

簡易充放電器の製作

1	はじめに	3
2	電池の特性	3
2.1	ニッケル水素電池	3
2.1.1	充電特性	3
2.1.2	充電効率	4
2.1.3	放電特性	4
2.1.4	放電率	5
2.1.5	放電終止電圧	6
2.1.6	放電深度と容量回復率	6
2.1.7	ニッケル水素電池のメモリー効果について	7
2.2	リチウムイオン電池	8
2.2.1	放電レート特性	8
2.2.2	充電方法	10
3	簡易充放電器全体構成	12
3.1	仕様	12
3.2	ブロック図&接続図	13
3.3	H/W	14
3.3.1	充電部	14
3.3.2	放電部	17
3.3.3	電流測定回路	19
3.3.4	電圧測定回路	20
3.3.5	温度センサー	20
3.4	PSoC S/W	20
3.4.1	PC Interface(HID)	20
3.4.2	各種設定画面	24
3.4.3	システムタイマー	24
3.4.4	AD値の扱いについて	25
3.4.5	温度補正	26
3.4.6	充電制御	27
3.4.7	放電制御	31
4	PC TOOL	32
5	電池評価	33
5.1	ニッケル水素電池	33

5.2	リチウムイオン電池.....	34
5.2.1	充放電特性.....	34
5.2.2	大容量リチウム電池を作ってみる.....	35
6	まとめ.....	37
7	課題と対策.....	37

1 はじめに

近年、地球温暖化が問題となり環境・エネルギーに対してとりわけ関心が高くなってきている。

その中で私たちの生活は電気エネルギーの消費することで成り立っているといっても過言ではない。特に最近携帯電話、デジカメ、ノート PC、モバイルゲームなどモバイルデジタル家電が身の回りにあふれ出し、それに伴いそれらの原動力となる電池の使用が多くなってきた。

最近の”エコ”の観点から何度も繰り返して充電して使用できる二次電池が主流になりつつあるが、電池の特性をよく理解して使用しないと有効に活用することができない。

二次電池は基本的に電気エネルギーを化学反応変化という形で蓄えるわけであるが、“化学反応”は環境温度・湿度、反応回数などの影響で必ず劣化するものである。

私の身の回りでも写真 1-1のようにこれまで使用していた単 3 型ニッケル水素電池が多数あるが、経年劣化と使用頻度それぞれ異なるためどの程度使えるものなるのかをなかなか判断しにくい。中には満充電したつもりでもすぐに電池が無くなってしまいうものもあり、困ったこともある。従って電池の劣化具合を客観的且つ定量的に測定し破棄する時期を把握したいと考えた。



写真 1-1:ニッケル水素電池

同じように携帯電話ではリチウムイオン使用されているが、これも長年使用していると充電周期が短くなってくる。この電池の交換時期も適切に知る方法もあればいいかと思った。

2 電池の特性

2.1 ニッケル水素電池

電池の詳細な内容については文献やインターネット上にあるので多くを割愛するが、ここでは要点だけ。パナソニックにニッケル水素電池の特性について述べたものがあったのでそこから抜粋してみることにする。

2.1.1 充電特性

充電とは、放電した電池を再びもとの使用できる状態に戻す操作です。ニッケル水素電池を充電するには充電器が必要となります。ニッケル水素電池の一般的な充電特性を図 2-1 に示します。ニッケル水素電池は充電を続けると、電池電圧、電池温度および電池内圧が時間と共に変化し、またこれらは充電電流の大きさや周囲温度の影響を受けます。ニッケル水素電池は負極の容量を正極より大きくすることで過充電時に負極から水素ガスの発生を少なくし、かつ正極から発生した酸素ガスを負極に効率よく吸収させることによって電池の密閉化を達成しています。この電池反応が速やかに起こるために、その種類や充電方法によってそれぞれ最適な充電電流値、充電時間が決められています。ニカド電池に比べ、充電反応は発熱反応であり、充電時の温度上昇がより高くなる特長があります。

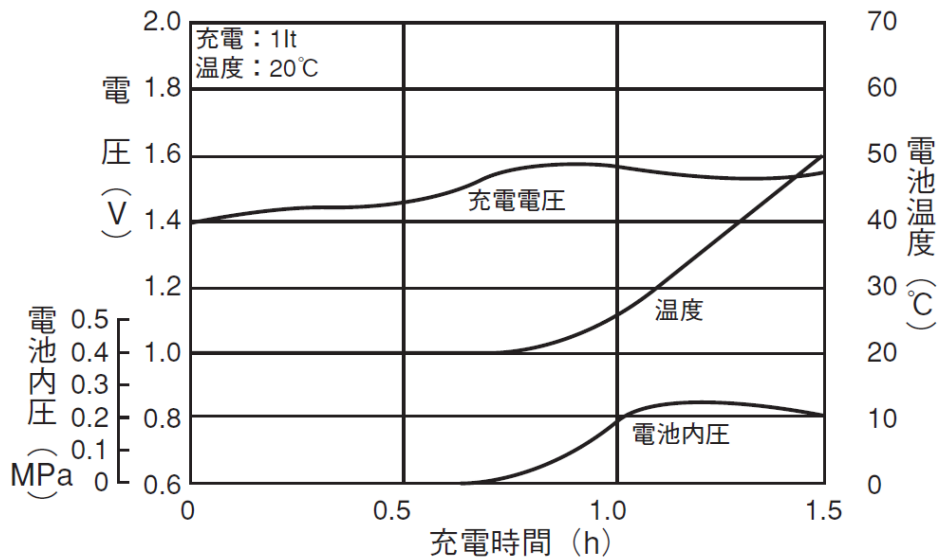


図 2-1: 充電特性の一例

2.1.2 充電効率

電池は通常 $0.1ItA$ の電流で 16 時間、すなわち 160% (完全充電) 充電することで 100% の容量を取り出しています。これは、充電エネルギーが①活物質を充電状態へ変換する反応②ガス発生などの副反応などに使用されるためです。図 2-2 に充電電気量と放電容量の関係を示します。充電エネルギーは充電初期から中期にかけてはほとんど①の反応に使用されますが、充電終期では電池が満充電状態となるため、②の反応が主体となります。ここで ItA とは充電電流 (または放電電流) の大きさを表すもので、電池の定格容量を表した数値の倍数に、 It と電流の単位を付けたものです。例えば、定格容量 1600mAh の電池の場合、 $0.1ItA$ は $0.1 \times 1600 = 160mA$ に相当します。充電効率は電池の種類、充電電流の大きさ、過充電率、周囲温度によって変化します。

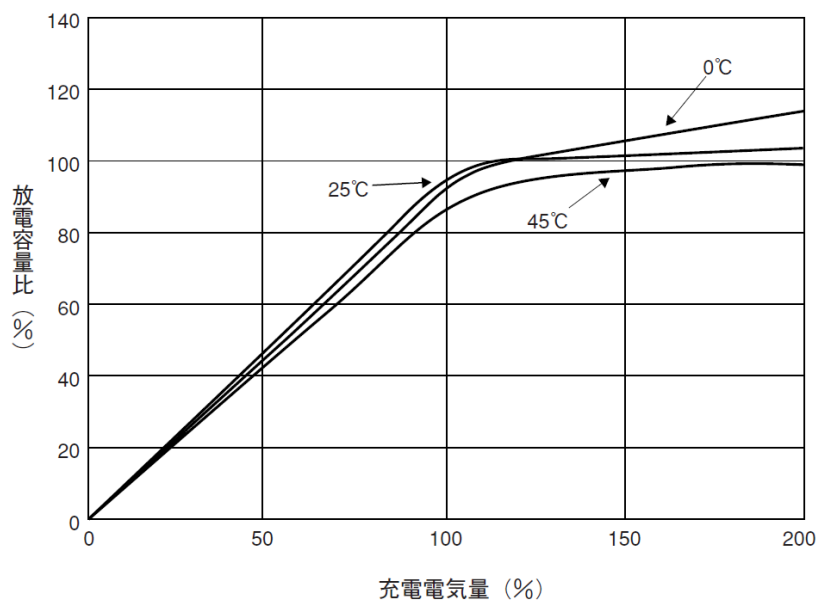


図 2-2: 充電効率の一例

2.1.3 放電特性

放電とは、充電により電池に蓄えられたエネルギーを取り出す操作です。放電における当社ニッケル水

素電池の大きな特徴は、常温はもちろん、低温でも大きなパワー（電流）が取り出せることができ、かつ放電終了間際までそのパワーが持続することです。電池の放電特性は、放電電圧と放電容量（時間）で特徴付けることができます。

