

# AVR474 : ATAVRSB202ファームウェア使用者の手引き

## 要点

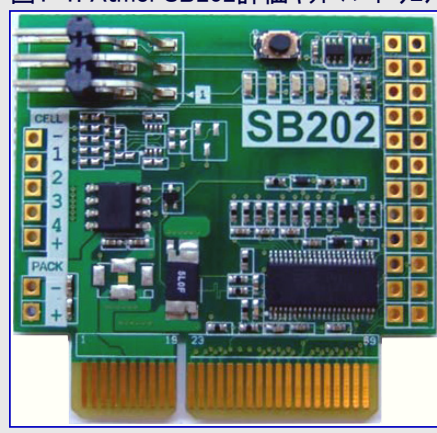
- リチウム イオン電池に関して必要とされる全ての安全な測定を提供
  - ・ 電圧
  - ・ 電流
  - ・ 温度
- 高精度の電圧と電流の測定
- 通信
  - ・ TWI/SMBus 1.1適合
  - ・ スマート電池使用に基づく命令1式
- HMAC-SHA2認証

## 1. 序説

この資料はAtmel®のATAVRSB202評価キットを用いるスマート電池実装例を記述します。この実装は再充電可能なリチウム イオン電池パックに関して正確な測定と最適な安全を得るのにAtmelのATmega16HVB/32HVBをどう使うかを実演します。全てのコードは容易な評価と更なる開発を許すために限定された許諾契約下で利用可能です。

SB202はリチウム イオンとリチウム ポリマの2、3、または4つの直列セルでのスマート電池が目的です。この製品は同じファームウェア ソース コードに基づきますが、各構成設定に対して異なるようにコンパイルされます。ATmega16HVB/32HVBデバイスの評価用のSB202ハードウェアとSB202実行可能ファイルはAtmelのATAVRSB202評価キットとして入手可能です。

図1-1. Atmel SB202評価キット ハードウェア



## 2. 概要

スマート電池実装の主な目的は過充電、過放電、それと温度と電流の限度外での使用から電池を保護することです。また、電池の残量を予測し、これをホスト/応用へ伝達します。

度々、スマート電池は電池を認証するための方法を提供します。認証は適合する電池だけが応用に使われるのを保証することを可能にします。これは応用が電池の寿命と安全性に関して意図されるような働きを保証することを応用の製造業者に許します。

AtmelのATmega16HVB/32HVBは回路短絡と両方向での過大な電流に対して保護するための専用の自律ハードウェアを持ちます。セル電圧レベルや動作温度のような他の安全条件はCPUによって扱われます。良好な性能を保証するために、電池の動作状態についての正確な情報を蓄積するのにクーロン カウンタA/D変換器と電圧A/D変換器(CCADCとVADC)が使われます。CPU/ファームウェアはこの情報を収集して処理し、それによって応じます。専用且つ特化されたハードウェア単位部はデータシートで記述され、この応用記述はそれらがそれらの最善にどう使われ得るかを実演します。

CCADCは固定間隔で割り込みを起動するように設定することができ、パワーセーブ動作形態で動きます。この理由のため、この実装に於いて時間元として使われます。CCADCは1秒間の電流を累積(平均)し、毎回変換が完了され、主繰り返しは1回の繰り返しを走行してATmega16HVB/32HVBは休止に戻ります。どの深さで休止し得るかは主繰り返しで開始された全ての単位部が終了されたか否か、実行中の通信があるか否かに依存します。

ハードウェアTWI単位部は通信に使われます。全ての転送はホスト/応用によって始められなければなりません、電池は応答することができます。

AtmelのSB202は2、3、または4セル応用のために設計されています。ヘッダ ファイル内の`#define "B ATTPARAM_CELLS_IN_SERIES"`はコードが2、3、または4セル構成設定を目標とするかを決めます。ハードウェアは各種セル数での使用に構成設定されなければなりません。これはハードウェア使用者の手引きで記述されます。



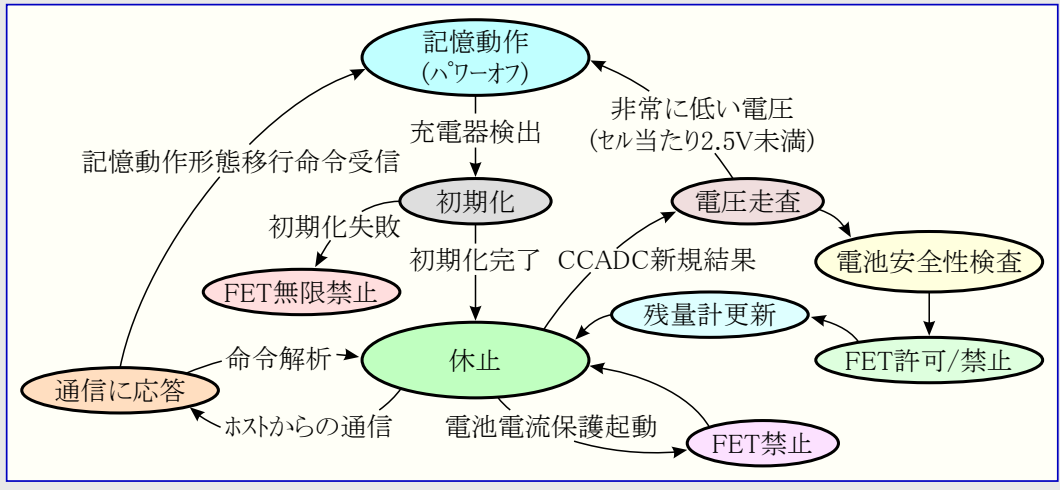
8ビット **AVR**<sup>®</sup>  
マイクロコントローラ

## 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. 8237A-03/11, 8237AJ2-01/21

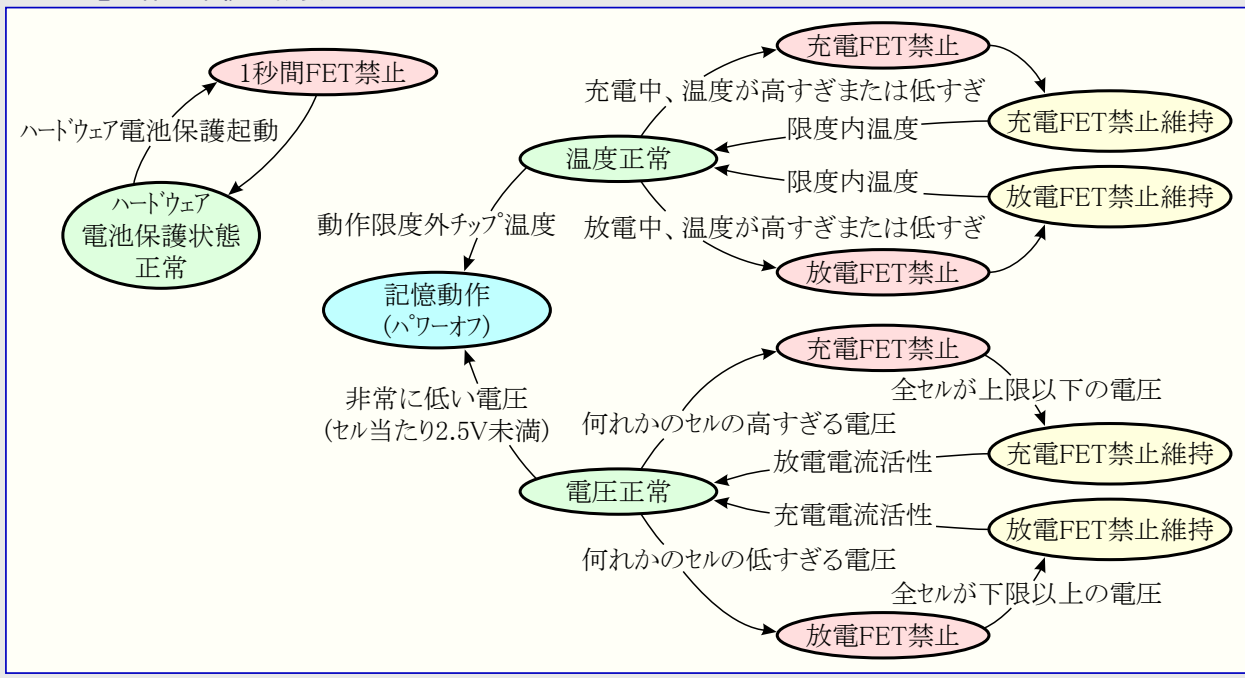
図2-1. Atmel SB2027ファームウェアがどう動くのかの概要



### 3. 電池の安全性

電池が使用者に対して安全に危険がないのを保証することはAtmelのATmega16HVB/32HVBに対する最も重要な作業です。リチウムイオン電池は過充電や多過ぎる電流が引き出される場合に非常に熱く、そして爆発さえ有り得、故に商品としての電池に対して良好な保護が必要とされます。

図3-1. 電池保全システムの概要



#### 3.1. 電池保護ハードウェア単位部

AtmelのATmega16HVB/32HVBのハードウェア電池保護は電池で大きすぎる電流が流れ込むまたは流れ出す時に両方のFETを禁止します。これは放電に関して3つ、充電に関して2つの限度を持ちます。限度は異なる電流と反応時間に設定することができます。これは非常に速い回路短絡検出を許しますが、ホスト/応用への供給断なしに低い充放電スパイクをも許します。表3-1.は3つの放電限度がどう設定し得るかの例を示します。

この実装では試験中のより速い放電を許すために、(放電高電流保護と呼ばれる)最低放電電流制限が禁止されます。このパラメータはEEPROM内に格納され、始動で単位部の構成設定に使われます。

