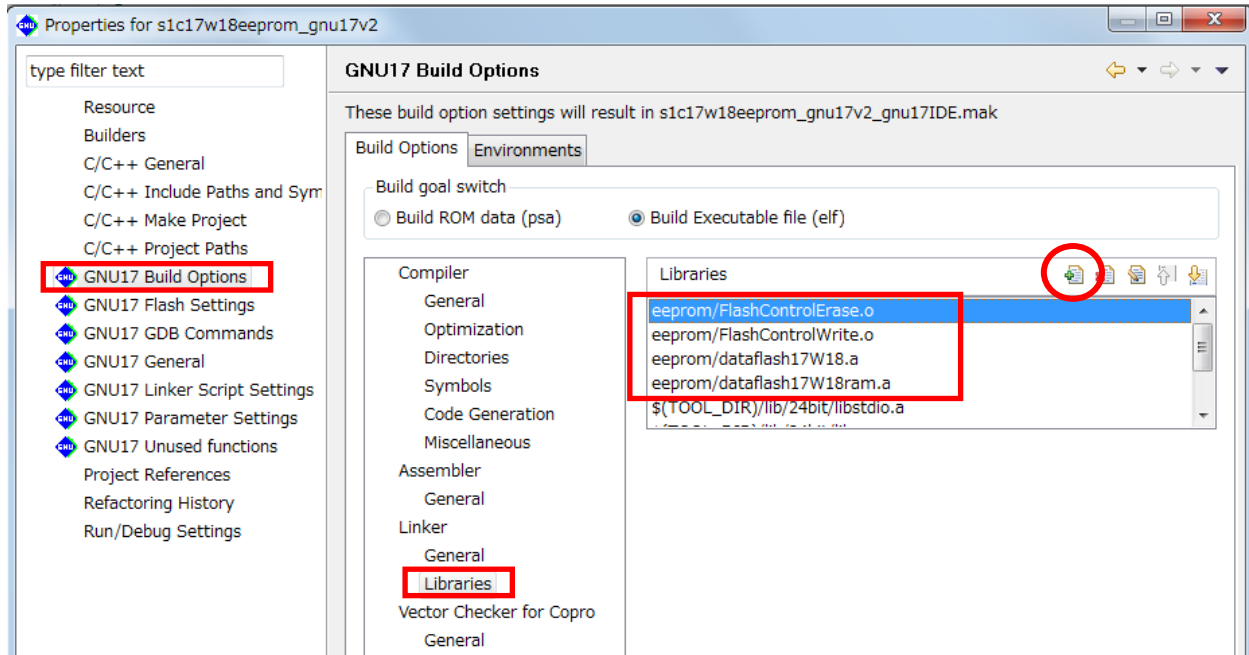
1. Importing library and header files

Import the eeprom folder included in this package into the project folder.



2. Adding libraries

The libraries imported must be added to the library list of the build option before they can be used. Open the [Properties] dialog box of the project and select [GNU17 Build Options] - [Linker] - [Libraries]. Click the [Add] button (indicated with a red circle in the figure below) and select “dataflash17W18.a,” “dataflash17W18ram.a,” “FlashControlErase.o,” and “FlashControlWrite.o,” which are included in the eeprom folder, to add them to the library list.

