

## AVR918 : Atmel Tinyプログラミング インターフェース(TPI)の使い方

Atmel 8ビット マイクロ コントローラ

## 要点

- TPI支援を持つデバイス用のTPIドライバ
- 以下に関するプログラミング例:
  - NVM読み書き
  - 施錠ビット読み書き
  - ヒューズビット読み書き
  - 校正ビット読み書き
  - デバイスID読み込み
  - チップ消去
- AVROSP(Atmel AVR911)適合

## 序説

Atmel®のTinyプログラミング インターフェース(TPI)は選ばれたAtmelの低位AVR®マイクロ コントローラでの特徴とされ、外部書き込み器にデバイスの不揮発性メモリ(NVM)のアクセスを許します。このインターフェースはデバイスの施錠ビット、プログラム用フラッシュメモリと識票、形態設定、校正領域へのアクセスを提供します。

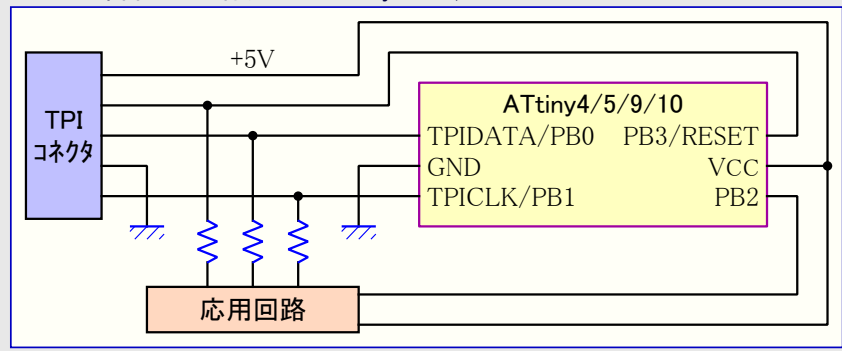
TPIは以下の3つのピン経由でアクセスすることができます。

RESET : TPI許可入力

TPICLK : TPIクロック入力

TPIDATA : TPIデータ入出力

図1. 外部書き込み器用Atmel Tinyプログラミング インターフェース



TPI規約のより多くの詳細については対応するデバイスのデータシートを参照してください。

この応用記述はTPIでデバイスをアクセスする、Atmel AVRマイクロ コントローラの使用方法を記述します。この設計はAtmelの「AVR911:AVR開放ソース書き込み器」応用記述で記述されるようなAVR開放ソース書き込み器(AVROSP)を使用します。

1. 不揮発性メモリ	3
1.1. 不揮発性メモリ施錠ビット	3
1.2. フラッシュメモリ	3
1.2.1. コード(プログラムメモリ)領域	3
1.2.2. 形態設定領域	3
1.2.3. 識票領域	3
1.2.4. 校正領域	3
2. TPI目的対象実装	3
2.1. TPIフレーム	3
2.2. TPI物理層	4
2.2.1. 直列データ受信	4
2.2.2. 直列データ送信	4
2.2.3. 方向変更	4
2.3. TPIアクセス層	4
3. 外部書き込み器実装	4
3.1. デバイス選択	4
3.2. 目的対象インターフェース	5
3.2.1. TPI許可	5
3.2.2. 命令1式	5
3.2.3. 外部プログラミング動作形態移行	5
3.2.4. 外部プログラミング動作形態抜け出し	5
3.2.5. NVMアクセス法	6
3.2.5.1. NVM指令	6
3.2.5.2. フラッシュのアドレス指定	6
3.2.5.3. フラッシュ読み込み	6
3.2.5.4. フラッシュ書き込み	7
3.2.5.5. チップ消去	7
3.2.5.6. 領域消去	7
4. 使用者インターフェース実装	7
4.1. 万能命令	8
4.2. 例: Atmel ATtiny10用コマンド行構文	8
5. 即時開始の手引き	9
6. 更新履歴	9

## 1. 不揮発性メモリ

本章はAtmelのTinyプログラミング インターフェース(TPI)経由でアクセスすることができる不揮発性メモリ (NVM)を記述します。本章で記述されるアクセス法はTPIを持つデバイスに特定され、他のAtmel AVRマイクロ コントローラに適用できないでしょう。

