

Atmel AVR JTAGICE mkII デバッグ



Atmel[®] AVR[®] JTAGICE mkIIは全てのAtmel AVR 8及び32ビット マイクロ コントローラとチップ上デバッグ能力を持つプロセッサでのチップ上デバッグとプログラミングを支援します。

支援されるインターフェースは次のとおりです。

- JTAG (32ビットAVR、AVR[®]XMEGA[®]、megaAVR[®]デバイス)
- PDI (AVR XMEGAデバイス) **注:**ハードウェア改訂01のみ
- デバッグWIRE (megaAVRとtinyAVR[®]デバイス)
- SPI (megaAVRとtinyAVRデバイス)
- aWire (32ビットAVRデバイス) **注:**ハードウェア改訂01のみ

| | | | |
|---|----|---------------------------------|----|
| Atmel AVR JTAGICE mkIIデバッグ | 1 | 8.5. Atmel AVR UC3 OCD | 23 |
| 1. 序説 | 3 | 9. 障害対策 | 24 |
| 1.1. Atmel JTAGICE mkIIの特徴 | 3 | 9.1. 障害対策の手引き | 24 |
| 1.2. システム要件 | 3 | 10. ファームウェア格上げ更新 | 25 |
| 1.3. ハードウェア改訂 | 3 | 11. 公開履歴と既知の問題 | 25 |
| 2. 開始に際して | 4 | 11.1. 新規情報 | 25 |
| 2.1. キット内容 | 4 | 11.2. ファームウェア公開履歴(Atmel Studio) | 25 |
| 2.2. Atmel JTAGICE mkIIの給電 | 4 | 11.3. 既知の問題 | 25 |
| 2.3. ホストコンピュータへの接続 | 5 | 11.3.1. 全般 | 25 |
| 2.4. シリアルポート接続 | 5 | 11.3.2. ハードウェア関連 | 25 |
| 2.5. USBドライバのインストール | 5 | 11.3.3. Atmel AVR XMEGA関連 | 26 |
| 2.5.1. Windows | 5 | 11.3.4. JTAG (mega)関連 | 26 |
| 2.6. デバッグ | 6 | 11.3.5. デバッグWIRE関連 | 26 |
| 3. Atmel JTAGICE mkIIの接続 | 6 | 11.3.6. 共通 | 26 |
| 3.1. JTAG目的対象への接続 | 6 | 12. 改訂履歴 | 27 |
| 3.1.1. JTAG 10ピンコネクタの使い方 | 6 | | |
| 3.2. PDI目的対象への接続 | 7 | | |
| 3.3. デバッグWIRE目的対象への接続 | 9 | | |
| 3.4. aWire目的対象への接続 | 10 | | |
| 3.5. SPI目的対象への接続 | 11 | | |
| 3.6. Atmel STK500とのJTAGICE mkIIの使い方 | 11 | | |
| 3.7. Atmel STK600とのJTAGICE mkIIの使い方 | 13 | | |
| 4. チップ上デバッグ | 15 | | |
| 4.1. チップ上デバッグ(OCD)の序説 | 15 | | |
| 4.2. 物理インターフェース | 15 | | |
| 4.2.1. JTAG | 15 | | |
| 4.2.2. aWire物性 | 17 | | |
| 4.2.3. PDI物性 | 17 | | |
| 4.2.4. デバッグWIRE | 17 | | |
| 4.2.5. SPI | 17 | | |
| 4.3. Atmel AVR OCD実装 | 18 | | |
| 4.3.1. Atmel AVR UC3 OCD (JTAGとaWire) | 18 | | |
| 4.3.2. Atmel AVR XMEGA OCD (JTAGとPDI物性) | 18 | | |
| 4.3.3. Atmel megaAVR OCD (JTAG) | 18 | | |
| 4.3.4. Atmel megaAVR/tinyAVR OCD (デバッグWIRE) | 18 | | |
| 5. ハードウェア説明 | 18 | | |
| 5.1. 物理的寸法 | 18 | | |
| 5.2. LED | 19 | | |
| 5.3. 背面 | 19 | | |
| 5.4. 基本構造説明 | 19 | | |
| 5.4.1. 電源 | 19 | | |
| 5.4.2. レベル変換器 | 19 | | |
| 5.4.3. 探針 | 20 | | |
| 6. ソフトウェア統合 | 20 | | |
| 6.1. Atmel Studio | 20 | | |
| 6.1.1. Atmel Studio | 20 | | |
| 6.1.2. Atmel StudioプログラミングGUI | 20 | | |
| 6.1.3. プログラミング任意選択 | 20 | | |
| 6.1.4. デバッグ任意選択 | 20 | | |
| 7. コマンド行ユーティリティ | 21 | | |
| 8. 特別な考慮 | 21 | | |
| 8.1. Atmel AVR XMEGA OCD | 21 | | |
| 8.2. Atmel megaAVR OCDとデバッグWIRE OCD | 21 | | |
| 8.3. Atmel megaAVR OCD (JTAG) | 22 | | |
| 8.4. デバッグWIRE OCD | 23 | | |

1. 序説

1.1. Atmel JTAGICE mkIIの特徴

- Atmel Studio、AVR32 Studio、AVR Studio®4とそれ以降に完全適合
- OCDを持つ全てのAtmel AVR 8及び32ビット マイクロ コントローラのデバッグを支援
- OCDを持つ全ての8及び32ビット AVRデバイスのプログラミングを支援
- 正確な電気的特性
- デジタルとアナログのチップ上機能の模倣(エミュレート)
- ソフトウェア中断点(ブレイクポイント) (注:ATmega128[A]を除く)
- プログラム メモリ中断点
- アセンブリ言語と高位言語でのソースレベル デバッグを支援
- フラッシュ メモリ、EEPROM、ヒューズ、施錠ビットのためのプログラミング インターフェース (デバッグWIREを除く)
- プログラミングと制御用のPCに対するUSB 1.1とRS232インターフェース
- 9～15V DC電力に対する調整された電源
- USBバスからの給電可
- 1.65～5.5Vの目的対象動作電圧範囲

1.2. システム要件

Atmel JTAGICE mkII本体はコンピュータに前置デバッグ環境(Atmel Studio、AVR32 StudioまたはAVR Studio 4.9またはそれ以降)と連携するユーティリティがインストールされることが必要です。これらの一括のシステム要件についてはwww.atmel.comを調べてください。

JTAGICE mkII本体は提供されたUSBまたはRS232ケーブルのどちらかを使ってホスト コンピュータに接続されなければなりません。2つの接続任意選択の機能は同じです。

注: Atmel StudioはRS232直列通信を支援しません。

JTAGICE mkII本体は加えて9～15V DC外部電源に接続し得ます。ケーブルはキットに含まれます。外部電源が接続される場合、USB電力は使われません。

1.3. ハードウェア改訂

改訂0のハードウェアはPDIやaWireのインターフェースを支援せず、ここで示されるようにA0～で始まる通番を持ちます。



改訂1のハードウェアはPDIとaWireの両方を支援します。通番はここで示されるようにA09-0041またはB0～で始まります。



改訂1は筐体内にUSB接続が行われた時に点灯する緑LEDもあります。

改訂1本体は完全にRoHS適合でもあります。

2. 開始に際して

2.1. キット内容

- ・ 探針付きAtmel AVR JTAGICE mkII本体
- ・ USBケーブル (1.8m)
- ・ RS232シリアル ケーブル
- ・ DC電源ケーブル
- ・ 10ピン(JTAG)⇒6ピン(SPI)探針アダプタ ケーブル
- ・ 10芯多色特注コネクタ(ハラ線)ケーブル
- ・ 予備30芯フレキシブル ケーブル
- ・ AVR技術ライブラリDVD
- ・ Atmel AVR XMEGAマイクロ コントローラ用PDIアダプタ

図2-1. JTAGICE mkIIキット内容



2.2. Atmel AVR JTAGICE mkIIの給電

Atmel AVR JTAGICE mkIIは9～15V DCを提供する外部電源を使って動作することができ、またはUSBバスから直接給電することができます。内部スイッチは既定で外部電源からの給電を選びます。けれども、これが接続されない、または外部電源が使用可能なレベル以下に落ちる場合、電力は(接続されていれば)USBから取られます。

例えどの極性でも動くとは言え、電力スイッチが接地にするため、DCジャックの望まれる極性は中心が-です。JTAGICE mkIIの電源投入時、電力LEDが直ちに点灯するでしょう。LEDが点灯しない場合、十分な電源が使われていることを調査してください。

注: JTAGICE mkIIがUSBだけから給電される場合、USBが最低500mA配給できる(自己給電ハブ)ことが必要とされます。

JTAGICE mkIIが目的対象とホストPCに正しく接続されると、電力をONにすることができます。給電されていないJTAGICE mkIIに目的対象から流れる電流の可能性を防ぐため、目的対象給電前にJTAGICE mkIIの電源投入が推奨されます。

「ハードウェア説明」章で示されるように2つのLEDは各々エミュレータ電力と目的対象電力を示します。

2.3. ホストコンピュータへの接続

最初にAtmel AVR JTAGICE mkIIに接続する前に、ホストコンピュータにUSBドライバをインストールことを確実にしてください。これはAtmelによって無料で提供される前置ソフトウェアをインストールする時に自動的に行われます。更なる情報や最終版の前置ソフトウェアのダウンロードについてはwww.atmel.comをご覧ください。

JTAGICE mkIIはUSBケーブルを通してホストPCに、またはシリアルケーブルを(PCのCOMポートに)接続することができます。

2.4. シリアルポート接続

AVR Studio 5とAtmel StudioはRS232直列通信を支援しません。

2.5. USBドライバのインストール

Atmel JTAGICE mkII用のUSBドライバはAtmel Studio、AVR Studio 4とそれ以降版、AVR32 Studio前置ソフトウェアでインストールされます。

2.5.1. Windows

Microsoft® Windows®が走行するコンピュータでJTAGICE mkIIをインストールすると、本体が最初に電源投入される時にUSBドライバが読み込まれます。

注: 最初に電源投入する前に、前置ソフトウェア一括のインストールを確実にしてください。

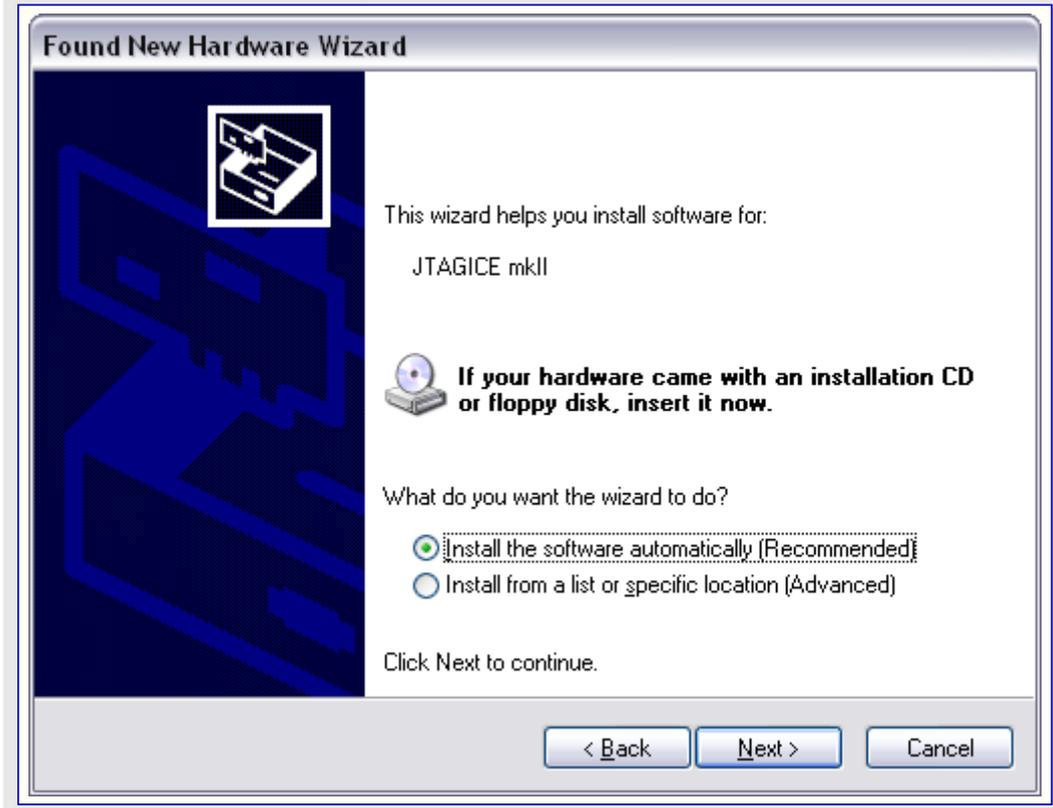
新規ハードウェア ウィザードを通して既定(“推奨”)任意選択で処理してください。



図2-2. JTAGICE mkIIのUSBドライバ インストール



図2-3. JTAGICE mkIIのUSBドライバインストール



自動的に検出されない場合、ウィザードを<Windowsルート>\Infフォルダに格納されるJTAGICEmkII.infと呼ばれる(Jungoによって提供された)デバイスドライバへ指示してください。

一旦、成功裏にインストールされると、JTAGICE mkIIはデバイス マネージャで”Jungo”装置として現れるでしょう。

JTAGICE mkIIは今や使用の準備が整いました。



2.6. デバッグ

Atmel Studioを使ってAtmel AVR JTAGICE mkIIとで開始する最も簡単な方法はASFからの多くの例プロジェクトの1つを構築することです。より多くの情報をご覧ください。

