

S1C17 Series Flash Memory Protect Function Application Note

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. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

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

Table of Contents

1. DESCRIPTIONS OF FLASH MEMORY PROTECT FUNCTION	1
2. FLASH MEMORY PROTECT SETTING	2
2.1 Protect Bits	2
2.2 Setting the Flash Memory Protect Functions	3
3. CHECKING THE FLASH MEMORY PROTECT SETTINGS	5
3.1 Data Read Protection	5
3.2 Data Write Protection	7
4. RELEASING THE FLASH MEMORY PROTECTION	8
5. NOTES	10
REVISION HISTORY	11

1. DESCRIPTIONS OF FLASH MEMORY PROTECT FUNCTION

This Application Note explains how to set up the Flash Memory Protect function using GNU17 and ICD software development tools.

The Flash Memory Protect function can protect data stored in the flash memory.

Once the Flash Memory Protect function is set, the program contents can be protected from reading or modifying from malicious third-party.

This function allow the user to protect each sector of flash memory areas by write-disable or read-disable settings.

Data can be write-disabled (called “data write protection”) or read-disabled (called “data read protection”) by setting the respective protect bits.

If the Write Protect bit is set, data is disabled to write in the area.

If the Read Protect bit is set, value “0x0000” is always shown when data is read from the area.

This Application Note explains the Flash Memory Protect functions using the S1C17702 device as a reference.

2. FLASH MEMORY PROTECT SETTING

2. FLASH MEMORY PROTECT SETTING

2.1 Protect Bits

To enable the Flash Memory Protect functions, set the Protect bits as follows. The data read or write protection is set if protect bits of the corresponding memory area are set to zero (0). The following table defines the Protect bits for the S1C17702 device.

0x27ffc - 0x27ffe: Flash Protect Bits

Address	Bit	Function	Setting			Init	R/W	
0x27ffc (16 bits)	D15- 8	reserved	—			—	—	
	D7	Flash write-protect bit for 0x24000–0x27fff	1	Writable	0	Protected	1	R/W
	D6	Flash write-protect bit for 0x20000–0x23fff	1	Writable	0	Protected	1	R/W
	D5	Flash write-protect bit for 0x1c000–0x1ffff	1	Writable	0	Protected	1	R/W
	D4	Flash write-protect bit for 0x18000–0x1bfff	1	Writable	0	Protected	1	R/W
	D3	Flash write-protect bit for 0x14000–0x17fff	1	Writable	0	Protected	1	R/W
	D2	Flash write-protect bit for 0x10000–0x13fff	1	Writable	0	Protected	1	R/W
	D1	Flash write-protect bit for 0x0c000–0x0ffff	1	Writable	0	Protected	1	R/W
0x27ffe (16 bits)	D15-8	reserved	—			—	—	
	D7	Flash data-read-protect bit for 0x24000–0x27fff	1	Readable	0	Protected	1	R/W
	D6	Flash data-read-protect bit for 0x20000–0x23fff	1	Readable	0	Protected	1	R/W
	D5	Flash data-read-protect bit for 0x1c000–0x1ffff	1	Readable	0	Protected	1	R/W
	D4	Flash data-read-protect bit for 0x18000–0x1bfff	1	Readable	0	Protected	1	R/W
	D3	Flash data-read-protect bit for 0x14000–0x17fff	1	Readable	0	Protected	1	R/W
	D2	Flash data-read-protect bit for 0x10000–0x13fff	1	Readable	0	Protected	1	R/W
	D1	Flash data-read-protect bit for 0x0c000–0x0ffff	1	Readable	0	Protected	1	R/W
D0	reserved	1* ^{Note1}			1	R/W		

*Note 1

- Always set bit D0 to logical 1 for address 0x27ffe. If this bit is set to 0, the program cannot boot up.
- The Protect bit settings vary depending on the devices used. Be sure to read and follow the instructions given in the Technical Manual of each device model.
- In some device model, the Flash Memory Protect function is not available. Refer to the Technical Manual of each device model.

