

AVR1516 : XMEGA-A1 Xplain練習 – XMEGA 事象システム

前提条件

- 必要な知識
 - ・ マイクロコントローラとCプログラミング言語の基礎知識
 - ・ AVR1512:XMEGA-A1 Xplain練習 – XMEGA 基礎の完了
 - ・ AVR1513:XMEGA-A1 Xplain練習 – XMEGA タイマ/カウンタの終了を推奨
- 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

事象システムは周辺機能間通信に関する機能の組です。これは或る周辺機能での状態の変化に対して他の周辺機能での自動的な活動起動の可能性を許します。周辺機能での何の状態変化が他の周辺機能での活動起動となるかはソフトウェアで形態設定可能です。これは簡単ですが、それが割り込みまたはCPUそれとDMA資源のどれをも使用することなく周辺機能の自律制御を許すため、強力なシステムです。

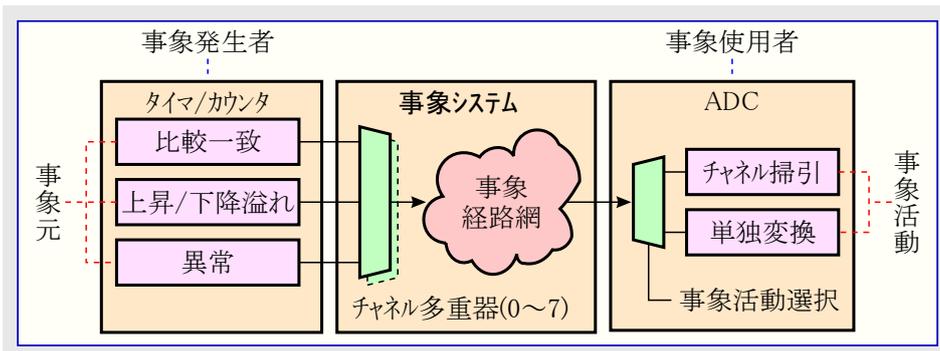
周辺機能での状態の変化の指示は事象として参照されます。事象は事象経路網と呼ばれる専用の配線網を使用して周辺機能間で渡します。これは事象が全ての多重器に配線される、8つの多重器から成ります。

2. 事象システムの紹介

この紹介は事象システムとこの練習での課題を理解するのに必要とされる用語の基礎概要と動きを与えることが意図されています。この練習での課題は事象システムがどう動くかをより詳細に示します。

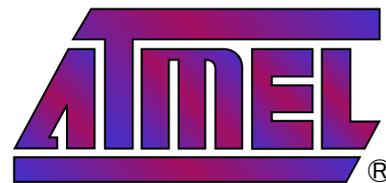
下図は事象システムを図解します。この図はこれを動かす各種部分、事象元、チャネル多重器、事象使用者/周辺機能での事象活動選択を示します。

図は事象生成者としての1つのタイマ/カウンタと事象使用者としての1つのA/D変換器(ADC)での簡易化した版を示します。事象チャネル多重器は対応する事象チャネルを通して配線されるべき3つの利用可能な供給元の1つを選ぶことができます。



事象は以下の周辺機能によって生成することができます。

- ・ タイマ/カウンタ (TCxn)
- ・ 実時間計数器 (RTC)
- ・ A/D変換器 (ADCx)
- ・ アナログ比較器 (ACx)
- ・ ポート (PORTx)
- ・ システムクロック (clk_{sys})



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. 8402A-07/11, 8402AJ1-03/14

これらの各周辺機能は事象用に様々な供給元を持ちます。供給元の例はタイマ/カウンタ溢れ、ポートでのピン変化、またはA/D変換完了です。利用可能な事象元の完全な一覧はATMEL® AVR® XMEGA® AU手引書で事象システムに関するレジスタ説明で示されます。

