

AVR458 : ATAVRBC100でのリチウム イオン電池の充電

要点

- リチウム イオン電池充電用の完全な機能設計
- 10ビットA/D変換器での高精度測定
- 単位部品化した“C”ソースコード
- 容易に調整可能な電池と充電のパラメータ
- 外部主装置との通信用直列インターフェース
- 電池EEPROMとの通信用単線インターフェース
- 電池IDと温度の読み込み用アナログ入力
- 強化した温度管理用の内部温度感知器
- 電池の記憶と走行時パラメータ用のチップ上EEPROM



8ビット **AVR**[®]
マイクロコントローラ

応用記述

1. 序説

この応用記述はATAVRBC100電池充電器参照基準設計(BC100)に基き、リチウム イオン(Li-Ion)電池を充電するのに参照設計をどう使用するのかに集中します。ファームウェアは(IAR[®] SystemsのEmbedded Workbenchを用いて)全体的にC言語で書かれ、他のAVR[®]マイクロ コントローラへの移設が簡単です。

この応用はATtiny861マイクロ コントローラに基きますが、ピン互換デバイスのATtiny261やATtiny461のような他のAVRマイクロ コントローラへ設計を移植することが可能です。ATtiny25/45/85のような少ピン数のデバイスも使用できますが、機能的に減らされています。

2. 動作の理屈

電池の充電は化学系に於いてエネルギーを再格納する可逆化学反応によって可能にされます。使用される化学物質に依存して、電池は或る特性を持ちます。電池に障害を与えるのを避けるために、これらの詳細な知識が必要とされます。

2.1. Li-Ion電池の技術

リチウム イオン(Li-Ion)電池は最近の再充電可能な電池に於いて最高のエネルギー/重量比とエネルギー/容量比を持ちます(16頁の「参照」の1をご覧ください)。これは現在、ノートブック コンピュータ、携帯電話、携帯メディアプレーヤー、パーソナル デジタル アシスタント(PDA)、電動ツール、医療装置のような最終応用での市場に於いて最速で成長している電池システムです。

従来の再充電可能な電池と比べ、Li-Ion電池は低い内部抵抗、高い繰り返し充電寿命、速い充電時間、低い自己放電、低い毒性を持ち、保守の必要がありません。例えば、コバルト陰極のリチウム イオンセルはニッケルに基く電池の2倍、鉛蓄電池の4倍のエネルギーを保持します(16頁の「参照」の2をご覧ください)。リチウム イオンは楽な保守体系で、他の殆どの化学特性が主張できない利点です。リチウム イオンではメモリ効果が全くなく、電池はその寿命を延ばすための計画的な充放電周期が必要ありません。リチウム イオンは低い自己放電で、環境に優しい電池です。処分は最小の害しか起しません。

Li-Ion電池の欠点は低い過充電許容力と組み込み保護回路の必要性を含みます。電気的な短絡は大きな電流の流れ、温度上昇、可燃性ガスが放出される熱暴走に帰着し得ます。

2.1.1. 安全性

リチウム イオン電池は安全で、充放電時に対応する或る予防処置を提供されます。加えて、電池製造業者は次のような3つ層の保護を追加することによって高水準の信頼性を保証します。

1. 活性材料の量はエネルギー密度と安全性の可動の平衡を達成するように制限されます。
2. 様々な安全機構が各セル内に含まれます。
3. 電気的な保護回路が電子パックの内側に追加されます。

セル保護装置は次のように働きます。

- PTC(正温度係数)デバイスは高電流波動を禁止するための保護として働きます。
- CID(回路中断デバイス)は極端に高い充電電圧が内部セル圧を上げる場合に電気的な経路を開き(切り)ます。
- 安全弁はセル圧での急速な増加の事態に於いて制御されたガス開放を許します。

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. 8080B-08/08, 8080BJ3-11/13



電氣的な保護回路は次のように動きます。

- ・ 半導体スイッチはどれかのセルの充電電圧が与えられた限度に達した場合に開かれます。
 - ・ ヒューズはセルの表面温度が概ね90°C(194°F)に接近した場合に電流の流れを切断します。
 - ・ 電流経路はセル電圧が与えられた閾値以下に落ちた場合に切断します。これは過放電から電池を保護するためです。
- 今日、毎年製造される数億を超えるセルで利用可能で安全な電池化学特性で、リチウムイオンは最も成功したものの1つです。

2.2. Li-Ion電池の充電

本質的にリチウムに基く電池を充電するのに1つの方法だけがあります(「参照」の3.)。リチウムイオンセル製造業者は充電手順に於いて非常に厳密な指針を持ち、電池パックは製造業者の“代表的な”充電技術によって充電されるべきです。

日本に於いては各種温度範囲に対して最大電圧と最大電流を定義する適切な規定もあります。これは2.2.3.副項でもっと詳細に説明されます。

Li-Ion電池は充電処理の初期の段階に於ける過熱を避けるため、電流制限された定電圧を用いて充電されます。充電は充電電流が製造業者によって設定された閾値以下に落ちた時に終了されます。電池は過熱から損傷を受けて過充電された場合に爆発するかもしれません。

2.2.1. 安全性

静電気や不完全な充電器は電池の保護回路を破壊して半導体スイッチを恒久的なON位置にするかもしれません。これは使用者が知らない間に起きるかもしれません。不完全な保護回路を持つ電池は正常に機能するかもしれませんが、誤用に対して保護を提供しません。

顧客品質のリチウムイオン電池は0°C(32°F)以下で充電することができません。低温で充電した場合、電池パックは正常に充電されると思われるかもしれませんが、セル内側の化学反応が恒久的な障害を引き起こすかもしれず、パックの安全性を危険に晒し得ます。

電池は衝撃、押し潰し、または高い頻度での充電を受けさせられた場合、より損傷を受け易くなります。

電池は冷たいままでなければなりません。充電中に熱を帯びた電池パックは使用されるべきではありません。

2.2.2. 初回充電と充電間隔

他の多くの再充電可能な電池形式と異なり、リチウムイオン電池は初回充電を必要としません。Li-Ion電池の初回充電は10回目や100回目の充電と何も変わりません。

リチウムイオン電池は度々充電されるかもしれず、そしてされるべきです。電池は完全(最大)放電よりもむしろ部分的な放電の方がより長く持ち堪えます。消耗のために完全放電は避けられるべきです。

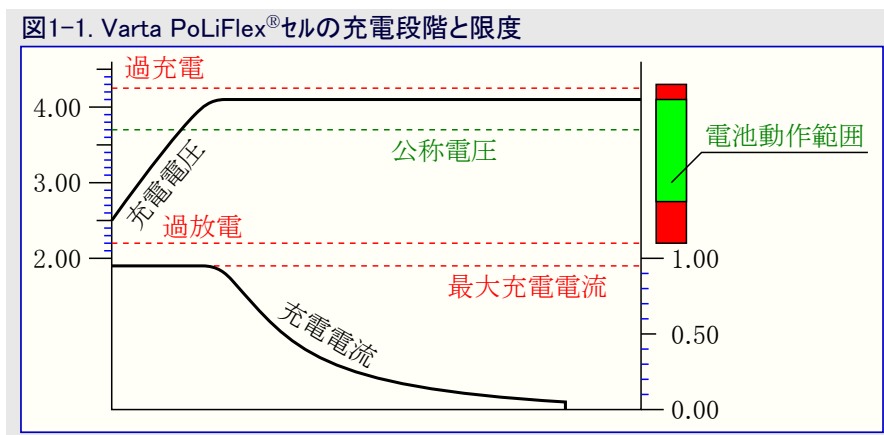
電池は使用されるか否かに拘らず、時の経過のために容量を失います。

2.2.3. 充電の段階

リチウムイオン電池には次のように2つの充電段階があります。

1. 定電流。Li-Ion電池の充電は電池に定電流を印加することで始まります。充電電流の大きさは電池依存で、製造業者によって与えられます。この段階は電池電圧が製造業者によって与えられた閾値に達した時に完了します。
2. 定電圧。電池閾値電圧到達後、充電器は定電流供給から定電圧供給に切り替えます。この段階は充電電流が製造業者によって与えられた閾値以下に落ちた時に完了します。

下図は充電中のリチウムイオン電池の電圧と電流を図解します。



上の図に於いて、“過充電”はセル保護回路が割り込んで半導体スイッチを開いて充電電流経路を中断する処のレベルです。この後、電流経路が回復される前に電池電圧は代表的に数100mVの低下が必要です。“過放電”は過放電から電池を守るために電流経路が切られる処のレベルです。電池動作電圧は過充電と過放電の限度から離れる余裕が推奨されます。

日本の規則に適合するために、充電の電流と電圧は表1-1.で示されるように4つの温度範囲に関して定義された最大の組を超えるられないことに注意してください。温度が指定された範囲外、換言すると、0°C未満または60°C越えの場合、充電は停止しなければなりません。この動きはこの応用記述で実装されますが、既定で許可されていません。形態設定の助けについては4.5.1.項をご覧ください。