一連で複数の充電過/高充電保護は機能不全の充電器を示します。その場合には充電FETが禁止され、スタンバイよりも高い放電電流が検出されるまで再び許可されません。このように、例えば充電器から切断されて再び接続されても、電池は保護されます。

表3-1. 電池保護用のタイミング例

電流超過	放電FET禁止反応時間
2A	20ms
3A	2ms
6A	125µs

## 3.2. ファームウェアに基づく電池保護

ファームウェアに基づく電池の動作状況の監視が過電圧、過小電圧、危険な温度から電池を保護します。

### 3.2.1. 電流

高い充放電電流によって電池寿命が減らされるため、電池の入出力電流の流れは電池特性データと実際の応用の筋書きに従って決められ得る妥当なレベルに制限されるべきです。これは単純な閾値よりもっと凝った電流制限の仕組み/方法の使用も望まれるかもしれませんが、これは実装が最終使用者に任せられます。

### 3.2.2. 温度

非常に高いまたは低い温度は電池の寿命を減らし、更に、高い温度は電池が損傷される場合でも有り得ます。従って温度が或る温度範囲外の時は充放電を禁止することが一般的な慣例です。これは一般的な安全と電池寿命の両方にとって良い考えです。本実装は(活動時間と電力消費を最小にするために)4秒毎に温度を検査します。これは急速な温度変化が予測されないの、殆どの電池に対して充分と考えられます。

既定では、組み立てに於いてサーミスタがAtmelのSB202上に半田付けされていないので、セル温度監視は禁止されています。けれどもサーミスタはSB202キットと共に提供されます。より多くの情報については11.3.項をご覧ください。代わりにATmega16HVB/32HVBの内部温度基準がセル温度と同じであると仮定されます。

温度が高すぎまたは低すぎの場合、電流が流れている方向に依存して、充電か放電のどちらかのFETが禁止されます。それは温度が許された範囲内になるまで再び許可されません。

ATmega16HVB/32HVBの内部温度が動作限度(-20~80°C)外の場合、ATmega16HVB/32HVBは自身をOFFにし、そしてFETが自動的に禁止されます。

電池に関する温度制限の全てがEEPROMに格納されます。

### 3.2.3. 電圧

電圧を検査することによって電池は過充電と過放電から自身を保護することができます。これら両方の状態が電池寿命を減らし、安全性を危険に晒します。従って、充電電流が流れる時にセルは過電圧に関して継続的に検査され、一方同様に放電時、それらは過小電圧に関して検査されます。

電圧が高すぎる時は充電FETが禁止されます。それはスタンバイよりも高い放電電流が流れるまで再び許可されません。保護は放電FETに対して同じように働き、過小電圧状態後、放電FETは充電電流が検出されるまで再び許可されません。

放電FETが禁止されている時に、その本体ダイオードが充電方向に導きます。けれどもダイオードは充電器と電池層間の電圧差を減らす電圧低下を引き起こし、それはまた減らされた充電電流にもさせます。電池電圧が高い(約3.55V以上)なら、充電器が標準の4.2Vで充電している時に実質的に充電電流は全く流れないでしょう。

この状況は電池からの高電流(例えば3C)流出のためにセル電圧が低電圧限度以下に落ちた場合に起き得ます。この場合は電池の残容量が未だかなり高いでしょう。低電圧限度に当たることによって放電FETが禁止されると、電池セルは回復されて電圧が増加します。電圧は2.7Vから3.6Vに増加し得て、これは充電を制限するのに十分な高さです。

この筋書きを処理するため、電池電圧が与えられた閾値以上、または充電電流が検出(数mAで充分)された場合に放電FETが許可されます。

電圧が非常に低く落ちた(例えば、各セルに対する最小動作電圧-0.2V)場合、ATmega16HVB/32HVBは更なる電池の放電を制限するために自身をOFFにします。例えば、応用に電池を挿入した時の高い突出電流のための短い放電尖頭での中断を避けるため、電圧は電力OFFが実行される前の2秒間低くなければなりません。

電圧は以下にも使われます。

- セル平衡がAtmelのSB202上で活性にされるべきかの検査。
- 電池が完全に充電/放電された時の検査。容量計の割合に有用です。
- 深下電圧回復(DUVR)動作の禁止。電圧が十分に高い時にDUVR動作が禁止され、ファームウェアがFETを制御できます。

電圧保護に関する全ての限度はEEPROMに格納されます。

## 4. 電池のパラメータ/設定

殆どの電池パラメータは容易な形態再設定のためにEEPROMに格納されます。これらの値は"物理的な"単位、即ち、mA、分などでです。複雑な計算を最少に減らすため、AtmelのSB202は内部的に他の単位を用い、従って始動に於いてEEPROM内のいくつかのパラメータは他の単位に変換されてSRAMに格納されます。

コードの再コンパイルを必要とする或る設定の変更はEEPROM内の値だけよりもっと影響を及ぼします。最も明白なのはセル数と外部温度感知器の効力です。

全てのパラメータと設定はbattery\_pack\_parameters.h/cファイル内に置かれて注釈されます。電池特性データはそれらを使うファームウェア単位部、例えば、cc\_gas\_gaugingとvoltage\_based\_SoCに格納されます。



## 5. クロックと校正

システム クロックとしてAtmelのATmega16HVB/32HVBの内部高速RC発振器が使われます。これは公称8MHzの周波数で走行しますが、システム クロック前置分周器はSB202用に1/8でヒューズが設定され、1MHzの周波数のシステム クロックに帰着すべきです。

高速RC発振器は温度に対して変化します。けれども、これはその周波数を温度の直線的な関数として決めることができる低速RC発振器の使用で、走行時に高速RCを校正することによって避けることができます。これは信頼に足るUART通信にするために十分な精度で高速RC発振器を校正することを可能にします。校正はチップ温度が2°C変化する度毎に動きます。

低速RC周期が既知(計算済み)の時に、その周期中に1MHzで走行すべき高速RCの計数がどれ位起こるかが分かります。校正時、低速RCの8周期間に高速RCクロック数が数えられ、その数は調整が免除されるのと同じ位正確になるまで校正レジスタが補正されます。校正レジスタは大きすぎる周波数変化を避けるために一度に1段階だけ補正されます。

### 5.1. 実時間計数器

AtmelのSB202では正確な実時間計数器(RTC)が全く必要ではありません。実時間は概ねの分数が経過した時を知ることにだけ使われ、どの計算も使われません。

RTCはCCADC変換時間を通して超低電力(ULP)RC発振器によってクロック駆動されます。実際のULP RC周期は低速RCを通して計算することができるので、必要ならばRTCの精度を改善することが可能です。

## 6. (リセット後の)初期化の流れ

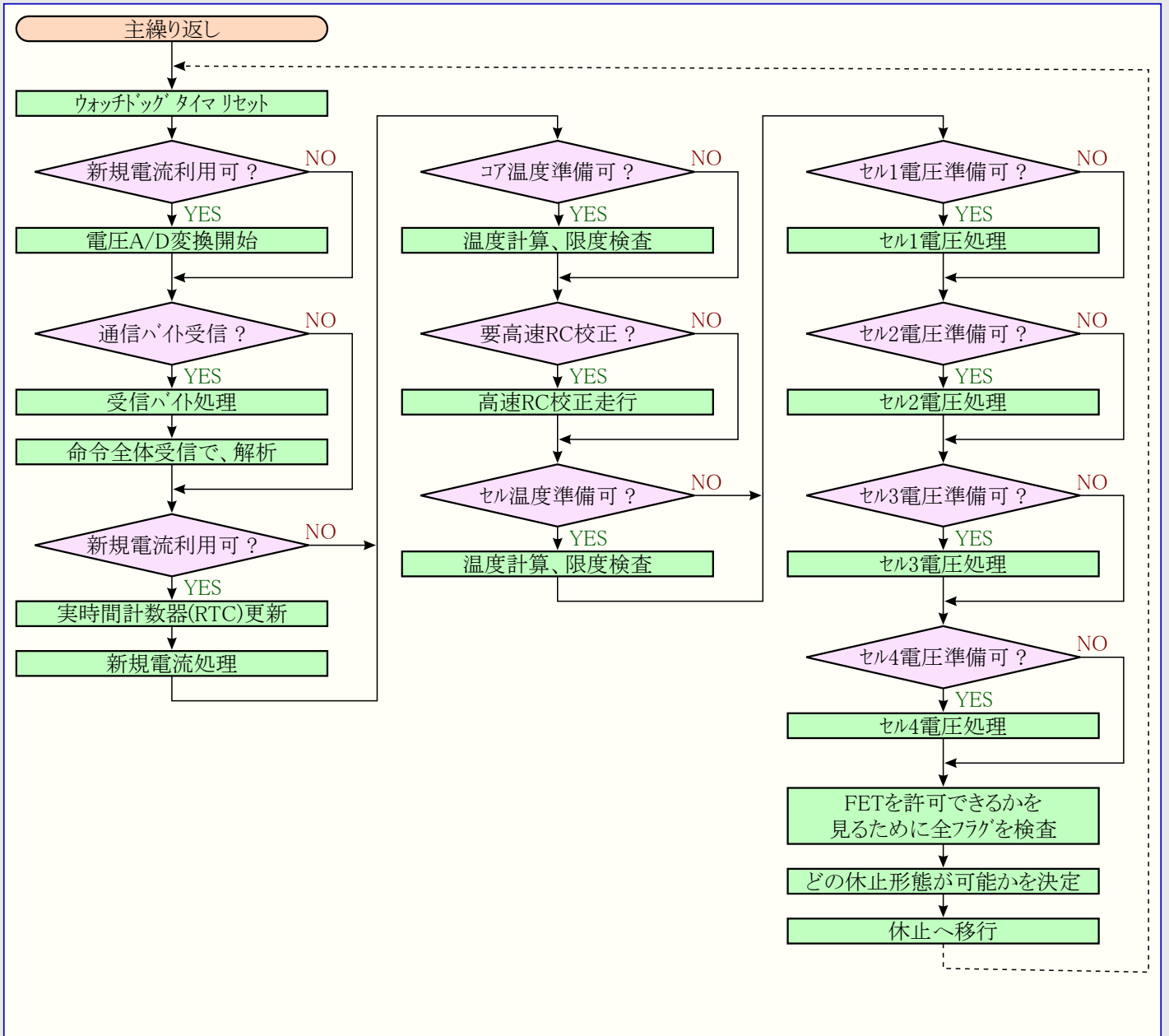
初期化の間に全ての周辺機能とファームウェアの単位部が構成設定され、校正され、そして動作の準備が成されます。正に初回の始動(即ち、プログラミング後の通電)で、CCADC変位(オフセット)補正も実行されます。初期化手順が以下で一覧にされます。

