

フラッシュ書き込み中の 周辺モジュール割り込みサービス TB3344



はじめに

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

Microchip AVR[®] EA マイクロコントローラ ファミリが備えるフラッシュメモリは RWW (Read-While-Write) セクションと NRWW (Non Read-While-Write) セクションに分割されており、RWW セクションに対する消去または書き込みの実行中に CPU は NRWW セクションからの命令を実行し続ける事ができます。フラッシュに対する消去/書き込み動作はミリ秒オーダーで処理されますが、RWW 機能を使うと消去/書き込み動作の完了を待つ事なく、この時間を他の処理用に使う事ができます。本書では、AVR EA デバイス上で NRWW フラッシュ セクションを使って RWW フラッシュ セクションへ書き込む方法と、その利点について説明します。

RWW/NRWW の概念とその AVR EA ファミリデバイス上での実装について説明するために、下記のユースケースを使います。

ユースケース：フラッシュへの書き込み中の周辺モジュール割り込みサービス

- このユースケースは、NRWW フラッシュメモリから RWW へ書き込む事で得られる利点を明らかに示します。
- 各種デバイスピンの状態遷移により、消去/書き込みサイクル中の重要イベントが示されます。
- これらのピンは周期的割り込み、ページバッファ ステータス、フラッシュ ステータスを示します。
- イベントの発生の仕方は、NRWW または RWW のどちらの領域内のプログラム ページを消去/書き込みするかに応じて異なります。

Note: このユースケース向けに 2 種類のサンプルコードが提供されます。1 つは MPLAB[®]内で Microchip Code Configurator (MCC)を使って作成されたコードであり、もう 1 つは AVR[®] EA 上で Microchip Studio を使って開発されたベアメタルコードです。どちらのコードも Microchip Discover で見つかります。

- [Microchip Studio 向けサンプルコード](#)
- [MPLAB X 向けサンプルコード](#)

目次

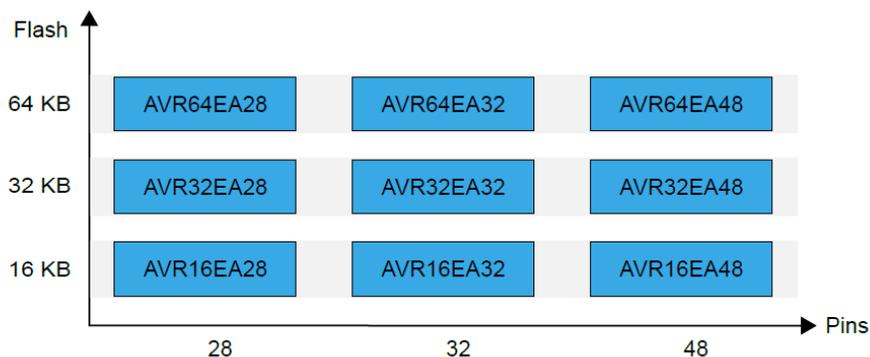
はじめに	1
1. 対応デバイス	3
2. 概要	4
3. ユースケース: フラッシュへの書き込み中の周辺モジュール割り込みサービス	7
4. MPLAB Discover からのサンプルコードの入手	11
5. 改訂履歴	12
Microchip 社の情報	13
Microchip 社ウェブサイト	13
製品変更通知サービス	13
カスタマサポート	13
Microchip 社のデバイスコード保護機能	13
法律上の注意点	13
商標	14
品質管理システム	15
各国の営業所とサービス	16

1. 対応デバイス

以下に、本書の内容に対応するデバイスを記載します。図 1-1 に、ピン数とメモリ容量に基づく AVR EA ファミリデバイスのラインアップを示します。

- 縦方向の上位デバイス (ピン数が同じでメモリ容量が大きなデバイス)へ移行する場合、機能が増えてもピン互換であるためコードを変更する必要はありません。
- 横方向の下位デバイス (メモリ容量が同じで少ピンのデバイス)へ移行する場合、使えなくなる機能があります。
- フラッシュメモリ容量が異なるデバイスでは、通常 SRAM と EEPROM の容量も異なります。