図 2-3 にニッケル水素電池と乾電池の放電特性の違いを示します。ニッケル水素電池の放電電圧は平坦性に優れ、放電終期まで安定しています。その電圧は、例えば放電電流が $1It$ の場合、約 $1.2V$ 程度です。これは、電池内の極板抵抗の変化が少ないためです。

放電容量は、放電電流と放電開始から放電終止電圧までの時間の積 (Ah または mAh) で表されます。電池の容量とは、電池を周囲温度 $20 \pm 5^{\circ}C$ の雰囲気下で、 $0.1ItA$ の電流で 16 時間、すなわち 160% 充電 (完全充電) し、1 時間放置した後、 $0.2ItA$ の電流で $1.0V$ まで放電した時の電気量です。

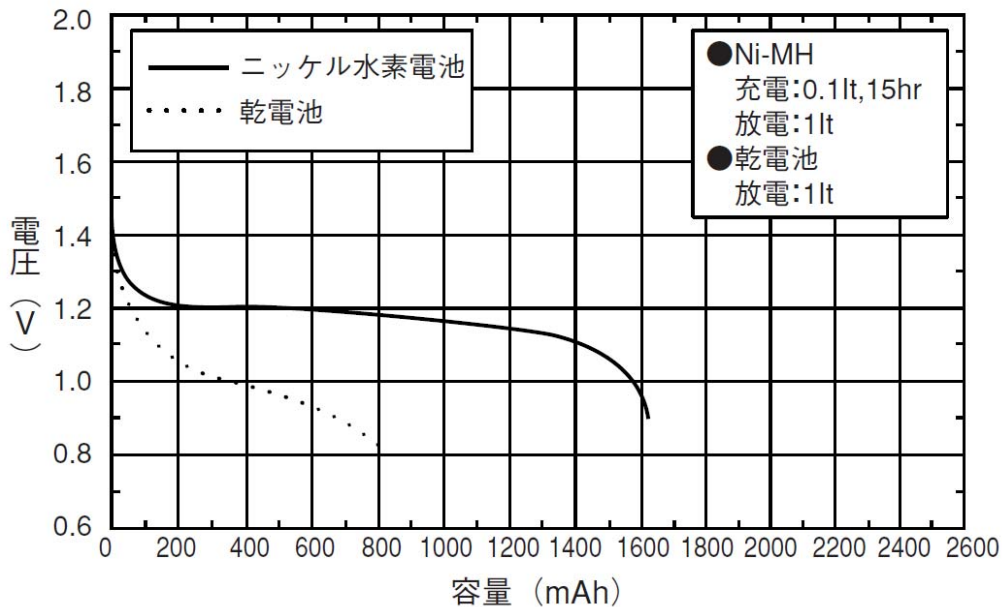


図 2-3: 放電特性の一例 (乾電池との比較)

2.1.4 放電率

図 2-4 に放電電流と放電電圧の関係を示します。一般的に電池は大きな電流で放電すればするほど、取り出せる容量 (放電率) が少なくなります。これは、放電電流が大きくなると極板中の活物質利用率が低下、あるいは電池内部抵抗の増加に伴い電池電圧が低下するためです。図 2-5 に標準タイプの放電電流と放電容量の関係を示します。一般的には放電電流が $2ItA$ を越えるときは、放電容量が著しく低下したり、発熱することがありますので、最大許容放電電流は $2ItA$ を目安にしてください。また、高温耐久性 H タイプでは $1ItA$ を目安にしてください。

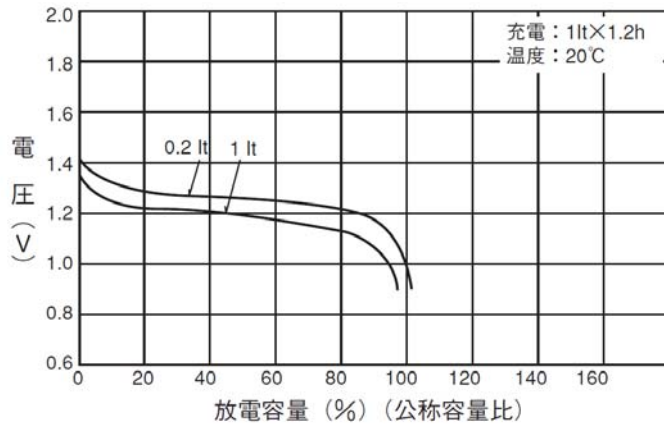


図 2-4: 放電電流と放電電圧の一例

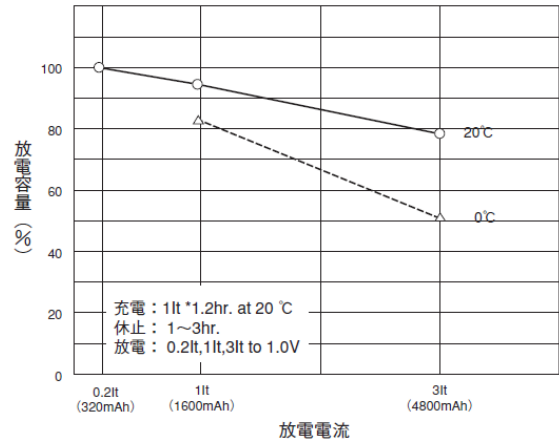


図 2-5: 放電電流と放電容量の一例

2.1.5 放電終止電圧

複数個の電池を直列に接続して放電する場合、深い放電を行うと個々の電池の容量差により、容量の低い電池が過放電され、転極をします。このような転極を繰り返すと放電性能は低下します。また、機器のスイッチ切り忘れ等により過放電をされると、電池性能は劣化します。このように転極に至る深い放電は、電池内部ガス圧力が上昇し、ガス排出弁が作動する結果となります。なお、電池を負荷に接続したままで長期放置しておきますと、電池電圧は 0V となり、ガス排出弁より電解液がクリーニングをおこします。これを防止するために過放電防止機構を設定することが望ましい使用法です。

放電終止電圧は、単セルの場合は 1.0V を目安としてください。また、パック電池の場合は過放電や逆充電による電池性能低下防止のため、次の式を参考にして放電終止電圧を設定してください。

1~6 セル : (電池個数 × 1.0) V
7~20 セル : {(電池個数 - 1) × 1.2} V

2.1.6 放電深度と容量回復率

放電深度とサイクル数の関係を図 2-6 に示します。ニッケル水素電池は、放電深度 (DOD: Depth of discharge) とサイクル数の対数がほぼ反比例の関係になります。ニッケル水素電池の寿命は、正極に使用されている導電剤が放電により還元されることによる正極利用率の低下と、負極で使用されている水素吸蔵合金の腐食による電解液の消費が主要因となっています。サイクル的な用途で使用する場合、DOD は 100% つまり完全放電状態まで使い切ることが一般的です。このとき正極は電気化学容量が残っていないため、負極の電位に近づく事になります。正極に用いられる導電剤は負極の電位に近づくことと還元されてしまい、導電剤としての機能が低下してしまいます。また、放電時に電池容量がなくなるとき、つまり電池電圧が 1V 以下になるときに、負極の電位も一時的に正極の電位に影響されて上昇し、負極の水素吸蔵合金が若干腐食する反応が起こります。また、負極の水素吸蔵合金は充放電により水素の吸蔵放出反応を行うときに結晶の膨張収縮が起こりますが、そのストレスにより、合金が割れていく微粉化反応が起こりやすくなります。従って、これらの反応が起こりにくい、浅い充放電を繰り返した場合、ニッケル水素電池のサイクル寿命は飛躍的に向上します。

