

AVR146 : ATmega16/32U4での USB経由リチウム イオン電池充電

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
- 電池IDと温度の読み込み用アナログ入力
- 使用者インターフェース用USB CDCクラス

1. 序説

この応用記述はATmega16/32U4に基づいて電源としてUSB接続を使ってリチウム イオン(Li-Io n)電池を充電するのにEVK527評価キットを使う方法に集中します。

USB CDCクラスは充電パラメータを表示するための容易なインターフェースを提供します。

この応用記述は「AVR458:ATAVRBC100でのリチウム イオン電池充電」に由来します。

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。

2. 説明

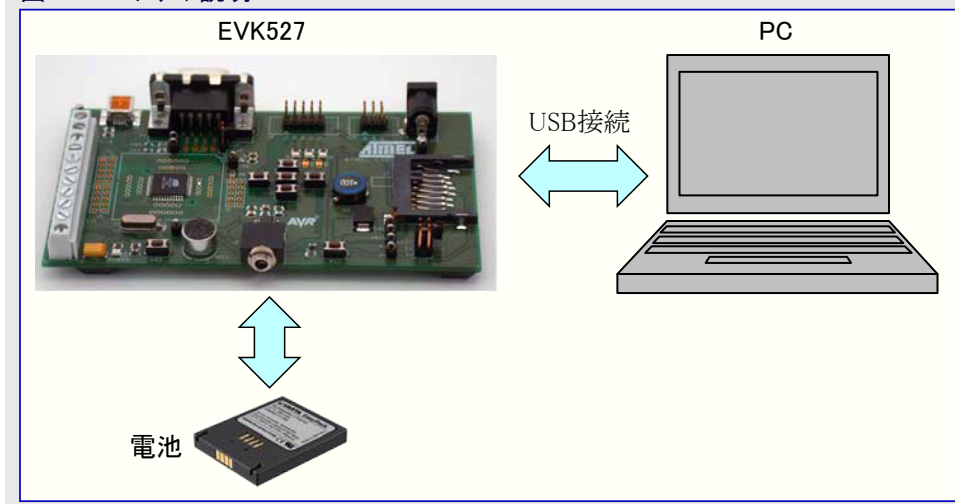
この資料はEVK527評価キットで走行する応用を記述します。EVK527はATmega16/32U4専用です。

USBはVBUSピンで5V電源を提供します。利用可能な電流範囲は100mA～500mAです。これはリチウム イオン電池セルを充電するのに充分です。

リチウム イオン セルは充電中に電圧と電流の正確な制御が必要です。

ATmega16/32U4はUSB全速(Full-speed)インターフェース、PWMチャネル、10ビットA/D変換チャネルを提供します。これらの機能の全てがUSB経由リチウム イオン電池充電器の実行に使われます。

図2-1. ハードウェア説明



使用者に好意的なインターフェースのため、全ての充電パラメータ(充電状態、電池電圧、充電電流、電池温度など)が測定道具を使わずにPCで表示されます。

USB列挙(接続認証)後、仮想通信ポートが宣言されます(図2-2をご覧ください)。この通信ポートに接続されたハイパー ターミナル ウィンドウが充電パラメータを表示します。この通信ポートは仮想で、従ってハイパー ターミナルのポート設定(速度、パリティなど)は考慮に入れられません。使用者は既定構成設定を選択することができます。



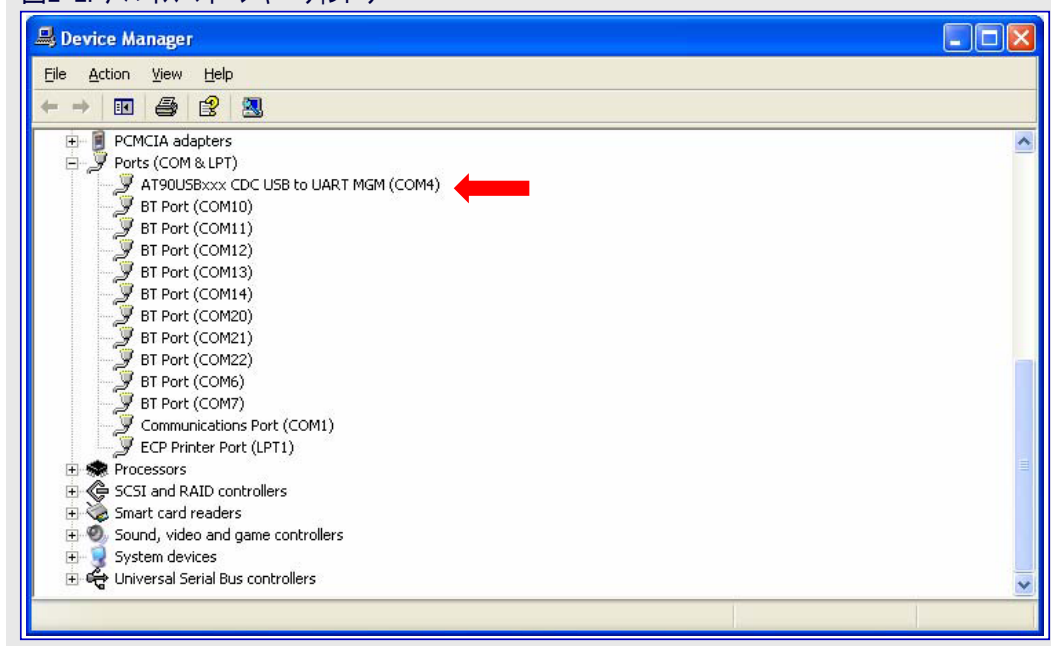
8ビット **AVR**®
マイクロ コントローラ

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. 7801A-06/08, 7801AJ3-01/21

図2-2. デバイス マネージャー ウィンドウ

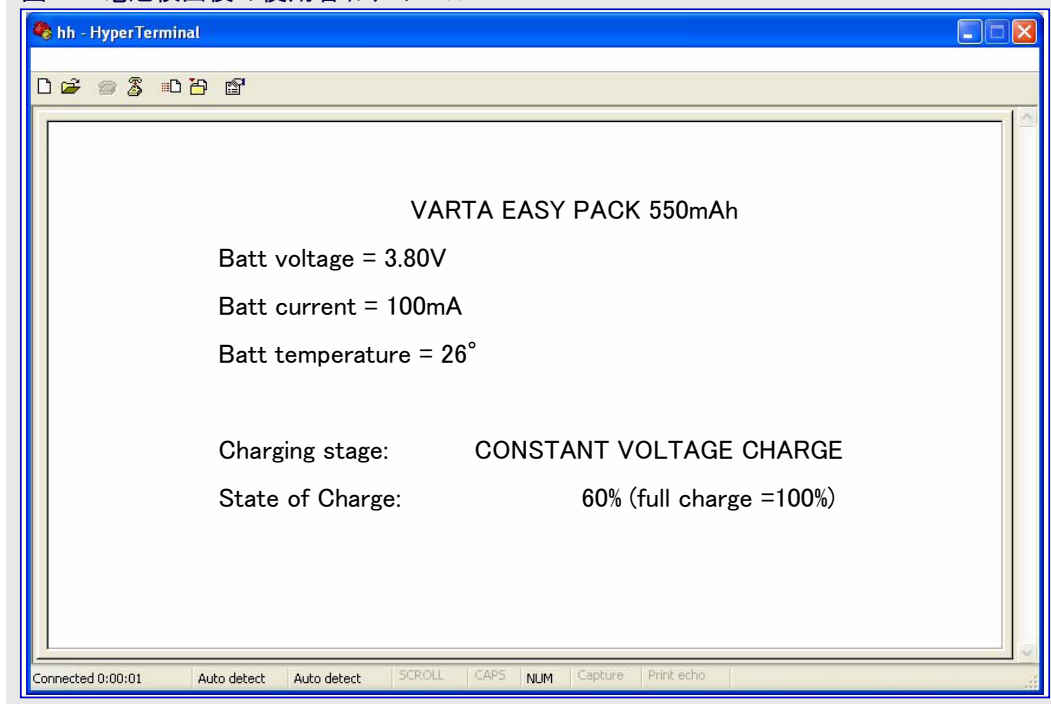


2.1. PCでの充電パラメータ表示

応用はPCで表示されるパラメータの継続的な更新を実行します。検出されて確認された電池がない場合、充電器は開始されません。充電の状態(State)は定電圧充電段階でだけ利用可能です(3.2.3項をご覧ください)。

