

AVR1501 : XMEGA-A1 Xplain練習 – XMEGA タイマ/カウンタ

前提条件

- 必要な知識
 - ・ AVR1512:XMEGA基礎練習の完了
- 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
 - ・ JTAGICEmk3(またはJTAGICEmk II かAVR ONE!)
- 予想完了時間
 - ・ 2時間

1. 序説

ATMEL® AVR® XMEGA®は最高で非常に柔軟な16ビット タイマ/カウンタ(TC)の組を持ちます。これらの基本的な能力は正確なプログラム実行タイミング、周波数と波形の生成、事象管理、デジタル信号の時間測定を含みます。

この実践ではXMEGA計時器、PWM生成、高分解能拡張、新波形拡張についてもっと学びます。

2. 概要

ATMEL AVR XMEGAは最高で非常に柔軟な16ビット タイマ/カウンタ(TC)の組を持ちます。これらの基本的な能力は正確なプログラム実行タイミング、周波数と波形の生成、事象管理、デジタル信号の時間測定を含みます。

タイマ/カウンタは計数器(COUNT)と比較/捕獲(CC)チャネルの組から成ります。それはタイミングに用いることができる方向(DIR)と定期(PER)の設定を持ちます。

高分解能拡張(Hi-Res)と新波形拡張(AWeX)はもっと進んだ特別な周波数と波形の生成機能の容易な実装のためにタイマ/カウンタと共に使用することができます。

ここはこの練習に於ける課題の短い概要です。

課題1. タイマ/カウンタ開始

最初の課題では前置分周と定期の設定を含み、タイマ/カウンタを開始するための初期構成設定を通して指導されます。

課題2. 比較一致

この課題では比較検査のために比較/捕獲(CCx)レジスタを使用する方法を学びます。

課題3. 波形生成

CCチャネルと比較一致は入出力ピン上での波形生成出力に使用することができ、この課題でこれを形態設定する方法を学びます。

課題4. AWeXと模様生成

タイマ/カウンタ拡張はもっと特殊な機能を許可するのに使用することができます。この課題で共通波形と模様生成動作形態を考察してください。

ご幸運を!



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. 8399A-07/11, 8399AJ1-03/14

3. 課題1: タイマ/カウンタ開始

タイマ/カウンタは動くのにクロック元が必要です。利用可能なクロック元は(前置分周された)周辺クロックと事象システムです。この課題では周辺クロックを使用しますが、事象システムを用いるようにTCを構成設定するのは同じように簡単です。

この課題の目的は以下です。

- ・制御レジスタA(CTRLA)の前置分周(CLKSELビット)を用いてタイマ/カウンタを開始する方法を知る。
- ・計数器がどのくらいの間計数すべきかを設定するために、定期(PER)レジスタを使用する方法を知る。



すること

課題:

1. ATMEL XMEGAのタイマ/カウンタフォルダに位置付けてください。xmega_timer.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. コードを通して眺めてどう構成設定されるかの理解を確実にしてください。
3. 異常がないことを確実にしてプロジェクトを構築してください。
4. デバッグ作業を開始してください。
5. I/OウィンドウでPWM C0を持つタイマ/カウンタを探し出してそれを展開し、故にデバッグ開始時に設定変更を見ることが出来ます。
6. while(1)宣言行に達するまでコードを通して1行実行(Single step)を行ってください。
7. 今やTCC0が前置分周なしの標準動作形態で動いており、IOウィンドウを使用してTCC0構成設定の詳細を見ることが出来ます。
8. 1行実行を続けて計数器(CNT)値が変化し、計数器(CNT)レジスタが\$30に達して丸められる時に割り込み要求フラグ(INT_FLAGS)レジスタの上昇/下降溢れ割り込み要求フラグ(OVFIF)が設定(1)されるのを見ましょう。
9. コードを走行(F5)してください。LED0とLED1の両方のONに注意してください。これはLEDの交互切り替えを見るのに対してコード走行が速すぎるからです。
10. 実行を中断(Ctrl+F5)してください。ATMELのAVR StudioのI/Oウィンドウを使用してTCが周辺クロックの64分周から動くようにクロック選択を変更してください。

