AVR274：単線ソフトウェアUART

要点

- ソフトウェア実装UART
- 半二重単線通信
- 割り込み駆動
- 1MHzシステム クロックで9600bpsまでの速度を支援
- 外部割り込みと8ビット タイマ/カウンタ比較割り込みを支援するどのAVR®にも適合

1. 序説

UART通信は一般的に送信と受信に対して独立したデータ線を使って実行されます。単線UARTは通信に単線だけを使い、従って高速全二重通信が必要とされない安価な解決策に於いて理想的です。この応用記述は単線UARTのソフトウェア実装を説明します。この規約は2つのデバイス間の半二重通信を支援します。必要条件は外部割り込みを支援する入出力ポートとタイマ/カウンタの比較一致割り込みだけです。

2. 動作の理屈

2.1. UARTフレーム

UART規約は非同期直列通信規格です。データは1度に1ビットで連続して転送されます。本実装は図2-1.で示されるように1開始ビット、8ビット データ、2停止ビットから成るフレームを使います。他の実装は5～9ビット データ、誤り制御用のパリティ ビット、1停止ビットから成る違うフレーム形式を使うかもしれません。送信が全く進行していない時の線はHighです。

図2-1. UARTフレーム形式

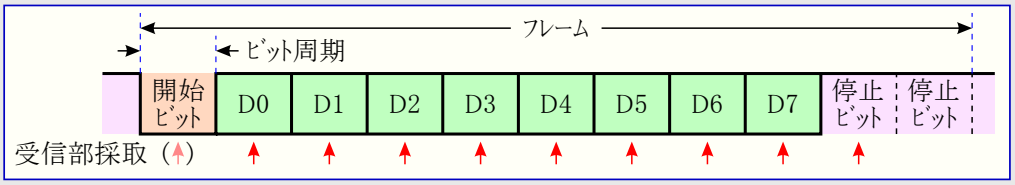
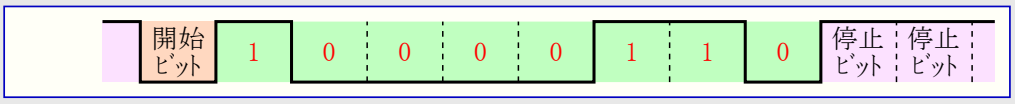


図2-2. ASCIIの'a'($61)に対する直列フレーム



2.2. 送信

送信は1ビット周期の間、開始ビット(線のLow引き込み)を送ることによって始められます。受信部は下降端を検知し、そして送信部と同期することができます。データ ビットの最下位ビットが先に送られます。

線を駆動するのにオープン ドレイン出力が用いられますが、両デバイスが同時に送信する場合、例え他の送信部がHighビットを送っても、Lowビットを送る送信部が線をLowに引き込みます。この状態を処理するため、UARTは受信中に決して送信を始めません。送信部は最後のビットが送信されてから線が変えられていないことを確認するために、次のビットを送信する前に線の採取も行います。Highビット送信持続時にLowビットが受信された場合、異常フラグが設定(1)されます。

2.3. 受信

受信は開始ビットが検出された時に始められます。そして各周期の中央でデータ ビットが採取されます。最初のデータ ビットは開始ビットが検出された後の1.5ビット周期後に採取されます。本実装はドライバによって費やされるクロック数を最小にするため、殆どのハードウェアUARTで見られる3採取の多数決を使いません。

2.4. ボーレート

UART通信に於ける速度はボーレートによって定義されます。この場合、ボーレートは開始ビットと停止ビットを含めた秒毎の送信ビット数と等価です。送信部と受信部は同じボーレート使用に設定されなければならず、さもないとそれらは同期することができません。一般的なボーレートは4800,9600,19200,28800,38400ですが、他の速度を使うこともできるかもしれません。

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. 8065A-03/07, 8065AJ4-04/21

