

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。

AVR[®] STK500

使用者の手引き



目 次

第1章 概要	1-1
1.1 スタートキットの特徴	1-1
1.2 支援デバイス	1-2
第2章 開始に際して	2-1
2.1 キット内容	2-1
2.2 必要環境	2-1
2.3 即時開始	2-1
2.3.1 機器接続	2-2
2.3.2 目的AVRデバイスのプログラミング設定	2-3
第3章 ハードウェア	3-1
3.1 使用者LED	3-1
3.2 使用者スイッチ	3-1
3.3 使用者LEDとスイッチの接続	3-2
3.4 I/Oポート コネクタ	3-3
3.5 使用者RS-232Cインターフェース	3-4
3.6 データ用フラッシュ メモリ	3-4
3.7 目的対象ソケット	3-6
3.7.1 ISPプログラミング	3-6
3.7.2 高電圧プログラミング	3-9
3.7.2.1 高電圧並列プログラミング	3-10
3.7.2.2 高電圧直列プログラミング	3-11
3.8 ジャンパ設定	3-12
3.8.1 目的VCC設定 VTARGET	3-13
3.8.2 アナログ基準電圧 AREF	3-14
3.8.3 リセット選択 RESET	3-15
3.8.4 クロック選択 XTAL1, OSCSEL	3-16
3.8.5 バイト選択2ジャンパ BSEL2	3-18
3.8.6 PJUMPジャンパ PJUMP	3-18
3.8.7 ATtiny26/261/461/861の高電圧並列プログラミング	3-19
3.8.8 ATmega406の高電圧並列プログラミング	3-19
3.9 拡張コネクタ	3-20
3.9.1 信号概要	3-21
3.10 PROG CTRL, PROG DATAヘッダ	3-21
3.11 その他スイッチ, LED	3-22
3.11.1 RESET押し釦	3-22
3.11.2 PROGRAM押し釦	3-22
3.11.3 主電源LED	3-22
3.11.4 目的AVR電源LED	3-22
3.11.5 状態LED	3-22
第4章 AVR Studioのインストール	4-1
第5章 AVR Studioの使用法	5-1
5.1 AVR Studio	5-1
5.2 AVR Studioの起動	5-1
5.2.1 STK500の始動	5-1
5.3 STK500使用者インターフェース	5-2
5.3.1 Mainタブ	5-2
5.3.1.1 デバイス選択と識別バイト Device and Signature Bytes	5-2
5.3.1.2 プログラミング種別と目的対象調整設定 Programming Mode and Target Settings	5-3
5.3.2 Programタブ	5-4
5.3.2.1 デバイス操作 Device	5-4
5.3.2.2 フラッシュ プログラミング Flash	5-4
5.3.2.3 EEPROMプログラミング EEPROM	5-4



5.3.2.4	ELF提出ファイル形式 ELF Production File Format	5-5
5.3.2.5	XMEGA消去種別	5-5
5.3.3	Fusesタブ	5-6
5.3.4	LockBitsタブ	5-7
5.3.5	Advancedタブ	5-8
5.3.5.1	発振校正値 Oscillator Calibration Byte (非XMEGA用)	5-8
5.3.5.2	発振校正値の読み書き (非XMEGA用)	5-8
5.3.5.3	XMEGA用Advancedタブ	5-9
5.3.5.4	製品識票 Production Signature	5-9
5.3.5.5	使用者識票 User Signature	5-9
5.3.6	HW Settingsタブ	5-11
5.3.6.1	電圧 Voltages	5-11
5.3.6.2	クロック発生器 Clock Generator	5-11
5.3.7	HW Infoタブ	5-12
5.3.8	Autoタブ	5-13
5.3.8.1	自動プログラミング設定	5-13
5.3.8.2	自動プログラミング記録ファイル	5-13
5.4	コマンド行ソフトウェア	5-14
5.5	引数	5-14
5.6	支援デバイス名	5-15
第6章	外部実装書き込み(ISP)	6-1
第7章	障害対策の指針	7-1
7.1	STK500ファームウェア手動更新	7-1
第8章	技術支援	8-1
第9章	試供プログラム	9-1
9.1	LEDとスイッチ制御	9-1
付録		
	付録A (STK500構成図)	A-1
	付録B (STK500実装図)	B-1
	付録C (STK500全回路図)	C-1

STK500 AVR[®] フラッシュ マイクロ コントローラ スタート キットの御購入有難う御座います。STK500は Atmel社のAVR フラッシュ マイクロコントローラ用の完全なスタート キットである開発装置です。それは新規設計でのAVR上のコード開発、試作、試験の迅速な開始を設計者に与えるために設計されました。

1.1 スタート キットの特徴

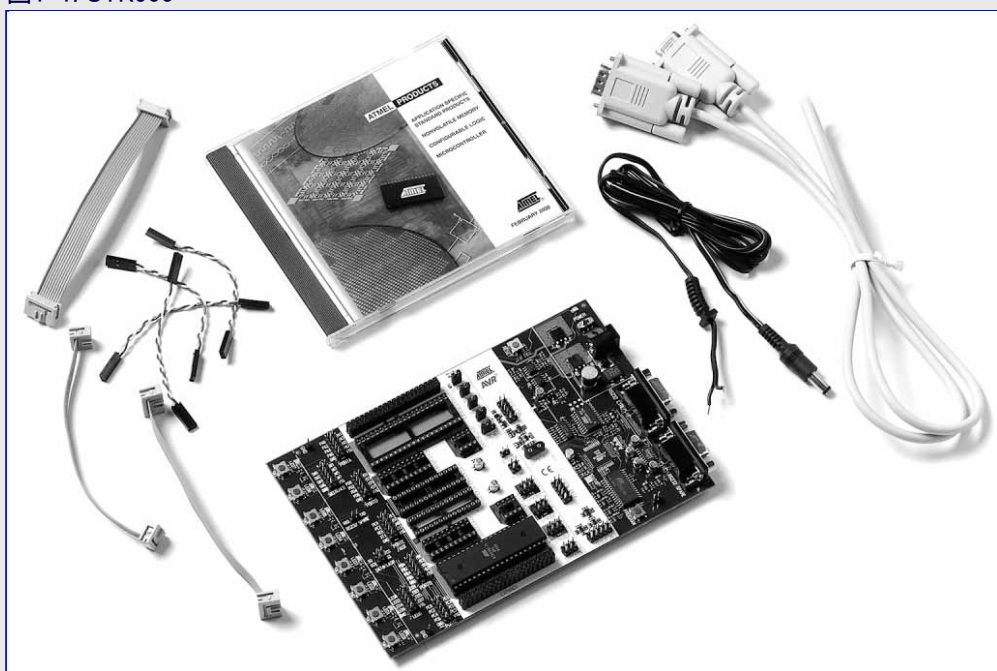
- AVR Studio[®] 適合
- パソコンからのプログラミングと制御用のRS-232Cインターフェース
- DC10～15V入力からの安定化電源供給
- 8,20,28,40ピンAVRデバイス用ソケット
- 高電圧並列及び直列プログラミング
- 実装書き込み(ISP)直列プログラミング
- 外部目的システム内のAVRデバイスに対する実装書き込み(ISP)直列プログラミング
- AVRデバイスの再プログラミング (書き換え)
- 8個の汎用押し釦スイッチ入力
- 8個の汎用LED出力表示
- ピン ヘッダ接続によるAVR全I/Oポートへの容易なアクセス
- 汎用RS-232Cポートの付加
- プラグイン規格化部品や試作領域用の拡張コネクタ
- 不揮発性データ記憶用に2Mビットのデータ フラッシュ(DataFlash[®]) 内蔵

注: AT45D021データ フラッシュは製造終了で、従って後期のSTK500基板では実装されていません。

注: いくつかのデバイスはSTK500と共に動作するために拡張アダプタを必要とします。これらのデバイスとアダプタについては**支援デバイス**を調べてください。

STK500,各種拡張アダプタ,AVR Studioなどの最新情報についてはAtmelのウェブサイト www.atmel.com をご覧ください。

図1-1. STK500



1.2 支援デバイス

現在のシステムソフトウェアは次のデバイスの全速度品種を支援します。

AT90S1200	AT90S2313	AT90S2323	AT90S2333
AT90S2343	AT90S4414	AT90S4433	AT90S4434
AT90S8515	AT90S8535		
ATmega48(P)	ATmega8	ATmega8515	ATmega8535
ATmega88(P,PA)	ATmega16	ATmega16HVA (注13)	ATmega161
ATmega162	ATmega163	ATmega164P	ATmega165(P) (注2)
ATmega168(P)	ATmega169(P) (注2)	ATmega32	ATmega32C1 (注8)
ATmega32M1 (注8)	ATmega32U4 (注13)	ATmega32HVB (注13)	ATmega323
ATmega324P	ATmega325(P) (注2)	ATmega3250(P) (注4)	ATmega328P
ATmega329(P) (注2)	ATmega3290(P) (注4)	ATmega406 (注12)	ATmega64 (注1)
ATmega640 (注3)	ATmega644(P)	ATmega645 (注2)	ATmega6450 (注4)
ATmega649 (注2)	ATmega6490 (注4)	ATmega103 (注1)	ATmega128 (注1)
ATmega1280 (注3)	ATmega1281 (注1)	ATmega1284P	ATmega2560 (注3)
ATmega2561 (注1)	AT90CAN32 (注6)	AT90CAN64 (注6)	AT90CAN128 (注6)
AT90PWM2 (注7)	AT90PWM2B (注7)	AT90PWM216 (注7)	AT90PWM3 (注7)
AT90PWM3B (注7)	AT90PWM316 (注7)	AT90USB82 (注10)	AT90USB162 (注10)
AT90USB646 (注9)	AT90USB647 (注9)	AT90USB1286 (注9)	AT90USB1287 (注9)
ATtiny11	ATtiny12	ATtiny13	ATtiny15
ATtiny22	ATtiny2313	ATtiny24 (注5)	ATtiny25
ATtiny26 (注11)	ATtiny261 (注11)	ATtiny28	ATtiny43U (注13)
ATtiny44 (注5)	ATtiny45	ATtiny461 (注11)	ATtiny48
ATtiny84	ATtiny85	ATtiny861 (注11)	ATtiny88
ATtiny167			
AT86RF401	AT89S51 (注12)	AT89S52 (注12)	

注1: STK501アダプタとで支援。

注2: STK502アダプタとで支援。

注3: STK503アダプタとで支援。

注4: STK504アダプタとで支援。

注5: STK505アダプタとで支援。

注6: STK501アダプタとSTK501CAN(ATADAPCAN1)アダプタとで支援。

注7: STK520アダプタとで支援。

注8: STK524アダプタとで支援。

注9: STK525アダプタとで支援。

注10: STK526アダプタとで支援。

注11: これらのデバイスはより容易な結線用のSTK505アダプタでも支援されます。

注12: ATmega406のピン配置は母基板やどのアダプタ基板でも支援されません。けれども高電圧並列プログラミングがSTK500ファームウェアに含まれます。

注13: 外部目的対象(ISPプログラミング)のみ支援。

新しいAVRデバイスの支援はAVR Studioの新版で追加されます。AVR Studioの最終版はwww.atmel.com から常に入手できます。



2.1 キット内容

- STK500 スタート キット評価基板
STK500用ケーブル
 - I/Oポートと並列プログラミング用10芯ケーブル ×2
 - 実装書き込み(ISP)用6芯ケーブル ×1
 - UARTとデータ フラッシュ接続用2芯ケーブル ×4
 - 9ピンRS-232C用ケーブル
 - DC電源ケーブル
 - データシートとソフトウェアが含まれるAtmelのCD-ROM
 - AT90S8515-8PC マイクロ コントローラ試供品
- (訳注) 実際のキットには、試供品に ATtiny11やATmega163/16なども含まれている場合があります。

2.2 必要環境

- ハードウェアとソフトウェアの最低必要条件は以下です。
- Pentium®クラスのCPU
 - 64Mバイト(Windows9x),128Mバイト(Windows NT/2000/XP)のRAM (この倍を推奨)
 - 100Mバイト ハードディスク空き容量 (AVR Studio用)
 - 115200bps RS-232Cポート (COMポート)
 - 10～15V、500mA以上のDC電源

2.3 即時開始

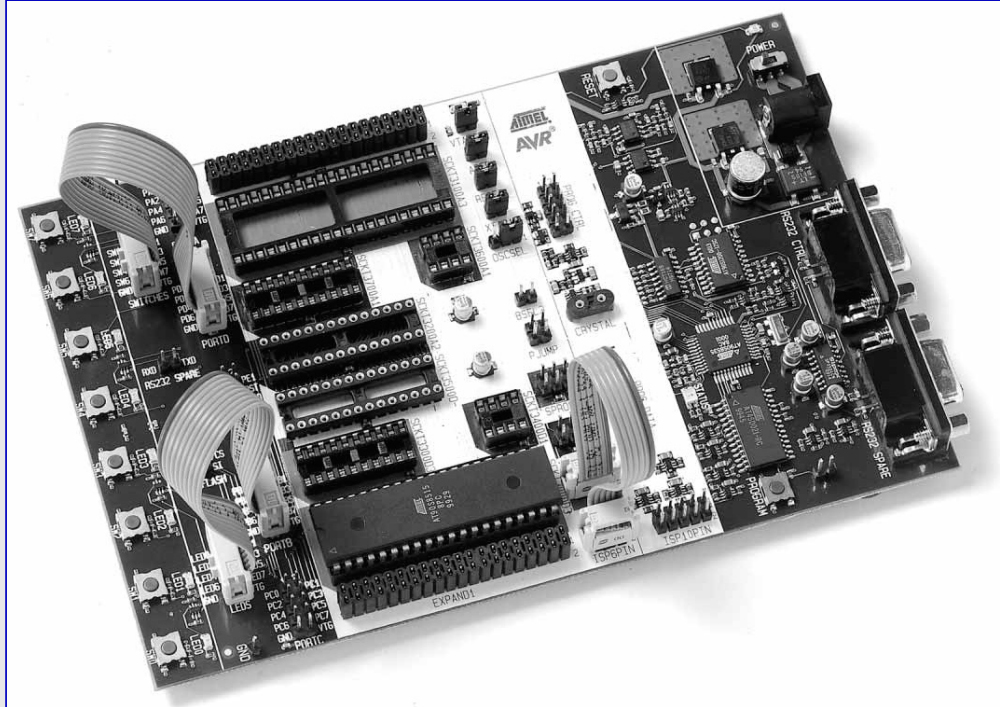
STK500スタート キットの**SCKT3000D3**と記載されたソケットにAT90S8515-8PCマイクロ コントローラが搭載されています。**既定のジャンプ設定**はSTK500基板上の電源とクロックでの実行をマイクロ コントローラにさせます。

このマイクロ コントローラにはLEDを点滅する試験プログラムが書き込まれています。AT90S8515内の試験プログラムは第9章で説明される**応用(プログラム)コード例**と同等のものです。AT90S8515内の試験プログラムを実行させるにはLEDとスイッチを接続してSTK500の電源を入れます。

PORTB, **LEDS**と記載されたヘッダの接続と**PORTD**, **SWITCHES**と記載されたヘッダの接続には供給された10芯ケーブルを使います。この接続は図2-1.で示されます。

10～15Vの外部DC電源が必要です。入力回路はブリッジ全波整流で、STK500は中心が+または-、両方のコネクタを自動的に扱います。中心+のコネクタが使われる場合、電源スイッチがGND端子を切断する(構造)のため、STK500のON/OFF切り替えは不可能かもしれません。この場合、接続されているRS-232Cケーブルのシールド経路、または代わりのGND接続経路でGNDが供給される可能性があります。電源とSTK500間を電源ケーブルで接続します。電源コネクタにDC10～15Vを供給します。**電源スイッチ**はSTK500主電源のONとOFFを切り替えます。電源ONで赤LEDが点灯し、この**状態LED**は赤→黄→緑になります。**緑LED**は目的(デバイスの)VCCが存在することを示します。AT90S8515で走行中のプログラムは押されたスイッチにLEDの点滅で応じます。

図2-1. STK500既定設定



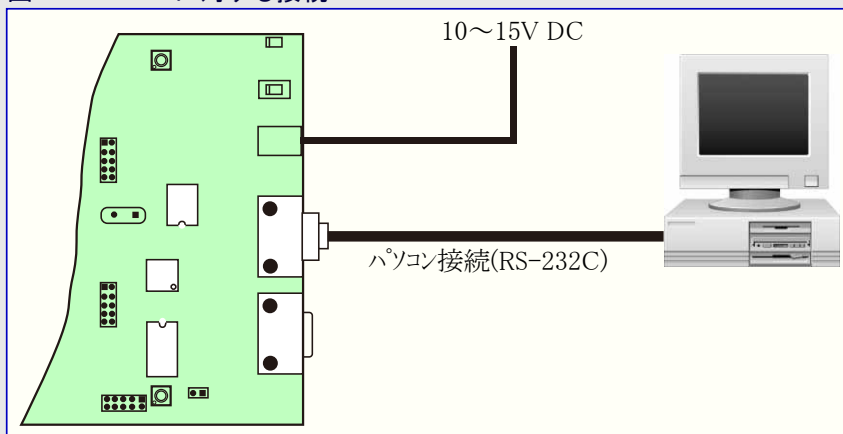
このスタートキットは種々の電源やクロックの設定ができます。ジャンパ設定の完全な内容は3-11頁の「3.8 ジャンパ設定」節内とスタートキット基板の裏面で説明されます。

2.3.1 機器接続

AT90S8515に書き込むには図2-1.で示されるようにISP6PINヘッダとSPROG3目標ISPヘッダ間を供給された6芯ケーブルで接続します。3-6頁の3.7.1節は実装書き込み(ISP)用ケーブル接続を記述します。

図2-2.で示されるように評価基板のRS232 CTRLと記載されたコネクタとパソコンのCOMポートに直列ケーブルを接続します。パソコンにAVR Studioソフトウェアをインストールします。AVR Studioのインストールと使用法の指示は5-1頁の第5章で提供されます。AVR Studioが開始される時にこのプログラムはSTK500がどのCOMポートに接続されているかを自動的に検出します。

図2-2. STK500に対する接続

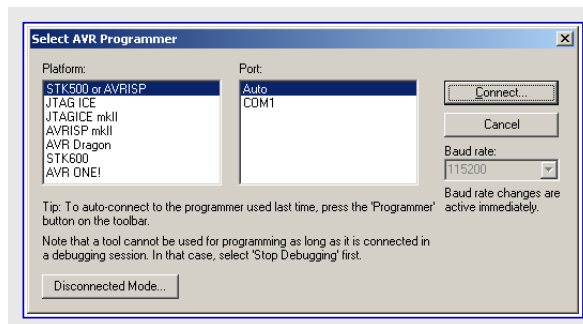


2.3.2 目的AVRデバイスのプログラミング設定

STK500は4.0版以降のAVR Studioで制御されます。AVR StudioはAVRでの応用の開発とデバッグ用の統合開発環境(IDE)です。AVR 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、インサーキットエミュレータのインターフェース、STK500のプログラミングインターフェースを提供します。

目的AVRデバイス内にHEXファイルを書き込むにはAVR StudioのToolsメニューからConnect...を選択してください。これは以下で示されるダイアログを表示します。

STK500やAVRISPに接続するにはPlatform:一覧でSTK500 or AVRISPを選択してください。そしてConnect...鈕を押下してください。するとSTK500とAVRISP用のドライバが開始され、STK500ダイアログが現れるでしょう。システムがJTAGICEやJTAGICE mk IIのような他のプログラミング装置も支援することに注意してください。



その後、Programタブの引き落としメニューから目的AVRデバイスを選択し、(ダウンロードする)目的のIntel-HEX形式ファイルを割り当てます。試験プログラムを書き込む場合は試験プログラムをアセンブルし、HEXファイルを作成します。

Erase鈕、その後にProgram鈕をクリックします。デバイス書き込み中の状態LEDは黄色に替わり、書き込み成功時、このLEDは緑に替わります。書き込みが失敗した場合、本LEDは書き込み後、赤に替わります。この場合は7-1頁の「第7章 障害対策の指針」を参照してください。

AVR StudioでのSTK500インターフェース使用の完全な記載は5-1頁の第5章で提供されます。

図2-3. AVR StudioによるSTK500でのプログラミングメニュー

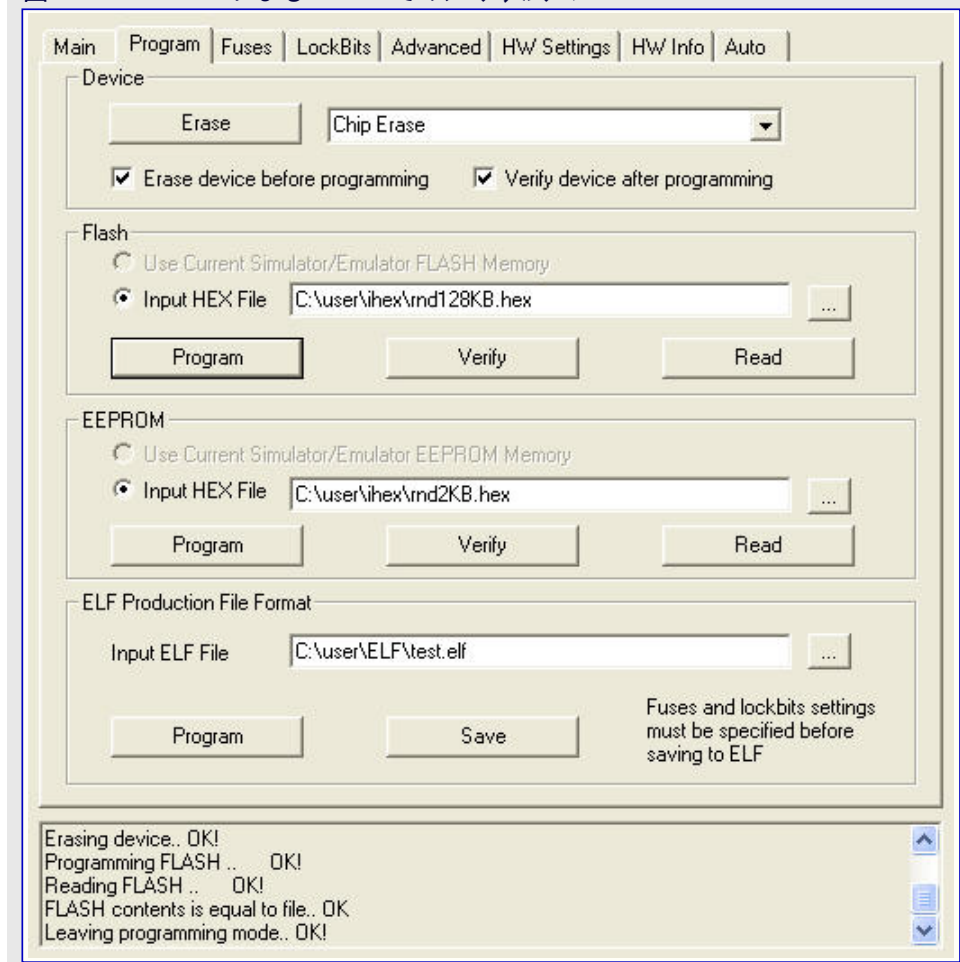
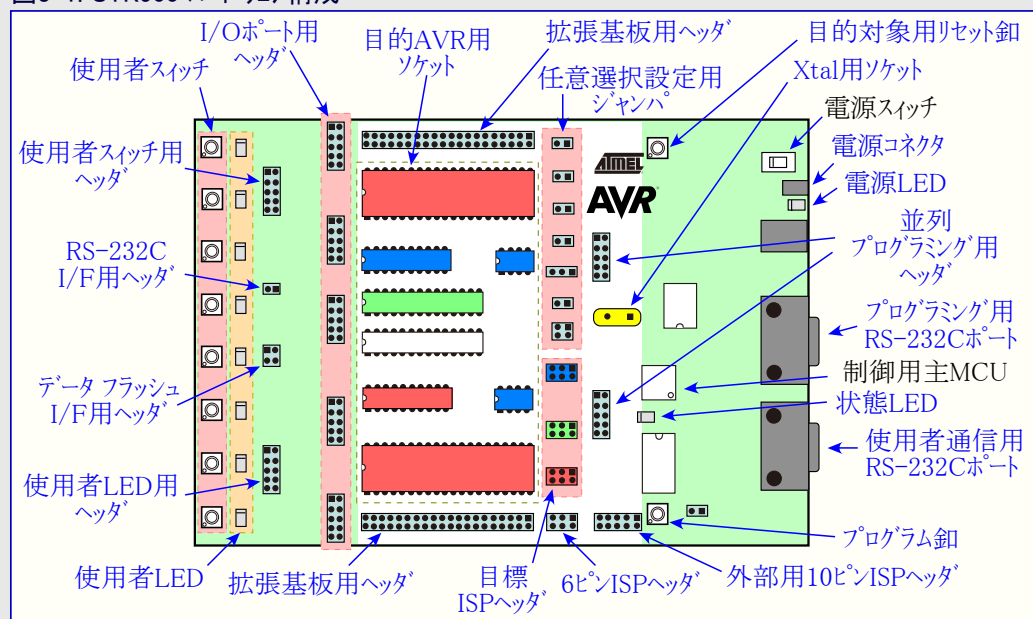


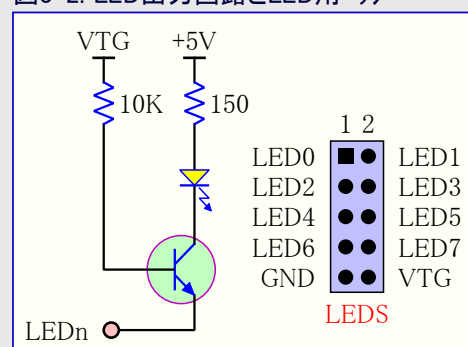
図3-1. STK500 ハードウェア構成



3.1 使用者LED

STK500スタータキットは8つの押し釦スイッチと8つの黄色LEDを含みます。このLEDとスイッチは基板の他の部分から分離され、デバッグ用ヘッダに接続されています。これらはAVR I/Oポートのヘッダへの提供された10芯ケーブルで、AVRデバイスに接続できます。図3-4はLEDとスイッチがI/Oポートヘッダにどう接続されるかを示します。I/OポートヘッダからLEDやスイッチへのケーブルは直結されるべきです。このケーブルはねじられるべきではありません。ケーブルの赤線が1ピンを示します。これが各ヘッダの1番ピンに接続されることを確かめます。図3-2はLED制御がどう実装されているかを示します。この方法は1.8~6.0Vの全目的電圧で同一LED光量を提供します。

