

## 1. 要約

この資料はM16C/63、64、64A、64C、65(プログラムROM1が512Kバイト以下の製品)、65C、6C、5LD、56D、5L、56、5M、57グループのユーザブート機能を使用したマイコン内蔵フラッシュメモリ(データフラッシュ、プログラムROM1)の書き換え方法を掲載しています。

ユーザブート機能は、ユーザが任意の通信方法でマイコン内蔵フラッシュメモリ(データフラッシュ、プログラムROM1)を書き換えるための機能です。

(ユーザアプリケーションプログラムのスタートアップの機能ではありません。)

以下、M16C/64を例に挙げて説明します。

## 2. はじめに

この資料で説明する応用例は次のマイコン、条件での使用に適用します。

- マイコン：M16C/63グループ
  - M16C/64グループ
  - M16C/64Aグループ
  - M16C/64Cグループ
  - M16C/65(プログラムROM1が512Kバイト以下の製品)グループ(注1)
  - M16C/65Cグループ
  - M16C/6Cグループ
  - M16C/5LDグループ
  - M16C/56Dグループ
  - M16C/5Lグループ
  - M16C/56グループ
  - M16C/5Mグループ
  - M16C/57グループ
- 発振周波数：8MHz
- 動作周波数：24MHz(CPU書き換えモード時は8MHz)

注1. プログラムROM1が512Kバイトを超える製品は、ソフトウェアコマンドが異なります。  
詳しくはハードウェアマニュアルを参照してください。

このアプリケーションノートに掲載するプログラムの使用に際しては十分な評価を行ってください。

## 3. 応用技術の説明

### 3.1 概要

本アプリケーションノートでは、ホストPCからシリアル通信を用いてマイコンのフラッシュメモリ書き換えを制御します。なお、MOTOROLA-Sフォーマットの転送はXmodemを使用します。また、本アプリケーションノートではフラッシュメモリの書き換えにおいて、通常の書き込み、ロックビットの状態を無視する消去、ロックビットの状態を無視しない消去の3種類の方法を用意しています。

### 3.2 ユーザブート機能の概要

M16C/63、64、64A、64C、65(プログラムROM1が512Kバイト以下の製品)、65C、6C、5LD、56D、5L、56、5M、57グループはブートモードにユーザブート機能を有しています。ブートモード(P5\_5端子に“L”、CNVSS端子に“H”、P5\_0端子に“H”)を入力してハードウェアリセットし動作するモード)で起動するとき、ユーザブートコード領域の状態により標準シリアル入出力モードまたはユーザブートモードを選択できます。

ユーザブートモードでは、フラッシュメモリのプログラムROM2領域(10000h番地~)に書き込んだプログラムを実行します。

ユーザブートモードでのフラッシュメモリ書き換え処理の特長を以下に示します。

- 使用するポート(端子)を自由に設計可能  
標準シリアル入出力モードではUART1を使用します。
- シリアルライター(注1)を自由に設計可能  
通信方法、送受信データ(コマンド、プログラムコード)の内容、通信タイミングなどのプロトコル設計も可能です。

注1. 標準シリアル入出力モードでは M16C/63、64、64A、64C、65、65C、6C、5LD、56D、5L、56、5M、57グループに対応したシリアルライターが必要です。

### 3.3 プログラムROM2のメモリ配置

図 3.1にプログラムROM2のユーザブートプログラム領域とユーザブートコード領域を示します。

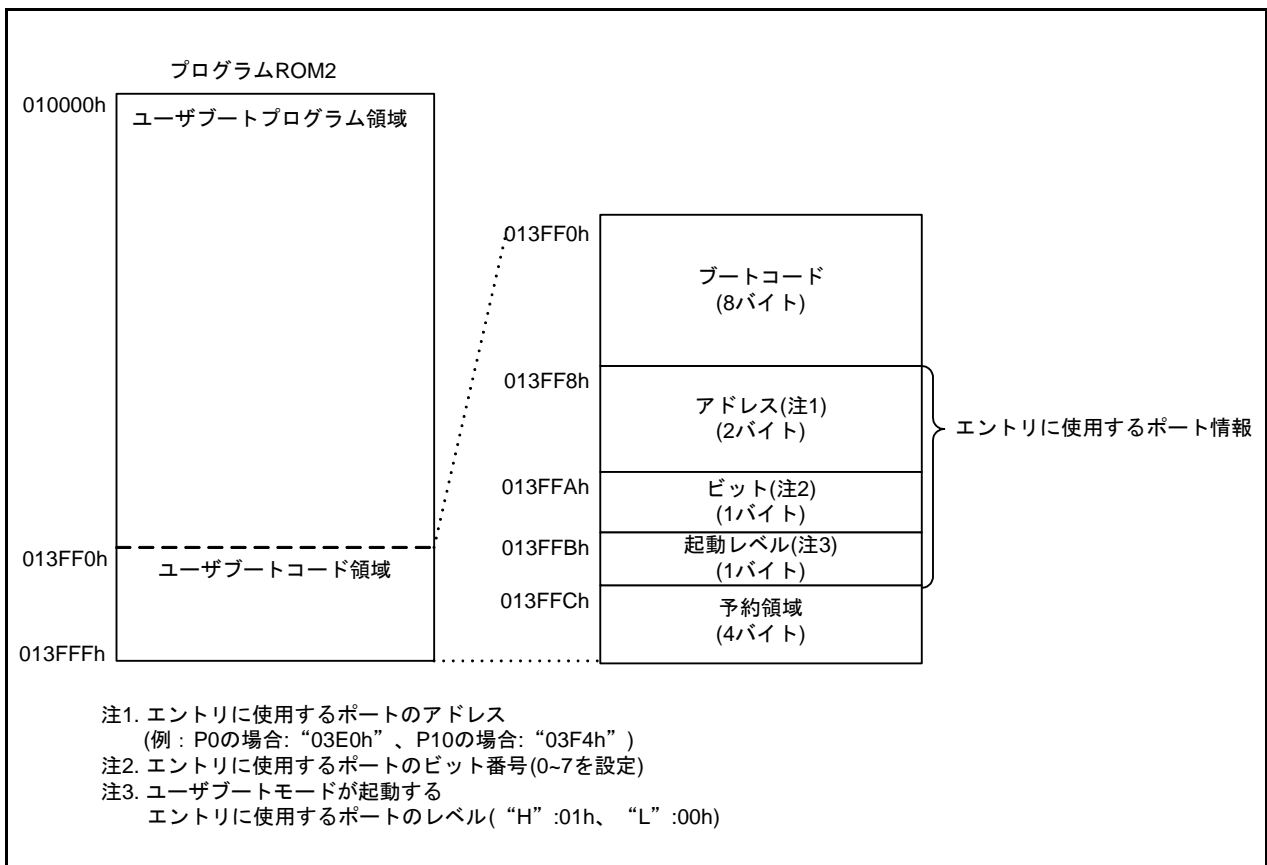


図 3.1 ユーザブートプログラム領域とユーザブートコード領域

### 3.4 ユーザブートコード領域の設定値と起動モード

エントリに使用するポートをポートPi<sub>j</sub> (i = 0~10, j = 0~7)とした場合を説明します。表 3.1にユーザブートコード領域の設定値と起動モードを示します。表 3.1 がない値、組み合わせを設定しないでください。

表 3.1 ユーザブートコード領域の設定値と起動モード

ブートコード (13FF0h~13FF7h番地)	エントリに使用するポート情報			ポート 入力レベル (ポートPi <sub>j</sub> )	起動モード
	アドレス (13FF8h~ 13FF9h番地)	ビット (13FFAh番地)	起動レベル (13FFBh番地)		
"UserBoot"(注1)	0000000h			—	ユーザブート
	Piレジスタの 番地(注2)	00h~07h (jの値)	00h	"H"	標準シリアル入出力
			01h	"L"	ユーザブート
	—	—	—	"H"	標準シリアル入出力
—	—	—	"L"	標準シリアル入出力	
"UserBoot"以外	—	—	—	—	標準シリアル入出力

注1. 「表 3.2 "UserBoot"のASCIIコード」を参照してください。

注2. 「表 3.3 エントリに使用できるポートのアドレス」を参照してください。

表 3.2 "UserBoot"のASCIIコード

番地	13FF0h	13FF1h	13FF2h	13FF3h	13FF4h	13FF5h	13FF6h	13FF7h
ASCIIコード	55h (U大文字)	73h (s小文字)	65h (e)	72h (r)	42h (B)	6Fh (o小文字)	6Fh (o小文字)	74h (t)

表 3.3 エントリに使用できるポートのアドレス

ポート	Piレジスタの番地
P0	03E0h
P1	03E1h
P2	03E4h
P3	03E5h
P4	03E8h
P5 (注1)	03E9h
P6	03ECh
P7	03EDh
P8	03F0h
P9	03F1h
P10	03F4h

注1. 次のいずれかの条件の場合、ブートコードに "UserBoot" を設定するとブートモード起動時には常にユーザブートモードを選択します。

- ・エントリに使用するポート情報にP5\_0を選択し、かつ起動レベルが"H"
- ・エントリに使用するポート情報にP5\_5を選択し、かつ起動レベルが"L"

