

AVR1321 : ATMEL AVR XMEGAの32ビット実時間計数器 と電池代替支援システムの使用法

要点

- 32ビット実時間計数器(RTC)
 - 32ビット計数器
 - 選択可能なクロック元
 - ・ 1.024kHz
 - ・ 1Hz
 - 長い溢れまでの時間
 - ・ クロック元として1Hz使用時に136年以上
 - 超低消費電力
 - パワーセーブ動作からMCUを起動復帰可能
 - 溢れと比較一致での任意選択の割り込み/事象
- 電池代替支援システム
 - 主電力消失時にVBAT電力ピンからの供給電圧を持つ32ビットRTCと2つの支援レジスタを提供
 - 主電力の低電圧検出(BOD)で主電力から電池電力への自動切換え
 - 主電力が再び利用可能時に電池電力から主電力への自動切換え
 - クリスタル停止検出監視器

1. 序説

ATMEL®のAVR® XMEGA® 32ビット実時間計数器部署は1.024kHzまたは1Hzのクロック元周期を計数するように設計されています。これは溢れ割り込みまたは(それと)比較一致割り込みによって指定される間隔で低電力形態からMCUを起こすのに使用することができます。32ビット分解能のお蔭で、最大溢れ時間は1Hzクロック元使用時に136年以上です。ソフトウェア算法で、RTCは年、月、曜日、日を導き出して暦機能を提供するのに容易に用いることができます。

32ビットRTCは主電力消失時に電池代替支援から給電することができる独立した電力領域である電池代替支援システム内に含まれます。主電力から支援電力への遷移は支援システムによって自動的に行われ、故にRTCの安全な動作を保証します。RTCを持つ代替支援システムは例えシステムの残りが電力なしの間でも、時間の経緯を保つ必要があるシステムに対して理想的な解決策を形成します。例えば電池給電される携帯装置を想像してください。或る時点で使用者は電池交換を必要とするかもしれませんが、代替支援システムが全く利用できない時にこれは時間の損失になるでしょう。完全に統合化された解決策は外部の実時間クロックと/または電力スイッチの費用を取り去ります。

この応用記述はいくつかのXMEGAデバイスで利用可能な32ビット実時間計数器と電池代替支援システムの詳細な使い方を網羅します。



8ビット **AVR**®
マイクロコントローラ

応用記述

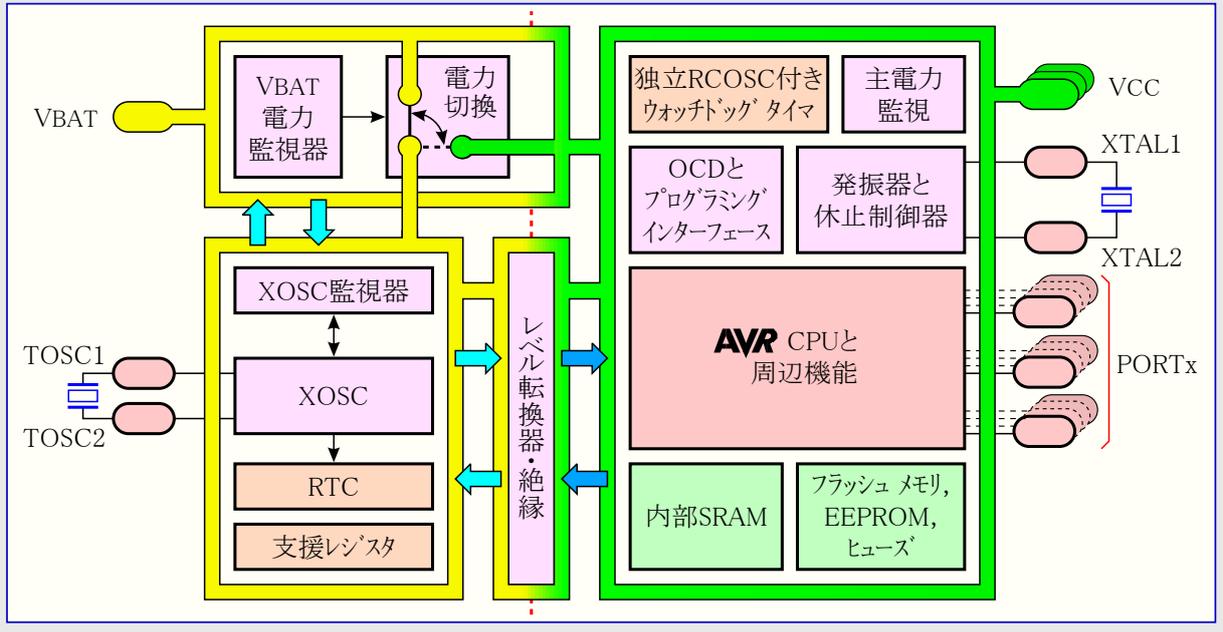
本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 8307B-02/11, 8307BJ1-03/14

