AVR311：I²C従装置としてのTWI部の使い方

要点

- TWI従装置用Cｺｰﾄﾞ
- PhilipsのI²C規約に適合
- ﾊｰﾄﾞｳｪｱTWI単位部使用
- 割り込み駆動転送
- 自身のｱﾄﾞ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。このﾊﾞｽは電子機器に於ける集積回路間の簡単、強力且つ費用効率的な通信を与えるように開発されました。TWIの能力は同じﾊﾞｽ上での128装置までのｱﾄﾞﾚｽ指定能力、調停、ﾊﾞｽに多数の主装置を持つことの可能性を含みます。

ﾊｰﾄﾞｳｪｱTWI単位部は利用可能なAVRﾃﾞﾊﾞｲｽの多くに内蔵されています。

この応用記述は完璧なﾄﾞﾗｲﾊﾞとこのﾄﾞﾗｲﾊﾞの使用例の形式でTWI従装置実装を記述します。ﾄﾞﾗｲﾊﾞは標準動作(<100kbps)と高速動作(<400kbps)の両方に従って転送を処理します。

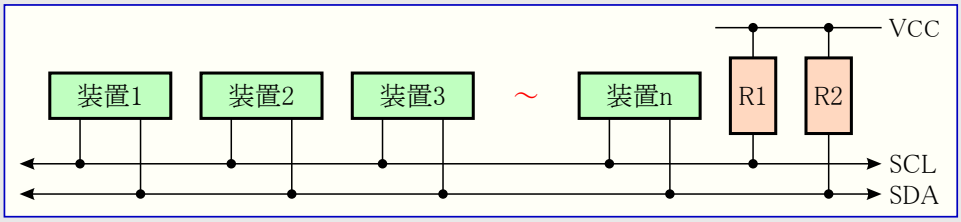
理屈

この章は一般的なTWIｲﾝﾀｰﾌｪｰｽとAVR上のTWI単位部の短い説明を与えます。より多くの詳細情報についてはﾃﾞ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つとﾃﾞｰﾀ(SDA)用の1つの2つの双方向ﾊﾞｽ線だけを使って最大128の個別ｱﾄﾞ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。ﾊﾞｽを実装するのに必要な外部ﾊｰﾄﾞｳｪｱはTWIﾊﾞｽ線の各々に対する単一ﾌﾟﾙｱｯﾌﾟ抵抗だけです。ﾊﾞｽに接続された全ての装置は個別のｱﾄﾞﾚｽを持ち、ﾊﾞ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ﾊﾞｽ相互連結



TWIﾊﾞｽはﾊﾞ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ﾊﾞｽです。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線にﾃﾞｰﾀを出すことだけを許されます。