T0は2つの異なる最大値を持つ特別な場合です。この温度範囲に於ける最速充電については以下のような充電法が使用されるべきです。

1. 電圧が4.10Vに達するまで(1.0C)定電流
2. 電流が0.5Cに減るまで(4.10V)定電圧
3. 電圧が4.25Vに達するまで(0.5C)定電流
4. 電流が閾値以下に減るまで(4.25V)定電圧

本来、充電器は異なる温度範囲間の遷移をも扱うことができなければなりません。電池の製造業者が制限的な限度を指定した場合、勿論それらがこれらの代わりに使用されるべきです。

2.2.4. 代表的な充電特性

電池仕様は常に製造業者のデータシートで確かめるべきです。右は代表的なリチウムイオン電池充電特性の要約です。実際のパラメータは変わるかもしれません。

表1-1. 日本の規則に従った最大充電電流と電圧

| 範囲識別子 | 温度範囲 | 最大電流 | 最大電圧 |
|-------|---------|------|-------|
| T0 | 0~10°C | 1.0C | 4.10V |
| | | 0.5C | 4.25V |
| T1 | 10~45°C | 1.0C | 4.25V |
| T2 | 45~50°C | 1.0C | 4.15V |
| T3 | 50~60°C | 1.0C | 4.10V |

表1-2. 代表的な充電特性

| 項目 | 代表値 |
|-----------------|-----------|
| 充電時間 | 3時間 |
| 充電電流 | 1C |
| 充電効率 | 99.9% |
| 充電電流閾値 | 0.03C |
| 充電電圧 | 4.20V |
| 充電電圧許容誤差(セル当たり) | ±0.05V |
| 温度範囲 | 0~45°C |
| 湿度範囲 | 65±20% RH |

2.2.5. 代表的な電池特性

下表はこの応用で使用される電池形式に関する製造業者のデータを要約します。他の形式の電池を使用できるかもしれませんが、ソフトウェアやハードウェアの調整が必要かもしれません。

表1-3. Varta PoLiFlexリチウムイオン電池に対する製造業者の範囲データ(「参照」の4.)

| 項目 | PLF443441 | PLF443441 | PLF443441 | PLF443441 | 単位 |
|------------|-----------|-----------|-----------|-----------|----|
| 公称電圧 | 3.70 | | | | V |
| 動作電圧範囲 | 2.75~4.20 | | | | |
| 充電電圧 | 4.20 | | | | |
| 充電電圧許容誤差 | ±50 | | | | mV |
| 充電電流 | 520 | 720 | 955 | 955 | mA |
| 最大充電時間 | 3 | 3 | 3 | 4 | 時間 |
| 充電停止電流 | 10 | 14 | 19 | 38 | mA |
| RID (登録ID) | 3.9 | 6.8 | 10 | 24 | kΩ |
| NTC | 10 | | | | |
| β値 | 3435 | | | | K |
| 過充電検出電圧 | 4.35 | | | | V |
| 過放電検出電圧 | 2.20 | | | | |

2.3. 電池充電器

この応用記述はATMEL®によるATAVRBC100電池充電器参照基準設計が目的対象です。この参照基準設計はむしろ複雑で特徴的な負荷を持ちますが、この応用は設計の核機能、降圧変換器だけに集中します。BC100のより多くの情報については「AVR451:BC100ハードウェア使用者の手引き」(「参照」の5.)を参照してください。

2.3.1. マイクロコントローラ

BC100は主(既定でATmega644)と従(既定でATtiny25/45/85またはATtiny261/461/861)の2つのマイクロコントローラを持ちます。主マイクロコントローラはこの応用の範囲外ですが、何時でも主装置が従装置からデータを要求するかもしれないように、マイクロコントローラがお互いに通信する能力があることが留意されるかもしれません。

従マイクロコントローラは電池充電に関連する全ての作業を処理する完全な能力があり、主マイクロコントローラの存在を必要としません。それは電池用のコネクタを継続的に走査し、(電池を)見つけたなら、必要とされる時にそれらを充電します。従マイクロコントローラは何らかの変則に関してハードウェアも継続的に監視します。

2.3.2. 電源

この応用記述は電源に注目しません。けれども動作が信頼に足ることを保証するためにファームウェアが継続的に入力電圧水準を監視することに留意されるかもしれません。

2.3.3. 降圧切り替え

従マイクロ コントローラ上のファームウェアはBC100基板上の3つの降圧変換器のどれかを制御します。既定は電圧を調整して電池に電流を流すためにマイクロ コントローラの高速PWM出力を使用することです。降圧変換器の電圧(と電流)はPWM信号のデューティサイクルに直接比例します。

3. 電池充電器ハードウェア

この応用記述はATAVRBC100電池充電器参照基準設計に基づきます。詳細なハードウェア説明はこの資料で提供されません。詳細情報については「AVR451:BC100ハードウェア使用者の手引き」をご覧ください。

3.1. 形態設定

ATAVRBC100電池充電器参照基準設計は下で詳述されるように形態設定されなければなりません。

3.1.1. マイクロ コントローラ

ハードウェアは以下のようにされるべきです。

- SC300ソケットが空であることを確認してください。
- ATtiny861をSC301ソケットに居させてください。

他のAVRマイクロ コントローラを使用することが可能ですが、この応用はATtiny861の使用に最適化されています。コンパイルされたコード量が減少されるなら、ATtiny261とATtiny461(「参照」の6.)のようなピン互換の置換が使用されるかもしれません。これはコンパイラの最適化を増すと、ファームウェアから不要な機能を取り去ることによって行うことができます。

他のマイクロ コントローラ選択にはATtiny25,ATtiny45,ATtiny85を含みます(「参照」の7.)。(他の8ピンAVRマイクロ コントローラと同じ様に)これらはBC100上のSC300ソケットを使用します。少ピン数のために8ピンのマイクロ コントローラが既定の20ピンよりも少ない機能しか提供しないことに注意されるべきです。

3.1.2. プログラミング コネクタ

マイクロ コントローラはSPIまたはデバッグWIREのどちらかを使用して6ピンのJ301コネクタ経由でプログラミングすることができます。

BC100の或るハードウェア改訂版に於いて、R303を取り去ってU202の15番ピンを切断する必要があるかもしれません。この手続きは外部の書き込み器やデバッグによる使用のために/RESET線を自由にしますが、主マイクロ コントローラが従装置をリセットする可能性を取り去ります。必要とされる限り、基板を変えないでください。代わりに、マイクロ コントローラは基板外で常にプログラミングすることができます。

3.1.3. ジャンパ

ジャンパは以下のように形態設定されるべきです。

- J401,J404,J407,J408 : 降圧変換器C(20V/1A)を使用するように設定してください。
- J405とJ406 : ジャンパを1/4に設定してください(最大測定可能電圧=10V)。

他の形態設定も可能ですが、ファームウェアの変更が必要かもしれません。ADC.hファイルでVBAT_RANGE変数をご覧ください。

3.1.4. 電池

この応用は特別な形式のリチウム イオン電池を使用し、ここで提供された全ての形態設定は製造業者のデータに基づきます。本質的に他のリチウム イオン電池も使用できるかもしれませんが、製造業者のデータシートから電池データを探してファームウェアとハードウェアに必要な調整が行われるのを保証するのは使用者の義務です。4.5.1.項とbattery.hファイルをご覧ください。

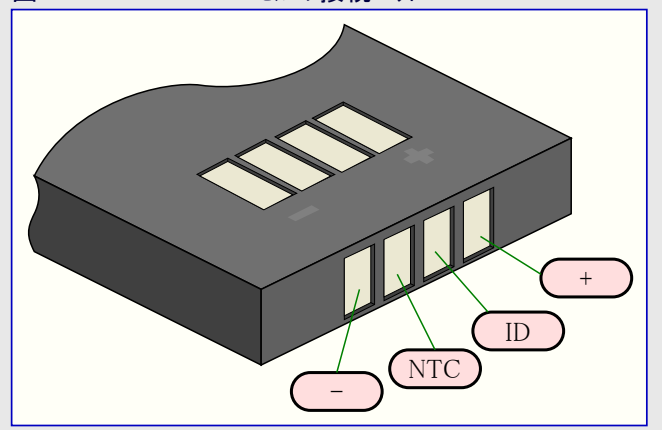
右図はこの応用で使用されるリチウム イオン電池の接続パッドを図解します。

電池は次のように電池充電器へ接続されます。

表1-4. 充電器への電池接続

| 電池コネクタ | 充電器コネクタ | 注意 |
|--------|----------|-------------|
| - (負極) | BATTERY- | |
| NTC | NTC/RID | 電池温度測定 |
| ID | SCL | RID、電池識別抵抗器 |
| + (正極) | BATTERY+ | |

