AVR105：ﾌﾗｯｼｭ ﾒﾓﾘでの電力効率的な高耐久性ﾊﾟﾗﾒｰﾀ記憶

8-bit AVR®
マイクロ コントローラ

応用記述

要点

- ﾊﾟﾗﾒｰﾀの高速記憶
- 高耐久性ﾌﾗｯｼｭ ﾒﾓﾘ記憶 – 350,000書き込み回数
- 電力効率的なﾊﾟﾗﾒｰﾀ記憶
- 任意容量のﾊﾟﾗﾒｰﾀ
- 半冗長ﾊﾟﾗﾒｰﾀ記憶
- 書かれたﾊﾟﾗﾒｰﾀの任意選択検証
- 電力不足での任意選択回復

序説

組み込みｼｽﾃﾑはﾘｾｯﾄや電力損失に渡って保護できるﾊﾟﾗﾒｰﾀを信頼します。いくつかのｼｽﾃﾑでは、この静的情報が始動で状態を正しくするためのｼｽﾃﾑ初期化に使用され、別のｼｽﾃﾑでは、ｼｽﾃﾑ履歴や累積されたﾃﾞｰﾀの記録に使用されます。EEPROMがこれに使用できますが、複数ﾊﾞｲﾄが同時に格納される必要がある時にﾌﾗｯｼｭ ﾒﾓﾘの速度と釣り合うことができません。

ﾌﾗｯｼｭ ﾒﾓﾘがより大きなﾊﾟﾗﾒｰﾀ群に対してより効果的な理由は、ﾍﾟｰｼﾞ ﾌﾟﾛｸﾞﾗﾐﾝｸﾞが使用できることで、そしてそれはﾌﾟﾛｸﾞﾗﾐﾝｸﾞ時間を減らします。ﾊﾞｲﾄ当たりのﾌﾟﾛｸﾞﾗﾐﾝｸﾞ時間は複数ﾊﾞｲﾄのﾊﾟﾗﾒｰﾀ群格納時にEEPROMに対するよりもﾌﾗｯｼｭ ﾒﾓﾘに対する方がそれによってより短くなります。より高速な記憶方法の直接的な結果として、より多くの時間が休止形態で費やせるために電力消費が削減できます。

本応用記述はAVRの自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ機能を使用する、ﾌﾗｯｼｭ ﾒﾓﾘでの高耐久性ﾊﾟﾗﾒｰﾀ記憶法の実装方法を記述します。ﾌﾗｯｼｭ ﾒﾓﾘのﾍﾟｰｼﾞ全体と循環緩衝部に対して使用されるのと同様な入出力方法を利用することによって、ﾌﾗｯｼｭ ﾒﾓﾘのﾍﾟｰｼﾞ内の各単一ﾒﾓﾘ位置は1つのﾒﾓﾘ位置が使用される度毎に書き込みｱｸｾｽされません。この方法は記憶部の耐久性を増し、記憶領域が"使い古し"でないことを保証します。記憶の耐久性はﾊﾟﾗﾒｰﾀ群容量と記憶"緩衝部"に対して割り当てられたﾍﾟｰｼﾞ容量間の比率に比例します。

動作理論

megaAVR®系統は"自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ"と呼ばれる機能を持ちます。この機能はAVRに対して内部ﾌﾗｯｼｭ ﾒﾓﾘの再ﾌﾟﾛｸﾞﾗﾐﾝｸﾞを可能にします。このﾌﾟﾛｸﾞﾗﾑ用ﾒﾓﾘは全てのAVRで定数、そして今や走行時のﾌﾗｯｼｭ ﾒﾓﾘ内容が変更可能なので、ﾊﾟﾗﾒｰﾀの格納にも使用できます。