2.5. 異常状態

起こり得る様々な異常状態があります。ﾎﾞｰﾚｰﾄが2つのﾃﾞﾊﾞｲｽで大きすぎるくらい違えば、それらは同期することができません。ﾎﾞｰﾚｰﾄがｸﾛｯｸ周波数に依存するため、この問題はｸﾛｯｸが予期した値と異なる場合に起き得ます。内蔵RC発振器を使う場合、UARTを使う前にそれを校正することが推奨されます。内蔵RC発振器の校正方法に関連する応用記述を参照してください。

受信したﾃﾞｰﾀ ﾊﾞｲﾄは単一ﾊﾞｲﾄ緩衝部に格納されます。次のﾊﾞｲﾄが受信される前にそのﾃﾞｰﾀが処理されない場合、緩衝部が溢れて旧ﾃﾞｰﾀが上書きされます。これが起きた場合は溢れﾌﾗｸﾞが設定(1)されます。この状況を救うためにUART速度を減らす、緩衝部を増やすことができ、または可能ならば頻繁に受信ﾙｰﾁﾝが呼び出されることかもしれません。

雑音ﾊﾟﾙｽが線をLowに強制した場合、AVRは下降端を検知して受信を開始します。開始ﾋﾞｯﾄがHighとして受信された場合、AVRは受信を中止して何のﾃﾞｰﾀも保存しませんが、雑音ﾊﾟﾙｽが(**訳補**:CPUｸﾛｯｸの)2周期よりももっと続くなら、不正なﾊﾞｲﾄが保存されるでしょう。

雑音は送信下のﾌﾚｰﾑも不正にするかもしれません。ﾋﾞｯﾄは1度しか採取されず、このためにこの1度だけの採取が雑音によって不正にされた場合、ﾌﾚｰﾑは不正な値を持ちます。これらの異常を検知するためにUARTﾌﾚｰﾑへﾊﾟﾘﾃｨ ﾋﾞｯﾄを追加することができます。

割り込み駆動近似法が用いられるため、応用ｺｰﾄﾞはUART通信と並行して実行するかもしれません。他の割り込み元が活性(有効)なら、それがUARTﾀｲﾐﾝｸﾞに影響を及ぼすかもしれず、そしてUART通信を失敗させるかもしれません。

主装置によって要求された時にだけ従装置がﾃﾞｰﾀを送る主従形態でﾄﾞﾗｲﾊﾞを使うことが推奨されます。これは両ﾃﾞﾊﾞｲｽが同時に送信する状況を防ぐでしょう。従装置が異常状態の場合、特定時間の間のUART線Lowをﾎﾟｰﾘﾝｸﾞすることによって主装置へ合図するかもしれません。そして主装置の異常ﾌﾗｸﾞが設定(1)され、従装置が線のLowを止めた時に通信を続けられるかもしれません。

3. 実装

この応用記述で説明されるｺｰﾄﾞはUART通信用のﾄﾞﾗｲﾊﾞとして書かれています。

3.1. ﾎﾞｰﾚｰﾄ設定

ﾋﾞｯﾄ採取とﾋﾞｯﾄ送信の間隔を生成するのにﾀｲﾏ/ｶｳﾝﾀ比較一致割り込みが使われます。ﾀｲﾏ/ｶｳﾝﾀは比較一致でのﾀｲﾏ/ｶｳﾝﾀ解除(CTC)動作でﾀｲﾏ/ｶｳﾝﾀが比較ﾚｼﾞｽﾀと等しい時に割り込みを生成するように設定されます。**式3-1.**で示されるように、各割り込み間の時間はｼｽﾃﾑ ｸﾛｯｸ、ﾀｲﾏ/ｶｳﾝﾀ前置分周器、比較ﾚｼﾞｽﾀの値に依存します。比較値10の設定は各割り込み間で11計数を生じます。ﾎﾞｰﾚｰﾄ設定はUARTﾍｯﾀﾞ ﾌｧｲﾙ(single_wire_UART.h)内で設定されます。

式3-1. ﾎﾞｰﾚｰﾄ計算

$$\text{ﾎﾞｰﾚｰﾄ} = \frac{\text{ｼｽﾃﾑ ｸﾛｯｸ周波数}}{(\text{1周期比較設定値}-1)\times\text{前置分周数}}$$