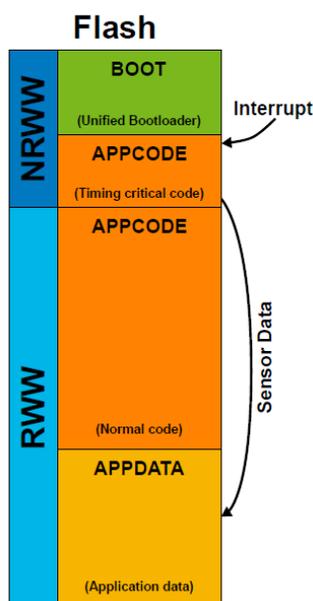
図 1-1. AVR® EA ファミリの概要



2. 概要

フラッシュメモリは複数のページで構成され、各ページは複数のワードを格納します。AVR デバイスでは、一般的に1ワードは2バイト(16ビット)です。AVR EA デバイスのページサイズは128バイトです。ページサイズとは別に、フラッシュは2つの物理セクション - NRWW (Non Read-While-Write)セクションと RWW (Read-While-Write)セクション - に分割されます。NRWW セクションはフラッシュの先頭から始まり、ヒューズ設定に応じて BOOT セクションおよび APPCODE/APPDATA セクションとオーバーラップします。あるセクションからコードを実行している時に同じセクションに書き込む事はできないという事を理解する事が重要です。例えば NRWW セクションが BOOT/APPCODE セクション間の境界または BOOT/APPDATA セクション間の境界とオーバーラップする場合、APPCODE または APPDATA セクション内の消去/書き込みサイクル中に CPU を停止せずに実行を継続するコードは BOOT セクション内に配置する必要があります。フラッシュ内に BOOT、APPCODE、APPDATA セクションを配置する場合、APPDATA セクションのプログラミング中に CPU を停止せず実行を継続するコードは、APPCODE セクションの NRWW 領域内に格納する事もできます。

図 2-1. RWW メモリフロー



「RWW セクション」とは、プログラミング(消去または書き込み)されるセクションを指します(読み出されるセクションを指すものではありません)。RWW セクションのプログラミング中は、CPU 命令の実行またはデータの読み出しによって NRWW セクション内のコードにのみアクセスできます。

NRWW セクションと RWW セクションの主な違いは以下の通りです。

1. NRWW セクションから実行されるコードを使って RWW セクション内に配置されたページに対する消去または書き込みを実行する場合、NRWW セクションからのコード実行またはデータ読み出しを継続できます。これにより、プログラムメモリに対する消去/書き込みを実行しながら CPU の動作を継続できます。
2. NRWW セクション内のページに対する消去または書き込み時には CPU の動作が停止します。例外として、BOOT セクションから NRWW セクションとオーバーラップする APPCODE または APPDATA セクションへの書き込み時には CPU は停止しません。

AVR EA デバイスの NRWW / RWW プログラムメモリ分割を利用するアプリケーションは、プログラムメモリへの書き込み中に CPU を停止する事なく実行を継続できるように構成できます。

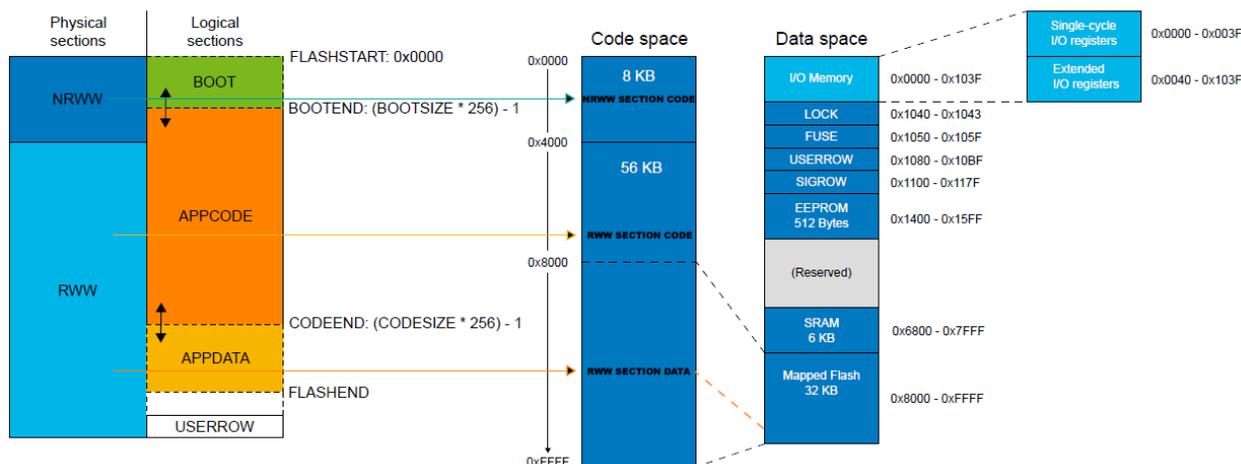
- そのようなユースケースの1つがブートローダ シナリオです。この例の場合、CPU はホストから通信周辺モジュール(I²C/USART)を介して受信したコマンドに従って動作しながら、同時にプログラムメモリ内の APPCODE/APPDATA セクションをプログラミングします。
- もう1つのユースケースはデータロギング シナリオです。この例の場合、プログラムメモリに保存する必要があるデータが入手可能になった時点でアナログ周辺モジュールが CPU 動作に割り込みます。

