

AVR311 : I²C従装置としてのTWI部の使い方

要点

- TWI従装置用コード
- PhilipsのI²C規約に適合
- ハードウェアTWI部署使用
- 割り込み駆動転送
- 自身のアドレス認証での全休止形態からの起動復帰

序説

2線直列インターフェース(TWI)はPhilipsのI²C規約と互換です。このバスは電子機器に於ける集積回路間の簡単、強力かつ費用効率的な通信を与えるように開発されました。TWIの能力は同じバス上での128装置までのアドレス指定能力、調停、バスに多数の主装置を持つことの可能性を含みます。

ハードウェアTWI部署は利用可能なAVRデバイスの多くに内蔵されています。

この応用記述は完璧なドライバとこのドライバの使用例の形式でTWI従装置実装を記述します。ドライバは標準動作(<100kbps)と高速動作(<400kbps)の両方から従って転送を処理します。

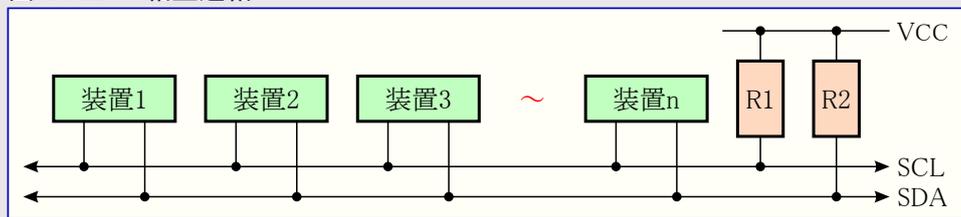
理屈

この章は一般的なTWIインターフェースとAVR上のTWI部署の短い説明を与えます。より多くの詳細情報についてはデータシートを参照してください。

2線直列インターフェース

2線直列インターフェース(TWI)は代表的なマイクロコントローラ応用に対して理想的に適合します。TWI規約はクロック(SCL)用の1つとデータ(SDA)用の1つの2つの双方向バス線だけを使用して最大128の個別アドレス指定可能な装置の相互連絡をシステム設計者に許します。バスを実装するのに必要な外部ハードウェアはTWIバス線の各々に対する単一プルアップ抵抗だけです。バスに接続された全ての装置は個別のアドレスを持ち、バス衝突を解決する仕組みはTWI規約に内在しています。

図1. TWIバス相互連結



TWIバスはバスを制御を行う能力のある1つ以上の装置を接続することができる複数主装置バスです。主装置だけがSCLとSDAの両線を駆動することができ、従装置はSDA線にデータを出すことだけを許されます。



8ビット **AVR**[®]
マイクロコントローラ

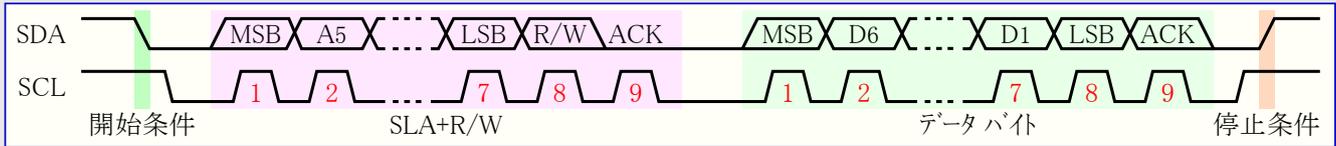
応用記述

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. 2565D-08/09, 2565DJ2-02/14

データ転送は常にバス主装置によって始められます。SCLがHighの間のSDA線上のHighからLowへの遷移が**開始条件**(または**再送開始条件**)に定義されます。

図2. TWIのアドレスとデータの packets 形式



開始条件は常に(固有の)7ビット従装置アドレスとその後のデータ方向ビットが後続します。アドレス指定された従装置は1クロック周期の間、SDAをLowに保持することによって直ぐに応答します。主装置がどんな(肯定)応答も受信しなかった場合、その転送は終了されます。データ方向ビットに依存して、主装置または従装置は直ぐにSDA線へ8ビットのデータを送信します。そして受信する装置はデータに対して応答します。主装置によって**再送開始条件**や**停止条件**が発行される前に単一方向で複数バイトを転送することができます。転送は主装置が**停止条件**を発行する時に終了されます。**停止条件**はSCLがHighの間のSDAでのLowからHighへの遷移によって定義されます。

