

AVR1612 : PDIプログラミングドライバ

要点

- 目的対象MCUとしてPDIプログラミング インターフェースを持つデバイス用のPDIドライバ
- PDI命令一式とタイミング
- ATAVR Xplain/AT90USB1287用のプログラミング応用例
 - ・ チップ消去
 - ・ 応用フラッシュ読み書き
 - ・ EEPROM読み書き
 - ・ ヒューズ読み書き
 - ・ 施錠ビット読み書き
 - ・ 目的対象デバイスID読み込み

1. 序説

プログラミングとデバッグ用インターフェース(PDI)はデバイスの外部プログラミングとチップ上デバッグ用のATMEL[®]専用のインターフェースです。

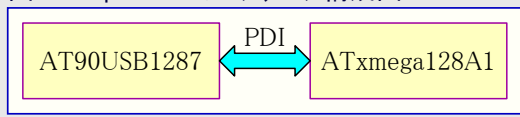
PDIは全不揮発性メモリ(NVM)空間、フラッシュ、EEPROM、ヒューズ、施錠ビット、使用者識別列の高速プログラミングを支援します。これはPDIインターフェースを通してNVM制御器をアクセスし、NVM制御器指令を実行することによって行われます。

PDIはクロック入力(PDI_CLK)用のリセットピンとデータ入出力用の専用(PDI_DATA)ピンを使用する2ピンインターフェースです。

この応用記述はPDIプログラミング実装方法を記述します。これはATMELのXplain評価基板に基づき、PDIプログラミングの規約とタイミングを明白にします。このキットでの開始方法については「AVR 1907:Xplainハードウェア使用者の手引き」応用記述も参照してください。

図1-1.はXplain PDIプログラミング構成図を示します。応用のファームウェアはAT90USB1287用ですが、USARTを特徴とする他のATMELのチップに簡単に移すことができます。

図1-1. Xplain PDIプログラミング構成図



2. PDI目的対象実装

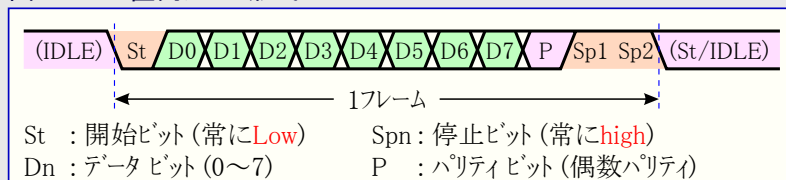
2.1. PDIフレーム形式

PDI物理層は標準的なUARTフレーム形式を使用します。直列フレームは開始と停止のビットと偶数パリティビットを持つ8つのデータビットの1文字になると定義されます。

書き込み衝突またはパリティ誤りの場合は中断文字(全データ、パリティ、停止ビットがLow設定)を送ることができます。中断文字についての更なる情報に関してはデータシートを参照してください。

図2-1.はPDI直列フレーム形式を記述します。

図2-1. PDI直列フレーム形式



8ビット 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。

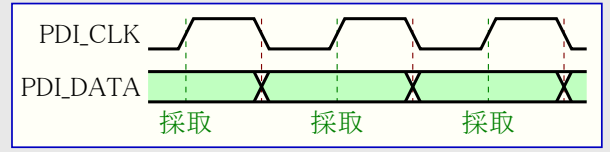
Rev. 8282A-11/10, 8282AJ3-11/14

2.2. 直列送受信

PDI物理層は送信(TX)または受信(RX)の動作形態のどちらかです。既定により、それは受信形態で、開始ビットを待ちます。

書き込み器とPDIは書き込み器によって提供されるPDI_CLKで同期して動きます。クロック端と、データ採取またはデータ変更間の依存性は固定化されます。図2-2.で図解されるように、(書き込み器またはPDIのどちらかからの)出力データは常にPDI_CLKの下降端で構成設定(変更)され、一方データは常にPDI_CLKの上昇端で採取されます。

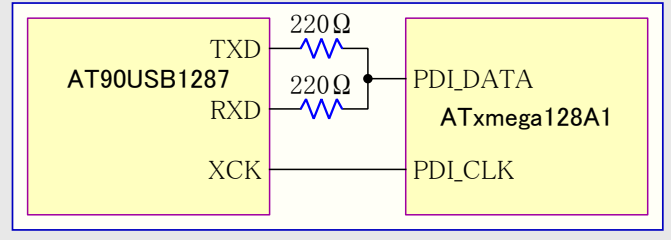
図2-2. データの変更と採取



2.3. AT90USB1287のUSARTとのハードウェア接続

ATMELのXplainキットはPDIポートのハードウェアを実装するのにATMELのAT90USB1287のUSARTを使用します。図2-3.はAT90USB1287とATMELのATxmega128A1間のPDIハードウェア接続を示します。