1. ウォッチドッグを構成設定して多過ぎるウォッチドッグ リセットが起こるか検査します。そうなら、それは大きな問題を示し、多分電池が禁止されるべきです(この実装では何も行われません)。
2. 電力消費を減らすために未使用の周辺機能とピンを禁止します。
3. 電池パラメータと識票列が正しい(CRC16)ことを検査します。それらがそうでなければ、デバイス是不正な値を測定し、電池保護に関して不正な限度を持ち得ます。故に検査誤りの場合に多分電池が禁止されるべきです(この実装では何も行われません)。
4. バンドギャップを校正します。失敗なら、ADC単位部からの読み込みは信頼することができず、従って電池は禁止されます。
5. ハードウェア電池保護単位部を初期化します。
6. 識票列からの利得係数と変位(オフセット)でVADC単位部を構成設定します。
7. CCADCからmA/mAhへ内部単位間を変換するための値がEEPROM内に格納された分圧抵抗から計算されます。
8. 秒毎の計算を行うのを避けるため、電池パラメータから度々 限度に使われるいくつかの内部単位に変換されます。
9. 最初の主繰り返しの間に値を持つように、VADCは利用可能な全ての入力の1走査を走行し、DUVR動作を禁止できるかも検査します。
10. 初期化処理に於いて初期に重大な誤りが起きなければ、CCADCとケロン カウンタ(CC)容量計単位部が構成設定されます。**CCに基づく容量計**の残容量が**電圧に基づく容量計**から充電の状態(SoC)を用いることによって設定されます。けれどもそれは不正確と記され、電圧に基づく容量計が正確な推測を持つと直ぐに更新されます。
11. それが正に初回の始動なら、CCADC変位(オフセット)を測定します。この処理は最大30秒かかります。
12. 最後に通信が初期化され、主繰り返しの移行する前に全体割り込みが許可されます。

## 7. 主繰り返し

主繰り返しの目的は割り込みからのデータを蓄積して反応することです。これは利用可能な全てのデータを通して走行し、全てが限度内であることを検査します。充電または放電中に異常が発見された場合、対応するFETが禁止され、異常フラグが設定されます。主繰り返しの最後で異常フラグが全く設定されていないならば、FETが再び許可されます。主繰り返しが秒当たり1度走行するため、ファームウェアが異常を検出した時にFETは最低1秒間禁止されます。

図1-7. 主繰り返しに関する簡単化した流れ図



ハードウェア電池保護単位部は(高すぎる電流によって引き起こされる)電池保護事象が起動された後の1秒以内で許可されることからFETを保護します。電流保護がハードウェアによって処理されるため、主繰り返いはFETを許可する前にそれに対する検査をしないけません。

## 8. 割り込み

この実装は割り込みに基きます。全ての処理は割り込み起動に基き、割り込み(とフラグ)はこの実装の重要部分です。この実装で使われる割り込みが表8-1.で一覧にされます。

表8-1. SB202スマート電池応用で使われる割り込み

割り込み元/関数名	説明
電圧調整器監視割り込み VREGMON_ISR()	様々な単位部に対して全ての割り込み設定を格納し、電圧調整器が再びOKになる、またはCREGが空になるまでパワーセーブ(動作形態)に移行します。電圧調整器がOKなら、割り込みを回復して動作を続けます。電圧調整監視器がどう動くかこの記述に関してはAtmelのATmega16HVB/32HVBのデータシートをご覧ください。
ウォッチドッグ計時完了割り込み WDT_Timeout_ISR()	デバイスを起こして電圧調整器がOKかを調べるために電圧調整監視器割り込みによってのみ使われます。
タイマ/カウンタ0比較一致割り込み RCCAL_InputCapture_ISR()	低速RC1周期中の高速RCクロック数を捕獲します。4頁の「5. クロックと校正」をご覧ください。
電池保護割り込み BATTPROT_BatteryProtection_ISR()	過剰電流または回路短絡のためにハードウェア電池保護単位部がFETを禁止した時に起動されます。これはホストに警告するのに使われるフラグを設定します。それが過充電だった場合、機能不全の充電器を検出し得るために起される回数を増します。
CCADC累積変換完了割り込み Ccadc_Acc_ISR()	CCADCが負極性に設定されたなら、その結果は無効にされ、さもなければそれは単にそれとして格納されます。極性切り換え補償(10.2.をご覧ください)が活性なら、ここで補償が行われます。
電圧ADC変換完了割り込み VADC_ISR()	チャンネルまたはチャンネル群が完了されたことを合図します。その後主繰り返しが結果を処理します。進行中の掃引での走査により多くのチャンネルがある場合、多重器を設定して再びVADCを開始します。
TWI(SMBus)割り込み要求 TWI_ISR()	通信を実装するためにTWI(SMBus)状態機構を保持します。
タイマ/カウンタ1溢れ割り込み T1OVF_ISR()	コード用の計時器計時を提供します。時間経過周期は2.048msです。
タイマ/カウンタ1比較B一致割り込み T1_CMPB_ISR()	押された釦の保持時間を検査。保持時間が予め定義された間隔よりも大きくなった後、LEDは容量の大きさ、または異常符号を示すために点灯します。LEDのON時間はLED_ON_TIMEOUTで設定されます。
タイマ/カウンタ1比較A一致割り込み T1CMPA_ISR()	0になるまでLEDのON時間を減らし、そしてLEDをOFFにします。
ピン変化割り込み0 PCINT0_ISR()	釦入力を調べ、釦上でピン変化がある場合に起動します。これはLEDでの状態符号を表示するための計時器を許可します。

## 9. FET制御

電流の流れを切る必要がある異常が起きた場合、FETは直ちに禁止になって異常フラグが設定されます。FETは異常が消滅、即ち、全ての異常フラグが解除されるまで再び許可されません。

フラグはコードに於いて各種変数に分けられますが、ここではフラグ名だけが使われます。

表9-1. SB202スマート電池応用で使われる割り込み

フラグ名	解除条件	説明
CriticalConditonDetected	決して解除されません。	バンドギャップまたはVADCが校正/初期化できず、何時も両FETが禁止を保つ場合に設定されます。
ChecksumFailure	決して解除されません。	識票またはEEPROM電池パラメータのどちらかのCRCが不正の場合に設定されます。この実装では行われませんが、チェックサムが不正の場合に電池は禁止されるべきです。
InDUVR	DUVR動作が禁止される時に解除されます。	始動でとDUVR動作が再び許可されたことをコードが検知した場合に設定されます。許可されることから両FETを防ぎますが、DUVRは充電FETの制御権を持ち、故に充電は可能です。
SystemIsInStandby	電流がどちらかの方向で活性(有効)な電流閾値よりも高い場合に解除されます。	電流が活性(有効)な電流閾値未満の場合に設定されます。既定でのそれは10mAです。
ReoccurringChargeProtection	活性な放電電流が検知された時に解除されます。	機能不全の充電器を示す、高すぎる充電電流が連続的に多過ぎる回数検出された場合に設定されます。設定時に充電FETを禁止して、解除されるまで再び許可されることからそれを防ぎます。
ChargingProhibited	活性な放電電流が検知された時に解除されます。	高すぎる電圧が検出された場合に設定されます。設定されている間に許可されることから充電FETを防ぎます。
DischargingProhibited	活性な充電電流が検知された時に解除されます。	低すぎる電圧が検出された場合に設定されます。設定されている間に許可されることから放電FETを防ぎます。
VoltageTooHigh	全セルに対して最後の電圧が最大以下だった場合に解除されます。	どのセルに対しても最後の電圧が高すぎる場合に設定されます。充電FETを禁止します。
VoltageTooLow	全セルに対して最後の電圧が最小以上だった場合に解除されます。	どのセルに対しても最後の電圧が低すぎる場合に設定されます。放電FETを禁止します。
CellTemperatureTooHigh	最後の温度読み取りが最大限度以下の場合に解除されます。	どれかのセル(またはサーミスタが接続されていない場合にはATmega16HVB/32HVBの内部温度基準)の温度が高すぎる場合に設定されます。設定時、どちらの方向に電流が流れているかに依存して、充電または放電のFETのどちらかを禁止します。設定されている間に許可されることから両方のFETを防ぎます。
CellTemperatureTooLow	最後の温度読み取りが最小限度以上の場合に解除されます。	どれかのセル(またはサーミスタが接続されていない場合にはATmega16HVB/32HVBの内部温度基準)の温度が低すぎる場合に設定されます。設定時、どちらの方向に電流が流れているかに依存して、充電または放電のFETのどちらかを禁止します。設定されている間に許可されることから両方のFETを防ぎます。



