

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

MESC TECHNICAL NEWS

No. M16C-22-9904

MESC TECHNICAL NEWS

「No.M16C-20-9903」の差し替え

MESC TECHNICAL NEWS「No.M16C-20-9903」に誤りがありましたので訂正いたします。訂正したテクニカルニュース「M16C/62 フラッシュメモリ版消費電力に関する注意事項」を「No.M16C-20-9903」のニュースに差し替えてください。

[訂正内容]

対策の例2は、フラッシュメモリ5V版専用でした。そのため、フラッシュメモリ3V版用として新たに例3を追加しました。

[添付]

訂正したテクニカルニュース No.M16C-22-9904

「M16C/62 フラッシュメモリ版消費電力に関する注意事項」.....10枚

MESC TECHNICAL NEWS

No. M16C-22-9904

M16C/62 フラッシュメモリ版 消費電力に関する注意事項

1. 対象品種

フラッシュメモリ 5V 版 : M30624FGFP、M30624FGGP、M30625FGGP

フラッシュメモリ 3V 版 : M30624FGLFP、M30624FGLGP、M30625FGLGP

2. 注意事項

【低速モードでの注意事項】

フラッシュメモリ版の内部メモリの制御方法がマスクROM版と異なっているために、低速モード(X_{IN} を停止し X_{CIN} で動作)であっても消費電力が、十分に低下しません。低速モードでの消費電流の標準値を以下に示します。

フラッシュメモリ 5V 版 : 約 8mA($V_{CC}=5V$ 、マスクROM/ワンタイムPROM版の標準値:90 μA)

フラッシュメモリ 5V 版 : 約 5mA($V_{CC}=3V$ 、マスクROM/ワンタイムPROM版の標準値:40 μA)

フラッシュメモリ 3V 版 : 約 700 μA ($V_{CC}=3V$ 、マスクROM/ワンタイムPROM版の標準値:40 μA)

【高速、中速モードでの注意事項】

フラッシュメモリ版の内部メモリの制御方法がマスクROM版と異なっているために、CPUクロックの動作周波数を下げても消費電力が、十分に低下しません。

3. 対策

例 1 ストップモード、ウェイトモードを使用する方法(フラッシュメモリ 5V 版、3V 版共通)

例 2 消費電力を低減するためのビットを使用する方法(フラッシュメモリ 5V 版専用)

例 3 内蔵RAMでプログラムを実行する方法(フラッシュメモリ 3V 版専用)

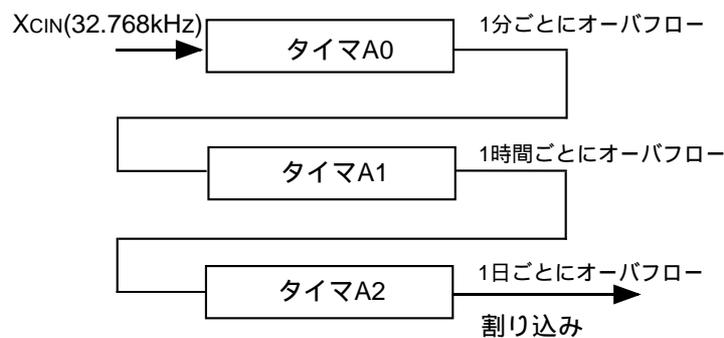
例1 ストップモード、ウエイトモードを使用する方法(共通)

ストップモード、ウエイトモードへの移行によりフラッシュメモリ制御レジスタへの特別な設定なしにマスクROM版と同等の消費電力が実現可能です。ストップモード、ウエイトモードの使用を検討ください。特にウエイトモードで時計をカウントする場合、タイマを連結することにより X_{CIN} から時間、さらに日までカウントできます。