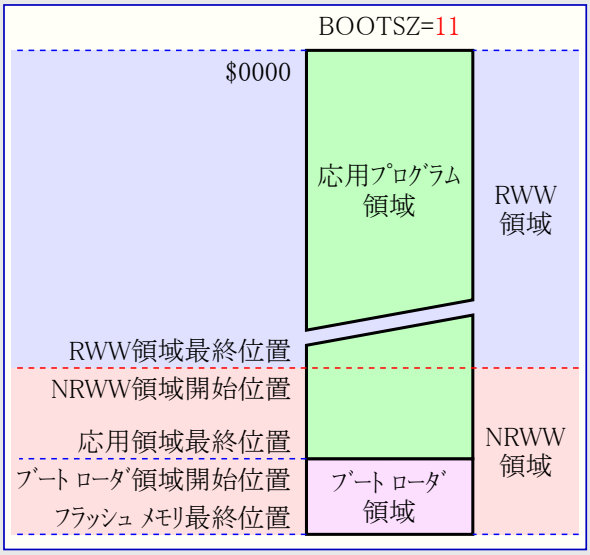
けれども、ﾊﾟﾗﾒｰﾀ記憶にﾌﾗｯｼｭ ﾒﾓﾘを使用することは、例えばEEPROMのｲﾝﾀｰﾌｪｰｽのように至極簡単ではありません。自己ﾌﾟﾛｸﾞﾐﾝｸﾞ機能はﾌｧｰﾑｳｪｱ更新に対して使用されることを意図されていますが、その柔軟性はﾌﾗｯｼｭ ﾒﾓﾘのﾊﾟﾗﾒｰﾀ更新に対して上手く使用されることを可能にします。本項はﾊﾟﾗﾒｰﾀ記憶に対してAVRの内部ﾌﾟﾛｸﾞﾗﾑ用ﾒﾓﾘを使用するために必要なﾌﾗｯｼｭ ﾒﾓﾘについての基本的な情報を記述します。

Rev. 2546A-09/03, 2546AJ3-01/14

書き込み中に読める(RWW:Read-While-Write)ﾌﾗｯｼｭ ﾒﾓﾘ

ﾌﾗｯｼｭ ﾒﾓﾘはSPM命令を使用して再ﾌﾟﾛｸﾞﾗﾐﾝｸﾞすることができます。SPM命令はこのﾒﾓﾘのﾌﾞｰﾄ領域からだけ実行できます。応用領域からのSPM命令実行は無効です。応用領域はｱﾄﾞﾚｽ$0000からﾌﾞｰﾄ領域の始まりまでに配置されます(**図1.**をご覧ください)。4つの異なるﾌﾞｰﾄ領域容量が選択できます。ﾌﾞｰﾄ領域容量はﾋｭｰｽﾞ設定によって決められます。選択できるﾌﾞｰﾄ容量は使用するAVRに依存します。

図1. ATmega128の応用領域とﾌﾞｰﾄ領域のﾒﾓﾘ配置



ﾌﾞｰﾄ領域は常に書き込み中に読めない(NRWW:No-Read-While-Write)領域として参照されるﾌﾗｯｼｭ ﾒﾓﾘの部分に配置されます。応用領域は常に書き込み中に読める(RWW:Read-While-Write)領域と呼ばれるるﾌﾗｯｼｭ ﾒﾓﾘの部分を含みますが(**図1.**をご覧ください)、選択したﾌﾞｰﾄ領域容量に依存して、NRWW部のいくらかも上手く含められます。最大ﾌﾞｰﾄ領域容量が選択される時にﾌﾞｰﾄ領域はNRWW部全体を取り、従って応用領域は純粋なRWWﾒﾓﾘです。NRWWとRWWの部分は固定され、ﾌﾞｰﾄ領域容量によって影響を及ぼされません。これについてのより詳細に関しては自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ付きﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

NRWW部とRWW部間の違いは、RWW用領域内のﾍﾟｰｼﾞ消去または書き込みの間、AVRがﾒﾓﾘのNRWW部に配置されたﾌﾟﾛｸﾞﾗﾑ ｺｰﾄﾞの実行を継続できることです。これはNRWW部内のﾍﾟｰｼﾞ消去または書き込み時には不可能で、ﾌﾗｯｼｭ ﾒﾓﾘのNRWW部内のﾍﾟｰｼﾞ変更の間、AVRｺｱは停止されます。荒っぽく言えば、応用領域内で変更されるﾍﾟｰｼﾞがRWWﾒﾓﾘ内に配置されていると仮定して、ﾌﾞｰﾄ領域に配置されたｺｰﾄﾞは応用領域がﾌﾟﾛｸﾞﾗﾐﾝｸﾞされている間に実行することができます。

