

要点

- AVRマイクロ コントローラ用の完全な実装書き込み法
- 実装書き込みを支援する全AVR マイクロ コントローラを網羅
- プログラム用フラッシュ メモリとデータ用EEPROM、両方の書き換え
- 低価格実装書き込み器用の完全な回路図
- 簡単な3線SPIプログラミング インターフェース

序説

実装書き込みは最終システム側に位置する、どのAVR マイクロ コントローラの書き込みと書き換えを許します。実装書き込み器は単純な3線SPIインターフェースを使ってAVR マイクロ コントローラと直列通信を行い、チップ上の不揮発性メモリを書き換えます。

実装書き込みはシステムからの物理的なチップの取り外しをなくします。これは開発中と現場でのソフトウェアまたはパラメータの更新時の両方で時間と費用を節約します。

本応用記述は実装書き込みを支援するためのシステム設計法を示します。標準9ピン シリアル ポートを装備する、どんなパソコンからも対象AVRマイクロ コントローラへの書き込みを可能にする低価格実装書き込み器の作成法も示します。更に自身の書き換えを許すために、実装書き込み器全体をシステム内に組み込むこともできます。

(訳補) 本応用記述が基本的に第1世代プロセス製品を基にしていることに注意してください。

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

目次

要点	1
序説	1
1. プログラミング インターフェース	3
2. ハードウェア設計での考慮	3
2.1. GND	3
2.2. RESET	3
2.3. SCK	4
2.4. MOSI	4
2.5. MISO	4
2.6. VCC	4
3. プログラミング規約	5
3.1. 命令形式	5
3.2. メモリ読み書き許可	5
3.3. デバイス符号	5
3.4. プログラム用フラッシュ メモリ読み書き	6
3.4.1. データ用EEPROM読み書き	7
3.4.2. 施錠ビット読み書き	7
3.4.3. チップ消去操作	7
4. 簡単低価格実装書き込み器	8
5. 部品表	8
6. 改訂履歴	9

1. プログラミング インターフェース

実装書き込みについて、書き込み器は可能な限り少ない線を使って対象に接続されます。何れかの対象システム上の何れかのAVR マイクロコントローラを書くために対象基板へ書き込み器を接続するのに、簡単な6線インターフェースが使われます。下図は必要とされる接続を示します。

直列周辺インターフェース(SPI)は直列クロック(SCK)、主装置入力従装置出力(MISO)、主装置出力従装置入力(MOSI)の3線から成ります。AVRのプログラミング時、実装書き込み器は常に主装置として扱い、対象システムは常に従装置として取り扱います。

実装書き込み器(主装置)はSCK信号線に通信用のクロックを提供します。SCK信号線の各パルスは主装置出力従装置入力(MOSI)上で書き込み器(主装置)から目的対象(従装置)へ1ビットを転送します。同時にSCK信号線の各パルスは主装置入力従装置出力(MISO)上で目的対象(従装置)から書き込み器(主装置)へ1ビットを転送します。

3線SPIでの正しい通信を保証するため、書き込み器のGNDを対象のGNDへ接続することが必要です。直列プログラミング動作への移行とそれに留まるために、AVR マイクロコントローラのリセット信号は有効(Low)に保たれなければなりません。また、チップ消去を実行するために、リセットはチップ消去周期の最後にパルス出力されなければなりません。プログラミング作業を容易にするために、第4の制御線(RST)を使い、本処理を自動化するために書き込み器が対象マイクロコントローラのリセット線の制御を行なうことが選ばれます。

許された何れかの電圧(2.7~6.0V)で走行する対象のプログラミングを許すため、書き込み器は対象システムから電力(VCC)を引き込みます。これは書き込み器用の独立した電源の必要をなくします。また、対象システムはプログラミング時に書き込み器から電力供給もでき、プログラミング間、通常の電源コネクタを通す対象システムへの電力の必要をなくします。

右図は対象システムへ接続するために本実装書き込み器で使われるコネクタを示します。この標準コネクタはピン間100mil(2.54mm)で2×3ピンヘッダ接触子を供給します。

