

---

## Differences between the CA78K0R compiler version and the CC-RL compiler version

---

This document contains notes and points for caution on using the Flash Self-Programming Library Type01 Ver.2.21 for the CC-RL compiler (hereinafter abbreviated as "FSL Type01 V2.21 for CC-RL") and specifications different from Flash Self-Programming Library Type01 Ver.2.20 for the CA78K0R compiler (hereinafter referred to as "FSL Type01 V2.20 for CA78K0R"). Please read this document before use.

You can understand the functionality of FSL Type01 V2.21 for CC-RL by reading this document and the user's manual of FSL Type01 V2.20 for CA78K0R.

## Contents

1. Notes.....	3
2. Details on Functions Related with Flash Memory .....	4
2.1 FSL_Init.....	4
2.2 FSL_PrepareFunctions .....	5
2.3 FSL_PrepareExtFunctions .....	5
2.4 FSL_ChangeInterruptTable.....	5
2.5 FSL_RestoreInterruptTable.....	6
2.6 FSL_Open .....	6
2.7 FSL_Close.....	6
2.8 FSL_BlankCheck .....	7
2.9 FSL_Erase .....	8
2.10 FSL_IVerify.....	9
2.11 FSL_Write .....	10
2.12 FSL_GetSecurityFlags .....	11
2.13 FSL_GetBootFlag .....	12
2.14 FSL_GetSwapState .....	13
2.15 FSL_GetBlockEndAddr.....	14
2.16 FSL_GetFlashShieldWindow .....	15
2.17 FSL_SetBlockEraseProtectFlag.....	16
2.18 FSL_SetWriteProtectFlag .....	16
2.19 FSL_SetBootClusterProtectFlag .....	17
2.20 FSL_InvertBootFlag .....	17
2.21 FSL_SetFlashShieldWindow.....	18
2.22 FSL_SwapBootCluster .....	19
2.23 FSL_SwapActiveBootCluster .....	19
2.24 FSL_ForceReset .....	20
2.25 FSL_StatusCheck .....	20
2.26 FSL_StandBy .....	21
2.27 FSL_WakeUp.....	21
2.28 FSL_GetVersionString .....	22
3. ROM and RAM Size Used by Flash self-programming library.....	23
4. Stack Size Used by Flash Functions.....	24

## 1. Notes

- Each segment (FSL\_FCD, FSL\_FECD, FSL\_RCD, FSL\_BCD, or FSL\_BECD) of the Flash Self-Programming Library for the CC-RL compiler for the RL78 family cannot be allocated to extend across the 64-Kbyte boundary. Be sure to allocate segments so that they do not extend across the 64-Kbyte boundary.
- When using an assembler of the CC-RL compiler from Renesas Electronics, the hexadecimal prefix representation (0x..) cannot be mixed together with the suffix representation (..H). Specify the representation method by editing the symbol definition in fsl.inc to match the user environment.

fsl.inc

`; __FSL_INC_BASE_NUMBER_SUFFIX .SET 1`

When symbol "`__FSL_INC_BASE_NUMBER_SUFFIX`" is not defined (initial state), the prefix representation will be selected.

fsl.inc

`__FSL_INC_BASE_NUMBER_SUFFIX .SET 1`

When symbol "`__FSL_INC_BASE_NUMBER_SUFFIX`" is defined, the suffix representation will be selected.

## 2. Details on Functions Related with Flash Memory

Though there are no changes in the features of each function of the FSL, the arguments or type declarations of function call have been changed. Therefore, the changed contents of each function are shown in the following pages

### 2.1 .FSL\_Init

#### (1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_Init( \_\_far fsl\_descriptor\_t\* descriptor\_pstr)

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_Init(**const** \_\_far fsl\_descriptor\_t\* descriptor\_pstr)

#### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__far fsl_descriptor_t* descriptor_pstr	AX(0-15), C(16-23)
<b>RENESAS (CC-RL)</b>	<b>const</b> __far fsl_descriptor_t* descriptor_pstr	<b>DE(0-15), A(16-23)</b>

The structure fsl\_descriptor\_t :

	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	typedef struct { fsl_u08 fsl_flash_voltage_u08; fsl_u08 fsl_frequency_u08; fsl_u08 fsl_auto_status_check_u08; } fsl_descriptor_t;	fsl_descriptor_str: DB fsl_flash_voltage_u08 DB fsl_frequency_u08 DB fsl_auto_status_check_u08
<b>RENESAS (CC-RL)</b>	typedef struct { fsl_u08 fsl_flash_voltage_u08; fsl_u08 fsl_frequency_u08; fsl_u08 fsl_auto_status_check_u08; } fsl_descriptor_t;	fsl_descriptor_str: <b>.DB</b> fsl_flash_voltage_u08 <b>.DB</b> fsl_frequency_u08 <b>.DB</b> fsl_auto_status_check_u08

#### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

#### (4) Destroyed register None

## 2.2 FSL\_PrepareFunctions

- (1) Function Prototype  
 RENESAS CA78K0R:  
 void FSL\_PrepareFunctions( void )  
 RENESAS CC-RL:  
 void **\_\_far** FSL\_PrepareFunctions( void )
- (2) Argument  
 None
- (3) Return value  
 None
- (4) Destroyed register  
 None

## 2.3 FSL\_PrepareExtFunctions

- (1) Function Prototype  
 RENESAS CA78K0R:  
 void FSL\_PrepareExtFunctions( void )  
 RENESAS CC-RL:  
 void **\_\_far** FSL\_PrepareExtFunctions( void )
- (2) Argument  
 None
- (3) Return value  
 None
- (4) Destroyed register  
 None

## 2.4 FSL\_ChangeInterruptTable

- (1) Function Prototype  
 RENESAS CA78K0R:  
 void FSL\_ChangeInterruptTable( fsl\_u16 fsl\_interrupt\_destination\_u16 )  
 RENESAS CC-RL:  
 void **\_\_far** FSL\_ChangeInterruptTable( fsl\_u16 fsl\_interrupt\_destination\_u16 )
- (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u16 fsl_interrupt_destination_u16	AX(0-15)
<b>RENESAS (CC-RL)</b>	fsl_u16 fsl_interrupt_destination_u16	AX(0-15)

- (3) Return value  
 None
- (4) Destroyed register  
 None

## 2.5 FSL\_RestoreInterruptTable

- (1) Function Prototype  
RENESAS CA78K0R:  
void FSL\_RestoreInterruptTable( void )  
RENESAS CC-RL:  
void **\_\_far** FSL\_RestoreInterruptTable( void )
- (2) Argument  
None
- (3) Return value  
None
- (4) Destroyed register  
None

## 2.6 FSL\_Open

- (1) Function Prototype  
RENESAS CA78K0R:  
void FSL\_Open( void )  
RENESAS CC-RL:  
void **\_\_far** FSL\_Open( void )
- (2) Argument  
None
- (3) Return value  
None
- (4) Destroyed register  
None

## 2.7 FSL\_Close

- (1) Function Prototype  
RENESAS CA78K0R:  
void FSL\_Close( void )  
RENESAS CC-RL:  
void **\_\_far** FSL\_Close( void )
- (2) Argument  
None
- (3) Return value  
None
- (4) Destroyed register  
None

## 2.8 FSL\_BlankCheck

### (1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_BlankCheck( fsl\_u16 block\_u16 )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_BlankCheck( fsl\_u16 block\_u16 )

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u16 block_u16	AX
<b>RENESAS (CC-RL)</b>	fsl_u16 block_u16	AX

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.9 FSL\_Erase

### (1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_Erase( fsl\_u16 block\_u16 )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_Erase( fsl\_u16 block\_u16 )

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u16 block_u16	AX
<b>RENESAS (CC-RL)</b>	fsl_u16 block_u16	AX

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None



## 2.10 FSL\_IVerify

### (1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_IVerify( fsl\_u16 block\_u16 )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_IVerify( fsl\_u16 block\_u16 )

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u16 block_u16	AX
<b>RENESAS (CC-RL)</b>	fsl_u16 block_u16	AX

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.11 FSL\_Write

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_Write( __near fsl_write_t* write_pstr )
```

RENESAS CC-RL:

```
fsl_u08 __far FSL_Write( __near fsl_write_t* write_pstr )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__near fsl_write_t* write_pstr	AX(0-15)
<b>RENESAS (CC-RL)</b>	__near fsl_write_t* write_pstr	AX(0-15)

The structure fsl\_write\_t :

	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	typedef struct { fsl_u08 __near *fsl_data_buffer_p_u08; fsl_u32 fsl_destination_address_u32; fsl_u08 fsl_word_count_u08; } fsl_write_t;	fsl_write_str: DW fsl_data_buffer_p_u08 DG fsl_destination_address_u32 DB fsl_word_count_u08
<b>RENESAS (CC-RL)</b>	typedef struct { fsl_u08 __near *fsl_data_buffer_p_u08; fsl_u32 fsl_destination_address_u32; fsl_u08 fsl_word_count_u08; } fsl_write_t;	fsl_write_str: <b>.DB2</b> fsl_data_buffer_p_u08 <b>.DB4</b> fsl_destination_address_u32 <b>.DB</b> fsl_word_count_u08

