

序説

著者: Henrik M. Arnesen, Egil Rotevatn, Microchip Technology Inc.

MicrochipのAVR[®] EAシステムマイクロ コントローラはフラッシュ メモリ領域が、1つの書き込み中読み込み可(RWW:Read While Write)領域と1つの書き込み中読み込み不可(NRWW:Non Read While Write)領域に分けられるフラッシュ メモリが特徴で、RWW領域の消去または書き込み中にNRWW領域からの命令走行を続けることをCPUに許します。フラッシュ メモリの消去/書き込みの操作がms単位で測定されるため、RWW機能は終了を待つことなくこの時間を利用することを許します。この文書の目的はAVR EAデバイスでRWWフラッシュ メモリ領域への書き込みに対してNRWWフラッシュ メモリをどう使うかとこれが何故有益かを記述することです。

この技術概説はRWW/NRWWの概念と以下の使用事例でAVR EAシステムマイクロ コントローラの実装を説明します。

フラッシュ メモリ書き込み中の周辺機能割り込みのサービス

- この例はNRWWフラッシュ メモリからRWWへ書けることの利点を披露します。
- 消去/書き込み周期で重要な事象を示すために各種デバイス ピンが継続的に切り替えられます。
- ピンが周期的割り込み、ページ緩衝部状態、フラッシュ メモリ状態を表します。
- NRWWまたはRWWのどちらかの領域でプログラム ページを消去と書く時に事象は各種の出来事を持ちます。

注: この文書で記述した各使用事例に対して、MPLAB[®]でMicrochipコード 構成部(MCC)を使って作成した物とAVR[®] EAでMicrochip Studioを使って開発した素の物の2つのコード 例が準備されます。両方共にMicrochip Discoverで見つけることができます。

- [Microchip Studio用例コード](#)
- [MPLAB X用例コード](#)

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

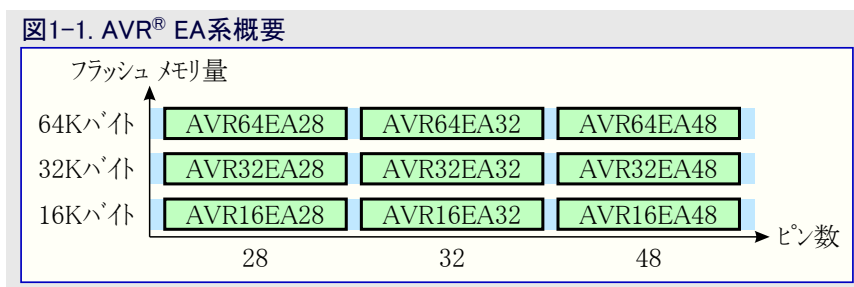
目次

序説	1
1. 関連デバイス	3
2. 概要	4
3. フラッシュメモリ書き込み中の周辺機能割り込みのサービス	6
4. MPLAB Discoverからのコード例取得	10
5. 改訂履歴	11
Microchip情報	12
Microchipウェブサイト	12
製品変更通知サービス	12
お客様支援	12
Microchipデバイスコード保護機能	12
法的通知	12
商標	13
品質管理システム	13
世界的な販売とサービス	14

1. 関連デバイス

本章はこの文書に関連するデバイスを一覧にします。下図はピン数の変種とメモリ量を展開して各種系統デバイスを示します。

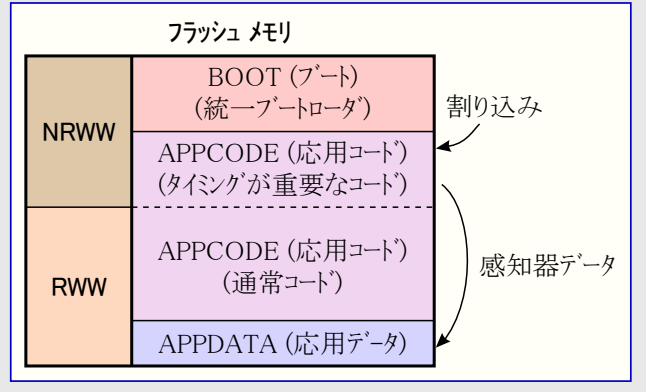
- これらのデバイスがピン互換で同じまたはより多くの機能を提供するため、垂直方向移植はコード変更なしで可能です。
 - 左への水平方向移植はピン数、従って利用可能な機能を減らします。
- 異なるフラッシュメモリ量を持つデバイスは一般的に異なるSRAMとEEPROMも持ちます。