図2-3. AT90USB1287とATxmega128A1間のハードウェア接続



2.4. PDI命令一式

PDIはPDIそれ自身と内部インターフェースへの全てのアクセスに使用される賢い命令一式を持ちます。全ての命令はバイト命令です。命令の殆どは命令に後続する数バイトのオペラントが必要です。命令は外部書き込み器にPDI制御器、NVM制御器、NVMメモリへのアクセスを許します。図2-4.はPDI命令一式を示し、一方表2-1.は各命令についてより多くの詳細を示します。

図2-4. PDI命令一式要約

	指令				A量				D量			
LDS	0	0	0	0	x	x	x	x				
STS	0	1	0	0	x	x	x	x				
									P種別		A/D量	
LD	0	0	1	0	x	x	x	x				
ST	0	1	1	0	x	x	x	x				
									CSアドレス			
LDCS	1	0	0	0	x	x	x	x				
STCS	1	1	0	0	x	x	x	x				
									D量			
REPEAT	1	0	1	0	0	0	x	x				
KEY	1	1	1	0	0	0	0	0				

A量 (直接アドレス指定時アドレス長)			D量 (データ長)			P種別 (間接アドレス指定時)		
0 0	バイト		0 0	バイト		0 0	*(ptr) (ポインタ間接データ)	
0 1	語 (2バイト)		0 1	語 (2バイト)		0 1	*(ptr++) (上記+ポインタ進行)	
1 0	3バイト		1 0	3バイト		1 0	ptr (ポインタレジスタ)	
1 1	ロング (4バイト)		1 1	ロング (4バイト)		1 1	ptr++ (予約)	

CSアドレス (制御/状態レジスタ)											
0 0 0 0	レジスタ0	0 1 0 0	(予約)	1 0 0 0	(予約)	1 1 0 0	(予約)	0 0 0 0	(予約)	1 1 0 0	(予約)
0 0 0 1	レジスタ1	0 1 0 1	(予約)	1 0 0 1	(予約)	1 1 0 1	(予約)	0 0 0 1	(予約)	1 1 0 1	(予約)
0 0 1 0	レジスタ2	0 1 1 0	(予約)	1 0 1 0	(予約)	1 1 1 0	(予約)	0 0 1 0	(予約)	1 1 1 0	(予約)
0 0 1 1	(予約)	0 1 1 1	(予約)	1 0 1 1	(予約)	1 1 1 1	(予約)	0 0 1 1	(予約)	1 1 1 1	(予約)

表2-1. PDI命令一式

PDI命令	命令説明
LDS (\$00)	直接アドレス指定を用いてPDIBUSデータ空間からデータを取得
STS (\$40)	直接アドレス指定を用いてPDIBUSデータ空間へデータを格納
LD (\$20)	間接アドレス指定を用いてPDIBUSデータ空間からデータを取得
ST (\$60)	間接アドレス指定を用いてPDIBUSデータ空間へデータを格納
LDCS (\$80)	PDI制御/状態レジスタ空間からデータを取得
STCS (\$C0)	PDI制御/状態レジスタ空間へデータを格納
KEY (\$E0)	活性化鍵設定
REPEAT (\$A0)	命令繰り返し計数器設定

注: より多くの情報についてはXMEGA手引書の「プログラミングとデバッグ用インターフェース」章内の「PDI命令一式要約」を参照してください。

PDI命令使用例:

- PDI制御器のポインタ内に32ビット アドレス書き込み
擬似命令 : ST ptr \$12345678
PDIBUS16進符号: \$6B \$12 \$34 \$56 \$78
注: "\$6B"は"ST ptr"命令を表します。
- PDI制御器を通して*(ptr++)命令でアドレスへ値書き込み
擬似命令 : ST *(ptr++) \$FF
PDIBUS16進符号: \$67 \$FF
注: "\$67"は"ST *(ptr++)"命令を表します。
- PDI制御器内に繰り返し数書き込み(REPEAT命令はビッグエンディアン データを使用します。)
擬似命令 : REPEAT \$1234
PDIBUS16進符号: \$A1 \$34 \$12
注: "\$A1"は"REPEAT"命令と2バイトのデータ長を表します。
- 間接アドレス指定を用いてPDIBUSデータ空間からデータを取得
擬似命令 : LD *(ptr++)
PDIBUS16進符号: \$27
注: "\$27"は"LD *(ptr++)"命令を表します。