2. 電池代替支援システム

電池代替支援システムは主電力が失われている間に任意選択の外部からの供給電力を持つ電力領域内に含まれる全ての周辺機能へ供給する任意選択を提供します。

図2-1. 電池代替支援システム概要



以下の部署が代替支援電力領域内に配置されます。

- 電池代替支援システム監視器
- 電力スイッチ
- 32.768kHzクリスタル用低電力発振器
- 発振器停止検出監視器
- 32ビット実時間計数器(RTC)
- 2つの電池代替支援レジスタ
- 採取式電池代替低電力検出器(BBBOD)
- 電池代替電源ON検出器(BBPOD)

電池代替支援システムはSRAMやI/Oレジスタ、デバイス内の不揮発性メモリのように、この電力領域の外側の他の部分へ電力を提供しません!

デバイスは主電力消失を検知するのに主電源上の低電圧検出器(BOD)を使用し、そして自動的にVBATに接続された代替支援電源へ切り替えます。加えて、代替支援電源を監視するために代替支援システムのBODがONにされます。主電力が回復された後、システムは自動的に主電力へ切り替え戻します。

電池代替支援は主電力が存在しない時に放電するだけです。これは電池代替支援に関して最大電池寿命時間を保証します。

2.1. 電力監視器

電力監視器は代替支援システムの始動と代替支援システム状態フラグの状態更新を扱います。監視器は電源ONと低電圧の検出器を持ち、これらから以下のようなBBPWR, BBPODF, BBBODFの状態フラグが生成されます。

電力検出(BBPWR)機能はデバイスがリセット状態を去る時毎にVBAT電圧を調べます。VBATピン上に電圧が全く存在しない場合、電池代替電力(BBPWR)フラグが設定(1)されます。これは主電力が失われていた間に代替電池が取り去られた、または放電されてしまったことを示します。このフラグはシステムの始動手順の間でだけ更新されるので、代替電池電力監視器としてこのフラグを使用することはできません。

電源ON検出(BBPOD)機能はVBATピンに電力が印加された時、換言すると、代替電池が挿入された時を検知します。これが起こる時に電池代替電源ON検出フラグ(BBPODF)が設定(1)され、デバイスが形態設定される前に代替電池流出を防ぐために電力スイッチが切断されます。このフラグはデバイスのリセット中にだけ更新されます。より多くの情報については2.2節を調べてください。このフラグは一度更新されるだけなので、代替電力の切断は(主電力が再印加されるように)デバイスのリセット後にだけ検出されます。

低電圧検出(BBBOD)機能は電池代替支援システムがVBATピンから給電されている時にVBAT電圧水準を監視します。この検出器は主電力が利用可能な時に活性ではなく、故に主電力から走行している間にこの検出器で電池代替低電圧を検知することはできません。代替電源からの走行時にVBAT電圧が閾値電圧以下に落ちた場合、電池代替低電圧検出フラグ(BBBODF)が設定(1)されます。BBBODは概ね1Hzの速度でVBAT電圧水準を採取し、遅い電圧変化を検出するように設計されています。