2.2 Setting the Flash Memory Protect Functions

The following explains how to set up the Flash Memory Protect functions.
This explanation is based on a simple sample project that uses the SVT17702 device.

The user must create a mask file (called the “psa” file hereafter) using the build of GNU17 and IDE tools, and execute the “protect.bat” file for embedding of protect data in the Protect bits of “psa” file.

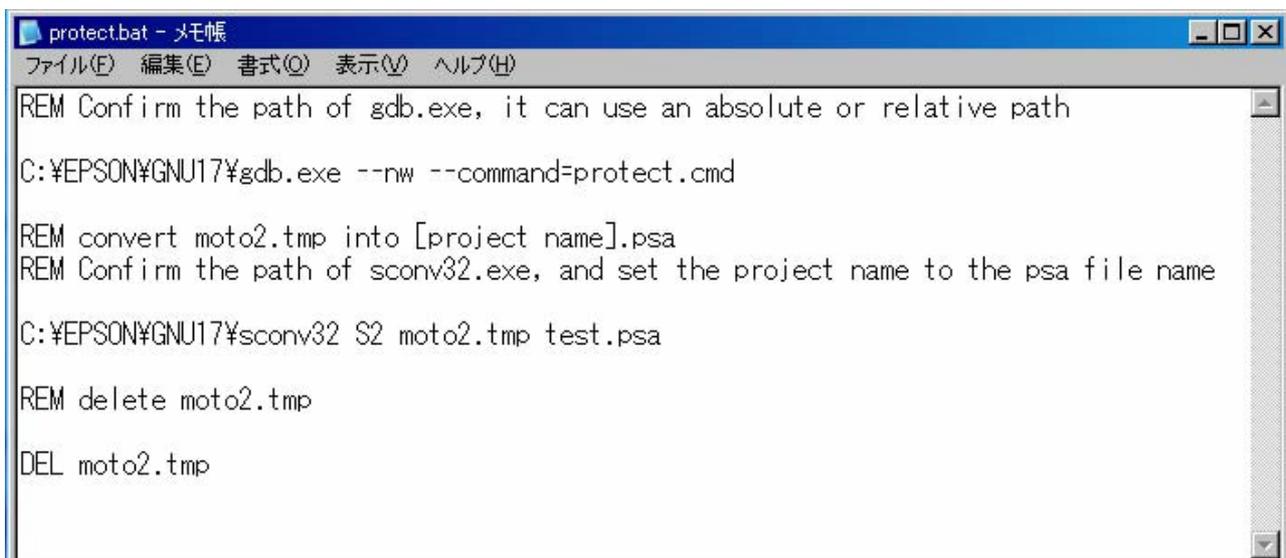
In the preparatory stage, place the two sample software files (“protect.bat” and “protect.cmd” files) in the project folder.

Enter the “red” character sections in the “protect.bat” and “protect.cmd” files. (Do not change the “black” character sections.)

● Setting the “protect.bat” file

Include the Protect bit data.

- Check the “gdb.exe” file path. It can be an absolute or relative file path.
`..\..\gdb --nw --command=protect.cmd`
- Rename the “moto2.tmp” file to the “project-name.psa”.
- Check the “sconv32.exe” file path and set the “psa” file name.
`..\..\sconv32 S2 moto2.tmp project-name.psa`
- Delete the intermediate file “moto2.tmp”.
`DEL moto2.tmp`