どちらのシナリオでも、不揮発性メモリ コントローラ(NVMCTRL)モジュールがプログラムメモリに対して消去/書き込みを実行中である時に CPU の制御権がブロックされない必要があります。消去/書き込みサイクルが進行中であっても実行する必要があるコードを NRWW セクション内に格納しておく事により、コマンドまたは割り込み要求が届いた時にフラッシュ消去/書き込みサイクルの完了を待つ事なく直ちに処理できます。

AVR EA 上の RWW (Read-While-Write)機能の実装を理解するには物理的観点、論理的観点、コード/データ空間的観点でプログラムメモリを見る事が助けとなります。物理的には、プログラムメモリは固定された NRWW セクションと RWW セクションにより構成されます。これらのセクションはデバイスのタイプ(製品番号)ごとに決まっており、変更はできません(データシート内の「メモリの概要」参照)。RWW ユースケースを使うには、コードとデータをそれぞれの物理セクションに保存します。

論理的には、プログラムメモリは BOOT、APPCODE、APPDATA セクションに分割されます。図 2-2 に示す通り、これらのセクションは BOOTSIZE および CODESIZE ヒューズを使って調節できます。ヒューズ設定に応じて、物理的 NRWW セクションは BOOT セクションと APPCODE/APPDATA セクションの両方に使う事ができます。詳細な説明は、データシート内の「メモリ構成」を参照してください。プログラムメモリのサイズが 32KB を超えるデバイスでは、フラッシュマッピング機能によりデータ空間を経由してコード空間にアクセスできます。詳細な説明はデータシート内の「メモリアクセス」を参照してください。

図 2-2. AVR64EA48 のフラッシュ セクション



このメモリ割り当てを実装するため、アドレスが割り当てられたリンク名前付き属性がリンクオプションとして与えられ、それらの名前付き属性は関数とデータの両方の宣言時に使われます。

RWW_DATA_SECTION 属性は、ページ書き込みによって保存するデータ向けに使います。これらのデータは BOOT セクションまたは APPCODE セクションから書き込む事ができます。下のサンプルコードは、APPDATA セクション内に配列を配置します。コンパイルする前に、リンク設定内で .rww_data アドレスを定義します。

```
// the .rww_data section is at address 0x2000 (word address 0x1000)
#define RWW_DATA_SECTION attribute ((used, section(".rww_data")))

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

フラッシュのプログラミング/消去中に**実行する可能性がある**ルーチン(例: 割り込み、app code フラッシュ ページ書き込み/消去、ブートローダ ホスト通信ドライバ等)には NRWW_PROG_SECTION 属性を使います。

下のサンプルコードは、フラッシュへの書き込み中に実行される割り込みから呼び出し可能な関数 `FillBuffer` を `NRWW` セクション内に配置します。コンパイルする前に、リンク設定内で `.nrww_program` アドレスを定義する必要があります。

```
// the .nrww_program section is at address 0x0400 (word address 0x0200)
#define NRWW_PROG_SECTION attribute ((section(".nrww_program")))