式3-2. 1周期比較設定値計算

$$\text{1周期比較設定値} = \frac{\text{ｼｽﾃﾑ ｸﾛｯｸ周波数}}{\text{ﾎﾞｰﾚｰﾄ}\times\text{前置分周数}} - 1$$

表3-1.はいくつかの一般的なｸﾛｯｸ速度とﾎﾞｰﾚｰﾄに対するﾀｲﾏ/ｶｳﾝﾀ設定を示します。誤差値は**式3-3.**を用いて計算されています。

表3-1. 1,2,4,8MHz発振器に対する1周期設定

ﾎﾞｰﾚｰﾄ	1MHz発振器			2MHz発振器			4MHz発振器			8MHz発振器		
	1周期設定	前置分周数	誤差	1周期設定	前置分周数	誤差	1周期設定	前置分周数	誤差	1周期設定	前置分周数	誤差
4800	207	1	-0.16%	51	8	-0.16%	103	8	-0.16%	207	8	-0.16%
9600	103	1	-0.16%	207	1	-0.16%	51	8	-0.16%	103	8	-0.16%
19200	-	-	-	103	1	-0.16%	207	1	-0.16%	51	8	-0.16%
28800	-	-	-	-	-	-	138	1	0.08%	34	8	0.82%
38400	-	-	-	-	-	-	103	1	-0.16%	207	1	-0.16%

式3-3. 誤差計算

$$\text{誤差}[\%] = \left(\frac{\text{最短一致時ﾎﾞｰﾚｰﾄ}}{\text{ﾎﾞｰﾚｰﾄ}} - 1\right)\times 100[\%]$$

これらがﾎﾞｰﾚｰﾄに関する最大設定であることに注意してください。これはﾀｲﾏ/ｶｳﾝﾀ割り込みが次の割り込みが生成される前に終了されなければならないからです。**式3-4.**が可能な最大ﾎﾞｰﾚｰﾄを与えます。割り込みは次の比較一致が起動される前に終わらなければなりません。比較一致割り込みに於ける最大周期数はｺﾝﾊﾟｲﾗ設定に依存して約100～110です。1MHzｼｽﾃﾑ ｸﾛｯｸ使用での最大ﾎﾞｰﾚｰﾄは約10000bpsです。UARTはこのﾎﾞｰﾚｰﾄで通信する時に殆ど全てのCPU資源を費やします。UARTを使う応用は次のﾌﾚｰﾑが受信される前に受信したﾃﾞｰﾀを読む時間も持たなければならず、さもないと溢れ異常が起こるでしょう。応用に依存して最大設定よりも低い適切なﾎﾞｰﾚｰﾄの設定が推奨されます。

式3-4. 最大ﾎﾞｰﾚｰﾄ設定

$$\text{ﾎﾞｰﾚｰﾄ} < \frac{\text{ｼｽﾃﾑ ｸﾛｯｸ周波数}}{\text{比較一致割り込みでの最大実行周期数}}$$

3.2. ハードウェア

本実装は外部プルアップ回路を使って設計されています。従って入出力ピンはオープンドレイン出力を使わなければなりません。加えてそれらはやって来るフレームを検知するために外部割り込みを生成する必要があります。

AVRマイクロ コントローラ上のプルアップ抵抗に対する代表値は15～40kΩ です。Highビットを送る、または受信する時にAVRのポートはHi-Zにされます。ポートをLow出力に構成設定することによってLowビットが送られます。

RS-232規格を支援する装置と通信する場合、約-15～15Vの電圧レベルが必要とされます。それで信号をこれらの電圧に変換する回路が必要とされます。単一チップのインターフェース回路例はMaximのMAX232です。これは単一5V電源で動き、チップ上にRS-232レベルを生成するためのDC/DC変換器を持ちます。