3. Atmel JTAGICE mkIIの接続

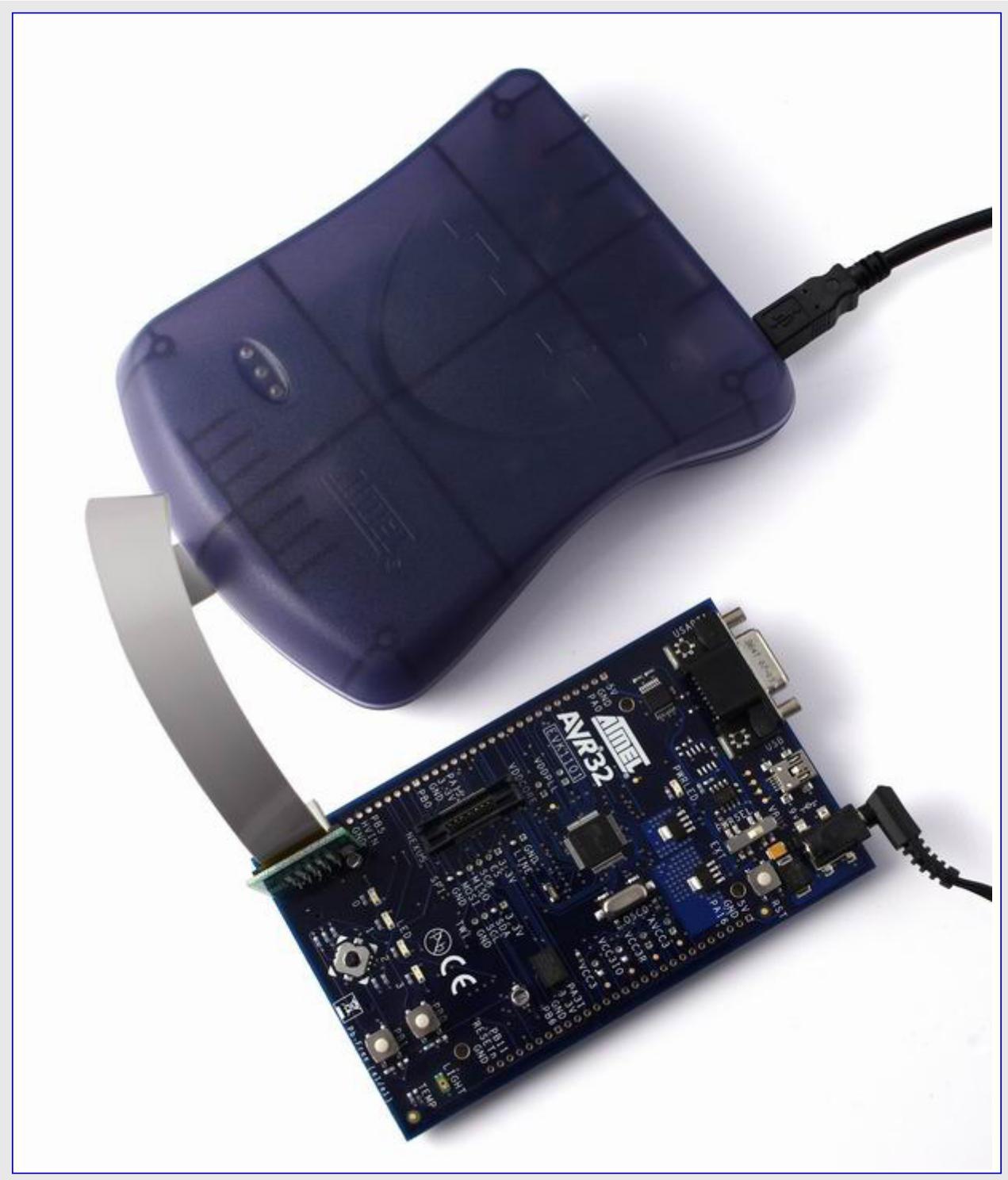
3.1. JTAG目的対象への接続

Atmel JTAGICE mkII探針は全てのデバッグとプログラミングを支援する2つの目的対象10ピン コネクタを持ちます。各コネクタのピン配列は同じです。JTAGデバッグに対して、この10ピン コネクタを直接使うことができます。他のインターフェースについてはアダプタが必要とされます。

3.1.1. JTAG 10ピン コネクタの使い方

10ピン JTAGコネクタのピン配列は図4-2. JTAGヘッダ ピン配列で示されます。

JTAGICE mkIIを目的対象応用PCBに接続する時に10ピン ヘッダを正しい向きで使うことを確実にしてください。必要なら、50mil独立アダプタがAtmelから入手可能です。



JTAG物理インターフェースのより多くの情報については「[物理インターフェース](#)」の「[JTAG](#)」をご覧ください。

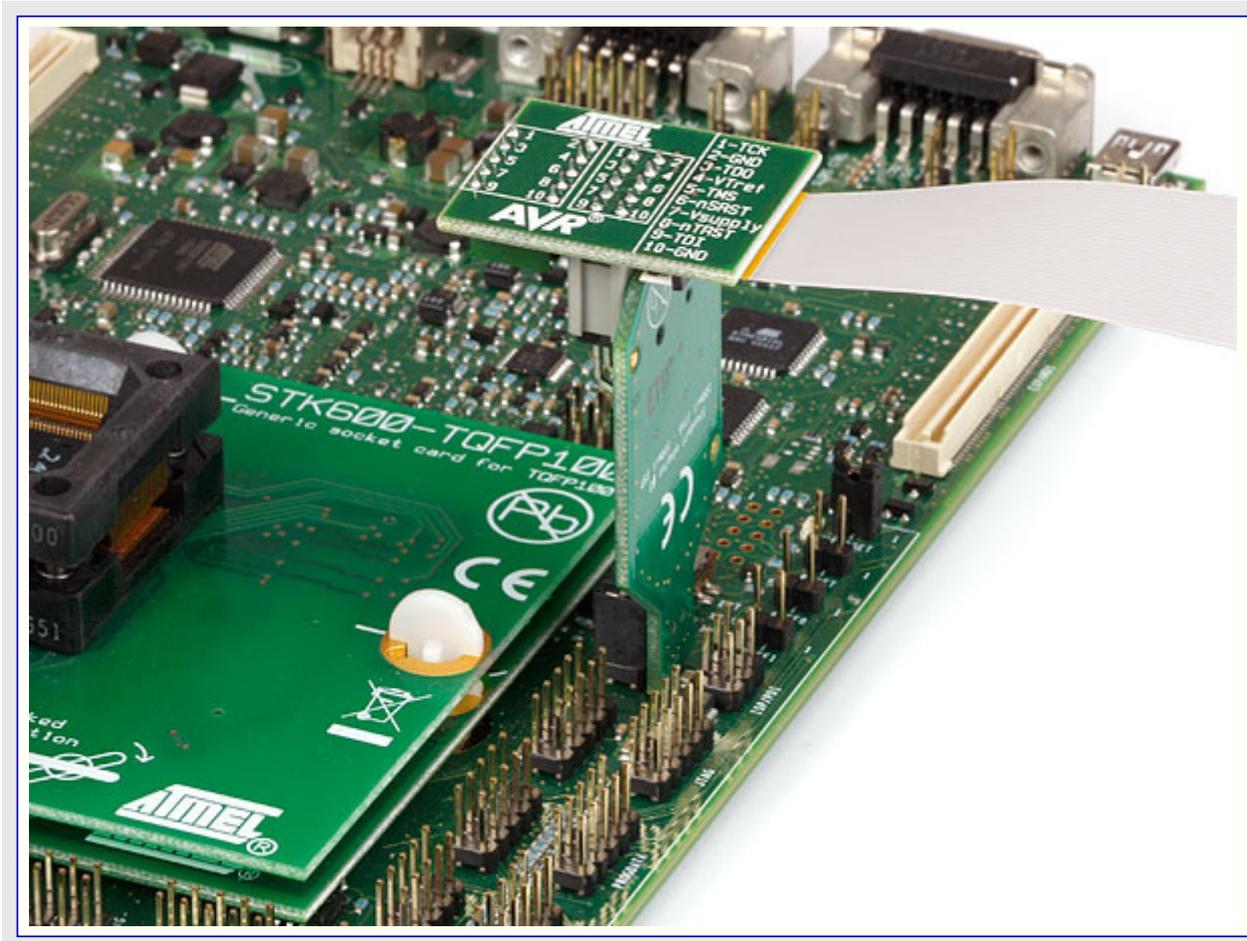
3.2. PDI目的対象への接続

6ピンPDIコネクタ用のピン配列は[図4-5. PDIヘッダピン配列](#)で示されます。

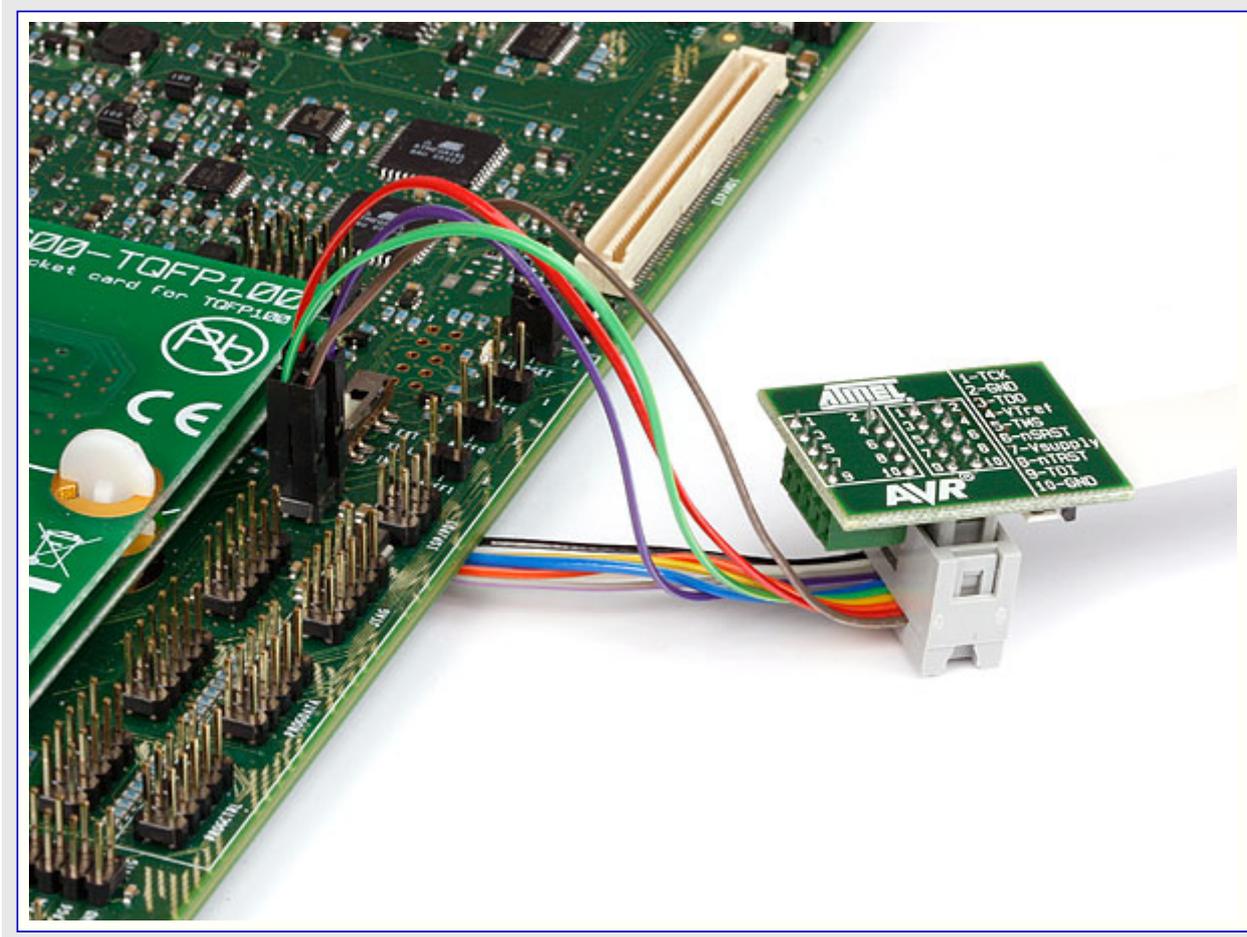
注: ハードウェア改訂0のAtmel AVR JTAGICE mkII本体はPDI能力を持ちません。本体がどのハードウェア改訂かのより多くの情報については「[ハードウェア改訂](#)」をご覧ください。

本体が改訂0の場合、このハードウェアを使うPDIのプログラミングとデバッグは不可能です。けれども、例えそれが目的対象デバイスの入出力ピンを使うとは言え、(目的対象デバイスで利用可能ならば)JTAGインターフェースはPDIと同じ機能を提供します。

上で言及されたピン配列は元々のAtmel STK[®]600だけでなく将来的なAtmel AVR XMEGA能力ツールによっても支援されます。このピン配列でJTAGICE mkIIを使うために、最寄りのAtmel代理店で入手可能なJTAGICE mkII用XMEGA PDIアダプタを使うことが必要です。



代わりに、PDIインターフェースはJTAGICE mkIIキットと共に出荷される多色”バラ線”ケーブルを使用して接続することもできます。



標準6ピンヘッダを持たない目的対象への接続時、JTAGICE E mkII 探針上の10ピンJTAGコネクタと目的対象基板間パラ線ケーブルを使うことができます。4つの接続が必要とされ、右表はそれらを接続する場所を記述します。