ハイパー ターミナル走行時、PCへの情報送出を開始するため、使用者はHWPB釦を押さなければなりません。

図2-3. 電池検出後の使用者インターフェース



3. 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。使わ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。

3.1. 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(21頁の「参考文献」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。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。

3.1.1. 安全性

確実な予防処置が充放電時に結合されるなら、リチウム イオン電池は安全です。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池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/NTC(正/負温度係数)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。

電気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。

3.2. Li-Ion電池の充電

リチウムに基く電池を充電するには、1つの方法だけがあります。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。

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、(公称充電電圧到達後)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。充電中に充電時間、電池温度などの多数のパラメータが監視されます。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。

3.2.1. 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われるべきではありません。

3.2.2. 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回目の充電と何も違いありません。

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。

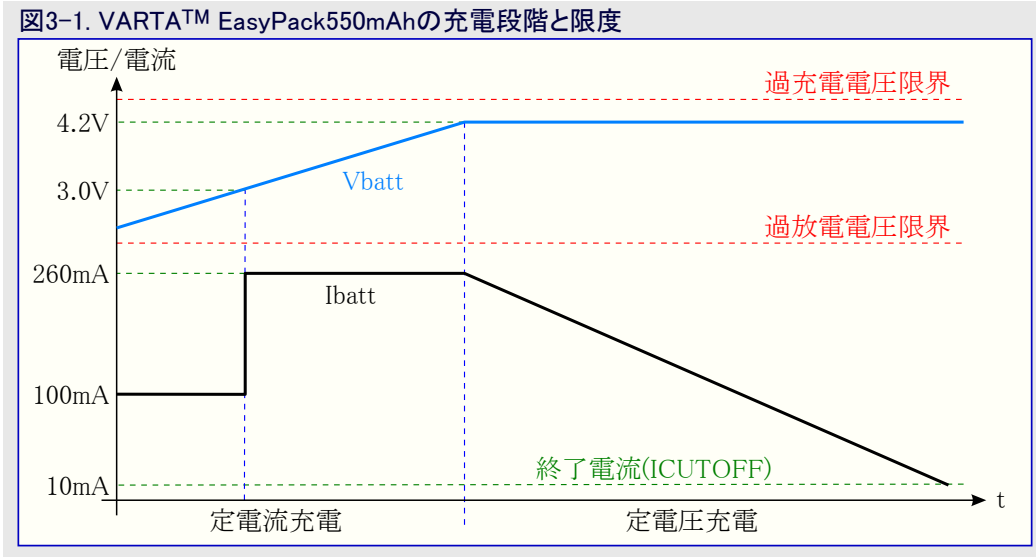
電池は使われるか否かに拘らず、時の経過のために容量を失います。

3.2.3. 充電の段階

リチウム イオン電池は以下の3つの段階に従います。

1. 事前限定電流。リチウム イオン電池の充電は電池電圧の調査で始まります。電圧が定義された閾値(PREQUAL_VOLTAGE)以下ならば、充電は固定された低電流で始めます
2. 定電流。電池に定電流を印加することで充電は続きます。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。
3. 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。

3.3. VARTA™電池

3.3.1. 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。下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3-1. 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 (注)
充電効率	99.9%
充電電流閾値	0.03C (注)
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℃
湿度範囲	65±20% RH

注: Cは代表定格容量値に対応(表3-2をご覧ください。)

3.3.2. 代表的な電池特性

下表はこの応用で使わ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え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3-2. VARTA™ EasyPackリチウム イオン電池に対する製造業者の範囲データ

項目	EZPack S-3.7V	EZPack M-3.7V	EZPack L-3.7V	EZPack XL-3.7V	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) (注1)	3.9	6.8	10	24	kΩ
NTC	10				
β 値 (注2)	3435				K
過充電検出電圧	4.35				V
過放電検出電圧	2.20				

注1: RID:電池内部抵抗が接続された電池の容量を識別します。

注2: β 値は温度公式で使われます。

3.3.3. 電氣的なピン配列

この応用は特別な形式のリチウム イオン電池を使い、ここで提供された全ての構成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 イオン電池も使え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されるのを保証するのは使用者の義務です。

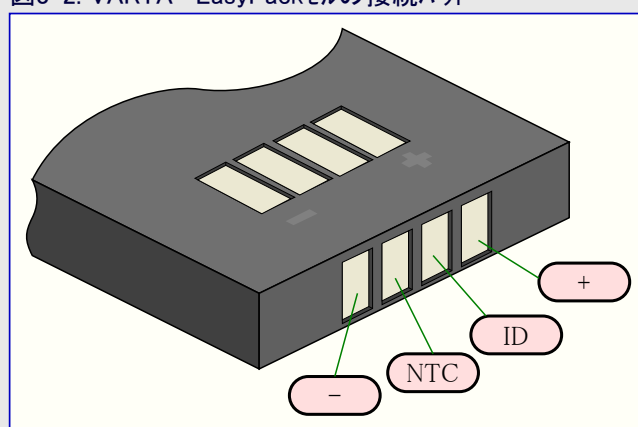
右図はこの応用で使われるリチウム イオン電池の接続パッドを図解します。

電池は次のように電池充電器へ接続されます。

表3-3. 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3-2. VARTA™ EasyPackセルの接続パッド



3.4. VBUS供給電圧

USB給電応用は以下の3つの分類に分かれます。

- ・ **低電力バス**：低電力バス給電機能はVBUSからそれらの電力全てを得て、USB規格に従って1単位負荷(100mA)より多く引き出してはなりません。また4.40V~5.25V間のVBUS電圧で動くことができなければなりません。
- ・ **高電力バス**：高電力バス給電機能はVBUSからそれらの電力全てを得て、構成設定されるまで100mAよりも多く引き出すことができます。一旦形態設計されると、その記述子でこれを要求することによって5単位負荷(500mA)まで引き出すことができます。全負荷に於いて、4.75V~5.25V間のVBUS電圧で動くことができなければなりません。
- ・ **自己給電**：自己給電機能はVBUSから最大100mAと残りを別の供給元から引き出すことができます。

EVK527給電と電池充電のための電流はVBUSから来ます。EVK527は必要ならば充電電流を制限しなければなりません。

簡単な解決策は参照表内のI充電パラメータを変更することです。

例えば、550mAh電池は260mA充電電流を許します。8MHz発振器でのEVK527消費が約10mAであることを知り、このパラメータの(例えば)90mAへの変更は低電力バスでの充電器接続を許します。この場合、事前限度電流も90mAに制限されなければなりません。

