Thank you for using the RL78 Family Data Flash Library Type04 Ver.1.05 Package Ver.2.00. This document contains notes and points for caution on using the Data Flash Library Type04 Package Ver.2.00. Please read this document before use.

Contents

Chapter 1  Target Product............................................................................................................2
Chapter 2  User’s Manual ........................................................................................................2
Chapter 3  Revisions..................................................................................................................2
Chapter 4  Supported Tools.......................................................................................................3
Chapter 5  Installation ...............................................................................................................3
5.1  Installation....................................................................................................................3
5.2  Uninstallation ................................................................................................................3
5.3  File Configuration ........................................................................................................4
Chapter 6  How To Build a Program ...................................................................................................5
6.1  Software to be used...........................................................................................................5
6.2  Building using CS+(former CubeSuite+) ...................................................................................5
   6.2.1  Building a C program................................................................................................5
   6.2.2  Building an assembly-language program.........................................................................8
6.3  Notes at Build ................................................................................................................11
   6.3.1  When the CA78K0R Compiler is Used .................................................................11
   6.3.2  When the CC-RL Compiler is Used .............................................................................12
Chapter 7  How To Debug a Program....................................................................................................13
7.1  Notes on Debugging........................................................................................................13
Chapter 8  Sample Program .................................................................................................................14
8.1  Initial Settings of the Sample Program .....................................................................................14
8.2  Settings of Option byte and On-Chip Debugging ....................................................................15
8.3  Compilation Switch for the C-Language Sample Program .....................................................16
8.4  Defining the Internal RAM Area...............................................................................................17
   8.4.1  When the CA78K0R Compiler is Used .........................................................................17
   8.4.2  When the CC-RL Compiler is Used .............................................................................18
Chapter 1  Target Product

The Data Flash Library Type04 Ver.1.05 Package Ver.2.00 is an integrated package of "Data Flash Library Type04 Ver.1.05B for the CA78K0R Compiler for the RL78 Family (JP_R_FDL_RL78_T04_V1.05_B_E.zip)" and "Data Flash Library Type04 Ver.1.05A for the CC-RL Compiler for the RL78 Family(JP_R_FDL_RL78_T04_V1.05_CCRL_A_E.zip)". Also, The Data Flash Library Type04 Package Ver.2.00 was changed to install files using the by an installer.

The following shows the target products for this release note.

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Ver.</th>
<th>Installer File Name</th>
<th>Ver.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Flash Library Type04 for CA78K0R Compiler</td>
<td>V1.05</td>
<td>RENESAS_RL78_FDL_T04_2V00.exe</td>
<td>V2.00</td>
</tr>
<tr>
<td>Compiler for the RL78 Family</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Flash Library Type04 for CC-RL Compiler</td>
<td>V1.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compiler for the RL78 Family</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 2  User's Manual

The following user's manual is available for this version.

<table>
<thead>
<tr>
<th>Title</th>
<th>Document Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL78 Family Data Flash Library Type04 User's Manual</td>
<td>R01US0049EJ0106</td>
</tr>
</tbody>
</table>

Chapter 3  Revisions

The revised contents of this version are as follows:

<table>
<thead>
<tr>
<th>No.</th>
<th>Package Ver.</th>
<th>Target</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V.2.00</td>
<td>Library V1.05 for CA78K0R Compiler</td>
<td>There are no changes in the library from the previous ZIP File (JP_R_FDL_RL78_T04_V1.05_B_E.zip).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Library V1.05 for CC-RL Compiler</td>
<td>There are no changes in the library from the previous ZIP File (JP_R_FDL_RL78_T04_V1.05_CCRL_A_E.zip).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User's manual</td>
<td>Revised from Rev.1.05 to Rev.1.06. For details on the corrections to the user's manual in response to the revision, refer to the revision history of the user's manual.</td>
</tr>
</tbody>
</table>
Chapter 4  Supported Tools

Use the following tool version when using the Data Flash Library Type04 Ver.1.05.

