

AVR307 : USI部を使う半二重UART

要点

- 半二重UART通信
- 14.75MHzで最大230.4kbpsの通信速度
- 割り込み制御通信
- 8ビットデータ、パリティなし、1停止ビット
- データ緩衝処理

序説

ATtiny26, ATtiny2313, ATmega169のようなAVRデバイスに存在する多用途直列インターフェース(USI)はTWIとSPIの通信用に設計された通信単位部です。けれどもUSIはそれら2つの通信規格に制限されません。UART通信にも使うことができます。

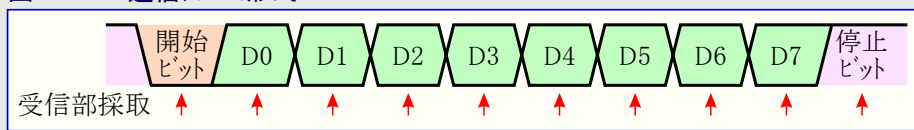
UART通信にUSIを使うことにより、ソフトウェアで実装されたUART通信に関連する不便のいくつかを避けることができます。USIを使うことにより、割り込みが通信を制御し、ビットが自動的に移動出力されます。このようにして通信によって専有される処理時間を減らし、通信中に応用の他の部分に対する処理時間を開放します。

この資料はUART通信に対してATtiny26のUSIを使う方法を記述します。送受信に関する通信ドライバ用ソースコードが提供されます。ドライバはコードの小変更によってATtiny2313, ATmega169のUSI単位部用に改造することができます。コードはデータ緩衝処理と、送信部と受信部の組み合わせで完全です。これらの機能は使用資源の最小化のために分割または削除することができます。

動作の理屈

UART規約は非同期直列通信規格です。フレーム形式は5～9ビットデータ、1つのパリティビット(任意選択)、1つまたは2つの停止ビットが後続する開始ビットとして指定されます。データは最下位ビット(LSB)が先に送られます。UART通信に関するフレーム形式は図1.で図解されます。

図1. UART通信フレーム形式



8ビット **AVR**[®]
マイクロコントローラ

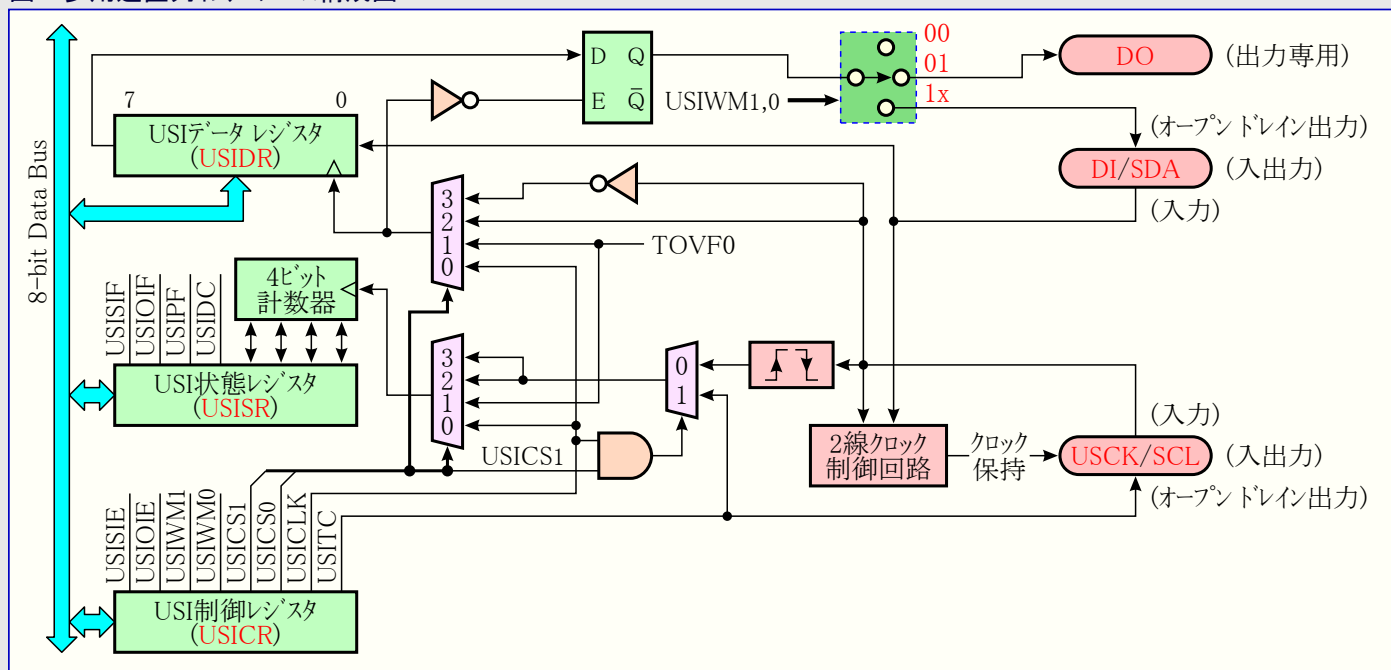
応用記述

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. 4300A-10/03, 4300AJ3-04/21