図1-1. 書き込み器と対象システム間の6線接続

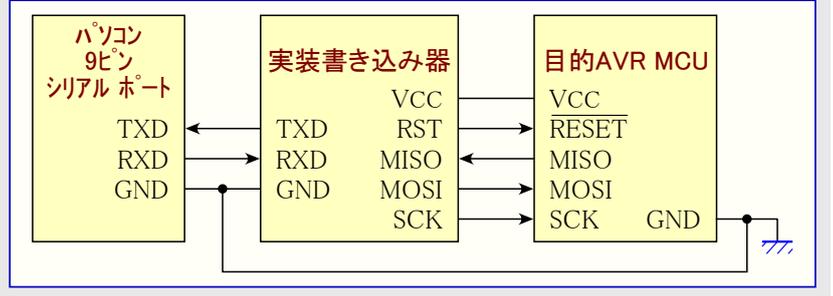


図1-2. 推奨実装書き込みインターフェースコネクタピン配置

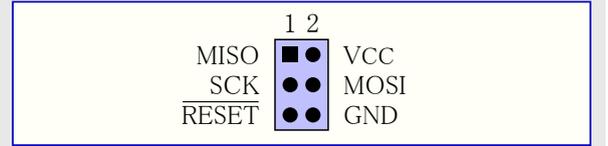


表1-1. 実装書き込みに必要な接続

ピン名	名称	説明
SCK	直列クロック	実装書き込み器(主装置)によって生成されたプログラミングクロック
MOSI	主装置出力従装置入力	実装書き込み器(主装置)からプログラミングされる目的対象AVR(従装置)への通信線
MISO	主装置入力従装置出力	目的対象AVR(従装置)から実装書き込み器(主装置)への通信線
GND	共通GND	2つのシステムは同じ共通GNDを共用しなければなりません。
RESET	目的AVR MCU リセット	実装書き込みを許可するため、対象AVRのリセットは有効(Low)を保持しなければなりません。これを簡単化するため、実装書き込み器は対象AVRのリセットを制御すべきです。
VCC	目的対象電源	どの電圧で動作する対象の簡単なプログラミングを許すため、実装書き込み器は対象から電力を引き込みます。代わりに対象はプログラミングの間、実装書き込みコネクタを通して供給される電力を得ることができます。

2. ハードウェア設計での考慮

AVRマイクロコントローラの実装書き込みを許すため、実装書き込み器はプログラミング中、ピン機能を無効にできなければなりません。本章はプログラミング操作に対して使われる各ピンの詳細を記述します。

2.1. GND

実装書き込み器と対象システムは同じ基準電圧で動作する必要があります。これは対象のGNDを書き込み器のGNDに接続することによって行なわれます。このピンに適用する特別な考慮はありません。

2.2. RESET

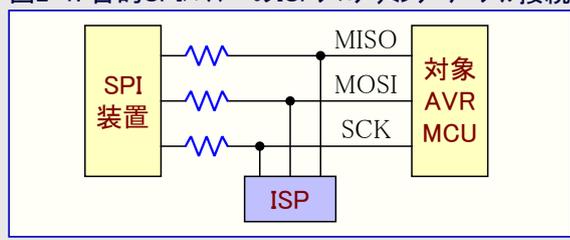
目的対象AVR マイクロコントローラはリセット信号線が有効(Low)の時だけ直列プログラミング動作へ移行します。リセット信号線はチップ消去時、消去周期の最後で(H/L)交互切り替えされなければなりません。この操作の単純化のため、対象のリセットは実装書き込み器によって制御できることが推奨されます。

リセットが有効になった後、直ちに実装書き込み器は、SCK,MISO,MOSIの3つの専用SPI線で通信を開始します。外部回路がこれらの信号線を駆動する可能性があるなら、駆動部の衝突を避けるために、3つの専用線の各々に直列抵抗器が配置されるべきです。この接続は右図で示されます。抵抗値はSPIバスに接続された回路に応じて選ばれるべきです。

注: リセット有効時、AVRマイクロコントローラは自動的に全I/Oピンをプルアップ禁止の入力に設定します。

問題を避けるために実装書き込み器はプログラミングの間中、対象システムリセット全体を保持できるべきです。リセットが有効の間、対象システムは、この3つのSPI線を駆動する試みを決してすべきではありません。

