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までのﾎﾞｰﾚｰﾄ処理能力
- 8ﾋﾞｯﾄ ﾀｲﾏ/ｶｳﾝﾀと外部割り込みを持つどのﾃﾞﾊﾞｲｽでも動作

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ﾋﾞｯﾄ ﾀｲﾏ/ｶｳﾝﾀ0と外部割り込みを使ってどのAVRﾃﾞﾊﾞｲｽでも半二重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ﾄﾞｳｪｱUARTがないﾃﾞﾊﾞｲｽにｼﾘｱﾙ ﾎﾟｰﾄを実装する、または既にUARTが装備されているAVRﾃﾞﾊﾞｲｽに2つ目のｼﾘｱﾙ ﾎﾟｰﾄを実装するのに用いることができます。ｱｲﾄﾞ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。開始ﾋﾞｯﾄは常に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。開始ﾋﾞｯﾄに続いてﾃﾞｰﾀ ﾋﾞｯﾄが来て、常に論理1の停止ﾋﾞｯﾄが後続します。この停止ﾋﾞｯﾄは次の開始ﾋﾞ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

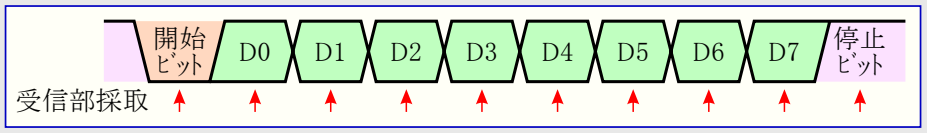
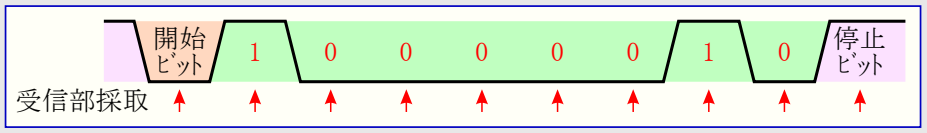


図1-2. ASCII"A"($41)の直列ﾌﾚｰﾑ



2. 動作の理屈

非同期直列ﾃﾞｰﾀ通信はﾃﾞｰﾀ転送に於いていくつかの簡単な規則に従います。ﾃﾞｰﾀは1度に1ﾋﾞｯﾄで連続的に送信されます。新しいﾊﾞｲﾄが到着したことを受信部に知らせるために、各ﾊﾞｲﾄは所謂開始ﾋﾞｯﾄと停止ﾋﾞ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開始ﾋﾞｯﾄ、8ﾃﾞｰﾀ ﾋﾞｯﾄ、1停止ﾋﾞｯﾄを持ちます。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、ﾊﾟﾘﾃｨ ﾋﾞｯﾄともっと停止ﾋﾞｯﾄも含むかもしれません。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。ﾃﾞｰﾀの正しい受信は全てのﾋﾞｯﾄ長を等しく保つことによって保証されます。受信部は開始ﾋﾞ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。ﾋﾞｯﾄ長は通信に使うﾎﾞｰﾚｰﾄによって決められます。UARTの場合はﾎﾞｰﾚｰﾄが1秒間の転送ﾋﾞｯﾄ数と等価です。送信部と受信部は正しい受信のために等しく同じﾎﾞｰﾚｰﾄを使う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(外部割り込みを使って)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ﾋﾞｯﾄ長での時間超過に設定されます。この1.5ﾋﾞｯﾄ長遅延は最初のﾃﾞｰﾀ ﾋﾞ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を使って予め定義された(1ﾋﾞｯﾄ長)遅延後に生成されます。