いくつかの他の機能が実行されてしまうまで、従装置がやって来るデータを処理することができない場合、主装置に待ち状態を強制するためにSCLをLowに保持することができます。

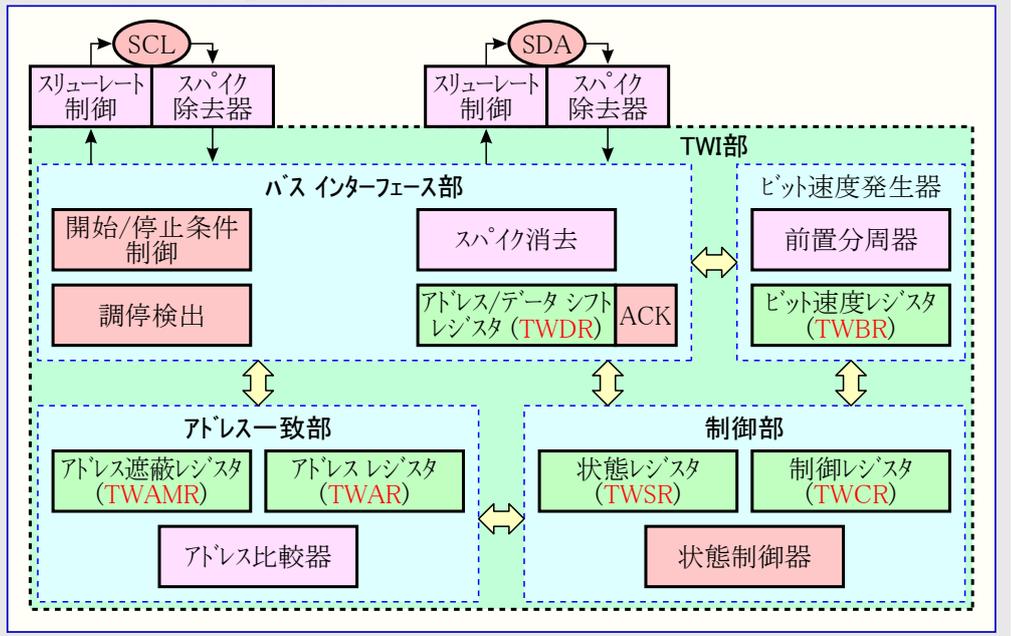
TWIバス上に送信される全てのデータパケットは1バイトのデータと応答ビットから成る9ビット長です。データ転送の間、主装置がクロックと**開始条件**、**停止条件**を生成し、一方受信部は受信に応答する責任があります。**肯定応答(ACK)**は第9SCL周期中に受信部がSDA線をLowに引っ張ることによって合図されます。受信部がSDA線をHighのままにするなら、**否定応答(NACK)**が合図されます。

AVR TWI部署

TWI部署は図3.で示されるように多数の補助部署から成ります。赤文字のレジスタ名の(訳注:原書は「太線で描かれた」)全てのレジスタがAVRのデータバスを通して入出力可能です。

(訳注) 図は代表的な構成に置き換えています。初期のTWI部署にはアドレス遮蔽レジスタ(TWAMR)が存在しません。一部のデバイスには関連する付加機能とそれらに対する付加レジスタが存在します。

図3. AVR内のTWI部署の概要



制御部

AVRのTWI部署は主装置と従装置の両方の動作で動くことができます。動作種別はTWI制御レジスタ(TWCR)の或るビットの使用とTWI状態レジスタ(TWSR)のTWI状態符号によって区別されます。

予め定義されている状態符号の組がTWI事象発生時にTWIがなり得る様々な状態を網羅します。状態符号は主装置と従装置の符号、更に送信と受信に関連する符号に分けられます。バスの異常とアイドルに対する状態符号も存在します。

