

# AVR1514 : XMEGA-A1 Xplain練習

## - 直接メモリ入出力制御器

### 前提条件

- 必要な知識
  - ・ AVR1512:XMEGA-A1 Xplain練習 - XMEGA基礎の完了
- ソフトウェア必要条件
  - ・ ATMEL® AVR Studio® 5
- ハードウェア必要条件
  - ・ XMEGA-A1 Xplain評価基板
  - ・ JTAGICEmk3(またはJTAGICEmk II かAVR ONE!)
- 予想完了時間
  - ・ 2時間

### 1. 序説

この応用記述はATMEL® AVR® XMEGA®の直接メモリ入出力制御器(DMAC)の基本的な機能を網羅します。この練習の目的はCPU時間を殆ど使用することなく、簡単なメモリ転送を開始させ、そして殆どどんなCPUの介入もなく周辺機能に読み/書きすることです。

個別の転送元、転送先、起動、塊容量を持つ4つのDMAチャネルがあります。DMA制御器は或るメモリ領域から別のメモリ領域、メモリと周辺機能間、周辺機能間でデータを移動できます。

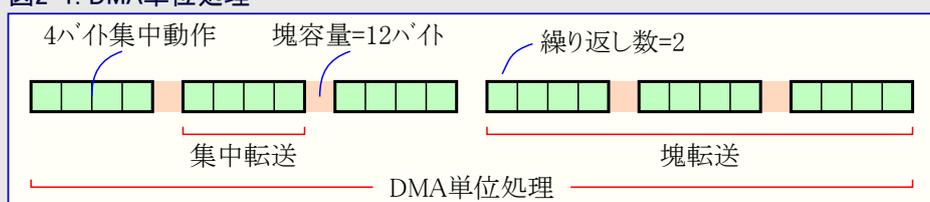
### 2. DMA制御器の構成設定

ATMEL® AVR® XMEGA®の直接メモリ入出力制御器(DMAC)は最小のCPU介在でメモリと周辺機能間でデータを転送する能力を持つ高い柔軟性の4チャネルDMA制御器です。CPUが低電力休止形態で時を費やす、または他の作業を実行している間に、XMEGAのDMACは或る領域から別の領域へ単にデータの複写を処理することによってCPUの負荷を無くします。以降の副節はこの練習に於いて課題を行う時の理解を補完することを目論まれています。

#### 2.1. DMA単位処理

メモリや周辺機能間での完全なDMA読み書き操作はDMA単位処理と呼ばれます。単位処理はデータの塊で行われ、単位処理の量(転送バイト数)はソフトウェアから選択可能で塊量と繰り返し計数器の設定によって制御されます。各塊転送はより小さな集中(転送)に分割されます(図2-1.をご覧ください)。

図2-1. DMA単位処理



DMAチャネルがデータ転送を要求すると、バス調停器はAVRコアがデータバスを使用しないようになるまで待つてDMA制御器にデータの転送を許します。転送は1,2,4または8バイトの集中(転送)で行われます。アドレス指定は静止、増加、減少にできます。各集中転送、塊転送後、転送完了時、または禁止時に、転送元や転送先のアドレス自動再設定を行うことができます。DMA転送は応用ソフトウェア、周辺機能、それと事象によって起動することができます。

塊転送の容量は塊転送計数レジスタによって設定され、1バイトから64Kバイトまでのどれにでもできます。繰り返し計数器は単位処理が完了する前の繰り返し塊転送数の設定を許すことができます。繰り返しは1から255までで、繰り返し回数を0に設定することによって無制限の繰り返し回数を達成することができます。

バス調停器はDMA制御器とAVRコアがバスを使用できる時を制御します。コアが常に優先権を持ち、故にコアがバスへのアクセスを要求する限り、どの保留集中転送も待たなければなりません。コアはSRAM、I/Oメモリ、EEPROM、外部バスインターフェースとデータを読み書きする命令を実行する時にバスのアクセスを要求します。



8ビット ATMEL  
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. 8400A-06/11, 8400AJ1-03/14

## 2.2. アドレス指定

転送元や転送先がSRAMの場合、使用者は多分、アドレス指示子の増加または減少を望みそうです。従って、増加または減少の形態でDMACの転送元/転送先を設定することが可能です。

DMACは増加動作形態でアドレス指示子を自動的に増加します。

SRAMアドレス	値
\$2066	34
\$2067	76
\$2068	13
\$2069	113

増加形態ではDMACが転送元アドレスで始まります。最初のバイトが送られてしまった後、次のバイトが直前のアドレス+1から読まれます。減少形態ではこれが逆で、DMACは与えられた転送元アドレスで始まり、前(-1)のアドレスで進みます。