void NRWW_SECTION_CODE FillBuffer(void);
```

フラッシュのプログラミング/消去中に**実行する必要のない**ルーチンには `RWW_PROG_SECTION` 属性を使います。これらはブロッキングの対象となり、物理 `RWW` セクション内に配置する必要があります。セットアップルーチンは、そのようなルーチンの一例です。

以下に、セクション属性の使用例を示します。

ブートローダをプロジェクト内で使う場合、ブートローダ コードが最初にフラッシュの先頭アドレス (0x0000)~`BOOTEND` (`BOOTSIZEx256-1`)に配置されます。`BOOT` セクションのサイズは `BOOTSIZExヒューズ` の設定によって決まります。`CODESIZE > BOOTSIZExヒューズ` である場合、`APPCODE` セクションが `BOOT` セクションの後に配置されます。`APPCODE` セクションが `NRWW` セクションと `RWW` セクションの両方にまたがって配置される場合、`NRWW` セクション内に配置された関数または割り込みは、`RWW` セクションに対する消去/書き込み中に実行可能です。しかし、`RWW` に対する消去/書き込みの実行中に `RWW` セクション内に配置されたコードへの割り込みまたはジャンプが発生してソフトウェアの実行が未確定の状態を終了しないよう措置を講じる必要があります。

表 2-1. `RWW` (Read-While-Write)シナリオ

消去/書き込みされるフラッシュ セクション	アクセスされるフラッシュ セクション	CPU
<code>NRWW</code> セクション	<code>NRWW</code> セクション	停止
<code>RWW</code> セクション	<code>NRWW</code> セクション	動作
<code>NRWW</code> セクション	<code>RWW</code> セクション	停止
<code>RWW</code> セクション	<code>RWW</code> セクション	停止

フラッシュメモリのサイズと `NRWW/RWW` セクションのサイズについては、`AVR EA` デバイスのデータシートを参照してください。

3. ユースケース: フラッシュへの書き込み中の周辺モジュール割り込みサービス

このユースケースは、NRWW フラッシュ セクションから RWW フラッシュ セクションへ書き込む事の利点を示します。割り込みイベントおよび各種メモリステータスを示すデバイスピンの動作をオシロスコープまたはロジックアナライザで観察する事により、コードフローとその動作方法を容易に理解できます。

試験

プログラムメモリは複数のページに分割されています。各ページは複数バイトを格納し、ページごとに書き込み/消去が可能です。AVR EA デバイスのフラッシュはページ単位で処理されるため、内容の変更が必要な場合は常に1ページ内の全てのバイトに対して消去/書き込みを実行します。これに対してEEPROMメモリは1バイト単位で消去/書き込みが行えます。

サンプルコードは最初にデータの保存先ページを消去し、バッファにデータを書き込み、最後にバッファ値を使ってページ書き込みを実行します。周期的なタイマ割り込みによりバッファには次第にデータが蓄積されるように構成しています。

リングバッファに `PROGMEM_PAGE_SIZE` 以上の新しいデータが貯まると `PROGMEM_PAGE_SIZE` 個のバイトがリングバッファからフラッシュページ バッファへ複製され、フラッシュページ書き込みが開始されます。このプロセスが2回繰り返されます。

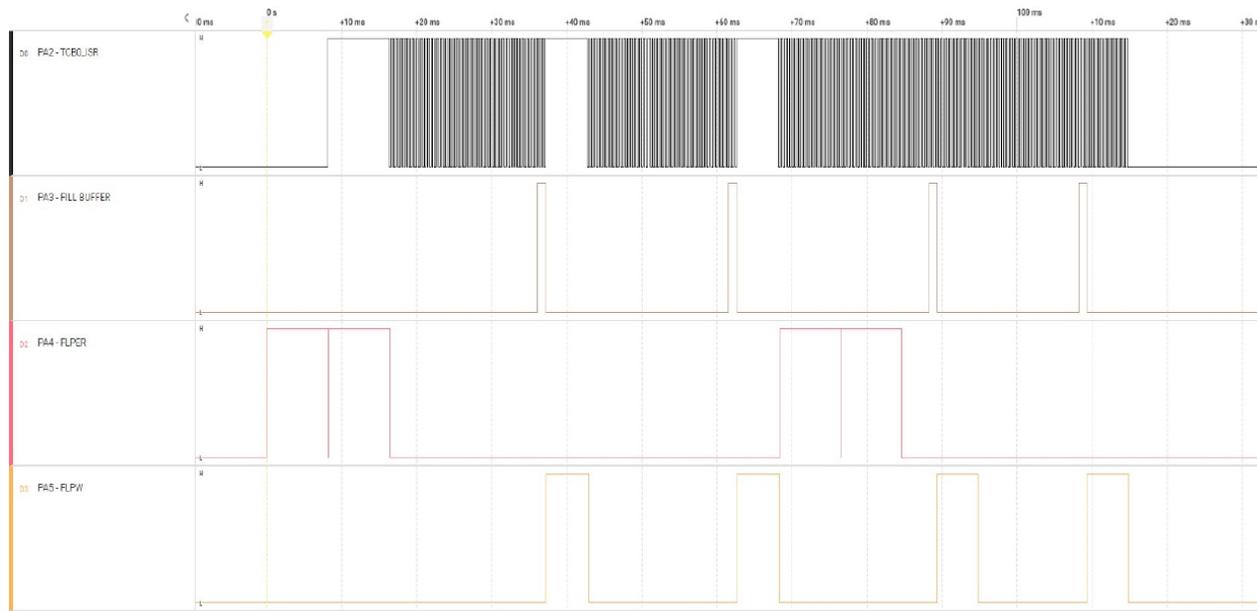
サンプルコードは、TCB タイマ/カウンタ内のキャプチャ/コンペア(CCMP)レジスタへの書き込みによりISR周期を変更します。

プログラムフローの観察を可能にするため、コードは以下のステータスピンをトグルします。

- `PA2_ISR` はTCB0 周期的割り込みISRが処理されるたびにトグルします。
- `PA3_Buffer` はフラッシュ ページバッファの書き込み中に High になります。
- `PA4_FLPFR` はフラッシュページの消去実行中に High になります。
- `PA5_FLPW` はフラッシュページの書き込み実行中に High になります。

アプリケーションはステートマシンに従って実行されるため、これらのピンは図 3-1 の通りに動作します。このフローの詳細は、サンプルコードの「README」ファイルに記載されています。

図 3-1. ステータスピンの動作



コードを理解する

2 種類のリンク済みサンプルコードが提供されます。1 つは MCC を使ってドライバを生成する MPLAB X Studio 向けであり、もう 1 つはベアメタル プロジェクトを作成する Microchip Studio 向けです。どちらのプロジェクトも同じコードフローを持ち、以下の部分に分割されています。

- main ファイルと関数
- Timer 0 - バッファ書き込み
- Timer 1 - スイッチ デバウンス
- NVM コントローラ

上記に加えて、BOOT および APPDATA セクションのサイズを定義するために以下の通りにヒューズを設定します。