図2-1. 目的SPIバスへのISPプログラミングケーブル接続



2.3. SCK

直列動作でのAVRプログラミング時、実装書き込み器はSCKピン上にクロック情報を供給します。このピンは常に書き込み器によって駆動され、目的対象システムは対象リセットが有効の時にこの線を駆動する試みを決してすべきではありません。リセットが有効になった後、直ちに本ピンは書き込み器によってLowに駆動されるでしょう。プログラミング区間の最初の段階の間、パルスが目的対象AVRに書き込み器との同期を失わせるので、パルスに対して開放のままのSCK信号線は危険です。同期化時、プログラミング許可命令の第3バイト実行時に第2バイト値(\$53)が戻るでしょう。\$53が戻らなかったなら、正パルスをリセットに与え、(新しい)プログラミング許可命令を実行してください。

注: 新規送信を開始する前に、プログラミング許可命令の全4バイトが送られなければならないことに注意してください。

目的対象AVRマイクロコントローラはリセットが有効の時は必ずSCKピンを常にプルアップなしの入力に設定します。リセット信号線の記述もご覧ください。

直列クロック(SCK)入力に対するHighとLowの最小時間はデータシートの「直列プログラミング」項で定義されます。AT90S1200については右のように定義されます。

Low区間 > 1 XTAL1 クロック周期
High区間 > 4 XTAL1 クロック周期

2.4. MOSI

直列動作でのAVRプログラミング時、実装書き込み器はMOSIピン上に目的対象へのデータを提供します。このピンは常に書き込み器によって駆動され、目的対象システムは目的対象リセットが有効の時にこの線を駆動する試みを決してすべきではありません。

目的対象AVRマイクロコントローラはリセットが有効の時は必ずSCKピンを常にプルアップなしの入力に設定します。リセット信号線の記述もご覧ください。

2.5. MISO

リセットが目的対象AVRマイクロコントローラに印加されるとき、MISOピンはプルアップなしの入力に設定されます。プログラミング許可命令が対象へ正しく送信されてしまった後にだけ、目的対象AVRマイクロコントローラはMISOピンが出力になることを設定します。この初めの間、実装書き込み器は目的対象マイクロコントローラによって駆動されるまで、MISO線の安定を保つためにプルアップを適用します。

2.6. VCC

目的対象マイクロコントローラのプログラミング時、書き込み器出力はDC特性で指示された範囲内に留まる必要があります。

どの目的対象電圧にも容易に適合させるため、書き込み器は目的対象システムから必要とする全電力を引き込みます。これは実装書き込み器が目的対象システムから、代表的には20mAより多くない、非常に小さな電力を引き込むだけなので許されます。この応用記述で示す本書き込み器はこの動作で動きます。

逆に目的対象システムは通信用に使う同じコネクタを通して書き込み器から供給された電力を得ることもできます。これは網的对象への外部的な電力供給なしのプログラミングを目的対象に許します。

表2-1. 実装書き込みを支援するハードウェア設計時の勧告

ピン名	勧告
GND	目的対象のGNDを実装書き込み器のGNDに接続してください。
RESET	目的対象システムのリセットを実装書き込み器に許してください。
SCK	目的対象AVRマイクロコントローラのリセット有効時、本信号線はISP書き込み器によって制御されるべきです。リセットがLowに引かれた後の本信号線のエッジは危険で、目的対象AVRマイクロコントローラに書き込み器との同期を失わせます。プログラミング時に於いて、AVRリセット有効時、本ピンの振幅はシステム周辺に黙許されるべきです。
MOSI	目的対象AVRマイクロコントローラのリセット有効時、本信号線はISP書き込み器によって制御されるべきです。プログラミング時に於いて、AVRリセット有効時、本ピンの振幅はシステム周辺に黙許されるべきです。
MISO	目的対象AVRマイクロコントローラのリセット有効時、本信号線はISP書き込み器によって制御されるべきです。プログラミング時に於いて、AVRリセット有効時、本ピンの振幅はシステム周辺に黙許されるべきです。
VCC	許されたどの目的対象電圧にも適合するため、目的対象システムから電力を引き込むことを実装書き込み器に許してください。書き込み器の電力で必要な最大電流は使われる書き込み器に応じて変わります。

3. プログラミング規約

目的対象AVRマイクロコントローラでリセットが有効になった後、直ちにチップはプログラミング動作への移行準備が整います。内部直列周辺インターフェース(SPI)は活性化され、書き込み器からの命令を受け入れる準備が整います。1つのエッジが目的対象に書き込み器との同期を失わせるので、AT90S1200でのSCKピンの安定を保つことが非常に重要です。他のデバイスについてはデータシートで指示された同期化手順を使います。リセットをLowに引いた後、最初の命令を実行する前に最低20ms待機してください。