2. 概要

フラッシュメモリはいくつかの語から成るページで構成されます。AVRに関して一般的に語は2バイトまたは16ビットです。AVR EAについてはページの大きさが128バイトです。ページの大きさは別に、フラッシュメモリは書き込み中読み込み不可(NRWW)と書き込み中読み込み可(RWW)の2つの物理的な領域に分けられます。NRWW領域はフラッシュメモリの始めて始まり、それにより、ヒューズ設定に応じてBOOT領域とAPPCODE/APPDATAの領域と重なります。コードが走行しているのと同じ領域への書き込みが不可能なことに注意するのが重要なことです。NRWWがBOOT/APPCODEまたはBOOT/APPDATAの領域間の境界と重なる例では、消去/書き込み周期がAPPCODE/APPDATAで進行中の間に走行するため、中断されないコードはBOOT領域に置かれることが必要です。BOOT/APPCODE/APPDATAを持つように構成設定する場合、CPUを停止することなくAPPDATAを書くために中断されないコードはAPPCODE領域のNRWW部にすることもできます。

図2-1. RWWメモリの流れ



構文の”RWW領域”は読まれる領域ではなく、プログラムされる(消去されるまたは書かれる)領域を指します。NRWWフラッシュメモリ内に置かれたコードだけがRWW領域をプログラミングする間にCPU命令実行またはデータ読み込みのどちらかによってアクセス可能です。

NRWWとRWWの領域間の主な違いは次のようになります。

1. NRWW領域で走行するコードを使ってRWW領域内に置かれたページを消去または書く時に、コードやデータはNRWW領域で走行または読むのを続けることができ、プログラムメモリ操作が走行中にCPUの継続操作を許します。
2. NRWW領域のページの消去または書き込み時、その操作中にCPUは停止されます。これに対する例外はBOOT領域がNRWW領域と重なるAPPCODEまたはAPPDATAの領域に書く時です(訳注:本行は矛盾するため誤りと思われれます)。

AVR EAのNRWW/RWWプログラムメモリ分割を活用する応用はプログラムメモリが書かれる間の妨害からCPUを守ります。

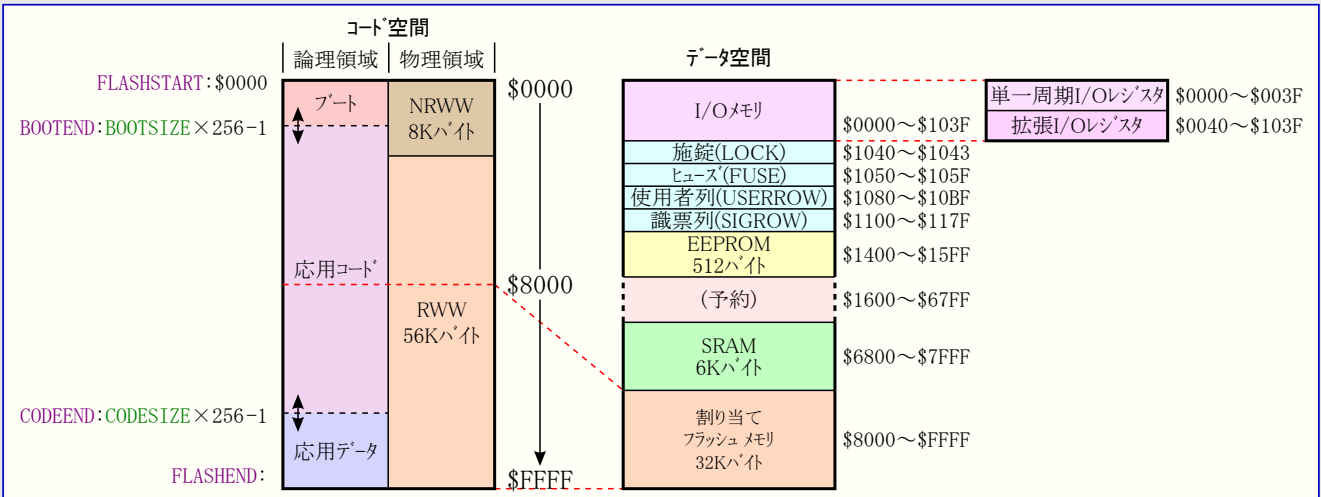
- 1つの例はプログラムメモリでAPPCODE/APPDATA領域をプログラミングする間で同時にホストから通信周辺機能(I²C/USART)経由で受け取った命令でCPUが働くブートローダの筋書です。
- 別の使用事例はプログラムメモリに一旦保存されなければならないデータが利用可能になった時にアナログ周辺機能がCPU操作に割り込むデータ記録の筋書です。