図3-2. LED出力回路とLED用ヘッダ



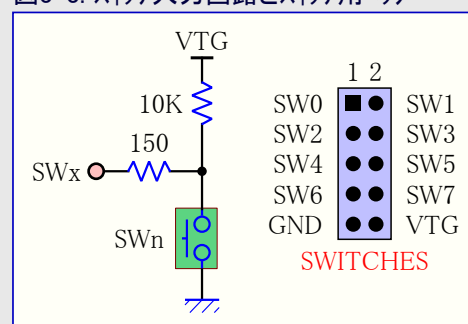
注: AVRは直接LEDを駆動するのに十分な吐き出しと吸い込みの電流を流せます。STK500の設計では、どんな目的電圧(VTG)でも同一LED光量を提供するのに、トランジスタと2本の抵抗が使われますが、VTGがなくなるとLEDをOFFにしてしまいます。

3.2 使用者スイッチ

デバッグ用ヘッダに接続されるスイッチは図3-3で示されるように実装されます。スイッチ押下は対応するSWxがLowに引き込まれる原因になり、一方開放はスイッチ用ヘッダに供すVTGに帰着します。有効な目的電圧範囲は1.8V<VTG<6.0Vです。

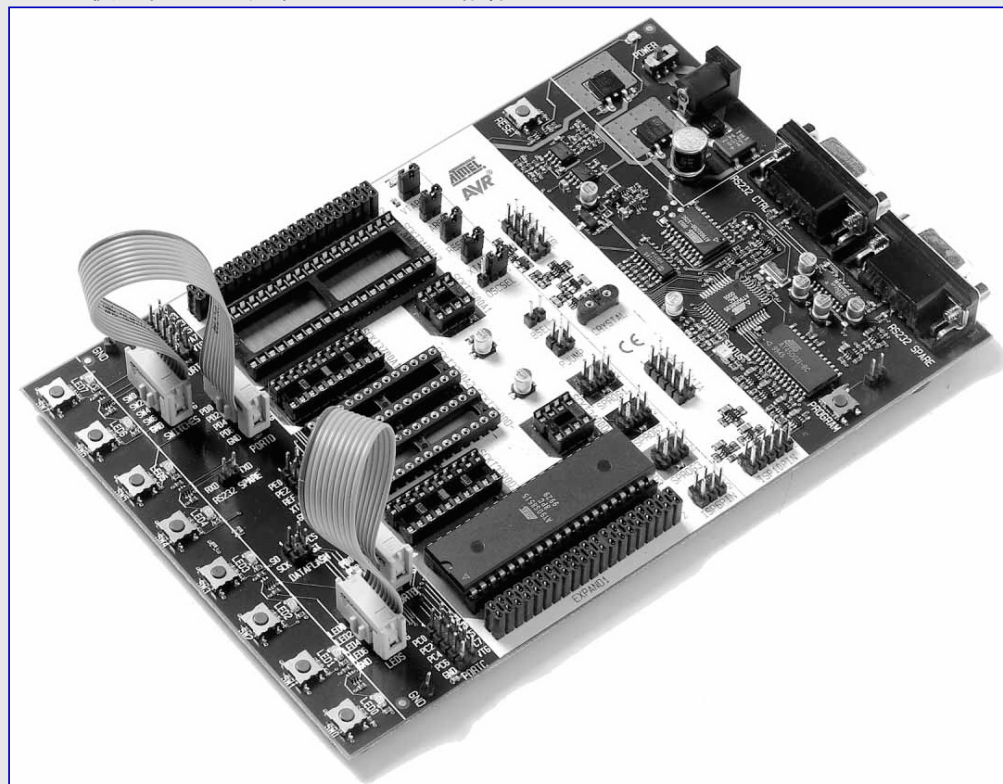
注: AVRでは入力ピンの内蔵プルアップを許可でき、押し釦スイッチの外部プルアップについては取り外しが必要です。STK500の設計では押し釦スイッチ非押下時にSWxの論理1を提供するために10kΩ外部プルアップが追加されています。150Ωの抵抗はAVRへの入力電流を制限します。

図3-3. スwitch入力回路とスイッチ用ヘッダ



3.3 LEDとスイッチの接続

図3-4. 使用者用LED、スイッチとI/Oポートの結線



AVRのどのI/Oポートも10芯ケーブルを使ってこのLEDやスイッチに接続できます。この(I/Oポート用)ケーブルは信号線に加えてVTG(目的電源電圧:VCC)とGNDを供給します。

3.4 I/Oポートコネクタ

標準I/Oポート用ヘッダのピン配置は図3-5で示されます。四角印が1番ピンを示します。

PORTE/AUXヘッダにはPORTEピンの信号に加え、いくつかの特別な信号と機能があります。このヘッダのピン配置は図3-6で示されます。

図3-5. 標準I/Oポート用ヘッダのピン配置

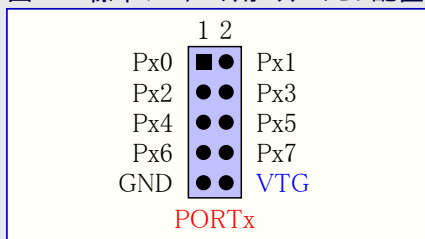
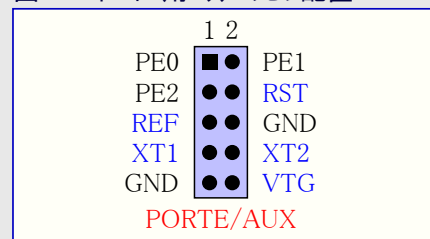


図3-6. ポートE用ヘッダのピン配置



このPORTE/AUXポートの特殊機能は以下のとおりです。

PE0～PE2 これらのピンは次表で示すようにデバイスによって異なります。

表3-1. PE0～2の割り当て（代表例）

信号名	ATmega161	AT90S4414/AT90S8515
PE0	PE0/ICP/INT2	ICP
PE1	PE1/ALE	ALE
PE2	PE2/OC1B	OC1B

REF アナログ基準電圧です。このピンは独立したアナログ基準電圧ピンを持つデバイスのA REFピンに接続されます。

XT1 XTAL1ピンです。全ソケットへの内部主クロック信号(出力)です。XTAL1ジャンパが切断される場合、このピンは外部クロック信号(入力)として使うことができます。

XT2 XTAL2ピンです。XTAL1ジャンパが切断される場合、このピンはXT1ピンとで外部クリスタル用に使うことができます。

LEDやスイッチ用ヘッダはI/Oポート用ヘッダと同じピン配置を使います。スイッチ用ヘッダのピン配置は図3-7で、LED用ヘッダのピン配置は図3-8で示されます。四角印が1番ピンを示します。

図3-7. 使用者スイッチ用ヘッダのピン配置

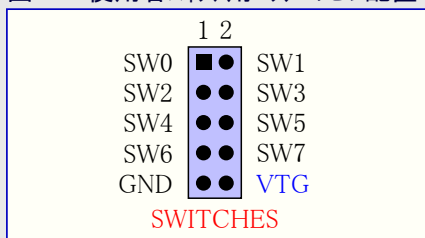
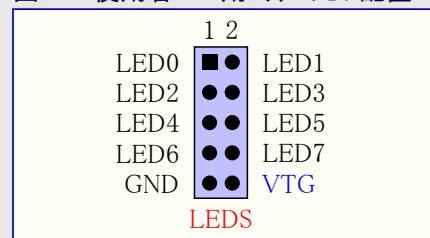


図3-8. 使用者LED用ヘッダのピン配置



3.5 使用者用RS-232C インターフェース

STK500は2つのRS-232Cポートを含みます。1つのRS-232C(**RS232 CTRL**)はAVR Studioとの通信に使われます。他方のRS-232C(**RS232 SPARE**)は直列ポートがRS-232Cに接続されたパソコンとソケット内の目的AVR マイクロ コントローラ間の通信に使えます。RS-232Cを使うにはAVRのUARTピンがRS232Cへ物理的に接続されなければなりません。

RS232 SPAREと記載された2ピンのヘッダはRS232Cレベル変換器をソケット上の目的AVRマイクロ コントローラのUARTピンへ接続するのに使えます。UARTピンをRS-232Cへ接続するには2芯ケーブルを使います。この接続は図3-9.で示されます。RS-232C接続の構成図は図3-10.で示されます。

図3-9. I/OピンのUART接続

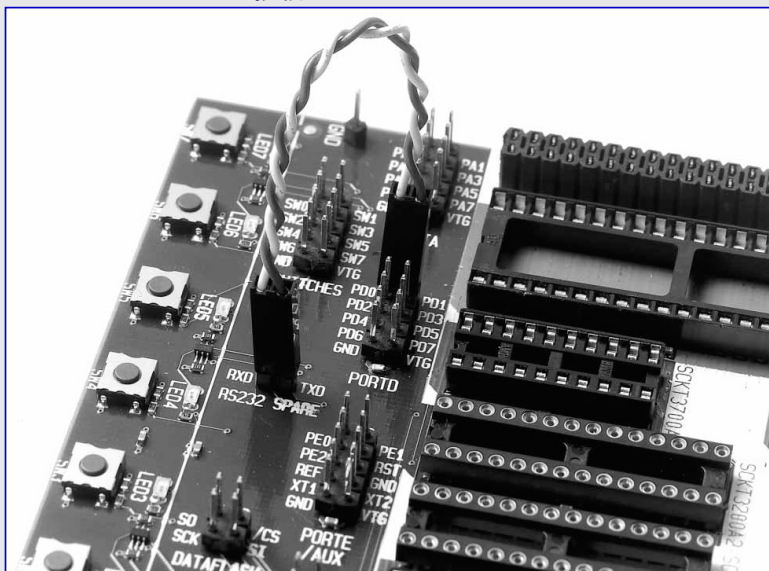
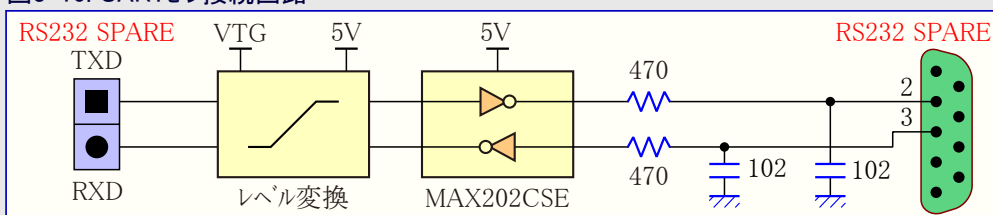


図3-10. UARTピン接続回路



3.6 データ用フラッシュ メモリ

注: AT45D021データフラッシュは製造終了で、従って後期のSTK500基板では実装されていません。

不揮発性データ保管場所としてAT45D021 2Mビット DataFlashがSTK500上に含まれます。DataFlashはSPI直列インターフェースの高密度フラッシュ メモリです。DataFlashの詳細データシートはAtmel CD-ROMのフラッシュ メモリ部門、またはAtmelのウェブサイトから入手できます。

DataFlashはマイクロ コントローラ ソケットのI/Oピンに接続できます。**DATAFLASH**と記載された4ピンヘッダはDataFlashのSPIインターフェースをソケット内の目的AVRマイクロ コントローラへ接続するのに使えます。DataFlashのI/Oピンへの接続用2芯ケーブルはSTK500に含まれます。DataFlashがAVRマイクロ コントローラのPORTBのハードウェアSPIインターフェースに接続される場合、供給された10芯ケーブルも使えます。I/Oピンの接続は図3-13.で示されます。ハードウェアSPIインターフェースへのDataFlashの接続に関するDataFlash接続の構成図は図3-14.で示されます。SPIインターフェース ピン配置は図3-11.と図3-12.で示されます。

図3-11. 40ピン デバイス SPI ピン配置

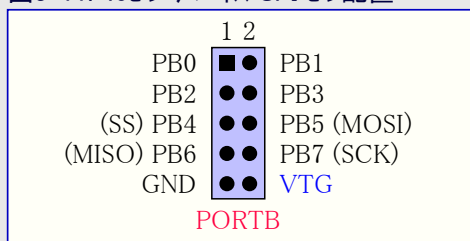
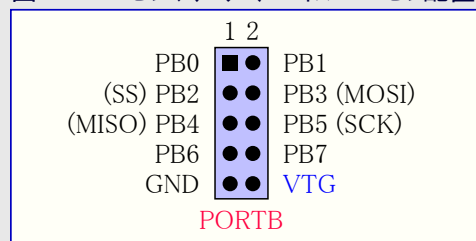


図3-12. 28ピン アナログ デバイス SPI ピン配置



3.7 目的対象AVRソケット

プログラミング部はスタータキット中央、白色領域内の8つのソケットで構成されます。これらのソケットには目的のAVRデバイスが応用での使用やプログラミング用に挿入できます。

注: ソケットには同時に1つのAVRデバイスだけが挿入されるべきです。

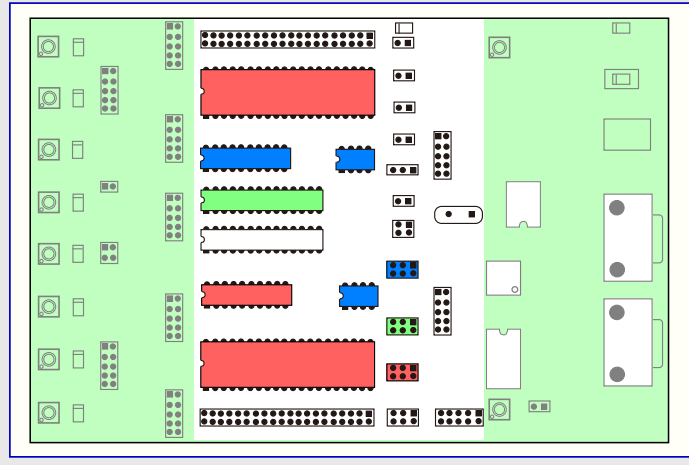
AVRのフラッシュメモリは1000(デバイスによっては10000)回のプログラミング操作に関して間違いがないことが保証されています。フラッシュメモリの代表的な寿命はもっとと長寿命です。

注: ソケットへのデバイス挿入時にデバイスの向きを確認してください。

デバイス短辺のノッチ(くぼみ)はソケットの窪みと一致しなければなりません。デバイスが誤った方法で挿入されると、デバイスやスタータキットに損害を与えるかもしれません。

ソケット部は応用(プログラム)実行とデバイスのプログラミング両方で使われます。

図3-15. STK500 プログラミング基部



ソケットに挿入されたデバイスはAVR Studioから2つの異なる方法でプログラミングできます。

1. デバイスが動作している通常供給電圧でのAVR実装書き込み(ISP)。
2. 供給電圧が常に5Vでの高電圧プログラミング。

4つの標準信号(VTARGET, RESET, XTAL1, AREF)がソケット部に接続できます。

後続の節は両方のプログラミング法の使用方法を記述します。AVR Studio プログラミング ソフトウェア使用の教示については5-1頁の「[第5章 AVR Studioの使用法](#)」を参照してください。

3.7.1 ISPプログラミング

実装書き込み(ISP)はAVRのフラッシュメモリとEEPROMにコード(データ)をダウンロード(書く)ために、AVRの内部SPI(直列周辺インターフェース)を使います。ISPプログラミングはVCC,GND,RESETとプログラミング用の3つの信号線だけが必要です。AT90C8534,ATtiny10/11/28,ATmega406を除く、全AVRデバイスでISPプログラミングができます。AVRは一般的に2.7~6.0(1.8~5.5)Vの通常動作電圧でプログラミングできます。高電圧信号は必要ありません。ISP書き込み器は内部フラッシュメモリとEEPROMの両方にプログラム(読み書き)できます。多くのデバイスではクロック元、起動時間、低電圧検出(BOD)リセットの選択用ヒューズビットもプログラム(読み書き)します。

ISP周波数(SCK)が目的クロックの1/4よりも低くなければならないことに**注意**してください。けれどもこれは目的クロックとISPクロックの両方で50%:50%のデューティ比を必要とします。1/5またはSTK500で勧められた目的クロックよりも低いISP周波数で走らせてください。ISP周波数はSTK500使用者インターフェースのMainタブでISP選択後、ISP周波数設定(Settings...)鉤からの設定によって設定されます。

高電圧プログラミングはISPプログラミングで支援されないデバイスもプログラム(読み書き)できます。いくつかのデバイスは特定ヒューズビットのプログラミングに高電圧プログラミングが必要です。高電圧プログラミング使用法の教示については3-8頁の「[高電圧プログラミング](#)」項を参照してください。

プログラミングインターフェースがデバイスによって異なるピンに配置されるため、プログラミング信号の正しいピンへの配線に3つのプログラミングヘッダ(SPROGn)が使われます。6芯ケーブルはISP信号の目的ISPヘッダへの接続用に供給されます。色分けと番号は各ソケットにどのISPヘッダが使われるかを示すのに使われます。

ISPプログラミング中、6芯ケーブルはISP6PINと記載されたヘッダへ常に接続されていなければなりません。青色ソケットのデバイスにプログラミングする時は、ケーブル他端をSPROG1 ISPヘッダに接続します。同様に緑色ソケットの時はSPROG2、赤色ソケットの時はSPROG3 ISPヘッダを使います。表3-2. はどのソケットがどのAVRデバイス用か、ISPプログラミングにどのSPROGn ISPヘッダを使うかを示します。



6芯ケーブルはISP6PINヘッダから正しいSPROG_n目的ISPヘッダへ直結されるべきです。このケーブルはねじられるべきではありません。ケーブルの色付き(赤)線が1番ピンを示します。これが各ヘッダの1番ピンに接続されることを確認します。

8ピンデバイスのプログラミング時、以下に注意してください。いくつかのデバイス(ATtiny11/12/13/15/25/45/85)で1番ピンはRESETとPB5の両方に使われます。8ピンソケット(SCKT3400D1, SCKT3600A1)の1番ピンはPB5に接続されます。従ってISPプログラミング中に使われるRESET信号はこれらソケットの1番ピンに接続されません。この信号はPORTE/AUXヘッダのRSTとPORTBヘッダのPB5間の配線によって接続されなければなりません。PORTE/AUXヘッダのXT1(7番ピン)とPORTBヘッダのPB3(4番ピン)を接続するために2芯ケーブルを使います。これはAVRデバイスにクロックシステムを接続します。

表3-2. AVRソケット、ISPヘッダ対応表

AVRデバイス	STK500ソケット	色	番号	ISPヘッダ	注意
AT90S1200/2313 ATtiny2313	SCKT3300D3	赤	3	SPROG3	
AT90S2323/2343 ATtiny12/13/22/25/45/85	SCKT3400D1	青	1	SPROG1	1,3
ATtiny11					5
ATtiny28	SCKT3500D-	無し	無し		
AT90S4414/8515 ATmega8515/161/162	SCKT3000D3	赤	3	SPROG3	7
AT89S51 AT89S52					
AT90S4434/8535, ATmega8535/16/163 /164P/32/323/324P/644(P)/1284P	SCKT3100A3				
AT90S2333/4433 ATtiny48/88 ATmega48(P)/8/88(P,PA)/168(P)/328P	SCKT3200A2	緑	2	SPROG2	
ATtiny15	SCKT3600A1	青	1	SPROG1	1
ATtiny26/261/461/861	SCKT3700A1	青	1		2,4,8
ATmega64/103/128/1281/2561 AT90CAN32/128	STK501アダプタを使います。				
ATmega165(P)/169(P)/325(P)/329(P)/ 645/649	STK502アダプタを使います。				
ATmega640/1280/2560	STK503アダプタを使います。				
ATmega3250(P)/3290(P)/6450/6490	STK504アダプタを使います。				
ATtiny24/44/84	STK505アダプタを使います。				
AT90PWM2/2B/216/3/3B/316	STK520アダプタを使います。				
AT90USB646/647/1286/1287	STK525アダプタを使います。				
AT90USB82/162	STK526アダプタを使います。				
AT86RF401	AT86RF401-EK1経由で使います。				6

注1: PORTE/AUXヘッダのRSTとPORTBヘッダのPB5を接続します。

注2: PORTE/AUXヘッダのRSTとPORTBヘッダのPB7を接続します。

注3: AT90S2323はPORTE/AUXヘッダのXT1とPORTBヘッダのPB3を接続します。

注4: PORTE/AUXヘッダのXT1とPORTBヘッダのPB4を接続します。

注5: 高電圧プログラミングのみです。

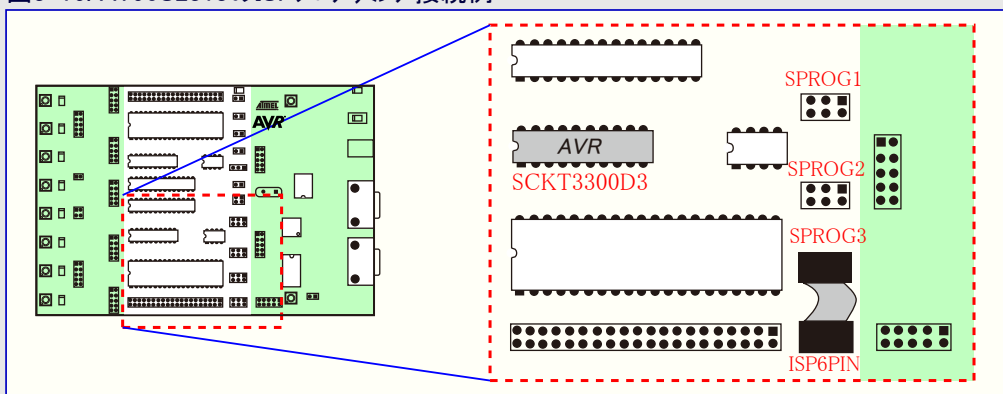
注6: ソケット支援なし、外部ISPプログラミングがAT86RF401評価キットを通して支援します。

注7: これらはMCS51コア互換デバイスで、AVRコアではありません。これらのデバイスではISPプログラミングだけが支援され、他の機能はありません。

注8: これらのデバイスはSTK505アダプタでも支援されます。

図3-16.はAT90S2313がISPプログラミングできる例を示します。6芯ケーブルはISP6PINヘッダから赤のSPROG3 ISPヘッダへ接続され、AT90S2313デバイスはSCKT3100D3と記載された赤のソケットに挿入されます。

図3-16. AT90S2313のISPプログラミング接続例



AVR内のプログラムを走行中にISPプログラミング用の6芯ケーブルを取り外す必要はありません。ISPプログラミングに使われるポートピンは使用者プログラムで他の目的に使えます。

3.7.2 高電圧プログラミング

高電圧プログラミングはAVRデバイスのRESETピンに12Vのプログラミング電圧が供給されます。全てのAVRデバイスが高電圧プログラミングでプログラミングでき、目的デバイスがソケットに実装されている間にプログラミングできます。

高電圧プログラミングについては2つの異なる方法が使われます。8ピンデバイスは直列プログラミングインターフェースを、一方他のデバイスは並列プログラミングインターフェースを使います。プログラミング信号はSTK500と共に供給されたケーブルを使って目的デバイスの正しいピンへ配線されます。

表3-3. はプログラミング法と高電圧プログラミング使用時の特別な考慮を示します。

表3-3. 高電圧プログラミング設定

AVRデバイス	STK500ソケット	色	番号	備考
AT90S1200/2313 ATtiny2313	SCKT3300D3	赤	3	高電圧並列プログラミング。PROG CTRLとPORTDヘッダ間、PROG DATAとPORTBヘッダ間を接続(基本)。(図3-17.参照)
AT90S4414/8515	SCKT3000D3			
AT90S4434/8535	SCKT3100A3			
ATtiny28	SCKT3500D-	無	無	
ATmega8515/161/162	SCKT3000D3	赤	3	高電圧並列プログラミング。基本接続にBSEL2ジャンパを装着。
ATmega8535/16/163/164P/32/323/324P/644(P)/1284P	SCKT3100A3			
AT90S2333/4433	SCKT3200A2	緑	2	高電圧並列プログラミング。基本接続にPJUMPジャンパを装着。 高電圧並列プログラミング。基本接続にPJUMPジャンパを装着。BSEL2とPC2間を接続。
ATtiny48/88 ATmega48(P)/8/88(P,PA)/168(P)/328P				
AT90S2323/2343 ATtiny11/12/13/22/25/45/85	SCKT3400D1	青	1	高電圧直列プログラミング。
ATtiny15	SCKT3600A1			
ATtiny26/261/461/861	SCKT3700A1	青	1	高電圧並列プログラミング。要特殊ケーブル(注1)。
ATtiny24/44/84	STK505使用	無	無	高電圧並列プログラミング。 高電圧直列プログラミング。
ATmega64/103/128/1281/2561, AT90CAN32/64/128	STK501使用 (STK501使用者の手引き参照)			
ATmega165(P)/169(P)/325(P)/329(P)/645/649	STK502使用 (STK502使用者の手引き参照)			
ATmega640/1280/2560	STK503使用 (STK503使用者の手引き参照)			
ATmega3250(P)/3290(P)/6450/6490	STK504使用 (STK504使用者の手引き参照)			
AT90PWM2/2B/216/3/3B/316	STK520使用 (STK520使用者の手引き参照)			
AT90USB646/647/1286/1287	STK525使用 (STK525使用者の手引き参照)			
AT90USB82/162	STK526使用 (STK526使用者の手引き参照)			

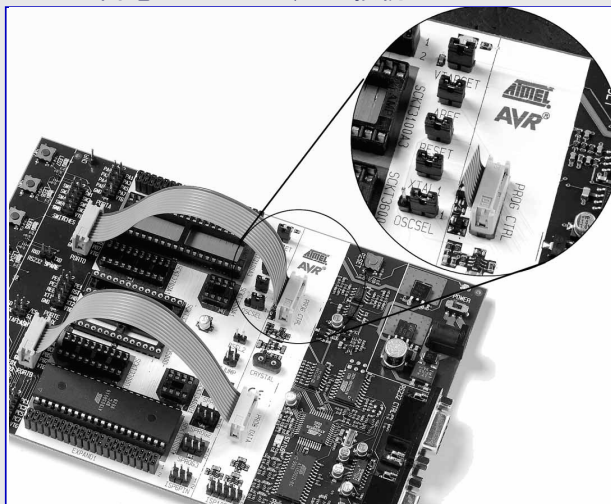
注: STK5xxアダプタ使用時の接続に関してはそのアダプタの使用者の手引きをご覧ください。

