マイクロコンピュータ技術情報

技術通知 78K0S/KU1+, KY1+, KA1+, KB1+ 使用制限事項の件	発行番号	ZBG－CC－06－0028	1/1
	発行日	2006年 7月 11日	
	発行元	NEC エレクトロニクス株式会社 第四システム事業本部 汎用マイコンシステム事業部 第一ソリューショングループ	

文書分類	○ 使用制限事項	バージョン・アップ	ドキュメント修正	その他

関連資料		
	78K0S/KY1+ ユーザーズ・マニュアル	資料番号：U16994JJ2V0UD00（第3版）
	78K0S/KA1+ ユーザーズ・マニュアル	資料番号：U16898JJ2V0UD00（第3版）
	78K0S/KB1+ ユーザーズ・マニュアル	資料番号：U17446JJ2V0UD00（第2版）

CP(K), 0

1. 対象製品

78K0S/KU1+
μPD78F9200／78F9201／78F9202

78K0S/KY1+
μPD78F9210／78F9211／78F9212

78K0S/KA1+
μPD78F9221／78F9222

78K0S/KB1+
μPD78F9232／78F9234

2. 制限事項内容

フラッシュ・セルフ・プログラミング使用時の制限事項

フラッシュ・セルフ・プログラミングを使用する場合、通常モード、およびセルフ・プログラミング・モードに移行する直前に FLCMD レジスタの値を 0 に設定してください。また、セルフ・プログラミング・モードに遷移する特定シーケンス処理のあとに NOP 命令、HALT 命令を実行するようにしてください。

※特定シーケンス処理の詳細は、別紙に記載します。

この制限事項は、HALT 命令によるスタンバイ機能とフラッシュ・セルフ・プログラミングを併用したときに、この 2 つの処理を繰り返し実行すると動作不定状態になる不具合を回避するためのものです。

