

AVR304 : 半二重割り込み駆動ソフトウェアUART

要点

- 非同期走行(割り込み駆動)
- 8MHzクロック周波数で38400bpsまでのボーレート処理能力
- 8ビット タイマ/カウンタと外部割り込みを持つどのデバイスでも動作

1. 序説

多くの制御応用は同時に一方向でだけ連続的に通信します(半二重通信)。この応用記述は8ビット タイマ/カウンタと外部割り込みを使用してどのAVRデバイスでも半二重UARTを作成する方法を記述します。このソフトウェアはハードウェアUARTがないデバイスにシリアルポートを実装する、または既にUARTが装備されているAVRデバイスに2つ目のシリアルポートを実装するのに用いることができます。アイドル線はその線を論理1に保つことによって合図されます。開始ビットは常に0で、UART受信部は最初の下降端によってフレームの開始を検出します。開始ビットに続いてデータビットが来て、常に論理1の停止ビットが後続します。この停止ビットは次の開始ビットが送られるまで1での安定を維持されます。

図1-1. UART通信フレーム形式

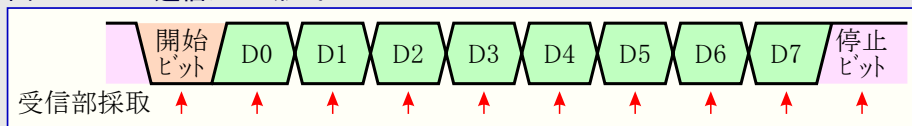
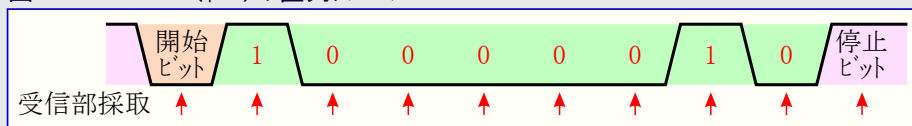


図1-2. ASCII "A" (\$41) の直列フレーム



2. 動作の理屈

非同期直列データ通信はデータ転送に於いていくつかの簡単な規則に従います。データは1度に1ビットで連続的に送信されます。新しいバイトが到着したことを受信部に知らせるために、各バイトは所謂開始ビットと停止ビットの間に配置されます。この構造はフレームと呼ばれます。フレーム形式は図1-1と図1-2で示されます。フレームは1開始ビット、8データビット、1停止ビットを持ちます。このフレーム形式が本応用記述で実装されます。フレーム形式は拡張することができ、パリティビットともっと停止ビットも含むかもしれませんが、非同期送信では受信部へ独立したクロックが全く提供されません。データの正しい受信は全てのビット長を等しく保つことによって保証されます。受信部は開始ビット先頭の下降端で同期し、次の採取時間を自身の計時器で見つけます。ビット長は通信に使用するボーレートによって決められます。UARTの場合はボーレートが1秒間の転送ビット数と等価です。送信部と受信部は正しい受信のために等しく同じボーレートを使用する設定でなければなりません。

図1-1で見られるようにフレームは下降端で始まります。この下降端が受信部で(外部割り込みを使用して)初期割り込みを生成します。この割り込みはタイマ/カウンタ0を開始し、そしてそれは正確に1.5ビット長での時間超過に設定されます。この1.5ビット長遅延は最初のデータビットの中央で次の採取事象を生成するのに必要とされます。次の8つの割り込みはタイマ/カウンタ0を使用して予め定義された(1ビット長)遅延後に生成されます。

データ送信は全てのビットが等しい長さを持ち、計時器を一定の(1ビット長)遅延に予め設定することができるので、ずっと簡単です。最初のビットは開始ビットです。これは常に論理0(またはスペース)です。このビットはデータが来ることを受信部に知らせます。そしてデータビットがLSB(最下位ビット)先行、MSB後行でシフト出力されます。最後に最終ビットは受信部がデータバイトを分けることができるように停止ビットでなければなりません。これは常に論理1(またはマーク)です。図2-2は直列データ送信用の流れ図を示します。

