

AVR067 : JTAGICEmk II 通信規約

要点

- AVR StudioからJTAGICEmk II へ送られる命令
- JTAGICEmk II からAVR Studioへ送られる返答
- 構成設定可能なパラメータ
- 各種メモリ形式
- パケット同期と異常制御用の特殊文字とパケット形式
- JTAGICEmk II で扱う中断点(ブレークポイント) (tinyAVR/megaAVRデバイスのみ)

1. 序説

この資料はAVR StudioとJTAGICEmk II 間で使われる通信規約を記述します。この資料の目的は第3の供給者に、JTAGICEmk II に対するそれら自身の前置処理部の設計を可能にすることです。

2. 動作の理屈

この通信規約は主/従環境で動作し、AVR Studioが主装置になり、JTAGICEmk II が従装置になります。

JTAGICEmk II 従装置は常に以下に列挙される3つの状態の1つで動きます。

1. 停止(STOPPED)
2. 走行(RUNNING)
3. プログラミング(PROGRAMMING)

いくつかの命令は従装置が停止(STOPPED)動作中にだけ実行することができ、他の命令は全動作で許され得ます。詳細は各個別命令に対して記述されます。

事象は主装置から与えられる命令が先行しない、従装置から送られるメッセージです。

直列通信は本質的に非同期で、そしてJTAGICEmk II もです。全ての命令と応答は受信したメッセージの有効性検証を受信側に許すことを含む証拠印(token:トークン)とCRC検査を持ちます。

この転送規約は受信した毎回のパケットが正しく照合されることを保証しますが、全てのパケットが受信されることを保証しません。再送と消失パケットは転送層ではなく、応用層によって扱われます。USB使用時、ハードウェア層は自動的に再送を処理します。これはRS232の場合ではありません。

この規約は"リトル エンディアン"で、メモリ アドレスに於いて常にLSBが先に送られます。

2.1. USBとRS-232接続

JTAGICEmk II はホスト コンピュータへ接続するためにUSBとRS-232の両インターフェースを持ちます。これらは同じメッセージ形式を共有します(3章をご覧ください)。

速度のためにUSBが好まれます。

2.1.1. RS-232接続

RS-232接続の電源投入時設定は19200bps、8ビット データ、パリティなし、1停止ビット、ハンドシェイクなしです。けれども、[エミュレータ パラメータ書き込み\(CMND_SET_PARAMETER\)命令](#)と[ボーレート パラメータ](#)を使ってボーレートを変更することができます。

2.1.2. USB接続

JTAGICEmk II はUSB全速(Full-speed:12Mbps)の能力があるUSBポートを持ちます。

データ転送に加えて、USBケーブルはエミュレータに電力を供給することができ、このために外部電力が必要とされません。列挙(接続)後、エミュレータはUSBハブから最大500mA引き出せます。

2.1.3. USB構成設定

JTAGICEmk II は2つの大量(バルク)転送エンドポイントを含む1つのインターフェースを持つ1つの構成設定を持ちます。USB記述子は7章で得られます。

JTAGICEmk II のメッセージは大量転送エンドポイントとして転送されます。最大パケット容量は[エンドポイント記述子](#)で示されます。



8ビット **AVR**[®]
マイクロコントローラ

応用記述

本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 2587D-11/09, 2587DJ3-05/21

JTAGICEmk II のフレーム形式によって提供されるフレーム構造に加えて、エミュレータはUSBパケットがJTAGINEmk II フレームでの最後の1つであることを示すために(INとOUTの両エンドポイントで)短いパケットを使います。ホストドライバはこの短いパケットも支援しなければなりません。

3. メッセージ形式

全ての**命令**、**応答**、**事象**は共通のメッセージ形式を共有します。

<MESSAGE_START, SEQUENCE_NUMBER, MESSAGE_SIZE, TOKEN, MESSAGE_BODY, CRC>

表3-1. メッセージの構成物

構成物名	使用法	形式
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)	[バイト]
順序番号 (SEQUENCE_NUMBER)	メッセージ送出毎に1増加。\$FFFE後、0へ丸め。\$FFFFは(非同期)事象用に予約。	[バイト]×2、LSB先行
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の容量	[バイト]×4、LSB先行
証拠印 (TOKEN)	規約をもっと強化する、ASCIIのSO文字(\$0E)	[バイト]
メッセージ本体 (MESSAGE_BODY)	メッセージ本体	[バイト]×メッセージ量
循環冗長符号 (CRC)	メッセージ開始とメッセージ本体を含むメッセージ内全文字を使用	[バイト]×2

メッセージ本体の先頭バイトは常にメッセージ識別子(MESSAGE_ID)です。(従って、)メッセージ本体(MESSAGE_BODY)の量は最低1バイトです。

- メッセージは\$01～\$3Fの範囲のメッセージ識別子(MESSAGE_ID)を使います。
- 成功応答は\$80～\$9Fの範囲のメッセージ識別子(MESSAGE_ID)を使います。
- 失敗応答は\$A0～\$BFの範囲のメッセージ識別子(MESSAGE_ID)を使います。
- 事象は\$E0～\$FFの範囲のメッセージ識別子(MESSAGE_ID)を使います。
- メッセージ開始はASCIIのESC文字(\$1B)で、変えられるべきではありません。
- 証拠印はASCIIのSO文字(\$0E)で、変えられるべきではありません。

4. メッセージ解析

4.1. 序説

どちらにしてもメッセージは状態機構により、バイト列として読まれるべきです。メッセージ開始(MESSAGE_START)文字が起きる毎に状態機構はメッセージの復号を開始し、メッセージを完了するか、またはメッセージ規則に違反した場合にバイト列を中止するかのどちらかです。メッセージはどんな順序で来ることもでき、命令間で必要とする時間枠に関する特別なタイミングはありません。エミュレータはメッセージ待ち行列を実装し、状態機構が解析、検証された完全なメッセージを持つ時に、それは順序付けされてメッセージ待ち行列に置かれます。エミュレータは待ち行列で得られた順にメッセージを実行します。メッセージに対する応答はそれらがエミュレータによって処理される時に与えられます。

RS232とUSBの両転送単位処理に同じ状態機構が使われます。JTAGICEmk II の通信制御プロセッサ上の割り込みルーチンは適切なUSARTを読んで、やって来るデータをSRAMに実装されたFIFOに置きます。状態機構がこのFIFOからデータを読んで解析します。

4.2. 共通状態機構実装

図4-1. 状態遷移図

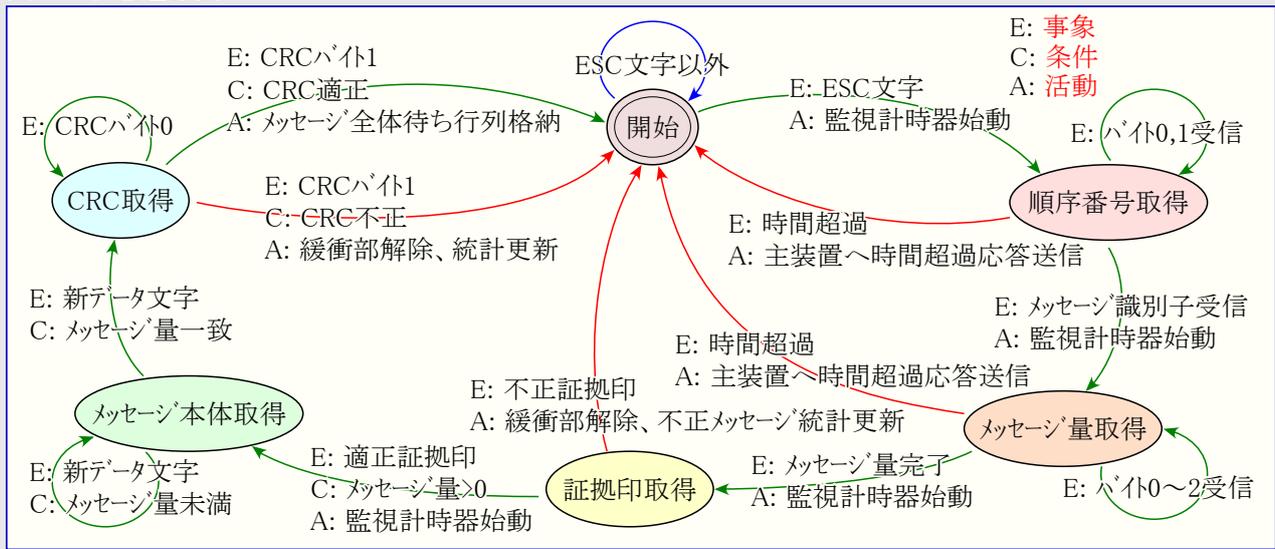


表4-1. 状態表 (赤背景の領域は失敗メッセージを表し、その他は成功メッセージです。)

現在の状態	事象	条件	活動	次の状態
開始	入力緩衝部から文字取得	文字=ASC ESC(\$1B)	監視計時器始動	順序番号取得
	入力緩衝部から文字取得	文字≠ASC ESC(\$1B)	なし	開始
順序番号取得	入力緩衝部から文字取得	順序番号バイト計数器<2	監視計時器始動	順序番号取得
	入力緩衝部から文字取得	順序番号バイト計数器=2	監視計時器始動	メッセージ量取得
	時間超過	なし	メッセージ統計更新	開始
メッセージ量取得	入力緩衝部から文字取得	メッセージ量バイト計数器<4	監視計時器始動	メッセージ量取得
	入力緩衝部から文字取得	メッセージ量バイト計数器=4	監視計時器始動,メッセージ量計算	証拠印取得
	時間超過	なし	メッセージ統計更新	開始
証拠印取得	入力緩衝部から文字取得	文字=ASC SO(\$0E)	監視計時器始動	本体データ取得
	入力緩衝部から文字取得	文字≠ASC SO(\$0E)	監視計時器停止,メッセージ統計更新	開始
	時間超過	なし	メッセージ統計更新	開始
本体データ取得	入力緩衝部から文字取得	データバイト計数器<メッセージ量	監視計時器始動	本体データ取得
	入力緩衝部から文字取得	データバイト計数器=メッセージ量	監視計時器始動	CRC取得
	時間超過	なし	メッセージ統計更新	開始
CRC取得	入力緩衝部から文字取得	CRCバイト計数器<2	監視計時器始動	CRC取得
	入力緩衝部から文字取得	CRCバイト計数器=2,CRC正常	監視計時器停止,CRC計算,メッセージ統計更新,命令実行	開始
	入力緩衝部から文字取得	CRCバイト計数器=2,CRC異常	監視計時器停止,CRC計算,メッセージ統計更新	開始
	時間超過	なし	メッセージ統計更新	開始

5. 命令と応答

5.1. 主装置命令

5.1.1. 停止命令(CMND_SIGN_OFF) [\$00]

デバッグ作業を終了する時に主装置がこの命令を発行します。

従装置の状態：全状態

表5-1. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	なし
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-2. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SIGN_OFF (\$00)	[バイト]

表5-3. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令は実行されました。
失敗 (RSP_FAILED)	JTAGICEmk IIは存在しますが、命令を理解できません。

5.1.2. エミュレータ存在検査命令(CMND_GET_SIGN_ON) [\$01]