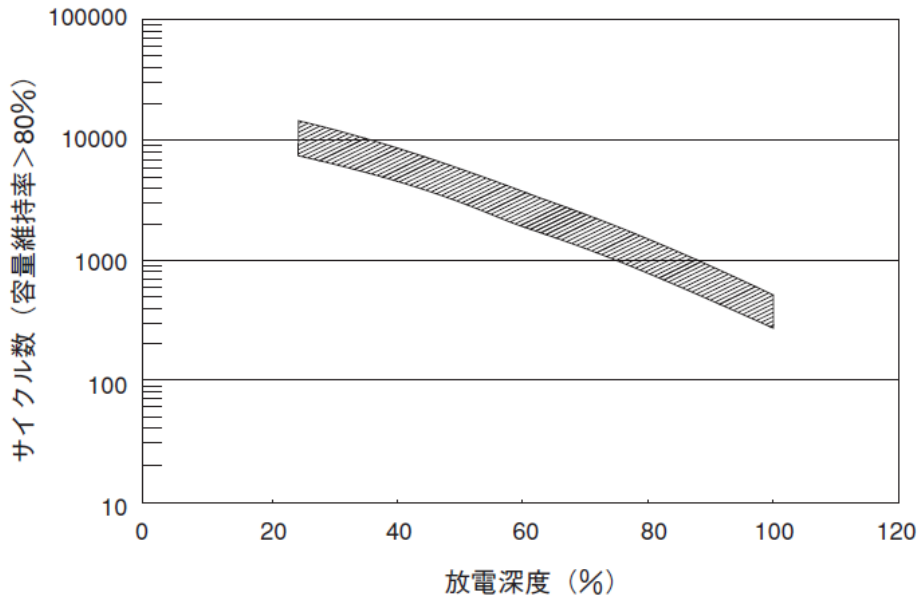


図 2-6: 放電深度とサイクル数の関係

2.1.7 ニッケル水素電池のメモリー効果について

一般的にニッケル水素電池は、電池を使いきらずに充放電(部分放電サイクルと呼ぶ)を繰り返すと、放電電圧が低下する現象が起こります。この現象をメモリー効果(Memory effect)と呼んでいます。電池の部分放電サイクルにより、正極板の活物質である水酸化ニッケルが結晶構造変化を起こすためと言われています。

このメモリー効果による一時的な放電電圧低下については、部分放電サイクルの後に1~2回の深い放電(1.00Vの放電終止電圧)を繰り返すことにより解消します。

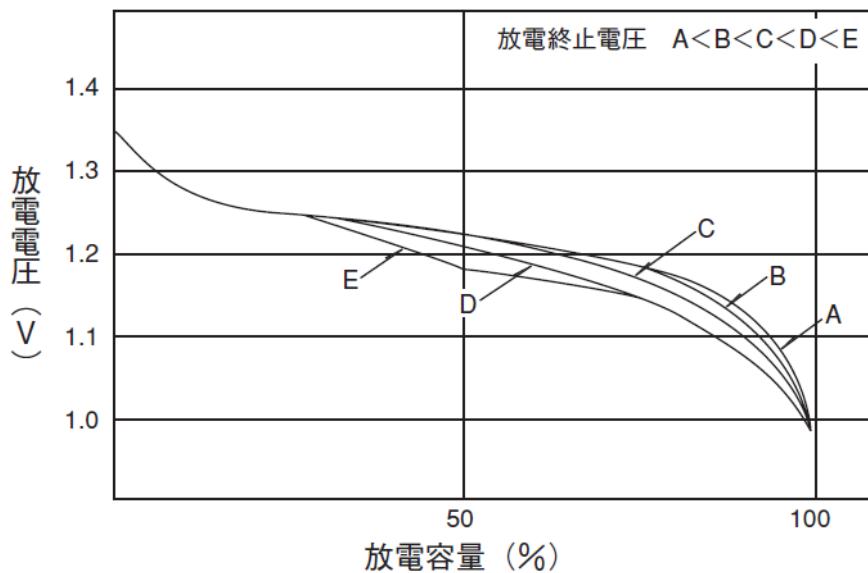


図 2-7: メモリ効果の一例

2.2 リチウムイオン電池

(<http://www.baysun.net/lithium/>から一部抜粋)

2.2.1 放電レート特性

横軸は放電容量もしくは放電時間(定電流放電ですから、結局放電容量を示しています。)、縦軸はセル電圧とし、放電温度は一定で、放電電流をパラメータとしたグラフです。通常示されるグラフは、放電温度 20°Cにおいて、放電レートが 0.2C と 1C です。1C とは公称容量値の容量を有するセルを定電流放電して、ちょうど 1 時間で放電終了となる電流値のことで、たとえば 2.2Ah の公称容量値のセルでは 1C=2.2A です。0.2C は公称容量値の容量を有するセルが 5 時間で放電終了となる電流値で、上述の 2.2Ah の公称容量値のセルでは 0.44A です。0.2C は電池メーカーが電池容量を規定するのに一般的に使う電流値です。電池は大きさによって容量値が決まり、電池特性は電流の絶対値ではなく、容量値に対する相対的な大きさで決まります。したがって、電池特性を示すには C 値(放電レートを示します。)を用いた相対表示が便利で、使われています。

放電レートが大きくなると電池の電圧が低下していきます。これは基本的にはセルの内部抵抗による電圧のドロップです。

放電レート特性

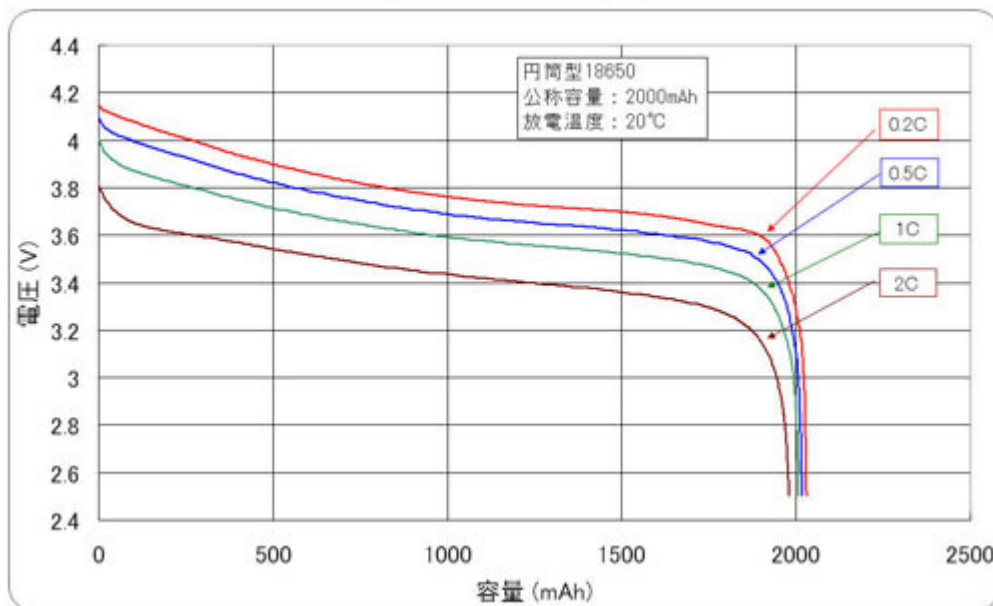


図 2-8:リチウム電池放電レート特性一例

(2) 放電温度特性

図 2-8の横軸は放電容量もしくは放電時間、縦軸はセル電圧とし、放電電流は一定で、放電中の環境温度をパラメータとしたグラフです。放電温度によらず、充電は 20°C で満充電を行っています。