2.2. 代替支援システム始動手順

主電力が回復された時毎に、代替支援システムはその始動手順を走行します。このリセット手順が代替支援システムのソフトウェアリセットによって起動されず、デバイスのリセットによってのみ起動されることに注意してください。代替支援システムに於いて内部的に以下の動きが実行されます。

- 1) 代替電池が十分な電力を持っているかを調べるために低電圧検出器はVBAT入力で採取します。
- 2) 状態(STATUS)レジスタで電池代替電力(BBPWR)フラグが格納されます。
- 3) VBATで十分な電力が測定されたなら、状態(STATUS)レジスタで電池代替低電圧検出フラグ(BBBODF)と電池代替電源ON検出フラグ(BBPODF)が格納され、有効な電力(BBPWR)が検出されない場合、これらのフラグは無効です。
- 4) VBATシステムは主電力に切り替え、VBAT入力を切断して通常形態に移ります。

BBPWR, BBPODF, BBBODFは代替支援システムのリセット手順の間にだけ講師Hんされるので、主電力から走行している間に代替電池の状態を監視するためにこれらのフラグを使用することはできません。

2.3. クリスタル用発振器と停止検出監視器

クリスタル用発振器(XOSC)は外部32.768kHzクリスタルの接続を支援します。これは1.024kHzまたは1Hzのどちらかの出力を持つ予め分周されたクロックを提供します。

クリスタル用発振器は超低電力消費用に設計され、従ってこれを達成するために制限された駆動能力を持ちます。減少された駆動能力は発振器を雑音や他の影響をより受け易くもします。このため、発振器は消費電力を犠牲にして駆動能力を増す、“高ESR”形態用の形態設定任意選択を提供します。

クリスタル用発振器停止監視器はクリスタルが恒久的または一時的に停止されたかを検知し、そしてクリスタル用発振器停止(XOSCFAIL)フラグを設定(1)します。停止検出機構は計数器へ供給する統合された低電力RC発振器を持ちます。この計数器はクリスタル用発振器が検出された場合にそのクロック計時毎にリセットされます。計数器が溢れる前にリセットされなければ、停止検出フラグが設定(1)されます。内部32kHz RC発振器の64周期間にクリスタル用発振器の周期が全く検知されない場合に停止が検出されます。監視発振器は約30%の精度を持ちます。

2.4. 代替支援レジスタ

代替支援システムは主電力消失中に保存されるべき情報を格納するのにも使用することができる2つの8ビットレジスタを提供します。

3. 32ビットRTC

XMEGAの32ビットRTCは32ビット計数器と単一比較チャネルを含みます。任意選択の溢れ割り込みは計数器値が指定した定期値に達した時に生成することができます。加えて、比較一致割り込みは計数器値が指定した比較値と等しい時に生成することができます。

3.1. レジスタ

RTCの計数、定期、比較の値は全て32ビット値です。データバスが8ビット幅なので、32ビット値を表現するのに4つのレジスタが使用されます。“3”の接尾子を持つレジスタが最上位バイトです。このRTC用の32ビットレジスタは、CNT3~0, PER3~0, COMP3~0です。