TWI部署は状態遷移機構として動き、事象駆動です。TWIアドレスが後続する**開始条件**が従装置のTWIアドレスレジスタ(TWAR)内のアドレスと一致した場合、TWI割り込み要求(TWINT)フラグが設定(1)され、(全体割り込みとTWI割り込みが許可されていれば)対応する割り込みの実行に帰着します。従装置のファームウェアはTWSRの状態符号を読むことによって反応し、そしてそれに応じて返答します。全てのTWI事象がTWINTフラグを設定(1)し、そしてファームウェアはTWSR内の状態符号に基いて応答しなければなりません。

TWINTフラグが設定(1)されている限り、SCL線はLowに保持されます。これはTWI転送の継続を許す前にその作業の完了を応用ソフトウェアに許します。

TWINTフラグは以下の状況で設定(1)されます。

- TWIが**開始条件/再送開始条件**を送出した後
- TWIが**SLA+R/W**を送信した後
- TWIがアドレスバイトを送信した後
- TWIが調停を失った(に敗れた)後
- TWIが一斉呼び出しましたは自身アドレスによってアドレス指定された後
- TWIがデータバイトを受信した後
- 従装置として未だアドレス指定されている間に**停止状態**または**再送開始条件**が受信された後
- 不法な**開始条件**または**停止条件**のためにバス異常が起きた時

ビット速度発生器

ビット速度発生器部は主装置動作で動いている時にSCLの周期を制御します。SCL周期はTWIビット速度レジスタ(TWBR)とTWI状態レジスタ(TWSR)内の前置分周器選択(TWPS1,0)ビットでの設定によって制御されます。従装置動作はビット速度と前置分周器の設定に依存しませんが、従装置のCPUクロック周波数はSCL周波数よりも最低16倍高くなければなりません。表1.は標準速と高速のTWI転送に対する最低CPUクロック速度を示します。

表1. SCL周波数に対する最低CPUクロック周波数

CPUクロック周波数(MHz)	SCL周波数(kHz)
>6.4	400
>1.6	100

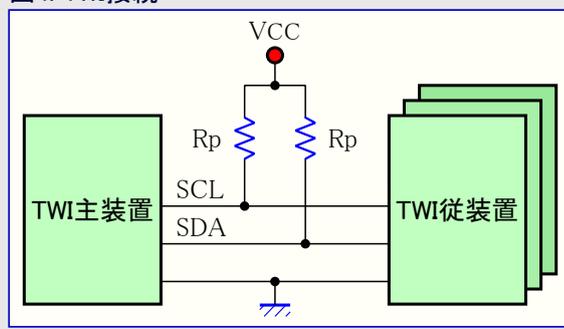
SCLピンとSDAピン

両TWI線(SDAとSCL)は双方向で、従ってTWIバスに接続する出力はオープンドレインまたはオープンコレクタでなければなりません。各線はプルアップ抵抗経路で供給電圧に接続されなければなりません。そして線を駆動する接続された装置が何も無い時に線は論理1(High)で、1つ以上が線をLowに駆動する場合に論理0(Low)です。

出力駆動部はスローレイトリミタを含みます。入力段は50nsよりも短い尖頭(雑音)を除去する尖頭除去部を含みます。データシートの「入出力ポート」章で記述されるように、SCLとSDAのピンに対応するPORTビットを設定(1)することによって、AVRのポート内の内部プルアップを許可することができることに注目してください。内部プルアップはいくつかのシステムで外部抵抗の必要をなくすることができます。

図4.はTWI部をTWIバスに接続する方法を示します。Rpの値はVCCとバス容量に依存します(代表的には4.7kΩ)

図4. TWI接続



アドレス一致部

アドレス一致部は従装置動作でだけ使用され、受信したアドレスバイトがTWIアドレスレジスタ(TWAR)内の7ビットアドレスに一致するかを調べます。アドレス一致で、制御部はそれを知らされ、取られるべき正しい動きを可能にします。TWIはTWI制御レジスタ(TWCR)の設定に依存して、そのアドレスに応答するかもしれないし、しないかもしれません。

TWIアドレス遮蔽レジスタ(TWAMR)を持つデバイスでは遮蔽されたアドレスの1群に応じることができます。

例え全ての休止形態に於いてTWIへのクロックシステムがOFFにされても、インターフェースはクロック元としてTWIバスクロックを使用することにより、一斉呼び出しアドレスまたは自身の従装置アドレスに未だ応答することができます。そしてデバイスは休止形態から起き、TWIは起動復帰中とTWI割り込み要求(TWINT)フラグが解除(0)されるまでSCLクロックをLowに保持します。