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

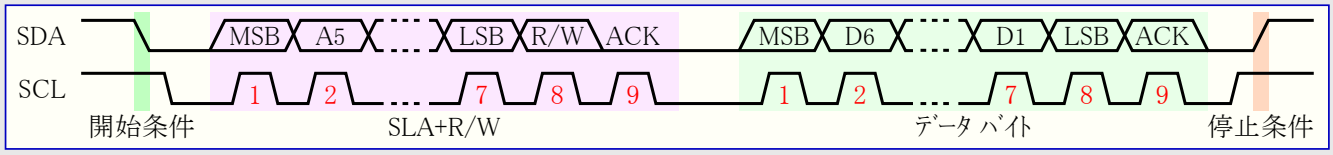
応用記述

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, 2565DJ3-04/21

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パケット形式



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。

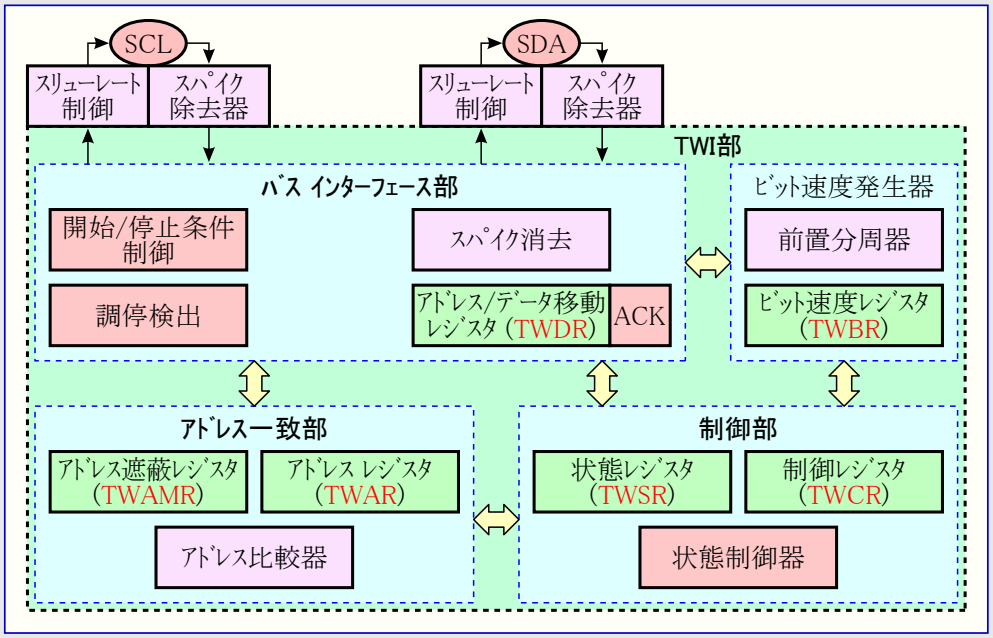
TWIバス上に送信される全てのデータ パケットは1バイトのデータと応答ビットから成る9ビット長です。データ転送の間、主装置がクロックと開始条件、停止条件を生成し、一方受信部は受信に応答する責任があります。肯定応答(ACK)は第9SCL周期中に受信部がSDA線をLowに引っ張ることによって合図されます。受信部がSDA線をHighのままにするなら、否定応答(NACK)が合図されます。

AVR TWI単位部

TWI単位部は図3.で示されるように多数の補助単位部から成ります。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単位部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単位部の概要



制御部

AVRのTWI単位部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単位部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ﾌﾗｸﾞ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がｱﾄﾞﾚｽ ﾊﾞｲﾄを送信した後
- TWIが調停を失った(に敗れた)後
- TWIが一斉呼び出しまたは自身ｱﾄﾞﾚｽによってｱﾄﾞﾚｽ指定された後
- TWIがﾃﾞｰﾀ ﾊﾞｲﾄを受信した後
- 従装置として未だｱﾄﾞ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ﾊﾞｽ異常が起きた時

ﾋﾞｯﾄ速度発生器

ﾋﾞ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ﾋﾞｯﾄ速度ﾚｼﾞｽﾀ(TWBR)とTWI状態ﾚｼﾞｽﾀ(TWSR)内の前置分周器選択(TWPS1,0)ﾋﾞｯﾄでの設定によって制御されます。従装置動作はﾋﾞ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

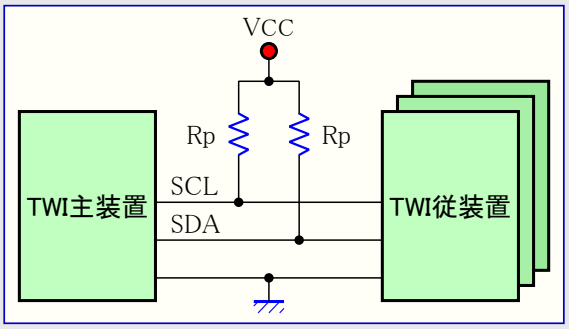
SCLﾋﾟﾝとSDAﾋﾟﾝ

両TWI線(SDAとSCL)は双方向で、従ってTWIﾊﾞｽに接続する出力はｵｰﾌﾟﾝ ﾄﾞﾚｲﾝまたはｵｰﾌﾟﾝ ｺﾚｸﾀでなければなりません。各線はﾌﾟﾙｱｯﾌﾟ抵抗経由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ない時に線は論理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。

