

AVR126 : シングル エント 形態でのmegaAVRのA/D変換器(ADC)

応用記述

序説

Atmel® megaAVR® デバイスは10ビットの分解能で15kspsまでの変換速度の能力がある逐次比較A/D変換器(ADC)を持ちます。これは多数のシングル エント 入力ピンとデバイス内のバンドギャップ基準電圧からの内部チャンネルで電圧を測定することをADCに許す柔軟な多重器が特徴です。シングル エント 入力チャンネルはGNDに対して参照されます。

この応用記述は開始に際してのAtmel ATmega88でのコード例で、シングル エント 動作形態でAtmelのmegaAVRデバイスに於けるADCの基本機能を記述します。このコード例はC言語で書かれ、機能性に関してはAtmel STK®600スターキットで試験されています。

特徴

- 10ビットまでの分解能
- Atmel ATmega88に関して最大76.9ksps(採取/秒)
- 最大分解能で最大15ksps(採取/秒)
- 自動起動と単独変換の動作形態
- A/D変換結果読み取りに対する左揃え任意選択
- 休止動作形態雑音消去器
- ATmega88用に含められたドライバ ソースコード
 - ATmega88 ADC – 単変換動作形態
 - ATmega88 ADC – 連続変換(自由走行)動作形態と変換完了割り込み
 - ATmega88 ADC – 起動元としてタイマ/カウンタ0の比較事象を用いる自動起動
 - ATmega88 ADC – バンドギャップ(内部)基準電圧の測定
 - ATmega88 ADC – 雑音低減休止形態

目次

序説	1
特徴	1
1. 部署概要	3
1.1. ADC操作	3
1.2. 入力元	3
1.2.1. 内部入力	4
1.2.2. シングルエンド入力	4
1.3. 変換の開始	4
1.4. A/D変換クロックと変換タイミング	4
1.5. チャンネルや基準電圧選択の変更	4
1.6. A/D変換雑音消去器	4
1.7. 変換結果	5
1.8. アナログ入力回路	5
1.9. 精度改善のための最善策	5
2. 始める前に	6
2.1. STK600でコードを試験するための全般指示	6
2.2. 単独変換形態	6
2.2.1. 試験方法	6
2.3. 連続変換(自由走行)形態と変換完了割り込み	6
2.4. 起動元としてタイマ/カウンタの比較事象を用いる自動起動	7
2.5. ハントキャップ(内部)基準電圧の測定	7
2.6. 雑音低減休止	8
3. デバイス データシート参照	8
4. ドライバ実装	8
5. 推奨読み物	8
6. 改訂履歴	9

1.2.2. 内部入力

入力として内部バンドギャップアナログ電圧またはGND信号を選択することができ、ADCによって測定されます。このバンドギャップ電圧は内部電圧基準に対する供給元で、マイクロコントローラ内部の正確な基準電圧です。

1.2.1. シングルエンド入力

シングルエンド測定に関しては入力として全てのアナログ入力ピンが使用できます。全てのシングルエンドチャンネルはGNDに対して参照されます。アナログ入力電圧はADCに対して選択された基準電圧よりも大きくすることができません。

1.3. 変換の開始

シングルエンド形態では変換の開始のためにA/D変換制御/状態レジスタA(ADCSRA)のA/D変換開始(ADSC)ビットが論理1を書かれなければなりません。このビットは変換が進行中の間は論理1に留まり、一旦変換が完了するとハードウェアによって解除(0)されます。

自動起動形態では、様々な供給元によって変換が自動的に起動されます。自動起動を許可するには、ADCSRAレジスタのA/D変換自動起動許可(ADATE)ビットが設定(1)されなければなりません。起動供給元はA/D変換制御/状態レジスタB(ADCSRБ)のA/D変換自動起動要因選択(ADTSn)ビットの手助けで選ぶことができます。自動起動形態は起動元に基づいて形態設定可能な固定間隔で変換開始の方法を提供します。

割り込み要求フラグは例え指定割り込みまたは全体割り込みが禁止されていても設定(1)されます。従って変換は割り込みなしでA/D変換完了割り込み要求フラグ(ADIF)を用いて起動することができます。自動起動形態に於いて次の割り込み事象で起動するためには、ADIFが手動で解除(0)されなければならないことに注意してください。