3.5. EVK527改訂

EVK527改訂1.0.0は以下のように更新されなければなりません。

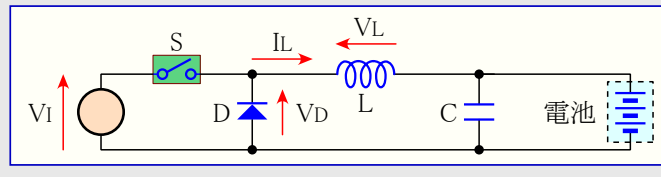
- ・ 差動入力動作形態を使うためにPF0とPF1間に分路抵抗が接続されなければなりません。PF0はPD4を置き換えなければならず、SP6は(既定構成設定と異なる)“半田なし”でなければなりません。
- ・ Q1Aのゲートピンとソースピンは切断されなければなりません。
- ・ R6とR7の新しい値は13k Ω です。
- ・ R3の新しい値は1 Ω です。

第5章で与えられる回路図はこれらの変更を示しません。

3.6. 降圧変換器

降圧変換器は電池電圧と電池電流を制御するためにEVK527に統合されます。切り換え器は高速PWM出力によって制御されます。

図3-3. 降圧変換器回路図



3.6.1. PWM周波数

PWM用のPWM周波数は最大(64MHz)に設定されます。供給元クロックはUSBとPWMの両方で使われるPLL出力(96MHz)です。

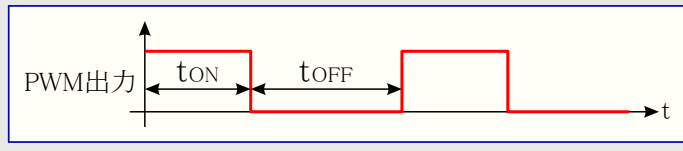
後置分周器はPLL信号用に1.5分周を提供します(96MHz/1.5=64MHz、PLL周波数制御(PLLREQ)レジスタの高速タイマ用PLL後置分周器選択(PLLTMI,0)をご覧ください)。

PWM出力信号の結果は250kHzの周波数です。

$$64\text{MHz}/256=250\text{kHz}$$

ここで256はタイマ/カウンタ4で使われるタイマ/カウンタ4比較A(OCR4A)レジスタのビットでの大きさです。

図3-4. PWM出力信号



ソフトウェアはPWM出力のデューティサイクルを変更することで電池電圧/電流を制御します。 t_{ON} が増加すると、電池電圧/電流はより重要なエネルギー負担を受け取ります。

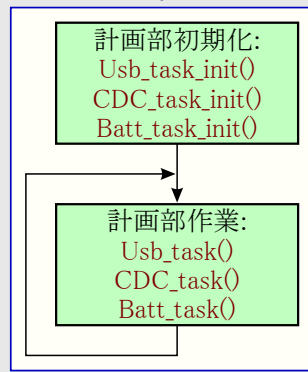
4. 電池充電器ソフトウェア

4.1. 計画部

計画部は定義された作業を漠然と呼び出すために実装されます。この無限繰り返しを開始する前に初期化関数が呼ばれます。

3つの作業があります。各作業は直前のものの最後の後で呼ばれます(先取りなし)。

図4-1. 計画部



4.2. ファイルの一覧

ファームウェアはIAR SystemsのEmbedded Workbench® 5.10版を使って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4-1. CDC応用のためのプロジェクト ファイル (IAR EWの作業空間ファイルをご覧ください。)

ファイル名	形式	注記
cdc_task.c	Cソース ファイル	CDC作業とCDC作業初期化関数
cdc_task.h	ヘッダ ファイル	
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	
power_drv.c	Cソース ファイル	電力管理低位ドライバ
power_drv.h	ヘッダ ファイル	
scheduler.c	Cソース ファイル	計画部ルーチン
scheduler.h	ヘッダ ファイル	
start_boot.c	Cソース ファイル	ブート関数
start_boot.h	ヘッダ ファイル	
time.c	Cソース ファイル	タイミング用関数
time.h	ヘッダ ファイル	
uart_lib.c	Cソース ファイル	このファイルは最小VT100端末アクセスを提供します。
uart_lib.h	ヘッダ ファイル	
uart_usb_lib.c	Cソース ファイル	UART USB関数
uart_usb_lib.h	ヘッダ ファイル	
usb_descriptor.c	Cソース ファイル	応用を識別するUSBパラメータ
usb_descriptor.h	ヘッダ ファイル	
usb_device_task.c	Cソース ファイル	USB装置制御部
usb_device_task.h	ヘッダ ファイル	
usb_drv.c	Cソース ファイル	USBドライバ ルーチン
usb_drv.h	ヘッダ ファイル	
usb_standard_request.c	Cソース ファイル	USB装置列挙(接続認証)要求
usb_standard_request.h	ヘッダ ファイル	
usb_specific_request.c	Cソース ファイル	使用者呼び戻し関数
usb_specific_request.h	ヘッダ ファイル	
usb_task.c	Cソース ファイル	USB作業とUSB作業初期化関数
usb_task.h	ヘッダ ファイル	

表4-2. 電池単位部用プロジェクト ファイル (IAR EWの作業空間ファイルをご覧ください。)

ファイル名	形式	注記
ADC.c	Cソース ファイル	A/D変換器に関連する関数
ADC.h	ヘッダ ファイル	
Batt_task.c	Cソース ファイル	電池作業と電池作業初期化関数
Batt_task.h	ヘッダ ファイル	
battery.c	Cソース ファイル	電池特有定義と、電池制御とデータ取得に関連する関数
battery.h	ヘッダ ファイル	
chargefunc.c	Cソース ファイル	充電関数
chargefunc.h	ヘッダ ファイル	
LIIONcharge.c	Cソース ファイル	リチウム イオン電池用充電状態関数
LIIONcharge.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	
PWM.c	Cソース ファイル	パルス幅変調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ファイルで定義された状態に関連する関数
statefunc.h	ヘッダ ファイル	

4.3. 概要

ファームウェアはリチウム イオン電池を充電するのに必要とされる全ての関数を統合します。

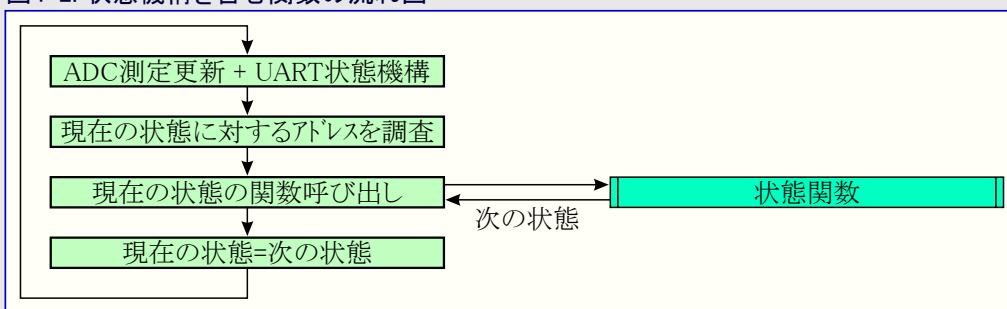
表4-3. ファームウェアのメモリ必要条件

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)	13900
	DATA (SRAM)	1109
	XDATA (EEPROM)	2

4.4. 状態機構

状態機構は電池作業内に実装されます。この状態機構はかなり簡単で関数ポインタを使います。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4-2. 状態機構を含む関数の流れ図



