

AVR1927 : XMEGA-A1 Xplain開始の手引き

要点

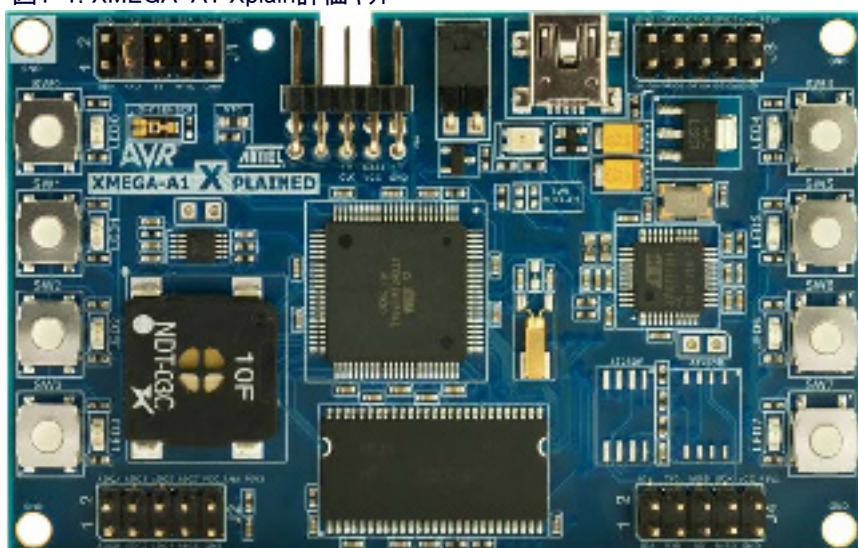
- USBケーブルと書き込まれたブート ロードだけの容易な再書き込み
- JTAGに基づくデバッグ/エミュレートでの容易なデバッグ コード
- ATMEL® STK600®, ATMEL AVR Dragon™, ATMEL AVR JTAGICEmk II, ATMEL AVR ONE!が使用可能

1. 序説

ATMELのXMEGA-A1 Xplain評価キットはATMELのATxmega128A1マイクロ コントローラを評価するハードウェア基盤です。

このキットは正しい方法でXMEGA周辺機能の使用を開始し、それら自身の設計でXMEGAデバイスと統合する方法の理解をATMELのAVR XMEGA使用者に許す特徴の範囲を提供します。この開始の手引きはXMEGA-A1 Xplain評価キットを使用し始めるのに必要とする基本段階の全てを簡単に片付けます。

図1-1. XMEGA-A1 Xplain評価キット



2. 何が開始に必要とされますか?

予め書き込まれたコードを走行し得るには、ATMELのXMEGA-A1 Xplain評価キットと、基板とPCを接続するためのUSBケーブル(標準A対ミニBまたはミニAB)が必要です。

XMEGA-A1 XplainをPCに接続する時に、直列通信ドライバをインストールするためにオペレーティングシステムはドライバ ファイルを要求します。このドライバ ファイルはこの資料と連携するzipファイルで入手可能です。このドライバ ファイルはWindows®XPとWindows 7の32ビットと64ビットの版を支援します。Linux®ではドライバのインストールが必要ありません。

例のコードを変更または新しいコードを書いてそれをコンパイルするには、コードの編集とデバッグのためのIDEとATMELのAVRマイクロ コントローラ用のツールチェーンが必要です。ATMELはATMELのAVR Studio® 4とATMELのAVR GNUツールチェーンの両方を提供します。IAR™のEmbedded Workbenchのような他のIDEとツールチェーンも利用可能です。

走行時にもコードのデバッグを望むなら、ATMELのAVR JTAGICEmk II やATMELのAVR ONE!のようなデバッグが必要です。これらのツールのより多くの情報はATMELのウェブサイトから入手可能です。より多くの詳細については本資料後部の「参照」項をご覧ください。

2.1. XMEGA-A1 Xplain用資料

「AVR1924:XMEGA-A1 Xplainハードウェア使用者の手引き」一括は、回路図、部品表、組み立て図、3D図、その他もつのような、キットについての詳細な情報を含みます。この一括はATMELのウェブサイトから入手可能です。

www.atmel.com/xplained



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. 8372A-04/11, 8372AJ1-03/14