## 10. CCADC

電流レベルを検査して(メモリを節約するために概算の)1分に渡る平均電流が計算され、そして電流の流入と流出の累積充電を計算するのにCCADCからの結果が用いられます。CCADCからの累積された結果は限度検査と充電を累積する時にだけ使われます。CCADCは1秒の変換時間と変換完了時に割り込みを起動するように設定されます。そのようにしてこれは実時間計数器(RTC)を更新するのもに使われ得ます。実際のタイミングは温度に依存しますが、4頁の「5. クロックと校正」で記述されるようにRTCは非常に正確な精度を必要としません。

実際の変換時間は電池を通して流れる充電を計数するために電流を累積する時に重要です。CCADCは低速RC発振器によってクロック駆動され、その周期はAtmelのATmega16HVB/32HVBの内部温度基準と識別列に格納されている低速RC温度予測値から計算できます。`cc_gas_gauging`単位部は容量を累積して実際の低速RC周期に対して補償する責任があります。

極性切り換えが許可されるなら、CCADC累積変換完了割り込みは常に充電電流に対して正と放電電流に対して負になるように結果を保存する責任があります。

CCADCからの新しい電流結果が準備可の時に、それは主繰返しによって**battery\_current\_monitoring**単位部に渡され、そこでは1分平均電流に加算されます。平均は**式10-1. 平均電流計算法**で示されるように計算されます。精度を改善するために尺度調整が用いられ、平均が返される時に(尺度調整が)取り去られます。1分移動平均電流を見積もるのに指数濾波器が使われることに注意してください。これは1分移動平均が非常にSRAMを消費するので、メモリを節約するために行われます。

`battery_current_monitoring`はそれを必要とする他の単位部用に電流と変位(オフセット)校正された電流への容易なアクセスも与えます。

### 式10-1. 平均電流計算法

$$\text{新平均値} = \frac{124 \times \text{旧平均値} - 4 \times \text{電流値} \times 2^n}{128}$$

$$n = \text{AVERAGE\_CURRENT\_SCALING (既定値=4)}$$

### 10.1. CCADC変位(オフセット)

CCADCは或る変位(オフセット)を持ちます。充電累積時、一定間隔で極性を切り替えることによって打ち消すことができます。しかし、変位を測定することが可能なので、それが瞬間的な電流として使われる時にその測定を修正するのにそれを使うことは良いことです。電池がスタンバイ動作形態か否かのどちらかかを検査する時の低電流に於いて最大の助けです。

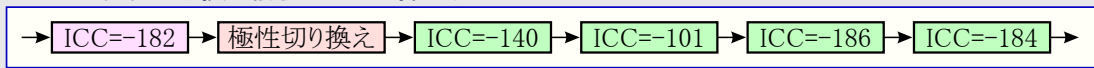
変位は電池が開始される初回に測定され、20~30秒かかり、その間中、両FETが禁止されます。切り替え後の初回採取を破棄して正と負のCCADC極性間の他の採取の平均を比較します。その差の半分が変位で、もっと正確な値を得るために後の測定に対してその後に(極性設定に依存して)加算または減算することができます。

極性切り換え後の最初の変換累積が正しくないため、多過ぎる頻度での極性切り換えは良い考えではありません。

### 10.2. 極性切り換えでの挑戦

CCADCからのA/D変換は(1秒変換時間)ハードウェアによって256回の瞬時変換の平均から計算されます。しかし、極性が切り替えられる毎の最初の2つまたは3つの瞬時電流変換は不正です。最初の1つは完全に外れており、2つ目はかなり不正で、そして3つ目は殆ど常に正しく、そうでなければ僅かに誤っているだけです。

図10-1. 極性切り換え前後でのICC採取例



これは安定に2または3のICC採取を必要とするCCADCの安定時間によって引き起こされます。図10-1の例に於ける1秒変換時間と-185の定常電流に関する誤差は以下のとおりです。

$$\text{誤差} = \frac{254 \times 185 + (-140 + 101) - (256 \times 185)}{256 \times 185} \% = -0.86\%$$

可能な限り多くの問題を取り除くために、切り換え後の2つまたは3つの採取はその秒に対する累積(ACC)結果から取り除かれて、それらの前後の瞬時(ICC)の平均で置換されるべきです。

それを行うには、ICCの結果を読むためにデバイスは極性切り換え後の3または4延長時間で起き上がらなければならない、それは活動時間、従って電力消費を追加します。また複雑さとコード量も追加します。

AtmelのSB202は誤差の殆どを更に取り去る、より単純な近似手法を取ります。切り換え後の2つの採取の和は1つのICC採取と比べて小さく、故にそれらは既に“取り去られた”と見做されます。そしてそれらの周辺にICC読み取りの平均を追加する代わりに、極性切り換えの直前からのICC採取だけが使われます。その採取は極性切り換え時に利用可能なので、実際的に全く走行時間を追加しません。

最終誤差計算として同じ仮定での誤差は今や以下ようになります。

$$\text{誤差} = \frac{(254 \times 185 + (-140 + 101) + 2 \times 182) - (256 \times 185)}{256 \times 185} \% = 0.095\%$$

注: 図10-1.に於いて結果は極性補償されず、故に切り換え前での結果は誤差計算でそれが使われる前に符号反転されます。



累積(ACC)の結果が異なる変換時間に対しても同じ範囲を持つため、結果はハードウェアによって縮尺されます。従って瞬時(ICC)の結果もそれがACCの結果に加算される前に縮尺されなければなりません。

異なる採取数と表10-1.からの縮尺に関するACCの結果は以下の通りです。

$$ACC = \left( \sum_{n=1}^{\text{採取数}} ICC[n] \right) \gg \text{縮尺}$$

2つの“失われた”ICC採取を置換することのため、切り換え直前でのICC採取はそれがACCの結果に加算される前に(2つ分で2倍され、そして3ビット分縮尺なので)4で割られなければなりません。

表10-1. 異なるACC変換時間に対するICC採取

変換時間	採取数	縮尺(ビット数)
1s	256	3
500ms	128	2
250ms	64	1
125ms	32	0

### 10.3. 分路校正

CCADCからの結果はAtmelのSB202上で $5m\Omega \pm 1\%$ の値を持つ外部分圧抵抗上の電圧低下です。この分圧抵抗は電流を計算するためにCCADCからの結果と共に使われます。校正任意選択のため、分圧値は通信インターフェース経由で新しい分圧抵抗値を送ることによって走行時に変更することができます。

電池の正確な流入/流出電流を測定するのに電流計を使うことができます。この情報はその後に分圧抵抗が実際にどれ位かを計算するのに使うことができます。これはAtmelのAVR491応用記述でもっと詳細に記述されます。

分圧抵抗は**分圧校正(Shunt Calibration)命令**を用いて4000~16000 $\mu\Omega$ 間に設定することができます。他の値も可能ですが、計算溢れを避けるためにコードの変更が必要です。

## 11. VADC

AtmelのATmega16HVB/32HVB MCUはセル電圧、チップ温度、セル/電池温度を測定するために5チャンネルのVADCを持ちます。チップ温度は内部ダイオードを使います。電池のセル温度を測定するには外部サーミスタが必要とされます。

CCADCの累積(ACC)の結果が準備可の後直ぐに、新しいVADC走査が構成設定されて開始されます。これは(VADC変換がデータ処理と比較して長い時間かかるため)主繰り返しがそれを処理し得る時にデータが準備されていることを保証するためです。CPUが情報処理で多忙な間にだけVADCが走行するのは理想的でしょう。さもなければ雑音低減動作形態/アイドル動作形態休止または活動動作形態で更にエネルギーが浪費されます。変換の完了を単に待つだけでなく、実際に変換の完了を待つことが必要とされる時は必ず雑音低減休止動作形態が使われます。

VADC走査を構成設定することは採取されるべきチャンネルを設定することを意味します。それはどれ位の数のセルとサーミスタが使われるのかに依存します。セル電圧は毎秒採取されますが、温度はそれらが早く変化しないので4秒毎に採取されます。ADC入力チャンネルはそれらが主繰り返しで処理されるのと同じ順番で走査されます。これは次の試料を変換している間にCPUがデータを処理できることを保証するためです。

1つのチャンネルが終了されるとVADC変換完了割り込みルーチンが主繰り返しに通知するためのデータ準備可フラグを設定します。これは例えば、VADCがセル2電圧を採取している間にセル1からの結果を処理することを主繰り返しに許します。割り込みルーチンは走査での次の入力チャンネル(VADC多重器の形態再設定)に進めるのか、またはADC採取をOFFにします。

### 11.1. VADC校正

VADC使用時に最適な精度を保証するため、全ての結果は変位(オフセット)と利得の誤差に対して修正されなければなりません。変位と利得の補償(値)はATmega16HVB/32HVBの識票列から読むことができます。より多くの詳細についてはデータシートを参照してください。

#### 式11-1. VADC結果の変位と利得の補償

$$\text{補償済みADC} = (\text{ADC結果} - \text{ADCOffset}) \times \text{ADCgain} \quad : (2^{14} \text{による縮尺})$$

### 11.2. チップ温度

チップ温度を計算するために、VADCの結果はデータシートで記述されるように識票列からのVPTAT値で乗算されて縮尺されます。更にそれはSBS仕様に従って必要とされる $0.1^\circ\text{K}$ での結果を得るために10倍されます。

