

## AVR109 : 自己プログラミング

### 要点

- 自己プログラミング用C言語試供応用
- フラッシュメモリとEEPROMの両方の読み書き
- 施錠ビットの読み書き
- ヒューズビットの読み込み
- AVR911:オープンソース書き込み器での動作のために設計
- AVRProg適合
- 効率的なプログラミングに最適化された規約

### 序説

この応用記述はSPM(Store Program Memory)命令とでAVRを自己プログラミング用に構成設定することができる方法を記述します。試供応用はAVR911応用記述からのAVRオープンソース書き込み器(AVROSP)を実行するPCとUART経由で通信します。これは外部書き込み器なしでのフラッシュメモリとEEPROMのプログラミングを可能にします。コードはAVRProgとも互換です(AVRProgによって支援されるデバイスに対してのみ)。

ブートローダプログラムはフラッシュメモリのブート領域内に配置されます。このプログラムはホストPCとの通信を扱い、フラッシュメモリとEEPROM両方のプログラミングを容易にします。一度プログラミングすると、フラッシュメモリのブートと応用の両方の部分に異なるレベルの保護が個別に適用されます。従ってAVRは広範なメモリ保護段階を使用者に許す独特な柔軟性を提供します。

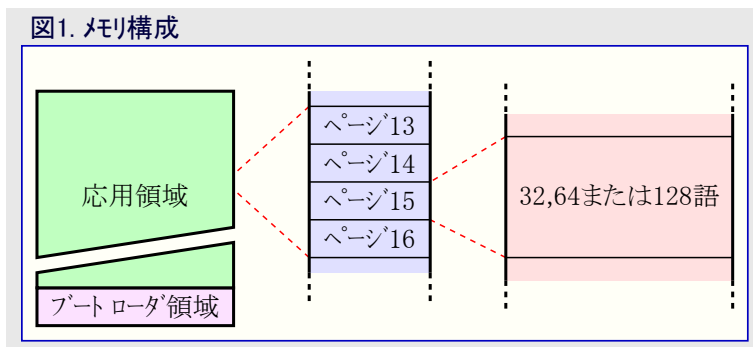
### SPMの説明

AVRの自己プログラミング能力のより良い理解を得るため、この機能の基本が以下で説明されます。

### メモリ構成

フラッシュメモリは1つの応用領域と1つのブートローダ領域の2つに分けられています。応用領域は応用の主コードを含み、一方ブートローダ領域は実際の自己プログラミング用コードを含みます。SPM命令はブートローダ領域からだけ実行できます(注:ブートローダ領域は通常の実用コード用にも使えます)。

フラッシュメモリは各々32,64,128語を含むページに分けられています。ページの使用は後で説明されます。応用領域とブートローダ領域の両方に及ぶメモリ全体がページに分けられています。例えば、8Kバイトのフラッシュメモリと32語(64バイト)のページ容量を持つデバイスは故に合計128ページを持ちます。メモリ構成は図1.で示されます。



ブートローダ領域の容量は2つのBOOTSZ<sub>n</sub>ヒューズを使って選べます。このヒューズは予め定義された4つの容量から1つを選びます。BOOTSZ<sub>n</sub>ヒューズは直列または並列プログラミングを使って変更することができます。詳細についてはデバイスのデータシートを参照してください。

ブートローダが実装される場合、呼び出し(CALL系命令)または分岐(JMP、BR<sub>xx</sub>系命令)を使って応用コードから直接的に、またはBOOTRSTヒューズのプログラム(0)により、このどちらでも呼び出せます。BOOTRSTヒューズがプログラム(0)されると、CPUはアドレス\$0000で開始する代わりにリセットでブートローダ領域での実行を開始します。BOOTRSTヒューズは直列または並列プログラミングを使って変更することができます。