図1-2. Varta PoLiFlexセルの接続パッド



3.1.5. データEPROM

いくつかの電池は充電と製造のデータを格納するための組み込みEPROMが装備されています。本応用は単線インターフェース経由でのEPROMの使用を支援します。既定は右のように接続されたDS2502 EPROMです。

EPROMが電池充電器に接続されない場合、応用は単にその不在を無視するでしょう。

表1-5. 充電器へのDS2502外部EEPROM接続

| EPROMピン | 充電器コネクタ |
|---------|------------|
| DATA | 1-WIRE/SDA |
| GND | BATTERY- |

3.1.6. 供給電圧

より高い供給電圧とより高い最小電流を降圧切り換えは提供することができます。例えば、供給電圧が約9Vで4.20Vで電池を充電するのに降圧変換器Cが使用される場合、得られる最小電流は約80mAです。この点に於けるPWMデューティサイクルでの最小減少(換言すると、1によるOCR1B内容の減少)は実際上、電池への電流をOFFにするでしょう。

電池充電電圧より約3V高い供給電圧を使用することが推奨されます。この応用では充電中に電池電圧が代表的に最大4.5Vを出力し、故に供給電圧は7.5Vが推奨されます。

ハードウェアが提供できる最小充電電流をより低めるための別の方法はより大きなインダクタを降圧変換器に使用することです。BC100でこれは降圧変換器Aを意味します。

4. 電池充電器ソフトウェア

ファームウェアはIAR SystemsのEmbedded Workbench 4.20版を使用してC言語で書かれています。ファームウェアが全体的にCで書かれているため、他のAVR Cコンパイラへ移設することは難しくないでしょう。いくつかのコンパイラ特有の項目は多分書き直される必要があるでしょう。

下表はコンパイラのプロジェクトに関連するファイルを一覧にします。

表1-6. プロジェクトファイル (IAR EWの作業空間ファイルBC100_tiny.ewwをご覧ください。)

| ファイル名 | 形式 | 注記 |
|--------------|----------|--------------------------------|
| ADC.c | Cソースファイル | A/D変換に関連する関数 |
| ADC.h | ヘッダファイル | |
| battery.c | Cソースファイル | 登録IDとNTCの参照表、電池制御とデータ獲得に関連する関数 |
| battery.h | ヘッダファイル | |
| chargefunc.c | Cソースファイル | 指定されたパラメータに従って実際に充電を行う関数 |
| chargefunc.h | ヘッダファイル | |
| enums.h | ヘッダファイル | ADC.cとtime.cで使用される列挙 |
| main.c | Cソースファイル | 主プログラム/プログラム入口 |
| main.h | ヘッダファイル | |
| menu.c | Cソースファイル | 状態機構定義 |
| menu.h | ヘッダファイル | |
| OWI.c | Cソースファイル | 単線インターフェースに関連する関数 |
| OWI.h | ヘッダファイル | |
| PWM.c | Cソースファイル | パルス幅変調出力の生成に関連する関数 |
| PWM.h | ヘッダファイル | |
| statefunc.c | Cソースファイル | menu.cで定義された状態に関連する関数 |
| statefunc.h | ヘッダファイル | |
| structs.h | ヘッダファイル | プロジェクト全体を通して使用される様々な構造体の宣言 |
| time.c | Cソースファイル | 時間維持に関連する関数 |
| time.h | ヘッダファイル | |
| USI.c | Cソースファイル | 直列インターフェースに関連する関数 |
| USI.h | ヘッダファイル | |

4.1. 概要

ファームウェアは2つのリチウムイオン電池を充電するのに必要とする全ての関数を統合します。電池は一方が充電され得る間に他方がアイドルとなるように独立したポートに接続されます。ファームウェアは完全に自動化され、自立した電池の監視と充電の能力がありますが、またBC100で実装されているもののように、主マイクロコントローラと共に使用されるかもしれません。

既定により、ファームウェアはATtiny861(構築任意選択:デバッグ)またはATtiny461(構築任意選択:提出)に適合します。ファームウェアのメモリ必要条件は下表で要約されます。

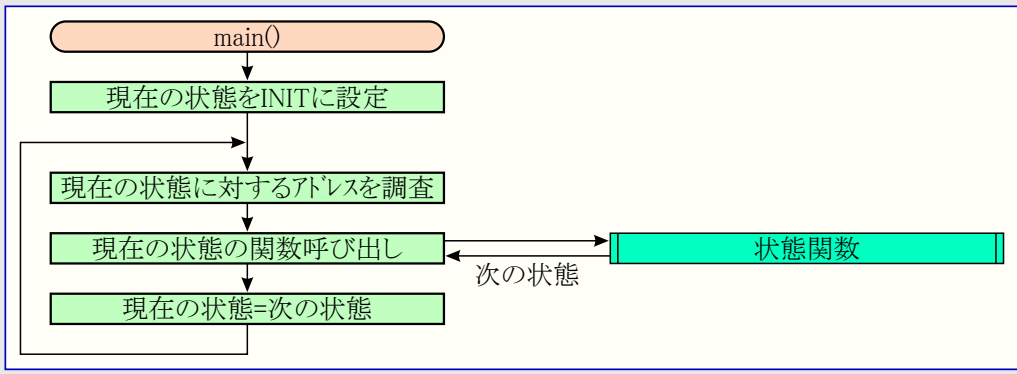
表1-7. ファームウェアのメモリ必要条件

| 構築任意選択 | メモリ | 概ねの値 (バイト) |
|--------|-----------------|------------|
| デバッグ | CODE (フラッシュメモリ) | 5800 |
| | DATA (SRAM) | 270 |
| | XDATA (EEPROM) | 130 |
| 提出 | CODE (フラッシュメモリ) | 3900 |
| | DATA (SRAM) | 270 |
| | XDATA (EEPROM) | 130 |

4.2. 状態機構

状態機構はかなり簡単でmain()関数に属します。単純に実行するための次の関数のアドレスを調べて、そしてその関数へ飛びます。状態機構の流れ図が下図で図解されます。

図1-3. 状態機構を含むmain関数の流れ図



戻りに於いて、状態機構は戻り引数として次の状態を示すことを期待します。承認された戻り符号が下表で記述されます。

表1-8. 状態機構符号(ソースコードとmenu.hをご覧ください)

| ラベル名 (注1) | 関連する関数 (注2) | 説明 |
|-------------|------------------|----------------|
| INIT | Initialize() | 入口の状態 |
| BATCON | BatteryControl() | ハードウェアと電池を調査 |
| PREQUAL | Charge() | 電池電圧上昇、安全性検査 |
| SLEEP | Sleep() | 低消費電力動作形態 |
| CCURRENT | Charge() | 定電流で充電 |
| CVOLTAGE | Charge() | 定電圧で充電 |
| MAXVOLTcurr | Charge() | 最大電圧または最大電流で充電 |
| ENDCHARGE | Charge() | 充電成功終了 |
| DISCHARGE | Discharge() | |
| ERROR | Error() | 可能ならば、異常を解決 |