内容

- ・タイマ A0 は、X_{CIN} をカウントソースとして 1 分までをカウントする
 - ・タイマ A1 は、タイマ A0 をカウントソースとして 1 時間までをカウントする
 - ・タイマ A2 は、タイマ A1 をカウントソースとして 1 日までをカウントする
- これを実現するための各タイマの設定例を示します。割り込みは、タイマ A2 のオーバーフローで発生させます。WAIT 命令実行前に、タイマ A2 の割り込みは許可してください。

タイマの接続



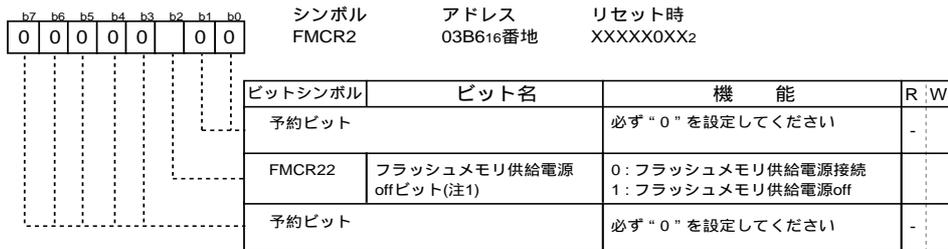
レジスタの設定

TA0MR(タイマ A0 モードレジスタ:0396 ₁₆ 番地)	=	" 0C0H "
タイマ A0(タイマ A0:0387 ₁₆ 、0386 ₁₆ 番地)	=	" 0EFFFH "
TA1MR(タイマ A1 モードレジスタ:0397 ₁₆ 番地)	=	" 001H "
TA1(タイマ A1:0389 ₁₆ 、0388 ₁₆ 番地)	=	" 0003BH "
TA2MR(タイマ A2 モードレジスタ:0398 ₁₆ 番地)	=	" 001H "
TA2(タイマ A2:038B ₁₆ 、038A ₁₆ 番地)	=	" 00017H "
TRGSR(トリガ選択レジスタ:0383 ₁₆ 番地)	=	" 00AH "

例2 消費電力を低減するためビットを使用する方法(5V版専用)

消費電力を低減するためのビットを新設しました。(図1)このビットを“1”にすることで、消費電力を低減することができますが、内蔵フラッシュメモリからプログラムを読み出すことはできなくなります。このビットを“1”にする動作、およびこのビットが“1”の間、フラッシュメモリ以外の領域で実行させてください。

フラッシュメモリ制御レジスタ2



注1. “1”を設定するためには、このビット2への“0”書き込み “1”書き込みを連続して行う必要があります。この手順でないと、“1”にできません。また、割り込み、DMA転送が入らないようにしてください。
 パラレル入出力モード時は、このビットによらず、端子により書き込みの制御がなされます。
 このビットへの書き込みは、内蔵フラッシュメモリ以外の領域のプログラムで行ってください。

図1 フラッシュメモリ制御レジスタ2

このビットを操作するためのフローチャートと参考プログラムを以下に示します。

プログラムの構成

参考プログラムは、以下のプログラムで構成されています。

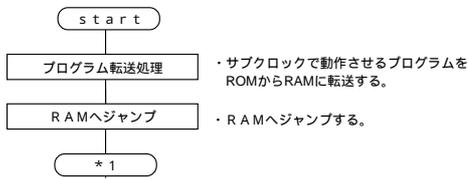
- ・低消費電力を実現するためのプログラム
 フラッシュメモリ制御レジスタを設定、クロック等の切り替え、WAIT命令を実行します。タイムA0割り込みが発生すると再度WAIT命令を実行し、外部割り込みが発生すると低速モードから復帰します。(フローチャート2)
- ・低速モードから復帰するための割り込みプログラム(INT0 割り込み)
 外部割り込みが発生すると、RAMにデータを設定します。
- ・時計をカウントするための割り込みプログラム(TA0 割り込み)
 時計をカウントします。
- ・プログラムをRAMに転送するプログラム
 上記プログラムをRAMに転送するためのプログラム(フローチャート1)