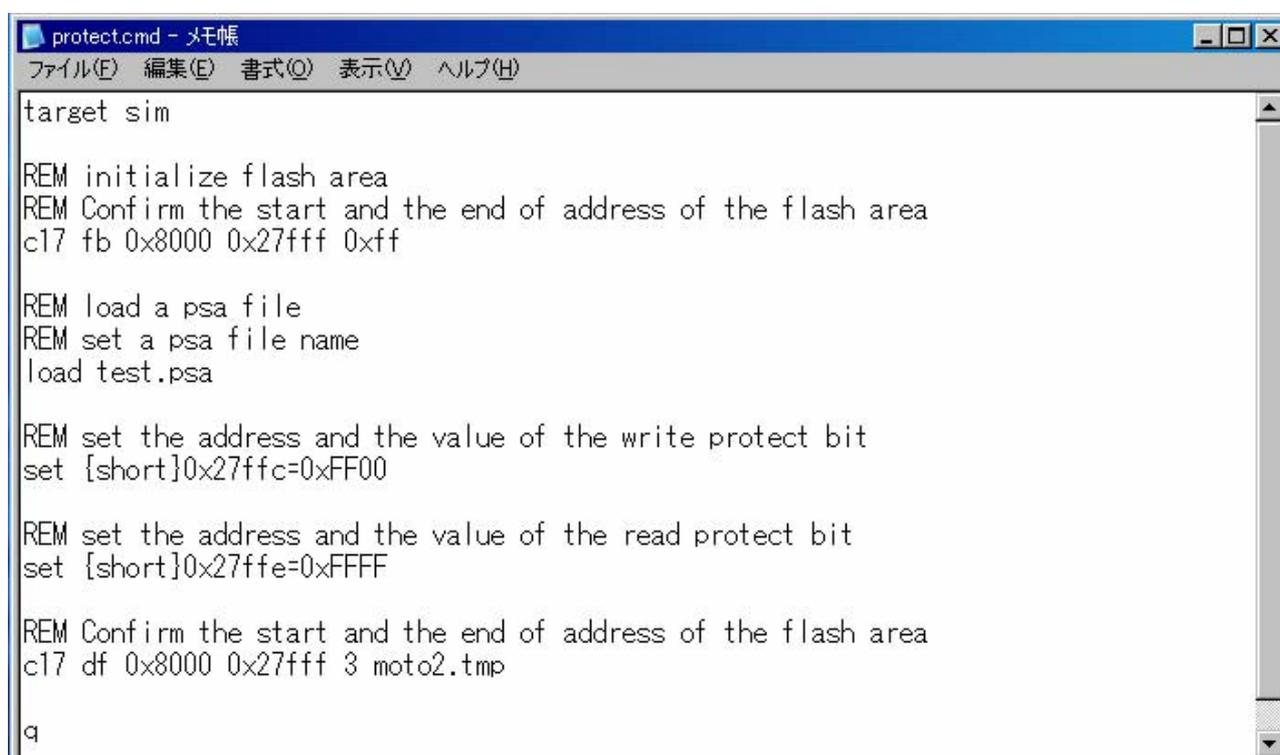
```
protect.bat - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
REM Confirm the path of gdb.exe, it can use an absolute or relative path
C:¥EPSON¥GNU17¥gdb.exe --nw --command=protect.cmd
REM convert moto2.tmp into [project name].psa
REM Confirm the path of sconv32.exe, and set the project name to the psa file name
C:¥EPSON¥GNU17¥sconv32 S2 moto2.tmp test.psa
REM delete moto2.tmp
DEL moto2.tmp
```

Figure 1 protect.bat Setup Screen

2. FLASH MEMORY PROTECT SETTING

● Setting the “protect.cmd” file

```
target sim
• Initialize the flash memory area.
• Check the flash memory area.
  c17 fb 0x8000 0x27fff 0xff
• Loads the “psa” file.
• Set the “psa” file name as follows.
  load project-name.psa
• Set the address and value of Write Protect bits as follows.
  set {short}0x27ffc=0xFFFF
• Set the address and value of Read Protect bits as follows.
  set {short}0x27ffe=0xFFFF
• Check the flash memory area.
  c17 df 0x8000 0x27fff 3 moto2.tmp
q
```



```
protect.cmd - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
target sim
REM initialize flash area
REM Confirm the start and the end of address of the flash area
c17 fb 0x8000 0x27fff 0xff
REM load a psa file
REM set a psa file name
load test.psa
REM set the address and the value of the write protect bit
set {short}0x27ffc=0xFF00
REM set the address and the value of the read protect bit
set {short}0x27ffe=0xFFFF
REM Confirm the start and the end of address of the flash area
c17 df 0x8000 0x27fff 3 moto2.tmp
q
```

Figure 2 protect.cmd Setup Screen

To write the protect data, double-click the “protect.bat” file. The “.psa” file will be overwritten by the data that has been written in the Protect bits.