温度は $0.1^\circ\text{K}$ (の単位)で格納されます。

**注:** 内部温度基準に対するVADCの結果は利得補償だけが必要です。変位(オフセット)補償は必要とされません。

### 11.3. セル温度

組み立て中にサーミスタがAtmelのSB202上に半田付けされないで、セル温度読み込み(と処理)は既定で禁止されています。代わりにシステム温度を評価する時にチップ温度がセル温度と同じと仮定されます。

1つまたは2つのサーミスタを使うには、AtmelのAVR455応用記述で記述されるようにそれらを接続し、`battery_pack_parameters.h`内の2つの定義を変更してください(`CELLTEMPERATURE_INPUT`を探してその直ぐ上の注釈を読んでください)。

VADCの結果をmVに変換するには、結果が変位(オフセット)校正され、利得係数で乗算され、その後縮尺されなければなりません。縮尺は最大精度の選択で、故に既定でのVADC単位部はmVで200×電圧を返します。

#### 式11-2. データシートに従ったVADCでの電圧計算法

$$ADC(mV) = \frac{1}{10} \times \frac{(ADC結果 - ADCOffset) \times ADCgain}{16384}$$

#### 式11-3. AtmelのSB202でのVADC結果からの縮尺電圧計算法

$$ADC(縮尺mV) = \frac{(ADC結果 - ADCOffset) \times ADCgain}{8192}$$

SB202で供給されるサーミスタは三菱のRH16-3h103fで、`ntc_rh103f.c`ファイル内のNTCファームウェア単位部がVADC単位部からの縮尺された電圧を摂氏または華氏の温度に変換します。別のNTCを使うには、ファームウェア単位部によって使われる他のデータが使われなければなりません。

### 11.4. セル電圧

VADCの結果から現実の電圧を得るには、データシートで書かれているように、変位(オフセット)が加算され、結果が利得係数で乗算され、そして縮尺されなければなりません。VADCの結果と実際の電圧間のこの変換は変換毎に1回の計算だけで、従って後のセル電圧の使用ではVADCの結果を電圧(V)に変換する時間を費やしません。

## 12. 容量計

容量計は容量がどれ位かと走行時に電池にどれ位残りがあのかを推測します。AtmelのATmega16HVB/32HVBは非常に正確なVADCとCCADCによって高い精度の容量計を達成することができます。フラッシュメモリ、SRAM、EEPROMの大きな記憶容量は正確な電池モードの使用を許します。

AtmelのSB202は電圧とクーロンカウンタの組み合わせに基く容量計を使います。電圧に基く容量計が残容量の初期化に用いられ、そしてその後の使用中にクーロンカウンタがそれを増減します。

### 12.1. SB202の容量計戦略

始動に於いて、電圧に基く容量計は電池の内部抵抗の近似を用いて充電状況(SoC)を推測します。残容量はそのSoCとEEPROMに格納された最後の満充電容量から計算されます。これは開回路電圧(OCV)が測定された時または電池が満容量に充電された時ほど推測が正確ではありませんが、SoCと残容量の妥当な初期推測を提供します。この理由のため、SoC/残容量が“不正確”で、可能な限り早くより良い推測が実行されるべきであることを示すためにフラグが使われます。

電圧に基く容量計単位部からのSoCは電流が30分またはそれ以上の間スタンバイ電流限度以下の時に正確と見做されます(スタンバイ電流の閾値はファームウェアで指定され、EEPROMに格納されます)。初回、これは始動後に起き、残容量が更新され、もはや不正確として合図されなくなります。

残容量はクーロンカウンタによって毎秒更新され、SB202がホスト/応用に提供できる全ての情報が計算に使われます。

電池から高電流が流出する時により早く放電終期電圧に達するため、空へ向かう走行時の計算は現在の水準を考慮に入れます。非常に小さな電流だけを費やす時に比べて各種電流水準で利用可能な容量変化を決めるために、この計算は指定電池セル形式用の特性付けデータを使います。理想的には電池特性が時間経過で僅かに変化するので、電池が古くなる時に特性データが更新されるべきですが、本応用でこれは提供されません。

### 12.2. 電圧に基く容量計

電圧に基く容量計は開回路電圧(OCV、即ち、電流が低いまたは全くない)からのSoCを決めるため、かなり上手く動きます。しかし“もっと電流が流れると直ぐに、電池の内部抵抗が端子電圧の減少をもたらします。この抵抗は温度、充電状況(SoC)、電池の使用時間で変わり、これはOCVだけからSoCを正確に推測することを挑戦的にします。

電圧に基く容量計では他に次のような2つの問題があります。

- %で電池の状況を与えるだけで電池の容量の指示が全くなければ、電池に関する正確な動作時間を与えることを難しくします。
- 電流停止時、電圧は直ぐにOCVに戻りません。どれ位長くかかるかは電池に依存しますが、一般的に約30分かかります。

けれども、OCVに基く推測とクーロンカウンタに基く推測の組み合わせは温度や経年などにあまり影響を受けない全体的に良好な推測を提供します。

### 12.3. クーロン カウンタ容量計

クーロン カウンタ近似は電池への充電とそれからの放電の両方でどれ位の電流が流れたかの経緯を保持します。それは高電流でさえ非常に正確です。クーロン カウンタに基づく容量計の不利な点は変化だけを測定できることです。これは絶対的な充電水準についての情報を提供せず、従って初期充電状況も必要とされます。

クーロン カウンタ容量計を行う時は、始動とスタンバイでの長時間後に於いても可能な時に電圧に基づく近似を用いて充電電流/残容量が推測されなければなりません。初期電圧に基づく推測が不正なら、クーロン カウンタに基づく推測も不正になります。けれども完全放電と満充電の状態に達した時に自身を校正することができます。

クーロン カウンタを使うことにより、電池が古くなった時に以下の方法で満充電容量を更新することができます。

それは電池が満充電されると完全に放電される時を知ることで、故にそれらの状態の間にどれ位の充電量が使われたかが満充電容量です。リチウム イオンセルでの満充電は充電電流が充分低く落ちた、度々10~15mAの範囲に下がる時に達成されます。内部抵抗のために高い電流で放電する時に電池が実際に空になる前に放電の最後(EoD)の電圧に達するので、完全放電は容量器に対して少しごまかしです。故にEoDは完全放電になるように充分低い電流で到達されなければなりません。

残充電量と走行時での空を推測しようとする時に、ホスト/応用が停止するため、代表的に電池が決して完全放電にならないことの実事が挑戦を意味します。

## 13. 電池特性付けデータ

SB202は充電状況(SoC)対開回路電圧(OCV)用の表が必要で、満充電容量が高い電流でどれ位容量が減るのかの表が利用可能で任意選択ですが、推奨されます。

SB200はSB202から入手可能な全データを記録するためにPC実演ソフトウェアと共に使うことができ、従って電池を特性付けするための基盤として使うことができます。

### 13.1. 満充電容量

電池の満充電容量は度々設計容量と異なります。電池はそれが正しくないと判断された場合に満充電容量を更新します。

設計容量だけでなく満充電容量も非常に低い、即ち約10mAの電流で利用可能な容量であるべきです。

このパラメータは**battery\_pack\_parametes**単位部に於いてEEPROMに格納されます。

### 13.2. 充電の状態と電圧

電圧に基づく容量計は0%~100%の各種充電状況(SoC)に於ける開回路電圧(OCV)の表を使います。SB202は表内の2つの最近点間を直線補間し、故にSoCは一定の間隔でなければなりません。これは殆どの電池に関してOCVが殆ど直線である70%~30%の間でより少ない値を、そして電圧がもっと多く変化する放電の最後近くでもっと値を持つことを許します。

**注:** SB202は正確に100%と0%の値を持つことが必要です。

表は**voltage\_bases\_SoC**単位部に格納されます。

### 13.3. 残容量補償

電池の内部抵抗とそれが放電の最後に向かって増加するという事実のために、より高い電流で容量と等しく電池の出力を得ることは不可能です。従ってSB202は電池がどれだけ長く負荷電流を維持するのか知るために、各種電流で利用可能な容量がどれ位減るかを知らなければならない。

SB202は残容量補償表に於ける任意の電流数を支援し、2つの最近点間で補間します。電流が表の最高値よりも高い場合、2つの最高値を用いて補償値を予想します。

電池の内部抵抗は温度と共に変化しますが、これはこの実装によって未だ支援されません。

## 14. セル平衡

複数セル電池パック使用時、電池パックから最大容量を得るためにセル電圧に関してそれらが平衡にされます。何故かを理解するには、リチウム イオン充電周期を理解することが必要とされます。

充電器は最大充電電流か最大充電電圧のどちらかで充電します。充電周期の始めに於いて、電池が殆ど空の時に充電電流は最大限度に達するでしょう。従って、電池が或る充電レベルの構築を始める時に、充電電圧は増加して結局電圧限度(代表的にLi-Ion セル当たり4.2V)に達し、故に充電器に対する制限要素になります。賢い充電器はこのレベル以上に増加することを電池パックの充電電圧に許さず、結果として充電電流が減少します。これら2つの段階は定電流充電と定電圧充電と呼ばれます。

問題は2つのセルを使う時で、充電器は8.4Vの電圧限度を持ちます。1つのセルが他よりも多く充電される、即ち不平衡の場合、それは他のセルよりも早く最大充電電圧に達するでしょう。この場合、殆ど充電されたセルは4.2Vに達し、一方少ない方の充電は4.1Vです。この状況では電池積層の電圧は8.3Vで、充電器は電池セルの1つがその最大充電電圧に達したことを通知しません。充電器は定電流充電を続け、そして電圧は電池パックが8.4Vになるまで増加を続けるでしょう。これは殆ど充電されたセルが4.25Vに達するかも知れず、それは安全性の理由のための絶対最大電圧で、AtmelのSB202は充電FETを禁止するでしょう。