エミュレータが接続されて生きているかを見るために主装置がこの命令を使います。

従装置の状態：全状態

表5-4. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	なし
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-5. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_GET_SIGN_ON (\$01)	[バイト]

表5-6. 応答パラメータ

パラメータ名	説明
開始 (RSP_SIGN_ON)	JTAGICEmk II に関する開始文字列を含みます。 他のエミュレータで発行される共通性が解決されなければなりません。 開始 (RSP_SIGN_ON : \$86)
失敗 (RSP_FAILED)	JTAGICEmk II は存在しますが、命令を理解できません。
時間超過	定義された制限時間内に従装置が全く応答しません。 従装置は存在しないか、または電源ONではありません。
ゴミ受信	RS232使用なら、ハードウェア異常、UART構成設定誤り(不正なボーレート、パリティ設定など)。USB使用なら、ハードウェア異常。

5.1.3. エミュレータパラメータ書き込み命令(CMND_SET_PARAMETER) [\$02]

エミュレータは多数の構成パラメータを処理します。この命令は全パラメータ書き込みに使われます。パラメータ識別子(PARAMETER_ID)が書かれるべきパラメータを識別します。

従装置の状態：パラメータに依存

表5-7. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PARAMETER_ID, PARAMETER_VALUE>

表5-8. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SET_PARAMETER (\$02)	[バイト]
パラメータ識別子 (PARAMETER_ID)	パラメータ識別子	[バイト]
パラメータ値 (PARAMETER_VALUE)	パラメータ値、どんな量にもできます。	[バイト]×n、LSB先行

表5-9. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	パラメータが正常に書かれました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正パラメータ (RSP_ILLEGAL_PARAMETER)	JTAGICEmk II は選択したパラメータを支援しません。 (AVR Studio 4とJTAGICEmk II のファームウェア不適合)
不正値 (RSP_ILLEGAL_VALUE)	与えられた値が無効または範囲外でした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で書けません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

パラメータの説明はこの資料の[後ろの章](#)で得られます。

5.1.4. エミュレータパラメータ読み込み命令(CMND_GET_PARAMETER) [\$03]

エミュレータ構成パラメータをAVR Studioに読み戻します。パラメータの説明はこの資料の後ろの章で得られます。

従装置の状態：パラメータに依存

表5-10. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	\$02
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PARAMETER_ID>

表5-11. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_GET_PARAMETER (\$03)	[バイト]
パラメータ識別子 (PARAMETER_ID)	要求されたパラメータ識別子	[バイト]

表5-12. 応答パラメータ

パラメータ名	説明
パラメータ値 (RSP_PARAMETER)	要求パラメータを返します。パラメータ返答 (RSP_PARAMETER : \$81)
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正パラメータ (RSP_ILLEGAL_PARAMETER)	JTAGICEmk II は選択したパラメータを支援しません。 (AVR Studio 4とJTAGICEmk II のファームウェア不適合)
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で読めません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.5. メモリ書き込み命令(CMND_WRITE_MEMORY) [\$04]

何れかのメモリ領域内の何れかのアドレスへメモリ塊を書きます。

従装置の状態：停止(STOPPED)

表5-13. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, MEMORY_TYPE, BYTE_COUNT, START_ADDRESS, DATA>

表5-14. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_WRITE_MEMORY (\$04)	[バイト]
メモリ種別 (MEMORY_TYPE)	メモリ種別(フラッシュ,SRAM,EEPROM...)	[バイト]
バイト数 (BYTE_COUNT)	書かれるべきバイト数	[バイト]×4、LSB先行
開始アドレス (START_ADDRESS)	開始メモリ アドレス	[バイト]×4、LSB先行
データ (DATA)	データ	[バイト]×バイト数

各種メモリ形式の定義とパラメータ値はメモリ形式章で得られます。

表5-15. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	メモリが正常に書かれました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正メモリ種別 (RSP_ILLEGAL_MEMORY_TYPE)	JTAGICEmk IIは選択したメモリを支援しません。
不正メモリ範囲 (RSP_ILLEGAL_MEMORY_RANGE)	メモリ書き込みは選択したメモリの領域範囲外でした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で書けません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.6. メモリ読み出し命令(CMND_READ_MEMORY) [\$05]

何れかのメモリ領域内の何れかのアドレスからメモリ塊を読みます。

従装置の状態：停止(STOPPED)

表5-16. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	10 (\$0A)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, MEMORY_TYPE, BYTE_COUNT, START_ADDRESS>

表5-17. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_READ_MEMORY (\$05)	[バイト]
メモリ種別 (MEMORY_TYPE)	メモリ種別(フラッシュ,SRAM,EEPROM...)	[バイト]
バイト数 (BYTE_COUNT)	読まれるべきバイト数	[バイト]×4、LSB先行
開始アドレス (START_ADDRESS)	開始メモリ アドレス	[バイト]×4、LSB先行

各種メモリ形式の定義とパラメータ値はメモリ形式章で得られます。

表5-18. 応答パラメータ

パラメータ名	説明
メモリ読み出し成功 (RSP_MEMORY)	メモリが正常に読まれました。メモリ読み出し (RSP_MEMORY: \$82)
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正メモリ種別 (RSP_ILLEGAL_MEMORY_TYPE)	JTAGICEmk IIは選択したメモリを支援しません。
不正メモリ範囲 (RSP_ILLEGAL_MEMORY_RANGE)	メモリ読み込みは選択したメモリの領域範囲外でした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.7. プログラム カウンタ書き込み命令(CMND_WRITE_PC) [\$06]

AVRのプログラム カウンタを書きます。

従装置の状態：停止(STOPPED)

表5-19. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証抛印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PROGRAM_COUNTER>

表5-20. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_WRITE_PC (\$06)	[バイト]
プログラム カウンタ (PROGRAM_COUNTER)	プログラム カウンタ値	[バイト]×4、LSB先行

表5-21. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	プログラム カウンタが正常に書かれました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.8. プログラム カウンタ読み出し命令(CMND_READ_PC) [\$07]

AVRのプログラム カウンタを読みます。

従装置の状態：停止(STOPPED)

表5-22. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証抛印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-23. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_READ_PC (\$07)	[バイト]

表5-24. 応答パラメータ

パラメータ名	説明
プログラム カウンタ読み込み成功 (RSP_PC)	プログラム カウンタが返されました。プログラム カウンタ読み (RSP_PC : \$84)
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.9. プログラム実行開始命令(CMND_GO) [\$08]

現在のプログラム カウンタでプログラム実行を開始します。

従装置の状態：停止(STOPPED)または走行(RUNNING)。命令実行時に状態は走行(RUNNING)に変更されるか、または走行(RUNNING)に留まるかのどちらかです。

表5-25. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	なし
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-26. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_GO (\$08)	[バイト]

表5-27. 応答パラメータ

パラメータ名	説明
実行開始成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.10. 1段実行命令(CMND_SINGLE_STEP) [\$09]

現在のプログラム カウンタ アドレスで1段実行を開始します。

従装置の状態：停止(STOPPED)。実行後、従MCUは次の中断点(ブレークポイント)に当たるまで走行(RUNNING)状態です。そして**中断事象**が発行され、従MCUは停止(STOPPED)に戻ります。

表5-28. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	3 (\$03)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	高位または低位のフラグ。以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, FLAG>

表5-29. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SINGLE_STEP (\$09)	[バイト]
フラグ (FLAG)	\$01: 低位, \$02: 高位	[バイト]
段動作種別 (STEP_MODE)	\$00: STEP_OVER (外側1行実行) \$01: STEP_INT0 (1命令実行) \$02: STEP_OUT (内側実行)	[バイト]

低位または上位のデバッグ(デバッグWIRE、けれどもJTAGICEの次の実装も)

表5-30. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.11. プログラム実行停止命令(CMND_FORCED_STOP) [\$0A]

プログラム実行を停止します。

従装置の状態：停止(STOPPED)または走行(RUNNING)。命令実行後の従MCUは停止(STOPPED)状態です。

表5-31. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, MODE>

表5-32. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_FORCED_STOP (\$0A)	[バイト]
動作種別 (MODE)	\$01: 低位, \$02: 高位(次の高位行で停止)	[バイト]

表5-33. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.12. 使用者プログラムリセット命令(CMND_RESET) [\$0B]

エミュレータはプログラム実行を再始動するための全ての動作を実行します。

従装置の状態：何れか。この操作前に従MCUが走行(RUNNING)動作なら、従MCUはこの命令の実行前に停止(STOPPED)状態に変わります。

表5-34. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。 上位または下位のフラグについて、 上位は代表的にmainへ戻ります(手続き(プロシージャ)を与えます。 後で走行するリセットを下位リセットを与えます。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, FLAG>

表5-35. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_RESET (\$0B)	[バイト]
フラグ (FLAG)	\$01: 低位, \$02: 高位(リセット後にmainを走行), \$04: デバッグWIRE禁止でリセット	[バイト]

表5-36. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.13. デバイス記述子設定命令(CMND_SET_DEVICE_DESCRIPTOR) [\$0C]

特定デバイス構成に関連する全パラメータを転送します。

従装置の状態：停止(STOPPED)

表5-37. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	特定デバイス構成に関する全パラメータ
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PARAMETERS>

表5-38. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SET_DEVICE_DESCRIPTOR (\$0C)	[バイト]
パラメータ群 (PARAMETERS)	現在のJTAGICE記述子と同様に、2進数列形式でデバイスを構成するのに必要とする全てのパラメータ	[バイト] × 298

表5-39. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.14. SPMページ消去命令(CMND_ERASEPAGE_SPM) [\$0D]

フラッシュメモリのページ全体を消去します。

従装置の状態：停止(STOPPED)

表5-40. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PAGE_ADDRESS>

表5-41. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_ERASEPAGE_SPM (\$0D)	[バイト]
ページアドレス (PAGE_ADDRESS)	消去すべきページのアドレス	[バイト]×4、LSB先行

表5-42. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.15. 同期獲得命令(CMND_GET_SYNC) [\$0F]

同期を失った場合にJTAGICEmk II との同期を復活するためにAVR Studioから送られます。

従装置の状態：何れか。従装置がこの命令に応答する場合、従装置は命令実行後に停止(STOPPED)状態になります。

表5-43. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-44. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_GET_SYNC (\$0F)	[バイト]

表5-45. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.16. 自己検査命令(CMND_SELFTEST) [\$10]

JTAGICEmk II に自己検査を実行させ、主装置へ報告を返します。

従装置の状態：停止(STOPPED)

表5-46. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	(何の検査かを選ぶ)自己検査用パラメータ
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, FLAGS>

表5-47. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SELFTEST (\$10)	[バイト]
フラグ (FLAGS)	どの検査が実行されるべきかを識別するフラグ。 ビット7: 内部検査(SRAM,FIFOなど) ビット6: (未使用) ビット5: (未使用) ビット4: (未使用) ビット3: STK500 リセット ジャンパ検出器 ビット2: JTAG駆動 ビット1: デバッグWIRE容量(キャパシタンス) ビット0: デバッグWIRE駆動	[バイト]×4、LSB先行

表5-48. 応答パラメータ

パラメータ名	説明
自己検査成功 (RSP_SELFTEST)	命令が実行されました。自己検査 (RSP_SELFTEST : \$85)
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.17. 中断点(ブレイクポイント)設定命令(CMND_SET_BREAK) [\$11]