2.2. AVRソフトウェア枠組み

AVRソフトウェア枠組み(ASF:AVR Software Framework)はATMELのAVRデバイス用の応用を構築するためのソフトウェアドライバとライブラリを提供します。それは開発を手助けしてソフトウェア設計の異なる部分と共にくっ付けるように設計されています。それはオペレーティングシステム(OS)内への容易な統合、または自立製品としての走行が行えます。

新しいAVRソフトウェア枠組み(2+版)は最終ソフトウェア更新を得るのに使用することができます。それは以下で得られます。

http://atmel.com/dyn/products/tools_card.asp?tool_id=4192&category_id=163&family_id=607&subfamily_id=2138

2.3. ATMEL AVRマイクロコントローラ用統合開発環境

2.3.1. AVR Studio 4

AVR Studio 4はATMELのAVR 8ビットMCU用の無料の統合開発環境です。これはコードを書いてマイクロコントローラをプログラミングしてデバッグするのに使用することができます。

ATMEL AVR Studio 4

<http://www.atmel.com/avrstudio>

2.3.2. ATMEL AVR用Embedded Workbench

IARのEmbedded Workbench®はプロジェクト管理ツールとエディタを持つ統合開発環境です。

ATMEL AVR用IAR Embedded Workbench

<http://www.iar.com/ewbsite1/1.0.1.0/107/1/>

2.4. FLIP(BatchISP)

BatchISPは基板制御器のフラッシュメモリとEEPROMをプログラミングするためのコマンド行ツールで、FLIPインストールの一部です。これは基板制御器上で予め書き込まれたブートローダとの通信に使用することができます。

FLIP(Flexible In-system Programmer)プログラミング用

http://atmel.com/dyn/products/tools_card.asp?tool_id=3886

2.5. ツールチェーン

ソースコードから応用を構築するために、アセンブラ、コンパイラ、リンカ、他に必要とされるツールを含むツールチェーンが必要とされます。本キット用のコードはAVR GNUとIAR AVRのツールチェーンでコンパイル可能です。

2.5.1. AVR GNUツールチェーン

これはATMELのウェブサイトからダウンロードすることができる無料のツールチェーンです。これはAVR応用開発用のC/C++コンパイラ、アセンブラ、リンカのような、全ての必要なツールを含みます。ツールチェーンはAVR GCCプラグインを通してATMELのAVR Studio 4応用と統合します。

AVR GNUツールチェーン(AVR 8/32ビットMCU用無料C/C++コンパイラ)

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

2.5.2. IAR AVRツールチェーン

これはAVR応用開発用のC/C++コンパイラ、アセンブラ、リンカのような、全ての必要なツールを含む商品のツールチェーンです。4K(コード量制限された)自動インストール版だけでなく30日評価版もIARのウェブサイトから入手可能です。

ATMEL AVR用IAR Embedded Workbench

<http://www.iar.com/ewbsite1/1.0.1.0/107/1/>

2.6. 書き込み器とデバッグ

応用がブートローダの使用なしでプログラミングされる、またはデバッグ支援が必要とされるどちらかの時に書き込み器またはデバッグが必要とされます。Xplainキットで直接的に使用することができる多くの各種ツールが利用可能です。最も一般的なツールが以下の節で記述されます。

2.6.1. AVR Dragon

ATMELのAVR DragonはXMEGA-A1 Xplainキットでのコード開発に使用することができる低価格のデバッグ/プログラミングツールです。

ATMEL AVR Dragon(チップ上プログラミングとデバッグのツール)

http://atmel.com/dyn/products/tools_card.asp?tool_id=3891

2.6.2. AVR JTAGICEmk II

ATMELのAVR JTAGICEmk IIはXMEGA-A1 Xplainキットでのコード開発に使用することができる中程度のデバッグ/プログラミングツールです。