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register None

## 2.12 FSL\_GetSecurityFlags

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_GetSecurityFlags( fsl_u08 __near *data_destination_pu08 )
```

RENESAS CC-RL:

```
fsl_u08 __far FSL_GetSecurityFlags( fsl_u08 __near *data_destination_pu08 )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)
<b>RENESAS (CC-RL)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.13 FSL\_GetBootFlag

### (1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_GetBootFlag( fsl\_u08 \_\_near \*data\_destination\_pu08 )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_GetBootFlag( fsl\_u08 \_\_near \*data\_destination\_pu08 )

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)
<b>RENESAS (CC-RL)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.14 FSL\_GetSwapState

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_GetSwapState( fsl_u08 __near *data_destination_pu08 )
```

RENESAS CC-RL:

```
fsl_u08 __far FSL_GetSwapState( fsl_u08 __near *data_destination_pu08 )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)
<b>RENESAS (CC-RL)</b>	fsl_u08 __near *data_destination_pu08	AX(0-15)

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.15 FSL\_GetBlockEndAddr

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_GetBlockEndAddr( __near fsl_getblockendaddr_t* getblockendaddr_pstr )
```

RENESAS CC-RL:

```
fsl_u08 far FSL_GetBlockEndAddr( __near fsl_getblockendaddr_t* getblockendaddr_pstr )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__near fsl_getblockendaddr_t* getblockendaddr_pstr	AX(0-15)
<b>RENESAS (CC-RL)</b>	__near fsl_getblockendaddr_t* getblockendaddr_pstr	AX(0-15)

The structure fsl\_getblockendaddr\_t :

	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	typedef struct { fsl_u32 fsl_destination_address_u32; fsl_u16 fsl_block_u16; } fsl_getblockendaddr_t;	fsl_getblockendaddr_str: DG fsl_destination_address_u32 DW fsl_block_u16
<b>RENESAS (CC-RL)</b>	typedef struct { fsl_u32 fsl_destination_address_u32; fsl_u16 fsl_block_u16; } fsl_getblockendaddr_t;	fsl_getblockendaddr_str: <b>.DB4</b> fsl_destination_address_u32 <b>.DB2</b> fsl_block_u16

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.16 FSL\_GetFlashShieldWindow

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_GetFlashShieldWindow( __near fsl_fsw_t* getfsw_pstr )
```

RENESAS CC-RL:

```
fsl_u08 far FSL_GetFlashShieldWindow( __near fsl_fsw_t* getfsw_pstr )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__near fsl_fsw_t* getfsw_pstr	AX(0-15)
<b>RENESAS (CC-RL)</b>	__near fsl_fsw_t* getfsw_pstr	AX(0-15)

The structure fsl\_fsw\_t :

	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	typedef struct { fsl_u16 fsl_start_block_u16; fsl_u16 fsl_end_block_u16; } fsl_fsw_t;	fsl_fsw_str: DW fsl_start_block_u16 DW fsl_end_block_u16
<b>RENESAS (CC-RL)</b>	typedef struct { fsl_u16 fsl_start_block_u16; fsl_u16 fsl_end_block_u16; } fsl_fsw_t;	fsl_fsw_str: <b>.DB2</b> fsl_start_block_u16 <b>.DB2</b> fsl_end_block_u16

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.17 FSL\_SetBlockEraseProtectFlag

(1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_SetBlockEraseProtectFlag( void )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_SetBlockEraseProtectFlag( void );

(2) Argument

None

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

(4) Destroyed register

None

## 2.18 FSL\_SetWriteProtectFlag

(1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_SetWriteProtectFlag( void )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_SetWriteProtectFlag( void )

(2) Argument

None

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

(4) Destroyed register

None



## 2.19 FSL\_SetBootClusterProtectFlag

(1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_SetBootClusterProtectFlag( void )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_SetBootClusterProtectFlag( void )

(2) Argument

None

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

(4) Destroyed register

None

## 2.20 FSL\_InvertBootFlag

(1) Function Prototype

RENESAS CA78K0R:

fsl\_u08 FSL\_InvertBootFlag( void )

RENESAS CC-RL:

fsl\_u08 **\_\_far** FSL\_InvertBootFlag( void )

(2) Argument

None

(3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

(4) Destroyed register

None

## 2.21 FSL\_SetFlashShieldWindow

### (1) Function Prototype

RENESAS CA78K0R:

```
fsl_u08 FSL_SetFlashShieldWindow( __near fsl_fsw_t* setfsw_pstr )
```