通常は、放電レートが 0.2C で、温度が 20°C と、仕様書で規定している放電温度範囲の上限と下限値でのデータを示すのが一般的です。温度が低温になると、セル電圧は低下していきます。高温側は 20°C のデータとほとんど同じです。

低温側で電圧が低下していく原因を簡単に言うと、放電中のリチウムイオンの移動が低温では動きにくくなり、これはセルの内部抵抗が上昇したことになります。このため、電圧ドロップが増加して電圧が低

下するものです。

放電レートでの説明もあわせて考えると、リチウムイオン電池は、低温でハイレート(大電流)放電のときに電圧ドロップが大きくなります。通常、セル単体では放電終止電圧を 2.5V とすることが多いのですが、しかし、 2.5V から電池パックが放電を禁止して放電を止めてしまう 2.3V まではほとんど容量がなく、大電流で使用していれば非常に短時間で電池パックが放電を止めてしまいます。このため、リチウムイオン電池を使用するセットでは、セット側で電池パックの電圧が **3V程度** になったときに電池残量が 0 であると判断するのが一般的です。(携帯電話では 3.2V を放電終止電圧と見る人が多いようです。) したがって、低温側でセル電圧が全体に低下していくと使える電池容量は大きく減少していきます。

このため、セルの選定に当たってはセットが使用される最低電圧で、かつセットが必要とする最大電流で、電池容量がどうなるかを見なければなりません。しかし、ここでマージンを大きく見ると、電池が大きくなり、コストも、サイズも、重量も増大していきますので、慎重な検討が必要です。

一般的にはセル特性はあまり詳細なデータまでは公表されていませんが、必要であれば、特別な環境温度や放電電流のデータを取得することになります必要があります。特に、電流に関しては通常、一定電流で放電させることはなく、時間とともに変化する電流パターンを有しています。その電流パターンをシミュレートして、電池の使用可能時間を見積もることもできますべきです。

放電温度特性

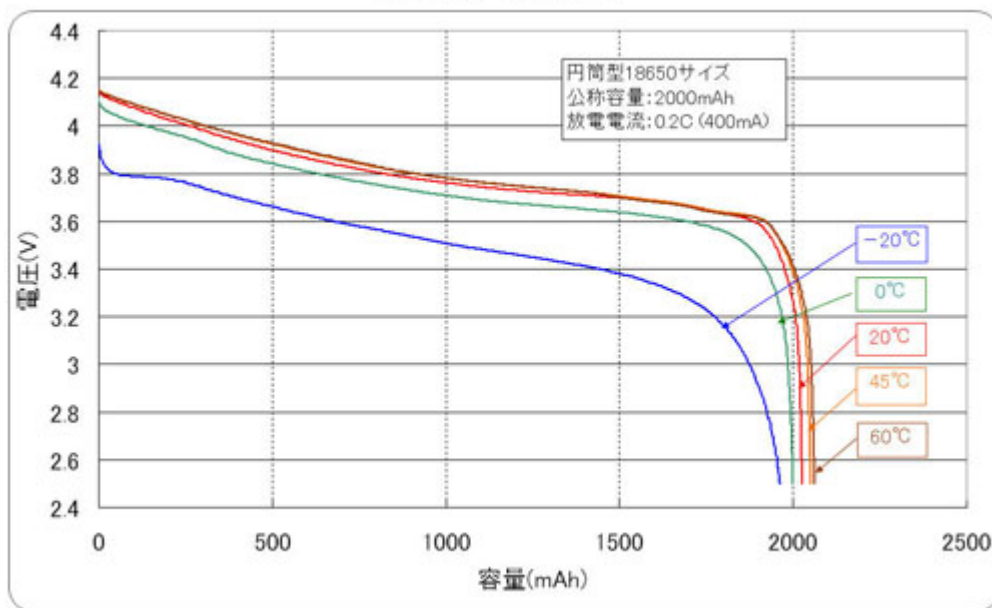


図 2-9: リチウムイオン電池放電温度特性

上の図 2-9において、 -20°C の放電パターンの放電開始直後に、電圧が急激に低下し、その後しばらく持ち直す挙動を示しています。これは温度が低いほど、また、電流が大きいほど顕著に現れる現象です。電圧低下が止まる原因は、放電電流によって、セル内部で抵抗損による自己発熱があり、これによりセル温度が上昇することによって、電圧値が上昇するためです。したがって、これは連続放電を行っている場合の特性を示しています。間欠的な放電では挙動が変わってきます。

-20°C で放電をさせることは、一般的には多くある状況ではありません。特にわが国における都市圏に

においては -20°C という環境温度はまずありえません。何故、低温特性が着目されるかという理由は、筆者としては次のように考えています。低温特性は特にいろいろな電池メーカーで特性が異なるものです。一般的には低温特性がよいほど、セルの特性がよいといわれています。すなわち、低温での放電の際の電圧ドロップが小さいということは、セルの抵抗値が小さく、大電流放電させた際のセルの温度上昇が小さく、長時間連続的に大電流放電を行えるセルであることを示しています。

(3) サイクル特性

電池は充放電を繰り返せば、当然、劣化していきます。通常はその程度を示すために、サイクル特性として、室温で、1C 充電と1C 放電を繰り返したときの容量値の推移を見ます。充放電サイクルにより、セルの内部抵抗も増大していくため、単純に容量が低下するだけでなく、放電レート特性や放電温度特性についても変化していくことが推定されますが、それはサイクル後のサンプルを実測するしか方法はないようです。

セルのメーカーがカタログ等で公表しているデータは多くの場合、室温で1C 充電、1C 放電のサイクルを繰り返したときの容量を示しています。

メーカーは通常 0.2C 放電のときの容量で仕様書上の容量を規定しています。したがって、1C 放電の容量値では0.2C 放電に対して、容量値を少なく見積もっていることとなります。しかし、0.2C 放電でサイクル試験を行うと、放電時間が 5 倍になってしまい、データ取得に非常に長い時間がかかってしまいます。

このため、サイクルは1C 放電で行い、0.2C 放電を定期的に(通常 50 サイクルごと)行う方法がとられます。サイクル試験結果をプロットした図を見ると、50 サイクルごとに容量値の大きなデータが繰り返しており、そのデータをつないで見ることにより、0.2C 放電のサイクルを行った場合のデータが得られます。実際には、0.2C 放電サイクルより1C 放電サイクルのほうがセルの劣化率が大きいので、厳密には寿命特性を厳し目に見ていることとなりますが、それ程の差はありません。

2.2.2 充電方法

リチウムイオン電池は充電方法を誤ると、事故をもたらすことがあります。充電はメーカーが指定した条件内で行うことが重要です。一般的には、電圧はセルあたり最大 4.2V、電流は1C 以下とされています。電圧は 4.1V とする場合もあるようです。10 年ほど前までは 4.1V が一般的で、容量増をねらって 4.2V が導入されてきたと理解しています。電圧が低い場合には当然、充電容量が減少します。4.1V 充電では **4.2V 充電に対して、約 10% 充電容量が低下**します。しかし、サイクル寿命は非常に改善されます。充電電流が規定値より大きすぎると、負極内に吸収されないリチウムイオンができてきて、これが金属リチウムになるといわれています。金属リチウムは良く知られているように、非常に活性な金属で、事故が発生しやすくなります。

