【注意事項】
RX ファミリ Firmware Integration Technology, RX Driver Package

R20TS1223JJ0100
Rev.1.00
2026.04.20

概要

以下の製品の使用上の注意事項を連絡します。

1. R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義を用いたインターフェースヘッダの定義を使用した時の注意事項
2. R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義、R_BSP_PRAGMA_UNPACK、R_BSP_PRAGMA_PACKOPTION の定義を用いたインターフェースヘッダの定義を使用した時の注意事項

1. R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義を用いたインターフェースヘッダの定義を使用した時の注意事項

1.1 該当製品

(1) グラフィック LCD コントローラモジュール Firmware Integration Technology

Rev.1.50 / R01AN1980JJ0150 以降 Rev.1.61 / R01AN1980JJ0161 までの全製品

(2) EPTPC モジュール Firmware Integration Technology

Rev.1.17 / R01AN1943JJ0117

(3) I²C バスインタフェース(RIIC)モジュール Firmware Integration Technology

Rev.2.49 / R01AN1692JJ0249 以降 Rev.3.11 / R01AN1692JJ0311 までの全製品

(4) ハイスピード I²C バスインタフェース(RIICHS)モジュール Firmware Integration Technology

Rev.1.00 / R01AN5552JJ0100 以降 Rev.1.23 / R01AN5552JJ0123 までの全製品

(5) RSCI 簡易 I²C モジュール Firmware Integration Technology

Rev.1.00 / R01AN8028JJ0100 および Rev.1.01 / R01AN8028JJ0101

(6) 簡易 I²C モジュール Firmware Integration Technology

Rev.2.49 / R01AN1691JJ0249 以降 Rev.2.91 / R01AN1691JJ0291 までの全製品

(7) バッテリバックアップ機能モジュール Firmware Integration Technology

Rev.2.10 / R01AN2796JJ0210 以降 Rev.2.33 / R01AN2796JJ0233 までの全製品

(8) RX Driver Package

上記(1)～(7)の製品は、以下の RX Driver Package にも同梱されています。

Ver.1.33 / R01AN6073JJ0133 以降 Ver.1.50 / R01AN8254JJ0150 までの全製品

1.2 該当デバイス

RX ファミリ

1.3 内容

1.4 発生条件で指定された条件が満たされた場合に、R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義内の bit_order に関する定義が無効になるため、参照時および設定時にそれぞれ以下の問題が発生します。

1.3.1 参照時の問題

該当変数を使用した API 関数の引数として変数を指定し、API 関数の実行後に該当変数をビットの構造体定義で参照したとき、参照先が正しくないため期待する値が読み取れない問題が発生します。

1.3.2 設定時の問題

該当変数をビットの構造体定義で設定した後 API 関数の引数として変数を指定し、API 関数を実行したとき、API 関数の中でビットの構造体定義で参照する処理があると、参照先が正しくないため、期待どおりの実行結果にならない問題が発生します。

1.4 発生条件

以下の条件をすべて満たした場合に発生します。

1.4.1 参照時の問題の発生条件

- CCRX かつ C++のプロジェクトを作成する。
- 1.1 該当製品のモジュールの(3), (4), (5), (6), (7)のいずれかをプロジェクトに組み込む。
- cpp ファイル内で R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義をインターフェースヘッダで使用している構造体定義を用いて変数を宣言し、関連する API 関数を実行する。
- cpp ファイル内で API 関数実行後にビットの構造体定義を使用して変数を参照する。

該当するモジュールの発生条件の一例を以下に示します。

(3) I²C バスインタフェース(RIIC)モジュール Firmware Integration Technology

```
volatile riic_return_t ret;
riic_info_t iic_info_m;
riic_mcu_status_t riic_status;

iic_info_m.ch_no = 0;
ret = R_RIIC_GetStatus(&iic_info_m, &riic_status);
/* ビットの構造体定義で参照 */
if( 0 == riic_status.BIT.SDAI )
{
    /* SDA が Low の場合の処理 */
}
```