これが起きた時はリチウムイオンセルが事実上少しの時間だけ最大充電電圧で充電されるべきなので、どちらのセルも完全に充電されません。問題が発生した場合、セル不平衡がますます悪化することを続けるのは高い危険があります。結果として、セルが高い不平衡ならば、充電器は意図したように動くことができず、電池パックはその最大容量に充電されないでしょう。

SB202によって用いられる解決策はセル間の電圧差が充分低くなるまで最も充電されたセルを放電することです。BATTPARAM\_MISBALANCE\_VOLTAGE\_THRESHOLDと呼ばれる閾値はbattery\_pack\_parameters単位部のEEPROMに格納されます。

不平衡が電池パック充電中の本来の問題のため、セル平衡は電池充電中にだけ活動します。放電中、セル平衡の活性化は制限された効果を持ちます。2つの高い不平衡セルが共に動作する場合、セルを平衡にするのに多くの充放電繰り返し回数がかかるかもしれないことに注意してください。

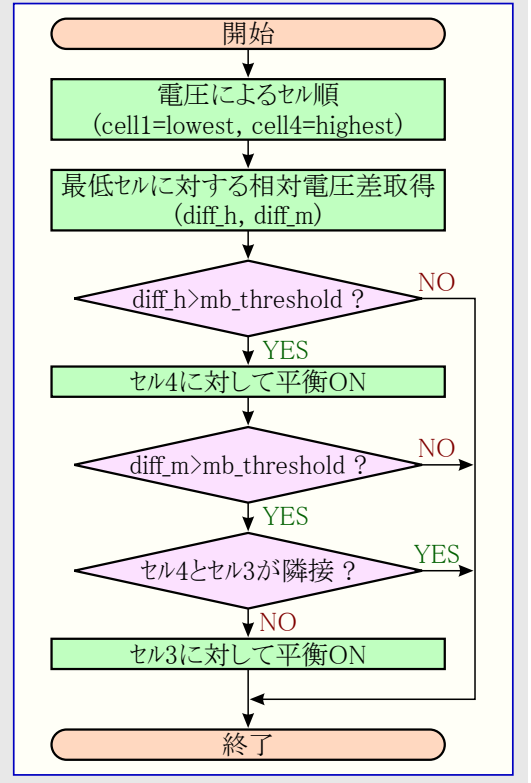
AtmelのATmega16HVB/32HVBでは平衡FETは内蔵で、外部FETは全く必要とされません。濾波する抵抗は内部FETを通して流すことができる最大平衡電流を制限します。これは平衡FETが内部で、浪費された電力がチップの温度に影響を及ぼすため、注意されるべきです。最大平衡電流に依存して、内部温度は室温で最大30°C上昇し得ます。従って、電池セル温度を監視するのに外部サーミスタ(NTC)が使われない場合、セル平衡は或るレベルで禁止されるべきです。

2セル電池パックに関して、セル平衡算法は2つのセルの最上位セルを平衡にするのと同様に簡単です。3セルと4セルの電池パックに関して、その算法はもっと複雑です。ATmega16HVB/32HVBはそれらが互いに隣接していない限り、1度に2つのセルまで平衡することができます。例えば、セル1とセル3は1度に平衡することができますが、セル1とセル2はできません。3セルと4セルの実装について、最高電圧のセルが平衡され、2番目に高いセル電圧が隣接セルでなければこれも平衡にされます。

この算法は第2のセルが最高のセルに隣接していて平衡にすることができない場合に3番目に高いセルを平衡にしません。これは最低のセルをより不平衡にするだけで、従ってそれらが最高のセルの電圧に達するまで平衡にされません。

加えて、充電効率を増加するために、この算法は充電の残りの状態が80%よりも大きい場合にだけ起こるようにセル平衡を許します。

図14-1. 4セル応用に関するセル平衡の流れ図



## 15. 電力管理

電池電力を節約するために、ATmega16HVB/32HVBは可能な限りパワーセーブ動作形態で休止します。けれども、パワーセーブ動作形態で動かない周辺機能単位部が使われるなら、パワーセーブ動作形態へ移行すべきではありません。この実装に於いて、これはVADCとタイマ/カウンタを意味します。

主繰り返しはVADCの1走査よりも速く実行される場合、主繰り返しの最後でVADCが未だ走行しているかもしれません。その場合、ATmega16HVB/32HVBはA/D変換雑音低減動作形態へ移行し、変換が完了した時のVADC割り込みによって起き上がります。

タイマ/カウンタは高速RC校正と通信によって使われます。それらが使われる時にデバイスはアイドル動作形態へだけ移行することができます。校正はファームウェアから始められるため、校正が動いている時は常に分かります。けれども、通信が活性かを判断するのは、それが外部事象によって引き起こされるので、ずっと困難です。

送信は外部割り込み要求を起動する開始ビットによって始められます。そしてメッセージ内のビットを採取するために割り込みルーチンが計時器を始動します。難題は通信が活性かを定めるのに検査した後に開始ビットが受信され得ることです。解決策は外部割り込みルーチン内で休止動作形態をアイドルに設定することです。

ATmega16HVB/32HVBは以下の条件の1つが起きた場合にパワーオフ(動作形態)へ移行します。

- チップ温度がATmega16HVB/32HVBの外側。3.2.2項をご覧ください。
- 電池電圧が非常に低い。3.2.3項をご覧ください。

## 16. 通信

AtmelのSB202はホスト/応用と通信するためにSMBusインターフェースを持ちます。ホストが通信を開始しなければならず、SB202は問われた場合にだけ通信を行います。いくつかの異常に於いて、SMBus仕様に従い、異常を検知するのにSCL線での時間超過の仕組みが使われます。

命令一式は<http://www.sbs-forum.org>からのSBS仕様1.1版です。仕様には完全に従っておらず、いくつかの命令は完全に支援されず、いくつかの予約命令が使われます。命令の一覧については17章をご覧ください。



## 16.1. SMBusインターフェース(TWI)

AtmelのATmega16HVB/32HVBはSMBus実装を支援する組み込みの最適化されたTWI制御部を持ちます。SB202は従装置送信形態と従装置受信形態のSMBus従装置動作形態だけを持ちます。

TWI逐次割り込みでは状態変更が実装されています。SB202側からの従装置アドレス一致から始まる1つの単位転送はATmega16HVB/32HVBをパワーセーブ動作形態から起こすことができます。しかし後続する同じ単位転送内の命令とデータの転送に関して、データの到着がアイドル動作形態よりも深い休止動作形態からデバイスを起こすことができないため、デバイスは活動動作形態またはアイドル動作形態を保たなければなりません。

SB202はパケット誤り検査符合(PEC:Packet Error-checking Code)を持つまたは持たない受信メッセージを受け入れます。送信時、SB202はメッセージ送信後、常にPEC送信が準備されます。ホストがPECを望まないなら、SB202に通知するために最終データバイト後にNACK信号を送ることができます。

通信速度はSMBusホストによって決められます。この速度は10kHz~100kHzの範囲にすることができます。

## 16.2. 規約

SBS仕様はSMBus規約を使います。SB202はSMBus仕様1.1に適合します。SBS仕様に従って電池パックアドレスは\$16に固定されます。

SB202は以下の4つの異なるメッセージ形式を支援します。

- ・ 語読み込み
- ・ 語書き込み
- ・ 塊読み込み
- ・ 塊書き込み

語命令は常に2バイトのデータを持ち、塊命令は常に32バイトで、これは可変長を持つSBS塊命令とは異なります。

開始条件信号後、ホストは上位側7ビットの従装置アドレスとLSBの読み/書きフラグ(0は書き込みを意味し、1は読み込みを意味します)を含むバイト(8ビット)を送ります。

全ての命令符号は1バイトで、語命令と塊命令に分けられます。全ての命令が何なのかを双方が同意しなければならないでしょう。

SB202ファームウェアはPEC実装を支援します。PECは命令と全てのデータバイトから計算される、多項式\$8DでのCRC8(CRC8-CCITT)です。

### 16.2.1. 読み込み命令

読み込み命令に於いて、ホストは最初にSB202に命令を書いて、その後に読み込みフラグで通信を再開します。その後、従装置は1つずつデータバイトをSDA線に置きます。

### 16.2.2. 書き込み命令

書き込み命令に於いて、ホストはSB202へ命令バイトとデータバイトを連続的に送ります。SB202が1バイトを受信した後、1ビットのACKまたはNACK信号で返答します。最後のデータバイト後、SB202はホストからのPECバイトを尋ねるためにACK信号を送り出します。しかしホストがPEC機能を支援しない場合、従装置がメッセージの終了を知るために停止条件(P)信号を送ります。