ADCが連続変換形態で動く場合、一度直前の変換が完了すると、次の変換が起動されます。連続変換(自由走行)形態に対して、ADCはADIFが解除(0)されているか否かのどちらかに拘らず連続的に変換を実行します。最初の変換はADSCビットの設定(1)によって始められなければなりません。

1.4. A/D変換クロックと変換タイミング

ADCは最大分解能を得るために50~200kHz間のA/D変換クロックを提供するようにシステムクロックを前置分周することができます。10ビットよりも少ないA/D変換分解能が必要とされる場合、A/D変換クロック周波数は200kHzよりも高くすることができます。1MHzに於いて8ビットまでの最大分解能の達成が可能です。

前置分周器値はA/D変換制御/状態レジスタA(ADCSRA)のA/D変換クロック選択(ADPSn)ビットで選ばれます。ADCSRAのA/D変換開始(ADSC)ビット設定(1)によるシングルエンド変換開始時、その変換は後続するA/D変換クロック周期の上昇端で始まります。

ADCがONに切り替わった(ADCSRAのA/D許可(ADEN)が設定(1)後の初回変換はアナログ回路を初期化するために25 A/D変換クロック周期かかります。その後の更なる変換については13(自動起動変換については13.5)A/D変換クロック周期かかります。バンドギャップ電圧がADCへの入力に使用される時は、電圧安定のために或る時間かかります。バンドギャップ基準電圧に対する始動時間はデータシートの「システムとリセットの特性」項で利用可能です。

1.5. チャンネルや基準電圧選択の変更

A/D変換多重器選択(ADMUX)レジスタのA/Dチャンネル選択(MUXn)と基準電圧選択(REFS1,0)はCPUが乱アクセスを持つための一時レジスタを通して単一緩衝されます。自動起動が使用される場合、ADMUXは以下で安全に更新することができます。

- 自動起動許可(ADATE)またはA/D許可(ADEN)が解除(0)されている時
- 変換中、起動事象後最低1 A/D変換クロック周期後
- 変換後、起動元として使用される割り込み要求フラグが解除(0)される前

これらの方法では、新しい設定が次のA/D変換に影響を及ぼします。

単独変換動作形態では変換開始前にチャンネルが選択されなければなりません。チャンネルはA/D変換開始(ADSC)ビット設定(1)後1クロック周期で変更することができますが、チャンネル変更前に変換完了を待つことがより良いことです。

連続変換(自由走行)動作形態では最初の変換を始める前にチャンネルを選んでください。これはA/D変換開始(ADSC)ビット設定(1)後1クロック周期で変更することができますが、チャンネル変更前に変換完了を待つことがより良いことです。次の変換が既に自動的に開始されるため、その変更は後続する次の変換に反映されます。

1.6. A/D変換雑音消去器

Atmel megaAVRのA/D変換器(ADC)は休止形態中の変換を許可する雑音消去器を持ち、これはCPUコアと他のI/O周辺機能から誘導された雑音を低減します。この機能はA/D変換雑音低減とアイドルの動作形態で利用可能です。この機能を使用するには、

- ADCが許可されて変換で多忙でないことを確実にしてください。単独変換動作形態が選択され、A/D変換完了割り込みが許可されなければなりません。
- A/D変換雑音低減(またはアイドル)動作形態に移行してください。一旦CPUが停止されると、ADCは変換を開始します。
- A/D変換完了前に他の割り込みが起きなければADC割り込みがCPUを起こしてA/D変換完了割り込み処理ルーチンを実行します。A/D変換が完了する前に別の割り込みがCPUを起こした場合、その割り込みが実行され、A/D変換完了割り込み要求はA/D変換が完了する時に生成されます。CPUは新しい休止(SLEEP)命令が実行されるまで活動形態に留まります。

1.7. 変換結果

ADC変換完了(ADIF設定(1))後、ADCHとADCLのレジスタで10ビットの結果が利用可能になります。シングルエンド変換について、その結果は次の通りです。

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}}$$

ここでの V_{IN} はアナログ入力電圧を表し、 V_{REF} は選択した基準電圧を表します。 $\$000$ はGNDを表し、 $\$3FF$ は基準電圧-1LSBを表します。