図3-1. AVR Studio 5 I/Oウィンドウ

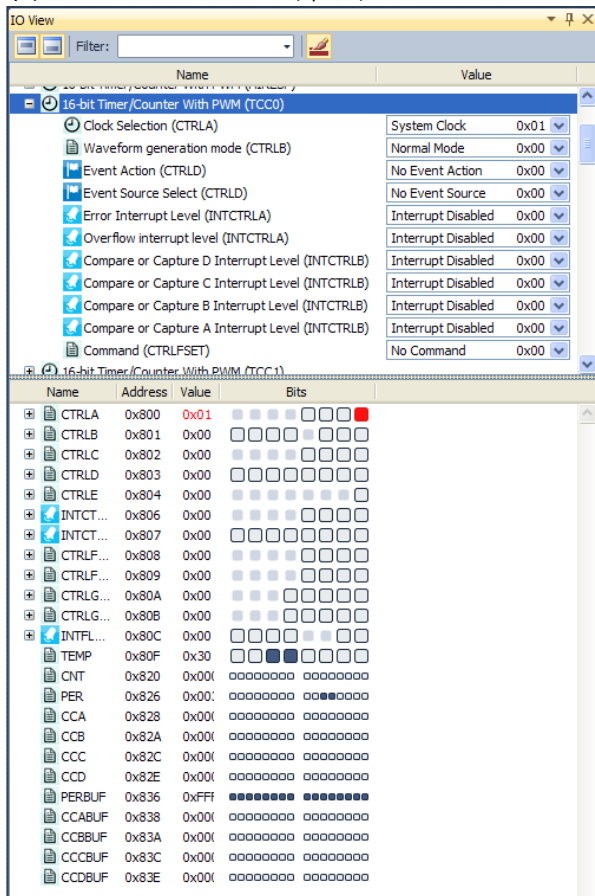
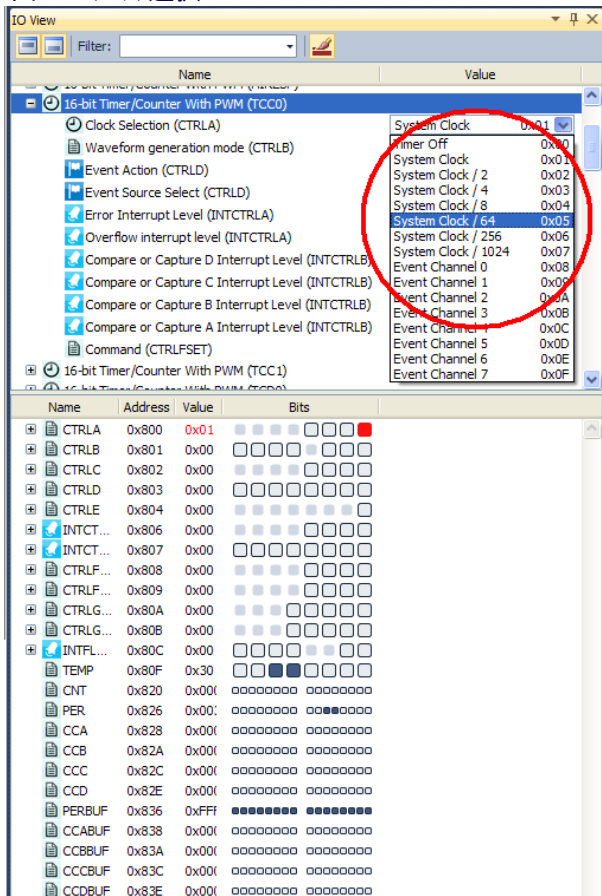


図3-2. クロック選択



11. コードを走行する時にLEDの交互切り替えを見ることができるよう定期(PER)レジスタをより高い値に変えてください。
12. より高いPER設定は計数器に対してより長い周期を与え、従ってLEDはよりゆっくりした繰り返しで切り替えます。