より多くの情報についてはXMEGA手引書の「命令一式要約」項を参照してください。

2.5. NVM指令

外部書き込み器からNVMメモリをアクセスするのに使用することができるNVM指令が表2-2.で一覧にされます。これは自己プログラミングに利用可能な指令の上位集合です。各指令はPDI命令を用いて目的対象のNVM指令(CMD)レジスタに設定されなければなりません。

外部書き込み器に関して、活動起動命令に対する起動は不揮発性メモリ(NVM)制御レジスタA(CTRLA)内の不揮発性メモリ指令実行(CMDEX)ビットを設定(1)することです。読み込み起動命令はPDI(PDI読み込み)から直接または間接の取得命令(LDSまたはLD)によって起動されます。書き込み起動命令はPDI(PDI書き込み)から直接または間接の取得命令(STSまたはST)によって起動されます。

表2-2. 外部プログラミングで利用可能なNVM指令

CMD6~0	指令/操作	起動源
\$00	無操作	-
\$40	チップ消去 (注1)	CMDEX
\$43	NVM読み込み	PDI読み込み
フラッシュ ページ緩衝部		
\$23	フラッシュ ページ緩衝部格納(設定)	PDI書き込み
\$26	フラッシュ ページ緩衝部消去	CMDEX
フラッシュ メモリ全体		
\$2B	フラッシュ ページ消去	PDI書き込み
\$2E	フラッシュ ページ書き込み	PDI書き込み
\$2F	フラッシュ ページ消去&書き込み	PDI書き込み
\$78	フラッシュ CRC	CMDEX
応用領域フラッシュ メモリ		
\$20	応用領域消去	PDI書き込み
\$22	応用領域ページ消去	PDI書き込み
\$24	応用領域ページ書き込み	PDI書き込み
\$25	応用領域ページ消去&書き込み	PDI書き込み
\$38	応用領域 CRC	CMDEX
ブートローダ領域フラッシュ メモリ		
\$68	ブートローダ領域消去	PDI書き込み
\$2A	ブートローダ領域ページ消去	PDI書き込み
\$2C	ブートローダ領域ページ書き込み	PDI書き込み
\$2D	ブートローダ領域ページ消去&書き込み	PDI書き込み
\$39	ブートローダ領域 CRC	CMDEX
校正と使用者識票領域		
\$01	使用者識票列読み込み	PDI読み込み
\$18	使用者識票列消去	PDI書き込み
\$1A	使用者識票列書き込み	PDI書き込み
\$02	校正列読み込み	PDI読み込み
ヒューズと施錠ビット		
\$07	ヒューズ読み込み	PDI読み込み
\$4C	ヒューズ書き込み	PDI書き込み
\$08	施錠ビット書き込み	PDI書き込み/CMDEX
EEPROM ページ緩衝部		
\$33	EEPROM ページ緩衝部格納(設定)	PDI書き込み
\$36	EEPROM ページ緩衝部消去	CMDEX
EEPROM		
\$30	EEPROM消去	PDI書き込み
\$32	EEPROMページ消去	PDI書き込み
\$34	EEPROMページ書き込み	PDI書き込み
\$35	EEPROMページ消去&書き込み	PDI書き込み
\$06	EEPROM読み込み	PDI読み込み

注: より多くの情報についてはXMEGA手引書の「外部プログラミング」項を参照してください。

注1: EESAVEヒューズがプログラム(0)されている場合、EEPROMはチップ消去の間、保護されます。

3. プログラミング インターフェース

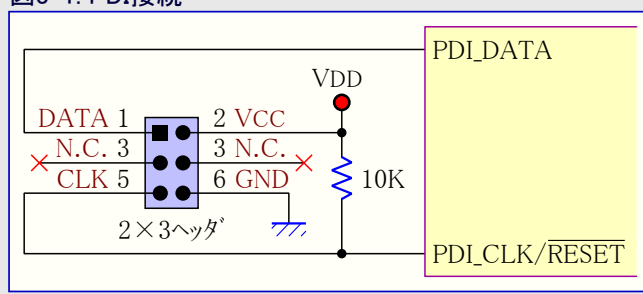
PDI低位ドライバ層はUSARTドライバに対して基本的なインターフェースを処理します。物理層は(USART形態でのUSARTとして)半二重双方向同期直列送信部と受信部を使用します。物理層はフレーム開始検出、フレーム異常検出、パリティ生成、パリティ誤り検出、衝突検出を含みます。PDIは2つのピンを通して入出力されます。表3-1はPDIインターフェース検査表を示します。

