

序説



Atmel JTAGICE3はチップ上デバッグ能力を持つARM® Cortex®-Mに基づくAtmel® SAMとAtmel AVR®マイクロコントローラのプログラミングとデバッグ用の強力な開発ツールです。

- JTAGとaWireの両インターフェースで全てのAtmel 32ビットAVRマイクロコントローラのプログラミングとチップ上デバッグ
- JTAGと2線PDIの両インターフェースで全てのAtmel AVR XMEGA® 8ビットマイクロコントローラ系デバイスのプログラミングとチップ上デバッグ
- JTAG、デバッグWIRE、またはUPDIのインターフェースでOCD支援を持つ全てのAtmel 8ビットAVRマイクロコントローラの(JTAG,SPI,UPDIでの)プログラミングとデバッグ
- SWDとJTAGの両インターフェースでARM Cortex-Mに基づく全てのAtmel SAMマイクロコントローラのプログラミングとデバッグ (ファームウェア 3.0とそれ以降版)

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。

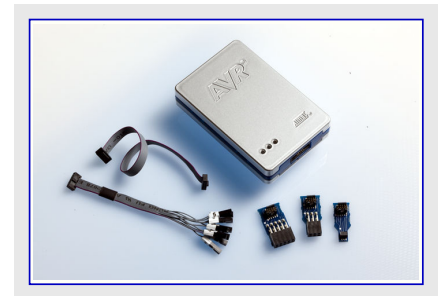
目次

序説	1	8.1. Atmel AVR UC3目的対象	17
1. 特徴	3	8.1.1. EVTI/EVTOの使い方	17
1.1. Atmel JTAGICE3の特徴	3	8.2. デバッグWIRE目的対象	18
1.2. システム要件	3	8.2.1. ソフトウェア中断点	18
2. Atmel JTAGICE3での開始に際して	3	9. ファームウェア格上げ更新	18
2.1. キット内容	3	10. 公開履歴と既知の問題	18
2.2. Atmel JTAGICE3の組み立て	4	10.1. 主版3での新規情報	18
2.3. Atmel JTAGICE3を開ける	5	10.2. ファームウェア公開履歴	19
2.4. Atmel JTAGICE3の給電	5	10.3. Atmel JTAGICE3に関する既知の問題	19
2.5. ホストコンピュータへの接続	5	10.3.1. Atmel AVR XMEGA OCD特有問題	19
2.6. USBドライバのインストール	5	10.3.2. Atmel megaAVR/tinyAVR OCD特有問題	19
2.6.1. Windows	5	11. 製品適法性	20
3. Atmel JTAGICE3の接続	6	11.1. RoHSとWEEE	20
3.1. JTAG目的対象への接続	6	11.2. CEとFCC	20
3.1.1. JTAG 10ピンコネクタの使い方	6	12. 改訂履歴	20
3.2. aWire目的対象への接続	7		
3.3. PDI目的対象への接続	7		
3.4. デバッグWIRE目的対象への接続	7		
3.5. SPI目的対象への接続	8		
3.6. SWD目的対象への接続	8		
3.7. UPDI目的対象への接続	9		
3.8. Atmel STK500とのAtmel JTAGICE3の使い方	9		
3.9. Atmel STK600とのAtmel JTAGICE3の使い方	9		
4. チップ上デバッグ	10		
4.1. チップ上デバッグ(OCD)の序説	10		
4.2. 物理インターフェース	10		
4.2.1. JTAG	10		
4.2.2. aWire	12		
4.2.3. PDI	12		
4.2.4. UPDI物理インターフェース	12		
4.2.5. デバッグWIRE	13		
4.2.6. SPI	13		
4.2.7. SWD	13		
4.3. Atmel OCD実装	13		
4.3.1. Atmel AVR UC3 OCD (JTAGとaWire)	13		
4.3.2. Atmel AVR XMEGA OCD (JTAGとPDI物性)	14		
4.3.3. Atmel megaAVR OCD (JTAG)	14		
4.3.4. Atmel megaAVR/tinyAVR OCD (デバッグWIRE)	14		
4.3.5. Atmel tinyX-OCD (UPDI)	14		
4.3.6. ARM CoreSight構成部	15		
5. ハードウェア説明	15		
5.1. LED	15		
5.2. 背面	15		
5.3. 底覆い	15		
5.4. 基本構造説明	16		
5.4.1. Atmel JTAGICE3主基板	16		
5.4.2. Atmel JTAGICE3目的対象コネクタ	16		
5.4.3. Atmel JTAGICE3目的対象コネクタ部品番号	16		
6. ソフトウェア統合	16		
6.1. Atmel Studio	16		
6.1.1. Atmel Studioでのソフトウェア統合	16		
6.1.2. プログラミング任意選択	16		
6.1.3. デバッグ任意選択	17		
7. コメント行ユーティリティ	17		
8. 高度なデバッグ技術	17		

1. 特徴

1.1. Atmel JTAGICE3の特徴

- Atmel Studioに完全適合
- 全てのAtmel AVR UC3 32ビット マイクロ コントローラのプログラミングとデバッグを支援
- 全ての8ビット AVR XMEGAデバイスのプログラミングとデバッグを支援
- OCDを持つ全てのAtmel 8ビット megaAVRとtinyAVRデバイスのプログラミングとデバッグを支援
- ARM Cortex-Mに基づく全てのSAMマイクロ コントローラのプログラミングとデバッグを支援
- 1.65～5.5Vの目的対象動作電圧範囲
- 動作中に目的対象のVTrefから3ma未満の引き出し
- 32kHz～15MHzのJTAGクロック周波数を支援
- 32kHz～10MHzのPDIクロック周波数を支援
- 4kビット/s～0.5Mビット/sのデバッグWIREホーレートを支援
- 7.5kビット/s～7.5Mビット/sのaWireホーレートを支援
- 8kHz～1.875MHzのSPIクロック周波数を支援
- 750kビット/sまでのUPDIホーレートを支援
- 32kHz～2MHzのSWDクロック周波数を支援
- USB高速(High-speed)ホスト インターフェース
- 最大1MB/sでのITM連続追跡捕獲
- 10ピン50mil JTAGコネクタだけでなく、アダプタを使って10ピン100mil、6ピン50mil SPI、6ピン100mil SPIのインターフェースも支援



1.2. システム要件

Atmel JTAGICE3本体はコンピュータにAtmel StudioかAVR Studio®の5.0またはそれ以降のような前置デバッグ環境がインストールされる必要があります。

JTAGICE3は提供されたUSBケーブルを使ってホスト コンピュータに接続されるべきです。

2. Atmel JTAGICE3での開始に際して

2.1. キット内容

- JTAGICE3本体
- USBケーブル (1.8m、高速(High-speed)、ミニB)
- 50mil SPI、100mil SPI、100mil JTAGのアダプタを含むアダプタ基板
- 50mil JTAGケーブル
- 50mil 10ピン ミニ バラ線ケーブル
- AVR技術ライブラリDVD

図2-1. JTAGICE3キット内容



2.2. Atmel JTAGICE3の組み立て

Atmel JTAGICE3本体は付随されるケーブルなしで出荷されます。組み立てキット内で次のような2つのケーブル任意選択が提供されます。

- ・ 50mil 10ピンIDCフラット ケーブル
- ・ 10×100milソケット付き50mil 10ピン ミニ バラ線ケーブル

殆どの目的に関し、組み立てキットで提供されるアダプタの1つと共に50mil 10ピンのIDCフラット ケーブルを使うことができます。1つの小さなPCB区画部分で3つのアダプタが提供されます。アダプタを分割するには、区分を前後に優しく曲げることによって単にそれぞれを折り取ってください。分割処理に起因するかもしれないどんな鋭い端をも避けるように注意してください。以下のアダプタが含まれます。

- ・ A09-0955 - A 100mil 10ピン JTAGアダプタ
- ・ A09-0955 - B 100mil 6ピン SPI/デバッグWIRE/PDI/aWireアダプタ
- ・ A09-0955 - C 50mil 6ピン SPI/デバッグWIRE/PDI/aWireアダプタ