Then, open the “.psa” file using a text editor, check the data located in the address of Protect bits, and start the debugger by selecting [Run] → [External Tools] options.

Dump the protect bit address in the Memory window, and make sure that the value is written in the Protect bits correctly. The memory is not protected yet at this time. To enable the memory protection, reset the system hardware. *1

*1 Hardware reset

- Press the Reset button of the system hardware.
- From the debugger console, execute the “c17 rstt” target reset command.

Note: When executing the “c17 rstt” command using the ICD-mini software tool, the user must connect the RESET line of the target board to the ICD-mini extension connector port.

3. CHECKING THE FLASH MEMORY PROTECT SETTINGS

The following gives the standard procedure to make sure that the Flash Memory Protect function has been set correctly.

3.1 Data Read Protection

If memory is read-protected, no data can be read from the memory. (The protected area data is shown as “0x0000”.) Check the memory read protection as follows.

2.2 Enable the memory read protection according to the Protect Bit Settings section.

Then, verify that the memory is read-protected.

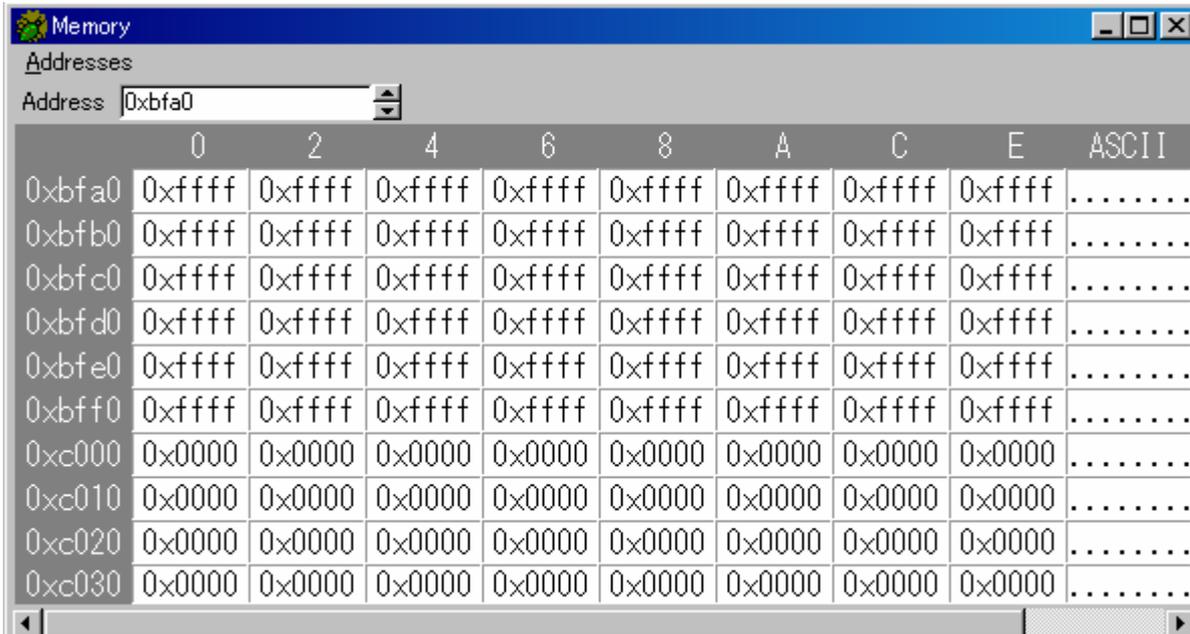
Address	0	2	4	6	8	A	C	E	ASCII
0x27ff0	0xffff	0xfffd						
0x28000	0x0000							
0x28010	0x0000							
0x28020	0x0000							
0x28030	0x0000							
0x28040	0x0000							
0x28050	0x0000							
0x28060	0x0000							
0x28070	0x0000							
0x28080	0x0000							

Figure 3 Protect Bit Checkout Example

If the user dumps data 0x027ffc in the Memory window of the debugger as shown in Figure 3, the user can confirm that data sets 0xffff and 0xfffd are written in 0x027FFC and 0x027FFE correctly. This shows that the Protect bits have been set correctly.