バスインターフェース部

この部署はデータとアドレスのソフトレジスタ(TWDR)、開始条件/停止条件制御器、調停検出ハードウェアを含みます。TWDRは送信されるべきアドレスまたはデータのバイト、または受信したアドレスまたはデータのバイトを含みます。加えて、送信するまたは受信した(N)ACKビットを含む(内部)レジスタも含みます。休止形態から起きた後のTWDRの内容が未定義なことに注意してください。換言すると、多数の従装置を支援する装置に於いて、その装置を休止形態から起すことを起動した従装置アドレスがどれかを判断するのにTWDRを使用することは不可能です。

開始条件/停止条件制御器は開始条件、再送開始条件、停止条件の生成と検出に関する責任があります。例えAVR MCUが休止形態の1つの時でも、開始条件/停止条件制御器は開始条件と停止条件を検知することができ、主装置によってアドレス指定されている場合にMCUを起すことを許します。TWIが主装置として転送を始めた場合、調停が進行中かの判断を試みる転送を調停検出ハードウェアが継続的に監視します。TWIが調停を失った(敗れた)場合、制御部はそれを通知されます。そして正しい動きを取ることができ、適切な状態符号が生成されます。

実装

この応用記述で実装するコードは純粋な従装置ドライバです。TWI部署は主装置動作も支援します。主装置ドライバの試供品については「AVR315:I²C主装置としてのTWI部の使い方」をご覧ください。主装置と従装置のドライバは主装置と従装置のドライバを結合して1つに合成できますが、それはこの応用記述の範囲ではありません。

従装置ドライバのCコードは3つのファイルから成ります。

- TWI_slave.cまたはTWI_multi_slave.c
- TWI_slave.h
- main.c

main.c内にドライバ使用法の例があります。TWI_slave.hファイルが主応用内にインクルードされなければならず、これは全ての関数宣言と全てのTWI状態符号に関する定義を含みます。TWI状態符号定義は異常メッセージの評価と適切な動きを取るのに使用することができます。TWI_slave.cとTWI_multi_slave.cのファイルは全てのドライバ関数を含みます。

いくつかのデバイスは多数のTWI従装置アドレスに応答することをデバイスで可能にする付加的なTWIアドレス遮蔽レジスタ(TWAMR)を持ちます。ここで記述された標準実装のその逃え版が応用記述付属物に含まれています。

関数

ドライバはTWI割り込み処理ルーチンと各種関数から成ります。全ての関数はドライバファイルの範囲外での使用が可能です。けれども、これらのいくつかはドライバそれ自身によって内部的にも使用されます。ドライバ内の全ての関数が表2で一覧にされます。IARコンパイラでコンパイルされた関数に対する実際のコード量は表4で一覧にされます。