3.1.1. バイトアクセス

殆どのCコンパイラは32ビットレジスタの抽象名が使用される時に自動的に32ビットアクセスを扱います。個別の8ビットレジスタアクセスも可能です。

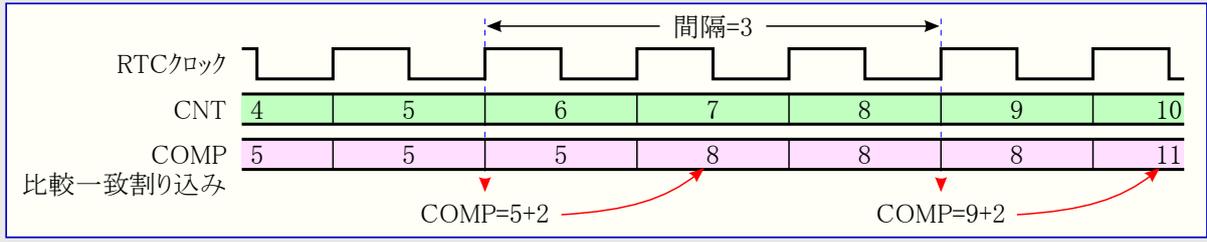
バイトアクセス法で32ビットRTC部署の32ビットレジスタを書くには最初に最下位バイトが書かれなければなりません。最上位バイト書き込み後、新しい値は多数のRTCクロック周期後に利用可能です。クロック領域の同期化については3.4節を参照してください。

3.2. 割り込み

RTCは溢れ割り込みと1つの比較割り込みの2つの割り込みを生成することができます。溢れ割り込みは定期(PER)レジスタによって制御されます。この割り込みはMCUを起こす、または事象を生成するのに用いることができます。比較レジスタは周期の変更または計時器の計数(CNT)レジスタを再設定することなく可変時間経過割り込みを設定する方法を提供します。使用者はLCDでの時間表示を更新するため、または暦の警報機能を実装するため、毎分でパワーセーブ形態からMCUを起こすのに、例えばこの割り込みを利用することができます。

比較一致割り込みは比較一致条件発生後の次の計数で引き起こされます。これは比較一致間隔に付加周期が追加されることを意味します。例えば、比較周期が2 RTCクロック周期($COMP=CNT+2$)に設定されるとの仮定で、比較一致間隔は3 RTCクロック周期です。

図3-1. 比較一致間隔での追加周期



3.3. クロック元

32ビットRTCは発振器の2つの出力の1つだけによってクロック駆動でき、それらの予め分周されたクロックは1Hzと1.024kHzです。電力消費を減少して溢れる前の最長時間を得るには1Hz出力を用いることができます。この場合、計時分解能は1秒で、故に溢れ最大時間は 2^{32} 秒で、これは136年以上です。クロック元として1.024kHz出力を使用することの恩恵は概ね1msのより高い分解能です。

3.4. クロック領域

RTCがCPUと異なるクロックで動くため、それはCPUと別の“クロック領域”を持ちます。CPUとRTCがクロック領域境界を渡って情報を交換する時に、2つのクロック領域間の同期が必要とされます。

同期を必要とする、制御(CTRL)レジスタ、計数(CNT)、定期(PER)、比較(COMP)の4つのレジスタがあります。同期はハードウェアによって制御されます。その意味は8ビットレジスタ(CTRL)の値変更、または32ビットレジスタ(CNT, PER, COMP)の最上位8ビット書き込みによって同期が起動されます。32ビットレジスタに関して下位側8ビットの書き込みは(同期に関して)何の効果も持ちません。従ってレジスタアクセスの正しい順番は、先に下位側の8ビット、その後最上位8ビットを書くことが適切です。

RTCからCPUの領域への同期は同期制御/状態(SYNCCNT)レジスタのCNTレジスタ同期許可(SYNCCNT)ビットによって起動されます。CNT同期だけが必要とされることに注意してください。

同期が完了されたかどうかを監視するため、SYNCCNTレジスタの同期化中(SYNCBUSY)フラグを調べることができます。CTRLまたはCNTの同期だけがこのフラグに影響します。

同期時間はレジスタ間で変わります。右表で記述されます。

表3-1. RTCLレジスタ同期周期数

レジスタ名	書き込み同期周期数	読み込み同期周期数
CTRL	0.5 RTCクロック周期	
CNT	12 周辺クロック周期	8 周辺クロック周期
PER	2 RTCクロック周期	
COMP	2 RTCクロック周期	

