

# AVR1510 : Xplain練習 – XMEGA USART

## 前提条件

- 必要な知識
  - ・ AVR1500:Xplain練習 – XMEGA<sup>®</sup> 基礎
  - ・ AVR1502:Xplain練習 – XMEGA 直接メモリ入出力制御器
- ソフトウェア必要条件
  - ・ ATMEL<sup>®</sup> AVR<sup>®</sup> Studio<sup>®</sup> 4.18またはそれ以降
  - ・ WinAVR/GCC 20100110またはそれ以降
- ハードウェア必要条件
  - ・ Xplain評価基板
  - ・ JTAGICEmk II
- 予想完了時間
  - ・ 1.5時間

## 1. 序説

USART(Universal Synchronous Asynchronous Receiver Transmitter)はコンピュータ、端末、その他の装置間で直列通信に於いて鍵となる要素です。

この練習はATMEL XMEGAのUSARTの基本的な構成設定と使い方を網羅し、3つの課題がポーリング形態と割り込み形態でのUSART使用法とCPUの相互作用なしにデータを転送するための直接メモリ入出力制御器(DMAC:Direct Memory Access Controller)の使用法を実演します。

## 2. 概要

この練習はATMEL XMEGA USARTの基本的な機能のいくつかを網羅します。

### 課題1. ポーリング形態

最初の課題はポーリング形態でUSARTを構成設定する方法を示します。いくつかの文字が環状戻り形態で転送されます。

### 課題2. 割り込み形態

この課題はUSARTを構成設定するのにドライバを使用する方法を示します。このドライバは開発者の暮らしにゆとりを持たせる環状緩衝部を持ちます。また、ハードウェア緩衝部はこの課題で示されます。

### 課題3. DMAC

ATMELのXMEGAは8ビットプロセッサに直接メモリ入出力制御器(DMAC)を導入します。USARTとDMACは殆どCPUの介在なしにデータに流れを許し、非常に有用です。この課題はDMACとのUSART構成設定法を示します。

## 3. 課題1: ポーリング形態でのUSART

USARTでポーリング形態を使用することは例えば応用をデバッグする時やプログラム内の段階が同期して起こることが予測される時に特に有用です。この課題ではUSARTから別のUSARTへデータを送るようにXplain評価基板を構成設定します。これはUSARTを試験する良い方法です。

この課題の目的はあなたが以下の方法を知ることです。

- ・ ポーリング形態でのUSART形態設定
- ・ 環状戻り形態でのいくつかの文字の送出
- ・ 送信が成功したことの検証



すること

1. ATMELのAVR Studioを開いてXMEGA-USARTフォルダ内のPolled\_Usart.apsプロジェクトファイルを開いてください。
2. Xplain評価基板に於いて、PD2とPD3のピン間をジャンパまたはケーブルで接続してください。



8ビット **AVR**<sup>®</sup>  
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. 8319A-06/10, 8319AJ1-03/14

### 3.1. ホーレート

ホーレートはパラメータとして周辺機能周波数( $f_{PER}$ )、ホーレート倍率(BSCALE)、ホーレート値(BSEL)を用いることによって計算されます。ATMEL XMEGAの目的周波数と周辺機能周波数は既定で2MHzに設定されます。BSELビットはホーレート設定で、BSCALEは更に多くの機能を加えますが、この課題では無視されます。BSCALEを0に設定することにより、BSELは式3-1.によって得ることができます。