組み込まれたNVMは以下を持ちます。

- 不揮発性メモリ施錠ビット
- 4つの領域を持つフラッシュ メモリ

### 1.1. 不揮発性メモリ施錠ビット

施錠ビットはプログラム(0)される時に、デバイスに対して付加的な安全を提供します。施錠ビットはチップ消去によってのみ消去することができ、従って、他のNVM領域がプログラミングされた(書かれた)後でそれらをプログラミングする(書く)ことが必要とされます。既定により、施錠ビットは非プログラム(1)にされます。

### 1.2. フラッシュ メモリ

組み込まれたフラッシュ メモリは以下の4つの領域を持ちます。

- コード(プログラム メモリ)
- 形態設定
- 識票
- 校正

#### 1.2.1. コード(プログラム メモリ)領域

プログラム メモリは直接的にアクセスすることができないので、それはデータ メモリに割り当てられています。割り当てられたプログラム メモリはデータ メモリ内のバイト アドレス\$4000で始まります(更なる情報についてはデバイスのデータシートをご覧ください)。これはプログラム メモリ内のプログラムはアドレス\$0000から実行を開始されるけれども、データ メモリ経由でアクセスされる時は同じメモリ領域が\$4000から始まるアドレスで指定されることを意味します。

フラッシュ プログラム メモリへの内部的な書き込み操作は禁止され、従ってプログラム メモリはファームウェアに対して読み込み専用として現れます。フラッシュ メモリは外部書き込み器によってのみ書くことができます。

#### 1.2.2. 形態設定領域

形態設定バイトは形態設定領域に属します。以下の機能は、形態設定バイトに適切な値を書くことによって形態設定することができます。

- 低電圧検出(BOD:Brown-Out Detection)レベル
- ポートピンでのシステム クロック出力
- ウォッチドッグ タイマ ON
- 外部リセット禁止

形態設定ビット値変更はデバイスがプログラミング動作形態を去った後に実施されます。

利用可能な機能の更なる詳細については個別デバイスのデータシートを調べてください。

#### 1.2.3. 識票領域

識票領域はデバイスの識票のような情報を格納するのに使用されます。代表的に、デバイスの識票は3バイトから成ります。

識票値についてはデバイスのデータシートを調べてください。

#### 1.2.4. 校正領域

校正領域は代表的に内部発振器用の1つの校正バイトを含みます。このバイトはデバイス製造時に格納される校正値を含みます。校正領域は読み込み専用です。

## 2. TPI目的対象実装

AtmelのTinyプログラミング インターフェース(TPI)は物理層とアクセス層の2つの層から成ります。TPI物理層は送信と受信の2つの動作形態を支援します。既定により、物理層は受信動作形態で、開始ビットを待ちます。TPIアクセス層は動作形態を制御します。

### 2.1. TPIフレーム

TPI物理層は固定のフレーム形式を支援します。フレームは長さ8ビット、1開始ビット、偶数パリティビット、2停止ビットの1キャラクタから成ります。データは最下位ビットが先で転送されます。12ビット長のLowレベルの中断(BREAK)キャラクタが支援され、これは12ビットを超えて延長することもできます。

図2-1. 直列フレーム形式

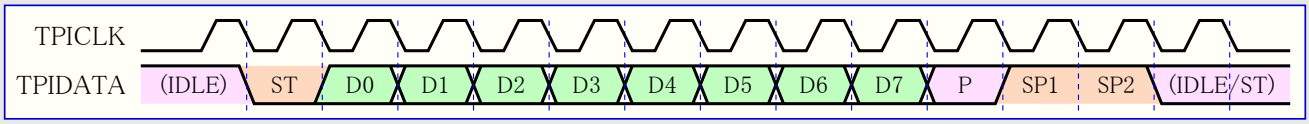


図2-1.で使用されるシンボルは次の通りです。

- ST : 開始ビット (常にLow)
- D0～D7 : データビット (最下位ビットが先に送出)
- P : パリティビット (偶数パリティを使用)
- SP1 : 停止ビット1 (常にhigh)
- SP2 : 停止ビット2 (常にhigh)