表2. TWI従装置ドライバに於ける関数の説明

関数名	説明
void TWI_Slave_Initialize (uchar TWI_ownAddress)	TWI従装置をその初期待機状態に設定するにはこの関数をお願いします。TWI初期化後に主応用で割り込みを許可することを忘れないでください。従装置アドレスと一斉呼び出しでの起動必要条件の両方を同じバイト内で渡してください。この関数呼び出し時に例えばこの表記法を使用してください。 <code>TWI_Slave_Initialize((TWI_slaveAddress<<TWI_ADR_BITS) (TRUE<<TWI_GEN_BIT));</code> TWI部署はどの要求に於いてもNACKに形態設定されます。TWIを開始するにはTWI_Start_Transceiver関数を使用してください。
void TWI_Start_Transceiver_With_Data (uchar *msg, uchar msgSize)	予め用意されたメッセージを送る、または受信部に送受信部を始動するにはこの関数をお願いします。SLA+Wが受信された時に送るべきデータへのポインタを含めてください。データはTWI緩衝部に複製されます。何バイトが送られるべきかも含めてください。主装置関数と同様ではなく、アドレスバイトがメッセージ緩衝部に含まれないことに注意してください。関数はTWI割り込み処理ルーチンが直前操作を完了し、そして次の操作を初期化して戻るまで実行を差し控えます(内部待機)。
void TWI_Start_Transceiver(void)	新規送信データの指定なしに送受信部を始動するにはこの関数をお願いします。送信再開、または受信部にただ送受信部を始動するのに有用です。ドライバは直前に送受信緩衝部に置かれたデータを再使用します。関数はTWI割り込み処理ルーチンが直前操作を完了し、そして次の操作を初期化して戻るまで実行を差し控えます(内部待機)。
uchar TWI_Transceiver_Busy(void)	TWI割り込み処理ルーチンが転送多忙かを調べるにはこの関数をお願いします。
uchar TWI_Get_State_Info(void)	直前操作の状態情報を取得するにはこの関数をお願いします。関数はTWI割り込み処理ルーチンが直前操作を完了するまで実行を差し控えます(内部待機)。異常がある場合、関数はTWI状態符号を返します。
uchar TWI_Get_Data_From_Transceiver (uchar *msg, uchar msgSize)	TWI送受信緩衝部から受信したデータを読み出すにはこの関数をお願いします。換言すると、TWI送受信部にデータを取得させるため、最初にTWI_Start_Transceiver関数をお願いします。そしてそれらが到着した時のデータを取得するためにこの関数を走らせてください。関数呼び出しに於いてデータを置く場所へのポインタと取得するバイト数を含めてください。関数はデータを読み出して戻る前に、TWI割り込み処理ルーチンが直前操作を完了するまで実行を差し控えます(内部待機)。異常がある場合、関数はTWI状態符号を返します。
_interrupt void TWI_ISR(void)	この関数は割り込み処理ルーチン(ISR)で、TWI事象が起きた時に必ず起動されるTWI割り込み時に自動的に呼ばれます。この関数は主応用から直接呼ばれるべきではありません。

表3. 送受信部最終操作からの状態情報を含むバイト変数説明 (バイト内のビット領域として利用可能)

TWI_statusReg	説明
TWI_statusReg.lastTransOK	操作が成功裏に完了したとき、1に設定されます。
TWI_statusReg.RxDataInBuf	この設定内容はlastTransOKが1に設定されている場合にだけ有効です。データが受信されてしまい、送受信緩衝部に格納されているとき、1に設定されます。
TWI_statusReg.genAddressCall	この設定内容はRxDataInBufが1に設定されている場合にだけ有効です。受信が一斉呼び出しのとき、1に設定されます。換言すると、0の場合にそれがアドレス一致。

図5.はドライバを通るTWIインターフェースに対するデータの受信と送信の処理の流れ図を示します。データは引数を通して関数に渡され、一方動作状況は状態全体変数を通して利用可能です。図6.はTWIドライバそれ自身に関する流れ図を含みます。TWI割り込み処理ルーチンに於ける各事象/状況に対する動きのより多くの詳細説明は図7.の流れ図で得られます。図7.で左側の列は割り込み移行時にTWI状態遷移機構がなり得る各種の状況/事象を含みます。状態切り換え器は割り込み呼び出しの原因に依存する各種動作を実行します。

送受信部は1つの転送緩衝部だけを使用します。主装置からのメッセージは常にこの緩衝部で処理されます。換言すると、主装置書き込みは内容を上書きし、一方主装置読み込みはこの緩衝部の内容を送信します。

TWI_Start_Transceiver_With_Data関数呼び出し時、メッセージ全体が送受信緩衝部内に複製されます。そして送受信部を開始するためにTWI割り込みを許可します。その後、割り込みは転送を完全に処理し、転送完了時または異常状態発生の場合に自身を禁止します。ドライバはこのように転送完了を調べるのに割り込み許可ビットをポーリングすることができます。主応用はTWI送受信部が多忙でない間にだけ、送受信部全体変数のアクセスを許されます。割り込みは結果としての異常状態を関数呼び出しを通して主応用で利用可能な変数に格納します。

各送受信部操作後にドライバがTWI部署を不動状態にすることに注意してください。これはどの新規要求への応答にも先立って主装置からのメッセージを読んで解釈することを応用に許すためです。TWI従装置が再開される前にやって来る主装置からの全ての新規メッセージは従ってNACKにされます。故に主装置は次の送出前に応答のための十分な時間を従装置に与えることが重要です。