ATMEL AVR JTAGICEmk II(チップ上プログラミングとデバッグのツール)

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3353

2.6.3. AVR ONE!

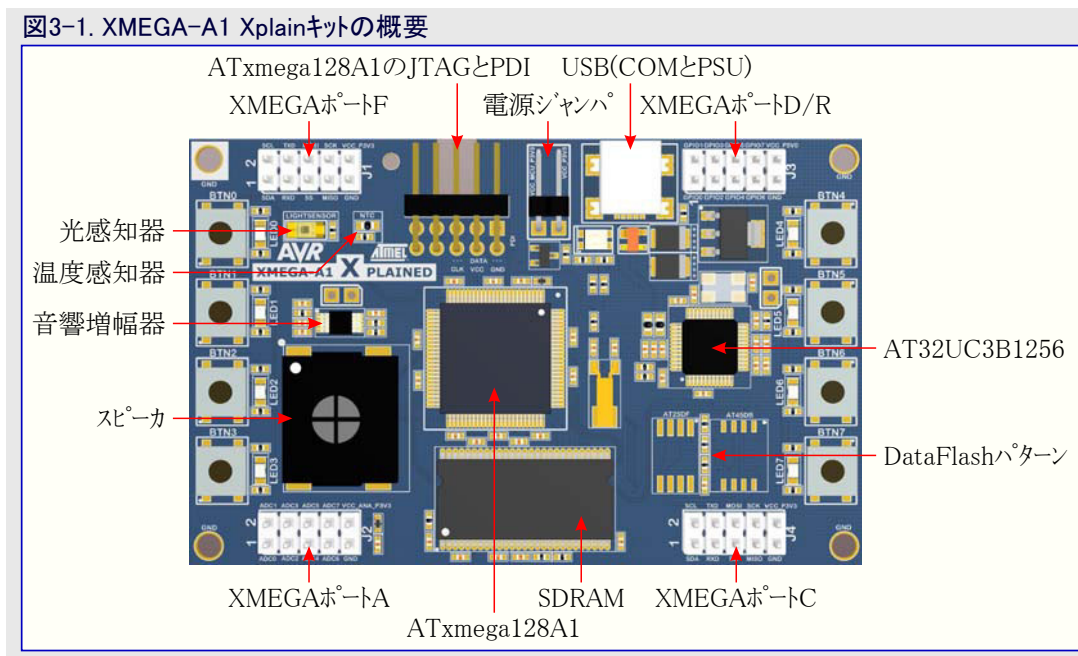
ATMELのAVR ONE!はXMEGA-A1 Xplainキットでのコード開発に使用することができる最上位のデバッグ/プログラミング ツールです。

ATMEL AVR ONE!(チップ上プログラミングとデバッグのツール)

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4279

3. 基板概要

図3-1は基板で利用可能な機能を示します。XMEGA-A1 Xplainハードウェアのより多くの情報については「AVR1924:XMEGA-A1 Xplainハードウェア使用者の手引き」応用記述をご覧ください。



4. 基板接続

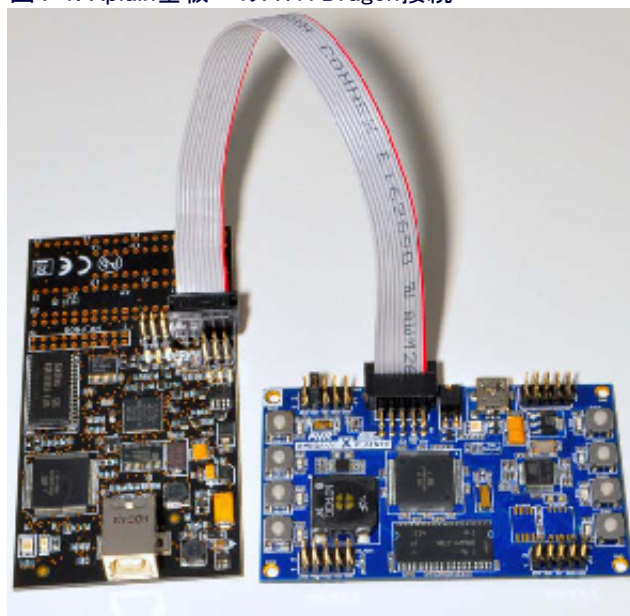
基板とPCまたはそれに通電するためにUSB電源間にUSBケーブルを接続してください。即ちその全てが必要です。電力が印加された時に電源/状態LEDが緑に点灯します。

