AVR1316 : XMEGA 自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ

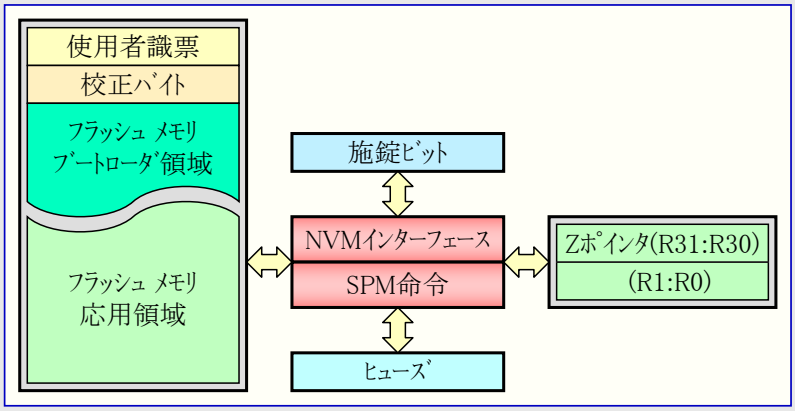
要点

- 不揮発性ﾒﾓﾘへのｿﾌﾄｳｪｱ ｱｸｾｽ
- 校正ﾊﾞｲﾄ読み込み
- ﾋｭｰｽﾞ ﾊﾞｲﾄ読み込み
- 施錠ﾋﾞｯﾄ読み書き
- 使用者識票の消去、読み込み、書き込み
- 応用領域の消去、読み込み、書き込み
- ﾌﾞｰﾄ領域の消去、読み込み、書き込み
- 応用領域とﾌﾞｰﾄ領域に対するCRC生成
- 最適化されたｱｾﾝﾌﾞﾘ言語実装
- ﾄﾞﾗｲﾊﾞのｿｰｽ ｺｰﾄﾞ内包

1. 序説

本応用記述は素早い準備と行動のための例と共にXMEGA®の自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ機能の基本的な機能を記述します。その上、Cｲﾝﾀｰﾌｪｰｽを持つｱｾﾝﾌﾞﾘ言語で書かれたﾄﾞﾗｲﾊﾞ ｲﾝﾀｰﾌｪｰｽが含まれています。

図1-1. 自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ概要



2. 部署概要

本項はXMEGA自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ機能の基本的な機能と形成任意選択の概要を提供します。

2.1. 不揮発性ﾒﾓﾘ部署

XMEGAのﾌﾗｯｼｭ ﾒﾓﾘ、識票と校正のﾊﾞｲﾄ、ﾋｭｰｽﾞと施錠のﾋﾞｯﾄは不揮発性ﾒﾓﾘ(NVM)部署を使用して全てｱｸｾｽすることができます。NVM部署はEEPROMへのｱｸｾｽにも使用されます。より多くの情報については「AVR1315:XMEGA EEPROMのｱｸｾｽ」またはﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

NVM部署が多数の目的を取り扱うので、ﾌﾗｯｼｭ ﾒﾓﾘなどのｱｸｾｽにこれを使用する前に(EEPROM更新のような)他の操作で多忙でないのを調べることが重要です。NVM状態(STATUS)ﾚｼﾞｽﾀのNVM多忙(NVMBUSY)ﾌﾗｸﾞはNVM部署が多忙の時に設定(1)されます。

自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ実行のためのNVM部署使用はNVM指令を伴います。これらは読み込み、書き込み、消去などに対する指令です。多くの指令はその指令の発行前に設定すべきｱﾄﾞﾚｽまたは指示子を必要とします。いくつかの指令は値も返します。

ﾒﾓﾘとｱｸｾｽ形式に依存して3つの異なる指令形式があります。指令形式は次からの3項目で記述されます。指令の概要とそれらの使用は以降の2.5.項で得られます。

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

応用記述

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、ATMEL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。

Rev. 8070B-11/08, 8070BJ3-03/14

2.1.1. NVM活動に基く指令

NVM活動に基く指令はNVM活動を開始するためにNVM制御ﾚｼﾞｽﾀA(CTRLA)のNVM指令実行(CMDEX)ﾋﾞｯﾄが設定(1)されなければならない命令です。NVM指令実行ﾋﾞｯﾄは必要とする制御ﾚｼﾞｽﾀが予め設定された後で設定(1)されなければなりません。NVM活動に基く指令はEEPROM、ﾋｭｰｽﾞ ﾋﾞｯﾄ、施錠ﾋﾞｯﾄのｱｸｾｽとﾌﾗｯｼｭ ﾒﾓﾘの領域に対するCRC計算に使用されます。

