

AVR1624 : USB-TWIブリッジとしての ATxmega128A1 Xplainの使用

要点

- 開放ソース C++コード
- 基準単位設計
- デバイスから独立
- ATMEL® AVR1622でのブートローダ支援

1. 序説

この応用記述はATMEL ATxmega128A1 XplainキットをUSB-TWIブリッジとして使用方法を記述します。この応用はTWI従装置機能を持つ様々な応用との通信に使用することができます。この応用記述ではATxmega128A1上のTWIブートローダ応用を走らせるTWI従装置を使用します。PCに於いてTWIインターフェースに直接データを送るハードウェアがないため、USB-TWIブリッジとしてATMEL AVR® Xplainキットが使用されます。

応用はPCからデバイスへ指令を送るためのコマンド行ユーティリティのTWIGENを提供します。開放ソースコードとその基本単位設計応用は移転を容易にします。このユーティリティはフラッシュメモリのページ容量以外、デバイスのどのパラメータにも依存しません。

このユーティリティにはUSB-TWIブリッジとしてのXplainキットを用いてTWIブートローダとでHEXファイルを直接デバイスにプログラミングする(書き込む)能力があります。HEXファイルのプログラミングとは別に、ユーティリティはどのTWI従装置とも通信することができ、デバイスとのバイトの読み書きの能力があります。従って開発されつつあるTWI従装置での機能を検査するための主装置として使用することもでき、TWIインターフェース上でデバイスへ送られるべきデータの流れを生成するのに使用することができます。

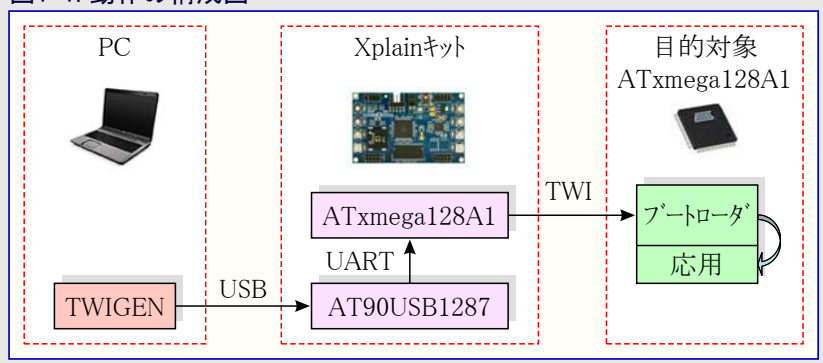
2. Xplainブリッジの準備と走行

本章はXplainブリッジの準備と走行のための基本段階を簡単に片付けます。適切な情報と共に必要な構成と必要条件が記述されます。

2.1. 動作の理屈

この項はATMEL AVR XplainキットがPCとTWIブートローダを持つデバイス間の橋渡し(ブリッジ)としてどのように働く能力があるかを説明します。

図1-1. 動作の構成図



PC上で走行するTWIGENからのデータはUSBインターフェースを通してXplainキットに送られます。Xplainキット上のATMEL AT90USB1287デバイスはこのUSBインターフェース上のデータを受け取ってUARTインターフェース上に送ります。Xplainキット上のATMEL ATxmega128A1はこのUART上のデータを受け取ってTWIインターフェース上でそれをTWIブートローダが走行するデバイスへ渡します。

2.2. USB⇒UART

Xplainキット上のAT90USB1287デバイスはUSBインターフェースからのデータをUARTインターフェース上のデータへ渡すのに使用されます。これは通信装置クラス(CDC:Communication Device Class)を使用して実装されます。CDCクラスは通信装置(モデム、イーサネットなど)に使用される標準USBクラスで仮想COMポートのエミュレートにも使用されます。



8ビット ATMEL
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. 8438A-09/11, 8438AJ1-03/14