表3-1. PDIポート インターフェース検査表

信号名	推奨ピン接続	説明
PDI_CLK	リセットのプルアップは10kΩまたはより弱い、または全体的に取り去られるべきです。 PDIプログラミングとデバッグ用インターフェースが使用される場合、リセットのどの雑音分離(デカップ)コンデンサも取り去られるべきです。 この線を駆動する他の外部リセット元は切断されるべきです。 (クロック線上のどの負荷もデータビットに遅すぎる採取/生成と結果として通信失敗を引き起こすクロック端での遅延を生じ得ます。)	PDI_CLK: クロック入力/リセットピン (内部プルアップ)
PDI_DATA	プログラミング ヘッド/検査点にだけ接続してください。	PDI_DATA: PDIデータ入出力 (内部プルダウン)

これら2つのピンに加えて、VCCとGNDも外部書き込み器/デバッグとデバイス間で接続されなければなりません。図3-1は代表的な接続を示します。

図3-1. PDI接続



3.1. PDIプログラミング手順の概要

PDIプログラミングの手順が以下に示されます。

1. PDI部署を許可
2. 外部プログラミング動作形態移行
3. PDIでメモリ(フラッシュ メモリ/EEPROM/ヒューズ/施錠ビット)を読み書き
4. PDIプログラミング動作形態抜け出し

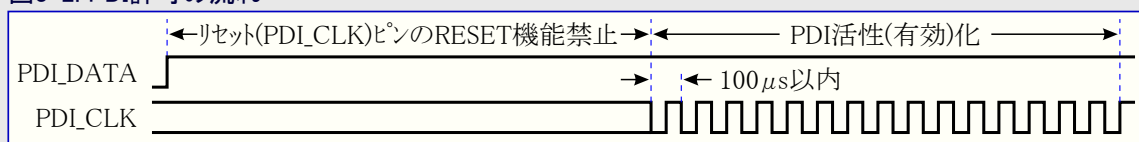
3.2. PDI部署許可

PDI物性はそれが使用され得る前に許可されなければなりません。これは最初に外部リセット最小パルス幅相当よりも長い時間PDI_DATA線をHighに強制することによって行われます(リセット特性についてはデバイスのデータシート参照)。

最初のPDI_CLK周期はRESETピンのリセット機能が禁止された後、100μsよりも遅れずに始まらなければなりません。これがその時に起こらなければ、RESETピンのリセット機能が再び自動的に許可され、この許可手順は再び全体を始めなければなりません。

この手順後、PDIが許可されて命令を受け取る準備が整います。この許可手順は図3-2で示されます。

図3-2. PDI許可の流れ



3.3. 外部プログラミング動作形態移行

外部書き込み器がPDI部署との通信を確立した後も、内部インターフェースは既定によってアクセス不能です。プログラミングのためにNVM制御器とNVMメモリへのアクセスを得るにはKEY命令を使用することによって固有の鍵で合図されなければなりません。内部インターフェースはPDIと内部インターフェース間の専用バス(PDIBUS)を用いて1つの直線状のアドレス空間としてアクセスされます。

PDI制御/状態レジスタ空間はSTCS(格納)とLDSCS(取得)の命令でアクセスすることができます。より多くの情報についてはXMEGA手引書の「レジスタ説明 - PDI制御/状態レジスタ」項と「命令一式要約」項を参照してください。

KEY命令を用いて送らなければならない鍵は64ビット長です。NVMプログラミングを許可する鍵は次の通りです。

\$1289AB45CDD888FF

外部プログラミング移行手順は次の通りです。

1. リセット識票(\$59)をPDIリセット(RESET)レジスタに設定
2. PDIに正しいNVM鍵を設定
3. 不揮発性メモリ許可(NVMEN)が設定(1)されるまでPDI状態(STATUS)レジスタのNVMENをポーリング