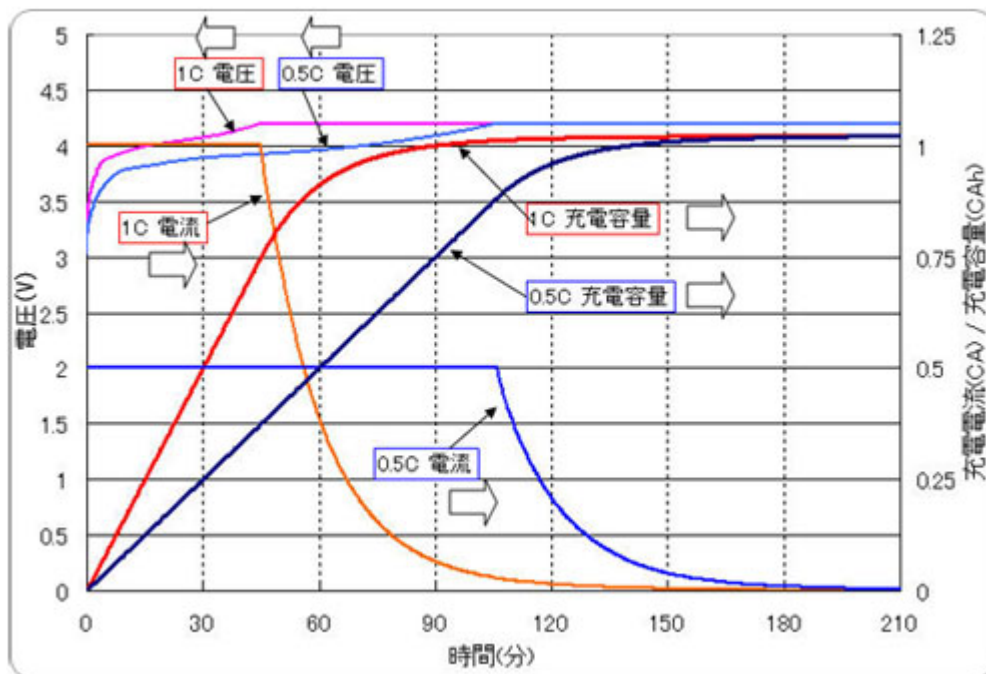
このように、リチウムイオン電池は、充電電圧と充電電流をともに管理する必要があります。これが **定電流定電圧充電方式**です。ちなみに、ニッケル水素電池やニッカド電池では、電流のみの管理で、定電流充電が行われます。しかし、ニッケル水素電池やニッカド電池では充電を止める判定が難しく、電圧の変化点や、セルの温度が急上昇するところを検出する機構が必要で、充電器としてはリチウムイオン電池用のほうがシンプルで、低コストになります。

電池パック内にサーミスタを入れて電池セルの温度を検出する場合がありますが、これはももとはニッ

ケル水素電池の充電で、満充電の時点を検出するための温度検出用であって、リチウムイオン電池では必ずしもサーミスタを入れる必然性はありません。電池メーカーが充電温度範囲を規定しているので、その温度範囲内にあることを検出するためにサーミスタを入れる、というのが本音のようです。リチウムイオン電池は低温(例えば 0°C よりだいぶ下の温度)で充電した場合は、正極から出たリチウムイオンが負極に吸収されにくくなり、リチウム金属が析出しやすいといわれています。もし、このような状態になれば、事故が発生しやすくなります。逆に、充電時の温度が高い場合は、充電電流による温度上昇が環境温度に加わり、電池が高温になる場合があることを心配します。これがリチウムイオン電池で、充電時の温度範囲が規定されている背景です。

多くの場合、充電温度範囲は $0^{\circ}\text{C} \sim 45^{\circ}\text{C}$ 程度に規定されています。実際に問題になるような低温もしくは高温というのはさらに極端に低温もしくは高温なのですが、しかし実用的には、例えば 0°C 以下で充電することはまずないだろうし、 45°C 以上で充電するケースもないでしょう。

定電流定電圧充電方式



定電流定電圧充電方式は図で示されているように、放電状態から充電を開始すると、当初電圧は低い
ため、定電流充電となります。次第に充電量が増加して、セル電圧が 4.2V に達すると、ここからは定電
圧充電となり、セル電圧が 4.2V を超えないように電流量が絞られていきます。

満充電の判定は、充電時間または充電電流の減少状態で行います。充電時間で規定する場合は、電
流 1C 、電圧 4.2V の定電流定電圧充電で充電時間 2.5 時間を満充電と定義することが多いようです。
電流 0.5C 、電圧 4.2V の定電流定電圧充電では 3.5 時間が満充電となります。

ニッケル水素電池に対するリチウムイオン電池の欠点の一つはいわゆる急速充電が困難であること
ですが、図からもわかるように、たとえば 1C 充電の場合には充電開始 1 時間後には 90% が充電されてい
ます。 0.5C 充電の場合では **120 分後には 90% 以上充電**されています。電池の容量を満充電の 90% と

割り切って考えてしまえば、ニッケル水素電池にほぼ匹敵する急速充電ができることがわかります。携帯電話等では充電時間を”短い印象”にするために、充電完了表示(ランプの色変更)を90分程度で行っているようです。充電電流は0.7C程度のもので主流ですから、セルの能力である満充電に対して、**90%程度の充電量で満充電表示**を行っています。表示が満充電となっても充電は継続している場合が多いので、急がない場合には使用する時まで充電台においておくことにより、充電量を増やすことができます。

充電量は**[電流×時間]**で決まります。

定電流定電圧充電方式で充電時間が長くなるのは、電圧が上昇したときに電流値を小さくしていくからです。セル電圧が充電中のごく短時間だけ、規格値である4.2Vを超えることを認め、電流値を大きくすることによって、充電時間を短縮しようという方法がパルス充電方式です。

この方式はいろいろな方法が提案され、各社特許を取得しているようですが、基本的な方法はセル電圧4.2V以下のときは例えば電流1Cの一定の時間幅のパルスで充電し、その後のセル電圧をモニターし、セル電圧が4.2V以上ならば充電は停止、4.2V以下ならば再度パルス充電を行うというものです。多少充電時間が短縮されますが、コストアップに見合うだけの効果は無く、また、各社の特許の権利関係が複雑です。

3 簡易充放電器全体構成

3.1 仕様

CPU	PSoC CY8C24794 (ROM:16kbyte, RAM:1kbyte)
表示	キャラクタLCD(16×2行)
放電部	
被測定数	1本(個)
放電モード	定電流モード
測定範囲	0V～4.6V, 0A～2A
放電終止電圧監視	ニッケル水素:0.9V リチウムイオン電池:2.9V
充電部	
供給能力	0V～4.6V, 0A～2A
電池充電方式	ニッケル水素1セル:定電流方式。 充電終止は温度監視(温度変化率(上昇率)0.5°C/分が4回連続で超えたとき、あるいは測定温度が52°Cを超えたときに充電終了) リチウムイオン1セル:定電流・定電圧(値は設定可能)方式
その他システム	
外部IF	PCとUSB HID IFを用いて通信
温度センサー	°C～80°C TEMP1:ニッケル水素用電池ケースに実装 TEMP2:外部サーミスタとの接続用(10kΩセンサーを想定)
電圧測定精度	0.01V
電流測定精度	1mA