以下はXplainキット上のAT90USB1287デバイスのプログラミング(書き込み)とそれの列挙(接続認証)のために従うべき手続きを説明します。

1. ATMEL AVR JTAGICE mk II ヘッドをXplainキット上の'JTAG USB'ヘッドに接続してください。
2. ATMEL AVR Studio® 5を開始してください。
3. Tools⇒AVR Programmingを選択してください。
4. 使用するツールとインターフェースを選択してください。
5. デバイスとしてAT90USB1287を設定してその設定を適用('Apply')してください。
6. 'Memories'タブを選択して'Flash'領域下で'Xplain_USB.a90'を含むフォルダを指定してください。
7. フラッシュメモリをプログラミングして(書き込んで)ください。
8. Xplainキットを切断して再び接続してください。
9. 'Xplain USB Gateway'と'Xplain Serial Port'の表示が出現し、「発見された新しいハードウェアのウィザード」でソフトウェアをインストールするために指示してください。
10. Windows®は仮想COMポートのための固有のドライバを提供せず、故に正しいドライバを取得/設定するためのINFファイルを提供することが必要です。「一覧または特定の場所からインストール(上級者)」を選択して、'at90usbxxx_cdc.inf'ファイルを含むフォルダを指示してください。
11. するとデバイスマネージャのポート(COM&LPT)下に於いて、それに割り当てられた特定のCOM番号で'Xplain Serial Port'として一覧にされるでしょう。

注: 使用されるヒューズ設定は拡張:\$F3、上位:\$99、下位:\$5Eです。

2.3. UART⇒TWI

ATMEL AVR Xplainキット上のATMEL ATxmega128A1デバイスにはUARTインターフェースを通してデータを得てそれをTWIインターフェース上に渡す責任があります。この場合にそれはTWID部署を使用します。

下の説明はXplainキット上のATxmega128A1デバイスをプログラミング(書き込みを)するのに従うべき手続きです。

1. ATMEL JTAGICE mk II ヘッドをXplainキット上の'JTAG&PDI XMEGA'ヘッドに接続してください。
2. 直前項で与えられた手続きの段階2.~4.に従ってください。
3. デバイスとしてAT90USB1287を設定してその設定を適用してください。
4. 'Memories'タブを選択して'Flash'領域下で'XplainSerialToI2CBootLoaderBridge.hex'を含むフォルダを指定してください。
5. フラッシュメモリをプログラミングして(書き込んで)ください。

注: 使用されるヒューズ設定はFUSEBYTE0:\$FF、FUSEBYTE1:\$33、FUSEBYTE2:\$F9、FUSEBYTE4:\$FE、FUSEBYTE5:\$FFです。

3. 目的対象ATxmega128A1デバイスの準備と走行

本章はTWIポートローダを走らせることによってその応用領域を更新されなければならない目的対象ATMEL ATxmega128A1デバイスの構成設定手続きを説明します。同じことを行うための手続きを説明するATMELの「AVR1622:XMEGA®用TWI7ポートローダ」を参照してください。

4. AVR TWIGENの準備と走行

本章はAVR TWIGEN PCユーティリティの準備と走行のための基本段階を簡単に説明します。AVR TWIGEN走行の実演についてはATMEL AVR1622を参照してください。

4.1. 即時開始情報

この項は応用コードの変更の必要がない場合にTWIGENの使用を開始するのに必要な段階を記述します。

twigen.exe実行可能ファイルはTWIGENを使用するためだけに必要とされます。それはこの応用記述と共に来るavr1624.zip内の'Windows side'フォルダに置かれます。ZIPファイルはATMEL AVR XplainキットにプログラミングされるべきHEXファイルと完全なソースコードも含みます。新しいフォルダに実行可能ファイルを複製してPATH環境変数にそのフォルダ名を追加してください。