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4-4. 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	関連する関数	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
PREQUAL_CTRL	Charge()	PREQUALの終了待ち
SLEEP	Sleep()	低消費電力動作形態
CCURRENT	Charge()	定電流で充電
CCURRENT_CTRL	Charge()	CCURRENTの終了待ち
CVOLTAGE	Charge()	定電圧で充電
CVOLTAGE_CTRL	Charge()	CVOLTAGEの終了待ち
ENDCHARGE	Charge()	充電成功終了
DISCHARGE	Discharge()	BATCON状態へ(将来実装用準備)
ERROR	Error()	可能ならば、異常を解決

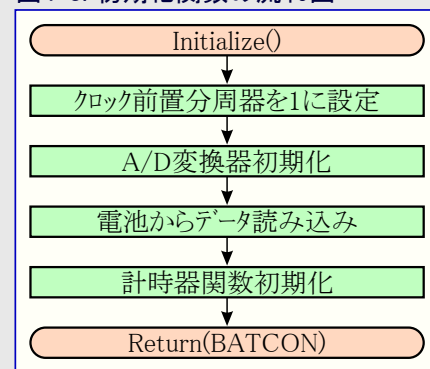
状態関数は以降の項で記述されます。

4.4.1. 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。

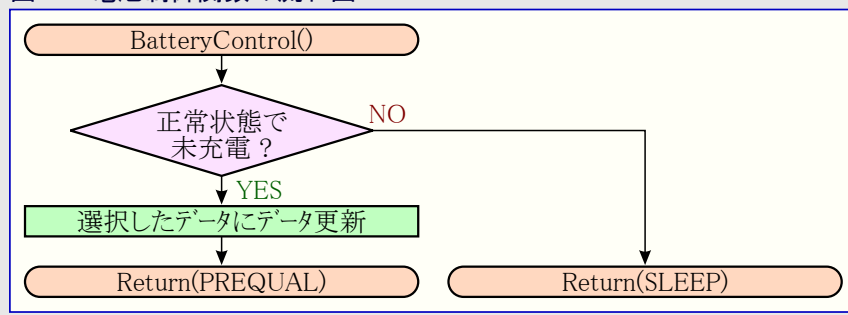
図4-3. 初期化関数の流れ図



4.4.2. 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て、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かめるために調べます。プログラムの流れは下図で図解されます。

図4-4. 電池制御関数の流れ図



4.4.3. Charge()

充電関数は段階に分けられた充電算法を含みます。この応用に関して、それは4つの段階を持ちます。

- ・ 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以下に落ちるか、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決める処です。

ChargeParametersとHaltParametersはこの関数の鍵変数です。各段階後、関数は次に望む状態をbatt_task()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4-5. 充電状態関数の流れ図

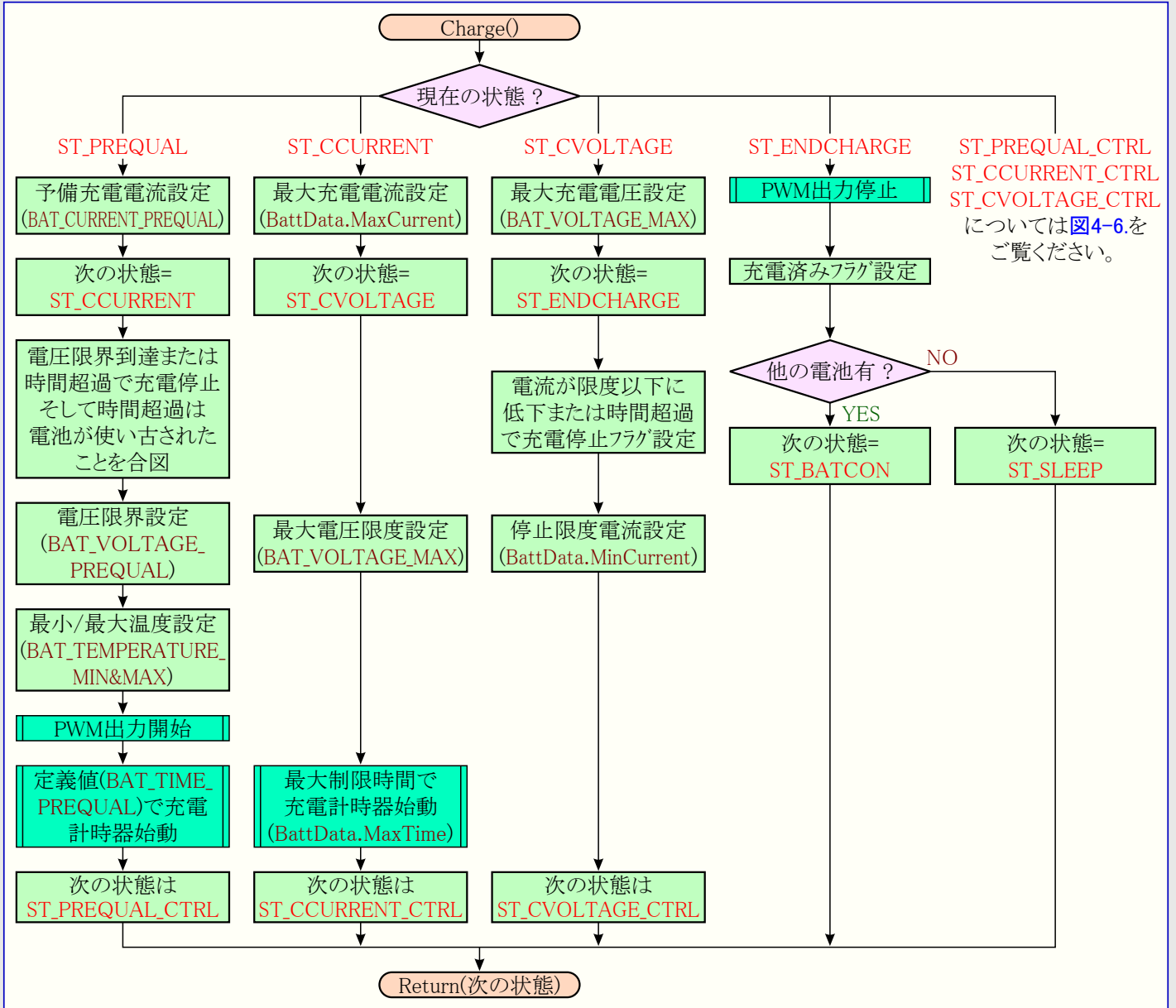
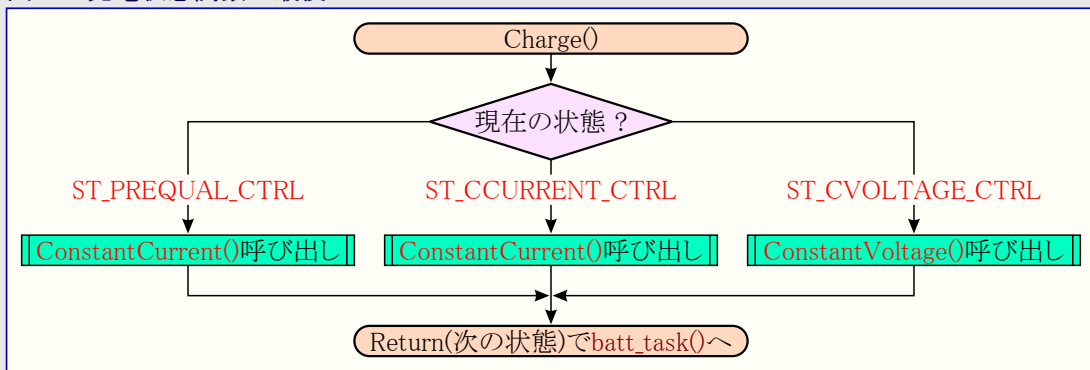