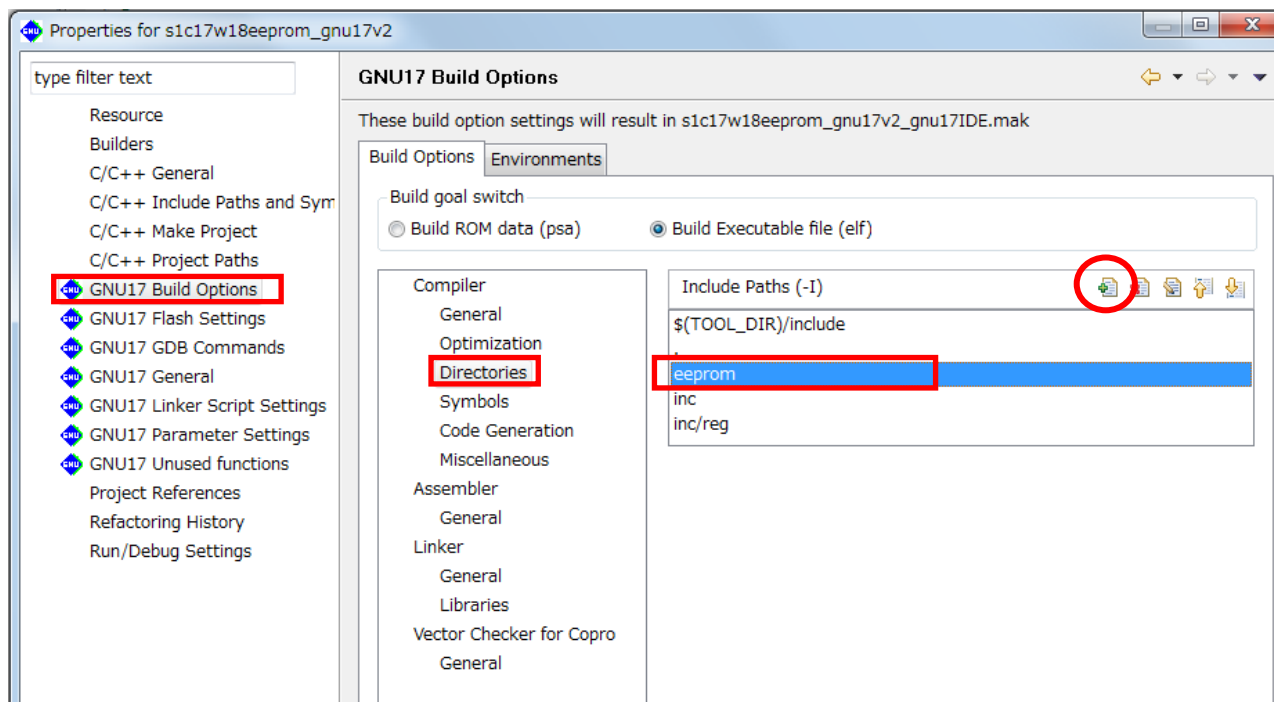


3. Setting include path

Set the include path to use “DataFlashCommand.h” included in the eeprom folder.

Open the [Properties] dialog box of the project and select [GNU17 Build Options] - [Directories]. Click the [Add] button (indicated with a red circle in the figure below) and select the eeprom folder as an include path.

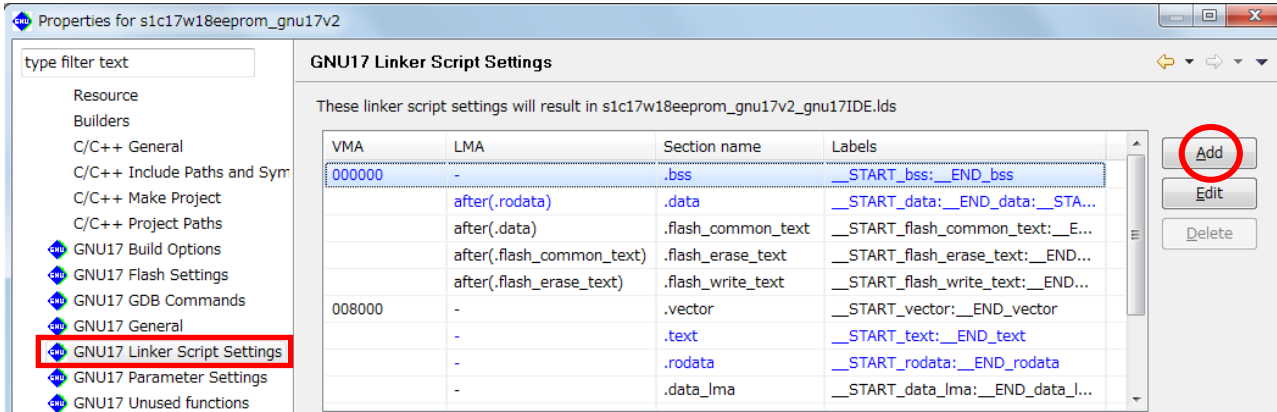
Note: This setting is not necessary if the include path is directly specified in the source file.