3. 制限事項詳細および回避策

上記制限事項の詳細および回避策は、別紙に記載いたします。

4. 改善計画

ソフトウェアにより回避可能なため、デバイス修正は行わず、制限事項とさせていただきます。
ユーザーズ・マニュアルは、制限事項を記載するための改版を行います。

以上

78K0S/KA1+，78K0S/KY1+使用制限事項一覧

1）対象製品

＜78K0S/KU1+＞

内容	UPD78F9200 UPD78F9201 UPD78F9202
フラッシュ・セルフ・プログラミング使用時の制限事項	△

＜78K0S/KY1+＞

内容	UPD78F9210 UPD78F9211 UPD78F9212
フラッシュ・セルフ・プログラミング使用時の制限事項	△

＜78K0S/KA1+＞

内容	UPD78F9221 UPD78F9222
フラッシュ・セルフ・プログラミング使用時の制限事項	△

＜78K0S/KB1+＞

内容	UPD78F9232 UPD78F9234
フラッシュ・セルフ・プログラミング使用時の制限事項	△

注 1）各記号はそれぞれ以下の意味を示します。

△：制限事項対象（修正予定なし）

2）使用制限事項の詳細

●フラッシュ・セルフ・プログラミング使用時の制限事項

＜不具合内容および原因＞

HALT 命令によるスタンバイ機能とフラッシュ・セルフ・プログラミングを併用してご使用される場合、下記の手順で実行すると動作不定状態になります。（図 1 参照）

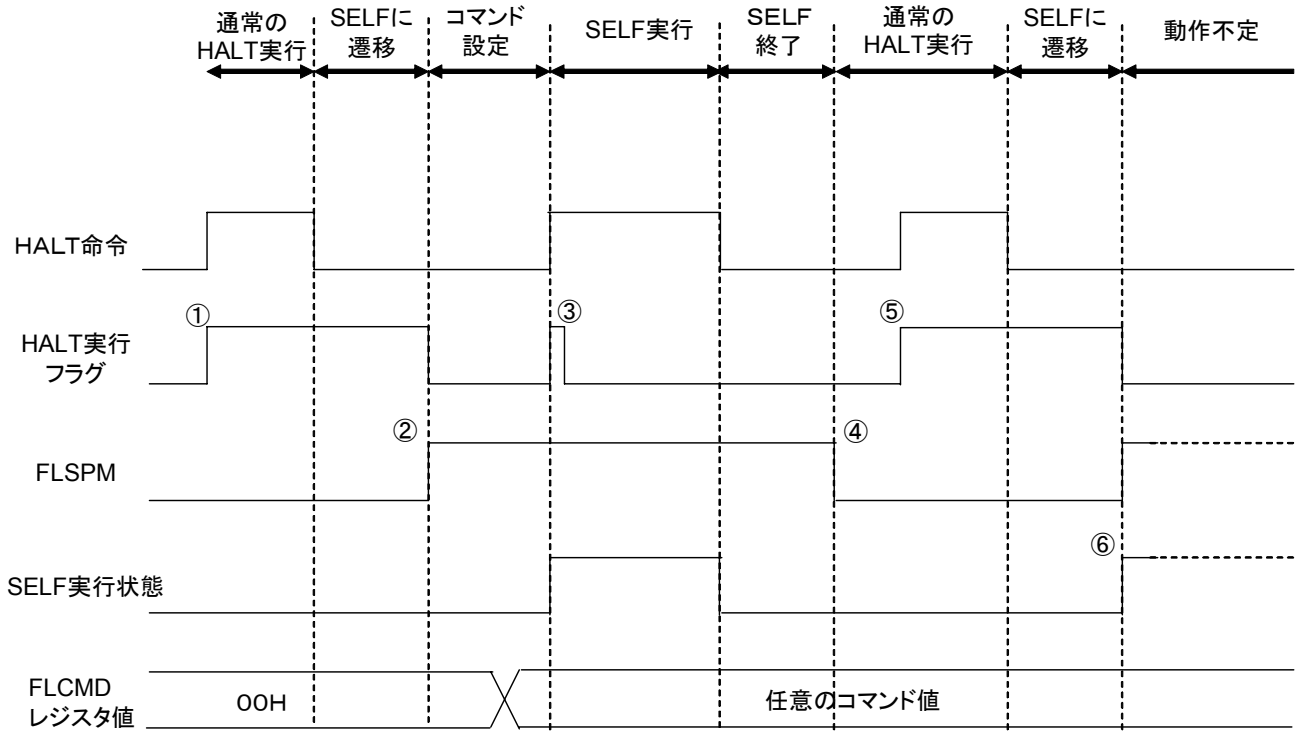
※特定シーケンスについて

この製品は、下記に示す 2 つのモードが存在します。

・通常モード：通常動作の実行状態です。HALT 命令を実行すると、スタンバイ状態になります。
・セルフ・プログラミング・モード：
セルフ・プログラミングのコマンドが実行可能になる状態です。コマンド、アドレス、書き込みデータなどを設定し、HALT 命令を実行すると、セルフ・プログラミングが行われます。

特定シーケンスとは、上記の通常モードとセルフ・プログラミング・モードを切り替えるためのレジスタ操作のことです。

図 1．不具合再現例



①． 通常の HALT 命令を実行すると、内部の HALT 実行フラグがセットされます。このフラグがセットされた状態で FLSPM がセットされることで、セルフ・プログラミングが実行されます。

②．特定シーケンスを実行し、セルフ・プログラミング・モードに遷移します。このとき、FLSPM が変化し、セルフ・プログラミングが実行可能な状態になったことを示します。しかし、この時点では、FLCMD レジスタが初期化状態(00H)であるため、セルフ・プログラミングのコマンドは実行されません。

③．FLCMD レジスタに任意のコマンド値を設定したあとに HALT 命令を実行すると、セルフ・プログラミングのコマンドが実行されます。セルフ・プログラミングのコマンドの実行と同じタイミングで、HALT 実行フラグはクリアされます。