(訳補) 上記の論理値は一般的なデバイスピンでのことです。通常、各信号線は反転器を通してコネクタに接続されます。従ってコネクタ(またはケーブル)上での論理は上記記述と反対になります。上記()内記述のスペースとマークの呼称はRS232規格上での呼称で、回線上の信号に対して用います。ここでは結果としてこれになるという意味です。



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. 0941C-04/08, 0941CJ2-02/14

図2-1. 直列データ受信用流れ図

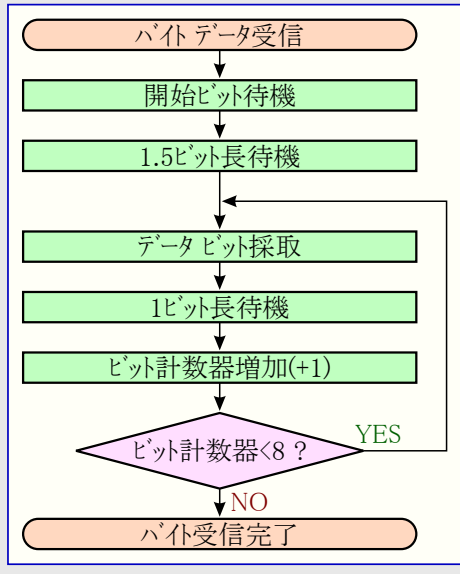
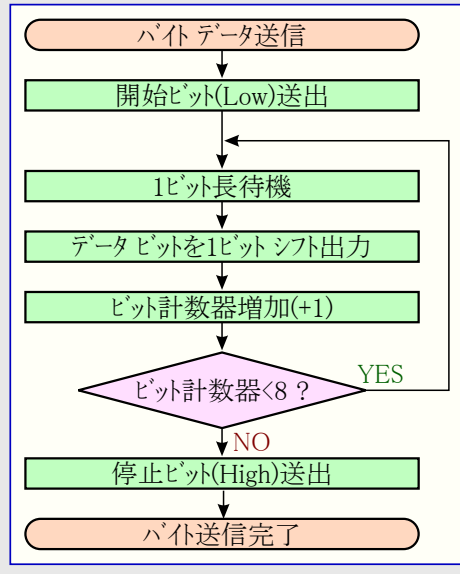


図2-2. 直列データ送信用流れ図

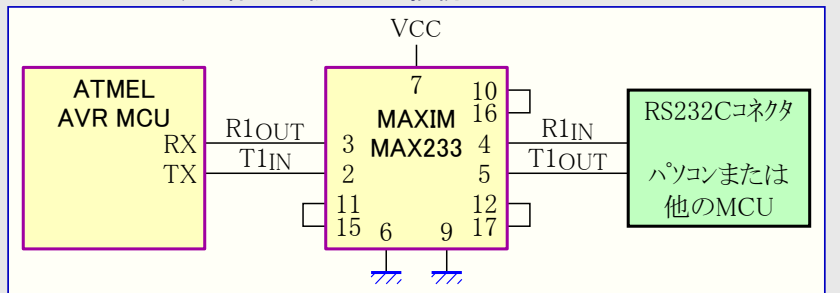


3. 接続

RS-232規格は±15Vの電圧レベルを必要とします(下記訳補参照)。これらの信号レベルを生成するためにMCUの電圧をRS232電圧に変換する独立したインターフェース回路が必要とされます。単独チップインターフェース回路の例はMAXIMのMAX233です。これは単一5V電源で動作し、5VをRS232信号レベルに変換するためのDC-DC変換器を持っています。

受信(RX)ピンは外部割り込みなのでINT0ピンに接続されなければなりません。送信(TX)ピンとしてどのピンが使用されるかは重要ではありません。図3-1はMCUが物理的にどうRS232線へ接続されるかを示します。

図3-1. RS232シリアル線への物理的な接続



(訳補) 上記記述の±15VはRS-232規格に於ける送出端の最大信号電圧を意味します。実際のRS-232規格に於ける信号電圧は送出端と受入端で個別に規定されており、送出端での最大値は上記のように±15Vで最低は±5Vです。受入端で保証されなければならない最低信号電圧は±3Vです。