1.4.2　設定時の問題の発生条件

- CCRX かつ C++のプロジェクトを作成する。
- 1.1 該当製品のモジュールの(1), (2), (7)のいずれかをプロジェクトに組み込む。
- cpp ファイル内で R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義をインターフェースヘッダで使用している構造体定義を用いて変数を宣言する。
- cpp ファイル内でビットの構造体定義を使用して変数を設定し関連する API 関数を実行する。

該当するモジュールの発生条件の一例を以下に示します。

(1) グラフィック LCD コントローラモジュール Firmware Integration Technology

```
volatile glcdc_err_t ret_glcdc;
glcdc_color_t bg_color;
/* ビットの構造体定義で変数を設定 */
bg_color.byte.r = 0x00h;
bg_color.byte.g = 0x00h;
bg_color.byte.b = 0xFFh;

ret_glcdc = R_GLCDC_Control(GLCDC_CMD_CHANGE_BG_COLOR, (void *)&bg_color);
```

1.5　回避策

1.1 該当製品のモジュールのインタフェースヘッダ(*_if.h)で R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT に関連する定義を使用している構造体定義に直接 bit_order left のプリプロセッサの宣言を追加してください。

回避策の修正の一例を以下に示します。該当するモジュールについて同様の修正を行ってください。

(3) I²C バスインタフェース(RIIC)モジュール Firmware Integration Technology

r_riic_rx_if.h 一部抜粋

修正前

```
typedef union
{
    uint32_t LONG;
    R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT_21
    (
        uint32_t rsv :12, /* reserve */
        uint32_t AAS2 :1, /* Slave2 address detection flag */
        uint32_t AAS1 :1, /* Slave1 address detection flag */
        uint32_t AAS0 :1, /* Slave0 address detection flag */
        uint32_t GCA :1, /* Generalcall address detection flag */
        uint32_t DID :1, /* DeviceID address detection flag */
        uint32_t HOA :1, /* Host address detection flag */
```

```
        uint32_t MST :1, /* Master mode / Slave mode flag */
        uint32_t TMO :1, /* Time out flag */
        uint32_t AL :1, /* Arbitration lost detection flag */
        uint32_t SP :1, /* Stop condition detection flag */
        uint32_t ST :1, /* Start condition detection flag */
        uint32_t RBUF :1, /* Receive buffer status flag */
        uint32_t SBUF :1, /* Send buffer status flag */
        uint32_t SCLO :1, /* SCL pin output control status */
        uint32_t SDAO :1, /* SDA pin output control status */
        uint32_t SCLI :1, /* SCL pin level */
        uint32_t SDAI :1, /* SDA pin level */
        uint32_t NACK :1, /* NACK detection flag */
        uint32_t TRS :1, /* Send mode / Receive mode flag */
        uint32_t BSY :1 /* Bus status flag */
    ) BIT;
} riic_mcu_status_t;
```

修正後(赤字部分を追加)

```
#pragma bit_order left
typedef union
{
    uint32_t LONG;
    R_BSP_ATTRIB_STRUCT_BIT_ORDER_LEFT_21
    (
        uint32_t rsv :12, /* reserve */
        uint32_t AAS2 :1, /* Slave2 address detection flag */
        uint32_t AAS1 :1, /* Slave1 address detection flag */
        uint32_t AAS0 :1, /* Slave0 address detection flag */
        uint32_t GCA :1, /* Generalcall address detection flag */
        uint32_t DID :1, /* DeviceID address detection flag */
        uint32_t HOA :1, /* Host address detection flag */
        uint32_t MST :1, /* Master mode / Slave mode flag */
        uint32_t TMO :1, /* Time out flag */
        uint32_t AL :1, /* Arbitration lost detection flag */
        uint32_t SP :1, /* Stop condition detection flag */
        uint32_t ST :1, /* Start condition detection flag */
        uint32_t RBUF :1, /* Receive buffer status flag */
        uint32_t SBUF :1, /* Send buffer status flag */
        uint32_t SCLO :1, /* SCL pin output control status */
        uint32_t SDAO :1, /* SDA pin output control status */
        uint32_t SCLI :1, /* SCL pin level */
```

```
        uint32_t SDAI :1, /* SDA pin level */
        uint32_t NACK :1, /* NACK detection flag */
        uint32_t TRS :1, /* Send mode / Receive mode flag */
        uint32_t BSY :1 /* Bus status flag */
    ) BIT;
} riic_mcu_status_t;
#endif bit_order
```