(JTAGインターフェースを持つmegaAVRデバイスのハードウェア中断点だけとソフトウェア中断点に関して)中断点(ブレイクポイント)を設定します。
従装置の状態：停止(STOPPED)

表5-49. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	中断点設定(下の形式をご覧ください。)
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, TYPE, NUMBER, ADDRESS, MODE>

表5-50. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SET_BREAK (\$11)	[バイト]
形式 (TYPE)	\$01: プログラムメモリ中断点, \$02: データ中断点, \$03: データ中断点用遮蔽値	[バイト]
中断点番号 (NUMBER)	中断点番号: \$01~\$04(4つまでの中断点を支援。\$00はソフトウェア中断点用)	[バイト]
アドレス (ADDRESS)	中断点アドレスまたは遮蔽値	[バイト]×4、LSB先行
動作種別 (MODE)	\$00: 読み込みで中断, \$01: 書き込みで中断, \$02: 読み書きで中断, \$03: プログラム中断点	[バイト]

表5-51. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正中断点 (RSP_ILLEGAL_BREAKPOINT)	中断点番号が存在しません。(この動作種別で)中断点番号は支援されません。不正中断点 (RSP_ILLEGAL_BREAKPOINT : \$A8)
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.18. 中断点(ブレイクポイント)取得命令(CMND_GET_BREAK) [\$12]

JTAGICEmk II内に設定されている現在の中断点(ブレイクポイント)を読み戻すためにホストによって使われます(JTAGインターフェースを持つmegaAVRデバイスのハードウェア中断点だけとソフトウェア中断点に関して)。

従装置の状態：停止(STOPPED)

表5-52. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, NUMBER>

表5-53. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_GET_BREAK (\$12)	[バイト]
中断点番号 (NUMBER)	中断点番号: \$01~\$04(4つまでの中断点が支援されます。)	[バイト]

表5-54. 応答パラメータ

パラメータ名	説明
中断点取得成功 (RSP_GET_BREAK)	命令が実行されました。中断点が返されます。 中断点読み出し (RSP_GET_BREAK : \$83)
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正中断点 (RSP_ILLEGAL_BREAKPOINT)	(この動作種別で)中断点番号は支援されません。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.19. チップ消去命令(CMND_CHIP_ERASE) [\$13]

完全なチップ消去を行うのに(JTAGインターフェースを持つmegaAVRデバイスでだけ)使われます。(デバッグWIREを持つデバイスはプログラミングの前にフラッシュ ページを自動的に消去します。XMEGAデバイスはこの資料の後ろで詳細にされる専用のXMEGA消去機能を使います。)

表5-55. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-56. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_CHIP_ERASE (\$13)	[バイト]

表5-57. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.20. プログラミング動作移行命令(CMND_ENTER_PROGMODE) [\$14]

プログラミング動作へ移行します。

表5-58. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-59. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_ENTER_PROGMODE (\$14)	[バイト]

表5-60. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正JTAG ID (RSP_ILLEGAL_JTAG_ID)	JTAG IDが目的デバイスのそれと一致しません。 不正JTAG ID (RSP_ILLEGAL_JTAG_ID : \$A9)
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.21. プログラミング動作抜け出し命令(CMND_LEAVE_PROGMODE) [\$15]

プログラミング動作を抜け出します。

表5-61. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-62. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_LEAVE_PROGMODE (\$15)	[バイト]

表5-63. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.22. 中断点(ブレークポイント)解除命令(CMND_CLR_BREAK) [\$1A]

(JTAGインターフェースを持つmegaAVRデバイスのハードウェア中断点だけとソフトウェア中断点に関して)中断点(ブレークポイント)を解除します。
従装置の状態：停止(STOPPED)

表5-64. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	中断点指定 (以下をご覧ください。)
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, NUMBER, ADDRESS>

表5-65. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_CLR_BREAK (\$1A)	[バイト]
中断点番号 (NUMBER)	中断点番号: \$01~\$04(4つまでの中断点が支援されます。)	[バイト]
中断点アドレス (ADDRESS)	中断点アドレス(ソフトウェア中断点用)	[バイト]×4、LSB先行

表5-66. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正中断点 (RSP_ILLEGAL_BREAKPOINT)	中断点番号が存在しません。(この動作種別で)中断点番号は支援されません。不正中断点 (RSP_ILLEGAL_BREAKPOINT : \$A8)
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.23. 指定アドレスまで実行命令(CMND_RUN_TO_ADDR) [\$1C]

現在のプログラムカウンタアドレスで実行を開始し、与えられたプログラムカウンタアドレスまで走行します。

従装置の状態：停止(STOPPED)。命令が実行されると、状態は走行(RUNNING)に変わります。

表5-67. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PCアドレス>

表5-68. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_RUN_TO_ADDR (\$1C)	[バイト]
停止プログラムカウンタ値	停止アドレス	[バイト]×4、LSB先行

表5-69. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.24. 万能SPI命令(CMND_SPI_CMD) [\$1D]

万能SPI命令を実行します。

従装置の状態：停止(STOPPED)

表5-70. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証抛印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, COMMAND[4]>

表5-71. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SPI_CMD (\$1D)	[バイト]
SPI命令 (COMMAND)	4バイトSPI命令	[バイト]×4

表5-72. 応答パラメータ

パラメータ名	説明
SPI返答成功 (RSP_SPL_DATA)	命令が実行されました。SPIデータ返答 (RSP_SPL_DATA : \$88)
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません。(SPI動作でなければなりません。)
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.25. 事象メモリ解除命令(CMND_CLEAR_EVENTS) [\$22]

JTAGICEmk II に(ソフトウェア中断点(ブレイクポイント)用の)事象メモリ解除を指示します。必要とする量のメモリだけが解除されることを保証するため、デバイス記述子設定後に呼ばなければなりません。

表5-73. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証抛印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-74. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_CLEAR_EVENTS (\$22)	[バイト]

表5-75. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.26. 目的対象回復命令(CMND_RESTORE_TARGET) [\$23]

JTAGICEmk II に目的デバイスを自由に走行できる状態への回復を指示します。これは装置停止前と(JTAGに関して)OCDを禁止する前に、デバッグ作業の終了で行われることです。この命令は以下を行います。

- 目的対象停止
- ソフトウェア中断点を元の命令で置換(回復)
- 目的対象リセット
- 目的対象走行

表5-76. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-77. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_RESTORE_TARGET (\$23)	[バイト]

表5-78. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.1.27. JTAG命令書き込み命令(CMND_JTAG_INSTR) [\$24]

IR(JTAG命令レジスタ)へJTAG命令を書きます。

表5-79. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, IR_VALUE>

表5-80. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_INSTR (\$24)	[バイト]
IR値 (IR_VALUE)	IRへ書くべき値	[バイト]

表5-81. 応答パラメータ

パラメータ名	説明
成功 (RSP_SCAN_CHAIN_READ)	命令が実行され、データが返されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.28. JTAGデータ読み書き命令(CMND_JTAG_DATA) [\$25]

DR(JTAGデータレジスタ)とデータを読み書きします。

表5-82. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, BITS, DATA>

表5-83. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_DATA (\$25)	[バイト]
ビット数 (BITS)	読み書きするビット数	[バイト]
データ (DATA)	DRへ書くデータ	[バイト]×4

表5-84. 応答パラメータ

パラメータ名	説明
成功 (RSP_SCAN_CHAIN_READ)	命令が実行され、データが返されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.29. SABデータ書き込み命令(CMND_JTAG_SAB_WRITE) [\$28]

AVR32のSABへデータを書きます。

表5-85. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	10 (\$0A)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, ADDRESS, DATA>

表5-86. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_SAB_WRITE (\$28)	[バイト]
SABアドレス (ADDRESS)	データを書くSABアドレス	[バイト]×5
データ (DATA)	SABへ書くデータ	[バイト]×4

表5-87. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正値 (RSP_ILLEGAL_VALUE)	与えられた値が無効または範囲外でした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.30. SABデータ読み込み命令(CMND_JTAG_SAB_READ) [\$29]

AVR32のSABへデータを書きます。

表5-88. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	6 (\$06)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, ADDRESS>

表5-89. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_SAB_WRITE (\$28)	[バイト]
SABアドレス (ADDRESS)	データを読むSABアドレス	[バイト]×5

表5-90. 応答パラメータ

パラメータ名	説明
成功 (RSP_SCAN_CHAIN_READ)	命令が実行され、データが返されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.31. JTAGデータ塊読み込み命令(CMND_JTAG_BLOCK_READ) [\$2C]

AVR32のSABからデータの塊を読みます。

表5-91. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	7 (\$07)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, SIZE, ADDRESS>

表5-92. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_BLOCK_READ (\$2C)	[バイト]
量 (SIZE)	読む32ビット語数	[バイト]
SABアドレス (ADDRESS)	データを読むSABアドレス	[バイト]×5

表5-93. 応答パラメータ

パラメータ名	説明
成功 (RSP_SCAN_CHAIN_READ)	命令が実行され、データが返されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.32. JTAGデータ塊書き込み命令(CMND_JTAG_BLOCK_WRITE) [\$2D]

AVR32のSABへデータの塊を書きます。

表5-94. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, ADDRESS, DATA>

表5-95. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_JTAG_SAB_WRITE (\$28)	[バイト]
SABアドレス (ADDRESS)	(埋め込みのために)データを書くSABアドレス	[バイト]×8
データ (DATA)	SABへ書くデータ	[バイト]×4

表5-96. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正値 (RSP_ILLEGAL_VALUE)	与えられた値が無効または範囲外でした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
時間超過	制限時間内に従装置が全く応答しません。

5.1.33. カプセル化SPI命令(CMND_ISP_PACKET) [\$2F]

JTAGICEmk IIへ何れかのISP命令を送るのに使われます。AVRISPmk II 互換のISPパケットがJTAGICEmk IIパケットにカプセル化されます。そしてJTAGICEmk IIはAVRISPmk IIと同じ様にISPパケットで動作します。表5-100.はJTAGICEmk IIで支援されるISP命令を示します。

ISP命令に関する更なる情報については「AVR069:AVRISPmk II 通信規約」資料を調べてください。

表5-97. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, <ISP_PACKET>>

表5-98. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_SPL_PACKET (\$2F)	[バイト]
ISPパケット (ISP_PACKET)	カプセル化されたAVRISPデータ	[バイト]×n

表5-99. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk IIによって理解されませんでした。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