正確な手順は次の通りです。

1. 必要なら、NVMｱﾄﾞﾚｽ(ADDRn)ﾚｼﾞｽﾀにｱﾄﾞﾚｽまたは指示子を格納してください。
2. 必要なら、NVMﾃﾞｰﾀ(DATAn)ﾚｼﾞｽﾀにﾃﾞｰﾀを格納してください。
3. NVM指令(CMD)ﾚｼﾞｽﾀ内に指令符号を格納してください。
4. 形態設定変更保護(CCP)ﾚｼﾞｽﾀ内にI/Oﾚｼﾞｽﾀ保護識票(ﾊﾞｲﾄ値$D8)を格納してください。これは次からの4CPU命令周期間、全ての割り込みを自動的に禁止します。
5. 次からの4CPU命令周期内で、NVM制御ﾚｼﾞｽﾀA(CTRLA)のNVM指令実行(CMDEX)ﾋﾞｯﾄを設定(1)してください。
6. その操作はNVM多忙(NVMBUSY)ﾌﾗｸﾞが解除(0)される時に終了されます。
7. その操作からの結果はNVMﾃﾞｰﾀ(DATAn)ﾚｼﾞｽﾀで利用可能です。

次の操作がNVM活動に基く指令を使用します。

- ﾋｭｰｽﾞ ﾊﾞｲﾄ読み込み
- 施錠ﾋﾞｯﾄ書き込み
- 応用領域に対するCRC生成
- ﾌﾞｰﾄ領域に対するCRC生成

2.1.2. LPMに基く指令

LPMに基く指令はLPM命令が使用される指令です。LPM命令は必要とする制御ﾚｼﾞｽﾀが予め設定された後で実行されます。LPMに基く指令はﾌﾗｯｼｭ ﾒﾓﾘの読み込みに使用されます。XMEGAﾃﾞﾊﾞｲｽでは応用ｺｰﾄﾞ、校正ﾊﾞｲﾄ、識票ﾊﾞｲﾄがﾌﾗｯｼｭ ﾒﾓﾘに配置されています。

正確な手順は次の通りです。

1. 必要なら、Zﾎﾟｲﾝﾀ(R31とR30)内にｱﾄﾞﾚｽまたは指示子を格納してください。
2. NVM指令(CMD)ﾚｼﾞｽﾀ内に指令符号を格納してください。
3. LPM命令を実行してください。
4. この操作は直ちに終了します。
5. この操作からの結果はR0ﾚｼﾞｽﾀで利用可能です。

次の操作がLPMに基く指令を使用します。

- 校正ﾊﾞｲﾄ読み込み
- 使用者ｱｸｾｽ可能な識票ﾊﾞｲﾄ読み込み
- ﾌﾗｯｼｭ ﾒﾓﾘ読み込み

2.1.3. SPMに基く指令

SPMに基く指令はCPUがSPM命令を実行する指令です。SPM命令は必要とする制御ﾚｼﾞｽﾀが予め設定された後で実行されます。SPMに基く指令はﾌﾗｯｼｭ ﾒﾓﾘの消去と書き込みに使用されます。XMEGAﾃﾞﾊﾞｲｽでは応用ｺｰﾄﾞ、校正ﾊﾞｲﾄ、識票ﾊﾞｲﾄがﾌﾗｯｼｭ ﾒﾓﾘに配置されています。

正確な手順は次の通りです。

1. 必要なら、Zﾎﾟｲﾝﾀ(R31とR30)内にｱﾄﾞﾚｽまたは指示子を格納してください。
2. 必要なら、R1とR0のﾚｼﾞｽﾀ内にﾃﾞｰﾀを格納してください。
3. NVM指令(CMD)ﾚｼﾞｽﾀ内に指令符号を格納してください。
4. 形態設定変更保護(CCP)ﾚｼﾞｽﾀ内にSPM保護識票(ﾊﾞｲﾄ値$9D)を格納してください。これは次からの4CPU命令周期間、全ての割り込みを自動的に禁止します。
5. 次からの4CPU命令周期内で、SPM命令を実行してください。
6. その操作はNVM多忙(NVMBUSY)ﾌﾗｸﾞが解除(0)される時に終了されます。