PDI STATUSレジスタのNVMENビットが設定されると、NVMインターフェースはPDIから活性(活動有効)です。

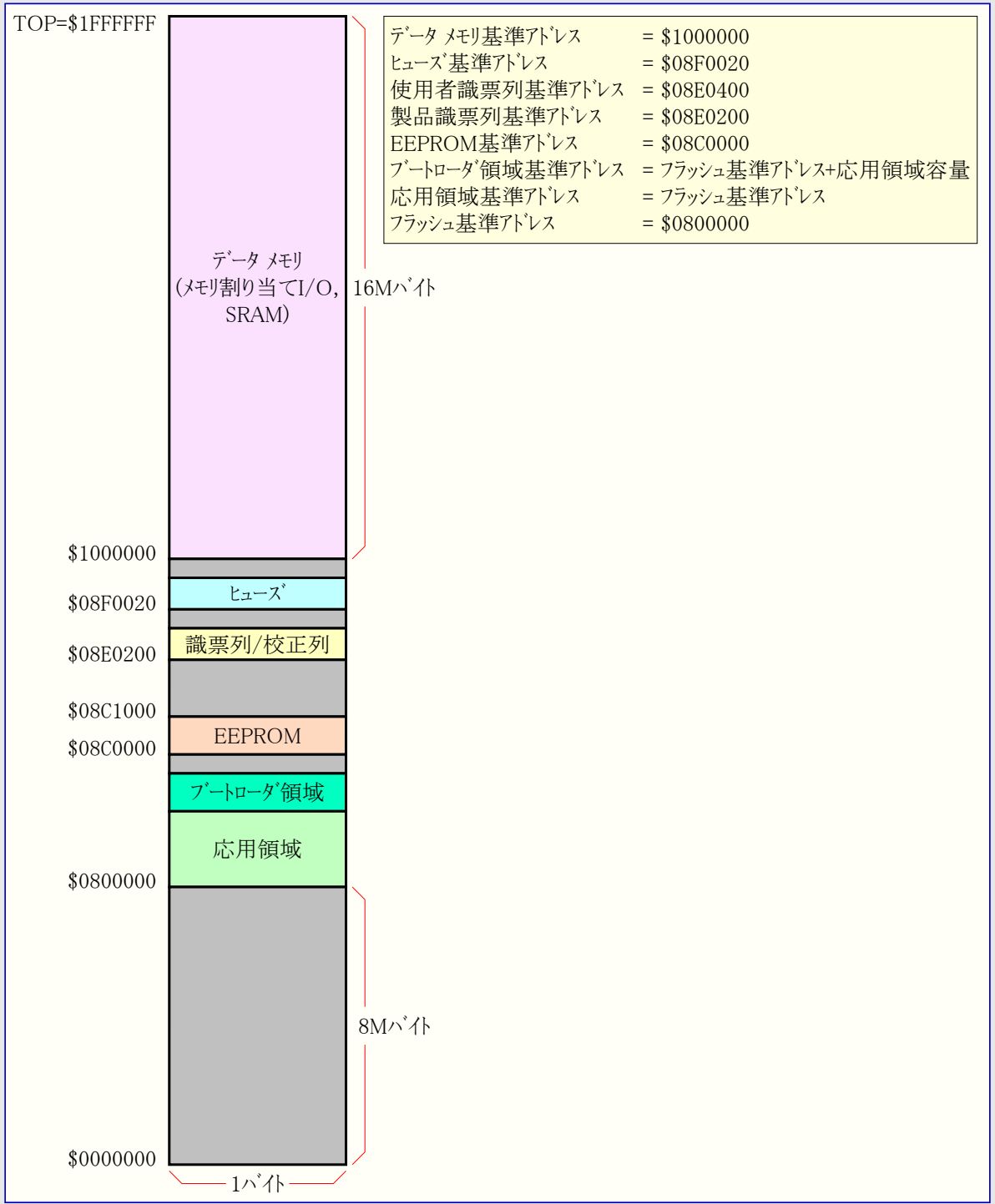
3.4. メモリプログラミング

本項は外部プログラミングでXMEGA®の不揮発性メモリ(NVM)のプログラミング方法を記述します。VNMはフラッシュプログラムメモリ、使用者識票、校正列、ヒューズと施錠ビット、それとEEPROMデータメモリから成ります。

外部プログラミングに関してデバイスはPDI物理接続を用いてPDIとPDI制御器を通してアクセスされます。PDIを通して、外部書き込み器はPDIバスを使用して全てのNVMメモリとNVM制御器をアクセスします。これを行う全てのデータとプログラム用のメモリ空間は直線状のPDIメモリ空間に割り付けられます。