8-bit **AVR**<sup>®</sup>  
マイクロコントローラ

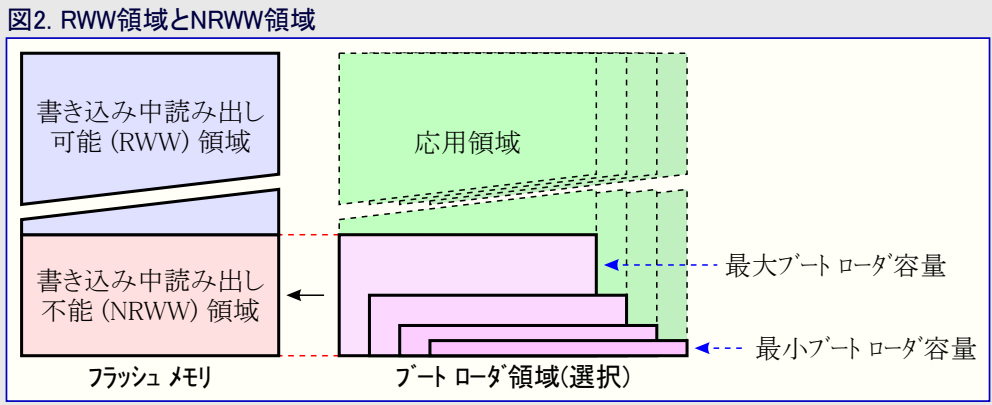
## 応用記述

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. 1644G-06/04, 1644GJ4-03/21

## 書き中に読める(Read-While-Write)能力

選択可能な応用領域とブートローダ領域間の分割に加えて、フラッシュメモリは2つの固定容量領域にも分けられます。最初の領域は書き中に読める(RWW:Read-While-Write)領域です。2つ目は書き中に読めない(NRWW:No-Read-While-Write)領域です。NRWW領域は常に選択可能なブートローダ領域の最大容量に等しく、故にブートローダ領域はNRWW領域の一部または全部を占有します。これは図2.で図解されます。



領域間の違いはNRWW領域がRWW領域を更新する間にアクセス可能なことです。RWW領域はそれが更新されている時にアクセスすることが不可能です。NRWWが更新される時(例えばブートローダコード自身を更新)、CPUは操作全体間で停止されます。別の言葉では、NRWW領域への書き込み間は読めませんが、RWW領域への書き込み間は読むことが可能です。詳細についてはデバイスのデータシートを参照してください。

この機能はRWW領域を更新する間に重要なコードの実行継続を可能にします。この重要なコードはNRWW領域内に含まれなければならないことに注意してください(ブートローダ領域である必要はありません)。より多くの情報については以降の[割り込みの項](#)をご覧ください。

ATmega163とATmega323のデバイスはNRWW領域とRWW領域を持たず、応用領域とブートローダ領域の選択可能な分割だけを持ちます。これらのデバイスでのフラッシュメモリへのどんな更新も、操作全体の間、CPUを停止します。

## SPM命令の使用

自己プログラミング操作はSPM命令を使って実行されます。動作はSPMCRLレジスタ(いくつかのデバイスではSPMCSR)を使って選ばれます。このレジスタは図3.で示されるように構成されています。

図3. SPMCRレジスタ

ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0
SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN

SPM機能使用時、SPM操作可(SPMEN)ビットは常にSPM命令を実行するのに先立つ4周期内に設定(1)されなければなりません。これは予期せぬフラッシュ更新の防止です。ソフトウェアはSPMENビット設定とSPM命令実行間に呼び出される割り込みルーチン、従って4周期制限を超えないことを保証しなければなりません。他の4つの強調表示されたビットは各種SPM機能を選択します。SPMENビットは操作完了時に機能ビットと共に自動的に解除(0)されます。

SPM機能は以下で記述されます。

### ページ消去

全てのフラッシュメモリ更新はページ単位で行われます。ページに新しいデータを書く前にページが消去されなければなりません。

Zレジスタは消去すべきページを選ぶのに使われます。消去されるべきページ内のバイトを指し示すようにZレジスタを設定してください。ページ内のバイトを選択する下位ビットは無視されます。例えば、32語(64バイト)のページ容量を持つデバイスでは、Zレジスタの下位6ビットが無視されます。