USIは基本的に入力と出力のピンに接続された8ビットの移動レジスタです。UART通信に使うとき、汎用計時器がデータビットの入出力クロック駆動を制御し、与えられたビットが移動入出力された時に専用のUSI計数器が割り込みを生成します。USIの構成図は図2.で見ることができます。

図2. 多用途直列インターフェース構成図



(訳補) 図中のTOVF0はATtiny26の場合です。これはタイマ/カウンタ0溢れまたは比較一致で、デバイスによって異なります。

UARTフレーム形式が合計7～14ビットで指定され、代表的なフレーム容量が10ビットなので、USI移動レジスタは代表的な形式に足りない2つのビット群になります。けれどもUSI単位部の機能を使ってこの制限を欺くことができます。

UART受信

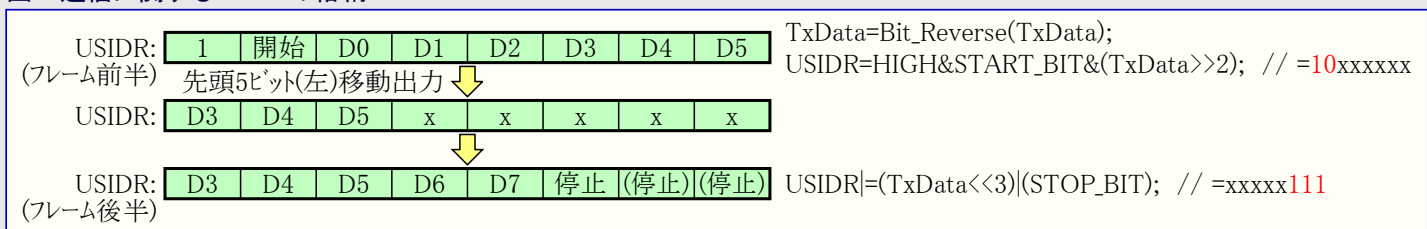
受信に対して示唆される解決策は開始ビットを検出し、そしてそれらの適切な時間間隔で8つのデータビットを採取することです(一方停止ビットを無視します)。最初のビット採取は半ビット間隔後に行われ、そしてこれは開始ビットの下降端から開始ビットの中央までの時間です。(その後の)継続的な採取はボーレートによって決められる1ビット幅の固定間隔で行われます。移動レジスタが8ビットしか保持できないため、開始ビットは移動レジスタに保持されませんが、これは最終データビットが移動入力される時に起きます。タイマ/カウンタ0の使用、遅延の制御、これによって使うビット速度を簡単にするため、この方法は次に最初のデータビットで最初の採取を実行することが好まれます。

AVRハードウェアUARTと、ソフトウェアまたはUSI単位部を使って実装されたUART通信間で入力データ採取法が異なることに注意すべきです。ハードウェアUARTは採取ビットの値を決めるのに3採取の“多数決”が用いられます。“多数決”はデータを不正にさせる雑音の可能性をなくします。完全なまたは部分的なソフトウェア実装の殆どでは各データビットの採取が1回だけ行われます。これはUART通信のUSI実装に対する場合もです。従って雑音がやって来るデータを不正にさせ得る可能性が増されます。この事実は受信に関してパリティ検査を使う誘引として見る事ができます。パリティ検査はこの資料で記述される受信ドライバに付加されていません。代替解決策はAtmelのウェブサイトでも入手可能なAVR350応用記述で記載されるようなメッセージ全体のCRC検査を使うことです。

UART送信

UART送信は僅かにもっと複雑で、フレームは2つに分割されなければなりません。USI移動レジスタの最上位ビット(MSB)はUSIデータ出力ピンに直接接続されています。従って正しいタイミングを保証するために、MSBは計時器が正しくデータの移動を開始するまでHigh状態を保たなければなりません。USI移動レジスタはHigh値、開始ビット、データの下位側6ビットを格納されます。そして送信が始まります。移動レジスタ内の最初の5ビットが移動出力されてしまった時に移動レジスタが再格納され、このようにしてフレームの残りが移動レジスタの内容に追加されます。送信に関するUSIデータレジスタ(USIDR)の格納/再格納の手順は図3.で図解されます。

図3. 送信に関するUSIDRの格納



USI動作種別

USIに対して各種通信動作種別または“線動作種別”を選択することができますが、USI単位部がTWIとSPIの通信用に設計されているため、UART通信専用の線動作種別がありません。選ぶことができる線動作種別は“3線”(SPI)または“2線”(TWI)です。UART通信に使われるべき動作種別は、これが独立したデータの入力と出力の線を持つ動作種別なので、SPI動作種別です。