図3-3はPDIメモリ空間とATxmega128A1での各々のメモリ空間に対する基準アドレスを示します。

図3-3. データとプログラムのメモリのPDIアクセス用メモリ割り当て



3.4.1. チップ消去

フラッシュプログラムメモリ、EEPROM、施錠ビットを消去するのにチップ消去指令が使用されます。使用者識票列、校正列、ヒューズは影響を及ぼされません。

チップ消去手順は次の通りです。

1. チップ消去指令(表2-2参照)をNVM指令(CMD)レジスタに設定してください。
2. NVM制御レジスタA(CTRLA)の指令実行(CMDEX)ビットを設定(1)してください。

注: チップ消去指令はPDI制御器とNVMを禁止します。間接的なPDI制御器の許可であるNVM許可(NVMEN)ビットが設定(1)されるまでこれをポーリングしてください。

3.4.2. フラッシュメモリとEEPROMのページ書き込み

フラッシュメモリとEEPROMのページ書き込みは最初に連携するページ緩衝部を満たしてその後にフラッシュメモリまたはEEPROM内の選択したページにページ緩衝部全体を書きます。

ページ緩衝部の大きさは各デバイスのフラッシュメモリとEEPROMの大きさに依存し、ページ容量とページ数の詳細は各デバイスのデータシートで記述されます。

3.4.2.1. フラッシュメモリとEEPROMの書き込み手順

フラッシュメモリまたはEEPROMのページ緩衝部内のデータをフラッシュメモリまたはEEPROMのページに書く前に、フラッシュメモリまたはEEPROMのページが消去されなければなりません。未消去のフラッシュメモリまたはEEPROMのページ書き込みはフラッシュメモリまたはEEPROMのページ内容を不正にします。

1. フラッシュメモリまたはEEPROMのページ緩衝部を消去してください。
2. フラッシュメモリまたはEEPROMのページ緩衝部を格納(設定)してください。
3. ページ消去&書き込みを実行してください。

3.4.2.2. ページ緩衝部消去

フラッシュページ緩衝部消去指令とEEPROMページ緩衝部消去指令はフラッシュメモリとEEPROMのページ緩衝部の消去に使用されます。

1. フラッシュ/EEPROMページ緩衝部消去指令をNVM指令(CMD)レジスタに設定してください。
2. NVM制御レジスタA(CTRLA)の指令実行(CMDEX)ビットを設定(1)してください。これは自己プログラミング中の時間制限CCP手順が必要です。

操作が完了されるまでNVM状態(STATUS)レジスタの多忙(BUSY)フラグが設定(1)されます。

3.4.2.3. ページ緩衝部格納(設定)

フラッシュページ緩衝部格納指令とEEPROMページ緩衝部格納指令はフラッシュメモリとEEPROMのページ緩衝部内に1バイトのデータを格納するのに使用されます。

1. フラッシュ/EEPROMページ緩衝部格納指令をNVM指令(CMD)レジスタに設定してください。
2. PDI書き込み操作を行うことによって選択したメモリアドレスを書いてください。

フラッシュページ緩衝部が語アクセスでPDIがバイトアドレス指定を用いるため、PDIは正しい順でフラッシュページ緩衝部を書かなければなりません。

3.4.2.4. ページ消去&書き込み

応用領域ページ消去&書き込み指令、ブートローダ領域ページ消去&書き込み指令、EEPROMページ消去&書き込み指令は1つの非分断操作で1つのページを消去してその後に設定したフラッシュ/EEPROMのページ緩衝部を選択したメモリ空間のページ内へ書くのに使用されます。

1. 応用領域/ブートローダ領域/使用者識票列/EEPROMのページ消去&書き込み指令をNVM指令(CMD)レジスタに設定してください。
2. PDI書き込みを行うことによって選択したページを書いてください。ページはページ内の何れかのバイト位置をアドレス指定することによって書かれます。

操作が終了されるまでNVM状態(STATUS)レジスタの多忙(BUSY)フラグが設定(1)されます。

3.4.3. NVM読み込み

フラッシュメモリ、EEPROM、ヒューズ、識票と校正列領域を読むのにNVM読み込み指令が使用されます。

1. NVM読み込み指令をNVM指令(CMD)レジスタに設定してください。
2. PDI読み込み操作を実行することによって選択したメモリアドレスを読んでください。

注: アドレスは各デバイスのデータシートで記述されるPDIメモリ空間アドレスです。

3.4.4. ヒューズ/施錠ビット書き込み

ヒューズ書き込み指令と施錠ビット書き込み指令はヒューズと、もっとも安全な設定に施錠ビットを書くのに使用されます。

1. ヒューズ/施錠ビット書き込み指令をNVM指令(CMD)レジスタに設定してください。
2. PDI書き込み操作を行うことによって選択したヒューズまたは施錠ビットを書いてください。