表5-100. JTAGICEmk II で支援されるAVRISPmk II 命令

命令	使用法
パラメータ設定命令 (CMD_SET_PARAMETER)	パラメータ設定
パラメータ取得命令 (CMD_GET_PARAMETER)	パラメータ取得
発振校正命令 (CMD_OSCCAL)	OSCCAL校正手順実行
アドレス指定命令 (CMD_LOAD_ADDRESS)	プログラミング機能用アドレス格納
ISPプログラミング動作移行命令 (CMD_ENTER_PROGMODE_ISP)	目的デバイスをプログラミング動作に置きます。
ISPプログラミング動作抜け出し命令 (CMD_LEAVE_PROGMODE_ISP)	目的デバイスはプログラミング動作を抜け出します。
ISPチップ消去命令 (CMD_CHIP_ERASE_ISP)	目的デバイスでチップ消去を実行
ISPフラッシュメモリ書き込み命令 (CMD_PROGRAM_FLASH_ISP)	目的デバイスのフラッシュメモリ書き込み
ISPフラッシュメモリ読み出し命令 (CMD_READ_FLASH_ISP)	目的デバイスのフラッシュメモリ読み出し
ISPEEPROM書き込み命令 (CMD_PROGRAM_EEPROM_ISP)	目的デバイスのEEPROM書き込み
ISPEEPROM読み出し命令 (CMD_READ_EEPROM_ISP)	目的デバイスのEEPROM読み出し
ISPヒューズビット書き込み命令 (CMD_PROGRAM_FUSE_ISP)	目的デバイスのヒューズ書き込み
ISPヒューズビット読み出し命令 (CMD_READ_FUSE_ISP)	目的デバイスのヒューズ読み出し
ISP施錠ビット書き込み命令 (CMD_PROGRAM_LOCK_ISP)	目的デバイスの施錠ビット書き込み
ISP施錠ビット読み出し命令 (CMD_READ_LOCK_ISP)	目的デバイスの施錠ビット読み出し
ISP識票バイト読み出し命令 (CMD_READ_SIGNATURE_ISP)	目的デバイスの識票バイト読み出し
ISP発振校正値読み出し命令 (CMD_READ_OSCCAL_ISP)	目的デバイスの発振校正値バイト読み出し
ISP複数命令 (CMD_SPL_MULT)	ISP命令実行に使用される標準命令

5.1.34. XMEGA消去命令(CMND_XMEGA_ERASE) [\$34]

XMEGAの消去機能。

従装置の状態：プログラミング(PROGRAMMING)

表5-101. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証抛印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, ERASE_MODE, ADDRESS>

表5-102. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	CMND_XMEGA_ERASE (\$34)	[バイト]
消去種別 (ERASE_MODE)	\$00 : XMEGA_ERASE_CHIP \$01 : XMEGA_ERASE_APP \$02 : XMEGA_ERASE_BOOT \$03 : XMEGA_ERASE_EEPROM \$04 : XMEGA_ERASE_APP_PAGE \$05 : XMEGA_ERASE_BOOT_PAGE \$06 : XMEGA_ERASE_EEPROM_PAGE \$07 : XMEGA_ERASE_USERSIG	[バイト]
アドレス (ADDRESS)	消去する領域の先頭位置(アドレス)	[バイト]×4、LSB先行

表5-103. 応答パラメータ

パラメータ名	説明
成功 (RSP_OK)	命令が実行されました。
失敗 (RSP_FAILED)	命令がJTAGICEmk II によって理解されませんでした。
不正エミュレータ動作 (RSP_ILLEGAL_EMULATOR_MODE)	操作はこのエミュレータ動作種別で実行できません(JTAG/デバッグWIRE)。
不正MCU状態 (RSP_ILLEGAL_MCU_STATE)	操作は現在の状態に於いて目的MCUで実行することができません。
時間超過	定義された制限時間内に従装置が全く応答しません。

5.2. 従装置応答

5.2.1. 成功(RSP_OK) [\$80]

受け取り通知: 従装置は命令を理解し、全てが正常でした。

表5-104. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-105. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_OK (\$80)	[バイト]

5.2.2. パラメータ返答(RSP_PARAMETER) [\$81]

エミュレータは要求されたパラメータを返します。

表5-106. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	以下をご覧ください。
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PARAMETER_VALUE>

表5-107. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_PARAMETER (\$81)	[バイト]
パラメータ値 (PARAMETER_VALUE)	パラメータ値	[バイト]×n

5.2.3. メモリ読み出し返答(RSP_MEMORY) [\$82]

エミュレータは要求されたメモリ塊を返します。

表5-108. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, DATA>

表5-109. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_MEMORY (\$82)	[バイト]
読み出しデータ値 (DATA)	メモリ読み出し値	[バイト]×n

5.2.4. 中断点(ブレイクポイント)読み出し返答(RSP_BREAK) [\$83]

エミュレータは要求された中断点(ブレイクポイント)だけを返します。

表5-110. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, TYPE, ADDRESS, MODE>

表5-111. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_BREAK (\$83)	[バイト]
形式 (TYPE)	\$01: プログラム メモリ中断点, \$02: データ中断点	[バイト]
アドレス (ADDRESS)	中断点アドレス	[バイト]×4、LSB先行
動作種別 (MODE)	\$00: (プログラム中断点-未使用), \$01: 読み込みで中断, \$02: 書き込みで中断, \$03: 読み書きで中断	[バイト]

5.2.5. プログラム カウンタ読み出し返答(RSP_PC) [\$84]

主装置へ現在のプログラム カウンタ値を返します。

表5-112. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	5 (\$05)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PROGRAM_COUNTER>

表5-113. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_PC (\$84)	[バイト]
プログラム カウンタ値 (PROGRAM_COUNTER)	プログラム カウンタ値	[バイト]×4、LSB先行

5.2.6. 自己検査返答(RSP_SELFTEST) [\$85]

自己検査の結果を返します。自己検査の内容は後で記述されます。自己検査は2つの目的を扱います。

1. 装置自身の健全性検証
2. 実際のデバッグ構成が動く(デバイス インターフェースが正常、動かない線がない、など)ことの検証

表5-114. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下の9部分から成ります。

<MESSAGE_ID, SELFTEST_0, ~, SELFTEST_7>

表5-115. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_SELFTEST (\$85)	[バイト]
自己検査0結果 (SELFTEST_0)	自己検査0結果 (TEST_SPECIFIC_RESULT	[バイト]
~	~	~
自己検査7結果 (SELFTEST_7)	自己検査7結果 (TEST_SPECIFIC_RESULT	[バイト]

自己検査命令(CMND_SELFTEST)によって8つまでの自己検査機能を選択することができます。各検査はそれが実行されたか否かに拘らず、応答のための1バイトを返します。個別の結果がSELFTEST値で、各特定検査に対して更に独自設定することができます。

表5-116. 返答

自己検査返答	値	意味
SELFTEST_SKIPPED	\$00	この自己検査は実行されませんでした。
SELFTEST_OK	\$01	自己検査合格
SELFTEST_FAILED	\$80	自己検査失敗
ST_USART_FAILURE	\$81	内部USART検査失敗
ST_FIFO_M_FAILURE	\$82	主装置側FOFO読み込み失敗
ST_FIFO_S_FAILURE	\$83	従装置側FOFO読み込み失敗
ST_FIFO_M_EMPTY_FAILURE	\$84	主装置側FOFO空ビット読み込み失敗
ST_FIFO_S_EMPTY_FAILURE	\$85	従装置側FOFO空ビット読み込み失敗
ST_FIFO_M_FULL_FAILURE	\$86	主装置側FOFO満ビット読み込み失敗
ST_FIFO_S_FULL_FAILURE	\$87	従装置側FOFO満ビット読み込み失敗
ST_FIFO_M_NINE_FAILURE	\$88	主装置側FOFO第9ビット読み込み失敗
ST_FIFO_S_NINE_FAILURE	\$89	従装置側FOFO第9ビット読み込み失敗
ST_SRAM_FAILURE	\$8A	内部SRAM読み書き検査失敗
ST_JTAG_TMS_STUCK_HIGH	\$8B	JTAG TMS線Low駆動不能
ST_JTAG_TCK_STUCK_HIGH	\$8C	JTAG TCK線Low駆動不能
ST_JTAG_TDI_STUCK_HIGH	\$8D	JTAG TDI線Low駆動不能
ST_JTAG_TMS_STUCK_LOW	\$8E	JTAG TMS線High駆動不能
ST_JTAG_TCK_STUCK_TMS	\$8F	JTAG線のTCKとTMSが共に結合の可能性
ST_JTAG_TDI_STUCK_TMS	\$90	JTAG線のTDIとTMSが共に結合の可能性
ST_JTAG_TCK_STUCK_LOW	\$91	JTAG TCK線High駆動不能
ST_JTAG_TMS_STUCK_TCK	\$92	JTAG線のTMSとTCKが共に結合の可能性
ST_JTAG_TDI_STUCK_TCK	\$93	JTAG線のTDIとTCKが共に結合の可能性
ST_JTAG_TDI_STUCK_LOW	\$94	JTAG TDI線High駆動不能
ST_JTAG_TMS_STUCK_TDI	\$95	JTAG線のTMSとTDIが共に結合の可能性
ST_JTAG_TCK_STUCK_TDI	\$96	JTAG線のTCKとTDIが共に結合の可能性

- JTAG自己検査メッセージは次の形式でしょう。
"JTAG接続失敗: %message. ICEと目的対象間の物理的な接続を調べてください。"
- デバッグWIRE容量(キャパシタンス)検査失敗の場合、次のようなメッセージが読めるでしょう。
"JTAGICEmkIIはデバッグWIRE通信に影響を及ぼすかもしれないRESET線上の高容量を検出しました。目的対象回路でリセット線から何れかの容量(コンデンサ)を取り去ってください。"
- デバッグWIRE駆動検査失敗の場合、次のようなメッセージが読めるでしょう。
"JTAGICEmkIIは目的応用に於いてリセット線をLowに引き込めません。強すぎる抵抗器(低すぎる抵抗値)の使用がこれを起こし得ます。推奨値は10kΩ台です。"

5.2.7. 開始返答(RSP_SIGN_ON) [\$86]

主装置の開始命令に対する応答。この応答の目的はどのデバイスが接続され、ハードウェアとファームウェアのどの版が使われるかをAVR Studioに告げることです。

表5-117. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, COMM_ID, M_MCU_BLDR, M_MCU_FW_MIN, M_MCU_FW_MAJ, M_MCU_HW, S_MCU_BLDR, S_MCU_FW_MIN, S_MCU_FW_MAJ, S_MCU_HW, SERIAL_NUMBER, DEVICE_ID_STR>

表5-118. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_SIGN_ON (\$86)	[バイト]
通信規約版番号 (COMM_ID)	通信規約版番号	[バイト]
主MCUフットローダ版番号 (M_MCU_BLDR)	主MCUフットローダ ファームウェア版番号	[バイト]
主MCUファームウェア副番号 (M_MCU_FW_MIN)	主MCUファームウェア版副番号	[バイト]
主MCUファームウェア主番号 (M_MCU_FW_MAJ)	主MCUファームウェア版主番号	[バイト]
主MCUハードウェア番号 (M_MCU_HW)	主MCUハードウェア版番号	[バイト]
従MCUフットローダ版番号 (S_MCU_BLDR)	従MCUフットローダ ファームウェア版番号	[バイト]
従MCUファームウェア副番号 (S_MCU_FW_MIN)	従MCUファームウェア版副番号	[バイト]
従MCUファームウェア主番号 (S_MCU_FW_MAJ)	従MCUファームウェア版主番号	[バイト]
従MCUハードウェア番号 (S_MCU_HW)	従MCUハードウェア版番号	[バイト]
通番 (SERIAL_NUMBER)	EEPROMに格納されたUSB用通し番号	[バイト]
装置識別文字列 (DEVICE_ID_STR)	"JTAGICE mkII",0 (0終端装置識別ASCII文字列)	[バイト]×n