④．セルフ・プログラミングのコマンド実行が完了し、特定シーケンスを再度実行することで、通常モードに遷移します。

⑤．再度 HALT 命令が実行されると、スタンバイ状態に移行し、HALT 実行フラグがセットされます。

⑥．スタンバイ解除後に特定シーケンスを実行し、セルフ・プログラミング・モードに遷移します。このとき、FLCMD に格納されているコマンド値が初期化されていない場合、FLSPM がセットされた瞬間、FLCMD に残っているコマンドが再度実行されます。その後、CPU 動作中にセルフ・プログラミングが実行されてしまい、CPU がフラッシュ・メモリから誤った命令を読み出してしまうため、想定しない動作に陥ります。

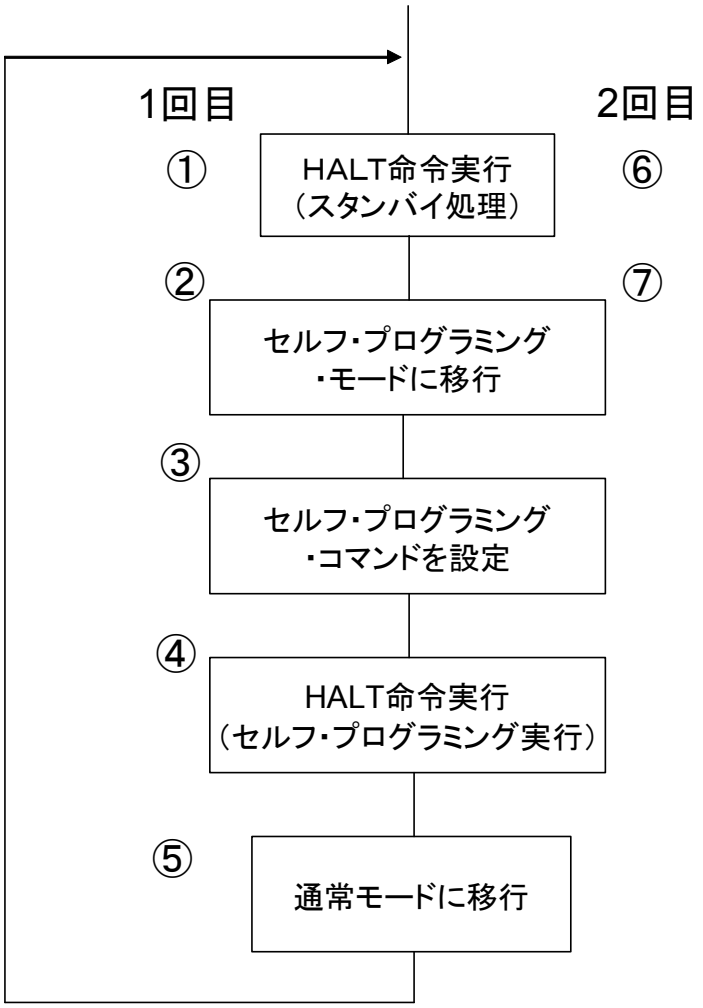
※ ①の前にフラッシュ・セルフ・プログラミングを実行した場合も同様の現象が発生します。

＜不具合回避方法＞

フラッシュ・セルフ・プログラミングを使用する場合、通常モード、およびセルフ・プログラミング・モードに移行する直前に FLCMD レジスタの値を 0 に初期化することで移行直後に不正なコマンド実行を防ぎます。また、セルフ・プログラミング・モードに遷移する特定シーケンスを完了した直後に NOP 命令、HALT 命令を実行することで、CPU とフラッシュ・メモリ制御部との実行するタイミングを制御します。

次の頁から、動作不定状態になる流れ図とソースコード例、回避策を実装した流れ図とソースコード例について記載します。

＜動作不定状態になる流れ図＞



①． 通常の HALT 命令を実行し、スタンバイ状態に入ります。その後、割込み等のスタンバイ・リリース信号により、スタンバイ状態を解除します。
②． セルフ・プログラミング・モードに移行するための、特定シーケンスを実行します。
③． 任意のセルフ・プログラミング・コマンド（ソースコードの例では、ブロック消去）を FLCMD レジスタに設定します。
④． HALT 命令を実行することで、セルフ・プログラミング・コマンドが実行されます。
⑤． ③で指定したセルフ・プログラミング処理が終了したら、通常モードに移行するための特定シーケンスを実行します。今回の例は、①～⑤までの処理を繰り返し行うことを想定しています。
⑥． 通常の HALT を実行し、スタンバイ・リリース信号を発生させ、スタンバイ状態を解除します。
⑦． 2 回目のセルフ・プログラミング・モードに移行するための特定シーケンスを実行した直後にセルフ・プログラミング・コマンド（ソースコードの例では、ブロック消去）が実行され、マイコンの動作が不定状態に入ります。