図3-2. PERレジスタ

Name	Address	Value	Bits
CTRLA	0x800	0x01	00000001
CTRLB	0x801	0x00	00000000
CTRLC	0x802	0x00	00000000
CTRLD	0x803	0x00	00000000
CTRLE	0x804	0x00	00000000
INTCT...	0x806	0x00	00000000
INTCT...	0x807	0x00	00000000
CTRLF...	0x808	0x00	00000000
CTRLF...	0x809	0x00	00000000
CTRLG...	0x80A	0x00	00000000
CTRLG...	0x80B	0x00	00000000
INTFL...	0x80C	0x00	00000000
TEMP	0x80F	0x30	00000011
ENT	0x820	0x0000	00000000 00000000
PER	0x826	0x0030	00000011 00000000
CCA	0x828	0x0000	00000000 00000000
CCB	0x82A	0x0000	00000000 00000000
CCC	0x82C	0x0000	00000000 00000000
CCD	0x82E	0x0000	00000000 00000000
PERBUF	0x836	0xFFFF	11111111 11111111
CCABUF	0x838	0x0000	00000000 00000000
CCBBUF	0x83A	0x0000	00000000 00000000
CCCBUF	0x83C	0x0000	00000000 00000000
CCDBUF	0x83E	0x0000	00000000 00000000

4. 課題2: 標準動作での比較一致

課題1は基本的な計時器機能の構成設定法を示しました。課題2は比較一致機能の使用法を示します。

計数器と定期的の設定に加えて、ATMEL AVR XMEGAのTC0型のタイマ/カウンタは比較一致、波形生成(WG)、または捕獲入力に使用することができる4つの比較/捕獲(CC)チャネルを持ちます。XMEGAのTC1型のタイマ/カウンタは利用可能な2つの捕獲チャネルを持ちます。ATxmegaA1は4つの0型計時器(TCC0, TCD0, TCE0, TCF0)と4つの1型計時器(TCC1, TCD1, TCE1, TCF1)を持ちます。

課題2では課題1のようにTCを標準動作で使用しますが、計数器が特定の値に達した時を検査するのに比較一致を使用します。計数器が特定の値に達した時を検知するのに1つのCCチャネルを使用します。これは割り込みと事象の生成に使用できますが、今のところ、比較一致状態を検出するのにソフトウェアポーリングを使用します。

この課題の目的は以下です。

- 比較検査のためのCCxレジスタの使用法を知る。
- 2重緩衝の働き方を知る。



すること

課題:

1. XMEGAのタイマ/カウンタフォルダに位置付けてください。xmega_timer.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を活性化に設定してください。
2. この課題に於いてクロック元を形態設定するのにTC_driver.cファイルからの関数をどう使うかに注意してください。TCドライバファイルに於いてXMEGAのTCを使用するために利用可能な基本的な関数を見ることができます。
3. task2.cでは比較一致に用いるためのCCチャネル用の比較値の設定が必要です。計数器と比較する値、例えば0x0300をCCAレジスタに設定するコードを追加してください。
4. プロジェクトを構築し、異常がないことを確実にしてコードを走行してください。LEDの交互切り替え、そして各溢れに対して比較値を増加するためにON/OFF時間が変わるのを見るべきです。
5. 今回、LEDは何故違うONとOFFの時間を持つのでしょうか。(ONとOFFの時間が同じだった課題1を思い出してください。)
6. (Ctrl+Shift+F5押下によって)デバッグ動作を抜け出してください。CCABUFレジスタを更新している場所のコードを、変わりにCCAレジスタを直接更新(CCA+=0x1000)するように変更してください。
7. コードを再コンパイルし、新規のデバッグ作業を開始してコードを走行してください。



LEDは前のように点滅しません。これは2重緩衝で行わなければならない、それは何故?。

5. 課題3: 波形生成動作

課題1と課題2に於いて計時器を動かして溢れまたは比較一致でソフトウェアによってLEDを交互切り替えするのに標準動作形態を使用しました。

標準動作形態に加えて、TCはポートピン上に直接周波数や波形を出力するのに用いることができる多数の波形生成(GW)動作形態を持ちます。

この課題の目的は以下です。

- 波形生成動作形態を許可して使用方法を知る。
- 対応するポートピンの出力レジスタを無効にするために個別CCチャネルを許可して波形を出力する方法を知る。
- TCからの出力信号を反転するためにポートピンを使用する方法を知る。