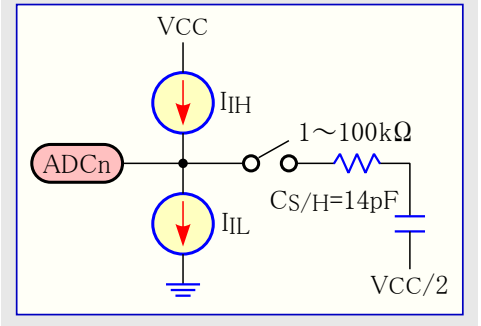
1.8. アナログ入力回路

シングルエンドチャンネル用のアナログ入力回路が下図で示されます。ADC入力ピンに印加されるアナログ供給元は例えADC用の入力として選択されていなくても、ピン容量とそのピンの入力漏れに従属されます。特定のチャンネルが選択されている時に供給元は入力経路で直列(複合)抵抗を通して採取/保持(S/H)コンデンサを駆動しなければなりません。

ADC部署は10k Ω またはそれ以下の出力インピーダンスを持つアナログ信号用に最適化されています。このような供給元が使用されるなら、採取時間は無視しても良いほどでしょう。

供給元インピーダンスが10k Ω よりも高い場合、コンデンサを変更するのにかかる時間が増加して(広範囲に変わり得る)不正確な結果を生成するかもしれません。ADC入力で使用される分圧器が抵抗網を使用する場合、その供給元インピーダンスは10k Ω よりも小さいことを確実にしてください。この充電転移用の時間を最小とするために、緩やかに変化する信号に対して低インピーダンスが使用されなければなりません。(予想不能な信号の渦からの歪を避けるため)ナイキスト周波数($f_{ADC}/2$)よりも高い周波数成分を低域通過濾波器で取り除くことが推奨されます。

図1-2. アナログ入力回路



1.9. 精度改善のための最善策

ADCの精度は入力信号と電源の品質に依存します。ADC測定の精度を改善するために以下の点が考慮に入れられるべきです。

- A/D変換器とその機能、そしてそれらがどう使用されるべく意図されているかを理解してください。
- 応用の必要条件を理解してください。
- アナログ基準電圧(V_{REF})とアナログ電源($AVCC$)のようなアナログ信号経路の設計時、大いに注意することが重要です。 $AVCC$ はADCへの供給電圧です。
- アナログ電源($AVCC$)がデジタル電源に接続される場合、濾波が使用されるべきです。即ち、デバイスの $AVCC$ ピンはLC網経由でデジタル電源(VCC)ピンに接続されるべきです。このLC網接続のより多くの情報については各々のAtmel megaAVRデバイスのデータシートを参照してください。 $AVCC$ と VCC が高インピーダンス経路によって内部的に接続されていることに注意してください。
- Atmel megaAVRデバイスに関して $AVCC$ は VCC から $\pm 0.3V$ よりも大きく違ってはなりません。
- 基準電圧は $AREF$ ピンとGND間にコンデンサを接続することによってより大きな雑音耐性にすることができます。
- アナログ信号経路は可能な限り短く保ってください。ADCチャンネルに対するPCB布線上のインピーダンスが大きくないことも重要で、これはADCの採取/保持(S/H)コンデンサのより長い充電周期に帰着します。
- アナログ布線はアナログGND面上の走行を確実にしてください。
- 高速切り替え雑音(即ち通信線、クロック信号)を持つデジタル信号経路に近いアナログ信号経路を避けてください。
- シングルエンド入力に対して信号入力とGND間のアナログ信号の雑音分離(デカップ)を考慮してください。
- 差動信号について、正と負の入力間で雑音分離(デカップ)をしなければなりません。雑音分離コンデンサ値は入力信号に依存します。信号が高速切り替えなら、雑音分離コンデンサはより低くなければなりません。
- 使用される採取速度に対して供給元インピーダンスが高すぎないことを確実にしてください。供給元インピーダンスが高すぎる場合、内部採取コンデンサが正しいレベルに充電されず、結果が正確にならないでしょう。
- 精度に影響を及ぼす切り替え雑音を防ぐため、ADC変換が進行中の間にポートピンの切り替えを避けるようにしてください。ADCは特にアナログ電源によって給電される入出力ピン(ATmega88の場合はポートC)の切り替えによりもっと敏感です。
- 電力消費を最小とするためにADCチャンネルに対応する入出力ピンのデジタル入力を禁止してください。
- 未使用周辺機能からのどんな雑音も無くするために電力削減レジスタ(PRR)で各々のビットを設定(1)することによって未使用周辺機能をOFFに切り替えてください。
- 精度を改善するため、測定に変位(オフセット)と利得の校正を加えてください。
- もっと正確な結果を得るためにA/D変換雑音低減動作を使用してください。
- ADC変換を起動する前に、ADCが進行中のどの変換も完了するまで待ってください。基準電圧と入力元に対して安定にされるべき十分な時間が与えられることを確実にしてください。例えば、ハントギャップ電圧はADC入力として選択される時に安定のための或る量の時間が必要です。