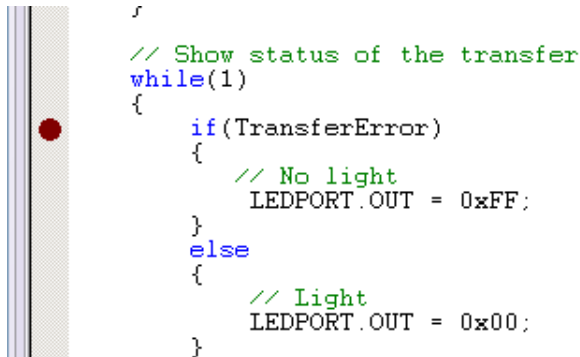
式3-1. BSEL値計算式

$$BSEL = \frac{f_{PER}}{2^{BSCALE} \times 16 \times f_{BAUD}}$$

### 3.2. ホーリングUSARTのデバッグ

6. コード全体を通して眺めて注釈を見てください。
7. プロジェクトを構築してデバッグ作業を開始(Playアイコンをクリック)してください。
8. 図3-1.で見えるようにwhile繰り返しに中断点を追加してください。
9. Rx\_BufとTx\_Bufを監視(Watch)に追加してください。
10. コードを走らせて(F5を押して)ください。
11. Rx\_BufとTx\_Bufが等しいことを確かめてください。
12. 1行実行(Single step, F11)して転送異常(LED点灯)が全く起きないことを検査してください。

図3-1. 最後のwhile繰り返しに中断点追加



## 4. 課題2: 割り込み形態でのUSART

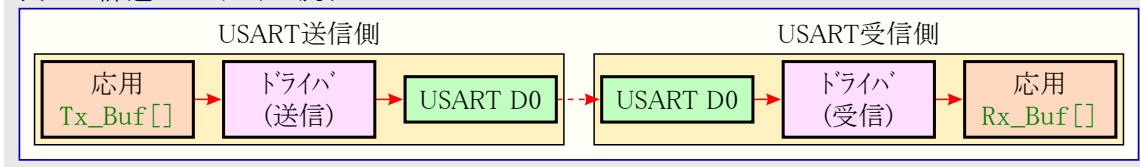
割り込み形態でのUSARTは送信レジスタが空か、または受信レジスタが新規データを含むかを見るためにマイクロコントローラがそれらをポーリングしないので、CPU周期を自由にします。このように、マイクロコントローラは待つよりもその他やもっと有用なことを行うことができ、その能力の使い方を増します。

ATMEL XMEGAの新規はデータを保持するための3バイトのハードウェア緩衝部です。この緩衝部の利点は緩衝部溢れの発生を減らすことです。例えばより高い優先権を持つ割り込み処理ルーチン(ISR)がより低い優先権を持つ他のISRを兵糧攻め(脚注)にする時に緩衝部溢れが起き得ます。

割り込み形態でのUSARTを構成設定するのにドライバが使用されます。ドライバは実装された環状緩衝部を持ち、これは簡単に調べられるでしょう。

図4-1.で見られるように、応用は受信するUSARTへケーブルを経由してそれを転送するドライバにデータを送ります。受信部ではドライバが応用にデータを送ります。

図4-1. 課題2でのデータの流れ



この課題の目的は以下を行うことです。

- 割り込み形態でATMEL XMEGAのUSARTを構成設定する方法を知る。
- 構成設定に関してドライバの使用法を理解する。
- ドライバ内の環状緩衝部をさっと調べてください。
- ハードウェア緩衝部がどう動くかを理解する。

**脚注:** 高優先権割り込みが繰り返し走行している故に、より低い優先権の割り込みルーチンは決して走行のための時間(CPU周期)を得られません。



すること

1. `InterruptControlled.aps`プロジェクトを見つけ出して開いてください。
2. ポートDのTxとRx、即ちPD2とPD3間をジャンパで接続してください。
3. コード(`task2.c`)全体を通して見て何が起るかを理解してみてください。
4. コードをコンパイルして警告や異常がないことを保証してください。
5. デバッグ作業を開始してください。
6. `USART_InterruptDriver_Initialize()`を内側実行(Step into, **F11**を押)してドライバファイルにアクセスしてください。
7. ドライバがレジスタをどう構成設定するかを理解してみてください。ドライバの外側実行、**Shift+F11**を押してください。
8. `receiveArray[]`緩衝部を見つけ出して監視(Watch)に加えてください(図4-2.をご覧ください)。

図4-2. 緩衝部を監視(Watch)に追加

Name	Value	Type	Loca
receiveArray	[...]	uint8_t[3]	0x2
[0]	0 ''	unsigned char	0x2
[1]	0 ''	unsigned char	0x2
[2]	0 ''	unsigned char	0x2