注: PDI_DATAは殆どのツールで使われる標準3番ピン接続と比較して、JTAGコネクタの9番ピンに接続されます。

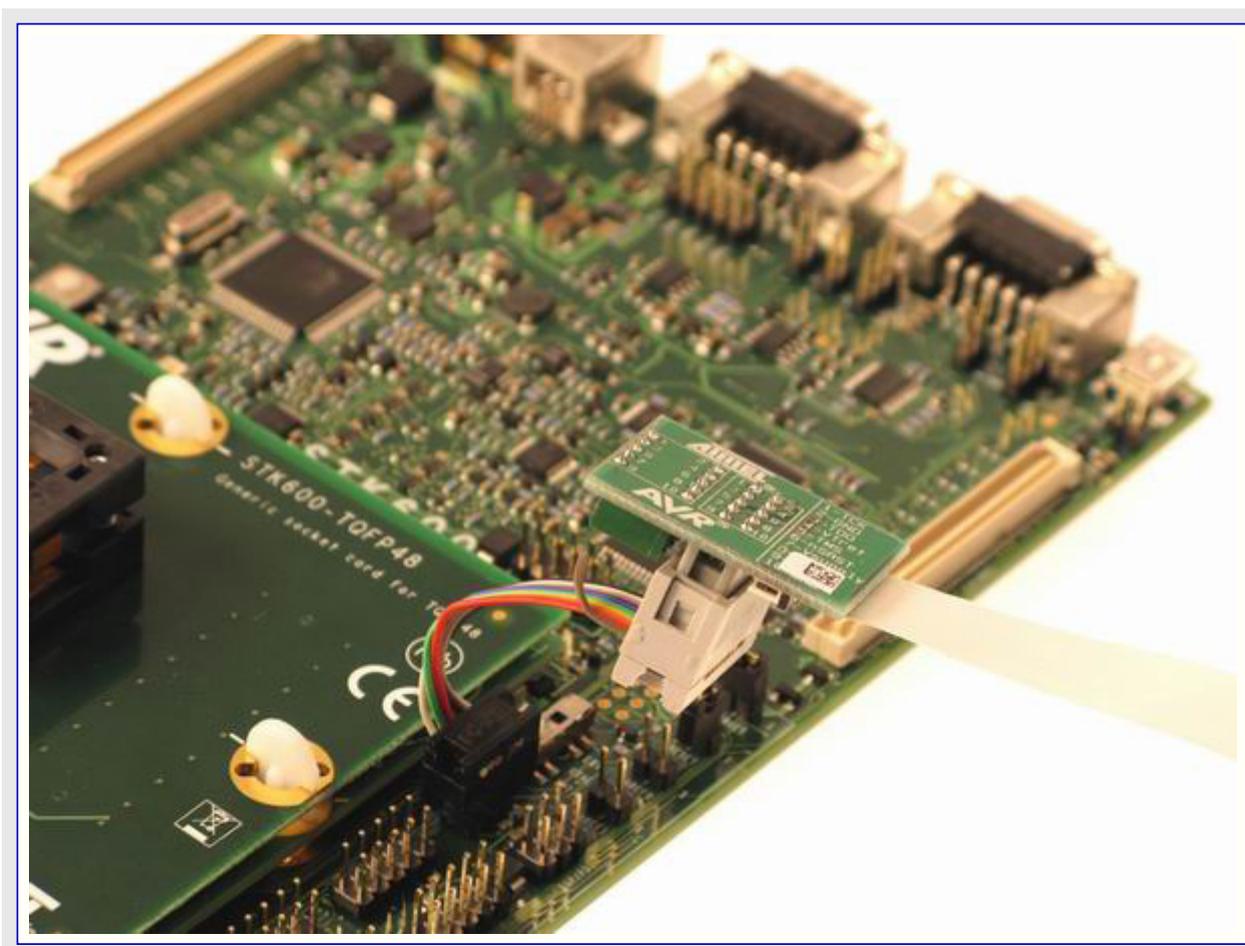
表3-1. パラ線ケーブルを使うPDIへの接続

| JTAGICE mkIIピン | 目的対象ピン | パラ線ケーブル色 | STK600 PDIピン |
|----------------|----------|----------|--------------|
| 1 (TCK) | | 黒 | |
| 2 (GND) | GND | 白 | 6 |
| 3 (TDO) | | 灰 | |
| 4 (VTref) | VTref | 紫 | 2 |
| 5 (TMS) | | 青 | |
| 6 (nSRST) | PDI_CLK | 緑 | 5 |
| 7 (未接続) | | 黄 | |
| 8 (nTRST) | | 橙 | |
| 9 (TDI) | PDI_DATA | 赤 | 1 |
| 10 (GND) | | 茶 | |

3.3. デバッグWIRE目的対象への接続

6ピン デバッグWIREコネクタ用のピン配列は図4-6. デバッグWIRE(SPI)ヘッダピン配列で示されます。

Atmel AVR JTAGICE mkIIを目的対象応用PCBに接続する時に6ピンヘッダを正しい向きで使うことを確実にしてください。



例え正しく動作するためにデバッグWIREインターフェースがVCC、GNDと1つの単線(RESET)だけが必要でも、SPIプログラミングを使ってデバッグWIREインターフェース(DWENヒューズ)を許可/禁止することができるように完全なSPIコネクタへの入出力を持つことが推奨されます。

デバッグWIREインターフェース上での正しい通信を保証するためにRESET線に関していくつかの予防策がとられなければなりません。RESET線にプルアップ抵抗がある場合、この抵抗は10kΩよりも大きくなければなりません。JTAGICE mkIIは内部RESETプルアップを持ちます。RESET線の容量性負荷は取り去られるべきです。RESET線に繋がれる他のどの論理回路も取り去られるべきです。

DWENヒューズが許可されると、OCD単位部がRESETピン上の制御を持つために、内部的にSPIインターフェースが無効にされます。デバッグWIRE OCDは(Atmel Studioのプロパティダイアログ内のデバッグタブ上の卸を使って)一時的にそれ自身を禁止する能力があり、故にRESET線の制御を解放します。(SPIヒューズがプログラム(0)される場合にだけ、)その後にSPIインターフェースが再び利用可能で、SPIインターフェースを使ってDWENヒューズを非プログラム(1)にすることを許します。DWENヒューズが非プログラム(1)にされる前に電力がOFF/ONされるとデバッグWIRE単位部は再びRESETピンの制御を取ります。単純にAtmel StudioにDWENヒューズの設定と解除を扱わせることを強く勧告します。

目的対象Atmel AVRの施錠ビットがプログラム(0)されている場合、デバッグWIREインターフェースを使うのは不可能です。常にDWENヒューズをプログラム(0)する前に施錠ビットが解除(1)されていることを確実にして、DWENヒューズがプログラム(0)されている間に施錠ビットを決して設定(0)しないでください。デバッグWIRE許可(DWEN)ヒューズと施錠ビットの両方が設定(0)される場合、チップ消去を行うのに高電圧プログラミングを使うことができ、そのようにして施錠ビットを解除(1)してください。施錠ビットが解除(1)されると、デバッグWIREインターフェースが再び許可されます。DWENヒューズが非プログラム(1)の時のSPIインターフェースはヒューズ読み出し、識票読み出し、チップ消去実行の能力だけです。

3.4. aWire目的対象への接続

6ピンaWireコネクタ用のピン配列は図4-4. aWireヘッダピン配列で示されます。

注: ハードウェア改訂0のAtmel AVR JTAGICE mkII本体はaWire能力を持ちません。本体がどのハードウェア改訂かのより多くの情報については「ハードウェア改訂」をご覧ください。

本体が改訂0の場合、このハードウェアを使うaWireのプログラミングとデバッグは不可能です。けれども、例えそれが目的対象デバイスの入出力ピンを使うとは言え、(目的対象デバイスで利用可能ならば)JTAGインターフェースはaWireと同じ機能を提供します。

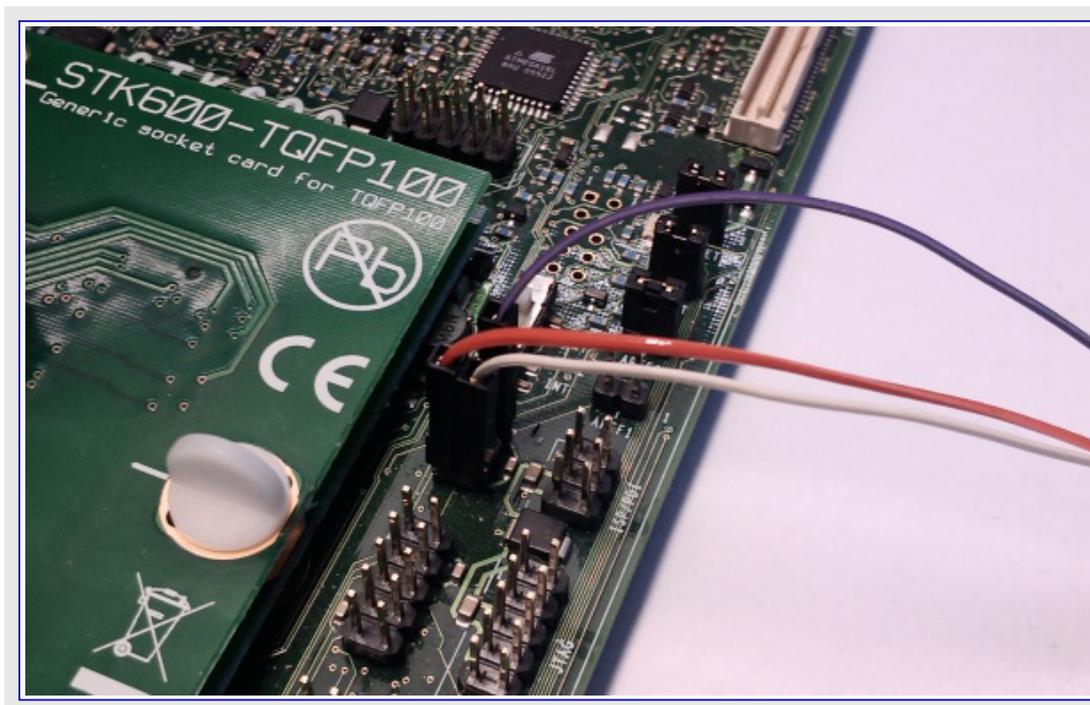
右で示されるピン配列はaWire用推奨ピン配列で、将来のaWire能力ツールによって元々支援されます。JTAGICE mkIIはこの配列への割り当てに使われる10ピン多色”パラ線”ケーブルが必要です。

表3-2. パラ線ケーブルを使うSPIへの接続

| JTAGICE mkIIピン | 目的対象ピン | パラ線ケーブル色 | SPIピン |
|----------------|--------|----------|-------|
| 1 (TCK) | SCK | 黒 | 3 |
| 2 (GND) | GND | 白 | 6 |
| 3 (TDO) | MISO | 灰 | 1 |
| 4 (VTref) | VTref | 紫 | 2 |
| 5 (TMS) | | 青 | |
| 6 (nSRST) | /RESET | 緑 | 5 |
| 7 (Vsupply) | | 黄 | |
| 8 (nTRST) | | 橙 | |
| 9 (TDI) | MOSI | 赤 | 4 |
| 10 (GND) | | 茶 | |

表3-3. パラ線ケーブルを使うaWireへの接続

| JTAGICE mkIIピン | 目的対象ピン | パラ線ケーブル色 | aWireピン |
|----------------|--------|----------|---------|
| 1 (TCK) | | 黒 | |
| 2 (GND) | GND | 白 | 6 |
| 3 (TDO) | | 灰 | |
| 4 (VTref) | VTref | 紫 | 2 |
| 5 (TMS) | | 青 | |
| 6 (nSRST) | | 緑 | |
| 7 (未接続) | | 黄 | |
| 8 (nTRST) | | 橙 | |
| 9 (TDI) | aWire | 赤 | 1 |
| 10 (GND) | | 茶 | |



3.5. SPI目的対象への接続

6ピンSPIコネクタ用のピン配列は図4-7. SPIヘッダピン配列で示されます。

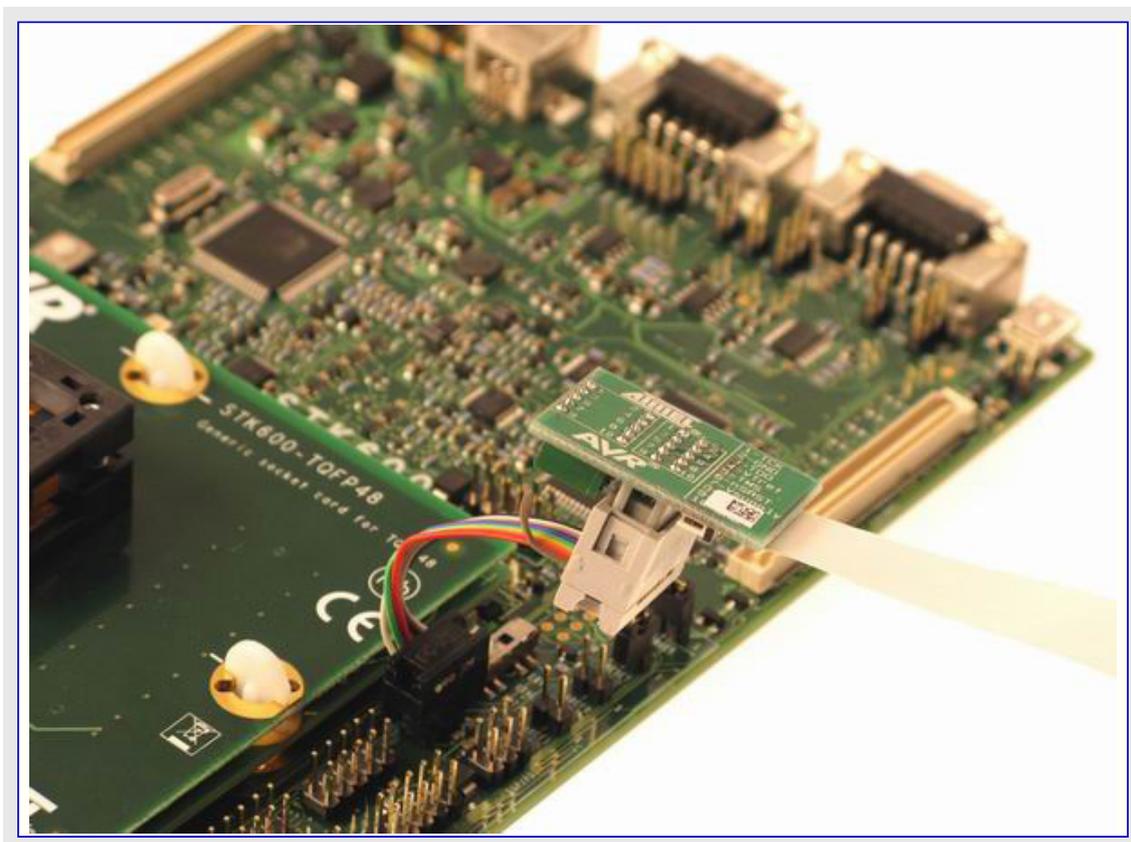
Atmel AVR JTAGICE mkIIを目的対象応用PCBに接続する時に6ピンヘッダを正しい向きで使うことを確実にしてください。

注: デバッグWIRE許可(DWEN)ヒューズがプログラム(0)されている時に、例えばSPIENヒューズもプログラム(0)されていても、SPIインターフェースは事実上禁止されます。SPIインターフェースを再許可するには、デバッグWIREでのデバッグ作業中に' disable debugWIRE' 命令が発行されなければなりません。この方法でのデバッグWIRE禁止はSPIENヒューズが既にプログラム(0)されていることが必要です。Atmel StudioがデバッグWIRE禁止を失敗する場合、多分SPIENヒューズがプログラム(0)にされていないでしょう。その場合、SPIENヒューズをプログラム(0)するのに高電圧プログラミングインターフェースを使うことが必要です。単純にAtmel StudioにDWENヒューズの設定と解除を扱わせることを強く勧告します。



表3-4. パラ線ケーブルを使用するSPIへの接続

| JTAGICE mkIIピン | 目的対象ピン | パラ線ケーブル色 | SPIピン |
|----------------|--------|----------|-------|
| 1 (TCK) | SCK | 黒 | 3 |
| 2 (GND) | GND | 白 | 6 |
| 3 (TDO) | MISO | 灰 | 1 |
| 4 (VTref) | VTref | 紫 | 2 |
| 5 (TMS) | | 青 | |
| 6 (nSRST) | /RESET | 緑 | 5 |
| 7 (Vsupply) | | 黄 | |
| 8 (nTRST) | | 橙 | |
| 9 (TDI) | MOSI | 赤 | 4 |
| 10 (GND) | | 茶 | |



