

AVR1505 : XMEGA練習 – A/D変換器(ADC)

前提条件

- 必要な知識
 - ・ ATME[®] AVR[®] XMEGA[®] – 基礎実践
 - ・ ATME[®] XMEGA – 事象システム実践(推奨)
- ソフトウェア必要条件
 - ・ ATME[®] AVR Studio[®] 4.18 SP2またはそれ以降
 - ・ WinAVR/GCC 20100110またはそれ以降
- ハードウェア必要条件
 - ・ 絶縁ドライバ
 - ・ Xplain評価基板
 - ・ JTAGICEmk II
- 予想完了時間
 - ・ 2時間

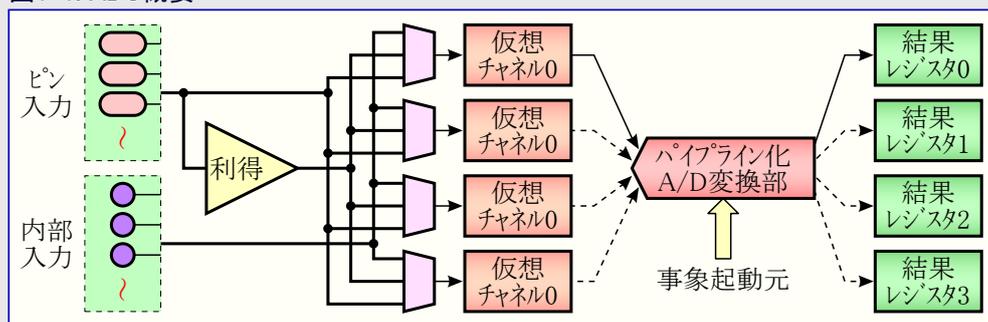
1. 序説

ATME[®] XMEGA A系統のA/D変換器(ADC)は12ビットの分解能を持ちます。それは1秒当たり2百万回(MSPS)までの変換能力があり、12ビットの結果に対して7 ADCクロック周期(利得が許可されている場合は1追加)の変換時間を持ちます。入力選択は柔軟で、シングルエンドと差動の両方の測定を行うことができます。ADCは符号なしと符号付きの両方の結果を提供することができ、動的範囲を増すために任意選択の利得段が利用可能です。

この練習はATxmega128A1マイクロコントローラを特徴とするATME[®] AVR Xplain評価キットに基づきます。ATME[®]の多くのAVRのADCと違い、XMEGA AのADCはパイプライン構造を持ちます。これは各ADCクロック周期で新しいADC測定を採取することができることを意味します。従って、新しいADC測定は他のADC測定が進行中に開始することができます。その結果は変換時間よりもずっと高い変換速度です。

ADC測定は継続的に行われる、または応用ソフトウェアやデバイス内の他の周辺機能からの事象によって開始されるのどちらでもできます。4つまでの異なる仮想チャンネルが提供されます。これらは応用に対してデータの経緯を保つのを容易にするために個別の結果と入力選択(MUX)のレジスタを持ちます。ADCの結果を直接メモリまたは周辺機能へ移動するのにDMAを使用することも可能です。

図1-1. ADC概要



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. 8320A-08/10, 8320AJ1-03/14

2. 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. シングル エント変換

ADC機能の最も一般的な使用法はシングル エント変換です。ここではATMELのXMEGAでこれがどう行われ得るかを学びます。

課題2. 比較機能

ADCは比較機能を持ちます。この課題はそれがどう動くのかと何故それが有用なのかを示します。

課題3. 仮想チャネル

この課題は多数のADC入力が入力の1つの応用で使用される時に仮想チャネルがどうソフトウェアを単純化して実装を容易にするかの基本的な例を示します。

課題4. 事象起動変換

1つまたは多数のADC変換を起動するのに事象を使用することができます。2つのADCを持つデバイスに関しては同時に2つのADCを起動することもできます。

ご幸運を!

3. 課題1: シングル エント変換

シングル エント変換は最も基本的な変換の形式です。外部ピンまたは内部基準のどちらかからの単一電圧がADCへの入力として使用されます。

この課題では仮想チャネル0(CH0)を使用します(仮想チャネルは課題3でもっと詳細に網羅されます)。Xplain基板上の可変抵抗器電圧を測定するように仮想チャネル0を形態設定します。

この課題の目的はあなたが以下の方法を理解することです。

- `adc_driver.c`を用いてADCを構成設定して許可します。
- 入力元としてXplainの可変抵抗器信号を使用します。
- LEDに信号を出力します。

構想は可変抵抗器電圧を変換してLEDポートに出力することです。



すること

1. XMEGA-ADCフォルダを見つけ出してATMELのAVR StudioでSingleEnded.apsプロジェクト ファイルを開いてください。全てのコードと共に必要な`task1.c`と`adc_driver.c`が既にインクルードされています。
2. コード、それがどう動くのか、そしてADCが設定されて単独変換動作で走行するのに何が必要とされるのかを理解するのに或る程度時間を費やしてください。
3. プロジェクトを構築し、異常がないことを確実にしてAVR Studioでデバッグ作業を開始してください。
4. コードを走らせてLEDが可変抵抗器電圧のA/D変換の結果を示すのを見るでしょう。
5. 可変抵抗器をUSBポート近くで見つけ出してドライバで調整してみてください。(潜在的な皮膚電圧のため)絶縁ハンドルの先端が推奨されます。