ﾃﾞｰﾀ送信は全てのﾋﾞｯﾄが等しい長さを持ち、計時器を一定の(1ﾋﾞｯﾄ長)遅延に予め設定することができるので、ずっと簡単です。最初のﾋﾞｯﾄは開始ﾋﾞｯﾄです。これは常に論理0(またはｽﾍﾟｰｽ)です。このﾋﾞｯﾄはﾃﾞｰﾀが来ることを受信部に知らせます。そしてﾃﾞｰﾀ ﾋﾞｯﾄがLSB(最下位ﾋﾞｯﾄ)先行、MSB後行で移動出力されます。最後に最終ﾋﾞｯﾄは受信部がﾃﾞｰﾀ ﾊﾞｲﾄを分けることができるように停止ﾋﾞ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.は直列ﾃﾞ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ﾃﾞﾊﾞｲｽ ﾋﾟﾝでのことです。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(またはｹｰﾌﾞﾙ)上での論理は上記記述と反対になります。上記( )内記述のｽﾍﾟ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の意味です。

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

応用記述

Rev. 0941C-04/08, 0941CJ3-04/21

図2-1. 直列ﾃﾞｰﾀ受信用流れ図

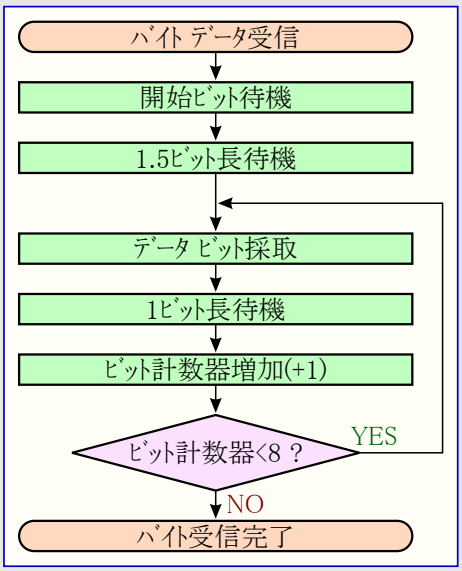
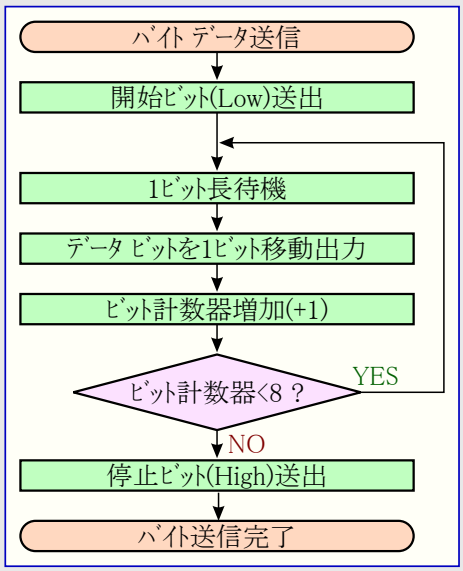


図2-2. 直列ﾃﾞｰﾀ送信用流れ図

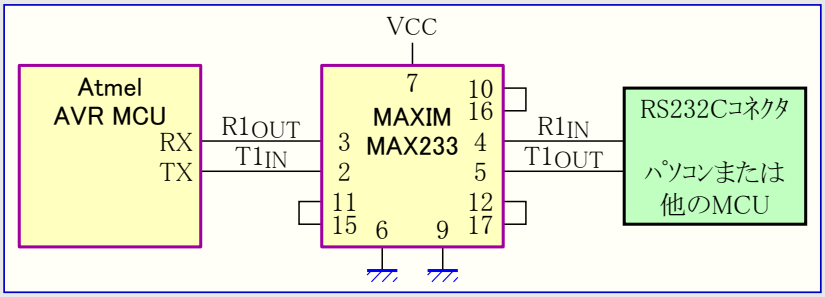


3. 接続

RS-232規格は±15Vの電圧ﾚﾍﾞﾙを必要とします(下記訳補参照)。これらの信号ﾚﾍﾞ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。単独ﾁｯﾌﾟ 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信号ﾚﾍﾞﾙに変換するためのDC-DC変換器を持っています。

受信(RX)ﾋﾟﾝは外部割り込みなのでINT0ﾋﾟﾝに接続されなければなりません。送信(TX)ﾋﾟﾝとしてどのﾋﾟﾝが使わ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。