図5.の“組み合わせ”流れ図で異常処理符号を追加する場所が示唆されます。この例にこれが実装されない場合、その異常状態が異常な送信を扱うようになり得る従装置送受信部の再開を引き起こすことに注意してください。

図5. 応用からのTWIドライバ呼び出し

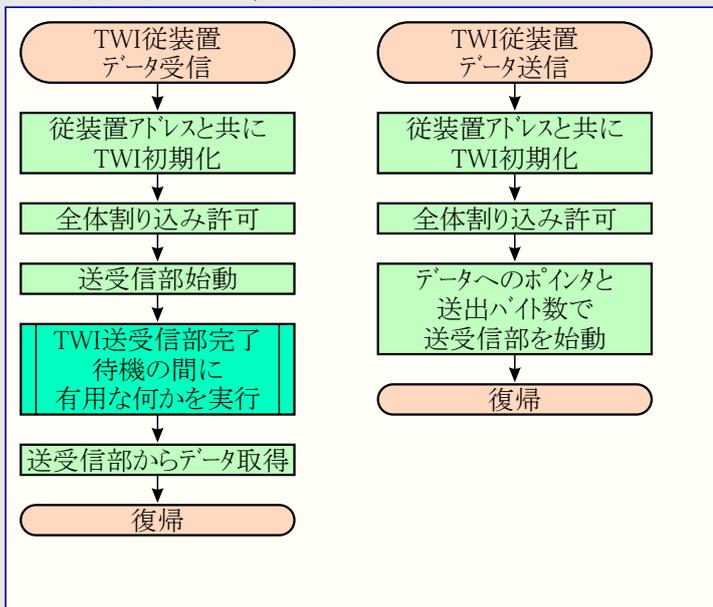


表4. IAR3.10コンパイラでのコード量

TWI従装置関数	量(バイト)
TWI_Slave_Initialize()	12
TWI_Transceiver_Busy()	6
TWI_Get_State_Info()	12
TWI_Start_Transceiver_With_Data()	68
TWI_Start_Transceiver()	8
TWI_Get_Data_From_Transceiver()	62
TWI_ISR()	162
(全体総計)	335