両線動作種別に於いて、USI単位部はいわゆる**開始条件**割り込みを生成することができます。SPI動作種別に於ける**開始条件**割り込みはSCKピンでのレベル変化によって起動されますが、DI(データ入力)ピンの変化での起動復帰が望まれるので、ピン変化割り込みを使うのがより良好です。ピン変化割り込みは全ての休止形態からAVRを起すことができます。従ってピン変化割り込みはUARTフレームの開始ビット検出に使うことができます。UART受信に於ける開始ビットはこの方法でピン変化割り込みを生成し、その後は割り込みルーチンで受信を処理することができます。従ってSCKピンは必要とされず、他の目的に使うことができます。

ビット速度

直列通信に対するビット速度を指定するのに8ビットタイマ/カウンタ0が使われます。ATtiny26ではUSI移動レジスタを起動するのにタイマ/カウンタ0溢れを使うことができます。ATmega169では移動レジスタの起動に比較一致が使われます。従って直列通信のビット速度を生成する方法は計時器(タイマ/カウンタ0)を動かしてタイマ/カウンタ溢れ/比較一致事象がビット移動を起動させることです。

この資料ではビット移動を起動するのにATtiny26のUSIと従ってタイマ/カウンタ0溢れが使われます。1ビットの幅に合う計時器の値(基)は**式1**で決めることができます。

図4. 式1.

$$\text{1ビットの計時数} = \frac{\text{システムクロック周波数}}{\text{ポーレート} \times \text{計時器前置分周係数}}$$

$$\text{タイマ/カウンタ0(TCNT0)設定値} = 256 - \text{1ビットの計時数}$$

計時器の基(TCNT0設定値)が計時器の頂上値に近接する場合、計時器の基の丸め/切捨ては重要なタイミング誤差を誘引し得ます。ビット毎の周期数に比べて丸め誤差は可能な限り低いことが推奨されます。誤り無しの通信を保証するため、経験則として総合タイミング誤差が2%未満であるべきです。総合誤差はシステムクロックの精度とUARTビットタイミングの両方の合計です。

更に、計時器の基が計時器の頂上値に近い場合、そしてそれは高速通信使用時に起き、各計時器割り込み間がかなり小さな周期になります。これは充分早く/繰り返して割り込みを扱うことに於いて潜在的な問題を起こし得ます。計時器割り込みを供給するのに必要とする周期数、特に他の割り込みが起き得る場合、それによって計時器溢れ割り込みを妨げることに注意して考慮することが推奨されます。加えて重要な要素として割り込み遅延も考慮すべきで、これに応じて計時器の基を修正してください。割り込み遅延は計時器の基を再設定するために用いる周期数と共に、計時器の基の値に加えられなければなりません。これは実装章で記述されます。

計時器の基の一覧は**表1**で得られます。計時器の基は**式1**を用いて計算され、割り込み遅延に関して修正されていません。

表1. 各種システムクロックと通信速度に対するタイマ/カウンタ0(TCNT0)の基

システムクロック (MHz)	可能な通信速度(bps)						
	9600	14400	19200	28800	57600	115200	230400
1	152	187 (注3)	利用不可	利用不可	利用不可	利用不可	利用不可
1.84	64	128	160	192 (注3)	利用不可	利用不可	利用不可
2	48	117	152	187 (注3)	利用不可	利用不可	利用不可
3.69	208 (注2)	(注1,2)	64	128	192 (注3)	利用不可	利用不可
4	204 (注2)	221 (注2)	48	117	186 (注3)	利用不可	利用不可
7.37	160 (注2)	192 (注2)	207 (注2)	(注1,2)	128	192 (注3)	利用不可
8	152 (注2)	187 (注2)	204 (注2)	221 (注2)	117	187 (注3)	利用不可
11.06	112 (注2)	160 (注2)	184 (注2)	208 (注2)	64	160	利用不可
14.75	64 (注2)	128 (注2)	160 (注2)	192 (注2)	(注1,2)	128	192 (注3)
16	48 (注2)	117 (注2)	152 (注2)	187 (注2)	221 (注2)	117	187 (注3)

注1: タイマ/カウンタ0計時器の基は必要ありません。

注2: 計時器の前置分周器はCK/8です。

注3: 危険なタイミングを示します。

いくつかの通信速度に於いて計時器の基は0です。これらの場合では一旦計時器溢れが起きれば基が丸めによってそれ自身を“再設定”するため、計時器の基を再設定する必要がありません。従ってタイマ/カウンタ0溢れ割り込みはこれらの場合に禁止されます。

受信に対する初回計時器値(基)が表1.で指定される計時器の基と異なることに注意してください。受信に用いる初回計時器値は、初回採取が開始ビットまたは最初のデータビットのどちらで行うつもりかに依存して、0.5または1.5ビット幅の遅延を生成すべきです。これらは式2.と式3.で示されるように計算することができます。

図5. 式2.