図16-1. 語読み出し命令

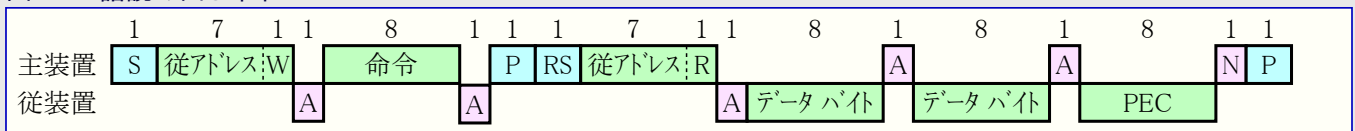


図16-2. 語書き込み命令

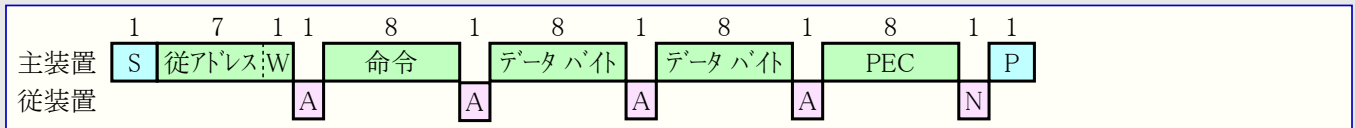


図16-3. 塊読み出し命令

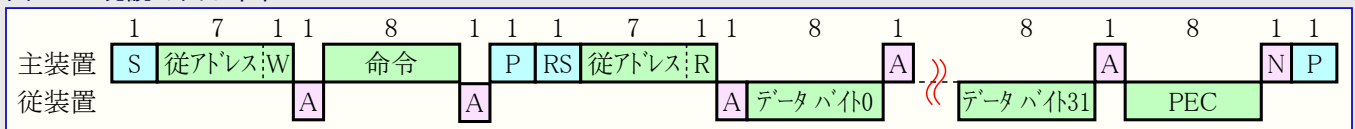
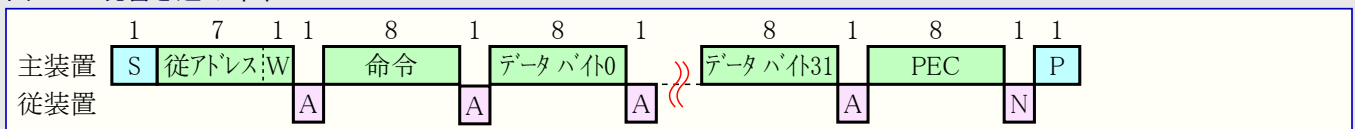


図16-4. 塊書き込み命令



### 16.2.3. 異常

通信中に異常が起きた場合、SB202は25msよりも長い間SCL線でクロックのLowを延長することができます。そしてその後にSB202はTWI線を開放してTWI制御部をリセットすることができます。異常は以下を含みます。

- SB202によって返された不正命令
- SB202から送られたデータでのフレーム異常
- SB202から送られたデータでのPEC不正

### 16.3. 認証

SB202は挑戦/応答認証に使うことができる、AES暗号化算法とSHA256に基づくHMACの両方を含みます。これはホストに接続された電池が互換型であることを保証することを許します。

両算法は同時に使うことができず、決定はAUTH\_USE\_AESとAUTH\_USE\_HMAC\_SHA2の2つの定義の使用でコンパイル時に行われなければなりません。

AESは16バイト(128ビット)の鍵と16バイトの返答量を使います。HMAC-SHA256は32バイト(256ビット)の鍵と31バイトの挑戦と返答の量を使います。AESとHMAC-SHA256を使うことは挑戦と返答の量での違いを除いて同一です。

電池とホストの両方が予め鍵を知らなければなりません。代表的に同じ型式の全ての電池は予め格納された同じ鍵を持ち、ホストは異なる電池型式に対して2つの鍵にするかもしれません。

HMAC-SHA256用の既定の鍵は“This is a long key for HMAC-SHA2”です。組み立てに於いてSB202に予めプログラミングされたAES用の鍵は“You cannot pass!”です。しかし、実際の応用は表示可能な文字だけから成らない鍵を使うべきです。

AESとHMAC-SHA256の両方は安全と見做され、数ある中でNSA(米国の国家安全保障局(National Security Agency))によって認定されています。(力付くを除いて)既知の攻撃は(2008年5月現在)どれ1つも全く知られていません。両算法はサイドチャネル攻撃に基づくタイミングを避けるために一定時間で走行します。それにも拘らず、認証の強さは、鍵がメモリにどう格納されるか、挑戦と返答がどう実行されるか、その他諸々のような、サイドチャネル攻撃に於いて搾取されるかもしれない他の要素も信頼します。非常に高い安全水準が必要とされるなら、応用開発者は良好な暗号化実算を学んでそれを適用することが奨励されます。

この応用記述で用いられる認証法は純粋な力付くからに対してかなり強力です。AESの128ビット鍵に於いて、力付くの攻撃は正しい鍵を見つけようとするのに平均 $2^{27}$ ( $1.7 \times 10^{38}$ )かかります。誰かが並列で100万( $10^6$ )台のコンピュータを用いて1年( $32.6 \times 10^6$ 秒)で鍵を破ろうとした場合、各コンピュータは以下を行うことが要求されるでしょう。

$$\frac{\text{鍵可能性数}}{\text{処理秒数} \times \text{コンピュータ台数}} = \frac{1.7 \times 10^{38}}{32.6 \times 10^6 \times 10^6} = 5.2 \times 10^{24} \text{ 試行回数/秒}$$

これはns(ナノ秒)あたりに128ビット鍵を用いて16バイトのAES暗号を概ね5200京回完了するのと等価です。故に、例え少しばかりの幸運があったとしても、力付くの攻撃に基づく中位の時間内に鍵を識別すると思いません。それでも、使われる暗号法が望む目的に対して十分に強く、そしてそれが最良の施行で使われることは設計者に任せられます。

#### 16.3.1. 認証手順

ホストが最初に電池がどの型式かを問わなければならないのは、使う鍵を了解するためです。そして電池は共用鍵とでデータに於いてAESまたはHMAC-SHA256を実行します。ホストが電池から返答を読む時に、電池が正しい鍵が使われたのを確かめることができ、それが言われた型式であることをかなり確信することができます。

段階的手順は次のとおりです。

1. ホストは**認証命令**(SBS塊書き込み命令)を電池へ送ります。この実装に於ける最初のバイトは常に\$01です(それは新しい挑戦が到着した時を知るために電池によって使われます)。その後AESに対して16バイト、HMAC-SHA256に対して31バイトの乱データが送られるべきです。
2. そして電池は受信したデータで算法を動かします。HMAC-SHA256は計算内に(データ内の先頭バイトとして)状態バイトも含み、故に返答を検査する時にホストはこれも行わなければならないことに注意してください。SHA356で穴埋め時に使われたメッセージ長が32バイトの挑戦長から計算されるべきであることも意味します。
3. ホストは変更されたデータを読むために再び認証命令を送りますが、今回はSBS塊読み込み命令としてです。先頭バイトは(表16-1で記述される)状態バイトで、残りが返答です。HMAC-SHA256に関して返答の最終バイトは32バイトの塊メッセージに合うように切断されなければなりません。
4. そしてホストは電池からのデータを照合し、算法が正しい鍵で走行したことを検査します。

表16-1. 読み込み認証状態バイト

状態バイト	説明
\$01	認証は未だ開始されていません。
\$02	認証が開始されましたが、完了されていません。
\$04	認証が完了しました。
\$08	異常が発生しました。

### 16.3.2. 乱データの重要性

ホストが認証毎に同じ挑戦を送らないことが重要です。そうした場合、電池は返答を固定符号にでき、鍵を知ることもなく、認証翻訳は互換電池が接続されているかを判断するのに役立ちません。少し違うだけのデータの組み合わせが使われる場合、同じ問題が適用されます。

この種の攻撃に対して保護するための最良の方法は各ホスト/応用に対して或る独特なデータで結ばれた擬似乱データを送ることです。独特なデータは例えば通番にすることができます。これについてのより多くの情報は暗号についてのインターネットと本で得られ、“反射攻撃”で検索してください。

## 17. 通信命令一覧

### 17.1. SBS準拠命令

以下の命令は<http://www.sbs-forum.org>で入手可能なスマート電池データ仕様改訂1.1で定義されるようにほぼ実装されます。AtmelのSB202は電流(mA)の代わりに電力(mW)でデータを使って返すことを支援せず、故に両方を支援すべき命令は電流だけを支援します。

• ManufacturerAccess	- 製造業者アクセス
• RemainingCapacityAlarm	- 残容量警告値
• RemainingTimeAlarm	- 残時間警告値
• AtRate	- 充放電速度値
• AtRateToFull	- AtRateでの満容量までの予測残り時間
• AtRateToEmpty	- AtRateでの空までの予測残り動作時間
• AtRateOK	- 10秒間、AtRateで供給可
• Voltage	- mVでの電池パック電圧
• Current	- mAでの電池電極通過(充放電)電流
• AverageCurrent	- 1分間移動平均によるmAでの電池電極通過(充放電)電流
• RelativeStateOfCharge	- 満充電容量に対する%での予測残容量
• AbsoluteStateOfCharge	- 理論容量に対する%での予測残容量
• RemainingCapacity	- 残容量
• FullChargeCapacity	- 満充電容量
• RunTimeToEmpty	- 現在の放電速度による分での予測残電池寿命
• AverageTimeToEmpty	- 1分間移動平均による分での予測残電池寿命
• AverageTimeToFull	- 1分間移動平均による満充電までの分での予測時間
• ChargingCurrent	- mAでの望む充電電流(速度)
• ChargingVoltage	- mVでの望む充電電圧
• CycleCount	- 電池が経験した充放電周回数
• DesignCapacity	- 新電池に於けるmAhでの理論容量値
• DesignVoltage	- 新電池に於けるmVでの理論電圧値
• SpecificationInfo	- 支援するスマート電池仕様版番号と電圧/電流の尺度調整
• ManufacturerDate	- セルパック製造日付
• SerialNumber	- 通番
• ManufacturerName	- 製造業者名
• DeviceName	- 電池名称
• DeviceChemistry	- 電池の化学特性
• ManufacturerData	- 電池製造業者によって決められる内容