次の操作がSPMに基く指令を使用します。

- 使用者識票列消去
- 使用者識票列書き込み
- ﾌﾗｯｼｭ ﾒﾓﾘ消去
- ﾌﾗｯｼｭ ﾒﾓﾘ ﾍﾟｰｼﾞ緩衝部格納(設定)
- ﾌﾗｯｼｭ ﾒﾓﾘ書き込み
- ﾌﾗｯｼｭ ﾒﾓﾘ ﾍﾟｰｼﾞ緩衝部破棄

2.2. フラッシュ メモリの消去と書き込み

フラッシュ メモリを更新するには非分離操作と分離操作の2つの方法があります。非分離書き込みでは、フラッシュ メモリ位置が1操作で消去されて書かれます。分離操作では消去と書き込みが独立した操作です。

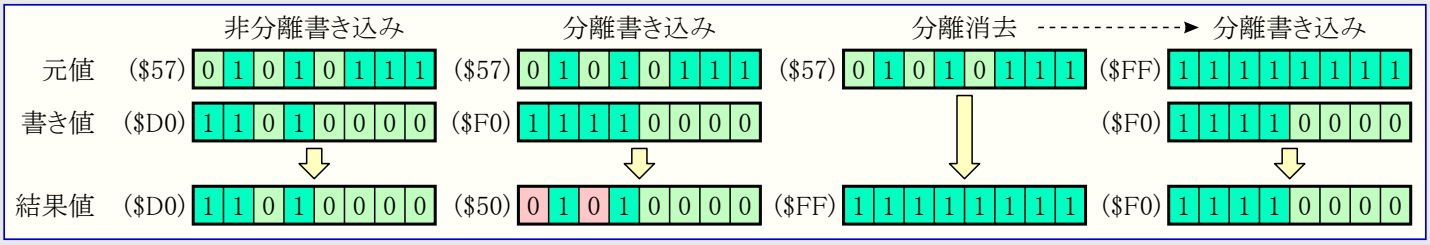
フラッシュ メモリ位置消去時、全ビットが論理1に設定されます。その後に分離書き込みは選択したビットを論理0にプログラミングすることができます。最初に論理1へ消去してその後に選択したビットに論理0を書く非分離書き込みと異なり、分離書き込み操作はビットを0から1に変更することができません。これは異なる値での1つの位置への複数書き込みがその間でその位置が消去されない場合に、結局全位置が論理0になりつつある結果に終わります。これは既存値と書き込み値間の(正)論理AND操作と同様です。

非分離書き込みは1つの消去または1つの書き込みの時間の概ね倍の時間がかかります。従って、予め(例えば初期化の間に)フラッシュ メモリ位置を消去することによって、時間を節約するのに使用することができます。例えば、応用が電力低下検知時に極めて重要なデータの保存を必要とする場合、分離書き込みは非分離書き込みよりも短い時間で済みます。

分離書き込みに関する他の有用な応用は、特に頻繁にフラッシュ メモリ位置を更新する応用に対してフラッシュ メモリの耐久性を増すことです。フラッシュ メモリ位置が消去されなければならない場合以外、フラッシュ メモリ位置が消去されない仕組みを実装することによって、分離操作機能はフラッシュ メモリの耐久性を増します。これはEEPROMに対する分離と非分離の書き込みと同様です。より多くの詳細については「AVR101:高耐久性EEPROM記憶」応用記述を参照してください。

消去操作と書き込み操作の違いは操作前後のバイト値と共に下の図2-1.で図解されます。例えフラッシュ メモリ領域が語(ワード)アドレス指定でも、簡単化のため、この図はバイト アドレスを示します。

図2-1. 非分離書き込み、分離消去と書き込みの動作



2.3. ページ一時緩衝部

フラッシュ メモリは順番が語(ワード)アドレス指定のページで構成されます。消去と書き込みの両動作はページで操作されます。ページ容量はメモリ容量に依存し、デバイスのデータ シートで与えられます。消去と書き込みの操作はフラッシュ メモリに書かれるべきページ全体の準備が整うまで個別語(ワード)を格納するのにページ一時緩衝部を使用します。