3.2 ブロック図&接続図

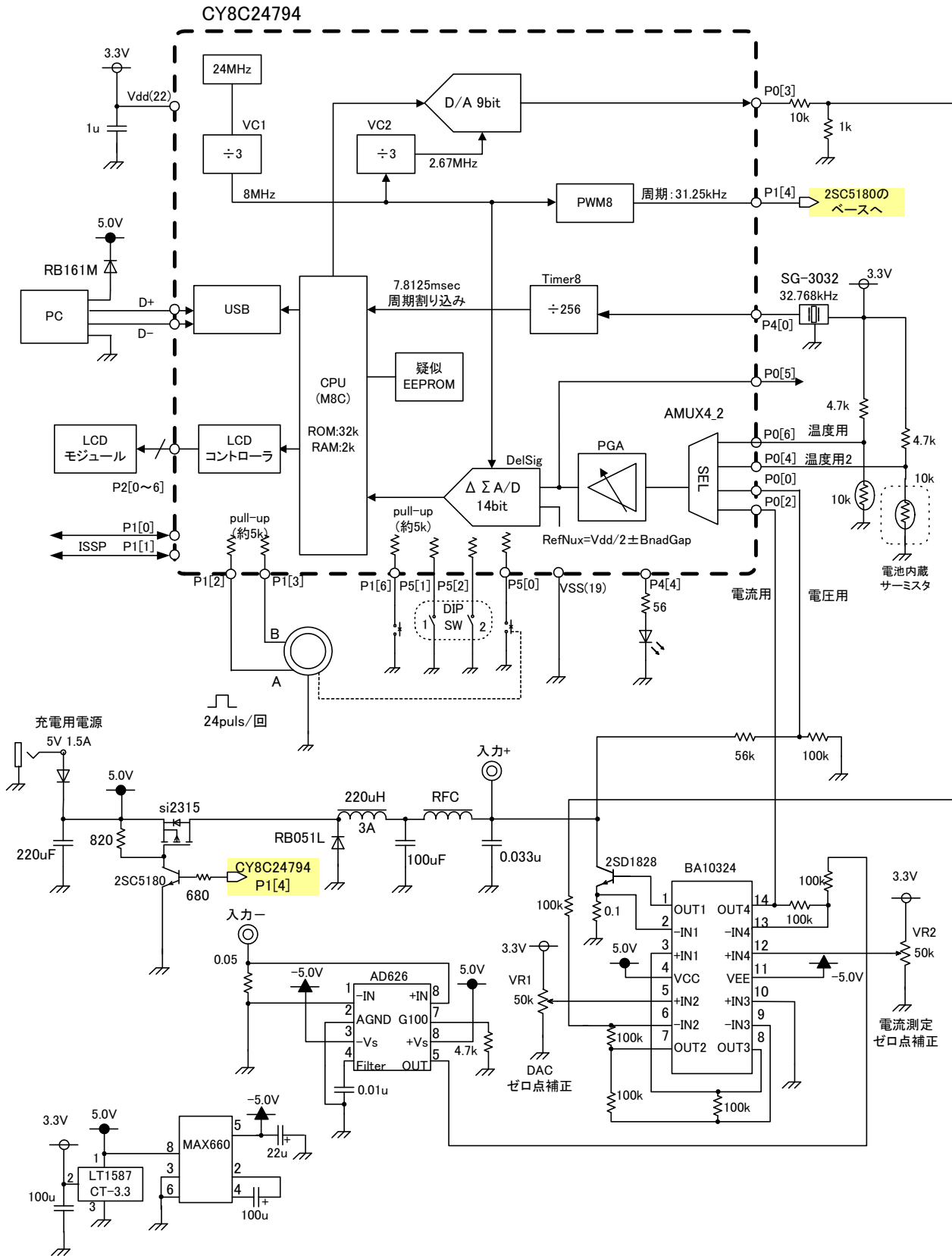


図 3-1:簡易充放電器全体回路図

3.3 H/W

3.3.1 充電部

3.3.1.1 DC/DC コンバータ

充電をおこなう方法としてニッケル水素電池は定電流電源、リチウム電池は定電圧定電流電源となり異なる。電源として上記の条件をみたすものとして結局定電圧定電流電源があればその設定値により対応できる。今回お手軽にこの装置を作るため、マイコン制御型の簡易降圧型チョッパ式 DC/DC コンバータ回路を採用することにした。元々電流値、電圧値をモニタする仕組みはあるので、設定電流あるいは設定電圧になるように DC/DC コンバータを駆動する PWM の幅を可変してやればよい。この PWM 制御には本来 16bit 程度の分解能で制御したいところだが、今回のマイコン(CY8C24794)のデジタルブロックのリソースの関係で 8bit の PWM しか構成できなかった。従って、電流(電圧)フィードバック制御時に設定値がふらつくおそれがあるが妥協することにする。

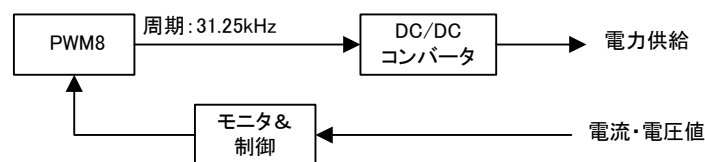


図 3-2: 充電電力供給概要ブロック

また、PWB周期を最長でも 31.25kHz に設定し、可聴周波数より高く設定することで制御時に耳障りな音を無くした。

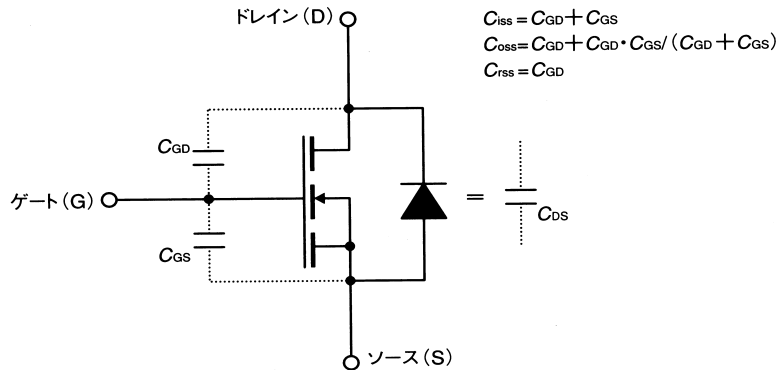
3.3.1.2 ブートストラップ回路

今回は簡易回路であるため採用していないが、DC/DC コンバータの効率と制御精度を上げるため「スイッチングの切れ」というのが重要になってくる。特に Power MOS FET ではゲート入力容量が大きいので等価的に数千 pF のコンデンサに見え、ドライブ側も駆動能力を要求される。

今回は電圧レベルの変換も兼ねてパワーMOSFET のゲートドライブには 2SC5180 で簡易的にすませる。

👉 Power MOS FET をドライブするにあたり注意する点

通常 MOS FET ではドレイン・ソースとゲート間は絶縁されているため電流が流れることはない。ただし、これは静特性の場合であり、実際には右の図のように端子間容量というのが存在する。



Ciss:入力容量

パワーMOSFET がオフ状態のときのゲート入力容量です。ゲート-ソース間容量 C_{GS} とゲート-ドレイン間容量 C_{GD} の和から求められます。数千 pF 程度になることが多い。入力容量は、スイッチング周波数などの駆動条件を決める重要なパラメータで、 C_{GD} は空乏層容量(可変容量ダイオード)の最大値、すなわと $V_{DS}=0V$ における値が重要。

Coss:出力容量

ドレイン-ソース間の容量です。言い換えると内蔵ダイオードの逆特性バイアス時の容量です。負荷回路が共振系の場合、 C_{oss} が共振回路の一部を構成するため、あらかじめ計算に入れておく必要があります。

Crss:ドレイン-ゲート間の容量 C_{GD} のことで、ミラー容量とも呼ばれます。高周波でスイッチングするときには重要なパラメータです。 C_{rss} は C_{iss} 、 C_{oss} よりも高周波特性への影響が大きいからです。優れた高周波特性を得るためには極力 C_{GD} の小さい素子を選択する必要があります。