5.2.8. 走査チェーン読み込み返答(RSP_SCAN_CHAIN_READ) [\$87]

成功したJTAG走査チェーンの直接読み込みからのデータを返します。

表5-119. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	バイト単位でのメッセージ本体の量
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, DATA>

表5-120. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_SPI_DATA (\$87)	[バイト]
データ値 (DATA)	JTAG走査チェーンからの読み込みデータ	[バイト]×n、LSB先行

5.2.9. SPI読み出しデータ返答(RSP_SPI_DATA) [\$88]

SPI命令からのクロック出力されたデータを返します。

表5-121. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, DATA>

表5-122. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_SPI_DATA (\$88)	[バイト]
データ値 (DATA)	SPI命令からクロック出力されたデータ	[バイト]

5.2.10. 失敗(RSP_FAILED) [\$A0]

従装置は命令を理解できず、何も行いません。メッセージ本体は実際に何が失敗したかの詳細情報を含めるために拡張でき、この点に於いて更新されるかもしれません。

表5-123. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-124. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_FAILED (\$A0)	[バイト]

5.2.11. 不正なパラメータ(RSP_ILLEGAL_PARAMETER) [\$A1]

主装置は存在しないエミュレータのパラメータの読み書きを試みました。

表5-125. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-126. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_PARAMETER (\$A1)	[バイト]

5.2.12. 不正メモリ形式アクセス(RSP_ILLEGAL_MEMORY_TYPE) [\$A2]

主装置は存在しないエミュレータのメモリ形式の読み書きを試みました。

表5-127. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-128. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_MEMORY_TYPE (\$A2)	[バイト]

5.2.13. 不正メモリ範囲アクセス(RSP_ILLEGAL_MEMORY_RANGE) [\$A3]

主装置は未支援のメモリ容量の読み書きを試みました。

表5-129. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-130. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_MEMORY_RANGE (\$A3)	[バイト]

5.2.14. エミュレータ動作種別による操作実行不能(RSP_ILLEGAL_EMULATOR_MODE) [\$A4]

主装置は現在のエミュレータ動作種別で実行できない操作を要求しました。

表5-131. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, CURRENT_MODE>

表5-132. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_EMULATOR_MODE (\$A4)	[バイト]
現在の動作種別 (CURRENT_MODE)	\$00: デバッグ(WIRE(EMULATOR_MODE_DEBUGWIRE)) \$01: JTAG(EMULATOR_MODE_JTAG) \$02: 未知(EMULATOR_MODE_UNKNOWN)	[バイト]

5.2.15. MCU状態による操作実行不能(RSP_ILLEGAL_MCU_STATE) [\$A5]

主装置は目的MCUの現在の動作状態で実行できない操作を要求しました。

表5-133. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, CURRENT_MODE>

表5-134. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_MCU_STATE (\$A5)	[バイト]
現在の動作状態 (CURRENT_MODE)	\$00: 停止(STOPPED) \$01: 走行(RUNNING) \$02: プログラミング(PROGRAMMING)	[バイト]

5.2.16. 不正値(RSP_ILLEGAL_VALUE) [\$A6]

主装置はエミュレータのパラメータに対して不正な値の書き込みを試みました。

表5-135. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-136. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_VALUE (\$A6)	[バイト]

5.2.17. 不正中断点(RSP_ILLEGAL_BREAKPOINT) [\$A8]

主装置は存在しない中断点(ブレイクポイント)の設定または取得を試みました。

表5-137. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-138. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_BREAKPOINT (\$A8)	[バイト]

5.2.18. 不正JTAG ID(RSP_ILLEGAL_JTAG_ID) [\$A9]

主装置は目的デバイスと一致しないJTAG IDでプログラミング動作への移行を試みました。

表5-139. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-140. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_JTAG_ID (\$A9)	[バイト]

5.2.19. 不正命令(RSP_ILLEGAL_COMMAND) [\$AA]

主装置は不正なエミュレータ命令の入出力、または未知の命令実行を試みました。

表5-141. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-142. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_COMMNAD (\$AA)	[バイト]

5.2.20. 目的対象不正電源状態(RSP_NO_TARGET_POWER) [\$AB]

主装置は目的デバイスが電源OFFまたは未接続で命令の実行を試みました。

表5-143. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-144. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_NO_TARGET_POWER (\$AB)	[バイト]

5.2.21. デバッグWIRE同期失敗(RSP_DEBUGWIRE_SYNC_FAILED) [\$AC]

主装置がデバッグWIRE動作への移行を試みましたが、リセット線に対する応答がありませんでした。デバッグWIREの同期が達成されませんでした。

デバッグWIREへのエミュレータ動作パラメータ設定命令(CMND_SET_PARAMETER)によってのみ、または他のデバッグWIRE動作命令の可能性で有り得ます。

表5-145. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-146. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_DEBUGWIRE_SYNC_FAILED (\$AC)	[バイト]

5.2.22. JTAGICEmk II 電力不足(RSP_ILLEGAL_POWER_STATE) [\$AD]

この応答はJTAGICEmk IIがUSB電力で動くけれど、100mA動作だけで接続された時のどの命令に対しても送られます。JTAGICEmk IIは節電動作で、どの命令に対しても通常のように応答しません。

表5-147. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	各パケット送出毎に1増加。
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表5-213. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	RSP_ILLEGAL_POWER_STATE (\$AD)	[バイト]

6. 事象

6.1. 序説

事象は他の方法で行う先行命令なしで、従装置(JTAGICEmk II)から主装置(AVR Studio)へ送られるメッセージです。事象の代表的な例は中断(BREAK)で、エミュレータは永遠に走行することができますが、急に中断条件が真になると、エミュレータは中断します。エミュレータは主装置へ中断(BREAK)事象を発行し、そして誰かが適切な動きを取ります。本章は全ての事象を一覧にします。

6.2. 事象

6.2.1. 中断事象(EVT_BREAK) [\$E0]

中断条件に出会うと、エミュレータは走行(RUNNING)状態から停止(STOPPED)状態になります。

表6-1. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	6 (\$06)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, PROGRAM_COUNTER, BREAK_CAUSE>

表6-2. メッセージ本体パラメータ (tinyAVR/megaAVRデバイス)

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_BREAK (\$E0)	[バイト]
プログラムカウンタ (PROGRAM_COUNTER)	プログラムカウンタ値	[バイト]×4、LSB先行
中断原因 (BREAK_CAUSE)	\$00=特定不能, \$01=プログラム中断, \$02=プログラムまたはデータ単独中断, \$03=プログラム,データまたは遮蔽単独中断	[バイト]

6.2.2. 目的対象電源ON事象(EVT_TARGET_POWER_ON) [\$E4]

エミュレータは目的対象の電源がOFFされてその後にONされた場合にこの事象を送ります。AVR Studioはこれにより、活動しなければなりません。

表6-3. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-4. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_TARGET_POWER_ON (\$E4)	[バイト]

6.2.3. 目的対象電源OFF事象(EVT_TARGET_POWER_OFF) [\$E5]

エミュレータは目的対象の電源がONされてその後にOFFされた場合にこの事象を送ります。AVR Studioはこれにより、活動しなければなりません。

表6-5. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-6. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_TARGET_POWER_OFF (\$E5)	[バイト]

6.2.4. 目的対象外部リセット事象(EVT_EXT_RESET) [\$E7]

目的対象が(未だ走行中、)外部的にリセットされました。

表6-7. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-8. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_EXT_RESET (\$E7)	[バイト]

6.2.5. 目的対象休止移行事象(EVT_TARGET_SLEEP) [\$E8]

目的対象は走行中ですが、休止状態(動作)へ移行しました。

表6-9. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-10. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_TARGET_SLEEP (\$E8)	[バイト]

6.2.6. 目的対象起動復帰事象(EVT_TARGET_WAKEUP) [\$E9]

目的対象が休止状態(動作)から起き上がりました。

表6-11. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-12. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_TARGET_WAKEUP (\$E9)	[バイト]

6.2.7. 不正電源状態事象(EVT_POWER_ERROR_STATE) [\$EA]

JTAGICEmk II が不正な電源状態に入り(例えば、非給電USBハブからの電力取得の試み)、デバッグを継続することができません。

表6-13. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-14. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_ICE_POWER_ERROR_STATE (\$EA)	[バイト]

6.2.8. 電源正常事象(EVT_POWER_OK) [\$EB]

JTAGICEmk II 電力供給元は全操作に関して大丈夫です。

表6-15. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-16. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_ICE_POWER_OK (\$EB)	[バイト]

6.2.9. IDR書き込み事象(EVT_IDR_DIRTY) [\$EC]

目的デバイスはIDRレジスタを書かれました。

表6-17. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	2 (\$02)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID, IDR値>

表6-18. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_IDR_DIRTY (\$EC)	[バイト]
IDR値 (IDR)	目的応用によってIDRに書かれた値	[バイト]

6.2.10. デバッグWIRE異常事象(EVT_ERROR_PHY_x) [\$xx]

表6-19. 異常事象

事象名	値
EVT_ERROR_PHY_FORCE_BREAK_TIMEOUT	\$E2
EVT_ERROR_PHY_RELEASE_BREAK_TIMEOUT	\$E3
EVT_ERROR_PHY_MAX_BIT_LENGTH_DIFF	\$ED
EVT_ERROR_PHY_SYNC_TIMEOUT	\$F0
EVT_ERROR_PHY_SYNC_TIMEOUT_BAUD	\$F4
EVT_ERROR_PHY_SYNC_OUT_OF_RANGE	\$F5
EVT_ERROR_PHY_SYNC_WAIT_TIMEOUT	\$F6
EVT_ERROR_PHY_RECEIVE_TIMEOUT	\$F7
EVT_ERROR_PHY_RECEIVED_BREAK	\$F8
EVT_ERROR_PHY_OPT_RECEIVE_TIMEOUT	\$F9
EVT_ERROR_PHY_OPT_RECEIVED_BREAK	\$FA
EVT_ERROR_PHY_NO_ACTIVITY	\$FB

デバッグWIRE異常事象はAVR Studioでデバッグ任意選択がチェックされている時にだけ見えるでしょう。

標準動作については異常状態発生時にただデバッグWIRE入出力失敗表示が起きます。使用者はこの低位異常メッセージで悩んではなりません。

表6-20. メッセージ形式

構成物名	説明
メッセージ開始 (MESSAGE_START)	ASCIIのESC文字(\$1B)
順序番号 (SEQUENCE_NUMBER)	(事象用に予約されている) \$FFFF
メッセージ量 (MESSAGE_SIZE)	1 (\$01)
証拠印 (TOKEN)	ASCIIのSO文字(\$0E)
メッセージ本体 (MESSAGE_BODY)	以下をご覧ください。
循環冗長符号 (CRC)	2バイトのCRC

メッセージ本体(MESSAGE_BODY)は以下から成ります。

<MESSAGE_ID>

表6-21. メッセージ本体パラメータ

パラメータ名	使用法	形式
メッセージ識別子 (MESSAGE_ID)	EVT_ERROR_PHY_x (\$E2~\$FB)	[バイト]