両筋書は不揮発性メモリ制御器(NVMCTRL)周辺機能がプログラムメモリの消去/書き込みをする間にCPUが妨害されない必要性を強調します。消去/書き込み周期が進行中に走行する必要があるコードがNRWW領域にあることを保証することにより、命令や割り込みがフラッシュメモリ消去/書き込み周期後ではなく、それらは到着する時に処理されることを保証することができます。

AVR EAで書き込み中の読み込み可(RWW)機能を実装する方法を理解するため、物理的、論理的、コード/データの空間の視点からプログラムメモリを見ることが役立ちます。物理的にプログラムメモリは固定のNRWWとRWWの領域で設計されています。これらは変更不可で部品番号に依存します。データシートの「メモリ構成」項をご覧ください。RWW使用例を使うため各々物理的な領域にコードとデータを格納してください。

論理的な領域はBOOT、APPCODE、APPDATAです。下の画像で示されるように、これらの領域はBOOTSIZEとCODESIZEのヒューズを通して調整することができます。注: ヒューズ構成設定に応じて、物理的なNRWW領域はBOOTとAPPCODE/APPDATAの両方に使うことができます。詳細説明についてはデータシートでNVMCTRL周辺機能の「メモリ構成」項をご覧ください。フラッシュメモリ割り当て機能を通してデータ空間経由でコード空間をアクセスでき、32Kバイトを超えるプログラムメモリを持つデバイスで利用可能なことに注意してください。詳細説明についてはデータシートでNVMCTRL周辺機能の「メモリアクセス」項をご覧ください。

図2-2. AVR64EA48フラッシュメモリ領域



この割り当てを実装するため、割り当てられたアドレスでリンク命名された属性がリンク任意選択として与えられ、命名された属性は関数とデータの両方を宣言する時に使われます。

RWW_DATA_SECTION属性はページ書き込みによって格納されるデータに対して使われます。これらのデータはBOOTまたはAPPCODEの領域のどちらかから書くことができます。下のコード断片はAPPDATA領域での配列の配置を示します。コンパイル前にリンク設定で、`rww_data`アドレスを定義してください。

```
// .rww_data部分はアドレス0x2000(語アドレス0x1000)です。
#define RWW_DATA_SECTION __attribute__((used, section(".rww_data")))

const RWW_DATA_SECTION uint8_t rww_array[DATA_SIZE] = {0};
```

フラッシュメモリのプログラミングまたは消去(例えば、割り込み、応用コードフラッシュメモリページに書き込みまたは消去、ブートローダ、ホスト通信ドライバ)と共に実行することができるルーチンに対してNRWW_PROG_SECTION属性を使ってください。下のコード断片ではフラッシュメモリ書き込み中に割り込み走行によって呼ばれることができるようにFillBufferと呼ばれる関数がNRWW領域に置かれます。再度、リンク設定は、`nrww_program`アドレスを定義しなければなりません。

```
// ,nrww_program部分はアドレス0x0400(語アドレス0x0200)です。
#define NRWW_PROG_SECTION __attribute__((section(".nrww_program")))

void NRWW_SECTION_CODE FillBuffer(void);
```

フラッシュメモリのプログラミングまたは消去中に実行しないルーチンに対してRWW_PROG_SECTION属性を使ってください。これらは阻止で物理的なRWW領域に配置されなければなりません。それらの例は準備ルーチンです。