注1: 特殊なケーブル接続が必要です。詳細については3.8.7節を参照してください。

3.7.2.1 高電圧並列プログラミング

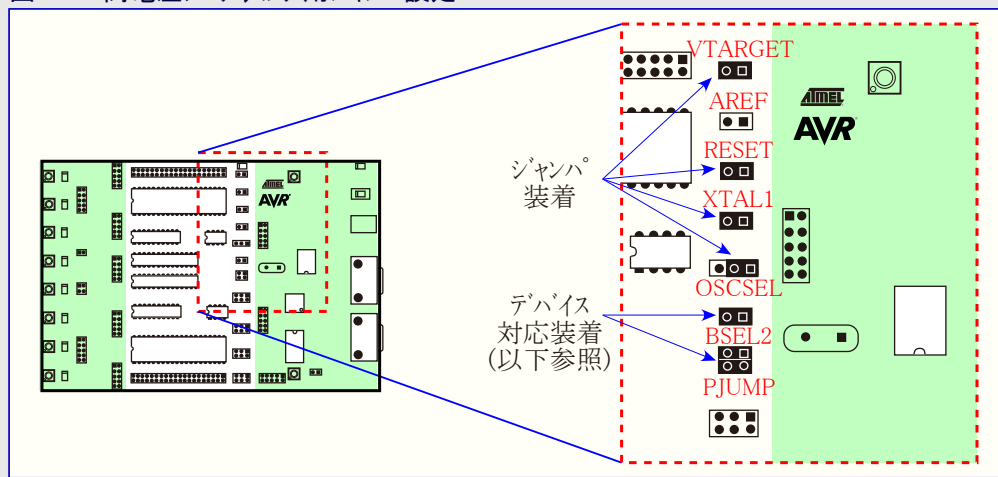
高電圧プログラミングを使うにはプログラミング信号がAVRのI/Oピンに配線されなければなりません。図3-17.で示されるようにSTK500と共に供給された2本の10芯ケーブルが**PROG DATA**ヘッダと**PORTB**ヘッダ、**PROG CTRL**ヘッダと**PORTD**ヘッダの接続に使えます。

図3-17. 高電圧並列プログラミング接続



高電圧プログラミング使用時にSTK500上のいくつかのジャンパ設定が変更されなければなりません。図3-18.はこれらのジャンパ設定を示します。

図3-18. 高電圧プログラミング用ジャンパ設定



高電圧並列プログラミング用のハードウェア設定を次に示します。

1. 電源をOFFにします。
2. 3-9頁の表3-3.に従って書き込むデバイスをソケットに装着します。
3. **PROG DATA**ヘッダと**PORTB**ヘッダを10芯ケーブルで接続します。
4. **PROG CTRL**ヘッダと**PORTD**ヘッダを10芯ケーブルで接続します。
5. ソフトウェアで制御されたクロックを選択するために**OSCSEL**の1番ピンと2番ピンにジャンパを装着します。
6. デバイスへ発振器信号を配線するために**XTAL1**ジャンパを装着します。
7. **VTARGET**と**RESET**ジャンパを装着します。
8. AT90S2333/4433, ATtiny48, ATmega48/48P/8/88/88P/88PA/168/168P/328Pのプログラミング時に**PJUMP**ジャンパを装着します。2芯ケーブルがジャンパの代わりに使えます。「[3.8.6 PJUMPジャンパ](#)」をご覧ください。
9. 必要とするデバイスに対して**BSEL2**ジャンパを装着します。「[3.8.5 BSEL2ジャンパ](#)」をご覧ください。
10. 目的応用回路をSTK500から切り離します。
11. 電源を投入します。
12. プログラミング前にVTARGET電圧が4.5～5.5V間かを確認します(5.3.6.1参照)。

ジャンパ設定の完全な説明については「[3.8 ジャンパ設定](#)」章を参照してください。

注: デバッグを開始する前には高電圧プログラミング用ハードウェア設定を取り外してください。



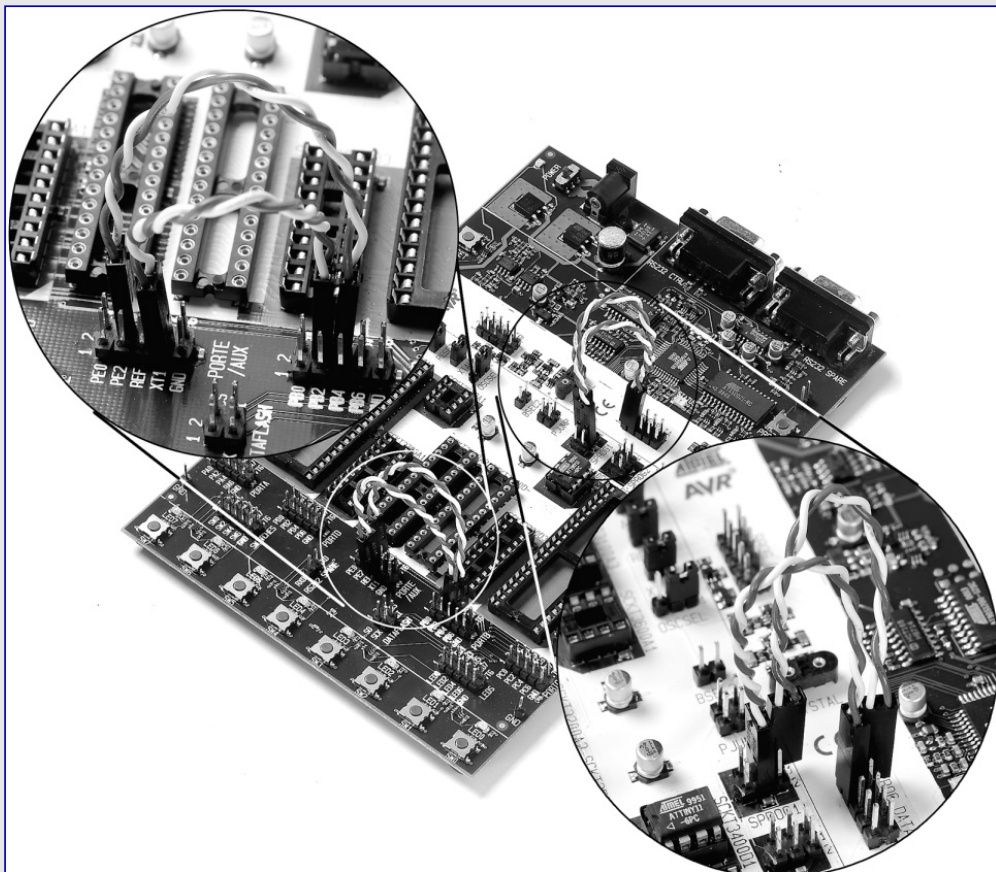
3.7.2.2 高電圧直列プログラミング

少数ピンのAVRは高電圧プログラミング中の並列通信を使うにはピン数が少なすぎます。これらは代わりに直列通信を使います。これはより少ない信号が配線されなければならないことを意味します。次に高電圧直列プログラミングのハードウェア設定を示します。

1. 電源をOFFにします。
2. 3-9頁の表3-3に従って書き込むデバイスをソケットに装着します。
3. ソフトウェアで制御されたクロックを選択するために**OSCSEL**の1番ピンと2番ピンにジャンパを装着します。
4. デバイスへ発振器信号を配線するために**XTAL1**ジャンパを装着します。
5. **VTARGET**と**RESET**ジャンパを装着します。
6. **PORTB**ヘッダのPB3(4番)ピンと**PORTE/AUX**ヘッダのXT1(7番)ピンを接続するために1つの2芯ケーブルを使います。これはクロックシステムをAVRデバイスに接続します。
7. **PORTB**ヘッダのPB5(6番)ピンと**PORTE/AUX**ヘッダのRST(4番)ピンを接続するために他の2芯ケーブルを使います。これはリセットシステムをAVRデバイスに接続します。
8. **SPROG1**ヘッダのPB0,PB2(4,3番)ピンと**PROG DATA**ヘッダのDATA0,DATA2(1,3)ピンを接続するために3つ目の2芯ケーブルを使います。
9. **SPROG1**ヘッダのPB1(1番)ピンと**PROG DATA**ヘッダのDATA1(2番)ピンを接続するために最後の2芯ケーブルを使います。
10. 電源を投入します。
11. プログラミング前にVTARGET電圧が4.5～5.5V間かを確認します。

全ての接続は図3-19に示されます。

図3-19. 高電圧直列プログラミング用接続



3.8 ジャンパ°設定

主マイクロコントローラと8つのジャンパ°がスタータキットのハードウェア設定を制御します。通常の動作中、これらのジャンパ°は既定位置が装着されるべきです。スタータキットの高度な使用設定のためにジャンパ°は脱着できます。ジャンパ°設定と使用法は次章で示されます。既定のジャンパ°設定は図3-20.で示されます。

図3-20. 既定ジャンパ°設定

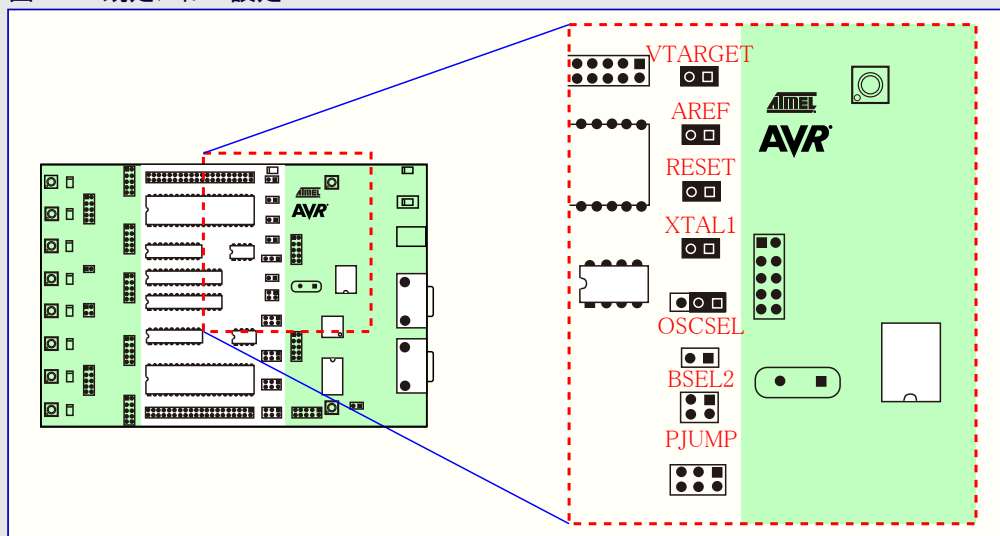


表3-4. ジャンパ°一覧

ジャンパ°名	既定	意味
VTARGET	装着	基板内VTARGET供給電源接続。
AREF	装着	基板内アナログ基準電圧接続。
RESET	装着	基板内リセット部信号接続。
XTAL1	装着	基板内クロック部信号接続。
OSCSEL	装着	基板内発振器選択。
BSEL2	未装着	高電圧プログラミング時に装着を必要するデバイスの一覧については「BSEL2ジャンパ°」をご覧ください。 ATmega48/48P/8/88/88P/168/168P/328Pの高電圧プログラミング時はPC2に接続。
PJUMP	未装着	高電圧プログラミング時に装着を必要するデバイスの一覧については「PJUMPジャンパ°」をご覧ください。

3.8.1 VTARGET 目的VCC設定

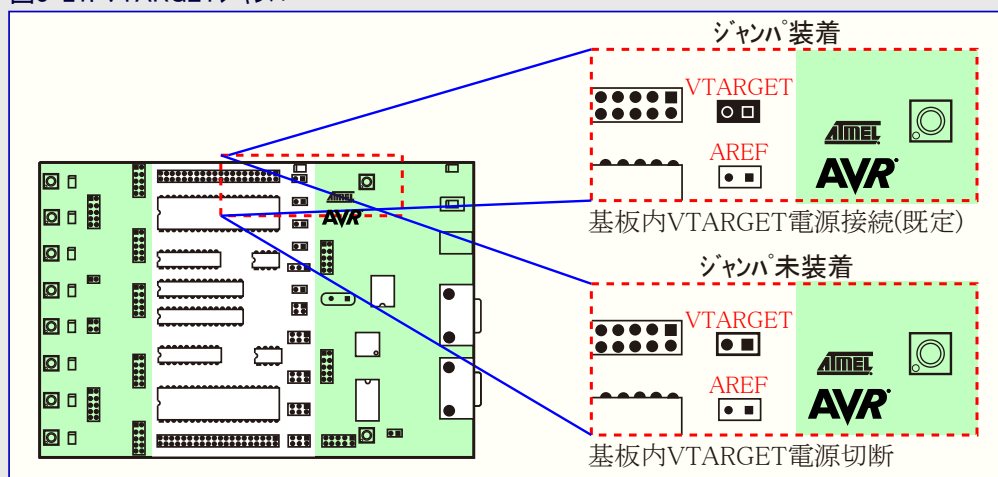
VTARGETは目的AVRマイクロコントローラのソケットへの供給電源を制御します。AVR Studioで制御される基板内電源か、外部電源かどちらかにできます。VTARGETジャンパ装着で基板内電源が接続されます。基板内電源はAVR Studioから0～6.0Vに調整できます。VTARGET電圧調整前にAVRデバイスの動作電圧に関して各々のデータシートを常に確かめてください。

基板内電源の使用は目的回路部に約0.5Aが配給できます。目的回路部の範囲について理解するには「付録A STK500構成図」を参照してください。

VTARGETジャンパ未装着の場合、目的対象のVCCは外部電源からPORTxヘッダのVTGピンの1つに供給されなければなりません。図3-21はVTARGETジャンパ切り替えを示します。

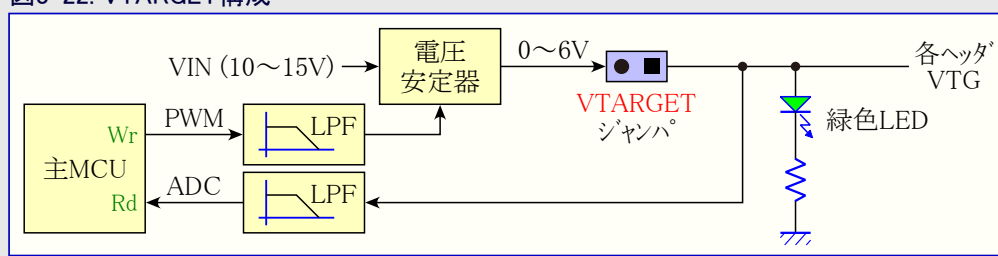
VTARGETに外部電源を使う時はVTARGETをAREF(アナログ基準)電圧以上に制御しなければなりません。外部VTARGET電圧使用時、常に共通GNDを接続します。

図3-21. VTARGETジャンパ



STK500の主マイクロコントローラは内蔵PWMを使用して目的電圧を制御します。図3-22はVTARGET信号の内部接続を示します。

図3-22. VTARGET構成



注: 緑色LEDはVTG利用可能時に点灯します。VTGなしでSTK500のプログラミングやデバッグ領域の使用は不可能です。

内部VTARGETには短絡防止回路があります。VTARGETが0.3Vよりも高く設定され、主マイクロコントローラが80ms間0.3V以下を計測すると、主マイクロコントローラはこのVTARGETと内部AREFを遮断します。これが起こると、状態LEDが高速点滅します。

外部VTARGETが使われ、且つ外部VTARGETがOFFに切り替わると、これも(偽の)短絡としてSTK500によって検出されます。後者の場合、内部VTARGET電圧を0.3V以下に下げることによって(発生を)無くせます。

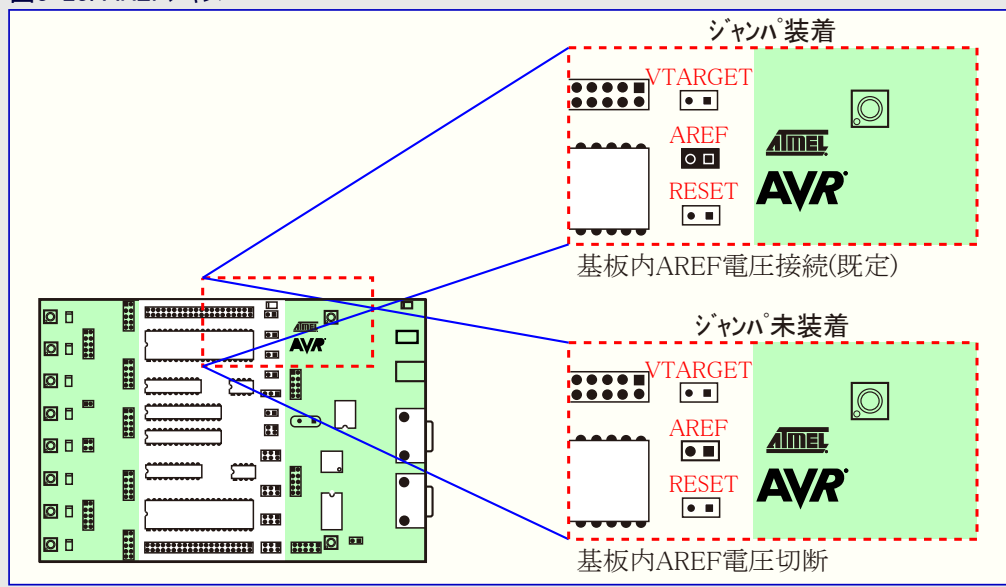
3.8.2 AREF アナログ基準電圧

アナログ基準電圧(AREF)はAVRの内蔵A/D変換器へ基準電圧を供給できます。AREFジャンパが装着されると、基板内アナログ基準電圧がAVRのAREFに接続されます。基板内アナログ基準電圧は0～6.0VにAVR Studioから調整できますが、VTARGET電圧を超えられません。

AREFジャンパ未装着時、AREF電圧は外部電源からPORTE/AUXヘッダのAREFピン(図3-6参照)に供給されなければなりません。図3-23はAREFジャンパ切り替えを示します。

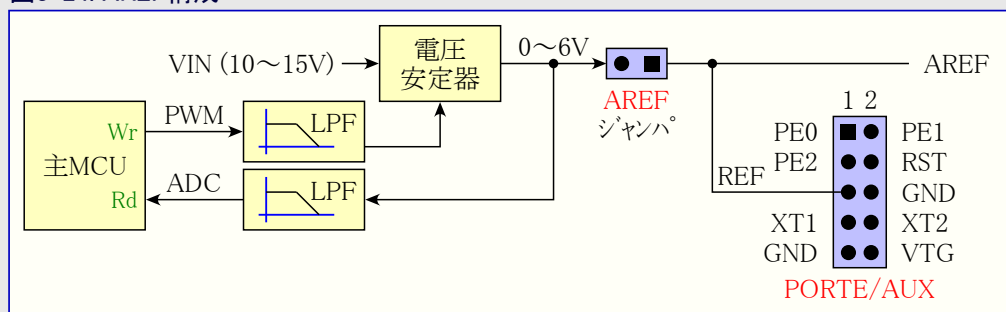
AREFに外部アナログ基準電圧を使う時はVTARGET(目的電源電圧)をAREF以上に制御しなければなりません。これはAREF設定前にAVR StudioからVTG値を読むことによって容易に制御できます。

図3-23. AREFジャンパ



STK500の主マイクロコントローラは内蔵PWMを使って基板内AREF電圧を制御します。AVRのAREF信号はPORTE/AUXヘッダでも入出力でき、このピンは外部AREF信号にも使えます。図3-24はAREF信号の内部接続を示します。

図3-24. AREF構成



AVR Studioで制御されるアナログ基準電圧はAVRのA/D変換計測に関して、またはアナログ比較器の入力としても使えます。AVRのAREF信号はVTGへの接続もできます。

内部AREFには短絡防止回路があります。AREF値が0.3Vよりも高く設定され、主マイクロコントローラが80ms間0.3V以下を計測すると、主マイクロコントローラはAREFを遮断します。これが起きると、状態LEDが低速点滅します。

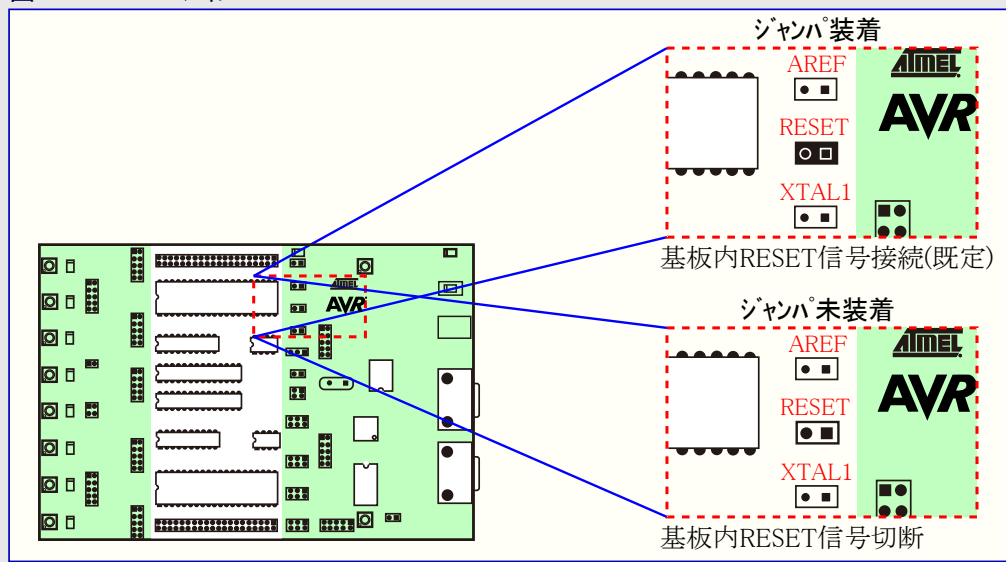
短絡回路がVTARGETで検出されると、AREFも主マイクロコントローラによって遮断されます(加えて内部VTARGETの遮断)。この場合、状態LEDは高速点滅します。

3.8.3 RESET リセット選択

RESETジャンパはSTK500のRESET信号を制御します。ソケット内の目的デバイスのISPプログラミング時、主マイクロコントローラは応用回路の邪魔をせずにAVRデバイスをプログラミングします。RESETジャンパ装着時、主マイクロコントローラはAVRのRESET信号を制御します。RESETジャンパ未装着時、RESET信号は切断されます。これは外部リセットシステム付きの応用試作品に有用です。

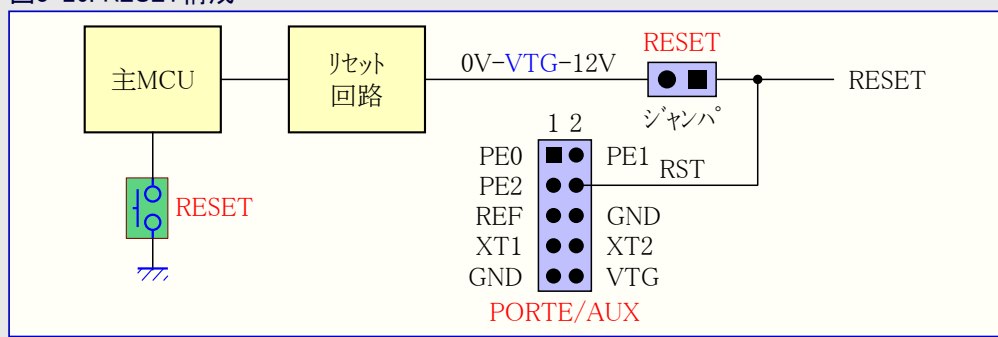
AVRデバイスの高電圧プログラミング時、常にRESETジャンパを装着しなければなりません。外部リセットシステム使用時、プログラミング中のリセットシステムは主マイクロコントローラによって制御されるリセット信号を許さなければなりません。RESETジャンパ未装着時、RESET鉤は切り離されます。図3-25はRESETジャンパ選択を示します。

図3-25. RESETジャンパ



STK500上の主マイクロコントローラは目的AVRへのRESET信号を制御します。AVRデバイスのRESET信号はPORTE/AUXヘッダでも入出力でき、このピンは外部RESET信号にも使えます。図3-26はRESET信号の内部接続を示します。

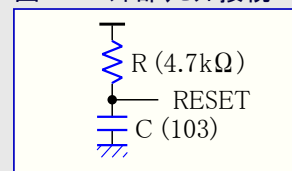
図3-26. RESET構成



注: 高電圧プログラミング中にSTK500はAVRのRESET信号へ12Vを供給します。従ってAVRの高電圧プログラミング前に、これを操作しかねない外部リセット回路は切断されなければなりません。

外部システムに接続される時に外部プルアップ抵抗がRESET信号線に接続されることが多々あります。代表的なリセット接続は図3-27に示されます。外部プルアップ抵抗が低すぎる(<4.7kΩ)場合、STK500はRESET信号線をLowに引き込めません。

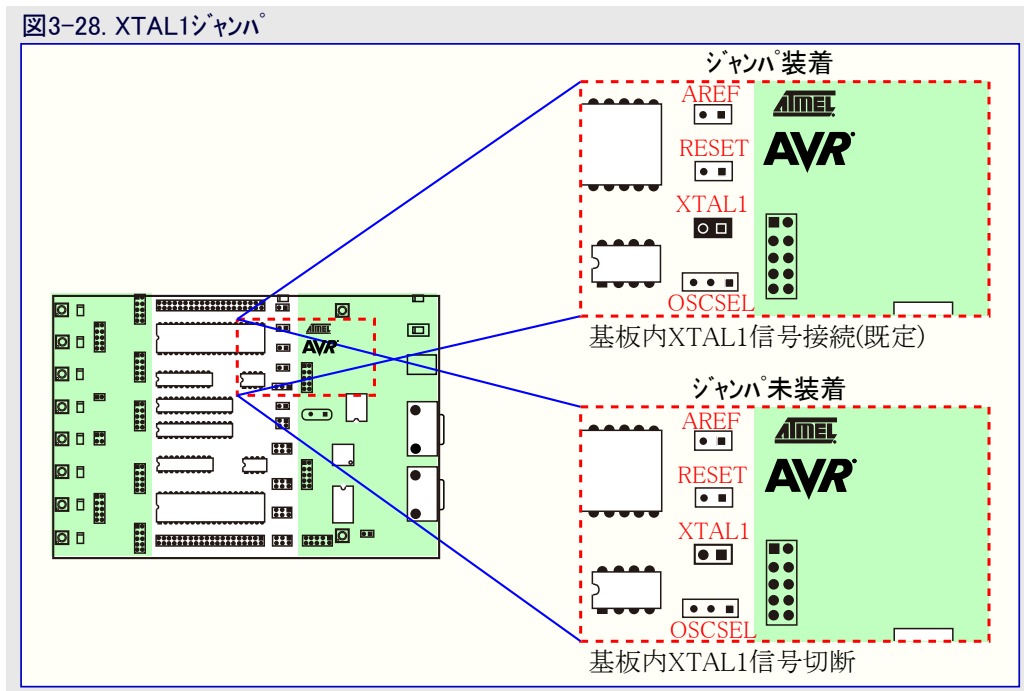
図3-27. 外部リセット接続



3.8.4 XTAL1, OSCSEL クロック選択

STK500には目的AVR用に多くのクロック元があります。**XTAL1**と**OSCSEL**ジャンパの設定はクロック選択を制御します。**OSCSEL**ジャンパはAVRのXTAL1ピンに配線する信号を決めます。