注意事項

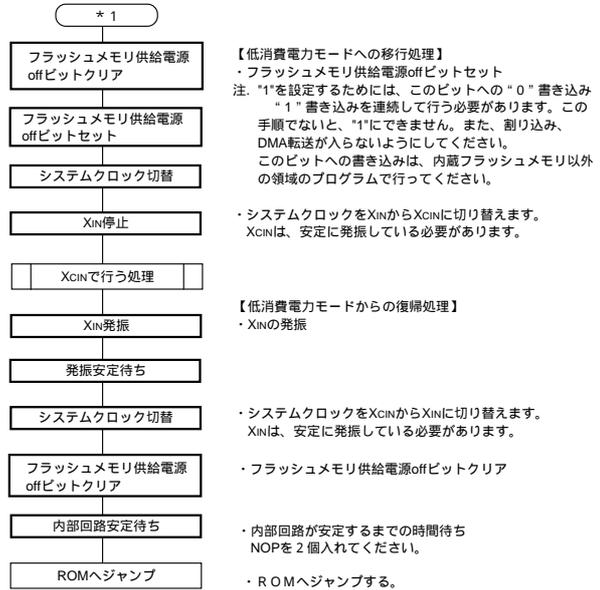
- 1 割り込みを使用する場合、割り込みプログラム、割り込みベクタテーブルもRAMに転送する必要があります。このとき、割り込みプログラム、割り込みベクタテーブルを転送してから割り込みテーブルレジスタ(INTB)を変更してください。また、割り込みテーブルレジスタ(INTB)を変更している間、割り込みを禁止してください。
- 2 参考プログラムには、フラッシュメモリ制御レジスタ2への設定とシステムクロックの切り替えのみ示しています。その他の低消費電力に移行する場合の設定(ポートの設定、割り込みの許可禁止、Vrefの切断等)は各システムにあわせて行ってください。
- 3 参考プログラムを組み込む場合、システムの評価は十分に行ってください。

フローチャート

ROM上でのプログラム
(フローチャート1)



RAM上でのプログラム
(フローチャート2)



参考プログラム

```

;//////////////////////////////////////
RAMプログラム転送処理
;//////////////////////////////////////
PROG_MOVE:
mov.w    #PROG_TOP&0FFFFh,A0
mov.b    #PROG_TOP>>16,R1H
mov.w    #Vram_P,A1
mov.w    #(PROG_END-PROG_TOP),R3
smovf.b
;//////////////////////////////////////
割り込みプログラム転送処理
;//////////////////////////////////////
VECT_PROG_MOVE:
mov.w    #VECT_PRG_TOP&0FFFFh,A0
mov.b    #VECT_PRG_TOP>>16,R1H
mov.w    #Vram_Vect,A1
mov.w    #(VECT_PRG_END-VECT_PRG_TOP),R3
smovf.b
;//////////////////////////////////////
; 可変ベクタ転送処理
;//////////////////////////////////////
VECT_TBL_MOVE:
mov.w    #VECT_TBL_TOP&0FFFFh,A0
mov.b    #VECT_TBL_TOP>>16,R1H
mov.w    #Vram_VectT,A1
mov.w    #(VECT_TBL_END-VECT_TBL_TOP),R3
smovf.b
fclr     i
ldintb   #Vram_VectT           ; intb change
nop
nop
fset     i
jmp.a    Vram_P               ; ram program jump
  