正しい操作を保証するため、周辺クロックは制御と計数のレジスタがアクセスされる時にRTCクロック(1.024kHzまたは1Hz)よりも8倍速くなければならない、計数レジスタが書かれる時に12倍の速さを求めます。

3.5. 休止形態に関する特別な関係

同期のために3 RTCクロック周期、それは同期のための2 RTC周期と割り込み遅延用の1追加RTC周期(より多くの詳細については2.2.節を参照)毎よりもっと頻繁にRTC割り込みによって休止形態から起こすのは不可能です。比較間隔が2 RTCクロック周期($COMP=CNT+1$)またはそれ以下に設定された場合、新しい比較値が同期された時に実際のCNT値は比較値よりも大きく、故に比較一致割り込みは予期したように起きません。

3.6. 事象システムへのRTC接続

32ビットRTCに関して溢れと比較一致で事象を生成することが可能です。事象システムの詳細についてはデータシートとAVR101応用記述を参照してください。

注: 事象システムはアイドル動作以外の他の休止形態で動作しません。

3.7. デバッグ形態でのRTCの動作

例えば中断点(ブレークポイント)により、デバッグ形態でコード実行が停止される時にRTCクロックは妨げられます。これは単一実行(シングルステップ)が求められるデバッグ作業で割り込みが継続的に生成されないことを保証します。けれども、RTCクロック元がCPUクロックに同期するため、コードの単一実行時にRTCのタイミングは影響を及ぼされます。

4. 電池代替支援システムとRTCの形態設定

4.1. 予防処置

4.1.1. 低電圧検出(BOD)形態設定

代替支援システムは主電力が失われた時を知る必要があるため、主BODの活性化は必須です。ATMEL® AVR® XMEGA®はデバイスの動作形態に対して異なる形態設定を提供するBOD形態設定に関して2つのヒューズを持ちます。代替支援システムを動作するために以下のヒューズ設定が必要です。

- ・活動形態とアイドル形態用のBODヒューズは継続的なBODに設定されなければなりません。
- ・パワーダウン形態用のBODヒューズは継続形態または採取形態のどちらかに設定されなければなりません。

4.1.2. 電源ON傾斜検査

主電源の活性化は単調上昇傾斜で終わるべきです。さもなければ傾斜が途中でいくらかの低下を持つ時にデバイスが2回リセットされることが起き得ます。2重リセットの結果は電池代替電源ON検出フラグ(BBPODF)と電池代替低電圧検出フラグ(BBBODF)を解除することになり、即ち2重リセットは代替電池が決して挿入されないように見え、代替支援システムが低電圧だったかどうか分かりません。

これを避けるため、主電源の上昇傾斜が単調であることを調べることを推奨されます。

加えて、応用コード内の非常に早期にBBPODFとBBBODFのフラグを調べるのが最善です。早期検査時、電源低下やウォッチドッグ タイムのような別のリセットのためにそれらが失われそうにもありません。