3. CHECKING THE FLASH MEMORY PROTECT SETTINGS

The next example shows the memory dump of the “read-protected” memory area.



	0	2	4	6	8	A	C	E	ASCII
0xbfa0	0xffff							
0xbf b0	0xffff							
0xbf c0	0xffff							
0xbf d0	0xffff							
0xbf e0	0xffff							
0xbf f0	0xffff							
0xc000	0x0000							
0xc010	0x0000							
0xc020	0x0000							
0xc030	0x0000							

Figure 4 Checkout Example of Memory Read Protection

Memory areas 0x0c000 to 0x0ffff are read-protected. If the read-protected memory area has data 0x0000 as shown in the Figure 4, the memory read protection has been set correctly.

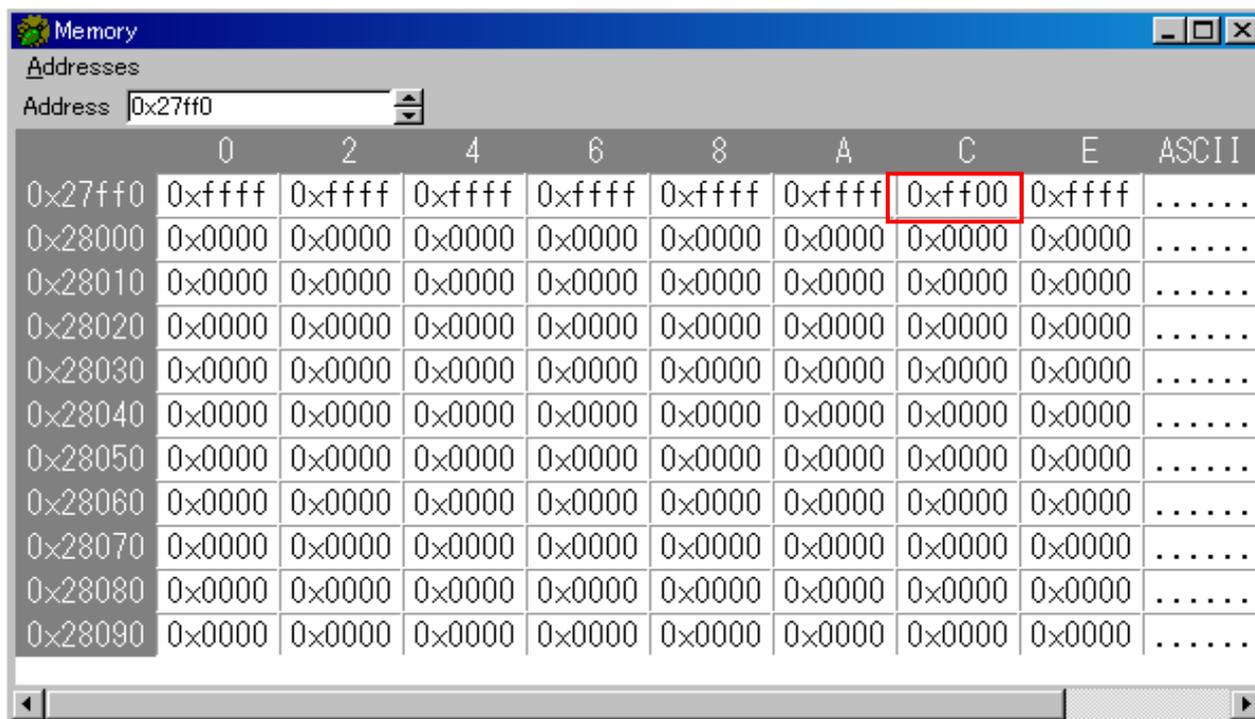
If the read-protected memory area does not have data 0x0000, it shows that the memory read protection has failed.