タイマ/カウンタ0(TCNT0)設定値 = $256 - (1\text{ビットの計時数} \times 0.5)$

図6. 式3.

タイマ/カウンタ0(TCNT0)設定値 = $256 - (1\text{ビットの計時数} \times 1.5)$

各々の策は注意しなければならない制限を持ちます。非常に低いビット当たりの周期数の場合、2で割ることがUSIを受信用に設定するのにかかる時間よりも短い初回計時器値を生じるかもしれません。これに反して、ビット当たりの周期数が非常に高い場合、1.5倍の乗算が256よりも大きな初回計時器値を生じるかもしれません。

推奨される近似手段は(ビット当たりの周期数 \times 1.5) $>$ 256の場合に(ビット当たりの周期数 \times 0.5)を初回検査にすることであり得ます。

ATmega169については比較一致が用いられるので、計時器の基の再設定に関連する考慮が最小にされます。比較一致は先に記述された計時器の基が0の場合の状況と同様に、計時器の基の自動設定と等価です。

USI計数器

タイマ/カウンタ0がビット速度を決め、一方USI計数器がバイト/フレームレベルに於けるデータの送信を処理します。USI計数器はUSIDRで送受信されるビット数を数えるのに使われる4ビット計数器です。この計数器の頂上値は16です。USI計数器に対するクロックはUSIクロック元選択(USICS1,0)ビットとUSIクロック ストローブ(USICLK)ビットに依存します。USIがUART通信に使われる時にこれらのビットはUSICS1,0=01とUSICLK=x(無関係)にすべきです。これらの設定でUSI計数器とUSIDR移動レジスタはタイマ/カウンタ0溢れによってクロック駆動されます。従ってUSI計数器は送受信されるビット数を数えます。

受信

USI計数器は溢れ割り込みを生成することができ、そしてそれはビット数が受信されてしまったことを示すのに使えます。これを成し遂げるためにUSI計数器は頂上(16)-データビット数(代表的に8ビット)が予め格納されます。式4.と式5.をご覧ください。

図7. 式4.

USI計数器設定値 = 計数器頂上値 - (開始ビット + データビット)

図8. 式5.

USI計数器設定値 = 計数器頂上値 - (データビット)

計数器の頂上値は16、開始ビットは1、データビットは代表的に8(1バイトの容量)です。選択は初回計時器値が開始ビットを採取するか否かのどちらかに依存します。より多くの情報については3頁の「ビット速度」に於ける初回計時器値の吟味をご覧ください。

送信

送信ではUSIDRが(10ビット フレーム全体を送ることができるようにするのに)必要とする再格納を決めるのにUSI計数器溢れが使われます。この場合、USI計数器は計数器頂上値-半フレーム量が予め格納され、そしてこれは計数器が再格納される前に送信されるべきビット数です(図3.をご覧ください)。送信に対する初回USI計数器値は式6.によって示されるように決められます。

図9. 式6.

USI計数器設定値 = 計数器頂上値 - (半フレーム量)

2つ目の“半フレーム”がUSIDR内に格納された後、USI計数器は一度フレーム全体が送信されてしまったならば割り込みを生成するために再び予め格納されます。送信に関する2回目のUSI計時器値は式7.によって決められます。

図10. 式7.

USI計数器設定値 = 計数器頂上値 - (全体フレーム量 - 半フレーム量)

要約

ビット速度を制御するのにタイマ/カウンタ0が使われ、4ビットUSI計時器溢れがフレーム処理に用いられます。受信と送信の両方がタイマ/カウンタ0とUSI計数器を必要とするため、半二重通信にだけUSIを使うことが可能です。

USIデータ入力ピンでのピン変化割り込みは受信に於いてフレーム開始を検知するのに用いられます。

送信では送信されるフレームの各々に対してUSIデータ レジスタが2度格納されなければなりません。

実装

この応用記述で記載されるコードはUART通信用ドライブとして書かれています。コードは真の半二重通信を与える組み合わせた受信と送信を可能にする制御要素を含みます。ドライブの使用を簡単にするために通信緩衝部が実装されています。これはUART通信での並行的な実行を応用に許します。送信と受信の両方、または通信緩衝部に関して必要でないものがあれば、量に於いてドライブを最小化するために削除し、そして性能を最適化することができます。

コードは全最適化許可でIAR EWAVR 2.28Aを使って書かれ、そしてコンパイルされています。試験とデバッグはAVR Studio 4.07とICE 50を使って行われています。