XTAL1ジャンパ装着時、STK500内部クロックシステムが目的AVRの主クロックとして使われます。**XTAL1**ジャンパ未装着時、内部クロックシステムは切り離されます。これはAVRへの目的クロック元として外部クロック信号またはクリスタルが使われるのを許します。図3-28は**XTAL1**ジャンパ選択を示します。



XTAL1ジャンパ未装着時、**PORTE/AUX**ヘッダに外部クロック信号またはクリスタルが接続できます。これは図3-30で示されます。

XTAL1ジャンパ装着時、STK500の内部クロックシステムが目的AVRへの主クロックとして使われます。内部クロックシステムは基板上のクリスタルソケット内のクリスタル発振子か、主マイクロコントローラからソフトウェア生成されたクロックのどちらかが使えます。ソフトウェア生成されたクロックの周波数は0~3.68 MHzに設定できます。既定値は3.68MHzです。5-11頁の5.3.6項はAVR Studioからこのクロック周波数を設定する方法を説明します。

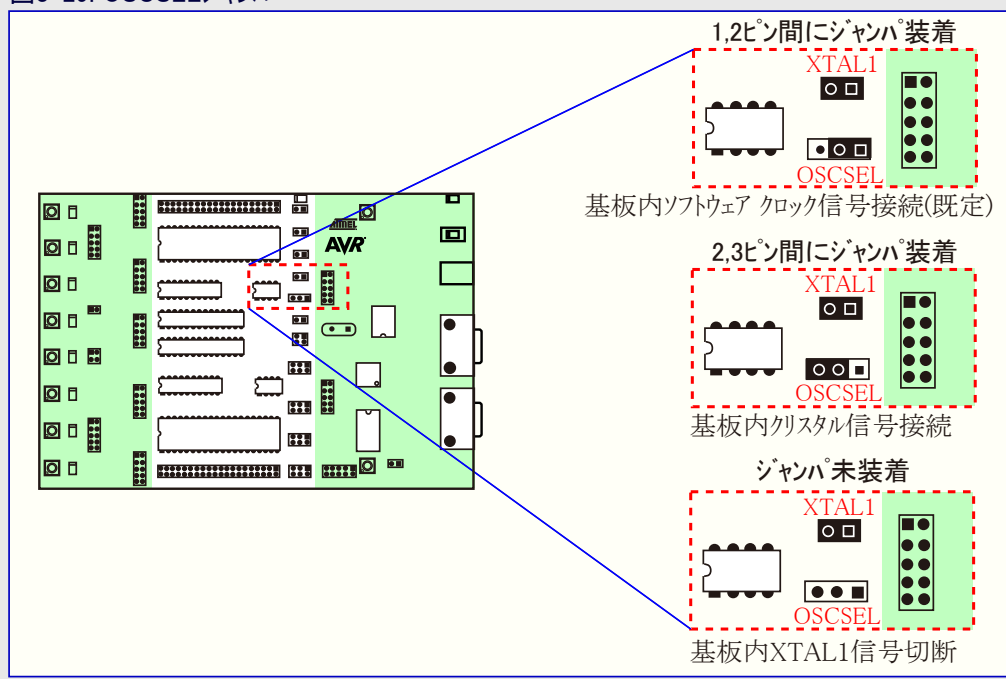
主クロックとしてSTK500のソフトウェア生成されたクロックを使う時に目的AVRマイクロコントローラのヒューズはクロック元として**外部クロック**用に設定されるべきです。これはマイクロコントローラに最短起動時間を与えます。起動時間の詳細についてはAVRマイクロコントローラのデータシートを参照してください。クロック元ヒューズ設定の説明については5-6頁の5.3.3項を参照してください。AVRデバイスの全てがクロック元としてクリスタルまたは発振器のどちらを使うか選ぶためのヒューズを持っている訳ではありません。

内部クロックシステムは**OSCSEL**ジャンパで選択されます。図3-29は**OSCSEL**ジャンパ選択を示します。

基板内発振器は2~20MHzのセラミック振動子またはクリスタル発振子(ATカット、並列共振)で動作します。



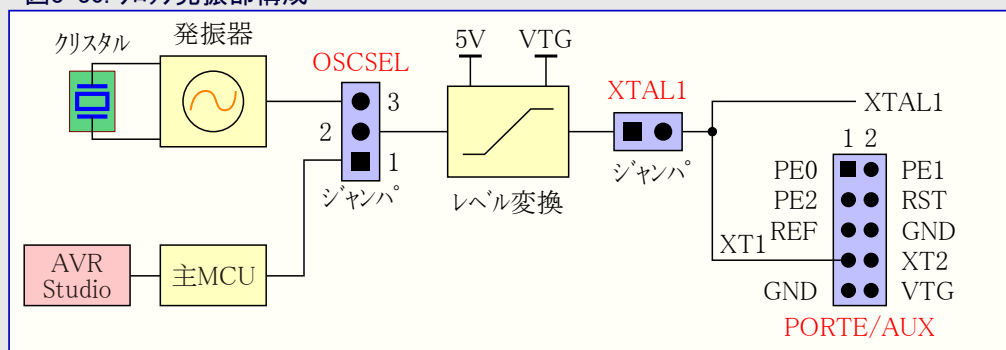
図3-29. OSCSELジャンパ



高電圧プログラミング動作でAVRをプログラミングする時に目的クロックの制御を主マイクロコントローラにさせるため、OSCSELは1,2番ピンが装着されるべきです。これは3-9頁の3.7.2節で詳細が説明されます。

注: クリスタルに接続された唯一のAVRでの実際の応用に於いて外部発振器回路は必要ありません。STK500には同じクロックシステムに接続された8つの異なるAVRソケットがあります。このシステム内の長い信号線はAVR内蔵発振器とクリスタルでの駆動を困難にしまいます。STK500の発振器は1.8~6.0Vの全目的電圧で動作するように設計されています。

図3-30. クロック発振部構成



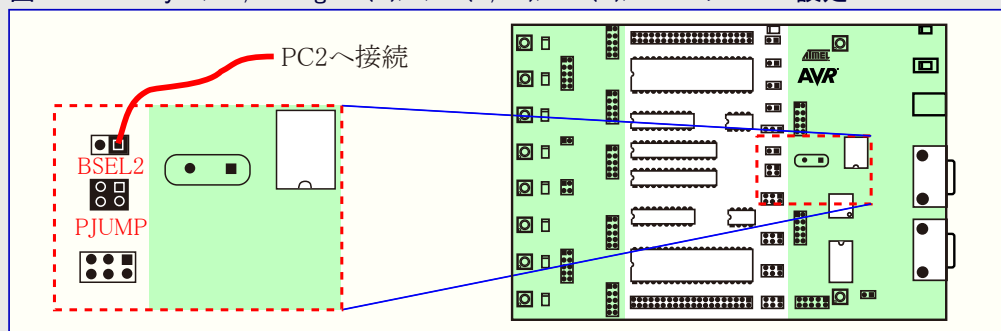
3.8.5 BSEL2

BSEL2ジャンパは以下で一覧されるデバイスの高電圧プログラミング用バイト選択2信号を接続します。BSEL2ジャンパは高電圧プログラミング時に装着されるべきです。

•ATmega8515	•ATmega8535	•ATmega16	•ATmega161	•ATmega162
•ATmega163	•ATmega164P	•ATmega165(P)	•ATmega169(P)	•ATmega32
•ATmega323	•ATmega324P	•ATmega325(P)	•ATmega3250(P)	•ATmega329(P)
•ATmega3290(P)	•ATmega64	•ATmega640	•ATmega644(P)	•ATmega645
•ATmega6450	•ATmega649	•ATmega6490	•ATmega103	•ATmega128
•ATmega1280	•ATmega1281	•ATmega1284P	•ATmega2560	•ATmega2561
•AT90CAN32	•AT90CAN64	•AT90CAN128	•AT90USB82	•AT90USB162
•AT90USB646	•AT90USB647	•AT90USB1286	•AT90USB1287	

ATtiny48, ATtiny88, ATmega48(P)/8/88(P,PA)/168(P)/328P使用時、右側のBSEL2ピンを目的対象領域内(PORTCヘッダ)のPC2へ接続します(図3-31.参照)。バイト選択2(BS2)信号の記述については対応するデバイスのデータシートでプログラミング章を参照してください。

図3-31. ATtiny48/88, ATmega48(P)/8/88(P,PA)/168(P)/328PのBSEL2設定



3.8.6 PJUMP

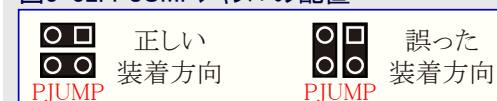
PJUMPジャンパは次のデバイスの高電圧プログラミング使用時にプログラミングピンをプログラミング信号に配線します。

•AT90S2333	•AT90S4433	•ATtiny48	•ATtiny88	•ATmega48(P)
•ATmega8	•ATmega88(P,PA)	•ATmega168(P)	•ATmega328P	

注: PJUMPジャンパはこれらのデバイスの高電圧プログラミング使用時だけ装着されるべきです。他のデバイスの高電圧プログラミング、ISPプログラミング、デバッグの間、これらのジャンパは装着されるべきではありません。

PJUMPジャンパは水平方向、換言すると例えばBSEL2ジャンパと同じ方向のように装着されるべきです。

図3-32. PJUMPジャンパの配置



3.8.7 ATtiny26/261/461/861 高電圧並列プログラミング

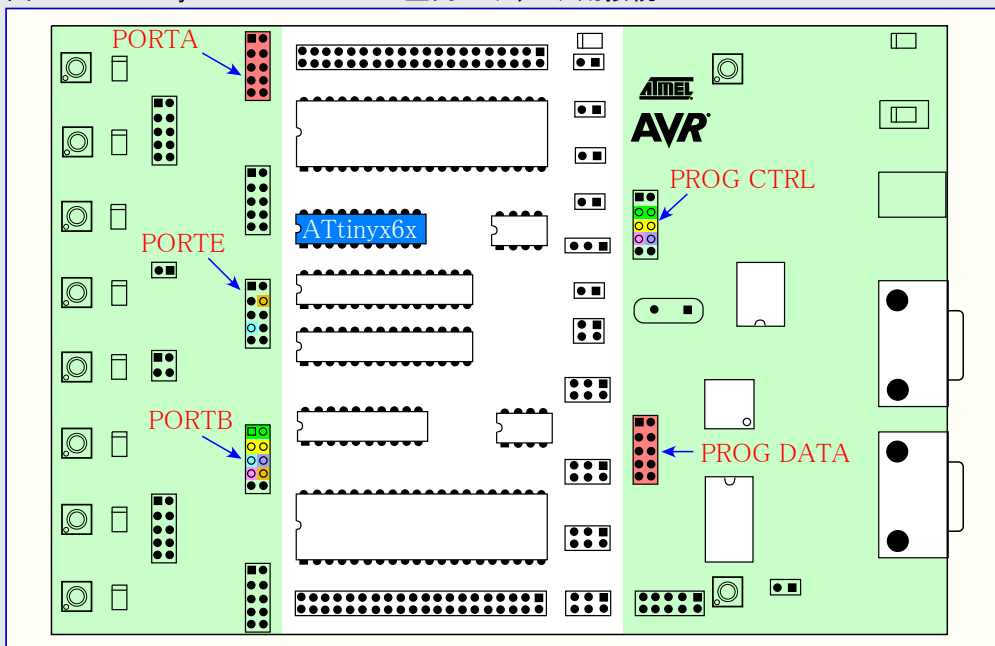
STK500には並列プログラミング時に使うための2つのヘッダがあります。これら2つのヘッダは**PROG DATA**、**PROG CTRL**と記されています。ATtiny26/261/461/861用**PROG CTRL**ケーブルは表3-5と図3-32A.で示されるように変更されなければならない、**PROG DATA**からのケーブルが**PORTA**に接続されなければならない。

注: ATtiny26/261/461/861はより容易な結線のSTK505アダプタでも支援されます。

表3-5. ATtiny6x 並列プログラミング用ピン配置

PORTB	PROG CTRL	PORTE
1	3	
2	4	
3	5	
4	6	
5		7 (XT1)
6	8	
7	7	
8		4 (RST)

図3-32A. ATtiny26/261/461/861の並列プログラミング用接続



次の点に注意してください。

- **PROG DATA**はSTK500目的対象領域上の**PORTA**に接続されなければならない。
- **BSEL2**ジャンパを装着してはいけません。
- ATtiny26/261/461/861はSTK500上の**SCKT3700A1**ソケットに配置されなければならない。

3.8.8 ATmega406 高電圧並列プログラミング

ATmega406はSTK500のファームウェアだけによって支援されます。これはSTK500母基板または何れかのアダプタのどれもがATmega406ピン配置を支援しないことを意味します。そしてATmega406がISPインターフェースを持たないため、高電圧並列プログラミングだけを支援します。ATmega406用の主プログラミングインターフェースはJTAGICEmk IIによって支援されるJTAGです。ATmega406の高電圧並列プログラミングはATmega406上のヒューズによってJTAGインターフェースが禁止された場合に使用されることの意味だけです。

本節はデバイスに高電圧並列プログラミングができ得るにはSTK500からどのピンがATmega406へ結線されなければならないかを記述します。

STK500上の以下のジャンパが装着されなければならない。

- **VTARGET**
- **RESET**
- **XTAL1**
- **OSCSEL** (1番ピンと2番ピン間)
- **BSEL2**

表3-6. ATmega406の並列プログラミング用結線

STK500	ATmega406
PROG DATA [0~7]	PORTB [0~7]
PROG CTRL [1~7]	PORTA [1~7]
PA0 (PORTA)	PA0
RESET (PORTE/AUX)	RESET
XTAL1 (PORTE/AUX)	XTAL1
VTG (4.5~5.5V)	VCC
GND	GND

3.9 拡張コネクタ

STK500にはプログラミング部両端に2つの拡張コネクタがあります。全AVRのI/Oポート、制御信号、プログラミング信号が拡張コネクタに配線されています。拡張コネクタはSTK500との応用試作を容易にさせます。拡張コネクタのピン配置は図3-34と図3-35で示されます。

図3-33. 拡張ヘッダ

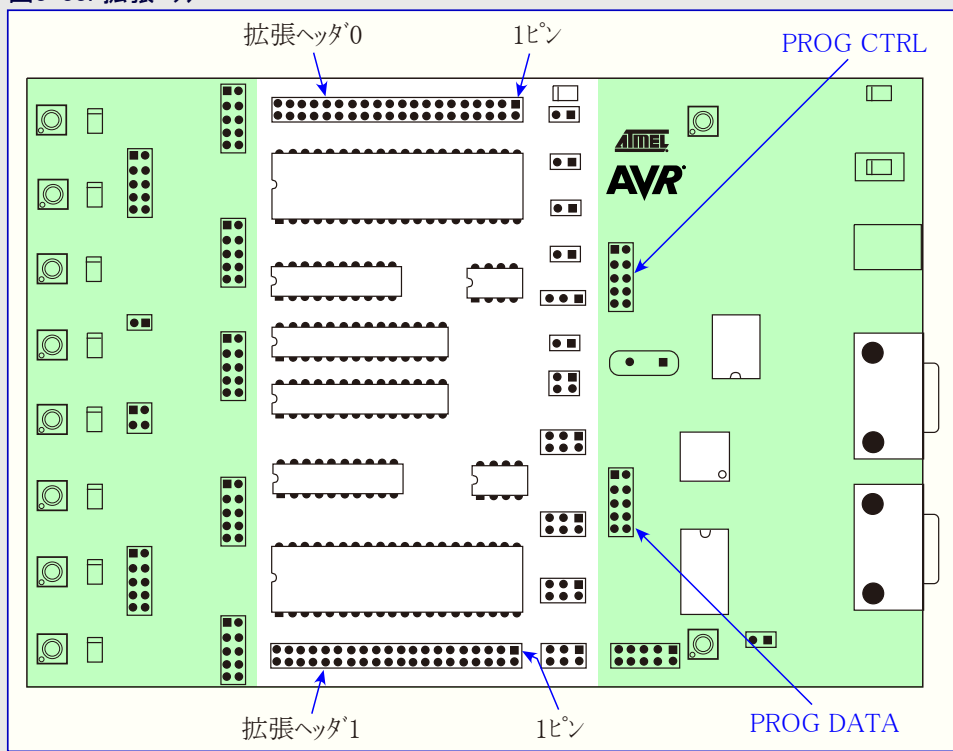


図3-34. 拡張コネクタ0 ピン配置

GND	1	2	GND
AUXI0	3	4	AUXO0
CT7	5	6	CT6
CT5	7	8	CT4
CT3	9	10	CT2
CT1	11	12	BSEL2
NC	13	14	REF
RST	15	16	PE2
PE1	17	18	PE0
GND	19	20	GND
VTG	21	22	VTG
PC7	23	24	PC6
PC5	25	26	PC4
PC3	27	28	PC2
PC1	29	30	PC0
PA7	31	32	PA6
PA5	33	34	PA4
PA3	35	36	PA2
PA1	37	38	PA0
GND	39	40	GND

図3-35. 拡張コネクタ1 ピン配置

GND	1	2	GND
AUXI1	3	4	AUXO1
DATA7	5	6	DATA6
DATA5	7	8	DATA4
DATA3	9	10	DATA2
DATA1	11	12	DATA0
SI	13	14	SO
SCK	15	16	CS
XT1	17	18	XT2
VTG	19	20	VTG
GND	21	22	GND
PB7	23	24	PB6
PB5	25	26	PB4
PB3	27	28	PB2
PB1	29	30	PB0
PD7	31	32	PD6
PD5	33	34	PD4
PD3	35	36	PD2
PD1	37	38	PD0
GND	39	40	GND

3.9.1 信号概要

AUXIO, AUXI1, AUXO0, AUXO1信号は将来使う予定です。応用のために、これらの信号へ接続してはいけません。

DATA7～0とCT7～0信号は**PROG DATA**と**PROG CTRL**ヘッダでも得られます。これらの信号とコネクタは3-21頁の3.10項で説明されます。

BSEL2信号は**BSEL2**ジャンパで得られるものと同じです。このジャンパは3-18頁の3.8.5節で説明されます。

SI, SO, SCK, $\overline{CS}$ 信号はDataFlashに接続されています。DataFlashの使用は3-4頁の3.6節で示されます。

NCはこのピンが接続されていないことを意味します。

残りの信号は3-3頁の3.4項で説明されるPORTxヘッダで得られるものと同じです。

注: DATA, CT, AUX信号は5V CMOSロジックに基準が置かれています。**VTG**に適合させるための電圧変換が、これらの信号では行われません。

3.10 PROG CTRL
PROG DATA
ヘッダ

PROG CTRLと**PROG DATA**ヘッダは目的AVRデバイスの高電圧並列プログラミングで使われます。ヘッダ位置は図3-33.で示されます。高電圧並列プログラミング中、**PROG CTRL**は目的デバイスの**PORTD**に配線されます。**PROG DATA**は**PORTB**に配線されます。高電圧プログラミングの完全な記述については3-9頁の3.7.2節を参照してください。**PROG CTRL**と**PROG DATA**のピン配置は図3-36.と図3-37.で示されます。AVRデバイスの高電圧プログラミングの多くの情報については各AVRデータシートのプログラミング章を参照してください。

注: **PROG CTRL**と**PROG DATA**コネクタの信号はレベル変換器なしで直接主マイクロコントローラに接続されます。これはこれらの信号が常に5Vロジックであることを意味します。

通常、**PROG CTRL**信号はAVRデバイスの高電圧並列プログラミング時の制御信号として使われます。

注: 全ての**PROG CTRL**信号は5V CMOSロジックに基準が置かれています。**VTG**に適合させるための電圧変換が、これらの信号では行われません。

PROG DATA信号はAVRデバイスの高電圧並列プログラミング時のデータバスとして使われます。ISPプログラミング中、DATA5はMOSI、DATA6はMISO、DATA 7はSCKとして使われます。

注: 全ての**PROG DATA**信号は5V CMOSロジックに基準が置かれています。**VTG**に適合させるための電圧変換が、これらの信号では行われません。

図3-36. PROG CTRLヘッダのピン配置

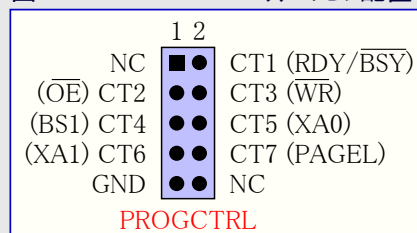
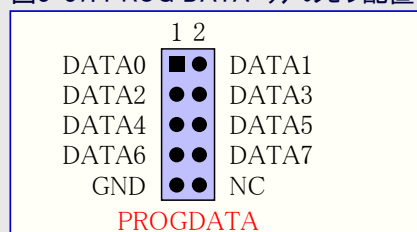


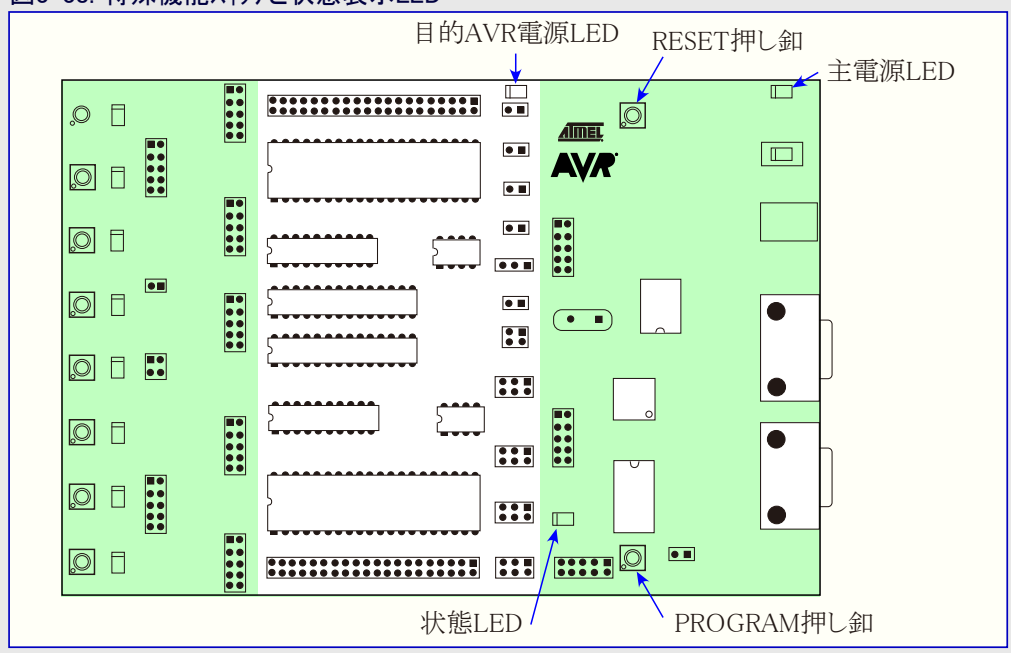
図3-37. PROG DATAヘッダのピン配置



3.11 その他スイッチ,LED

STK500には特殊機能と状態表示用に2つの押し釦と3つのLEDがあります。後続項はこれらの機能を説明します。図3-38.はこれら機能の配置を示します。

図3-38. 特殊機能スイッチと状態表示LED



3.11.1 RESET押し釦

RESET押し釦は押されると、目的AVRデバイスをリセットします。主マイクロ コントローラはRESET押し釦によって制御されません。RESETジャンパ未装着時、RESET押し釦は禁止されます。

3.11.2 PROGRAM押し釦

AVR Studioの将来版はSTK500上の主マイクロ コントローラ プログラムを更新するかもしれません。AVR StudioはSTK500の旧版ソフトウェアを検出し、主マイクロ コントローラのプログラム用フラッシュ メモリを更新します。これを行うにはSTK500電源投入時、PROGRAM釦の押下が必要とされます。更新手続き中、AVR Studioは更新実行方法を発給します。

3.11.3 主電源LED

この赤色電源LEDはSTK500の主電源に直接接続されます。この電源LEDはSTK500に電力が供給されているとき、常に点灯します。

3.11.4 目的AVRデバイス 電源LED

この目的電源LEDはソケット内の目的AVRデバイスのVCC信号線(VTG)に接続されています。この目的電源LEDは目的AVRデバイスに電力が供給されているとき、常に点灯します。

3.11.5 状態LED

状態LEDは3色LEDです。書き込み中、このLEDは黄色です。目的AVRデバイスが首尾よく書かれると、このLEDは緑に変わります。書き込み失敗時、書き込み失敗を示すためにこのLEDは赤に変わります。書き込み失敗時、7-1頁の「第7章 障害対策の指針」を照合してください。起動中、この状態LEDは赤から変化し、黄色を経由して、主マイクロ コントローラの準備完了を示すための緑に変化します。



AVR Studioのインストール

AVR Studio 統合開発環境(IDE)は全てのAVR開発用の理想的なソフトウェアです。それにはエディタ、アセンブラ、デバッガと、STK500スタートキットやAVRエミュレータの前処理部があります。

AVR Studioをインストールするには供給されたAtmelの参考資料CD-ROMをコンピュータに挿入してProducts⇒AVR 8-bit RISC⇒Softwareの順で選択します。AVRstudio.exeファイル上でマウスを右クリックし、ファイルに保存を選択します。任意の空フォルダを選択し、ファイルを保存します。

保存したAVRstudio.exeファイルを実行します。このファイルは自己解凍ファイルで、必要とされる全ファイルが現在のフォルダに解凍されます。Setup.exeファイルを実行します。これは構成手続きを通して指示がでます。

注: STK500を支援するにはAVR Studioの3.2版以降が必要とされます。

5.1 AVR Studio

この章ではSTK500支援ソフトウェアが示され、利用可能なプログラミング機能の詳細な記述が与えられます。

5.2 AVR Studioの起動

STK500開発基板との通信に使われるソフトウェアは3.2版以降のAVR Studioに含まれます。このソフトウェア(AVR Studio)のインストール方法の情報については4-1頁の第4章を参照してください。一旦インストールされると、AVR StudioアイコンのダブルクリックによってAVR Studioが開始できます。既定インストールが使われる場合、本プログラムは**スタート⇒プログラム⇒Atmel AVR Tools**フォルダに配置されます。

5.2.1 STK500の始動

AVR Studio ツールバーの **Connect** 鈕を押下が **図5-1**で示されるAVR書き込み器選択(Select AVR Programmer)ダイアログを開始します。