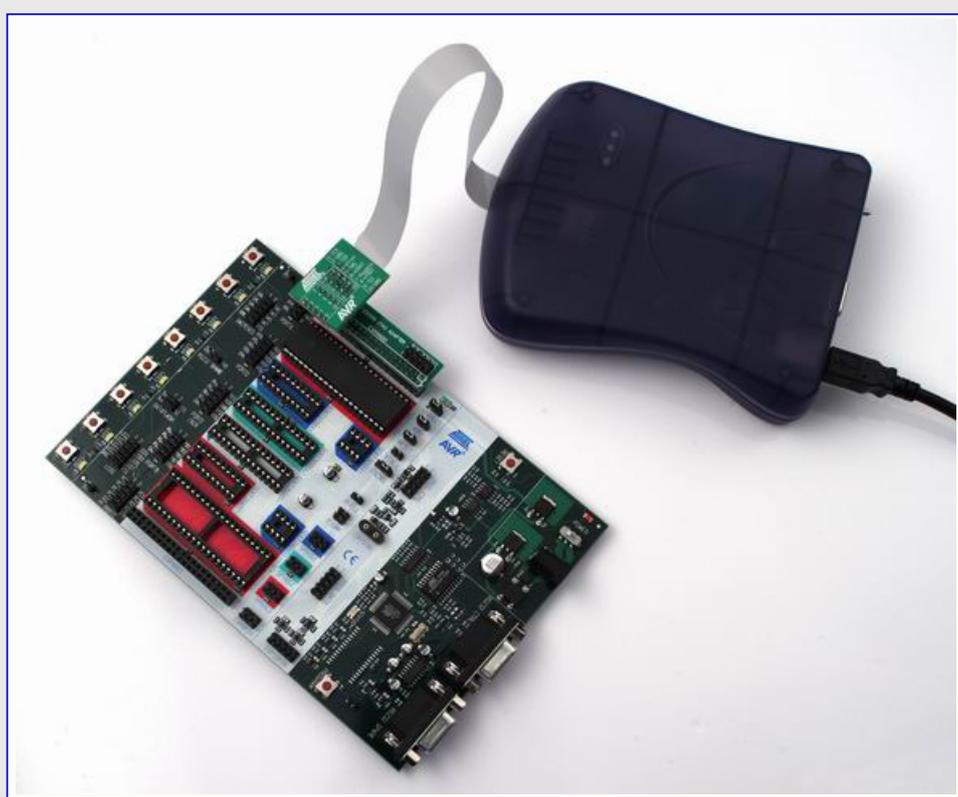
3.6. Atmel STK500とのAtmel JTAGICE mkIIの使い方

Atmel STK500スタータキットはJTAG、デバッグWIRE、SPIインターフェースを通してAtmel AVR JTAGICE mkIIを接続することができるAtmel AVRデバイスの収容場所に使うことができます。

JTAG目的対象への接続時、単純にここで示されるATSTK500_JTAG_ADAPTERを使ってください。



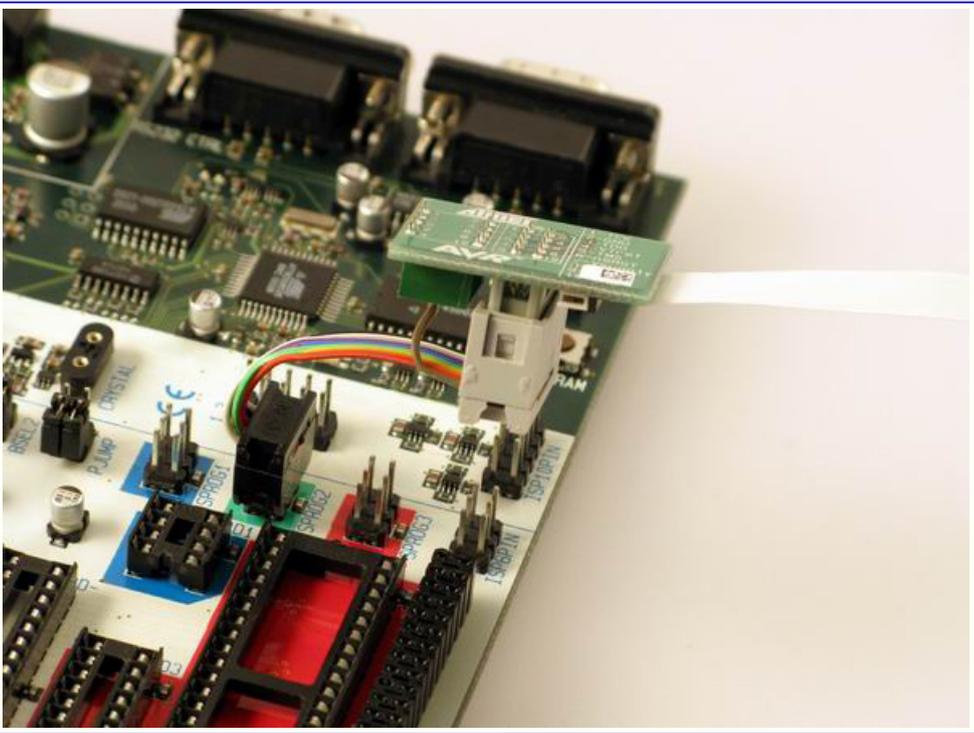
いくつかのSTK500(と初期のJTAGICE mkIIキット)と共に出荷したSTK500 JTAGアダプタはSTK500上のSCKT3100A3とSCKT3000D3と一致するJTAGを持つAVRデバイスに対して簡単にSTK500へ接続するのに使うことができます。



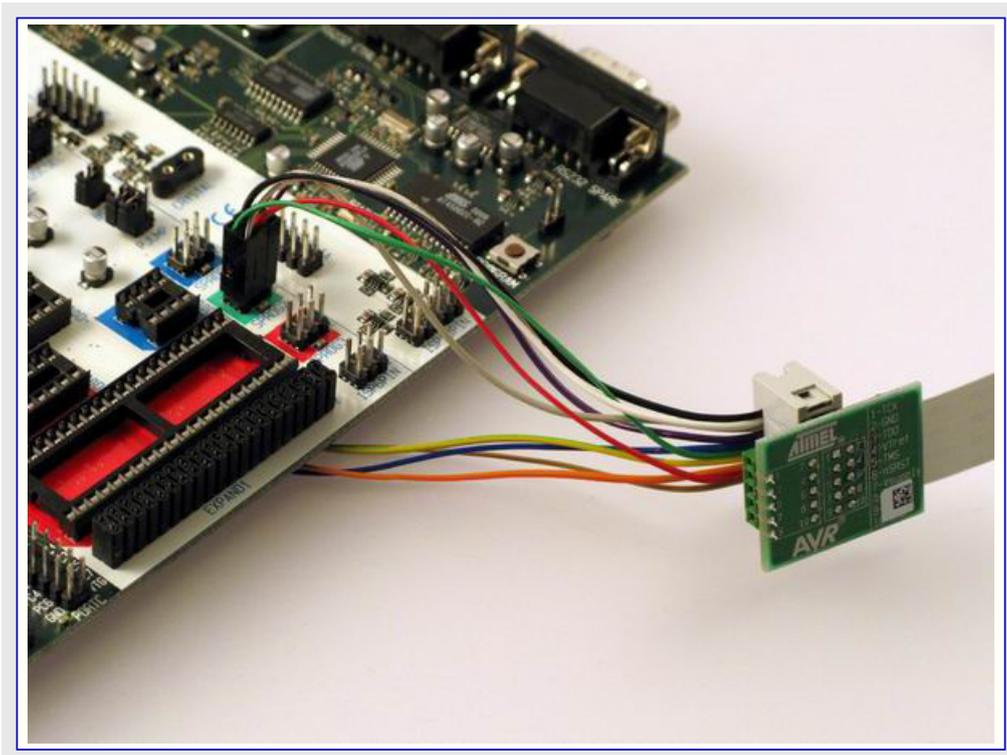
注: 例えばSTK501/502のようなSTK500用アドオンカードは通常専用のJTAGコネクタを持ちます。

利用可能なSTK500 JTAGアダプタを持たない場合、STK500のポートCのPC5~2でデバイスのJTAGポートに直接接続するのに、10ピン多色”パラ線”ケーブルも使うことができます。

デバッグWIREとSPIの目的対象への接続は同じ10ピンまたは6ピンのリボンケーブルを使って行われます。デバッグWIREインターフェース使用时、必要とされるように駆動されることをリセット線に許すために、STK500のRESETジャンパを取り去ることを確実にしてください。



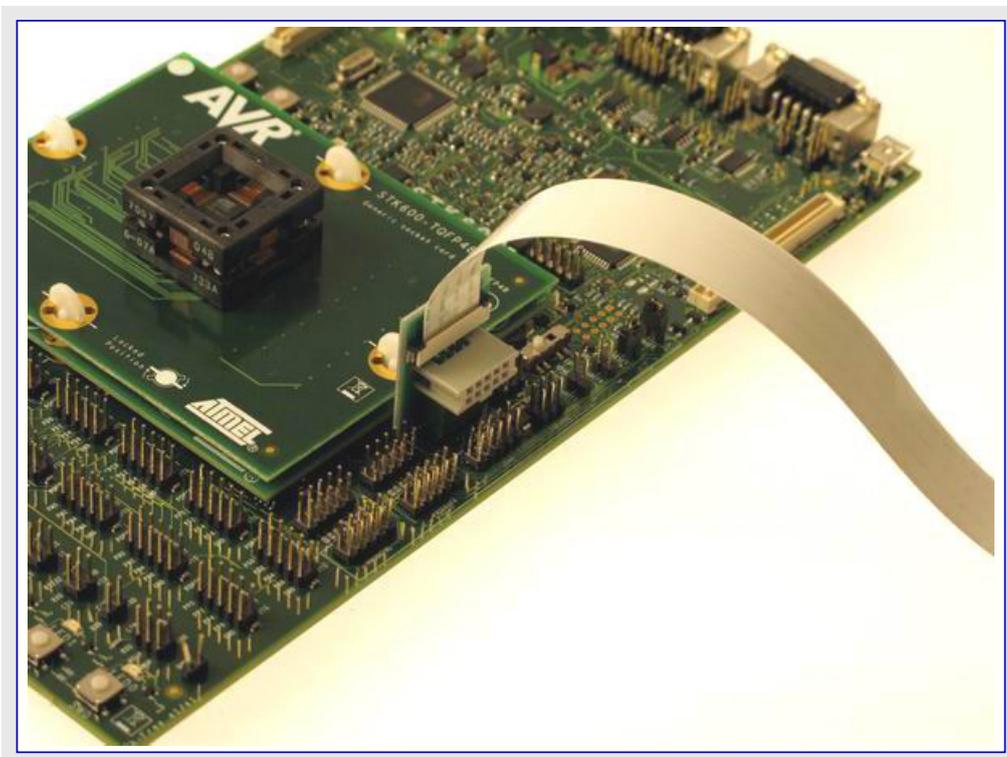
代わりに、JTAGICE mkIIは(提供された)10ピン”バラ線”ケーブルを使ってどの目的対象インターフェースにも接続することができます。



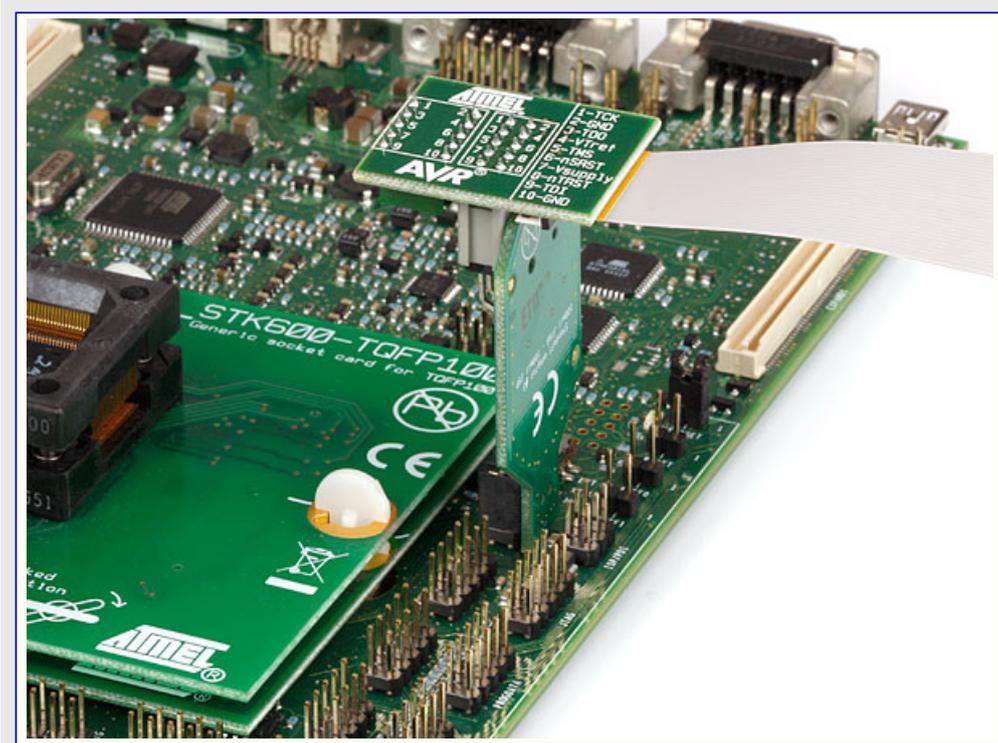
STK500でデバッグWIREインターフェースを使ってデバイスを手元でデバッグする時に、(内部RCを使わない限り)デバイスへのクロック接続を確実にしてください。これは度々 Atmel tinyAVRデバイスに対していくつかのジャンパ(線)が必要です。また、RESET信号が正しく接続され、STK500上のRESETジャンパが取り去られることを確実にしてください。