4.2. コマンド行構文規則

全てのパラメータは-(マイナス)、1つ以上の文字、任意選択の数値で始まらなければなりません。入力ファイル名(-i)を除き、パラメータ(-r, -w, など)と任意選択の値の文字(<バイト数>, <バイト0>, など)の間に1つの空白があるべきです。支援されるコマンド行パラメータは表4-1.で一覧にされます。

表4-1. コマンド行パラメータ

パラメータ	説明
-e	デバイス消去
-i<入力ファイル>	指定したHEXファイルでフラッシュ書き込み。ファイル形式はIntel拡張HEX。
-r<バイト数>	指定したバイト数を読み込み
-w<バイト数> <バイト0> <バイト1> ~	指定したバイト数を書き込み。データはバイト0、バイト1、~によって定義されます。
-a<バイトでのアドレス量> <アドレスバイト0> <アドレスバイト1> ~	指定した補助アドレス量を使用。アドレスはアドレスバイト0、アドレスバイト1、~によって定義されます。
-x	待ち行列にされた指令を実行
-v	操作実行後に指定した操作を検証
-c	操作実行なしで指定した操作を検証
-s<従装置アドレス>	TWI従装置アドレス指定
-p<COMポート>	Xplain基板との通信に使用するPC上のCOMポートを指定
-q	指定した指令を待ち行列へ。待ち行列にされた指令は-x指令実行によって実行することができます。
-d<遅延>	待ち行列にされた指令に対してmsで遅延を指定。この遅延は指令実行前に処理されます。
-n	従装置によって応答されなかった場合にその単位転送を再試行しないことを指定。この任意選択は待ち行列にされた指令に対してだけ有効です。この任意選択が指定されない場合、既定によって各指令は再試行されます。

4.3. コマンド行の例

この項はいくつかの例でAVR TWIGENの動きを説明します。

```
twigen -e -iLEDCHASER.hex -a 3 0x00 0x00 0x00 -s 0x55 -p 14
```

上の例は最初にデバイスを消去してその後にCOMポートの14で、指定したHEXファイルを従装置\$55に3バイトの補助アドレスでプログラミング(書き込み)ます。

```
twigen -e -a 3 0x00 0x00 0x00 -s 0x55 -p 14
```

上の例はCOMポートの14での従装置\$55のフラッシュメモリを消去します。アドレスパラメータは全ての指令に於いて本質的に渡されるべきです。

```
twigen -r 0x20 -a 1 0x00 -s 0x5E -p 26
```

上の例はCOMポートの26と単一バイトの補助アドレスの\$00を用いて従装置\$5Eから\$20バイトを読みます。

```
twigen -r 0x30 -a 2 0x00 0x00 -s 0x52 -p 26
```

上の例はCOMポートの26と2バイトの補助アドレスの\$0000を用いて従装置\$52から\$30バイトを読みます。

```
twigen -w 1 0x11 -a 1 0x07 -s 0x5E -p 26
```

上の例はCOMポートの26と単一バイトの補助アドレスの\$07を用いて従装置\$5Eへ値\$11の1バイトを書きます。

```
twigen -w 2 0x55 0xAA -a 2 0x00 0x10 -s 0x52 -p 26
```

上の例はCOMポートの26と2バイトの補助アドレスの\$0010を用いて従装置\$52へ値\$55AAの2バイトを書きます。

4.4. 規則

この項はAVR TWIGEN使用時に従うべき規則を説明します。

- 一度に1つの操作だけを指定することができます。
- vと-cは同時に指定することができません。
- vと-cは0バイトの書き込みまたは何れかの量の読み込みで指定することができません。
- nは待ち行列にされた指令に対してだけ指定することができます。