図4-6. 充電状態関数の最後



4.4.4. Discharge()

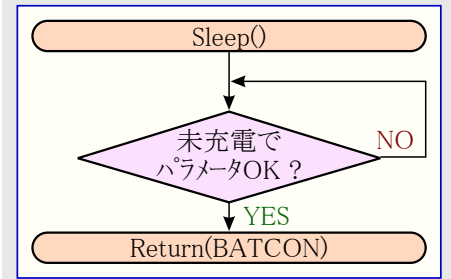
この関数は実装されていません。

4.4.5. 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4-7. 休止関数の流れ図

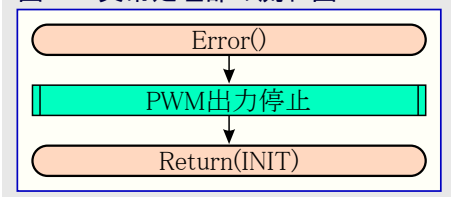


4.4.6. Error()

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。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右図で図解されます。

図4-8. 異常処理部の流れ図



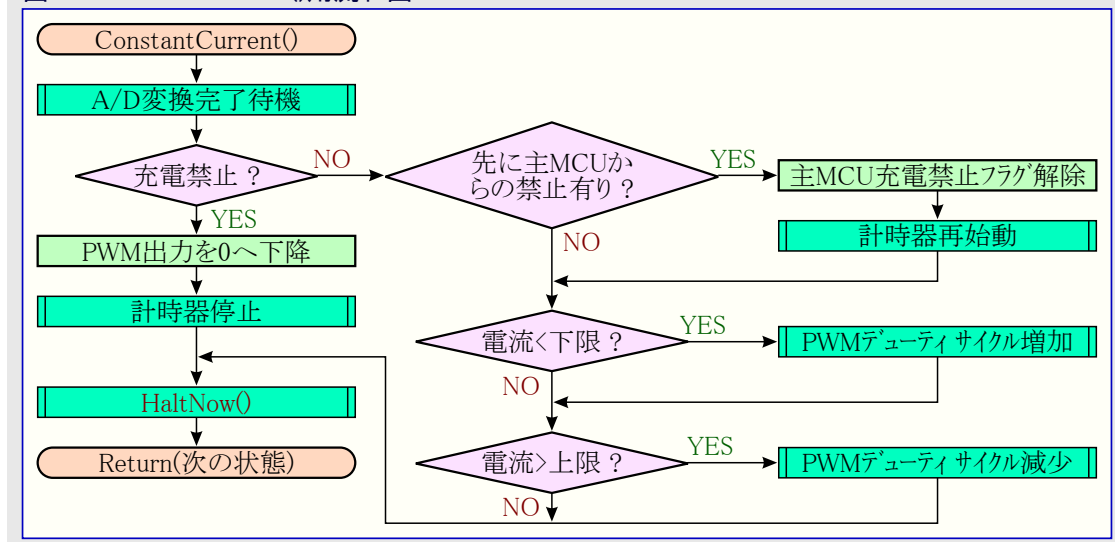
4.5. 充電制御関数

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。

4.5.1. 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変数を使います。

図4-9. ConstantCurrent()用流れ図



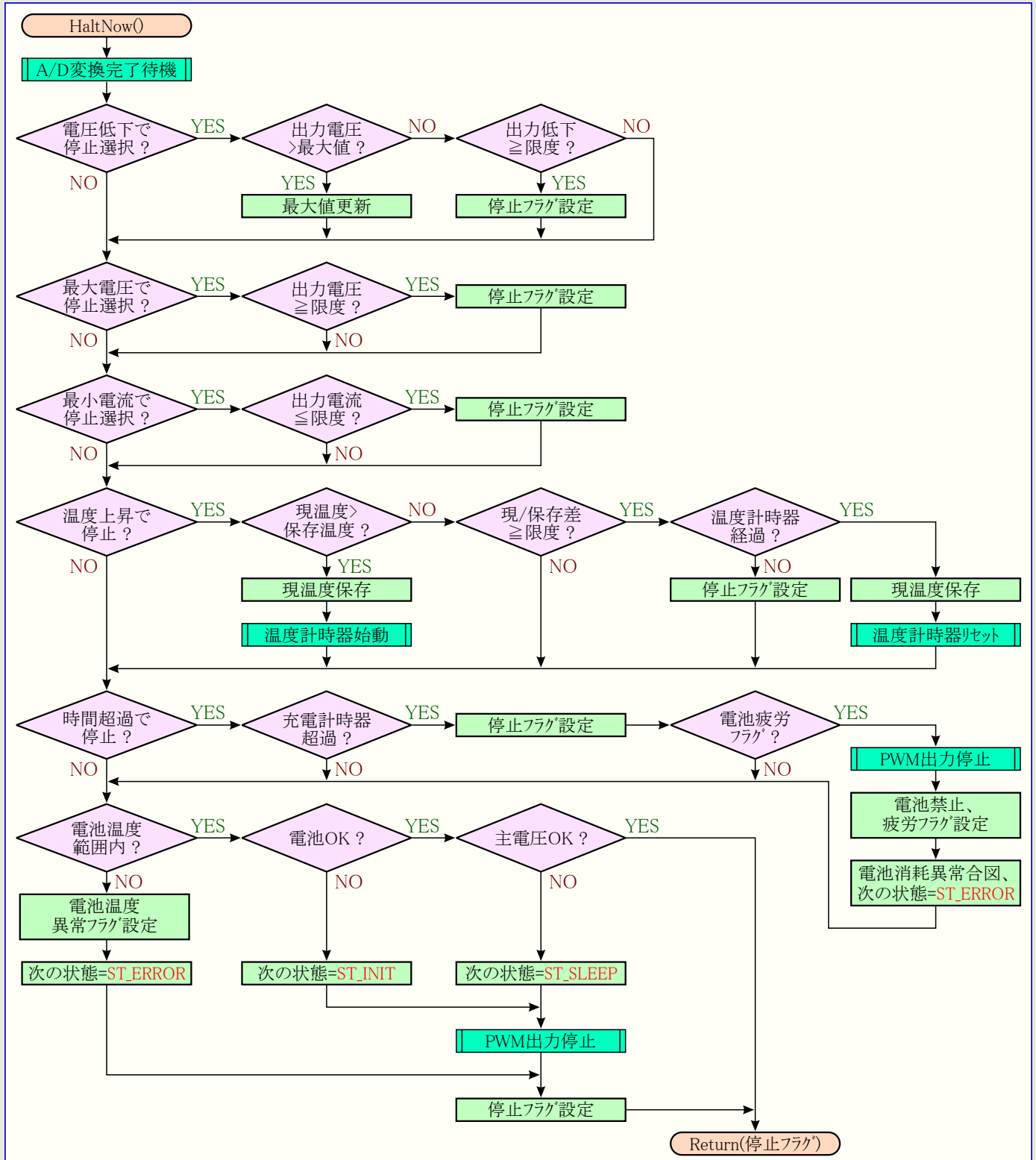
4.5.2. 充電停止判定

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()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。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4-10. HaltNow()用流れ図