下は属性部分の使用例です。

プロジェクトでブートローダを使う場合、ブートローダコードは先頭、フラッシュメモリ開始(0x0000)からBOOTEND(BOOTSIZ*256-1)までに置かれます。BOOTSIZヒューズ設定はBOOT領域の大きさを決めます。応用コード(APPCODE)は(CODESIZE>BOOTSIZである限り)BOOTの後に配置され、これは応用コードがNRWWとRWWの両方に属すかもしれないことを意味し、NRWW領域への関数移動を可能にします。NRWW領域に置かれた関数や割り込みはRWW領域消去/書き込み中に走行することができます。けれども、ソフトウェアが未知の状態になることに至る、RWWでの操作が進行中の間にRWW内に置かれたコードへの割り込みや分岐がないことを確実にする処置を取ってください。

表2-1. 書き込み中の読み込みの筋書

フラッシュメモリ領域消去/書き込み	フラッシュメモリ領域アクセス	CPU
NRWW領域	NRWW領域	停止
RWW領域	NRWW領域	走行
NRWW領域	RWW領域	停止
RWW領域	RWW領域	停止

AVR EAのデータシートでフラッシュメモリの大きさとNRWW/RWW領域を見つけてください。

3. フラッシュメモリ書き込み中の周辺機能割り込みのサービス

この例はNRWWフラッシュメモリ領域からRWWフラッシュメモリ領域に書くことの優位性を示します。割り込み事象と各種メモリ状態に対してデバイスピンを使うことによってコードの流れの概要とそれがオシロスコープまたはロジックアナライザでどう動くかを得ることが容易です。

試験

プログラムメモリはページと呼ばれるより小さな断片に分割されます。各ページは独立して消去と書き込みをされ、多数のバイトから成ります。フラッシュメモリはAVR EAに対してページの粒度を持ち、変更が必要な時は必ずページ内の全てのバイトに対して消去/書き込みを実行することを意味します。比べると、EEPROMはバイトの粒度を持ち、一度に単一バイトを消去と書き込みをすることができます。

コードは最初にデータが保存されるページを消去し、緩衝部を満たし、その後に緩衝部の値を使ってページ書き込みを実行します。周期的な計時器割り込みが継続的に緩衝部へ追加します。

環状緩衝部が新しいデータで満たされる(≧PROGMEM_PAGE_SIZE)と、環状緩衝部からページ緩衝部にPROGMEM_PAGE_SIZE数のバイトを複製してフラッシュメモリページ書き込みを開始します。再びこの処理を繰り返します。