注1: 先行する"ST_"を除いたラベル名

注2: ソースコードで宣言された関数名

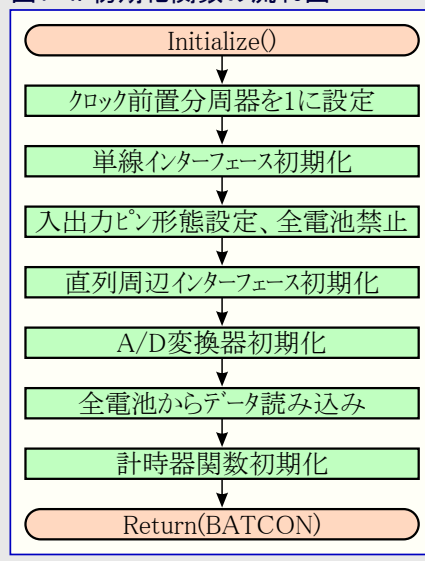
状態関数は以降の項で記述されます。

4.2.1. Initialize()

初期化関数はデバイスリセット後に実行される最初の状態関数です。この関数の流れ図は右図で示されます。

初期化関数は常に電池制御のための状態関数を示す同じ戻り符号で抜け出します。

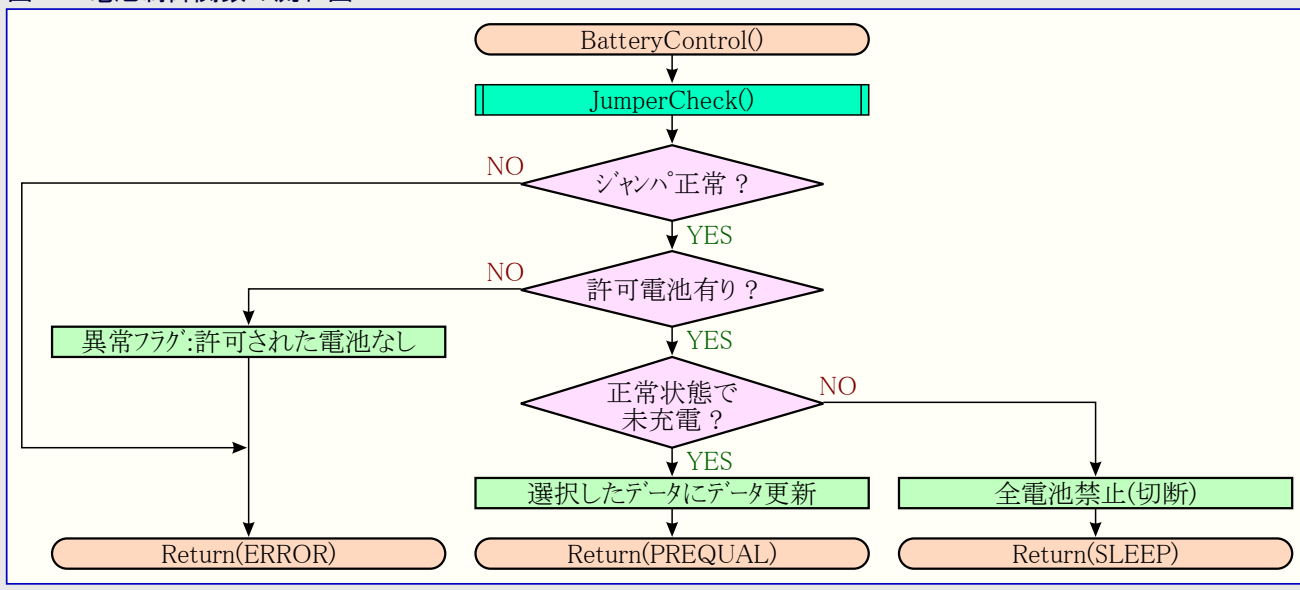
図1-4. 初期化関数の流れ図



4.2.2. BatteryControl()

電池制御関数はジャンパが正しく設定されていること確かめ、そして充電を必要とする何れかの許可された電池が存在するかを確認するために調べます。プログラムの流れは下図で図解されます。

図1-5. 電池制御関数の流れ図



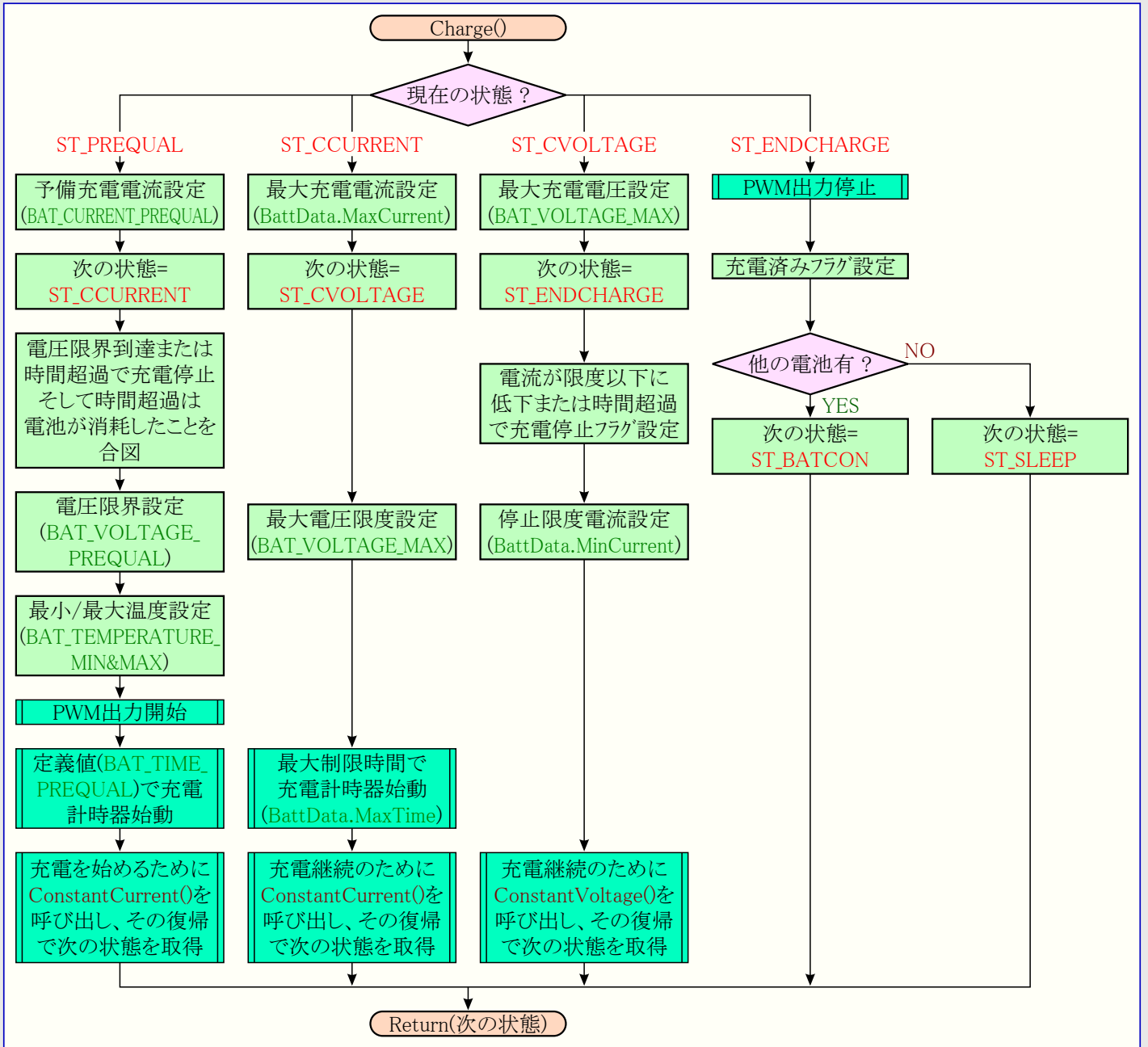
4.2.3. Charge()

現在の実装ではそれが日本の規則に適合するように形態設定するか否かに依存して、充電関数2つの異なる動作形態を持ちます。非適合形態は最も簡単で4つの段階から成ります。

- 予備検定 - 十分な充電電圧に至るまで電池が定電流で充電される区間。与えられた時間内にこれが起きるなら、電池は良好と見做され、充電器は次の段階を続けるかもしれません。電圧到達前に時間超過、または電池温度が範囲外になった場合、電池はダメと見做され、充電は中止されます。
- 定電流充電 - 電池電圧がその最大に達するまでより高い電池仕様の電流で電池が充電される区間。電池の最大充電制限時間内にこれが起きたなら、充電器は次の段階へ行きます。制限時間経過、または電池温度が範囲外になった場合、電池はダメと見做され、充電は中止されます。
- 定電圧充電 - 充電電流が電池指定の停止限度未満に落ちるか、または最大充電制限時間経過まで、最大電池電圧で電池が充電される区間。ここでも電池温度が範囲外になった場合に充電が停止されます。
- 充電終了 - 休止状態へ行くのか、または他の電池の充電を試みるのかのどちらかを定める処です。

ChargeParametersとHaltParametersはこの関数に於ける中心的な変数です。各段階後、関数は次に望む状態をmain()に返します。この状態関数のプログラムの流れは下図で図解されます。

図1-6. 充電状態関数の流れ図



順法形態は1つのST_MAXVOLTcurr内に組み合わせられた第2(ST_CCURRENT)と第3(ST_CVOLTAGE)の状態を持ちます。この関数は本質的に充電電圧と充電電流の両方が温度範囲に対して制限内であることを保証します。

4.2.4. Discharge()

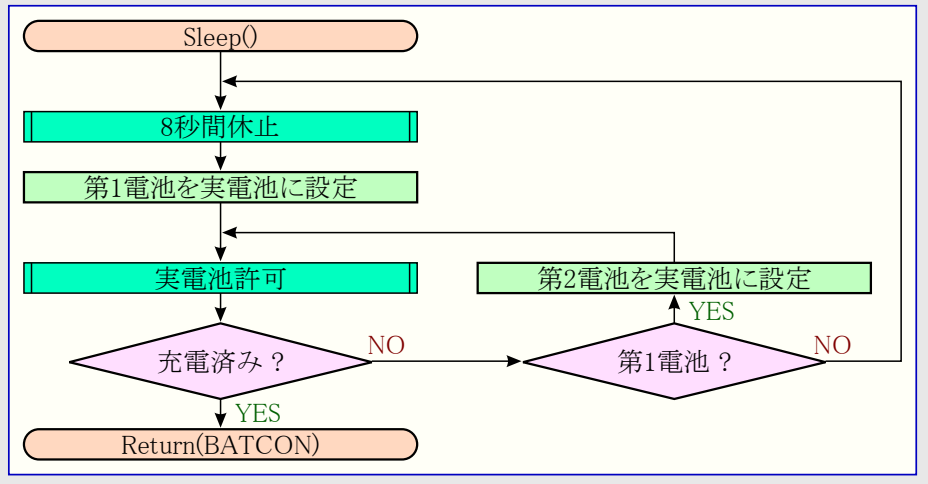
この関数は実装されていません。

4.2.5. Sleep()

応用は電池が完全に充電されてしまった時に休止形態へ移行します。それは電池の現在の状態を調べるために一定間隔で起き上がります。休止形態は何れかの電池が充電を必要とすると直ぐに終了されます。