```
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 Section
    .CODESIZ = 1,      // 0x01F00 -> FLASH_END : APPDATA Section
};
```

BOOTSIZ は 31 ブロック(1 ブロックは 256 バイト)に設定され、CODESIZ は 1 に設定されます。その他のヒューズは全て既定値に設定されます。CODESIZ が 1 に設定されるため、プログラムメモリは論理的に BOOT セクションと APPDATA セクションに分割されます。

main ファイルとセットアップ

システムの main ファイルはサンプルコードの初期化部および main の構造部を格納します。RWW/NRWW セクションの定義と、割り込みベクタテーブルが BOOT セクションの先頭へ移動される事に注意する事が重要です。最終的に、プログラムメモリは各セクションの先頭アドレス (.nrww_program、.nrww_data、.rww_dat)によってマッピングされます。これらの初期化により、実行コードはプログラムメモリの NRWW セクション内にのみ配置され、データは RWW および NRWW セクションの両方にまたがって配置されます。

Timer 0

```
// TCB with capture interrupt
TCB0.INTCTRL = TCB_CAPT_bm;
TCB0.CCMP = TOP_VALUE;
TCB0.CTRLA = TCB_CLKSEL_DIV1_gc;
```

このタイマは、アクティブ時のキャプチャ発生時にトリガされる周期的割り込み用に設定されます。このタイマは初期化によって有効にされない事に注意してください。下記の割り込みルーチンは、呼び出された時にデータバッファに書き込みます。

```
// TCB0 capture interrupt
FillBuffer();
// Clear intflag
TCB0.INTFLAGS = TCB_CAPT_bm;
```

Timer 1

```
// TCB with periodic interrupt
TCB1.INTCTRL = TCB_CAPT_bm;
TCB1.CCMP = TCB1_TOP_VALUE;
TCB1.CTRLA = TCB_CLKSEL_DIV1_gc | TCB_ENABLE_bm;
```

もう 1 つのタイマも周期的割り込みの実行用に設定されますが、こちらのタイマは常時有効です。この割り込みは、誤ったトリガを防ぐためのデバウンス機能を実行します。

```
// TCB1 capture interrupt
DebounceSW0();
```

```
// Clear intflag
TCB1.INTFLAGS = TCB_CAPT_bm;
```

プログラムメモリの消去

```
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);