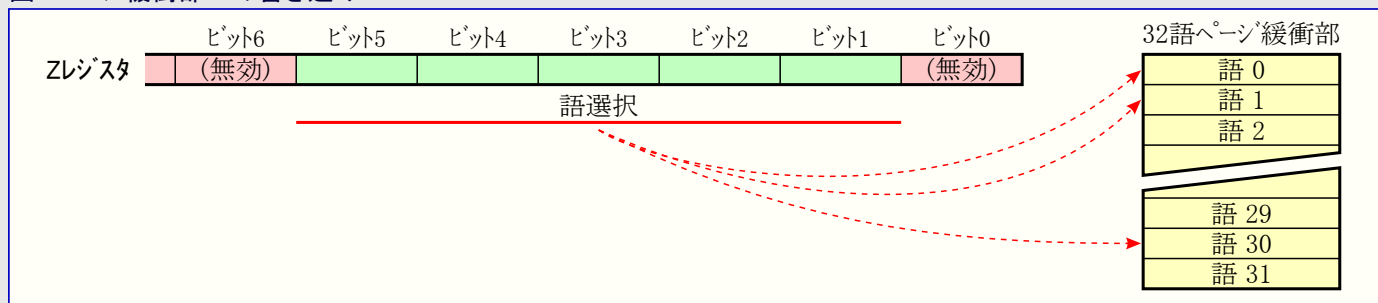
ページを消去するには、SPMCRレジスタでページ消去(PGERS)とSPMENビットを設定(1)し、SPM命令を実行してください。

## ページ緩衝部格納

ページに新しいデータを書くには、ページ緩衝部が先に満たされていなければなりません。ページ緩衝部は1つの一時的なページを保持する(SRAMではなく)独立した書き込み専用緩衝部です。この緩衝部は語単位で満たされなければなりません。緩衝部は或る1操作でフラッシュメモリに複写されます。

Zレジスタは緩衝部内に書かれるべき語を選択するのに使われます。常に1操作で語全体が書かれるために、Zレジスタの最下位ビット(LSB)は無視されます。従って単一バイトアクセスは不可能です。ページを選択するZレジスタの上位ビットはページ緩衝部への書き込み時に無視されます。32語(64バイト)ページに対するZレジスタビット構成は図4.で示されます。より大きなページ容量は語選択用ビットをもっと使います。

図4. ページ緩衝部への書き込み

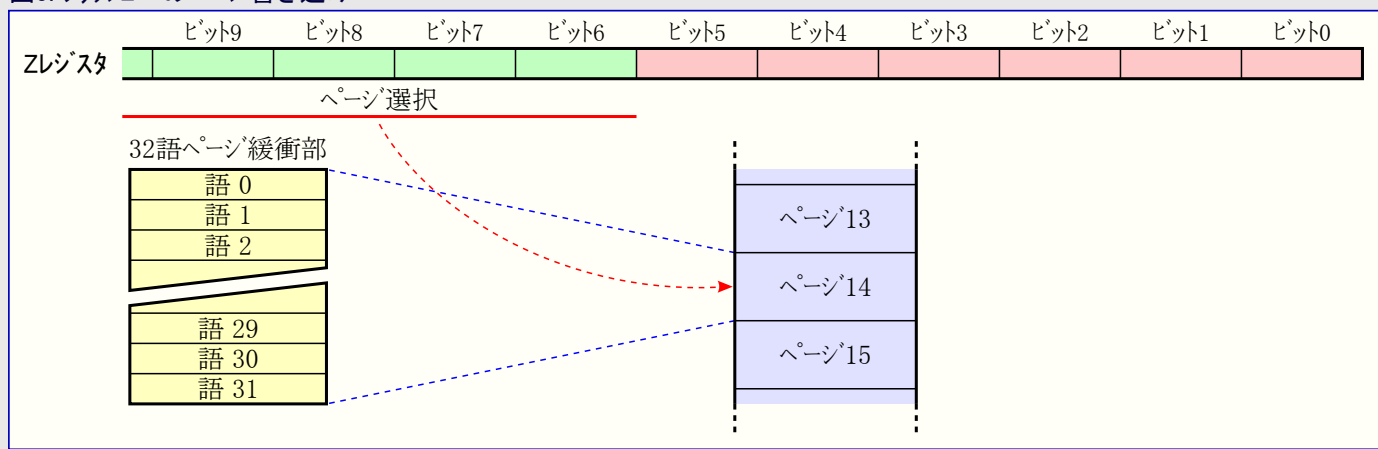


ページ緩衝部に語を書くために、R1:R0レジスタ内に語を格納してください。正しい語を指し示すようにZレジスタを設定し、SPMCRレジスタでSPMENビットだけを設定(1)してください。その後4周期内にSPM命令が実行されなければなりません。