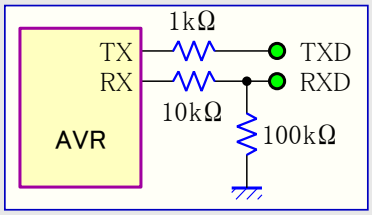
ここで両端規定電圧が正側だけをみるとVCCに近いことに注目してください。両方向について考察してみます。RS-232適合ﾄﾗﾝｼｰﾊﾞの例として本例使用のMAX233を参照します。

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。実際に使われる電圧は殆どが±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)ﾂｪﾅｰﾀﾞｲｵｰﾄﾞ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、ﾁｯﾌﾟ内の入力保護ﾀﾞｲｵｰﾄﾞ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。現状使わ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。入力ｲﾝﾋﾟｰﾀﾞﾝｽ(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。現実にこれを適用する場合は未接続時にRX線をLow(ｱｲﾄﾞﾙ)に固定化するﾌﾟﾙﾀﾞ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ﾋﾟﾝの内蔵ﾌﾟﾙｱｯﾌﾟ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います。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われますが、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、ここでの$Xtal$はシステムの周波数です。

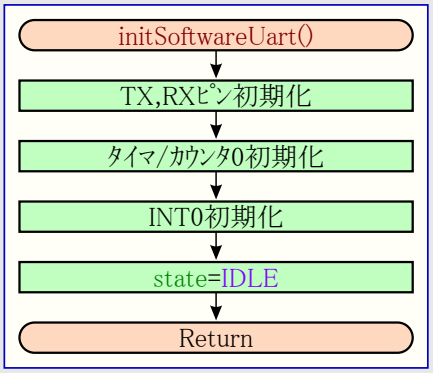
$$N = \frac{Xtal}{\text{ボーレート} \times C}$$

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を使っ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'_disble_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。

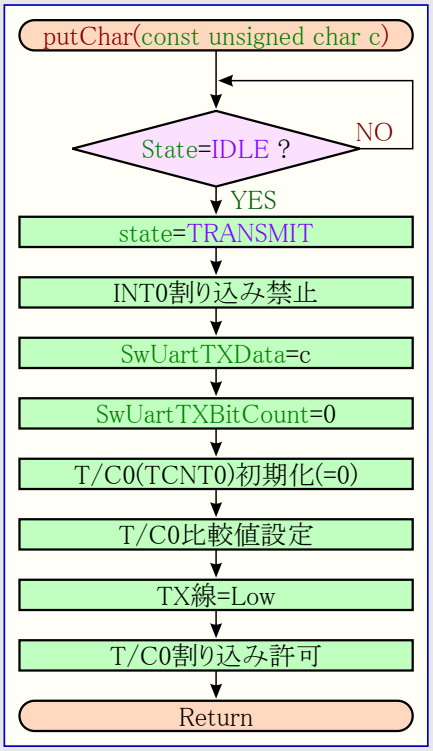
図5-1. "initSoftwareUart()"流れ図



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われます。これはアイドル状態を持ち、その後に(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