フラッシュ ページ緩衝部は「AVR1315:XMEGA EEPROMのアクセス」応用記述で記述される、EEPROMページ緩衝部と非常に良く似ています。違いは例えフラッシュ ページ緩衝部内に少数のデータだけが格納されていても、データ書き込み時にページ全体が更新される、またはページ消去実行時にページ全体が消去されます。EEPROMページ緩衝部については、EEPROMページ緩衝部の書き込みまたは消去時に格納した緩衝部位置だけが影響を及ぼされます。使い方の詳細についてはコード例を学習してください。

実際のフラッシュ ページ書き込み操作は2つの操作、(1)ページ緩衝部へのデータ格納と(2)フラッシュ メモリのページへのデータ書き込みから成ります。ページ緩衝部への設定時、ページ緩衝部内の語(ワード)を選択するのにフラッシュ メモリ アドレスの下位部分が使用され、一方上位部分は無視されます。ページの書き込みまたは消去時、アドレスの上位部分がページを選択し、一方下位部分は無視されます。フラッシュ メモリが語(ワード)アドレス指定のため、アドレスの最下位ビット(LSB)は常に無視されます。

最初に語(ワード)が一旦緩衝部内に格納されると、NVM状態(STATUS)レジスタのフラッシュ ページ緩衝部設定中(FLOAD)フラグが設定(1)されます。このビットは緩衝部が破棄されるか、またはページ書き込み(非分離または分離の書き込み)が実行されるかのどちらかまで設定(1)に留まります。

ページ緩衝部内の各語(ワード)位置が緩衝部の破棄またはページ書き込みの前に1度だけ書くことができることに注意してください。

2.4. 応用領域とﾌﾞｰﾄ領域

XMEGAのﾌﾗｯｼｭ ﾒﾓﾘは応用領域とﾌﾞｰﾄ領域の2つの領域に分けられています。応用領域は通常の応用ﾌｧｰﾑｳｪｱを格納し、一方ﾌﾞｰﾄ領域は大抵ﾌﾞｰﾄﾛｰﾀﾞ応用を格納するのに使用されます。

ﾌﾞｰﾄ領域は通常の応用ﾌｧｰﾑｳｪｱにも使用することができます。SPMに基く指令を実行するｺｰﾄﾞがﾌﾞｰﾄ領域からだけ走行することができるので、この領域は大抵ﾌﾞｰﾄﾛｰﾀﾞやﾌﾗｯｼｭ ﾒﾓﾘの更新または他のSPMに基く指令の走行を必要とする応用ﾌｧｰﾑｳｪｱの一部に使用されます。

応用領域とﾌﾞｰﾄ領域には分離された施錠ﾋﾞｯﾄがあり、故にこの2つの領域に対する読みと/または書きは個別に制限することができます。また、応用領域の上部は残り部分から独立した自身の施錠ﾋﾞｯﾄの組を持っています。この上位領域はﾌﾗｯｼｭ ﾒﾓﾘでの大きな表の格納と保守を典型とするので、応用表領域と呼ばれます。