3.3. 状態レジスタ

単線UARTは以降の4つのフラグを含む状態レジスタを持ちます。

- SW_UART_TX_BUFFER_FULL

送信データが用意されている時に設定(1)されます。SW_UART_Transmit関数を呼び出す時にこのフラグは0でなければなりません。

- SW_UART_RX_BUFFER_FULL

データが受信緩衝部で利用可能な場合に設定(1)されます。SW_UART_Receive関数を呼び出す時にこのフラグは1でなければなりません。

- SW_UART_RX_BUFFER_OVERFLOW

やって来たデータが受信緩衝部での溢れのために失われた場合に設定(1)されます。

- SW_UART_FRAME_ERROR

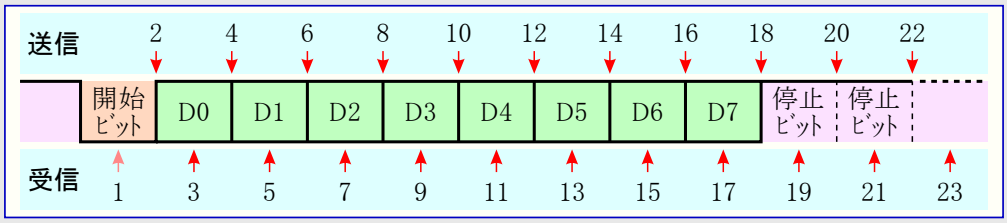
受信時にHighの開始ビットまたはLowの停止ビットが採取された場合に設定(1)されます。送信時に於いて直前に送信したビットと異なるビットが採取された場合にも設定(1)されます(**訳補**:比較一致割り込み処理に於いて今回のビット値を出力する前に前回出力値と現在のポート入力値を照合します)。

コード量を減らして速度を増すため、利用可能なら、状態レジスタをGPIORに置けるかもしれません(例のATmega32では利用不可)。

3.4. UART計数器

状態とUARTによってどのビットが送受信されるべきかを制御するためにUARTに於いて計数器変数が使われます。UARTはこの計数器値が0の時にアイドルです。この計数器値は**図3-1.**で示されるように送信時に偶数で、受信時に奇数です。

図3-1. UART計数器値



3.5. UART関数

ドライバは3つの全域関数から成ります。

- void SW_UART_Enable(void)
- void SW_UART_Transmit(uint8_t)
- uint8_t SW_UART_Receive(void)

SW_UART_statusはUART状態フラグを保持する全域変数です。single_wire_UART.hで定義した状態フラグをアクセスするのにSET_FLAG, CLEAR_FLAG, READ_FLAGのマクロを使うことができます。

UART実行に於いて以下の割り込みルーチンが使われます。

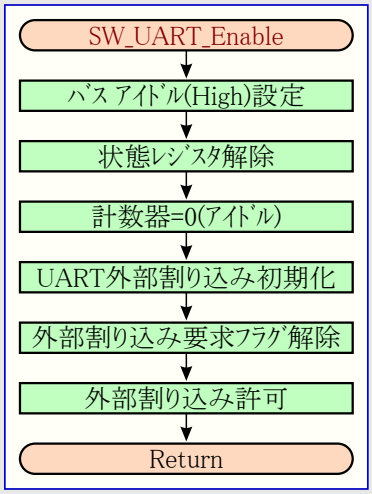
- _interrupt void External_SW_UART_ISR()
- _interrupt void Timer_Compare_SW_UART_ISR()

次の項はUART関数に対する簡単な説明と流れ図を含みます。

3.5.1. SW_UART_Enable

ﾃﾞｰﾀが送受信され得る前にUARTはSW_UART_Enable関数を呼び出すことによって許可されることが必要です。それはUARTﾋﾟﾝをHi-Zにし、故に線はHighのｱｲﾄﾞﾙになります。状態ﾚｼﾞｽﾀと計数器が解除され、このために進行中の送信は停止されます。外部とﾀｲﾏ/ｶｳﾝﾀの割り込みの禁止がUARTを停止します。