- ・ 分解能を増して乱雑音を無くすために過採取を使用してください。
- ・ 入力多重器(MUX)設定や基準電圧選択が変更された時には必ず最初の変換結果を破棄することが推奨されます。
- ・ (利得設定を持つ)差動チャネルへ切り換える時は自動変位(オフセット)消去回路に対する安定時間が必要とされるため、最初の変換が貧弱な精度を持つかもしれないことに注意してください。
- ・ 応用の必要に基づいて1点(変位(オフセット))と2点(変位と利得)のような直線補完法を使用することができます。

2. 始める前に

本章は多重器(MUX)設定での簡単なADC変換と実験とで開始するための基本的な段階を簡単に片付けます。必要なレジスタが関係するビット設定と共に記述されます。この応用記述で供給されるコードでは、容易にするために、同じレジスタへ書く多段階は結合され、実効値は単一段階で書かれます。

2.1. STK600でコードを試験するための全般指示

- ・ 指定した配線カードとソケットカード(STK600-RC032M-29とSTK600-TQFP32)を使用してAtmel ATmega88デバイスをSTK600上に配置してください。
- ・ Atmel StudioでメニューのTools⇒Device Programmingからデバイスをアクセスしてみてください。
- ・ AREF1からの電圧がADCへの入力として使用することができます。Atmel StudioでメニューのTools⇒Device Programming⇒Board settings経由でAREF1電圧を調整してください。
- ・ STK600上でAREF1を例で使用されるADC入力チャネル(ADC0(PC0))に接続してください。
- ・ STK600に於いて、PB0をLEDに接続してください。このピンは視覚的表示を与えるために以下の例で使用されます。

2.2. 単独変換形態

課題：ADC入力1の1つのシングルエンド変換

このプログラムに於いてADC部署を初期化するのに'initialize()'ルーチンが使用されます。応用がADC変換を必要とする時は必ず、'convert()'ルーチンが呼ばれなければなりません。

1. ADCチャネル0を選択するためにA/D変換多重器(ADMUX)レジスタのA/Dチャネル選択(MUX3~0)ビット領域を0000と等しく設定してください。
2. ADC部署を許可するためにA/D変換制御/状態レジスタA(ADCSRA)のA/D許可(ADEN)ビットを設定(1)してください。
3. システムクロックを16前置分周するためにADCSRAの前置分周選択(ADPS2~0)ビット領域を100と等しく設定してください。
4. 内部1.1V基準電圧を選択するためにADMUXの基準電圧選択(REFS1,0)ビット領域を11と等しく設定してください。
5. 単独変換を開始するためにADCSRAのA/D変換開始(ADSC)ビットを設定(1)してください。
6. 新しい変換が完了されたことを示す、設定(1)されるべきADCSRAのA/D変換完了割り込み要求フラグ(ADIF)をポーリングして(待ってください)。
7. ADIFビットが1になった後、2バイト値として10ビットの変換結果値を得るためにADCデータレジスタ対(ADCL/ADCH)を読んでください。

2.2.1. 試験方法

1. プロジェクトを構築してhexファイルをデバイスに設定して(書いて)ください。
2. 上の「STK600でコードを試験するための全般指示」で記述されるようにSTK600で準備してください。
3. PC0に印加される電圧を調整し、その電圧に依存してLED表示が更新されるかどうかを調べてください。
4. PC0の電圧が約0.5Vよりも高い場合、PB0に接続されたLEDはOFFに留まり、さもなければLEDはONに留まります。