出力駆動部はｽﾘｭｰﾚｰﾄ制限器を含みます。入力段は50nsよりも短い尖頭(雑音)を除去する尖頭消去部を含みます。ﾃﾞｰﾀｼｰﾄの「**入出力ﾎﾟｰﾄ**」章で記述されるように、SCLとSDAのﾋﾟﾝに対応するPORTﾋﾞｯﾄを設定(1)することによって、AVRのﾊﾟｯﾄﾞ内の内部ﾌﾟﾙｱｯﾌﾟを許可することができることに注目してください。内部ﾌﾟﾙｱｯﾌﾟ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ﾊﾞｽに接続する方法を示します。Rpの値はVCCとﾊﾞｽ容量に依存します(代表的には4.7kΩ)

図4. TWI接続



ｱﾄﾞﾚｽ一致部

ｱﾄﾞﾚｽ一致部は従装置動作でだけ使われ、受信したｱﾄﾞﾚｽ ﾊﾞｲﾄがTWIｱﾄﾞﾚｽ ﾚｼﾞｽﾀ(TWAR)内の7ﾋﾞｯﾄ ｱﾄﾞﾚｽに一致するかを調べます。ｱﾄﾞﾚｽ一致で、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制御ﾚｼﾞｽﾀ(TWCR)の設定に依存して、そのｱﾄﾞ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ｱﾄﾞﾚｽ遮蔽ﾚｼﾞｽﾀ(TWAMR)を持つﾃﾞﾊﾞｲｽでは遮蔽されたｱﾄﾞﾚｽの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ﾊﾞｽ ｸﾛｯｸを使うことにより、一斉呼び出しｱﾄﾞﾚｽまたは自身の従装置ｱﾄﾞﾚｽに未だ応答することができます。そしてﾃﾞﾊﾞ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)ﾌﾗｸﾞが解除(0)されるまでSCLｸﾛｯｸをLowに保持します。

ﾊﾞｽ ｲﾝﾀｰﾌｪｰｽ部

この単位部はﾃﾞｰﾀとｱﾄﾞﾚｽの移動ﾚｼﾞｽﾀ(TWDR)、開始条件/停止条件制御器、調停検出ﾊｰﾄﾞｳｪｱを含みます。TWDRは送信されるべきｱﾄﾞﾚｽまたはﾃﾞｰﾀのﾊﾞｲﾄ、または受信したｱﾄﾞﾚｽまたはﾃﾞｰﾀのﾊﾞｲﾄを含みます。加えて、送信するまたは受信した(N)ACKﾋﾞｯﾄを含む(内部)ﾚｼﾞｽﾀも含みます。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ﾄﾞﾚｽがどれかを判断するのにTWDRを使うことは不可能です。

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、主装置によってｱﾄﾞ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ﾄﾞ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ﾄﾞは純粋な従装置ﾄﾞﾗｲﾊﾞです。TWI単位部は主装置動作も支援します。主装置ﾄﾞﾗｲﾊﾞ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。主装置と従装置のﾄﾞﾗｲﾊﾞは主装置と従装置のﾄﾞﾗｲﾊﾞを結合して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。

従装置ﾄﾞﾗｲﾊﾞのCｺｰﾄﾞは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内にﾄﾞﾗｲﾊﾞ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ﾄﾞ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ｼﾞの評価と適切な動きを取るのに使うことができます。TWI.slave.cとTWI_multi_slave.cのﾌｧｲﾙは全てのﾄﾞﾗｲﾊﾞ関数を含みます。

