

AVR4013 : picoPower基礎

要点

- 休止形態
- PRRLレジスタ
- 発振器校正
- USART

1. 序説

この応用記述でファームウェアだけを変更することで多数の要素によって我々の応用の電池寿命を延ばす方法を示します。変更のいくつかは非常に単純でいくつかのレジスタの設定を必要とするだけで、一方他の変更はいくつかのコードの書き換えが必要であろうことを知ります。

それが同じ量のエネルギーで充電することが容易なため、この実演での電池としてコンデンサを使用します。電力消費の測定として放電時間を使用します。

実演応用は1つのA/D変換を行い、処理に値する1000周期を実行し、そして"More oomph to your amps, picoPower!"文字列を送ってUARTで計数を繰り返します。これは各秒毎に繰り返されます。この応用記述を読む前に、以下からATmega88PA用のデータシートをダウンロードすべきです。

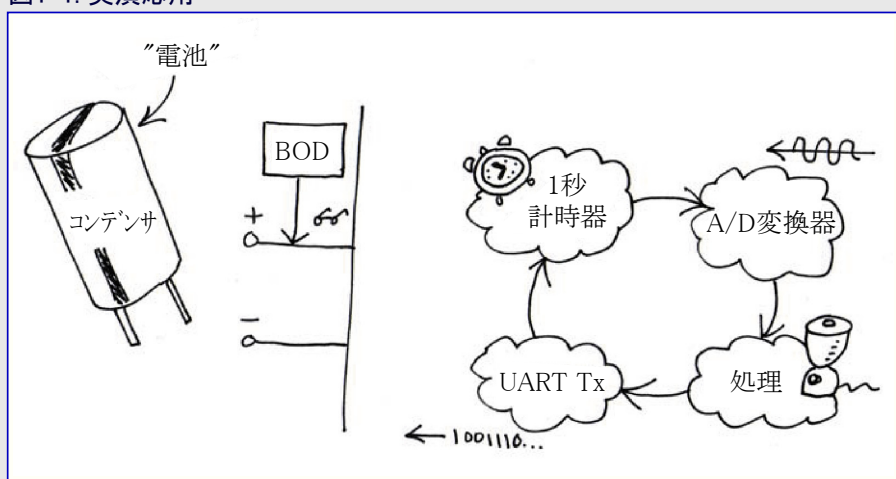
http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf



8ビット AVR[®]
マイクロコントローラ

応用記述

図1-1. 実演応用



2. 回路図

図2-1. picoPower実演回路図

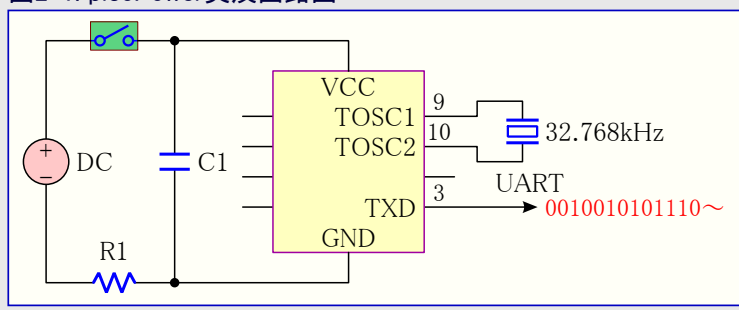


図2-1から実演回路がかなり簡単を知ることができます。スイッチが押された時にコンデンサを充電するのに使用される5.5Vに設定されたDC電圧源があります。スイッチが開放されると、ATmega88PAはコンデンサC1に蓄えられたエネルギーを流出させます。デバイス用の主クロック元は内蔵RC発振器です。タイマ/カウンタ2は外部クリスタルに接続され、休止形態のいくつかで機能することができます。これを実演で使用します。

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. 8349A-11/10, 8349AJ1-12/13