## ページ書き込み

ページ緩衝部が新しいデータで格納されると、それはフラッシュメモリに書かれなければなりません。これを行うには、[ページ消去に関する項](#)で記述されたのと同じ方法でZレジスタを設定してください。そしてSPMCRレジスタでページ書き込み(PGWRT)とSPMENビットを設定(1)し、4周期内にSPM命令を実行してください。R1:R0レジスタの内容は無視されます。32語(64バイト)ページ書き込み用のZレジスタ使用は図5.で示されます。

図5. フラッシュへのページ書き込み



CPUが更なるページを更新するための準備を整える時を発見するのに、SPMENビットをポーリングすることができます。更新手順は割り込み制御にもできます。より多くの情報については以降の[割り込みの項](#)をご覧ください。

## RWW領域多忙フラグ

RWW領域でのページ消去またはページ書き込み実行時、この領域がアクセス不能であることを示すRWW領域多忙(RWWSB)フラグがハードウェアによって設定(1)されます。RWWSBフラグはSPM操作が完了される時にソフトウェアで解除されるべきです。これはSPMCRレジスタでRWW領域再許可(RWWSRE)とSPMENビットを設定(1)し、後続する4周期内のSPM命令によって行われます。代わりに、このフラグはページ緩衝部格納の開始によっても自動的に解除(0)されます。RWWSBフラグはRWW領域の現在のアクセス状況を調べるために応用の他の部分によって使うことができます。より多くの情報についてはデバイスのデータシートを参照してください。

ZレジスタとR1:R0レジスタの内容がRWWSRE機能使用時に無視されることに留意してください。

RWW領域が消去または書き込み後に再許可なしでアクセスされる場合、RWW領域の全てのアドレスが\$FFFFを読むことに注意してください。これはLPM命令を使ってフラッシュを読む時とRWW領域への呼び出しや分岐を実行する場合の両方に適用されます。再許可なしでのRWW領域への分岐実行の結果は、従ってプログラムカウンタが最初の実行可能コードに会うまでコード空間を通して“落ちる”ことを結局引き起こすでしょう。その場合、最初の実行可能コードはNRWW領域の最初のアドレスで出会うでしょう。

## ブート施錠ビット

応用領域とブートローダ領域は異なるレベルで保護できます。両領域に対して4つの保護レベルがあります。以下は保護種別の短い説明です。

表1. ブート施錠種別

種別	ブート施錠ビット	内容
1	1 1	完全な読み書きアクセス
2	1 0	書き込みアクセス禁止
3	0 0	別の領域からの読み込み(データまたは割り込み実行)と書き込み禁止
4	0 1	別の領域からの読み込み(データまたは割り込み実行)禁止

一旦プログラム(0)されると、直列または並列プログラミングなしで再びビットを非プログラム(1)にすることは不可能です。例えば、一度更新されるべき応用を実装するには、応用領域をブート施錠種別1に、ブートローダ領域を種別4に設定してください。これはブートローダをアクセスすることから応用を妨げ、同時に応用領域を更新するための完全なアクセスをブートローダに与えます。一旦更新されると、ブートローダは応用領域を種別3に設定し、従って全ての更なるアクセスを防止します。

ブート施錠ビットをプログラム(0)するには、正しいビットをR0レジスタに格納し、SPMCRでブート施錠ビット設定(BLBSET)とSPMENビットを設定(1)して4周期内にSPM命令を実行してください。Zレジスタの内容は無視されます。

SPM命令に代わるLPM命令の使用は、これらのビットを読みます。

## 割り込みの考慮

RWW領域書き込み間の割り込み使用は可能ですが、RWW領域への他の如何なるアクセスも防がなければなりません。別の言葉で、RWW領域更新間に行われるべき割り込み処理ルーチンは、割り込みベクタを含めてNRWW領域に配置されなければなりません。

MCUCR(一部のデバイスはGICR)レジスタの割り込みベクタ選択(IVSEL)ビットを使い、応用は2つの独立した割り込みベクタ表の実装に使うことができます。応用領域で1つと、RWW領域を更新する時に使われるべきブートローダ領域での1つです。これは重要な処理、例えば自己プログラミング中の安全監視の継続を応用に許します。割り込みとIVSELビットのより多くの詳細についてはデバイスのデータシートを参照してください。