元の転送元と転送先のアドレスはDMA制御器によって保存され、故に転送元と転送先のアドレスは以下の時点で再設定するように個別に形態設定できます。

- ・各集中転送の最後
- ・各塊転送の最後
- ・転送処理単位の後
- ・再設定なし

SPIのような周辺機能でDMACを使用する時はデータレジスタが固定で、アドレス指定形態が静止であることが重要です。

## 2.3. 転送の起動

DMA起動の主な目的は転送速度で周辺機能を同期することです。例えば、9600bpsでUSARTを用いて転送する時に、DMAは大抵転送完了フラグの設定(1)で起動されるべきではありません。

DMA転送はDMA転送要求が検出された時にだけ開始することができます。転送要求はソフトウェア、外部起動元(周辺機能)、または事象から起動することができます。各DMAチャンネルに対して専用の起動元選択があります。利用可能な起動元はデバイス毎に変わるかもしれませんが、デバイスに存在する部署や周辺機能に依存します(各種転送起動元についてはXMEGA手引書をご覧ください)。

## 2.4. 割り込み

DMA制御器はDMAチャンネルで異常が検出された時、またはDMAチャンネルに関して転送が完了した時に割り込みを生成することができます。各DMAチャンネルは独立した割り込みベクタを持ち、異常と転送処理単位完了に対して異なる割り込みフラグがあります。例え無制限繰り返し許可されていても、転送処理単位完了フラグは各塊転送の最後で設定(1)されます。

## 3. 概要

ここはこの練習に於ける課題の短い概要です。

### 課題1. メモリ複写

この課題はCPUの介在なしに或る位置から別の位置にメモリの塊を複写する方法を示します。この課題は以下のように2つの部分に分けられます。

1. 最初の部分はDMACでSRAMのメモリを複写する方法を示します。
2. 2つ目の部分は転送が終わったことを示すように割り込みを構成設定する方法を示します。

### 課題2. 割り込み形態

この課題は10秒間のスイッチ押下の流れを記録してLEDでそれらを再生する簡単な記録器の実装方法を示します。記録と再生の処理はCPUの介在なしで実行されます。

## 4. 課題1: メモリ複写

DMA制御器は或る位置から別の位置へメモリの大きな塊を複写するのに理想的です。この課題はドライブファイルからの塊メモリ複写ルーチンを実演します。転送が終わった時に割り込みを得るためにPMICを構成設定する方法も示します。

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を構成設定するためにドライブコードを用いてメモリ複写操作を開始
- ・転送が終わった時に割り込みを起動するようにPMICを構成設定

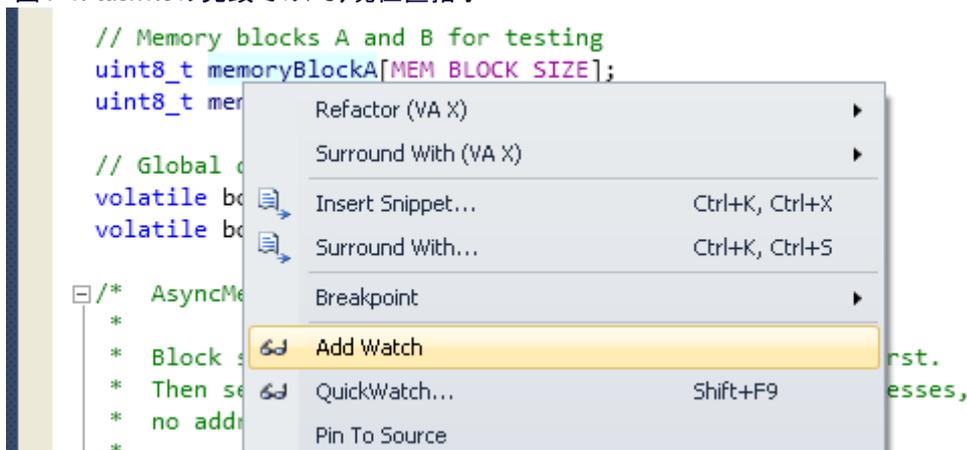
### 4.1. メモリ複写: ホーリング手法



すること

1. Atmel XMEGA-DMACフォルダに位置付けてください。
2. `xmega_dmac.avrsln`解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによってTask1を活性化に設定してください。
3. プロジェクトを構築(F7を押)してデバッグを始めて(Play鈕をクリックまたはAlt+F5を押下して)ください。
4. `memoryBlockA`と`memoryBlockB`の定義を探し出してそれらの両方に対してデータ監視(Watch)を追加(右クリックでのメニューで“Add watch...”)してください(図4-1.をご覧ください)。