```

```

=====
;
;   RAM 転送プログラム
=====
PROG_TOP:
    mov.b    #00h,3B6h
    bset     2,3B6h                ; Flash memory supply voltage off
    bset     PRC0
    bset     CM02
    bset     CM07                ; CPU CLOCK = SUB
    bset     CM05                ; Xin STOP
    mov.b    #11111111b,P1
?:
    wait                    ; program wait
    nop
    cmp.b    #1,INT_FLG
    jne     ?-
    mov.b    #0,INT_FLG
    bset     PRC0
    bclr     CM05                ; Xin START
    bclr     CM07                ; CPU CLOCK = MAIN
    mov.b    #00h,3B6h          ; Flash memory supply voltage on
    nop
    nop
    jmp.a    ROM_TEST_PRG_TOP
PROG_END:
;////////////////////////////////////
;   割り込み
;////////////////////////////////////
VECT_PRG_TOP:
Dummy:
    REIT
=====
;
;   タイマ A0 割り込み(時間のカウンタ)
=====
TA0i:
    mov.b    #1,ONE_SEC_FLG
    add.b    #1,SEC
    cmp.b    #60,SEC
    jne     CHK_end
    mov.b    #0,SEC
    add.b    #1,MIN
MIN_chk:
    cmp.b    #60,MIN
    jne     CHK_end
    mov.b    #0,MIN
    add.b    #1,HUR
CHK_end:
    REIT
=====
;
;   外部割り込み(低消費電力モードからの復帰信号)
=====
INT0i:
    mov.b    #0,INT_FLG
    btst     P8_2
    jne     INT0_end
    mov.b    #1,INT_FLG
INT0_end:
    REIT
VECT_PRG_END:

```

```

=====
;
;   INT TABLE Area for RAM
;
=====
VECT_TBL_TOP:
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+TA0i- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+INT0i- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
VECT_TBL_END:

```

* プログラム中のラベル

Vram_P :実行プログラムを格納する RAM の先頭アドレス
Vram_Vect :割り込みプログラムを格納する RAM の先頭アドレス
Vram_VectT :割り込みベクタテーブルを格納する RAM の先頭アドレス

例3 内蔵RAMでプログラムを実行する方法(3V版専用)

プログラムの構成

参考プログラムは、以下のプログラムで構成されています。

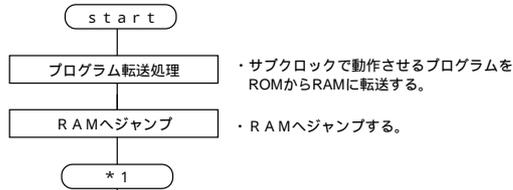
- ・低消費電力を実現するためのプログラム
クロック等の切り替え、WAIT命令を実行します。タイマA0割り込みが発生すると再度WAIT命令を実行し、外部割り込みが発生すると低速モードから復帰します。(フローチャート2)
- ・低速モードから復帰するための割り込みプログラム(INT0割り込み)
外部割り込みが発生すると、RAMにデータを設定します。
- ・時計をカウントするための割り込みプログラム(TA0割り込み)
時計をカウントします。
- ・プログラムをRAMに転送するプログラム
上記プログラムをRAMに転送するためのプログラム(フローチャート1)

注意事項

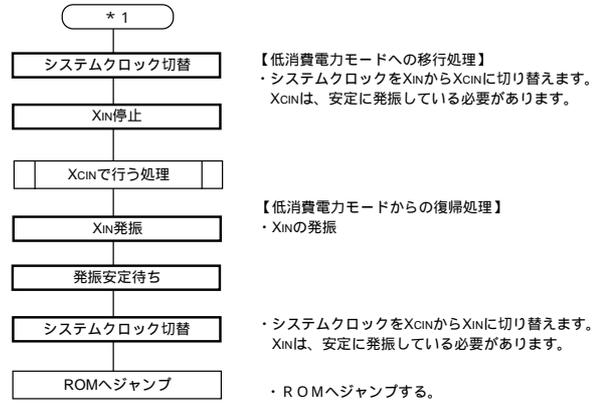
- 1 割り込みを使用する場合、割り込みプログラム、割り込みベクタテーブルもRAMに転送する必要があります。このとき、割り込みプログラム、割り込みベクタテーブルを転送してから割り込みテーブルレジスタ(INTB)を変更してください。また、割り込みテーブルレジスタ(INTB)を変更している間、割り込みを禁止してください。
- 2 参考プログラムには、システムクロックの切り替えのみ示しています。その他の低消費電力に移行する場合の設定(ポートの設定、割り込みの許可禁止、Vrefの切断等)は各システムにあわせて行ってください。
- 3 参考プログラムを組み込む場合、システムの評価は十分に行ってください。