### 3.4.1 ユーザブートコード領域の設定例

ユーザブートコード領域はユーザブートプログラムのROMDATAとして設定します。以下にユーザブートコード領域の設定例を示します。

- セクション名を“ubtcd\_sec”、  
   エントリポートはポートP10 (P10)、  
   エントリビットはビット0 (P10\_0)、  
   起動レベルは“H”とした場合のユーザブートコード領域の設定例

<セクション定義>

- アセンブリ言語でアドレス指示を行う場合

```

;-----
; User boot code area section.
;-----
.section          ubtcd_sec, ROMDATA
.org              013FF0H
.section          ubtcd_sec_FE, ROMDATA, ALIGN

```

- リンカージェディタ (ln30) でアドレス指示を行う場合

ln30へのコマンドオプションに“-ORDER ubtcd\_sec\_FE=13FF0”を追加します。

<ユーザブートコード領域定義>

```

/*-----*/
/* define of "User boot code area" */
/*-----*/
typedef struct UBTCDEF {
    unsigned char    btcd[8];           /* Boot-code */
    unsigned short   eptaddr;          /* SFR address of entry port */
    unsigned char    eptbitn;          /* Bit of the entry port */
    unsigned char    exptlvl;          /* Enable port level */
    unsigned long    ubtrsv;           /* Reserved area */
} ubtcd_def;
#pragma SECTION rom ubtcd_sec          /* The section name "rom" is changed to "ubtcd_sec". */
const far ubtcd_def UserBootCode = { {'U','s','e','r','B','o','o','t'}, /* Boot-code = "UserBoot" */
    0x03F4,                               /* Entry port = "P10" */
    0x00,                                   /* Entry port bit = bit0 */
    0x01,                                   /* Enable port level = "H" */
    0xFFFFFFFF                               /* Reserved area */
};
#pragma SECTION rom rom                 /* The section name is returned to "rom". */

```

### 3.5 ユーザブートプログラムの留意点

ユーザブートプログラムコードはプログラムROM2領域に収まるように作成してください。また、ユーザブートモードで実行する先頭番地は10000h番地です。

オンチップデバッガなどでユーザブートプログラムのデバッグを行う場合、リセットベクタに10000h番地を設定して、シングルチップモードで立ち上げてください。

また、リセットベクタを含むブロックは消去しないでください。

#### 3.5.1 フラッシュメモリ書き換え

フラッシュメモリ(データフラッシュ、プログラムROM1)の書き換えは、CPU書き換えモードのEW0モード、EW1モードいずれも使用できます。

表3.4にフラッシュメモリ書き換えに関する制約事項と対処方法を示します。図3.2にユーザブートプログラムの再配置を、図3.3にフラッシュメモリ書き換え動作の概念図を示します。

表 3.4 フラッシュメモリ書き換えに関する制約事項と対処方法

制約事項	対処方法
フラッシュメモリはブロック単位でしか消去できません。(フラッシュメモリに書き込み、消去回数の制限がある)	フラッシュメモリへの書き込みは、ブロックごとの消去(イレーズ)回数を減らすよう考慮してください。