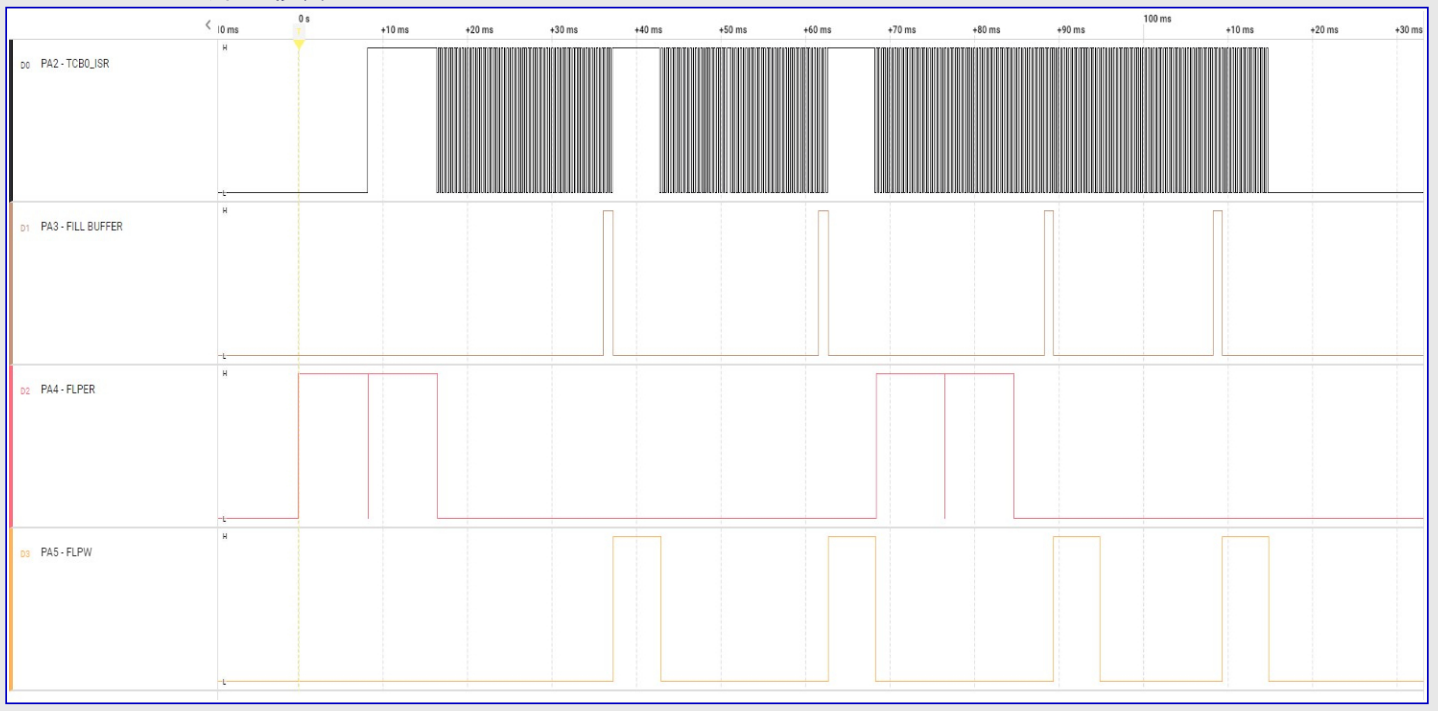
TCBタイマ/カウンタの捕獲/比較(CCMP)レジスタへの書き込みは割り込み処理ルーチン(ISR)周期を変更します。

コードはプログラムの流れを詳しく調べることを可能にする以下のようないくつかの状態ピンを切り替えます。

- PA2_ISRはTCB0の周期的割り込みISRが処理される時毎に交互切り替えます。
- PA3_Bufferはフラッシュメモリページ緩衝部が設定されている時にHighです。
- PA4_FLPWRはフラッシュメモリページ消去が進行中の際Highです。
- PA5_FLPWはフラッシュメモリページ書き込みが進行中の際Highです。

この応用が状態機構を通して走行するため、これらのピンは下の画像で見られるように切り替えます。この流れのより多くの詳細な記述は例”README”ファイルで見ることができます。

図3-1. ピン交互切り替え複製



コードの理解

リンクしたコード例は駆動部を生成するのにMCCを使うMPLAB X StudioとMicrochip Studioで作成した素のプロジェクトの種類でやって来ます。プロジェクトは同じコードの流れを持ち、以下の部分に分けられます。

- 主ファイルと関数
- 計時器0 - 緩衝部充填
- 計時器1 - 切り替え器跳ね返り抑制
- NVM制御器

上の部分に加えて、次に示されるようにBOOTとAPPDATAの領域の大きさを定義するためにヒューズを設定してください。


```
FUSES =
{
    .SYSCFG0 = FUSE_SYSCFG0_DEFAULT,
    .SYSCFG1 = FUSE_SYSCFG1_DEFAULT,

    .WDTCFG = FUSE_WDTCFG_DEFAULT,
    .BODCFG = FUSE_BODCFG_DEFAULT,
    .OSCCFG = FUSE_OSCCFG_DEFAULT,

    .BOOTSIZ = 31, // 0x0000~0x1F00 : BOOT領域
    .CODESIZ = 1,  // 0x01F00~FLASH_END : APPDATA領域
};
```

256バイトの塊を31に設定するBOOTSIZとCODESIZを除き、殆どのヒューズはそれらの既定値に設定されます。CODESIZは1に設定され、プログラムメモリはBOOTとAPPDATAの領域で論理的に分割されます。