4. Editing linker script

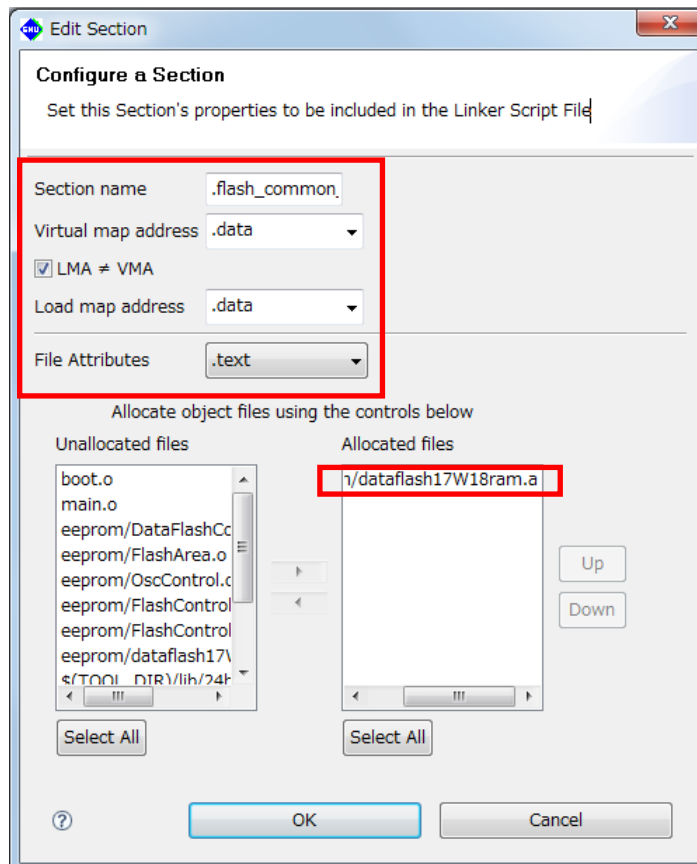
Edit the linker script for the library imported.

Open the [Properties] dialog box of the project and select [GNU17 Linker Script Settings]. Click the [Add] button (indicated with a red circle in the figure below) and add the sections to which the libraries will be placed.

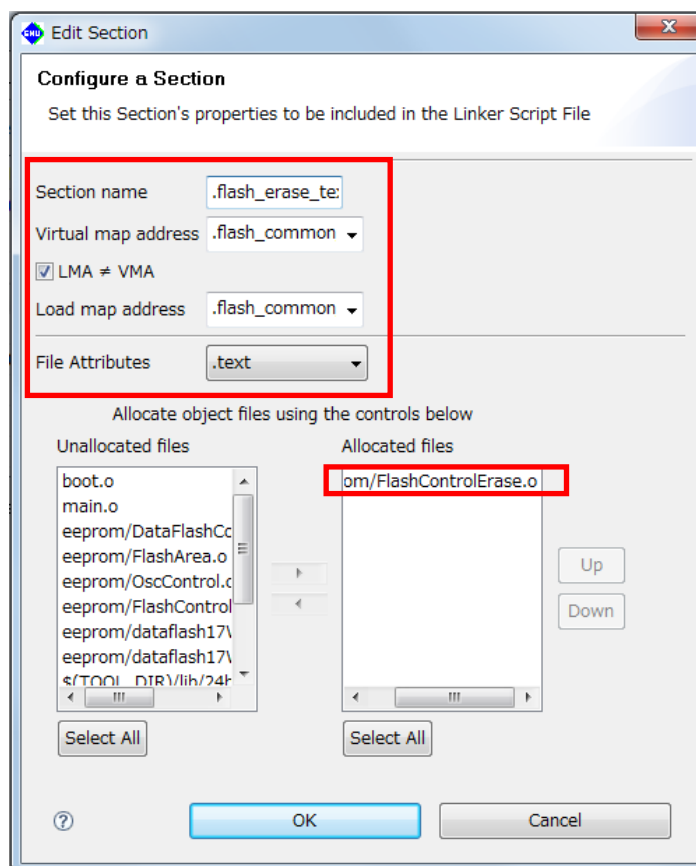


Add the “.flash_common_text,” “.flash_erase_text,” “.flash_write_text,” and “.flashdata_address” sections as below. The section name must begin with a dot (.).

Place “dataflashW18ram.a” in the “.flash_common_text” section.