調整時、値がLEDで直ちに变化しているのを見るべきです。



この課題で使用された基準信号は何ですか?(助言:Cコードを見てください)。

- a) a)での基準使用時に入力信号の上限は何ですか?。
- b) 符号付き形態と符号なし形態間の違いは何ですか?。この課題ではどの形態を使用しましたか?。

4. 課題2: 比較機能

通常のシングル エント変換に加えてADCは比較機能も持ちます。自由走行形態(継続的な採取、連続変換動作)との組み合わせでこの機能が使用されると、ADCは選択可能な閾値の上または下の時に結果を合図(即ち、チャネルの割り込みフラグを設定)します。

この機能は入力信号が特定の電圧よりも高いか低いかを調べたいだけの応用に関して非常に有用です。比較機能なしでは、毎回の変換に対してソフトウェアで結果を調べることが自然です。ATMEL XMEGAのADCに於ける比較機能付きでは、閾値に達した時に割り込みまたは事象経由でADCが応用に合図します。これはCPU負荷を減らし、資源の使用を最小に保ちます。

この課題の目的はあなたが以下をすることです。

- 比較機能を理解する
- 各種の比較割り込み形態と比較値を構成設定できる



すること

1. CompareFunction.apsプロジェクトを見つけ出して`task2.c`を開いてください(コードの殆どは既にインクルードされています)。この課題に対して割り込みを使用することに注意してください。CH0割り込みに対する簡単な割り込み処理部があなたのために既に準備されています。

2. 入力選択に関するコードは直前の課題で既に終了していますが、以下のためにいくつかのコードが追加されなければなりません。
 - a. PMICでの中位割り込み許可
 - b. 正しい割り込み動作形態の設定
3. プロジェクトを構築し、異常がないことを確実にしてATMELのAVR Studioでデバッグ作業を開始してください。
4. コードを走行してください。入力信号を低く(0V近く)を保とうとするなら、LEDには何も起きないでしょう。けれども、入力電圧を比較値以上に増すと、LEDは値の変更を始めるでしょう。
5. AVR Studioのエディタで、結果が閾値よりも低い時の割り込みに対する割り込み動作形態を変更してください(助言:`adc_ch_intmode_below_gc`)。
6. 再コンパイルして再びコードを走らせてください。今や結果が閾値よりも低い時に割り込みを得ることを見るでしょう。

5. 課題3: 仮想チャネル

ATMEL XMEGAのADCは最大4つ(ATMEL XMEGA D系列は2つ)の入力多重器(MUX)と結果レジスタの組を持ち、これらの組は仮想チャネル(CH0~CH3)として参照されます。各仮想チャネルは完全に他と独立しています。これらは全て独立した以下を持ちます。

- 専用の多重器制御(`CHn.MUXCTRL`)レジスタを通した入力選択
- `CHn.RES`と名付けられた結果レジスタ
- 割り込み制御とベクタ
- 変換開始ビット

課題1と課題2では`CH0.MUXCTRL`, `CH0.RES`, 以下同様の仮想チャネル0を使用しました。このような4つのチャネルで、4つの独立した設定を持ち、そしてどんな形態設定の変更もなしに異なるチャネルで変換を開始することの可能性を持ちます。

この課題の目的はあなたが以下をすることです。

- 仮想チャネルの概念とそれらがどう動くかを理解する
- 自由走行(連続変換)動作形態での仮想チャネルの掃引変換の構成設定と開始のこつ



すること

1. `VirtualChannels.aps`プロジェクトを見つけ出してATMELのAVR Studioで開いてください。`task3.c`ファイルを開いてコードを習熟してください。

課題1と課題2との主な違いはCH0とCH1の両方が異なる入力選択設定で構成設定されることです。CH0は前と同様にピン1でのピン入力を使用し、一方CH1は内部チャネル、組み込み温度感知器を読みます。

2. `main.c`の先頭で、掃引動作を許可するためのコードが追加されなければならない場所を見つけ出して、必要なコードを追加してください。`adc_driver.h`ヘッダファイルを参照してください。
3. あなたが見たように、CH0はシングルエンド動作形態で構成設定されます。CH1は入力として(外部ピンではなく)内部供給元を使用するように構成設定されます。内部温度信号を使用するのに欠けているコードを追加してください。
4. 最後に、自由走行(連続変換)動作を許可するコードを追加してください。`adc_driver.h`を参照してください。
5. プロジェクトを構築し、異常がないことを確実にしてAVR Studioでデバッグ作業を開始してください。
6. 下で示されるようにコードに中断点を配置してください。

```

do { } while( (ADCA.INTFLAGS & ADC_CH1IF_bm) != ADC_CH1IF_bm);
ADCA.INTFLAGS |= ADC_CH1IF_bm; //Clear CH1IF
  