休止形態は右図で図解されます。

図1-7. 休止関数の流れ図

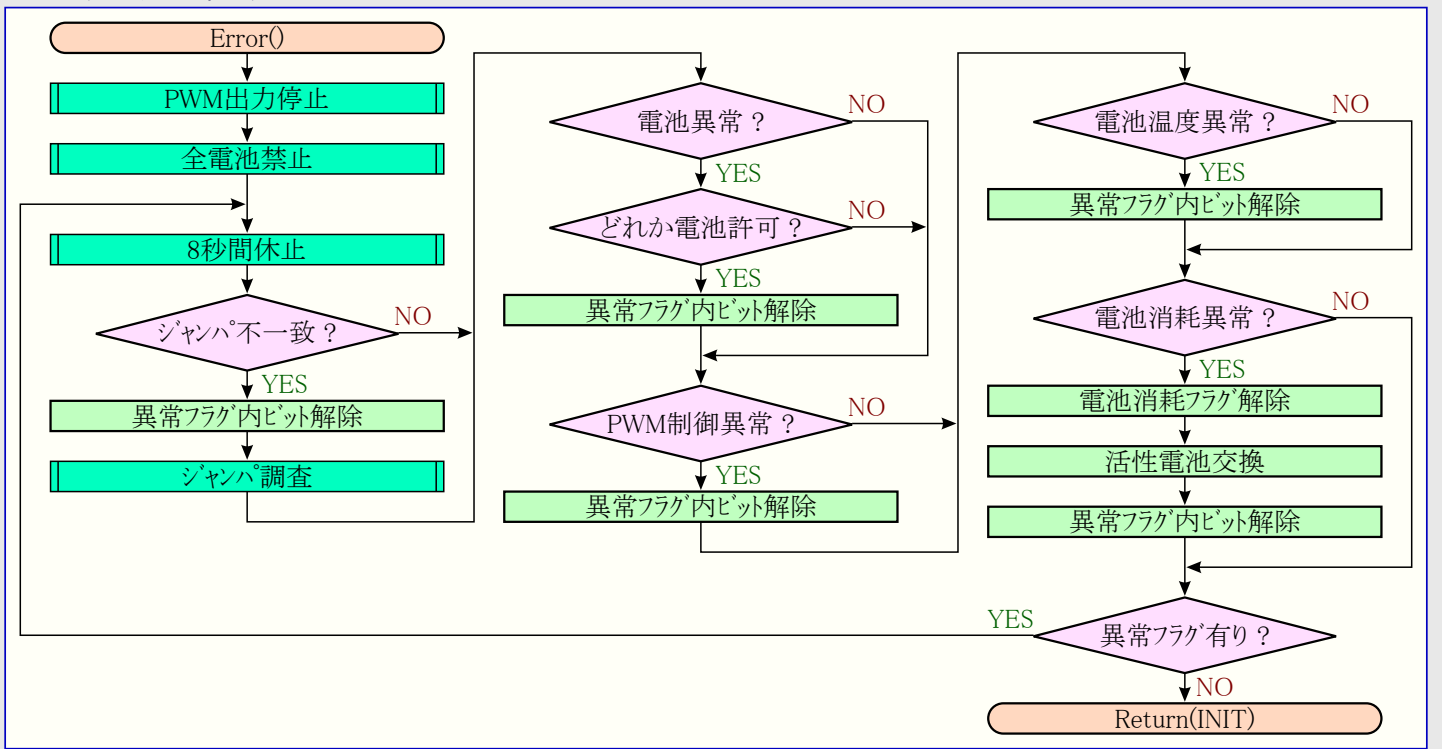


4.2.6. Error()

異常が起きた時にプログラムの流れはここで転換されます。異常処理部は殆どの一般的な問題の解決を試みるためのいくつかの簡単な算法を含みます。プログラム実行は全ての異常元が解除された時に異常処理部を抜け出します。

このプログラムの流れは下図で図解されます。

図1-8. 異常処理部の流れ図



4.3. 充電関数

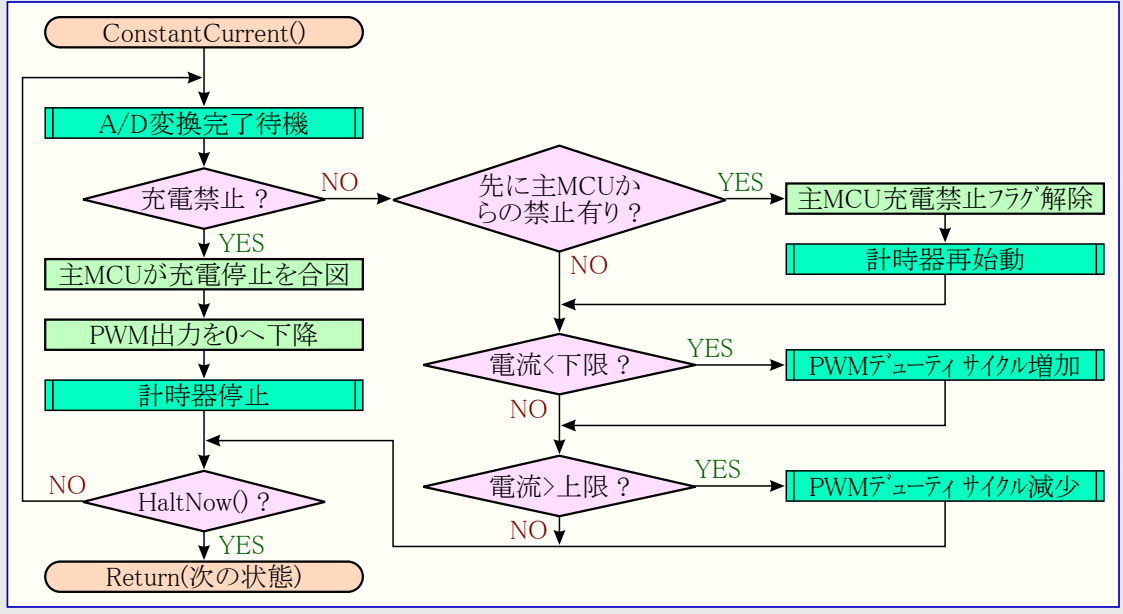
これらの関数は全てのパラメータが設定された後でCharge()によって呼ばれます。

4.3.1. 定電流と定電圧

それらが限度内に保持を試みるためのA/D変換測定が何かを別にして、これら2つの関数は同じです。従って、ConstantCurrent()の流れ図だけが下図で図解されます。これら両方ともChargeParameter変数を使用します。