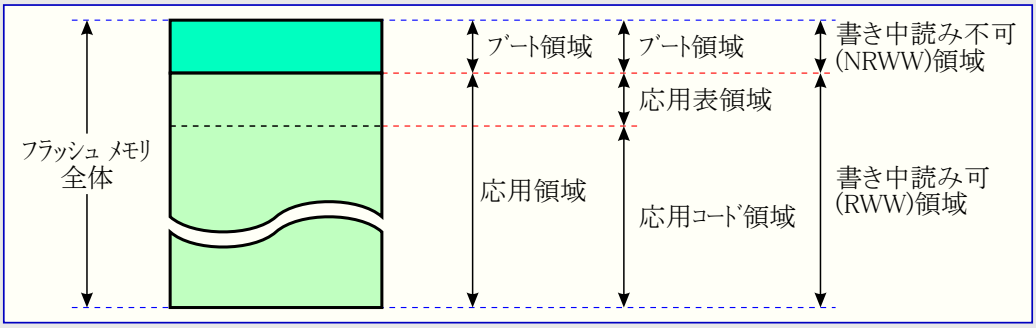
応用表領域を使用する一般的な筋書きは次の通りです。

- 応用領域に応用ﾌｧｰﾑｳｪｱを格納し、対応する施錠ﾋﾞｯﾄを使用して施錠してください。
- 応用表領域に大きな表の格納と維持をし、後で更新を必要する場合は不施錠のままにしてさい。
- ﾌﾞｰﾄ領域に(SPMに基く指令を使用して)ﾌﾗｯｼｭ ﾒﾓﾘの表を保守するﾌｧｰﾑｳｪｱ部分を格納し、対応する施錠ﾋﾞｯﾄを使用してﾌﾞｰﾄ領域を施錠してください。

応用表領域容量が常にﾌﾞｰﾄ領域容量と等しいことに注意してください。

下の図2-2.は相互に関連する各種ﾌﾗｯｼｭ ﾒﾓﾘ領域の概要を示します。

図2-2. ﾌﾗｯｼｭ ﾒﾓﾘ構成



2.4.1. 書き中読み不可(NRWW)領域

ﾌﾞｰﾄ領域全体は書き中読み不可(NRWW)領域として参照されます。NRWW領域が書かれつつある間、ﾌﾗｯｼｭ ﾒﾓﾘのどの位置からの読み込みも不可能です。読み込みにはｺｰﾄﾞ実行とLPMに基く指定を使用するﾌﾗｯｼｭ ﾒﾓﾘのﾃﾞｰﾀ読み込みを含みます。

SPMに基く指令はﾌﾞｰﾄ領域からだけ実行することができます。ﾌﾞｰﾄﾛｰﾀﾞ応用が自身の更新を欲した場合、ﾌﾞｰﾄ領域が書かれつつある間、命令実行は停止されます。

2.4.2. 書き中読み可(RWW)領域

応用領域全体は書き中読み可(RWW)領域として参照されます。これはこの領域が書かれつつある間にこの領域からの読み込みが可能なことを意味しません。それは応用領域が更新されつつある間にﾌﾞｰﾄ領域からｺｰﾄﾞを読んで実行するのが可能なことを意味します。

この特性は応用ﾌｧｰﾑｳｪｱまたは応用表領域のﾌﾗｯｼｭ ﾒﾓﾘ表を更新する間、重要な機能の走行を保つことを可能にします。

2.4.3. SPM施錠

或る応用は始動設定後に応用領域を更新し、その後は次のﾘｾｯﾄまたは電源OFF/ONまでそのままにします。この使い方に対する安全な予防策として、NVM部署は如何なるSPMに基く指令に対しても更なる全てのｱｸｾｽを施錠することができます。

SPM施錠指令はNVM活動に基く指定の変形を必要とします。正確な手順は次の通りです。

1. 形態設定変更保護(CCP)ﾚｼﾞｽﾀ内にI/Oﾚｼﾞｽﾀ保護識票(ﾊﾞｲﾄ値$D8)を格納してください。これは次からの4CPU命令周期間、全ての割り込みを自動的に禁止します。
2. 次からの4CPU命令周期内で、NVM制御ﾚｼﾞｽﾀB(CTRLB)のSPM施錠(SPMLOCK)ﾋﾞｯﾄを設定(1)してください。

ｿﾌﾄｳｪｱ例はこの機能の実装を含みます。

2.4.4. ﾌﾞｰﾄ ﾘｾｯﾄ ﾍﾞｸﾀ

応用領域を更新するﾌﾞｰﾄﾛｰﾀﾞ ﾌｧｰﾑｳｪｱを使用する応用は電源投入またはﾘｾｯﾄ後、応用領域の代わりにﾌﾞｰﾄ領域のｺｰﾄ実行を始めるようにXMEGAを形態設定する必要があります。応用領域の代わりにﾌﾞｰﾄ領域の最初の位置(RESET割り込みﾍﾞｸﾀ)にｱﾄﾞﾚｽを初期設定する形態設定に対して専用のﾋｭｰｽﾞ ﾋﾞｯﾄが使用されます。ﾋｭｰｽﾞ設定とﾌﾟﾛｸﾞﾗﾐﾝｸﾞの情報についてはﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