7. USB記述子

本章はJTAGICEmk IIのUSB通信関連で使われる記述子を一覧にします。

7.1. 装置記述子

表7-1. 装置記述子

変位	領域	値	説明
0	bLength	\$12	この記述子の容量は18バイトです。
1	bDescriptorType	\$01	装置記述子形式
2	bcdUSB	\$0110	装置はUSB仕様1.10版に適合
4	bDeviceClass	\$FF	装置クラスは供給者指定
5	bDeviceSubClass	\$00	各インターフェースがそれ自身の補助クラス情報を指定
6	bDeviceProtocol	\$00	装置基本規約なし
7	bMaxPacketSize0	\$10	エンドポイント0に対する最大パケット容量は16です。
8	idVendor	\$03EB	供給者識別は1003: Atmel Corporationです。
10	idProduct	\$2103	製品識別は\$2103です。
12	bcdDevice	\$0200	装置公開番号は2.00です。
14	iManufacturer	\$01	製造業者を記述する文字列記述子の指標は1です。
15	iProduct	\$02	製品を記述する文字列記述子の指標は2です。
16	iSerialNumber	\$03	通番を記述する文字列記述子の指標は3です。
17	bNumConfigurations	\$01	装置は利用可能な1つの構成設定を持ちます。

7.2. 構成設定記述子

表7-2. 構成設定記述子

変位	領域	値	説明
0	bLength	\$09	この記述子の容量は9バイトです。
1	bDescriptorType	\$02	構成設定記述子形式
2	wTotalLength	\$0020	本構成設定に関するデータ合計長は32です。 これは返される全記述子の組み合わせ長を含みます。
4	bNumInterface	\$01	この構成設定は1つのインターフェースを支援します。
5	bConfigurationValue	\$01	この構成設定を選択するのに値1が使われるべきです。
6	iConfiguration	\$00	装置はこの構成設定を記述する文字列記述子を持ちません。
7	bmAttributes	\$80	構成設定特性、ビット7: (予約)
8	bMaxPower	\$FA	この構成設定での装置の最大消費電力は500mAです。

7.3. インターフェース記述子

表7-3. インターフェース記述子

変位	領域	値	説明
0	bLength	\$09	この記述子の容量は9バイトです。
1	bDescriptorType	\$04	インターフェース記述子形式
2	bInterfaceNumber	\$00	このインターフェース番号は0です。
3	bAlternateSetting	\$00	本インターフェースに対する代替設定選択に使う値は0です。
4	bNumEndpoints	\$02	本インターフェースで使われるエンドポイント数は(エンドポイント0を除き)2です。
5	bInterfaceClass	\$FF	インターフェースクラスは供給者指定
6	bInterfaceSubClass	\$00	補助クラス符号は\$00です。
7	bInterfaceProtocol	\$00	本インターフェースはどのクラス特定規約も使いません。
8	iInterface	\$00	装置はこのインターフェースを記述する文字列記述子を持ちません。

7.4. INエンドポイント記述子

表7-4. INエンドポイント記述子

変位	領域	値	説明
0	bLength	\$07	この記述子の容量は7バイトです。
1	bDescriptorType	\$05	エンドポイント記述子形式
2	bEndpointAddress	\$82	これは(エンドポイント番号)2のアドレスを持つINエンドポイントです。
3	bmAttributes	\$02	形式 - 転送:大量(バルク)、同期:なし、使用法:データ エンドポイント
4	wMaxPacketSize	\$0040	本エンドポイントに対する最大パケット容量は\$40(ビット12,11:追加転送谷処理/フレーム)です。
6	bInterval	\$0A	ポーリング間隔は10(ms)です。(大量(バルク)時無効)

7.5. OUTエンドポイント記述子

表7-5. OUTエンドポイント記述子

変位	領域	値	説明
0	bLength	\$07	この記述子の容量は7バイトです。
1	bDescriptorType	\$05	エンドポイント記述子形式
2	bEndpointAddress	\$02	これは(エンドポイント番号)2のアドレスを持つOUTエンドポイントです。
3	bmAttributes	\$02	形式 - 転送:大量(バルク)、同期:なし、使用法:データ エンドポイント
4	wMaxPacketSize	\$0040	本エンドポイントに対する最大パケット容量は\$40(ビット12,11:追加転送谷処理/フレーム)です。
6	bInterval	\$0A	ポーリング間隔は10(ms)です。(大量(バルク)時無効)

8. パラメータ

以下の表はエミュレータ パラメータ書き込み命令(CMND_SET_PARAMETER)とエミュレータ パラメータ読み込み命令(CMND_GET_PARAMETER)を通して保持されるJTAGICEmk II の全パラメータを記述します。

表8-1. パラメータ説明

識別子	説明	形式	読み書き	値
\$01	ハードウェア版番号	[バイト]主MCU版番号 [バイト]従MCU版番号	R	ハードウェア版番号文字列
\$02	ファームウェア版番号	[バイト]主MCU副番号 [バイト]主MCU主番号 [バイト]従MCU副番号 [バイト]従MCU主番号	R	ファームウェア版番号文字列
\$03	エミュレータ動作種別 (デバッグWIREまたはJTAG)	[バイト]	R/W	\$00:デバッグWIRE, \$01:megaAVR用JTAG, \$02:未知/なし(既定), \$03:SPI, \$04:AVR32用JTAG, \$05:XMEGA用JTAG, \$06:XMEGA用PDI
\$04	(未使用)			(未使用)
\$05	ボーレート(既定=4:19200bps)。 RS232構成は8ビット データ,パリティなし, 1停止ビット	[バイト]	R/W	\$01:2400, \$02:4800, \$03:9600, \$04:19200(既定), \$05:38400, \$06:57600, \$07:115200, \$08:14400
\$06	OCD 目的対象電圧(Vtarget)	[バイト]×2、LSB先行	R	目的対象からの読み取り
\$07	OCD JTAGクロック	[バイト]	R/W	JTAGクロックの設定/解除間の遅延。 (4MHz以上の対象)遅延なし=\$00
\$08	OCD 中断原因	[バイト]	R	中断に於いて目的対象からの読み取り
\$09	目的デバイス停止動作時計器動作	[バイト]	R/W	\$00:計時器停止(既定), \$01:計時器走行
\$0A	流れの変化での中断	[バイト]	R/W	\$00
\$0B	中断アドレス1	[バイト]×2、LSB先行	R/W	(未使用)
\$0C	中断アドレス2	[バイト]×2、LSB先行	R/W	(未使用)
\$0D	組み合わせ中断制御	[バイト]	R/W	(未使用)
\$0E	JTAG ID文字列	[バイト]×4	R	デバイス特定

[次頁へ続く](#)

表8-1 (続き). パラメータ説明

識別子	説明	形式	読み書き	値
\$0F	(未使用)			(未使用)
\$10	(未使用)			(未使用)
\$11	(未使用)			(未使用)
\$12	(未使用)			(未使用)
\$13	外部リセット	[バイト]	R/W	\$00:なし, \$01:有り(既定)
\$14	フラッシュメモリページ容量	[バイト]×2、LSB先行	R/W	デバイス指定
\$15	EEPROMページ容量	[バイト]	R/W	デバイス指定
\$16	(未使用)			(未使用)
\$17	PSB1	[バイト]×2、LSB先行	R/W	(未使用)
\$18	PSB2	[バイト]×2、LSB先行	R/W	(未使用)
\$19	(未使用)			(未使用)
\$1A	目的MCU状態	[バイト]	R	\$00:停止(STOPPED), \$01:走行(RUNNING) \$02:プログラミング(PROGRAMMING)
\$1B	デバッグチェーン情報	[バイト]前デバイス数 [バイト]後デバイス数 [バイト]前ビット数 [バイト]後ビット数	R/W	
\$1C	ブートアドレス	[バイト]×4、LSB先行	R/W	
\$1D	目的デバイス識別	[バイト]×2、LSB先行	R	目的デバイスから読み取り
\$1E	(未使用)			(未使用)
\$1F	プログラム開始点	[バイト]×4、LSB先行	W	
\$22	CANメールボックス読み込み	[バイト]	R/W	\$00:CANメールボックス読み込みなし, \$01:CANメールボックス読みこみ
\$23	IDR事象許可	[バイト]	W	\$00:OSCCALアクセス, \$01:IDRアクセス
\$24	走査チェーンでのページ書き込み許可	[バイト]	W	\$00:不許可, \$01:許可
\$2D	(停止(SIGN-OFF)後の)ソフトリセット	[バイト]	W	\$00:ハードリセット, \$03:リセットなし
\$31	NVM制御器のPDIオフセット	[バイト]×4、LSB先行	W	デバイス指定
\$33	フラッシュ応用領域のPDIオフセット	[バイト]×4、LSB先行	W	デバイス指定
\$33	フラッシュブート領域のPDIオフセット	[バイト]×4、LSB先行	W	デバイス指定
\$37	AVR32 JTAG許可手順	[バイト]	W	\$00:JTAG無許可, \$01:JTAG許可手順使用
\$38	プログラミング後走行	[バイト]	W	\$00:目的対象停止(STTOPED)保持, \$01:目的対象走行許容
\$40	パケット解析異常	[バイト]×4、LSB先行	R	
\$41	有効受信パケット	[バイト]×4、LSB先行	R	
\$42	相互通信送信失敗	[バイト]×4、LSB先行	R	
\$43	相互通信受信失敗	[バイト]×4、LSB先行	R	
\$44	CRC異常	[バイト]×4、LSB先行	R	
\$45	電力源	[バイト]	R	\$00:外部, \$01:USB

9. デバイス記述子ファイル

以下はデバイス記述子設定命令(CMND_SET_DEVICE_DESCRIPTOR)によって使われる定義済みパラメータです。

```

unsigned char ucReadIO[8]; // LSB=I/O位置 0, MSB= I/O位置 63
unsigned char ucReadIOShadow[8]; // LSB=I/O位置 0, MSB= I/O位置 63
unsigned char ucWriteIO[8]; // LSB=I/O位置 0, MSB= I/O位置 63
unsigned char ucWriteIOShadow[8]; // LSB=I/O位置 0, MSB= I/O位置 63
unsigned char ucReadExtIO[52]; // LSB=I/O位置96, MSB= I/O位置511
unsigned char ucReadIOExtShadow[52]; // LSB=I/O位置96, MSB= I/O位置511
unsigned char ucWriteExtIO[52]; // LSB=I/O位置96, MSB= I/O位置511
unsigned char ucWriteIOExtShadow[52]; // LSB=I/O位置96, MSB= I/O位置511
unsigned char ucIDRAddress; // IDRアドレス
unsigned char ucSPMCRAddress; // SPMCRレジスタ アドレスとdW基準PC
unsigned long ulBootAddress; // デバイスのブートローダ開始アドレス
unsigned char ucRAMPZAddress; // SRAM I/O空間でのRAMPZレジスタ アドレス
unsigned int uiFlashPageSize; // デバイスのフラッシュ ページ容量(本値での2のべき乗が容量)
unsigned char ucEepromPageSize; // デバイスのバイトでのEEPROMページ容量
unsigned int uiUpperExtIOLoc; // 拡張I/O最終位置(0=拡張I/Oなし)
unsigned long ulFlashSize; // デバイスのフラッシュ メモリ容量
unsigned char ucEepromInst[20]; // EEPROM読み書き用命令手順(命令群)
unsigned char ucFlashInst[3]; // フラッシュ メモリ読み書き用命令手順(命令群)
unsigned char ucSPHaddr; // スタック ポインタ上位
unsigned char ucSPLaddr; // スタック ポインタ下位
unsigned int uiFlashpages; // フラッシュ メモリ内ページ数
unsigned char ucDWDRAddress; // DWDRレジスタ アドレス
unsigned char ucDWBasePC; // プログラム カウンタ(PC)の基準/遮蔽値
unsigned char ucAllowFullPageBitstream; // 全ての新デバイスに於いてFALSE
unsigned int uiStartSmallestBootLoaderSection //
unsigned char EnablePageProgramming; // JTAGデバイス専用(既定:TRUE)
unsigned char ucCacheType; // キャッシュ形式、$00=標準,$01=CAN形式
unsigned int uiSramStartAddr // SRAM開始アドレス
unsigned char ucResetType; // リセット形式、$00
unsigned char ucPCMaskExtended; // 拡張PC付きデバイス専用
unsigned char ucPCMaskHigh; // PC上位遮蔽値
unsigned char ucEindAddress; // EINDレジスタ アドレス
unsigned int EECRAAddress; // EECR I/Oアドレス
  
