

AVR105 : フラッシュ メリでの電力効率的な 高耐久性パラメータ記憶

要点

- パラメータの高速記憶
- 高耐久性フラッシュ メリ記憶 - 350,000書き込み回数
- 電力効率的なパラメータ記憶
- 任意容量のパラメータ
- 半冗長パラメータ記憶
- 書かれたパ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パ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われ、別のシステムでは、システム履歴や累積されたデータの記録に使われます。EEPROMがこれに使えますが、複数バイトが同時に格納される必要がある時にフラッシュ メリの速度と釣り合うことができません。

フラッシュ メリがより大きなパラメータ群に対してより効果的な理由は、ページプログラミングが使えることで、そしてそれはプログラミング時間を減らします。バイト当たりのプログラミング時間は複数バイトのパラメータ群格納時にEEPROMに対するよりもフラッシュ メ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の自己プログラミング機能を使うフラッシュ メリでの高耐久性パラメータ記憶法の実装方法を記述します。フラッシュ メリのページ全体と循環緩衝部に対して使わ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 メリのページ内の各単一メモリ位置は1つのメモリ位置が使わ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。記憶の耐久性はパラメータ群容量と記憶“緩衝部”に対して割り当てられたページ容量間の比率に比例します。

動作理論

megaAVR®系統は“自己プログラミング”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 メリの再プログラミングを可能にします。このプログ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 メリ内容が変更可能なので、パラメータの格納にも使えます。

けれども、パラメータ記憶にフラッシュ メリを使うことは、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。自己プログラミング機能はファームウェア更新に対して使わ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 メリのパラメータ更新に対して上手く使われることを可能にします。本項はパラメータ記憶に対してAVRの内部プログラム用メモリを使うために必要なフラッシュ メリについての基本的な情報を記述します。



8-bit **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. 2546A-09/03, 2546AJ4-03/21

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

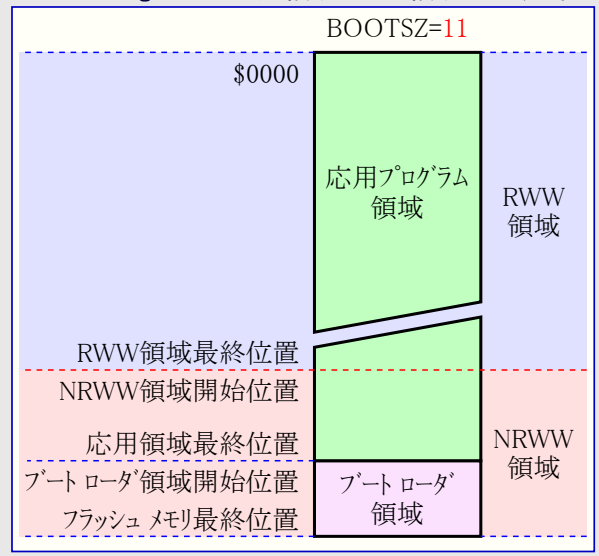
フラッシュ メモリはSPM命令を使って再プログラミングすることができます。SPM命令はこのメモリのブート領域からだけ実行できます。応用領域からのSPM命令実行は無効です。応用領域はアドレス\$0000からブート領域の始まりまでに配置されます(図1をご覧ください)。4つの異なるブート領域容量が選択できます。ブート領域容量はヒューズ設定によって決められます。選択できるブート容量は使うAVRに依存します。

ブ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)、選択したブ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。最大ブート領域容量が選択される時にブート領域は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、ブ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プログラミング付きデバイスのデ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用領域内のページ消去または書き込みの間、AVRがメモリのNRWW部に配置されたプログラムコードの実行を継続できることです。これはNRWW部内のページ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部内のページ変更の間、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ページがRWWメモリ内に配置されていると仮定して、ブート領域に配置されたコードは応用領域がプログラミング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 パラメータを更新するコード部分はブート領域に配置されなければならない、AVRがパラメータ更新間に動作を継続するなら、フラッシュ パ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。これはパラメータ書き込みの間に割り込みが防がなければならないかの実証が求められます。