図5-1. AVR書き込み器選択ダイアログ

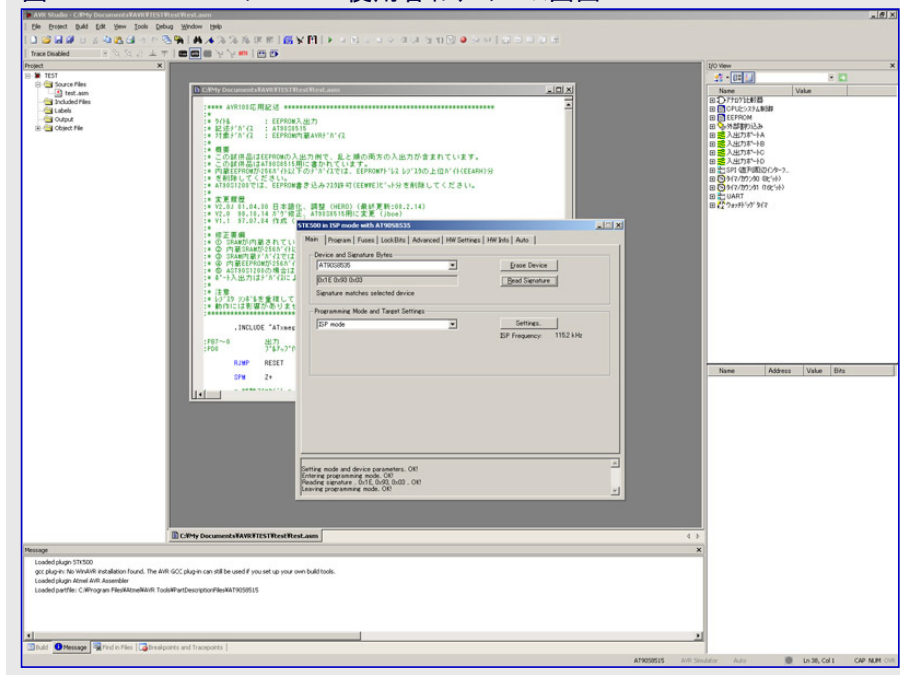


Platformと**Port**を選択して、**Connect...**鈕を押下してください。プログラミング前処理ソフトウェアが数秒後に現れるでしょう(図5-2)。

AVR書き込み器選択(Select AVR Programmer)ダイアログが最後に開かれていた時に選択されていた基盤とポートへ直接的に接続するにはツールバーの **AVR** 鈕を押下してください。

選択した基盤が検出できなかった場合、数秒後にAVR書き込み器選択(Select AVR Programmer)ダイアログが再び現れるでしょう。これが起きたなら、RS232またはUSBケーブルが正しく接続されているのと、ツールが既に同じまたは別のAVR Studioでデバッグまたはプログラミング作業に接続されていないことを確認してください。

図5-2. AVR StudioのSTK500使用者インターフェース画面



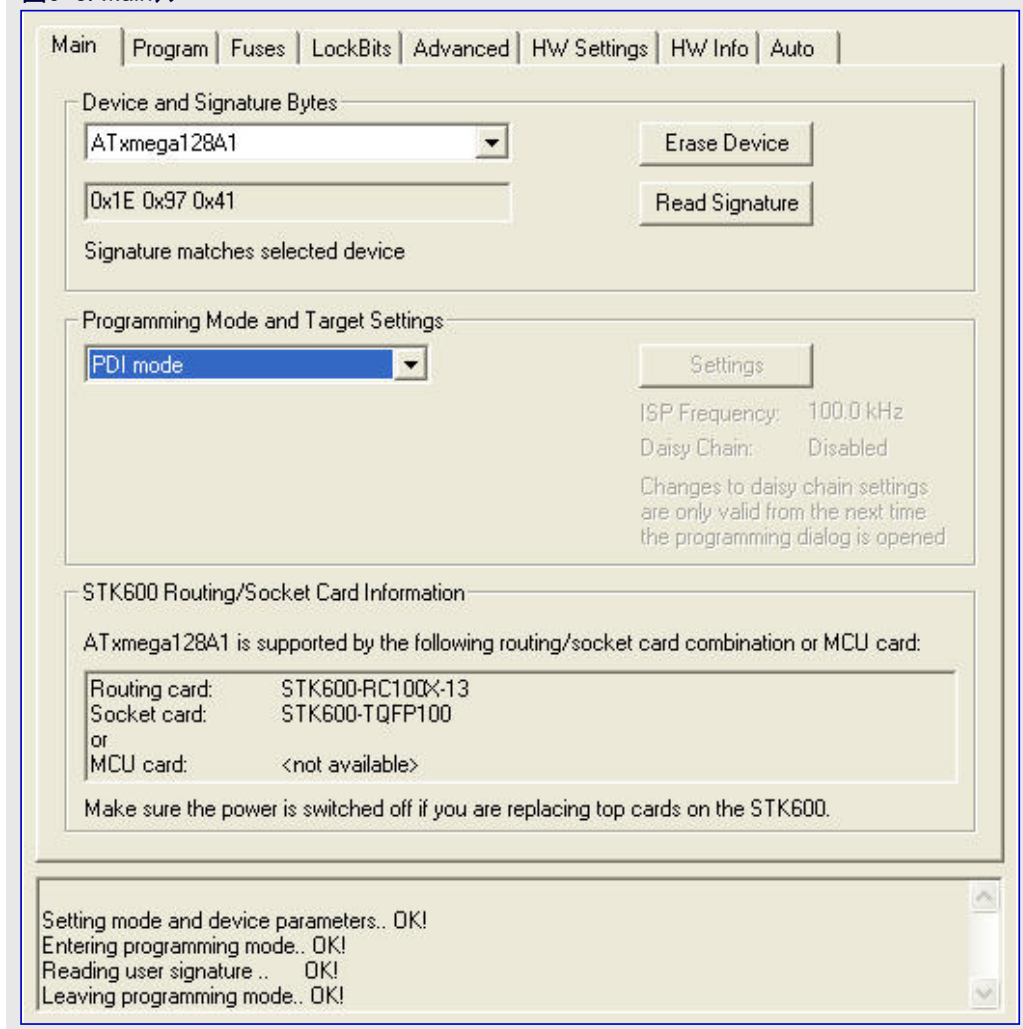
5.3 STK500 使用者インターフェース

STK500使用者インターフェースはSTK500開発基板用の強力な機能を含みます。利用可能な設定は8つの群に分けられ、適切なタブのクリックによって各々選択可能です。異なるデバイスには異なる機能を持つため、利用可能な機能と選択はどのデバイスが選択されるかに依存します。利用不能の機能は灰色表示になります。

5.3.1 Mainタブ

デバイスとプログラミング動作種別がこのMainタブで選択されます。このタブはデバイス消去と識票読み込み用の釦も含みます。

図5-3. Mainタブ



5.3.1.1 Device and Signature Bytes デバイス選択と識票バイト

どんなプログラミング操作にも先立って、正しいデバイスとプログラミングインターフェースが設定されなければなりません。

デバイスコンボイでデバイスを選択してください。これは正しいプログラミング方法がそのデバイスに使われるのと、デバイスに対して利用可能な機能だけがダイアログで表示されることを確かめます。次に以下の項で記述されるようにプログラミングインターフェースを指定してください。

デバイスの識票を読むにはRead Signature釦を押下してください。このダイアログは識票が選択デバイスと一致するかを検査します。識票バイトについての多くを理解するにはAVRのデータシートを参照してください。

完全なチップ消去はErase Device釦の押下によって実行されます。これはフラッシュメモリ、(EESAVEヒューズのプログラム(0時を除き))EEPROM、施錠ビットを含め、接続されたデバイスの全体の内容を消去します。

5.3.1.2 Programing Mode and Target Settings

プログラミング種別と 目的対象調整設定

選択したインターフェース経由での目的デバイスへのツール接続方法は基盤固有のヘルプで記述されます。プログラミング前処理ソフトウェアはそれに応じて初期設定されなければなりません。現在利用可能なプログラミングインターフェースは以下です。

- ISP (In-System Programming:実装書き込み)
- PP/HVSP (高電圧並列プログラミングまたは高電圧直列プログラミング)
- JTAG (注: STK500を他のツールまたはアダプタと組み合わせた場合)
- PDI (XMEGAシステム専用) (注: STK500を他のツールまたはアダプタと組み合わせた場合)

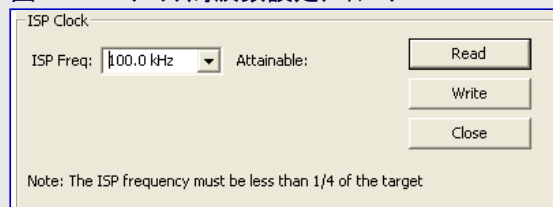
ISPまたはJTAGのどちらかが選択された場合、Settings鈕のクリックによってそのインターフェースに関する更なる設定が指定できます。PP/HVSPインターフェースに対する更なる設定はありません。

5.3.1.2.1 ISP動作

ISP動作でのSettings鈕押下はISPクロック周波数設定ダイアログを引き出します。

ISPクロック周波数(SCK)は目的クロックの1/4よりも低くなければなりません。然しながら、これは目的クロックとISPクロックの両方で50%/50%デューティ比が必要です。目的クロックの1/5よりも低いISP周波数の使用が推奨されます。

図5-4. ISPクロック周波数設定ダイアログ



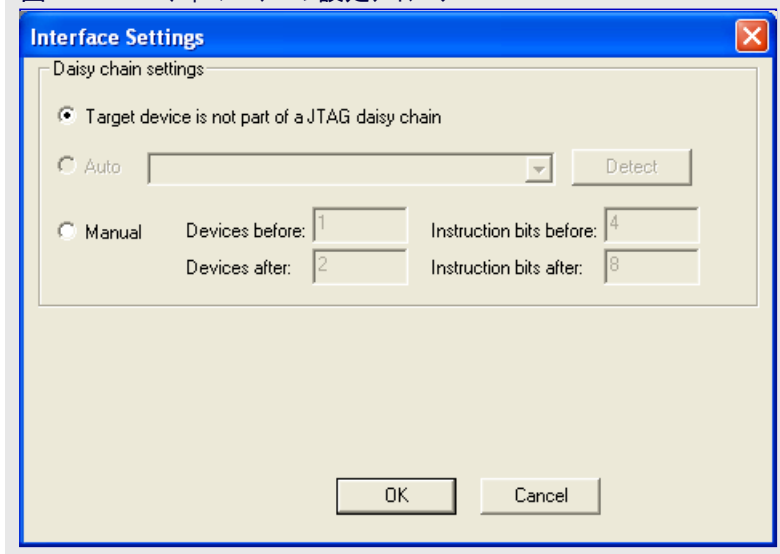
注: いくつかのAVRデバイスはCKDIV8ヒューズを持ちます。このヒューズがプログラム(0)の時にAVRコアは分周されたクロックで走行します。ISP周波数がこのクロックに応じて正しく設定されているのを確認してください。例えばCKDIV8(8分周)ヒューズがプログラム(0)されている場合、ISP周波数は目的クロック元の1/32(1/40推奨)よりも低くなければなりません。

5.3.1.2.2 JTAG動作

JTAG動作でのSettings鈕押下はJTAGデジーチェーン設定ダイアログを引き出します。

JTAGデジーチェーン設定がスタートキット/エミュレータの初期設定の一部のため、これらのパラメータに対する効果を表すにはプログラミングダイアログを閉じて再び開かなければならないことに注意してください。

図5-5. JTAGデジーチェーン設定ダイアログ

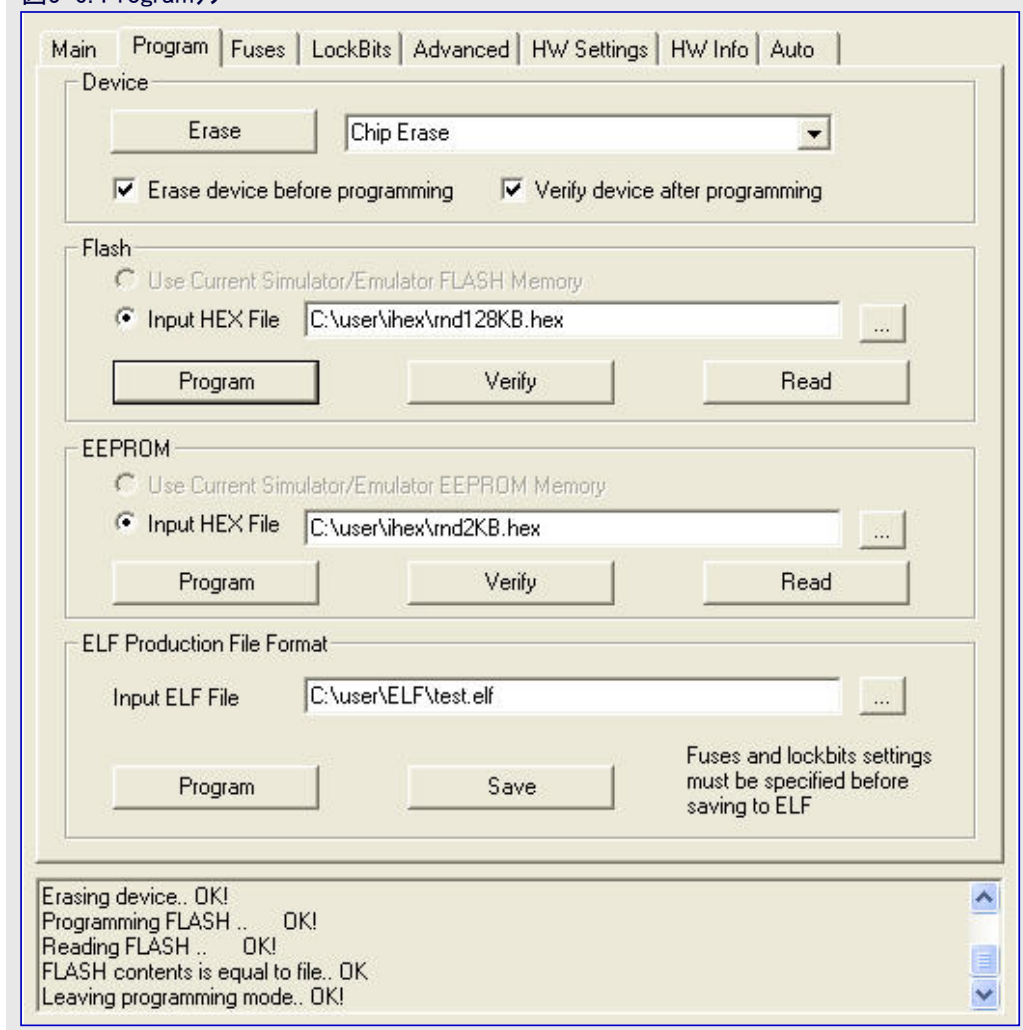


- “目的対象はJTAGデジーチェーンの一部ではありません。(Target is not part of a JTAG daisy chain)” : 1つの目的デバイスだけを使う時にこの任意選択を選んでください。
- “自動(Auto)” : JTAGデジーチェーンに付属しているデバイスを検出するには検出(Detect)鈕を押下してください。通信するデバイスを引き落とし一覧から選択してください。これを正しく動かすためには全てのデバイスが8ビットのAVRデバイスでなければならないことに注意してください。
- “手動(Manual)” : デジーチェーン設定を手動で指定するにはこの設定を使ってください。

5.3.2 Programタブ

このProgramタブはチップ消去の実行だけでなくメモリプログラミングも許します。XMEGA系統に対して新しい消去機能が追加されています。[XMEGA消去種別](#)をご覧ください。

図5-6. Programタブ

5.3.2.1 Device
デバイス操作

完全なチップ消去はErase Device釦の押下によって実行されます。チェック枠はその後のプログラミングに関する以下の任意選択、メモリプログラミング操作前のチップ消去実行(Erase device before programming)とメモリプログラミング後の自動検証(Verify device after programming)を指定します。

5.3.2.2 Flash
フラッシュプログラミング

適切な釦の押下により、接続デバイスのフラッシュメモリ内容のプログラミング(書き込み)、検証、読み出しができます。フラッシュメモリが書かれる前に入力ファイルが指定されていなければなりません。元となるコードが独立したHEXファイル内に格納されている場合、Input HEX File任意選択を選んでください。テキスト領域内に完全なパスとファイル名を書くか、または  釦押下によって正しいファイルを開覧/選択してください。選択したファイルは"Intel HEX"形式または"拡張Intel HEX"形式でなければなりません。

プロジェクトがAVR Studioに読まれるとき、(メモリウィンドウ内に表示されるのと同じ、)デバッガ内のフラッシュメモリ内容を接続されたデバイスに転送することができます。これを行うにはUse Current Simulator/Emulator FLASH Memory任意選択を選択してください。

5.3.2.3 EEPROM
EEPROMプログラミング

EEPROMプログラミングに関する任意選択はフラッシュメモリと同じです。

5.3.2.4 ELF Production File Format

ELF提出ファイル形式

この提出ファイル形式はフラッシュメモリとEEPROM両方の内容だけでなく、ヒューズと施錠ビット設定も1つの単独ファイルに保持することができます。この形式はELF(Executable and Linkable Format:実行/リンク可能形式)に基づいています。

このELFファイル生成に先立って、フラッシュメモリとEEPROM内容が指定され、ヒューズと施錠ビットが設定されていなければなりません。ProgramタブのFLASHとEEPROMの入力ファイルを指定してください。そしてFusesタブへ行き、望むヒューズビット設定を指定してください。LockBitsタブでも同様に行ってください。Programタブに戻り、ELFファイルを生成するSave鈕を押下してください。

生成したELFファイルはこのプログラミングダイアログへの入力として指定でき、故にファイルの全内容が1操作でプログラミングされます。Input ELF Fileエディット枠でファイル名を指定するか、またはそのファイルを開覧/選択し、その後Program鈕を押下してください。

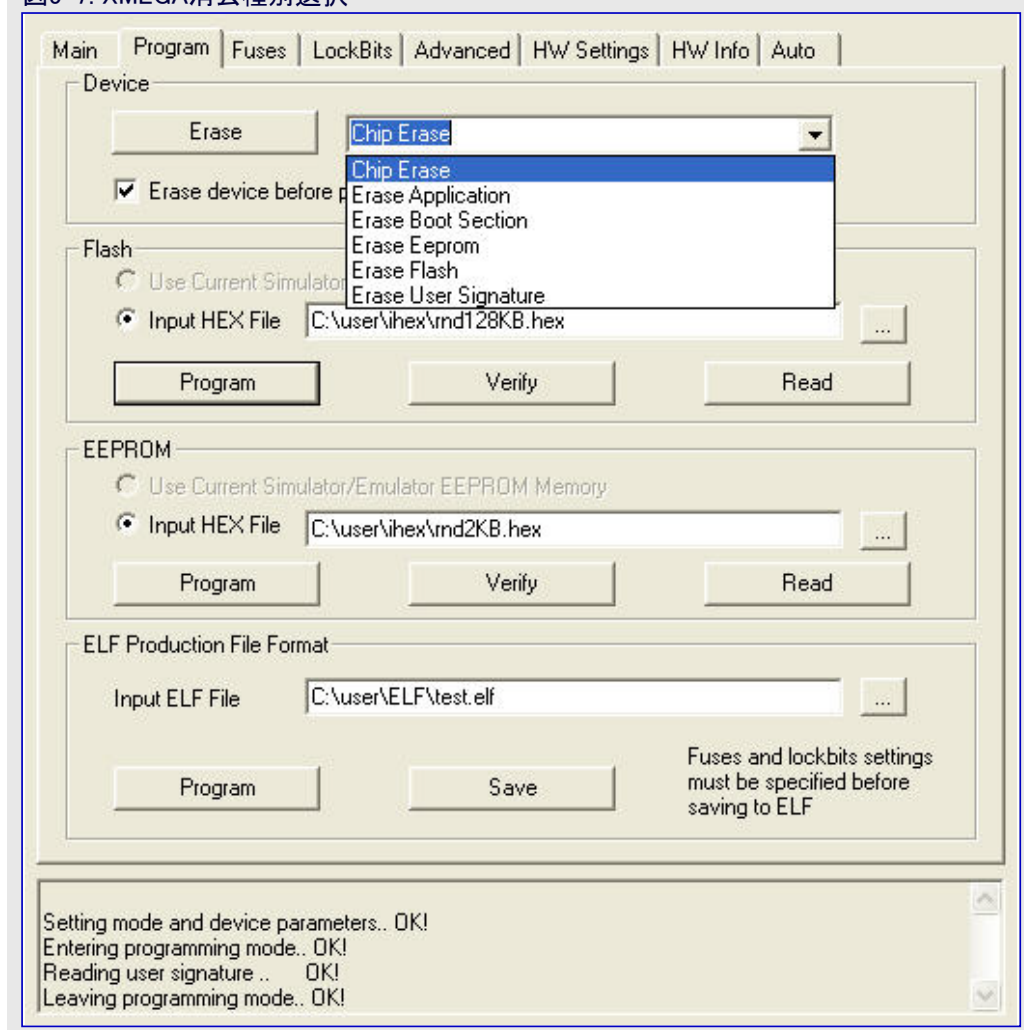
このELFファイルはSTK600用のコマンド行ツールへの入力としても使えます。他の基盤用のコマンド行ツールは将来支援されるでしょう。(訳補:従って現状のSTK500では支援されません。)

5.3.2.5 XMEGA消去種別

XMEGAデバイスシステム使用時に新しい消去機能が利用可能なことに注意してください。新しい消去機能は各種消去任意選択を使用者に許します。以下の任意選択が利用可能です。

- ・チップ消去(Erase Chip)
- ・応用領域消去(Erase Application)
- ・ブート領域消去(Erase Boot Section)
- ・EEPROM消去(Erase Eeprom)
- ・フラッシュ消去(Erase Flash)
- ・使用者識票消去(Erase User Signature)

図5-7. XMEGA消去種別選択



5.3.3 Fusesタブ

Fusesタブは選択したデバイスのヒューズを表します。

ヒューズの現在値を読むにはRead鈕を、現在のヒューズ設定をデバイスに書くにはWrite鈕を押下してください。ヒューズ設定状態はチェック枠または引き落とし一覧として表されます。

各プログラミング動作種別でどのヒューズが利用可能かと、それらの機能の詳細情報はデバイスのデータシートで得られます。選択したヒューズ設定がチップ消去区間を伴うデバイス消去(例えばProgramタブでのChip Erase押下)によって影響されないことに注意してください。

ヒューズ値は下の欄で拡張、上位、下位ヒューズへ直接的にも書けます。

下部のチェック枠は、Fusesタブが表示される時にデバイス内のヒューズが自動的に読まれるべきかどうか(Auto read)、使用者が潜在的に危険なヒューズ設定を選択した場合に警告が表示されるべきかどうか(Smart warnings)、書き込み後にヒューズが自動的に検証されるべきかどうか(Verify after programming)を指定します。

図5-8. Fusesタブ

The screenshot shows the 'Fuses' tab in AVR Studio. It contains two tables of fuse settings, checkboxes for 'Auto read', 'Smart warnings', and 'Verify after programming', and buttons for 'Program', 'Verify', and 'Read'. At the bottom is a status log window.

Parameter	Value
JTAGUSERID	0xFF
WDWFP	32 cycles (32ms @ 3.3V)
WDP	32 cycles (32ms @ 3.3V)
DVSDON	☒
BOOTRST	Application Reset
BODACT	BOD Disabled
BODPD	BOD Disabled
SUT	0 ms
WDLOCK	☐
JTAGEN	☒
EESAVE	☒
BODLVL	1.6 V

FUSEBYTE0	0xFF
FUSEBYTE1	0x22
FUSEBYTE2	0x7F
FUSEBYTE4	0xFE
FUSEBYTE5	0xF7

☒ Auto read
☐ Smart warnings
☒ Verify after programming

Program Verify Read

Leaving programming mode.. OK!
 Entering programming mode.. OK!
 Reading fuses (low to high).. 0xFF, 0x22, 0x7F, 0xFF, 0xFE, 0xF7 .. OK!
 Fuse bits verification.. OK
 Leaving programming mode.. OK!