従って、ﾌﾗｯｼｭ ﾊﾟﾗﾒｰﾀを更新するｺｰﾄﾞ部分はﾌﾞｰﾄ領域に配置されなければならず、AVRがﾊﾟﾗﾒｰﾀ更新間に動作を継続するなら、ﾌﾗｯｼｭ ﾊﾟﾗﾒｰﾀは応用領域のRWW部分に配置されなければなりません。これはﾊﾟﾗﾒｰﾀ書き込みの間に割り込みが防がれなければならないかの実証が求められます。

ﾌﾗｯｼｭ ﾒﾓﾘ ｾﾙの消去とﾌﾟﾛｸﾞﾗﾐﾝｸﾞ

ﾌﾗｯｼｭ ﾒﾓﾘは各々が単一ﾋﾞｯﾄを表す独立したｾﾙから成ります。ﾌﾗｯｼｭ ｾﾙはﾌﾛｰﾃｨﾝｸﾞ ｹﾞｰﾄ ﾄﾗﾝｼﾞｽﾀ技術を基にし、ﾄﾗﾝｼﾞｽﾀのｹﾞｰﾄ上での電荷"捕獲"がﾌﾗｯｼｭ ｾﾙの読む論理ﾚﾍﾞﾙを決めます。次の方が僅かに平易でしょう。ｾﾙの動きは以下のように説明できます。ｾﾙを消去するとき、ｹﾞｰﾄに充電が行われてｾﾙは論理1として読まれます。ﾌﾗｯｼｭの"ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ(0書き込み)"は論理ﾚﾍﾞﾙを0に持って来る、ｹﾞｰﾄ放電と当価です。消去(充電)されているｾﾙのﾌﾟﾛｸﾞﾗﾐﾝｸﾞ(放電)だけが可能です。

ﾌﾗｯｼｭ ﾒﾓﾘはﾍﾟｰｼﾞで配置されます。ﾌﾗｯｼｭの消去とﾌﾟﾛｸﾞﾗﾐﾝｸﾞは自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ使用時、ﾍﾟｰｼﾞで行われます。ﾌﾗｯｼｭ消去はﾍﾟｰｼﾞ全体で実行され、ﾌﾗｯｼｭ書き込みはﾍﾟｰｼﾞ全体で実行されます。

ﾋﾞｯﾄが個別にﾌﾟﾛｸﾞﾗﾐﾝｸﾞできることに注意してください。ﾌﾟﾛｸﾞﾗﾑ(0)されるべきﾋﾞｯﾄだけが放電されるため、残りの非ﾌﾟﾛｸﾞﾗﾑ(1)ﾋﾞｯﾄは未だ充電されています。どの非ﾌﾟﾛｸﾞﾗﾑ(1)ﾋﾞｯﾄも後の段階でﾌﾟﾛｸﾞﾗﾑ(0)にできます。従って、既に書かれたことのあるﾊﾞｲﾄを、その間での消去なしで書くことは、新旧値間のﾋﾞｯﾄ単位ANDの結果になるでしょう。

ﾌﾗｯｼｭのﾊﾞｲﾄへの値$01書き込みはそのﾊﾞｲﾄ(8ﾌﾗｯｼｭ ｾﾙ)が最初に消去される必要があり、そしてそれはそのﾊﾞｲﾄを値$FFにします。値を書く(ﾌﾟﾛｸﾞﾗﾑする)ために上位7ﾋﾞｯﾄ(ﾌﾗｯｼｭ ｾﾙ)が放電されます。ﾌﾗｯｼｭが先行して消去されない場合、意図した値にﾌﾟﾛｸﾞﾗﾑできないかもしれません。そのﾊﾞｲﾄ値が$FEで、$01でﾌﾟﾛｸﾞﾗﾑされたとの仮定で、その結果はLSBが0から1に変更できないため、$00になるでしょう。

ﾌﾗｯｼｭ ﾒﾓﾘの耐久性

ﾊﾟﾗﾒｰﾀ(**注**)が単一ﾌﾗｯｼｭ ﾍﾟｰｼﾞ内で多数回格納できる事実はﾊﾟﾗﾒｰﾀ保存に適応するﾌﾗｯｼｭにします。ﾌﾗｯｼｭ ｾﾙは10,000消去/書き込み回数の保証された耐久性を持ちます。現在の各ｾﾙは10,000回の消去とﾌﾟﾛｸﾞﾗﾑができます。従って、ﾊﾟﾗﾒｰﾀが毎回違う位置を使用することによって1ﾍﾟｰｼﾞ内で10回書かれ得るなら、記憶(ﾍﾟｰｼﾞ)の耐久性は全体として100,000書き込み回数として見ることが出来ます。これはﾍﾟｰｼﾞ内一面にﾊﾟﾗﾒｰﾀを移動することによって、各ｾﾙが10,000回の消去と書き込みをされるだけだからです。