いくつかのﾃﾞﾊﾞｲｽは多数のTWI従装置ｱﾄﾞﾚｽに応答することをﾃﾞﾊﾞｲｽで可能にする付加的なTWIｱﾄﾞﾚｽ遮蔽ﾚｼﾞｽﾀ(TWAMR)を持ちます。ここで記述された標準実装のその誂え版が応用記述付属物に含まれています。

関数

ﾄﾞﾗｲﾊﾞはTWI割り込み処理ﾙｰﾁﾝと各種関数から成ります。全ての関数はﾄﾞﾗｲﾊﾞ ﾌｧｲﾙの範囲外での使用が可能です。けれども、それらのいくつかはﾄﾞﾗｲﾊﾞそれ自身によって内部的にも使われます。ﾄﾞﾗｲﾊﾞ内の全ての関数が**表2.**で一覧にされます。IARｺﾝﾊﾟｲﾗでｺﾝﾊﾟｲﾙされた関数に対する実際のｺｰﾄﾞ量は**表4.**で一覧にされます。

表2. TWI従装置ﾄﾞﾗｲﾊﾞ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呼んでください。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。従装置ｱﾄﾞﾚｽと一斉呼び出しでの起動必要条件の両方を同じﾊﾞ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ってください。 TWI_Slave_Initialize((TWI_slaveAddress<<TWI_ADR_BITS) | (TRUE<<TWI_GEN_BIT)); TWI単位部はどの要求に於いてもNACKに構成設定されます。TWIを開始するにはTWI_Start_Transceiver関数を使ってください。
void TWI_Start_Transceiver_With_Data (uchar *msg, uchar msgSize)	予め用意されたﾒｯｾｰｼﾞを送る、または受信用に送受信部を始動するにはこの関数を呼んでください。SLA+Wが受信された時に送るべきﾃﾞｰﾀへのﾎﾟｲﾝﾀを含めてください。ﾃﾞｰﾀはTWI緩衝部に複写されます。何ﾊﾞ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、ｱﾄﾞﾚｽ ﾊﾞｲﾄがﾒｯｾｰｼﾞ緩衝部に含めら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)	新規送信ﾃﾞｰﾀの指定なしに送受信部を始動するにはこの関数を呼んでください。送信再開、または受信用にただ送受信部を始動するのに有用です。ﾄﾞﾗｲﾊﾞは直前に送受信緩衝部に置かれたﾃﾞ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呼んでください。
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呼んでください。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ﾃﾞｰﾀを読み出すにはこの関数を呼んでください。換言すると、TWI送受信部にﾃﾞｰﾀを取得させるため、最初にTWI_Start_Transceiver関数を呼んでください。そしてそれらが到着した時のﾃﾞ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ﾃﾞｰﾀを置く場所へのﾎﾟｲﾝﾀと取得するﾊﾞｲﾄ数を含めてください。関数はﾃﾞ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ﾊﾞｲﾄ変数説明 (ﾊﾞｲﾄ内のﾋﾞ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。ﾃﾞ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の場合にそれがｱﾄﾞﾚｽ一致。

図5.はﾄﾞﾗｲﾊﾞを通るTWIｲﾝﾀｰﾌｪｰｽに対するﾃﾞｰﾀの受信と送信の処理の流れ図を示します。ﾃﾞ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ﾄﾞﾗｲﾊﾞ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います。主装置からのﾒｯｾｰｼﾞ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、ﾒｯｾｰｼﾞ全体が送受信緩衝部内に複写されます。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。ﾄﾞﾗｲﾊﾞはこのように転送完了を調べるのに割り込み許可ﾋﾞｯﾄをﾎﾟｰﾘﾝｸﾞ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ﾄﾞﾗｲﾊﾞがTWI単位部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ｼﾞ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ｼﾞは従ってNACKにされます。故に主装置は次の送出前に応答のための充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。