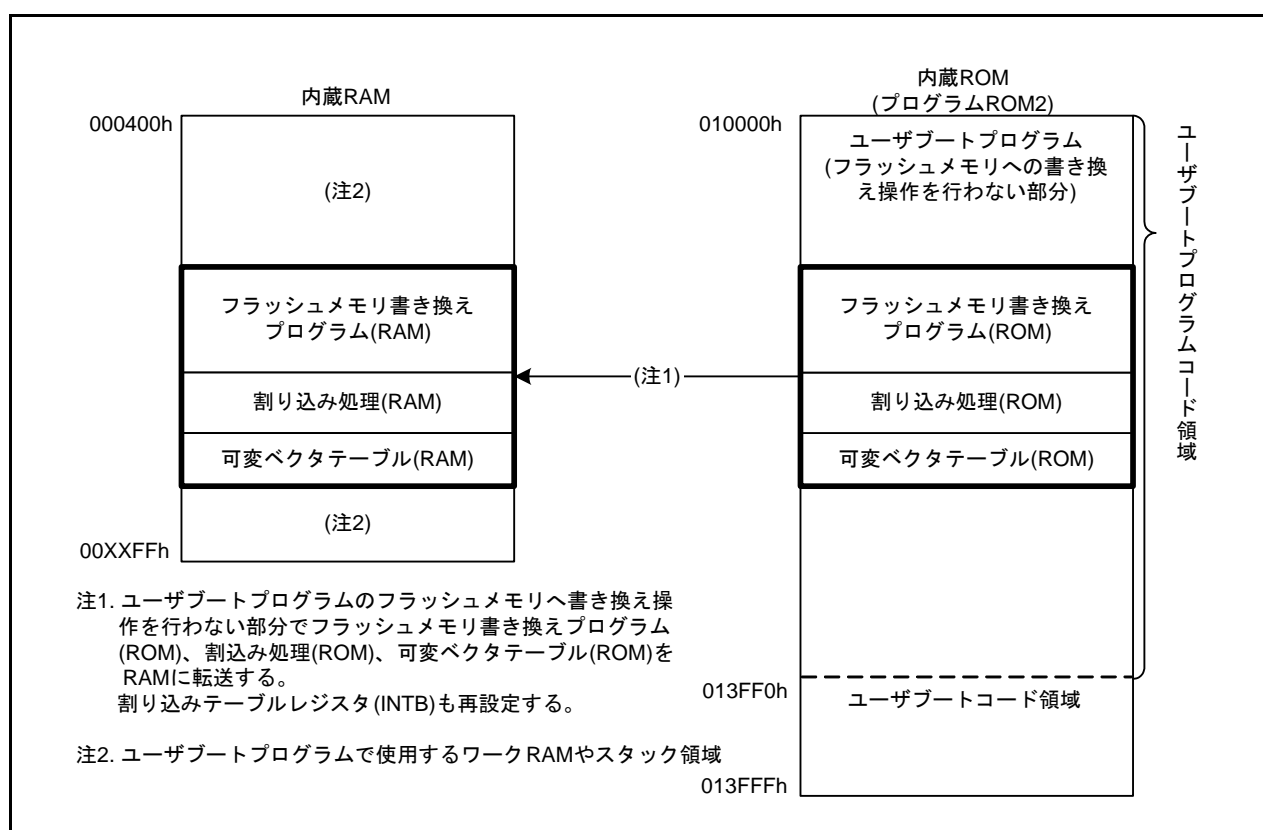


図 3.2 ユーザブートプログラムの再配置

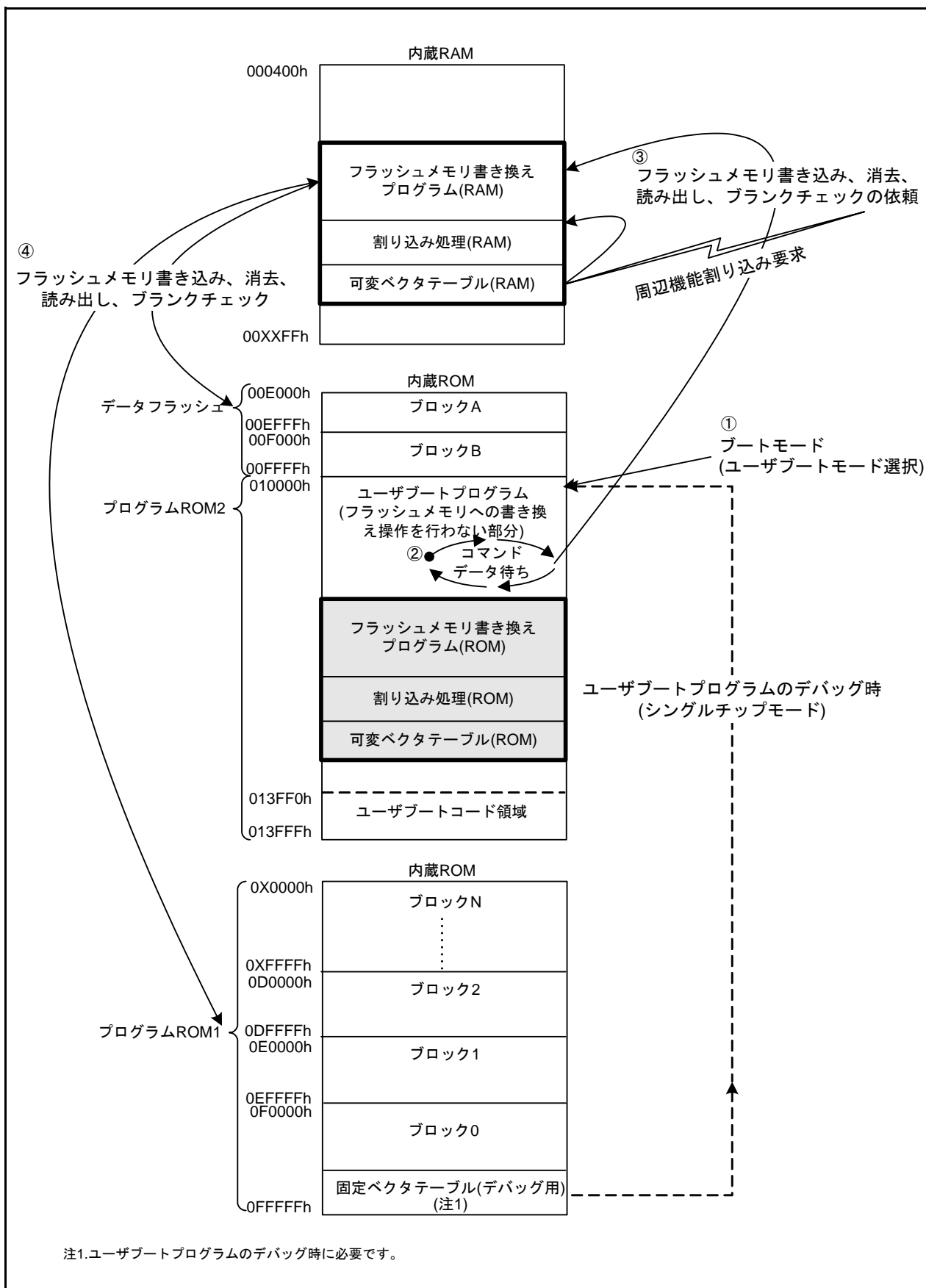


図 3.3 フラッシュメモリ書き換え動作の概念図

### 3.5.2 ユーザブートプログラムの動作速度

CPU書き換えモード中はCPUクロックを10MHz以下にしてください。また、内部RAM、内部ROMアクセスはPM1レジスタのPM17ビットを“1”(1ウェイトあり)にしてください。

### 3.5.3 ユーザブートモード選択時のマイコン状態

SFRはリセット後の値、内蔵RAMは不定です。(ソフトウェアリセット時も内蔵RAMは不定です。)

図 3.4にユーザブート選択時のCPUレジスタ状態を示します。

表 3.5 CPUリセット後とユーザブートモード選択時に状態が変化するSFR一覧

番地	レジスタ	シンボル	リセット後の値	ユーザブートモード選択時の値
0220h	フラッシュメモリ制御レジスタ0	FMR0	0000 0001b	0010 0001b(注1)

注1. FMR0レジスタを操作する場合、ビット5を“1”にしてください。

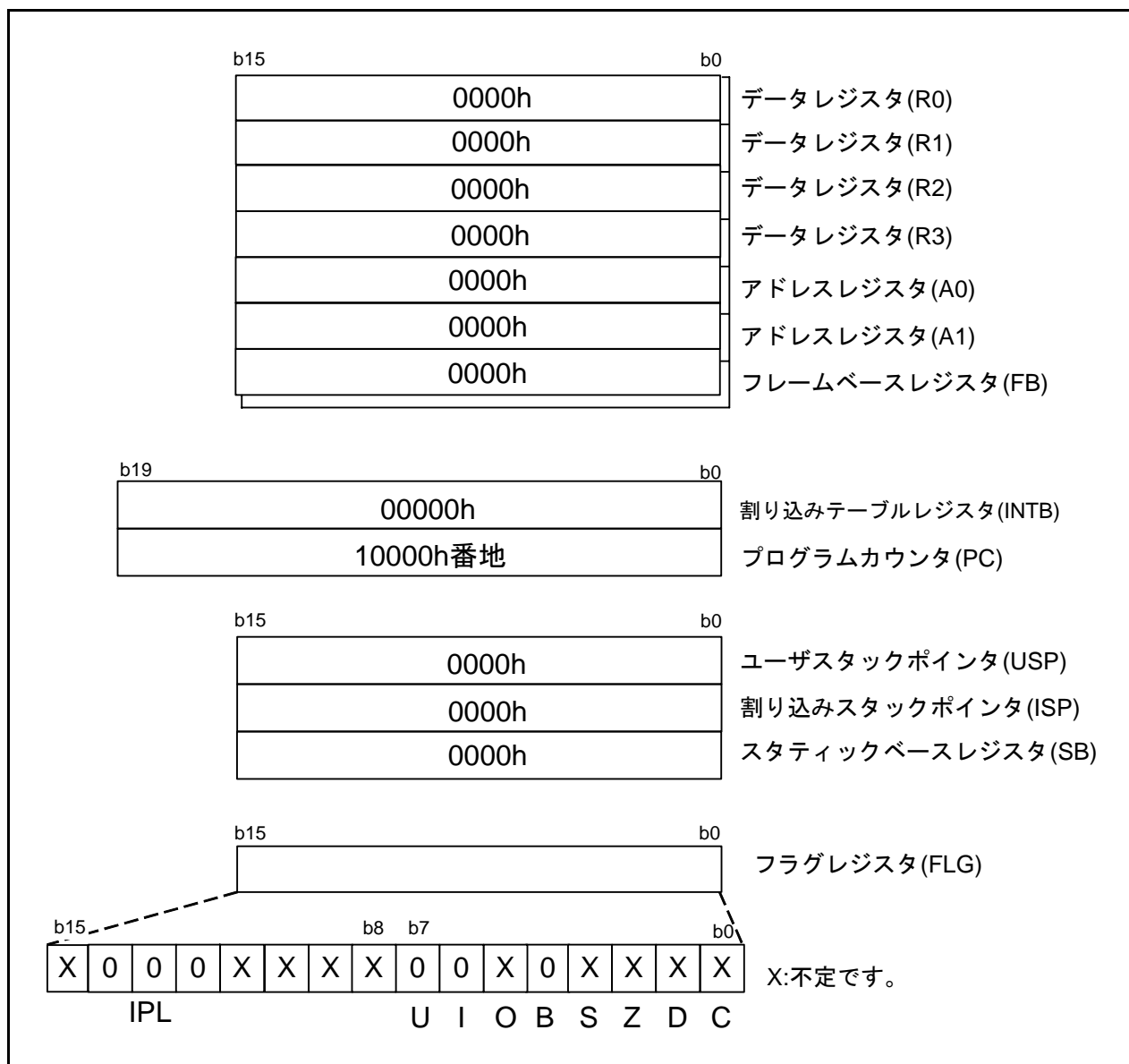


図 3.4 ユーザブート選択時のCPUレジスタ状態



### 3.5.4 ウォッチドッグタイマのリフレッシュ処理

プログラムROM1のOFS1(FFFFFFh番地)の設定がWDTON="0"(リセット後、ウォッチドッグタイマは自動起動)などウォッチドッグタイマをスタートする場合、ユーザブートプログラムでウォッチドッグタイマの初期化を実施してください。