3.1. 命令形式

全ての命令は4バイトから成る共通形式を持ちます。最初のバイトは操作と目的対象メモリを選択する命令符号を含みます。第2、第3バイトは選択したメモリ空間のアドレスを含みます。第4バイトはどちらかの方向となるデータを含みます。

通常、対象から返されるデータは直前に送ったデータです。下表は2つの連続する命令が対象へ送られる例を示します。返された全バイトが丁度受信したバイトと等しいことに注意してください。いくつかの命令は対象のメモリから1バイトを返します。このバイトは常に最終(第4)バイトで返されます。データは常に最上位ビット(MSB)先行でMOSIとMISO上に送られます。

利用可能な命令の詳細についてはデータシートの「直列プログラミング」項を参照してください。

3.2. メモリ読み書き許可

リセットピンが始めて有効(Low)に引かれるとき、SPIインターフェースによって受け入れ可能な命令は「プログラミング許可」だけです。この命令だけがフラッシュとEEPROMメモリへの読み書きを開放し、この命令なしでの他の如何なる命令も無視されます。下表は先にチップへ送った命令でメモリ読み書きが許可される例を示します。

表3-1. メモリ読み書き許可とチップ消去の例

動作	目的対象AVRへの送信(MOSI)	目的対象AVRからの返り(MISO)
プログラミング許可	\$AC \$53 \$xx \$yy	\$zz \$AC \$53 \$xx
アドレス\$00の識票符号\$1E読み出し	\$30 \$nn \$00 \$mm	\$yy \$30 \$nn \$1E

「プログラミング許可」命令が対象へ送られてしまった後、現在の施錠ビット保護設定に従ってチップの不揮発性メモリへの読み書き(権)が与えられます。

目的対象AVRマイクロコントローラは「プログラミング許可」命令への承認での返答を行いません。命令が目的対象AVRマイクロコントローラによって受け入れられたのを調べるために、デバイス符号を読むことができます。デバイス符号は識票バイトとしても知られます。

3.3. デバイス符号 (識票バイト)

「プログラミング許可」命令がSPIインターフェースによって完全に読まれてしまった後、書き込み器はデバイス符号を読むことができます。デバイス符号はチップ供給者(Atmel)、デバイス系統(AVR)、Kバイト単位のフラッシュ容量、系統番号(例えばAT90S1200)を識別します。「識票バイト読み出し」命令形式はデータシートの「直列プログラミング」項で得られます。例としてAT90S1200に対する本命令は[\$30,\$xx,\$adr,\$code]です。有効なアドレス(adr)は\$00,\$01,\$02です。下表は見込まれた結果が何かを示します。

その次の表は度々\$FFとして読まれるデバイス符号を示します。これが起こる場合、デバイス符号がデバイス内にプログラムされていません。これは異常を示しませんが、デバイスは書き込み器に手動で識別されなければなりません。

デバイス符号\$FFは対象の準備できていない場合や、MISO信号線が常にHighへ引かれている場合にも起きるかもしれません。書き込み器は目的対象へ送った命令が\$FFとして返されることでの検知によっても、この状態を検出できます。

目的対象が供給者符号\$00、デバイス系統\$01、デバイス番号\$02を報告する場合、両方の施錠ビットがプログラム(0)されています。これは応答からメモリ領域を保護し、書き込み器から受信した現在のアドレスを表す値が返されます。施錠ビットを消去するには有効な「チップ消去」を実行する必要があります。

表3-2. 許されたデバイス符号

アドレス	符号	有効な符号
\$00	供給者符号	\$1EはAtmelによって製造されたことを示します。 \$00はデバイスが閉ざされているのを示します。以下をご覧ください。
\$01	デバイス系統とフラッシュ容量	\$9nは2 ⁿ KバイトフラッシュメモリのAVRを示します。
\$02	デバイス番号	デバイスを識別します。支援デバイスの全一覧についてはavr910.asmファイルをご覧ください。

表3-3. デバイス番号識別例

デバイス系統とフラッシュ容量	デバイス番号	デバイス
\$90	\$01	AT90S1200
\$91	\$01	AT90S2313
\$92	\$01	AT90S4414
\$93	\$01	AT90S8515
\$FF	\$FF	消去されたデバイス符号(または対象なし)
\$01	\$02	閉ざされたデバイス