2つ目の割り込みベクタが使われない場合、RWW領域更新中、割り込みは禁止されなければなりません。

## SPM割り込み

ATmega163とATmega323デバイスを除き、自己プログラミングを支援する全てのデバイスでは、割り込みを使ってフラッシュ更新操作を制御することが可能です。SPMCRレジスタでのSPM操作可割り込み許可(SPMIE)ビットの設定(1)がSPM操作可割り込みを許可します。これは現在のSPM操作が終了された時を示すのに使うことができます。

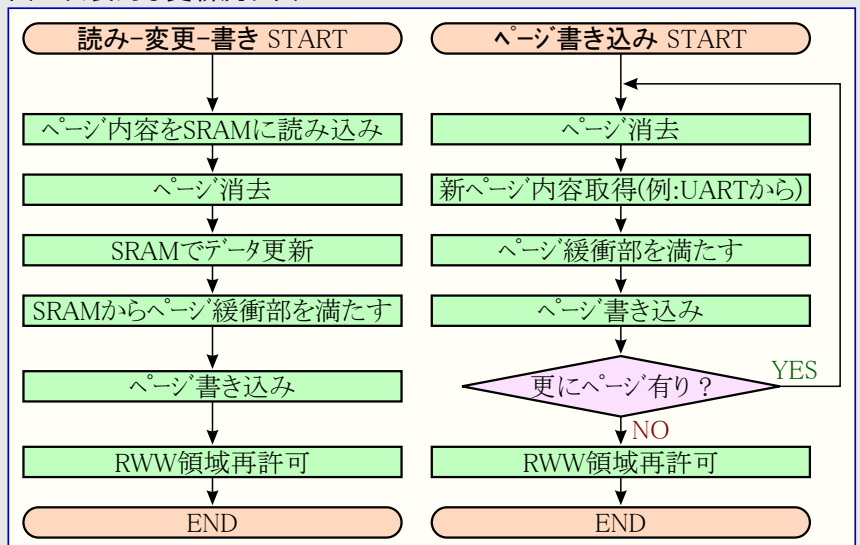
## EEPROM競合

SPM命令を実行する前に、EEPROMへの全ての書き込み操作が終了されていない(またその逆も)ことに注意してください。フラッシュメモリとEEPROMの消去/書き込みは同時に起き得ません。

## 代表的な更新手順

2つの共通する更新手順が図6.で示されます。左の流れ図はフラッシュの小さな部分、例えばフラッシュメモリに含まれる文字列定数を更新するのに使う読み-変更-書き操作を記述します。右の流れ図は直前の内容を読むことなくページ全体、例えばUARTからの受信データを書くのに使うページ書き込みを記述します。

図6. 代表的な更新流れ図



## ブートローダ例

本応用記述で表されたブートローダソフトウェアは使用者インターフェースとしてAVRオープンソース書き込み器(AVROSP)を使います。応用例は目的デバイスのフラッシュメモリとEEPROMを読むまたは更新するための機能を実装します。それはデバイスのヒューズビット読み込み、及び施錠ビットの読み込みと更新も可能です。

## AVRProg互換

AVRProg互換性はデバイス符号を当てにし、そしてそれは全てのデバイスに対しては定義されていません。preprocessor.xlsファイルは定義済みデバイス符号を含みますが、いくつかは未だAVRProgで実装されていません。その場合は同じメモリ容量のデバイスからの符号を使ってください。これに反してAVROSPは識票バイトだけを当てにし、自己プログラミング能力を持つ全てのデバイスを受け入れます。

## 規約

ブートローダプログラムで使われる規約はAVRProgに関して定義された規約のサブセットです。支援される命令の一覧は表2で示されます。全ての命令が単一文字で始まります。書き込み器は\$0D(キャリッジリターン、10進数の13)または、命令終了後の要求されたデータを返します。未知の命令は"?"で応答されます。