いくつかのヒューズは高電圧プログラミング動作でだけ利用可能です。これらのヒューズは表示されますが、MainタブでISPプログラミング動作が選択されている場合はアクセス(操作)することができません。また、いくつかのデバイスはISPまたは高電圧プログラミング動作を支援していません。チェック枠の外見はヒューズが活動中のプログラミング種別で読み書き可能かどうかを表現します。

- ☐ ☒ 活動中のプログラミング種別で読み書き可能なヒューズビット(標準)
- ☐ ☒ 活動中のプログラミング種別で読み込み可能/書き込み不能なヒューズビット
- ☐ ☒ 活動中のプログラミング種別で読み込み不能/書き込み可能なヒューズビット
- ☐ ☒ 活動中のプログラミング種別で読み書きどちらも不能なヒューズビット

ヒューズが活動中のプログラミング種別で読み込み不能の場合、ダイアログ内に表示された値は信用できません。

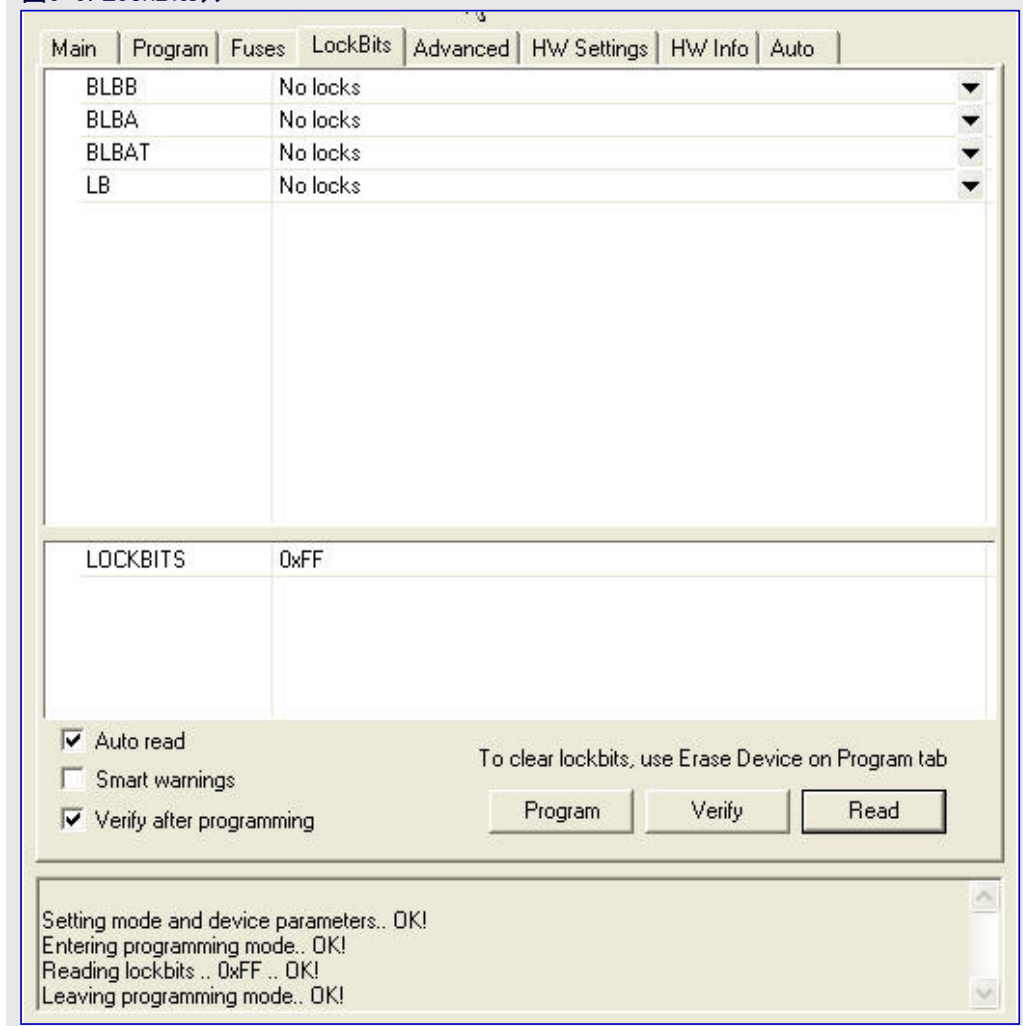
5.3.4 LockBitsタブ

Fusesタブ同様、LockBitsタブは選択デバイスにどの施錠種別が適用可能かを表示します。

施錠種別は複数の施錠ビット設定の組み合わせから成るかもしれません。これはSTK500使用者インターフェースによって操作され、選択した施錠種別に対して正しい施錠ビットが自動的に書かれます。一旦“施錠種別”保護段階が許可されると、保護の程度を低く選択したり、違う施錠種別を設定することによって保護段階を低くすることはできません。

プログラム(0)された施錠ビットを削除する方法は完全なチップ消去を実行することだけです。

図5-9. LockBitsタブ



ISPプログラミングを支援するデバイスについては全ての施錠ビットがISPと高電圧プログラミング動作種別で入出力できます。いくつかのデバイスがISPプログラミングまたは高電圧プログラミングのどちらかを支援しないことに注意してください。LockBitsタブの外見は施錠ビットが活動中のプログラミング種別で読み書き可能かどうかを表現します。

- ☐ ☒ 活動中のプログラミング種別で読み書き可能な施錠ビット
- ☐ ☒ 活動中のプログラミング種別で読み込み可能/書き込み不能な施錠ビット
- ☐ ☒ 活動中のプログラミング種別で読み込み不能/書き込み可能な施錠ビット
- ☐ ☒ 活動中のプログラミング種別で読み書きどちらも不能な施錠ビット

施錠ビットが活動中のプログラミング種別で読み込み不能の場合、ダイアログ内に表示された値は信用できません。

5.3.5 Advancedタブ

Advancedタブはデバイス系統に基づいて2つの見方を持ち得ることに注意してください。旧来のタブはtinyとmegaデバイスを支援し、一方xmegaは自身のタブを持ちます。

5.3.5.1 Oscillator Calibration Byte

発振校正値

(非XMEGA用)

発振校正値は可能な限り選択したクロック周波数近くでの動作に内蔵RC発振器を調整するために、選択デバイスに基づいたOSCCALレジスタへ書かれ得る値です。

発振校正値は製造中にデバイスへ格納され、使用者による消去や変更ができません。これはデバイスに依存して、デバイス起動中にOSCCALレジスタへ自動的に転送されるか、またはプログラムの初期化中に設定するかです。プログラムの初期化中に応用が設定するデバイスでは、プログラミングダイアログまたはコマンド行ツールを使って、先にこの値をフラッシュメモリかEEPROMに転送しておかなければなりません。

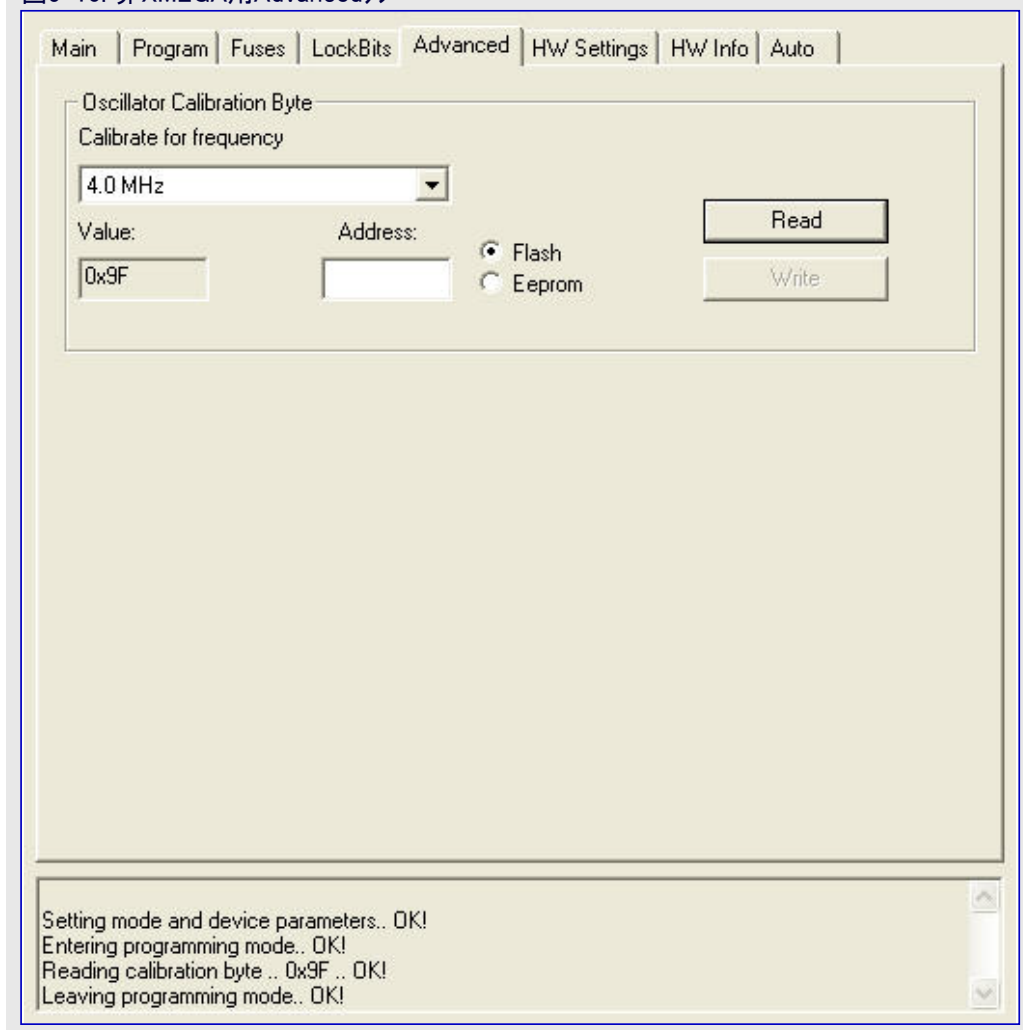
5.3.5.2 発振校正値の読み書き

(非XMEGA用)

Read釦押下により、校正値がデバイス内の記憶域から読まれ、Value:テキスト枠に表示されます。この任意機能が灰色表示の場合、選択デバイスには調整可能な内蔵RC発振器がありません。

Write釦押下によって校正値がフラッシュメモリまたはEEPROM内に書かれます。先にメモリ形式(種別)とアドレスが指定されなければなりません。

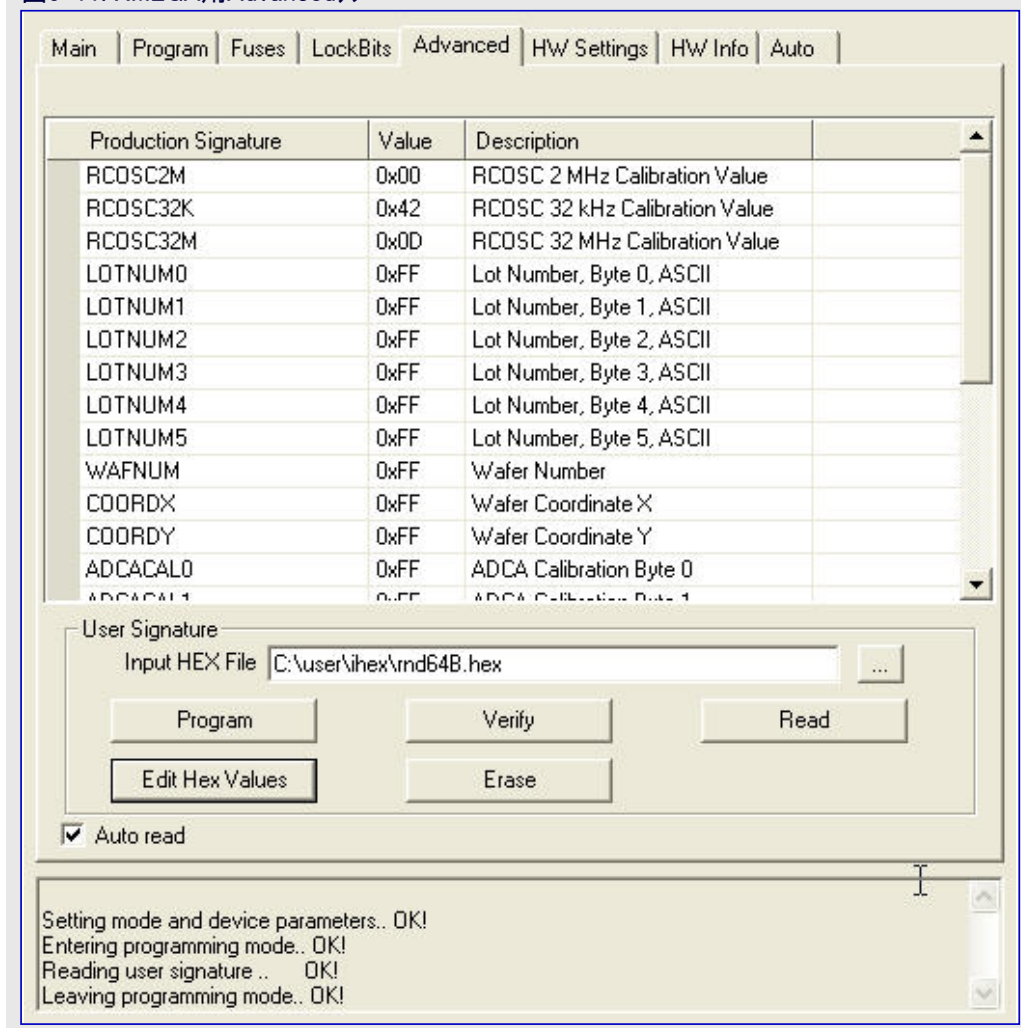
図5-10. 非XMEGA用Advancedタブ



5.3.5.3 XMEGA用Advancedタブ XMEGAデバイス使用時、Advancedタブは下のタブに変わります。製品識票(Production Signature)と使用者識票(User Signature)と呼ばれる2つの新しいメモリ形式が現れます。

5.3.5.4 Production Signature 製品識票 XMEGAの製造識票はデバイスの製造に関するデバイス特有情報を含みます。例えば、ウェハ番号、ウェア座標、ロット番号、その他諸々です。下図をご覧ください。

図5-11. XMEGA用Advancedタブ



5.3.5.5 User Signature 使用者識票 XMEGA系統は使用者定義値用のメモリ形式を含みます。このメモリ形式は例えば校正値の格納に使うことができます。このメモリ形式の容量は1つのフラッシュページと同じです。一般的に512バイトです。

使用者識票メモリ空間は2つの方法で編集することができます。入力としてHEXファイルを使うことによって、または16進エディタでメモリを直接編集すること(Edit Hex Values 鈕)によって行えます。16進エディタについては次の図をご覧ください。

図5-10. 非XMEGA用Advancedダンプ

0000	50 52 4F 47 52 41 4D 4D 49 4E 47 FF FF FF FF FF	PROGRAMMING.....
0010	44 49 41 4C 4F 47 20 4F 4E 4C 49 4E 45 20 2E 2E	DIALOG ONLINE ..
0020	48 45 4C 50 20 53 43 52 45 45 4E 53 48 4F 4F 54	HELP SCREENSHOOT
0030	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0040	55 53 45 52 20 53 49 47 4E 41 54 55 52 45 FF FF	USER SIGNATURE..
0050	48 45 58 20 45 44 49 54 4F 52 FF FF FF FF FF FF	HEX EDITOR.....
0060	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0070	46 4F 52 20 58 4D 45 47 41 20 4F 4E 4C 59 FF FF	FOR XMEGA ONLY..
0080	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0090	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00A0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00B0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00C0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00D0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00E0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
00F0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0100	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0110	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0120	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0130	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0140	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0150	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0160	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0170	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0180	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
0190	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01A0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01B0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01C0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01D0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01E0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
01F0	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	

Read Write Erase OK Cancel

5.3.6 HW Settingsタブ

HW SettingsタブはSTK500開発基板上の動作条件を表示します。

注: STK500に関してはデバイスに対する推奨または最大仕様を超えて動作条件を強制することができます。これを行うことは推奨されず、目的デバイスに損傷を与えるかもしれません。デバイスに対する推奨動作条件はデバイスのデータシートで得られます。

5.3.6.1 Voltages 電圧

VTarget設定は目的基板に対する動作電圧を制御します。STK500でのこの電圧は0.1V単位で0～6.0V間に調整することができます。STK500とSTK600以外の他のツールに対するVTargetは読み込み専用です。

AREF設定はA/D変換器用のアナログ基準電圧を制御します。この設定はA/D変換器付きのデバイスだけに適用されます。STK500でのAREFは0.1V単位で0～6.0V間に調整することができます。選択デバイスに対する有効電圧範囲についての情報に関してはデバイスのデータシートを参照してください。

AVRに恒久的な損傷を与えるため、**AREF**を**VTarget**よりも高く設定することはできません。

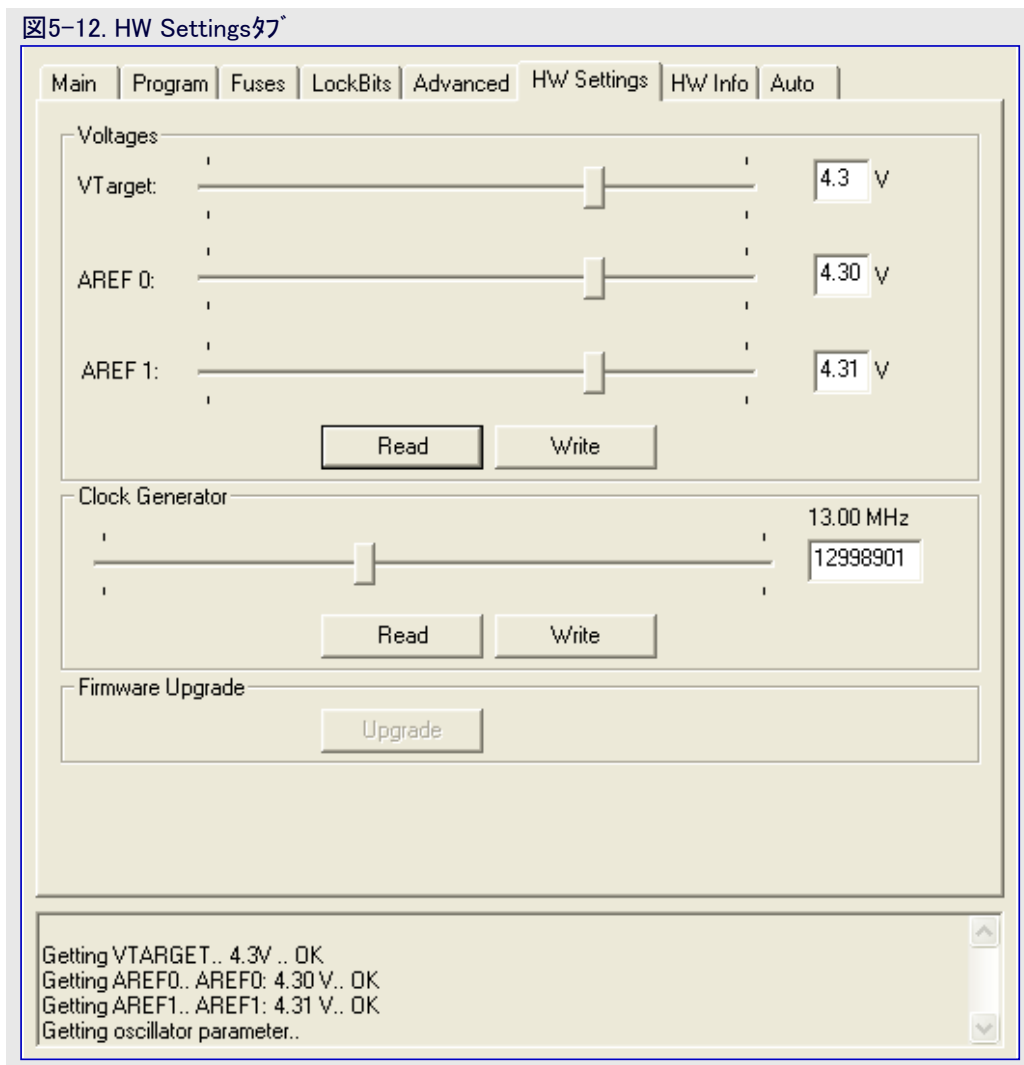
全ての電圧は**Read**釐押下によって読まれ、**Write**釐押下によって書かれ(目的対象基盤に設定され)ます。

VTarget電圧の物理的な接続は3-13頁の図3-22で、AREF電圧の物理的な接続は3-14頁の図3-24で示されます。

5.3.6.2 Clock Generator クロック発生器

Clock Generator設定はSTK500のXTAL1に生成する目的クロックの表示と制御を行います。この設定はSTK500とSTK600以外の他のツールに対して利用できません。

図5-12. HW Settingsタブ



5.3.7 HW Infoタブ

HW Infoタブは接続されたツールのハードウェアとファームウェアの改訂(Rev)番号を表示します。STK600に対しては導入されている配線カードとソケットカードについての詳細情報を表示します。現在配置されている配線カードとソケットカードによって支援されるデバイスが引き落とし一覧で一覧され、配線カード、ソケットカード、拡張カードがそれらのカードで許される最大動作電圧と共に表示されます。

図5-13. HW Infoタブ

The screenshot shows the HW Info tab in AVR Studio. The tab is titled 'HW Info' and is part of a larger window. The window has a menu bar with 'Main', 'Program', 'Fuses', 'LockBits', 'Advanced', 'HW Settings', 'HW Info', and 'Auto'. The HW Info tab is active, showing the following information:

- Hardware and Firmware Revision**
 - Hardware Revision: 0x02
 - Firmware Version: 0x0203 0x0200 0x0200
- Supported Devices by Current Routing/Socket Card**
 - ATxmega64A1
- Routing, Socket and Expansion Card**

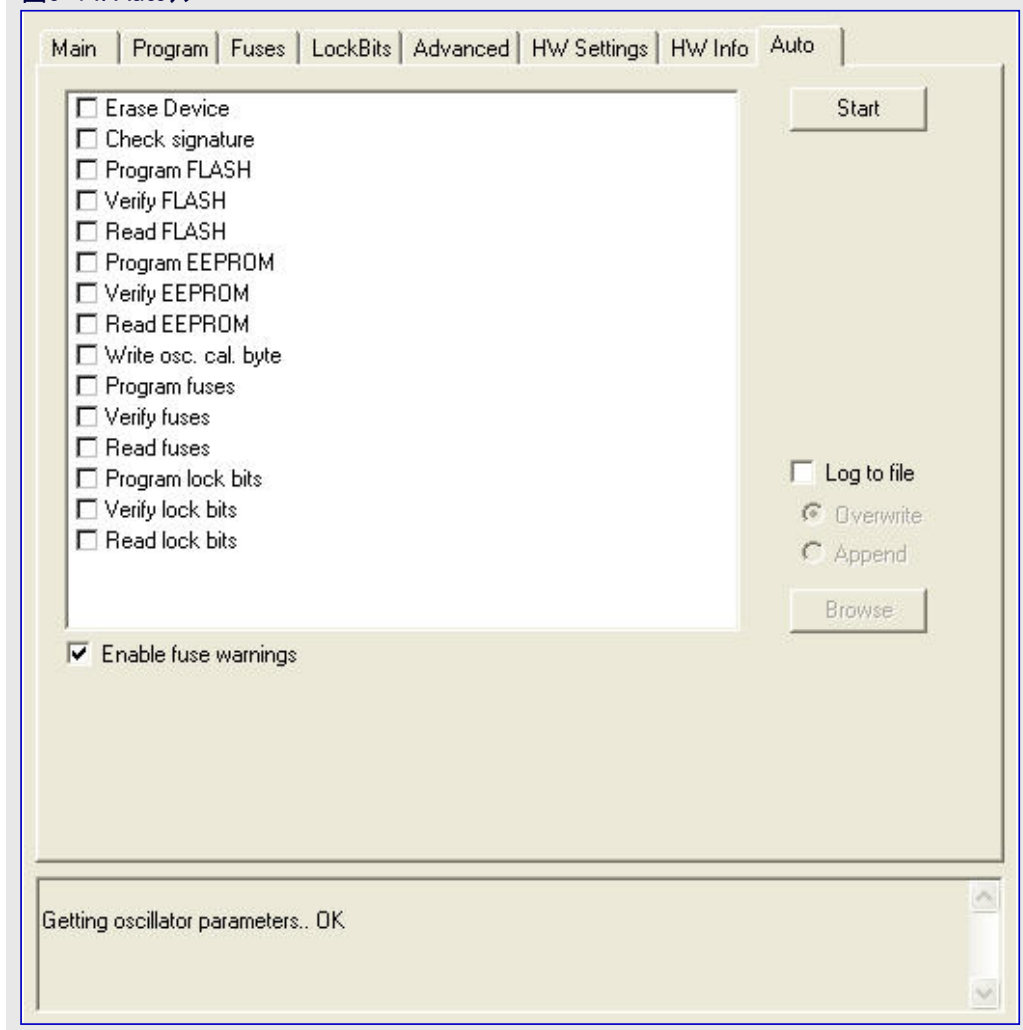
Routing Card: STK600-RC100X-13	Max Voltage: 3.6 V
Socket Card: STK600-TQFP100	
Expansion Card: Not Connected	

At the bottom of the window, there is a status bar that says 'Getting oscillator parameters.. OK'.