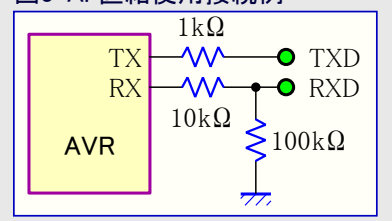
ここで両端規定電圧が正側だけをみるとVCCに近いことに注目してください。両方向について考察してみます。RS-232適合トランシーバの例として本例使用のMAX233を参照します。

AVRの受信(RX)信号については上記規定のように±5~±15Vで送られて来ます。実際に使用される電圧は殆どが±9~±12Vの範囲です。AVRの入力閾値は1.4~2.5V前後なので閾値の問題はありません。負電圧とVCCを越える電圧だけが問題になります。これはRX線に直列(電流制限)抵抗とVCC電圧をやや下回る(例えば4.7Vの)ツェナーダイオードを挿入することによってこれを回避できます。あまり推奨されていませんが、チップ内の入力保護ダイオードを利用して直列抵抗のみとすることもできます。この場合の抵抗値は負荷を軽減するために比較的大きな値、例えば10kΩが適当でしょう。

AVRの送信(TX)信号については相手側の入力特性に依存します。現状使用されている殆どの機器の入力特性はいくつかの理由から殆ど等価で、従ってここではMAX233の特性に基づきます。当然ですがAVRは0=GND値、1=VCC値として出力します。実際の出力電圧は負荷に依存し、重負荷ほど出力電圧が中心電圧へ寄ります。無負荷であっても完全にGND値またはVCC値ではありませんが、これはこの考察に於いて無関係なので問題になりません。MAX233の閾値電圧はLow認識側が最小0.8V、High認識側が最大2.4Vです。入力インピーダンス(AVR出力負荷)は最小3kΩで、VCC=5VのAVRにとって約1.67mAの負荷電流になります。この負荷電流に於けるAVR出力電圧変化は(代表特性上)±0.1V以下で、従って0出力に対して0.1V、1出力に対して4.9Vが保証されると見做せます。これらから、伝送路に於ける(雑音を含む)電圧低下がLow側で0.8V-0.1V=0.7V、High側で4.9V-2.4V=2.5V未満であれば問題がないことになります。

上記から、最低RX線の直列抵抗のみで物理的にインターフェース可能で、入出力の論理値を逆に変更することで本例も利用可能です。現実これを適用する場合は未接続時にRX線をLow(アイドル)に固定化するプルダウン抵抗と出力短絡によるAVR保護用のTX線の直列抵抗が必要でしょう。RXピンの内蔵プルアップが禁止されなければならないことに注意してください。

図3-A. 直結使用接続例



4. 実装

これらのソフトウェアUARTルーチンはタイマ/カウンタ0と1つの外部割り込みを使用します。MCUに対して供給されるクロックが達成可能な最大ボーレートを制限します。このソフトウェアUARTは8MHzクロック周波数に於いて38400bpsまでのボーレートを処理する能力があります。この速度では全ての処理能力近く使用されますが、MCUは未だ送出される各バイト間で他の作業に利用可能です。

ビット長は別の(異なる)溢れを生成するのに必要とする($C \times N$)周期数によって決められます。タイマ/カウンタに於いて N はタイマ/カウンタ比較レジスタに設定される値で、 C はAVRデータシートのタイマ/カウンタ前置分周器で記述されるようにタイマ/カウンタ前置分周係数です。値 N は次式で計算することができます、ここでの $Xtal$ はシステムの周波数です。

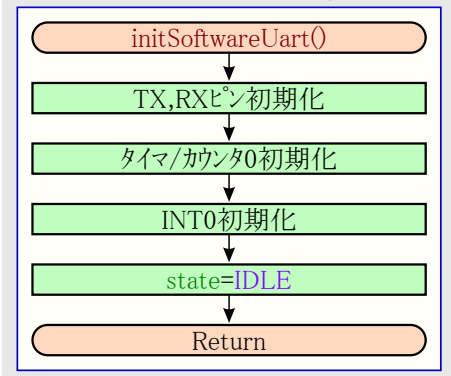
$$N = \frac{Xtal}{\text{ボーレート} \times C}$$