### 3.6 ユーザブートモード選択までの処理遷移

図 3.5 にブートモードエントリからユーザブートモード選択までの処理遷移を示します。

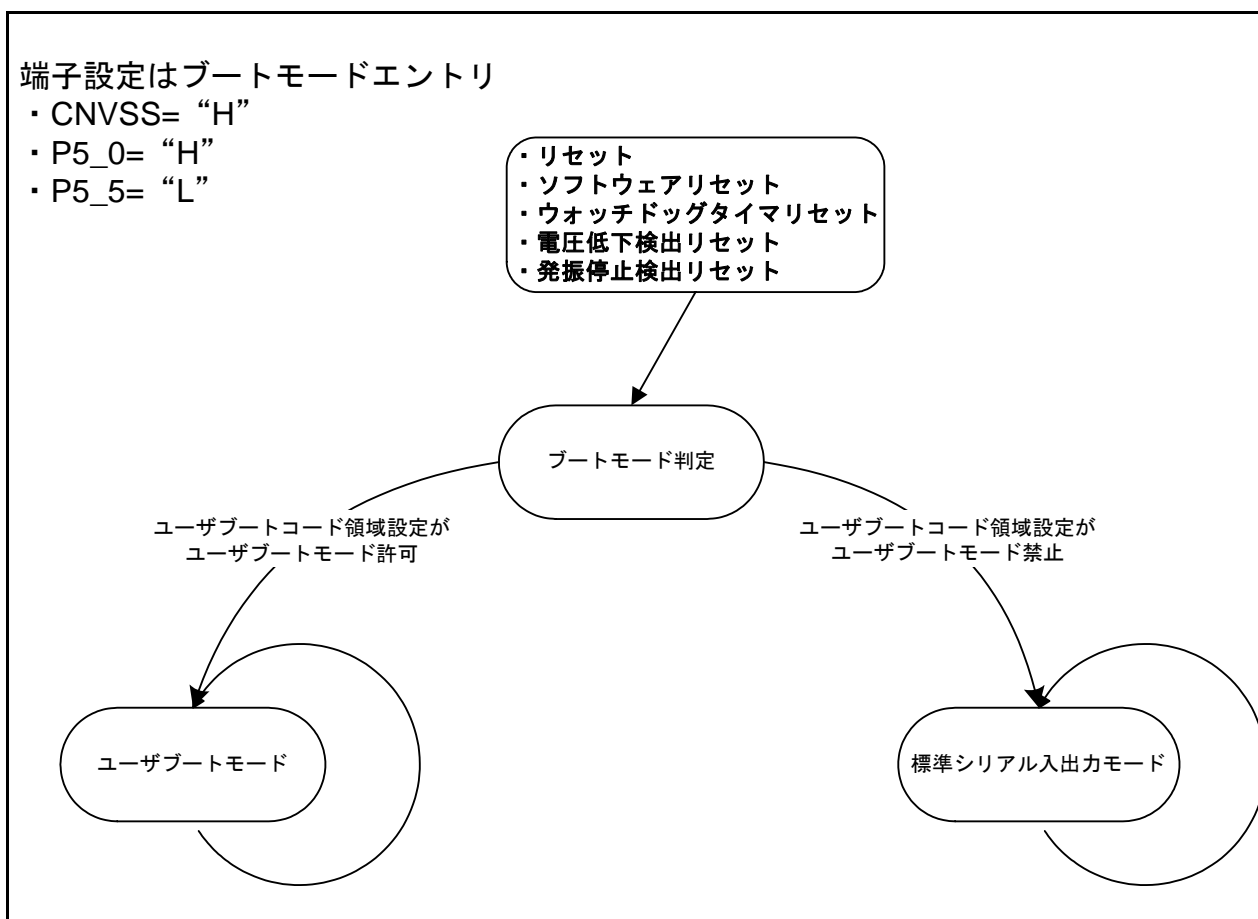


図 3.5 ユーザブートモード選択までの処理遷移

## 4. 参考ユーザブートプログラムの説明

### 4.1 概要

M16C/63、64、64A、64C、65(プログラムROM1が512Kバイト以下の製品)、65C、6C、5LD、56D、5L、56、5M、57グループのCPU書き換えモード(EW0モード)と、ユーザブート機能を使用し、内蔵フラッシュメモリの書き換えを行います。

書き換え対象

- データフラッシュ (ブロック A、B)
- プログラムROM1

フラッシュメモリに対する処理の選択はホストPCで起動するターミナルソフト(注1)で通信します。内蔵フラッシュメモリに書き込むデータはXMODEMプロトコルでMOTOROLA-Sフォーマット(注2)のテキストデータを通信します。

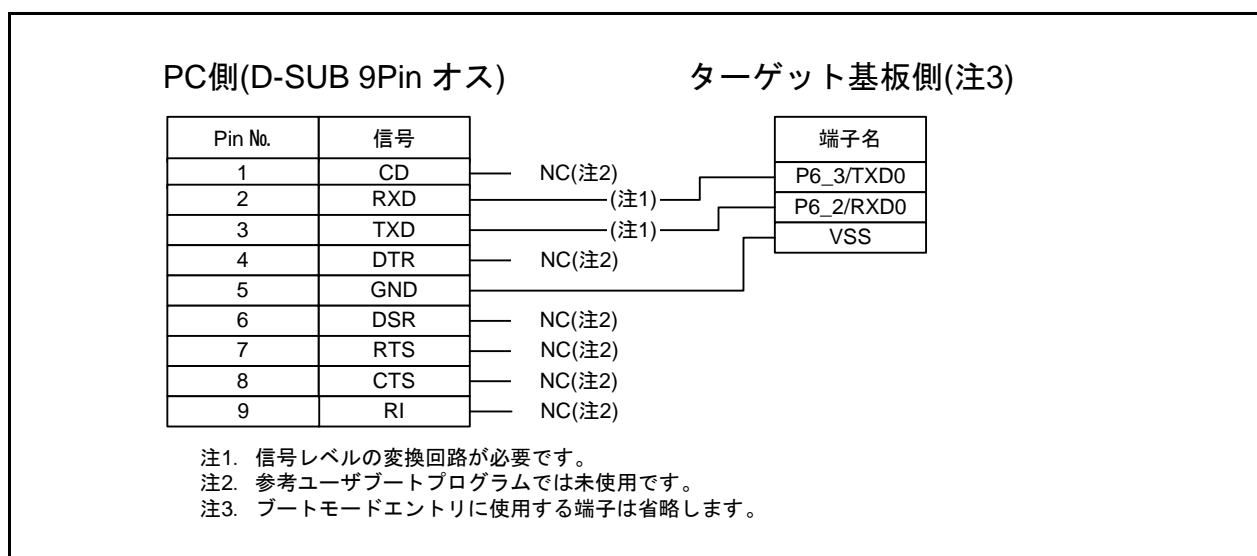
受信したMOTOROLA-Sフォーマットのデータをバイナリ形式に変換した後、内蔵フラッシュメモリに書き込みます。

注1. ターミナルソフトの通信設定は、115200bps、データ長8ビット、パリティなし、1ストップビット、フロー制御なしとしてください。

注2. S1レコード(16ビットのアドレス長)、S2レコード(24ビットのアドレス長)のみ書き込みデータ対象として処理します。

### 4.2 ホストPCとの接続

マイコンを搭載するターゲット基板と、ホストPCをRS-232Cケーブルで接続します。



## 4.2.1 使用端子

表 4.1 使用端子と機能

分類	端子名	入出力	電源系統	備考
電源	VCC1	入力	—	電源電圧
	VCC2	入力	—	電源電圧
	VSS	入力	—	0Vの入力
リセット入力	RESET	入力	VCC1	—
CNVSS	CNVSS	入力	VCC1	VCC1の入力(注1)
バス制御端子	P5_0(CE)/WRL/WR	入力	VCC2	VCC2の入力(注1)
	P5_5(EPM)/HOLD	入力	VCC2	VSSの入力(注1)
シリアルインタフェース UART0	P6_2/RXD0/SCL0	入力	VCC1	シリアルデータ入力
	P6_3/TXD0/SDA0	出力	VCC1	シリアルデータ出力
入出力ポート	P10_0	入力	VCC1	ユーザブートモードの選択用VCC1の入力(注2)

注1. ブートモードのエントリ設定です。

注2. 参考ユーザブートプログラムでの設定値。詳細は「表 3.3 エントリに使用できるポートのアドレス」を参照してください。

## 4.3 処理概要

ターミナルソフト通信で選択する処理を表 4.2 に示します。各処理ではタイムアウトを設けています。一定時間入力しなかった場合、対象コマンドはキャンセルされます。キャンセルされた場合、最初からコマンドを入力し直してください。

表 4.2 処理一覧

番号	処理	概要
1	ブランクチェック	内蔵フラッシュメモリのブランクチェックをブロック単位で行います。ブランクチェック対象はデータフラッシュ(ブロックA、B)、プログラムROM1(ブロック0~3)です。
2	イレーズ	ロックビットが非ロック状態のとき、内蔵フラッシュメモリの消去を行います。消去対象はデータフラッシュ(ブロックA、B)、プログラムROM1(ブロック0~3)です。
3	選択ブロックのイレーズ	ロックビットが非ロック状態のとき、選択した内蔵フラッシュメモリのブロックを消去します。消去選択対象はデータフラッシュ(ブロックA、B)、プログラムROM1(ブロック0~3)です。
4	送信ファイルのプログラム	MOTOROLA-Sフォーマットデータ(注1)をXMODEMプロトコルで通信し、内蔵フラッシュメモリに書き込みます。
5	チェックサムとCRCの計算	データフラッシュ(ブロックA、Bの8Kバイト)とプログラムROM1(ブロック0~3の256Kバイト)のチェックサムとCRCをそれぞれ計算します。
6	ロックビットのロック設定	内蔵フラッシュメモリをブロック単位でロック状態にします。ロック対象はデータフラッシュ(ブロックA、B)、プログラムROM1(ブロック0~3)です。
7	ロックビットステータスの表示	ロックビットの状態を読み出し、ブロック単位で表示します。読み出す対象はデータフラッシュ(ブロックA、B)、プログラムROM1(ブロック0~3)です。
8	選択ブロックのロックビットおよびデータのイレーズ	選択したブロックをロックビットの状態にかかわらず消去します。消去したブロックのロックビットも消去されます。

注1. ブロックA、B、0~3以外に配置しないでください。

### 4.3.1 ユーザインタフェース

ターミナルソフト通信でホストPCの画面に表示する内容と選択値について説明します。

なお、通信設定やXMODEM通信プロトコルによるMOTOROLA-Sフォーマットのファイル送信操作は使用するターミナルソフトの操作方法などを参照してください。

(1) メニュー表示

「表 4.2 処理一覧」の番号に対応した処理を“1”~“8”の数字で選択します。

選択処理番号(1~8)以外を入力すると、再びメニューを表示します。

ターミナルソフト画面
M16C/64 User Boot Menu v2.00 ----- 1...Blank Check 2...Erase 3...Erase of selection block 4...Program Flash via XModem Download 5...Checksum calculation and CRC calculation 6...Lock of lock bit 7...Display of lock bit status 8...Unlock of lock bit and erase of selection block >

## (2) ブランクチェック

“1”キーを入力すると内蔵フラッシュメモリのブロックA、B、0~3がブランク(書き込みデータなし)かどうかチェックします。チェック結果はブロックごとに表示します。

ターミナルソフト画面
5...Checksum calculation and CRC calculation
6...Lock of lock bit
7...Display of lock bit status
8...Unlock of lock bit and erase of selection block
>1
Blank checking user area...
Block-A is a blank.
Block-B is a blank.
Block-0 is a blank.
Block-1 is not a blank.
Block-2 is a blank.
Block-3 is a blank.
Blank check is done.