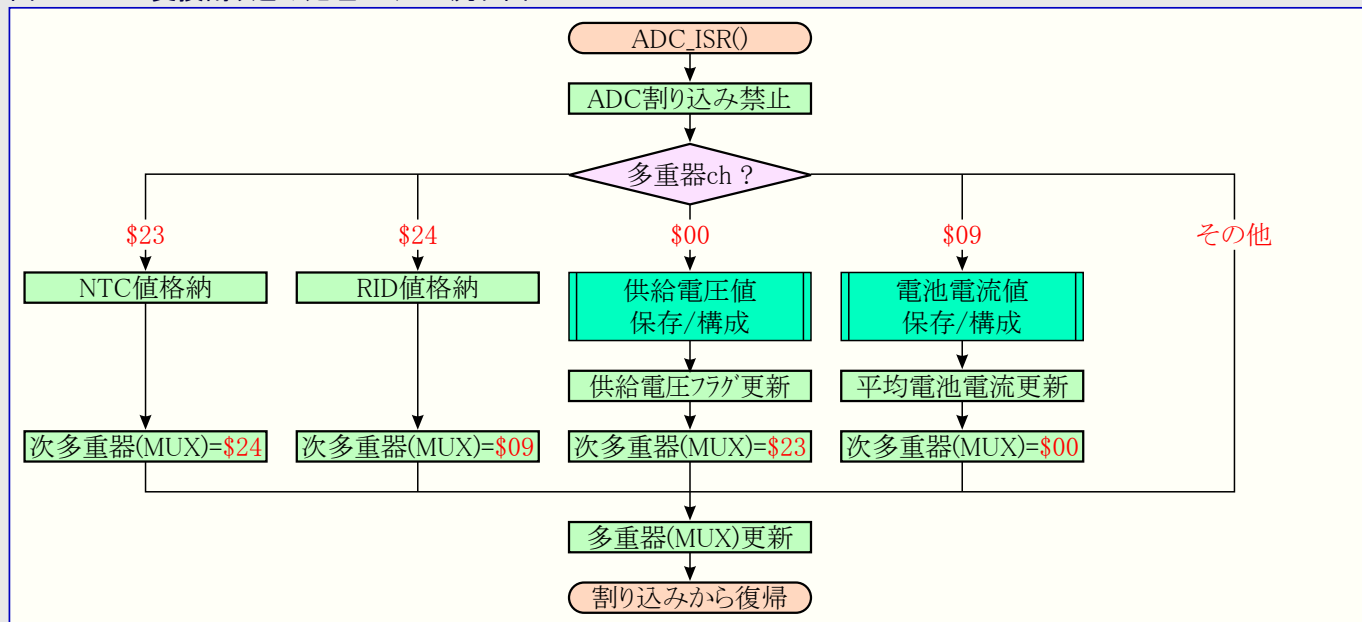
(訳注) 原書の図5-4.~7.は図4-10.として纏めました。

4.6. その他の関数

4.6.1. 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。全ての多重器(MUX)値について、A/D変換基準電圧は2.56V内部基準電圧です。

図4-11. A/D変換割り込み処理ルーチンの流れ図



4.7. 実装

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

4.7.1. 構成設定

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。更なるプログラム定数については**battery.h**ファイルをご覧ください。

表4-5. 電池に関連するコンパイル時定数 (battery.hソース ファイルをご覧ください。)

ラベル名	説明
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	充電されるべきセルでの電圧。
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	充電電流ヒステリシス。
BAT_VOLTAGE_HYST	充電電圧ヒステリシス。
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疲労と記されます。
BAT_VOLTAGE_HYST	予備検定(充電)段階で費やされる最大時間。
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。さもなければ充電は停止されます。
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[]	温度参照表。

4.7.2. コンパイル

IARとGCCの両方のプロジェクトが利用可能です。GCCプロジェクトは外部makefile(¥gcc¥default内のMakefleをご覧ください。)またはAVR Studioプロジェクトで定義される任意選択を使うことができます。

表4-6. コンパイラ構成設定

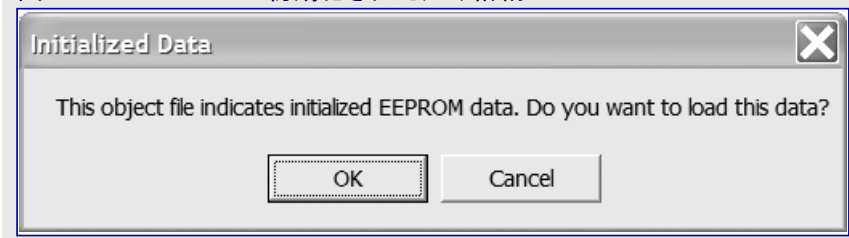
分類	ターゲット	領域	値
General Options	Target	Processor configuration	ATmega16/32U4
		Memory model	Small
	System	Data stack	0x100
		Return address stack	32
		Enable bit definitions ...	なし
C/C++ Com+iler	Language	Require prototypes	チェック
Linker	Output	Format	Other: ubrof8
	Extra Options	Command Line	-y(CODE) -Ointel-extended,(DATA)=\$EXE_DIR\$¥\$PROJ_FNAME\$_data.hex -Ointel-extended,(XDATA)=\$EXE_DIR\$¥\$PROJ_FNAME\$_eeprom.hex

4.7.3. プログラミング

コンパイルしたコードはAVR Studio®とJTAGICEmk 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。

必要ならばコンパイルされたコードはEEPROMデータを含む準備が整っていることに注意してください。この特徴は更なる開発専用です。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4-12. 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。既定工場構成設定から外れるヒューズ設定が右図で一覧にされます。

EVK527改訂1.0.0ではJTAGピンとジョイスティック鉤が同じ入出力を共用します。これはCDCキー押下応用取り外しのためです。

ハイパーターミナルへのデータ送出を開始するのにHWP鉤が使われます。

AVR Studioでのソフトウェア書き込み後、ATmega16/32U4ブートローダーが消去されます。FLIP(AtmelのISP)を使うことによる書き込みが必要とされる場合、(AVR Studioで)ブートローダーソフトウェアの書き込みが必要とされます。

表4-7. 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. EVK527改訂1.0.0回路図

図5-1. 基板改訂1.0.0に対応する回路図改訂3.0.0 (1/5)