図2-2. JTAGICE3ケーブル

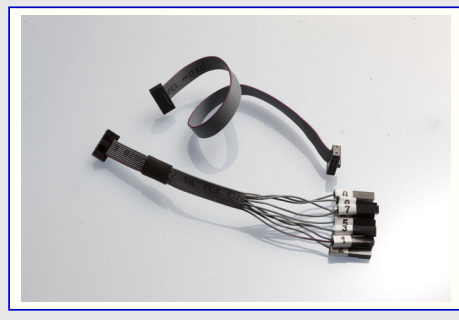
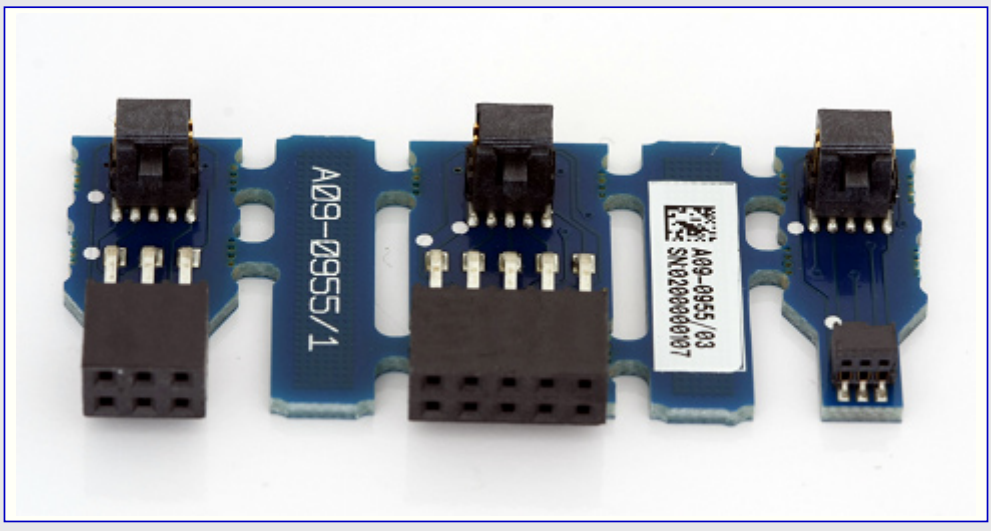


図2-3. JTAGICE3アダプタ



50mil JTAGアダプタが提供されないことに注意してください。これは50mil JTAGヘッダに直接的に接続するのに50mil 10ピンIDCケーブルを使うことができるためです。50mil 10ピンコネクタに使われる部品の部品番号については「[目的対象コネクタ部品番号](#)」をご覧ください。

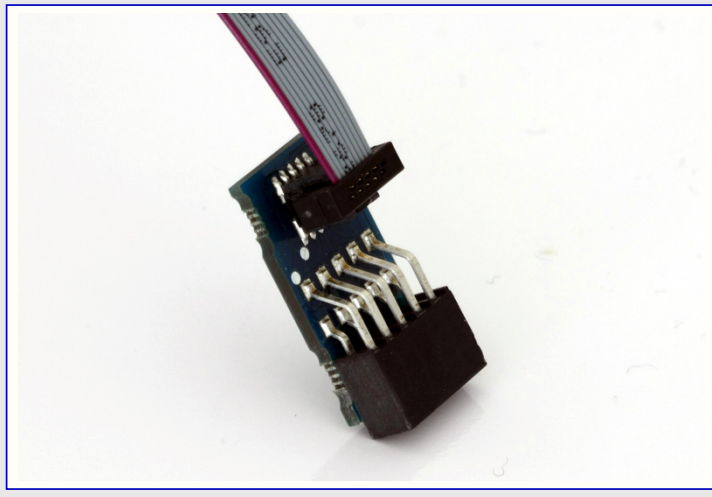
JTAGICE3をその既定構成設定に組み立てるには、右図で示されるように本体に10ピンの50mil IDCケーブルを接続してください。ケーブルの赤い線(1番ピン)が筐体の青い帯の三角目印と整列するよう、ケーブルの向きに注意してください。ケーブルは本体から上側へ接続されるべきです。

次に、アダプタ区画部から100mil 10ピンJTAGアダプタをポキッと折り、それをIDCケーブルの他端側に接続してください。Atmel JTAGICE3は今やその基本的なJTAG構成設定で使う準備が整いました。

図2-4. Atmel JTAGICE3ケーブル接続



図2-5. JTAGICE3アダプタ接続



2.3. Atmel JTAGICE3を開ける

注: 通常操作に関して、Atmel JTAGICE3本体が開かれてはなりません。本体を開けることはあなた自身の責任に於いて行われます。

JTAGICE3の筐体は、組み立て中に共に嵌められる上覆い、底覆い、それと青い帯の分離した3つのプラスチック部品から成ります。本体を開けるには単に大きなマイナス(-)ドライバを青い帯の隙間に挿入し、或る程度内側へ圧力を印加して、優しく捻ってください。他の嵌め穴でこれを繰り返し、上覆いを取り上げてください。

図2-6. JTAGICE3を開ける (訳注:原書での図2-6,7,8.は図2-6.として纏めました。)



本体を再び閉じるには、単に上覆いと底覆いを正しく整列し、共に固く押してください。

2.4. Atmel JTAGICE3の給電

Atmel JTAGICE3はUSBバス電圧によって給電されます。それは動作するのに100mA未満が必要で、故にUSBハブを通して給電することができます。電力LEDは本体が接続された時に点灯します。活動するプログラミングまたはデバッグの作業で接続されない時は、コンピュータの電池を保護するために本体が低電力消費へ移行します。JTAGICE3は電力断にすることができず、未使用時に接続を外されるべきです。

2.5. ホスト コンピュータへの接続

最初にAtmel JTAGICE3で接続する前に、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。

JTAGICE3は提供されたUSBケーブルを使ってホスト コンピュータで利用可能なUSBポートへ接続されなければなりません。JTAGICE3はUSB 2.0適合制御器を含み、全速(Full-speed)と高速(High-speed)の両動作で動作することができます。最良の結果のために、提供されたケーブルを使ってホスト コンピュータのUSB 2.0適合高速ハブへ直接接続してください。

2.6. USBドライバのインストール

2.6.1. Windows

Microsoft® Windows®が走行するコンピュータでAtmel JTAGICE3をインストールすると、JTAGICE3が最初に接続される時にUSBドライバが読み込まれます。

注: 最初に本体を接続する前に、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2-9. JTAGICE3 USBドライバのインストール

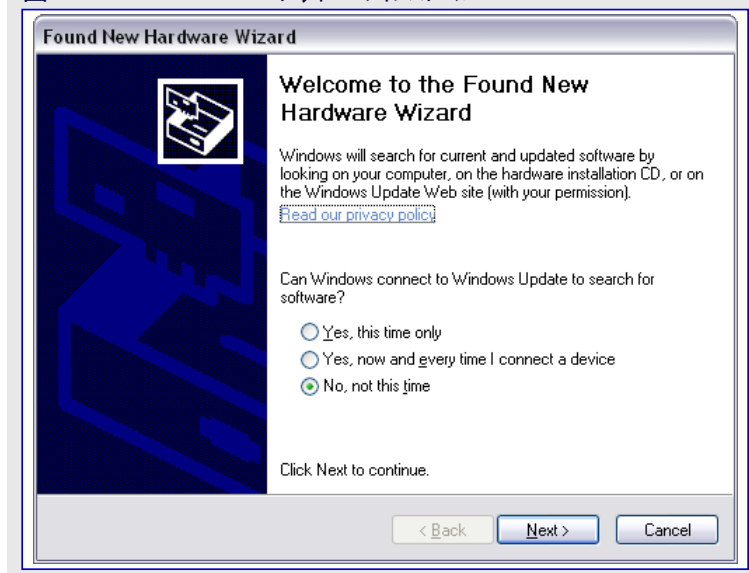
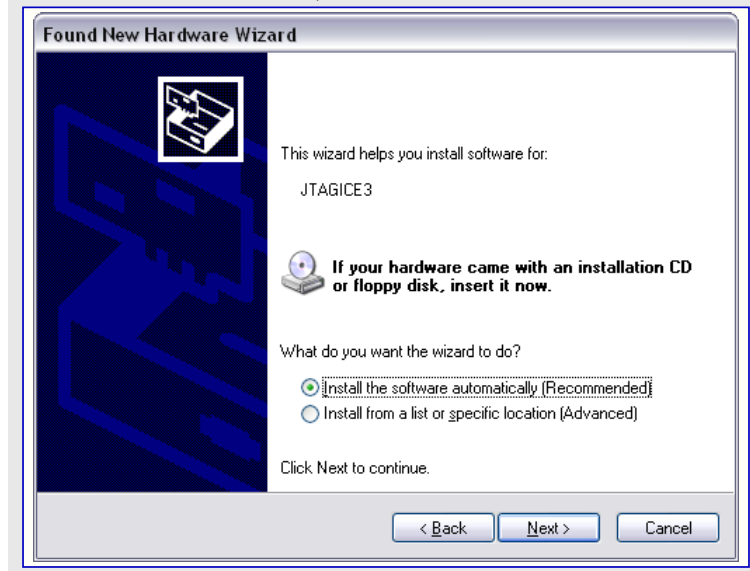


