# CubeSuite+ 78K0 Compiler CA78K0 V1.21
Release Note

R20UT0879EJ0100
Rev.1.00
14 Oct, 2011

## Contents

# 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.

| Manual Name | Document Number |
|---|---|
| CubeSuite+ V1.01.00 78K0 Coding Edition | R20UT0782EJ0100 |
| CubeSuite+ V1.01.00 78K0 Build Edition | R20UT0783EJ0100 |
| CubeSuite+ V1.01.00 Message | R20UT0736EJ0100 |

# 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 V1.21".

# Chapter 4. Changes

This chapter describes change from V1.20 to V1.21.

There is a possibility that the code is changed by the following.

## 4.1 Caution eliminated

The restriction described below has been eliminated.

Caution for error of the list converter.

Description: The path name length of the “Bin” folder that ‘**.exe*’ for CA78K0 in CubeSuite+ is located in should be within 67 characters. If the list converter is executed, it becomes an error.

# Chapter 5. Cautions

This section describes cautions for using CA78K0 V1.21.

## 5.1 Caution for the Memory bank relocation support tool

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 1.21

### 6.1.1 List of restrictions for the CA78K0

#### (1) List of restrictions for Assembler

| No. | **Description** |
|---|---|
| 1 | An error occurs if a control statement is crossed in a structured assembly language description.(Only PM+ version) |
| 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*. |

#### (2) List of restrictions for Compiler

| No. | **Description** |
|---|---|
| 16 | Bit fields with type *signed* are handled as unsigned bit fields. |
| 43 | Output conversion on I/O functions in the standard libraries causes illegal behavior. |
| 44 | The size of the minimum value (-32768) of types *int/short* is 4. |
| 47 | If the parameter type and the type of the identifier in a function definition do not match, an error is output. |
| 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. |
| 49 | The # operator cannot be expanded correctly. |

### 6.1.2 Restriction Details for the CA78K0

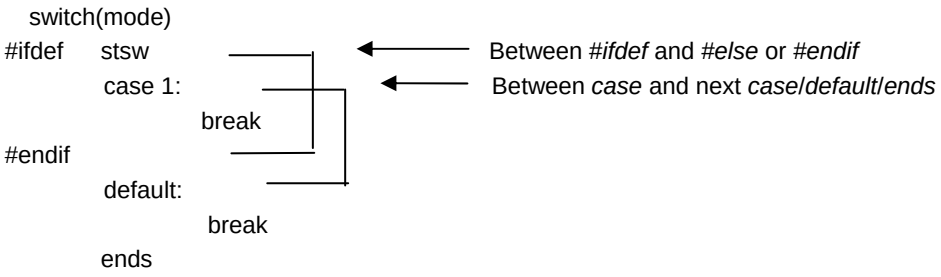
Below are details for the limitations for CA78K0 V1.21.

#### (1) Restriction details for Assemblert

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

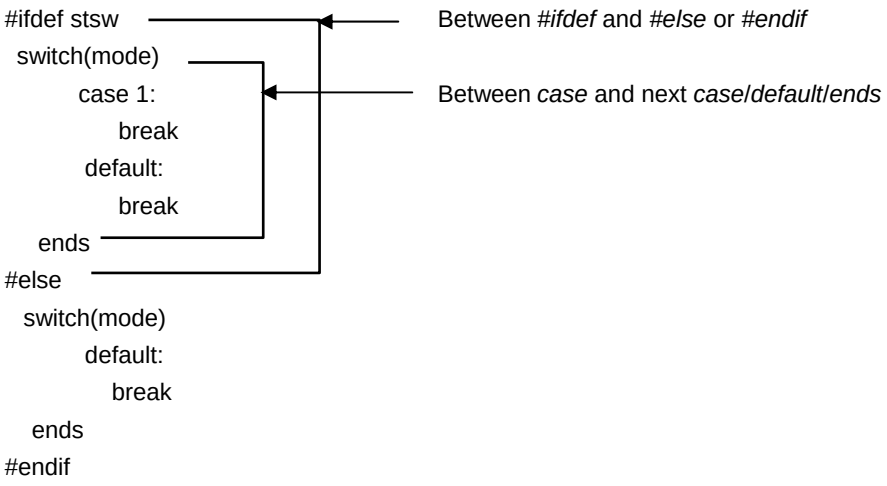
**[Description]** If a control statement is divided or crossed by code between *#ifdef* and *#endif*, an error occurs if *#ifdef* is true.

*Example:*




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

*Example:*



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 *vsprintf* 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:*

```
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:*

```
int  fn_char(int);
int  fn_char(c)
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:*

```
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:*

```
#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:*

```
#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

| No. | Description |
|---|---|
| 1 | Restriction relating to *__flashf* function |
| 2 | Restriction relating to specifying options. |
| 3 | Restriction relating to link directives. |
| 4 | Restriction relating to *callt* area. |

### 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*.

# Chapter 7. Changes in User's Manual

This section describes errata in CubeSuite+ documentation. The same content is also contained in the Help file, and should be replaced by this content.

## 7.1 Modifications in Build

### 7.1.1 Added description about [Memory Model] category of the [Compile Options] tab

**[Location]** Page 189 → [Output object for flash]

**[After addition]** Select whether to output the object for flash.

This corresponds to the -zf option of the compiler.

**If [Yes (-zf)] is selected**

- **On the [Compiler Options] tab, in the [Startup] category, if the [Use standard startup] property is not set to [No], it is set to [Yes (for Flash area)].**
- **On the [Link Options] tab, in the [Device] category, set the [Set Flash start address] property to [No].**

# Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
   - "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.
   - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
   - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 them 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 malfunction 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 system manufactured by you.
10. 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 noncompliance with applicable laws and regulations.
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 product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

**SALES OFFICES** Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.


**Renesas Electronics America Inc.**
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

**Renesas Electronics Europe GmbH**
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

**Renesas Electronics Malaysia Sdn.Bhd.**
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141





Colophon 1.1