チャンネル多重器(MUX)は利用可能な8つの事象システムの各々に配線された何かの供給元を選びます。各々の事象システムチャンネルはそのチャンネルのために事象を生成する1つの供給元を許します。事象チャンネルx多重器(EVSY.S.CHxMUX)レジスタは各チャンネルに対する事象元を制御します。

事象は以下の周辺機能によって使用され得ます。

- ・タイマ/カウンタ (TCx_n)
- ・A/D変換器 (ADC_x)
- ・D/A変換器 (DAC_x)
- ・直接メモリ入出力制御器 (DMAC)

事象の使い方は個別の周辺機能で制御されます。個別周辺機能の形態設定レジスタは入力として使用するどれかのチャンネルと、そのチャンネルに対する、どれかの事象活動の選択を許します。多数の周辺機能が入力として同じ事象チャンネルを使用することができます。これは同時に多数の活動開始を許すのに便利です。

例えば、A/D変換器で変換を開始すると同時にタイマ/カウンタの捕獲を開始することです。利用可能な事象活動は各々の周辺機能に対するレジスタ説明で示されます。

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. 32ビット タイマ/カウンタ

この課題は事象使用者と事象生成者での基本事象システム構成設定と、これが32ビット計時器を作るのにどう使用され得るかを示します。

課題2. 濾波付き捕獲

タイマ/カウンタでの捕獲入力はATMEL XMEGAでの事象で制御され、この課題はこれがどのように柔軟なのかを示します。

課題3. 同期した起動

1つよりも多くの周辺機能が1つの事象チャンネルからの事象を使用することができ、これは周辺機能での同期事象活動に使用することができます。

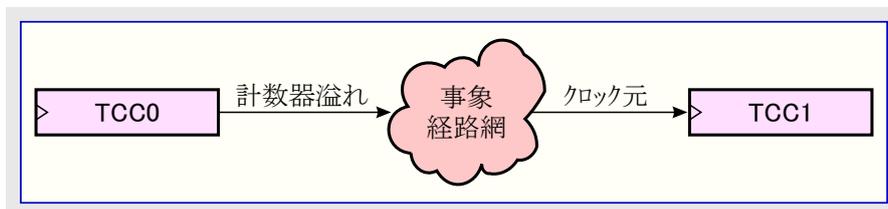
課題4. 手動生成事象

事象はソフトウェアから生成することができ、この課題はこれをどう行うのかの基本例を与えます。

ご幸運を!

4. 課題1: 32ビット タイマ/カウンタ

クロック入力/供給元として或るタイマ/カウンタから別のタイマ/カウンタに溢れ事象を用いることにより、32ビット タイマ/カウンタを作るのに事象システムを使用することが可能です。この構成では32ビット捕獲入力を作ることも可能です。タイマ/カウンタ実践作業に於いて、タイマ/カウンタTCC0への入力として周辺クロックを使用し、この課題に於いて事象システムはタイマ/カウンタTCC1への入力として事象システムを使用します。以下の図はこの概念を示します。



「AVR1001:XMEGAの事象システムでの着手に際して」この応用記述は捕獲入力を持つ32ビット タイマ/カウンタの実装法のコード例を含みます。

この課題の目的はあなたが以下の方法を知ることです。

- ・事象システムの使い方の基礎の理解と事象チャンネルの形態設定法
- ・周辺機能部署での事象チャンネル使用のこつ
- ・計時器をクロック駆動するための使用法の理解

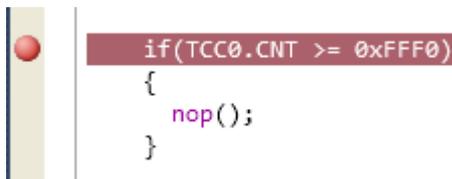


課題:

すること

1. Atmel XMEGA-EventSystemフォルダに位置付けてください。xmega_eventsystem.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を活性に設定してください。
2. コード、それがどう動くのかを理解するのに或る程度時間を費やし、事象システム構成設定法の基礎知識を確実にしてください。