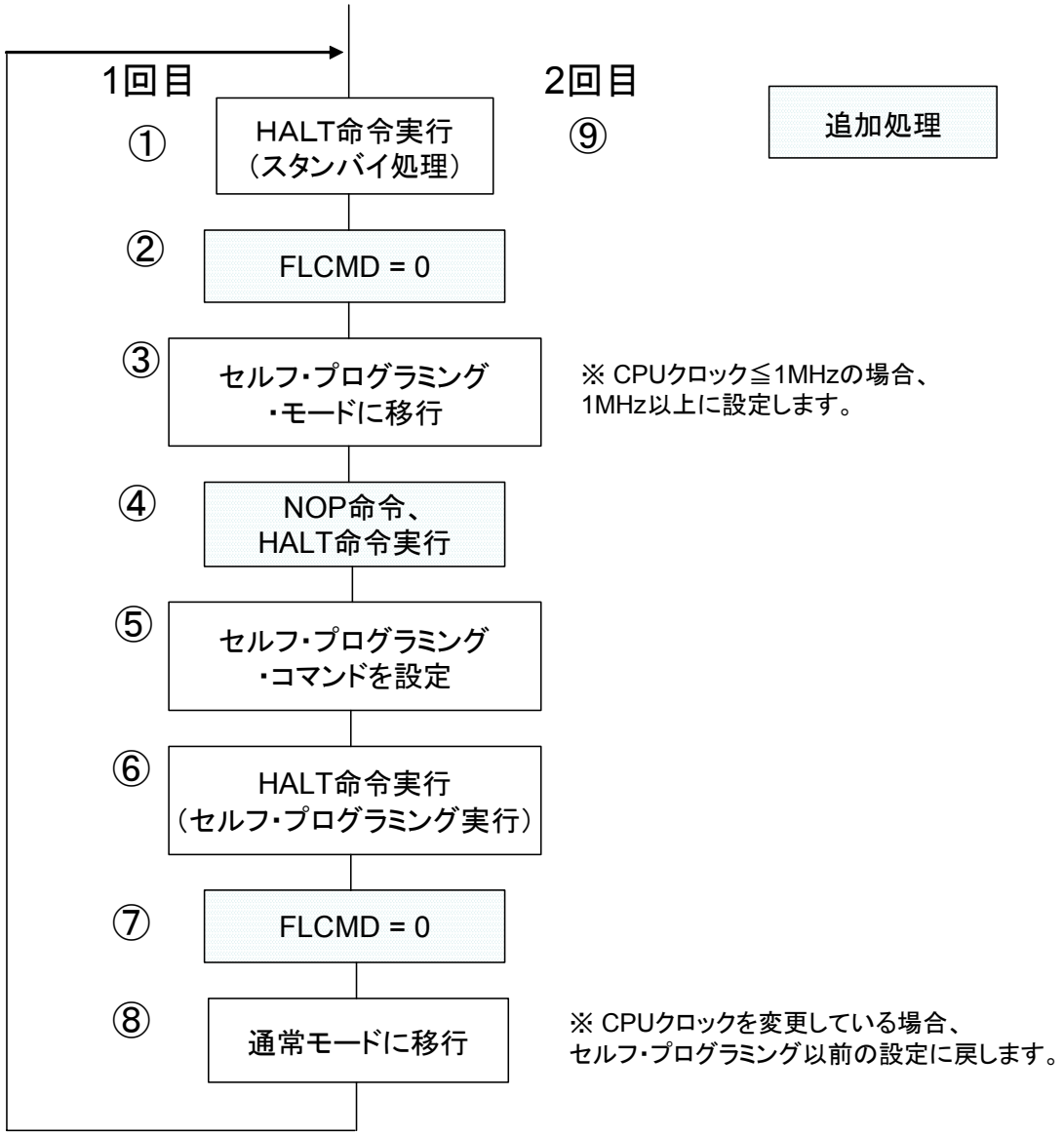
＜動作不定状態になるソースコード例（アセンブラ）＞

```
MAINLOOP:
    ; HALT を実行し、スタンバイ状態にします。フロー①、⑥
    HALT

    ; セルフ・プログラミングを実行するので、割り込みマスクを退避します。
    DI                              ; 割込み禁止
    MOV     A,MK0                   ; 割り込みマスクの退避
    XCH     A,X
    MOV     A,MK1                   ; KU1+, KY1+は、MK0 のみ退避
    PUSH    AX
    MOV     MK0,#0FFH
    MOV     MK1,#0FFH

    ; セルフ・プログラミング・モードに移行するための特定シーケンスを実行　フロー②、⑦
ModeOnLoop:
    MOV     PFCMD,#0A5H             ; PFCMD レジスタ制御
    MOV     FLPMC,#01H              ; FLPMC レジスタ制御(設定値)
    MOV     FLPMC,#0FEH             ; FLPMC レジスタ制御(設定値の反転)
    MOV     FLPMC,#01H              ; セルフ･プログラミング･モード設定
    ; 外付けの発振子や外部クロック入力の場合、ここで 16μS ウエイトさせます。

    ; 2 回目のセルフ・プログラミング・モードへの移行で、動作が不定になります。
    BT      PFS.0,$ModeOnLoop       ; 移行完了確認

    ; コマンドの設定を行います。　フロー③
    MOV     A, #0FH
    MOV     FLAPH,A                 ;消去ブロック番号設定
    MOV     FLAPHC,A                ;消去ブロックのコンペア番号設定(FLAPH と同じ値)
    MOV     FLCMD,#03H              ;フラッシュ制御コマンド設定（ブロック消去）
    MOV     PFS,#00H                ;フラッシュ･ステータス･レジスタのクリア
    MOV     WDTE,#0ACH              ;WDT をクリア＆リスタート

    ; 消去コマンド実行　フロー④
    HALT                            ;セルフ･プログラミング実行

    ; 通常モードに移行するための特定シーケンスを実行 フロー⑤
ModeOffLoop:
    MOV     PFS,#00H
    MOV     PFCMD,#0A5H             ; PFCMD レジスタ制御
    MOV     FLPMC,#00H              ; FLPMC レジスタ制御(設定値)
    MOV     FLPMC,#0FFH             ; FLPMC レジスタ制御(設定値の反転)
    MOV     FLPMC,#00H              ; 通常モード設定
    BT      PFS.0,$ModeOffLoop      ; 移行完了確認

    POP     AX                      ;割り込みマスク復帰
    MOV     MK1,X
    XCH     A,X
    MOV     MK0,A
    BR      MAINLOOP
```