注: この文脈での用語"ﾊﾟﾗﾒｰﾀ"は単一ﾊﾟﾗﾒｰﾀとﾊﾟﾗﾒｰﾀ群の両方を網羅します。故に"ﾊﾟﾗﾒｰﾀ"は任意容量のﾃﾞｰﾀ群への参照で有り得ます。

書き込み時間

EEPROM書き込み時は1ﾊﾞｲﾄが同時に書かれます。ﾍﾟｰｼﾞ ﾌﾟﾛｸﾞﾗﾐﾝｸﾞが利用されるので、これはﾌﾗｯｼｭ使用時の場合ではありません。従ってﾊﾟﾗﾒｰﾀ容量が単一ﾊﾞｲﾄよりも大きい時にﾌﾗｯｼｭ記憶を使用するのが有利です。より大きなﾊﾟﾗﾒｰﾀは格納されるﾊﾞｲﾄ当たりでより少ない時間が使用されます。

long型のﾊﾟﾗﾒｰﾀ書き込みは概ね4msかかります(消去時間を除く、それは同じ時間です)。故にﾌﾗｯｼｭへのlong型ﾊﾟﾗﾒｰﾀ書き込みはﾊﾞｲﾄ当たり1msかかります。比較すれば、EEPROM(注)への同量ﾃﾞｰﾀ書き込みは32msかかります。EEPROMに関しては消去が含まれますが、ﾌﾗｯｼｭの消去はﾌﾗｯｼｭのﾍﾟｰｼﾞが使い古された時にだけ行われるので、この比較は公平です。以下、ATmega128でのﾌﾗｯｼｭ ﾊﾟﾗﾒｰﾀ記憶使用の考察です。このﾃﾞﾊﾞｲｽに対するﾍﾟｰｼﾞ容量は256ﾊﾞｲﾄです。1ﾍﾟｰｼﾞで256/4=64回のlong変数格納が可能です。従ってlong変数の格納に対して平均消去/書き込み時間を確定するために、消去時間は64書き込みに渡って平均化されるべきです。概ね4msの消去時間は64で割られた・・・大した値ではありません。

従って結論は書き込み時間に関してより大きなﾊﾟﾗﾒｰﾀ群がﾌﾗｯｼｭ記憶でより効率的なことです。故により多くの時間が貴重な電力を節約する休止形態で費やすことができます。

注: 例としてはATmega128使用で、EEPROMﾌﾟﾛｸﾞﾗﾐﾝｸﾞ時間はﾃﾞﾊﾞｲｽ依存です。これらの詳細については使用するﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄを参照してください。

ﾌﾗｯｼｭ ﾒﾓﾘ書き込みの束縛

ﾌﾗｯｼｭ ﾒﾓﾘとEEPROMはﾒﾓﾘの消去とﾌﾟﾛｸﾞﾗﾐﾝｸﾞに関する部署を共用します。束縛の結果はEEPROMとﾌﾗｯｼｭ ﾒﾓﾘが同時に書けない(または消去できない)ことです。これはﾌﾟﾛｸﾞﾗﾑ用ﾒﾓﾘを更新するために自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ機能を使用する前に、EEPROM書き込み周期が実行中かどうかを検査すべきであることを意味します。もしそうならば、自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞはEEPROM書き込みの終了を待たなければなりません。これは他の色々な方向にも適用されます。(例えば)EEPROM書き込みはﾌﾗｯｼｭ書き込みの終了を待たなければなりません。

ﾌﾗｯｼｭ ﾒﾓﾘのﾃﾞｰﾀ保持期間

ﾌﾟﾛｸﾞﾗﾑ可能なﾒﾓﾘが正しいﾃﾞｰﾀを保持できる期間はﾃﾞｰﾀ保持時間(または全体ﾃﾞｰﾀ保持力)として参照されます。何故ﾒﾓﾘが無限のﾃﾞｰﾀ保持力を持たないのかの理由はﾒﾓﾘ ｾﾙが漏洩することです。これはｾﾙ内に保存された電荷が時間経過で弱められ、或る時に電荷は持つべき論理ﾚﾍﾞﾙをもはや表しません。消去とﾌﾟﾛｸﾞﾗﾑの両方をされたｾﾙは予想できない状態へ向かって漏れます。ﾃﾞｰﾀ保持期間が過ぎると、ﾌﾗｯｼｭ ﾒﾓﾘの内容は信用できなくなります。