```

10. メモリ形式

SRAM読み込み時、アドレス範囲に依存して3つの異なる動きが取られ得ます。アドレスが\$0000～\$001Fの範囲なら、JTAGICEmkIIは汎用作業レジスタファイルからデータを取得します。アドレスが\$0020～\$005Fの範囲なら、データはI/Oレジスタファイルから取得されます。AVRが拡張I/Oレジスタを支援し、アドレスが\$00FF以下の範囲なら、データは拡張I/Oレジスタファイルから取得されます。

拡張I/Oレジスタファイルが使われない時にアドレスが\$005Fを越える場合、データは内部/外部SRAMから取得されます。拡張I/Oレジスタファイルが使われる場合、内部/外部SRAMからデータを取得するのに、アドレスは\$00FFを越えなければなりません。

メモリ形式SPM読み込みは単一プログラム語を読む一方でFLASH_PAGEはページ全体を読みます。メモリ書き込みはフラッシュメモリ全体を書くのにFLASH_PAGEを使います。SPMは単一ページを書くためにだけ使われます。

メモリ形式EEPROM_PAGEはページ書き込みとバイト読み込みで使います。メモリ形式FUSE_BITS、LOCK_BITS、OSCCAL_BYTEはバイトの読み/(書き)に使います。

JTAGICEmkIIへ送られるアドレスは全メモリ形式についてバイトアドレスです。開始アドレスの供給だけを必要とします。

エミュレータのプログラムはメモリ読み込み異常をResp_FAILEDとして報告します。未知のメモリ形式読み込みは無視されます。未知のメモリ形式書き込みも無視されます。

メモリ形式CANはこのメモリ形式がアクセスされる毎に300バイトを読むことによって処理されます。メモリ形式CANのアクセスはCANメールボックス読み込みパラメータによって制御されます。CANメールボックス読み込みパラメータが真なら、300バイト全てを読み、偽なら読みません。

CANメールボックスへの書き込みもメモリ形式CANを使うことによって行われます。各バイトの編集/書き込みに対して、そのバイトのアドレスと共にこのメモリ形式でメモリ書き込み命令(CMND_WRITE_MEMORY)を呼ぶべきです。

表10-1. メモリ形式

メモリ領域	MEM_TYPE値
IO_SHADOW	\$30
SRAM	\$20
EEPROM	\$22
EVENT	\$60
SPM	\$A0
FLASH_PAGE	\$B0
EEPROM_PAGE	\$B1
FUSE_BITS	\$B2
LOCK_BITS	\$B3
SIGN_JTAG	\$B4
OSCCAL_BYTE	\$B5
CAN	\$B6
XMEGA_APPLICATION_FLASH	\$C0
XMEGA_BOOT_FLASH	\$C1
XMEGA_USER_SIGNATURE	\$C5
XMEGA_CALIBRATION_SIGNATURE	\$C6

11. 中断点(ブレイクポイント)

OCDシステムは中断点(ブレイクポイント)を設定するのに中断点比較器を使います。中断点制御部は2つの単一プログラムメモリ中断点と2つの組み合わせ中断点を含みます。どちらも一緒にして4つの中断点を構成することができます(1つの中断点は常に命令1段実行(シングルステップ)用に使われます)。

- 4つの単一プログラムメモリ中断点
- 3つの単一プログラムメモリ中断点 + 1つの単一データメモリ中断点
- 2つの単一プログラムメモリ中断点 + 2つの単一データメモリ中断点
- 2つの単一プログラムメモリ中断点 + 1つの遮蔽付きプログラムメモリ中断点(範囲中断点)
- 2つの単一プログラムメモリ中断点 + 1つの遮蔽付きデータメモリ中断点(範囲中断点)

データメモリ中断点は、データメモリ読み込み、データメモリ書き込み、データメモリ読み書きの3つの形態の1つに設定することができます。データメモリ中断は中断条件を引き起こした命令の終了後にAVR CPUを停止(STOPPED)状態に設定します。

データ内容での中断は支援されません。

OCDシステムは中断点制御部内に各種レジスタを含みます。

PSB0とPSB1(単一アドレスでの中断)はCPUからのプログラムカウンタに対する16ビット比較レジスタです。

PDMSB(プログラム/データの遮蔽/単一中断)はプログラムメモリまたはデータメモリのどちらかのアドレスでの単一中断点を設定するのに使われるレジスタです。また、PDMSBはPDSB比較器に対するアドレスの遮蔽としても働き、それによって“範囲中断”を実行します。

PDSB(プログラム/データ単一中断)はプログラムメモリまたはデータメモリのどちらかのアドレスでの単一中断点を設定するのに使われます。また、PDMSBはPDSB比較器に対するアドレスを遮蔽することができ、これによって“範囲中断”を実行します。

BCR(中断制御レジスタ)は上で言及した4つの各種中断レジスタの設定を制御するのに使われる数ある中の1つです。

JTAGICEmkIIに中断点を設定するため、AVR StudioはPSB0とPSB1のレジスタを設定するのにエミュレータパラメータ書き込み命令(CMND_SET_PARAMETER)を使います。この場合、パラメータ書き込み命令のアドレスは中断点が置かれる場所のアドレスで、値はPSB0またはPSB1のどちらの中断点がこの位置に設定されるべきかを示します(0=PSB0,1=PSB1)。必要なBCRの変更は自動的に行われます。

エミュレータパラメータ書き込み命令(CMND_SET_PARAMETER)と中断点アドレス1,2のパラメータはPDMSBとPDSBのレジスタを設定します。この場合、BCRは自動的に変更されません。PDMSBとPDSBの中断点を活性(有効)にするため、BCRを正しい値に設定するのにエミュレータパラメータ書き込み命令(CMND_SET_PARAMETER)と組み合わせ中断制御のパラメータが使われるべきです。JTAGICEmkII中断時、全ての中断点が解除され、従って中断点は各走行に先立って設定されなければなりません。中断点解除命令(CMND_CLR_BREAK)の必要はありません。

12. CRC16計算

以下のソースコードがCRC16計算を実行してJTAGICEmk II 通信規約を検証します。受信した命令の完全性を検証するのにVerifyChecksum関数を使います。メッセージの完全性が維持されていれば真(TRUE)を返し、さもなければ偽(FALSE)を返します。

メッセージを送る前にメッセージに2バイトのCRC16を追加するのにAppendChecksum関数を使います。

crc16.h

```
#ifndef CRC16_H
#define CRC16_H
class Crc16
{
public: // メッセージのチェックサム計算
    static unsigned short Checksum( const unsigned char* message, unsigned long length, unsigned short crc = 0xffff);
    // 最終2バイトがメッセージの有効なCRC(LSB先行)であることを検証
    static bool VerifyChecksum( const unsigned char* message, unsigned long length);
    // メッセージに2バイトのCRC(LSB先行)を追加。lengthはCRCを除くメッセージの容量です。
    // CRCバイト用の空間が先に割り当てられていなければなりません!
    static void AppendChecksum( unsigned char* message, unsigned long length);
}
#endif
```

crc16.cpp

```

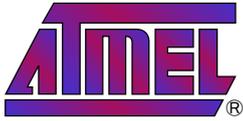
#include "Crc16.h"
// データ転送層crc16.hから取られる符号
// CRC16定義
const unsigned short crc_table[256] =
{
    0x0000, 0x1189, 0x2312, 0x329B, 0x4624, 0x57AD, 0x6536, 0x74BF,
    0x8C48, 0x9DC1, 0xAF5A, 0xBED3, 0xCA6C, 0xDBE5, 0xE97E, 0xF8F7,
    0x1081, 0x0108, 0x3393, 0x221A, 0x56A5, 0x472C, 0x75B7, 0x643E,
    0x9CC9, 0x8D40, 0xBFDB, 0xAE52, 0xDAED, 0xCB64, 0xF9FF, 0xE876,
    0x2102, 0x308B, 0x0210, 0x1399, 0x6726, 0x76AF, 0x4434, 0x55BD,
    0xAD4A, 0xBCC3, 0x8E58, 0x9FD1, 0xEB6E, 0xFAE7, 0xC87C, 0xD9F5,
    0x3183, 0x200A, 0x1291, 0x0318, 0x77A7, 0x662E, 0x54B5, 0x453C,
    0xBDCB, 0xAC42, 0x9ED9, 0x8F50, 0xFBEB, 0xEA66, 0xD8FD, 0xC974,
    0x4204, 0x538D, 0x6116, 0x709F, 0x0420, 0x15A9, 0x2732, 0x36BB,
    0xCE4C, 0xDFC5, 0xED5E, 0xFCD7, 0x8868, 0x99E1, 0xAB7A, 0xBAF3,
    0x5285, 0x430C, 0x7197, 0x601E, 0x14A1, 0x0528, 0x37B3, 0x263A,
    0xDECD, 0xCF44, 0xFDDF, 0xEC56, 0x98E9, 0x8960, 0xBBFB, 0xAA72,
    0x6306, 0x728F, 0x4014, 0x519D, 0x2522, 0x34AB, 0x0630, 0x17B9,
    0xEF4E, 0xFEC7, 0xCC5C, 0xDD55, 0xA96A, 0xB8E3, 0x8A78, 0x9BF1,
    0x7387, 0x620E, 0x5095, 0x411C, 0x35A3, 0x242A, 0x16B1, 0x0738,
    0xFFCF, 0xEE46, 0xDCDD, 0xCD54, 0xB9EB, 0xA862, 0x9AF9, 0x8B70,
    0x8408, 0x9581, 0xA71A, 0xB693, 0xC22C, 0xD3A5, 0xE13E, 0xF0B7,
    0x0840, 0x19C9, 0x2B52, 0x3ADB, 0x4E64, 0x5FED, 0x6D76, 0x7CFF,
    0x9489, 0x8500, 0xB79B, 0xA612, 0xD2AD, 0xC324, 0xF1BF, 0xE036,
    0x18C1, 0x0948, 0x3BD3, 0x2A5A, 0x5EE5, 0x4F6C, 0x7DF7, 0x6C7E,
    0xA50A, 0xB483, 0x8618, 0x9791, 0xE32E, 0xF2A7, 0xC03C, 0xD1B5,
    0x2942, 0x38CB, 0x0A50, 0x1BD9, 0x6F66, 0x7EEF, 0x4C74, 0x5DFD,
    0xB58B, 0xA402, 0x9699, 0x8710, 0xF3AF, 0xE226, 0xD0BD, 0xC134,
    0x39C3, 0x284A, 0x1AD1, 0x0B58, 0x7FE7, 0x6E6E, 0x5CF5, 0x4D7C,
    0xC60C, 0xD785, 0xE51E, 0xF497, 0x8028, 0x91A1, 0xA33A, 0xB2B3,
    0x4A44, 0x5BCD, 0x6956, 0x78DF, 0x0C60, 0x1DE9, 0x2F72, 0x3EFB,
    0xD68D, 0xC704, 0xF59F, 0xE416, 0x90A9, 0x8120, 0xB3BB, 0xA232,
    0x5AC5, 0x4B4C, 0x79D7, 0x685E, 0x1CE1, 0x0D68, 0x3FF3, 0x2E7A,
    0xE70E, 0xF687, 0xC41C, 0xD595, 0xA12A, 0xB0A3, 0x8238, 0x93B1,
    0x6B46, 0x7ACF, 0x4854, 0x59DD, 0x2D62, 0x3CEB, 0x0E70, 0x1FF9,
    0xF78F, 0xE606, 0xD49D, 0xC514, 0xB1AB, 0xA022, 0x92B9, 0x8330,
    0x7BC7, 0x6A4E, 0x58D5, 0x495C, 0x3DE3, 0x2C6A, 0x1EF1, 0x0F78 };
// CRC計算マクロ
#define CRC_INIT 0xFFFF
#define CRC(crcval, newchar) crcval = (crcval >> 8) ^ crc_table[(crcval ^ newchar) & 0x00ff]
unsigned short Crc16::Checksum( const unsigned char* message, unsigned long length, unsigned short crc)
{
    for(unsigned long i = 0; i < length; i++)
        { CRC(crc, message[i]);
        }
    return crc;
}
bool Crc16::VerifyChecksum(const unsigned char* message, unsigned long length)
{
    // メッセージ内の最後の2バイトが先行するバイトのCRCの場合に真(TRUE)を返します。
    unsigned short expected = Checksum(message, length - 2);
    return (expected & 0xff) == message[length - 2] && ((expected >> 8) & 0xff) == message[length - 1];
}
void Crc16::AppendChecksum(unsigned char* message, unsigned long length)
{
    unsigned long crc = Checksum(message, length);
    message[length] = (unsigned char)(crc & 0xff);
    message[length+1] = (unsigned char)((crc >> 8) & 0xff);
}