主マイクロコントローラが存在する場合、充電禁止の合図によって一時的に充電を停止するかもしれません。これは長い直列転送中に於ける電池の損傷を防ぐためです。

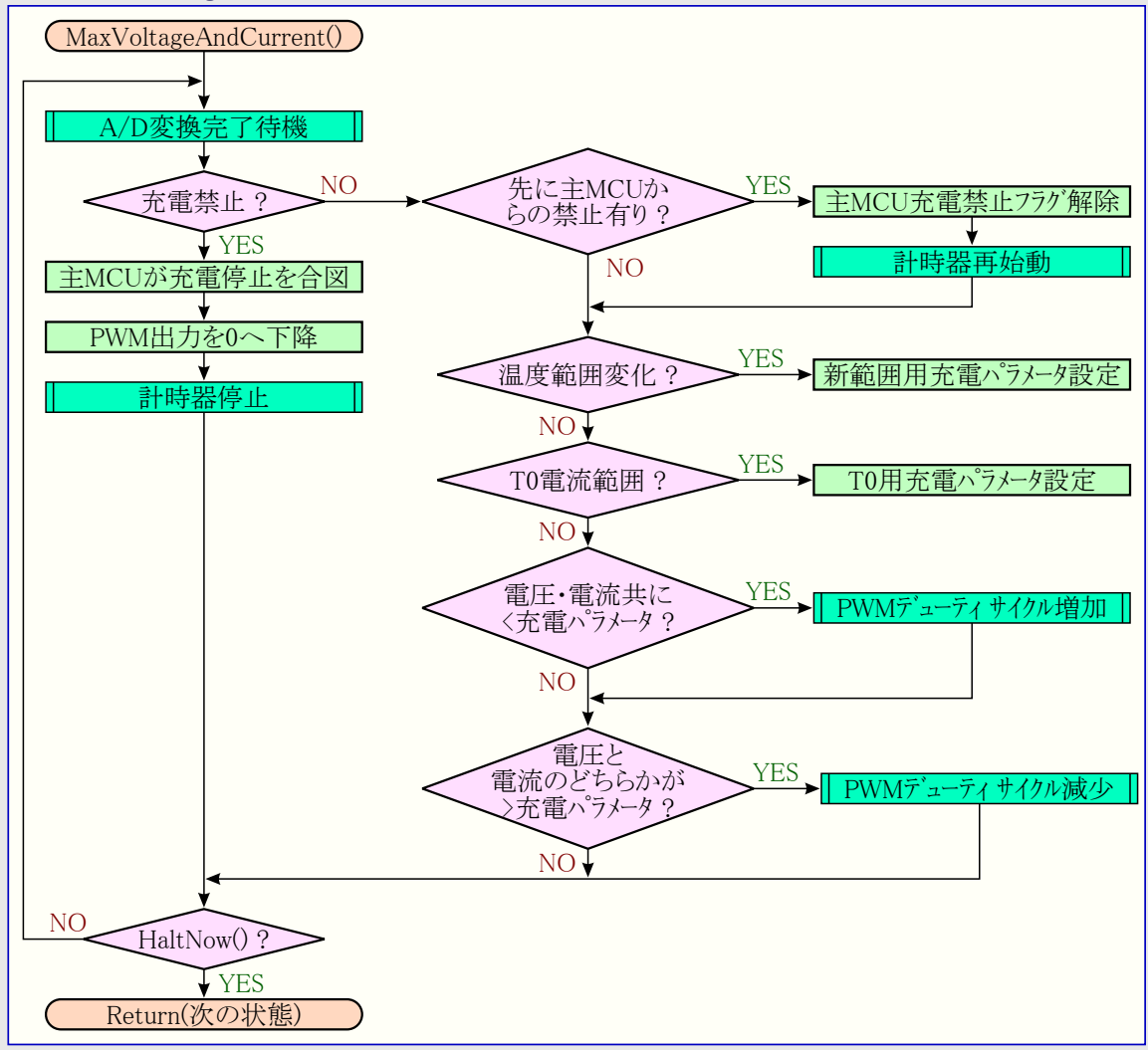
図1-9. ConstantCurrent()用流れ図



4.3.2. 最大電圧と最大電流

電池充電に関する日本の規則のため、第3の充電関数MaxVoltageAndCurrent()が提供されます。この関数は電池の温度を継続的に監視し、充電項目(電流と電圧の両方)が先の表1-1.で示される規則によって定義された限度内であることを保証します。本質的にこれは他の2つの関数が組み合わさった温度依存版です。図1-10.は幾分簡単化したこの関数の流れ図を示します。

図1-10. MaxVoltageAndCurrent()用流れ図



充電停止が次に記述される判定関数によって決定されるため、温度が範囲外(0°C未満または60°C越え)の場合にこの関数が充電を中止しないことに注意してください。

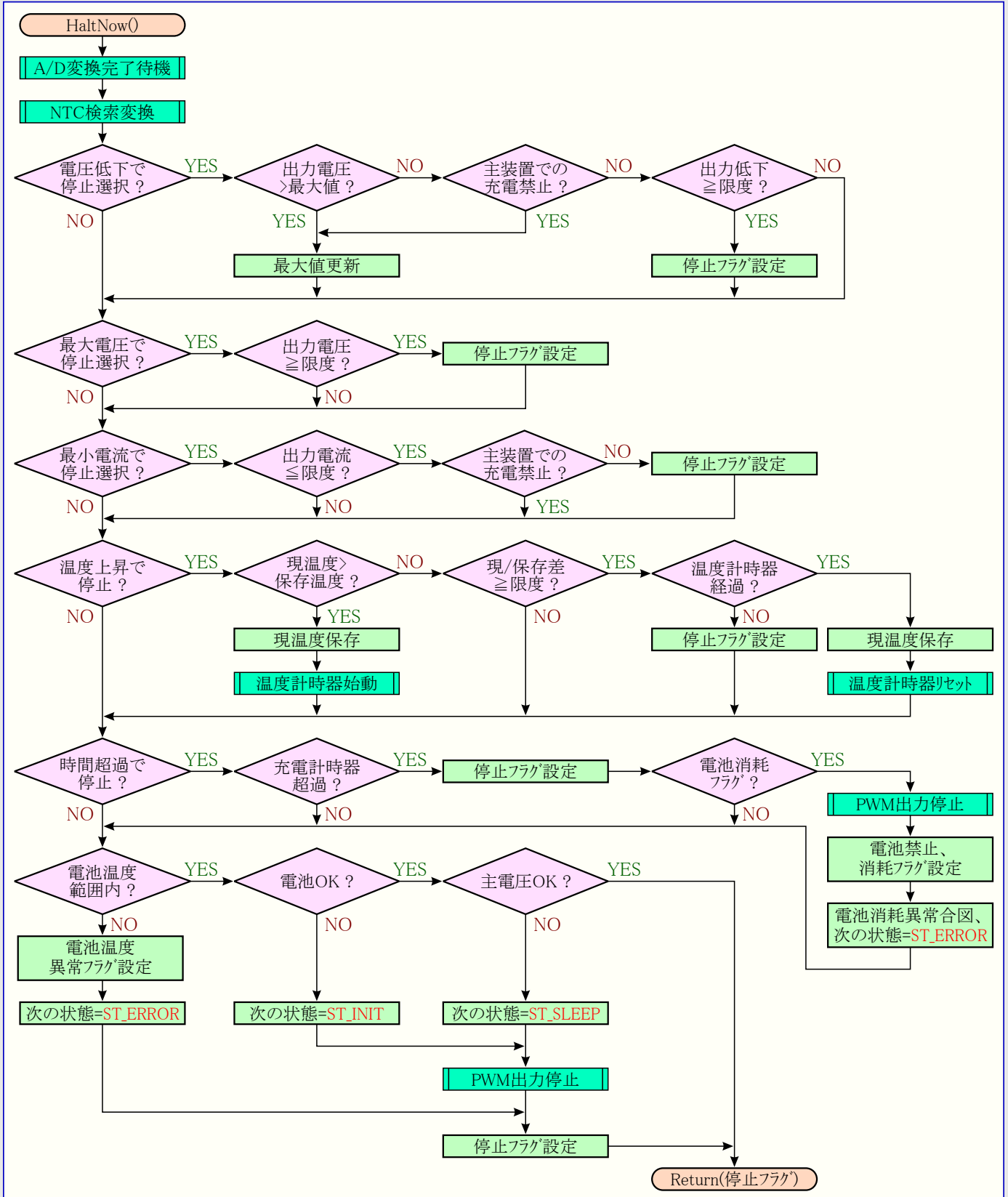
4.3.3. 充電停止判定

充電停止はHaltNow()によって判断されます。この関数は充電の段階が終わったかどうかを決めるために、それらが繰り返す度毎にConstantCurrent()、ConstantVoltage()、MaxVoltageAndCurrent()によって呼び出されます。

HaltParameters変数で使用者は何の項目で充電が停止されるべきで、例えば制限時間経過の場合に異常が合図されるべきかを指定することができます。主MCUが充電を禁止する場合、電圧低下または充電電流が最大閾値以下に下降することが停止を起動しないことに注意してください。また異常フラグは次の状態として設定されるST_ERRORに帰着し、これによって充電を中止します。何も異常が合図されなければ、Charge()内で先に設定された次に望まれる状態が適用されます。

最後に、関数は温度が限度内か、電池が正常か、主電圧が最小以上かを調べます。これらの検査のどれかが誤りなら、次の状態は適切な異常処理部(ST_ERROR, ST_INIT, ST_SLEEP)に設定され、充電は中止されます。