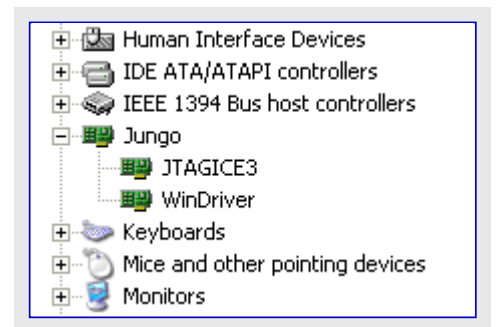
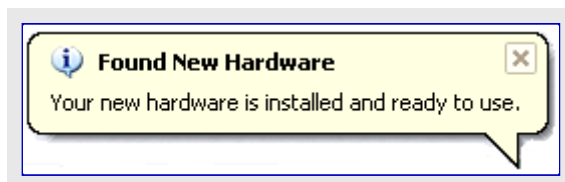
図2-10. JTAGICE3 USBドライバのインストール



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jtagice3.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、JTAGICE3はデバイス マネージャで”Jungo”装置として現れるでしょう。

JTAGICE3は今や使用の準備が整いました。



注: ファームウェア 3.0版とそれ以降は通信用にHIDインターフェースを使います。この場合、オペレーティング システムによって提供されるUSBドライバが自動的に読み込まれます。

3. Atmel JTAGICE3の接続

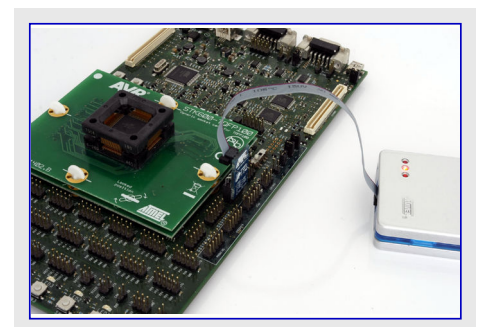
3.1. JTAG目的対象への接続

Atmel JTAGICE3探針はツールの筐体の前面で利用できる50mil 10ピンのJTAGコネクタを持ちます。このキットは目的対象基板上の50mil JTAGヘッダへ直接接続するのに使うことができる50mil 10ピン ケーブルを含みます。目的対象基板は(例えばAtmel STK[®]600の)100mil JTAGヘッダに適合させるべきで、キットで提供される(A08-0735-A)100milアダプタを使うことができます。目的対象基板が50 milまたは100milの適合した10ピン JTAGヘッダを持たない場合、100milソケットの個別10ヶ所への入出力を与える、提供された10ピンの”ミニ パラ線”ケーブルを使って独自配列に割り当てることができます。

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.で示されます。

JTAGICE3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。探針ケーブルの赤い線は100milアダプタでの’点’としての1番ピンを示します。



3.2. aWire目的対象への接続

aWireインターフェースはVCCとGNDに加えて1つのデータ線だけが必要です。Atmel JTAGICE3目的対象コネクタと目的対象基板間を接続するのに10ピンのミニパラ線ケーブルが使われるべきです。右表で示されるように、3つの接続が必要とされます。

表3-1. ミニパラ線ケーブルを使うaWireへの接続

JTAGICE3ピン	目的対象ピン	ミニパラ線ピン	aWireピン
1 (TCK)		1	
2 (GND)	GND	2	6
3 (TDO)	DATA	3	1
4 (VTG)	VTG	4	2
5 (TMS)		5	
6 (nSRST)		6	
7 (未接続)		7	
8 (nTRST)		8	
9 (TDI)		9	
10 (GND)		0	

3.3. 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.で示されます。

このキットで提供される(A-08-0735-BとA-08-0735-C)アダプタは50milまたは100milのどちらかの任意選択でこのピン配列に接続するのに使うことができます。

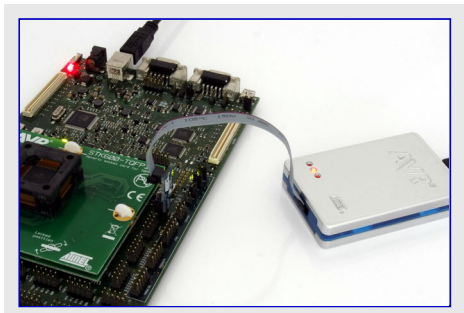
Atmel JTAGICE3を目的対象応用PCBに接続する時に6ピンヘッダを正しい向きで使うことを確実にしてください。アダプタ基板上的の‘点’が1番ピンを示します。

標準6ピンヘッダを持たない目的対象への接続時、JTAGICE3目的対象コネクタと目的対象基板間に10ピンミニ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、下表はそれらを接続する場所を記述します。

注: PDI_DATAが9番ピンに接続されるJTAGICE mk II のJTAG 探針とは違いがあります。JTAGICE3はAtmel AVR ONE!とAtmel AVR Dragon製品によって使われるピン配列と互換性があります。

表3-2. ミニパラ線ケーブルを使うPDIへの接続

JTAGICE3ピン	目的対象ピン	ミニパラ線ピン	STK600 PDIピン
1 (TCK)		1	
2 (GND)	GND	2	6
3 (TDO)	PDI_DATA	3	1
4 (VTG)	VTG	4	2
5 (TMS)		5	
6 (nSRST)	PDI_CLK	6	5
7 (未接続)		7	
8 (nTRST)		8	
9 (TDI)		9	
10 (GND)		0	



3.4. デバッグWIRE目的対象への接続

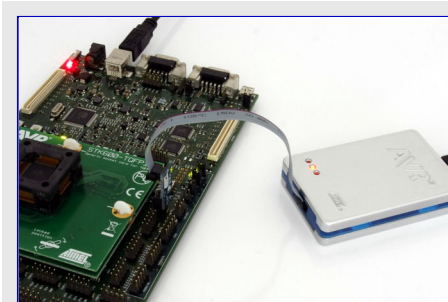
6ピンデバッグWIREコネクタ用のピン配列は図4-7.で示されます。

このキットで提供される(A-08-0735-BとA-08-0735-C)アダプタは50milまたは100milのどちらかの任意選択でこのピン配列に接続するのに使うことができます。

Atmel JTAGICE3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。アダプタ基板上的の‘点’が1番ピンを示します。

例えば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インターフェースを許可/禁止することができますように完全なSPIコネクタへの入出力を持つことが推奨されます。

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鉤を使って)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。(SPIENヒューズだけがプログラム(0)されるなら、)その後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。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、識別読み出し、チップ消去実行の能力だけです。

表3-3. ミニ バラ線ケーブルを使うデバッグWIREへの接続

JTAGICE3ピン	目的対象ピン	ミニ バラ線ピン
1 (TCK)		1
2 (GND)	GND	2
3 (TDO)		3
4 (VTG)	VTG	4
5 (TMS)		5
6 (nSRST)	RESET	6
7 (未接続)		7
8 (nTRST)		8
9 (TDI)		9
10 (GND)		0