消去/書き込みによるﾃﾞｰﾀ保持力低下

ﾌﾗｯｼｭ ｾﾙの消去と書き込み時、それらは物理的に消耗させられます。読み込みはﾃﾞｰﾀ保持力に影響を及ぼしません。消去/書き込み回数が増えると、ｾﾙからの漏れが増えます。その結果は充電(電荷)がより早く弱められることで、従ってﾃﾞｰﾀがより直ぐに無効になることです。別の言葉では、ﾃﾞｰﾀ保持時間が減少されます。ｾﾙでの消去/書き込み回数が保証された10,000消去/書き込み回数を越える場合、ﾃﾞｰﾀ保持力は予測された20年未満に減ります。

ﾌﾗｯｼｭでのﾊﾟﾗﾒｰﾀ記憶との関連では、ﾌﾗｯｼｭ ﾍﾟｰｼﾞが他の各々から独立しているのを知ることが重要で、1つのﾌﾗｯｼｭ ﾍﾟｰｼﾞがﾊﾟﾗﾒｰﾀ記憶に使用される場合に、このﾍﾟｰｼﾞはﾃﾞｰﾀ保持力が減じられるかもしれない一方で、ﾌﾟﾛｸﾞﾗﾑ ｺｰﾄﾞに使用される残りのﾌﾗｯｼｭはﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞのﾃﾞｰﾀ保持力減少によって影響を及ぼされません。

電力損失の考察

どんな組み込みｼｽﾃﾑでも電力不足に晒され得ると仮定しなければなりません。これはﾊﾟﾗﾒｰﾀに対して不揮発性形式のﾒﾓﾘを使用する理由の1つで、そしてそれは電力OFF/ON後のﾊﾟﾗﾒｰﾀ回復が可能になるでしょう。

ﾊﾟﾗﾒｰﾀが正しく書かれたことを検証する多くの異なる方法があります。むしろ好ましい選択は検証を行うために利用可能な時間とｺｰﾄﾞ空間に依存します。非常に小さなﾒﾓﾘ空間と時間を使用する安全な方法は、ﾒﾓﾘの静的部分に書き込み完了ﾌﾗｸﾞを保持することです。このﾌﾗｸﾞは電力不足から回復する時に、最後の書き込みが正しく完了されたかを決めるのに使用できます。そうでなければ適切な処置が講じられ得ます。

ﾌﾗｯｼｭ ﾒﾓﾘ化けに関連する考慮はEEPROM化けに適用されるそれらと似ています(EEPROM化けを避けることに関してはAVRﾃﾞｰﾀｼｰﾄを参照してください)。従って、電力損失のためのﾌﾗｯｼｭ化けを避けるには、自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ機能使用時に低電圧(ﾌﾞﾗｳﾝｱｳﾄ)検出機能を常に許可することが推奨されます。

自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞの使用

自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ実行に使用する手順に関する詳細はﾃﾞﾊﾞｲｽのﾃﾞｰﾀｼｰﾄと「**AVR109:自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ**」で学べます。ﾌﾗｯｼｭ ﾒﾓﾘへの書き込み周期が外部ﾘｾｯﾄまたは低電圧検出(BOD:Brown Out Detector)ﾘｾｯﾄによって停止されないことに気付くのが重要です。電源ONﾘｾｯﾄだけが実行中のﾌﾗｯｼｭ ﾒﾓﾘ書き込み周期を停止します。結果は、例え供給電圧が最低推奨動作電圧未満に低下しても書き込み周期が可能な限り長く継続することです。これはﾃﾞｰﾀ化けの危険を誘引しません。この機能はﾌﾗｯｼｭ ﾍﾟｰｼﾞの消去/書き込みが完了される可能性を増します。