図1. ATmega128の応用領域とブート領域のメモリ配置



フラッシュ メモリ セルの消去とプログラミング

フラッシュ メモリは各々が単一ビ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グ ゲートトランジスタ技術を基にし、トランジスタのゲート上での電荷“捕獲”がフラッシュ セルの読む論理レベルを決めます。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、ゲ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“プログラミング(0書き込み)”は論理レベルを0に持って来る、ゲート放電と当価です。消去(充電)されているセルのプログラミング(放電)だけが可能です。

フラッシュ メモリはページで配置されます。フラッシュの消去とプログラミングは自己プログラミング使用時、ページで行われます。フラッシュ消去はページ全体で実行され、フラッシュ書き込みはページ全体で実行されます。

ビットが個別にプログラミングできることに注意してください。プログラム(0)されるべきビットだけが放電されるため、残りの非プログラム(1)ビットは未だ充電されています。どの非プログラム(1)ビットも後の段階でプログラム(0)にできます。従って、既に書かれたことのあるバイトを、その間での消去なしで書くことは、新旧値間のビット単位ANDの結果になるでしょう。

フラッシュのバイトへの値\$01書き込みはそのバイト(8フラッシュセル)が最初に消去される必要があり、そしてそれはそのバイトを値\$FFにします。値を書く(プログラムする)ために上位7ビット(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、意図した値にプログラムできないかもしれません。そのバイト値が\$FEで、\$01でプログ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

パラメータ(注)が単一フラッシュ ページ内で多数回格納できる事実はパ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回の消去とプログラムができます。従って、パラメータが毎回違う位置を使うことによって1ページ内で10回書かれ得るなら、記憶(ページ)の耐久性は全体として100,000書き込み回数として見るができます。これはページ内一面にパ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“パラメータ”は単一パラメータとパラメータ群の両方を網羅します。故に“パラメータ”は任意容量のデ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バイトが同時に書かれます。ページプログラミングが利用されるので、これはフラッシュ使用時の場合ではありません。従ってパラメータ容量が単一バイトよりも大きい時にフラッシュ記憶を使うのが有利です。より大きなパラメータは格納されるバイト当たりでより少ない時間が使われます。

long型のパ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型パラメータ書き込みはバイト当たり1msかかります。比較すれば、EEPROM(注)への同量データ書き込みは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ページ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でのフラッシュパラメータ記憶使用の考察です。このデバイスに対するページ容量は256バイトです。1ページで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パ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プログラミング時間はデバイス依存です。これらの詳細については使うデバイスのデ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はメモリの消去とプログラミングに関する単位部を共用します。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。これはプログラム用メモリを更新するために自己プログラミング機能を使う前に、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、自己プログラミングは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データ保持期間

プログラム可能なメモリが正しいデータを保持できる期間はデータ保持時間(または全体データ保持力)として参照されます。何故メモリが無限のデ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ベルをもはや表しません。消去とプログ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。デ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/書き込みによるデータ保持力低下

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。読み込みはデ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、従ってデータがより直ぐに無効になることです。別の言葉では、デ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、データ保持力は予測された20年未満に減ります。

フラッシュでのパラメータ記憶との関連では、フラッシュページが他の各々から独立しているのを知ることが重要で、1つのフラッシュページがパラメータ記憶に使われる場合に、このページはデータ保持力が減じられるかもしれない一方で、プログラムコードに使われる残りのフラッシュはパラメータページのデ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。これはパラメータに対して不揮発性形式のメモリを使う理由の1つで、そしてそれは電力OFF/ON後のパラメータ回復が可能になるでしょう。

パ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ド空間に依存します。非常に小さなメモリ空間と時間を使う安全な方法は、メモリの静的部分に書き込み完了フラグを保持することです。このフラグ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えます。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デ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、自己プログラミング機能使用時に低電圧(ブラウンアウト)検出機能を常に許可することが推奨されます。

自己プログラミングの使用

自己プログラミング実行に使う手順に関する詳細はデバイスのデータシートと「AVR109:自己プログラミング」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。これはデータ化けの危険を誘引しません。この機能はフラッシュページの消去/書き込みが完了される可能性を増します。