警告: USBコネクタ隣の装着されたジャンパまたは装着された電流計を持たずに基板へ通電しないでください。そうでなければ、デバイス は入出力ピンを通した電力流れ出しによって損傷されるかもしれません。電力測定ヘッダは図3-1.で示されるようにUSBコネクタの 隣に配置されています。

4.1. AVR Dragonの接続

ATMELのXMEGA-A1 Xplain基板にATMELのAVR Dragonツールを接続するには10ピンヘッダケーブルが必要とされます。AVR DragonキットのJTAGコネクタとXMEGA-A1 XplainのJTAGコネクタ間にケーブルを接続してください。接続がどう行われるべきかを見るには図4-1.をご覧ください。

図4-1. Xplain基板へのAVR Dragon接続



4.2. AVR JTAGICEmk II の接続

XMEGA-A1 Xplain基板に接続する時にATMELのAVR JTAGICEmk II 探針の灰色のコネクタが使用されなければなりません。この接続を行う方法の参照基準については図4-2をご覧ください。灰色コネクタの向きは基板の切断側に配置されなければなりません。

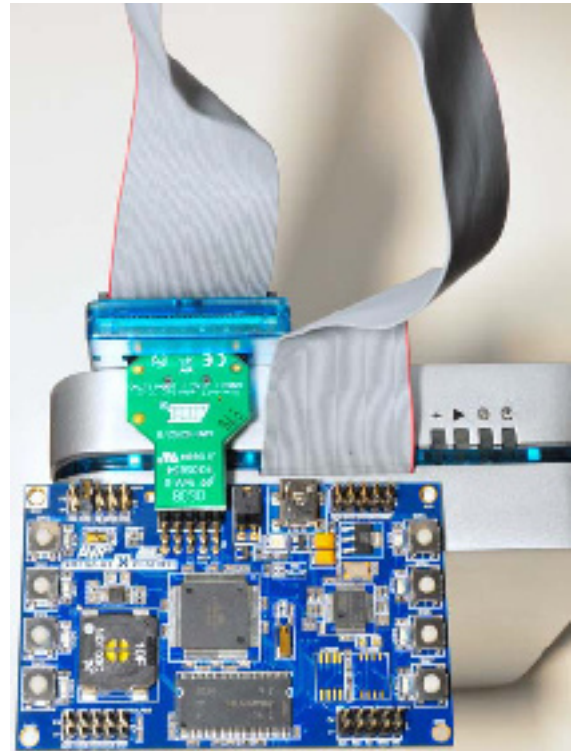
図4-2. Xplain基板へのAVR JTAGICEmk II 探針接続



4.3. AVR ONE! の接続

ATMELのAVR ONE!は図4-3.で示されるように延長アダプタでXMEGA-A1 Xplain基板に接続することができます。

図4-3. Xplain基板へのAVR ONE!デバッグ接続



5. キットのプログラミング

5.1. ブート ローダ経路のプログラミング

ATMELのATxmega128A1に予め書き込まれたブート ローダは、USARTインターフェース経由でデバイスをプログラミングする可能性を提供します。基板制御器の既定ファームウェアがUART-USBブリッジとして働くため、ATxmega128A1はキットへのUSB接続経由でPCからプログラミングすることができます。

ATxmega128A1上のブート ローダはリセット中に機械的なスイッチ(SW0)を押すことによって呼び出されます。プログラミングはインストールしたFLIPからBatchISPコマンド行ツールを使用することによって実行することができます。