＜動作不定状態になるソースコード例（C 言語）＞

```
while(1){
        /* HALT を実行し、スタンバイ状態にします。フロー①、⑥ */
        HALT();

        /* 割り込みマスクを退避 */
        DI();                   // 割込み禁止
        ch_mask_bak0 = MK0;     // KU1+，KY1+は、MK0 のみ退避
        ch_mask_bak1 = MK1;     // ch_mask_bak0/1 は、退避用の変数です。

        /* セルフ・プログラミング・モードへ移行 フロー②、⑦ */

        do{
                PFS = 0;                // フラッシュ･ステータス･レジスタのクリア
                PFCMD = 0xA5;           // PFCMD レジスタ制御
                FLPMC = 0x01;           // FLPMC レジスタ制御(設定値)
                FLPMC = 0xFE;           // FLPMC レジスタ制御(設定値の反転)
                FLPMC = 0x01;           // セルフ･プログラミング･モード設定
                /* 外付けの発振子や外部クロック入力の場合、
                   ここで 16μS ウエイトさせます。*/

                /* 2 回目のセルフ・プログラミング・モードへの移行で、
                   動作が不定になります。 */
        }while(PFS.0 == 1);     // 移行完了確認

        /* コマンドの設定を行います。フロー③ */
        FLAPH = FLAPHC = 0x0F;          // 消去ブロックの指定
        FLCMD = 0x03;                   // 消去コマンドの指定
        PFS = 0x00;                     // フラッシュ･ステータス･レジスタのクリア
        WDTE = 0xAC;                    // WDT のカウンタをクリア

        /* 消去コマンド実行　フロー④ */
        HALT();                         // 消去コマンド実行

        /* 通常モードへ移行 フロー⑤ */
        do{
                PFS = 0;                // フラッシュ･ステータス･レジスタのクリア
                PFCMD = 0xA5;           // PFCMD レジスタ制御
                FLPMC = 0x00;           // FLPMC レジスタ制御(設定値)
                FLPMC = 0xFF;           // FLPMC レジスタ制御(設定値の反転)
                FLPMC = 0x00;           // 通常モード設定
        }while(PFS.0 == 1);     // 移行完了確認

        /* 割り込みマスクを復帰 */
        MK0 = ch_mask_bak0;
        MK1 = ch_mask_bak1;

}
```