指令が終了されるまでNVM状態(STATUS)レジスタの多忙(BUSY)フラグが設定(1)されます。施錠ビット書き込みに関しては施錠ビット書き込み指令も使用できます。

3.5. PDIプログラミング抜け出し

概ね100 μ s間、PDI_CLK線上の動きがない場合、PDIは自動的に禁止されます。その後PDI_CLKをHigh、PDI_DATAをLowに設定してください。

4. プログラミング手順の例

PDI部署を許可して外部プログラミング動作形態移行後、デバイスはメモリの読み書きの準備が整います。PDI初期化についてのより多くの情報に関しては2.章を参照してください。

■ (フラッシュメモリ、EEPROM、使用者識別票、ヒューズビットを含む)メモリ読み込み

1. "NVM読み込み指令(\$43)"をNVM制御器指令(CMD)レジスタへ書くのにSTS命令を使用してください。CMDレジスタのアドレスは\$01CA (NVM制御器の基準アドレスが\$01C0でCMDレジスタの変位アドレスは\$0A)です。より多くの情報についてはXMEGA手引書の「レジスタ要約 - NVM制御器」項と「周辺機能部署アドレス割り当て」項を参照してください。
2. 2.4項で記述される"ST ptr"命令でメモリアドレスを設定してください。
3. "REPEAT"命令で繰り返し計数器内にデータ長を設定してください。
4. PDI制御器に"LD *(ptr++)"命令を送ってください。
5. データ配給完了までPDIBUSを読むためにポーリングしてください。

■ フラッシュメモリの消去と書き込み

1. フラッシュページ緩衝部消去
 - i. アドレス\$00000000を設定するのに"ST ptr"命令を使用してください。
 - ii. NVM制御器に"フラッシュページ緩衝部消去"(\$26)指令を書いてください。
 - iii. "ST *(ptr++)"命令でフラッシュページ緩衝部を偽装で書いてください。
 - iv. NVM多忙(BUSY)フラグが解除(0)されるまでそれをポーリングしてください。
2. フラッシュページ緩衝部格納(設定)
 - i. NVM制御器に"フラッシュページ緩衝部格納"(\$23)指令を書いてください。
 - ii. 開始アドレスを設定するのに"ST ptr"命令を使用してください。
 - iii. "REPEAT"命令で繰り返し計数器内にデータ長を設定してください。
 - iv. PDI制御器に"ST *(ptr++)"命令を送ってください。
 - v. PDIBUSを通して書き込みデータを送ってください。
3. NVM制御器に"応用領域ページ消去&書き込み"(\$25)指令を書いてください。
4. 開始アドレスを設定するのに"ST ptr"命令を使用してください。
5. "ST *(ptr++)"命令でフラッシュページを偽装で書いてください。
6. NVM多忙(BUSY)フラグが解除(0)されるまでそれをポーリングしてください。

■ EEPROMの消去と書き込み

1. EEPROMページ緩衝部消去
 - i. アドレス\$00000000を設定するのに"ST ptr"命令を使用してください。
 - ii. NVM制御器に"EEPROMページ緩衝部消去"(\$36)指令を書いてください。
 - iii. "ST *(ptr++)"命令でEEPROMページ緩衝部を偽装で書いてください。
 - iv. NVM多忙(BUSY)フラグが解除(0)されるまでそれをポーリングしてください。
2. EEPROMページ緩衝部格納(設定)
 - i. NVM制御器に"EEPROMページ緩衝部格納"(\$33)指令を書いてください。
 - ii. 開始アドレスを設定するのに"ST ptr"命令を使用してください。
 - iii. "REPEAT"命令で繰り返し計数器内にデータ長を設定してください。
 - iv. PDI制御器に"ST *(ptr++)"命令を送ってください。
 - v. PDIBUSを通して書き込みデータを送ってください。
3. NVM制御器に"EEPROMページ消去&書き込み"(\$35)指令を書いてください。
4. 開始アドレスを設定するのに"ST ptr"命令を使用してください。
5. "ST *(ptr++)"命令でEEPROMページを偽装で書いてください。
6. NVM多忙(BUSY)フラグが解除(0)されるまでそれをポーリングしてください。