3.7. Atmel STK600とのAtmel JTAGICE mkIIの使い方

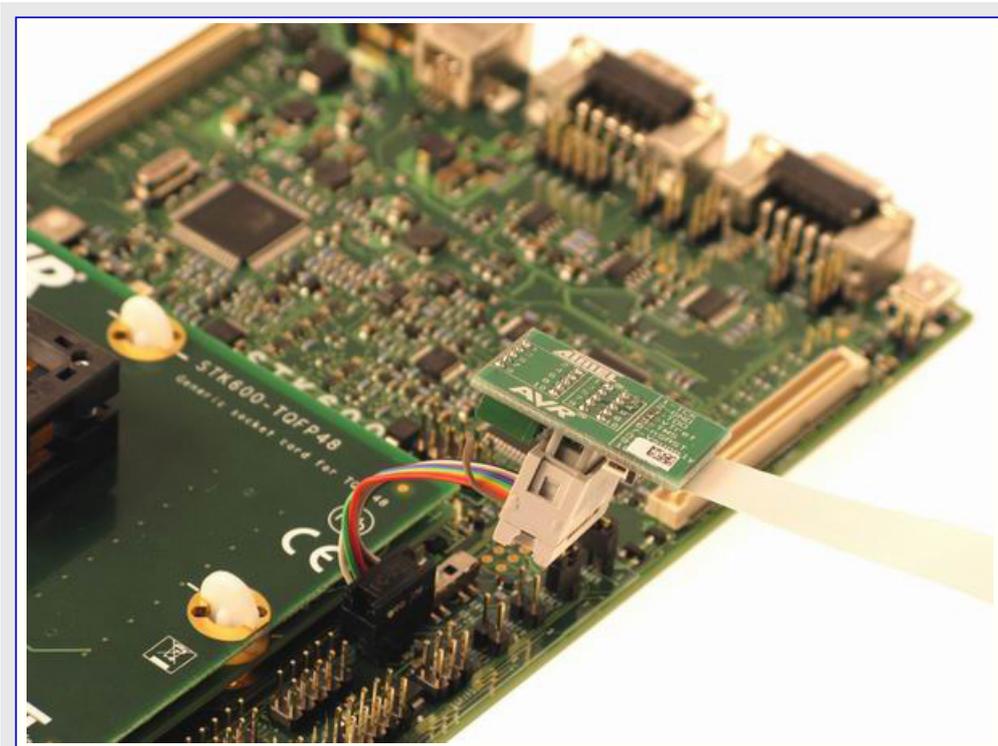
Atmel STK600スタータ キットはJTAG、デバッグWIRE、PDI、SPI、aWireインターフェースを通してAtmel JTAGICE mkIIを接続することができる Atmel AVRデバイスの収容場所に使うことができます。



JTAG目的対象への接続時、単純にJTAGICE mkII探針をSTK600上の内部JTAGピンヘッダ'に接続してください。
PDI目的対象への接続時、SPI/PDIコネクタに接続するのに単純に(提供された)6ピン100milアダプタを使ってください。



デバッグWIREまたはSPIの目的対象への接続時、SPI/PDIコネクタに接続するのに単純に(提供された)10ピン-6ピン100milリボンケーブルを使ってください。



aWire目的対象への接続時、指定されたピン配列に接続するのに(提供された)10ピン多色”バラ線”ケーブルが使われなければなりません。「aWire目的対象への接続」項をご覧ください。

4. チップ上デバッグ

4.1. チップ上デバッグ(OCD)の序説

伝統的なエミュレータは目的対象デバイスの正確な動きを真似ようとする道具です。より近いこの動きは実際のデバイスの動きで、より良い模倣です。

Atmel AVR JTAGICE mkIIは伝統的なエミュレータではありません。代わりに、JTAGICE mkIIはデバイスの実行を監視して制御するための機構を提供する目的対象Atmel AVRデバイス内の内部チップ上デバッグシステム(OCD:On-Chip Debug system)とインターフェースします。この方法ではデバッグをされつつある応用がエミュレートされませんが、現実のAVR目的対象デバイス上で実際に実行されます。

OCDシステムとで、応用は伝統的なエミュレータでは技術的に実現可能ではない何か、即ち目的対象システムで正確な電氣的及びタイミングの特性を保って静かに実行することができます。

走行動作

走行動作時、コードの実行はJTAGICE mkIIと完全に独立です。JTAGICE mkIIは中断条件が発生したかを知るために目的対象AVRを継続的に監視します。これが発生すると、OCDシステムはデバッグインターフェースを通してデバイスに質問し、使用者にデバイスの内部状況を見ることを許します。

停止動作

中断点(ブレークポイント)到達時、プログラム実行は停止されますが、全ての入出力は中断点が起きなかったように走行を継続します。例えば中断点発生時に丁度USART送信が初期化(/設定)されたと仮定します。この場合は例えコアが停止動作であっても、USARTは送信を完了する全速で走行を継続します。

ハードウェア中断点

AVR OCD部はハードウェアで実装された多数のプログラムカウンタ比較器を含みます。プログラムカウンタが比較器レジスタの1つに格納された値と一致すると、OCDは停止動作へ移行します。ハードウェア中断点はOCD部で専用のハードウェアを必要とするため、利用可能な中断点数は目的対象AVRに実装されるOCD部の大きさに依存します。通常、デバッグによって内部使用のために1つのこのようなハードウェア比較器が「予約」されます。様々なOCD部で利用可能なハードウェア中断点のより多くの情報については「[Atmel AVR OCD実装](#)」項をご覧ください。

ソフトウェア中断点

ソフトウェア中断点は目的対象デバイス上のプログラムメモリに配置されたBREAK命令です。この命令が読み込まれると、プログラム実行が中断され、OCDは停止動作へ移行します。実行を継続するには、OCDから「start」命令を与えなければなりません。全てのAVRがBREAK命令を支援するOCD部を持っている訳ではありません。様々なOCD部で利用可能なソフトウェア中断点のより多くの情報については「[Atmel AVR OCD実装](#)」項をご覧ください。

OCDシステム使用時の考慮と制限の更なる情報については「[特別な考慮](#)」章をご覧ください。

4.2. 物理インターフェース

Atmel AVR JTAGICE mkIIはこれ以降の項で記述されるように多数のハードウェアインターフェースを支援します。

4.2.1. JTAG

JTAGインターフェースはIEEE® 1149.1規格に適合する4線検査入出力ポート(TAP:Test Access Port)制御器から成ります。IEEE規格は回路基板の接続性(境界走査)を効率的に検査する工業標準的な方法を提供するために開発されました。Atmel AVRデバイスは完全なプログラミングとチップ上デバッグの支援を含むように拡張されたこの機能を持ちます。

JTAGインターフェースを持つAtmel AVRを含む応用PCB設計時、「[図4-2. JTAGヘッダピン配列](#)」で示されるようなピン配列を使うことが推奨されます。JTAGICE mkIIの100mil探針アダプタはこのピン配列を支援します。

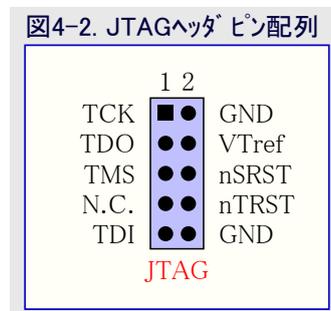
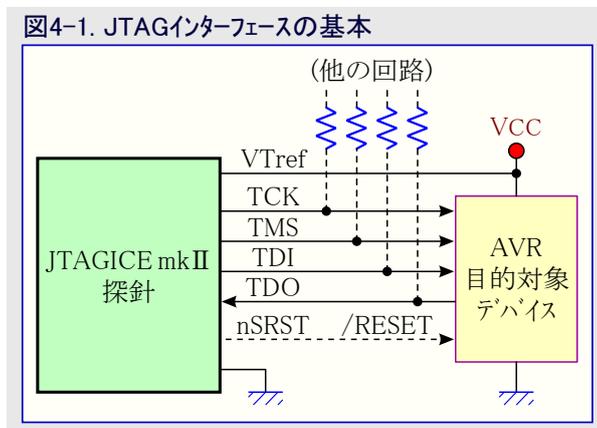


表4-1. JTAGピン説明

| 名前 | ピン番号 | 説明 |
|-------|------|--|
| TCK | 1 | 検査クロック (JTAGICE mkIIから目的対象デバイスへのクロック信号) |
| TMS | 5 | 検査種別選択 (JTAGICE mkIIから目的対象デバイスへの制御信号) |
| TDI | 9 | 検査データ入力 (JTAGICE mkIIから目的対象デバイスへ送出されるデータ) |
| TDO | 3 | 検査データ出力 (目的対象デバイスからJTAGICE mkIIへ送出されるデータ) |
| nTRST | 8 | 検査リセット (任意、いくつかのAVRデバイスのみ)。JTAG TAP制御器のリセットに使用 |
| nSRST | 6 | 元リセット (任意)。目的対象デバイスのリセットに使用。或る筋書でのデバッグを必要とし得る、リセット状態で目的対象デバイスを保持することをJTAGICE mkIIに許すために、このピンの接続が推奨されます。例えば応用ファームウェアによってJTAG禁止(JTD)ビットが設定(1)された場合、JTAGインターフェースを禁止します。nSRSTピンはJTAGICE mkIIで内部プルアップを持ちます。 |
| VTG | 4 | 目的対象基準電圧。JTAGICE mkIIは正しくレベル変換器を給電するために、このピンで目的対象デバイス電圧を採取します。JTAGICE mkIIはこのピンから1mA未満を引き出します。 |
| GND | 2,10 | 接地。JTAGICE mkIIと目的対象デバイスが同じ接地基準を共用するのを保証するために両方が接続されなければなりません。 |

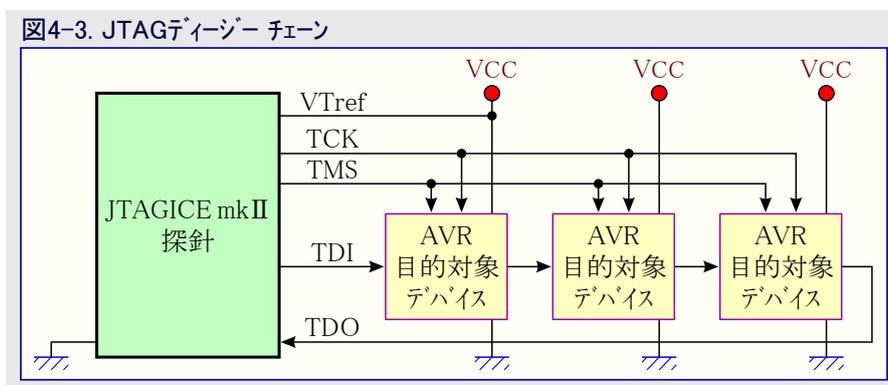
助言: 4番ピンとGND間に雑音分離(デカップ)コンデンサを含むことを覚えて置いてください。

注: JTAGICE mkIIは目的対象によって給電することができません。Vsupply(7番ピン)は**未接続**にして置くべきです。

目的対象応用で外部回路がJTAGデバッグ線を共用する時は、駆動部の衝突を避けるため、**図4-1. JTAGインターフェースの基本**で示されるように直列抵抗が使われるべきです。抵抗の値は外部回路とAVRがそれらの最大定格(即ち、引き込みと流れ出しの過度な電流)を超えないように選ばれるべきです。1kΩが一般的に使われる値です。

これらの要素がJTAG信号によって放電され、コンデンサの残存電圧によって影響を及ぼされて誤った論理レベルを引き起こす可能性のため、JTAG作業中に(抵抗の外側であるべき)これらの線上のどのアナログ濾波器も切断されることが推奨されます。濾波器が切断できない場合、電圧を安定に保つために作業中に目的対象VCCを直接コンデンサに印加することが推奨されます。これを行う時にはコンデンサとJTAG線の間には充分大きな抵抗を必ず使ってください!。

JTAGインターフェースは多数のデバイスに対してデジーチェーン構成設定内で単一インターフェースへ接続することを許します。目的対象デバイスは全てが同じ供給電圧によって給電され、共通接地節を共用しなければならず、「**図4-3. JTAGデジーチェーン**」で示されるように接続されなければなりません。



デジーチェーンでデバイス接続時、以下の点が考慮されなければなりません。

- 全てのデバイスはJTAGICE mkII探針のGNDに接続された共通接地(GND)を共用しなければなりません。
- 全てのデバイスは同じ目的対象電圧で動作しなければなりません。JTAGICE mkII探針のVTrefはチェーン内の最初のデバイスのVCCに接続されなければなりません。
- TMSとTCKは並列で接続されます。TDIとTDOは直列連鎖で接続されます。
- チェーン内のデバイスのどれか1つがJTAGポートを禁止する場合、JTAGICE mkII探針のnSRSTはデバイスのRESETに接続されなければなりません。
- "Devices before"はTDI信号が目的対象デバイスに到達する前にデジーチェーン内を通過しなければならないJTAGデバイス数を参照します。同様に"Devices after"は信号がJTAGICE mkIIのTDOピンに到達する前に目的対象デバイスの後を通過しなければならないデバイス数です。
- "Instruction bits before"と"Instruction bits after"はデジーチェーン内で目的対象デバイスの前後に接続される全てのJTAGデバイスの命令レジスタ(IR)長の総合計を参照します。
- 総IR長(Instruction bits before+Atmel AVR IR長+Instruction bits after)は最大32ビットに制限されます。

デバッグチェーン例：TDI ⇒ ATmega1280 ⇒ **ATxmega128A1** ⇒ ATUC3A0512 ⇒ TDO

Atmel AVR XMEGAデバイスに接続するためのデバッグチェーン設定は次のとおりです。

Devices before : 1

Devices after : 1

Instruction bits before : 4 (AVRデバイスは4ビットのIRを持ちます。)

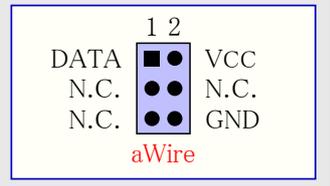
Instruction bits after : 5 (Atmel AVR 32ビット マイクロ コントローラは5ビットのIRを持ちます。)

4.2.2. aWire物性

aWireインターフェースはRESETピンを使う少ピン数のAVR 32ビット デバイスのプログラミングとデバッグ用の単ピンインターフェースです。JTAGインターフェースを通して利用可能なOCDシステムの全ての機能はaWireを使ってアクセスすることもできます。

aWireインターフェースを持つAVRを含む応用PCBの設計時、**図4-4. aWireヘッダピン配列**で示されるピン配列が使われるべきです。将来のaWire能力を持つツールがこのピン配列を支援する一方で、Atmel JTAG ICE mkIIはこのピン配列に割り当てるために10ピンの”バラ線”ケーブルの使用が必要です。

図4-4. aWireヘッダピン配列



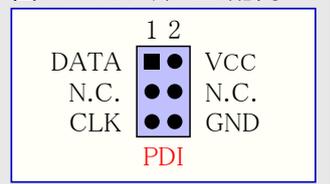
4.2.3. PDI物性

プログラミング/デバッグ用インターフェース(PDI:Program and Debug Interface)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専有インターフェースです。PDI物性は目的対象デバイスとの双方向半二重同期通信を提供する2ピン インターフェースです。

PDIインターフェースを持つAtmel AVRを含む応用PCBの設計時、**図4-5. PDIヘッダピン配列**で示されるピン配列が使われるべきです。そしてJTAGICE mkII探針を応用PCBへ接続するのに、Atmel AVR JTAG ICE mkIIキットと共に提供される6ピン アダプタの1つを使うことができます。