表3-4. AT90S1200からのデバイス符号(予期値:\$1E,\$90,\$01)読み出し例

動作	目的対象AVRへの送信(MOSI)	目的対象AVRからの返り(MISO)
アドレス\$00の供給者符号読み出し	\$30 \$xx \$00 \$yy	\$zz \$30 \$xx \$1E
アドレス\$01のデバイス系統とメモリ容量読み出し	\$30 \$nn \$01 \$mm	\$yy \$30 \$nn \$90
アドレス\$02のデバイス番号読み出し	\$30 \$xx \$02 \$yy	\$mm \$30 \$xx \$01

3.4. プログラム用フラッシュ メモリ読み書き

デバイスが認識されてしまうと、フラッシュメモリの読み書きを始める時間です。フラッシュメモリのプログラミングに先立ってチップ消去命令が実行されるべきです。もくてき対象デバイスによってフラッシュメモリは「バイト」または「ページ」の動作を使って書き込まれます。

バイト書き込み動作を持つデバイスについては、各フラッシュメモリ位置が個別に書かれて、(最終的に)仕上げられます。ページ書き込み動作では、ページ一時緩衝部が初めに満たされ、その後の単一書き込み周期で書かれます。この動作はフラッシュメモリ合計書き込み時間を短縮します。(訳補:初期の)デバイスはこれら動作の1つだけが利用可能です。バイト書き込み動作(訳補:だけ)のデバイスにはページ書き込み任意選択がありません。フラッシュメモリのページ書き込み動作を持つデバイスでも、EEPROMについてはバイト書き込みを使ってください(訳補:後期デバイスではEEPROMのページ書き込みも可能)。

デバイスが「バイト書き込み動作」または「ページ書き込み動作」を使うかに拘らず、フラッシュメモリは「プログラム用フラッシュメモリ読み出し」命令を使ってバイト単位で読めます。この命令は16ビット語を選択するためにメモリアドレス(\$HH,\$LL)を送り、命令バイト内のPビットで上位(=1)または下位(=0)バイトを選択します。このアドレスに格納されているバイト値は、その後の目的対象AVRマイクロコントローラからの第4バイトで返されます。

通常、フラッシュメモリ内の各16ビット語は1つのAVR命令を含みます。アドレス\$0104に格納された命令が“ADD R16,R17”と仮定すると、この命令に対する命令符号は\$0F01として格納されているでしょう。アドレス\$0104を順次読むと、第4バイトで返される期待される結果は、上位バイトからが\$0F、下位バイトからが\$01です。MISOとMOSI信号線上のデータは右表で示されるように見えるでしょう。

表3-5. フラッシュメモリ位置\$0104から\$0F01として“ADD R16,R17”読み出し例

動作	目的対象AVRへの送信(MOSI)	目的対象AVRからの返り(MISO)
アドレス\$0104下位バイトでの\$01読み出し	\$20 \$01 \$04 \$xx	\$zz \$20 \$01 \$01
アドレス\$0104上位バイトでの\$0F読み出し	\$28 \$01 \$04 \$yy	\$xx \$28 \$01 \$0F

然しながらフラッシュメモリへの書き込みは利用可能な書き込み動作によって異なります。

バイト書き込み動作を使うデバイスに対しては、バイトが「プログラム用フラッシュメモリ書き込み」命令で書かれます。この命令は16ビット語を選択するためにメモリアドレス(\$HH,\$LL)を送り、命令バイト内のPビットで上位(=1)または下位(=0)バイトを選択します。格納すべきバイト値は、その後の目的対象AVRマイクロコントローラへの第4バイトで送られます。

ページ書き込み動作を使うデバイスに対しては、フラッシュメモリが2段階で書かれます。初めにページ一時緩衝部が「フラッシュページ内バイト設定」命令を使って満たされます。この緩衝部内の各バイトは直接アドレス指定ができます。ページ一時緩衝部が満たされれば、「フラッシュページ書き込み」命令を使ってフラッシュメモリに書くことができます。