＜回避策を実装した流れ図＞



①． 通常の HALT 命令を実行し、スタンバイ状態に入ります。その後、割込み等のスタンバイ・リリース信号により、スタンバイ状態を解除します。

②． セルフ・プログラミング・モードに移行する特定シーケンスを実行する前に FLCMD レジスタの値に 0 を設定します。また、CPU クロックを 1MHz 以上に設定します。

③．セルフ・プログラミング・モードに移行するための特定シーケンスを実行します。

④． セルフ・プログラミング・モードに移行するための特定シーケンスの処理（二回目の FLPMC に 1 を代入）のあとに NOP 命令と HALT 命令を実行します。このとき HALT 命令が解除される時間は、最大 10μS です。

⑤． 任意のセルフ・プログラミング・コマンド（書き込み、消去等）を FLCMD レジスタに設定します。

⑥．HALT 命令を実行することで、セルフ・プログラミング・コマンドが実行されます。

⑦． 通常モードに移行する特定シーケンスを実行する前に FLCMD レジスタの値に 0 を設定します。

⑧． 通常モードに移行するための特定シーケンスを実行します。このとき、CPU クロックを変更していた場合は、セルフ・プログラミング以前の設定に戻します。

⑨． 上記の②、④、⑦、⑧の処理を追加することにより、今回の不具合が回避されます。

＜回避策を実装したソースコード例（アセンブラ）＞

```
MAINLOOP:
    ; HALT を実行し、スタンバイ状態にします。フロー①、⑨
    HALT

    ; セルフ・プログラミングを実行するので、割り込みマスクを退避します。
    DI                              ; 割込み禁止
    MOV     A,MK0                   ; 割り込みマスクの退避
    XCH     A,X
    MOV     A,MK1                   ; KU1+, KY1+は、MK0 のみ退避
    PUSH    AX
    MOV     MK0, #0FFH
    MOV     MK1, #0FFH

    ; FLCMD レジスタの値を初期化します。フロー②
    MOV     FLCMD, #00H

    ; CPU クロック≦1MHz の場合、1MHz 以上に設定します。

    ; セルフ・プログラミング・モードに移行するための特定シーケンスを実行　フロー③
ModeOnLoop:
    MOV     PFCMD,#0A5H             ; PFCMD レジスタ制御
    MOV     FLPMC,#01H              ; FLPMC レジスタ制御(設定値)
    MOV     FLPMC,#0FEH             ; FLPMC レジスタ制御(設定値の反転)
    MOV     FLPMC,#01H              ; セルフ･プログラミング･モード設定

    ; NOP 命令、HALT 命令を実行します。 フロー④
    NOP
    HALT
    ; 外付けの発振子や外部クロック入力の場合、ここで 8μS ウエイトさせます。

    BT      PFS.0,$ModeOnLoop       ; 移行完了確認

    ; コマンドの設定を行います。　フロー⑤
    MOV     A, #0FH
    MOV     FLAPH,A                 ;消去ブロック番号設定
    MOV     FLAPHC,A                ;消去ブロックのコンペア番号設定(FLAPH と同じ値)
    MOV     FLCMD,#03H              ;フラッシュ制御コマンド設定（ブロック消去）
    MOV     PFS,#00H                ;フラッシュ･ステータス･レジスタのクリア
    MOV     WDTE,#0ACH              ;WDT をクリア＆リスタート

    ; 消去コマンド実行　フロー⑥
    HALT                            ;セルフ･プログラミング実行

    ; FLCMD レジスタを初期化します。 フロー⑦
    MOV     FLCMD, #00H

    ; CPU クロックを変更している場合、セルフ・プログラミング以前の状態に戻します。
```

```
    ; 通常モードに移行するための特定シーケンスを実行　フロー⑧
ModeOffLoop:
    MOV     PFS,#00H
    MOV     PFCMD,#0A5H             ; PFCMD レジスタ制御
    MOV     FLPMC,#00H              ; FLPMC レジスタ制御(設定値)
    MOV     FLPMC,#0FFH             ; FLPMC レジスタ制御(設定値の反転)
    MOV     FLPMC,#00H              ; 通常モード設定
    BT      PFS.0,$ModeOffLoop      ; 移行完了確認

    POP     AX                      ;割り込みマスク復帰
    MOV     MK1,X
    XCH     A,X
    MOV     MK0,A
    BR      MAINLOOP
```

