

RL78 ファミリ EEPROM エミュレーション・ライブラリ Pack02

R20UT3486JJ0100
Rev.1.00
2015.07.15

CA78K0R コンパイラ版と CC-RL コンパイラ版の差分

この資料では、CC-RL コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02 Ver.1.01(以降、『CC-RL 用 EEL Pack02 V1.01』と省略させていただきます。)をお使いいただく上での制限事項および注意事項等、および、CA78K0R コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02 Ver.1.01 (以降、『CA78K0R 用 EEL Pack02 V1.01』)との差分情報等について記載しております。ご使用の前に、必ずお読みくださいますようお願い申し上げます。

本資料と CA78K0R 用 EEL Pack02 V1.01 のユーザーズマニュアルをあわせて読むことで、CC-RL 用 EEL Pack02 V1.01 の機能を理解することができます。

目次

1. 注意事項.....	2
2. フラッシュ関数詳細.....	2
2.1 FDL_Init.....	3
2.2 FDL_Open.....	4
2.3 FDL_Close	4
2.4 EEL_Init.....	5
2.5 EEL_Open.....	5
2.6 EEL_Close	6
2.7 EEL_Execute.....	7
2.8 EEL_Handler	8
2.9 EEL_GetSpace.....	8
2.10 EEL_GetVersionString	9
3. EEPROM エミュレーション・ライブラリが使用する ROM/RAM サイズ	10

1. 注意事項

- RL78 ファミリ CC-RL コンパイラ用 EEPROM エミュレーション・ライブラリ Pack02 の各セグメント (FDL_CODE, FDL_CNST, EEL_CODE, EEL_CNST) は、64KB 境界を跨いで配置することができません。必ず、64KB 境界を跨がないように配置してください。
- ルネサス製コンパイラ CC-RL のアセンブラを使用する場合、16 進数の Prefix 表現 (0x..) と Suffix 表現 (.H) は混在できません。ユーザの環境に合わせて fdl.inc 内のシンボル定義を編集することで表現方法を指定してください。

fdl.inc

```
; __FDL_INC_BASE_NUMBER_SUFFIX .SET 1
```

シンボル“__FDL_INC_BASE_NUMBER_SUFFIX”を定義しない場合 (初期状態)、Prefix 表現が選択されます。

fdl.inc

```
__FDL_INC_BASE_NUMBER_SUFFIX .SET 1
```

シンボル“__FDL_INC_BASE_NUMBER_SUFFIX”を定義する場合、Suffix 表現が選択されます。

2. フラッシュ関数詳細

EEL の各関数機能の変更はありませんが、関数呼び出しの引数や型宣言が変更になっていますので、以降、各関数の変更内容を示します。

2.1 FDL_Init

(1) Function Prototype

RENESAS CA78K0R:

```
fdl_status_t __far FDL_Init(const __far fdl_descriptor_t* descriptor_pstr)
```

RENESAS CC-RL:

```
fdl_status_t __far FDL_Init(const __far fdl_descriptor_t* descriptor_pstr)
```

(2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	const __far fdl_descriptor_t* descriptor_pstr	AX(0-15), C(16-23)
RENESAS (CC-RL)	const __far fdl_descriptor_t* descriptor_pstr	DE(0-15), A(16-23)

The structure fdl_descriptor_t :

	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	<pre>typedef struct { fdl_u16 eel_pool_bytes_u16; fdl_u16 fdl_pool_bytes_u16; fdl_u16 fdl_delay_u16; fdl_u08 eel_pool_blocks_u08; fdl_u08 fdl_pool_blocks_u08; fdl_u08 fx_MHz_u08; fdl_u08 wide_voltage_mode_u08; } fdl_descriptor_t;</pre>	
RENESAS (CC-RL)	<pre>typedef struct { fdl_u16 eel_pool_bytes_u16; fdl_u16 fdl_pool_bytes_u16; fdl_u16 fdl_delay_u16; fdl_u08 eel_pool_blocks_u08; fdl_u08 fdl_pool_blocks_u08; fdl_u08 fx_MHz_u08; fdl_u08 wide_voltage_mode_u08; } fdl_descriptor_t;</pre>	

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	fdl_status_t	C
RENESAS (CC-RL)	fdl_status_t	A

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	AX,B
RENESAS (CC-RL)	X, BC, DE, HL

2.2 FDL_Open

(1) Function Prototype

RENESAS CA78K0R:

void __far FDL_Open(void)

RENESAS CC-RL:

void __far FDL_Open(void)

(2) Argument

None

(3) Return value

None

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	None
RENESAS (CC-RL)	AX

2.3 FDL_Close

(1) Function Prototype

RENESAS CA78K0R:

void __far FDL_Close(void)

RENESAS CC-RL:

void __far FDL_Close(void)

(2) Argument

None

(3) Return value

None

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	None
RENESAS (CC-RL)	C

2.4 EEL_Init

(1) Function Prototype

RENESAS CA78K0R:

eel_status_t __far EEL_Init(void);

RENESAS CC-RL:

eel_status_t __far EEL_Init(void);

(2) Argument

None

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	eel_status_t	C
RENESAS (CC-RL)	eel_status_t	A