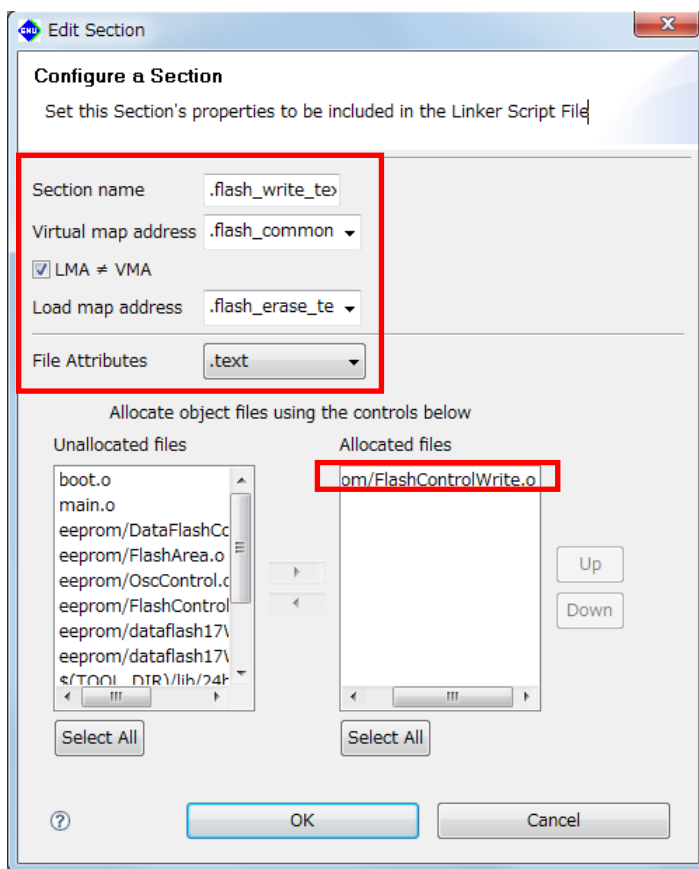


Place “FlashControlErase.o” in the “.flash_erase_text” section.
Set “.flash_common_text” to VMA and LMA.



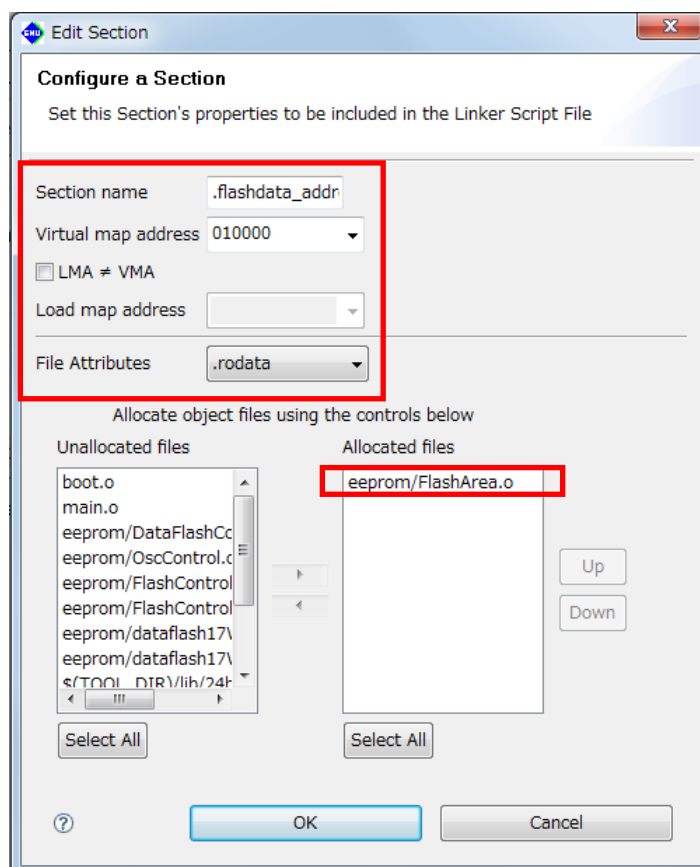
Appendix

Place “FlashControlWrite.o” in the “.flash_write_text” section.
Set “.flash_common_text” and “.flash_erase_text” to VMA and LMA, respectively.



Place “FlashArea.o” in the “.flashdata_addr” section.

Set the leading address of the EEPROM emulation area in the flash memory to VMA. The address must be specified in 0x80 increments.