前置分周係数が1,8,64,256,1024内の1つの値であるべきことに注意してください。

5. "initSoftwareUart()"関数 - UART初期化

UARTを使用してデータを転送する前に、"initSoftwareUart()"関数を呼び出すことによってUARTが初期化されなければなりません。この関数は通信に必要なタイマ/カウンタ前置分周器の初期設定、タイマ/カウンタと外部割り込みの許可を行います。このサブルーチンからの復帰で、全体割り込みを許可するために'_enable_interrupt()' (SEI)命令が続くべきです。これがUARTを許可します。後で'_disable_interrupt()' (CLI)命令を発行することでUARTを禁止することができます。

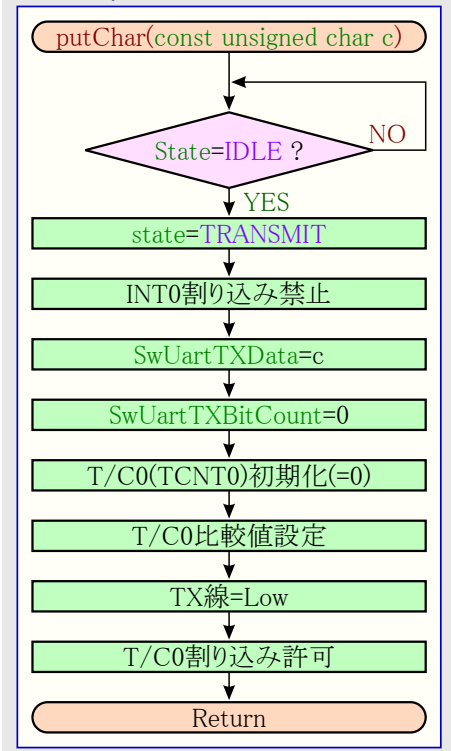
図5-1. "initSoftwareUart()"流れ図



6. "putChar(const unsigned char)"関数 - バイト送信

このルーチンはソフトウェアがデータの送信を必要とする時に使用されます。これはアイドル状態を持ち、その後(状態変数を)'TRANSMIT'に設定して(送信時に受信を禁止するために)外部割り込みを禁止し、正しいボーレート(タイマ/カウンタ割り込み)を設定して開始ビットを出力します。

図6-1. "putChar"流れ図

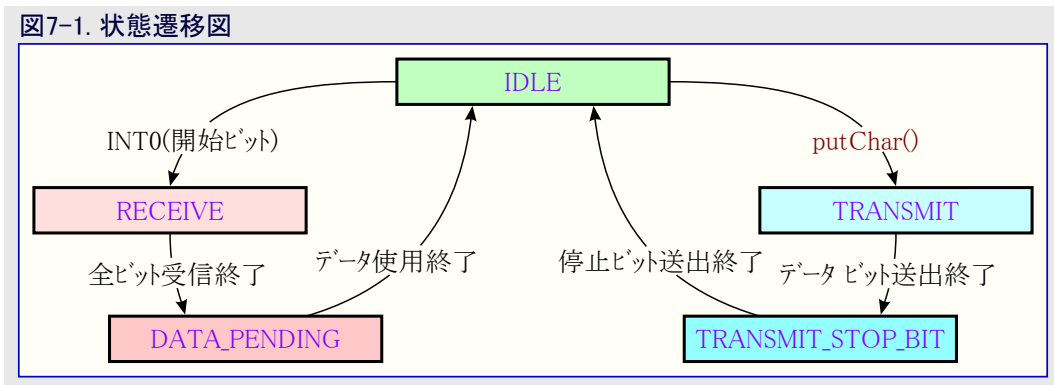


7. “state”番号制御

state列挙子は全ての関数で使用されます。これは5つの実行状態を持ちます。

1. **IDLE** : システムはアイドル状態で送受信両方が可能です。
2. **TRANSMIT** : ビット送信。システムは受信できません。
3. **TRANSMIT_STOP_BIT** : 停止ビット状態。これは最低1停止ビットが送出されるのを保証します。1停止ビット送出後、システム状態が**IDLE**に設定されます。
4. **RECEIVE** : ビット受信。INT0の割り込み時、システムはstateを**RECEIVE**に変更し、**DATA_PENDING**に変更する前に8ビットを受信します。
5. **DATA_PENDING** : 受信終了時。受信データはSwUartDataに格納され処理される準備が整います。データが処理された後で状態は**IDLE**に設定されるべきです。この状態は新しい受信発生でこれを停止せず、故に直ぐに処理されない場合にデータが失われるかもしれません。