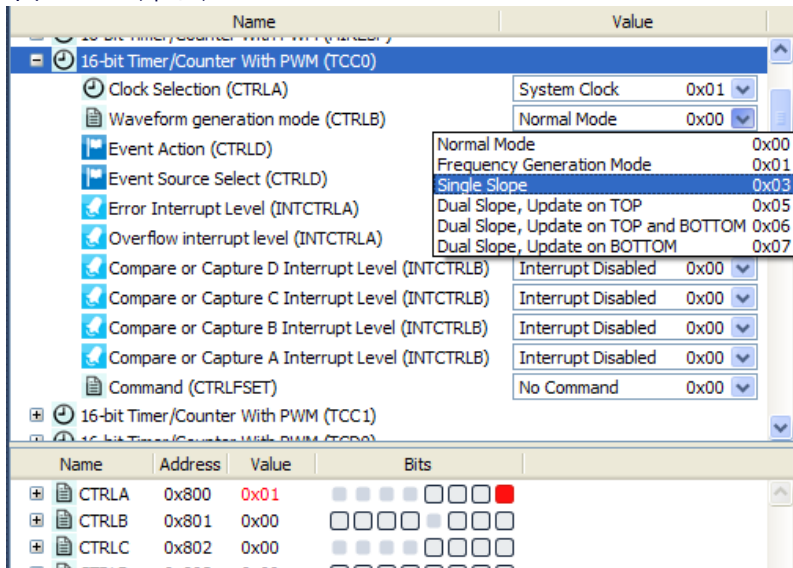


課題:

すること

1. ATMEL XMEGAのタイマ/カウンタフォルダに位置付けてください。 `xmega_timer.avrslin` 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. 基本的な計時器関数は課題1と課題2のように既に構成設定されていますが、正しいWG動作形態を許可して無効化許可信号を設定する必要があります。
3. `main()`関数の先頭で、コードが欠けている場所を探し出して上で説明されるように欠けたコードを追加してください。WG動作形態用の群形態設定を探すためにATMELのXMEGA手引書または `ATxmega128A1.h` ヘッダファイルを使用してください。
4. プロジェクトを構築し、異常がないことを確実にし、AVR Studioでデバッグ動作に入ってください。
5. デバッグを開始し、while繰り返しに達するまで初期化を外側実行(Step over)してください。
6. TCE0用のI/Oウィンドウを開くなら、直ぐに計時器の現在の動作形態と利用可能な動作形態を見ることができます。形態設定を変更してみて、それが出力にどう影響を及ぼすかを見てみてください。