主ファイルと構成

システムの主ファイルは初期化と例の主構造を含みます。注意する重要な事はRWW/NRWW領域がどう定義され、割り込みベクタ表がBOOT領域の先頭に移動されることです。最後に、割り当てられたプログラムメモリは最初の領域(.nrww_program、.nrww_data、.rww_dataの位置)を指し示します。これらの初期化はプログラムメモリのNRWW領域にだけ存在するコードを走行することと、それら既存のメモリがデータに対してRWWとNRWWの両領域でアドレス指定することを保証します。

計時器0

```
// 捕獲割り込みでのTCB
TCB0.INTCTRL = TCB_CAPT_bm;
TCB0.CCMP = TOP_VALUE;
TCB0.CTRLA = TCB_CLKSEL_DIV1_gc;
```

この計時器は活性時に捕獲を起動する周期的割り込みとして設定されます。初期化がこの計時器を許可しないことに注意してください。割り込みルーチンは呼ばれた時に下で見られるように、データ緩衝部を満たします。

```
// TCB0捕獲割り込み
FillBuffer();
// 割り込みフラグ解除(0)
TCB0.INTFLAGS = TCB_CAPT_bm;
```

計時器0

```
// 周期的割り込みでのTCB
TCB1.INTCTRL = TCB_CAPT_bm;
TCB1.CCMP = TCB1_TOP_VALUE;
TCB1.CTRLA = TCB_CLKSEL_DIV1_gc | TCB_ENABLE_bm;
```

2つ目の計時器は周期的割り込みで動くようにも構成されますが、全ての時間で許可されます。割り込みはその後に望まれない起動を避けるために跳ね返り抑制関数を走らせます。

```
// TCB1周期的割り込み
DebounceSW0();
// 割り込みフラグ解除(0)
TCB1.INTFLAGS = TCB_CAPT_bm;
```

プログラムメモリ消去

```
// TCB1周期的割り込み
DebounceSW0();
// 割り込みフラグ解除(0)
TCB1.INTFLAGS = TCB_CAPT_bm;

NRWW_DATA_last_addr = (((uint16_t) &nrww_array) & 0x7FFF) + MAPPED_PROGMEM_START + DATA_SIZE);

while ((uint16_t)nrwwFlashPointer < NRWW_DATA_last_addr)
{
    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_NOCMD_gc);

    // NVMCTRLのアドレスレジスタを更新するため、このアドレスに偽装書き込みを実行
    *nrwwFlashPointer = 0;
```

```

_PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_FLPW_gc);

while (NVMCTRL.STATUS & NVMCTRL_FLBUSY_bm)
{
    ; // フラッシュメモリ消去完了待機
}
SCOPE_PORT.OUTCLR = SCOPE_FLPW_bm;

nrwwFlashPointer = nrwwFlashPointer + PROGMEM_PAGE_SIZE;
}

```

NRWWとRWWのデータ領域に対するフラッシュメモリ消去操作は同じです。消去される空間に対してフラッシュメモリページ緩衝部を設定することによって始まります。**FLPER**ピンはHighに駆動され、故にフラッシュメモリ消去操作が正に開始した時を視覚器で見ることが可能です。指令衝突異常を避けるため、直前のNVM指令を解消するのにNVM制御器に**NOCMD**指令を送り、偽装の書き込み/格納が後続します。ページ消去指令がNVM制御器に送られます。フラッシュメモリが多忙でなく、次の消去のアドレスがページ量によって増されたことをNVMCTRL.**STATUS**レジスタが示す時点で**FLPER**ピンがLowに駆動されます。フラッシュメモリ消去操作はデータ領域が消去された時点で完了します。

プログラムメモリ書き込み

```

// ページが完全に満たされたか検査
if ((writeIndex - readIndex) >= PROGMEM_PAGE_SIZE)
{
    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_NOCMD_gc);

    for (uint8_t i = 0; i < PROGMEM_PAGE_SIZE; i++)
    {
        *nrwwFlashPointer++ = buffer[readIndex++];
    }
    SCOPE_PORT.OUTCLR = SCOPE_BUFFER_bm;
    SCOPE_PORT.OUTSET = SCOPE_FLPW_bm;

    // フラッシュメモリページ書き込み
    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_FLPW_gc);
    while (NVMCTRL.STATUS & NVMCTRL_FLBUSY_bm)
    {
        ; // フラッシュメモリページ書き込み操作完了待機
    }
}
}