状態遷移は図7-1.で示されます。



8. “Timer0_CompareA_interrupt()”割り込み処理ルーチン

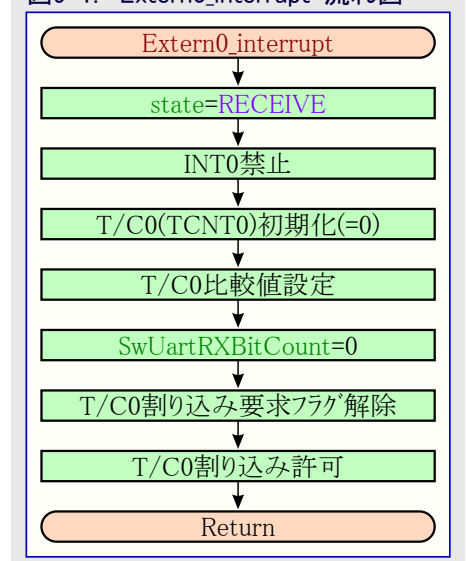
このルーチンは転送に於ける各ビットの送受信を行います。このルーチンは次のビットを送信または受信するためにタイマ/カウンタ比較一致で自動的に呼び出されます。

タイマ/カウンタ比較一致割り込みは送受信の開始ビット時に各々“putChar”と“Extern0_interrupt”によって許可されます。ルーチンは抜け出す前に次のビットを処理します。処理したビットが停止ビットの場合、タイマ/カウンタ比較一致割り込みが禁止され、外部割り込みが再び許可されます。

9. “Extern0_interrupt()”割り込み処理ルーチン

外部割り込み0はUARTがアイドルである限り常に活性(有効)です。外部割り込みで“Extern0_interrupt()”ルーチンが呼び出されます。このルーチンは直列データの受信を始めます(代替関数名は“uart_receptoin”でしょう)。外部割り込みはINT0ピンの下降端で起きます(下降端は開始ビットの開始を示します。図1-1.をご覧ください)。これはタイマ/カウンタ比較一致割り込みを活性(許可)にし、開始ビットの先頭に対して1.5ビットの遅延を生成します。抜け出す前に、到着バイト内の下降端での受信部再初期化を防ぐために外部割り込みが禁止されます。

図9-1. “Extern0_interrupt”流れ図



(訳注) 原書に於いて主にタイマ/カウンタ0溢れ割り込みで記述されていますが、実際の試供プログラムでは比較一致割り込みです。これは本応用記述が元々AT90S1200用アセンブリ言語で書かれていたためで、この試供プログラムは現在公開されていません。現在公開されているのはC言語で書かれた比較的最近のデバイス用のものです。

本書では実際の試供プログラム合わせて比較一致で記述しています。また元々のアセンブリ言語試供プログラムも同梱しています。

10. プログラム例

本応用記述にプログラム例が含まれています。このプログラムは(送られてくる)文字を待ちます。'a'が送られた場合、受信に於いてソフトウェアUARTは"atmel avr"メッセージを返します。'A'が送られた場合は"Atmel avr"、他の全ての場合は"Unknown command"です。

11. 要約

本応用記述でソフトウェアUARTが実装されます。MCUは8MHzクロック周波数で38400bpsを使用する能力があります。UARTは"`initSoftwareUart()`"呼び出しによって初期化され、そして全体割り込みを許可します。UARTがアイドルなら、到着データを自動的に受信します。データを送信するために、`unsigned char`として関数へ渡されるデータと共に"`putChar()`"と呼ばれるサブルーチンが呼び出されます。



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 2008. 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AVR304応用記述(doc0941.pdf Rev.0941C-04/08)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。