＜回避策を実装したソースコード例（C 言語）＞

```
while(1){
        /* HALT を実行し、スタンバイ状態にします。フロー①、⑨ */
        HALT();

        /* 割り込みマスクを退避 */
        DI();                   // 割込み禁止
        ch_mask_bak0 = MK0;     // KU1+，KY1+は、MK0 のみ退避
        ch_mask_bak1 = MK1;     // ch_mask_bak0/1 は、退避用の変数です。

        /* FLCMD を初期化します。フロー② */
        FLCMD = 0;

        /* CPU クロック≦1MHz の場合、1MHz 以上に設定します。  */

        /* セルフ・プログラミング・モードへ移行 フロー③ */
        do{
                PFS = 0;                // フラッシュ･ステータス･レジスタのクリア
                PFCMD = 0xA5;           // PFCMD レジスタ制御
                FLPMC = 0x01;           // FLPMC レジスタ制御(設定値)
                FLPMC = 0xFE;           // FLPMC レジスタ制御(設定値の反転)
                FLPMC = 0x01;           // セルフ･プログラミング･モード設定
        }while(PFS.0 == 1);     // 移行完了確認

        /* NOP 命令、HALT 命令を実行します。 フロー④ */
        NOP();
        HALT();
        /* 外付けの発振子や外部クロック入力の場合、ここで 8μS ウエイトさせます。*/

        /* コマンドの設定を行います。フロー⑤ */
        FLAPH = FLAPHC = 0x0F;          // 消去ブロックの指定
        FLCMD = 0x03;                   // 消去コマンドの指定
        PFS = 0x00;                     // フラッシュ･ステータス･レジスタのクリア
        WDTE = 0xAC;                    // WDT のカウンタをクリア

        /* 消去コマンド実行  フロー⑥ */
        HALT();                         // 消去コマンド実行

        /* FLCMD レジスタを初期化します。 フロー⑦ */
        FLCMD = 0;

        /* CPU クロックを変更している場合、
          セルフ・プログラミング以前の状態に戻します。 */

        /* 通常モードへ移行 フロー⑧ */
        do{
                PFS = 0;                // フラッシュ･ステータス･レジスタのクリア
                PFCMD = 0xA5;           // PFCMD レジスタ制御
                FLPMC = 0x00;           // FLPMC レジスタ制御(設定値)
                FLPMC = 0xFF;           // FLPMC レジスタ制御(設定値の反転)
```

```
        FLPMC = 0x00;           // 通常モード設定
    }while(PFS.0 == 1);     // 移行完了確認

    /* 割り込みマスクを復帰 */
    MK0 = ch_mask_bak0;
    MK1 = ch_mask_bak1;

}
```