通信緩衝部

受信と送信のルーチンはやって来るデータと出て行くデータの緩衝用に2ⁿの剰余でアドレス指定する循環緩衝を使います。緩衝容量はルーチンを使う前に定義されなければなりません。バイトでの緩衝容量でUART_RX_BUFFER_SIZEとUART_TX_BUFFER_SIZEの変数を設定してください。これらの変数が2のべき乗でなければならないことに注意してください。そうでなければ、コンパイラ異常メッセージで合図されるでしょう。

緩衝部の先頭と後尾の位置指示子を調査することにより、新しいデータが利用可能か、それは書き込み位置指示子が読み込み位置指示子と異なるか、を判断することができます。

際どいタイミングでの高速通信が必要とされる場合に受信緩衝容量を増やすことが推奨されます。さもなければ、データが緩衝部から読まれてしまったことを受信ドライブが確かめられない(緩衝部空き待ちをしない)ため、データが失われるかもしれません。

例コードに付加関数が追加されています。USI_UART_Data_In_Receive_Buffer関数は受信緩衝部が何のデータも含まない場合に0を返します。この関数はUSI_UART_Receive_Byte関数と対照的にやって来るデータを待ちませんが、緩衝部の状態を直ちに返します。

注: この関数は緩衝部内のバイト数を返しません。

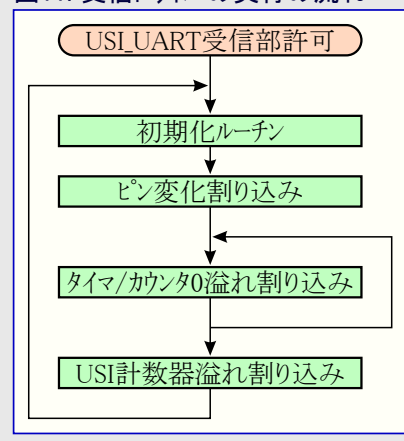
USI UART受信部

ドライブは4つの部分から成ります。

- ・ USI UART初期化
- ・ ピン変化割り込み処理ルーチン
- ・ タイマ/カウンタ0溢れ割り込み処理ルーチン
- ・ USI計数器溢れ割り込み処理ルーチン

これらの部分の各々は流れ図とそれらの流れ図に対する注釈によって、以下で記述されます。

図11. 受信ドライブの実行の流れ



USI_UART_Initialize_Receiver

この関数は受信用にデバイスを準備します。

USIは専用のUSI開始条件割り込みを持ちますが、これはSCK線と連携されます。UART通信に関しては独立したクロック線が使われず、受信はデータ入力(DI)線での動きとして検知されます。UARTフレームの開始ビットのために、最初の動きは常にバス上でのHighからLowへの遷移です。開始された送信を検知するためにピン変化割り込みが用いられます。従ってUSI_UART初期化ルーチンはやって来るデータを待つように、ピン変化割り込みを初期設定します。

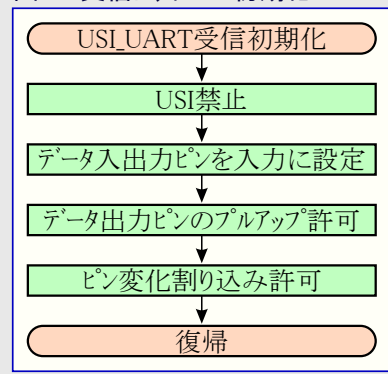
USIの機能はピン変化機能を無効にし、従ってUSI単位部は禁止されます。その他、DIとDOの両ピンが入力として定義されます。DIはこのピンでデータが予測されるため、DOはデータが意図せず移動出力されないことを保証するためです。

UART通信ピンのアイドルレベルはHighで、DOが送信中以外に入力なので、DOピンのプルアップが許可されます。

最後にピン変化割り込みが許可されます。

この初期化ルーチンで全体割り込みが許可されないことに注意してください。従ってこれは主ルーチンで行われるべきです。流れ図が図12.で示されます。

図12. 受信ドライブの初期化



IO_Pin_Change_ISR – 受信部

ピン変化割り込みの目的は一旦開始ビットの下降端が検出されたならばタイマ/カウンタ0を開始することです。開始ビットの下降端だけがピン変化割り込みを生成すべきなので、一旦開始ビットが検出されたならばピン変化割り込みは禁止されるべきです。USIが同じピン上のピン変化機能を侵略するため、これはUSIを許可することによって自動的に行われます。一旦完全なフレームが受信されてしまったならば、ピン変化割り込みはUSI計数器溢れ割り込み処理ルーチンで再び許可されます。