5.1.1. 事前必要条件

デバイスをプログラミングするにはFLIPをインストールしたフォルダ下で見つかるBatchISPが必要です。FLIP 3.4.2またはそれ以降版が必要です。

BatchISPを用いてデバイスに新しいファームウェアを書き込むには、基板制御器内のUART-USB交換器をそっくりそのまま持つことが必要です。

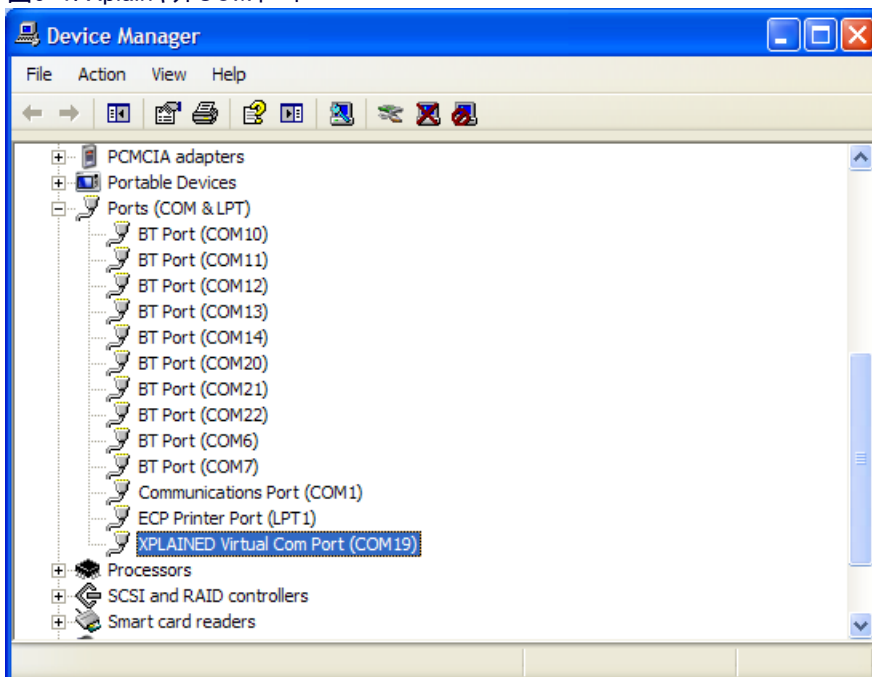
FLIPはデバイスに関するXML形態設定ファイルが必要です。FLIP 3.4.2版を使用する場合、XML形態設定ファイルは含められません。この応用記述と連携するzipファイルからXMLファイルを複製し、そのファイルをFlip 3.4.2¥bin¥PartDescriptionFilesインストール フォルダに置いてください。

5.1.2. 段階的の手引き

デバイスをプログラミングするには以下のこれらの段階に従ってください。

1. キットからUSBケーブルを切断してください。
2. USBケーブルを再接続している間、釦(SW0)を押して保持してください。
3. デバイスマネージャを開いてキットによって使用されているCOMポートに位置付けしてください。

図5-1. XplainキットCOMポート



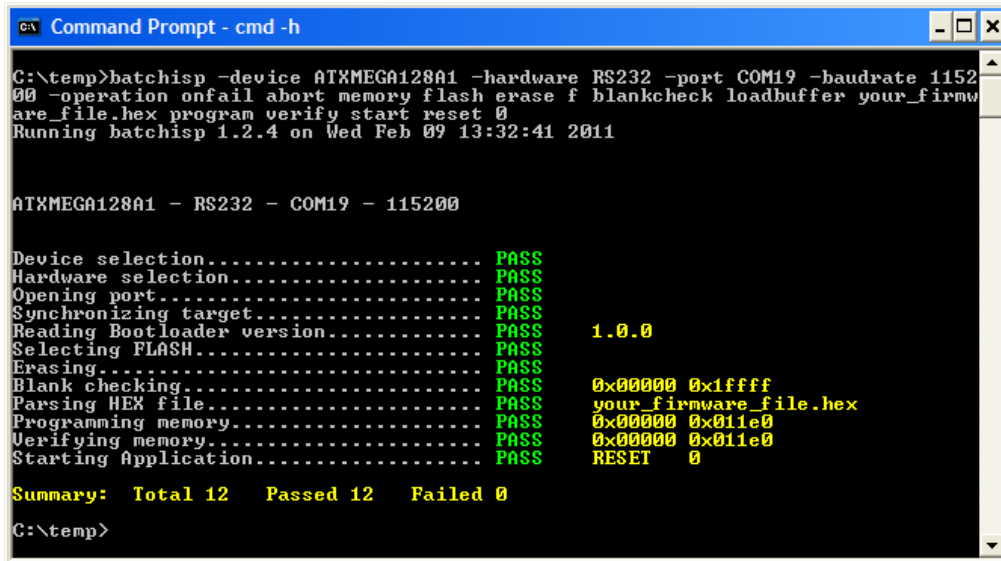
4. 正しいCOMポートを使用して以下の命令を走らせてください。

```
Batchisp -device ATXMEGA128A1
          -hardware RS232
          -port COM19
          -baudrate 115200
          -operation onfail abort
memory flash
erase f
blankcheck
loadbuffer あなたのファームウェア ファイル. hex
program verify
start reset 0
```