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4-8.で示されます。
このキットで提供される(A-08-0735-BとA-08-0735-C)アダプタは50milまたは100milのどちらかの任意選択でこのピン配列に接続するのに使うことができます。

Atmel JTAGICE3を目的対象応用PCBに接続する時に6ピン ヘッドを正しい向きで使うことを確実にしてください。アダプタ基板上の‘点’が1番ピンを示します。

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!。

Atmel JTAGICE3目的対象コネクタとSPI目的対象基板を接続するのに10ピン ミニ バラ線ケーブルを使うこともできます。右表で示されるように、6つの接続が必要とされます。

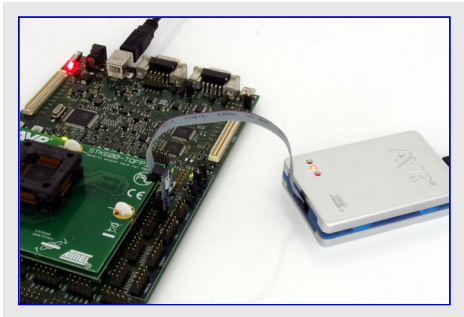


表3-4. ミニ バラ線ケーブルを使うSPIへの接続

JTAGICE3ピン	目的対象ピン	ミニ バラ線ピン	SPIピン
1 (TCK)	SCK	1	3
2 (GND)	GND	2	6
3 (TDO)	MISO	3	1
4 (VTG)	VTG	4	2
5 (TMS)		5	
6 (nSRST)	/RESET	6	5
7 (未接続)		7	
8 (nTRST)		8	
9 (TDI)	MOSI	9	4
10 (GND)		0	

3.6. SWD目的対象への接続

ARM SWDインターフェースはTCKとTMSのピンを利用するJTAGインターフェースの一部分で、SWDデバイスに接続する時に技術的に10ピンJTAGコネクタを使えることを意味します。けれどもARM JTAGとAVR JTAGのコネクタはピン互換ではなく、故にこれは使う目的対象基板の配置に依存します。STK600使用またはAVR JTAGピン配列利用時、標準JTAGコネクタを使うことができます。ARM JTAGピン配列を利用する基板への接続時、10ピンの”ミニ バラ線”ケーブルが使われなければなりません(この割り当てを行うAtmelからの利用可能なアダプタはありません)。

表3-5. ミニ バラ線ケーブルを使うSWDへの接続

JTAGICE3ピン	目的対象ピン	ミニ バラ線ピン	ARM SWDピン
1 (TCK)	SWDCLK	1	4 (SWDCLK)
2 (GND)	GND	2	3,5,9 (GND)
3 (TDO)	SWO(任意)	3	6
4 (VTG)	VTG	4	1 (VCC)
5 (TMS)	SWDIO	5	2 (SWDIO)
6 (nSRST)	RESET	6	10 (nRESET)
7 (未接続)		7	
8 (nTRST)		8	
9 (TDI)		9	
10 (GND)		0	3,5,9 (GND)

3.7. UPDI目的対象への接続

6ピンPDIコネクタ用のピン配列は「UPDI物理インターフェース」で示されます。

けれども、JTAGICE3は同時に(AVRヘッダの)6番ピンを駆動することなく(AVRヘッダの)3番ピンで1線UPDIインターフェースを駆動することができないハードウェアを持ちます。この理由のためにUPDIに対して10ピンのミニパラ線ケーブルを使うことが推奨されます。

注: Atmel-ICE及びより新しいツールはUPDIヘッダに直接接続する能力があります。

表3-6. ミニパラ線ケーブルを使うUPDIへの接続

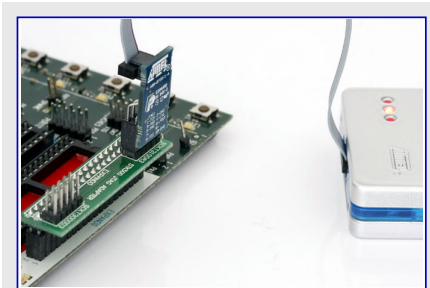
JTAGICE3ピン	目的対象ピン	ミニパラ線ピン	Atmel STK600 UPDIピン
1 (TCK)		1	
2 (GND)	GND	2	6
3 (TDO)	UPDI_DATA	3	1
4 (VTG)	VTG	4	2
5 (TMS)		5	
6 (nSRST)		6	
7 (未接続)		7	
8 (nTRST)		8	
9 (TDI)		9	
10 (GND)		0	

3.8. Atmel STK500とのAtmel JTAGICE3の使い方

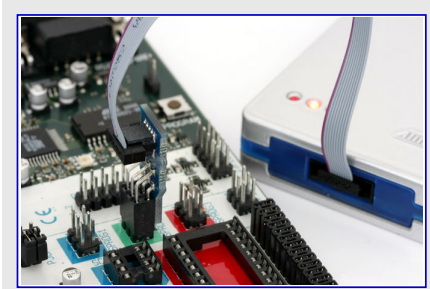
Atmel STK500スタータキットはJTAG、デバッグWIRE、SPIインターフェースを通してAtmel JTAGICE3を接続することができるAtmel AVRデバイスの収容場所に使うことができます。

JTAG目的対象への接続時、単純にATSTK500_JTAG_ADAPTERを使ってください。

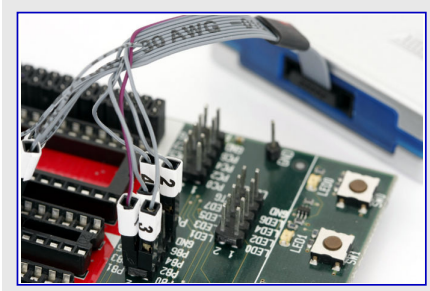
利用可能なSTK500 JTAGアダプタを持ってない場合、STK500のPORTC5～2でデバイスのJTAGポートに直接接続するのに10ピンの”パラ線”ケーブルを使うこともできます。



デバッグWIREとSPIの目的対象への接続は6ピン100milアダプタを使って行われます。デバッグWIREインターフェース使用時、リセット線に必要な時の駆動を許すためにSTK500のRESETジャンパを取り外すことを確実にしてください。



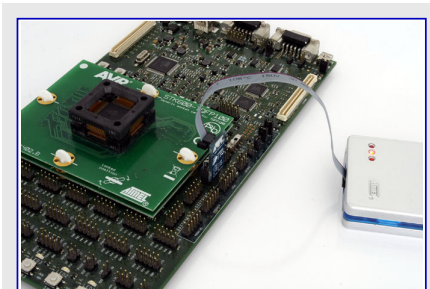
代わりに、JTAGICE3は提供された10ピンのミニパラ線ケーブルを使ってどの目的対象インターフェースへも接続することができます。



3.9. Atmel STK600とのAtmel JTAGICE3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 JTAGICE3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STK600上のJTAGコネクタへ接続するのに供給された10ピン100milアダプタを使ってください。



PDI、デバッグWIRE、SPI、aWireの目的対象への接続時、単純にSPI/PDIヘッダコネクタに接続するのに提供された6ピン100milアダプタを使ってください。



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 JTAGICE3は伝統的なエミュレータではありません。代わりに、JTAGICE3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JTAGICE3と完全に独立です。JTAGICE3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 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 OCD実装](#)」項をご覧ください。

4.2. 物理インターフェース

Atmel JTAGICE3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](#)で示されるようなピン配列を使うことが推奨されます。JTAGICE3はこのピン配列の100milと50milの両変種に接続することができます。