表4. IAR3.10ｺﾝﾊﾟｲﾗでのｺｰﾄﾞ量

TWI従装置関数	量(ﾊﾞｲﾄ)
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

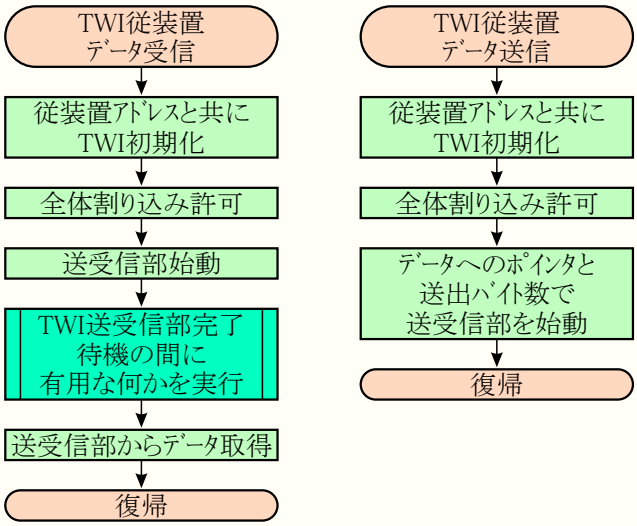


図5. 応用からのTWIﾄﾞﾗｲﾊﾞ呼び出し

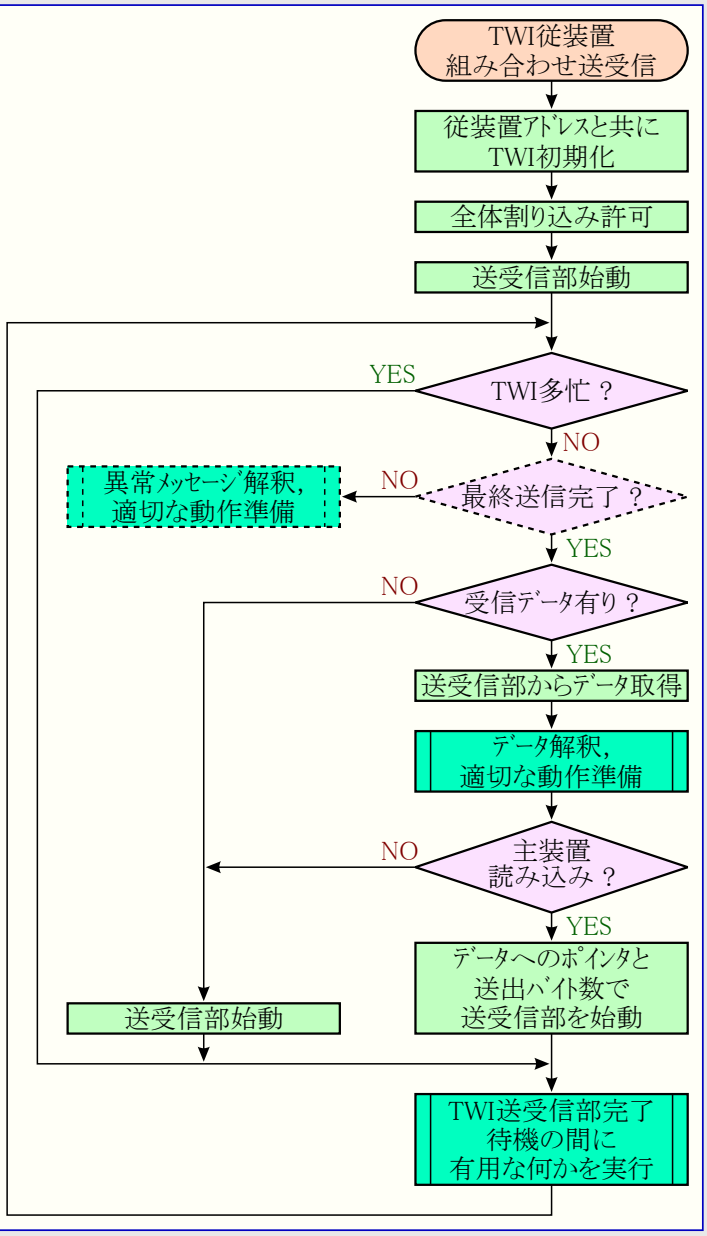


図6. TWIﾄﾞﾗｲﾊﾞ関数