2.3. 連続変換(自由走行)形態と変換完了割り込み

課題：ADCチャネル0での連続変換(自由走行)。変換完了割り込みの使い方

このプログラムではADC部署を初期化するのにinitialize()ルーチンが使用されます。A/D変換完了割り込みが許可されると、変換が完了されると直ぐにこの割り込み用の割り込み処理ルーチン(ISR:Interrupt Service Routine)が起動されます。連続変換(自由走行)形態では、変換完了後、直ちに新しい変換が開始されます。変換中、A/D変換開始(ADSC)ビットは1に留まります。2つの連続するADC採取間の時間はADC変換時間に依存します。

1. 上の「単独変換形態」の手順1~4を繰り返してください。
2. A/D変換完了割り込みを許可するためにA/D変換制御/状態レジスタA(ADCSRA)のADC割り込み許可(ADIF)ビットを1と等しく設定してください。
3. 自動起動動作形態を許可するためにADCSRAの自動起動許可(ADATE)ビットを1と等しく設定してください。既定により、A/D変換制御/状態レジスタB(ADCSRB)の自動起動要因選択(ADTS2~0)ビット領域は連続変換(自由走行)を表す000に設定されています。
4. 初回変換を開始するためにADCSRAのA/D変換開始(ADSC)ビットを設定(1)してください。
5. 変換終了(ADIFビットが1になった)後、CPUは10ビットのA/D変換結果を得るためにADCデータレジスタ対(ADCL/ADCH)を読む、ADC割り込み処理ルーチンを実行します。

試験方法

6頁の「試験方法」を参照してください。

2.4. 起動元としてタイマ/カウンタ0の比較事象を用いる自動起動

課題：起動元としてタイマ/カウンタ0を使用することによるADCチャンネル0での自動起動変換

このプログラムではADC部署を初期化するのにinitialize_adc()ルーチンが使用されます。initialize_Timer()ルーチンはタイマ/カウンタ0を比較A一致事象に形態設定します。ADCの結果は変換が完了されると直ぐに対応するISR内で読まれます。ADC部署はタイマ/カウンタ0がその比較一致値に達する時に必ず(即ちこの例では10ms毎に)、変換を開始します。

'OCR0A'比較レジスタとタイマ/カウンタ0制御レジスタB('TCCR0B')のタイマ/カウンタ0クロック前置分周ビットの変更により、応用の必要によって変換間隔を(10msから)変更することができます。

- 6頁の「単独変換形態」の手順1.~4.を繰り返してください。
- ADC割り込みを許可するためにA/D変換制御/状態レジスタA(ADCSRA)のADC割り込み許可(ADIF)ビットを設定(1)してください。
- 自動起動動作形態を許可するためにADCSRAの自動起動許可(ADATE)ビットを設定(1)してください。
- ADC開始の引き金としてタイマ/カウンタ0比較A一致事象を使用するように、A/D変換制御/状態レジスタB(ADCSR)の自動起動要因選択(ADTS2~0)ビット領域を'011'に設定してください。
- タイマ/カウンタ0制御レジスタA(TCCR0A)の波形生成種別(WGM1,0)ビット領域を'10'(比較一致タイマ/カウンタ解除)に設定してください。
- TCCR0Bのクロック選択(CS2~0)ビット領域を'011'(前置64分周)に設定してください。
- ADCが10ms(比較一致事象)間隔毎にアナログ値を変換するように、比較Aレジスタ(OCR0A)を望む値(この例では156)に設定してください。
- 変換終了(ADIFビットが1になった)後、CPUは10ビットのA/D変換結果を得るためにADCデータレジスタ対(ADCL/ADCH)を読む、ADC割り込み処理ルーチンを実行します。A/D変換は形態設定されたタイマ/カウンタ0の比較が起こる時に必ず起動されます。

試験方法

6頁の「試験方法」を参照してください。

2.5. バントギャップ(内部)基準電圧の測定

課題：バントギャップ基準電圧測定

このプログラムではADC部署を初期化するのにinitialize()ルーチンが使用されます。measure_gnd()ルーチンはGND値を測定します。measure_bandgap()ルーチンは内部バントギャップ基準電圧を測定します。

