[Notes]
Smart Configurator for RH850

R20TS0500EJ0100
Rev.1.00
Oct. 16, 2019

Outline

When using Smart Configurator for RH850, note the following points.

1. When using data CRC
2. When using one-pulse outputs

1. When Using Data CRC

1.1 Applicable Products

Smart Configurator for RH850 V1.0.0 or later

1.2 Applicable Devices

RH850 family: RH850/F1KM group

- RH850/F1KM-S1 group (48-pin, 64-pin, 80-pin, and 100-pin products)
- RH850/F1KM-S4 group (100-pin, 144-pin, 176-pin, and 233-pin products)

1.3 Details

Because unnecessary initialization code is generated when using the following data CRC function A (DCRA), CRC calculation is not carried out correctly.

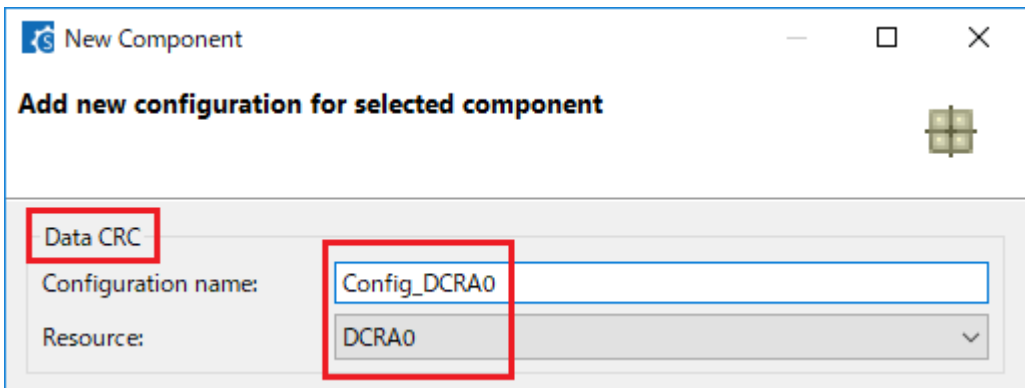
- RH850/F1KM-S1 group: 48-pin and 64-pin products

  DCRA0
- RH850/F1KM-S1 group: 80-pin and 100-pin products

  DCRA0 to DCRA3
- RH850/F1KM-S4 group: 100-pin, 144-pin, 176-pin, and 233-pin products

  DCRA0 to DCRA3

■ When using DCRA unit 0



1.4 Workaround

Manually delete the unnecessary initialization code from the generated code in the following source file of data CRC function A(Note):

- Functions in source file “<configuration-name>.c”:

  “void R_<configuration-name>_Create(void)”

  “void R_<configuration-name>_Input32bitData(const uint32_t * data, uint32_t data_num)”

  “void R_<configuration-name>_Input16bitData(const uint16_t * data, uint32_t data_num)”

  “void R_<configuration-name>_Input8bitData(const uint8_t * data, uint32_t data_num)”

Note: If code is generated again, the previous state is restored. Modification is necessary each time you perform code generation.

The following is an example of the required modification when <*configuration-name*> is Config_DCRA*n* in the RH850/F1KM group. Delete the unnecessary initialization code shown in red.

Details of modification

```
void R_Config_DCRAn_Create(void)
{
    ...
    DCRAn.CIN = _DCRA_CLEAR_DATA;
    /* Synchronization processing */
    ...
}

void R_Config_DCRAn_Input32bitData(const uint32_t * data, uint32_t data_num)
{
    ...
    DCRAn.CIN = _DCRA_CLEAR_DATA;

    for(i=0; i<data_num; i++)
    ...
}

void R_Config_DCRAn_Input16bitData(const uint16_t * data, uint32_t data_num)
{
    ...
    DCRAn.CIN = _DCRA_CLEAR_DATA;

    for(i=0; i<data_num; i++)
    ...
}

void R_Config_DCRAn_Input8bitData(const uint8_t * data, uint32_t data_num)
{
    ...
    DCRAn.CIN = _DCRA_CLEAR_DATA;

    for(i=0; i<data_num; i++)
    ...
}
```

n = unit number

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version. (Scheduled to be released in January 2020.)