```

NRWWとRWWのデータ領域に対するフラッシュメモリプログラミング(書き込み)操作は同じです。フラッシュメモリページ全体に書かれるために十分なデータが利用可能かを調べることによって始まります。そうでないなら、十分なデータが生成されるまで待ちます。そうなら、指令衝突異常を避けるため、直前のNVM指令を解消するのにNVM制御器に**NOCMD**指令を送ります。その後、**BUFFER**ピンがHighに駆動され、故に緩衝部設定操作が正に始まる時を見るために視覚器を使うことが可能です。データがデータ緩衝部から読まれ、フラッシュメモリページ緩衝部に格納されます。完全なページが読まれて格納されたのと同等の時に**BUFFER**ピンがLowに駆動されます。**FLPW**ピンがHighに駆動され、ページ書き込み指令がNVM制御器に送られます。フラッシュメモリが多忙でなく、次のページ書き込みのアドレスがページ量によって増されたことをNVMCTRL.**STATUS**レジスタが示す時点で**FLPW**ピンがLowに駆動されます。フラッシュメモリ書き込み操作はデータ領域が書かれた時点で完了します。

上の消去と書き込みの関数に対して、このコード断片は素の例からで、MCC例については違うように構成されます。けれども、機能は各例について同じです。

緩衝部を満たす

```

void FillBuffer(void)
{
    // いくつかのデータで緩衝部を見たす。
    SCOPE_PORT.OUTTGL = SCOPE_ISR_bm;
    buffer[writeIndex++] = (data >> 2);
    data++;

    // 緩衝部溢れか調査
    if (writeIndex == readIndex)
    {
        // 溢れ
    }
}

```



```
SCOPE_PORT. OUTTGL = SCOPE_OVERFLOW_bm;  
}  
}
```

FillBuffer関数は計時比較計数器が設定したTOP値に達した時に計時器0での周期的割り込みによって起動されます。ISRピンが切り替わり、緩衝部がデータで満たされます。OVERFLOWは緩衝部が溢れた場合に切り替わります。

SW0動作と跳ね返り抑制

NRWWとRWWの両データ領域に対する消去と書き込みの操作はSW0押下によって開始されます。その後のSW0押下は同じ処理を起動します。新しいデータの種としてRWWデータ領域の最後のデータ位置の値が使われます。この動きのため、SW0が押されてプログラムメモリが読まれる時にNRWWとRWWのデータ領域での変化を観察することができます。SW0が押されたことを確認するため、LED0が点灯します。計測ピンの交互切り替えを捕獲することができる外部計測(オシロスコープやロジックアナライザ)を設定するのに必要なため、釦押下によって例の全体操作を起動することが必要です。簡単な跳ね返り抑制算法が選ばれ、物理的な切替器のSW0は消去と書き込みの操作の望まれない起動を防ぎます。フラッシュメモリプログラミングはそれを超える操作が保証されない制限された消去/書き込み耐久性を持つことに注意してください(更なる情報と詳細についてはデバイスの電気的特性を調べてください)。

割り込みベクタ表の移動

割り込みベクタ配置は制御A(CPUINT.CTRLA)レジスタの割り込みベクタ選択(IVSEL)ビット値に依存します。割り込み元を使わない場合、通常のプログラムコードをそれらの位置に配置することができます。IVSELへの1書きこみにより、割り込みベクタはフラッシュメモリのBOOT領域の先頭に配置され、RWW領域の消去/書き込み操作が走行している間に割り込みが通常のように走行することを許します。「概要」章で言及したように、構文の”RWW領域”は読まれる領域ではなく、プログラムされる(消去されるまたは書かれる)領域を指します。RWW領域がプログラミングされている間にCPU命令実行またはデータ読み込みのどちらかによって、RWWフラッシュメモリ内に置かれたコードだけがアクセス可能です。

4. MPLAB Discoverからのコード例取得

MPLAB Discoverは複数の供給元から利用可能な資源を1つの入口に束ねます。

MPLAB Discoverウェブ頁: [MPLAB Discover](#)