## 2.2. TPI物理層

TPI物理層は基本的な低位直列通信を処理します。物理層は双方向、半二重、直列の受信部と送信部を使用します。それは直列⇒並列と並列⇒直列のデータ変換、フレーム開始検出、フレーム誤り検出、パリティ誤り検出、パリティ生成、衝突検出を含みます。

TPI物理層は外部書き込み器によって提供されるTPICLK信号に同期して動きます。データは下降端で変更され、上昇端で採取されます。

### 2.2.1. 直列データ受信

受信動作形態では開始ビットが検出されてしまうと直ぐにデータ受信が開始されます。シフトレジスタに完全なフレームが存在すると、受信したデータはTPIアクセス層に対して利用可能です。

受信動作形態では、フレーム誤り、パリティ誤り、中断(BREAK)検出の3つの有り得る例外があります。これら全ての例外はTPIアクセス層に合図され、そしてそれは異常状態へ移行し、TPI物理層を受信動作形態に置いて中断キャラクタを待ちます。

### 2.2.2. 直列データ送信

送信は送信されるべきデータをシフトレジスタに設定することによって始められます。新しいデータでシフトレジスタが設定されると、送信部はTPICLKによって与えられる転送速度でTPIDATA線上に1つの完全なフレームをシフト出力します。

転送中に衝突が検出された場合、出力駆動部が禁止されます。TPIアクセス層は異常状態へ移行し、TPI物理層を受信動作形態に置いて中断キャラクタを待ちます。

### 2.2.3. 方向変更

方向変更中に正しいタイミングを保証するため、物理層に保護時間機構が実装されています。TPI物理層が受信から送信の動作形態へ変更する時に、開始ビットが送信されるのに先立って形態設定可能な追加のアイドルビット数が挿入されます。既定の保護時間は128ビットです。送受信動作形態間の最小遷移時間は2アイドルビットです。総アイドル時間は保護時間+2アイドルビットで指定されます。

外部書き込み器が受信動作形態から送信へ変更する時に、開始ビットが送信されるのに先立って最低1アイドルビットが挿入されるべきです。

## 2.3. TPIアクセス層

TPIアクセス層はTPI物理層上のキャラクタ転送方向を制御します。これは例外後の異常状態からの回復も扱います。TPIアクセス層はメッセージ形式で通信を扱います。各メッセージはオペランドが後続するバイトの大きさの命令から成ります。命令は常に書き込み器から送られ、一方オペランドは命令の形式に依存して、書き込み器またはデバイスのどちらかからできます。

AtmelのTinyプログラミングインターフェースの制御/状態空間(CSS:Control and Status Space)はTPIアクセス層の制御/状態レジスタに割り当てられます。アクセス層はアドレスポインタとしてポインタレジスタ(PR:Pointer Register)を用いて、直接的または間接的のどちらでもデータ空間にアクセスすることができます。

例外処理の更なる情報については個別デバイスのデータシートを調べてください。

## 3. 外部書き込み器実装

この応用記述で記述される書き込み器はTPIを実装するのにUSARTを使用します。USARTは同期動作形態で使用され、XCKがTPI CLKに使用されます。使用者インターフェースとしてAtmelの「AVR911:AVR開放ソース書き込み器」応用記述で記述されるAVROSPが使用され、書き込み器は別のUSART経由でAVROSP規約を実装します。

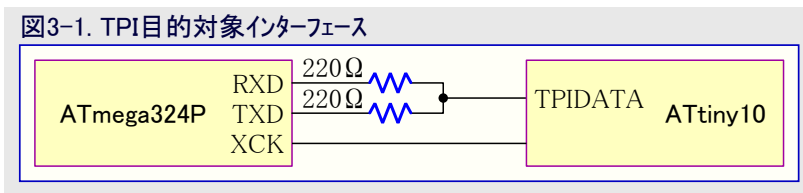
### 3.1. デバイス選択

この実装はAtmelのATmega324Pで試験されていますが、2つの内蔵USARTを持つAtmelのどのAVRデバイスでも実装のための最小の変更で使用することができます。外部書き込み器で使用されるデバイスは、TPI実装用の1つとAVROSP通信用の別の1つで、2つのUSARTが必要です。