- プロジェクトを構築し、異常がないことを確実にして(警告は無視できます。)デバッグ作業を開始してください。
- コードを走らせてください。事象システムからクロック駆動されている最上位TCのクロック計数速度でLEDが上向きで計数しているのを見ることがでしょう。
- 中断して下で示されるように中断点を配置してください。



```

if(TCC0.CNT >= 0xFF0)
{
    nop();
}

```

- 再びコードを走らせ、最下位計時器が溢れ近くの時に中断することを確認してください。数回1行実行(Single step)したなら、計時器が溢れ、LEDが計数しているのを見ましょう。TCC1に対するI/Oウィンドウを展開した場合、TCC0溢れ時にCNTがどう増加するかを見ることができます。
- この課題にいくつかのコードを書こうと思うなら、48や64ビット計時器を作るためにコードを追加することができます⑥。

5. 課題2: 濾波付き捕獲

課題1ではタイマ/カウンタ(TC)へのクロック元である事象を起動するのに事象システムを使用しました。一般的に、周辺機能は事象を受け取る時に各種活動を実行できます。例えば、TCに関して、計時器事象活動はATMEL XMEGA A手引書のタイマ/カウンタのレジスタ説明で示される制御レジスタD(CTRLD)で示されます。計時器がどの事象活動を使用することができるかを理解するためにこの表に目を通してください。

捕獲はATMEL XMEGA上の事象システムを通してだけ利用可能です。どれかの入出力ピンの変化が事象を起動するのに使用できるため、全ての入出力ピンを捕獲入力として使用することができます。

事実、どの事象もピン変化だけでなく、捕獲入力を起動することができます。この課題では基本に忠実に、捕獲入力としてピン変化事象を用います。この作業をするために以下の形態設定が必要です。

- チャンネル0で事象がある時に捕獲チャンネルA(CCA)で捕獲入力を実行する計時器TCC0
- 事象チャンネル0にポートEのピン0事象を経路付けるために配線する事象システム
- 上昇端、下降端、両端、またはピンのレベルで事象を生成するかを指定するためのポートEピン0での入力感知

この課題の目的はあなたが以下の方法を知ることです。

- 計時器と事象システムを用いて捕獲入力を構成設定する方法の理解
- 事象システムによって処理される入出力ピン上での濾波のこつ



すること

課題:

- Atmel XMEGA-EventSystemフォルダに位置付けてください。xmega_eventsystem.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Task2を活性に設定してください。
- このコードは殆ど終了していますが、この事象チャンネルへの入力として切り替えポートのピン0を使用するように事象チャンネル0の多重器(MUX)形態設定が必要です。手助けを必要とするなら、課題1でこれがどう行われたかを見てください。
- プロジェクトを構築し、異常がないことを確実にして(警告は無視できます。)、AVR Studioでデバッグファイルを開いてください。
- コードを走らせてください。スイッチを数回押してください。
- 各押下が捕獲入力を起動する事象(また、実際には2つの事象)を生成します。捕獲値は継続的に読まれてLEDに出力されます。