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	None
RENESAS (CC-RL)	X, BC, D, HL

2.5 EEL_Open

(1) Function Prototype

RENESAS CA78K0R:

void __far EEL_Open(void)

RENESAS CC-RL:

void __far EEL_Open(void)

(2) Argument

None

(3) Return value

None

(4) Destructed Register

None

2.6 EEL_Close

(1) Function Prototype

RENESAS CA78K0R:

void __far EEL_Close(void)

RENESAS CC-RL:

void __far EEL_Close(void);

(2) Argument

None

(3) Return value

None

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	None
RENESAS (CC-RL)	A

2.7 EEL_Execute

(1) Function Prototype

RENESAS CA78K0R:

```
void __far EEL_Execute(__near eel_request_t* request_pstr)
```

RENESAS CC-RL:

```
void __far EEL_Execute(__near eel_request_t* request_pstr)
```

(2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	__near eel_request_t* request_pstr	AX(0-15)
RENESAS (CC-RL)	__near eel_request_t* request_pstr	AX(0-15)

The structure of request_pstr :

	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	typedef struct { __near eel_u08* address_pu08; eel_u08 identifier_u08; eel_command_t command_enus; eel_status_t status_enus; } eel_request_t;	; byte index within EEL-request structure ; ----- EEL_REQUEST_ADDR EQU (000H) EEL_REQUEST_IDENTIFIER EQU (002H) EEL_REQUEST_COMMAND EQU (003H) EEL_REQUEST_ERROR EQU (004H)
RENESAS (CC-RL)	typedef struct { __near eel_u08* address_pu08; eel_u08 identifier_u08; eel_command_t command_enus; eel_status_t status_enus; } eel_request_t;	; byte index within EEL-request structure ; ----- EEL_REQUEST_ADDR .EQU (000H) EEL_REQUEST_IDENTIFIER .EQU (002H) EEL_REQUEST_COMMAND .EQU (003H) EEL_REQUEST_ERROR .EQU (004H)

(3) Return value

None

(4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	AX
RENESAS (CC-RL)	AX, BC, DE, HL

2.8 EEL_Handler

- (1) Function Prototype
 RENESAS CA78K0R:
 void __far EEL_Handler(void)
 RENESAS CC-RL:
 void __far EEL_Handler(void);

- (2) Argument
 None

- (3) Return value
 None

- (4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	None
RENESAS (CC-RL)	AX, BC, DE, HL

2.9 EEL_GetSpace

- (1) Function Prototype
 RENESAS CA78K0R:
 eel_status_t __far EEL_GetSpace(__near eel_u16* space_pu16)
 RENESAS CC-RL:
 eel_status_t __far EEL_GetSpace(__near eel_u16* space_pu16)

- (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	__near eel_u16* space_pu16	AX(0-15)
RENESAS (CC-RL)	__near eel_u16* space_pu16	AX(0-15)

- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
RENESAS (CA)	eel_status_t	C
RENESAS (CC-RL)	eel_status_t	A

- (4) Destructed Registers

	<i>Destructed Registers</i>
RENESAS (CA)	AX
RENESAS (CC-RL)	X, C, HL

2.10 EEL_GetVersionString

(1) Function Prototype

RENESAS CA78K0R:

__far eel_u08* __far EEL_GetVersionString(void)

RENESAS CC-RL:

__far eel_u08* __far EEL_GetVersionString(void)

(2) Argument

None

(3) Return value

	Parameter types	
	C Language	Assembler Language
RENESAS (CA)	__far eel_u08*	BC(0-15), DE(16-31)
RENESAS (CC-RL)	__far eel_u08*	DE(0-15), A(16-23)

(4) Destructed Registers

None

- ・ライブラリのバージョン情報のフォーマット
各文字は ASCII コードです。

"NM..MTTTCVVVV"

"N" はライブラリ名を示します。

'E' = EEL

"M..M" は対象デバイスのファミリ名を示します。

'RL78' = RL78

"TTT" は Type 番号を示します。

'TXX' = TypeXX

"CCCCC"(5 文字) はコンパイラ情報(対象ツール)を示します。(CC-RL の場合 : "CCCCC"(6 文字))

'RXXXG' = RENESAS C Compiler **CA78K0R** VX.XX supports all memory models.

'LXXXXG' = RENESAS C Compiler CC-RL VX.XX.0X supports all memory models.

"VVVV" はライブラリバージョンを示します。

'VXXX' = Version X.XX(正式版)

例)

"ERL78T01**R**110GV100" は、

"EEL RL78 Type02 V1.00 for RENESAS C Compiler **CA78K0R** V1.10"を示します。

"ERL78T01**L**1000GV100" は、

"EEL RL78 Type02 V1.00 for RENESAS C Compiler **CC-RL** V1.00.00"を示します。

3. EEPROM エミュレーション・ライブラリが使用する ROM/RAM サイズ

CC-RL 用 EEL Pack02 V1.01 が使用するユーザ ROM のコード・サイズおよびフラッシュ関数のスタック・サイズを以下に示します。

表 3-1 EEPROM エミュレーション・ライブラリのコード・サイズ

	最大容量 (バイト)
ROM使用サイズ	3400
スタック・サイズ	64

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、
各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>