2.4.5. 割り込みﾍﾞｸﾀ表位置

或る応用は例え応用領域が更新されつつある時でも選ばれた重要な機能の走行を保つことを必要とします。これらの機能は度々割り込み制御され、更新中の全ての時間で許可されて実行可能なことを必要とします。応用領域更新時、ｺｰﾄﾞは応用領域から実行することができません。割り込み処理ｺｰﾄﾞはﾌﾞｰﾄ領域に2重化される必要があり、割り込みﾍﾞｸﾀ表が再配置されなければなりません。

一般的な割り込みの情報と割り込みﾍﾞｸﾀ表の移動方法の詳細については「**AVR1305:XMEGAの割り込みと設定可能な多段割り込み制御器**」応用記述を参照してください。

2.4.6. 消費電力削減

既知のどの時間でも未使用のﾌﾗｯｼｭ ﾒﾓﾘ領域(応用領域またはﾌﾞｰﾄ領域)を禁止するようにNVM部署を形態設定することが可能です。禁止されたﾌﾗｯｼｭ ﾒﾓﾘ領域では消費電流がないので、この機能は応用での消費電力を最小にします。既定での両領域は許可ですが、NVM制御ﾚｼﾞｽﾀB(CTRLB)のﾌﾗｯｼｭ電力削減動作許可(FPRM)ﾋﾞｯﾄの設定(1)によって、ｺｰﾄﾞを実行している領域だけが許可されます。実行の流れが或る領域から別の領域へ移動する(例えば関数呼び出しの)場合、CPUは6ｸﾛｯｸ周期停止され、同時に或る領域が許可され、別の領域が禁止されます。

この6周期の罰則はLPM操作が禁止されたﾌﾗｯｼｭ ﾒﾓﾘ領域をｱｸｾｽする時にも適用します。

応用表領域は応用領域の一部で、応用領域が許可される時に許可されることに注意してください。

2.5. 指令要約

下の**表2-1.**はｿｰｽ ｺｰﾄﾞで使用した象徴的な名称に沿った自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ操作に対して利用可能な全てのVNM指令の概要を与えます。指令のﾊﾞｲﾄ値、その形式、簡単な内容も含まれます。より多くの詳細については例のｿｰｽ ｺｰﾄﾞを学習してください。

表2-1. 関連NVM指令

指令	符号	形式	内容
NO_OPERATION	$00	(LPM)	無操作/ｱﾄﾞﾚｽZからR0内にﾌﾗｯｼｭ ﾊﾞｲﾄ読み込み
READ_USER_SIG_ROW	$01	LPM	指示子Zの使用者識票ﾊﾞｲﾄをR0内に読み込み
READ_CALIB_ROW	$02	LPM	指示子Zの校正ﾊﾞｲﾄをR0内に読み込み
READ_FUSES	$07	NVM	指示子ADDR0のﾋｭｰｽﾞ ﾊﾞｲﾄをDATA0内に読み込み
WRITE_LOCK_BITS	$08	NVM	DATA0を施錠ﾋﾞｯﾄに書き込み
ERASE_USER_SIG_ROW	$18	SPM	指示子Zの使用者識票ﾊﾞｲﾄを消去
WRITE_USER_SIG_ROW	$1A	SPM	R0を指示子Zの使用者識票ﾊﾞｲﾄに書き込み
ERASE_APP	$20	SPM	応用領域全体を消去
ERASE_APP_PAGE	$22	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ｱﾄﾞﾚｽZの応用領域ﾍﾟｰｼﾞを消去
LOAD_FLASH_BUFFER	$23	SPM	ｱﾄﾞﾚｽ下位ﾋﾞｯﾄだけ使用し、 R1:R0の語(ﾜｰﾄﾞ)をﾍﾟｰｼﾞ緩衝部内のｱﾄﾞﾚｽZに格納(設定)
WRITE_APP_PAGE	$24	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ﾍﾟｰｼﾞ緩衝部をｱﾄﾞﾚｽZの応用領域ﾍﾟｰｼﾞに書き込み
ERASE_WRITE_APP_PAGE	$25	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ｱﾄﾞﾚｽZの応用領域ﾍﾟｰｼﾞを消去して、 ﾍﾟｰｼﾞ緩衝部をｱﾄﾞﾚｽZの応用領域ﾍﾟｰｼﾞに書き込み
ERASE_FLASH_BUFFER	$26	SPM	ﾍﾟｰｼﾞ緩衝部を破棄(消去)
ERASE_BOOT_PAGE	$2A	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ｱﾄﾞﾚｽZのﾌﾞｰﾄ領域ﾍﾟｰｼﾞを消去
WRITE_BOOT_PAGE	$2C	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ﾍﾟｰｼﾞ緩衝部をｱﾄﾞﾚｽZのﾌﾞｰﾄ領域ﾍﾟｰｼﾞに書き込み
ERASE_WRITE_BOOT_PAGE	$2D	SPM	ｱﾄﾞﾚｽ上位ﾋﾞｯﾄだけ使用し、ｱﾄﾞﾚｽZのﾌﾞｰﾄ領域ﾍﾟｰｼﾞを消去して、 ﾍﾟｰｼﾞ緩衝部をｱﾄﾞﾚｽZのﾌﾞｰﾄ領域ﾍﾟｰｼﾞに書き込み
APP_CRC	$38	NVM	応用領域からCRCを計算し、DATA2:DATA1:DATA0に格納
BOOT_CRC	$39	NVM	ﾌﾞｰﾄ領域からCRCを計算し、DATA2:DATA1:DATA0に格納