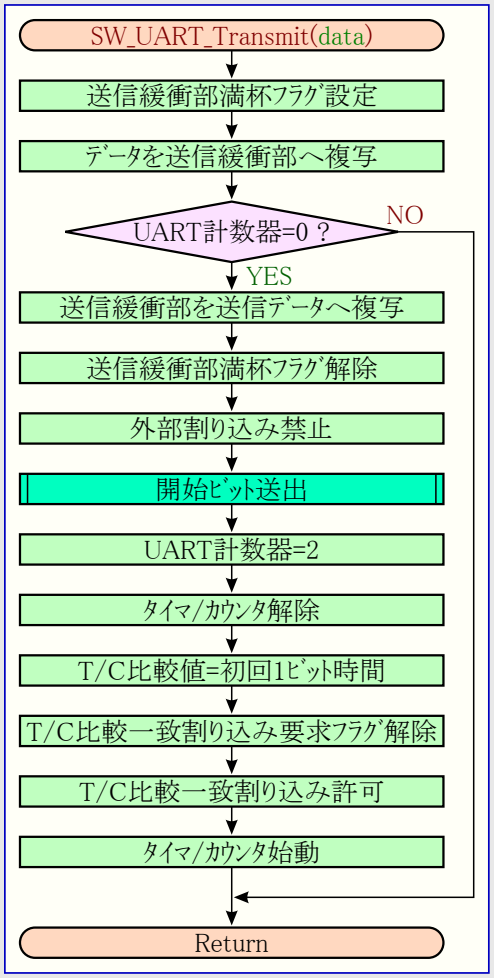
図3-2. SW_UART_Enable()関数



3.5.2. SW_UART_Transmit

SW_UART_Transmit()関数は引数として1ﾊﾞｲﾄを取り、このﾊﾞｲﾄを送信緩衝部に置きます。この関数を呼ぶ時に、SW_UART_TX_BUFFER_FULLﾌﾗｸﾞは0でなければならず、さもないとﾃﾞｰﾀが失われます。この関数が呼ばれた時にﾃﾞｰﾀ転送が進行中でなければ、開始ﾋﾞｯﾄを送ってﾀｲﾏ/ｶｳﾝﾀ比較一致割り込みを許可することによって新規送信が開始されます。

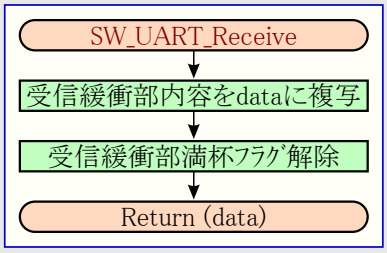
図3-3. SW_UART_Transmit()関数



3.5.3. SW_UART_Receive

この関数はRx_Dataの受信ﾊﾞｲﾄを返します。Rx_Data内に有効なﾃﾞｰﾀがあるのを保証するため、この関数を呼び出す前にSW_UART_RX_BUFFER_FULLﾌﾗｸﾞが調べられなければなりません。多数のﾊﾞｲﾄを受信するとき、次ﾊﾞｲﾄが受信される前にこの関数が呼び出されることが重要で、さもないと受信緩衝部が溢れてﾃﾞｰﾀが失われます。