Blank (書き込みデータなし)時の表示

非ブランク(書き込みデータあり)時の表示

## (3) イレーズ

“2”キーを入力すると、内蔵フラッシュメモリのブロックA、B、0~3の消去(イレーズ)を実行します。消去結果をブロックごとに表示します。ロックビットでロックされているブロックの消去(イレーズ)は失敗となります。非ロック状態でブランクのブロックの場合は消去(イレーズ)を実行せずに、消去成功を表示します。

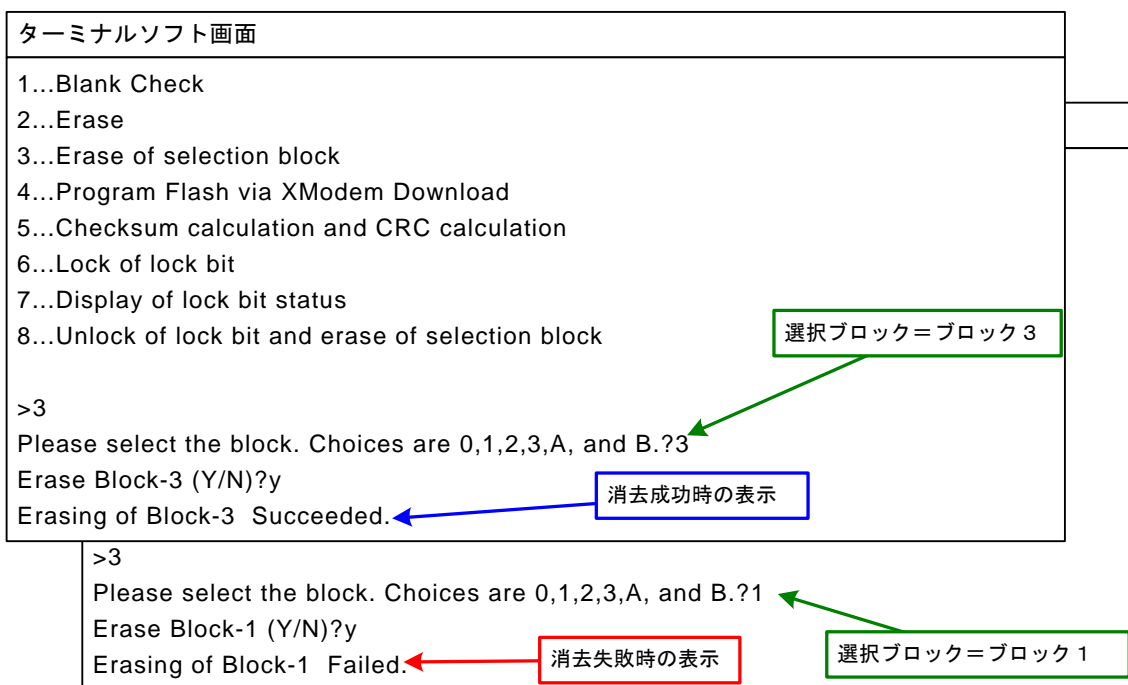
ターミナルソフト画面	
5...Checksum calculation and CRC calculation	
6...Lock of lock bit	
7...Display of lock bit status	
8...Unlock of lock bit and erase of selection block	
>2	
Really erase ALL user blocks (Y/N)?y	
Erasing of Block-A ... Succeeded.	} → 消去成功時の表示
Erasing of Block-B ... Succeeded.	
Erasing of Block-0 ... Succeeded.	
Erasing of Block-1 ... Failed.	
Erasing of Block-2 ... Failed.	
Erasing of Block-3 ... Succeeded.	
Erasing all blocks ended.	

## (4) 選択ブロックのイレーズ

“3”キーを入力すると選択された内蔵フラッシュメモリのブロックA、B、0~3から選択した1ブロックを消去(イレーズ)します。ロックビットでロックされているブロックの消去(イレーズ)は失敗となります。非ロック状態でブランクのブロックの場合は消去(イレーズ)を実行せずに、消去成功を表示します。ブロックの選択キーは表 4.3に示します。

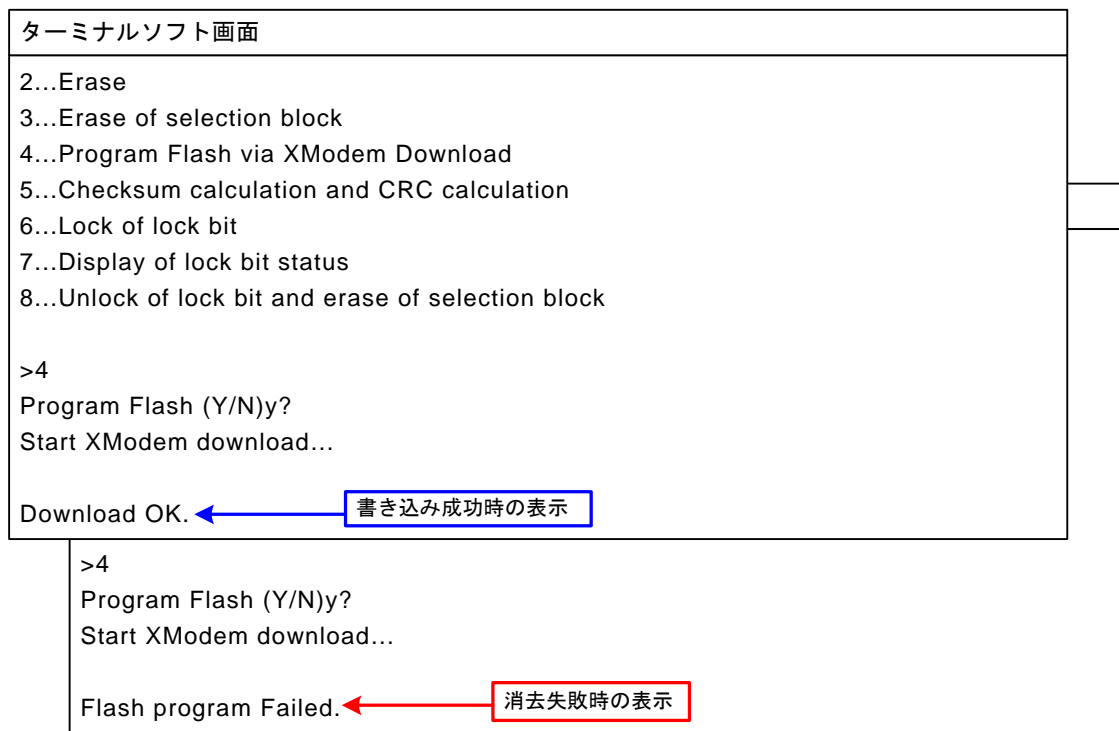
表 4.3 ブロック選択キー

ブロック種別	ブロック名	サイズ	アドレス	選択キー
データフラッシュ	ブロックA	4Kバイト	E000h~EFFFh	A
	ブロックB	4Kバイト	F000h~FFFFh	B
プログラムROM1	ブロック3	64Kバイト	C0000h~CFFFFh	3
	ブロック2	64Kバイト	D0000h~DFFFFh	2
	ブロック1	64Kバイト	E0000h~EFFFFh	1
	ブロック0	64Kバイト	F0000h~FFFFFh	0



## (5) 送信ファイルのプログラム

“4”キーを入力すると内蔵フラッシュメモリに書き込む準備をします。書き込むファイルを送信してください。通信設定やXMODEMによるファイル送信操作は、使用するターミナルソフトのマニュアルなどを参照してください。書き込む内蔵フラッシュメモリ(ブロックA、B、0~3)がロックビットでロックされているか非ブランクの場合、書き込みは失敗します。





## (6) チェックサムとCRCの計算

“5”キーを入力すると内蔵フラッシュメモリのブロックA、B、0~3のチェックサムとCRC演算を実行します。チェックサムとCRC演算の結果はデータフラッシュ (E000h~FFFFh番地)、プログラムROM1(C0000h~FFFFFh番地)ごとに2バイトで表示します。

ターミナルソフト画面	
1...Blank Check	
2...Erase	
3...Erase of selection block	
4...Program Flash via XModem Download	
5...Checksum calculation and CRC calculation	
6...Lock of lock bit	
7...Display of lock bit status	
8...Unlock of lock bit and erase of selection block	
>5	
Sum calculation user area	チェックサム値の表示
Data flash... Sum is F000h, Crc is 85F4h	CRC値の表示
Program ROM... Sum is 0000h, Crc is 8444h	
Sum calculation is done.	