実装

ﾌﾗｯｼｭ ﾊﾟﾗﾒｰﾀ記憶例の実装はIAR EQAVR 2.27bｺﾝﾊﾟｲﾗを使用して行われています。この実装は他のｺﾝﾊﾟｲﾗへ移転できますが、これはIARｺﾝﾊﾟｲﾗから多くの固有関数がこのｺｰﾄﾞ例で利用されてしまっているためにいくつかの作業を必要とするかもしれません。このｺｰﾄﾞ例の狙いは、複数ﾊﾞｲﾄのﾊﾟﾗﾒｰﾀが保存でき、またこの記憶方法で高耐久性を得ることができるﾄﾞﾗｲﾊﾞを作ることでした。この記憶の耐久性はﾊﾟﾗﾒｰﾀ容量と使用するﾌﾗｯｼｭ ﾍﾟｰｼﾞ容量に依存します。このｺｰﾄﾞ例では7ﾊﾞｲﾄの構造体ﾊﾟﾗﾒｰﾀがATmega128のﾌﾗｯｼｭ ﾍﾟｰｼﾞ内に保存されます。このﾃﾞﾊﾞｲｽのﾍﾟｰｼﾞ容量は256ﾊﾞｲﾄです。1つのﾍﾟｰｼﾞは35までのﾊﾟﾗﾒｰﾀの複製に加えて書き込み完了ﾌﾗｸﾞを保持できます。ﾌﾗｯｼｭ記憶の保証耐久性は以下に依ります。

耐久性 = セルの耐久性×複製数 = 10,000×35 = 350,000回書き込み

最低保証耐久性よりむしろﾌﾗｯｼｭ上のﾃﾞｰﾀの代表的な耐久性を信用することで、最低耐久性の10倍程度まで(ﾊﾟﾗﾒｰﾀの数百万更新)を期待できます。

更に、ﾌﾗｯｼｭ記憶方法を使用する応用の総電力消費の最小化に焦点を当てます。これはﾌﾗｯｼｭ ﾒﾓﾘのRWW部にﾊﾟﾗﾒｰﾀを置くことによって成し遂げられます。これによってﾊﾟﾗﾒｰﾀが更新されている間にﾌﾗｯｼｭ ﾒﾓﾘのNRWW部からのｺｰﾄﾞ実行の継続、即ち時間節約が可能です。このｺｰﾄﾞは割り込みが動作を継続するように作られており、従ってRWW領域内の割り込み駆動ｺｰﾄﾞは未だ実行可能です。

任意選択

より一層高い信頼性の記憶方法を得るには、書き込み完了ﾌﾗｸﾞの使用を"許可"することで可能です。このようなﾌﾗｸﾞが使用される場合、ﾊﾟﾗﾒｰﾀ書き込みと(書き込み完了ﾌﾗｸﾞでもある)ﾊﾟﾗﾒｰﾀ位置ﾌﾗｸﾞは2つの独立した書き込み操作で行われるでしょう。書き込み完了ﾌﾗｸﾞがﾌﾟﾛｸﾞﾗﾑされている場合にﾊﾟﾗﾒｰﾀが正しく保存されていると決めることは、この方法で可能です。けれども、ﾊﾟﾗﾒｰﾀと書き込み完了ﾌﾗｸﾞの保存は1操作で両方を保存するのに比べて2倍の時間がかかるでしょう。

保存方法の強靭さを増すため、ﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞが消去される間、ﾊﾟﾗﾒｰﾀは一時的にEEPROMへ保存されます。これは消去動作と書き込み動作間の電力不足がﾊﾟﾗﾒｰﾀ損失に終わらないことを保証するために行われます。従って、消去中に電力不足が起こると、ﾊﾟﾗﾒｰﾀはEEPROMから回復され、そしてﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞにﾌﾟﾛｸﾞﾗﾑ(再消去/書き込み)されます。

これらの機能はFlashStorageDriver.cﾌｧｲﾙによって制御され、既定で許可されています。

必要条件

本例はATmega128を目的にしており、従ってこのﾃﾞﾊﾞｲｽのﾒﾓﾘ形態に適合されています。本例が他のﾃﾞﾊﾞｲｽで使用される場合、ﾘﾝｶ命令ﾌｧｲﾙとFlashStorageDriver.cﾌｧｲﾙ内のﾃﾞﾊﾞｲｽ依存情報は変更を必要とします。