図4-1. JTAGインターフェースの基本

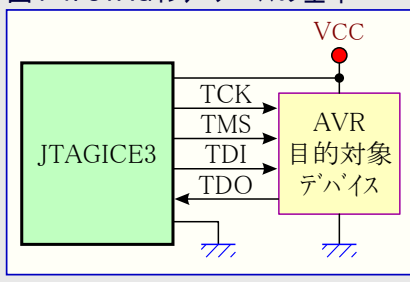


図4-2. JTAGヘッダピン配列

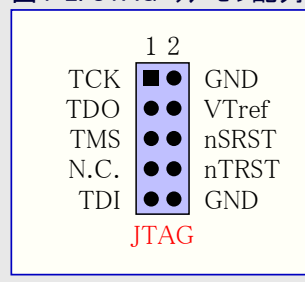


表4-1. JTAGピン説明

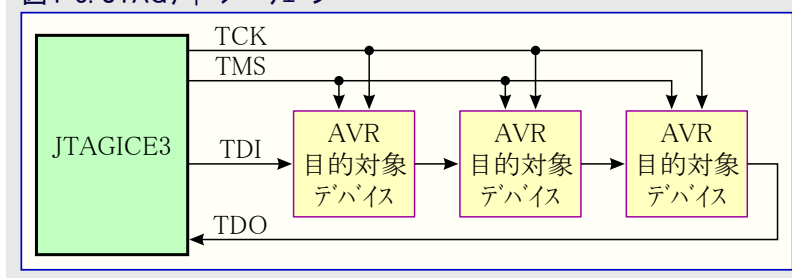
名前	ピン番号	説明
TCK	1	検査クロック (JTAGICE3から目的対象デバイスへのクロック信号)
TMS	5	検査種別選択 (JTAGICE3から目的対象デバイスへの制御信号)
TDI	9	検査データ入力 (JTAGICE3から目的対象デバイスへ送出されるデータ)
TDO	3	検査データ出力 (目的対象デバイスからJTAGICE3へ送出されるデータ)
nTRST	8	検査リセット (任意、いくつかのAVRデバイスのみ)。JTAG TAP制御器のリセットに使用
nSRST	6	リセット (任意)。目的対象デバイスのリセットに使用。或る筋書でのデバッグを必要とし得る、リセット状態で目的対象デバイスを保持することをJTAGICE3に許すために、このピンの接続が推奨されます。
VTG	4	目的対象基準電圧。JTAGICE3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。JTAGICE3はこのピンから1mA未満を引き出します。
GND	2,10	接地。JTAGICE3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。

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、下図で示されるように接続されなければなりません。

図4-3. JTAGデジーチェーン



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JTAGICE3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。JTAGICE3のVTGはこの電圧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何れかがJTAGポートを禁止する場合、JTAGICE3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して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JTAGICE3のTDOピンに到達する前に目的対象デバイスの後を通して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256ビットに制限されます。チェーン内のデバイス数は前が15、後ろが15に制限されます。

デジーチェーン例：TDI ⇒ ATmega1280 ⇒ ATxmega128A1 ⇒ ATUC3A0512 ⇒ TDO

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

Devices before : 1
 Devices after : 1
 Instruction bits before : 4 (8ビット AVRデバイスは4ビットのIRを持ちます。)
 Instruction bits after : 5 (32ビット AVRデバイスは5ビットのIRを持ちます。)

4.2.2. aWire

aWireインターフェースはプログラミングとデバッグの機能を許すためにAVRデバイスのRESET線を利用します。Atmel JTAGICE3によって、ピンの既定RESET機能を禁止する特別な許可手順が送出されます。

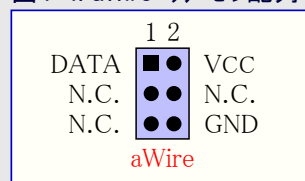
aWireインターフェースを持つAtmel AVRを含む応用PCBの設計時、右図で示されるようなピン配列を使うことが推奨されます。JTAGICE3はこのピン配列を支援する100milと50milの両アダプタと共に出荷されます。



助言: aWireが半二重インターフェースなので、方向変更時に開始ビット検出の失敗を避けるため、RESET線上に約47kΩのプルアップ抵抗が推奨されます。

aWireインターフェースはプログラミングとデバッグの両方のインターフェースとして使うことができ、10ピンJTAGインターフェースを通して利用可能なOCDシステムの全機能はaWireを通してアクセスすることもできます。

図4-4. aWireヘッダ ピン配列

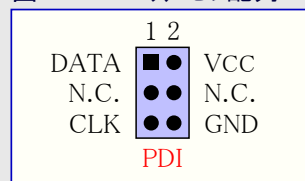


4.2.3. PDI

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、右図で示されるピン配列が使われるべきです。そしてJTAGICE3探針を応用PCBへ接続するのに、Atmel JTAGICE3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。

図4-5. PDIヘッダ ピン配列

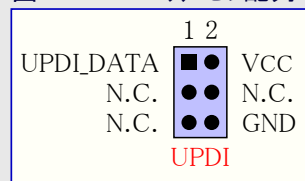


4.2.4. UPDI物理インターフェース

統合プログラミング/デバッグ用インターフェース(UPDI:Unified 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。これは全てのAVR XMEGAデバイスで見られるPDI 2線物理インターフェースの後継です。UPDIはプログラミングとデバッグの目的のために目的対象デバイスとの双方向半二重非同期通信を提供する単線インターフェースです。

UPDIインターフェースを持つAtmel AVRを含む応用PCBの設計時、右図で示されるピン配列が使われるべきです。そしてJTAGICE3探針を応用PCBへ接続するのに、Atmel JTAGICE3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。

図4-6. UPDIヘッダ ピン配列



4.2.4.1. JTAGICE3を使うUPDI

JTAGICE3はUPDIインターフェースを使うプログラミングとデバッグを支援します。けれども、JTAGICE3は同時に(AVRヘッダの)6番ピンを駆動することなく(AVRヘッダの)3番ピンで1線UPDIインターフェースを駆動することができないハードウェアを持ちます。この理由のためにUPDIに対して10ピンのミニ バラ線ケーブルを使うことが推奨されます。

注: Atmel-ICE及びより新しいツールはUPDIヘッダに直接接続する能力があります。

4.2.4.2. UPDIと/RESET

UPDI単線インターフェースは目的対象AVRデバイスに依存して専用ピンまたは共用ピンになり得ます。更なる情報についてはデバイスのデータシートを調べてください。

UPDIインターフェースが共用ピンの時にそのピンはRSTPINCFG1,0ヒューズを設定することによってUPDI、/RESETまたは汎用入出力(GPIO)のどれかに構成設定することができます。

RSTPINCFG1,0ヒューズはデータシートで記述されるように以下の構成設定を持ちます。各選択の実際上の影響はここで与えられます。

表4-2. JTAGピン説明

RSTPINCFG1,0	構成設定	使い方
0 0	GPIO	汎用入出力ピン。UPDIをアクセスするためにはこのピンに12Vのパルスが印加されなければなりません。外部リセット元は利用できません。
0 1	UPDI	プログラミング/デバッグ専用ピン。外部リセット元は利用できません。
1 0	/RESET	リセット信号入力。UPDIをアクセスするためにはこのピンに12Vのパルスが印加されなければなりません。
1 1	(予約)	-

注: より古いAVRデバイスは(直列と卑劣の変種が存在する)“高電圧プログラミング”として知られるプログラミング インターフェースを持ちます。一般的にこのインターフェースはプログラミング作業の間に/RESETピンに印加されるべき12Vが必要です。UPDIインターフェースは全く異なるインターフェースです。UPDIピンは代替機能(/RESETまたはGPIO)を持つようにヒューズ設定することができる、主にプログラミングとデバッグのピンです。代替機能が選択された場合、UPDI機能を再活性するためには12Vのパルスが必要とされます。

注: ピンの制限のためにUPDI信号の共用を設計が必要とする場合、デバイスがプログラミングできるのを保証するために段階が取られなければなりません。UPDI信号が正しく機能するのを保証するだけでなく、12Vのパルスから外部部品への損傷を避けるには、デバイスのプログラミングやデバッグを試みる時にこのピン上のどの部品も切断されることが推奨されます。これは既定で実装されデバッグの間に取り去られるか、またはピン ヘッドによって置換される0Ω抵抗を使って行うことができます。事実上この構成設定はデバイスを実装する前にプログラミングが行われるべきであることを意味します。

重要: JTAGICE3はUPDI線での12Vを支援しません。換言すると、UPDIピンはGPIOまたは/RESETとして構成設定されていた場合、JTAGICE3はUPDIインターフェースを許可することができません。