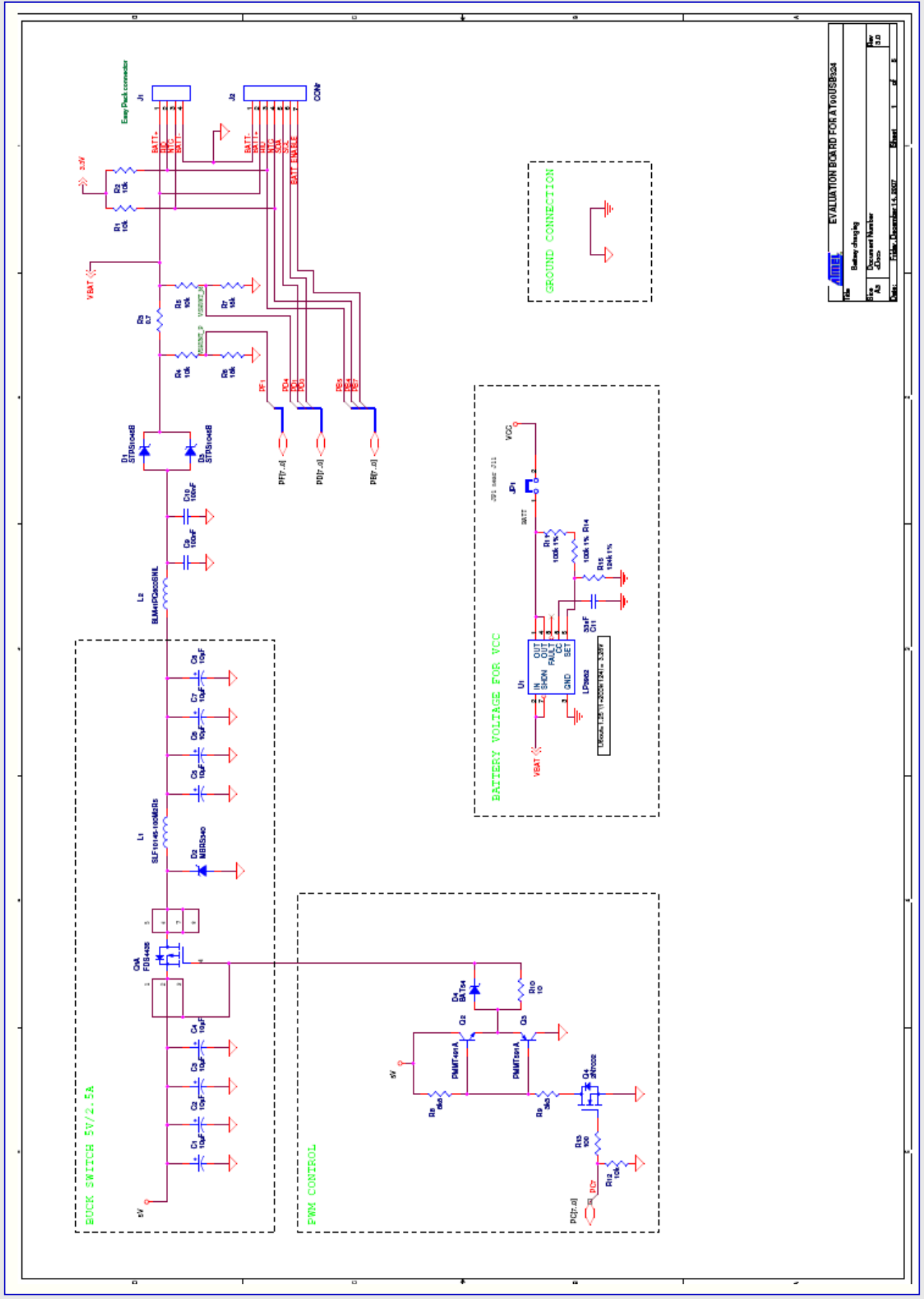


図5-3. 基板改訂1.0.0に対応する回路図改訂3.0.0 (3/5)

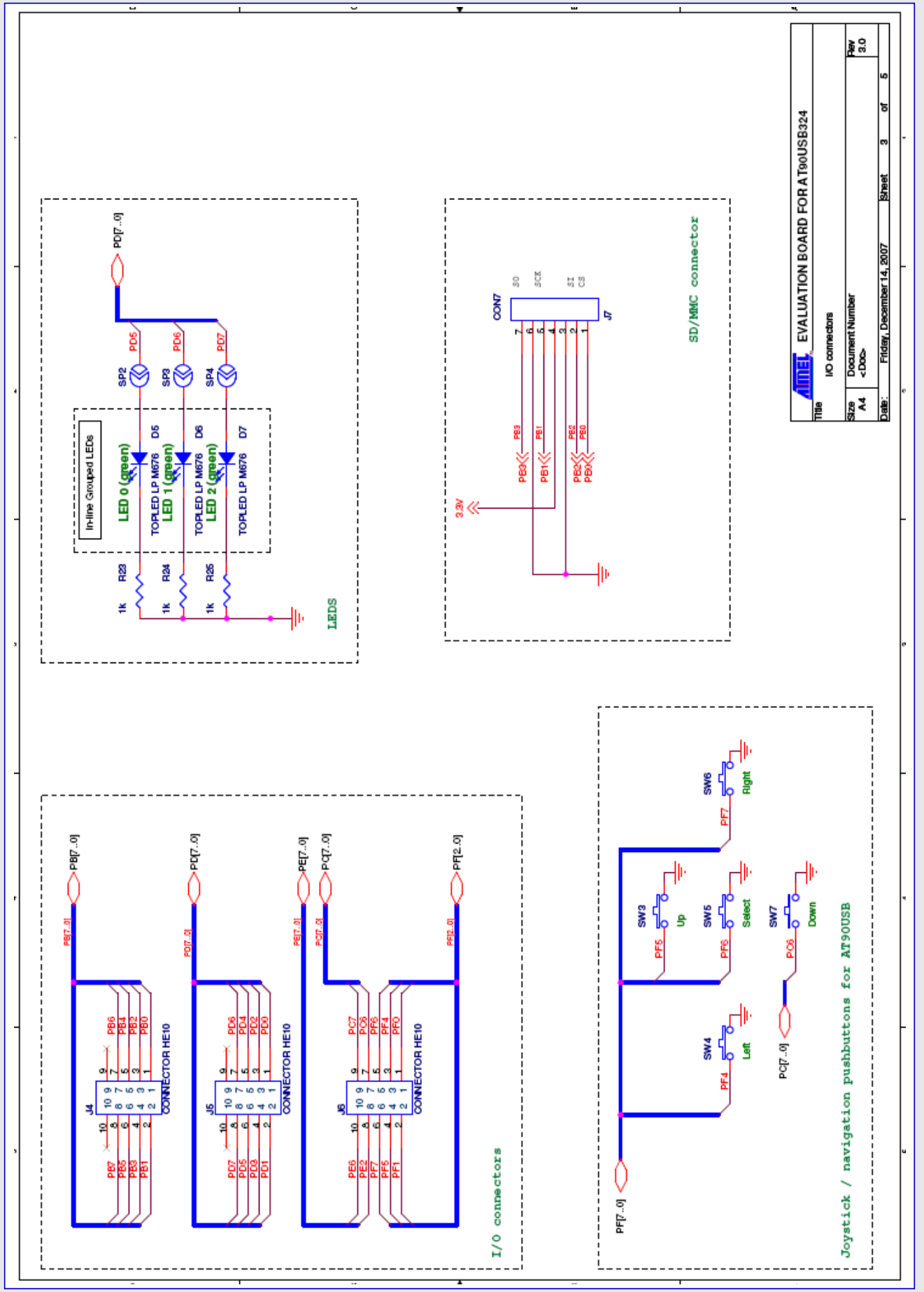


図5-4. 基板改訂1.0.0に対応する回路図改訂3.0.0 (4/5)

