Contents

Chapter 1. Target Devices ........................................................................................................................ 2
Chapter 2. User's Manuals........................................................................................................................ 3
Chapter 3. Key Word for Uninstallation..................................................................................................... 4
Chapter 4. Changes .................................................................................................................................. 5
  4.1 Changes of CA78K0.......................................................................................................................... 5
    4.1.1 Improved Translation Limit ............................................................................................................. 5
    4.1.2 Improved #asm statements ............................................................................................................ 5
    4.1.3 Removal of restriction ..................................................................................................................... 5
Chapter 5. Cautions................................................................................................................................... 6
  5.1 Caution for the Memory bank relocation support tool ..................................................................... 6
Chapter 6. Restrictions.............................................................................................................................. 7
  6.1 Restrictions for the CA78K0.............................................................................................................. 7
    6.1.1 List of restrictions for the CA78K0 ................................................................................................. 7
    6.1.2 Restriction Details for the CA78K0 ................................................................................................. 8
  6.2 Restrictions for the Memory Bank Relocation Support Tool ........................................................... 14
    6.2.1 List of restrictions for the Memory Bank Relocation Support Tool .............................................. 14
    6.2.2 Detail of restrictions for the Memory Bank Relocation Support Tool ....................................... 15
Chapter 1. Target Devices

The target devices supported by the CA78K0 are listed on the Website.
Please see this URL.
CubeSuite+ Product Page:

http://www.renesas.com/cubesuite+
Chapter 2. User's Manuals

Please read the following user's manuals together with this document.

<table>
<thead>
<tr>
<th>Manual Name</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>CubeSuite+ V1.03.00 78K0 Coding Edition</td>
<td>R20UT2141EJ0100</td>
</tr>
<tr>
<td>CubeSuite+ V1.01.00 78K0 Build Edition</td>
<td>R20UT0783EJ0100</td>
</tr>
<tr>
<td>CubeSuite+ V1.03.00 Message</td>
<td>R20UT2147EJ0100</td>
</tr>
</tbody>
</table>
Chapter 3. Key Word for Uninstallation

There are two ways to uninstall this product.

- Use the integrated uninstaller (uninstalls CubeSuite+)
- Use separate uninstaller (uninstalls this product only)

To use the separate uninstaller, select the following from the Control Panel:

- Add/Remove Programs (Windows XP)
- Programs and Features (Windows Vista, Windows 7)

Then select "CubeSuite+ CA78K0 V1.30".
Chapter 4. Changes

This chapter describes change of CA78K0.
There is a possibility that the code is changed by the following.

4.1 Changes of CA78K0
This section describes changes of CA78K0 from V1.21 to V1.30.

4.1.1 Improved Translation Limit

- Number of "case" labels for one "switch" statement. 257 → 1024
- Number of characters in one logical source line. 2048 → 32767
- Number of macro identifiers simultaneously defined in one translation unit. 32767 → 60000
- Number of members of a single structure or single union. 256 → 1024
- Macro nesting. 200 → 10000
- Number of nesting levels of an expression enclosed by parentheses in a complete expression. 32 → 1024

4.1.2 Improved #asm statements

Improved about the following #asm statements
(1) Make an external definition / an external reference declaration of the symbol besides CC78K0 management by "#asm" with C source,
(2) When the symbol name length who did the external definition / the external reference is 9 or more characters, generate OMF of unjust symbol information and it becomes an error.

4.1.3 Removal of restriction

No.77 Restriction for a conditional operator
Description: The code will be incorrect if the result of a conditional operator is a Boolean-type value.
(Example)
__boolean b1;
unsigned char uc1;
void func()
{
    b1 = (uc1 & 0x80) ? (__boolean)1 : (__boolean)0;
}
Chapter 5. Cautions

This section describes cautions for using CA78K0 V1.30.

5.1 Caution for the Memory bank relocation support tool

Description: In the re-link function, it may become an error if a memory bank relocation support tool is started to a flash area.
Chapter 6. Restrictions

This section describes the restrictions for the CA78K0.