4.2.5. デバッグWIRE

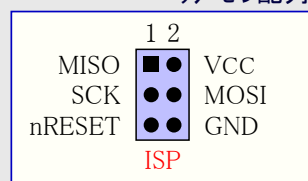
デバッグWIREインターフェースは小ピン数デバイスで使うためにAtmelによって開発されました。4ピンを使うJTAGインターフェースとは異なり、デバッグWIREはデバッグ ツールと共に双方向半二重非同期通信に単一(RESET)ピンだけを利用します。

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、右図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4-8.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ぬことを意味します。

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(オープントレイン)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。

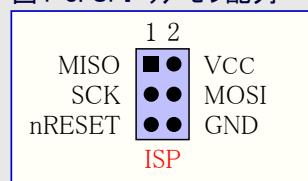
図4-7. デバッグWIRE(SPI) ヘッド ピン配列



4.2.6. 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、右図で示されるピン配列が使われるべきです。

図4-8. SPIヘッド ピン配列

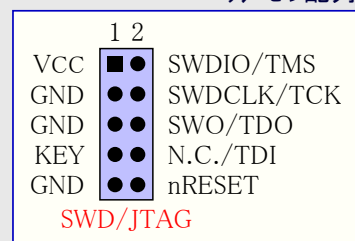


4.2.7. SWD

ARM SWDインターフェースはJTAGインターフェースの一部分で、TCKとTMSのピンを利用します。けれどもARM JTAGとAVR JTAGのコネクタはピン互換ではなく、故にSWDまたはJTAGのインターフェースを持つSAMデバイスを使う応用PCBを設計する時に、右図で示されるARMピン配列の使用が推奨され、JTAGICE3によって使われるAVRピン配列に割り当てるのにパラ線ケーブルかアダプタを使ってください(Atmelから入手可能なこの割り当てをするアダプタはありません)。

JTAGICE3はUART形式ITM追跡をホスト コンピュータへ流す能力があります。追跡は10ピン ヘッドのTRACE/SWO(JTAGのTDO)ピンで捕獲されます。データはJTAGICE3で内部的に緩衝され、HIDインターフェース上でホスト コンピュータへ送られます。信頼できる最大データ速度は概ね1MB/sです。

図4-9. 推奨ARM SWD/JTAG ヘッド ピン配列



4.3. Atmel 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(ブレーク 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
- 実時間プログラム カウンタ分岐追跡、データ追跡、処理追跡(Atmel JTAGICE3によって不支援)

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ホリック中断点(予約)
- 4つのハードウェア比較器
- (BREAK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

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を使う)無制限数のプログラム中断点 (ATmega128[A]を除く)

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

デバッグWIRE OCDは小ピン数のAtmel 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

4.3.5. Atmel tinyX-OCD (UPDI)

Atmel tinyX-OCDはUPDI(単線PDI)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
- 2つのプログラム メモリ (ハードウェア)中断点 (1つは予約)
- (BREAKを使う)無制限数のプログラム中断点

4.3.5.1. UPDIと/RESET

UPDI単線インターフェースは目的対象AVRデバイスに依存して専用ピンまたは共用ピンになり得ます。更なる情報についてはデバイスのデータシートを調べてください。

UPDIインターフェースが共用ピンの時にそのピンはRSTPINCFG1,0ヒューズを設定することによってUPDI、/RESETまたは汎用入出力(GPIO)のどれかに構成設定することができます。

RSTPINCFG1,0ヒューズはデータシートで記述されるように以下の構成設定を持ちます。各選択の実際上の影響はここで与えられます。

表4-3. JTAGピン説明		
RSTPINCFG1,0	構成設定	使い方
0 0	GPIO	汎用入出力ピン。UPDIをアクセスするためにはこのピンに12Vのパルスが印加されなければなりません。外部リセット元は利用できません。
0 1	UPDI	プログラミング/デバッグ専用ピン。外部リセット元は利用できません。
1 0	/RESET	リセット信号入力。UPDIをアクセスするためにはこのピンに12Vのパルスが印加されなければなりません。
1 1	(予約)	-

注: より古いAVRデバイスは(直列と卑劣の変種が存在する)“高電圧プログラミング”として知られるプログラミング インターフェースを持ちます。一般的にこのインターフェースはプログラミング作業の間に/RESETピンに印加されるべき12Vが必要です。UPDIインターフェースは全く異なるインターフェースです。UPDIピンは代替機能(/RESETまたはGPIO)を持つようにヒューズ設定することができる、主にプログラミングとデバッグのピンです。代替機能が選択された場合、UPDI機能を再活性するためには12Vのパルスが必要とされます。

注: ピンの制限のためにUPDI信号の共用を設計が必要とする場合、デバイスがプログラミングできるのを保証するために段階が取られなければなりません。UPDI信号が正しく機能するのを保証するだけでなく、12Vのパルスから外部部品への損傷を避けるには、デバイスのプログラミングやデバッグを試みる時にこのピン上のどの部品も切断されることが推奨されます。これは既定で実装されデバッグの間に取り去られるか、またはピン ヘッドによって置換される0Ω抵抗を使って行うことができます。事実上この構成設定はデバイスを実装する前にプログラミングが行われるべきであることを意味します。

 **重要:** JTAGICE3はUPDI線での12Vを支援しません。換言すると、UPDIピンはGPIOまたは/RESETとして構成設定されてしまった場合、JTAGICE3はUPDIインターフェースを許可することができません。

4.3.5.2. tinyX-OCD (UPDI)の特別な考慮

UPDIデータピン(UPDI_DATA)は目的対象AVRデバイスに依存して専用ピンまたは共用ピンになり得ます。共用されたUPDIピンは12Vに耐え、/RESETまたはGPIOとして使うように構成設定することができます。これらの構成設定でこのピンを使う方法の更なる詳細については「UPDI物理インターフェース」をご覧ください。

巡回冗長検査メモリ操作(CRCSCAN:Cyclic Redundancy Check Memory Scan)単位部を含むデバイスでこの単位部はデバッグ中に背面継続動作で使われるべきではありません。OCD単位部は制限されたハードウェア中断点(ブレイクポイント)比較器資源を持ち、故にもっとも中断点が必要とされる時、または更にソースレベルのコード段階実行の間、フラッシュメモリ内に**BREAK**命令が挿入されるかもしれません(ソフトウェア中断点)。CRC単位部はフラッシュメモリ内容の不正としてこの中断点を不正に検出し得ます。

CRCSCAN単位部はブート前にCRC走査を実行するように構成設定することもできます。CRC不一致の場合、デバイスはブートせず、固着状態が出現します。この状態からデバイスを回復する方法は完全なチップ消去と有効なフラッシュメモリ情報を書き込むか、またはブート前CRCSCANを禁止するかのどちらかを実行することだけです(単純なチップ消去は無効なCRCを持つ空白フラッシュメモリに帰着し、従ってデバイスは未だブートしません)。Atmel Studioはこの状態でデバイスをチップ消去する時に自動的にCRCSCANヒューズを禁止します。

UPDIインターフェースが使われる目的対象応用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。

- UPDI線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。プルダウン抵抗は使われるべきではなく、またはUPDI使用時に取り外されるべきです。UPDI物性はプッシュプル能力があり、故に線がアイドルの時に誤った開始ビットの起動を防ぐために弱いプルアップ抵抗が必要とされるだけです。
- UPDIピンがRESETとして使われる場合、それがインターフェースの正しい動作を妨害するため、どの安定化コンデンサもUPDI使用時に切断されなければなりません。
- UPDIピンがRESETまたはGPIOのピンとして使われる場合、それらがインターフェースの正しい動作を妨害するため、線上の全ての外部駆動部はプログラミングまたはデバッグの間、切断されなければなりません。

4.3.6. ARM CoreSight構成部

Cortex-Mに基づくAtmel ARMマイクロコントローラはOCD構成部に適合するCoreSight™を実装します。これらの構成部の機能はデバイス毎に変わり得ます。更なる情報についてはデバイスのデータシートを調べてください。