デバイスは外部11.059MHzクリスタルによって駆動されます。

## 3.2. 目的対象インターフェース

TPICLKとしてXCKが使用され、RXDとTXDは図3-1.で示されるように2つの220Ω抵抗経由でTPIDATAに接続されています。

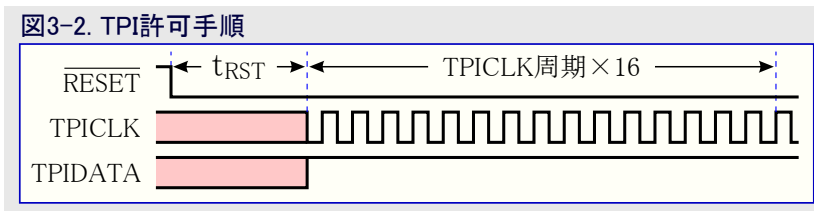


### 3.2.1. TPI許可

以下の手順がTPIを許可します(手引きについては図3-2.をご覧ください)。

- VCCとGND間に5Vを印加してください。
- t<sub>TOUT</sub>待ち、その後にRESETピンをLowに設定してください。これはデバイスをリセットしてTPI物理層を許可します。その後、RESETピンはプログラミング作業全体の間、Lowを保たなければなりません。
- t<sub>RST</sub>待ってください。
- 16 TPICLK周期間、TPIDATAピンをHighに保持してください。

注: t<sub>TOUT</sub>とt<sub>RST</sub>の値については個別デバイスのデータシートを調べてください。



### 3.2.2. 命令1式

TPIは表3-1.で示されるように、簡潔な命令1式を持ちます。これらの命令はNVM制御/状態空間とデータ空間をアクセスするのに使用することができます。外部書き込み器はこれらの命令を使用することによってNVM制御器とNVMメモリをアクセスすることができます。全ての命令は1バイトです。SKEYを除き、他の全ての命令は命令に続く1バイトのオペランドが必要です。SKEY命令は8バイトのデータが後続します。

表3-1. TPI命令1式

ニーモニック	オペランド	説明	動作
SLD	data,PR	間接アドレス指定を使用してデータ空間から直列取得	data←DS[PR]
	data,PR+	間接アドレス指定を使用してデータ空間から直列取得,事後増加	data←DS[PR],PR←PR+1
SST	PR,data	間接アドレス指定を使用してデータ空間へ直列格納	DS[PR]←data
	PR+,data	間接アドレス指定を使用してデータ空間へ直列格納,事後増加	DS[PR]←data,PR←PR+1
SSTPR	PR,a	直接アドレス指定を使用してポインタレジスタへ直列格納	PR[a]←data
SIN	data,a	データ空間から直列入力	I/O[a]←data
SOUT	a,data	データ空間へ直列出力	data←I/O[a]
SLDCS	data,a	直接アドレス指定を使用して制御/状態空間から直列取得	CSS[a]←data
SSTCS	a,data	直接アドレス指定を使用して制御/状態空間へ直列格納	data←CSS[a]
SKEY	Key,{8{data}}	鍵直列設定	Key←{8{data}}

### 3.2.3. 外部プログラミング動作形態移行

プログラミング動作形態に移行するためには、「3.2.1. TPI許可」項で記述される手順に従うことによってTPI通信を許可することが必要とされます。TPI許可後、

- TPI経由でSKEY命令を送ってください。
- TPI経由のバイト単位でNVMプログラミング許可データ(\$1289AB45CDD888FF)を送ってください。
- それが設定されるまでTPI状態レジスタ(TPISR)のNVM許可(NVMEN)ビットの状態をポーリングしてください。