<table>
<thead>
<tr>
<th>Target library</th>
<th>Tool Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library for CA78K0R Compiler</td>
<td>Integrated development environment CubeSuite+</td>
<td>V1.00.00 or later</td>
</tr>
<tr>
<td></td>
<td>Integrated development environment CS+</td>
<td>V3.00.00 or later</td>
</tr>
<tr>
<td>Library for CC-RL Compiler</td>
<td>Integrated development environment CS+</td>
<td>V3.01.00 or later</td>
</tr>
</tbody>
</table>

Chapter 5  Installation

This chapter describes how to install and uninstall the Data Flash Library Type04 Package Ver.2.00.

5.1  Installation

Install the Data Flash Library Type04 by using the following procedure:

(1) Start Windows.
(2) Decompress the file that contains the Data Flash Library Type04 Package and run of the installer.
(3) Select "Asia/Oceania - English" from the drop-down list.
(4) Click on the "OK" button to proceed installation according to the instructions of the installer.

5.2  Uninstallation

Uninstall the Data Flash Library Type04 by using the following procedure:

(1) Start Windows.
(2) Delete the folder that contains the Data Flash Library Type04 files and was placed at the location chosen by the user.
5.3 File Configuration

The file organization after this library is installed is shown below.

<table>
<thead>
<tr>
<th>Installation folder</th>
</tr>
</thead>
<tbody>
<tr>
<td>r20ut0749ejxxxx.pdf  : Release Note (this document)</td>
</tr>
<tr>
<td>suppor.txt           : Support information file for the data flash library</td>
</tr>
<tr>
<td>CA78K0R_110 or CC-RL_100</td>
</tr>
<tr>
<td>lib</td>
</tr>
<tr>
<td>pfdl.lib             : Data flash library</td>
</tr>
<tr>
<td>pfdl.h               : Include file for C</td>
</tr>
<tr>
<td>pfdl.inc             : Include file for assembler</td>
</tr>
<tr>
<td>pfdl_types.h         : Include file that specifies definitions for C</td>
</tr>
<tr>
<td>Sample</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>r_pfdl_sample_c.c    : Sample program file for C Note2</td>
</tr>
<tr>
<td>r_pfdl_sample_c.dr   : link directive file for Sample program Note2</td>
</tr>
</tbody>
</table>

Notes:
1. x indicates the omitted numerals in version or revision numbers.
2. To use the sample program, the program file (*.c) and link directive file (*.dr) should be embedded together.
   The link directive file (*.dr) that specifies the link information is not provided with the library for CC-RL.
   The link information for the sample program for CC-RL should be specified through the link setting window on CS+.
Chapter 6  How To Build a Program

This chapter describes how to build a program using the Data Flash Library Type04.

6.1  Software to be used

The following integrated development environment is necessary for building programs using the Data Flash Library Type04.

- Integrated development environment CS+ V3.00.00 or later for CA78K0R compiler
- Integrated development environment CubeSuite+ V1.00.00 or later for CA78K0R compiler
- Integrated development environment CS+ V3.01.00 or later for CC-RL compiler

6.2  Building using CS+(former CubeSuite+)

This section describes how to include the Data Flash Library Type04 in a user-created program and build the user program by using CS+.

6.2.1  Building a C program

(1) Creating a project and specifying the source file

Create a project by using CS+. In the Project Tree window displayed on the left, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-1).

Next, click the Files of type drop-down list to display a list of the file types. Select C source file (*.c), and then register the user-created program as the source file.

![Figure 6-1. Registering the User Program File](image)
(2) Specifying the include file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-2). Next, click the Files of type drop-down list to display a list of the file types. Select Header file (*.h;*.inc), and then register the header files (pfdl.h, pfdl_types.h) of the data flash library.

![Figure 6-2. Registering the Include Files](image)

(3) Specifying the library file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-3). Next, click the Files of type drop-down list to display a list of the file types. Select Library file (*.lib), and then register the data flash library file (pfdl.lib).
(4) Specifying the link directive file (only when the CA78K0R compiler is used)

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-4).