- RESET line is disconnected from the extended connector port of ICDmini and user target board:
Check the protection with the following procedure.
 - Reset the CPU once to enable the memory protection.
 1. Terminate the debugger.
 2. Reset the target device.
 - Check the status of memory read protection (the memory must be read-protected).
 1. Start the debugger alone (by executing the “./GNU17/gdb.exe” file).
 2. From the console of the debugger, execute the “target icd usb” command and establish a connection to the target device.
 3. In the Memory window of the debugger, dump the data of the read-protected memory area.
 4. Make sure that the read-protected memory area has data 0x0000.

3. CHECKING THE FLASH MEMORY PROTECT SETTINGS

3.2 Data Write Protection

If memory is write-protected, no data can be written to the area and its sectors are disabled to delete. To check the memory write protection, dump the data of the protected bit area and make sure that Protect bits are written as Figure 5.



The screenshot shows a window titled "Memory" with a table of memory addresses and their corresponding data. The address 0x27ff0 is selected, and its data is 0xffff. The data at address 0x27ff0 is highlighted with a red box, showing the value 0xff00. The table is as follows:

Address	0	2	4	6	8	A	C	E	ASCII
0x27ff0	0xffff	0xffff	0xffff	0xffff	0xffff	0xffff	0xff00	0xffff
0x28000	0x0000							
0x28010	0x0000							
0x28020	0x0000							
0x28030	0x0000							
0x28040	0x0000							
0x28050	0x0000							
0x28060	0x0000							
0x28070	0x0000							
0x28080	0x0000							
0x28090	0x0000							

Figure 5 Checkout Example of Memory Write Protection write

If the attempted data is written in Protect bits as shown in Figure 5, the memory is write-protected correctly.

4. RELEASING THE FLASH MEMORY PROTECTION

4. RELEASING THE FLASH MEMORY PROTECTION

The flash memory protect function can be released by chip erasing.

Chip erasing means the clearing of flash memory area. It can be set using the command file.

If the Flash Memory Protect function is set, it is made effective only when the debugger is started and when the CPU hardware is reset.

Similarly, the Flash Memory Protect function is released only when the chip erasing is made and when the target hardware is reset.

If this protection is not released yet, an error occurs.

To release the Flash Memory Protect function, specify “c17 fls 0x8000 0 0” and then “c17 rstt” in the command file.

Figure 6 gives a typical example of command file.