5. 実装

本章は読者がオブジェクト志向のプログラミングの概念、特にC++プログラミング言語の一定の知識を持っているとの仮定です。

最上位レベルの作業はtwigenクラスでカプセル化されています。それはHEXファイルを読み書きするのにHEXfileのオブジェクトを使用します。UtilityとErrorMsgの2つのヘルパークラスが応用全体を通して使用されます。

twigenはコマンド行を復号してそれによって処理します。それは主作業を処理し、COMポート処理はcomportファイルによって処理されます。

5.1. クラス記述

この項はクラスの目的の簡単な序説で各クラスと、その後の返り形式、パラメータ、目的でその公開メソッドの各々を記述します。

5.1.1. HEXFile

これはIntel®拡張HEXファイルの読み書きの基本的な機能を提供するクラスです。これは使用されるべきメモリ範囲を定義するためのメソッドも持ちます。これはAVRメモリの部分だけを読み書きするのに有用です。

5.1.1.1. HEXFile

これはクラスに対するコンストラクタです。これは必要とされる最大データ緩衝部容量を示すlongと、緩衝部初期化時に使用されるべき既定バイト値を含むlongの2つのパラメータを取ります。十分なメモリが利用可能でない場合に例外が出されます。

5.1.1.2. ~HEXFile

これはクラスに対するデストラクタです。これは過去に割り当てられた全てのメモリを割り当て解除します。

5.1.1.3. readFile

このメソッドはHEXファイルからデータを読みます。このメソッドはHEXファイル名の1つのパラメータを取り、全く値を返しません。何れかのファイルアクセス異常が起こる、またはファイル形式が不正の場合に例外が出されます。

5.1.1.4. writeFile

このメソッドはHEXファイルへデータを書きます。このメソッドはHEXファイル名の1つのパラメータを取り、全く値を返しません。何れかのファイルアクセス異常が起こる場合に例外が出されます。

5.1.1.5. setUsedRange

このメソッドはメモリ範囲指示子を無効にします。これは読み書き操作に対して範囲を制限するのに使用することができます。メソッドは各々新しい開始と終了の境界を含む2つのlong変数の2つのパラメータを取ります。メソッドは全く値を返しません。提供された範囲が不正な場合に例外が出されます。

5.1.1.6. clearAll

このメソッドはデータ緩衝部全体を望むバイト値に設定します。メソッドは望むバイト値を含むlong変数の1つのパラメータを取り、全く値を返しません。

5.1.1.7. getRangeStart

これは現在の範囲の開始アドレスのためのアクセスメソッドです。メソッドは全くパラメータを取らず、開始アドレスを返します。

5.1.1.8. getRangeEnd

これは現在の範囲の終了アドレスのためのアクセスメソッドです。メソッドは全くパラメータを取らず、終了アドレスを返します。

5.1.1.9. getData

これは緩衝部内のデータのためのアクセスメソッドです。これはバイトアドレスを含むlongの1つのパラメータを取り、バイト値を含むlongを返します。アドレスが適正な範囲外の場合に例外が出されます。

5.1.1.10. setData

これは緩衝部内にデータを設定するためのアクセスメソッドです。これはバイトアドレスを含むlongとバイト値を含むlongの2つのパラメータを取ります。アドレスが適正な範囲外の場合に例外が出されます。

5.1.1.11. getSize

これは緩衝部容量を取得するためのアクセスメソッドです。メソッドは全くパラメータを取らず、バイトでの緩衝部容量を含むlongを返します。

5.1.2. twigen

これはコマンド行パラメータに関して抽出された全ての情報を保持するクラスです。このクラスは必要な操作を実行するための機能も含まれます。

5.1.2.1. twigen

これはクラスに対するコンストラクタです。これはフラッシュへプログラミングされる(書き込まれる)べき入力ファイル名を保持する文字列を消去します。

5.1.2.2. parseArguments

このメソッドはコマンド行パラメータを解析します。これはmain()関数からのありふれたint argcとchar *argv[]の2つのパラメータを取ります。メソッドはメソッドの結果を含むintを返します。成功時に0を、さもなければ-1を返します。