Next, click the Files of type drop-down list to display a list of the file types. Select Link directive file (*.dr;*.dir), and then register the link directive file that has the same name as the user-created program.

(5) Building

On the CS+ Build menu, click Build Project to build the project.
6.2.2 Building an assembly-language program

(1) Creating a project and specifying the source file
Create a project by using CS+. In the Project Tree window displayed on the left, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-5).
Next, click the Files of type drop-down list to display a list of the file types. Select Assemble file (*.asm), and then register the user-created program as the source file.

(2) Specifying the include file
In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-6).
Next, click the Files of type drop-down list to display a list of the file types. Select Header file (*.h;*.inc), and then register the header file (pfdl.inc) of the data flash library.
(3) Specifying the library file

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-7). Next, click the Files of type drop-down list to display a list of the file types. Select Library file (*.lib), and then register the data flash library file (pfdl.lib).
(4) Specifying the link directive file (only when the CA78K0R compiler is used)

In the CS+ Project Tree window, right-click the File node, click Add, and then click Add File. The Add Existing File dialog box is displayed (as shown in Figure 6-8). Next, click the Files of type drop-down list to display a list of the file types. Select Link directive file (*.dr;*.dir), and then register the link directive file that has the same name as the user-created program.

![Figure 6-8. Registering the Link Directive File](image)

(5) Removing the automatically generated files (only when the CC-RL compiler is used)

CS+ for the CC-RL compiler automatically generates some files under the File node in the Project Tree window. Among these, the processing of the "main.c" and "hdwinit.asm" files is included in the data flash library. Therefore, remove these two files from the target of the build process.

![Figure 6-9. Removing the Automatically Generated Files](image)

(5) Building

From the CS+ Build menu, click Build Project to build the project.
6.3 Notes at Build

6.3.1 When the CA78K0R Compiler is Used

(1) When the on-chip debugging function is in use

After the on-chip debugging function is enabled in the CS+, building a program generates the following type of error.

RA78K0R error E3212: Default segment can't allocate to memory - ignored
Segment '?OCDROM' at xxxxxH-200H

This error occurs when the segment for the monitor area (OCDROM) used by the on-chip debugging function cannot be allocated. Therefore, to avoid this error, add the following code to the link directive file (*.dr) embedded in the project and prepare a separate area for allocating the segment.

MEMORY OCD_ROM : ( 0xxxxxH, 00200H )

Remarks: 1. xxxxx indicates the start address at which the error has occurred.
2. The area name "OCD_ROM" is an example of the notation.
6.3.2 When the CC-RL Compiler is Used

(1) When the on-chip debugging function is in use

After the on-chip debugging function is enabled in the CS+, building a program may generate the following type of error.

```
E0562321:Section ".monitor2" overlaps section "xxxxx"
```

This error occurs when the section for the monitor area (OCDROM) used by the on-chip debugging function cannot be allocated. Therefore, to avoid this error, right click the CC-RL (Build Tool) node (1) in the CS+ Project Tree window, select Property to open the CC-RL Property panel (2), and select the Link Options tab (3). In the Section category (4), modify the setting for Section start address (5) so that no other areas overlap the area where the section for the on-chip debugger monitor is allocated (monitor2: the initial address range is 0xFE00 to 0xFFFF in R5F100LE). (See Figure 6-10.)


Remark 1. xxxxx: Indicates the section name.
Chapter 7   How To Debug a Program

For details on how to perform debugging by using IECUBE or the on-chip debugging emulator E1, E2, E2 emulator Lite or E20, see the following document:

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
</table>

Note: You can download this document from the "CS+ Integrated Development Environment" page of the Renesas Electronics website.

7.1 Notes on Debugging

(1) When a command of the Data Flash Library Type04 Ver.1.05 is executed using the on-chip debugging emulator E1, E2, E2 emulator Lite or E20 in a CS+ whose version is earlier than Ver.1.01, do not execute a break until completion of the sequencer has been confirmed. Otherwise, the sequencer does not operate correctly.