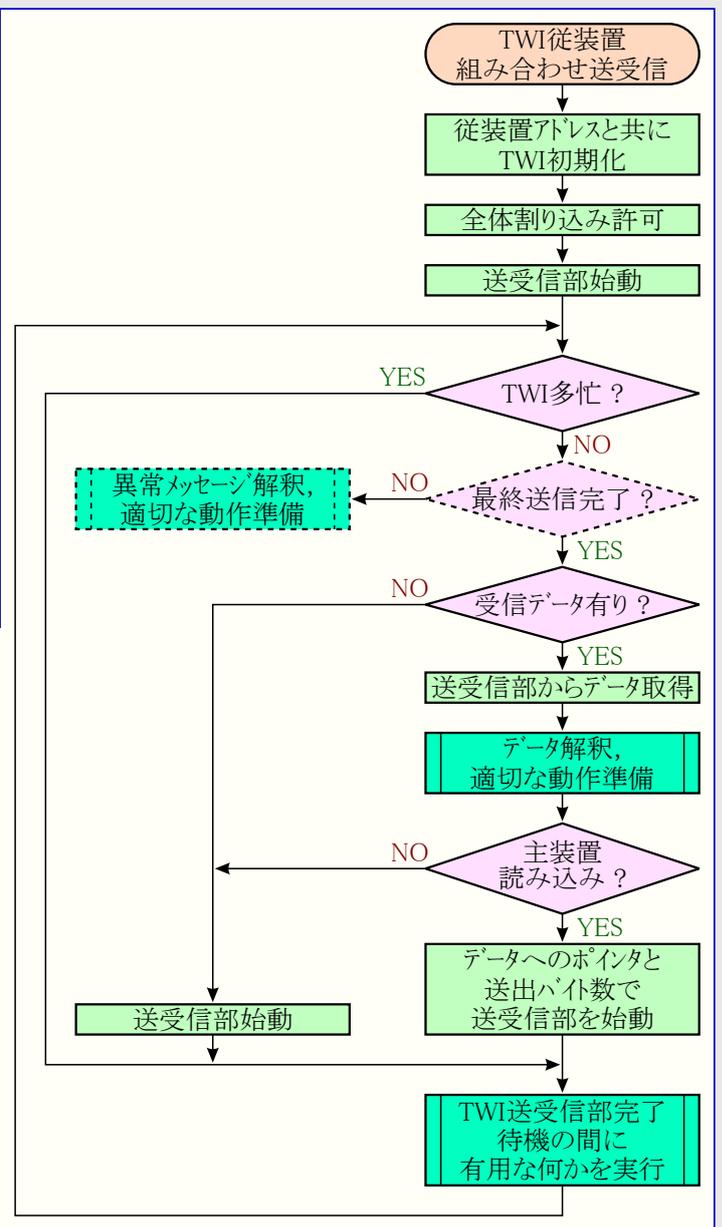


図6. TWIドライバ関数

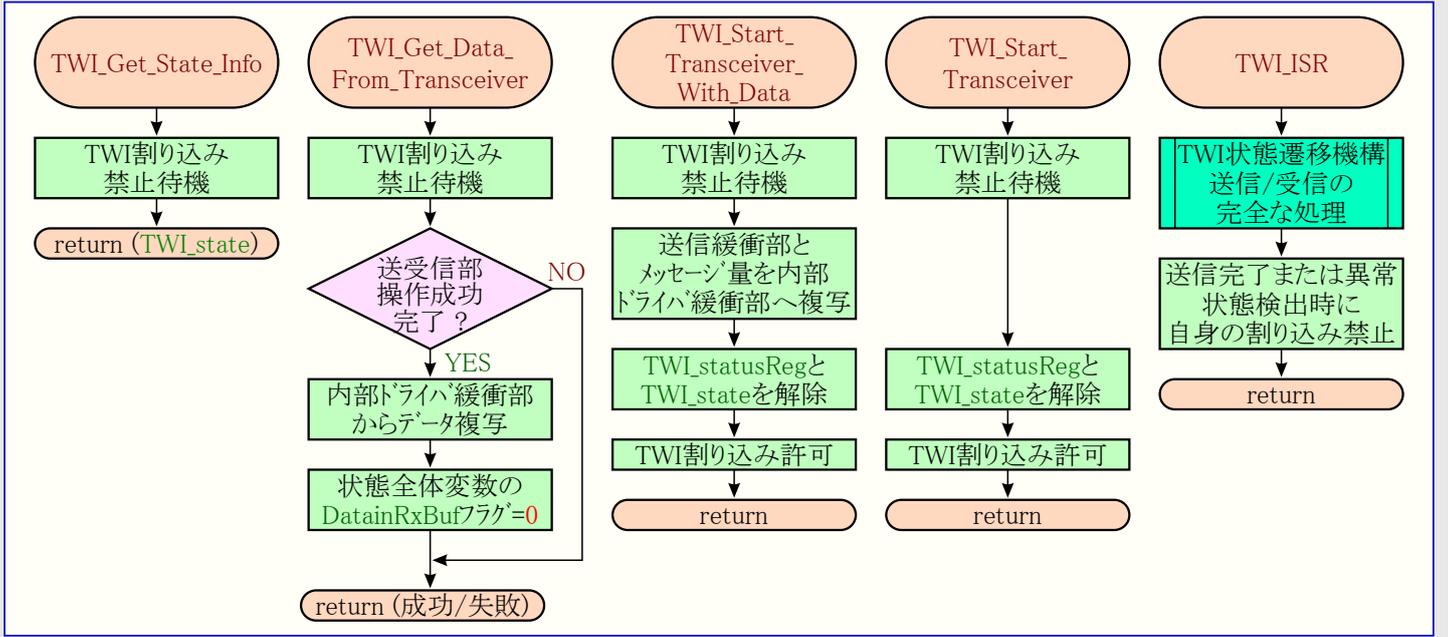
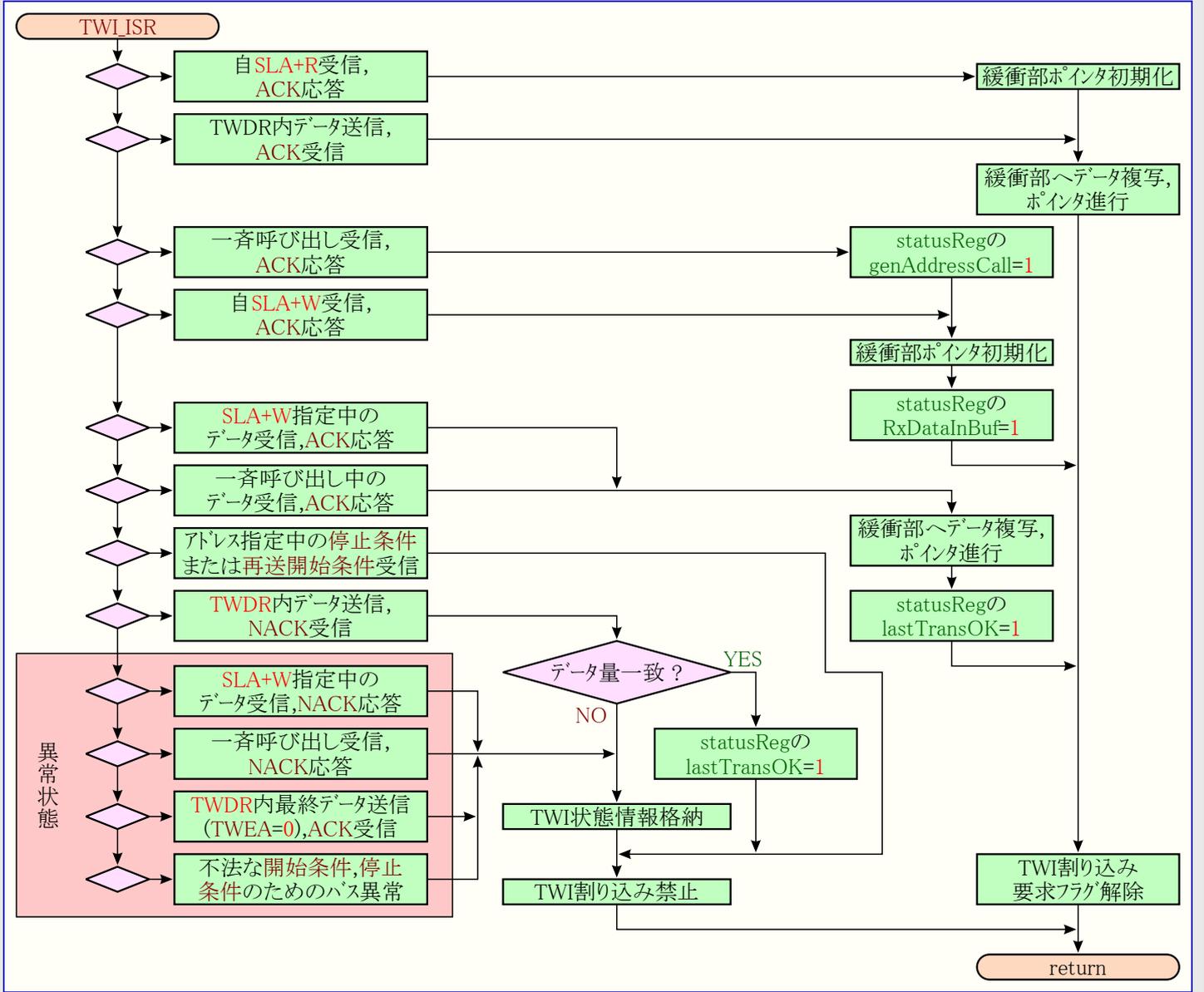
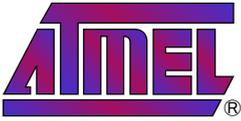


図7. TWI割り込み処理ルーチン





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. 全権利予約済 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付属物です。他の用語と製品名は一般的に他の商標です。

© HERO 2014.

本応用記述はATMELのAVR311応用記述(doc2565.pdf Rev.2565D-08/09)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。