9. さて、**F5**を押すことによって暫くの間コードを走らせてください。
10. 短時間の後、実行を中断(**Ctrl+F5**を押)してください。
11. さて、`receiveArray[]`を調べてください。文字が正しく転送されていましたか？。
12. デバッグ作業をリセット(**Shift+F5**)して割り込み処理ルーチン(ISR)に中断点を配置してください。プログラムを走らせて期待したように働くか見てください。
13. `usart_driver.h`を開いて`USART_RX_BUFFER_SIZE`と`USART_TX_BUFFER_SIZE`の両方を2に設定してください。
14. `Task2.c`で`NUM_BYTES`を7に設定してプロジェクトを再コンパイルしてください。
15. ATMELのAVR StudioのメニューでDebug⇒Remove all Breakpoints(全中断点削除)を押してください。
16. 暫くの間プログラムを(**F5**を押して)走らせて実行を中断(**Ctrl+F5**を押)してください。



何故、プログラムは完全に走行することができないのですか？。(助言:ソフトウェア環状緩衝部とハードウェア緩衝部の容量はいくつで、送るための配列の容量はいくつでしょうか?)

## 5. 課題3: DMAを用いるUSART

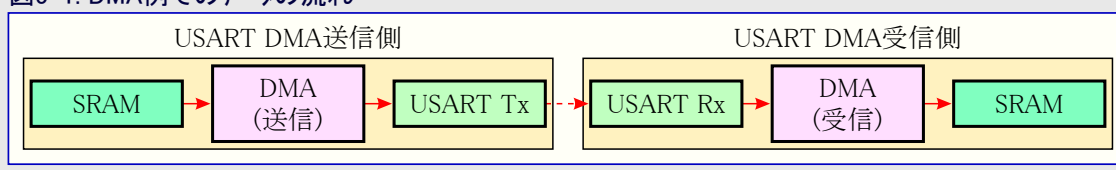
ATMELのXMEGAは8ビットマイクロコントローラに対してDMA制御器を導入します。DMAを使用することはデータ伝送を扱う時にCPUの負荷を無くしてマイクロコントローラの性能をかなり増すのを手助けします。

この課題はDMACドライバとでUSARTを構成設定する方法を示します。図5-1.はDMAを用いたこの課題に於けるデータの流れを図解します。

この課題の目的はあなたが以下の方法を知ることです。

- DMACを用いるUSARTの構成設定
- SRAMからデータを読むようにDMACを構成設定
- USARTへデータを書くようにDMACを構成設定
- USARTからデータを読むようにDMACを構成設定
- SRAMへデータを書くようにDMACを構成設定

図5-1. DMA例でのデータの流れ





すること



1. ATMELのAVR StudioでUSART\_DMA.apsプロジェクト ファイル開いてtask3.cを開いてください。
2. ポートDのTxとRx、即ちPD2とPD3間をジャンパで接続してください。
3. 送信チャンネルがどう構成設定されるか(SetupTransmitChannel)を学んでください。
  - a. Tx\_BufはDMA送信チャンネル用の入力です。
  - b. DMAはTx\_Bufのアドレスを増すように構成設定されます。何故?。
  - c. USARTデータレジスタはDMA送信チャンネル用の出力です。
  - d. DMAは伝送中にデータレジスタを固定するためにアドレスを維持するように構成設定されます。何故?。
  - e. ATMELのXMEGA手引書で正しい起動元(データレジスタ空き)を確かめることができますか?。(6C)
4. 受信チャンネルがどう構成設定されるか(SetupReceiveChannel)を学んでください。
  - a. USARTデータレジスタはDMA受信チャンネル用の入力です。
  - b. Rx\_BufはDMA受信チャンネル用の出力です。
  - c. DMAは固定の受信データレジスタアドレスを持ち、Rx\_Bufのアドレスを増すように構成設定されます。何故?。
  - d. XMEGA手引書で正しい起動元(受信完了)が使用されているのを確かめることができますか?。
5. プロジェクトを構築してデバッグ作業を開始(Playアイコンをクリック)してください。
6. DMAが完了した("LEDPORT.OUT = ~")後の最初の行に中断点を置いてください。コードを走らせて(F5を押して)ください。
7. LEDで何が示されましたか?。CPUはDMAの完了を待つ間に変数iを増す(+1)ことができますか?。
8. LEDPORT.OUT = Rx\_Buf[i];行に中断点を置いてください。
9. コードを走らせて(F5を押して)ください。
10. コードとの比較でLEDの状態を調べてください。意図したようにLEDが点滅しますか?。
11. Rx\_BufとTx\_Bufを監視(Watch)に追加してそれらを比べてください。それらは等しく、a~tの文字を含みます。図5-2をご覧ください。