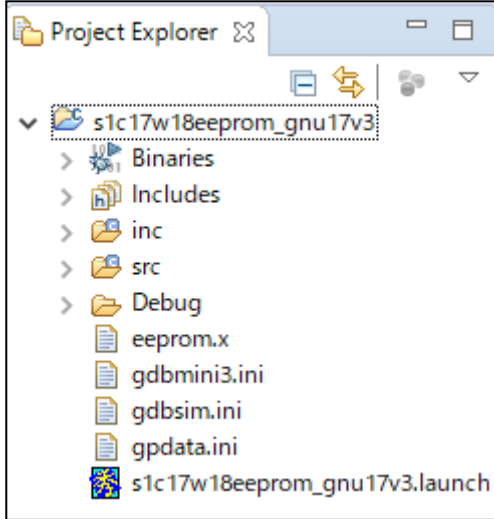


B. How to Incorporate Library into Project (GNU17 Ver. 3.x)

The following describes how to handle this library with GNU17 Ver. 3.x using S1C17W18 as an example. For detailed information on usage of GNU17 Ver. 3.x, refer to the “S5U1C17001C Manual (Ver. 3.x.x).”

1. Importing library and header files

Import the eeprom folder included in this package into the src folder in the project.



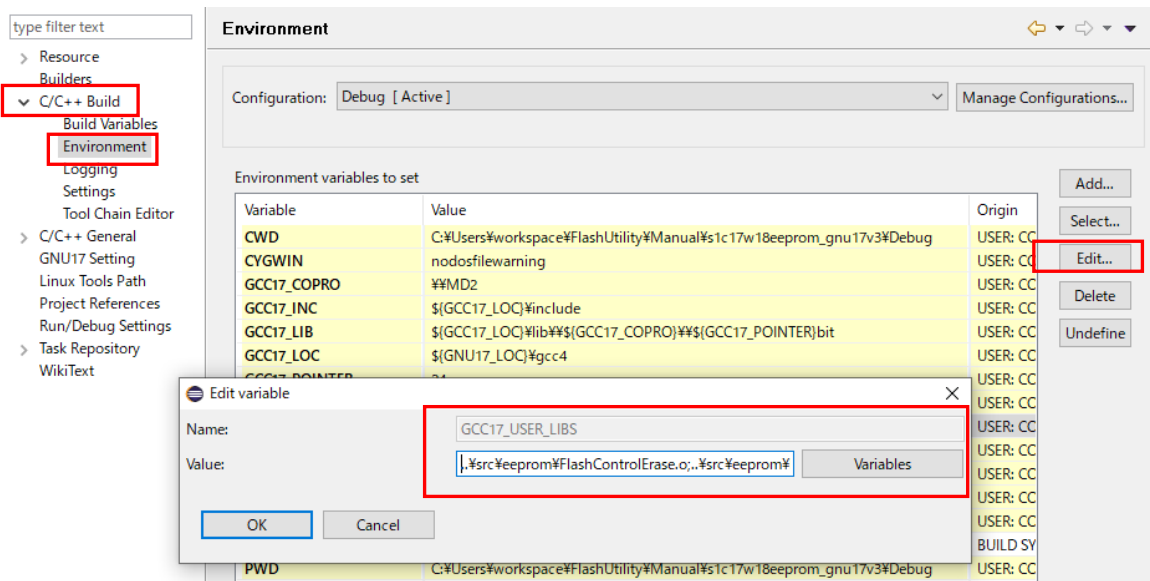
2. Adding libraries

The libraries imported must be added to the library list before they can be used.

Open the [Properties] dialog box of the project and select [C/C++ Build] - [Environment]. Add “dataflash17(XXX).a,” “dataflash17(XXX)ram.a,” “FlashControlErase.o,” and “FlashControlWrite.o,” which are included in the src/eeprom folder, to the Value of Variable GCC17_USER_LIBS.

```

./src/eeprom/FlashControlErase.o;./src/eeprom/FlashControlWrite.o;
./src/eeprom/dataflash17W18.a;./src/eeprom/dataflash17W18ram.a
    
```