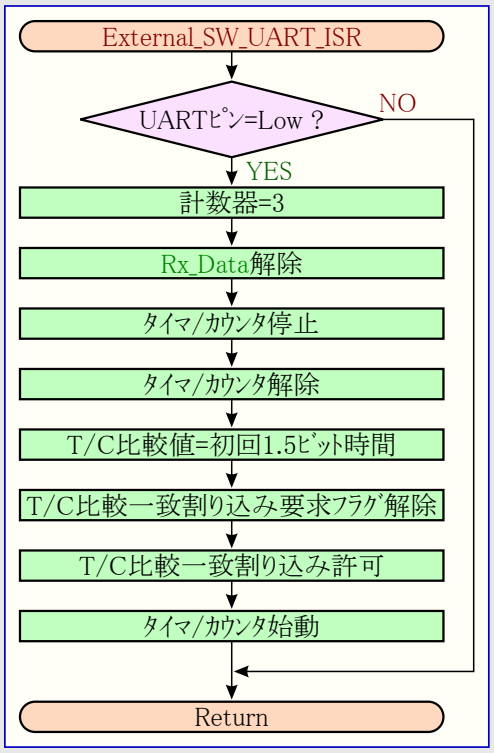
図3-4. SW_UART_Receive()関数



3.5.4. External_SW_UART_ISR

External_SW_UART_ISR()は送信または受信が進行中でない時に線で下降端が検出された時に起動されます。ﾙｰﾁﾝはUARTﾋﾟﾝがLowかを調べます。そうでなければ受信は開始されません。更なる外部割り込みも禁止されるため、ﾃﾞｰﾀ ﾋﾞｯﾄでの下降端が新しい割り込みを起動することは全くなく、故にこれは開始ﾋﾞｯﾄの検出にだけ使われます。

図3-5. External_SW_UART_ISR()



3.5.5. Timer_SW_UART_ISR

Timer_SW_UART_ISR()(図3-6.)はﾌﾚｰﾑの送受信に関する処理を制御します。それは比較一致割り込みが設定され、比較ﾚｼﾞｽﾀがﾀｲﾏ/ｶｳﾝﾀと等しいとき、自動的に呼ばれます。ﾃﾞｰﾀはUART計数器が奇数ならば送信され、この計数器が偶数の時に受信されます。受信処理(図3-8.)と送信処理(図3-7.)はこのﾀｲﾏ/ｶｳﾝﾀ割り込み内で直接実行されます。各計数器値の詳細については図3-1.を参照してください。

図3-6. Timer_SW_UART_ISR()