(2) The data flash library cannot be debugged by a simulator. To perform debugging, either use the on-chip debugging function of the RL78 microcontroller or prepare the IECUBE.
Chapter 8    Sample Program

The attached sample program (r_pfdl_sample_c.c) is provided to enable the usage method of the Data Flash Library Type04 to be easily confirmed on the QB-R5F100LE-TB boards with R5F100LEA (RL78/G13) as the target microcontrollers. The sample program is just a reference example and the user program does not have to be created to match the sample program. The sample program should be used as a simple program to confirm operation.

The link directive file (r_pfdl_sample_c.dr) for the sample program for the CA78K0R compiler has a purpose to specify that a stack or data buffer used by the sample program is not allocated to an area where allocation is prohibited Note1. When using the sample program, this file should also be embedded with the sample program. Note2, 3

The sample program for the CC-RL compiler does not need the link directive file (r_pfdl_sample_c.dr), but sections should be allocated appropriately in the Section category on the Link Options tabbed page in the CS+ window so that a stack or data buffer used by the sample program is not allocated to an area where allocation is prohibited Note1.

Notes:
1. For details, refer to chapter 2.2 "Software Environment" in the user’s manual.
2. In the supplied link directive file, the RAM area size is set to 2 Kbytes. Even when the target microcontroller has 2 Kbytes or larger RAM, the sample program (r_pfdl_sample_c.c) can be used for building without modifying the defined area setting.
3. The data in usage may be placed at an unintended area depending on how the environment in use or the program is changed. After an execution module is generated, the map file and allocation state of programs or data must be confirmed. For the definition method and allocation conditions of each code or data, refer to the user’s manual of the CS+.

8.1    Initial Settings of the Sample Program

The sample program operates with the following initial settings. When these settings need to be changed, modify the sample program.

- CPU operating frequency: High-speed on-chip oscillator 32 MHz
- Voltage mode: High-speed mode
8.2 Settings of Option byte and On-Chip Debugging

When performing on-chip debug, set "Set enable/disable on-chip debug by link option" to "Yes" and specify "84" for "Option byte values for OCD". For the CC-RL compiler, set "Set debug monitor area" to "Yes".

The sample program normally operates by setting the high-speed on-chip oscillator at 32 MHz.

After setting "Set user option byte" to "Yes" on the "Link Options" tabbed page, specify "xxxxE8" for "User option byte value" and set the high-speed on-chip oscillator at 32 MHz.

Figure 8-1 (a) Setting of Option byte (CA78K0R Compiler)

Figure 8-1 (b) Setting of Option byte (CC-RL Compiler)
8.3 Compilation Switch for the C-Language Sample Program

The sample program has a compilation switch as shown below. This compilation switch is used to turn on the LED to confirm operation on the QB-R5F100LE-TB board. To use this, modify "#if 0" to "#if 1" so that the #define declaration for the target CPU board becomes valid.

```
#ifndef __QB_R5F100LE_TB__ Can be modified to #if 1
#define __QB_R5F100LE_TB__ when QB-R5F100LE-TB is used

#else
#define __NON_TARGET__
#endif
```

/* Symbol for program switch of sample program */
/*************************************************************************/
/* Can be set when QB-R5F100LE-TB is used alone */
#if 0 /* Other boards */
#else
#define __NON_TARGET__
#endif
8.4 Defining the Internal RAM Area

8.4.1 When the CA78K0R Compiler is Used

When the CA78K0R compiler is used, the entire internal RAM area is automatically defined as an area with the name "RAM" in the initial state. Unless otherwise stated in the link directive file, the stack and data buffers are to be allocated to this area\(^\text{Note}\). However, in this case, the stack and data buffers would be allocated by default to an area (FFE20H to FFEFFH in self-RAM) for which use by the data flash library is prohibited, so the program may not run correctly.