■ ヒューズ ビット書き込み

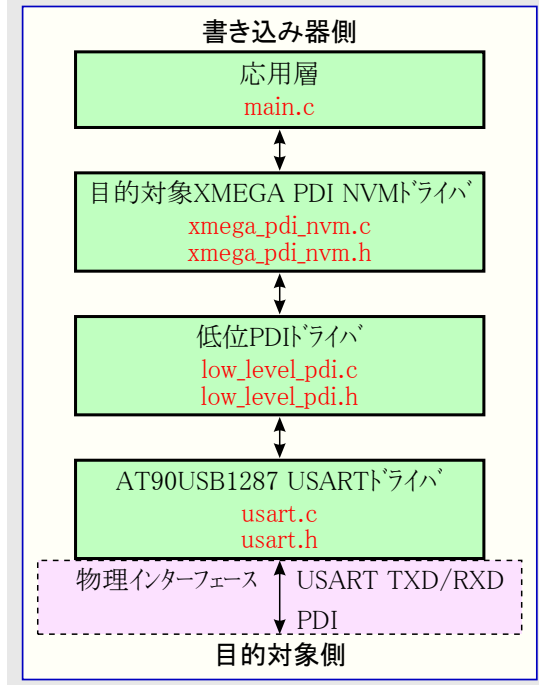
1. NVM制御器に“ヒューズ書き込み”(\$4C)指令を書いてください。
2. ヒューズ ビット値を設定するのに“STS ptr”命令を使用してください。
3. NVM多忙(BUSY)フラグが解除(0)されるまでそれをポーリングしてください。

5. コードの体系

この応用のコードは4つの層に分けることができます。図5-1はこの応用コードの体系を示します。

1. 応用層：各部署を統合。使用者コードはこの層の一員になります。
2. 高位目的対象XMEGA NVMドライバ、これは低位PDIドライバをインターフェースします。
3. 低位PDIドライバは書き込み器との通信のためにPDIインターフェース用の削減した命令一式と基本PDI操作用のUSARTドライバを使用します。
4. AT90USB1287 USARTドライバはPDI低位層から与えられた設定に基づいて物理的なUSART部署を構成設定する責任があります。

図5-1. 試供コードの単位部署図



6. 即時開始の手引き

本章はATMELのXplainキットを用いてATMELのATxmega128A1チップのプログラミングを始める方法の段階的な個人指導として意図されています。

注: AT90USB1287のヒューズは8MHzでデバイスをクロック駆動するように設定することが必要で、これはCKDIV8ヒューズがチェックされている場合にチェックを外す必要があることを意味します。

1. AVR1612用ソースコードをダウンロードして解凍してください。
2. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725からATMELのAVR Studio® 4.18(またはそれ以降)を、<http://sourceforge.net/projects/winavr/files/>からWINAVR-20090313(またはそれ以降)をダウンロードしてインストールしてください。
3. AVR1612のコードを特定の位置(どの位置も使用できますが、ここでは“C:\xplain_pdi_prog”を使用)に抽出してください。
4. GCCプロジェクトファイルまたはIAR™プロジェクトファイルを開いてソースコードをコンパイルしてください。
5. Xplain基板に電力を提供するためにATMEL XplainのUSBケーブルを接続してください。
6. ATMELのXplain上のJTAG USBヘッダに書き込み器(JTAICEm II または同類)を接続してください。
7. ATMELのAVR Studioを開始してTools⇒Program AVR⇒Connect⇒そして正しい書き込み器とポートを選んでください。コンパイルしたHEXファイルでATMELのAT90USB1287デバイスを書いてください。
8. プログラミング インターフェースを抜け出してください。そして基板のリセットが必要ならばXplain USBケーブルを切断して再接続してください。あらゆる物が動いているなら、ATMELのXplain上のLEDが今や点滅すべきで、これはATMELのATxmega128A1目的対象デバイスがATMELのAT90USB1287のフラッシュ内に格納された試験プログラムでプログラミングされて(書かれて)いることを示します。



Atmel Corporation

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

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

© 2010 Atmel Corporation. 全権利予約済

ATMEL®、ロゴとそれらの組み合わせ、それとAVR®、AVR Studio®、XMEGA®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のウェブサイトに表示する販売の条件と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AVR1612応用記述(doc8282.pdf Rev.8282A-11/10)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。