図1-11. HaltNow()用流れ図



(訳注) 原書の図1-11.~14.は図1-11.として纏めました。

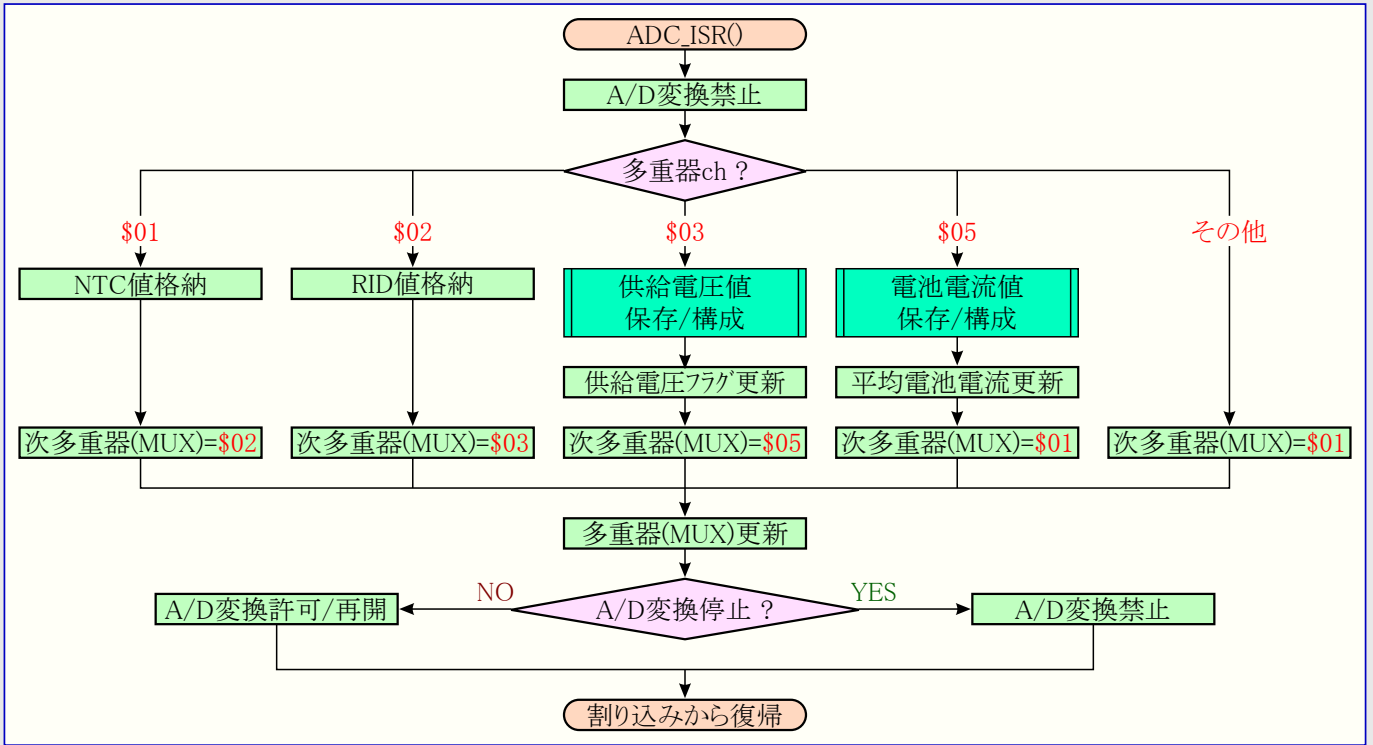
4.4. その他の関数

プログラムの流れは主に状態に基づきますが、いくつかの処理は背面で行われます。これはA/D変換、時間維持、直列インターフェースの処理を含みます。これらの関数の全てが割り込み駆動です。

4.4.1. A/D変換

A/D変換は多数のチャンネルからデータを読むのに多重器を用います。下の流れ図で図解されるように、変換の最後でA/D変換割り込み処理ルーチン(ISR)が呼ばれます。ISR完了後、プログラム実行は通常に戻ります。

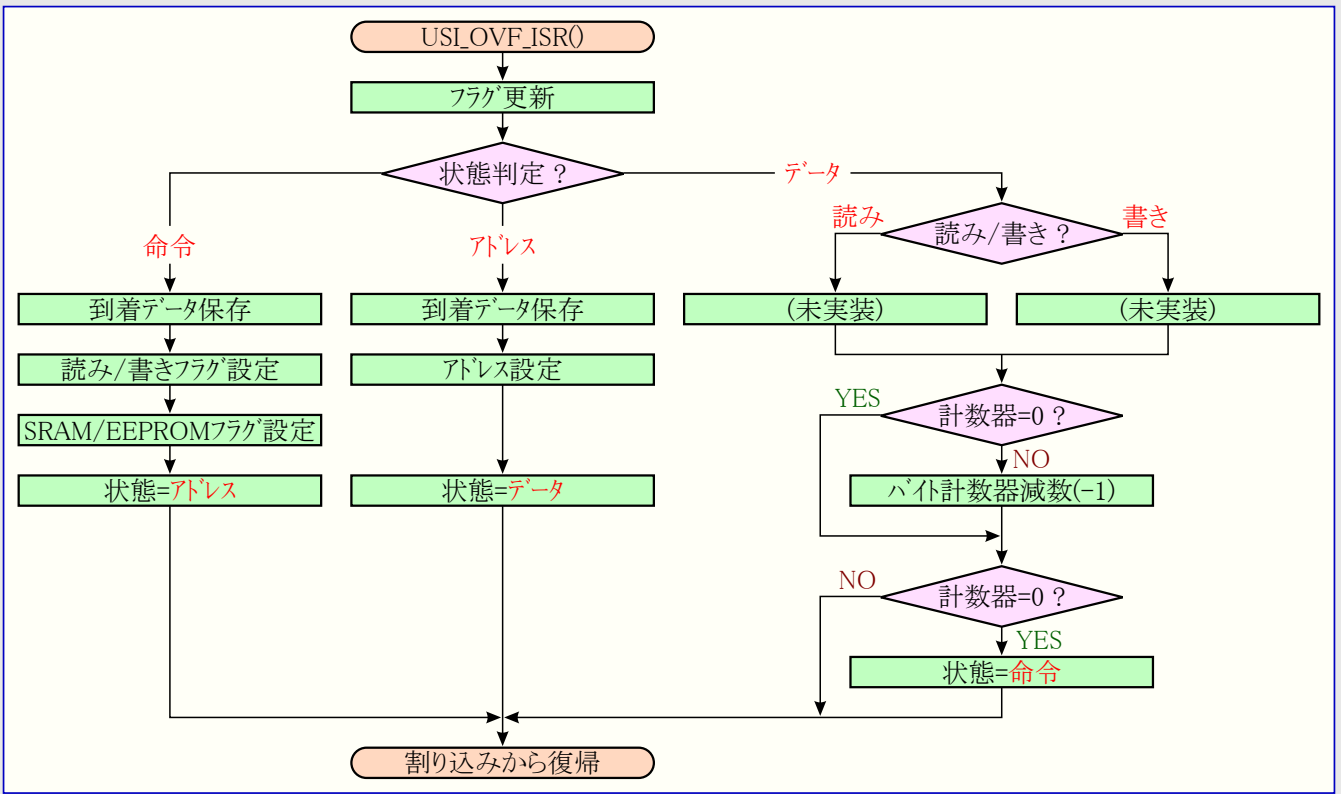
図1-15. A/D変換割り込み処理ルーチンの流れ図



4.4.2. 主/従通信

この応用は自立型として動くように設計されていますが、他のマイクロコントローラでの協力動作も支援します。多用途直列インターフェース(USI)はマイクロコントローラ間の通信に使用することができます。このインターフェースに対する基本規約は開発されていますが、完成させるにはいくつかの機能が必要です。

図1-16. USI溢れ割り込み処理ルーチンの流れ図



4.5. 実装

本項はソフトウェアの形態設定、作成、書き込み(プログラミング)方法を記述します。

4.5.1. 形態設定

コンパイル時に重要な殆どの定数が下表で一覧で言及されます。

表1-9. 電池に関連するコンパイル時定数 (LIIONspecs.hソースファイルをご覧ください。)

| ラベル名 | 説明 |
|---|--|
| BAT_CELL_NUMBER | 電池のセル数。BAT_VOLTAGE_MAX、_LOW、_MIN、_PREQUALを定義するために、定義されたセル電圧の各々がこれで乗算されます。 |
| CELL_VOLTAGE_SAFETY | 充電されるべき電池が不一致の場合に、電池内の追加セル毎、換言するとBAT_CELLNUMBER-1に関して、この定数がCELL_VOLTAGE_MAXから引かれます。 |
| CELL_VOLTAGE_MAX_T0 CELL_VOLTAGE_MAX CELL_VOLTAGE_MAX_T2 CELL_VOLTAGE_MAX_T3 | 各種温度範囲に於ける、充電のための最大セル電圧。 |
| CELL_VOLTAGE_LOW | セルが充電されたと見做される最低電圧。充電は電圧がこの水準以下に落ちた時に始まります。 |
| CELL_VOLTAGE_MIN | 充電が始められる最低電圧。一般的に電池の更なる放電が損傷を引き起こす下限電圧に設定されるべきです。 |
| CELL_VOLTAGE_PREQUAL | 予備検定(充電)中に充電されるべき電池の電圧。 |
| BAT_TEMPERATURE_MAX | 電池許容最高温度。これ越えて充電は停止されるか、または開始されません。 |
| BAT_TEMPERATURE_MIN | 電池許容最低温度。これ未満で充電は停止されるか、または開始されません。 |
| BAT_CURRENT_PREQUAL | 予備検定(充電)中の充電電流。 |
| BAT_CURRENT_HYST | 充電電流ヒステシス。目標±この値内の時に電流は調整されません。 |
| BAT_VOLTAGE_MAX_T0 BAT_VOLTAGE_MAX BAT_VOLTAGE_MAX_T2 BAT_VOLTAGE_MAX_T3 | BAT_CELL_NUMBER,CELLVOLTAGE_MAX[...],CELL_VOLTAGE_SAFETYから計算された、充電のための電池最大電圧 |
| BAT_VOLTAGE_HYST | 充電電圧ヒステシス。目標±この値内の時に電圧は調整されません。 |
| BAT_VOLTAGE_PREQUAL | 予備検定(充電)段階中の目標電圧。この電圧未到達の場合に電池は消耗と記されます。これはBAT_CELL_NUMBERとCELL_VOLTAGE_PREQUALから計算されます。 |
| BAT_TIME_PREQUAL | 予備検定(充電)段階で費やされる最大時間。 |