5. ハードウェア説明

5.1. LED

Atmel JTAGICE3の上覆い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	機能	説明
左	目的対象電源	目的対象の電源がOKの時に 緑 。点滅は目的対象電源異常を示します。プログラミング/デバッグ作業の接続が開始するまで点灯しません。
中央	主電源	主基板の電源がOKの時に 赤
右	状態	目的対象が走行/段階実行時に 緑 点滅。目的対象停止時に消灯



5.2. 背面

Atmel JTAGICE3の背面はUSBミニBコネクタを収容します。



5.3. 底覆い

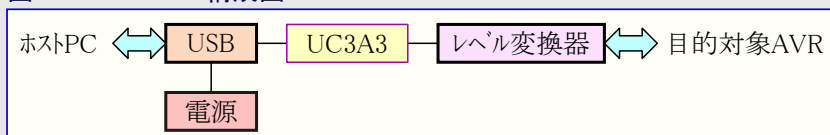
Atmel JTAGICE3の底覆いは通番と製造日付を示す張り札を持ちます。技術支援を求める時にこれら詳細を含めてください。



5.4. 基本構造説明

Atmel JTAGICE3の基本構造は右図で示されます。

図5-1. JTAGICE3構成図



5.4.1. Atmel JTAGICE3主基板

電力はUSBバスからAtmel JTAGICE3へ供給され、降圧スイッチング調整器によって3.3Vに調整されます。VTGピンによって目的対象応用からも少量の電力(1mA未満)が引き出され、それは基板上のレベル変換器の可変電圧側の電力に使われます。JTAGICE3主基板の心臓部は処理される作業に依存して1~60MHzで走行するAtmel AVR UC3マイクロ コントローラのAT32UC3A3256です。このマイクロ コントローラはデバッグとの高いデータ単位処理量を許す、チップ上のUSB 2.0高速(High speed)部を含みます。

JTAGICE3と目的対象デバイス間の通信は目的対象の動作電圧とJTAGICE3の内部電圧レベル間の信号を転換する一揃いのレベル変換器を通して行われます。また、信号経路には直列終端抵抗とESD保護ダイオードがあります。全ての信号チャネルは1.65~5.5Vの範囲で動作することができます。最大動作周波数は使う目的対象インターフェースに従って変わります。

5.4.2. Atmel JTAGICE3目的対象コネクタ

Atmel JTAGICE3はアクティブ探針を持ちません。直接または提供されるアダプタの1つのどちらかで目的対象に接続するのに50mil IDCケーブルが使われます。配線とアダプタのより多くの情報については「[JTAGICE3の組み立て](#)」項をご覧ください。

5.4.3. Atmel JTAGICE3目的対象コネクタ部品番号

JTAGICE3 50mil IDCケーブルを目的対象基板へ直接接続するため、どの標準50mil 10ピン ヘッドも充分であるべきです。それらがこのキットに含まれるアダプタ基板で使われるように、目的対象への接続時に正しい向きを保証するため、込み栓付きヘッドの使用が勧告されます。

SAMTECからのこのヘッドの部品番号はFTSH-105-01-LDV-K-A-Pです。

6. ソフトウェア統合

6.1. Atmel Studio

6.1.1. Atmel Studioでの統合

Atmel StudioはWindows環境でAtmel AVR応用を書いてデバッグするための統合開発環境(IDE: Integrated Development Environment)です。Atmel Studioはプロジェクト管理ツール、ソース ファイル エディタ、シミュレータ、アセンブラとC/C++用前処理部、エミュレーション、チップ上デバッグを提供します。

Atmel StudioまたはAVR Studio 5.0またはそれ以降版はAtmel JTAGICE3と共に使われなければなりません。

6.1.2. プログラミング任意選択

Atmel StudioはAtmel JTAGICE3を使ってAtmel AVRとAtmel SAM D20 ARM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、SWDの動作を使うように構成設定することができます。

クロック周波数構成設定時、各種インターフェースと目的対象システムに対して以下の各種規則が適用されます。

- SPIプログラミングは目的対象クロックを利用します。そのクロック周波数を目的対象デバイスが現在走行している周波数の1/4未満に構成設定してください。
- Atmel megaAVRデバイスでのJTAGプログラミングは書き込み器によってクロック駆動されます。これはプログラミング クロック周波数がデバイスそれ自身の最大動作周波数(通常、16MHz)に制限されることを意味します。
- JTAGとPDIの両インターフェースでのAVR XMEGAプログラミングは書き込み器によってクロック駆動されます。これはプログラミング クロック周波数がデバイスそれ自身の最大動作周波数(通常、32MHz)に制限されることを意味します。
- JTAGインターフェースでのAVR UC3プログラミングは書き込み器によってクロック駆動されます。これはプログラミング クロック周波数がデバイスそれ自身の(33MHzに制限される)最大動作周波数に制限されることを意味します。
- aWireインターフェースでのAVR UC3プログラミングは書き込み器によってクロック駆動されます。最適周波数は目的対象デバイスのSABバス速度によって与えられます。JTAGICE3デバッグはこの判定基準に合うようにaWireのボーレート自動的に調節します。通常必要ないとしても、使用者は(例えば、雑音環境で)必要な場合に最大ボーレートを制限することができます。
- SWDインターフェースでのSAM D20 ARMプログラミングは書き込み器によってクロック駆動されます。JTAGICE3で支援される最大周波数は2MHzです。周波数は目的対象CPU周波数×10を超えるべきではありません。

6.1.3. 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の1/4未満であるべきです。

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、故に周波数設定は必要ありません。Atmel JTAGICE3はデバッグ作業の開始で通信用の正しいボーレートを自動的に選択します。けれども、雑音性のデバッグ環境に関連した確実性の高い問題を経験した場合、デバッグWIRE速度をその '推奨(recommended)' 設定の何分の1かに強制することが可能です。

AVR XMEGA目的対象デバイスのデバッグ作業はデバイスそれ自身の最大速度(通常32MHz)までクロック駆動することができます。

JTAGインターフェースでのAVR UC3目的対象デバイスのデバッグ作業はデバイスそれ自身の(33MHzに制限される)最大速度までクロック駆動することができます。けれども、最適周波数は目的対象デバイスの現在のSABクロックより僅かに低くなります。

aWireインターフェースでのUC3目的対象デバイスのデバッグ作業はJTAGICE3それ自身によって最適ボーレートへ自動的に調節されます。けれども、雑音性のデバッグ環境に関連した確実性の高い問題を経験した場合、aWire速度を構成設定可能限度以下に強制することが可能です。

SWDインターフェースでのSAM D20 ARM目的対象デバイスのデバッグ作業は目的対象CPUクロック×10(最大2MHzに制限)までクロック駆動することができます。

- EEPROM保護(Preserve EEPROM)

デバッグ作業前に目的対象の書き込み中に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。

- 外部リセット使用(Use External Reset)

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Atmel SAM4S Xplained Pro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 6.2 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 6.2¥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命令: `atprogram --help` を入力してください。

8. 高度なデバッグ技術

8.1. Atmel AVR UC3目的対象

8.1.1. EVTI/EVTOの使い方

EVTIとEVTOのピンはAtmel JTAGICE3でアクセスできません。けれども、それらは他の外部装置と共に未だ使うことができます。

EVTIは以下の目的で使うことができます。

- 目的対象は外部事象への応答で実行の停止を強制することができます。DCレジスタ内の制御での事象(EIC:Event In Control)ビットが '01' を書かれる場合、EVTIピンでのHigh⇒Low遷移が中断点(ブレイク ポイント)状態を生成します。これが起こる時にDS内の外部中断点(EXB:External Breakpoint)ビットが設定(1)されます。
- 追跡同期メッセージ生成。JTAGICE3によって使われません。

EVTOは以下の目的で使うことができます。

- CPUがデバッグ動作へ移行したことを指示。DC内のEOSビットの '01' 設定は、目的対象デバイスがデバッグ動作移行時に1CPUクロック 周期間Lowへ引かれることをEVTOピンに起こします。この信号は外部オシロスコープ用の起動元として使うことができます。
- CPUが中断点または監視点に到達したことを指示。対応する中断点/監視点制御レジスタ内のEOCビットの設定により、中断点または監視点の状態がEVTOピンで示されます。DC内のEOSビットはこの機能を許可するために '10' に設定されなければなりません。そして監視点タイミングを調べるためにEVTOピンは外部オシロスコープに接続することができます。
- 追跡タイミング信号生成。JTAGICE3によって使われません。