5. 今やあなたのファームウェアを動かすべきです。

注: ブートローダは電源投入後に一度だけ同期することができ、そしてそれはBatchISPを用いて一度だけ目的対象に接続することができることを意味します。一度で行う必要がある全てを行えなかった場合、基板の電源をOFF/ONして再び命令を動かすことが必要です。ブートローダ経由でデバイスをプログラミングする例は図5-2.で見ることができます。

図5-2. BatchISP経由でのプログラミング例



```

C:\temp>batchisp -device ATXMEGA128A1 -hardware RS232 -port COM19 -baudrate 115200 -operation onfail abort memory flash erase f blankcheck loadbuffer your_firmware_file.hex program verify start reset 0
Running batchisp 1.2.4 on Wed Feb 09 13:32:41 2011

ATXMEGA128A1 - RS232 - COM19 - 115200

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Synchronizing target..... PASS
Reading Bootloader version..... PASS 1.0.0
Selecting FLASH..... PASS
Erasing..... PASS
Blank checking..... PASS 0x00000 0x1ffff
Parsing HEX file..... PASS your_firmware_file.hex
Programming memory..... PASS 0x00000 0x011e0
Verifying memory..... PASS 0x00000 0x011e0
Starting Application..... PASS RESET 0

Summary: Total 12 Passed 12 Failed 0
C:\temp>
  
```

ブートローダは以下のメモリの読み書きを支援します。

- フラッシュ (\$00000~\$1FFFF、フラッシュメモリの応用領域)
- EEPROM (\$000~\$7FF)
- 使用者 (\$000~\$1FF、使用者識別列)

BatchISPコマンド行ツールについてのより多くの情報に関して、任意選択とパラメータの一覧を得るにはコマンド行から“batchisp -h”を走らせてください。情報についてはFLIPのヘルプファイルを読むこともできます。

5.2. ブートローダの回復

ブートローダが消去されてしまった時は先に言及したプログラミングツールの何れかで回復することができます。

このブートローダ用のhexファイルはこの応用記述と連携するzipファイルで得られ、atxmega128a1_batchisp_uart_bootloader.hexと名付けられています。

注: ブートローダ領域は保護されており、外部書き込み器でだけ消去することができます。

6. 応用例

多くの例はATMELのウェブサイトのASF部からダウンロードすることができるASFで入手可能です。

http://atmel.com/dyn/products/tools_card.asp?tool_id=4192&category_id=163&family_id=607&subfamily_id=2138

各々の例は素早い開始を得るための予め準備されたバイナリだけでなく資料付きの完全なソースコードを持ちます。例はGCCまたはIARのどちらかのコンパイラを用いてコンパイルすることができます。

6.1. 予め書き込まれたファームウェア

ATMELのXMEGA-A1 Xplain基板上的ATMELのATxmega128A1 MCUはUARTブートローダと既定の例ファームウェアで予め書き込まれています。