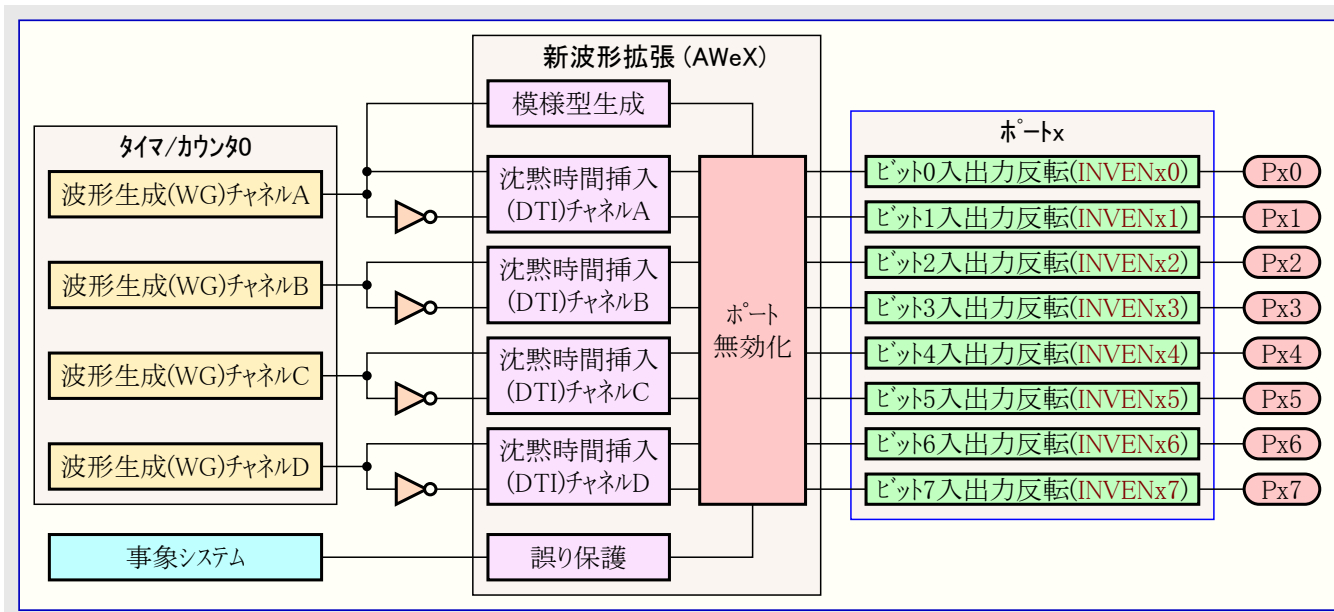
図5-1. I/Oウィンドウ



6. 課題4: AWeXと模様生成

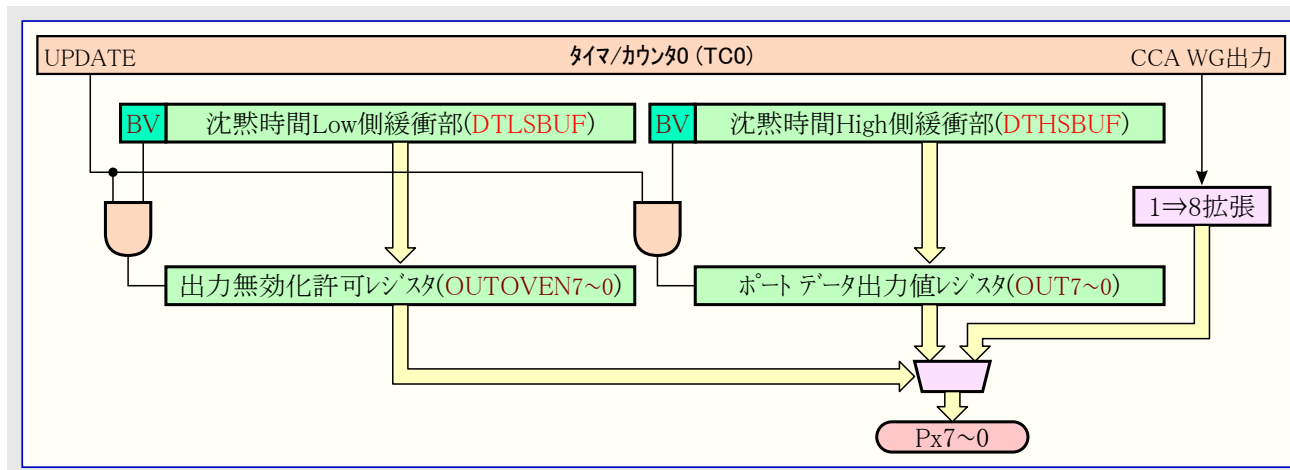
新波形拡張(AWeX)はWG(波形生成)使用時、TCに追加機能を提供します。AWeXは進んだ電動機制御(AC、ブラシレスDC、スイッチリ
ラクタンス、ステッピング モータ)と電力制御応用の容易且つ強力な実装を許します。