もし、スイッチング速度を高速化したい場合、ゲートドライブ回路を追加するとよい。

パワーMOSFET の場合 ON 速度よりも OFF 速度の方が遅くなる傾向にある。

以下にゲートドライブ回路にブートストラップを入れたものとそうでないものとで、40nsec幅のパルス(約 12.5MHz 相当)入力時のスイッチング応答速度の比較を示す。

スイッチングトランジスタは ROHM 製の”高速スイッチング用”の SPCIE ライブラリを使用した。

また p-ch MOSFET も ROHM 製で ON 抵抗は約 1Ω のもの。(ちょっと大きいかもしれない(^^))

ON 抵抗がこのくらいだと C_{iss} などが 50pF と小さいようだ。si2315 では C_{iss} は 1235pF とかなり大きい。

3.3.1.3 ゲートドライブ回路のシミュレーション

簡単ではあるがゲートドライブ回路のシミュレーションを行ってみた。

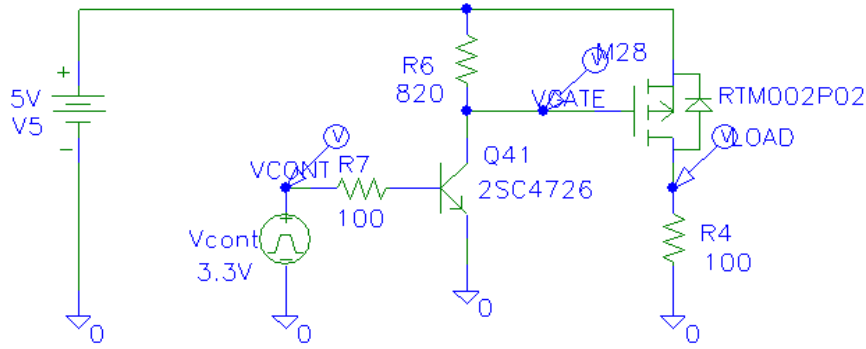


図 3-3: 通常の簡易的なゲートドライブ

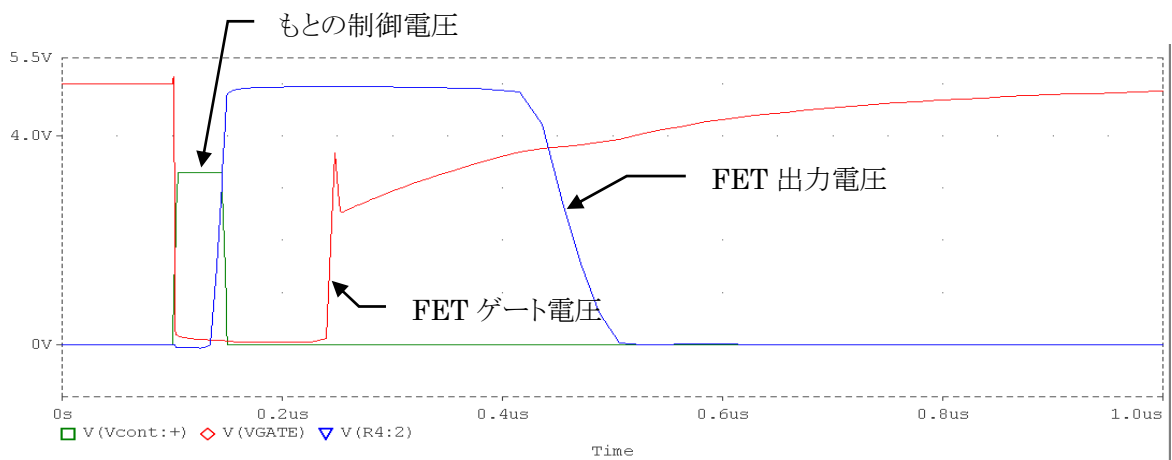


図 3-4: 簡易ゲートドライブの応答速度シミュレーション値

図 3-3 はゲートドライブの簡易的なもの。このシミュレーション結果 図 3-4 からゲート電圧を OFF(high) にしてもだだ電位があがっていくため FET の OFF は約 500nsec のところとなる。この立ち上がりは R6 の抵抗値のドライブ能力に依存する。立ち上がりを速くしようと抵抗値小さくすればいいのだが、そうすると Q41 を ON にしたときに無駄に電流が流れることとなり、R6 は大きいにこしたことはない。

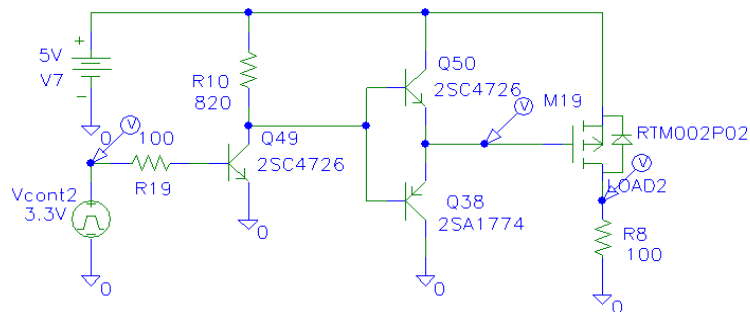


図 3-5: ゲートドライブにブートストラップ回路を用いたもの

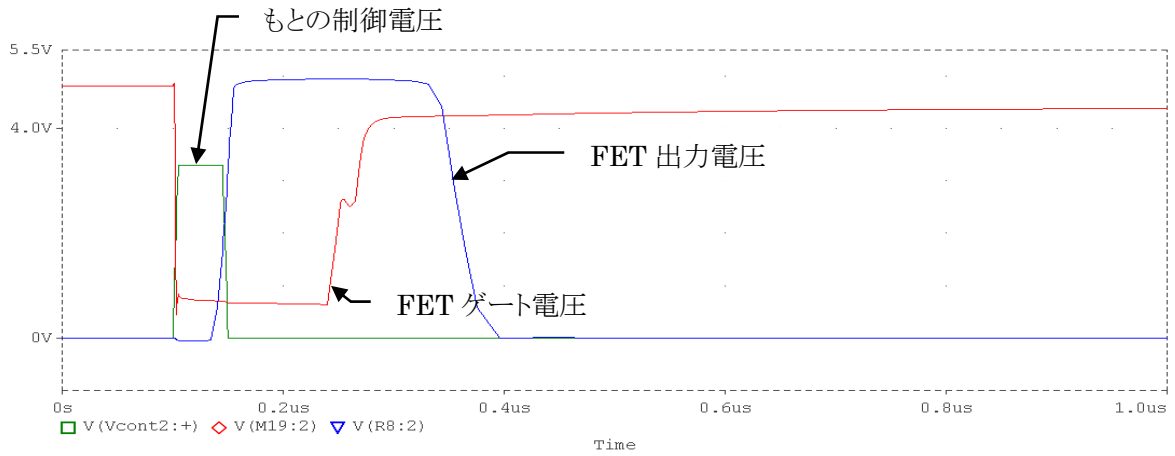


図 3-6:ゲート回路にブートストラップを用いた場合の過渡特性

一方、**図 3-5**ではゲートドライブにPNP,NPNのトランジスタをプッシュプル型で構成したブートストラップ回路を挿入した。Q49 のトランジスタは制御論理を合わせるのと同時に制御電圧のレベル変換も行っている(3.3V→5.0V)。この回路ではM19 のパワーMOSFETのゲートをHIGHレベルにするのにQ50(2SC4726)のトランジスタで積極的に電荷を注入しており、前述の回路よりFETゲート制御の応答(キレ)が改善されており、FET出力で 100nsec程度速くOFFになっていることがわかる。でも、見方によってはこれだけの回路を入れてやっても 100nsecしか改善しないのでその効果はいかかなものかと思った。