```

7. コードを走らせて中断点が設定されている場所でAVR Studioが中断するのを確実にしてください。
8. 数回1行実行(Single step)を行い、2つの仮想チャネルの結果が独立したレジスタで読み出しに関してどう利用可能で、例え(殆ど)同時に異なる入力を測定しても改めて形態設定の必要がないことを見てください。



仮想チャネルがどう動くかを理解しましたか?、そしてそれらの恩恵は何ですか?。

6. 課題4: 事象起動変換

今まで各変換に対してチャネルn単独変換開始(`CHn.START`)ビットを設定(1)することにより、または自由走行(連続変換)動作にADCを設定することによってソフトウェアからA/D変換を開始しました。

この課題では変換を起動するのに事象を使用します。ADCは単独変換または2つ以上の仮想チャネルでの掃引変換を開始するための事象を受け入れるように構成することができます。

加えて、ADCは同期採取を行う、即ちADCを破棄して次の周辺クロック周期で新しい変換を開始するように構成することができます。これは正確な測定のタイミングが必要な時に大変有用です。

この課題の目的はあなたが以下をすることです。

- A/D変換器(ADC)に関して事象起動変換を構成設定できる
- 2つのADCで同期採取をできる
- ADCが入力としてポートAとポートBの両方を使用できることを知る
- コードの基本的なデバッグを行うことができる



すること

1. **EventTriggered.aps**プロジェクトを見つけ出してATMELのAVR Studioで開いてください。**task4.c**ファイルを考察してコードを習熟してください。いくつかのCコードが欠けているのも見るでしょう。

結局、欠けたコードが満たされた時にADC部署の1つ(ADCBは基板上の周辺機能から測定することができ、一方ADCA入力は10ピンのポートAヘッダで利用可能です。)を事象起動と同期変換のために使用することができます。変換はタイマ/カウンタC0が溢れる度ごとに事象を生成するのに事象システムを使用することによって同期されます。

2. 最初に、ADC部署へのポインタを構成設定することが必要です。ポインタは名称**ADC_Pointer**を持つ型**ADC_t ***で、ADCB (&ADCB)のアドレスを含むべきです。main()関数の先頭で**"ADD CODE"**文字を探してそれを満たしてください。

3. ATMEL XMEGA A手引書を開いて事象システム チャネル0多重器(**EVSYS.CH0MUX**)を構成するマクロを探すためにヘッダファイル(**iox128a1.h**)内を覗いてみてください。

4. **InitADC()**関数に於いて、掃引とその同期の構成設定が必要です。欠けているコードを追加してください。これを良く知らない場合、XMEGA A手引書を参照してください。

ADC構成用のコードは直ぐに完成すべきです。

5. プロジェクトを構築し、異常がないことを確実にしてAVR Studioでデバッグ動作を開始してください。

6. コードを走行してください。(課題1でのように)可変抵抗器を調整することによって電圧を変化する時に、コードが正しければLEDが更新すべきです。

7. コード内の何かの修正が必要な場合、プロジェクトを再び構築して試験を行ってください。レジスタを調べるのにI/Oウィンドウを用いてJTAGICEmk IIでいくつかの基本的なデバッグを行うように中断点を設定してください。

8. ポートBのADCBの代わりにポートAのADCAを使用するようにコードを変更してください。コードの1行の変更を必要とするだけです。再構築してプログラムを走らせてください。測定用の信号を得るためにXplain上の**PORTA**ヘッダにアナログ供給元が接続されなければならないことに注意してください。

9. 時間が有るなら、ポートAピンとポートBピンの両方からの変換結果をLEDに出力するようにコードを変更してください。あなたは結果を2つの部分に分け、従って結果の各々の4ビットを8つのLEDに出力するかもしれません。この最後の練習の目的は1つの事象で同時に2つの同期した変換を行うことです。

7. 要約

ここはA/D変換器(ADC)練習の要点のいくつかです。

- 単独変換動作
- 自由走行(連続変換)動作
- 比較機能
- 仮想チャンネル
- 起動への事象システムの使用と同期変換

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/>

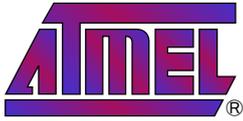
9. ATMEL技術支援センター

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



本社

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan

104-0033 東京都中央区
新川1-24-8
東熱新川ビル 9F
アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ウェブサイト

www.atmel.com

技術支援

avr@atmel.com

販売窓口

www.atmel.com/contacts

文献請求

www.atmel.com/literature

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、または利益の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。

© Atmel Corporation 2010. 全権利予約済 ATMEL[®]、ロゴとそれらの組み合わせ、AVR[®]とその他はATMEL Corporationの登録商標、XMEGA[®]とその他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。

© HERO 2014.

本応用記述はATMELのAVR1505応用記述(doc8320.pdf Rev.8320A-08/10)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。