In the attached link directive file for the sample program, as a solution, re-define the area with the name "RAM" so that it does not include the above area, ensuring that stack and so on are not allocated to the area for which usage is prohibited.

```
MEMORY RAM       : (0FEF88H, 000E98H)
```

The above statement redefines the area with the name "RAM" to be the E98H bytes area starting from the address FEF88H (FFE88H to FFEF1H)\(^\text{Note}\). This prevents attempted use of the area which the data flash library is prohibited to use by excluding the prohibited portion from the area with the name "RAM".

However, if this is the only change setting that is explicitly made, the area from FFE20H to FFEFFH is also unusable for any other purpose. Accordingly, separately add the following definition. No particular restrictions apply to the name of this area.

```
MEMORY SADDR_RAM:(0FFE20H, 0000E0H)
```

If there is a self-RAM area, automatic allocation of variables to this area can be restricted by defining its range as an area with the name "SELFRAM".

```
MEMORY SELFRAM : (0FEF00H, 000088H)
```

An example of the settings for an RL78/G13 (the product with 4 Kbytes of RAM and 64 Kbytes of ROM) is given below.

```
; Define new memory entry for Self-RAM
;----------------------------------------------------------
MEMORY SELFRAM  : (0FEF00H, 000088H) 
; Redefined default data segment RAM
;----------------------------------------------------------
MEMORY RAM      : (0FEF88H, 000E98H) 
; Define new memory entry for saddr area
;----------------------------------------------------------
MEMORY RAM_SADDR: (0FFE20H, 0000E0H)
```

Note: The CA78K0R linker allocates data with a non-specified destination for allocation (segment types DSEG and BSEG) to the internal RAM area according to the re-allocation attribute of the data. Accordingly, specific data may not be allocated to the area with the name "RAM" in some situations.

For details on the methods of defining and allocating the individual categories of data, refer to the user’s manual for CS+.

Reference to the map file (*.map) generated at the time of building is required to confirm the state of allocation.
8.4.2 When the CC-RL Compiler is Used

(1) Adding the include path

In CS+ for the CC-RL compiler, no include path is specified in the initial state; the include paths for the header files used by the data flash library need to be added. The data flash library uses header files "pfdl.h", "pfdl_types.h", and "iodefine.h" (this file is automatically generated by CS+). Add the include path for each header file in the Additional include paths in the Preprocess category on the Compile Options tabbed page.

(2) Defining sections

When CS+ for the CC-RL compiler is used, the sections used for the ROM and RAM areas need to be defined. Sections can be defined in the Section category on the Link Options tabbed page in the CS+ window. When the Layout sections automatically property is set to No, select the Section start address property to open the Section Settings dialog box and add the sections necessary for the data flash library to the ROM area (Figure 8-2). (In this example, the PFDL_COD section that are necessary for operation of the sample program is added.)

Figure 8-2. Example of Section Settings for the data flash library (ROM Area)
(3) Allocating the Self-RAM Area

In the initial state of the section settings in CS+ for the CC-RL compiler, the user RAM area is allocated at the beginning of the internal RAM area (from address FEF00H for R5F100LEA, which is the target microcontroller of the sample program). However, in R5F100LEA, the data flash library uses the address range from 0xEF00 to 0xEF87 as the self-RAM area. Therefore, the user RAM area must be allocated outside this area. In this example, the user data start address 0xEF00 is changed to 0xEF88.

Figure 8-3. Example of Changing the User RAM Area Allocation (RAM Area)

Note: The sections including the user-specified sections are automatically re-allocated when the Layout sections automatically property is temporarily set to No, the user RAM allocation is changed, and then the property is again set to Yes. In this case, sections may be allocated to areas that are not specified by the user; that is, data may be placed in unintended areas. Be sure to refer to the map file to check if the software resources (especially RAM data) used by the data flash library are placed in relocatable areas.
All trademarks and registered trademarks are the property of their respective owners.