```

13. 目次

要点	1
1. 序説	1
2. 動作の理屈	1
2.1. USBとRS-232接続	1
2.1.1. RS-232接続	1
2.1.2. USB接続	1
2.1.3. USB構成設定	1
3. メッセージ形式	2
4. メッセージ解析	2
4.1. 序説	2
4.2. 共通状態機構実装	3
5. 命令と応答	4
5.1. 主装置命令	4
5.1.1. 停止命令(CMND_SIGN_OFF) [\$00]	4
5.1.2. エミュレータ存在検査命令(CMND_GET_SIGN_ON) [\$01]	4
5.1.3. エミュレータパラメータ書き込み命令(CMND_SET_PARAMETER) [\$02]	5
5.1.4. エミュレータパラメータ読み込み命令(CMND_GET_PARAMETER) [\$03]	6
5.1.5. メモリ書き込み命令(CMND_WRITE_MEMORY) [\$04]	6
5.1.6. メモリ読み出し命令(CMND_READ_MEMORY) [\$05]	7
5.1.7. プログラムカウンタ書き込み命令(CMND_WRITE_PC) [\$06]	8
5.1.8. プログラムカウンタ読み出し命令(CMND_READ_PC) [\$07]	8
5.1.9. プログラム実行開始命令(CMND_GO) [\$08]	9
5.1.10. 1段実行命令(CMND_SINGLE_STEP) [\$09]	9
5.1.11. プログラム実行停止命令(CMND_FORCED_STOP) [\$0A]	10
5.1.12. 使用者プログラムリセット命令(CMND_RESET) [\$0B]	11
5.1.13. デバイス記述子設定命令(CMND_SET_DEVICE_DESCRIPTOR) [\$0C]	11
5.1.14. SPMページ消去命令(CMND_ERASEPAGE_SPM) [\$0D]	12
5.1.15. 同期獲得命令(CMND_GET_SYNC) [\$0F]	12
5.1.16. 自己検査命令(CMND_SELFTEST) [\$10]	13
5.1.17. 中断点(ブレークポイント)設定命令(CMND_SET_BREAK) [\$11]	14
5.1.18. 中断点(ブレークポイント)取得命令(CMND_GET_BREAK) [\$12]	14
5.1.19. チップ消去命令(CMND_CHIP_ERASE) [\$13]	15
5.1.20. プログラミング動作移行命令(CMND_ENTER_PROGMODE) [\$14]	16
5.1.21. プログラミング動作抜け出し命令(CMND_LEAVE_PROGMODE) [\$15]	16
5.1.22. 中断点(ブレークポイント)解除命令(CMND_CLR_BREAK) [\$1A]	17
5.1.23. 指定アドレスまで実行命令(CMND_RUN_TO_ADDR) [\$1C]	17
5.1.24. 万能SPI命令(CMND_SPI_CMD) [\$1D]	18
5.1.25. 事象メモリ解除命令(CMND_CLEAR_EVENTS) [\$22]	18
5.1.26. 目的対象回復命令(CMND_RESTORE_TARGET) [\$23]	19
5.1.27. JTAG命令書き込み命令(CMND_JTAG_INSTR) [\$24]	19
5.1.28. JTAGデータ読み書き命令(CMND_JTAG_DATA) [\$25]	20
5.1.29. SABデータ書き込み命令(CMND_JTAG_SAB_WRITE) [\$28]	20
5.1.30. SABデータ読み込み命令(CMND_JTAG_SAB_READ) [\$29]	21
5.1.31. JTAGデータ塊読み込み命令(CMND_JTAG_BLOCK_READ) [\$2C]	21
5.1.32. JTAGデータ塊書き込み命令(CMND_JTAG_BLOCK_WRITE) [\$2D]	22
5.1.33. カプセル化SPI命令(CMND_ISP_PACKET) [\$2F]	23
5.1.34. XMEGA消去命令(CMND_XMEGA_ERASE) [\$34]	24
5.2. 従装置応答	25
5.2.1. 成功(RSP_OK) [\$80]	25
5.2.2. パラメータ返答(RSP_PARAMETER) [\$81]	25
5.2.3. メモリ読み出し返答(RSP_MEMORY) [\$82]	25
5.2.4. 中断点(ブレークポイント)読み出し返答(RSP_BREAK) [\$83]	26
5.2.5. プログラムカウンタ読み出し返答(RSP_PC) [\$84]	26
5.2.6. 自己検査返答(RSP_SELFTEST) [\$85]	27
5.2.7. 開始返答(RSP_SIGN_ON) [\$86]	28
5.2.8. 走査チェーン読み込み返答(RSP_SCAN_CHAIN_READ) [\$87]	28
5.2.9. SPI読み出しデータ返答(RSP_SPI_DATA) [\$88]	29

5.2.10.	失敗(RSP_FAILED) [\$A0]	29
5.2.11.	不正なパラメータ(RSP_ILLEGAL_PARAMETER) [\$A1]	30
5.2.12.	不正メモリ形式アクセス(RSP_ILLEGAL_MEMORY_TYPE) [\$A2]	30
5.2.13.	不正メモリ範囲アクセス(RSP_ILLEGAL_MEMORY_RANGE) [\$A3]	30
5.2.14.	エミュレータ動作種別による操作実行不能(RSP_ILLEGAL_EMULATOR_MODE) [\$A4]	31
5.2.15.	MCU状態による操作実行不能(RSP_ILLEGAL_MCU_STATE) [\$A5]	31
5.2.16.	不正値(RSP_ILLEGAL_VALUE) [\$A6]	31
5.2.17.	不正中断点(RSP_ILLEGAL_BREAKPOINT) [\$A8]	32
5.2.18.	不正JTAG ID(RSP_ILLEGAL_JTAG_ID) [\$A9]	32
5.2.19.	不正命令(RSP_ILLEGAL_COMMAND) [\$AA]	32
5.2.20.	目的対象不正電源状態(RSP_NO_TARGET_POWER) [\$AB]	33
5.2.21.	デバッグWIRE同期失敗(RSP_DEBUGWIRE_SYNC_FAILED) [\$AC]	33
5.2.22.	JTAGICEmk II 電力不足(RSP_ILLEGAL_POWER_STATE) [\$AD]	34
6.	事象	34
6.1.	序説	34
6.2.	事象	34
6.2.1.	中断事象(EVT_BREAK) [\$E0]	34
6.2.2.	目的対象電源ON事象(EVT_TARGET_POWER_ON) [\$E4]	35
6.2.3.	目的対象電源OFF事象(EVT_TARGET_POWER_OFF) [\$E5]	35
6.2.4.	目的対象外部リセット事象(EVT_EXT_RESET) [\$E7]	35
6.2.5.	目的対象休止移行事象(EVT_TARGET_SLEEP) [\$E8]	36
6.2.6.	目的対象起動復帰事象(EVT_TARGET_WAKEUP) [\$E9]	36
6.2.7.	不正電源状態事象(EVT_ICE_POWER_ERROR_STATE) [\$EA]	36
6.2.8.	電源正常事象(EVT_ICE_POWER_OK) [\$EB]	37
6.2.9.	IDR書き込み事象(EVT_IDR_DIRTY) [\$EC]	37
6.2.10.	デバッグWIRE異常事象(EVT_ERROR_PHY_x) [\$xx]	38
7.	USB記述子	39
7.1.	装置記述子	39
7.2.	構成設定記述子	39
7.3.	インターフェース記述子	39
7.4.	INエンドポイント記述子	40
7.5.	OUTエンドポイント記述子	40
8.	パラメータ	40
9.	デバイス記述子ファイル	42
10.	メモリ形式	43
11.	中断点(ブレークポイント)	43
12.	CRC16計算	44
13.	目次	46



本社

Atmel Corporation

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

国外営業拠点

Atmel Asia

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan

104-0033 東京都中央区
新川1-24-8
東熱新川ビル 9F
アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ウェブサイト

www.atmel.com

技術支援

avr@atmel.com

販売窓口

www.atmel.com/contacts

文献請求

www.atmel.com/literature

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のウェブサイトに位置する販売の条件と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。たとえば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、または利益の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。

© Atmel Corporation 2009. 不許複製 Atmel®、ロゴとそれらの組み合わせ、AVR®とその他はAtmel Corporationの登録商標または商標またはその附属物です。他の用語と製品名は一般的に他の商標です。

© HERO 2021.

本応用記述はAtmelのAVR067応用記述(doc2587.pdf Rev.2587D-11/09)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。