## (7) ロックビットのロック設定

“6”キーを入力すると内蔵フラッシュメモリのブロックA、B、0~3にロックビットプログラムを実行し、ロックビットを“0”(ロック状態)にします。ロックビットプログラムの結果をブロックごとに表示します。ロックビットがすでに“0”のブロックの場合は失敗を表示します。

ターミナルソフト画面	
5...Checksum calculation and CRC calculation	
6...Lock of lock bit	
7...Display of lock bit status	
8...Unlock of lock bit and erase of selection block	
>6	
The lock bit is put into the state of the lock.	
Lock of Block-A Failed.	ロックビットプログラムの失敗時の表示
Lock of Block-B Failed.	
Lock of Block-0 Succeeded.	ロックビットプログラムの正常動作
Lock of Block-1 Succeeded.	
Lock of Block-2 Succeeded.	
Lock of Block-3 Succeeded.	
Block lock bit program is done.	

## (8) ロックビットステータスの表示

“7”キーを入力すると内蔵フラッシュメモリのブロックA、B、0~3のロックビットステータスをリードします。ロックビットステータスの結果をブロックごとに表示します。

ターミナルソフト画面	
5...Checksum calculation and CRC calculation	
6...Lock of lock bit	
7...Display of lock bit status	
8...Unlock of lock bit and erase of selection block	
>7	
The state of the lock bit is displayed.	
Block-A is unlocked.	} ← 非ロック状態の表示
Block-B is unlocked.	
Block-0 is locked.	} ← ロック状態の表示
Block-1 is locked.	
Block-2 is locked.	
Block-3 is locked.	
Read Block lock bit status is done.	

## (9) 選択ブロックのロックビットおよびデータのイレース

“8”キーを入力するとロックビットのロックを無効にして、内蔵フラッシュメモリのブロックA、B、0~3から選択して、対象の1ブロックを消去(イレース)します。このとき、ロックビットも消去され、ロックビットステータスが“1”(非ロック状態)になります。非ロック状態でブランクのブロックの場合は消去(イレース)を実行せずに、消去成功を表示します。ブロックの選択キーは表 4.3を参照してください。

ターミナルソフト画面	
1...Blank Check	
2...Erase	
3...Erase of selection block	
4...Program Flash via XModem Download	
5...Checksum calculation and CRC calculation	
6...Lock of lock bit	
7...Display of lock bit status	
8...Unlock of lock bit and erase of selection block	
>8	
Please select the block. Choices are 0,1,2,3,A, and B.?0	
Unlock and erase Block-0 (Y/N)?y	
Unlock and erasing of Block-0 Succeeded.	← ロックビットおよびデータの消去成功時の表示

### 4.3.2 使用例

参考ユーザブートプログラムを用いた内蔵フラッシュメモリの書き換え例のフローチャートを図4.1に示します。

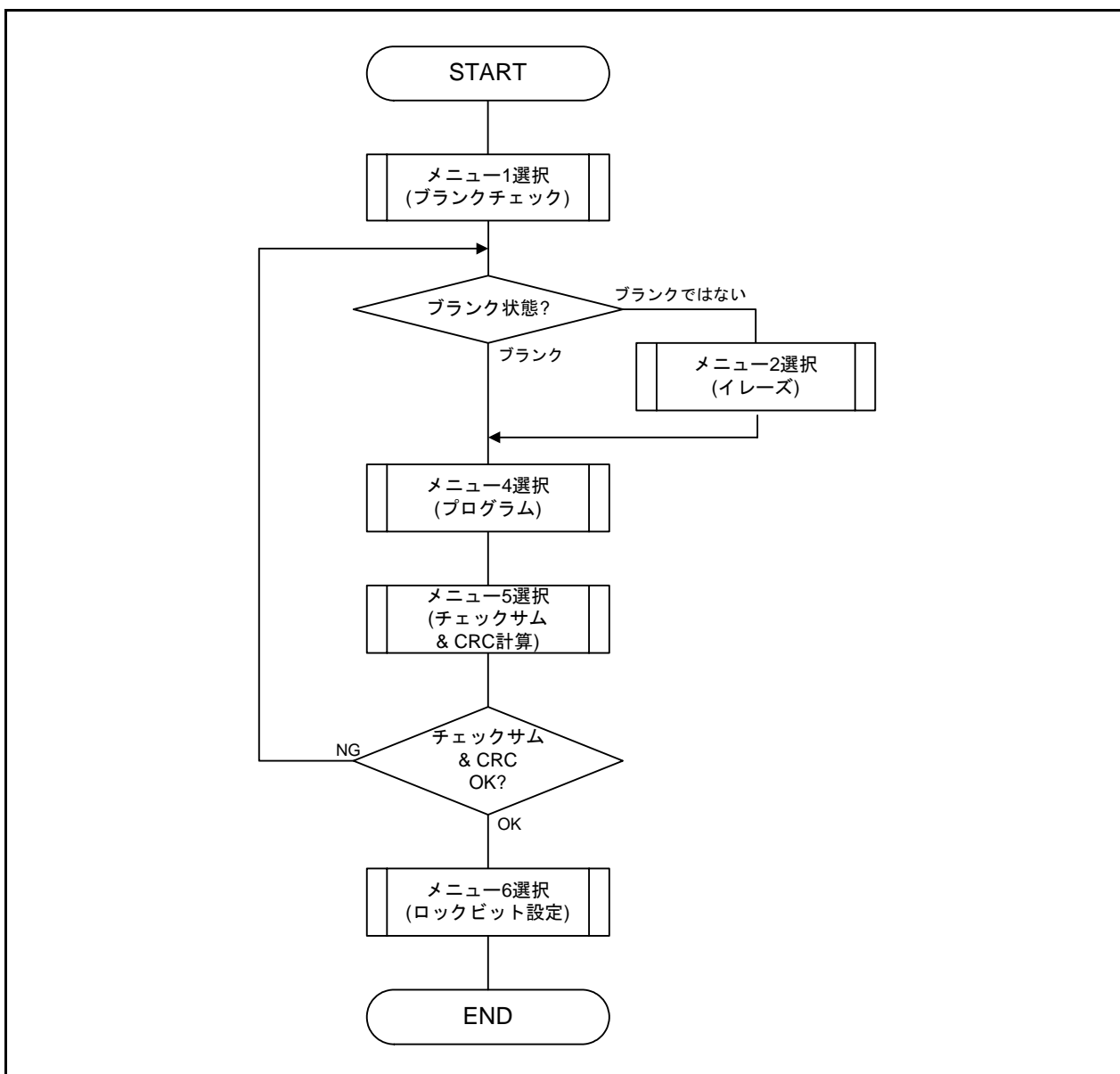


図 4.1 参考ユーザブートプログラムの使用例

## 4.4 プログラム説明

### 4.4.1 ファイル構成

参考ユーザブートプログラムを構成するプログラムのファイル一覧を表 4.4に示します。

表 4.4 ファイル一覧

ファイル名	解説	備考
common.c	共通処理モジュール	
common.h	共通処理モジュールの外部参照用インクルードヘッダ	型定義、共通関数のプロトタイプ宣言
lowlevelinit.c	マイコン初期化モジュール	
lowlevelinit.h	マイコン初期化モジュールの外部参照用インクルードヘッダ	
int_dmy.c	未定義割り込み(ダミー)処理	
int_dmy.h	未定義割り込み処理のプロトタイプ宣言	
flash_drv.c	CPU書き換えモード(EW0モード)での内蔵フラッシュメモリ書き換えモジュール	
flash_drv.h	内蔵フラッシュメモリ書き換えモジュールの外部参照用インクルードヘッダファイル	
serial0_drv.c	シリアル(UART0)データ通信モジュール	
serial0_drv.h	シリアル(UART0)データ通信モジュールの外部参照用インクルードヘッダ	
timerA0_drv.c	タイマA0(タイマモード)操作モジュール	タイムアウト時間計測用
timerA0_drv.h	タイマA0(タイマモード)操作モジュールの外部参照用インクルードヘッダ	
timerB5_drv.c	タイマB5(タイマモード)操作モジュール	ウォッチドッグタイマリフレッシュ周期計測用
timerB5_drv.h	タイマB5(タイマモード)操作モジュールの外部参照用インクルードヘッダ	
wdtRefresh.h	ウォッチドッグタイマリフレッシュマクロ定義	
xmodem.c	XMODEM通信プロトコル処理モジュール	
xmodem.h	XMODEM通信プロトコル処理モジュールの外部参照用インクルードヘッダ	
flash_menu.c	ユーザブートメニュー処理モジュール	サンプルソースファイルの使用例
flash_menu.h	ユーザブートメニュー処理モジュールの外部参照用インクルードヘッダ	
UserBoot.c	ユーザブートメイン処理	User boot code領域の定義を含みます
UserBoot.h	RAM領域での実行番地定義	可変ベクタ、割り込みハンドラ、RAM領域実行コード、データのRAM領域番地の定義

#### 4.4.2 内蔵フラッシュメモリ書き換えモジュールの説明

参考ユーザブートプログラムの内蔵フラッシュメモリ書き換えモジュール (flash\_drv) のモジュールインタフェース一覧を表 4.5に、データ型定義を表 4.6に示します。