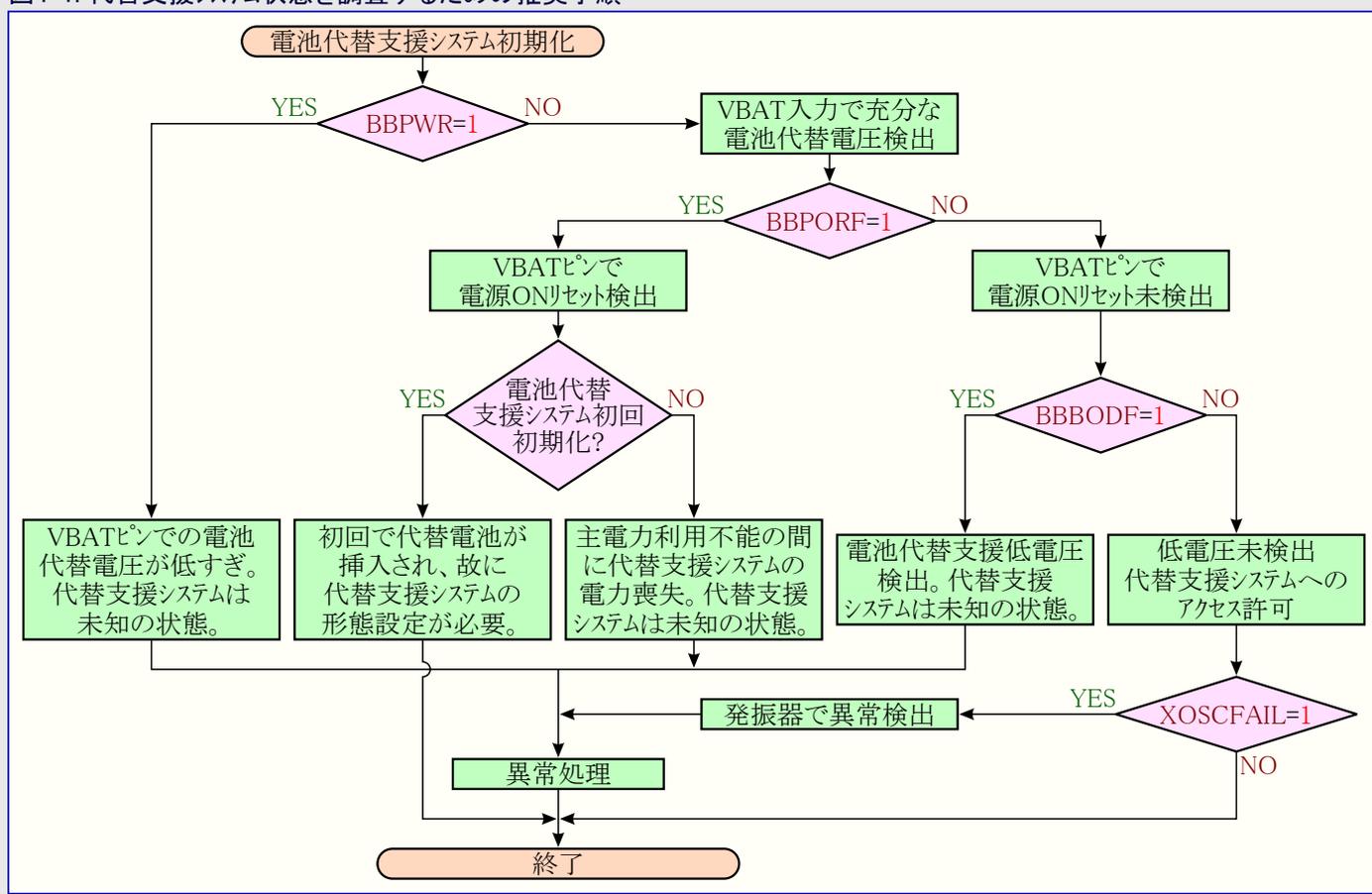
4.1.3. 発振器停止検出機構での追加安全性

正しい時間が重要な応用に関しては、発振器停止検出機構を許可することが推奨されます。この機能は発振器が最低64周期間動作を停止されたかどうかを検知します。従って発振器の問題は始動で外部用発振器停止検出(XOSCFAIL)レジスタの停止検出割り込み要求フラグ(XOSCFDIF)を調べることと、応用でこのフラグを監視することによって検知することができます。

4.2. 始動後の代替支援システム検査

代替支援機能が使用される時に、主電力が不在中に何か問題があったかを評価するために、主電力が印加されている時に代替支援システムの状態を調べる必要があります。この調査は「電源ON傾斜検査」節で検討されるように応用での最初の活動の1つであるべきです。

図4-1. 代替支援システム状態を調査するための推奨手順



代替支援システムの状態を決めるための推奨方法が図4-1.で概説されます。システムが初回に初期化されるのかまたはVBATで電力消失が検出されたのかのどちらかを知ることができるために、この情報を定常的に、例えばEEPROMに格納する必要があります。