### 3.2.4. 外部プログラミング動作形態抜け出し

プログラミング動作形態を抜け出すためには、以下の手順に従わなければなりません。

- TPI状態レジスタ(TPISR)のNVM許可(NVMEN)ビットを解除(0)してください。
- RESETを開放してください。

### 3.2.5. NVMアクセス法

NVMはデータメモリに割り当てられ、従って、データメモリ内でそれらが割り当てられた位置経由でアクセスすることができます。

プログラミング作業は不揮発性メモリ指令(NVMCMD)レジスタへの適切なNVM命令(表3-2.)の設定後に開始されます。不揮発性メモリ制御/状態レジスタ(NVMCSR)のNVM多忙(NVMBUSY)フラグはNVM制御器が指示された操作を実行する時に設定(1)されます。NVM指令レジスタはNVMBUSYフラグが活性(1)である限り、書き込みアクセスが妨げられます。これは新しい指令を始める前に現在の指令が完全に実行されることを保証するためです。

#### 3.2.5.1. NVM指令

NVM指令は操作に依存して、全般、領域、語の操作に分類することができます。全体では表3-2.で記述されるように、NVMをアクセスするために4つの指令があります。

表3-2. NVM指令

操作形式	NVMCMD	ニモニック	説明
全般	\$00	NO_OPERATION	無操作
	\$10	CHIP_ERASE	チップ消去
領域	\$14	SECTION_ERASE	領域消去
語 (注1)	\$1D	WORD_WRITE	語書き込み
2語 (注2)	\$1D	DWORD_WRITE	2語書き込み
フラッシュ語 (注3)	\$1D	CODE_WRITE	フラッシュ語書き込み

注1: Atmel ATtiny4/5/9/10

注2: Atmel ATtiny20

注3: Atmel ATtiny40

#### 3.2.5.2. フラッシュのアドレス指定

データ空間はバイトアドレス指定を使用しますが、フラッシュ領域が語としてアクセスされ、ページで構成されるので、データ空間のバイトアドレスはフラッシュ領域の語アドレスに変換されなければなりません。ページアドレスと語アドレスは共にフラッシュ内の語の絶対アドレスを形成します。データ空間アドレスの最上位側ビットはデータメモリに割り当てられたNVM施錠ビットまたはフラッシュ領域を選びます。フラッシュ領域アドレスの最下位ビットは語の上位または下位のバイトの選択に使用されます。

プログラミング中、データ空間位置はポインタレジスタ(PR)によって指示され、そしてこのアドレスはデータがアクセスされる前に格納されなければなりません。ポインタレジスタにアドレスを格納する手順は次の通りです。

- 0に設定したLSBを持つSSTPR命令を送ってください。
- アドレスの下位バイトを送ってください。
- 1に設定したLSBを持つSSTPR命令を送ってください。
- アドレスの上位バイトを送ってください。

#### 3.2.5.3. フラッシュ読み込み

フラッシュはデータメモリに割り当てられた位置から一度に1バイトずつ読むことができます。読み込み操作に関しては語アドレス内の上位または下位のバイトを選択するのに最下位ビット(ビット0)が使用されます。このビットが0ならば下位バイトが読まれ、それが1ならば上位バイトが読まれます。NVMからデータバイトを読むための手順は次の通りです。

- 事後増加付きSLD命令を送ってください。
- 受信のためにTPI制御レジスタ(TPICR)で設定された保護時間+2クロック周期のアイドル時間を待機してください。
- データバイトを受信してください。



### 3.2.5.4. フラッシュ書き込み

フラッシュは使用するデバイスに依存して、一度に語単位または複数語(2または4)単位で書くことができます。目的位置は書き込み操作の前に消去されなければなりません。さもなければ、フラッシュ語はその内容を不正にします。消去操作はその領域全体に対してだけ実行することができ、従って、書き込みを開始する前にフラッシュ領域を消去することが必要とされます。