受信に関してUSIを使うために線動作種別が“3線動作”(SPI)に設定されます。USIクロック元はタイマ/カウンタ0溢れが選択されます。

USI計数器は必要なビット数が受信されてしまった時に溢れを生じるように初期化されます。

タイマ/カウンタ0は開始ビットの下降端から採取されるビットの中央までの遅延となる、初回遅延を生成するために必要な計時器値(基)を格納することによって初期化されます。フレーム形式に関しては図1をご覧ください。正確なタイミングに関してDIPピンが値を変える時からタイマ/カウンタ0が走行を開始する時までの遅延も含めることができます。この値は割り込み処理ルーチンでタイマ/カウンタ0をどれ位早く事前格納して開始できるかに依存します。この操作を割り込み処理ルーチンの先頭に置き、遅延を計算するために最終コードの擬似実行を行ってください。

最後に必要とする全ての割り込みが許可されます。タイマ/カウンタ0(TCNT0)値(基)が0の場合、タイマ/カウンタ0溢れ割り込みを許可する必要はありません。8頁の「Timer0_OVF_ISR – 送信部」項でこれの注釈をご覧ください(訳注:参照先に該当情報はありません)。

Timer0_OVF_ISR – 受信部

タイマ/カウンタ0溢れ割り込み処理ルーチンの目的はやって来るデータビットの採取を制御する計時器値(基)を再設定することです。割り込みルーチンは図14の流れ図によって説明されます。タイマ/カウンタ0溢れ割り込み処理ルーチンは受信部ドライバと送信部ドライバの両方に対して同様です。

計時器値(基)が0の場合(表1をご覧ください)、計時器値(基)を再設定しないので、溢れ割り込み処理ルーチンは必要とされません。タイマ/カウンタ0溢れ割り込みを無効にできるなら、UART通信のタイミングは他の割り込み元の存在による影響がより少なくなるでしょう。更にUART受信(と送信)でより少ない処理資源が使われます。

USI_Counter_Overflow_ISR – 受信部

受信ドライバではバイト全体が受信されてしまった時にUSI計数溢れが起動されます。これによって受信の完了を示し、次のフレームの受信に対して準備するために、受信したデータの受信緩衝部への移動とUSIの再初期化を処理します。割り込みルーチンは図15で可視化されます。

最初にタイマ/カウンタ0が停止されます。これはデータの更なる移動入力をさせないことを保証します。データが循環データ緩衝部へ移動されます。データは使われる前にビット順反転を必要とします。これは受信緩衝部から読み出される時か、またはこのどちらかで行うことができます。

受信部としてのUSIの再初期化はUSIの禁止とピン変化割り込みの許可を意味します。

図13. ピン変化割り込み処理ルーチン

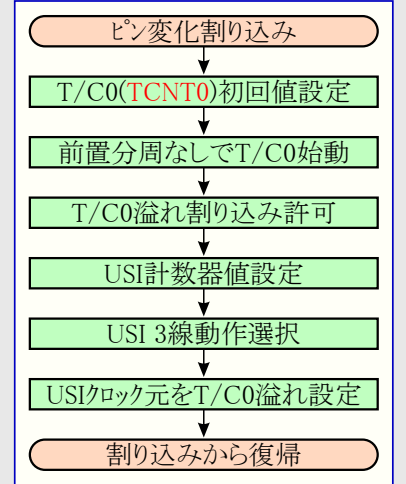


図14. T/C0溢れ割り込み処理ルーチン

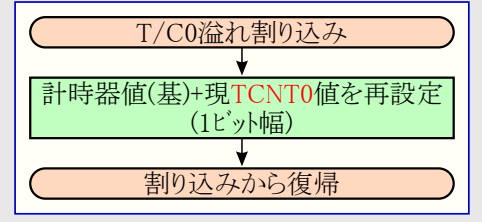
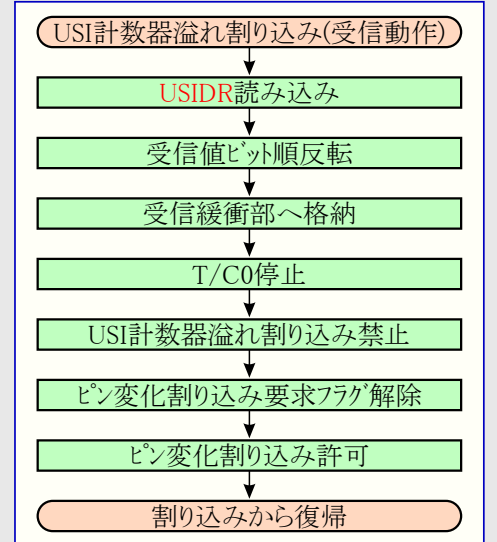


図15. USI計数器溢れ割り込み処理ルーチン



使用資源

受信部は、タイマ/カウンタ0、USI単位部/ピン、ピン変化割り込みを使います。

受信ドライバに対するメモリの必要条件は表2.で表示されます。コードメモリの過半数、合計236バイト内の136バイトは割り込みによって必要とされます。データのビット順反転は多量のメモリを要求するルーチンの1つです。コードの補助部の多くが同等で、故に送信部と共用され、(表での)合計よりも小さな総コード量になることに注意してください。