4.3. 初期化

代替支援機能が使用されるか否かに拘らず、発振器とRTCの初期化は常に同じです。これらを形態設定するには以下の段階が必要とされます。

1. 電池代替支援システムリセット(RESET)を実行してください。
2. 電池代替支援システムの制御(CTRL)レジスタで電池代替支援部署アクセス許可(ACCEN)ビットを設定(1)してください。
3. 発振器停止検出を許可してください。
4. 任意でRTCへの入力クロックを選択してください。
5. 任意で発振器に対して高ESR形態を許可してください。
6. 200 μ s待ってください。
7. 発振器を許可してください。
8. クリスタル用発振器準備可(XOSC RDY)フラグをポーリングすることによってクリスタルが安定するまで待機してください(フラグは発振器の約8K周期が検出された後で設定(1)されます)。
9. RTC部署の定期(PER)レジスタを書くことによって周期値を設定してください。
10. 任意で比較一致値(COMP)と初期計数値(CNT)を設定してください。
11. 任意で比較一致と溢れの割り込みに対して割り込みレベルを設定してください。
12. RTC部署の制御(CTRL)レジスタでRTC許可(ENABLE)ビットを設定(1)することによって32ビットRTCを許可してください。

4.4. RTC周期変更

1. RTC部署を禁止してください。
2. RTC部署の同期制御/状態(SYNCCTRL)レジスタで同期化中(SYNCBUSY)フラグを調べることで、同期終了まで待ってください。
3. 定期(PER)レジスタを書いてください。
4. RTC部署を再許可してください。

4.5. 計数値読み込み

1. 同期制御/状態(SYNCCTRL)レジスタのCNTレジスタ同期許可(SYNCCNT)ビットを設定(1)してください。
2. 同期終了まで待ってください。
3. 計数(CNT)レジスタを読んでください。

4.6. 比較一致値または計数値の変更

1. 同期終了まで待ってください。
2. 比較(COMP)または計数(CNT)のどちらかのレジスタを書いてください。

5. 応用実装

5.1. ドライバ

この応用記述はC言語で実装された基本ドライバと共にソースコード1式を含みます。

このドライバが高い可読性と周辺機能部署の使用法の一般的な例として書かれていることに注意してください。応用でこのドライバを使用する場合、関数呼び出し数を減らすため、必要とされるコードの関連部分を複写することが望まれるかもしれません。

5.2. 例

ソースコード1式は2つの例から成ります。以下はそのファイルです。

- `rtc32_driver.c` – ドライバ ソース ファイル
- `rtc32_driver.h` – ドライバ ヘッダ ファイル
- `vbat.c` – VBATシステム用ドライバ ソース ファイル
- `vbat.h` – VBATドライバ用ヘッダ ファイル
- `rtc32_example1.c` – ドライバを使用するコード例

5.2.1. 例1

この例は32ビットRTCの使用法と3秒毎にパワーセーブ形態からMCUを起こす方法を示します。

5.2.1.1. 消費電力計算

殆どの応用に於いて消費電力は非常に重要な項目です。求められた電池の寿命時間を提供するのに平均消費電力が充分低いかどうかを知るために、非常に注意深くそれを計算することが必要です。平均電流は右式によって定義されます。

式5-1. 平均電流式

$$\text{平均電流} = \frac{\text{活動時間} \times \text{活動電流} - \text{休止時間} \times \text{休止電流}}{\text{活動時間} - \text{休止時間}}$$

活動時間は活動形態で費やす時間を表します。これは次式によって計算することができます。

式5-2.