いくつかのデバイスでフラッシュメモリ書き込み周期の終了時を検出する方法がありません。このため、本応用記述内で示した書き込み器は、インターフェースへ他の命令の送を試みる前にNms待機します(遅延Nは目的対象デバイスに依存し、これはデータシートの「直列プログラミング」項で得られます)。いくつかのデバイスについてはポーリングの使用が利用可能です(訳補:第2世代プロセス製品の或る時点以降のデバイスでは「多忙/準備可検査」命令も利用できます)。フラッシュメモリまたはEEPROM内にバイトが書かれるとき、書かれる位置の読み出しは値M(多くは\$FF)を与えるでしょう。デバイスが新規バイトに対する準備が整うと、書いた値が正しく読めるでしょう。これは次バイトを書ける時を決めるのに使えます。書く値がMのとき、ポーリングは動作せず、次の値を書く前に遅延Nが使われるべきです。ポーリング動作はデバイスを書くために必要な時間を短縮するでしょう。

表3-6. フラッシュメモリ位置\$010Cへ\$0F12として“ADD R17,R18”(バイト)書き込み例

動作	目的対象AVRへの送信(MOSI)	目的対象AVRからの返り(MISO)
アドレス\$010C下位バイトでの\$12書き込み	\$60 \$01 \$0C \$12	\$zz \$60 \$01 \$0C
Nms待機		
アドレス\$010C上位バイトでの\$0F書き込み	\$68 \$01 \$0C \$0F	\$xx \$68 \$01 \$0C
Nms待機		

3.4.1. データ用EEPROM読み書き

「データ用EEPROM読み出し」命令の使用でEEPROM内容がバイト単位で読めます。この命令はEEPROM内のバイト位置を選択するためにメモリアドレス(\$HH,\$LL)を送ります。

表3-7. EEPROM位置\$003Fから\$AB読み出し例

動作	目的対象AVRへの送信 (MOSI)	目的対象AVRからの返り (MISO)
アドレス\$003Fでの\$AB読み出し	\$A0 \$00 \$3F \$xx	\$zz \$A0 \$00 \$AB
Nms待機		

EEPROMは「データ用EEPROM書き込み」命令でバイト単位に書かれます(訳補:第2世代プロセス製品の或る時点以降のデバイスではページ書き込みも利用できます)。この命令は「データ用EEPROM読み出し」命令と同じように書くバイト(位置)を選択し、目的対象へ送る最終(第4)バイトで書かれるべきデータを転送します。いくつかのデバイスで書き込み周期の終了時を検出する方法がありません。書き込み器はインターフェースへ他の命令を送ろうとする前に単にNms待つべきです(遅延Nは対象デバイスに依存し、これはデータシートの「プログラミング」章で得られます)。書き込み速度の増加に対しては「プログラム用フラッシュメモリ読み書き」項で記載されるように、ポーリングを使うことができます。EEPROM書き込みの例は右表で示されます。

表3-8. EEPROM位置\$0011へ\$0F書き込み例(バイト書き込み動作)

動作	目的対象AVRへの送信 (MOSI)	目的対象AVRからの返り (MISO)
アドレス\$0011での\$0F書き込み	\$C0 \$00 \$11 \$0F	\$zz \$C0 \$00 \$11
Nms待機		

3.4.2. 施錠ビット読み書き

権限のない読み出しや予期せぬ上書きからメモリ内容を守るために、施錠ビットをメモリ内容保護に設定できます。下表で示されるようにメモリは更なる書き込みからの保護、またはチップ上のメモリの読み書き両方の完全な禁止のどちらかにできます。

いくつかのデバイスの施錠ビットは読むことができず、施錠ビット設定は書き込み器によって照合することができません。これらのデバイスで施錠ビットが正しく設定されてしまっていることを調べために、EEPROMの位置切り替えを試みるべきです。施錠ビット1(LB1)が設定(プログラム(0))されていると、メモリ位置は切り換えられません。施錠ビット1(LB1)と施錠ビット2(LB2)が設定(プログラム(0))されると、位置が読めず、返された結果は命令で通したアドレスの下位バイトです。施錠ビット2(LB2)だけの設定(プログラム(0))は保護効果がありません。チップが読み出しから保護される前に書き込みからの保護が完了されなければなりません。

施錠ビットはメモリ内容変更からプログラミングインターフェースを妨げるだけです。コアは施錠ビット設定と無関係に通常どおりプログラム用フラッシュメモリの読み込みとEEPROMの読み書きができます。

表3-9. 施錠ビットの保護種別

施錠ビット		保護種別
1 (LB1)	2 (LB2)	
1	1	メモリ保護なし
0	1	フラッシュメモリとEEPROMの更なる書き込み禁止
0	0	フラッシュメモリとEEPROMの更なる書き込みと読み出し禁止