3. コード

このコードは1秒に一度データを送信する感知器装置を偽装するように計画されています。初期化後(これは各最適化法で異なります)、コードは無限繰り返しに移ります。

この繰り返しは以下の段階に分けられます。

- A/D変換
- 1000周期の処理擬似実行
- 番号をASCII文字列に変換
- UART上にデータ送出("More oomph to your amps, picoPower! [数値繰り返し]")
- 繰り返し

この繰り返しはいくつかのアナログ装置からデータ収集を偽装するためにA/D変換を行うことによって始まります。そしてデバイスは1000周期のデータ処理を偽装する関数へ移ります。多くの応用で数値をそのASCII表現に変換することが好まれ、故に繰り返し回数をASCIIに変換します。最後に、UART上に"More oomph to your amps, picoPower! [繰り返し回数]"文字列を送ります。その後この繰り返しを繰り返します。

4. ソースコードの有効性

5つの最適化レベルの全てに対するコードが独立した.zipファイルに含まれ、AVR Studio 4.18RC2とWinAVR-20100110ツールチェーンで検査されています。

5. 実演段階

コードは応用で各レベルが増された節電を提供する、5つの最適化レベルから成ります。

- 最適化なし
- 未使用入出力ピンでのプルアップ許可と使用されない部署の禁止
- 8MHzから2MHzへクロック前置分周
- 次送信待機中にパワーセーブ休止形態使用
- より高いボーレートと許可するように発振器校正

実演を開始する前に内蔵RC発振器を校正しなければなりません。これは全電圧範囲に渡ってボーレート発生器が安定なクロック源を生成することができることを保証するためです。

発振器校正は高いデバイス依存でATmega88PAデータシートを参照すると、内蔵RC発振器の工場校正は±10%の精度を持ち、使用者校正精度は±1%以内です(データシートで「28.4.1. 校正付き内蔵RC発振器精度」項をご覧ください)。

これを行うため、8MHzに最も近い周波数を見つけるためにオシロスコープを用いてCKOUTヒューズ(データシートで「27.2. ヒューズビット」項をご覧ください)を許可することができます。

発振器を校正するのに発振器校正(OSCCAL)レジスタ(データシートで「発振器校正レジスタ(OSCCAL)」をご覧ください)を使用し、発振器を校正する良い方法についてはAVR053応用記述を参照してください。

この例で使用されるATmega88PAで、我々はOSCCALレジスタでの値\$66が最良の周波数を与えることを見つけました。

5.1. 最適化なし

基準点を作成するために最適化なしでコードを動かすことによって始めます。デバイスを8MHzに設定してボーレート発生器を19200ボー設定でコードを実行します。

次送信待機の間、マイクロコントローラは最後の送信開始から1秒が経過したかを調べるためにタイマ/カウンタ2割り込み要求フラグレジスタ(TIFR2)をポーリングします。この方法は何時も活動形態でコードを走行します。

走行時間：6秒

5.2. 未使用入出力ピンのプルアップ許可と使用されない部署の禁止

いくつかの電力消費を取り去るのに、論理レベルを限定させて不要な切り替えを避けるために未使用入出力ピンでのプルアップを許可します。入出力ピンでのプルアップを許可するため、ポート方向レジスタでピンを入力に設定してポート出力レジスタをHighレベルに設定します(データシートで「13.2.1. ピン設定」項をご覧ください)。

コード:

```
DDRA = 0x00; // 全ピンで方向を入力に設定
PORTA = 0xFF; // ピンでのプルアップを許可
```

別の節電機能は電力削減レジスタ(PRR)でチップ上の未使用部署を禁止することです(データシートで「9.10. 消費電力の最小化」項をご覧ください)。

TWI、タイマ/カウンタ0、タイマ/カウンタ1、直列周辺インターフェース部署は使用せず、従ってこれらの部署を禁止します。

コード:

```
void enable_prr ()
{
    power_spi_disable ();
    power_timer0_disable ();
    power_timer1_disable ();
    power_twi_disable ();
}
```