$$\text{活動時間} = \frac{N}{f_{clk}}$$

変数 N は活動形態で実行されたクロック周期の量を表します。

この下はIARTMコンパイラによって生成された例1のオブジェクトコードから逆アセンブルされた代表的な繰り返し周回です。

コード例:

```

RETI                ;Interrupt return
RJMP                PC-0x000D ;Relative jump
LDI                 R16, 0x02  ;Load immediate
LDI                 R17, 0x00  ;Load immediate
LDI                 R18, 0x00  ;Load immediate
LDI                 R19, 0x00  ;Load immediate
CALL                0x000001DB ;Call subroutine
LDI                 R30, 0x21  ;Load immediate
LDI                 R31, 0x04  ;Load immediate
LDD                 R20, Z+0   ;Load indirect with displacement
ORI                 R20, 0x10  ;Logical OR with immediate
STD                 Z+0, R20   ;Store indirect with displacement
LDD                 R20, Z+0   ;Load indirect with displacement
SBRC                R20, 4    ;Skip if bit in register cleared
LDI                 R30, 0x24  ;Load immediate
LDD                 R20, Z+0   ;Load indirect with displacement
LDD                 R21, Z+1   ;Load indirect with displacement
LDD                 R22, Z+2   ;Load indirect with displacement
LDD                 R23, Z+3   ;Load indirect with displacement
ADD                 R20, R16   ;Add without carry
ADC                 R21, R17   ;Add with carry
ADC                 R22, R18   ;Add with carry
ADC                 R23, R19   ;Add with carry
LDI                 R30, 0x28  ;Load immediate
LDD                 R16, Z+0   ;Load indirect with displacement
LDD                 R17, Z+1   ;Load indirect with displacement
LDD                 R18, Z+2   ;Load indirect with displacement
LDD                 R19, Z+3   ;Load indirect with displacement
CP                  R16, R20   ;Compare
CPC                 R17, R21   ;Compare with carry
CPC                 R18, R22   ;Compare with carry
CPC                 R19, R23   ;Compare with carry
BRCC                PC+0x09   ;Branch if carry cleared
LDI                 R30, 0x2C  ;Load immediate
STD                 Z+0, R20   ;Store indirect with displacement
STD                 Z+1, R21   ;Store indirect with displacement
STD                 Z+2, R22   ;Store indirect with displacement
STD                 Z+3, R23   ;Store indirect with displacement
RET                 ;Subroutine return
LDI                 R16, 0x02  ;Load immediate
STS                 0x0607, R16 ;Store direct to data space
LDI                 R16, 0x07  ;Load immediate
STS                 0x0048, R16 ;Store direct to data space
SLEEP               ;Sleep

```

上の命令は73クロック周期に展開します。主クロックが32MHzなら、式5-2.に従って以下が得られます。

$$\text{活動時間} = \frac{73}{32\text{MHz}} = 2.28125\mu\text{s}$$

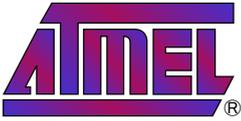
3.0V電源の場合、ATxmega256A3Bのデータシートに従い、活動電流=15.7mA、休止電流=0.65 μ Aです。故に平均電流は以下のように計算することができます。

$$\text{平均電流} = \frac{2.28125\mu\text{s} \times 15790\mu\text{A} + (3 \times 10^6\mu\text{s} - 2.28125\mu\text{s}) \times 0.65\mu\text{A}}{3 \times 10^6\mu\text{s}} \doteq 0.66\mu\text{A}$$

5.3. Doxygen

全てのソースファイルはDoxygenを使用する自動資料生成用に準備されています。

Doxygenは特別なキーワードを使用してソースコードを分析することによって、ソースコードから資料を作成するツールです。Doxygenについてのより多くの詳細に関しては<http://www.doxygen.org>を訪ねてください。予めコンパイルされたDoxygen資料は本応用記述に伴うソースコードと共に供給され、ソースコードフォルダの[readme.html](#)ファイルから利用可能です。



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®、ロゴとそれらの組み合わせ、それとAVR®、AVR Studio®、XMEGA®、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AVR1321応用記述(doc8307.pdf Rev.8307B-02/11)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。