### 17.2. 非SBS準拠命令

以降の命令の殆どはスマート電池データ仕様中存在しますが、それらはSB202に於いて僅かに異なって実装されます。いくつかの新しい命令は任意選択SBS命令または予約命令のどちらでも使われます。

#### ManufacturerAccess

\$00

語読み込みと語書き込み

この命令は何もせず、常に\$0000として読みます。

**BatteryMode**
**\$03**
**語読み込み**

SBSで記述されて定義されたビットが使われますが、(SB202が既定設定を支援しないため、)それらはそれらの既定値に設定されず、これらの変更は不可能です。

いくつかの予約ビットはセル平衡情報とFET状態に置換されています。

ビット	説明
15	CAPACITY_MODE : 0に設定 - "mAまたはmAhで報告"
14	CHARGER_MODE : 1に設定 - SB202が通信を開始できないので、"充電電圧と充電電流の広域送信禁止"
13	ALARM_MODE : 1に設定 - SB202が通信を開始できないので、"警告広域送信禁止"
12	BALANCING_CELL3 : セル平衡を保つためにセル3が放電されつつある時に設定(1)
11	BALANCING_CELL2 : セル平衡を保つためにセル2が放電されつつある時に設定(1)
10	BALANCING_CELL1 : セル平衡を保つためにセル1が放電されつつある時に設定(1)
9	PRIMARY_BATTERY : 0に設定 - "第2の役割で電池動作"
8	CHARGE_CONTROLLER_ENABLED : 0に設定 - "内部充電制御禁止"
7	CONDITION_FLAG : 0に設定 - "電池OK"
6	BALANCING_CELL4 : セル平衡を保つためにセル4が放電されつつある時に設定(1)
5	FCSRのDUVRD : FET制御/状態レジスタの深下電圧回復動作禁止ビットに設定されます。
4	FCSRのCPS : FET制御/状態レジスタの電流保護状態ビットに設定されます。
3	FCSRのDFE : FET制御/状態レジスタの放電FET許可ビットに設定されます。
2	FCSRのCFE : FET制御/状態レジスタの充電FET許可ビットに設定されます。
1	PRIMARY_BATTERY_SUPPORT : 0に設定 - 機能未支援
0	INTERNAL_CHARGE_CONTROLLER : 0に設定 - 機能未支援

**Temperature**
**\$08**
**語読み込み**

これは電池パックの温度を返します。SB202では0.1°Kでのチップ温度を返します。電池温度を得るためにTemperatureNTC1/TemperatureNTC2を使うことができます。

**MaxError**
**\$0C**

この命令は支援されません。

**RemainingCapacity**
**\$0F**
**語読み込み**

仕様に従い、これは0.2C放電電流での残容量を返します。SB202は平均電流での残容量を返します。



## BatteryStatus

\$16

## 語読み込み

この命令は予約ビットが使われ、異常符号が支援されないことを除き、基本的にSBSで指定されるように実装されます。

ビット	説明
15	OVER_CHARGED_ALARM : 充電禁止(chargingProhibited)フラグによって制御されます。
14	TERMINATE_CHARGE_ALARM : 電圧高すぎ(voltageTooHigh)フラグによって制御されます。
13	VREGMON_TRIGGERED : SBSでの予約ビット、これは最後に電池状態が読まれてから電圧調整監視器が起動された場合に設定(1)されます。
12	OVER_TEMP_ALARM : セル温度高すぎ(cellTemperatureTooHigh)フラグによって制御されます。
11	TERMINATE_DISCHARGE_ALARM : 残容量が定義された限度(SBS_TERMINATE_DISCHARGE_LIMIT)以下の場合に設定(1)され、充電が検出時に解除(0)されます。それが走行時に構成設定可能なので、代わりにREMAINING_CAPACITY_ALARMを使うことが推奨されます。
10	BATTERY_PROTECTION_TRIGGERED : 最後にこの命令が読まれてから電池保護ハードウェア単位部が禁止されたFETを持つ場合に設定(1)されます。
9	REMAINING_CAPACITY_ALARM : 残容量警告(RemainingCapacityAlarm)命令によって設定された値よりも残容量が低い場合に設定(1)されます。分毎に1回更新されます。
8	REMAINING_TIME_ALARM : 残容量警告(RemainingTimeAlarm)命令によって設定された値よりも空までの平均時間(AverageRimeToEmpty)が低い場合に設定(1)されます。分毎に1回更新されます。
7	INITIALIZED : 重大状況検出(criticalConditionDetected)フラグの反転によって制御されます。そのフラグが設定(1)される時に両FETが禁止されるため、それが充電器にも接続されている場合にだけ、ホスト/応用は重大状況検出が設定(1)されていることを検知できます。
6	DISCHARGING : 直前の秒で電流が放電だった場合に設定(1)されます。
5	FULLY_CHARGED : 電池が満充電と見做された場合に設定(1)され、スタンバイよりも高い放電電流が検出されると直ぐに解除(0)されます。
4	FULLY_DISCHARGED : 完全に容量が全くない時に設定(1)されます。より早い警告を得るためにTERMINATE_DISCHARGE_ALARMと残容量/時間警告(RemainingCapacity/TimeAlarm)機能を使うことができます。相対充電状況(RelativeStateOfCharge)が20%以上の時に解除(0)されます。
3~0	異常符号 (未使用)

## Authentication

\$24

## 塊読み込みと塊書き込み

16.3項で記述されます。

## ShuntCalibration

\$2A

## 語読み込みと語書き込み

非SBS命令。EEPROMに分圧抵抗値を設定するのに使われます。値は $\mu\Omega$ で設定され、4000~16000 $\mu\Omega$ 間の値を支援しますが、値の検査は全く行われません。この命令の読み込みは書き込み命令が成功裏に受信されたことを保証するのに使うことができます。

## FETDisable

\$2B

## 語書き込み

非SBS命令。この命令は充電と放電のFETの禁止を個別に強制するのに使うことができ、これは実演と試験の目的に有用で有り得ます。語内のビット0が1の場合、充電FETが禁止され、ビット0が0で再びこの命令が送られるまで禁止を保ちます。ビット1は放電FETに対して同じ様に働きます。

この命令がFETを許可に強制できず、それらを禁止に強制することだけができることに注意してください。

**StorageMode**
**\$2C**

語書き込み

非SBS命令。この命令はATmega16HVB/32HVBにパワーオフ動作形態移行を強制します。実際にパワーオフが望まれたことを確実にするために、送信語は\$FADEでなければならず、さもなければこの命令は何もしないでしょう。

**TemperatureNTC2**
**\$2D**

語読み込み

非SBS命令。第2のNTCが接続されているなら、これは0.1°Kでその温度を返します。NTCが接続されていない場合は0°Kが返されます。

**TemperatureNTC1**
**\$2E**

語読み込み

非SBS命令。第1のNTCが接続されているなら、これは0.1°Kでその温度を返します。NTCが接続されていない場合は0°Kが返されます。

**VoltageCell4**
**\$3C**

語読み込み

セル4上の電圧をmVで返します。3つのセルだけが使われる場合は0を返します。

**VoltageCell3**
**\$3D**

語読み込み

セル3上の電圧をmVで返します。2つのセルだけが使われる場合は0を返します。

**VoltageCell2**
**\$3E**

語読み込み

セル2上の電圧をmVで返します。

**VoltageCell1**
**\$3F**

語読み込み

セル1上の電圧をmVで返します。

## 18. 目次

要点	1
1. 序説	1
2. 概要	1
3. 電池の安全性	2
3.1. 電池保護ハードウェア単位部	2
3.2. ファームウェアに基づく電池保護	3
3.2.1. 電流	3
3.2.2. 温度	3
3.2.3. 電圧	3
4. 電池のパラメータ/設定	3
5. クロックと校正	4
5.1. 実時間計数器	4
6. (リセット後の)初期化の流れ	4
7. 主繰り返し	5
8. 割り込み	6
9. FET制御	7
10. CCADC	8
10.1. CCADC変位(オフセット)	8
10.2. 極性切り換えでの挑戦	8
10.3. 分路校正	8
11. VADC	9
11.1. VADC校正	9
11.2. チップ温度	9
11.3. セル温度	10
11.4. セル電圧	10
12. 容量計	10
12.1. SB202の容量計戦略	10
12.2. 電圧に基づく容量計	10
12.3. ケーロン カウンタ容量計	11
13. 電池特性付けデータ	11
13.1. 満充電容量	11
13.2. 充電の状態と電圧	11
13.3. 残容量補償	11
14. セル平衡	11
15. 電力管理	12
16. 通信	12
16.1. SMBusインターフェース(TWI)	13
16.2. 規約	13
16.2.1. 読み込み命令	13
16.2.2. 書き込み命令	13
16.2.3. 異常	14
16.3. 認証	14
16.3.1. 認証手順	14
16.3.2. 乱データの重要性	15
17. 通信命令一覧	15
17.1. SBS準拠命令	15
17.2. 非SBS準拠命令	15
18. 目次	19



#### *Atmel Corporation*

2325 Orchard Parkway  
San Jose, CA 95131  
USA  
TEL (+1)(408) 441-0311  
FAX (+1)(408) 487-2600  
[www.atmel.com](http://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*

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)(3)-3523-3551  
FAX (+81)(3)-3523-7581

© 2011 Atmel Corporation. 不許複製 / 改訂CORP072610

Atmel<sup>®</sup>、ロゴとそれらの組み合わせ、それとAVR<sup>®</sup>、AVRロゴその他は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 2021.

本応用記述はAtmelのAVR474応用記述(doc8237.pdf Rev.8237A-03/11)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。