図5-2. DMA転送検証

Watch				
Name	Value	Type	Location	
<input checked="" type="checkbox"/> Tx_Buf	[...]	char [20]	0x2000	[SRAM]
<input type="checkbox"/> Rx_Buf	[...]	char [20]	0x2014	[SRAM]
[0]	0x61 'a'	char	0x2014	[SRAM]
[1]	0x62 'b'	char	0x2015	[SRAM]
[2]	0x63 'c'	char	0x2016	[SRAM]
[3]	0x64 'd'	char	0x2017	[SRAM]
[4]	0x65 'e'	char	0x2018	[SRAM]
[5]	0x66 'f'	char	0x2019	[SRAM]
[6]	0x67 'g'	char	0x201A	[SRAM]
[7]	0x68 'h'	char	0x201B	[SRAM]
[8]	0x69 'i'	char	0x201C	[SRAM]
[9]	0x6A 'j'	char	0x201D	[SRAM]
[10]	0x6B 'k'	char	0x201E	[SRAM]
[11]	0x6C 'l'	char	0x201F	[SRAM]
[12]	0x6D 'm'	char	0x2020	[SRAM]
[13]	0x6E 'n'	char	0x2021	[SRAM]
[14]	0x6F 'o'	char	0x2022	[SRAM]
[15]	0x70 'p'	char	0x2023	[SRAM]
[16]	0x71 'q'	char	0x2024	[SRAM]
[17]	0x72 'r'	char	0x2025	[SRAM]
[18]	0x73 's'	char	0x2026	[SRAM]
[19]	0x74 't'	char	0x2027	[SRAM]

## 6. 要約

ここはこの練習からの要点のいくつかです。

- ポーリング形態でのUSART
- 割り込み形態でのUSART
- USARTドライバ
- USARTソフトウェア環状緩衝部
- USARTハードウェア緩衝部
- USART DMA

## 7. 資料

- XMEGAの手引書とデータシート
  - <http://www.atmel.com/xmega>
- ATMELのヘルプ ファイル付きAVR studio
  - <http://www.atmel.com/products/AVR/>
- WINAVR GCCコンパイラ
  - <http://winavr.sourceforge.net/>
- ATMEL用IAR Embedded Workbench<sup>®</sup>コンパイラ
  - <http://www.iar.com/>

## 8. ATMEL技術支援センター

ATMELは以下の利用可能な多数の支援チャネルを持ちます。

- ウェブ入り口 : <http://www.atmel.no/> 全てのATMELマイクロ コントローラ
- Eメール : [avr@atmel.com](mailto:avr@atmel.com) 全てのATMEL AVR製品
- Eメール : [avr32@atmel.com](mailto:avr32@atmel.com) 全ての32ビットAVR製品

以下のサービスへのアクセスを得るにはウェブ入り口で登録してください。

- 豊富なFAQデータベースへのアクセス
- 技術支援要請の容易な依頼
- あなたの過去の全支援要請の履歴
- ATMELマイクロ コントローラ時事通信の受信のための登録
- 利用可能な練習と練習材料についての情報取得



## 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](http://www.atmel.com)

### 技術支援

[avr@atmel.com](mailto:avr@atmel.com)

### 販売窓口

[www.atmel.com/contacts](http://www.atmel.com/contacts)

### 文献請求

[www.atmel.com/literature](http://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 2010. 全権利予約済 ATMEL<sup>®</sup>、ロゴとそれらの組み合わせ、AVR<sup>®</sup>とその他はATMEL Corporationの登録商標、XMEGA<sup>®</sup>とその他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AVR1510応用記述(doc8319.pdf Rev.8319A-06/10)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。