施錠ビット設定後にメモリへの読み書きを復活するには、「チップ消去」命令でチップ全体を消去することによる方法だけです。施錠ビットは全メモリ位置の消去完了直後だけに、保護を禁止する1に消去されます。

チップ消去で施錠ビットはビットが消去されていることを示す値1を得ます。保護を許可する操作が施錠ビットの「設定」として参照される换言之、保護を許可するためには0値が本ビットに書かれるべきです。

表3-10. 更なる書き込み禁止のための施錠ビット1(LB1)設定例

動作	目的対象AVRへの送信 (MOSI)	目的対象AVRからの返り (MISO)
書き込み禁止の施錠ビット1(LB1)設定	\$AC \$FD \$xx \$yy	\$zz \$AC \$FD \$xx
Nms待機		

(訳補) 本記述は第1世代プロセス製品用です。基本的に第2世代プロセス製品では命令形式が異なっていることに注意してください。

3.4.3. チップ消去操作

プログラム用フラッシュメモリに新しい内容が書かれる前に、そのメモリは消去されなければなりません。消去なしにはフラッシュメモリ内の0へのビット書き込みだけが可能で、選択的な1へのビット設定はできません。メモリ消去は「チップ消去」命令で実行されます。この命令はプログラム用フラッシュメモリとEEPROMの両方の全メモリ内容を消去します。

メモリの消去完了後にだけ、施錠ビットが消去されます。この方法は全データが完全に消去されてしまうまでメモリ内のデータが安全を維持することを保証します。

チップ消去後、全メモリ内容は\$FFとして読めます。

チップ消去を終えるのはリセット信号線の一時的な開放による方法だけです。

表3-11. プログラム用フラッシュメモリとEEPROMの全内容消去例

動作	目的対象AVRへの送信 (MOSI)	目的対象AVRからの返り (MISO)
チップ消去	\$AC \$8x \$yy \$nn	\$zz \$AC \$8x \$yy
Nms待機		
消去終了のためのRESET開放		

(訳補) 本応用記述ではヒューズビットの読み書きについて触れていませんが、ファームウェア(avr910.asm)では万能命令と新万能命令によってAVR StudioのAVRProgから支援されています。

4. 簡単低価格実装書き込み器

本応用記述は実装書き込み器の全ての状況を検討していません。代わりにAT90S1200と少しのディスクリート部品だけを使う簡単低価格書き込み器の作成方法を示します。

この書き込み器は何れかのパソコンの何れかのシリアルポートに接続します。AT90S1200はハードウェアUARTがありませんが、タイマ/カウンタをデータクロックに使うことによってソフトウェアが半二重UARTを実行します。AT90S1200はソフトウェアで完全な主装置SPIを実行することによって目的対象AVRのプログラミングも実行します。

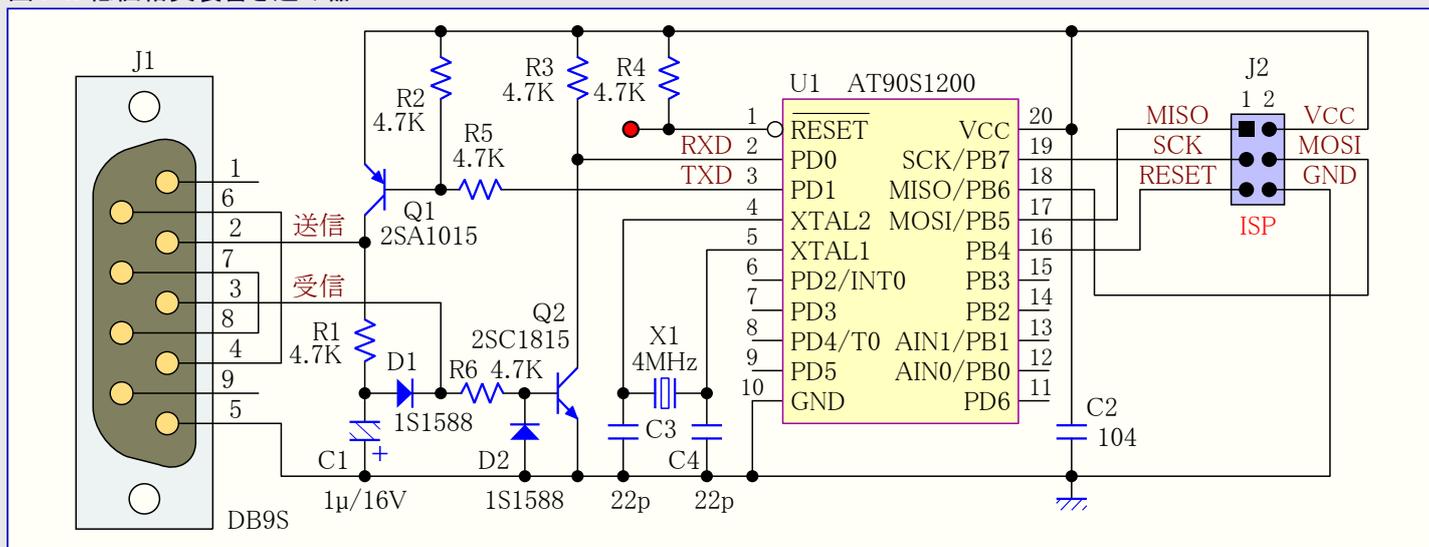
書き込み器の回路図は下図で見ることができます。AT90S1200への電力は目的対象システムから供給されます。パソコンとのシリアル通信で必要とされる負電圧は論理1受信(信号線の負電圧)時、C1に充電されます。