実装

フラッシュパラメータ記憶例の実装はIAR EQAVR 2.27bコンパイラを使って行われています。この実装は他のコンパイラへ移転できますが、これはIARコンパイラから多くの固有関数がこのコード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。このコード例の狙いは、複数バイトのパ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ドライバを作ることでした。この記憶の耐久性はパラメータ容量と使うフラッシュページ容量に依存します。このコード例では7バイトの構造体パラメータがATmega128のフラッシュページ内に保存されます。このデバイスのページ容量は256バイトです。1つのページは35までのパラメータの複製に加えて書き込み完了フラグを保持できます。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デ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(パ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う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部にパラメータを置くことによって成し遂げられます。これによってパラメータが更新されている間にフラッシュメモリのNRWW部からのコード実行の継続、即ち時間節約が可能です。このコード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ド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、書き込み完了フラグの使用を“許可”することで可能です。このようなフラグが使われる場合、パラメータ書き込みと(書き込み完了フラグでもある)パラメータ位置フラグは2つの独立した書き込み操作で行われるでしょう。書き込み完了フラグがプログラムされている場合にパ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、パラメータと書き込み完了フラグの保存は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、パラメータページが消去される間、パ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パ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、パラメータはEEPROMから回復され、そしてパラメータページにプログラム(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

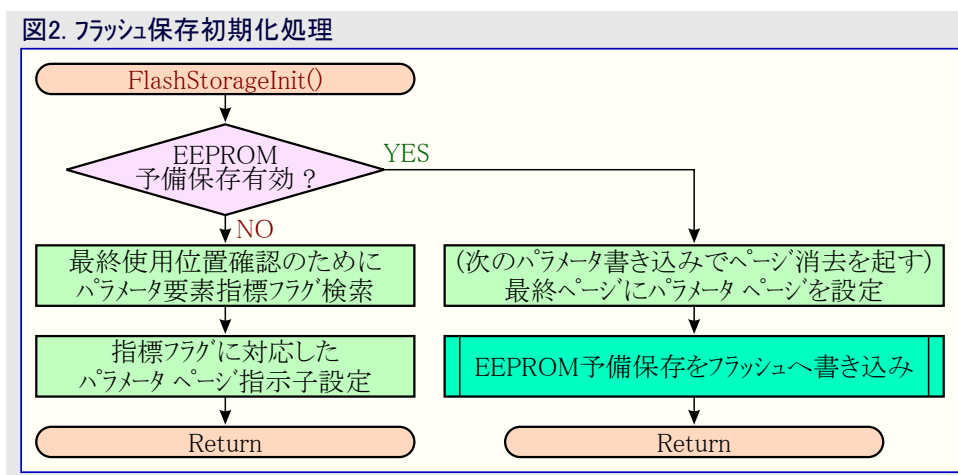
本例はATmega128を目的にしており、従ってこのデバイスのメモリ形態に適合されています。本例が他のデバイスで使われる場合、リンク命令ファイルとFlashStorageDriver.cファイル内のデバイス依存情報は変更を必要とします。

ファームウェア内容

このドライバは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。パラメータがEEPROM一時記憶に保存されている場合(EEPROM予備保存有効フラグで決められる)、そのパ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った位置が指標フラグから確認されます。