    // Perform a dummy write to this address to update the address register in NVMCTL
    *nrwwFlashPointer = 0;

    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_FLPER_gc);

    while (NVMCTRL.STATUS & NVMCTRL_FLBUSY_bm)
    {
        ; // wait for Flash erase to complete
    }
    SCOPE_PORT.OUTCLR = SCOPE_FLPER_bm;

    nrwwFlashPointer = nrwwFlashPointer + PROGMEM_PAGE_SIZE;
}
}
```

フラッシュの消去動作は NRWW データ セクションでも RWW データ セクションでも同じです。消去動作は、消去する領域に対してフラッシュ ページバッファを設定する事により始まります。FLPER ピンが High に駆動される事により、フラッシュ消去動作が始まるタイミングをスコープで確認できます。コマンド コリジョンエラーを防ぐために NOCMD 命令が NVM コントローラへ送信されて直前の NVM コマンドがクリアされた後に、ダミーの書き込み/保存が実行されます。次いでページ消去コマンドが NVM コントローラへ送信されます。NVMCTRL.STATUS レジスタによってフラッシュが非ビジーである事が示されると FLPER ピンが Low に駆動され、次の消去動作のアドレスがページサイズ分インクリメントします。フラッシュ消去動作は、データセクションが書き込まれると完了します。

プログラムメモリの書き込み

```
// Check if we have a full page to fill
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;

    // Write the Flash page
    _PROTECTED_WRITE_SPM(NVMCTRL.CTRLA, NVMCTRL_CMD_FLPW_gc);
    while (NVMCTRL.STATUS & NVMCTRL_FLBUSY_bm)
    {
        ; // wait Flash write page operation to complete
    }
}
}
```

フラッシュのプログラミング動作は NRWW データセクションでも RWW データセクションでも同じです。この動作は、フラッシュページを完全に埋めるのに十分なデータが存在するかどうかを確認する事により始まります。存在しない場合、十分なデータが生成されるまで待機します。十分なデータが存在する場合、コマンド コリジョンエラーを防ぐために NOCMD 命令が NVM コントローラへ送信されて直前の NVM コマンドがクリアされます。その後に BUFFER ピンが High に駆動される事により、バッファ書き込み動作が始まるタイミングをスコープで確認できます。データがデータバッファから読み出されてフラッシュ ページバッファに保存されます。完全な 1 ページ分のデータが読み出されて保存されると、BUFFER ピンは Low に駆動されます。次いで FLPW ピンが High に駆動され、ページ書き込みコマンドが NVM コントローラへ送信されます。NVMCTRL.STATUS レジスタによってフラッシュが非ビジーである事が示されると FLPW ピンは Low に駆動され、次のページ書き込みのアドレスがページサイズ分インクリメントします。フラッシュ書き込み動作は、データセクションが書き込まれると完了します。

前述の消去および書き込み動作の説明に使ったサンプルコードはペアメタル用であり、MCC 用のコードとは書式が異なりますが、どちらのサンプルコードでも機能は同じです。

バッファの書き込み

```
void FillBuffer(void)
{
    // Fill the buffer with some data
    SCOPE_PORT.OUTTGL = SCOPE_ISR_bm;
    buffer[writeIndex++] = (data >> 2);
    data++;