送信信号線はトランジスタQ1がOFFの時にこのC1からの負電圧が供給されます。これが送信信号線上で論理1を送ります。論理0(正電圧)は送信信号線へVCC(現実的にはVCC-0.2V)を接続する、Q1のONによって送信されます。

古いパソコンシステムのいくつかは論理0として+10V以下の電圧を受け付けられないシリアルポートを持っているかもしれません。然しながら、これは既存パソコンの多くで問題ではありません。

avr910.asmファイルはAT90S1200用のファームウェアを含みます。

図4-1. 低価格実装書き込み器



5. 部品表

表5-1. 部品表

記号	名称	製造者	数量	備考
U1	AT90S1200-4PC	Atmel	1	20ピン DIP AVR マイクロコントローラ
Q1	2SA1015	東芝	1	小信号用PNPトランジスタ
Q2	2SC1815	東芝	1	小信号用NPNトランジスタ
D1,D2	1S1588	東芝	2	小信号スイッチング用ダイオード
R1~R6	4.7kΩ	-	6	1/8W以上,20%,炭素皮膜固定抵抗器
C1	1µF/16WV	-	1	アルミ電解コンデンサ
C2	0.1µF/16V	-	1	セラミックコンデンサ
C3,C4	22pF/16V	-	2	セラミックコンデンサ
X1	4.000MHz	-	1	水晶発振子またはセラミック振動子
J1	DB9S	-	1	9ピンメスD-SUBコネクタ
J2	6Pヘッダ	-	1	6ピンヘッダ

(**訳注**) 回路図と部品表は一般用に変更しています。部品名は旧来からの代表的名称としています。発振用コンデンサ内蔵セラミック振動子を使う場合はC3,C4が不要です。

現在に於いては殆どのパソコンのRS232レシーバがRS232レベルとTTLレベルの兼用入力仕様になっているため、レベル変換回路が不要と考えられます。その場合、ファームウェアでRXD(PD0)とTXD(PD1)を論理反転すれば、論理反転回路も不要になります。そうすると、X1,C3,C4の発振回路とC2のパスコン以外には、RXDへの電流制限抵抗(10kΩ程度)と開放時の論理安定用のプルダウン抵抗(数10kΩ程度)と、TXDの電流制限抵抗(1kΩ程度)のみで実現できます。開放時のRXD論理安定用のプルダウン抵抗はピンに対してではなく、外部との信号線(即ちRXD電流制限抵抗のPD0側ではなくDB9Sコネクタ側)に対して行なうことに注意してください。

6. 改訂履歴

文書改訂	日付	注釈
8456A	-	初版文書公開
8456B	-	-
8456C	-	-
8456D	-	-
8456E	2008年8月	-
8456F	2016年11月	文書内で2ヶ所の”loose”を”lose”に修正 いくつかの微細な修正 新文書雛形

Atmel®、Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®、AVR®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。他の用語と製品名は一般的に他の商標です。

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。Atmelのウェブサイトに表示する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益と損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

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

© HERO 2021.

本応用記述はAtmelのAVR910応用記述(Rev.0943F-11/2016)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳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。