RENESAS CC-RL:

```
fsl_u08 __far FSL_SetFlashShieldWindow( __near fsl_fsw_t* setfsw_pstr )
```

### (2) Argument

	<i>Parameter Passing</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__near fsl_fsw_t* setfsw_pstr	AX(0-15)
<b>RENESAS (CC-RL)</b>	__near fsl_fsw_t* setfsw_pstr	AX(0-15)

The structure fsl\_fsw\_t :

	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	typedef struct { fsl_u16 fsl_start_block_u16; fsl_u16 fsl_end_block_u16; } fsl_fsw_t;	fsl_fsw_str: DW fsl_start_block_u16 DW fsl_end_block_u16
<b>RENESAS (CC-RL)</b>	typedef struct { fsl_u16 fsl_start_block_u16; fsl_u16 fsl_end_block_u16; } fsl_fsw_t;	fsl_fsw_str: <b>.DB2</b> fsl_start_block_u16 <b>.DB2</b> fsl_end_block_u16

### (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

### (4) Destroyed register

None

## 2.22 FSL\_SwapBootCluster

- (1) Function Prototype  
 RENESAS CA78K0R:  
     fsl\_u08 FSL\_SwapBootCluster( void )  
 RENESAS CC-RL:  
     fsl\_u08 **\_\_far** FSL\_SwapBootCluster( void )

- (2) Argument  
 None

- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

Remark For normal completion, the return value cannot be checked.

- (4) Destroyed register  
 None

## 2.23 FSL\_SwapActiveBootCluster

- (1) Function Prototype  
 RENESAS CA78K0R:  
     fsl\_u08 FSL\_SwapActiveBootCluster( void )  
 RENESAS CC-RL:  
     fsl\_u08 **\_\_far** FSL\_SwapActiveBootCluster( void )

- (2) Argument  
 None

- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

- (4) Destroyed register  
 None

## 2.24 FSL\_ForceReset

- (1) Function Prototype  
 RENESAS CA78K0R:  
 void FSL\_ForceReset( void )  
 RENESAS CC-RL:  
 void **\_\_far** FSL\_ForceReset( void )
- (2) Argument  
 None
- (3) Return value  
 None
- (4) Destroyed register  
 None

## 2.25 FSL\_StatusCheck

- (1) Function Prototype  
 RENESAS CA78K0R:  
 fsl\_u08 FSL\_StatusCheck( void )  
 RENESAS CC-RL:  
 fsl\_u08 **\_\_far** FSL\_StatusCheck( void )
- (2) Argument  
 None
- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

- (4) Destroyed register  
 None

## 2.26 FSL\_StandBy

- (1) Function Prototype  
 RENESAS CA78K0R:  
     fsl\_u08 FSL\_StandBy( void )  
 RENESAS CC-RL:  
     fsl\_u08 **\_\_far** FSL\_StandBy( void )

- (2) Argument  
 None

- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

- (4) Destroyed register  
 None

## 2.27 FSL\_WakeUp

- (1) Function Prototype  
 RENESAS CA78K0R:  
     fsl\_u08 FSL\_WakeUp( void )  
 RENESAS CC-RL:  
     fsl\_u08 **\_\_far** FSL\_WakeUp( void )

- (2) Argument  
 None

- (3) Return value

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	fsl_u08	C
<b>RENESAS (CC-RL)</b>	fsl_u08	<b>A</b>

- (4) Destroyed register  
 None

## 2.28 FSL\_GetVersionString

(1) Function Prototype

RENESAS CA78K0R:

\_\_far fsl\_u08\* FSL\_GetVersionString( void )

RENESAS CC-RL:

\_\_far fsl\_u08\* **\_\_far** FSL\_GetVersionString( void )

(2) Argument

None

(3) Return value

Pointer to version strings.

	<i>Parameter types</i>	
	<i>C Language</i>	<i>Assembler Language</i>
<b>RENESAS (CA78K0R)</b>	__far fsl_u08*	BC(0-15), DE(16-31)
<b>RENESAS (CC-RL)</b>	__far fsl_u08*	<b>DE(0-15), A(16-23)</b>

(4) Destroyed register

None

- Format of the library version information.

Each string is ASCII code.

"NM..MTTCCCCCVVVV"

"N" means library name.

'S' = FSL

"M..M" means family name of microcontroller.

'RL78' = RL78

"TTT" means type number.