図4-1. task1.cの先頭でのメモリ塊位置指示



5. main()でMemCopyに対する呼び出し上で右クリックして右クリックでのメニュー(またはCtrl+F10)で“Run to Cursor”を選んでください(図4-2をご覧ください)。
6. 監視(Watch)ウィンドウでmemoryBlockAとmemoryBlockBを展開してそれらの内容を調べてください。それらは等しいですか?(図4-3をご覧ください)。

図4-2. カーソル位置まで実行

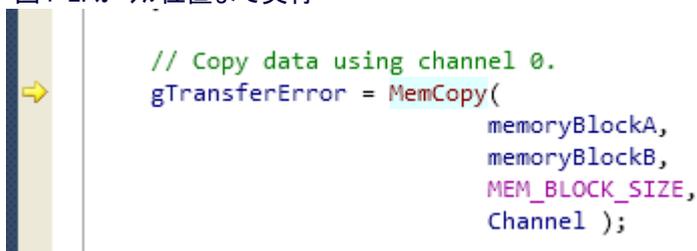


図4-3. memoryBlockAとmemoryBlockBの比較

Watch 1		
Name	Value	Type
memoryBlockA	{uint8_t[100]{data}@0x2066}	uint8_t[100]{data}@0x2066
[0]	0	uint8_t{data}@0x2066
[1]	1	uint8_t{data}@0x2067
[2]	2	uint8_t{data}@0x2068
[3]	3	uint8_t{data}@0x2069
[4]	4	uint8_t{data}@0x206a
[5]	5	uint8_t{data}@0x206b
[6]	6	uint8_t{data}@0x206c
[7]	7	uint8_t{data}@0x206d
[8]	8	uint8_t{data}@0x206e

7. MemCopy呼び出しを外側実行(Step over)して再びmemoryBlockAとmemoryBlockBの内容を調べてください。今はそれらが等しいですか?。
8. 実行をリセット(Shift+F5)して再びMemCopyを走らせてください。今回はこの関数をアクセスするのに内側実行(Step into: F11押下)を行ってください。更に、ドライバがDMACをどう形態設定するかを見るためにDMA\_SetupBlockも内側実行してください。これらの問いを答えるためにあなたは導入節の参照を望むかもしれません。
  - a) メモリ複写は静止、増加、減少のどれとして構成設定されますか?。何故このように構成設定されるのでしょうか?。
  - b) 別のDMAチャネルを使用することができましたか?。
  - c) このメモリ間転送に関して既定としてソフトウェア起動元が使用されます。SRAMからSRAMへの複写時に何故これが便利と考えますか?。
  - d) 集中転送長はいくつですか?。何故これが形態設定可能と考えますか?。

## 4.2. メモリ複写: 割り込み手法



課題:

1. 図4-4.で示されるように、試験2の開始へ中断点を追加してください。そして走行するためにF5を押してください。

すること 図4-4. ここに中断点追加

```
// Enable LO interrupt level for the complete transaction and
// error flag on DMA channel 0.
DMA_SetIntLevel( Channel, DMA_CH_TRNINTLVL_LO_gc, DMA_CH_ERRINTLVL_LO_gc );
PMIC_CTRL |= PMIC_LOLVLEN_bm;
```

2. 直前の試験が快調(転送で異常なし)なら、次の試験の準備を整えます。いくつかの段階を1行実行(F10押下)し、次のDMA転送単位処理完了時に割り込みが起動するように構成設定されることに気をつけてください。
3. AsyncMemCopy関数を考察してください。MemCopyと比べて、DMA\_ReturnStatus\_non\_blocking呼び出しがないことに注意してください。

DMA\_ReturnStatus\_non\_blocking関数はDMA転送が終わるまでホールドします。転送終了を示す代わりに割り込みを使用する時に、CPUは他の操作を行うように自由になります。

4. task1.cの下部に於いて、図4-5.で示されるように、ISR(DMA\_CH0\_vect)割り込み処理ルーチンで中断点を追加してください。走行するためにF5を押してください。

図4-5. 割り込み待機

```
// DMA CH0 Interrupt service routine. Clear interrupt flags after check
ISR(DMA_CH0_vect)
{
    if (DMA_INTFLAGS & DMA_CH0TRNIF_bm)
    {
        DMA_INTFLAGS |= DMA_CH0TRNIF_bm;
        gTransferError = false;
    } else
    {
        DMA_INTFLAGS |= DMA_CH0ERRIF_bm;
        gTransferError = true;
    }
    gInterruptDone = true;
}
```

5. 完了するために走行(F5を押)し、暫くして実行を中断(Ctrl+F5)してください。memoryBlockAとmemoryBlockBを比べることによって転送が上手く行われたことを検証してください。