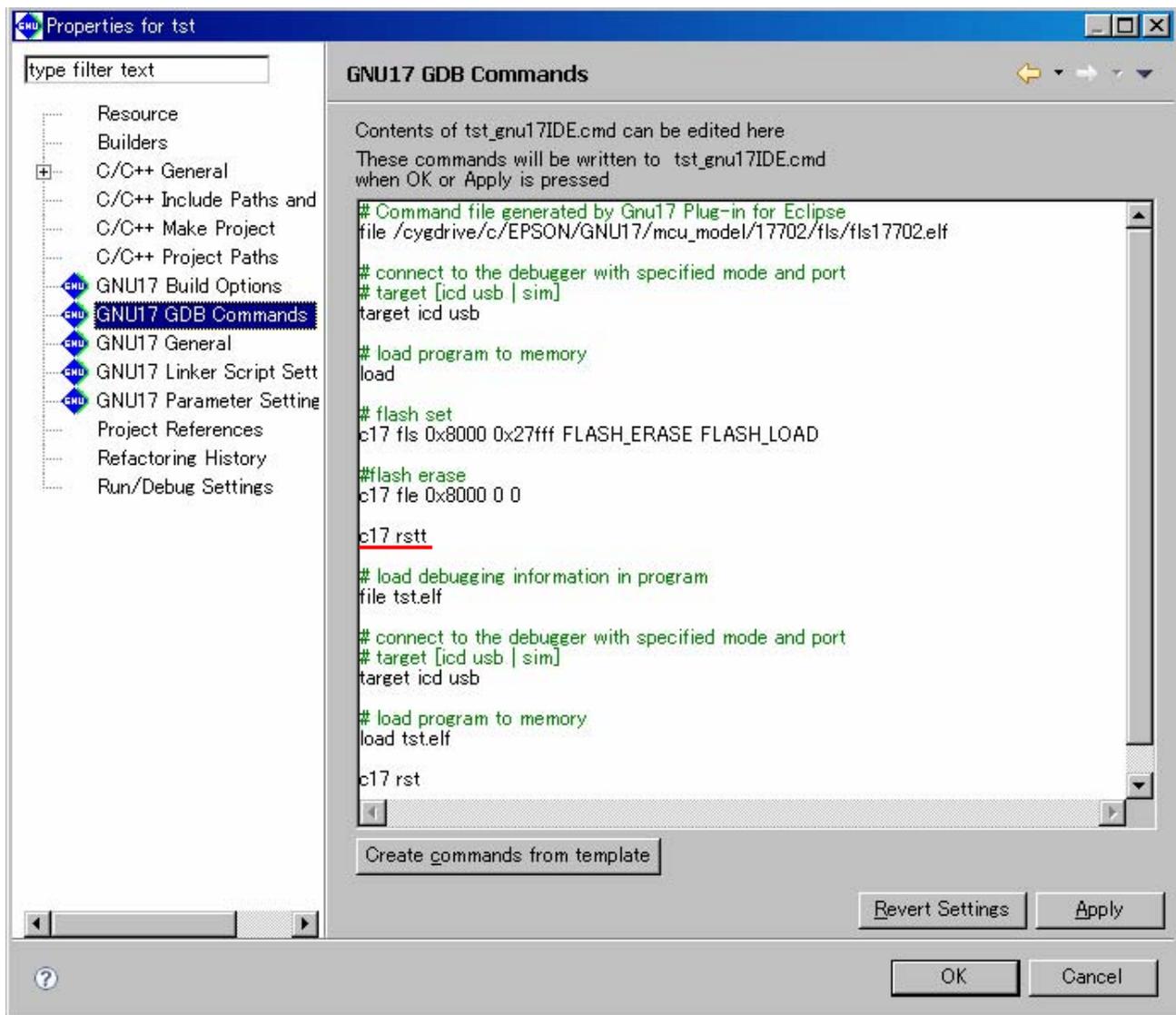


Figure 6 Example Command File to Release Flash Memory Protect Function

4. RELEASING THE FLASH MEMORY PROTECTION

As shown in Figure 6, the Flash Memory Protect function is released when the command file is set and the program is compiled.

If the RESET line is disconnected from the extended connector port of user target device, clear the memory area by chip erasing as follows.

1. Delete the “# load debugging information in program” and subsequent lines from the example command file, as shown in Figure 7.