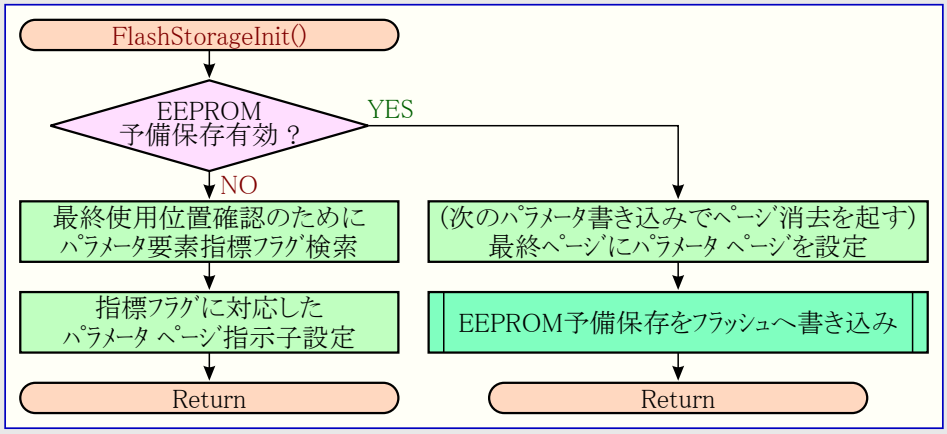
ﾌｧｰﾑｳｪｱ内容

このﾄﾞﾗｲﾊﾞは3つの関数から成ります。これらは流れ図(**図2.～4.**)と後続項によって記述されます。

FlashStorageInit

ﾌﾗｯｼｭ記憶の初期化はﾌﾗｯｼｭ記憶の状態の調査です。ﾊﾟﾗﾒｰﾀがEEPROM一時記憶に保存されている場合(EEPROM予備保存有効ﾌﾗｸﾞで決められる)、そのﾊﾟﾗﾒｰﾀはそれらから回復されてﾌﾗｯｼｭ記憶内に複写されます。さもなければ、ﾌﾗｯｼｭ緩衝部で最後に使用した位置が指標ﾌﾗｸﾞから確認されます。