表2. AVRProg命令

命令内容	ホスト送信(命令)		ホスト受信(応答)	
	命令文字	データ	データ	肯定応答(終了子)
プログラミング動作へ移行	"P"			\$0D
アドレス自動進行	"a"		dd	
アドレス指定	"A"	aH aL		\$0D
フラッシュメモリ下位バイト書き込み	"c"	dd		\$0D
フラッシュメモリ上位バイト書き込み	"C"	dd		\$0D
ページ書き込み実行	"m"			\$0D
施錠ビット読み込み	"r"		dd	
フラッシュメモリ読み込み	"r"		dH dL	
EEPROM読み込み	"d"		dd	
EEPROM書き込み	"D"	dd		\$0D
チップ消去	"e"			\$0D
施錠ビット書き込み	"l"	dd		\$0D
ヒューズビット読み込み	"F"		dd	
上位ヒューズビット読み込み	"N"		dd	
拡張ヒューズビット読み込み	"Q"		dd	
プログラミング動作抜け出し	"L"			\$0D
デバイス形式選択	"I"	dd		\$0D
識票バイト読み込み	"s"		dd dd dd	
支援デバイス符号取得	"t"		dd ×n	
ソフトウェア識別子取得	"S"		7バイト文字列	
ソフトウェア版番号取得	"V"		dd dd	
書き込み器形式取得	"p"		dd	
LED点灯	"x"	dd		\$0D
LED消灯	"y"	dd		\$0D
ブートローダ抜け出し	"E"			\$0D
ブロック読み書き支援調査	"b"		"Y" cH cL	
フラッシュ塊書き込み	"B"	cH cL "F" dd ×n		\$0D
EEPROM塊書き込み	"B"	cH cL "E" dd ×n		\$0D
フラッシュ塊読み込み	"g"	aH aL "F"		
EEPROM塊読み込み	"g"	aH aL "E"		



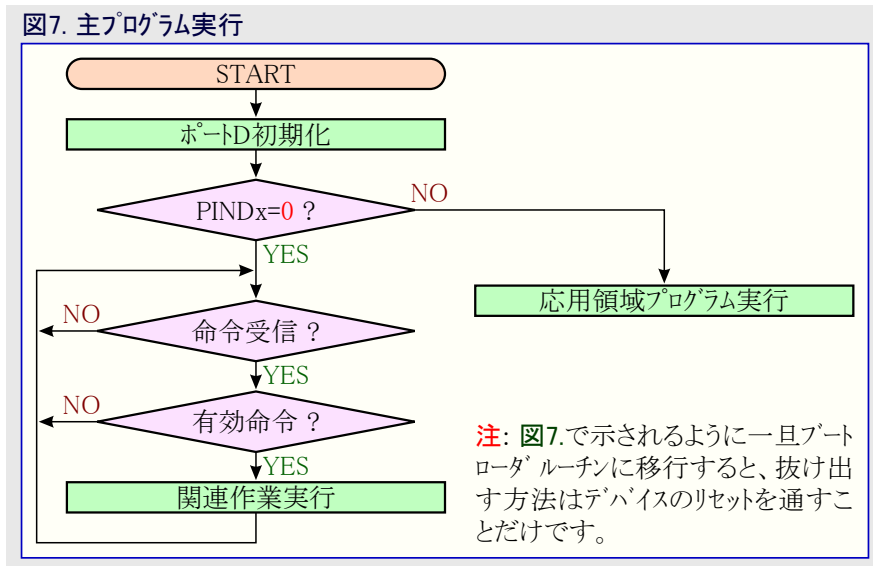
## プログラム内容

主プログラムはプログラミングが行われるべきか、または応用領域内のプログラムが実行されるべきかを調べることによって始まります。本応用ではこれがポートDの値によって指示されます。ポートDの使用者指定ピンがリセット間でLowに保持される場合、プログラムはプログラミング動作へ移行します。このピンがHighなら、(通常のリセットが起こる場合のように)プログラム実行は\$0000番地から始まります。

プログラミング動作ではプログラムがUART経由でAVROSPから命令を受信します。各命令は関連作業を実行します。ブートローダプログラムで認証されないどんな命令もAVROSPへ送り返される“?”に終わります。違うAVRとブートローダ容量に対してブートローダを変更する方法の例については、[preprocessor.xls](#) Microsoft Excelファイル内の段階的教えを参照してください。

### Main.c

`main.c`プログラムはホストPCとの通信を取り扱い、受信した命令を実行します。図7は操作を図解する流れ図を示します。



### Serial.c

UARTルーチン(`serial.c`)は単純にポーリングするUARTルーチンを実装します。先で記述されたように、このポーリングを行う理由は或るブート施錠ビット設定に対して割り込みがブート領域で許されていないためです。



## 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](http://www.atmel.com/literature)

## © Atmel Corporation 2004.

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AVR109応用記述(doc1644.pdf Rev.1644G-06/04)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。