注: 右で示されるピン配列はAtmel STK600、Atmel AVR ONE!, AVR Dragon、そしてPDIインターフェースを持つ将来のツールと一致します。JTAGICE mkIIは(キットと共に出荷された、または最寄りのAtmel 代理店から入手可能な)JTAGICE mkII用のAVR XMEGA PDIアダプタを使ってこのピン配列を支援します。

図4-5. PDIヘッダピン配列



4.2.4. デバッグWIRE

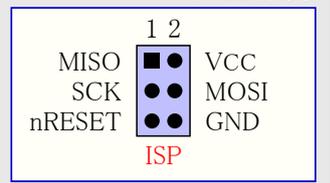
デバッグWIREインターフェースは少ピン数デバイスで使うためにAtmelによって開発されました。4ピンを使うJTAGインターフェースとは異なり、デバッグWIREはデバッグ ツールと共に双方向半二重非同期通信に単一(RESET)ピンだけを利用します。

デバッグWIREインターフェースを持つAtmel AVRを含む応用PCBの設計時、**図4-6. デバッグWIRE(SPI)ヘッダピン配列**で示されるピン配列が使われるべきです。

注: デバッグWIREインターフェースはプログラミング インターフェースとして使うことができません。これは目的対象をプログラミングするために、(図4-7. SPIヘッダピン配列で示されるような)SPIインターフェースも利用可能でなければならないことを意味します。

デバッグWIRE許可(DWEN)ヒューズがプログラム(0)され、施錠ビットが非プログラム(1)の時に、目的対象内のデバッグWIREシステムが活性(有効)にされます。RESETピンはプルアップ許可のワイヤードAND(オープンドレイン)双方向入出力ピンとして構成設定され、目的対象とデバッグ間の通信中継器になります。

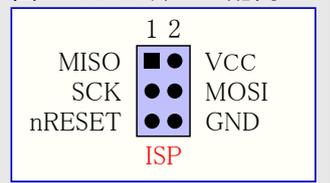
図4-6. デバッグWIRE(SPI)ヘッダピン配列



4.2.5. SPI

実装書き込み(ISP:In-System Programming)はフラッシュ メモリとEEPROM内にコードを書き込むのに目的対象のAtmel AVRの内部SPI(Serial Peripheral Interface)を使います。これはデバッグ インターフェースではありません。SPIインターフェースを持つAVRを含む応用PCBの設計時、**図4-7. SPIヘッダピン配列**で示されるピン配列が使われるべきです。

図4-7. SPIヘッダピン配列



4.3. Atmel AVR OCD実装

4.3.1. Atmel AVR UC3 OCD (JTAGとaWire)

Atmel AVR UC3 OCDシステムは高い柔軟性と強力で開かれた32ビット マイクロ コントローラ用のチップ[®]上デバッグ規格であるNexus 2.0規格 (IEEE[®]-ISTO 5001-2003)に従って設計されています。これは以下の機能を支援します。

- Nexus適合デバッグ解決策
- どのCPU速度も支援するOCD
- 6つのプログラム カウンタ ハードウェア中断点(ブレイク ポイント)
- 2つのデータ中断点
- 監視点として構成設定できる中断点
- 範囲での中断を与えるように結合することができるハードウェア中断点

このデバッグ インターフェースに関するより多くの情報については「[特別な考慮](#)」をご覧ください。

AVR UC3 OCDシステムに関するより多くの情報についてはwww.atmel.com/uc3に置かれているAVR32UC技術参考書を調べてください。

4.3.2. Atmel AVR XMEGA OCD (JTAGとPDI物性)

Atmel AVR XMEGA OCDは他にPDI(Program and Debug Interface)として知られます。(JTAGとPDI物性の)2つの物理インターフェースがデバイス内で同じOCD実装へのアクセスを提供します。これは以下の機能を支援します。

- 完全なプログラムの流れ制御
- 1つの専用プログラム アドレス比較器またはシンボリック中断点(予約)
- 4つのハードウェア比較器
- (BREAK命令を使う)無制限数の使用者プログラム中断点
- システム クロック周波数での制限なし

このデバッグ インターフェースに関するより多くの情報については「[特別な考慮](#)」をご覧ください。

4.3.3. Atmel megaAVR OCD (JTAG)

Atmel megaAVR OCDはJTAG物理インターフェースに基づきます。これは以下の機能を支援します。

- 完全なプログラムの流れ制御
- 4つのプログラム メモリ (ハードウェア)中断点 (1つは予約)
- データ中断点形式に結合することができるハードウェア中断点
- (BREAK命令を使う)無制限数のプログラム中断点 (ATmega128[A]を除く)

このデバッグ インターフェースに関するより多くの情報については「[特別な考慮](#)」をご覧ください。

4.3.4. Atmel megaAVR/tinyAVR OCD (デバッグWIRE)

デバッグWIRE OCDは小ピン数のAVRデバイス用に特に設計された制限された機能一式を持つ特殊化したOCD部です。これは以下の機能を支援します。

- 完全なプログラムの流れ制御
- (BREAK命令を使う)無制限数の使用者プログラム中断点
- 目的対象クロックに基づく自動転送速度構成設定

このデバッグ インターフェースに関するより多くの情報については「[特別な考慮](#)」をご覧ください。

5. ハードウェア説明

5.1. 物理的寸法

- 主本体 : 140mm×110mm×30mm
- 探針ケーブル : 180mm
- 探針 : 「[探針](#)」に対する記述をご覧ください。

5.2. LED

Atmel AVR JTAGICE mkIIは現在のデバッグまたはプログラミングの作業の状態を示す3つのLEDを持ちます。

表5-1. LED

| LED | 位置 | 説明 |
|-----------------|----|---|
| 目的対象電源 | 1 | 目的対象基板の電源がONの時に緑 |
| JTAGICE mkII 電源 | 2 | JTAGICE mkII本体が給電される時に赤。点滅はUSBハブが必要とされる電力量を割り当てられないことを示します。 |
| 状態 | 3 | 緑 : データ転送。緑点滅は目的対象走行を示します。 橙 : ファームウェア更新また初期化 赤 : アイドル、未接続 消灯 : アイドル、接続中 |



5.3. 背面

Atmel AVR JTAGICE mkIIの背面はDCジャック、電源スイッチ、USBとRS-232のコネクタを収容します。

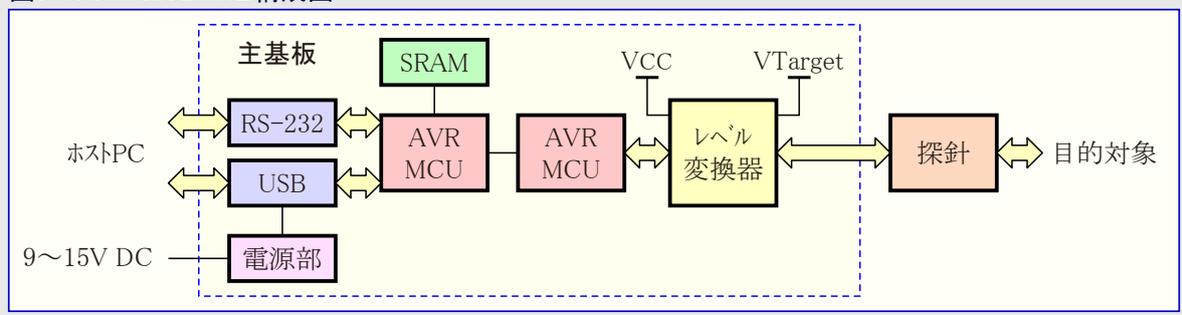
通番は本体底面の張り紙で示されます。



5.4. 基本構造説明

Atmel AVR JTAGICE mkIIの基本構造は図5-1. JTAGICE mkII構成図の構成図で示されます。

図5-1. JTAGICE mkII構成図

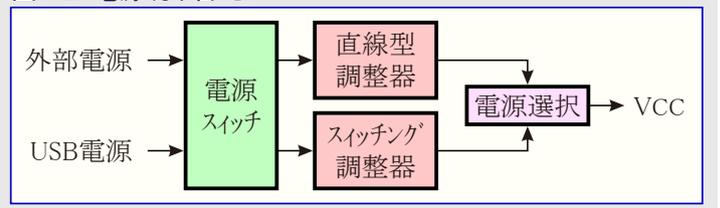


5.4.1. 電源

Atmel AVR JTAGICE mkIIの電源は右図で示されるように実装されます。JTAGICE mkIIの背面で見つかる電源スイッチは、外部とUSBの両方の電力をONに切り替え、内部スイッチはどの電源を使うかを選びます。外部電源は、それが十分な電力を提供する場合に既定で選ばれます。

注: JTAGICE mkIIは目的対象応用から給電できません。

図5-2. 電源切り替え

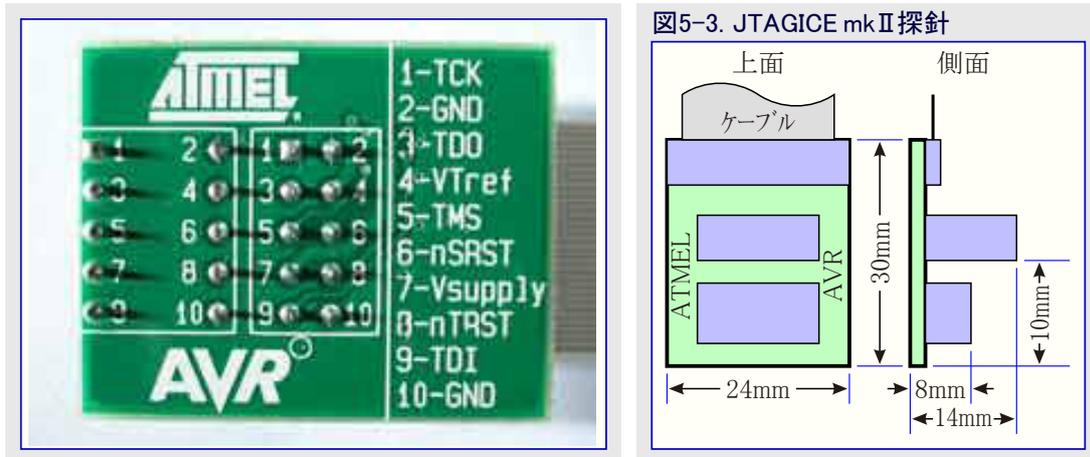


5.4.2. レベル変換器

レベル変換器の目的はAtmel AVR JTAGICE mkIIそれ自身と異なる電圧で走行する目的対象基板との成功裏の通信を提供することです。レベル変換器は1.65~5.5Vの目的対象電圧を支援するように設計されています。

5.4.3. 探針

目的対象アダプタが下で表されます。20芯のフレキシブル ケーブルが目的対象アダプタをAtmel AVR JTAGICE mkIIに接続します。目的対象アダプタは同じピン配列と信号を持つ2つの10ピン コネクタを持ちます。目的対象基板に最良適合する方を使ってください。どんな時でもコネクタの1つだけが接続されるべきです。



探針を目的対象応用にご接続するかの更なる情報については「[Atmel JTAGICE mkII の接続](#)」章をご覧ください。

6. ソフトウェア統合

6.1. Atmel Studio

6.1.1. Atmel Studio

Atmel StudioはVisual Studio®シェルに基づいてAtmel AVR 8ビットと32ビット マイクロ コントローラ用の応用を開発するための統合開発環境 (IDE: Integrated Development Environment)です。Atmel Studioが提供するための何を持つのかの更なる情報についてはAtmel Studioのヘルプをご覧ください。

Atmel AVR JTAGICE mkIIはAVR Studio 4とAtmel Studioにも適合します。

6.1.2. Atmel StudioプログラミングGUI

Atmel AVR JTAGICE mkIIはAtmel Studio IDEの一部であるGUI環境を使ってAtmel AVRデバイスをプログラミングするのに使うことができます。

6.1.3. プログラミング任意選択

Atmel StudioはAtmel AVR JTAGICE mkIIを使ってAtmel AVRデバイスのプログラミングを支援します。プログラミング ダイアログは選択した目的対象デバイスに従ってJTAG、aWire、SPI、PDIの動作を使うように構成設定することができます。

クロック周波数はSPIインターフェースに対してだけ構成設定することができます。JTAG、aWire、PDIのインターフェースについてはプログラミングが目的対象のクロックと独立して行われ、故に構成設定が必要とされません。

6.1.4. デバッグ任意選択

Atmel Studioを使ってAtmel AVRデバイスをデバッグする時に、プロジェクト プロパティ ウィンドウ内の 'Tool' タブはいくつかの重要な構成設定任意選択を含みます。更なる説明が必要な任意選択は以下です。

- 目的対象クロック周波数(Target Clock Frequency)

目的対象クロック周波数：JTAGインターフェースでAtmel megaAVRデバイスの信頼に足るデバッグを達成するには目的対象クロック周波数の正確な設定が重要です。この設定はデバッグされる応用の目的対象AVRデバイスの最低動作周波数に反映されます。より多くの情報については「[特別な考慮](#)」章をご覧ください。

デバッグWIRE目的対象デバイスでのデバッグ作業は目的対象デバイスそれ自身によってクロック駆動され、故に周波数設定は全く必要とされません。Atmel AVR JTAGICE mkIIはデバッグ作業の開始で通信用の正しいポーレートを自動的に選択します。

aWireインターフェースを使うデバッグ時、JTAGICE mkIIはポーレートを最適値に調整し、使用者設定は全く必要とされません。

Atmel AVR XMEGA目的対象デバイスはJTAGに対して最大速度、PDIに対して固定の1MHzクロックでクロック駆動されます。XMEGAデバイスは全く同期が必要とされず、故にクロックは構成設定不可です。

- EEPROM保護(Preserve EEPROM)

デバッグ作業前に目的対象の再書き込み中にEEPROMの消去を避けるにはこの任意選択を選択してください。

- デバイスプログラミング時に外部リセット常時活性(Always activate external reset when reprogramming device)

目的対象がJTAGインターフェースを禁止する場合、プログラミング中に外部リセットがLowに引かれなければなりません。外部リセットを使うかどうかを繰り返し問われるのを避けるにはこの任意選択を選択してください。

7. コマンド行ユーティリティ

Atmel StudioはJTAGICE mkIIを使って目的対象をプログラミングするのに使うことができるatprogramと呼ばれるコマンド行ユーティリティと共に来ます。Atmel Studioのインストール中にスタートメニューのAtmelフォルダ内にAtmel Studio 7.0 Command Promptと呼ばれるショートカットが作成されます。このショートカットのダブルクリックにより、コマンドプロンプトが開き、プログラミング命令を入力することができます。コマンド行ユーティリティはAtmel Studioインストールパス内のAtmel¥Atmel Studio 7.0¥atbackend¥フォルダ内にインストールされます。