フローチャート

ROM上でのプログラム (フローチャート1)



RAM上でのプログラム (フローチャート2)



参考プログラム

```

;////////////////////////////////////
RAMプログラム転送処理
;////////////////////////////////////
PROG_MOVE:
  mov.w    #PROG_TOP&0FFFFh,A0
  mov.b    #PROG_TOP>>16,R1H
  mov.w    #Vram_P,A1
  mov.w    #(PROG_END-PROG_TOP),R3
  smovf.b
;////////////////////////////////////
割り込みプログラム転送処理
;////////////////////////////////////
VECT_PROG_MOVE:
  mov.w    #VECT_PRG_TOP&0FFFFh,A0
  mov.b    #VECT_PRG_TOP>>16,R1H
  mov.w    #Vram_Vect,A1
  mov.w    #(VECT_PRG_END-VECT_PRG_TOP),R3
  smovf.b
;////////////////////////////////////
; 可変ベクタ転送処理
;////////////////////////////////////
VECT_TBL_MOVE:
  mov.w    #VECT_TBL_TOP&0FFFFh,A0
  mov.b    #VECT_TBL_TOP>>16,R1H
  mov.w    #Vram_VectT,A1
  mov.w    #(VECT_TBL_END-VECT_TBL_TOP),R3
  smovf.b
  fclr    i
  ldintb  #Vram_VectT                ; intb change
  nop
  nop
  fset    i
  jmp.a   Vram_P                    ; ram program jump
  
```

```

=====
;
;   RAM 転送プログラム
=====
PROG_TOP:
    bset    PRC0
    bset    CM02
    bset    CM07                ;CPU CLOCK = SUB
    bset    CM05                ;Xin STOP
    mov.b   #11111111b,P1
?:
    wait                    ; program wait
    nop
    cmp.b   #1,INT_FLG
    jne     ?-
    mov.b   #0,INT_FLG
    bset    PRC0
    bclr    CM05                ;Xin START
    bclr    CM07                ;CPU CLOCK = MAIN
    jmp.a   ROM_TEST_PRG_TOP
PROG_END:
;//////////////////////////////////////
;   割り込み
;//////////////////////////////////////
VECT_PRG_TOP:
Dummy:
    REIT
=====
;
;   タイマ A0 割り込み(時間のカウント)
=====
TA0i:
    mov.b   #1,ONE_SEC_FLG
    add.b   #1,SEC
    cmp.b   #60,SEC
    jne     CHK_end
    mov.b   #0,SEC
    add.b   #1,MIN
MIN_chk:
    cmp.b   #60,MIN
    jne     CHK_end
    mov.b   #0,MIN
    add.b   #1,HUR
CHK_end:
    REIT
=====
;
;   外部割り込み(低消費電力モードからの復帰信号)
=====
INT0i:
    mov.b   #0,INT_FLG
    btst    P8_2
    jne     INT0_end
    mov.b   #1,INT_FLG
INT0_end:
    REIT
VECT_PRG_END:

```

```
=====
;
;   INT TABLE Area for RAM
;
;=====
VECT_TBL_TOP:
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+TA0i- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+INT0i- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
.LWORD (Vram_Vect+Dummy- VECT_PRG_TOP)
VECT_TBL_END:
```

* プログラム中のラベル

Vram_P :実行プログラムを格納する RAM の先頭アドレス
Vram_Vect :割り込みプログラムを格納する RAM の先頭アドレス
Vram_VectT :割り込みベクタテーブルを格納する RAM の先頭アドレス