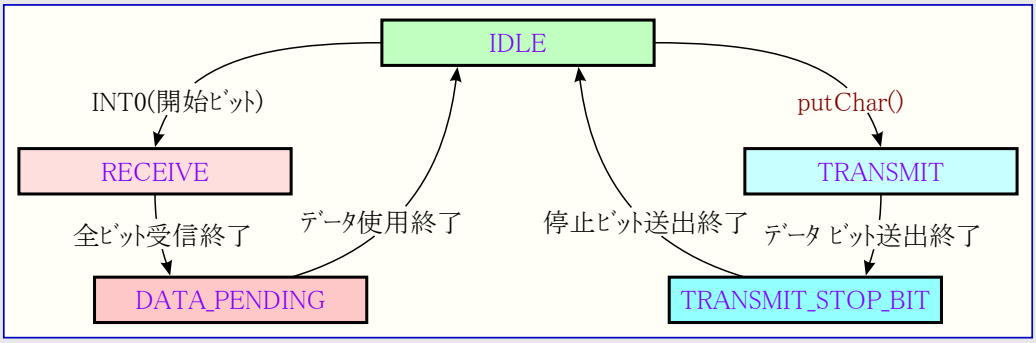
7. "state"番号制御

state列挙子は全ての関数で使われます。これは5つの実行状態を持ちます。

1. IDLE：ｼｽﾃﾑはｱｲﾄﾞﾙ状態で送受信両方が可能です。
2. TRANSMIT：ﾋﾞｯﾄ送信。ｼｽﾃﾑは受信できません。
3. TRANSMIT_STOP_BIT：停止ﾋﾞｯﾄ状態。これは最低1停止ﾋﾞｯﾄが送出されるのを保証します。1停止ﾋﾞｯﾄ送出後、ｼｽﾃﾑ状態がIDL Eに設定されます。
4. RECEIVE：ﾋﾞｯﾄ受信。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ﾋﾞｯﾄを受信します。
5. DATA_PENDING：受信終了時。受信ﾃﾞｰﾀはSwUartDataに格納され処理される準備が整います。ﾃﾞ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ﾃﾞ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。

図7-1. 状態遷移図



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

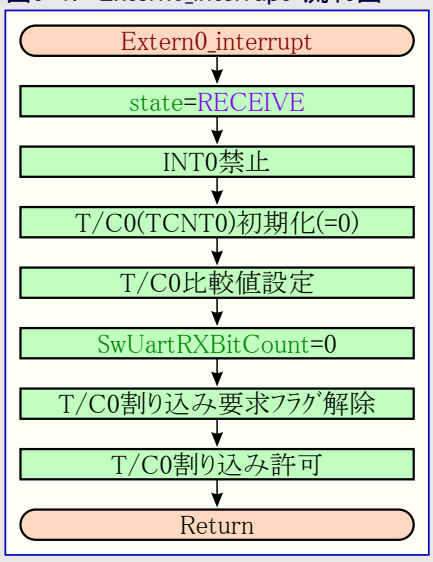
このﾙｰﾁﾝは転送に於ける各ﾋﾞｯﾄの送受信を行います。このﾙｰﾁﾝは次のﾋﾞｯﾄを送信または受信するためにﾀｲﾏ/ｶｳﾝﾀ比較一致で自動的に呼び出されます。

ﾀｲﾏ/ｶｳﾝﾀ比較一致割り込みは送受信の開始ﾋﾞｯﾄ時に各々"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ﾋﾞｯﾄを処理します。処理したﾋﾞｯﾄが停止ﾋﾞｯﾄの場合、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がｱｲﾄﾞ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ﾃﾞ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ﾋﾟﾝの下降端で起きます(下降端は開始ﾋﾞ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、開始ﾋﾞｯﾄの先頭に対して1.5ﾋﾞｯﾄの遅延を生成します。抜け出す前に、到着ﾊﾞ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、実際の試供ﾌﾟﾛｸﾞ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用ｱｾﾝﾌﾞﾘ言語で書かれていたためで、この試供ﾌﾟﾛｸﾞ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ﾃﾞﾊﾞｲｽ用のものです。

本書では実際の試供ﾌﾟﾛｸﾞﾗﾑ合わせて比較一致で記述しています。また元々のｱｾﾝﾌﾞﾘ言語試供ﾌﾟﾛｸﾞﾗﾑも同梱しています。

10. ﾌﾟﾛｸﾞﾗﾑ例

本応用記述にﾌﾟﾛｸﾞﾗﾑ例が含まれています。このﾌﾟﾛｸﾞﾗﾑは(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"ﾒｯｾｰｼﾞを返します。'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を使う能力があります。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がｱｲﾄﾞﾙなら、到着ﾃﾞｰﾀを自動的に受信します。ﾃﾞｰﾀを送信するために、unsigned charとして関数へ渡されるﾃﾞｰﾀと共に"putChar()"と呼ばれるｻﾌﾞ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

© *HERO* 2021.

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。