## 5. 課題2: 記録器

DMACが実際に有用な応用はADCやDACのような周辺機能とメモリ間でデータを複写することです。この課題は或る時間の間スイッチを採取し、SRAM緩衝部にそれを格納し、その後何度も何度もLEDでデータを再生する簡単な記録器の実装方法を示します。ADCで音を記録してDACでそれを再生する場合、正確に同じ原理が用いられます。



この課題の目的はあなたが以下の方法を知ることです。

- 周辺機能からSRAMへデータを複写するようにDMACを形態設定
- SRAMから周辺機能へデータを複写するようにDMACを形態設定
- DMAデータ転送用起動元として計時器の計時を使用



すること

1. Atmel XMEGA-DMACフォルダに位置付けてください。
2. xmega\_dmac.avrln解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによってTask2を活性に設定してください。
3. プロジェクトを構築(F7を押)し、デバッグを開始(Play鈕をクリック)して、そして記録器を使用するためにコードを走らせて(F5を押)してください。
  - a) 一度LEDが発光します。
  - b) どれかスイッチを押してください。
  - c) LEDが再び発光します。
  - d) DMACがスイッチの状態を記録する間にスイッチを叩くのに、概ね10秒あります。
  - e) LEDが再び発光します。
  - f) どれかスイッチを押してください。



- g) LEDが再び発光します。
  - h) DMACはスイッチが押されるまでLED上でスイッチの記録を再生します。(注)
  - i) この手順をやり直してください。
4. SAMPLE\_COUNT定義または採取速度を変更してみてください。再コンパイルして走らせて、何が起きるかを見てください(図5-1.をご覧ください)。

図5-1. SAMPLE\_COUNT変更試行

```
#define SAMPLE_COUNT 300 // How many switch state samples to store.
uint8_t samples[SAMPLE_COUNT]; // Store switch state samples here.
```

5. コードを簡単に片付けて何が起きるかを理解してみてください。これらの問いに答える時に導入節を参照するかもしれません。
- a) SetupReadChannelに於いて、何が転送元で何が転送先ですか？。
  - b) SetupWriteChannelに於いて、何が転送元で何が転送先ですか？。
  - c) ソフトウェア起動元でDMACを構成設定したなら、その場合は何が起きますか？。
  - d) 読み込みチャネルに関して、何故転送元アドレスが固定で転送先アドレスが増加なのですか？。
  - e) 集中(転送)長は4や8のような違う長さにはできますか？。



(注) XMEGA-A1 Xplain使用時にLED6とLED7が常に点灯することに注意してください。これに関する理由はPORTDとPORTR間にスイッチが配給されているからです。

## 6. 要約

- ・メモリ複写、SRAMからSRAMへのDMAC。
- ・転送の最後を示すための割り込み元の使用。
- ・DMACを用いたスイッチ押下記録。この応用はポートピンの入力状態を記録してSRAMにその情報を保存しました。SRAMに保存された情報はその後LEDに出力設定されました。起動元として計時器溢れが使用されました。

## 7. 資料

- ・XMEAの手引書とデータシート
  - <http://www.atmel.com/xmega>
- ・ATMEL AVR Studio 5
  - [http://www.atmel.com/microsite/avr\\_studio\\_5](http://www.atmel.com/microsite/avr_studio_5)
- ・ATMEL用IAR Embedded Workbench®コンパイラ
  - <http://www.iar.com/>

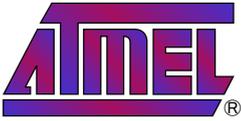
## 8. ATMEL技術支援センター

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



#### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL (+1)(408) 441-0311  
FAX (+1)(408) 487-2600  
[www.atmel.com](http://www.atmel.com)

#### *Atmel Asia Limited*

Unit 01-5 & 16, 19F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG  
TEL (+852) 2245-6100  
FAX (+852) 2722-1369

#### *Atmel Munich GmbH*

Business Campus  
Parking 4  
D-85748 Garching b. Munich  
GERMANY  
TEL (+49) 89-31970-0  
FAX (+49) 89-3194621

#### *Atmel Japan*

141-0032 東京都品川区  
大崎1-6-4  
新大崎勸業ビル 16F  
アトメル ジャパン合同会社  
TEL (+81)(3)-6417-0300  
FAX (+81)(3)-6417-0370

#### © 2011 Atmel Corporation. 全権利予約済

ATMEL<sup>®</sup>、ATMELロゴとそれらの組み合わせ、それとXMEGA<sup>®</sup>、AVR Studio<sup>®</sup>、AVR<sup>®</sup>、AVR<sup>®</sup>ロゴとその他は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。

**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、または利益と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。

#### © HERO 2014.

本応用記述はATMELのAVR1514応用記述(doc8400.pdf Rev.8400A-06/11)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。