6.1.1. 予め書き込まれたファームウェアの機能

ATxmega128A1 MCUに予め書き込まれたファームウェアは与えられた何れかの様式でXplain基板上の釦を押すことによって使用者が様式を入力することができる記憶ゲームを含みます。様式が格納されると、使用者は基板によって与えられる還元に従うことで様式の複写を試みることができます。

基板は動作形態を変更するために利用可能ないくつかの形態設定を持ちます。形態設定はPORTA(J2)上にジャンパを配置することによって変更することができます。利用可能な形態設定は次の通りです。

- ポート上にジャンパ装着なしで、基板は並列点灯走行を示します。走行する点灯は釦押下手順がEEPROMに格納されるまで時間経過を記し、記憶のゲームが保存された釦押下を用いて開始されます。基板が時間超過する時に釦が押されなければ、その時間超過がリセットします。
- 電源投入(リセット)中にポートピンの0と1の間にジャンパが置かれた場合、基板は存在すればEEPROM内に格納された既存の釦押下の格納を試み、直接メモリを演奏します。

- 電源投入またはリセットの間中にポートピンの2と3の間にジャンパが置かれた場合、基板は試供音の演奏の代わりに441HzのAから始まる半音階からの音を演奏します。

6.2. 更なるプロジェクト例とチュートリアル

ATxmega128A1の更なる探求についてはデバイスに適用できる応用記述をご覧ください。また、AVRソフトウェア枠組み部署内にはATxmega128A1用に利用可能な多数のプロジェクト例とドライバがあります。正にXMEGA-A1 Xplainキットを目的にした予め作成されたプロジェクトも利用可能です。

ATxmega128A1デバイスでの習熟を得るための基本的な練習についてはAVR1500～AVR1510の応用記述が良い開始点です。これらの応用記述はATMELのウェブサイトから入手可能です。

http://atmel.com/dyn/products/tools_docs.asp?category_id=163&family_id=607&subfamily_id=1965&tool_id=4506

7. プロジェクト例のコンパイル方法

プロジェクトをコンパイルする時にコンパイラは2つのファイルを生成します。HEXファイルはデバイス内に書き込まれる実際のデータを含みます。このHEXファイルは2.6項で一覧にされた書き込み器の1つを用いて、または予め書き込まれたブートローダを用いることによってデバイス内に書き込むことができます。

生成された他のファイルはデバイスに書き込まれるべきデータとデバッグ中に読めるコードを作るためのデバッグ情報の両方を含むデバッグファイルです。このファイルは単一実行(シングルステップ)とデバッグのためにATMELのAVR Studio 4で開くことができます。デバッグファイルのファイル拡張子はコンパイラに依存し、AVR GNUツールチェーンに対する`.elf`、またはIAR Embedded Workbenchに対する`.dbg`のどちらかです。

7.1. AVR GNUツールチェーン

AVR GNUツールチェーンを用いてこのプロジェクトを作るにはシェル/DOSプロンプトを開いて`gcc`副フォルダに誘導してください。`"make"`を入力して[ENTER]を押してください。以下のようなバイナリとデバッグのファイルが作成されます。

```
xmega_applications_xplain_demo_atxmega128a1_xmega_a1_xplained.hex
xmega_applications_xplain_demo_atxmega128a1_xmega_a1_xplained.elf
```

7.2. IAR Embedded Workbench

IAR Embedded Workbenchでプロジェクトを開き、`iar`副フォルダに誘導して以下のプロジェクトを開いてください。

```
xmega_applications_xmega_a1_xplained_demo_atxmega128a1_xmega_a1_xplained.eww
```

プロジェクトはキーボードのF7キーの押下、またはプロジェクトメニューから`"make"`を選択することによってコンパイルすることができます。以下のようなバイナリとデバッグのファイルが作成されます。

```
xmega_applications_xmega_a1_xplained_demo_atxmega128a1_xmega_a1_xplained.hex
xmega_applications_xmega_a1_xplained_demo_atxmega128a1_xmega_a1_xplained.dbg
```

8. 参照

[<http://www.atmel.com/Xplained>]