6.1 Restrictions for the CA78K0

Below is a list of restrictions for the CA78K0 V 1.30

6.1.1 List of restrictions for the CA78K0

(1) List of restrictions for Assembler

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>An error occurs if a control statement is crossed in a structured assembly language description. (Only PM+ version)</td>
</tr>
<tr>
<td>5</td>
<td>The assembler performs illegal processing if the label receiving the effect of optimization is described in the saddr part when an EQU definition is performed for a bit symbol with the value saddr.bit.</td>
</tr>
</tbody>
</table>

(2) List of restrictions for Compiler

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Bit fields with type signed are handled as unsigned bit fields.</td>
</tr>
<tr>
<td>43</td>
<td>Output conversion on I/O functions in the standard libraries causes illegal behavior.</td>
</tr>
<tr>
<td>44</td>
<td>The size of the minimum value (-32768) of types int/short is 4.</td>
</tr>
<tr>
<td>47</td>
<td>If the parameter type and the type of the identifier in a function definition do not match, an error is output.</td>
</tr>
<tr>
<td>48</td>
<td>In an identifier list in a function definition, a parameter that is not declared is not handled as type int, and an error results.</td>
</tr>
<tr>
<td>49</td>
<td>The # operator cannot be expanded correctly.</td>
</tr>
</tbody>
</table>
6.1.2 Restriction Details for the CA78K0

Below are details for the limitations for CA78K0 V1.30.

(1) Restriction details for Assembler

No. 1  A multidimensional array without the size defined may exhibit incorrect behavior

[Description] If a control statement is divided or crossed by code between #ifdef and #endif, an error occurs if #ifdef is true.

Example:

```
switch(mode)
#ifdef stsw                  Between #ifdef and #else or #endif
  case 1:
    break
  #endif
  default:
    break
end
```

[Work-around] Nesting will not cause an error. Rewrite the source so that the scopes of the control statements do not cross.

Example:

```
#ifdef stsw                                     Between #ifdef and #else or #endif
switch(mode)
  case 1:                                     Between case and next case/default/ends
    break
  #else
    switch(mode)
      case 1:
        break
default:
      break
end
  #endif
```
No. 5  The assembler performs illegal processing if the label receiving the effect of optimization is described in the saddr part when an EQU definition is performed for a bit symbol with the value saddr.bit.

[Description]  The assembler performs illegal processing if the label receiving the effect of optimization is described in the saddr part when an EQU definition is performed for a bit symbol with the value saddr.bit.

Illegal processing is performed in the following cases.

(1) When saddr.bit is 0FD20H, path 1 of a label is outside the area, and path 2 is inside the area, an error is output in path 1 for the EQU definition line, but not in path 2. At this time, the object is created but it is incorrect.

(2) When saddr is 0FF1FH, path 1 of a label is inside the area, and path 2 is outside the area, no error is output in path 1 for the EQU definition line, while an error is output in path 2. The following assembly error will be output for a label that is defined after this EQU symbol has been referenced.

[F410 Phase error]

When this label is referenced, the object becomes incorrect.

[Work-around]  None.
(2) Restriction details for Compiler

No. 16  Bit fields with type `signed` are handled as unsigned bit fields.

[Description]  Bit fields with type `signed` are handled as unsigned bit fields.

[Workaround]  None.

No. 43  Output conversion on I/O functions in the standard libraries causes illegal behavior.

[Description]  When output conversion is performed for the `printf`, `sprintf`, `vprintf`, and `vssprintf` functions, operation will become illegal under the following conditions.

If precision is specified as ".2" for the `d`, `i`, `o`, `u`, `x` or `X` conversion specifier, the 0 flag will not be ignored.

```
Example:
#include <stdio.h>
void func()
{
    printf("%04.2d\n", 77);
}
```

Remark  Illegal operation: "0077"
Correct operation: " 77"

For the `g`, and `G` conversion specifiers, the result is "specified precision + 1".

```
Example:
#include <stdio.h>
void func()
{
    printf("%.2g", 12.3456789);
}
```

Remark  Illegal operation: "12.3"
Correct operation: "12"

[Workaround]  None.
No. 44   The size of the minimum value (-32768) of types int/short is 4

[Description] The size of the minimum value (-32768) of types int/short is 4.

Example:
```c
int x;
void func()
{
  x = sizeof(-32768);
}
```

Remark   Illegal operation: The value of x is 4
Correct operation: The value of x is 2

[Workaround] Write as (-32767-1).

No. 47   If the parameter type and the type of the identifier in a function definition do not match, an error is output.