先に下位バイトが一時緩衝部に書かれなければならない、その後の上位バイト書き込みがフラッシュ書き込み操作を起動します。

複数語書き込み中、全ての語の下位バイトと上位バイトはフラッシュ書き込み操作を開始するための正しい順序で書かれるべきです。

フラッシュ語書き込みの手順は次の通りです。

- 対応するメモリ書き込み指令を送ってください。
- 事後増加付きSST命令を送ってください。
- 書かれるべきデータの下位バイトを送ってください。
- 事後増加付きSST命令を送ってください。
- 書かれるべきデータの上位バイトを送ってください。
- デバイスが一度に1語よりも多くの書き込みを支援する場合、1つのアイドル キャラクタと書かれるべき必要とされる語数(2または4)の間、上の書き込み手順を繰り返してください。
- NVM多忙(NVMBUSY)フラグが解除(0)されるのを待ってください。

複数語書き込みを支援するデバイスで形態設定領域を書く時に、実際のデータを書いた後に形態設定領域へ偽装バイトが書かれなければならない。

### 3.2.5.5. チップ消去

チップ消去指令はフラッシュ メモリのコード領域全体とNVM施錠ビットを消去します。NVM施錠ビットはコード領域が完全に消去されてしまう前にリセットされません。形態設定、識票、校正の領域がチップ消去によって影響を及ぼされないことに注意してください。チップ消去を実行する手順は次の通りです。

- **CHIP\_ERASE指令**を送ってください。
- コード内側の位置にアドレスを設定してください。
- アドレス指定した位置の上位バイトに偽装バイトを書いてください。
- NVM多忙(NVMBUSY)フラグが解除(0)されるのを待ってください。

### 3.2.5.6. 領域消去

領域消去命令は選択したNVM領域を消去します。コード領域、NVM施錠ビット、形態設定の領域を消去することができます。校正領域は読み込み専用で、従ってそれに消去や書き込みの操作を実行することはできません。

- **SECTION\_ERASE指令**を送ってください。
- 選択した領域内側の位置にアドレスを設定してください。
- アドレス指定した位置の上位バイトに偽装バイトを書いてください。
- NVM多忙(NVMBUSY)フラグが解除(0)されるのを待ってください。

## 4. 使用者インターフェース実装

この応用記述で記述されるAtmelのTPI書き込み器はAtmel AVR開放ソース書き込み器(AVROSP)を支援します。AVROSPはAtmelのAVR Studio®に含まれるAtmelのAVRProgツールと等価です。それはAVR Studioでの他のコマンド行ツールと同じ構文を使用するコマンド行ツールです。使用者インターフェースがどう動くかのより多くの情報についてはAtmelの「[AVR911:AVR開放ソース書き込み器](#)」応用記述を参照してください。

必要条件は利用可能なavrosp.exeファイルを持つことだけです。通信ポート設定(ボーレート、パリティなど)はAVROSPを使用する前に手動で設定されなければなりません。例えば、COM1を115200bps、パリティなし、8データビットに設定するには、以下のDOSコマンドを走らせてください。

```
mode com1 baud=115200 parity=n data=8
```

支援されるAVROSP命令は表4-1.で記述されます。

表4-1. 支援されるAVROSP命令と活動

AVROSP命令	活動	返し
'T'	AVROSP支援用偽装読み込み実行	'Yr'
'P'	TPI許可	'Yr'
'L'	TPI禁止	'Yr'
'a'	自動進行支援	'Yr'
'A'	NVM語アドレス取得	'Yr'
'c'	書かれるべきデータの下位バイト取得	'Yr'
'C'	書かれるべきデータの上位バイト取得、VNM書き込み開始	'Yr'
'm'	1ページ書き込み終了、AVR911支援用	'Yr'
'R'	VNMから1語読み込み、AVROSPにデータを返す。	データ上位バイトとデータ下位バイト
'e'	チップ消去実行	'Yr'
'.'	万能命令の4バイト読み込み	'Yr'
'N'	上位ヒューズバイト読み込み、AVROSP用偽装書き込み支援実行	偽装バイト(\$FF)
'S'	書き込み器識別	"AVR ISP"
's'	デバイスID読み込み	上位先行で3バイトのデバイスID
'Y'	内部RC発振器校正 (注)	OSCCAL値