1.6 恒久対策

1.1 該当製品の(2)を除き、今後のバージョンで改修予定です。

2. R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義、R_BSP_PRAGMA_UNPACK、R_BSP_PRAGMA_PACKOPTION の定義を用いたインターフェースヘッダの定義を使用した時の注意事項

2.1 該当製品

(1) グラフィック LCD コントローラモジュール Firmware Integration Technology

Rev.1.50 / R01AN1980JJ0150 以降 Rev.1.61 / R01AN1980JJ0161 までの全製品

(2) I²C バスインタフェース(RIIC)モジュール Firmware Integration Technology

Rev.2.49 / R01AN1692JJ0249 以降 Rev.3.11 / R01AN1692JJ0311 までの全製品

(3) ハイスピード I²C バスインタフェース(RIICHS)モジュール Firmware Integration Technology

Rev.1.00 / R01AN5552JJ0100 以降 Rev.1.23 / R01AN5552JJ0123 までの全製品

(4) RSCI 簡易 I²C モジュール Firmware Integration Technology

Rev.1.00 / R01AN8028JJ0100 および Rev.1.01 / R01AN8028JJ0101

(5) RSPI モジュール Firmware Integration Technology

Rev.3.03 / R01AN1827JJ0303 以降 Rev.3.81 / R01AN1827JJ0381 までの全製品

(6) RSPIA モジュール Firmware Integration Technology

Rev.1.10 / R01AN5684JJ0110 以降 Rev.1.63 / R01AN5684JJ0163 までの全製品

(7) 簡易 I²C モジュール Firmware Integration Technology

Rev.2.49 / R01AN1691JJ0249 以降 Rev.2.91 / R01AN1691JJ0291 までの全製品

(8) USB Host コミュニケーションデバイスクラスドライバ(HCDC)

Rev.1.31 / R01AN2027JJ0131 以降 Rev.1.45 / R01AN2027JJ0145 までの全製品

(9) USB Host コミュニケーションデバイスクラスドライバ(HCDC) for USB ミニファームウェア

Rev.1.20 / R01AN2167JJ0120 以降 Rev.1.32 / R01AN2167JJ0132 までの全製品

(10) USB ペリフェラルコミュニケーションデバイスクラスドライバ(PCDC)

Rev.1.31 / R01AN2030JJ0249 以降 Rev.1.45 / R01AN2030JJ0145 までの全製品

(11) USB ペリフェラルコミュニケーションデバイスクラスドライバ(PCDC) for USB ミニファームウェア

Rev.1.20 / R01AN2170JJ0249 以降 Rev.1.32 / R01AN2170JJ0132 までの全製品

(12) バッテリバックアップ機能モジュール Firmware Integration Technology

Rev.2.10 / R01AN2796JJ0100 以降 Rev.2.33 / R01AN2796JJ0233 までの全製品

(13) RX Driver Package

上記(1)～(12)の製品は、以下の RX Driver Package にも同梱されています。

Ver.1.33 / R01AN6073JJ0133 以降 Ver.1.50 / R01AN8254JJ0150 までの全製品

2.2 該当デバイス

RX ファミリ

2.3 内容

2.4 発生条件で指定された条件を満たした場合に、R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義内の bit_order に関する定義が無効になります。また、R_BSP_PRAGMA_UNPACK、R_BSP_PRAGMA_PACKOPTION の定義の無効になります。そのため、ソースファイル内ではプリプロセッサの宣言をしているのにコンパイル後はプリプロセッサの宣言が無効となるので、宣言の状態と実体と合わない問題が発生します。