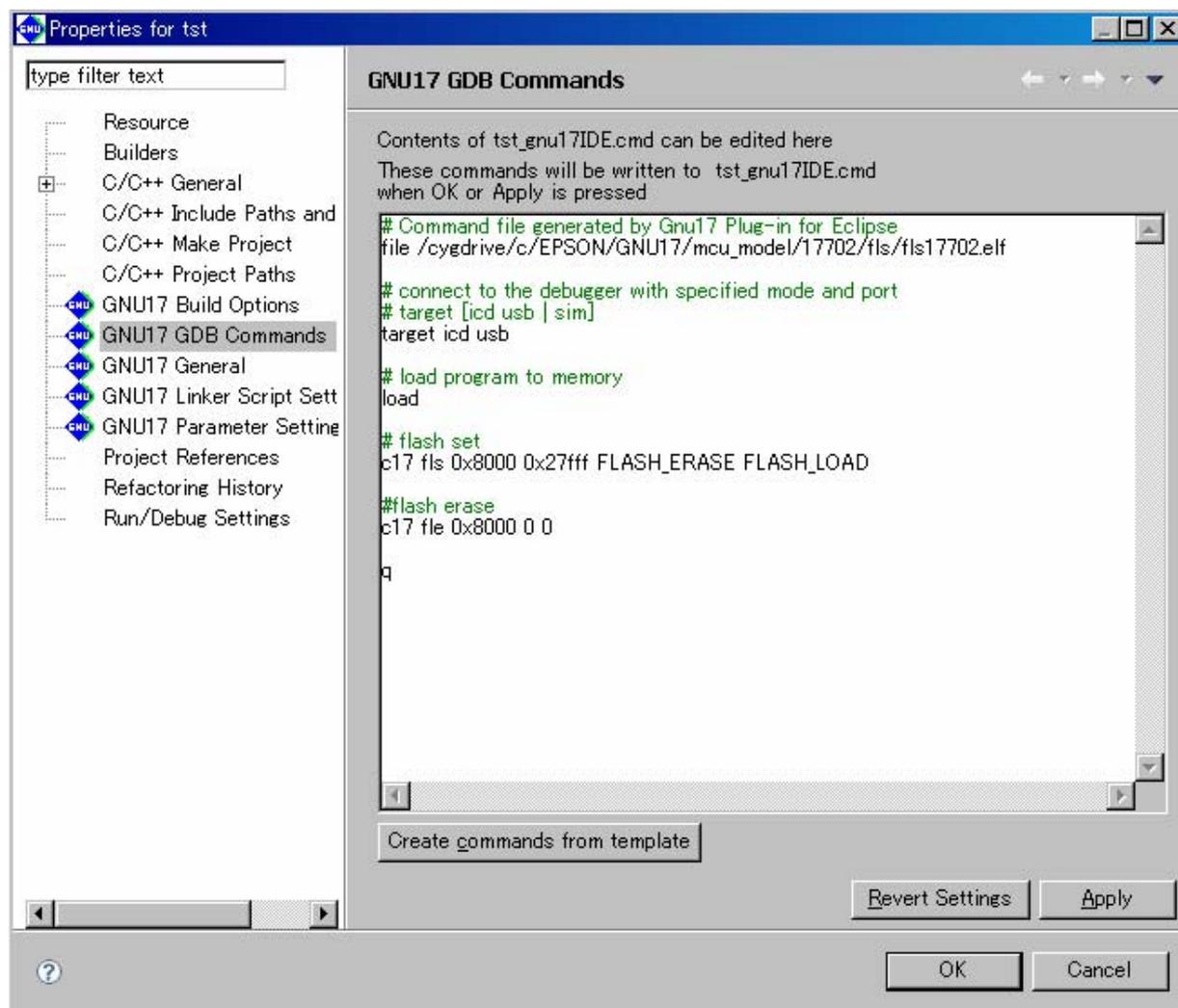


Figure 7 Example Command File for Chip Erasing of User Target Device

2. Start the debugger. (This debugger will terminate automatically.)
3. Reset the target hardware.

The Flash Memory Protect function has been released. Build the command file by returning it to the original status, and start the debugger by selecting [Run] → [External Tools] options.

Also, the user can release the Flash Memory Protect function by erasing the sector that contains the Protect bits. Note that this function can be released in the User mode only (that is, when the debugger is disconnected). If the sector having the Protect bits is write-protected, this function cannot be released.

5. NOTES

5. NOTES

- A sector having “.vector”, “.data” or “.rodata” section must not be read-protected.
- A mask ROM product is not affected by this function settings even if the mask ROM is write-protected.
- Without releasing the memory read protection and a program is written again, a verify error occurs during writing. This writing fails.If the flash memory is read-protected, it must be released. Then, the user can write a program in this memory.
- If a memory area is read-protected, data of this area cannot be copied to the built-in ROM, RAM and other internal storage devices.

AMERICA

EPSON ELECTRONICS AMERICA, INC.

2580 Orchard Parkway,
San Jose, CA 95131, USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH

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

ASIA

EPSON (CHINA) CO., LTD.

7F, Jinbao Bldg., No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, CHINA
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH

7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

SHENZHEN BRANCH

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

EPSON HONG KONG LTD.

20/F, Harbour Centre, 25 Harbour Road,
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.

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

EPSON SINGAPORE PTE., LTD.

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

SEIKO EPSON CORP.**KOREA OFFICE**

50F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

SEIKO EPSON CORP.**SEMICONDUCTOR OPERATIONS DIVISION****IC Sales Dept.****IC International Sales Group**

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