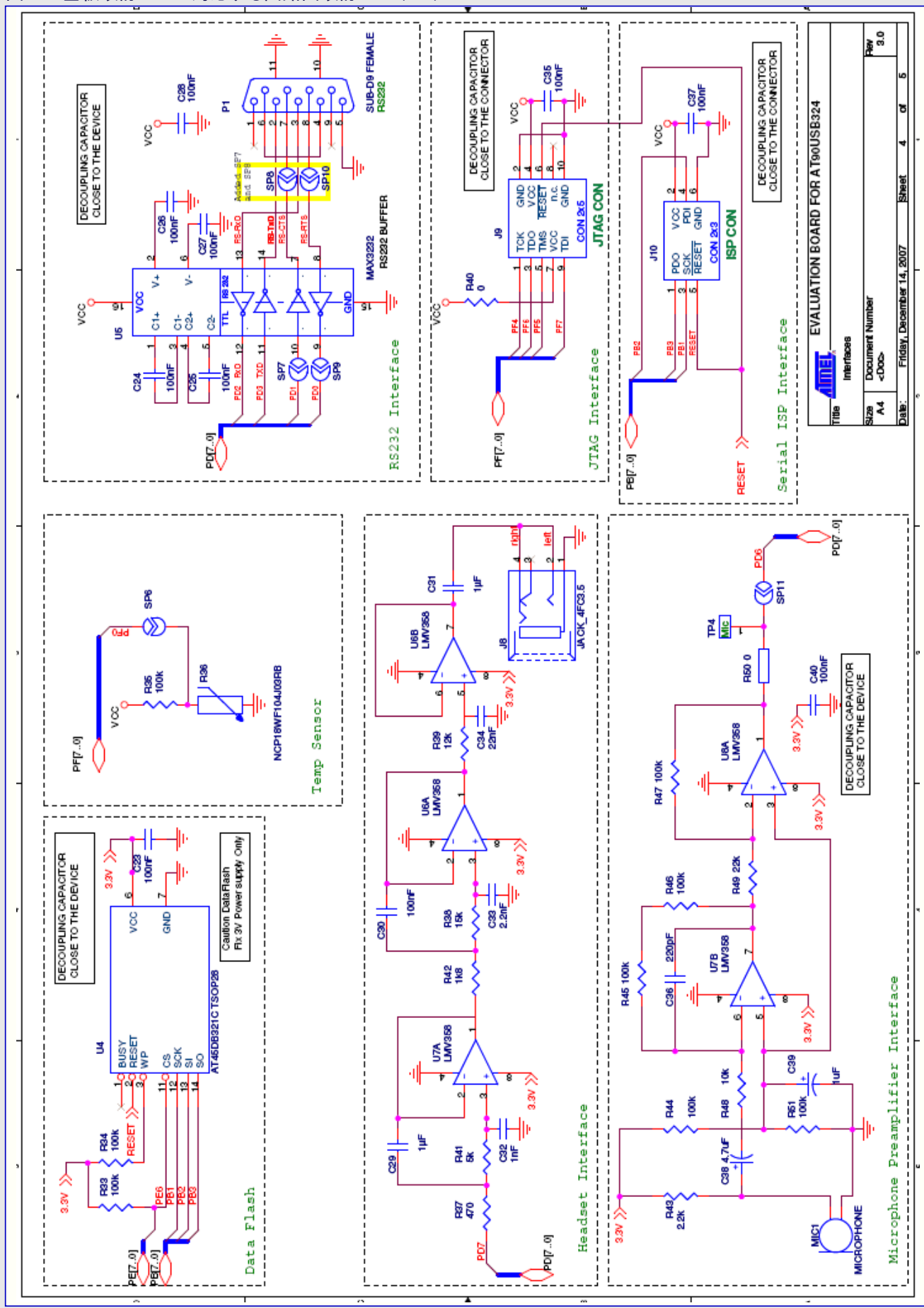
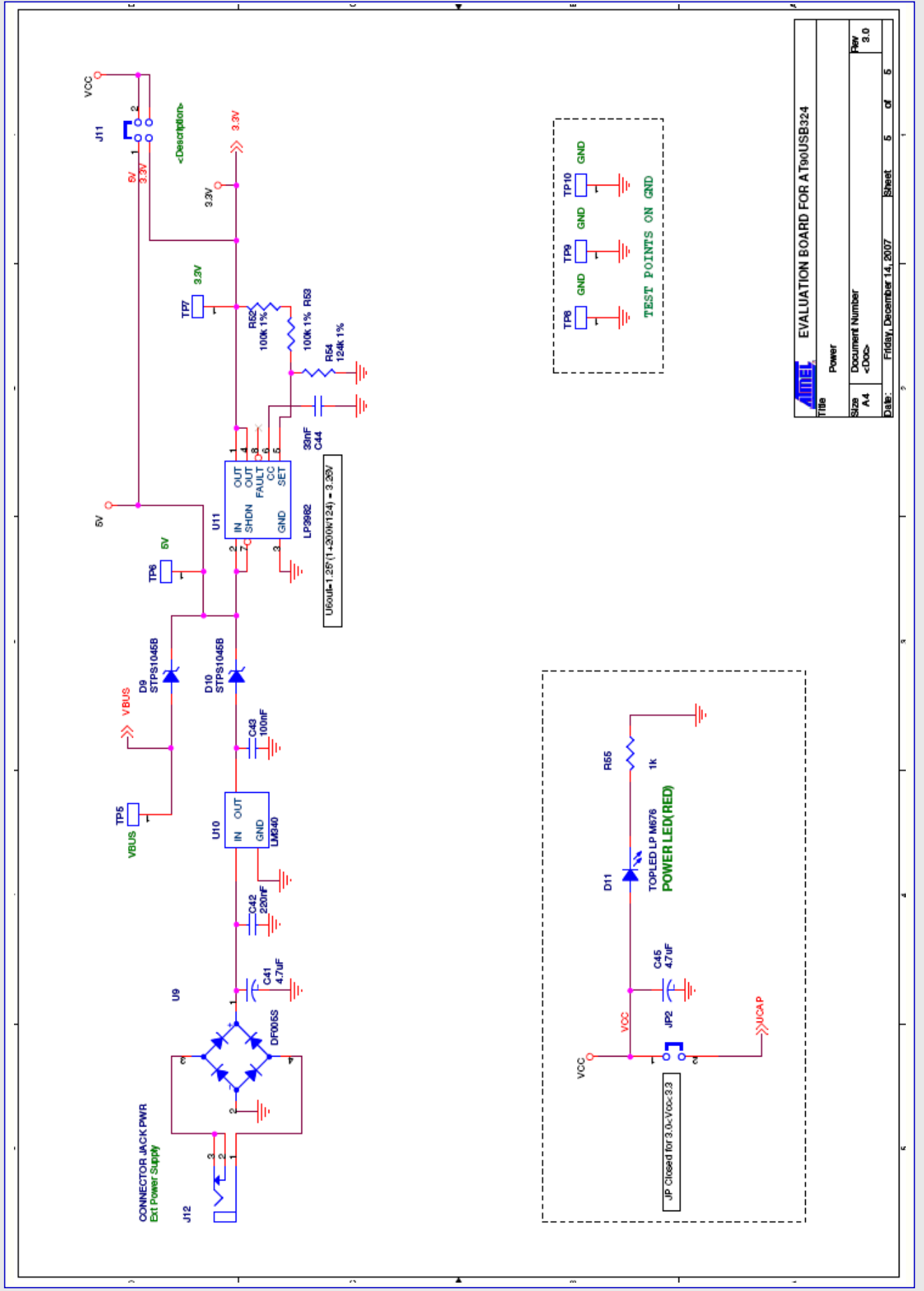


図5-5. 基板改訂1.0.0に対応する回路図改訂3.0.0 (5/5)



6. 参考文献

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™ EasyPack"データシート
550mA : http://www2.varta-microbattery.com/en/mb_data/documents/promotion_varta_easypack/688819.pdf
750mA : http://www2.varta-microbattery.com/en/mb_data/documents/promotion_varta_easypack/688820.pdf
1000mA : http://www2.varta-microbattery.com/en/mb_data/documents/promotion_varta_easypack/688821.pdf
2000mA : http://www2.varta-microbattery.com/en/mb_data/documents/promotion_varta_easypack/688822.pdf
5. "ATmega32U4"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. 不許複製 Atmel®、ロゴとそれらの組み合わせ、AVR®、AVR Studio®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 2021.

本応用記述はAtmelのAVR146応用記述(doc7801.pdf Rev.7801A-06/08)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。