[<http://www.atmel.com/avrstudio>]

8.1. IARコンパイラ

これはAVR 8ビットMCUに利用可能な商品のC/C++コンパイラです。4K(コード量制限された)自動インストール版だけでなく30日評価版もIARのウェブサイトから入手可能です。

ATMEL AVR用IAR Embedded Workbench

<http://www.iar.com/ewbsite1/1.0.1.0/107/1/>

8.2. AVRソフトウェア枠組み

これはATMELのウェブサイトからダウンロードして無料ででの使用が利用可能なドライバとコードの大きなライブラリです。

ATMEL AVRソフトウェア枠組み(ASF)

http://atmel.com/dyn/products/tools_card.asp?tool_id=4192&category_id=163&family_id=607&subfamily_id=2138

9. 評価基板/キット重要通知

この評価基板/キットは**工作、開発、実演を促進する、または評価目的だけ**の使用を意図されています。これは完成された製品ではなく、(基板/キットに於いて他の方法で注記されるかもしれないのを除き、)リサイクル(WEEE)、FCC、CE、またはULの電磁適合性に関する制限や指令なしで完成製品へ応用できる、含めることの何かまたは何れかの技術的または法律上の必要条件に(未だ)適合しないかもしれません。ATMELは販売者と更にその先の使用者単独の危険に於いて、全ての障害と共に何の保証もなく、“現状そのまま”でこの基板/キットを供給しました。使用者は商品の適切で安全な取り扱いのために全ての義務と責任を負います。また使用者は商品の使用や取り扱いから起こる全ての請求からATMELを保護します。製品の開放構造のため、静電放電と他のどんな技術的または法的な利害関係に関して何れか若しくは全ての適切な予防処置を取るのは使用者の責任です。

上で述べる保障の範囲までを除き、使用者とATMELは間接、特別、付带的、または必然的な損害に関して互いに責任がないでしょう。

そのようなATMELの製品やサービスがあるかもしれない、または使用されることに於いて、どんな機械、処理、または組み合わせに関連または網羅するATMELのどんな特許権や他の知的財産の下でも承諾は全く授けられません。

郵便住所: Atmel Corporation, 2325 Orchard Parkway, San Jose, CA 95131

Copyright © 2011, Atmel Corporation

10. 目次

要点	1
1. 序説	1
2. 何が開始に必要とされますか?	1
2.1. XMEGA-A1 Xplain用資料	1
2.2. AVRソフトウェア枠組み	2
2.3. ATMEL AVRマイクロコントローラ用統合開発環境	2
2.3.1. AVR Studio 4	2
2.3.2. ATMEL AVR用Embedded Workbench	2
2.4. FLIP(BatchISP)	2
2.5. ツールチェーン	2
2.5.1. AVR GNUツールチェーン	2
2.5.2. IAR AVRツールチェーン	2
2.6. 書き込み器とデバッグ	2
2.6.1. AVR Dragon	2
2.6.2. AVR JTAGICEmk II	2
2.6.3. AVR ONE!	3
3. 基板概要	3
4. 基板接続	3
4.1. AVR Dragonの接続	3
4.2. AVR JTAGICEmk IIの接続	4
4.3. AVR ONE!の接続	4
5. キットのプログラミング	5
5.1. ブートローダ経由のプログラミング	5
5.1.1. 事前必要条件	5
5.1.2. 段階的手引き	5
5.2. ブートローダの回復	6
6. 応用例	6
6.1. 予め書き込まれたファームウェア	6
6.1.1. 予め書き込まれたファームウェアの機能	6
6.2. 更なるプロジェクト例とチュートリアル	7
7. プロジェクト例のコンパイル方法	7
7.1. AVR GNUツールチェーン	7
7.2. IAR Embedded Workbench	7
8. 参照	7
8.1. IARコンパイラ	7
8.2. AVRソフトウェア枠組み	7
9. 評価基板/キット重要通知	8
10. 目次	8



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

© 2011 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[®]、STK[®]、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AVR1927応用記述(doc8372.pdf Rev.8372A-04/11)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。