    // Check if we have a buffer overflow
    if (writeIndex == readIndex)
    {
        // Overflow
        SCOPE_PORT.OUTTGL = SCOPE_OVERFLOW_bm;
    }
}
```

FillBuffer 関数は、timer 0 のタイマコンペア カウンタが TOP 値に達した時点で timer 0 上の周期的割り込みによってトリガされます。ISR ピンがトグルし、バッファにデータが書き込まれます。バッファがオーバーフローすると OVERFLOW 端子がトグルされます。

SW0 の動作とデバウンス

NRWW および RWW データセクションに対する消去および書き込み動作は、SW0 を押す事により開始されます。その後も SW0 を押すたびに同じ処理がトリガされます。RWW データセクション内の直近のデータ位置の値は、新しいデータ向けのシードとして使われます。このため、SW0 が押されてプログラムメモリが読み出された時に、ユーザは NRWW および RWW データセクション内の変更を観察できます。SW0 が押された事の確認として、LED0 が点灯します。測定用ピンのトグルをキャプチャできる外部機器(オシロスコープやロジックアナライザ)へのトリガを与えるため、SW0 ボタンを押してサンプルコードの全体の動作を開始させます。シンプルなデバウンス アルゴリズムが選択されて実装されます。TCB1 はこの目的で使われます。ソフトウェアで物理スイッチ SW0 をデバウンス処理する事により、消去および書き込み動作の誤ったトリガリングを防ぎます。フラッシュには消去/書き込み耐性の制限があり、制限を超える動作は保証されません(詳細はデバイスの電気的特性を参照してください)。

割り込みベクタテーブルの移動

割り込みベクタテーブルの配置は、制御 A (CPUINT.CTRLA)レジスタ内の割り込みベクタ選択(IVSEL)ビットの値によって決まります。割り込み要因が何も存在しない場合、通常のプログラムコードがこれらの位置に配置されます。IVSEL ビットに 1 を書き込むと割り込みベクタはフラッシュ内の BOOT セクションの先頭に配置され、RWW セクションの消去/書き込み中に割り込みを正常に実行できるようになります。「概要」で述べた通り、「RWW セクション」とはプログラミング(消去または書き込み)されるセクションを指します(読み出されるセクションを指すものではありません)。RWW セクションのプログラミング中は、CPU 命令の実行またはデータの読み出しによって NRWW セクション内のコードにのみアクセスできます。

4. MPLAB Discover からのサンプルコードの入手

MPLAB Discover は、様々なソースから入手可能なリソースを 1 つにまとめたポータルサイトです。

MPLAB Discover のウェブページ: [MPLAB Discover](#)



サンプルコード

- [Microchip Studio 向けサンプルコード](#)
- [MPLAB X 向けサンプルコード](#)

サンプルコードは、MPLAB Discover から直接 .zip ファイルとしてダウンロードできます。「*studio*」で終わるリポジトリは Microchip Studio 内で開く事ができます。「*mplab-mcc*」で終わるリポジトリは MPLAB X 内で開く事ができます。

サンプルコードが [GitHub](#) 上で提供される場合、MPLAB Discover はリンク「*Open with Github*」を提供します。そこからサンプルコードをダウンロードするか、お客様のコンピュータ上の `git` ツールを使ってローカル リポジトリ クローンを作成する事ができます。

5. 改訂履歴

リビジョン	日付	改訂内容
A	2023年10月	本書は初版です。

Microchip 社の情報

Microchip 社ウェブサイト

Microchip 社はウェブサイト(www.microchip.com)を通してオンライン サポートを提供しています。当ウェブサイトでは、お客様に役立つ情報やファイルを簡単に見つけ出せます。以下を含む各種の情報をご覧になれます。

- **製品サポート** - データシートとエラッタ、アプリケーション ノートとサンプル プログラム、設計リソース、ユーザガイドとハードウェア サポート文書、最新のソフトウェアと過去のソフトウェア
- **技術サポート** - FAQ(よく寄せられる質問)、技術サポートのご依頼、オンライン ディスカッション グループ、Microchip 社のデザイン パートナー プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** - 製品セレクトと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い合わせ先(営業所/正規代理店)の一覧

製品変更通知サービス

Microchip 社の製品変更通知サービスは、お客様に Microchip 社製品の最新情報をお届けする配信サービスです。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

<http://www.microchip.com/pcn> にアクセスし、登録手続きをしてください。

カスタマサポート

Microchip 社製品をお使いのお客様は、以下のチャンネルからサポートをご利用になれます。

- 正規代理店
- 技術サポート

サポートは正規代理店にお問い合わせください。各地の営業所もご利用になれます。本書の最後のページに各国の営業所の一覧を記載しています。

技術サポートは以下のウェブページからもご利用になれます。 www.microchip.com/support

Microchip 社のデバイスコード保護機能

Microchip 社製品のコード保護機能について以下の点にご注意ください。

- Microchip 社製品は、該当する Microchip 社データシートに記載の仕様を満たしています。
- Microchip 社では、通常の場合ならびに仕様に従って使った場合、Microchip 社製品のセキュリティ レベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- Microchip 社はその知的財産権を重視し、積極的に保護しています。Microchip 社製品のコード保護機能の侵害は固く禁じられており、デジタル ミレニアム著作権法に違反します。
- Microchip 社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。コード保護機能は常に進歩しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。

法律上の注意点

本書および本書に記載されている情報は、Microchip 社製品を設計、テスト、お客様のアプリケーションと統合する目的を含め、Microchip 社製品に対してのみ使用する事ができます。それ以外の方法でこの情報を使用する事はこれらの条項に違反します。デバイス アプリケーションの情報は、ユーザの便宜のためにのみ提供されるものであり、更新によって変更となる事があります。