コマンド行ユーティリティのより多くのヘルプを得るには以下の命令を入力してください。

```
atprogram --help
```

8. 特別な考慮

8.1. Atmel AVR XMEGA OCD

OCDとクロック駆動

MCUが停止動作に移行すると、OCDクロックはMCUクロックとして使われます。OCDクロックはJTAGインターフェースが使われている場合はJTAG TCK、またはPDIインターフェースが使われている場合はPDI_CLKのどちらかです。

Atmel AVR JTAGICE mkIIはAtmel AVR XMEGA目的対象に対して可変クロック速度を提供しません。

停止動作でのSDRAMリフレッシュ

OCDが停止動作の時は上の項で記述されるようにMCUはPDIまたはJTAGのクロックによってクロック駆動されます。デバッグまたはOCDによってこの周波数は何も知られていないため、自動的に低リフレッシュ周期(0x10)が使われます。この値は使用者によって変更することはできません。

停止動作でのI/O単位部

殆どのAtmel megaAVRデバイスと異なり、XMEGAのI/O単位部は停止動作で停止されます。これはUSART送信が中断され、計時器(とPWM)が停止されることを意味します。

ハードウェア中断点

2つのアドレス比較器と2つの値比較器で4つのハードウェア中断点(ブレイクポイント)比較器があります。これらは以下のような一定の制限を持ちます。

- 全ての中断点は同じ形式(プログラムまたはデータ)でなければなりません。
- 全てのデータ中断点は同じメモリ領域(I/O、SRAM、またはXRAM)でなければなりません。
- アドレス範囲が使われる場合は1つの中断点だけができます。

これは設定することができる各種組み合わせです。

- 2つの単独データまたはプログラムのアドレス中断点
- 1つのデータまたはプログラムのアドレス範囲中断点
- 単一値比較を持つ2つの単独データアドレス中断点
- アドレス範囲、値範囲、またはその両方を持つ1つのデータ中断点

中断点が設定できない場合にAtmel Studioは何故かを告げます。ソフトウェア中断点が利用可能の場合、データ中断点がプログラム中断点を超える優先権を持ちます。

外部リセットとPDI物性

PDI物理インターフェースはクロックとしてリセット線を使います。デバッグ中、リセットのプルアップは10kΩまたはそれ以上か、または全てを取り去られるべきです。どんなリセットコンデンサも取り去られるべきです。他の外部リセット元は切断されるべきです。

8.2. Atmel megaAVR OCDとデバッグWIRE OCD

I/O周辺機能

例え中断点(ブレイクポイント)によってプログラム実行が停止されても、殆どのI/O周辺機能は動作を続けます。例:UART送信中に中断点に到達した場合、その送信は完了されて対応するビットが設定されます。例え実施のデバイスに於いて普通はもっと後で起こっても、送信完了(TXC)フラグが設定(1)されて次のコードの単一段階実行(シングルステップ)で利用可能になります。

以下の2つの例外を除いて全てのI/O周辺機能は停止動作で動作を続けます。

- タイマ/カウンタ (ソフトウェア前処理部を使って構成設定可能)
- ウォッチドッグ タイマ (デバッグ中のリセットを防ぐため、常に停止されます。)

単一段階実行I/Oアクセス

停止動作でI/Oが動作を続けるため、或るタイミングの問題を避けることに注意が払われるべきです。例えば次のコードです。

```
OUT    PORTB, $AA
IN     TEMP, PINB
```

このコードを普通に走行すると、IN動作によって採取される時にデータが物理的に未だピンにラッチされていないので、TEMPレジスタは\$AAを読み戻さないでしょう。PINレジスタに正しい値が存在することを保証するには、OUTとINの命令間にNOP命令が配置されなければなりません。

けれども、OCDを通してこの関数を単一段階実行(シングル ステップ)すると、例え単一段階実行中にコードが停止されていても、I/Oが全速で動くため、このコードは常にPINレジスタで\$AAを与えます。

単一段階実行とタイミング

或るレジスタは制御信号許可後に与えられた周期数内で読みまたは書きされることが必要です。停止動作でI/Oクロックと周辺機能は全速で動作を続けるため、そのようなコード全体を通しての単一段階実行(シングル ステップ)はタイミングの必要条件に合いません。2つの単一段階実行間で、I/Oクロックは100万回の周期を走行するかもしれません。そのようなタイミングの必要条件を持つレジスタの読みまたは書きを成功するには、デバイスを全速で走行して非分断動作として読みまたは書きの手順が実行されるべきです。これはコードを実行するのにマクロまたは関数呼び出しを使うことによって行われるか、またはデバッグ環境で「カーソルまで走行」機能を使うことができます。

16ビット レジスタのアクセス

Atmel AVR周辺機能は代表的に8ビット データ バス経由でアクセスすることができる多数の16ビット レジスタ(例えば、16ビット タイマ/カウンタのTCNTn)を含みます。16ビット レジスタは2つの読みまたは書きの操作を使ってバイト アクセスされなければなりません。16ビット アクセスの中間での中断、またはこの状況全体を通しての単一段階実行(シングル ステップ)は誤った値に帰着し得ます。

制限されるI/Oレジスタ アクセス

或るレジスタはそれらの内容に影響を及ぼさずに読むことができません。このようなレジスタは読むことによって解除(0)されるフラグを含むそれら、または緩衝されたデータレジスタ(例えば、UDR)を含みます。ソフトウェア前処理部はOCDデバッグの意図された邪魔しない性質を守るため、停止動作時にこれらのレジスタの読み込みを防ぎます。加えて、いくつかのレジスタは副作用の発生なしに安全に書くことができません。それらのレジスタは読み込み専用です。例えば以下です。

- どれかのビットに'1'を書くことによって解除(0)されるフラグがあるフラグ レジスタ。これらのレジスタは読み込み専用です。
- UDRとSPDRのレジスタは単位部の状態に影響を及ぼさずに読むことができません。これらのレジスタはアクセス不可です。

8.3. Atmel megaAVR OCD (JTAG)

ソフトウェア中断点

これが初期のOCD単位部を含むため、ATmega128[A]はソフトウェア中断点(ブレーク ポイント)用のBREAK命令の使用を支援しません。

JTAGクロック

目的対象クロック周波数はデバッグ作業を開始する前にソフトウェア前処理部で正確に指定されなければなりません。同期の理由に関して、信頼に足るデバッグのためにJTAG TCK信号は目的対象クロック周波数の1/4未満でなければなりません。高すぎる目的対象クロック周波数設定はプログラミング完了のすぐ後でのデバッグ作業の失敗を引き起こすでしょう。これは表示される多数の偽のSLEEP、WAKE UP、IDRのメッセージが伴われるかもしれません。JTAGインターフェース経由でのプログラミング時、TCK周波数は実際に使われるクロック周波数ではなく、目的対象デバイスの最大周波数定格によって制限されます。

内部RC発振器使用時、この周波数がデバイス間で変わり得て、温度とVCCの変化によって影響を及ぼされることに注意してください。目的対象クロック周波数を指定する時に控え目にしてください。ソフトウェア前処理部を使って目的対象クロック周波数を設定する方法の詳細については「ソフトウェア統合」をご覧ください。

JTAGENとOCDENのヒューズ

JTAGインターフェースは既定でプログラム(0)にされているJTAGENヒューズを使って許可されます。これはJTAGプログラミング インターフェースへのアクセスを許します。この機構を通して、OCDENヒューズをプログラム(0)にすることができます(既定のOCDENは非プログラム(1))。これはデバイスのデバッグを容易にするためのOCDへのアクセスを許します。ソフトウェア前処理部は作業を終える時にOCDENヒューズを常に非プログラム(1)にして置くことを保証し、それによってOCD単位部による不要な電力消費を制限します。JTAGENヒューズが予期せず禁止された場合、SPIまたはPPでのプログラミング法を使って再許可することだけができます。

JTAGENヒューズがプログラム(0)される場合、JTAGインターフェースはファームウェアでJTAG禁止(JTD)ビットを設定(1)することによって未だ禁止することができます。これはコードをデバッグ不能にし、デバッグ作業を試みる時に行われるべきではありません。デバッグ作業開始時にこのようなコードがAtmel AVRデバイスで既に実行した場合、Atmel AVR JTAGICE mkIIは接続している間、RESET線を活性化にします。この線が正しく配線されているなら、目的対象AVRデバイスをリセットに強制し、それによってJTAG接続を許します。

JTAGインターフェースが許可される場合、JTAGピンは交換ピン機能に使うことができません。これらはプログラム コードからJTDビットを設定(1)することによってJTAGインターフェースが禁止されるか、またはプログラミング インターフェースを通してJTAGENヒューズが解除(1)されることによってかのどちらかまで専用JTAGピンに留まります。

IDR事象

応用プログラムがデバッグされつつあるAVRデバイスのOCDRレジスタにデータのバイトを書くと、JTAGICE mkIIはこの値を読み取ってソフトウェア前処理部のメッセージ ウィンドウにそれを表示します。IDRレジスタは100ms毎にポーリングされ、故にそれへのより高い頻度での書き込みは信頼に足る結果を生じないでしょう。デバッグ中にAVRデバイスが電力を失うと、偽のIDR事象が報告され得ます。これは目的対象電圧がAVRの最小動作電圧以下に落ちる時にJTAGICE mkIIが未だポーリングをし得るために起きます。

8.4. デバッグWIRE OCD

デバッグWIRE通信(dW)ピンは物理的に外部リセット(RESET)と同じピンに配置されます。従って、デバッグWIREインターフェースが許可された時は外部リセット元が支援されません。

デバッグWIRE許可(DWEN)ヒューズはデバッグWIREが機能するために目的対象デバイスで設定(0)されなければなりません。このヒューズはAtmel AVRデバイスが工場から出荷される時に既定によって非プログラム(1)にされます。デバッグWIREインターフェース自身はこのヒューズを設定するのに使うことができません。DWENヒューズを設定するには、SPI動作が使われなければなりません。ソフトウェア前処理部は必要なSPIピンが接続されていれば自動的にこれを処理します。Atmel Studioのプログラミング ダイアログからもSPIプログラミングを使って設定することができます。

• 以下か、

デバッグWIRE部でデバッグ作業を開始を試みてください。デバッグWIREインターフェースが許可されていない場合は、Atmel Studioは再試行を提案するか、またはSPIプログラミングを使ってデバッグWIREの許可を試みるでしょう。完全なSPIヘッダ接続があれば、デバッグWIREが許可され、目的対象の電源OFF/ONを尋ねられます。これはヒューズ変更を有効にするのに必要とされます。

• または、

SPI動作でプログラミング ダイアログを開き、識票が正しいデバイスと一致していることを確認してください。デバッグWIREを許可するためにDWENヒューズに選択印を付けて(をチェックして)ください。

注: SPI許可(SPIEN)ヒューズがプログラム(0)にされ、リセット禁止(RSTDISBL)ヒューズがプログラム(0)にされて置くことが重要です!。これを行わないことはデバッグWIRE動作でデバイスを固着状態にし、DWENヒューズを戻すために高電圧プログラミングが必要とされます。

デバッグWIREインターフェースを禁止するには、DWENヒューズを非プログラム(1)にするのに高電圧プログラミングを使ってください。代わりに、SPIENヒューズが設定(0)されているなら、SPIプログラミングに実行を許すために、自身を一時的に禁止するのにデバッグWIREそれ自身を使ってください。

注: SPIENヒューズがプログラム(0)にされて置かれなければ、Atmel Studioはこの操作を完了すつことができず、高電圧プログラミングが使われなければなりません。

• デバッグ作業中、'Debug'メニューから'Disable debugWIRE and Close'メニュー任意選択を選んでください。デバッグWIREは一時的に禁止され、Atmel StudioはDWENヒューズを非プログラム(1)にするのにSPIプログラミングを使います。

プログラム(0)にされたDWENヒューズを持つことは全ての休止動作形態で走行すべき、クロックシステムのいくつかの部分の許します。これは休止動作中のAVRの電力消費を増します。従って、DWENヒューズはデバッグWIREが使われない時に、常に禁止されるべきです。

デバッグWIREが使われる目的対象PCBの設計時、正しい動作のために以下の考慮が行われなければなりません。

- dW/(RESET)線上のプルアップ抵抗は10kΩよりも小さく(強く)てはなりません。デバッグがこれを提供するため、デバッグWIRE機能に関してプルアップ抵抗は必要とされません。
- VCCへのRESETピン直接接続はデバッグWIREインターフェースを失敗にさせます。
- RESETピンに接続されたどの安定用コンデンサも、それがインターフェースの正しい動作を妨げるため、デバッグWIRE使用時に切断されなければなりません。
- RESET線上の全ての外部リセット元または他の活性な駆動部は、それらがインターフェースの正しい動作を妨げるため、切断されなければなりません。

目的対象デバイスの施錠ビットは決してプログラム(0)にしてはなりません。デバッグWIREインターフェースは正しく機能するために施錠ビットが解除(1)されていることが必要です。

8.5. Atmel AVR UC3 OCD

JTAGインターフェース

いくつかのAtmel AVR UC3マイクロ コントローラで、JTAGポートは既定で許可されていません。これらのデバイスの使用時、Atmel AVR JTAGICE mkIIがJTAGインターフェースを許可できるようにRESET線を接続することが重要です。

RESETピンに接続されたどの安定用コンデンサも、それがインターフェースの正しい動作を妨げるため、aWire使用時に切断されなければなりません。

aWireインターフェース

データがそれら2つの領域間で同期されなければならないため、aWire通信のホーレートはシステム クロックの周波数に依存します。JTAGICE mkIIはシステム クロックが低かったことを自動的に検出してそれによってホーレートを再校正します。自動校正は8kHzのシステム クロック周波数への低下でだけ動きます。デバッグ作業中のより低いシステム クロックへ切り替えは目的対象との接触を失わせるかもしれません。

必要とされるなら、aWireホーレートはツール チェーンでaWireパラメータを設定することによって制限することができます。自動検出は未だ動きますが、上限値は結果を強いるでしよ

停止休止動作

いくつかのAVR UC3デバイスは1.8V調整されたI/O線を持つ3.3V供給動作で使うことができる内部調整器を持ちます。これは内部調整器がコアと殆どのI/Oの両方に給電することを意味します。JTAGICE mkIIはこの調整器が停止する停止休止動作を支援しません。換言すればこの休止動作はデバッグ中に使えません。デバッグ中にこの休止動作を使うことが必要条件なら、代わりにAtmel AVR ONE!デバッグを使ってください。