- 6頁の「単独変換形態」の手順1.~3.を繰り返してください。
- ADC基準電圧としてAVCCを選択するためにA/D変換多重器(ADMUX)レジスタの基準電圧選択(REFS1,0)ビット領域を01と等しく設定してください。
- ADC割り込みを許可するためにA/D変換制御/状態レジスタA(ADCSRA)のADC割り込み許可(ADIF)ビットを1と等しく設定してください。
- スイッチが押されたなら、多重器選択(MUX3~0)ビット領域を1111と等しく設定することによってGNDを測定するようにADCを形態設定してください。これはADCのコンデンサを放電するために行われます。
- GND測定の間中、ADC割り込みと自動起動(連続変換(自由走行))動作は禁止されます。
- 変換を開始するためにADCSRAのA/D変換開始(ADSC)ビットを設定(1)してください。
- 変換終了(ADIFビットが1になった)後、10ビットの変換結果値を得るためにADCデータレジスタ対(ADCL/ADCH)を読んでください。
- ADC入力としてバントギャップ基準電圧を選択するために多重器選択(MUX3~0)ビット領域を1110に形態設定してください。バントギャップ基準電圧の始動時間のため、これは70 μ s遅延後に測定されるべきです。
- 変換完了を調べるのにポーリング法が使用されます。自動起動(連続変換(自由走行))動作形態は禁止されます。
- 変換を開始するためにADCSRAのA/D変換開始(ADSC)ビットを設定(1)してください。
- ADIFビットが1になった後、10ビットの変換結果値を得るためにADCデータレジスタ対(ADCL/ADCH)を読んでください。

試験方法

- PB0をSTK600で利用可能なスイッチの1つに接続してください。
- Atmel Studio 7でプロジェクトを開いてください。デバッグを開始するためにAt+F5を押してください。
- デバッグWIREが既に許可されていない場合、Atmel StudioはデバッグWIREを許可するように指示するでしょう。
- デバッグ動作へ行った後、measure_bandgap関数の最後に中断点(ブレークポイント)を設定してください。
- コードを走らせてPB0に接続したスイッチを押してください。
- この点で、実行はmeasure_bandgap関数内に設定された中断点に当たります。
- ADC読み出しを見るために監視(Watch)ウィンドウにbg_val変数を追加してください。ADC結果レジスタは(メニューのDebug⇒Windows⇒I/Oを用いて)I/Oウィンドウ経由で調べることができます。

2.6. 雑音低減休止

課題：A/D変換雑音低減休止の使い方

このプログラムではADC部署を初期化するのにinitialize()ルーチンが使用されます。この例はA/D変換雑音低減休止動作形態を使用します。'sleep'命令を実行することによって休止動作形態へ移行してください。ADCはCPUが停止してしまっただ後に変換を開始します。ADC変換完了割り込みが許可されているため、変換が完了されると直ぐにこの割り込み用の割り込み処理ルーチン(ISR:Interrupt Service Routine)が起動されます。A/D変換開始(ADSC)ビットは変換中、1に留まります。ISRの実行が終了されると、実行はmainルーチンになり、再び'sleep'命令を実行して別の変換を起動します。

1. 「**単独変換形態**」からの手順1.~4.を繰り返してください。
2. ADC変換完了割り込みを許可するためにA/D変換制御/状態レジスタA(ADCSRA)のADC割り込み許可(ADIF)ビットを1と等しく設定してください。
3. 自動起動動作形態を許可するためにADCSRAの自動起動許可(ADSCF)ビットを1と等しく設定してください。既定により、A/D変換制御/状態レジスタB(ADCSRB)の自動起動要因選択(ADTS2~0)ビット領域は連続変換(自由走行)動作を表す000に設定されます。
4. 最初の変換を開始するためにADCSRAのA/D変換開始(ADSC)ビットを設定(1)してください。
5. 変換終了(ADIFビットが1になった)後、CPUは10ビットのA/D変換結果を得るためにADCデータレジスタ対(ADCL/ADCH)を読む、ADC割り込み処理ルーチンを実行します。