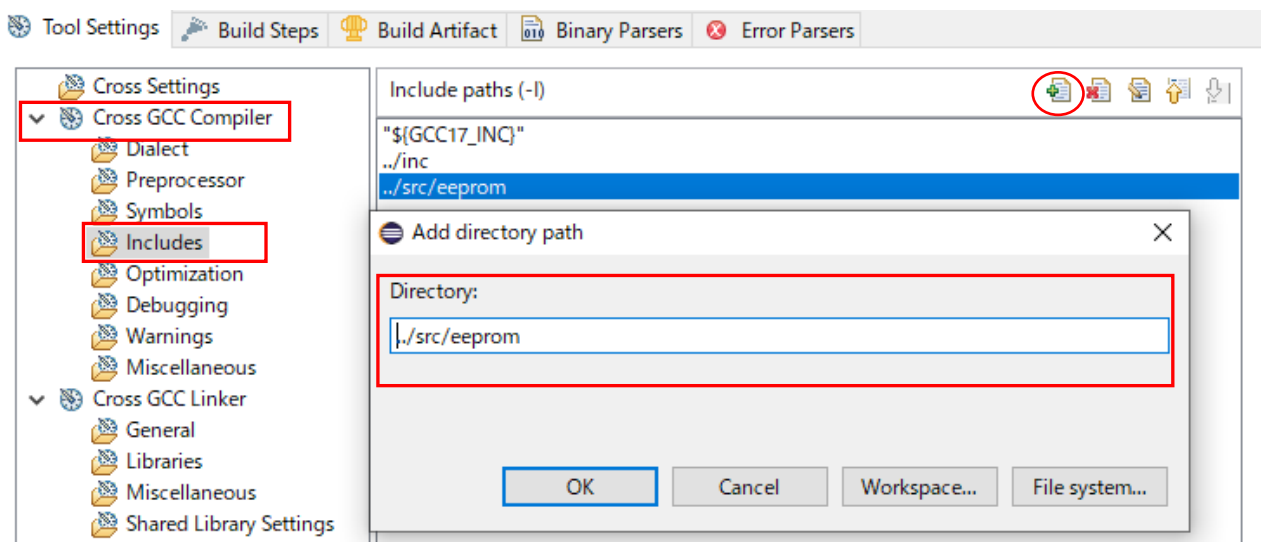


3. Setting include path

Set the include path to use “DataFlashCommand.h” included in the eeprom folder.

Open the [Properties] dialog box of the project and select [C/C++ Build] - [Settings] - [Tool Settings] - [Cross GCC Compiler] - [Includes]. Set the include path to the src/eeprom folder.

```
"/../src/eeprom"
```

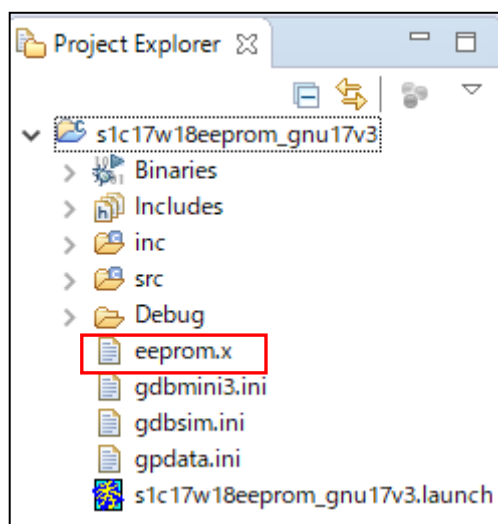


4. Setting linker script

Specify the linker script for the library.

A sample linker script file for the EEPROM emulation library exists in the folder shown below. Copy it to the project folder.