CRC生成に関する詳細な方法はﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄで得られます。

2.5.1. 特殊状態: 施錠ﾋﾞｯﾄ読み込み

施錠ﾋﾞｯﾄはNVM施錠ﾋﾞｯﾄ(LOCKBITS)ﾚｼﾞｽﾀ内にI/O割り当てされ、どんなNVM指令もなしに直接読むことができます。

2.6. 割り込みの考慮

同一応用で割り込みと自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ操作を使用する時は以下が考慮されるべきです。

- ﾌﾗｯｼｭ ﾍﾟｰｼﾞ緩衝部内容を不正にしないよう注意してください。ﾍﾟｰｼﾞ緩衝部設定時、割り込み処理がﾍﾟｰｼﾞ緩衝部をｱｸｾｽしないことを確認してください。割り込み処理がﾍﾟｰｼﾞ緩衝部のｱｸｾｽに使用される場合、同時に応用の他の部分がﾍﾟｰｼﾞ緩衝部をｱｸｾｽしないことを確認してください。
- NVM操作終了時を検知するのにNVM多忙(NVMBUSY)ﾌﾗｸﾞを継続的にﾎﾟｰﾘﾝｸﾞする代わりに、NVM割り込み制御(INTCTRL)ﾚｼﾞｽﾀのSPM操作可割り込みﾚﾍﾞﾙ(SPMLVL)ﾋﾞｯﾄ領域で適切な割り込みﾚﾍﾞﾙを設定することによってSPM割り込みを許可することが可能です。対応する割り込み処理はNVM多忙(NVMBUSY)ﾌﾗｸﾞが設定(1)されていない時に必ず呼び出されます。これは割り込み制御ﾌﾗｯｼｭ ﾒﾓﾘ更新の実装に使用することができます。割り込みのより多くの情報は「**AVR1305:XMEGAの割り込みと設定可能な多段割り込み制御器**」応用記述で得られます。

3. 始める前に

本項はXMEGAの自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞでの実験とでの準備と実行に対する基本段階を簡単に片付けます。必要なﾚｼﾞｽﾀが関連ﾋﾞｯﾄ設定と共に記述されます。少しの一般的筋書きが以下の項で記述されます。更なる例と詳細については例のｿﾌﾄｳｪｱを学習してください。

3.1. 全応用領域更新

この一般的筋書きはﾊﾟｿｺﾝの応用を通して直列通信からﾌﾗｯｼｭ ﾒﾓﾘのﾃﾞｰﾀを更新するﾌﾞｰﾄﾛｰﾀﾞ用です。応用領域全更新の場合に推奨される手順は次の通りです。