表1-10. 既定電池、RID、NTC用コンパイル時定数 (battery.c,battery.h)

| ラベル名 | 説明 |
|------------------------|---|
| DEF_BAT_CAPACITY | 既定電池容量。 |
| DEF_BAT_CURRENT_MAX | 既定最大充電電流。 |
| DEF_BAT_TIME_MAX | 既定最大充電時間。 |
| DEF_BAT_CURRENT_MIN | 既定中止充電電流。 |
| ALLOW_NO_RID | 定義されていれば、RIDなしの(または参照表に一致しない)電池は既定電池(情報)を用いて充電させます。さもなければRIDなしは充電が開始されないことを意味します。 |
| RID[].LowとRID[].High | 値がこれらの限度内の場合にRID抵抗一致と仮定します。 |
| RID[].Capacity | 与えられたRIDに対する電池容量。 |
| RID[].Icharge | 与えられたRIDに対する充電電流。 |
| RID[].tCutOff | 与えられたRIDに対する最大充電時間。 |
| RID[].IcutOff | 与えられたRIDに対する充電終了電流。 |
| NTC[].ADC | 各種温度に対するA/D変換値(配列指標は(A/D変換)値に関連する温度と等価です)。 |
| NTC[].ADCsteps | 測定された温度から変換された0.5°C当たりのA/D変換段階値。 |

表1-11. 順法充電形態設定 (charge.h)

| ラベル名 | 説明 |
|-----------|---------------------------|
| JAPANREGS | 定義されているなら、充電器は日本の規則に従います。 |

表1-12. 順法充電用温度設定 (chargefunc.h)

| ラベル名 | 説明 |
|-----------|---|
| TEMP_HYST | 温度ヒステリシス。充電器が日本の規格に適合するように形態設定された場合にだけ使用。 |
| T0 | 各種温度範囲に於ける最大温度を定義します(BAT_TEMPERATURE_MAX/MINが充電を停止することを覚えて置いてください)。 |
| T1 | |
| T2 | |
| T3 | |

4.5.2. コンパイル

コードをコンパイルする前に以下の形態設定が行われるべきです。

表1-13. コンパイル形態設定

| 分類 | タブ | 領域 | 値 |
|-----------------|---------------|----------------------------|---|
| General Options | Target | Processor configuration | ATtiny861 (注) |
| | | Memory model | Small |
| | System | Data stack | 0x40 |
| | | Return address stack | 24 |
| | | Enable bit definitions ... | チェック |
| C/C++ Compiler | Language | Require prototypes | チェック |
| Linker | Output | Format | Other: ubrof8 |
| | Extra Options | Command Line | -y(CODE) -Ointel-extended,(DATA)=\$EXE_DIR\$¥\$PROJ_FNAME\$_data.hex -Ointel-extended,(XDATA)=\$EXE_DIR\$¥\$PROJ_FNAME\$_eeprom.hex |

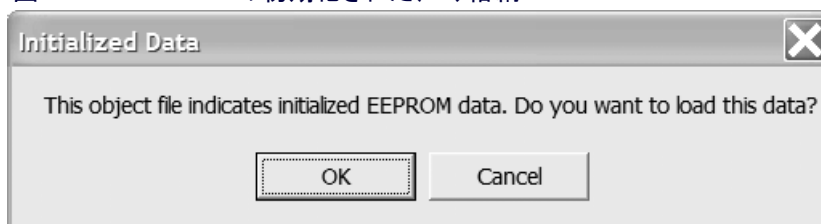
注: 他の選択も可能です。より多くの情報については4頁の3.1.1項をご覧ください。

4.5.3. プログラミング

コンパイルしたコードはAVR Studio®とJTAICEmk IIのような選りすぐりのデバッガまたは書き込みツールを用いて便利に目的対象デバイスへ書かれます。

ソフトウェアが動くために、EEPROMデータを含むコンパイルされたコードが目的対象に格納され(書かれ)なければならないことに注意してください。EEPROM内容が格納されるべきかをAVR Studioが尋ねる時にOKで答えてください。これは下図で図解されます。

図1-17. EEPROMへの初期化されたデータ格納



プログラムは内部発振器の使用とクロック信号が前置分周されないことを期待します。正しいプログラム実行を保証するために、いくつかのヒューズビットが設定されなければなりません。既定から外れるヒューズ設定が右表で一覧にされます。

表1-14. 既定でないヒューズビット設定

| ヒューズビット | 設定 | 説明 |
|----------|-----------|----------|
| CKDIV8 | 1(非プログラム) | 8分周を行わない |
| CKSEL3~0 | 0010 | 内部発振器を使用 |

5. 既知の制限

ここは一覧にされた設計の既知の制限です。

5.1. 電池電流測定

電池電流は非常に低い抵抗値を持つ分圧抵抗器を用いて感知されます。これは測定される信号に於いて雑音が容易に拾われ、非常に低い振幅を持つ雑音でさえ、測定を乱すかもしれないことを意味します。救済策として、測定された電池電流は4採取に渡って平均化されます。

更に、1または2LSBの程度の変動に出会うのは珍しくありません。既定では(3.1.3項をご覧ください)、これが7または14mAの測定誤差を意味します(ADC.cファイルのScale()関数をご覧ください)。実際問題として、これは早まった充電周期の終りに帰着するかもしれません。

提案される解決策は分圧抵抗(R410:より大きな方がよい)と抵抗分圧器(R400~410,R427,R428,R446,R447)の値を最適化することです。

5.2. RID感知

電池識別抵抗はPA2(ADC2)ピン経由で感知されます。この線上の既定プルアップ抵抗(ATAVRBC100電池充電器参照基準設計のR305)は4.7kΩです。これは感知抵抗値を最大約14.7kΩに制限します。

Varta PILiFlex電池使用時、これは信頼に足る感知ができる最大電池容量が1000mAhであることを意味します。より大きな感知抵抗/電池容量については、BC100上のプルアップ抵抗が変更されなければなりません。加えて、ソフトウェアは新しいプルアップ抵抗値を反映するように更新されなければなりません。

5.3. 降圧充電器

降圧充電器(と供給電圧)の選択は最小充電電流をどの位低くできるかの限度を設定します。より高い供給電圧とより小さい高圧切替インダクタは最小充電電流をより高くするでしょう。これは或る形態設定が早まった充電周期の終りに帰着するかもしれないことを意味します。

救済策は低い供給電圧と大きなインダクタを持つ降圧切り替えを使用することです。

6. 参照

1. "What's the best battery?" – Battery Universityから2007年4月3日に取得:
<http://www.batteryuniversity.com/partone-3.htm>
2. "Lithium-ion safety concerns" – Battery Universityから2007年4月3日に取得:
<http://www.batteryuniversity.com/partone-5B.htm>
3. "Charging lithium-ion batteries" – Battery Universityから2007年4月3日に取得.
<http://www.batteryuniversity.com/partone-12.htm>
4. "VARTA PoLiFlex Sales Program and Technical Handbook" – VARTA Microbatteryから2007年5月10日に取得:
<http://www.varta-microbattery.com/en/oempages/index.htm>
5. "AVR451:BC100ハードウェア使用者の手引き" – ATMELのウェブサイトから入手可能:
<http://www.atmel.com/products/avr/>
6. "ATtiny261/461/861データシート" – ATMELのウェブサイトから入手可能:
<http://www.atmel.com/products/avr/>
7. "ATtiny25/45/85データシート" – ATMELのウェブサイトから入手可能:
<http://www.atmel.com/products/avr/>



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 2008. 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。

© HERO 2013.

本応用記述はATMELのAVR458応用記述(doc8080.pdf Rev.8080B-08/08)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。