タイマ/カウンタ0からの波形生成器出力の各々はAWeX機能が許可された時に補完出力対に分配されます。



この課題で模様生成動作形態(PGM:Pattern Generation Mode)を使用し、これはTCE0が接続されるポート(PORTE)での同期したビット模様を生成するのに使用されます。加えて、CCAチャンネルからの波形生成出力が分配され、全てのポートピンを無効にします。PGMはDCモータ制御と同様の応用に関して理想的ですが、この練習に対してモータが利用可能でないため、より単純な近似を用い、模様を制御するのにスイッチを使用します。モータ制御応用では多分、整流子手順が模様を制御します。

(波形を出力すべきピンを制御する)ビット模様はPGMが許可されている時に沈黙時間Low側緩衝(DTLSBUF)レジスタに格納されます。これはDTLSBUFレジスタのビット0がピン0での出力、以下同様で、制御することを意味します。DTLSBUFのビット0の設定(1)はピン0での波形出力を許可します。High側レジスタは対応するLow側ビットがポート値を無効にするように設定されない時に使用されるべき既定のPORT値を保持します。更新(UPDATE)条件に於いてDTLSBUFレジスタに有効データ(DTLSBUFV)があれば、正しい模様でビット模様を更新されます。



この課題の目的は以下です。

- AWeXの機能を許可してそれを使用する方法を知る。
- 共通波形チャンネル動作形態がどう動くかを理解する。
- 模様生成が何かを知る。
- AWeXが(丁度TCのように)独立した出力無効化許可ビットを持つことを知る。



課題:

1. ATMEL XMEGAのタイマ/カウンタフォルダに位置付けてください。xmega_timer_avrslm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。

全てのピンにCCA波形出力を許可するために共通波形動作形態(CGM:Common Waveform Mode)の許可が必要です。これは制御(CTRL)レジスタの共通波形チャネル動作(CWCM)を設定(1)することによって行われます。CTRLレジスタの模様型生成動作(PGM)を設定(1)することによって模様生成動作形態の許可も必要です。

2. main()関数の先頭を探し出し、コードが欠けている場所を探してください。AWeXを正しく許可するように2つの失われている行を挿入してください。
3. コードに於いて、AWeX許可時にCCチャネルに対してポート無効化(実質上書き)を許可するために、チャネルx沈黙時間挿入許可(AWEX_DTICCxEN)ビットがどう設定(1)されなければならないかに注意してください。これはタイマ/カウンタに関する無効化許可ビットと同じです。
4. プロジェクトを構築し、異常がないことを確実にし、AVR Studioでデバッグファイルを開いてください。
5. コードを走らせてスイッチが押された時に模様はどう変化するかを見てください。複数変化を避けるため、短時間だけのスイッチ押下を守ってください。
6. デバッグ作業を中断し、以下の行に中断点を追加してください。

```
AWEXC.DTLSBUF = new_pattern;
```

7. 数回1行実行(Single step)を行い、新波形拡張Eを考察するためにI/Oウィンドウを展開し、出力無効化許可(OUTOVEN)と沈黙時間Low側緩衝部(DTLSBUF)のレジスタが今やどのように異なる値を持つかを検証してください。

Name	Address	Value	Bits
CTRL	0xA80	0x0F	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
FDEMASK	0xA82	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
FDCTRL	0xA83	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
STATUS	0xA84	0x01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>
DTBOTH	0xA86	0xF0	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
DTBOTHBUF	0xA87	0xC3	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DTLS	0xA88	0xF0	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
DTHS	0xA89	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DTLSBUF	0xA8A	0xC3	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
DTHSBUF	0xA8B	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
OUTOVEN	0xA8C	0x0F	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

8. 中断点を取り去り、中断前のコードを再び走行してください。OUTOVENとDTLSBUFがどうして同じ値を持つかに注意してください。



OUTOVENが更新される時、それとこれを起こすタイマ/カウンタ条件は何でしょうか?。

7. 要約

この練習でATMEL XMEGAの計時器、PWM生成と新波形拡張についてと、それらがどう形態設定されて応用でどう使用されるかを学びます。

8. 資料

- XMEA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®コンパイラ
 - <http://www.iar.com/>

9. 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のウェブサイトには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AVR1513応用記述(doc8399.pdf Rev.8399A-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。