試験方法

6頁の「**試験方法**」を参照してください。

3. デバイス データシート参照

この応用記述はAtmel ATmega88系のADC部署を記述します。Atmel megaAVRデバイスのチップ上ADCの操作は殆ど同じです。より多くの詳細については各々のデバイスのデータシートを参照してください。例えば、(Atmel ATmega48Pのような)いくつかのAtmel megaAVRデバイスの別種はADC入力の1つとしてチップ上温度感知器を持ち、(Atmel ATmega2560のような)いくつかのデバイスは形態設定可能な利得を持つ差動ADCチャネルと2.56V内部基準電圧を持ち、(Atmel AT90PWM1のような)いくつかのデバイスは125kspsのような高いADC速度を支援します。特定デバイスのADCの詳細な情報についてはデータシートの「A/D変換器」章を参照してください。

データシートの「電気的特性」下の「A/D変換器特性」と「電気的特性」下の「システムとリセットの特性」は積分直線性(INL)、微分直線性(DNL)、絶対精度、変換時間、変位(オフセット)誤差、利得誤差、入力抵抗、バンドギャップ基準電圧始動時間などのようなADCに関連する特性を含みます。

データシートの「障害情報」章はそのデバイスに対してADCに関連する何らかの既知の不具合についての情報を含みます。言及された対策を持つ何かの障害情報がある場合、その対策が実装されているのを確実にしてください。

4. ドライバ実装

この応用記述はC言語で実装された基本ADCドライバと共にソースコード一式を含みます。それはAtmel ATmega88デバイス用にAtmel Studio 7で開発されています。このADCドライバが高性能モードでの使用が意図されていないことに注意してください。これはADCで始める際のライブラリとして設計されています。

1. 単独変換動作
2. 連続変換(自由走行)動作と変換完了割り込み
3. 自動起動元としてタイマ/カウンタ0を用いる自動起動
4. バンドギャップ基準電圧の測定
5. 雑音低減休止

5. 推奨読み物

- AVR042：ハードウェア設計の考察 - この応用記述はAtmel AVRマイクロコントローラを伴う設計開始時に直面する疑問や問題のいくつかに対する回答を提供します。
- AVR120：AVRのA/D変換器の特性付けと校正 - この応用記述は様々なADC(A/D変換器)特性パラメータ、それらがADC測定にどう影響を及ぼすか、それらを測定する方法、走行時補償を実行する方法を説明します。
- AVR121：過採取によるA/D変換分解能増強 - この応用記述は外部ADCの使用なしに、より高い分解能を達成するための「過採取と間引き」と呼ばれる方法とこの方法を正しく動かすために満たされるべき必要な条件を説明します。
- AVR122：AVRの内部温度基準の校正 - この応用記述はATtiny25/45/85からの温度測定の校正と補償の方法を記述します。これは内部温度感知器を持つ他のAVRマイクロコントローラでも使用することができます。
- AVR125：シングルエンド形態でのtinyAVRのA/D変換器(ADC) - この応用記述は開始に際するAtmel ATtiny88でのコード例と共にシングルエンド形態でのAtmel tinyAVRデバイス内のADCの基本的な機能を記述します。
- AVR127：A/D変換器(ADC)パラメータの理解 - この応用記述はA/D変換器(ADC)の基本的な概念とADCの性能を決める様々なパラメータについて検討します。これらのADCパラメータはそれらがADCの出力の精度を決める一部なのでかなり重要です。

6. 改訂履歴

資料改訂	日付	注釈
8444A	2011年10月	初版資料公開
8444B	2016年3月	新しい例を追加

Atmel®, Atmelロゴとそれらの組み合わせ、Enabling Unlimited Possibilities®, AVR®, STK®, megaAVR®とその他は米国及び他の国に於けるAtmel Corporationの登録商標または商標です。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。

安全重視、軍用、車載応用のお断り: Atmel製品はAtmelが提供する特別に書かれた承諾を除き、そのような製品の機能不全が著しく人に危害を加えたり死に至らしめることがかなり予期されるどんな応用(“安全重視応用”)に対しても設計されず、またそれらとの接続にも使用されません。安全重視応用は限定なしで、生命維持装置とシステム、核施設と武器システムの操作の装置やシステムを含みます。Atmelによって軍用等級として特に明確に示される以外、Atmel製品は軍用や航空宇宙の応用や環境のために設計も意図もされていません。Atmelによって車載等級として特に明確に示される以外、Atmel製品は車載応用での使用のために設計も意図もされていません。

© HERO 2016.

本応用記述はAtmelのAVR126応用記述(Rev.8444B-03/2016)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。