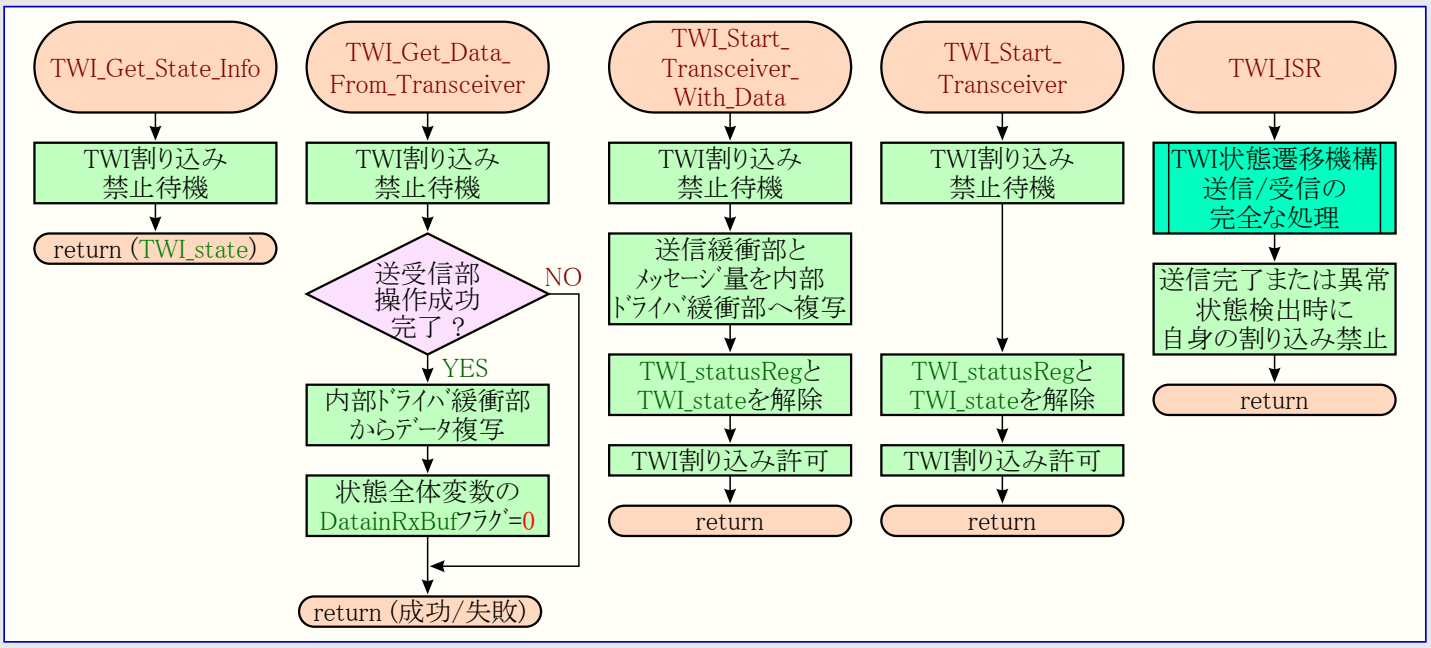
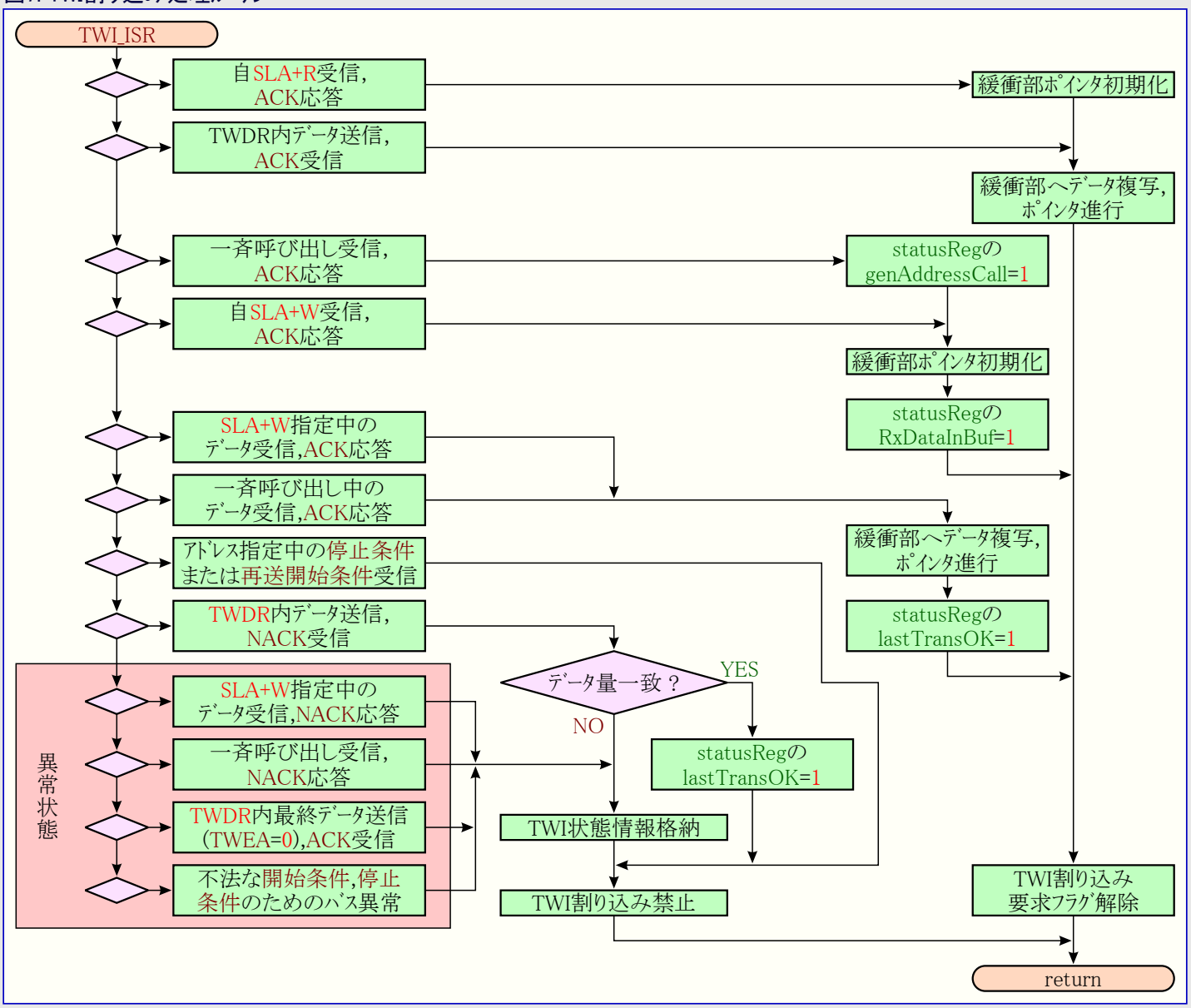


図7. TWI割り込み処理ルーチン



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
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ｳｪﾌﾞｻｲﾄ
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のｳｪﾌﾞ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. **不許複製** Atmel®、ﾛｺﾞ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* 2021.

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。