表 4.5 モジュールインタフェース一覧

インタフェース名	機能	備考
BLOCK_A_IDX	内蔵フラッシュメモリのブロック番号 (データフラッシュのブロックAを表します)	FLASH_BLOCK_NUM型のデータ
BLOCK_B_IDX	内蔵フラッシュメモリのブロック番号 (データフラッシュのブロックBを表します)	FLASH_BLOCK_NUM型のデータ
BLOCK_0_IDX	内蔵フラッシュメモリのブロック番号 (プログラムROM1のブロック0を表します)	FLASH_BLOCK_NUM型のデータ
BLOCK_1_IDX	内蔵フラッシュメモリのブロック番号 (プログラムROM1のブロック1を表します)	FLASH_BLOCK_NUM型のデータ
BLOCK_2_IDX	内蔵フラッシュメモリのブロック番号 (プログラムROM1のブロック2を表します)	FLASH_BLOCK_NUM型のデータ
BLOCK_3_IDX	内蔵フラッシュメモリのブロック番号 (プログラムROM1のブロック3を表します)	FLASH_BLOCK_NUM型のデータ
FDS_OK	成功	FLASH_STATUS型のデータ
FDS_NOT_BLANK	ブランクではない	FLASH_STATUS型のデータ
FDS_BLOCK_LOCK	ブロックロックビットは“0”(ロック状態)	FLASH_STATUS型のデータ
FDS_BLOCK_UNLOCK	ブロックロックビットは“1”(非ロック状態)	FLASH_STATUS型のデータ
FDS_LBP_FAIL	ロックビットプログラムエラー	FLASH_STATUS型のデータ
FDS_PROG_FAIL	書き込みエラー (フラッシュメモリから書き込みエラーの通知あり)	FLASH_STATUS型のデータ
FDS_ADDR_ERROR	書き込みアドレスエラー (4バイトのアライメントではない)	FLASH_STATUS型のデータ
FDS_ERASE_FAIL	消去エラー (フラッシュメモリから消去エラーの通知あり)	FLASH_STATUS型のデータ
BLOCK_IDX_NUM	フラッシュメモリのブロック数	データフラッシュとプログラムROM1のブロック数合計
BLOCK_FIRST_IDX	フラッシュメモリのブロック定義開始インデックス	
LBP_DISABLE	ロックビット無効	
LBP_ENABLE	ロックビット有効	
InitFlashDrv	内蔵フラッシュメモリ書き換えモジュールの初期化	
ClearStatusRegister	内蔵フラッシュメモリのステータスレジスタのクリア要求	
BlankReadCheck	指定ブロックのブランク(フラッシュメモリ読み込み値=0xFFFF)チェック	
BlockBlankCheck	指定ブロックのブランク(消去後の状態)チェック要求	
BlockErase	指定ブロックの自動消去要求	
Program128bytes	128バイト連続自動書き込み要求	
LockBitProgram	指定ブロックのロックビット、ロック要求	
ReadLockBitStatus	指定ブロックのロックビット状態、取得要求	
FlashReadSumCalc	指定ブロックの1バイトチェックサム値とCRC-CCITT値の計算	

表 4.6 データ型定義

データ型名	データ型	定義ヘッダ
SBYTE	signed char	common.h
UBYTE	unsigned char	common.h
SWORD	signed short	common.h
UWORD	unsigned short	common.h
SDWORD	signed long	common.h
UDWORD	unsigned long	common.h
FLASH_BLOCK_NUM	enum	flash_drv.h
FLASH_STATUS	enum	flash_drv.h

以下に参考ユーザブートプログラムの内蔵フラッシュメモリ書き換えモジュール (flash\_drv) の関数について説明します。

• InitFlashDrv

概要	内蔵フラッシュメモリ書き換えモジュールの初期化		
	インタフェース名	内容	取得元
外部参照	PRCR	SFR(プロテクトレジスタ)	sfr64.h
	PM10	SFR(データフラッシュ有効ビット)	sfr64.h
	memcpy	標準関数(メモリ領域の複写)	string.h
	WDT_INITIALIZE	ウォッチドッグタイマ初期化	wdtRefresh.h
	RAM_DATA_ADDRESS	データ複写先RAM番地	UserBoot.h
	RAM_BASED_ROM_DATA_ADDRESS	データ複写元ROM番地	UserBoot.h
	SIZE_OF_RAM_BASED_ROM_DATA	データ複写バイト数	UserBoot.h
	RAM_PROG_ADDRESS	プログラムコード複写先RAM番地	UserBoot.h
	RAM_BASED_ROM_PROG_ADDRESS	プログラムコード複写元ROM番地	UserBoot.h
	SIZE_OF_RAM_BASED_ROM_PROG	プログラムコード複写バイト数	UserBoot.h
宣言	void InitFlashDrv (void);		
引数	データ型	意味	
	void	引数なし	
戻り値	データ型	意味	
	void	戻り値なし	
機能説明			
<ul style="list-style-type: none"> <li>データフラッシュ (0E000h~0FFFFh番地) を有効にします。</li> <li>内蔵フラッシュメモリを操作するプログラムコード部分、データをRAM領域に複写します。</li> <li>ウォッチドッグタイマリフレッシュ処理(ウォッチドッグタイマ初期化)を実行します。</li> </ul>			
備考			
内蔵フラッシュメモリ書き換えモジュール (flash_drv) の関数を使用する場合、本関数の実行(1回)は必須です。			

## • ClearStatusRegister

概要	内蔵フラッシュメモリのステータスレジスタのクリア要求		
外部参照	インタフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR06	SFR(プログラムステータスフラグ)	sfr64.h
	FMR07	SFR(イレーズステータスフラグ)	sfr64.h
宣言	void ClearStatusRegister (void);		
引数	データ型	意味	
	void	引数なし	
戻り値	データ型	意味	
	void	戻り値なし	
機能説明	<ul style="list-style-type: none"> <li>• CPU書き換えモード(EW0モード)に移行して内蔵フラッシュメモリにソフトウェアコマンド(クリアステータスレジスタ)を発行します。</li> <li>• ステータスレジスタのクリアを確認後、CPU書き換えモード(EW0モード)から復帰します。</li> </ul>		
備考	<p>内蔵フラッシュメモリがレディ (FMR00=1) かつプログラムステータス正常 (FMR06=0)かつイレーズステータス正常 (FMR07=0)の状態になるまで、本関数から復帰しません。</p> <p>CPU書き換えモード(EW0モード)中は 割り込み禁止(Iフラグのクリア)状態です。</p>		

## • BlankReadCheck

概要	指定ブロックのブランク(フラッシュメモリ読み込み値=0xFFFF)チェック		
外部参照	インタフェース名	内容	取得元
	なし		
宣言	FLASH_STATUS BlankReadCheck (FLASH_BLOCK_NUM BlockNum, UWORD *Data);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリのブランクチェック(フラッシュメモリ読み出し値=0xFFFF)対象ブロック番号	
	UWORD *	ブランクではないときの内蔵フラッシュメモリ読み出し値	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_OK : ブランク判定 FDS_NOT_BLANK : ブランクではない	
機能説明	<ul style="list-style-type: none"> <li>• 内蔵フラッシュメモリの指定ブロック領域すべてのデータが0xFFFFであれば、ブランク判定(FDS_OK)を返却します。</li> <li>• 指定ブロック領域内のデータが0xFFFFでない場合は、引数の*Dataに読み出したデータを格納し、ブランクではない(FDS_NOT_BLANK)を返却します。</li> </ul>		
備考	なし		

## • BlockBlankCheck

概要	指定ブロックのブランク (消去後の状態) チェック要求		
外部参照	インタフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR07	SFR(イレーズステータスフラグ)	sfr64.h
宣言	FLASH_STATUS BlockBlankCheck (FLASH_BLOCK_NUM BlockNum);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリのブランク (消去後の状態) チェック対象ブロック番号	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_OK : ブランク判定 FDS_NOT_BLANK : ブランクではない	
機能説明			
<ul style="list-style-type: none"> <li>• CPU書き換えモード(EW0モード)に移行して内蔵フラッシュメモリにソフトウェアコマンド(ブロックブランクチェック)を発行します。</li> <li>• イレーズステータス(FMR07)の値により戻り値を決定し、CPU書き換えモード(EW0モード)から復帰します。 <ul style="list-style-type: none"> <li>a) イレーズステータスが正常(FMR07=0)の場合は ブランク判定(FDS_OK)を返却します。</li> <li>b) イレーズステータスが異常(FMR07=1)の場合は ブランクではない(FDS_NOT_BLANK)を返却します。</li> </ul> </li> </ul>			
備考			
「ブランクではない」とき、クリアステータスレジスタを発行しています。この場合、内蔵フラッシュメモリのステータスがイレーズステータス正常(FMR07=0)の状態になるまで、本関数から復帰しません。 CPU書き換えモード(EW0モード)中は 割り込み禁止(Iフラグのクリア)状態です。			