5.1.2.3. checkArguments

このメソッドはパラメータが先の「規則」章で宣言された規則に従っているか否かを調べます。これは全くパラメータを取らず、メソッドの結果を含むintを返します。

5.1.2.4. displayBytes

このメソッドは要求されたバイト数を表示します。これはバイト数を含むshortとデータが格納されるべき緩衝部のアドレスを含むchar *の2つのパラメータを取ります。これは全くパラメータを返しません。

5.1.2.5. getAck

このメソッドは応答バイトを得ます。これは全くパラメータを取らず、応答状態を返します。応答があった場合に0を、さもなければ-1を返します。

5.1.2.6. Ack

このメソッドは応答バイトを送ります。これは全くパラメータを取りません。メソッドは応答が成功裏に送られた場合に0を、さもなければ-1を返します。

5.1.2.7. executeQueue

このメソッドは待ち行列にされた指令を実行します。これはパラメータを全く取らず、メソッドの状態を含むintを返します。これは待ち行列にされた操作が成功裏に実行された場合に0を、さもなければ-1を返します。

5.1.2.8. write

このメソッドは指定されたデータを指定された従装置の指定されたアドレスに書きます。これは従装置アドレスを含むchar、データが書かれるべきアドレスを含むchar *、バイト数を含むshort、それと書かれるべきデータがある緩衝部のアドレスを含むchar *の4つのパラメータを取ります。これは成功裏に書かれた場合に0を、さもなければ-1を返します。

5.1.2.9. read

このメソッドは指定した従装置の指定したアドレスから指定バイト数を読み取ります。これは従装置アドレスを含むchar、データが読まれるべきアドレスを含むchar *、バイト数を含むshort、それとデータが格納されるべき緩衝部のアドレスを含むchar *の4つのパラメータを取ります。これは成功裏に読まれた場合に0を、さもなければ-1を返します。

5.1.2.10. eraseFlash

これは緩衝部内にデータを設定するためのアクセスメソッドです。これはバイトアドレスを含むlongとバイト値を含むlongの2つのパラメータを取ります。アドレスが適正な範囲外の場合に例外が出されます。

5.1.3. utility

このクラスは度々使用される関数の入れ物と名前空間として役に立ちます。これはソースファイル内で実証され、Utilオブジェクトへの外部参照はヘッダファイルで提供されます。

5.1.3.1. Utility

これはクラスに対するコンストラクタです。これはパラメータを全く取りません。このコンストラクタは記録と進捗の両方のメッセージを許可するために内部の記録と進捗の状態を初期化します。

5.1.3.2. ~Utility

これはクラスに対するデストラクタです。これは現在全く機能を持たず、将来の拡張のためのみの置き場です。

5.1.3.3. muteLog

このメソッドは画面に表示されつつある更なる全ての記録メッセージを防ぎます。これはパラメータを全く取らず、値を全く返しません。

5.1.3.4. muteProgress

このメソッドは画面に表示されつつある更なる全ての進捗メッセージを防ぎます。これはパラメータを全く取らず、値を全く返しません。

5.1.3.5. log

このメソッドは無表示でなければ、画面に記録形式メッセージを表示します。メソッドはメッセージ文字列の1つのパラメータを取ります。メソッドは値を全く返しません。



5.1.3.6. progress

このメソッドは無表示でなければ、画面に進捗形式メッセージを表示します。メソッドはメッセージ文字列の1つのパラメータを取ります。メソッドは値を全く返しません。

5.1.3.7. convertHex

このメソッドは16進文字列を数値に変換します。このメソッドは文字列の1つのパラメータを取り、変換された数値を含むlongを返します。何れかの変換誤りが起きた場合に例外が出されます。

5.1.3.8. convertLong

このメソッドは指定された基数を用いて数値を文字列に変換します。これは変換されるべき数値を含むlongと使用されるべき望む基数を含むlongの2つのパラメータを取ります。