走行時間は9秒に増やされます。

5.3. 8MHzから2MHzへクロック前置分周

8MHzで使うために低電圧検出(BOD:Brown Out Detection)レベルは2.7Vに設定しなければなりません。ATmega88PAは8MHzクロック設定に於いてより低い電圧で動くことは評価されていません(データシートで「28.3. 速度勾配」項をご覧ください)。従って、2.7Vで実行を停止し、これはこの設定で利用できない多くのエネルギーがコンデンサに残されていることを意味します。

8MHzで走行するデバイスの処理能力が必要ではなく、代わりにシステム周波数を2MHzに減らすべきで、これはより低い全体電力消費とBOD設定を1.8Vに変更することを許します。

未だ19200の安定なボーレートを生成することができます。4で前置分周したRC発振器出力は2MHz(8MHz/4=2MHz)を得、これにはシステムクロック前置分周レジスタ(CLKPR)を使用します。

AVRツールチェーンはpower.hヘッダファイルで見つかる、これ用のマクロを含みます。

コード:

```
clock_prescale_set (clock_div_4); // これはクロックを4分周します。
```

走行時間は40秒に増やされます。

5.4. 次送信待機中にパワーセーブ形態を使用

次のUARTデータ送信を待つ間、Amega88PAが何のデータも処理しないので、更に電力消費を低減するためにデバイスを休止形態に置くことは良い考えです。

パワーセーブ動作形態(データシートで「9.1. 休止形態種別」項をご覧ください)で起動元としてタイマ/カウンタ2を使用することができます。

データが全く不正になり得ないため、パワーセーブ休止形態で低電圧検出(BOD)は必要とされません。これは休止からの起き上がりで自動的に再許可され(データシートで「9.2. 低電圧検出器(BOD)禁止」項をご覧ください)、故に休止中のBOD禁止を選びます。

AVRツールチェーンは休止形態を設定し、BODを禁止し、そしてデバイスを休止に置くための関数を持つsleep.hヘッダファイルを含みます。これらの関数は以下のように呼び出されます。

- set_sleep_mode() - SLEEP命令が実行される時に使用するどれかの休止形態を選びます。
- sleep_enable() - 休止形態を許可します。
- sleep_bod_disable() - 休止注のBODを禁止します。
- sleep_cpu() - SLEEP命令を実行します。

走行時間は今や198秒です。

5.5. より高いボーレートを許可するように発振器を校正

電力消費を低減する別の接近方法は活動形態で費やされる時間を減らすことです。

この応用ではUART送信が終わるのを待つ間、CPUが活性です。活動形態で費やされる時間を減らすためにUARTでより高いボーレートを許す周波数へ発振器を校正します。

115.2kbpsの安定なボーレートを生成するために7.3728MHzに近い周波数が必要です(データシートで「19.11. ボーレート設定例」項をご覧ください)。全電圧範囲で利用できるように7.3728MHzよりも僅かに高い周波数の設定が必要です。未だ1.8VのBOD設定でデバイスを動かすことができるように、1.8432MHzを得るのにクロックを4分周することも求められます。

OSCCAL値を\$5Eに変更します。

走行時間は今や217秒に増やされます。

5.6. 結び

あなたの応用を構築してこの資料で言及した節電技法を理解することにより、あなたのPCBにどんな変更も行うことなくエネルギー消費を減らすことができます。

この例では、同じ電源からの流出で走行時間が6から217秒に増やされました。

6. 参照

[1]: ATmega88PAデータシート: http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf

[2]: AVR053応用記述: http://www.atmel.com/dyn/resources/prod_documents/doc2555.pdf



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

© 2010 Atmel Corporation. 全権利予約済

ATMEL®、ロゴとそれらの組み合わせ、それとその他は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 2013.

本応用記述はATMELのAVR4013応用記述(doc8349.pdf Rev.8349A-11/10)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。