図2. ﾌﾗｯｼｭ保存初期化処理

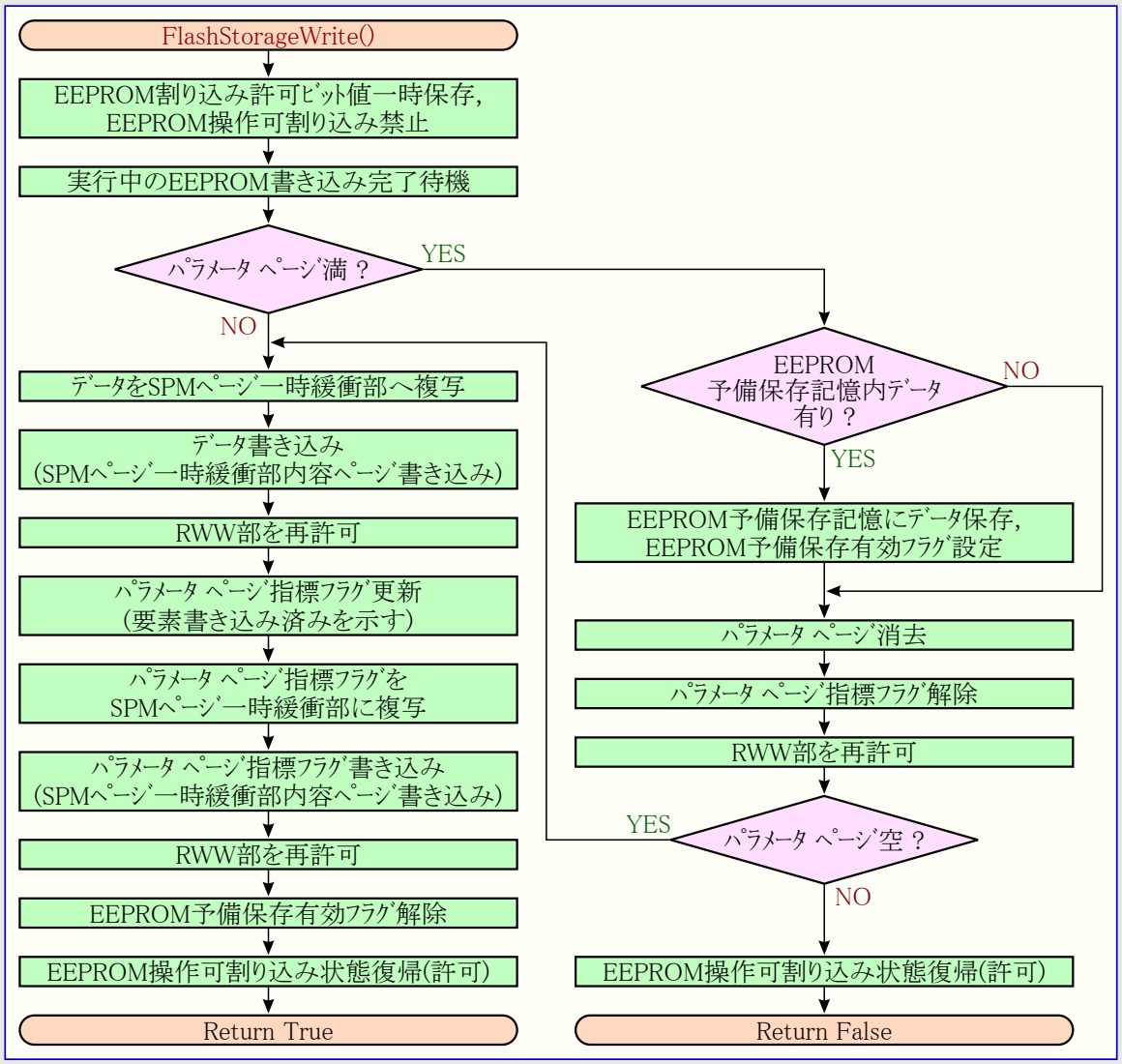


FlashStorageWrite

ﾊﾟﾗﾒｰﾀ保存ﾙｰﾁﾝは最初にEEPROM入出力が実行中かを調べます。実行中の場合、ﾙｰﾁﾝはEEPROM割り込みを禁止してその入出力の完了を待ちます。ﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞが一杯なら、初めにﾊﾟﾗﾒｰﾀがEEPROM内に保存されます。次にﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞが消去されます。EEPROMがﾊﾟﾗﾒｰﾀを保持している時にEEPROM予備保存有効ﾌﾗｸﾞが設定されます。一旦消去が完了されると、ﾊﾟﾗﾒｰﾀがﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞに書かれて、EEPROM予備保存有効ﾌﾗｸﾞが解除されます。この方法ではﾊﾟﾗﾒｰﾀ位置が有効なことを初期化中に決めることが可能です。最後にEEPROM割り込みが元の状態に復帰されます。

上で一覧される動作に加えて、関数から返る時にﾒﾓﾘ ｱｸｾｽが許可されるように、ﾒﾓﾘのRWW部のｱｸｾｽ制御が取り扱われます。

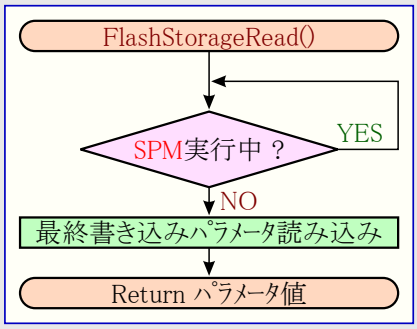
図3. ﾌﾗｯｼｭ保存書き込み処理



FlashStorageRead

ﾊﾟﾗﾒｰﾀの現在位置が応用で知れないため、応用に関してﾌﾗｯｼｭ ﾊﾟﾗﾒｰﾀを直接読むことは不可能です。故にﾊﾟﾗﾒｰﾀを読むために特別な関数が使用されます。この関数はﾊﾟﾗﾒｰﾀが読めるのとSPMが実行中でないことを検証し、そしてﾊﾟﾗﾒｰﾀ ﾍﾟｰｼﾞからのﾊﾟﾗﾒｰﾀを読んで返します。

図4. ﾌﾗｯｼｭ保存読み込み処理



文献一覧

1. 自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ付きﾃﾞﾊﾞｲｽに関するAVRﾃﾞｰﾀｼｰﾄ（ATMELのｳｪﾌﾞｻｲﾄで利用可能）
2. 「AVR109:自己ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ」応用記述（ATMELのｳｪﾌﾞｻｲﾄで利用可能）

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

製造拠点

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求
www.atmel.com/literature

本応用記述はATMELのAVR105応用記述(doc2546.pdf Rev.2546A-09/03)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。