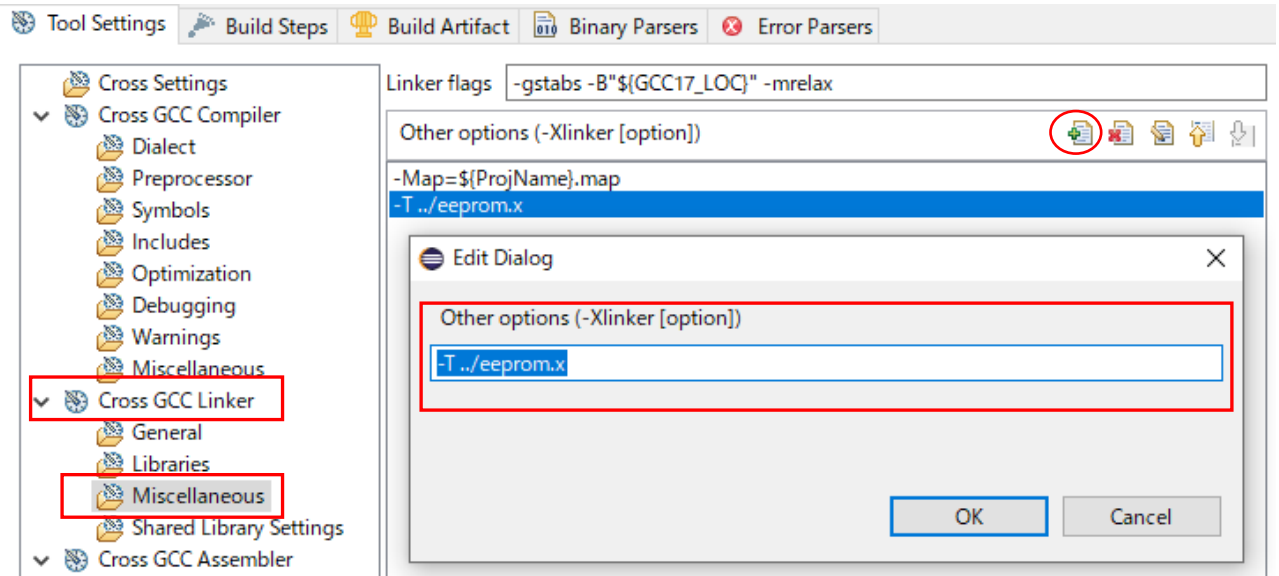
```
/c17 (xxx) _sample_gnu17v3/eeprom.x
```



Appendix

Open the [Properties] dialog box and select [C/C++ Build] - [Settings] - [Tool Settings] - [Cross GCC Linker] - [Miscellaneous]. Enter the option shown below into [Other options] to specify the linker script copied.

```
-T ../eeprom.x
```



This linker script defines the symbols shown below that are required for the processing of the library and arranges the library execution address in the internal RAM.

```
__START_flash_common_text_lma
__START_flash_erase_text_lma
__START_flash_write_text_lma
```

The script shown below specifies that “FlashControlCommon.o,” “FlashControlWrite.o,” and “FlashControlErase.o” will not be placed in the RAM.

```
* (EXCLUDE_FILE (*FlashTimeTable*.o*FlashControlCommon.o
                 *FlashControlWrite.o*FlashControlErase.o) .text)
```

The “.flashdata_address” specifies the leading address of the EEPROM emulation area in the flash memory. The address must be specified in 0x80 increments if you change this setting.

```
.flashdata_address 0x010000 :
```


Revision History

Attachment-1

Rev. No.	Date	Page	Category	Contents
Rev 1.0	2015/10/08	All	New	New establishment
Rev 1.1	2017/04/07	i,1,2,3,5, (15)	Revision	Made corrections to support S1C17M30/M31/M32/M33/M34.
Rev. 1.2	2018/07/01	1, 5, 15, 16	Revision	Removed the descriptions for the S1C17M30/M31/M32/M34 to return back to the original, as a separate manual for these models will be provided.
		3	Revision	Modified the description of "2. Setting EEPROM size and number of write retries."
		4	Revision/ addition	Modified and added descriptions of "3. Setting clock source."
		5	Revision/ addition	Corrected the source code described under "4. Adding EEPROM read/write functions." Modified and added descriptions in Section 2.4, "Precautions on Use of Library."
Rev. 1.3	2018/10/18	6	Addition	Added descriptions in Section 2.4, "Precautions on Use of Library."
Rev. 1.4	2019/11/01	17	Revision	Corrected descriptions in Appendix B, "2 Adding libraries."
Rev. 1.5	2020/05/26	5	Revision	Modified descriptions in Section 2.4, "Precautions on Use of Library."
Rev 1.6	2022/03/04	4	Revision	Modified descriptions in Section 3. "Setting clock source" of 2.1 "Settings for Using Library in Application Program".
		5	Addition	Added descriptions in Section 5. "About the data type of EEPROM read / write function" of 2.1 "Settings for Using Library in Application Program". And, added descriptions in Section 2.3 "Write Time".
		9	Addition	Added descriptions in Section 3.1 "Library Function Details".
		18-20	Addition	Added figure in Section Appendix "B.How to Incorporate Library into Project (GNU17 Ver. 3.x)".

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: +86-755-3299-0588 FAX: +86-755-3299-0560

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

MD Sales & Marketing Department

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