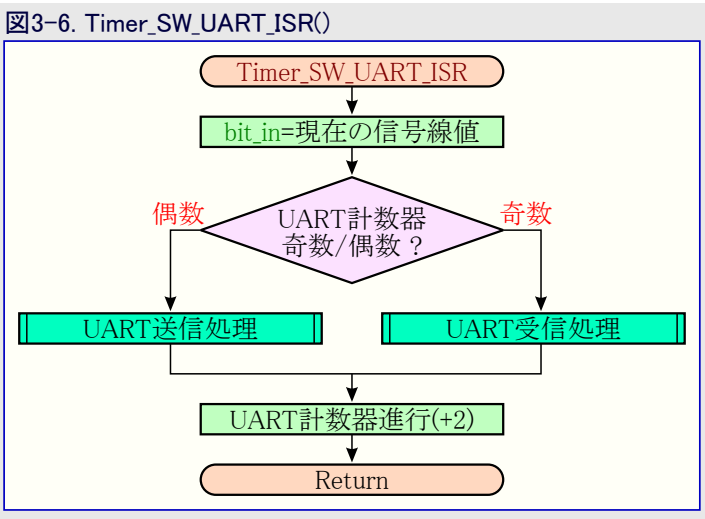


図3-7. UART送信処理

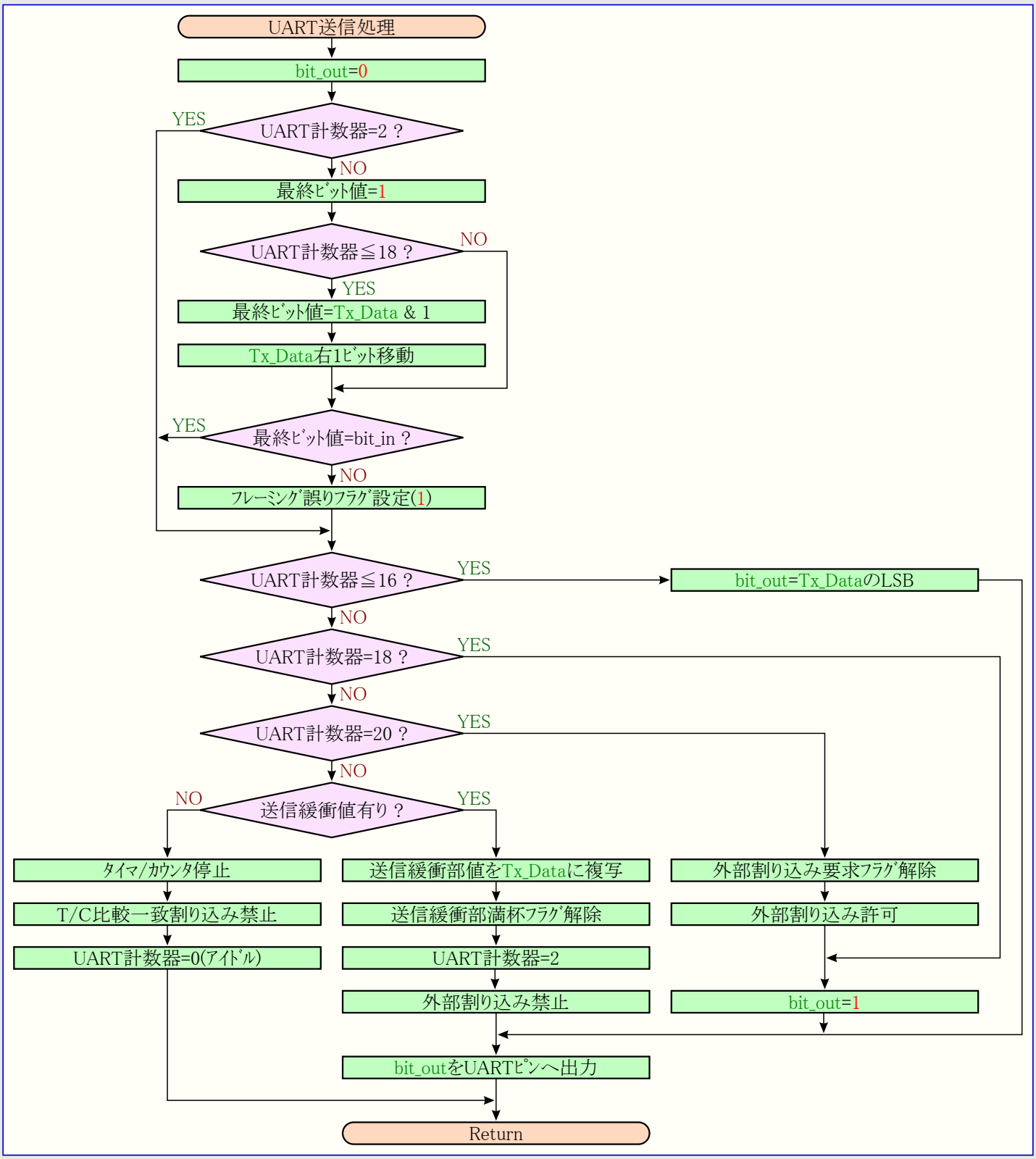
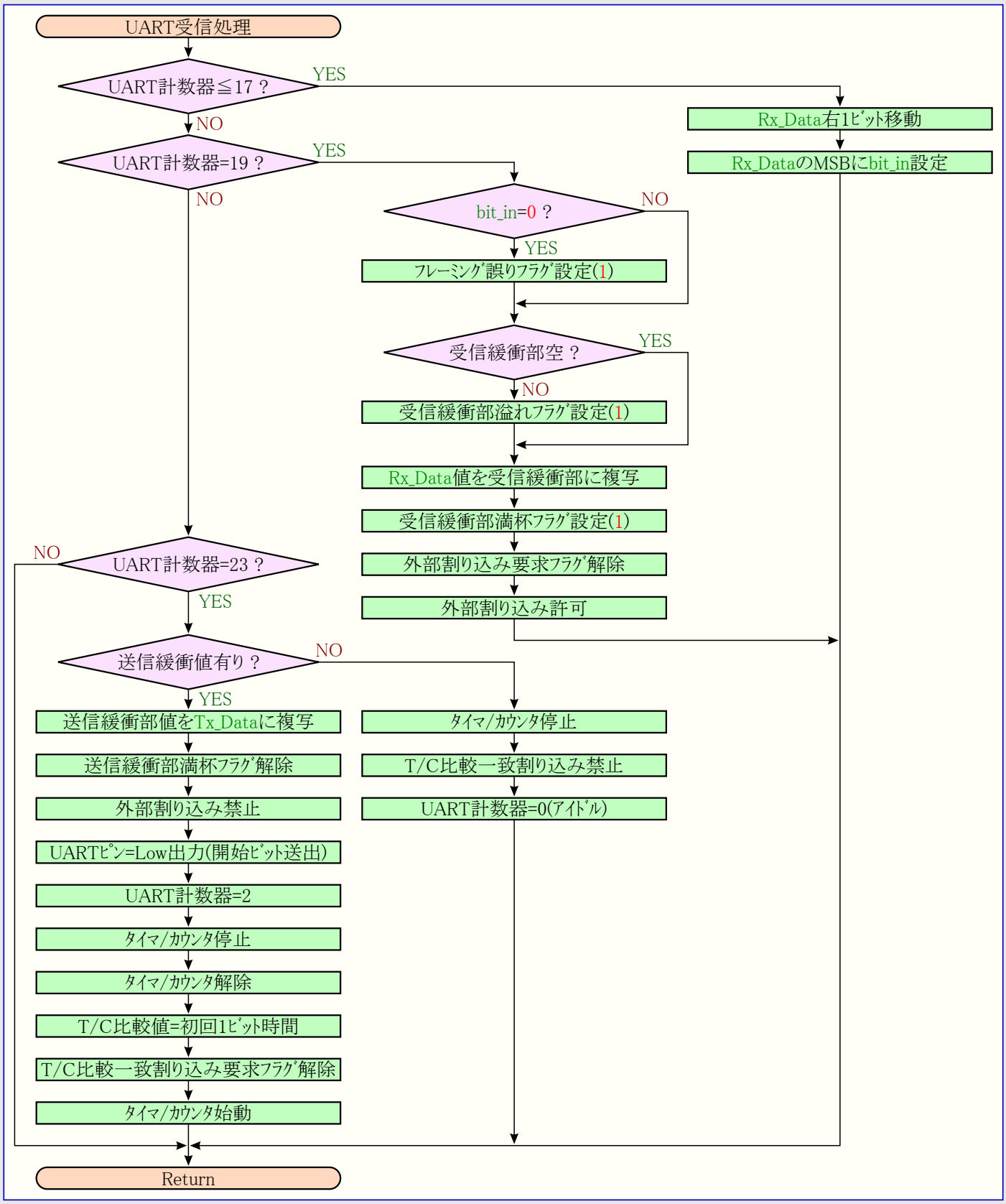