2.4 発生条件

以下の条件をすべて満たした場合に発生します。

- CCRX かつ C++のプロジェクトを作成する。
- 2.1 該当製品のモジュールのいずれかをプロジェクトに組み込む。
- cpp ファイル内で R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義、R_BSP_PRAGMA_UNPACK の定義をインターフェースヘッダで使用している構造体定義を用いて変数を宣言し、使用する。

2.5 回避策

2.5.1 R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義を用いるモジュールでの回避策

2.1 該当製品のモジュールのインタフェースヘッダ(*_if.h)で R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT に関連する定義を使用している構造体定義に直接 bit_order right のプリプロセッサの宣言を追加してください。

回避策の修正の一例を以下に示します。該当するモジュールについて同様の修正を行ってください。

(5) RSPI モジュール Firmware Integration Technology

r_rspi_rx_if.h 一部抜粋

修正前

```
typedef union rspi_command_word_s
{
    R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT_11 (
        rspi_spcmd_cpha_t        cpha         :1,
        rspi_spcmd_cpol_t        cpol         :1,
        rspi_spcmd_br_div_t      br_div       :2,
        rspi_spcmd_ssl_assert_t   ssl_assert    :3,
        rspi_spcmd_ssl_negation_t ssl_negate    :1,
        rspi_spcmd_bit_length_t   bit_length    :4,
        rspi_spcmd_bit_order_t    bit_order     :1,
        rspi_spcmd_spnden_t      next_delay    :1,
        rspi_spcmd_slnden_t      ssl_neg_delay :1,
        rspi_spcmd_sckden_t      clock_delay   :1,
        rspi_spcmd_dummy_t       dummy        :16
);
    uint16_t word[2];
} rspi_command_word_t;
```

修正後(赤字部分を追加)

```
#pragma bit_order right
typedef union rspi_command_word_s
{
    R_BSP_ATTRIB_STRUCT_BIT_ORDER_RIGHT_11 (
        rspi_spcmd_cpha_t        cpha         :1,
        rspi_spcmd_cpol_t        cpol         :1,
        rspi_spcmd_br_div_t      br_div       :2,
        rspi_spcmd_ssl_assert_t   ssl_assert    :3,
        rspi_spcmd_ssl_negation_t ssl_negate    :1,
        rspi_spcmd_bit_length_t   bit_length    :4,
        rspi_spcmd_bit_order_t    bit_order     :1,
        rspi_spcmd_spnden_t      next_delay    :1,
        rspi_spcmd_slnden_t      ssl_neg_delay :1,
        rspi_spcmd_sckden_t      clock_delay   :1,
        rspi_spcmd_dummy_t       dummy        :16
);
    uint16_t word[2];
} rspi_command_word_t;
#endif bit_order
```

2.5.2 R_BSP_PRAGMA_UNPACK、R_BSP_PRAGMA_PACKOPTION の定義を用いるモジュールでの回避策

2.1 該当製品のモジュールのインタフェースヘッダ(*_if.h)の R_BSP_PRAGMA_UNPACK の定義部分に直接 unpack のプリプロセッサの宣言を追加してください。R_BSP_PRAGMA_PACKOPTION の定義部分には直接 packoption のプリプロセッサの宣言を追加してください。

修正前

```
R_BSP_PRAGMA_UNPACK
```

```
R_BSP_PRAGMA_PACKOPTION
```

修正後(赤字部分を追加)

```
R_BSP_PRAGMA_UNPACK
#pragma unpack
```

```
R_BSP_PRAGMA_PACKOPTION
#pragma packoption
```

2.6 恒久対策

今後のバージョンで改修予定です。

以上

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	Apr.20.26	-	新規発行

本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

TS Colophon 4.3