[Description] Because argument promotion is not performed for the type of an identifier in a function definition, the parameter type and the type of the identifier in the function definition do not match, thus causing the E0747 error.

Example:
```c
int fn_char(int);
int fn_char(char)
char c;
{
  return 98;
}
```

[Workaround] Make sure that the type of the parameter matches that of the identifier in the function definition.
No. 48 In an identifier list in a function definition, a parameter that is not declared is not handled as type `int`, and an error results.

[Description] In an identifier list in a function definition, a parameter that is not declared is not handled as type `int`, thus causing the E0706 error.

*Example:*

```c
void func(x1, x2, f, x3, lp, fp)
  int (*fp)( );
  long *lp;
  float f;
{
  }
```

[Workaround] Declare all parameters in a function definition.
No. 49  The # operator cannot be expanded correctly.

[Description]  Expansion will not be performed correctly under either of the following conditions.

1. [“”] cannot be expanded correctly with the # operator, causing a compile error.

   Example for condition 1:

```c
#include <string.h>
#define str( a) (# a)
int x;
void func()
{
    if (strcmp(str(""), "\"")) == 0) x++;
}
```

   **Remark:** Illegal operation: Compile error
   Correct operation: if (strcmp( "\""), "\"")) == 0) x++;

2. Macros that contain a # operator and a nested structure cannot be expanded correctly.

   Example for condition 2:

```c
#define str(a) #a
#define xstr(a) str(a)
#define EXP 1
char *p;
void func()
{
    p = xstr(12EEXP);
}
```

   **Remark:** Illegal operation: "p = ("12E1") ;"
   Correct operation: "p = ("12EEXP") ;"

[Workaround]  None.
6.2 Restrictions for the Memory Bank Relocation Support Tool

Below is a list of restrictions for the Memory Bank Relocation Support Tool.

6.2.1 List of restrictions for the Memory Bank Relocation Support Tool

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Restriction relating to <code>__flashf</code> function</td>
</tr>
<tr>
<td>2</td>
<td>Restriction relating to specifying options.</td>
</tr>
<tr>
<td>3</td>
<td>Restriction relating to link directives.</td>
</tr>
<tr>
<td>4</td>
<td>Restriction relating to <code>callt</code> area.</td>
</tr>
</tbody>
</table>
6.2.2 Detail of restrictions for the Memory Bank Relocation Support Tool

See below for details of restrictions for the Memory Bank Relocation Support Tool.

No. 1 Restriction relating to __flashf function

[Description] The __flashf function must be allocated to the common area, but is relocated to an area other than the common area because the Memory Bank Relocation Support Tool cannot identify the __flashf function. Therefore, an error may occur during build.

[Workaround] Open the C source file properties for the file that includes the __flashf function, and from the [Build Properties] tab, under the [Memory Bank] category, set [Select common/bank Area] to "Common area".

No. 2 Restriction relating to specifying options.

[Description] The Memory Bank Allocation Support Tool does not analyze the options specified in the [Other options] box in the Compiler/Assembler/Linker Options dialog boxes.

[Workaround] Do not input options into the [Other options] box if they can be specified in other areas in the option dialog boxes.

No. 3 Restriction relating to link directives.

[Description] The link directive file is not supported. The Memory Bank Allocation Support Tool relocates C source files based on the memory area defined in the device file. (One example of this limitation is that the relocation of const data, initialization data, and the like outside of functions is not supported if it is allocated to the bank area.)

[Workaround] From the [Memory Bank Options] tab, from the [Margin] category, adjust the free space in each area.

No. 4 Restriction relating to callt area.

[Description] If the section name of @@CALT is changed as a result of a #pragma section specification or quasi directive, the Memory Bank Relocation Support Tool will output an illegal value to the reference information file as the reference count of a function called via a callt area.

[Workaround] Do not change the section name of @@CALT.

All trademarks and registered trademarks are the property of their respective owners.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software or information.

2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.

3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You should not alter, modify, copy or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.

   "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment, and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implants etc.), or may cause serious property damage (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.

6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.

7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain-use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunctions prevention, appropriate treatment for aging-degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.

8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your non-compliance with applicable laws and regulations.

9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems where manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.

11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

[Note 1] “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

[Note 2] “Renesas Electronics products” means any product developed or manufactured by or for Renesas Electronics.