## • BlockErase

概要	指定ブロックの自動消去要求		
外部参照	インタフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR07	SFR(イレーズステータスフラグ)	sfr64.h
	WDT_INITIALIZE	ウォッチドッグタイマ初期化	wdtRefresh.h
宣言	FLASH_STATUS BlockErase (FLASH_BLOCK_NUM BlockNum, UWORD LBPEnableFlag);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリの自動消去対象ブロック番号	
	UWORD LBPEnableFlag	LBP_ENABLE: ロックビット有効(FMR02=0) LBP_DISABLE: ロックビット無効(FMR02=1)	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_OK : 消去成功 FDS_ERASE_FAIL : 消去エラー	
機能説明			
LBPEnableFlagがLBP_ENABLE(ロックビット有効)の場合、消去対象のブロックが非ブランクもしくはロック状態であれば対象ブロックのイレーズを行います。 消去対象のブロックがブランクかつ非ロック状態であれば、イレーズを実行せずに消去成功(FDS_OK)を返します。 LBPEnableFlagがLBP_DISABLE(ロックビット無効)の場合、消去対象のブロックが非ブランクもしくはロック状態であればロックビットによるロックを無効にして対象ブロックのイレーズを行います。 消去対象のブロックがブランクかつ非ロック状態であれば、イレーズを実行せずに消去成功(FDS_OK)を返します。 イレーズエラーまたは、イリーガルコマンドエラーが発生した場合、戻り値として消去失敗(FDS_ERASE_FAIL)を返します。 イレーズが正常終了した場合、消去成功(FDS_OK)を返します。			
備考			
内蔵フラッシュメモリが自動消去中(FMR00=0)、ウォッチドッグタイマのリフレッシュ処理を実施しています。 CPU書き換えモード(EW0モード)中は 割り込み禁止(Iフラグのクリア)状態です。			



## • Program128bytes

概要	128バイト連続自動書き込み要求		
外部参照	インタフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR06	SFR(プログラムステータスフラグ)	sfr64.h
宣言	FLASH_STATUS Program128bytes (UDWORD Address, UWORD *Data);		
引数	データ型	意味	
	UDWORD	内蔵フラッシュメモリの書き込み開始アドレス	
	UWORD *	書き込みデータ(128バイト)格納バッファポインタ	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_OK : 書き込み成功 FDS_PROG_FAIL : 書き込みエラー FDS_ADDR_ERROR : 書き込みアドレスエラー	
機能説明			
<ul style="list-style-type: none"> <li>書き込み開始アドレス(UDWORD Address)が4バイトのアライメントでなければ、戻り値に書き込みアドレスエラー(FDS_ADDR_ERROR)を設定し、処理を終了します。</li> <li>書き込み開始アドレス(UDWORD Address)が4バイトのアライメントであれば、CPU書き換えモード(EW0モード)に移行して内蔵フラッシュメモリにソフトウェアコマンド(プログラム)を128バイトデータ分発行します。</li> <li>プログラムステータス(FMR06)の値により戻り値を決定し、CPU書き換えモード(EW0モード)から復帰します。 <ul style="list-style-type: none"> <li>a) プログラムステータスが正常(FMR06=0)の場合は書き込み成功(FDS_OK)を返却します。</li> <li>b) プログラムステータスが異常(FMR06=1)の場合は書き込みエラー(FDS_PROG_FAIL)を返却します。</li> </ul> </li> </ul>			
備考			
CPU書き換えモード(EW0モード)中は割り込み禁止(Iフラグのクリア)状態です。			

## • LockBitProgram

概要	指定ブロックのロックビット、ロック要求		
外部参照	インタフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR06	SFR(プログラムステータスフラグ)	sfr64.h
宣言	FLASH_STATUS LockBitProgram (FLASH_BLOCK_NUM BlockNum);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリのロックビットプログラム対象ブロック番号	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_OK : ロックビットプログラム成功 FDS_LBP_FAIL : ロックビットプログラムエラー	
機能説明			
<ul style="list-style-type: none"> <li>CPU書き換えモード(EW0モード)に移行して内蔵フラッシュメモリにソフトウェアコマンド(ロックビットプログラム)を発行します。</li> <li>プログラムステータス(FMR06)の値により戻り値を決定し、CPU書き換えモード(EW0モード)から復帰します。 <ul style="list-style-type: none"> <li>a) プログラムステータスが正常(FMR06=0)の場合はロックビットプログラム成功(FDS_OK)を返却します。</li> <li>b) プログラムステータスが異常(FMR06=1)の場合はロックビットプログラムエラー(FDS_LBP_FAIL)を返却します。</li> </ul> </li> </ul>			
備考			
CPU書き換えモード(EW0モード)中は割り込み禁止(Iフラグのクリア)状態です。			

## • ReadLockBitStatus

概要	指定ブロックのロックビット状態、取得要求		
外部参照	インターフェース名	内容	取得元
	FMR00	SFR(RY/BYステータスフラグ)	sfr64.h
	FMR16	SFR(ロックビットステータスフラグ)	sfr64.h
宣言	FLASH_STATUS ReadLockBitStatus (FLASH_BLOCK_NUM BlockNum);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリのロックビット状態取得対象ブロック番号	
戻り値	データ型	意味	
	FLASH_STATUS	FDS_BLOCK_LOCK : ブロックロックビットは“0”(ロック状態) FDS_BLOCK_UNLOCK : ブロックロックビットは“1”(非ロック状態)	
機能説明			
<ul style="list-style-type: none"> <li>• CPU書き換えモード(EW0モード)に移行して内蔵フラッシュメモリにソフトウェアコマンド(リードロックビットステータス)を発行します。</li> <li>• ロックビットステータス(FMR16)の値により戻り値を決定し、CPU書き換えモード(EW0モード)から復帰します。 <ul style="list-style-type: none"> <li>a) ロックビットステータスが正常(FMR16=0)の場合は ブロックロックビットは“0”(ロック状態)(FDS_BLOCK_LOCK)を返却します。</li> <li>b) ロックビットステータスが異常(FMR16=1)の場合は ブロックロックビットは“1”(非ロック状態)(FDS_BLOCK_UNLOCK)を返却します。</li> </ul> </li> </ul>			
備考			
CPU書き換えモード(EW0モード)中は 割り込み禁止(Iフラグのクリア)状態です。			

## • FlashReadSumCalc

概要	指定ブロックのチェックサム値(2バイト)とCRC-CCITT値(2バイト)の演算処理		
外部参照	インターフェース名	内容	取得元
	CRCD	SFR(CRCデータレジスタ)	sfr64.h
	CRCIN	SFR(CRCインプットレジスタ)	sfr64.h
宣言	void FlashReadSumCalc (FLASH_BLOCK_NUM BlockNum, UWORD *SumValue, UWORD *CrcValue);		
引数	データ型	意味	
	FLASH_BLOCK_NUM	内蔵フラッシュメモリのチェックサム値、CRC値の計算対象ブロック番号	
	UWORD *	チェックサム値バッファポインタ	
	UWORD *	CRC値バッファポインタ	
戻り値	データ型	意味	
	void	戻り値なし	
機能説明			
• 内蔵フラッシュメモリの指定ブロック領域すべてのデータを読み出してチェックサム値、CRC値を計算します。			
備考			
呼び出し元でチェックサム値(UWORD *SumValue)、CRC値(UWORD *CrcValue)の初期化を実施してください。			

## 5. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

## 6. 参考ドキュメント

M16C/63 ユーザーズマニュアル ハードウェア編 Rev.2.00  
M16C/64 ユーザーズマニュアル ハードウェア編 Rev.1.05  
M16C/64A ユーザーズマニュアル ハードウェア編 Rev.2.00  
M16C/64C ユーザーズマニュアル ハードウェア編 Rev.1.00  
M16C/65 ユーザーズマニュアル ハードウェア編 Rev.2.00  
M16C/65C ユーザーズマニュアル ハードウェア編 Rev.1.00  
M16C/6C ユーザーズマニュアル ハードウェア編 Rev.2.00  
M16C/5LD、M16C/56D ユーザーズマニュアル ハードウェア編 Rev.1.10  
M16C/5L、M16C/56 ユーザーズマニュアル ハードウェア編 Rev.1.00  
M16C/5M、M16C/57 ユーザーズマニュアル ハードウェア編 Rev.1.01  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース  
(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

Cコンパイラマニュアル  
M16Cシリーズ、R8Cファミリ用 Cコンパイラパッケージ V.5.45  
Cコンパイラユーザーズマニュアル Rev.2.00  
(最新版をルネサス エレクトロニクスホームページから入手してください。)

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ  
<http://japan.renesas.com/>

お問合せ先  
<http://japan.renesas.com/inquiry>

改訂記録	M16/63,64,64A,64C,65,65C,6C,5LD,56D,5L, 56,5M,57グループ ユーザブート機能の使用
------	--

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2009.03.31	—	初版発行
1.01	2011.02.28	—	デバイス追加 M16C/63,M16C/64C,M16C/65C,M16C/6C,M16C/5LD, M16C/56D,M16C/5L,M16C/56,M16C/5M,M16C/57
1.02	2011.07.29	1	要旨 「以下、M16C/64を例に挙げて説明します。」追記
		2	3.1 概要追加
		10-26	4. 全面改訂
		27	6. ユーザーズマニュアルRev.更新

すべての商標および登録商標は、それぞれの所有者に帰属します。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違っていると、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が異なる製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事事務の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。  
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）  
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/inquiry>