8.2. デバッグWIRE目的対象

8.2.1. ソフトウェア中断点

デバッグWIRE OCDはAtmel megaAVR (JTAG) OCDと比べた時に徹底的に縮小されています。これはデバッグ目的に使用者に対して利用可能などんなプログラム カウンタ中断点(ブレーク ポイント)比較器も持たないことを意味します。カーソルまで実行や一段実行の目的用にこのような比較器が1つ存在しますが、使用者中断点はハードウェアで支援されません。

代わりに、デバッグはAtmel AVRの**BREAK**命令を利用しなければなりません。この命令はフラッシュ メモリに置かれ、それが実行のために読み込まれた時にAVR CPUを停止動作へ移行させます。デバッグ中に中断点を支援するため、デバッグは使用者が求める中断点の点でフラッシュ メモリ内に**BREAK**命令を挿入しなければなりません。元の命令は後で置き換えるために貯えられなければなりません。**BREAK**命令での一段実行時、デバッグはプログラムの動きを維持するために貯えられた元の命令を実行しなければなりません。極端な場合では、**BREAK**がフラッシュ メモリから取り去られて後で置換されなければなりません。これら全ての筋書は中断点からの一段実行時に明白な遅延を引き起こし得て、目的対象クロック周波数が非常に低い時に悪化されるでしょう。

故に可能な以下の指針を守ることが推奨されます。

- ・ デバッグ中、常に可能な限り高い周波数で目的対象を動かしてください。デバッグWIRE物理インターフェースは目的対象クロックからクロック駆動されます。
- ・ 各1つが目的対象で置換されるべきフラッシュ ページを必要とするため、中断点の追加と削除を最小とするよう試みてみてください。
- ・ フラッシュ ページの書き込み操作数を最小にするため、同時に少数の中断点を追加または削除するよう試みてください。
- ・ 可能なら、2語命令に中断点を配置することを避けてください。

9. 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。

10. 公開履歴と既知の問題

10.1. 主版3での新規情報

ファームウェア第3版、SAM支援追加

Atmel Studio 6.1 SP2インストール時、JTAGICE3は第2版から第3版へ増した主ファームウェアを受け取ります。第2版はデバイス マネージャでJungo USBツールとして列挙(認識)され、一方第3版はXplained Pro系統の評価キットに含まれるEDBGのようなものである、(新しいUSB PIDでの)HID装置として現れます。

新しいJTAGICE3ファームウェアは全てのAVR機能を保つと同時にJTAGICE3にARM Cortex-Mに基づく全てのAtmel SAMマイクロ コントローラ用支援を齎します。

3.11版の時にJTAGICE3は直列ITM追跡捕獲も支援します。

注:

ファームウェア第3版へ格上げ更新後、JTAGICE3は以前の版のAtmel Studioによって認証されません。旧版のAtmel StudioでJTAGICE3を使うには、最初にファームウェア第2版へ格下げ更新しなければなりません。

atfw.exeコマンド'行ファームウェア格上げ更新ユーティリティはファームウェア第2版zipファイルへのパスをユーティリティに与えるために**-a archive-name**スイッチと共に使われなければなりません。

atfw.exeコマンド'行格上げ更新ユーティリティはAtmelウェブサイトまたはAtmel展示室からダウンロード'することができます。Atmel展示室のダウンロードは**atfw.exe**と必要とするファームウェア版を含みます。

SWDを使ってJTAGICE3をSAMデバイスに接続する情報については「**SWD目的対象への接続**」をご覧ください。

10.2. ファームウェア改訂履歴

表10-1. 公開ファームウェア版

ファームウェア版 (10進)	日付	関連変更
1.08	–	JTAGICE3の初公開
1.17	–	LEDの動き改善 デバッグWIRE信頼性改善 aWire速度改善
1.22	–	状態LED改善
1.34	–	JTAGデイスリーチェーン支援 外部及びソフトウェアでのリセット処理を改善 Atmel AVR XMEGAデバイスでの高SUT値を支援 aWire自動ポーレート計算改善 より新しいAtmel AVR XMEGAデバイスに対して修正されたチップ消去制限時間 (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 デバッグWIREの一段実行性能を改善
2.10	–	些細なバグ修正
2.15	–	些細なバグ修正
3.7	–	SAM D20 ARM Cortex-M0+に基づくマイクロ コントローラを支援
3.18	–	全てのSAM ARM Cortex-Mに基づくマイクロ コントローラを支援 ITM追跡捕獲の支援を追加
3.23	–	発振器校正命令を修正 デバッグWIRE信頼性改善
3.30	–	目的対象電源LEDの問題を修正 64命令ビットよりも多い命令ビットを持つJTAGデイスリーチェーンに関連する問題を修正
3.34	2016年9月29日	UPDIインターフェース(tinyXデバイス)の支援を追加 USBエンドポイントの大きさ構成設定を作成

10.3. Atmel JTAGICE3に関する既知の問題

10.3.1. Atmel AVR XMEGA OCD特有問題

- ATxmegaA1系については改訂Gまたはそれ以降版だけが支援されます。
- PDIクロック信号上に過剰下振れが多くある場合DPIは或る周波数で動かないかもしれません。いくつかの目的対象基板構成設定に於いて、PDIクロックはかなりの量の過剰下振れが見られます。この過剰下振れはAtmel JTAGICE3が目的対象と通信できなくなる結果に帰着するかもしれません。この問題を逃れるため、PDIクロック周波数はより低くされるべきです。これはプログラミング ダイアログ内のインターフェース設定(Interface settings)タブか、またはデバッグ プロジェクトのプロジェクト プロパティ内のToolタブで行うことができます。より高いPDIクロック周波数が必要とされる場合、PDIクロック線に何らかの終端が追加されなければなりません。例えば、目的対象基板上でPDIクロックとVCC間に(概ね100pFの)小容量コンデンサを追加することは効果を証明されています。直列抵抗の追加はそれがJTAGICE3主基板に追加されない限り、効果がありません。

10.3.2. Atmel megaAVR/tinyAVR OCD特有問題

- デバッグ作業中のATmega32U6での電源ON繰り返しはデバイスとの通信中断を引き起こすかもしれません。

11. 製品適法性

11.1. RoHSとWEEE

Atmel JTAGICE3(ATJTAGICE3キット)とその付属品はRoHS指令(2002/95/EC)とWEEE指令(2002/96/EC)の両方に従って製造されます。

11.2. CEとFCC

Atmel JTAGICE3本体は以下のような必須の必要条件とその他の関連指令条項に従って検査されています。

- 2004/108EC(B級)指令
- FCC区分15副区分B
- 2002/95/EC(RoHS,WEEE)

評価には以下の規格が使われます。

- EN 61000-1 (2007)
- EN 61000-3 (2007)
- FCC CFR 47区分15 (2008)

技術構成ファイルは以下に置かれます。

Atmel Norway
Vestre Rosten 79
7075 Tiller
Norway

全ての努力がこの製品からの電磁放射を最小にさせました。けれども、或る条件下で、システム(目的対象応用回路に接続された本製品)は前述の規格によって許される最大値を超える個別電磁成分周波数を放射するかもしれません。この放射の周波数と大きさは使われる製品と目的対象応用の配置と配線を含む様々な要素によって決められます。



12. 改訂履歴

資料改訂	日付	注釈
42634A	2015年11月	初版資料公開
42634B	2016年10月	UPDIインターフェース追加とファームウェア公開履歴更新



Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA TEL:(+1)(408) 441-0311 FAX: (+1)(408) 436-4200 | www.atmel.com

© 2016 Atmel Corporation. / 改訂:Atmel-42634B-JTAGICE3_User Guide-10/2016

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。ARM[®]、ARM Connected[®]ロゴとその他はARM Ltd.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JTAGICE3使用者の手引き(Rev.42634B-10/2016)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。