図3-8. UART受信処理



3.6. 例ﾌﾟﾛｸﾞﾗﾑ

main.cはUART試験用の例ﾌﾟﾛｸﾞﾗﾑを含みます。これはﾃﾞｰﾀをﾊﾞｲﾄ配列内へ受信し、この配列が一杯、またはCR符号($0D)が受信された時にこのﾃﾞｰﾀを送り返します。

3.7. ｺｰﾄﾞ量

IAR® EWAVR 4.21A、最高速最適化調整でのｺﾝﾊﾟｲﾙ時のUARTﾄﾞﾗｲﾊﾞに関するｺｰﾄﾞ量は500ﾊﾞｲﾄです。

4. 始める前に

ｿｰｽ ｺｰﾄﾞはwww.atmel.com/avrからZIPﾌｧｲﾙとしてﾀﾞｳﾝﾛｰﾄﾞすることができます。ｺｰﾄﾞはATmega32用に書かれ、IAR EWAVR 4.20A ｺﾝﾊﾟｲﾗを使ってｺﾝﾊﾟｲﾙされます。何の変更もなしにｿｰｽをｺﾝﾊﾟｲﾙするにはIAR EWAVRｺﾝﾊﾟｲﾗが必要とされます。Doxygen資料がreadme.htmlで利用可能です。

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 Chantrerie
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

本応用記述はAtmelのAVR274応用記述(do8065.pdf Rev.8065A-03/07)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。