注: 「AVR057:Atmel ATtiny4/5/9/10/20/40用内部RC発振器校正」応用記述を参照してください。

### 4.1. 万能命令

AVROSPによって送られる万能命令はピリオド(.)で始まり、4バイトの命令が後続します。表4-2の詳細をご覧ください。

表4-2. 支援される4バイト命令と活動

4バイト命令	活動
\$38 \$00 \$xx \$00	OSCCAL読み込み
\$AC \$E0 \$00 データ	施錠バイト書き込み
\$58 \$00 \$00 \$00	施錠バイト読み込み
\$AC \$A0 \$00 データ	ヒューズ下位書き込み
\$AC \$A8 \$00 データ	ヒューズ上位書き込み
\$50 \$00 \$00 \$00	ヒューズ下位読み込み
\$58 \$08 \$00 \$00	ヒューズ上位読み込み

### 4.2. 例: Atmel ATtiny10用コマンド行構文

表4-3. ATtiny10用コマンド行構文例

操作	コマンド行構文
識別読み込み	avrosp.exe -dattiny10 -s
ヒューズバイト読み込み	avrosp.exe -dattiny10 -q
施錠バイト読み込み	avrosp.exe -dattiny10 -y
ヒューズバイト書き込み	avrosp.exe -dattiny10 -f'ヒューズバイト'
施錠バイト書き込み	avrosp.exe -dattiny10 -l'施錠バイト'
チップ消去	avrosp.exe -dattiny10 -e
OSCCAL読み込み	avrosp.exe -dattiny10 -0
'ファイル名.hex'にフラッシュ読み込み	avrosp.exe -dattiny10 -rf -of'ファイル名.hex'
'ファイル名.hex'からフラッシュ書き込み	avrosp.exe -dattiny10 -g -e -pf -if'ファイル名.hex'



## 5. 即時開始の手引き

以下はどうか開始するかの段階的な手引きとして意図されています。

- 4頁の「3. 外部書き込み器実装」で記述されるようなハードウェア設定に従ってください。
- AtmelのAVR918用のソースコードをダウンロードして解凍してください。
- `avr918\AVR918.atsln`を開いてください。
- Open Project⇒AVR918 properties⇒Toolchain⇒AVR/GNU C Compiler⇒Symboles⇒'ATTINY10'をダブルクリックし、目的対象としてATtiny20またはATtiny40が使用される場合はこのシンボルを変更してください。
- 全保存(Save all)、(Build⇒Clean Solutionを使用して)一掃、(Build⇒Rebuild Solutionを使用して)再構築を行ってください。
- AtmelのATmega324Pに書いてください。
- コマンドプロンプトを動かして`avr918`フォルダへ行ってください。
- 8頁の「4.2. Atmel ATtiny10用コマンド行構文」で言及された何れかのコマンド行構文を入力してください。

## 6. 改訂履歴

資料改訂	日付	注釈
8373A	2011年2月	初版資料公開
8373B	2012年10月	新雛形。表4-1.:新AVROSP命令('Y')追加。5.:即時開始の手引き修正。



Enabling Unlimited Possibilities®

*Atmel Corporation*

1600 Technology Drive  
San Jose, CA 95110  
USA  
TEL (+1)(408) 441-0311  
FAX (+1)(408) 487-2600  
[www.atmel.com](http://www.atmel.com)

*Atmel Asia Limited*

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG  
TEL (+852) 2245-6100  
FAX (+852) 2722-1369

*Atmel Munich GmbH*

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY  
TEL (+49) 89-31970-0  
FAX (+49) 89-3194621

*Atmel Japan G.K.*

141-0032 東京都品川区  
大崎1-6-4  
新大崎勧業ビル 16F  
アトメル ジャパン合同会社  
TEL (+81)(3)-6417-0300  
FAX (+81)(3)-6417-0370

© 2012 Atmel Corporation. 全権利予約済 / 改訂:8373B-AVR-10/2012

Atmel®, Atmelロゴとそれらの組み合わせ、AVR®, AVR Studio®, Enabling Unlimited Possibilities®とその他はAtmel Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

**お断り:** 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイト位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえばAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© HERO 2014.

本応用記述はAtmelのAVR918応用記述(doc8373.pdf Rev.8373B-10/2012)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。