2. When Using One-Pulse Outputs

2.1 Applicable Products

Smart Configurator for RH850 V1.0.0 or later

2.2 Applicable Devices

RH850 family: RH850/F1KM group

- RH850/F1KM-S1 group (48-pin, 64-pin, 80-pin, and 100-pin products)
- RH850/F1KM-S4 group (100-pin, 144-pin, 176-pin, and 233-pin products)

2.3 Details

Because interval timer mode is set in the TAUB*n* channel mode OS register (TAUB*n*CMORm)(Note) and TAUD*n* channel mode OS register (TAUD*n*CMOR*m*)(Note) when using one-pulse outputs in the following timer array units, Smart Configurator does not operate properly.

Note: *n* = unit number, *m* = channel number

- RH850/F1KM-S1 group: 48-pin and 64-pin products

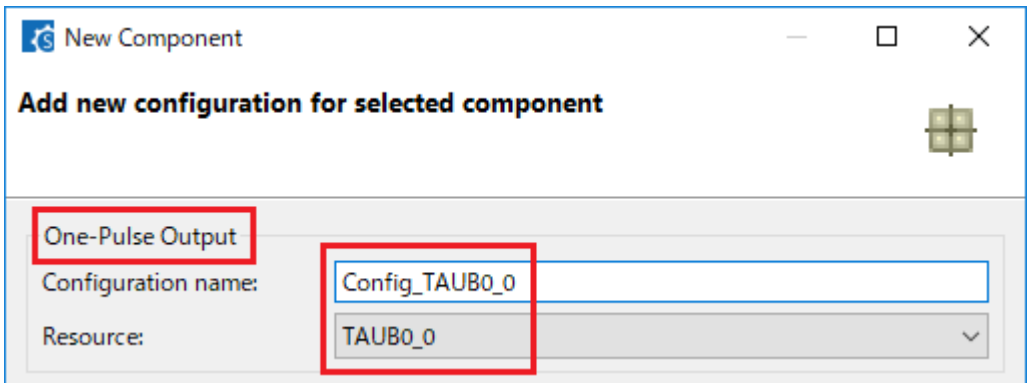
  TAUD0
- RH850/F1KM-S1 group: 80-pin and 100-pin products

  TAUB0 and TAUD0
- RH850/F1KM-S4 group: 100-pin and 144-pin products

  TAUB0 and TAUD0
- RH850/F1KM-S4 group: 176-pin and 233-pin products

  TAUB0, TAUB1, and TAUD0

■ When using one-pulse outputs of channel 0 of TAUB0



2.4 Workaround

Manually modify the generated code in the following source file for one-pulse outputs(Note):

Note: If code is generated again, the previous state is restored. Modification is necessary each time you perform code generation.

- For TAUB*n*:

Function in source file “<configuration-name>.c”:

“void R_<configuration-name>_Create(void)”

The following is an example of the required modification when <*configuration-name*> is Config_TAUB*n_m* in the RH850/F1KM group. The modification is shown in red.

Before modification

```
void R_Config_TAUBn_m_Create(void)
{
    ...
    /* Set channel m setting */
    TAUBn.CMORm = ... | _TAUB_INTERVAL_TIMER_MODE | ...;
    ...
}
```

n = unit number, *m* = channel number

After modification

```
void R_Config_TAUBn_m_Create(void)
{
    ...
    /* Set channel m setting */
    TAUBn.CMORm = ... | _TAUB_PULSE_ONECOUNT_MODE | ...;
    ...
}
```

n = unit number, *m* = channel number

- For TAUD*n*:

  Function in source file “<configuration-name>.c”:

  “void R_<configuration-name>_Create(void)”

The following is an example of the required modification when <*configuration-name*> is Config_TAUD*n*_*m* in the RH850/F1KM group. The modification is shown in red.

Before modification

```
void R_Config_TAUDn_m_Create(void)
{
    ...
    /* Set channel m setting */
    TAUDn.CMORm = ... | _TAUD_INTERVAL_TIMER_MODE | ...;
    ...
}
```

After modification

```
void R_Config_TAUDn_m_Create(void)
{
    ...
    /* Set channel m setting */
    TAUDn.CMORm = ... | _TAUD_PULSE_ONECOUNT_MODE | ...;
    ...
}
```

2.5 Schedule for Fixing the Problem

This problem will be fixed in the next version. (Scheduled to be released in January 2020.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Oct.16.19	-	First edition issued

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

TS Colophon 4.0