5.3.8 Autoタブ

同じコードでの複数デバイスのプログラミング時、Autoタブが命令の使用者定義手順を通した自動実行の強力な方法を提供します。命令はそれらが(選択されていたなら)実行される順番で一覧されます。命令を許可するには適切なチェック枠がチェックされるべきです。例えば、**Program FLASH**だけがチェックされた場合、**Start** 釦押下によってProgramタブで指定したHEXファイルでフラッシュメモリが書かれるでしょう。

図5-14. Autoタブ



Log to file チェック枠のチェックによって命令実行をテキストファイルへ記録することができます。

5.3.8.1 自動プログラミング設定

使用者インターフェースの実行を望む命令に対するチェック枠をクリック(チェック)してください。代表的な手順は例えば1回でデバイスが消去され、そして書かれることでしょう。チップが消去され、(フラッシュメモリとEEPROMの)両方が書かれて照合され、そして最後にヒューズと施錠ビットがプログラミングされます。

一度設定されると、**Start** 釦が押下されると何時でも同じプログラミング手順が実行されます。これは操作誤りのための異常に関する可能性とその作業の両方を低減します。

5.3.8.2 自動プログラミング記録ファイル

Log to file チェック枠のクリック(チェック)により、命令からの全出力がテキストファイルに書かれます。このファイルは **Browse** 釦の押下と、ファイルが置かれたまたは作成されるべきファイルの位置への誘導によって選択/作成されます。出力はこのファイルに書かれ、テキストエディタを使って閲覧や編集を行うことができます。



5.4 コマンド行ソフトウェア

STK500のコマンド行ソフトウェアは外部エディタからのSTK500プログラミングや、量産書き込み器での使用に有用です。簡単なバッチファイルが自動プログラミングを促進できます。ヘルプは**STK500 -h**と打ってください。

注: 引数を伴うコマンド識別子の何れについても、コマンド識別子と引数間にスペースがあるべきではありません。例えば、**-d AT90S8515**ではなく、**-dAT90S8515**とすべきです。

このプログラムは操作が成功だったならばERRORCODE 0を返し、操作が失敗だったならばERRORCODE 1を返します。

```
C:>STK500 [-d device] [-m s|p] [-ip infile] [-if infile] [-ie infile] [-of outfile]
          [-oe outfile] [-s] [-O address] [-Sf addr] [-Se addr] [-e]
          [-p a|f|e|b] [-r f|e|b] [-v a|f|e|b] [-l value] [-L value] [-y]
          [-f value] [-E value] [-F value] [-G value] [-q] [-Y] [-Z address]
          [-c port] [-ut value] [-Wt value] [-ua value] [-wt] [-wa] [-b h|s]
          [-! freq] [-t] [-I freq] [-J] [-h] [-?]
```

5.5 引数

- d** デバイス名。プログラミングするデバイス(後記参照)を指定します。
- m** プログラミング種別、直列(**s**)、並列(**p**)、PDI(**d**)設定。本引数省略時、直列動作が既定です。
- ip** ELF用入力ファイル名。全てのメモリ形式の書き込みと照合に必要です。ファイル形式はAVR-8ビットELFです。
- if** フラッシュまたは使用者識票用入力ファイル名。フラッシュメモリまたは使用者識票の書き込みと照合に必要です。ファイル形式は拡張インテル形式です。
- ie** EEPROM用入力ファイル名。EEPROMの書き込みと照合に必要です。ファイル形式は拡張インテル形式です。
- of** フラッシュまたは使用者/製品識票用出力ファイル名。フラッシュメモリまたは使用者/製品識票の読み出しに必要です。ファイル形式は拡張インテル形式です。
- oe** EEPROM用出力ファイル名。EEPROMの読み出しに必要です。ファイル形式は拡張インテル形式です。
- s** 識票バイトの読み込み。
- O** 発振校正値読み出し。**address**はデバイスのデータシートで記述された校正値バイトのアドレスです。
- Sf** フラッシュメモリに発振校正値書き込み。**addr**はバイト単位アドレスです。
- Se** EEPROMに発振校正値書き込み。**addr**はバイト単位アドレスです。
- e** デバイス消去。他の書き込み引数と共に与えられ、書き込み実行に先立って消去されます。
- p** メモリ書き込み。全メモリ形式(**a**)、フラッシュ(**f**)、EEPROM(**e**)、両方(**b**)、使用者識票(**u**)。対応する入力ファイルが必要です。
- r** メモリ読み込み。フラッシュ(**f**)、EEPROM(**e**)、両方(**b**)、使用者識票(**u**)、製品識票(**p**)。対応する出力ファイルが必要です。
- v** メモリ照合。全メモリ形式(**a**)、フラッシュ(**f**)、EEPROM(**e**)、両方(**b**)。-pと共に、または単独で使えます。対応する入力ファイルが必要です。
- l** 施錠ビット設定。**value**は8ビットの16進値です。
- L** 施錠ビット検証。**value**は比較照合用の8ビットの16進値です。
- y** 施錠ビット読み出し。
- f** ヒューズビット設定。**value**は上位/下位ヒューズ用の設定を記述する16ビットの16進値です。
- E** 拡張ヒューズビット設定。**value**は拡張ヒューズを記述する8ビットの16進値です。
- XE** xmegaヒューズバイト4,5設定。**value**は16ビットの16進値です。記述する値はxmegaの最後の2つのヒューズに設定します。
- F** ヒューズビット検証。**value**は比較照合用の16ビットの16進値です。
- XV** xmegaヒューズバイト4,5検証。**value**は比較照合用の16ビットの16進値です。
- G** 拡張ヒューズビット検証。**value**は比較照合用の8ビットの16進値です。
- q** ヒューズビット読み出し。
- Y** 発振器校正手順実行。より多くの情報についてはAVR053応用記述をご覧ください。
- Z** チップ消去に先立ってEEPROM位置から値を取得し、その後S任意選択を使って、フラッシュまたはEEPROM内に書き込めます。**address**は16進値です。より多くの情報についてはAVR053応用記述をご覧ください。
- c** 通信ポート選択。**COM1**～**COM8**。この引数が省略されると、プログラムはSTK500に関してCOMポートを走査します。

[次頁へ続く]



- ut 目的電圧(VTARGET)設定。valueは今度の電圧を記述する0.0～6.0の浮動小数点値です。
- ua アナログ基準電圧(AREF)設定。valueは今度の電圧を記述する0.0～6.0の浮動小数点値です。
- wt 現在の目的電圧(VTARGET)読み出し。
- wa 現在のアナログ基準電圧(AREF)読み出し。
- Wt 目的電圧(VTARGET)が与えられた値の5%以内であることを検査します。valueは0.0～6.0の浮動小数点値です。
- b 改訂番号取得。ハードウェア改訂(h)、ソフトウェア改訂(s)。
- ! 発振器周波数設定。freqはHz, kHz, MHzの全体ができます。
- t 発振器周波数取得。
- I ISP周波数設定。freqはHz, kHz, MHzの全体ができます。
- J ISP周波数取得。
- h ヘルプ情報(他の全ての設定を無視)。

支援デバイス：支援されるデバイスを表示するには" STK500.EXE -?"を入力してください。

5.6 支援デバイス名

支援デバイス名は「1.2. 支援デバイス」で示される全デバイスです。

例: `STK500.EXE -dATmega128 -e -ifFlash.hex -pf -vf`
`STK500.EXE -dATmega128 -fF73A -FF73A -EFF -GFF`

スイッチ間とそれらのパラメータ間に空白(スペース)がなく、また16進値が前置子'0x'を持たないことに注意してください。

例1: `stk500 -dAT90S8515 -e -pf -if8515test.hex`

説明

stk500	STK500.EXE起動
-dAT90S8515	デバイスをAT90S8515に設定
-e	チップ消去
-if8515test.hex	フラッシュメモリへ書き込むべき8515test.hexファイルを選択
-pf	上記で選択したファイル内容をフラッシュメモリに書き込み

例2: `stk500 -ccom1 -ut4.5 -ua4.5 -!3686000 -dATMEGA8515 -lFF -fD9E4 -ms -ifC:¥input.hex -pf -vf -e`

説明

stk500	STK500.EXE起動
-ccom1	COM1ポートを選択
-ut4.5	目的の電圧を4.5Vに設定
-ua4.5	目的のアナログ基準電圧を4.5Vに設定
-!1228800	STK500のクロック周波数を約3.68MHzに設定
-dATMEGA8515	デバイスをATmega8515に設定 (支援されるデバイスを見るには <code>stk500.exe -?</code> と入力してください。)
-e	チップ消去
-lFF	施錠ビットを\$FFに設定
-fD9E4	ヒューズを\$D9E4(上位:\$D9, 下位:\$E4)に設定
-ms	直列プログラミング動作設定
-ifC:¥input.hex	フラッシュメモリへ書き込むべきinput.hexファイルを選択
-pf	上記で選択したファイル内容をフラッシュメモリに書き込み
-vf	上記で選択したファイル内容とフラッシュメモリ内容を比較検証

例1: `stk500 -dATmega128 -e -ipmyfile.elf -pa -va`

説明

stk500	STK500.EXE起動
-dATmega128	デバイスをATmega128に設定
-e	チップ消去
-ipmyfile.elf	書き込むべきmyfile.elfファイルを選択
-pa	上記で選択したelfファイルに格納された全メモリ形式を書き込み
-va	上記で選択したelfファイル内容と全メモリ内容を比較検証



外部実装書き込み(ISP)

STK500は他の応用でAVRデバイスを書くための書き込み器として使えます。6ピンと10ピンの2つの異なるピン配置のISPコネクタが利用できます。両方共STK500によって支援されます。

STK500上でデバイスをプログラミングするのと同じ方法で書かれるデバイスを選択します。目的応用回路のVCCはSTK500によって検出され、信号は目的システムに適合する電圧レベルに変換されます。

図6-1. 6ピン ISPコネクタ ピン配置

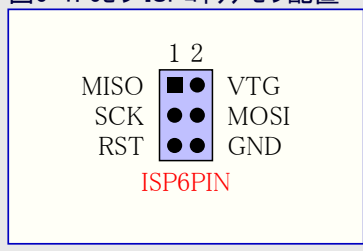
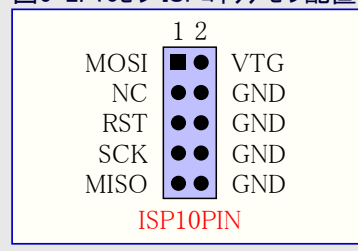


図6-2. 10ピン ISPコネクタ ピン配置



注: 他(外部)の応用回路が自身の電力をVTGに供給する場合、STK500を他の応用回路に接続する前にVTARGETジャンパを取り除かれなければなりません。VTARGETジャンパが取り除かれしないと、STK500は損傷を受けるかもしれません。

障害対策の指針

表7-1. 障害対策の指針

問題	原因	対処
電源LEDが赤色点灯しない。	DC電源ケーブルが接続されていない。	DC電源ケーブルをDCジャック(図2-2)に接続します。
	異常な電源の使用。	電源がDC10～15V,500mA以上を供給できるか検査します。
	電源スイッチがOFF。	電源スイッチをONにします。(図3-1)
試供プログラム実行でLEDが点滅せず。	ソケットにAVRデバイスがない。	右側ソケットにAVRデバイスを装着します。
	LEDがI/Oポートに接続されていない。	LEDSとPORTD、SWITCHESとPORTBの各ヘッダ間を接続します。(図2-1)
	フラッシュメモリが消去されている。	STK500とパソコンを接続し、AVRデバイスに再書き込みをします。(2.3.2)
AVRデバイスに書き込めない。	パソコンに直列ケーブルが接続されていない。	RS232PROGポートとパソコンのCOMポートを接続します。(図2-2)
	AVRデバイスが不正なソケットに挿入。	正しいソケットを使っているか検査します。(表3-2)
	AVRデバイスが不正な向きで挿入。	AVRデバイスとソケットのノッチ(くぼみ)を一致させ挿入します。
	目標ISPヘッダが接続されていない。	ISP6PINと目的対象の正しいSPROGn ISPヘッダを6芯ケーブルで接続します。(表3-2)
	ジャンパ設定が不正。	ジャンパを既定設定にします。(表3-4)
	VTARGET電圧が低すぎる。	AVRデバイスのデータシートで最低動作電圧を確認します。
	メモリ施錠ビットが設定(プログラム)されている。	書き込む前にチップ消去を行います。
	リセット禁止ヒューズが設定(プログラム)されている。	リセット禁止(RSTDISBL)ヒューズを調べます。
	プログラミングが速過ぎる。	STK500の発振器設定を調べ、実際のクロックよりも速くないか確認します。
	CKDIV8ヒューズが設定されている。	ISPクロック周波数を下げます(5.3.1.2.1項)。
	RESET信号の外部プルアップ抵抗値が低すぎる。	外部プルアップ抵抗が4.7kΩ以上か確認します。
AVR StudioがSTK500を検出できない。	直列ケーブル未接続か電源がOFF。	直列ケーブルをRS232PROGポートに接続し、電源接続を確認します。(図2-2)
	パソコンのCOMポートが使用できない。	パソコンのCOMポートを使う他のプログラムを禁止します。 パソコンのCOMポートを変更します。
STK500の電源OFFができない。	中央+の電源コネクタを使っている。	中央-の電源コネクタを使います。 電源コネクタを抜くことで電源OFFにします。
状態LED低速点滅。	アナログ電源(AREF)の短絡。	短絡原因を取り除きます。
状態LED高速点滅。	目的電源(VTARGET)の短絡。	短絡原因を取り除きます。
	外部電源使用で外部電源がOFF。	AVR StudioからVTARGET設定を0.3V以下にします。
ATtiny13プログラミングが動作しない。	短い起動時間が設定されている。	詳細については障害情報をご覧ください。
外部電源動作時、LEDが動作しない。	LED動作にはSTK500の電源が必要です。	STK500に電源を供給し、電源をONにします。

7.1 STK500ファームウェア 手動更新

現状版のAVR Studioはどんな古いSTK500ファームウェア版も自動的に検出し、ファームウェアの更新を促します。何らかの理由で更新失敗(更新中にシリアルケーブルが外れる、電力が無くなるなど)の場合、次にAVR StudioがSTK500に接続を試みる時に基板が検出されず、自動更新手順は始まりません。

以下はSTK500基板の手動更新の仕方の手順です。この手続きは全ての場合について行われるべきです。

- ① STK500の電源を切ります。
- ② STK500電源投入起動中の間、**PROGRAM**釐を押下します。
- ③ パソコンとSTK500の**RS232 CTRL**と記載されたDSUB 9ピン コネクタとの直列接続を確認して、AVR Studioを開始します。
- ④ AVR Studioのメニューバーの**Tools**内に配置されたプログラム(**AVR prog**)を開始します。
- ⑤ **AVR prog Hex file**欄の**Browse**釐を押下(クリック)することによって、ファームウェア更新HEXファイル(**Stk500.ebn**)を位置付けます。AVR Studio標準インストールでの**Stk500.ebn**ファイルのパスは **C:\Program Files\Atmel\AVR Studio\Stk500**です。
- ⑥ **AVR prog Flash**欄の**Program**釐を押下(クリック)します。進捗(プログレス)バーが現れ、同時に付加情報が表示されます。照合動作が完了されるまで待ちます。
- ⑦ **AVR prog**プログラムを閉じます。
- ⑧ STK500基板の電源を切ってから入れます。STK500は今や新しいファームウェアで使われる用意ができました。



STK500に関する技術的な問い合わせはavr@atmel.comにお願い致します。問い合わせに際しては次に示す情報も併せてご連絡ください。

- AVR Studioの版 (版番号はAVR StudioのメニューからHelp⇒Aboutで表示)
- パソコンのプロセッサ型式とクロック速度
- パソコンのOS名と版番号
- 目的AVRデバイス名 (完全な部品番号)
- 書き込み電圧
- ジャンパ設定
- 問題点の詳細説明

9.1 LED,スイッチ制御

実行に際しては**PORTB**と**LEDS**、**PORTD**と**SWITCHES**を接続します。
LEDはどのスイッチが押されたかによって異なる動作をします。
AVR Studioへ以下のプログラムを複写し、アセンブル、書き込み後、実行します。
このプログラムの先頭で目的デバイスの定義ファイルをインクルードするのを忘れないでください。

```

;          * STK500 LEDとスイッチの実演 *

. INCLUDE "8515DEF. INC"          ; AT90S8515用定義ファイル読み込み

. DEF     TEMP      =R16          ; LED出力値保持変数
. DEF     DELAY1     =R17          ; 遅延用計数値1(内側計数値)
. DEF     DELAY2     =R18          ; 遅延用計数値2(外側計数値)

;          * 初期化 *

RESET:    SER        TEMP        ; TEMP=11111111
          OUT        DDRB, TEMP  ; PORTB全ビット出力設定

;          * 入出力処理 *

LOOP:     OUT        PORTB, TEMP ; LED表示更新
          SBIS       PIND, 0      ; スイッチ0=OFFでスキップ
          INC        TEMP         ; スイッチ0=ONでLED減少計数(負論理)
          SBIS       PIND, 1      ; スイッチ1=OFFでスキップ
          DEC        TEMP         ; スイッチ1=ONでLED増加計数(負論理)
          SBIS       PIND, 2      ; スイッチ2=OFFでスキップ
          ROR        TEMP         ; スイッチ2=ONでLEDを1ビット右回転
          SBIS       PIND, 3      ; スイッチ3=OFFでスキップ
          ROL        TEMP         ; スイッチ3=ONでLEDを1ビット左回転
          SBIS       PIND, 4      ; スイッチ4=OFFでスキップ
          COM        TEMP         ; スイッチ4=ONでLED反転表示
          SBIS       PIND, 5      ; スイッチ5=OFFでスキップ
          NEG        TEMP         ; スイッチ5=ONでLED反転後+1
          SBIS       PIND, 6      ; スイッチ6=OFFでスキップ
          SWAP       TEMP         ; スイッチ6=ONでLED上下ニブル入れ替え表示

;          * 表示値保持時間用待機 *

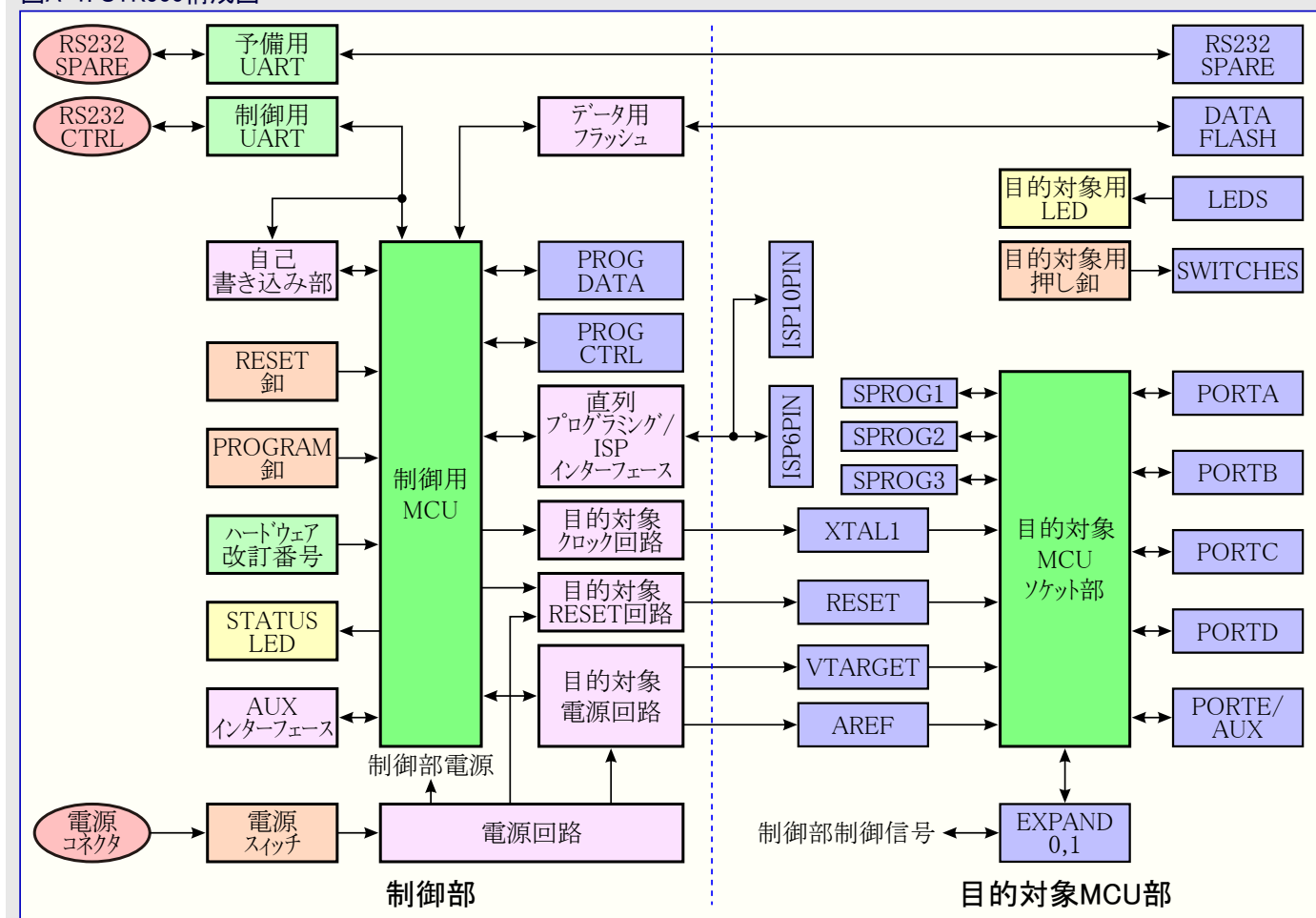
DLY:      DEC        DELAY1       ; 内側計数値減数
          BRNE       DLY          ; 256計数まで継続
          DEC        DELAY2       ; 外側計数値減数
          BRNE       DLY          ; 外側計数値256計数まで継続

;          * 無限繰り返し *

          RJMP       LOOP         ; 無限繰り返しへ

```

図A-1. STK500構成図

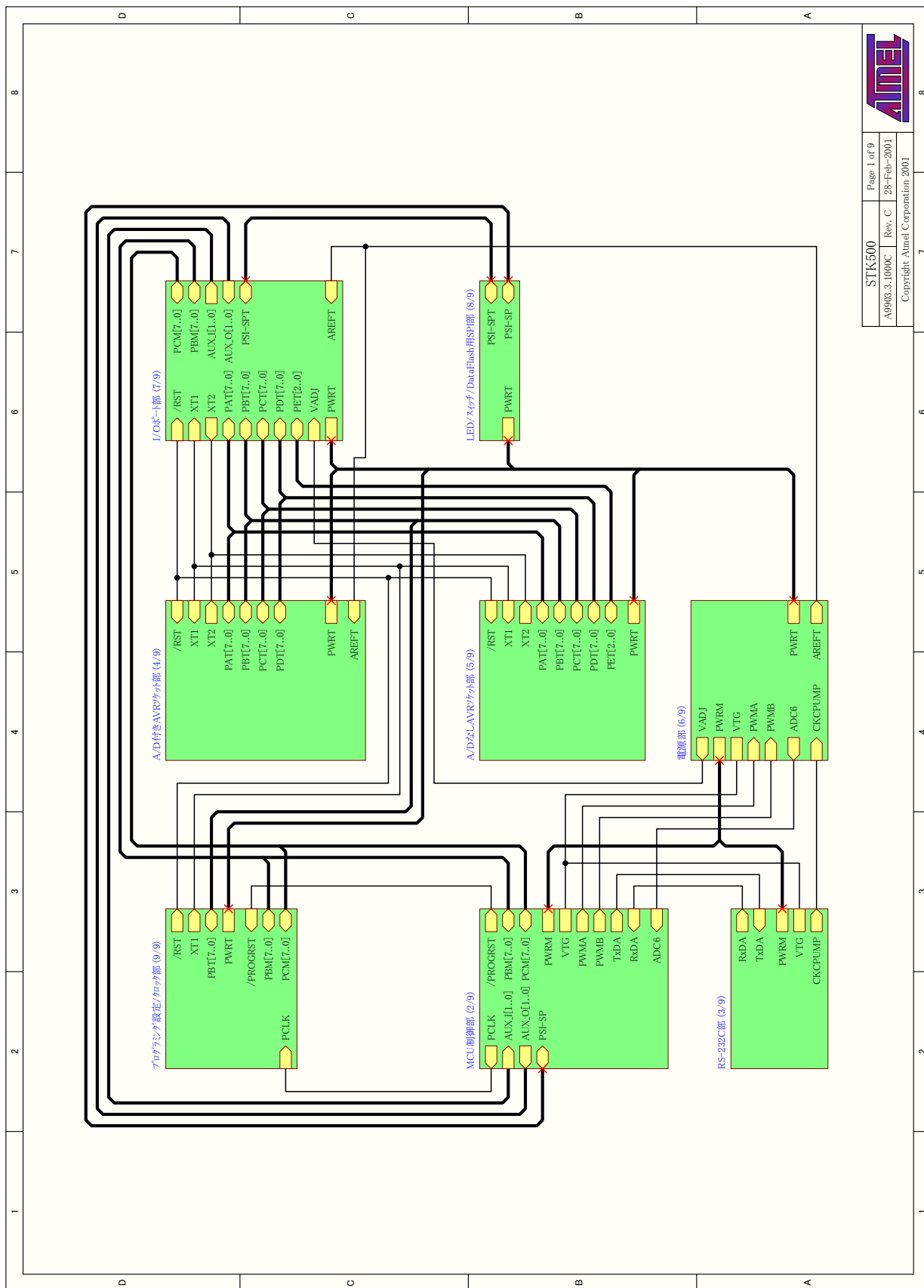


図B-1. 実装図

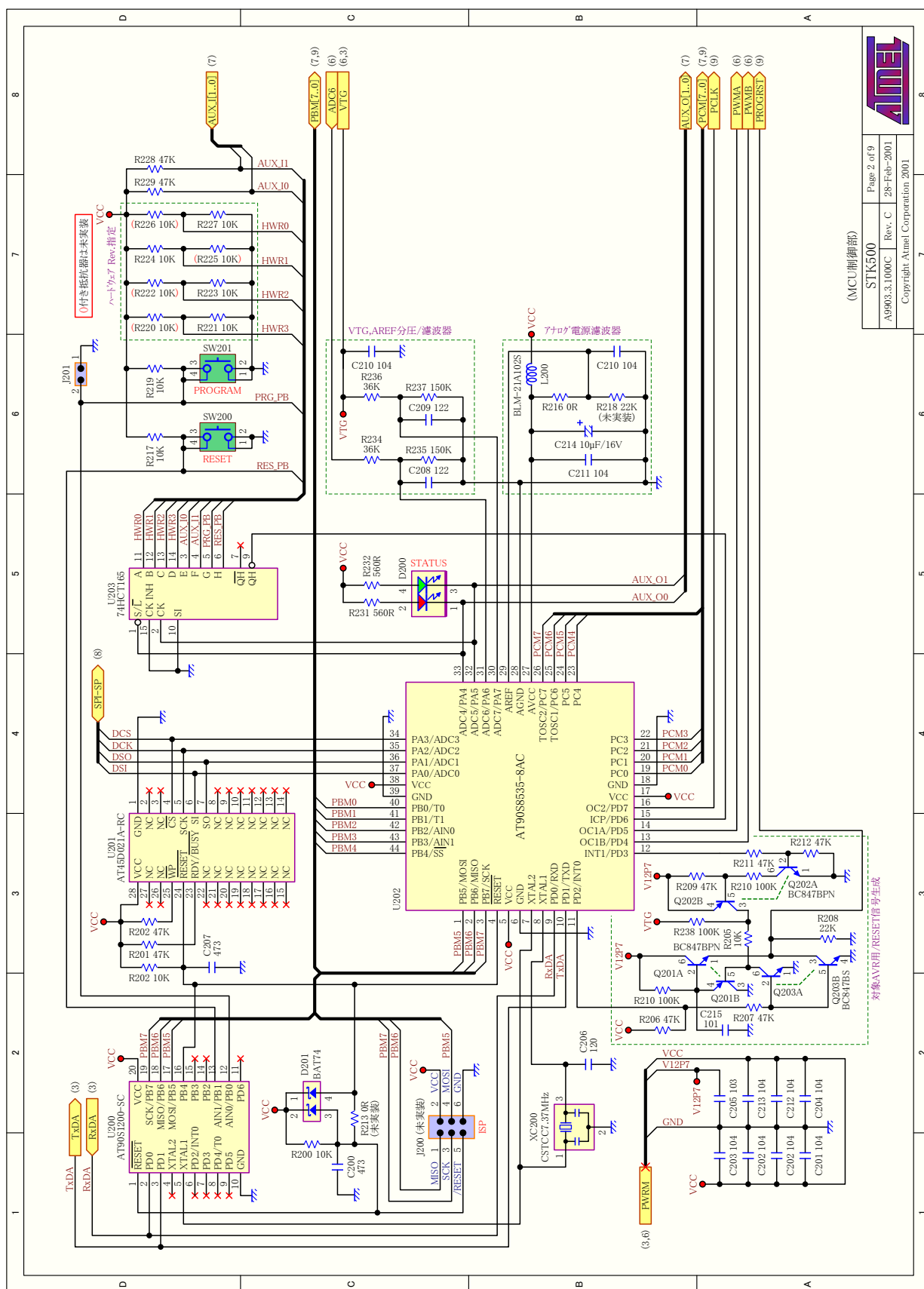


以下にSTK500改訂Cの全回路図が示されます。

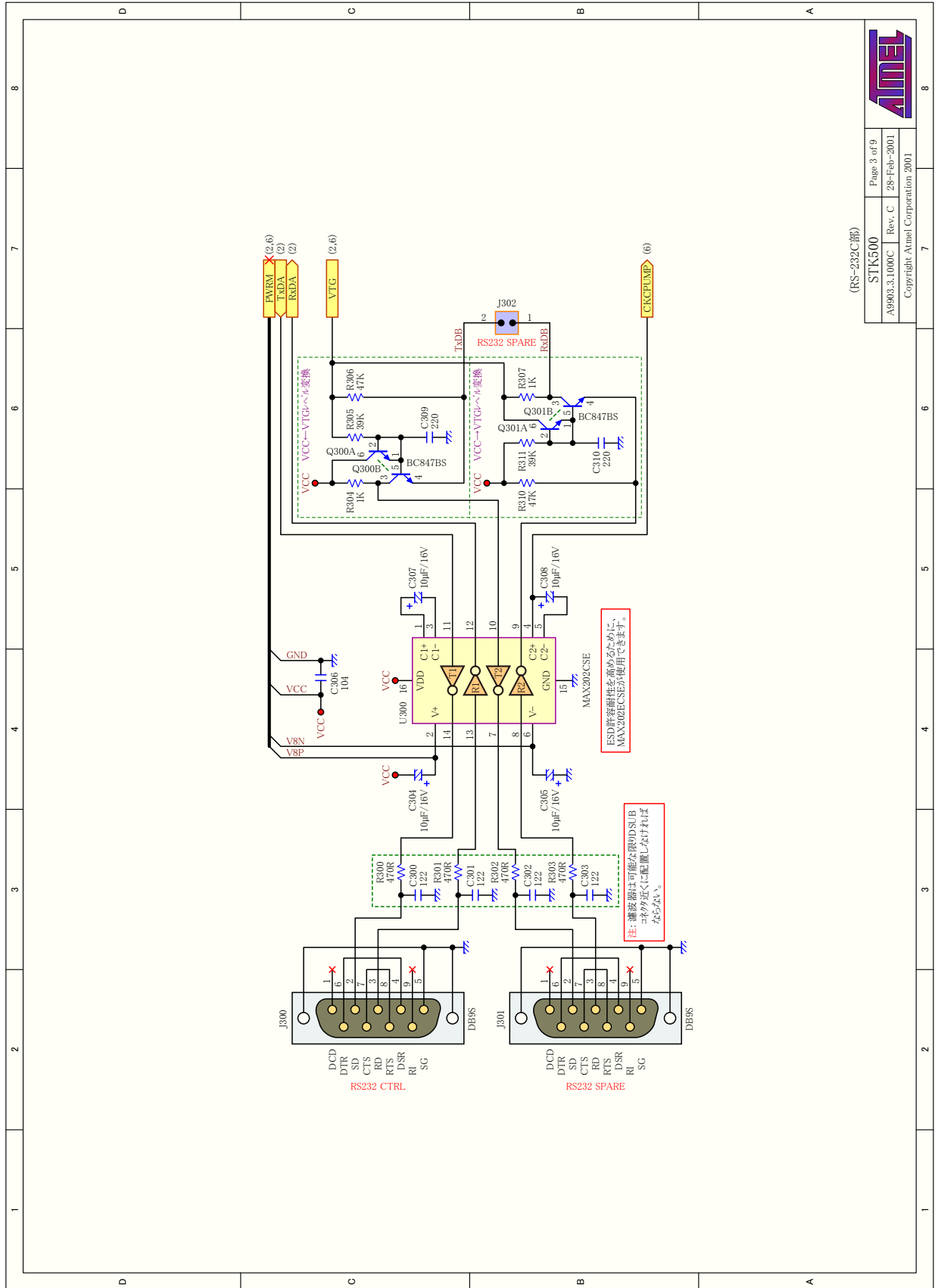
図C-1. 回路図 (1/9)