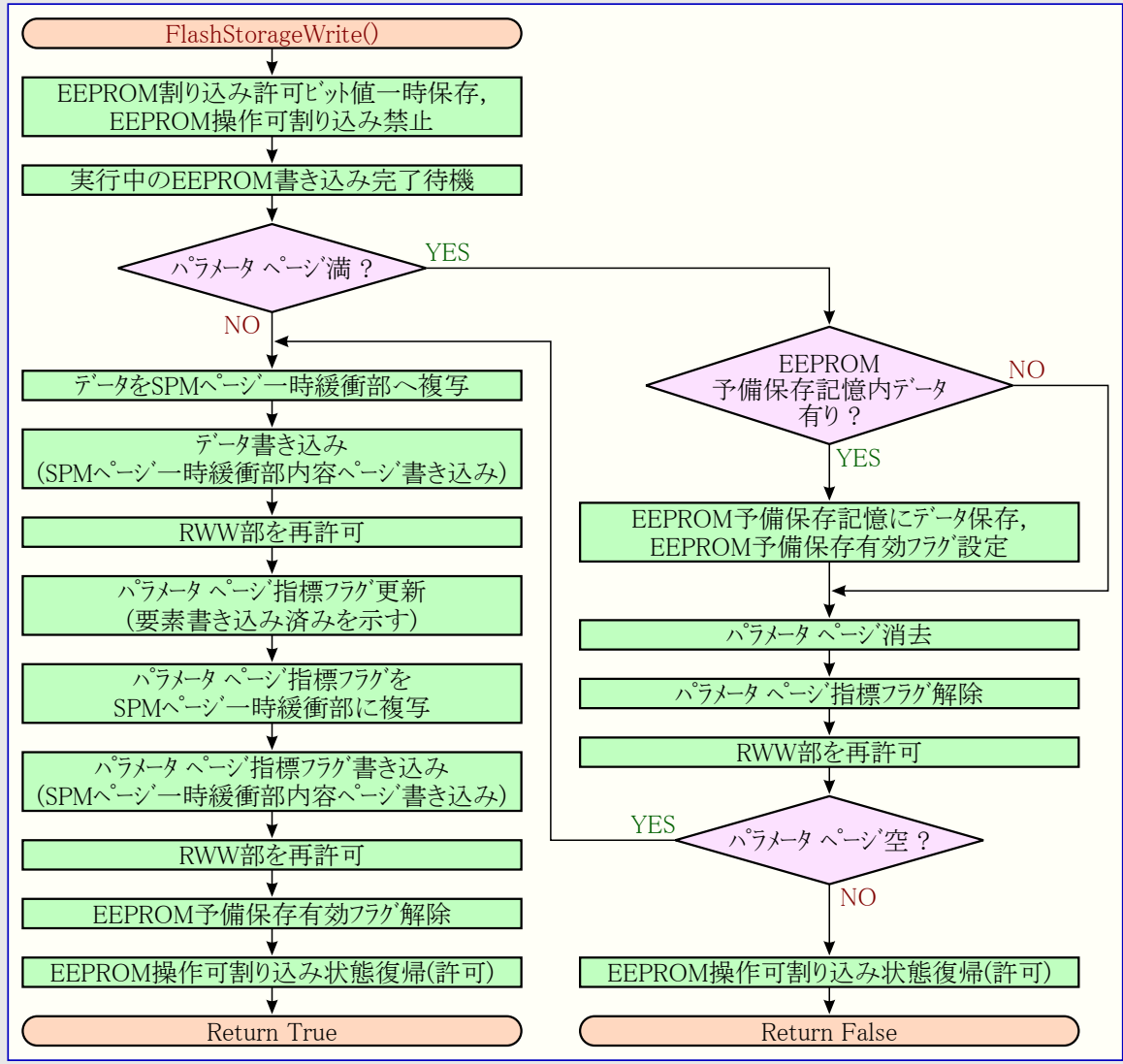


FlashStorageWrite

パ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。パラメータページ一杯なら、初めにパラメータがEEPROM内に保存されます。次にパラメータページが消去されます。EEPROMがパラメータを保持している時にEEPROM予備保存有効フラグが設定されます。一旦消去が完了されると、パラメータがパラメータページに書かれて、EEPROM予備保存有効フラグが解除されます。この方法ではパ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。

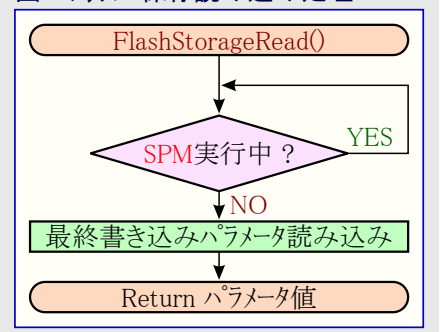
図3. フラッシュ保存書き込み処理



FlashStorageRead

パ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パラメータを直接読むことは不可能です。故にパラメータを読むために特別な関数が使われます。この関数はパラメータが読めるのとSPMが実行中でないことを検証し、そしてパラメータページからのパ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. 自己プログラミング付きデバイスに関するAVRデータシート (Atmelのウェブサイトで購入可能)
2. 「AVR109:自己プログラミング」応用記述 (Atmelのウェブサイトで購入可能)



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
東熱新川ビル 9F
アトメル ジャパン株式会社
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

© Atmel Corporation 2003.

Atmel製品は、ウェブサイト上にあるAtmelの定義、条件による標準保証で明示された内容以外の保証はありません。本製品は改良のため予告なく変更される場合があります。いかなる場合も、特許や知的技術のライセンスを与えるものではありません。Atmel製品は、生命維持装置の重要部品などのような使用を認めておりません。

本書中の®、™はAtmelの登録商標、商標です。
本書中の製品名などは、一般的に商標です。

© HERO 2021.

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。