表2. 受信部ドライバのメモリ必要条件

	ルーチン	量(バイト)
初期化とデータ/緩衝処理	USI_UART_Initialize_Receiver()	30
	USI_UART_Receive_Byte()	28
	Bit_Reverse()	42
	(合計)	100
採取とデータ格納	IO_Pin_Change_ISR()	68
	USI_Counter_Overflow_ISR()	68
	(合計)	136

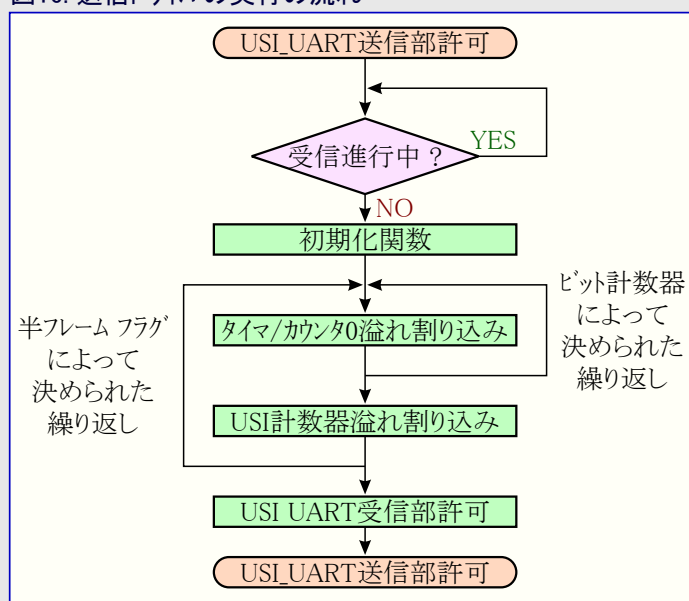
USI UART送信部

送信ドライバは関数呼び出しとして実装され、そして送信するバイトは関数呼び出しの引数として渡されます。

送信ドライバは4つの部分から成ります。

- ・送信関数
- ・USI UART初期化
- ・タイマ/カウンタ0溢れ割り込み処理ルーチン
- ・USI計数器溢れ割り込み処理ルーチン

図16. 送信ドライバの実行の流れ



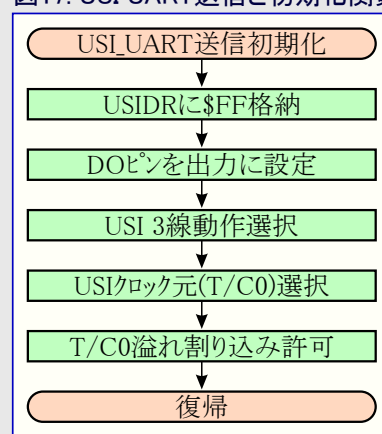
送信関数

送信を始めるために送信関数が呼び出されます。この関数は応用インターフェースであるドライバの一部です。送信関数は次のように宣言されます。

```
void USI_UART_Transmit_Byte( unsigned char );
```

送信されるべきバイトのビット順を反転することにより、送信に対してデータが準備されます。そしてそれは送信緩衝部に格納されます。既に送信稼動中なら、割り込みルーチンが緩衝部からデータを自動的に取り出すので、他の動きは全く必要ありません。送信進行中でない場合、USIを送信用に初期化する前に、進行中の受信があるかどうかの検査が行われます。USIはその動作用に構成設定されますが、初期計数器値(基)は割り込みルーチン移行までの最小時間に設定されます。データレジスタと計数器値(基)は割り込みルーチンで正しい内容を格納されます。

図17. USI UART送信と初期化関数



USI_UART_Initialize_Transmitter – 送信部

3線動作が選択されると、DOピンの方向は対応するDDRxnビットで制御されるので、これを出力に設定します。このピンの初期状態が入力なると受信時にプルアップ付き入力に切り換えられることから、このピンの外部プルアップが必要でしょう。(訳注:前2行修正)

送信に必要な割り込みが許可されます。

Timer0_OVF_ISR – 送信部

タイマ/カウンタ0溢れ割り込み処理ルーチンは受信ドライバで使われるものと同様です。このルーチンの詳細については6頁の「Timer0_OVF_ISR – 受信部」項を参照してください。

USI_Counter_Overflow_ISR – 送信部

USI計数器溢れ割り込みが送信と受信の両方に使われることに注意してください。従って最初に取りれた動きが現在の動作種別を決めます。

データは送信前に(ビット順の)反転が必要です。これはUSI_UART_Transmit_Byte関数で既に行われています。これはUARTフレーム形式(図1をご覧ください)に従ってLSB先行でデータを送信するためです。そして送信バイト(USI_UART_TxData)は上位値と開始ビットで合成されます。合成されたデータがUSIDRに複写されます(図3をご覧ください)。USIDRが最初の“半フレーム”、開始ビットと送信バイトの下位側6ビットを格納できてしまったことを示すために状態フラグが設定されます。この状態フラグはUSI計数器溢れ割り込み処理ルーチンで使われます。