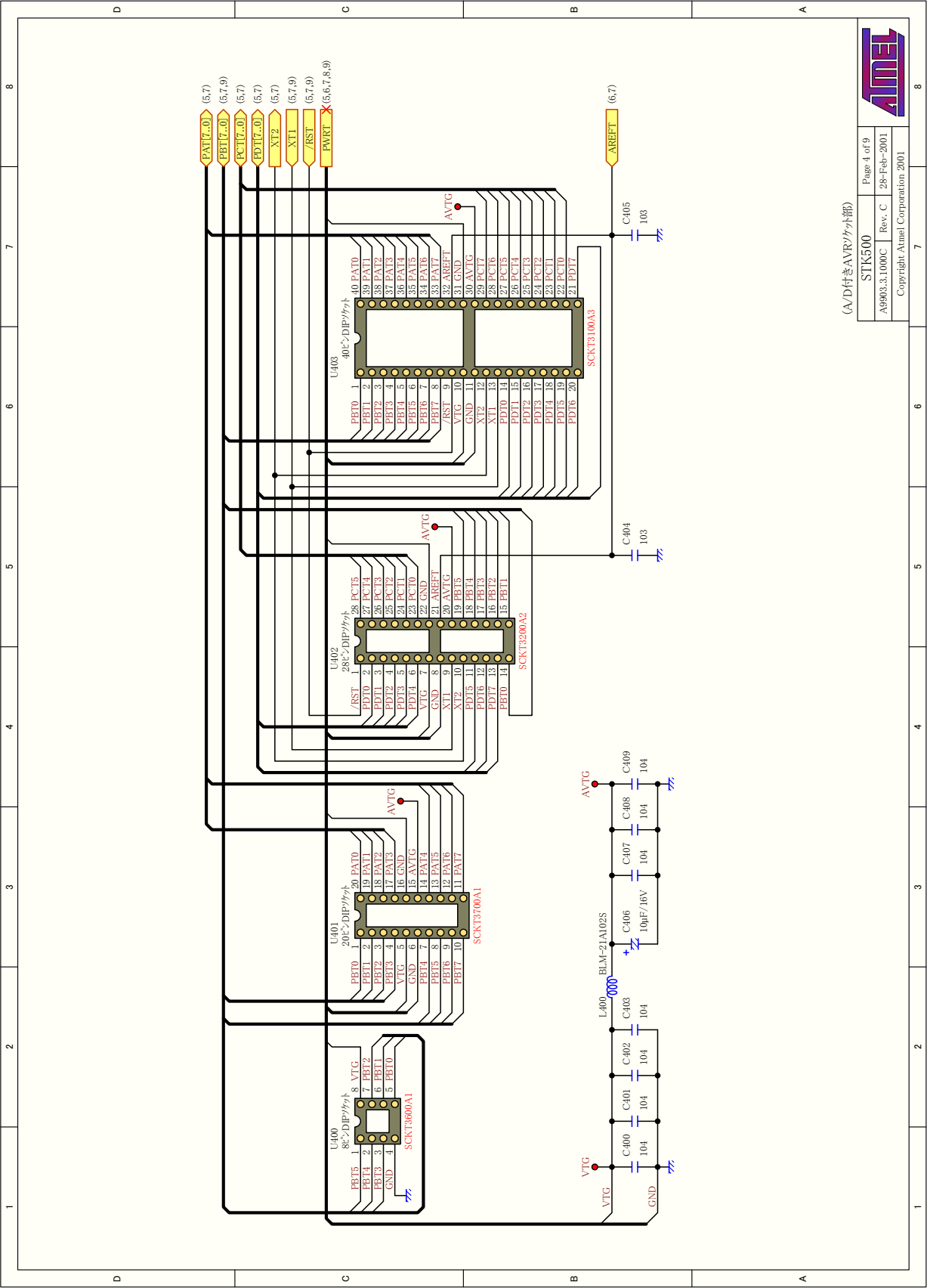
STK500 使用者の手引き



図C-3. 回路図 (3/9)



図C-4. 回路図 (4/9)

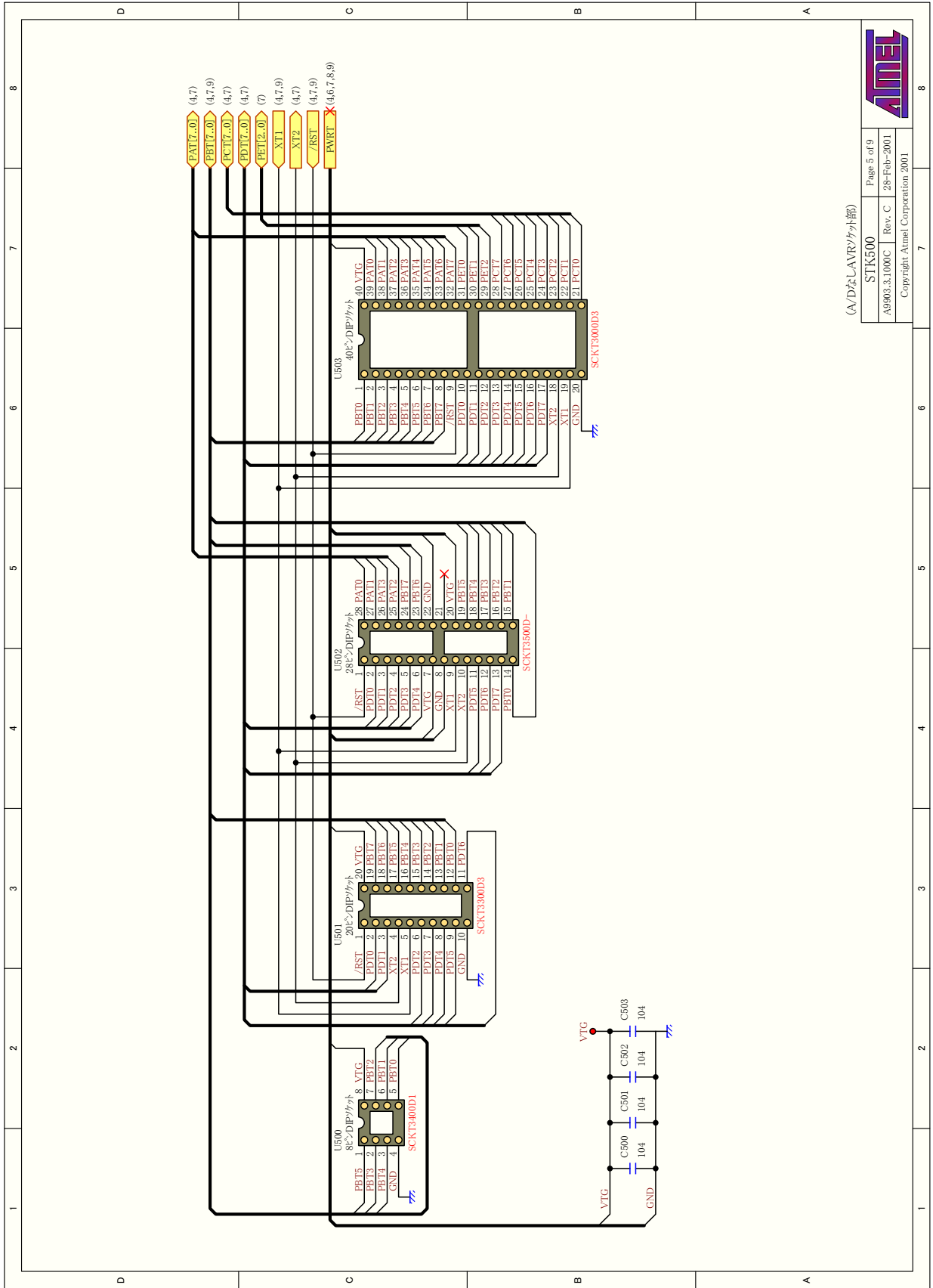


(A/D付きAVR/ATmega)

Page 4 of 9
Rev. C
28-Feb-2001
Copyright Atmel Corporation 2001



図C-5. 回路図 (5/9)



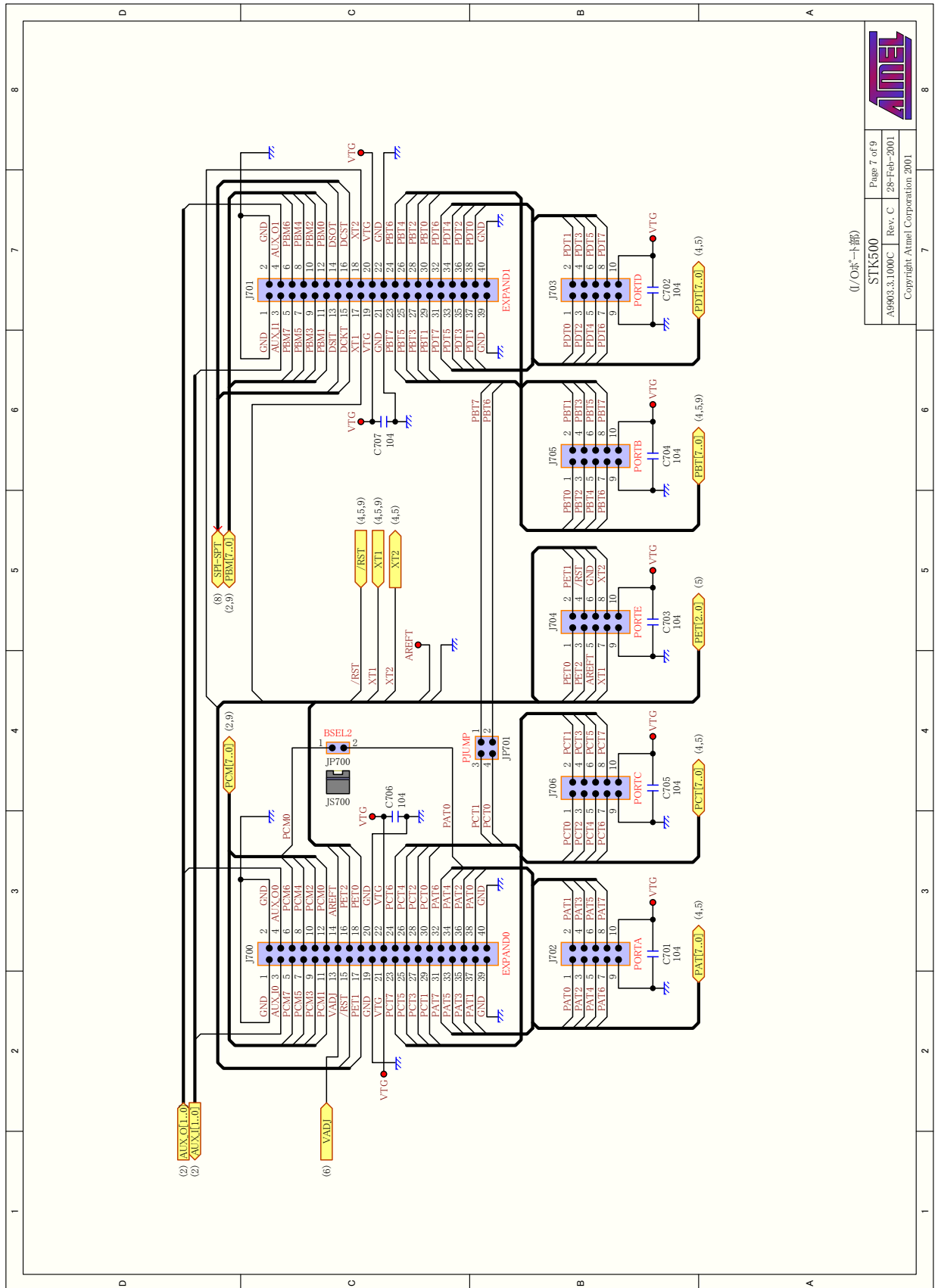
(A/DおよびLVR/リセット部)

Page 5 of 9
Rev. C
28-Feb-2001
Copyright Atmel Corporation 2001

STK500 使用者の手引き



図C-7. 回路図 (7/9)

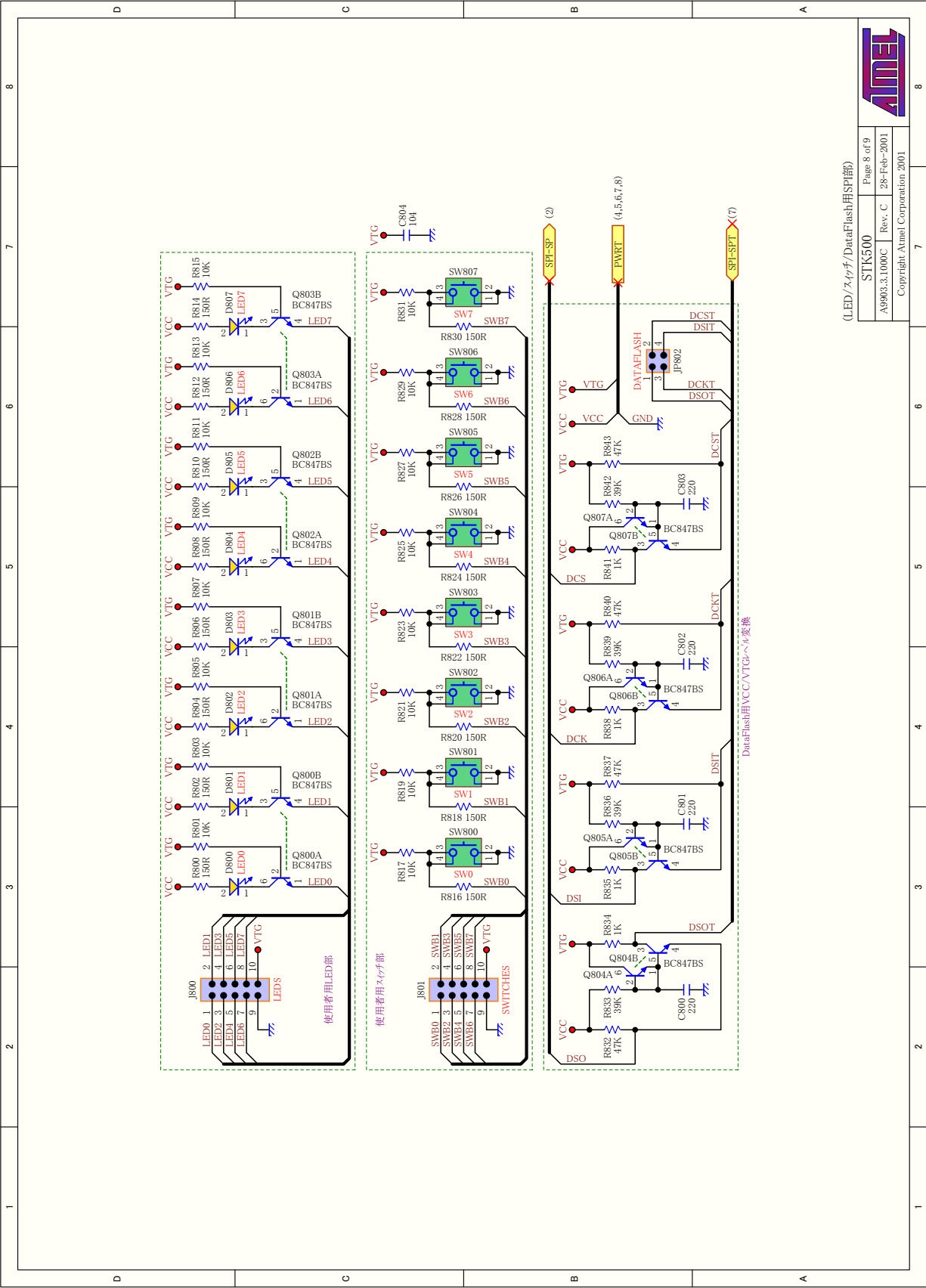


(1/0ホ*十部)

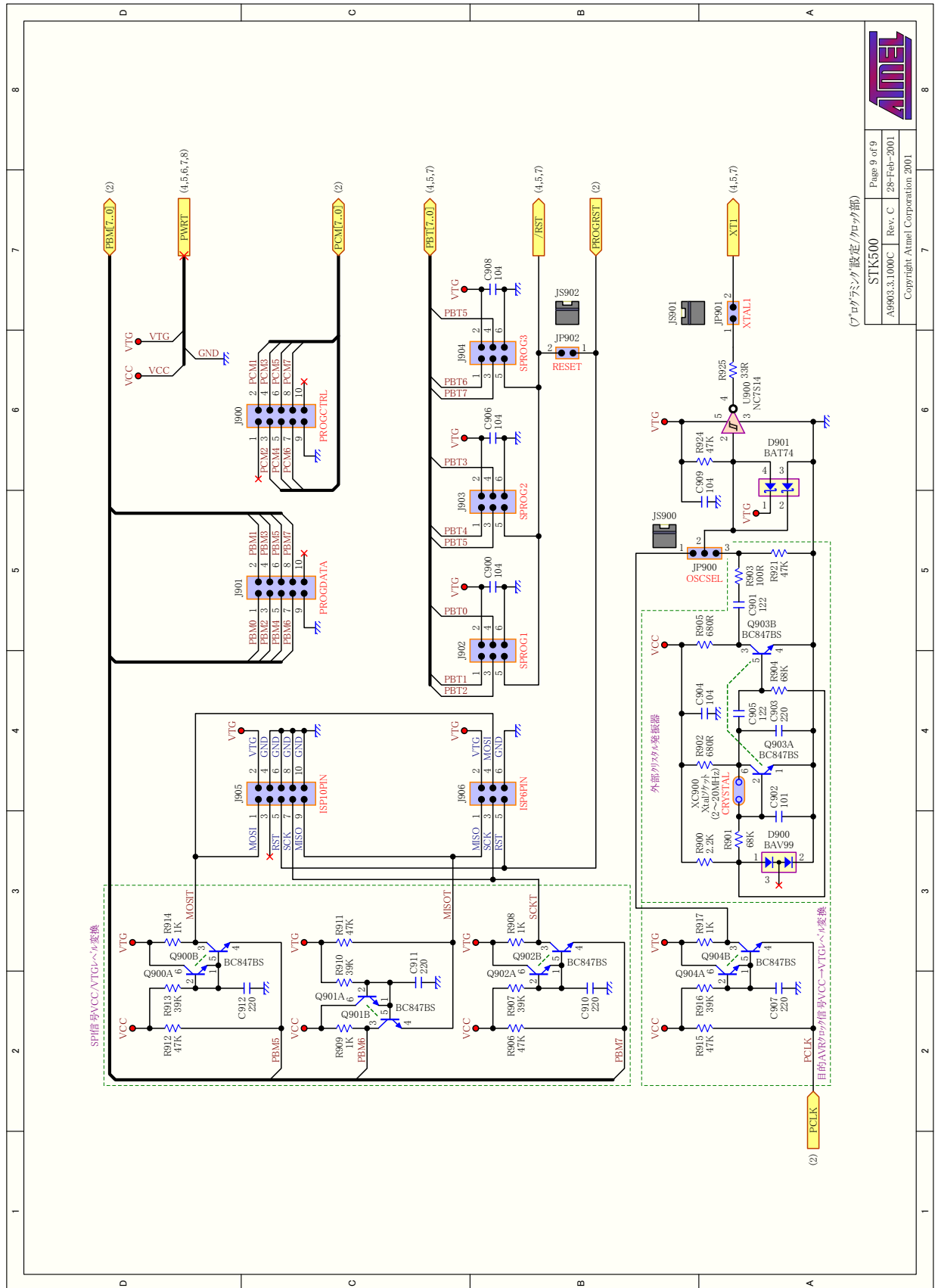
STK500	Page 7 of 9
A9903.3.1000C	Rev. C 28-Feb-2001
Copyright Atmel Corporation 2001	



図C-8. 回路図 (8/9)



図C-9. 回路図 (9/9)



(プログラマー設定/加工部)



Page 9 of 9	8
Rev. C	7
28-Feb-2001	6
Copyright Atmel Corporation 2001	5



1. ファームウェアの更新

ファームウェアの更新を完全に成し遂げるために全てのケーブルとデバイスがSTK500の目的対象ソケット領域から取り去られているのを確認してください。

ファームウェア 2.xx版はAVR Studio 4.11とそれ以降のAVR Studioでだけ動作します。STK500のファームウェア版とAVR Studioの版は整合している必要があります。

2. ATtiny13の障害

改訂Dよりも前のATtiny13 E.S.(技術試供品)には再プログラミングからデバイスを固定化させるヒューズビットの特定の組み合わせに関する障害があることに注意してください。次のヒューズビット設定の組み合わせがこの状態を発生します。

発振器設定	起動時間設定	デバッグWireとリセット設定
内蔵128kHz CKSEL[1,0]=11	最短 SUT[1,0]=00	デバッグWire許可(DWEN=0) または RESET禁止(RSTDISBL=0)
内蔵9.6MHz CKSEL[1,0]=10		
内蔵4.8MHz CKSEL[1,0]=01		

対策: 上記のヒューズの組み合わせを避けてください。より長い起動時間設定がこの問題をなくします。

クロック元と前置分周選択

ATtiny13にはクロック元と前置分周に関する多くの設定があります。もしもATtiny13で使われる前置分周クロックが32kHzよりも高く、且つSTK500で使われる選択発振器設定以下の場合にだけ、STK500はISPを使ってプログラミングできることに注意してください。

対策: ISPに代えて高電圧直列プログラミング(HVSP)を使ってください。

3. ATtiny2313の障害

改訂Cよりも前のATtiny2313の並列プログラミングには異常があり、従ってこれはSTK500で支援されないことに注意してください。

警告: ISPインターフェースを禁止するヒューズ設定を避けてください。これは更なるプログラミングに対してデバイスを固定化してしまいます。

クロック元と前置分周選択

ATtiny2313にはクロック元と前置分周に関する多くの設定があります。もしもATtiny2313で使われる前置分周クロックが32kHzよりも高く、且つSTK500で使われる選択発振器設定以下の場合にだけ、STK500はISPを使ってプログラミングできることに注意してください。

4. ATmega169の障害

\$FFバイトの連続を含むHEXファイルでのATmega169 EEPROMのISPプログラミングが失敗することに注意してください。

対策: ISPプログラミング使用ではプログラミング前にEEPROMを消去し、\$FF以外の値のバイトだけをプログラミングします。

5. リセット信号線のプルアップ

STK500目的対象リセット信号線上のプルアップは他の書き込み器が目的基板としてSTK500を使う関係に於いて強すぎます。

対策: STK500上の目的対象プログラミングに外部書き込み器を使う時にRESETジャンパを取り去ってください。

以下の内容は訳者らがSTK500使用中に体験した不具合点です。

1. 電源電圧変化

AVR StudioからSTK500のBoradタブでVTarget電圧を設定して読み込むと、設定電圧によって-0.1Vまたは+0.1V変化する場合があります。操作によっては意図しない電圧設定を行ってしまうかもしれません。十分に注意してください。

出荷時期によってはこの問題が発生しないものもあります。

2. ATtiny2313Aの障害

ATtiny2313Aの並列プログラミングが正しく機能しません。



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

製造拠点

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chanterrie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求

www.atmel.com/literature

© Atmel Corporation 2003.

Atmel製品は、ウェブサイト上にあるAtmelの定義、条件による標準保証で明示された内容以外の保証はありません。本製品は改良のため予告なく変更される場合があります。いかなる場合も、特許や知的技術のライセンスを与えるものではありません。Atmel製品は、生命維持装置の重要部品などのような使用を認めておりません。

本書中の®、™はAtmelの登録商標、商標です。

本書中の製品名などは、一般的に商標です。

© HERO 2020.

この使用者の手引きはAtmelのSTK500英語版使用者の手引き(Rev.1925C-03/03)の翻訳日本語版使用者の手引きですが、AVR Studio付属のヘルプ(STK500.CHM)の最新情報など、作成時点に於ける最新情報に追加修正されています。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。

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。