1. SPM指令ERASE_APPで応用領域全体を消去してください。
2. NVM多忙(NVMBUSY)ﾌﾗｸﾞが解除(0)されるのを待ってください。
3. 通信ﾁｬﾈﾙを伝わって来る値するﾃﾞｰﾀを1ﾍﾟｰｼﾞ取得してください。
4. SPM指令LOAD_PAGE_BUFFERを使用してﾃﾞｰﾀをﾍﾟｰｼﾞ緩衝部に格納(設定)してください。
5. 新ﾃﾞｰﾀをﾌﾗｯｼｭ ﾒﾓﾘのﾍﾟｰｼﾞへ書くのに、SPM指令WRITE_APP_PAGEを使用してください。
6. NVM多忙(NVMBUSY)ﾌﾗｸﾞが解除(0)されるのを待ってください。
7. 全てのﾍﾟｰｼﾞが更新されるまで**手順3.**からを繰り返してください。

3.2. 選択位置更新

他の一般的筋書きはﾌﾗｯｼｭ ﾒﾓﾘ上の選択位置、例えばﾌﾗｯｼｭ ﾒﾓﾘ(なるべくなら応用表領域)に格納された定数表またはﾊﾟﾗﾒｰﾀを更新することです。このような読み-変更-書き操作に対して推奨される手順は次の通りです。

1. 素直にLPM読み込み指令(NO_OPERATION)を使用して、更新されるべき位置を含むﾌﾗｯｼｭ ﾒﾓﾘのﾍﾟｰｼﾞをSRAM緩衝部内に読んでください。
2. SRAM緩衝部内で選択位置を更新してください。
3. SPM指令LOAD_PAGE_BUFFERを使用して更新されたSRAM緩衝部からﾃﾞｰﾀをﾍﾟｰｼﾞ緩衝部に格納(設定)してください。
4. 以前の内容を消去して新ﾃﾞｰﾀをﾌﾗｯｼｭ ﾒﾓﾘのﾍﾟｰｼﾞへ書くのに、SPM指令ERASE_WRITE_APP_PAGEを使用してください。
5. NVM多忙(NVMBUSY)ﾌﾗｸﾞが解除(0)されるのを待ってください。

4. ﾄﾞﾗｲﾊﾞ実装

本応用記述はCｲﾝﾀｰﾌｪｰｽを持つｱｾﾝﾌﾞﾘ言語で実装された基本的な自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ ﾄﾞﾗｲﾊﾞの一括ｿｰｽ ｺｰﾄﾞを含みます。より多くの詳細についてはﾄﾞﾗｲﾊﾞのｿｰｽ ｺｰﾄﾞとﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

4.1. ファイル

一括ｿｰｽ ｺｰﾄﾞは4つのﾌｧｲﾙから成ります。

- sp_driver.s　　：自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ ﾄﾞﾗｲﾊﾞ ｿｰｽ ﾌｧｲﾙ
- sp_driver.s90　：自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ ﾄﾞﾗｲﾊﾞ ｿｰｽ ﾌｧｲﾙ
- sp_driver.h　　：自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ ﾄﾞﾗｲﾊﾞ ﾍｯﾀﾞ ﾌｧｲﾙ
- sp_example.c　：自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ ﾄﾞﾗｲﾊﾞを使用するｺｰﾄﾞ例

利用可能なﾄﾞﾗｲﾊﾞ ｲﾝﾀｰﾌｪｰｽ関数とそれらの使用の完全な概要についてはｿｰｽ ｺｰﾄﾞの資料を参照してください。

4.2. Doxygen資料化

全てのｿｰｽ ﾌｧｲﾙはDoxygenを使用する自動資料生成用に準備されています。Doxygenは特別なｷｰﾜｰﾄﾞを使用してｿｰｽ ｺｰﾄﾞを分析することによって、ｿｰｽ ｺｰﾄﾞから資料を作成するﾂｰﾙです。Doxygenについてのより多くの詳細に関してはhttp://www.doxygen.orgを訪ねてください。予めｺﾝﾊﾟｲﾙされたDoxygen資料は本応用記述に伴うｿｰｽ ｺｰﾄﾞと共に供給され、ｿｰｽ ｺｰﾄﾞ ﾌｫﾙﾀﾞのreadme.htmlﾌｧｲﾙから利用可能です。

本社

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan
104-0033 東京都中央区
新川1-24-8
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ｳｪﾌﾞｻｲﾄ
www.atmel.com

技術支援
avr@atmel.com

販売窓口
www.atmel.com/contacts

文献請求
www.atmel.com/literature

本応用記述はATMELのAVR1316応用記述(doc8070.pdf Rev.8070B-11/08)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訳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。