5.1.3.9. getRegistryValue

このメソッドはWindowsのレジストリ データベースから値を取得します。このメソッドはレジストリ キーのパスを含む文字列とキー名を含む文字列の2つのパラメータを取ります。メソッドは取得した値を含む文字列を返します。データベース操作中に何れかの異常が起きた場合に例外が出されます。

5.1.4. ErrorMessage

このクラスは例外として出されるべき異常メッセージの入れ物として役に立ちます。

5.1.4.1. ErrorMessage

これはクラスに対するコンストラクタです。これは異常メッセージ文字列の1つのパラメータを取ります。

5.1.4.2. ~ErrorMessage

これはクラスに対するデストラクタです。これは現在全く機能を持たず、将来の拡張のための只の置き場です。

5.1.4.3. What

これは異常メッセージ文字列のためのアクセス メソッドです。これはパラメータを全く取らず、異常メッセージの複製を返します。

5.1.5. Commport

このファイルはクラスを持ちませんが、COMポートを通して通信するための基本単位関数を持ちます。

5.1.5.1. CommPortOpen

この関数は指定された名前でもCOMポートを開いてそのハンドルを得ます。これはCOMポートへのハンドルとCOMポート名を含む文字列の2つのパラメータを取ります。これは関数が成功した場合に真、さもなければ偽のブール値を返します。

5.1.5.2. CommPortInit

この関数は指定されたボーレートで指定されたハンドルのCOMポートを初期化します。これはCOMポートへのハンドル、ボーレートを含むlong、秒での待ち時間を含むcharの3つのパラメータを取ります。これは関数が成功した場合に真、さもなければ偽を返します。

5.1.5.3. CommPortClose

この関数は指定されたハンドルのCOMポートを閉じます。これはCOMポートへのハンドルのパラメータを取ります。これは関数が成功した場合に真、さもなければ偽を返します。

5.1.5.4. CommPortTx

この関数は指定されたCOMポートに指定された出力データを送信します。これはCOMポートへのハンドル、送信されるべきデータを含む緩衝部、バイト数を含むlongの3つのパラメータを取ります。これは関数が成功した場合に真、さもなければ偽を返します。

5.1.5.5. CommPortRx

この関数は指定されたCOMポートから指定された量のデータを指定された緩衝部に受け取ります。これはCOMポートへのハンドル、受信したデータが格納されるべき緩衝部、バイト数を含むlongの3つのパラメータを取ります。これは関数が成功した場合に真、さもなければ偽を返します。

5.1.5.6. CommPortGetLastError

この関数はパラメータを全く取らず、最後の異常を含むdouble wordを返します。

5.1.5.7. CommPortFlushRxBuffer

この関数は受信したデータを格納する緩衝部を破棄します。これはCOMポートへのハンドルのパラメータを取ります。この関数は成功した場合に真、さもなければ偽を返します。

5.1.5.8. CommPortRxReady

この関数はCOMポートがデータを受信する準備が整っているか否かを調べます。これはCOMポートへのハンドルとバイト数を含むlongの2つのパラメータを取ります。これはデータ受信の準備が整っている場合に真、さもなければ偽を返します。

6. 目次

要点	1
1. 序説	1
2. Xplainブリッジの準備と走行	1
2.1. 動作の理屈	1
2.2. USB⇒UART	1
2.3. UART⇒TWI	2
3. 目的対象ATxmega128A1デバイスの準備と走行	2
4. AVR TWIGENの準備と走行	2
4.1. 即時開始情報	2
4.2. コマンド行構文規則	3
4.3. コマンド行の例	3
4.4. 規則	3
5. 実装	4
5.1. クラス記述	4
5.1.1. HEXFile	4
5.1.2. twigen	4
5.1.3. utility	5
5.1.4. ErrorMessage	6
5.1.5. Commport	6
6. 目次	7



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®、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AVR1624応用記述(doc8438.pdf Rev.8438A-09/11)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。