'TXX' = TypeXX

"CCCCC"(5characters) means compiler information. (In case of CC-RL:"CCCCC" (6characters))

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

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

"VVVV" means library version .

'VXXX' = Version X.XX(release version)

Example)

"SRL78T01**R**110GV220" means "FSL RL78 Type01 V2.20 for RENESAS C Compiler **CA78K0R** V1.10"

"SRL78T01**L**1000GV221" means "FSL RL78 Type01 V2.21 for RENESAS C Compiler **CC-RL** V1.00.00"

### 3. ROM and RAM Size Used by Flash Self-programming Library

The following table shows the code size of the user's ROM and RAM used by FSL Type01 V2.21 for CC-RL.

Table 3-1 shows the code size required when all flash self-programming library functions are allocated to ROM.

Table 3-1 Code size of flash self-programming library 1 (when all functions are allocated to ROM)

Conditions	RAM Size (Bytes)	ROM Size (Bytes)
Code size when all functions are registered * Some functions cannot be used.	0	1294
Code size when all the following functions are used <ul style="list-style-type: none"> <li>• FSL_Init</li> <li>• FSL_Open</li> <li>• FSL_Close</li> <li>• FSL_PrepareFunctions</li> <li>• FSL_BlankCheck</li> <li>• FSL_Erase</li> <li>• FSL_IVerify</li> <li>• FSL_Write</li> <li>• FSL_StatusCheck</li> </ul>	0	502

Table 3-2 shows the code size required when the background operation (BGO) feature is used during flash self-programming.

Table 3-2 Code size of flash self-programming library 2 (when the BGO feature is in use)

Conditions	RAM Size (Bytes)	ROM Size (Bytes)
Code size when all functions are registered	468 (FSL_RCD)	826 + size of program that must be ROMized (468)
Code size when all the following functions are used: <ul style="list-style-type: none"> <li>• FSL_Init</li> <li>• FSL_Open</li> <li>• FSL_Close</li> <li>• FSL_PrepareFunctions</li> <li>• FSL_BlankCheck</li> <li>• FSL_Erase</li> <li>• FSL_IVerify</li> <li>• FSL_Write</li> <li>• FSL_StatusCheck</li> </ul>	88 (FSL_RCD)	502 + size of program that must be ROMized (88)

**Remark** The above tables only describe the code size of the flash self-programming library.

#### 4. Stack Size Used by Flash Functions

Table 4-1 Stack Size Used by Flash Functions

Function Name	Bytes	Function Name	Bytes
FSL_Init	44	FSL_GetBlockEndAddr	40
FSL_Open	2	FSL_GetFlashShieldWindow	50
FSL_Close	2	FSL_SwapBootCluster	40
FSL_PrepareFunctions	12	FSL_SwapActiveBootCluster	46
FSL_PrepareExtFunctions	12	FSL_InvertBootFlag	46
FSL_ChangeInterruptTable	32	FSL_SetBlockEraseProtectFlag	46
FSL_RestoreInterruptTable	32	FSL_SetWriteProtectFlag	46
FSL_BlankCheck	46	FSL_SetBootClusterProtectFlag	46
FSL_Erase	46	FSL_SetFlashShieldWindow	46
FSL_IVerify	46	FSL_StatusCheck	34
FSL_Write	46	FSL_StandBy	34
FSL_GetSecurityFlags	50	FSL_WakeUp	46
FSL_GetBootFlag	50	FSL_ForceReset	2
FSL_GetSwapState	40	FSL_GetVersionString	2

**Note** Each size does not include the stack size used by the caller to call the FSL function.



## 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 implantations etc.), or may cause serious property damages (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 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 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 noncompliance with applicable laws and regulations.
  9. 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. 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 product(s)" means any product developed or manufactured by or for Renesas Electronics.



### 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.**

2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

#### **Renesas Electronics Canada Limited**

9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3  
Tel: +1-905-237-2004

#### **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: +49-211-6503-0, Fax: +49-211-6503-1327

#### **Renesas Electronics (China) Co., Ltd.**

Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

#### **Renesas Electronics (Shanghai) Co., Ltd.**

Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333  
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

#### **Renesas Electronics Hong Kong Limited**

Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2265-6688, Fax: +852 2886-9022

#### **Renesas Electronics Taiwan Co., Ltd.**

13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

#### **Renesas Electronics Singapore Pte. Ltd.**

80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

#### **Renesas Electronics Malaysia Sdn.Bhd.**

Unit 1207, 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 India Pvt. Ltd.**

No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India  
Tel: +91-80-67208700, Fax: +91-80-67208777

#### **Renesas Electronics Korea Co., Ltd.**

12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141