コード例

- [Microchip Studio](#) 用例コード
- [MPLAB X](#) 用例コード

MPLAB Discoverから直接.zipファイルとしてコード例をダウンロードすることができます。”studio”で終わる保存場所はMicroship Studioで開くことができます。”mplab-mcc”で終わる保存場所はMPLAB Xで開くことができます。

コード例が[GitHub](#)で提供される時にMPLAB Discoverは”Open with GitHub(GitHubで開く)”リンクを提供します。そこで、例コードをダウンロードするか、または手元の保存場所複製を作成するためにPCでgitツールを使うことができます。

5. 改訂履歴

文書改訂	日付	注釈
A	2023年10月	初版文書公開

Microchip情報

Microchipウェブ サイト

Microchipはwww.microchip.com/で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **一般的な技術支援** – 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- **Microchipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するにはwww.microchip.com/pcnへ行って登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 組み込み解決技術者(ESE:Embedded Solutions Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はwww.microchip.com/supportでのウェブ サイトを通して利用できます。

Microchipデバイス コード保護機能

Microchip製品での以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは動作仕様内で意図した方法と通常条件下で使われる時に、その製品系統が安全であると考えます。
- Microchipはその知的所有権を尊重し、積極的に保護します。Microchip製品のコード保護機能を侵害する試みは固く禁じられ、デジタル ミレニアム著作権法に違反するかもしれません。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が”破ることができない”ことを当社が保証すると言うことを意味しません。コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。

法的通知

この刊行物と契約での情報は設計、試験、応用とのMicrochip製品の統合を含め、Microchip製品でだけ使えます。他の何れの方法でのこの情報の使用はこれらの条件に違反します。デバイス応用などに関する情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。追加支援については最寄りのMicrochip営業所にお問い合わせ頂くか、www.microchip.com/en-us/support/design-help/client-support-servicesで追加支援を得てください。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責することに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Microchip、Adaptec、AVR、AVR、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maxStylus、maxTouch、MediaLB、megaAVR、Microsemi、Microsemi、MOST、MOST、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST、Super Flash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

AgileSwitch、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus、Quiet-Wire、SmartFusion、SyncWorld、TimeCesium、TimeHub、TimePictra、TimeProvider、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、Clockstudio、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、Espresso T1S、EtherGREEN、EyeOpen、GridTime、IdealBridge、IGaT、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、IntelliMOS、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、MarginLink、maxCrypto、maxView、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified、MPLIB、MPLINK、mSiC、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、Power MOS IV、Power MOS 7、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、SmartHLS、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、Trusted Time、TSHARC、Turing、USBCheck、VariSense、Vector Blox、VeriPHY、ViewSpan、WiperLock、XpressConnect、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adaptec、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2023年、Microchip Technology Incorporatedとその子会社、不許複製

品質管理システム

Microchipの品質管理システムに関する情報についてはwww.microchip.com/qualityを訪ねてください。

日本語© HERO 2024.

本技術概説はMicrochipのTB3344技術概説(DS90003344A-2023年10月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: www.microchip.com/support ウェブアドレス: www.microchip.com	オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 中国 - 東莞 Tel: 86-769-8702-9880 中国 - 広州 Tel: 86-20-8755-8029 中国 - 杭州 Tel: 86-571-8792-8115 中国 - 香港特别行政区 Tel: 852-2943-5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 - 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040	インド - ハンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - プネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100	オーストラリア - ウェルズ Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4485-5910 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガルピング Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイムブロン Tel: 49-7131-72400 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア - ハートバ Tel: 39-049-7625286 オランダ - テルネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ポーランド - ワルシャワ Tel: 48-22-3325737 ルーマニア - ブカレスト Tel: 40-21-407-87-50 スペイン - マドリード Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン - イェテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム Tel: 46-8-5090-4654 イギリス - ウォーキングム Tel: 44-118-921-5800 Fax: 44-118-921-5820
アランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ホストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 デトロイト Novi, MI Tel: 248-848-4000 ヒューストン TX Tel: 281-894-5983 インディアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンゼルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 ローリー NC Tel: 919-844-7510 ニューヨーク NY Tel: 631-435-6000 サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078			