タイマ/カウンタに対する計時器値(基)とUSI計数器に対する計数器値(基)が設定され、タイマ/カウンタ0が始動されます。タイマ/カウンタ溢れがUSIDR内のビットを移動出力するためのクロックを生成します。

以降のUSI計数器溢れ割り込みはフレーム(前半フレーム)の最初の5ビットが転送されてしまった時と、フレーム送信が完了されてしまった時に生成されます。USI計数器溢れ割り込み処理ルーチンは図15で図解されます。受信部が同じUSI計数器溢れ割り込みも使うため、この図がそれも示すことに注意してください。

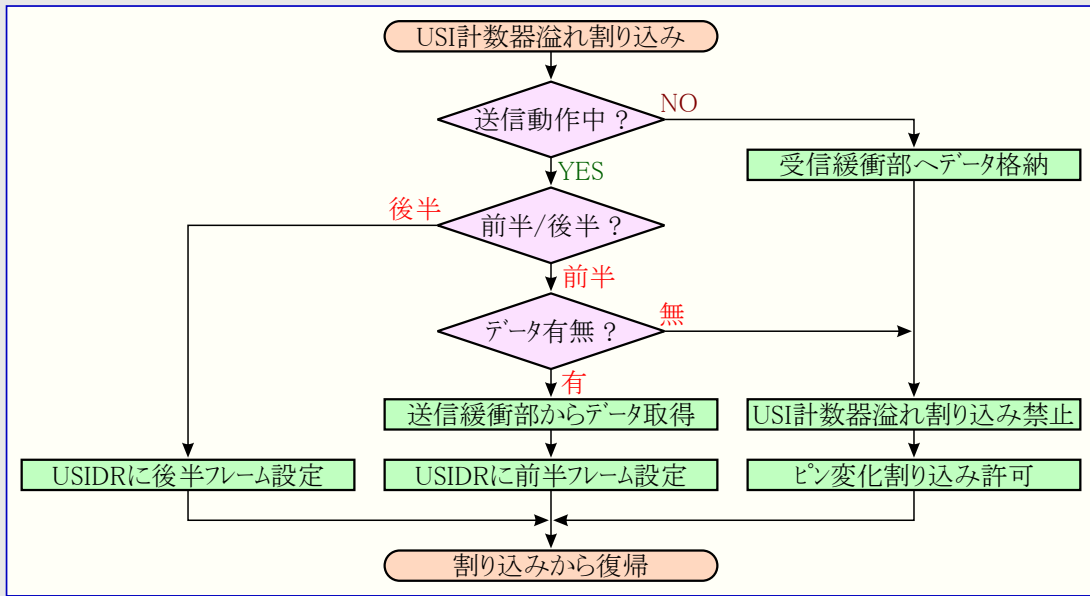
割り込みが前半フレームの送信によるものなら、データの残りと停止ビットがUSIDR内に合成されます(図3をご覧ください)。そしてUSIDRが前半フレームを格納されていることを示す状態フラグが解除されます。

割り込みが完全なフレームの転送によって起動された場合、タイマ/カウンタ0が停止され、送信進行中(USI_UART_status.ongoing_Transmission_From_Buffer)フラグが解除されます。

最後に、割り込み要求フラグ解除が割り込みベクタ起動によってハードウェアで行われないため、USI計数器溢れ割り込み要求フラグが解除(0)されます(全フラグが解除されます)。

この溢れ割り込みはUSIを禁止してUSI UART受信動作に再移行する前に、緩衝部を完全に送信し終わるまで動きます。

図18. USI計数器溢れ割り込み処理ルーチン



使用資源

送信部はタイマ/カウンタ0、USI単位部/ピンを使います。

送信ドライバに対するメモリの必要条件は表3で表示されます。コードメモリの過半数、合計252内の116バイトはUSI計数器溢れ割り込みによって必要とされます。データのビット順反転は多量のメモリを要求するルーチンの1つです。コードの補助部の多くが同等で、故に受信部と共用され、(表での)合計よりも小さな総コード量になることに注意してください。送受併せた総コード量は～450バイトです。

表3. 送信部ドライバのメモリ必要条件

ルーチン		量(バイト)
初期化とデータ/緩衝処理	USI_UART_Initialize_Transmitter()	46
	USI_UART_Transmit_Byte()	48
	Bit_Reverse()	42
	(合計)	136
採取とデータ格納	USI_Counter_Overflow_ISR()	116
	(合計)	116



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 Chantierie
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

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のウェブサイトには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 2003. 不許複製 Atmel®、ロコ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AVR307応用記述(doc4300.pdf Rev.4300A-10/03)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。