お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。その他のサポートについては、弊社または代理店にお問い合わせになるか、www.microchip.com/en-us/support/design-help/client-support-services をご覧ください。

Microchip 社は本書の情報を「現状のまま」で提供しています。Microchip 社は明示的、暗黙的、書面、口頭、法定のいずれであるかを問わず、本書に記載されている情報に関して、非侵害性、商品性、特定目的への適合性の暗黙的保証、または状態、品質、性能に関する保証をはじめとするいかなる類の表明も保証も行いません。

いかなる場合も Microchip 社は、本情報またはその使用に関連する間接的、特殊的、懲罰的、偶発的または必然的損失、損害、費用、経費のいかににかかわらず、また Microchip 社がそのような損害が生じる可能性について報告を受けていた場合あるいは損害が予測可能であった場合でも、一切の責任を負いません。法律で認められる最大限の範囲を適用しようとも、本情報またはその使用に関連する一切の申し立てに対する Microchip 社の責任限度額は、使用者が当該情報に関連して Microchip 社に直接支払った額を超えません。

Microchip 社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途に Microchip 社の製品を使用する事は全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。特に明記しない場合、暗黙的あるいは明示的を問わず、Microchip 社が知的財産権を保有しているライセンスは一切譲渡されません。

商標

Microchip 社の名称とロゴ、Microchip ロゴ、Adapttec、AVR、AVR ロゴ、AVR Freaks、BesTime、BitCloud、CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi ロゴ、MOST、MOST ロゴ、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST ロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、UNI/O、Vectron、XMEGA は米国およびその他の国における Microchip Technology Incorporated の登録商標です。

AgileSwitch、ClockWorks、The Embedded Control Solutions Company、EtherSynch、Flashtec、Hyper Speed Control、HyperLight Load、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus ロゴ、Quiet-Wire、SmartFusion、SyncWorld、TimeCesium、TimeHub、TimePictra、TimeProvider、ZL は米国における 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 ロゴ、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、SQL、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、Total Endurance、Trusted Time、TSHARC、Turing、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、WiperLock、XpressConnect、ZENA は米国およびその他の国における Microchip Technology Incorporated の商標です。

SQTP は米国における Microchip Technology Incorporated のサービス マークです。

Adapttec ロゴ、Frequency on Demand、Silicon Storage Technology、Symmcom はその他の国における Microchip Technology Incorporated の登録商標です。

GestIC は、その他の国における Microchip Technology Germany II GmbH & Co. KG (Microchip Technology Inc.の子会社)の登録商標です。

その他の商標は各社に帰属します。

© 2024, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved.

ISBN: 978-1-6683-4983-0

AMBA、Arm、Arm7、Arm7TDMI、Arm9、Arm11、Artisan、big.LITTLE、Cordio、CoreLink、CoreSight、Cortex、DesignStart、DynamIQ、Jazelle、Keil、Mali、Mbed、Mbed Enabled、NEON、POP、RealView、SecurCore、Socrates、Thumb、TrustZone、ULINK、ULINK2、ULINK-ME、ULINK-PLUS、ULINKpro、 μ Vision、Versatile は、米国およびその他の国における Arm Limited (またはその子会社)の商標または登録商標です。

品質管理システム

Microchip 社の品質管理システムについては www.microchip.com/quality をご覧ください。

各国の営業所とサービス

北米	アジア/太平洋	アジア/太平洋	ヨーロッパ
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel:480-792-7200 Fax:480-792-7277 技術サポート: http://www.microchip.com/support URL: 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 SAR 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 - バンガロール Tel:91-80-3090-4444 インド - ニューデリー Tel:91-11-4160-8631 インド - プネ 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	オーストリア - ヴェルス 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 - ガーヒング 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 - パドヴァ Tel:39-049-7625286 オランダ - ドリュエネン Tel:31-416-690399 Fax:31-416-690340 ノルウェー - トロンハイム Tel:47-7288-4388 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 - マドリッド Tel:34-91-708-08-90 Fax:34-91-708-08-91 スウェーデン - ヨーテボリ 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