

## CS+用 RXコード生成 ご使用上のお願い

CS+用 RXコード生成 (CS+ Code\_Generator for RX) の注意事項を連絡します。

### 1. マルチファンクションタイマパルスユニット3の注意事項

対象: RX64Mグループ

### 2. シリアルコミュニケーションインタフェースの注意事項

対象: RX111,RX113,RX64M および RX71M グループ

### 1. マルチファンクションタイマパルスユニット3(以下 MTU3)の注意事項

#### 1.1 該当製品

CS+用 RXコード生成  
(CS+ Code\_Generator for RX) V1.03.00以降

#### 1.2 該当マイコン

RXファミリ: RX64Mグループ

#### 1.3 内容

MTU3を下記の条件で設定した場合、コード生成を実行した時のポート設定出力コードが正しくありません。

<条件>

"周辺機能"の設定: マルチファンクションタイマパルスユニット3

- 対象チャンネル: MTU3
- 機能設定: ノーマルモード
- TGRD3: アウトプットコンペアレジスタ
- MTIOC3D端子の出力: PC4

#### 1.4 回避策

r\_cg\_mtu3.cファイルのvoid R\_MTU3\_Create(void) 関数にあるポート設定コードを下記のように修正してください。

なお、コード生成後は常に下記の修正が必要です。

修正前:

```
-----  
void R_MTU3_Create(void)  
{  
    :  
    :  
    /* Set MTIOC3D pin */  
    MPC.PC4PFS.BYTE = 0x01U;  
    PORTB.PMR.BYTE != 0x10U;          <-- 誤ったポート設定コード  
-----
```

修正後:

```
-----  
void R_MTU3_Create(void)  
{  
    :  
    :  
    /* Set MTIOC3D pin */  
    MPC.PC4PFS.BYTE = 0x01U;  
    PORTC.PMR.BYTE != 0x10U;          <-- ポート設定コードを修正  
-----
```

## 1.5 恒久対策

次期バージョンで改修する予定です。

## 2. シリアルコミュニケーションインタフェース(以下 SCI)の注意事項

### 2.1 該当製品

CS+用 RXコード生成  
(CS+ Code\_Generator for RX) V1.03.00以降

### 2.2 該当マイコン

RXファミリ: RX111, RX113, RX64M および RX71Mグループ

### 2.3 内容

(1) SCIを下記の条件で設定した場合、コード生成を実行した時のハンドラ出力コードが正しくありません。マスタ受信を行った場合、1バイト余分にデータ受信用クロックが生成されます。また、余分に受信したデータは、指定アドレスへ格納されません。

<条件>

- "周辺機能"の設定: シリアルコミュニケーションインタフェース
- 対象チャネル: すべて
  - 機能設定: 簡易I2Cバス

- I2C割り込みモードセレクト:受信割り込み および 送信割り込みを使用

(2) SCIで簡易SPIバスを選択した場合、コード生成を実行した時の出力コードが正しくありません。R\_SCIn\_Stop(void)関数の実行後、R\_SCIn\_Start(void)関数を実行しても受信できません。

## 2.4 回避策

(1) 2.3(1)の回避策

r\_cg\_sci\_user.cファイルの

static void r\_scin\_transmit\_interrupt(void)関数にある条件文を下記のように修正してください。

なお、コード生成後は常に下記の修正が必要です。

修正前:

```
-----  
if (g_scin_rx_length == g_scin_rx_count)    <-- 誤った条件文  
{  
    SCIn.SIMR2.BIT.IICACKT = 1U;  
    /* Write dummy */  
    SCIn.TDR = 0xFFU;  
  
    /* Generate stop condition */  
    g_scin_iic_cycle_clag = _00_SCI_IIC_STOP_CYCLE;  
    R_SCIn_IIC_StopCondition();  
}  
-----
```

修正後:

```
-----  
if (( g_scin_rx_length - 1) == g_scin_rx_count) <-- 条件文を修正  
{  
    SCIn.SIMR2.BIT.IICACKT = 1U;  
    /* Write dummy */  
    SCIn.TDR = 0xFFU;  
}  
else if (g_scin_rx_length == g_scin_rx_count)    <-- 条件文を修正  
{  
    /* Generate stop condition */  
    g_scin_iic_cycle_clag = _00_SCI_IIC_STOP_CYCLE;  
    R_SCIn_IIC_StopCondition();  
}  
-----
```

(2) 2.3(2)の回避策

r\_cg\_sci.cファイルのR\_SCIIn\_Stop(void)関数(注)にある誤ったコードを削除してください。

注: nはコード生成設定時のチャンネル  
なお、コード生成後は常に下記の修正が必要です。

修正例:

```
-----  
void R_SCI0_Stop(void)  
{  
    /* Set SMOSI0 pin */  
    PORT2.PMR.BYTE &= 0xFEU;           <- この行を削除  
    :  
    :  
}-----
```

## 2.5 恒久対策

次期バージョンで改修する予定です。

---

### [免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。