9. 障害対策

9.1. 障害対策の手引き

表9-1. 障害対策の手引き

| 問題 | 有りそうな原因 | 解決策 |
|--|--|--|
| JTAGデバッグ開始、その後突然に失敗。 | <ol style="list-style-type: none"> 1. Atmel AVR JTAGICE mkIIが十分に給電されていない。 2. MCUC(S)レジスタ内のJTAG禁止ビットが応用によってうっかり書かれてしまった。 3. 同期が失われた。 | <ol style="list-style-type: none"> 1. JTAGICE mkIIがUSBだけから給電されている場合、USBは500mA供給できることが必要とされます。 2. 制御を取り戻すためにRESETをLowに保持してJTAG禁止ビットが書かれないようにコードを変更してください。 3. JTAGICE mkIIと目的対象基板を電源OFF/ONしてください。PCとJTAGICE mkII間の通信速度を下げるが必要とされるかもしれません。 |
| デバイスにコードを書き込むのにJTAGICE mkII使用後、もはやエミュレータが動かない。 | <ol style="list-style-type: none"> 1. JTAG許可(JTAGEN)ヒューズが禁止されてしまっている。 2. プログラミング インターフェースが未だ活性。OCDとプログラミングの両方を同時に使うことは不可能です。 | <ol style="list-style-type: none"> 1. JTAGENヒューズをプログラム(0)にしてください。 2. プログラミング インターフェースを閉じてその後にエミュレータ動作へ移行してください。 |
| JTAGICE mkIIがAtmel Studioまたは他のソフトウェア前処理部によって検出されるが、目的対象デバイスへ接続しない。 | <p>JTAG : JTAGENヒューズがプログラム(0)にされていない。</p> <p>デバッグWIRE : デバッグWIRE許可(DWEN)ヒューズがプログラム(0)にされていない。</p> | <p>JTAG : JTAGENヒューズをプログラム(0)にするのに他のプログラミング インターフェースを使ってください。</p> <p>デバッグWIRE : DWENヒューズをプログラム(0)にするのに他のプログラミング インターフェースを使ってください。</p> |
| Atmel Studioが電圧不在メッセージを出す。 | <ol style="list-style-type: none"> 1. 目的対象基板上に電力なし。 2. VTrefが接続されていない。 3. 目的対象電圧が低すぎる。 | <ol style="list-style-type: none"> 1. 目的対象基板に電力を印加してください。 2. JTAGコネクタがVTref信号を含むのを確実にしてください。 3. 目的対象電源が十分な電力を提供できることを確実にしてください。 |
| OCD許可(OC DEN)ヒューズが禁止されるが、JTAGICE mkIIを使ってOCDが未だ可能。 | JTAGICE mkIIはOCDが禁止されている場合にOC DENヒューズを自動的にプログラム(0)にします。 | これは正しい動作です。 |
| Atmel StudioのI/OウィンドウでいくつかのI/Oレジスタが正しく更新されない。 | 邪魔しない読み戻しが不可能な時にJTAGICE mkIIはAtmel StudioのI/Oウィンドウでこの位置を更新しません。 | このI/O位置を一時レジスタ内に読んで、デバッグ中にそれを見てください。これによってどのレジスタが影響を及ぼされるかについての情報に関しては「特別な考慮」章をご覧ください。 |
| Atmel STK500とAtmel STK502とでのATmega169デバッグが外部クロック使用時に動かない。 | STK502上のTOSCスイッチがTOSC位置になっている。 | STK502基板でこのスイッチをXTAL位置に設定してください。 |
| ソフトウェア中断点(ブレークポイント)使用後時々目的対象が”固まって”正しく動かない。 | JTAGICE mkIIのデバッグ作業が正しく閉じられず、フラッシュメモリ領域に未だBREAK命令が存在。 | JTAGICE mkIIのデバッグ作業を正しく閉じることを確実にするか、または正しいhexファイルでフラッシュメモリを再書き込みしてください。 |
| デバッグWIREエミュレート開始後OK、その後突然に失敗。 | <ol style="list-style-type: none"> 1. JTAGICE mkIIが十分に給電されていない。 2. 駆動部または容量(コンデンサ)がRESET線上の通信を邪魔する。 3. 同期が失われた。 | <ol style="list-style-type: none"> 1. JTAGICE mkIIがUSBだけから給電されている場合、USBは500mA供給できることが必要とされます。 2. RESET線上の全ての駆動部または容量(コンデンサ)を取り去ってください。どのプルアップ抵抗も10kΩよりも大きいことを確実にしてください。 3. JTAGICE mkIIと目的対象基板で電源をOFF/ONしてください。PCとJTAGICE mkII間の通信速度を下げるが必要とされるかもしれません。 |
| デバッグWIRE作業後のSPIプログラミングが不可能。 | デバッグWIREが許可されると、SPIインターフェースは禁止される。 | 「特別な考慮」の「デバッグWIRE OCD」項で記述されるようにSPIインターフェースを再許可してください。SPIインターフェースを許可するのにコマンド行ソフトウェアを使ってください。 |
| SPIとデバッグWIREのどちらの接続も動かない。 | SPIとデバッグWIREのインターフェースが禁止されている。デバッグWIREは施錠ビットがプログラム(0)されている場合に動きません。 | 高電圧プログラミングで目的対象に接続してください。SPIまたはデバッグWIREを許可して、デバッグWIREを使う場合は施錠ビットが解除(1)してください。 |
| デバッグWIREまたはJTAG使用時に異常メッセージまたは他の奇妙な動き。 | 目的対象が安全動作領域外で走行。最大周波数対VCC。 | 実際のデバイス用データシートで「電気的特性」章で記述されるような安全動作領域内で目的対象が走行することを確実にしてください。周波数低減や電圧増加を行ってください。 |

10. ファームウェア格上げ更新

ファームウェアを格上げ更新する方法の情報についてはAtmel Studio使用者の手引きをご覧ください。

11. 公開履歴と既知の問題

11.1. 新規情報

表11-1. 本公開での新規事項

| | |
|-----------|----------------------|
| ファームウェア版 | 主: 7.26、従: 7.26 |
| Studio公開版 | Atmel Studio 6.2 SP1 |
| 注釈 | サインオフでの状態LED修正 |

11.2. ファームウェア改訂履歴 (Atmel Studio)

表11-2. 以前の公開版

| | |
|-----------|---|
| ファームウェア版 | 主: 7.25、従: 7.25 |
| Studio公開版 | Atmel Studio 6.2 |
| 注釈 | 発振器校正を修正 |
| ファームウェア版 | 主: 7.24、従: 7.24 |
| Studio公開版 | Atmel Studio 6.1 SP2 |
| 注釈 | <ul style="list-style-type: none">・ 施錠されたPDIデバイスのアクセスを修正・ megaAVRに対する開始の問題を修正 |
| ファームウェア版 | 主: 7.20、従: 7.20 |
| Studio公開版 | AVR Studio 5.1 |
| 注釈 | <ul style="list-style-type: none">・ 改善されたデバッグWIRE単一段階実行(シングル ステップ)性能・ 大きな(>320KB)Atmel AVR XMEGAデバイスでのソフトウェア中断点(ブレーク ポイント)を支援・ aWire自動ホールド計算を改良・ (低電圧で見られる)XMEGAフラッシュ ページ プログラミング異常を修正・ 新しいXMEGAデバイスに対して修正されたチップ消去制限時間・ XMEGAデバイスでの高いSUT値に対する支援 |
| ファームウェア版 | 主: 7.12、従: 7.12 |
| Studio公開版 | AVR Studio 5.0 一般公開 |
| 注釈 | |
| ファームウェア版 | 主: 7.11、従: 7.11 |
| Studio公開版 | AVR Studio 5.0 公開β2 |
| 注釈 | 改善されたaWire速度 |
| ファームウェア版 | 主: 7.06、従: 7.06 |
| Studio公開版 | AVR Studio 5.0 公開β1 |
| 注釈 | なし |

11.3. 既知の問題

それら各々の分野での既知の問題が以下の項で記述されます。

11.3.1. 全般

- ・ ソースレベルでのGCCで生成されたコードの単一段階実行(シングル ステップ)は常に可能でないかもしれません。最良の結果のために最適化レベルを最低に設定し、必要な時に逆アセンブラ ウィンドウを使ってください。

11.3.2. ハードウェア関連

- ・ Atmel AVR JTAGICE mkIIをOFFに切り替える前に常に目的対象の電力をOFFに切り替えてください。電流が応用から漏れてエミュレータへの損傷に帰着し得るため、給電された応用へ接続されたJTAGICE mkIIを決して電力断のままにしないでください。
- ・ 目的対象応用がJTAGピンを汎用入出力用に使う場合、JTAGICE mkIIは(JTAG許可ヒューズが設定されていれば)JTAGピン経由で目的対象をプログラミングするのに未だ使われ得ます。けれども、目的対象デバイスのRESETピンをエミュレータのnSRSTピンに接続することを確実にしてください。この接続なしでは、目的対象応用がプログラミング後の走行を妨げられ得ます。応用がJTAGピンを出力として駆動する場合、それらはエミュレータや目的対象に損傷を与える結果になり得る、エミュレータとの信号競合になるでしょう。

11.3.3. Atmel AVR XMEGA関連

- メモリ割り当て動作でEEPROMをアクセスするコードを通しての段階実行は各EEPROMページのバイト0の破損を引き起こすかもしれません。

11.3.4. JTAG(mega)関連

- 目的対象電力が失われた、または外部リセットが印加された時に、いくつかの偽事象メッセージが現れるかもしれません。
- 自己変更コードを含むプロジェクトについて：プロジェクトをアップロードする(目的対象デバイスからPCへ内容を読む)時に、リセットベクタへリセットする前に数ms間、目的対象AVRが走行(RUN)動作に入ります。これは目的対象アプリケーションが目的対象Atmel AVRのフラッシュメモリとEEPROMの状態と変わったかもしれないことを意味します。

対策：アプリケーションは自己変更コードが意図せずアクセスされないように書かれなければなりません。

11.3.5. デバッグWIRE関連

- デバイスのコードがデバイスをリセットを強制すると、デバッグWIRE通信は失われます。BOD、WDTまたは他のリセット元がデバッグWIRE通信を失わせます。再確立された通信を得るには高電圧プログラミングが必要とされます。
- プログラムメモリの最終アドレスに中断点(ブレークポイント)が設定された場合、その中断点到達後に実行を続けることが不可能です。段階実行や実行はデバイスを走行にさせないでください。中断点の命令は一旦中断後に実行されないでください。
- 終結しないソースコードの最後に渡る段階実行はdWを異常メッセージを生成させるでしょう。これはデバイスがここに消去されてしまっていないコードを含むためです。使われないフラッシュページがデバッグ開始時に消去されないことに注意してください。
- デバッグ作業中に目的対象の電圧や周波数を変更してはいけません。デバッグ作業が先に終わらされなければなりません。
- 発振校正(OSCCAL)とクロック前置分周(CLKPR)のレジスタはデバッグ中にアプリケーションで書くことができません。これはAtmel AVR JTAGICE mkIIに同期を失わせ、Atmel StudioはJTAGICE mkIIと通信することができないでしょう。この問題を修正するためにデバッグ作業を再開してください。
- 電力削減(PRR)レジスタを持つデバイスで直列周辺インターフェース電力削減(PRSSP)ビットは1を書かれてはなりません。このビットが1を書かれた場合、デバッグWIRE単位部へのクロックが禁止され、JTAGICE mkIIとデバッグWIREインターフェース間の全ての通信が停止されます。現在、このビットはATmega48/88/168に存在します。
- 多すぎる中断点(ブレークポイント)の挿入は非常に低い目的対象クロック周波数に於いて通信を失わさせ得ます。中断点の挿入または取り外しの時に、変更された中断点を含む各フラッシュページに対して、JTAGICE mkIIは目的対象デバイスに対してフラッシュページ全体を読んで変更して書かなければなりません。非常に低いクロック周波数(kHz程度)での走行時、これはAtmel Studioを制限時間超過にさせ得ます。
対策：群で中断点を挿入して挿入間を単一段階実行(シングルステップ)してください。
- 128kHzクロック元での走行時、CLKDIV8ヒューズを設定(0)しないでください。これはインターフェース速度が低すぎるためにデバッグ作業を失敗させます。成功するデバッグ用の最小クロック速度は128kHzです。
- CLKDIV8ヒューズの設定(0)はデバッグWIRE使用時に接続問題を引き起こし得ます。最良の結果のため、デバッグ中にこのヒューズを非プログラム(1)のままに置いてください。

11.3.6. 共通

- I/OウィンドウでI/Oビットを編集時、”そのビット位置に1を書くことによって解除(0)される”フラグを解除(0)するには、最初にそれを解除(0)してその直後にそれを設定(1)してください。それは設定(1)去れた後、次の周期で目的対象デバイスによって自動的に解除(0)されます。設定(1)したフラグがレジスタに書き戻され、故に自動的に解除(0)されるので、レジスタ内で設定(1)されたどのフラグもレジスタを編集することによって解除(0)されることにも注意してください。
- SLEEP命令の単一段階実行(シングルステップ)はデバイスを休止動作に置きません。単一段階実行の代わりに実行(Run)動作を使ってください。
- Atmel Studioのプログラムメモリウィンドウを使って目的対象のフラッシュ内容を編集する、または監視(Watch)ウィンドウでフラッシュ内容を編集することは不可能です。
- USB使用時、同じAtmel Studioの実体からデバッグ作業で1つのAtmel AVR JTAGICE mkIIとプログラミング用に別の1つを使うことは不可能です。デバイスをデバッグすることと、また利用可能なプログラミングインターフェースを持つことが必要な時は、独立したAtmel Studioの実体を使ってください。
- バス給電USBハブ経由でJTAGICE mkIIが接続される場合、USB上でJTAGICE mkIIを格上げ更新することは不可能です。
- 安全機能としてどれかの施錠ビットが設定(0)された時にチップ上デバッグシステムが禁止されることに注意してください。
- 互換ヒューズを持つデバイスに対してJTAGICE mkIIでデバッグをする間、互換ヒューズを設定(0)しないでください。
- いくつかのデバイスで、LPM命令直後に2つの命令の1つでの実行中断は逆アセンブリウィンドウで表示されるフラッシュメモリを不正にするようです。
対策：LPMコードを単一段階実行(シングルステップ)しない、そしてLPM命令直後に中断点(ブレークポイント)を挿入しないでください。

12. 改訂履歴

| 資料改訂 | 日付 | 注釈 |
|--------|---------|--------|
| 42710A | 2016年4月 | 初版資料公開 |

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, AVR Studio®, megaAVR®, STK®, tinyAVR®, XMEGA®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。Windows®は米国と他の国に於けるMicrosoft Corporationの登録商標です。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。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作用の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2020.

本応用記述はAtmelのJTAGICE mkII使用者の手引き(Rev.42710A-04/2016)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。