各押下が2つ事象を生成するのは何故ですか？。

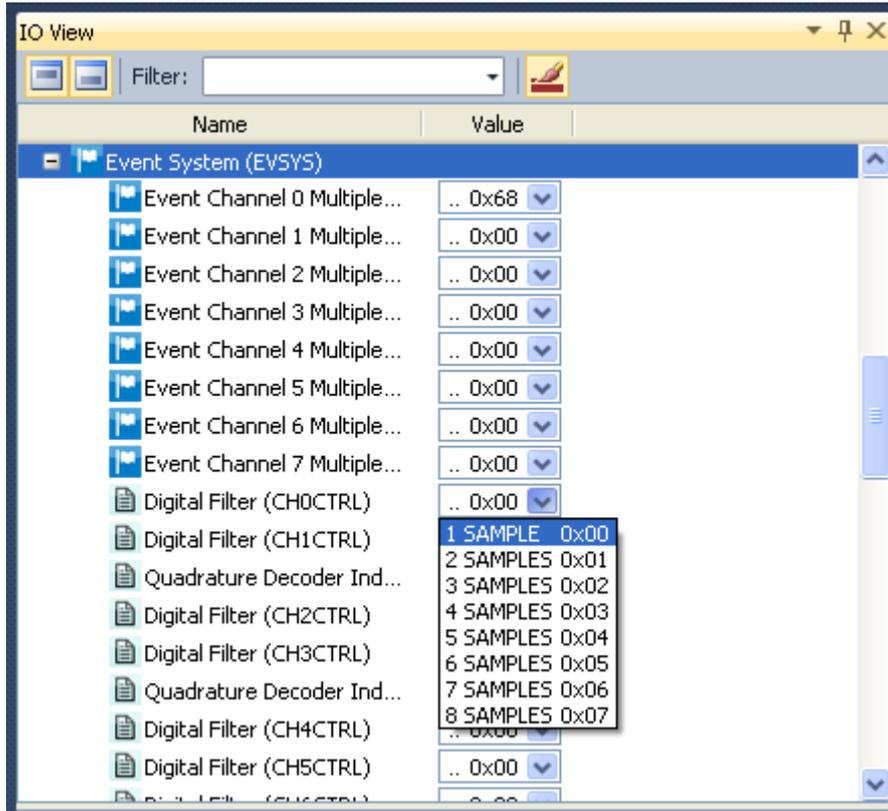
- 異常フラグを調べてみましょう。異常フラグはTCの捕獲レジスタに関して緩衝溢れがある時に設定(1)されます。これがどう動くのかを見るためにATMEL XMEGA A手引書で14.5.項(2重緩衝)を学習することができます。
- 溢れ異常数を追いかけることができるようにerror_countを監視(Watch)ウィンドウに追加してください。



Name	Value	Type
error_count	5	uint16_t(data)@0x3ff7 (FP+1)

- コードを走らせ、短い間隔で多数回スイッチを押し、中断してどんな異常が起こったかを見てください。あなたは何かを生成することができるべきです。
- 緩衝溢れを得るのに充分素早く生成するのに鉤からの雑音を使用します。溢れを生成するようにスイッチを押すのが難しいなら、CCAレジスタが度々読まれないようにwhile(1)繰り返しの遅延(例えば、_delay_ms(200);)を追加できます。

10. デバッグ作業が未だ停止されている間に、I/Oウィンドウで事象システムを探し出して現在の形態設定を見るためにそれを展開してください。



11. デジタル濾波器の最初の事例は事象チャンネル0に対する濾波器を表します。
12. 事象が全体に渡される前に周辺クロックによって採取されなければならないピン変件事象数を設定するようにデジタル濾波器を変更してください。
13. 再びコードを走らせ、スイッチを押して異なる濾波器値がerror_countをどう減らせるかを見てください。

6. 課題3: 同期した起動

異なる周辺機能に入力として同じ事象チャンネルを使用することが可能で、これを行うことによって多数の周辺機能が同じ事象を使用することができます。これは例えば同期活動に用いることができます。ここはこれが有用な少しの例です。

- 変換に時刻印を与えるために、同時に捕獲を起動してA/D変換を開始すること。
- 2つのA/D変換器で同時に変換を開始すること。
- A/D変換器、D/A変換器、タイマ/カウンタ、DMAのその他の組み合わせ

この課題では同時に3つの計時器(TCC0, TCD0, TCE0)で捕獲を始めるために、単一を保って事象システムを使用するつもりです。事象起動元(生成者)としてタイマ/カウンタのTCF0溢れが使用されます。

この課題の目的はあなたが以下の方法を知ることです。

- 事象システムを形態設定できる。
- 同期起動の理解。



課題:

すること

1. **Atmel XMEGA-EventSystem**フォルダに位置付けてください。 **xmega_eventsystem.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**Task3**を活性に設定してください。
2. コードは殆ど完成していますが、いくつかの変更を行うことが必要です。
 - a. TCC0と同じ方法でタイマ/カウンタのTCD0とTCE0を構成設定するコードを加えてください。
 - b. 事象チャンネル0への入力としてタイマ/カウンタTCF0の溢れを構成設定してください。
 - c. プロジェクトを構築し、異常がないことを確実にし、ATMELのAVR Studioでデバッグを開始してください。
3. コードを走らせて予期したように事象が起動されると、捕獲が起こることを確実にしてください。
4. 実行を中断し、下で示されるように中断点を配置してください。

```

capture_values[0] = TCC0.CCA;
capture_values[1] = TCD0.CCA;
capture_values[2] = TCE0.CCA;
LEDPORT.OUT = ~capture_values[0];
}

```

- 変数上を右クリックして“Add watch : “capture_values””を選択することによってcapture_values変数での監視(Watch)を追加してください。
- コードを走らせてcapture_values変数が更新されるのを観測してください。



全ての捕獲値が殆ど同じことに気が付き得ます。それらが正確に同じでないのは何故？。

時間が有るなら、while繰り返しの前に全てのタイマ/カウンタの計数(CNT)値を0に設定するコードを追加してください。今回は捕獲値が同じですか？ それは何故？。

7. 課題4: 手動事象生成

事象はソフトウェアから手動で生成することができます。これは事象発動(STROBE)レジスタ、またはチップ上デバッグ中に直接レジスタをアクセスすることによって行われます。STROBEレジスタ書き込みはその動作を起動します。

一度に多数のビット位置に書くことによって同時に多数のチャネルで事象を生成することが可能です。これは事象活動の同期化、チップ上デバッグ、またはプログラム実行の状態の経緯を保つために事象を使用するのに有用で有り得ます。

手動で生成した事象は1クロック周期間持続し、このクロック周期間は他の事象元からの事象を上書きします。

この課題の目的はあなたが以下の方法を知ることです。

- ソフトウェアから事象を生成すること
- ソフトウェアから事象を生成する時が有用で有り得ることの理解
- 多数のタイマ/カウンタを同期すること



すること

課題:

- Atmel XMEGA-EventSystemフォルダに位置付けてください。xmega_eventsystem.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Task4を活性に設定してください。
- task4.cファイルを開いてコードを習熟してください。このコードは課題3と同様ですが、TCC0,TCD0,TCE0が今や(CPUと同じ速度の)分周なしのクロック前置分周器で走行していることに注意してください。加えてタイマ/カウンタは今や事象を受け取った時にRESTARTに形態設定されます。
- プロジェクトを構築し、異常がないことを確実にし、プロジェクトのデバッグを開始してください。
- 主繰り返しの中断点を配置してください。
- 比較値の経緯を保つことができるようにcapture_valuesを監視(Watch)に追加してください。

```

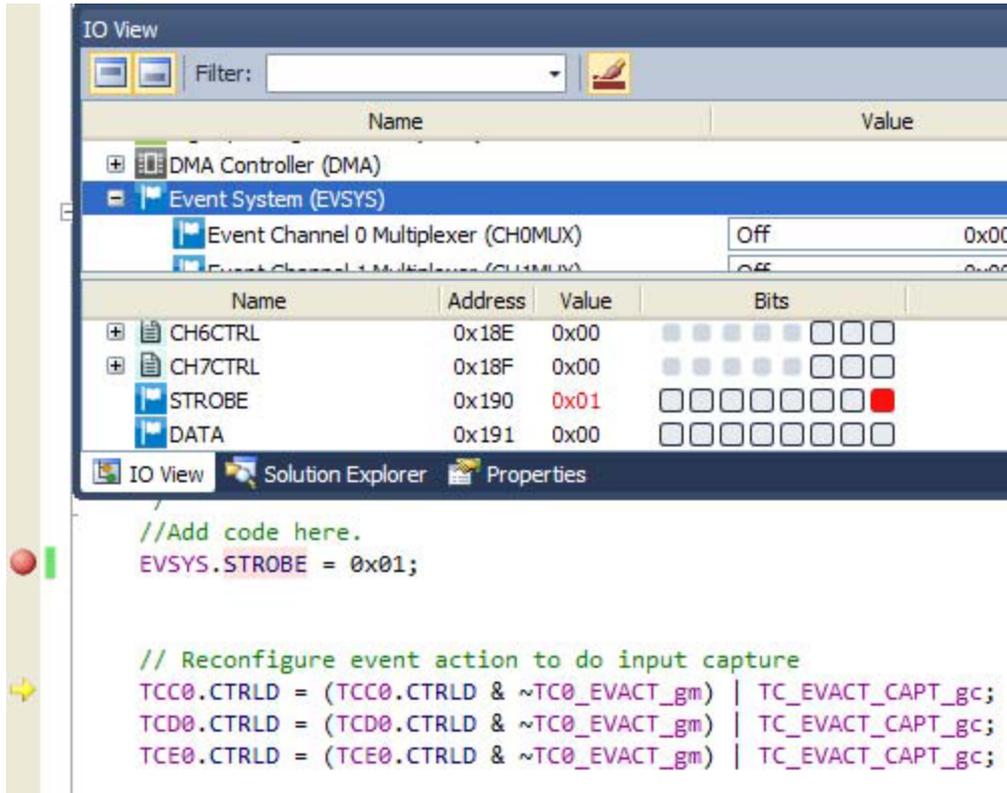
while (1)
{
    // Wait for capture interrupt to update capture values
    do {} while ((TCC0.INTFLAGS & TC0_CCAIF_bm) == 0);

    TCC0.INTFLAGS = TC0_CCAIF_bm;
}

```

- 例え捕獲が正確なクロック周期で起きても、capture_valuesが今違う値であることに注意してください。これは計時器が異なるクロック周期で開始されるためで課題3でのよう(と同じ)です。
- コードは殆ど完成していますが、事象チャネル0でソフトウェアから事象を生成するためにコードの追加が必要です。
- ソフトウェア事象が生成された後で1行実行(Single step)を行えるよう、コードに中断点を配置してください。
- コードを走らせて中断点で停止するのを確実にしてください。
- I/Oウィンドウで事象システムを開いてSTROBEレジスタが書かれつつあるのと、次の周期再び解除(0)されるのを見るために1行実行(Single step)を行ってください。

図7-1. AVR Studio 5



11. 事象起動された捕獲後にcapture_values値が新しい値で更新されるのを見るために“カーソルまで実行(Run to cursor)”を使用してください。今や全ての計時器が完璧に同期していることに注意してください。
12. STROBEレジスタはチップ上デバッグ中、例えばビットを設定(1)するのにI/Oウィンドウを使用することによって書くことができます。書かれたビットは次の周期で解除(0)されます。これを試験することができますが、緩衝部が満杯の場合に計時器に対する捕獲値が保たれないことを覚えて置いてください。代わりに異常を得るでしょう。

8. 要約

この実践で事象システムがどう動くか、その形態設定法を学び、そして事象システムに関する潜在的な利用法が示されました。

9. 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/avrstudio>
- ATMEL用IAR Embedded Workbench®コンパイラ
 - <http://www.iar.com/>

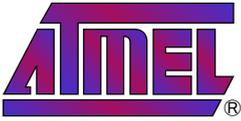
10. ATMEL技術支援センター

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 全てのATMEL AVR製品
- Eメール : 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

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®、ATMELロゴとそれらの組み合わせ、それとXMEGA®、AVR Studio®、AVR®、AVR®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のウェブサイトに表示する販売の条件と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AVR1516応用記述(doc8402.pdf Rev.8402A-07/11)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。