ただしこのブートストラップ回路はそのドライブするデバイスの周波数特性に依存するのでなるべくスイッチング性能のいいものを選びたい。

3.3.2 放電部

3.3.2.1 電子負荷

放電部はもっとも一般的な定電流モードが用いられ、基本的には**図 3-7**に示すような定電流回路となる。

トランジスタ Q4 に流れる電流 I_i を R_{s2} で電圧に変換し、この電圧と制御信号 V_4 が等しくなるように Q4 のベース電流を制御します。 R_{s2} の電圧が V_4 より小さくなると OP アンプ U16A の出力電圧が上昇し I_i を増加させます。逆に R_{s2} の電圧が V_4 より大きくなると U16A の出力電圧が降下し I_i を減少させます。流れる電流は

$$I_i = \frac{1}{R_{s2}} \times V_4$$

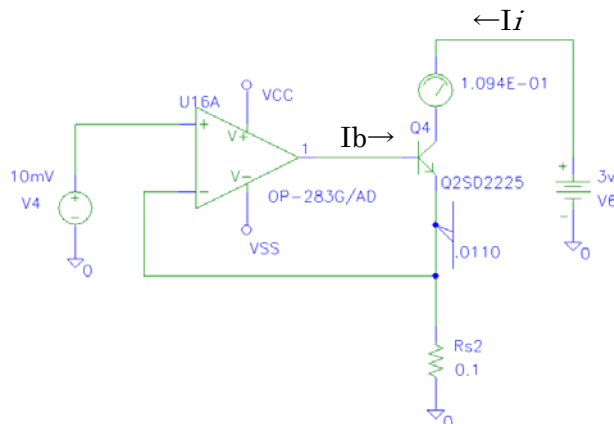


図 3-7:定電流制御回路

ゲート電圧を 10mV、 R_{s2} を 0.1 Ω としているため、流れる電流は 100mA となっている。

電流制御部で用いる Q4 のトランジスタで供給される電力を熱に変換するため放熱をつける必要がある。また、この手のパワートランジスタは通常 1 個では h_{FE} が 50 くらいなので負荷電流制御を 2A とした場合、 $I_b = 2A / 50 = 40mA$ となり OP アンプがドライブできないほどの電流となっています。そのため、この Q4 はダーリントトランジスタ 2SD1828 を用いる。このトランジスタでは h_{FE} がおよそ 1000~

2000 程度なので I_b も $1\text{m}\sim 2\text{mA}$ 程度となり現実的な値となる。

図 3-8は図 3-7で使用していたドライバのトランジスタをPower MOS FETに置き直したものです。

シミュレーション的にはOKなのですが、図 3-7でもし被測定電池が 0.6V 以下になった場合、トランジスタを使っているとベースからコレクタに電流が流れてしまいます。またパワートランジスタは h_{FE} が小信号タイプのものより小さいのでダーリントン構造のものを使用する必要があります。

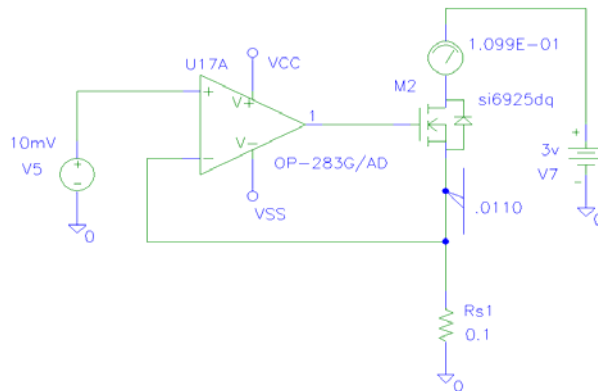


図 3-8:トランジスタを MOSFE に変更する

それと比較して MOS-FET であれば原理的にゲートに電流が流れ込むことがないのでドライバーの設計負荷が軽くなる。

図 3-9に定抵抗負荷回路を示す。

$$I_i = \frac{k}{R_1} \times V_i \quad \text{ただし } k = \frac{R_{21}}{R_{20} + R_{21}}$$

これは負荷としては $\frac{V_i}{I_i}$ が一定となる負荷である。

あまり電池試験には用いられない

V_i を R_{20}, R_{21} で分圧し、 I_i の検出抵抗 R_1 の大きさと等しくなるように動作します。つまり給電電圧 V_i が大きくなれば流れる電流 I_i も供給電圧に比例して大きくなります。従って、分圧値を変えて抵抗値を変化させます。マイコンで電圧をモニタしているので間接的にこの動作をさせることは可能だが今回は見送ることにする。

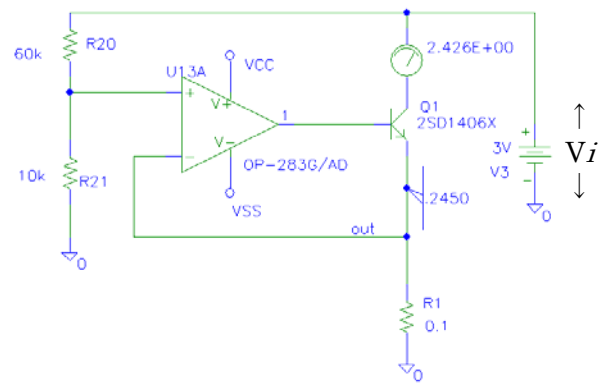


図 3-9: 定抵抗負荷回路

3.3.2.2 電子負荷の制御

電子負荷の制御方法は図 3-7の回路で V_4 のリファレンス電圧を制御してやればよい。 $I_i = \frac{1}{R_{s2}} \times V_4$

(ここで $R_{s2}=0.1\Omega$) であるから、最大 2A の電流負荷まで設定できるようにするのであればリファレンス電圧である V_4 は 200mV まで可変できればいいことになる。

今回、この可変電圧制御としてPSoCのCY8C24794 のアナログブロックのDAコンバータ部を使用した。このDAコンバータは $1/2V_{CC} \pm \text{Bandgap}(1.3\text{V})$ なので $1.65\text{V} \pm 1.3\text{V}$ となり $0.35\text{V} \sim 2.95\text{V}$ となり 9bitなので約 5mV/LSB となる。このままの電圧範囲では大きすぎるので分圧抵抗で $1/11$ にしているので $31.8\text{mV} \sim 268\text{mV}$ となる。それでも 31.8mV のオフセットがあるため電流設定では $31.8 \times 10 =$

318(mA)の設定値となってしまう。そのため反転アンプを 2 段入れてオフセット調整により 0digitのとき 0mAになるように設定する(図 3-1のVR1)。9bitなので 1digitあたりは約 4.6mAの設定値となる。実際には設定電流値の小さいときDACのリニアリティ特性が悪いため思うような設定値にはならないが、あまり小さい設定値は使わないので実用的な範囲では問題ない。

3.3.3 電流測定回路

電流測定はシャント抵抗を使った方法にした。このときドロップ電圧を精度よく測定する IC として定評のあるアナログデバイゼスの AD626 を使用することにした。ちょうどこれもアナデバの HP に AD626 の SPICE ライブラリがあったのでそのシンボルを作成してシミュレーションしてみた。今回は充放電器なので”充電”も”放電”も電流を測定できるようにしたいので、充電時には”マイナス(-)」、放電には”プラス(+)”になるようにした。ということはAD262 の出力は負電圧になるということで、今回負電源を用意することにした。また負電源があることで 0V 付近のリニアリティ特性がよくなり、電圧範囲の変換(反転、オフセット)もやりやすくなる。図 3-10 に電流測定回路を示す。ここではAD626 では正負の電圧が出力されるが、マイコンのADで取り込む際負電圧は入力できないので、後段のオペアンプでプラスにオフセットさせ正電圧の範囲になるようにする(図 3-1の VR2)。

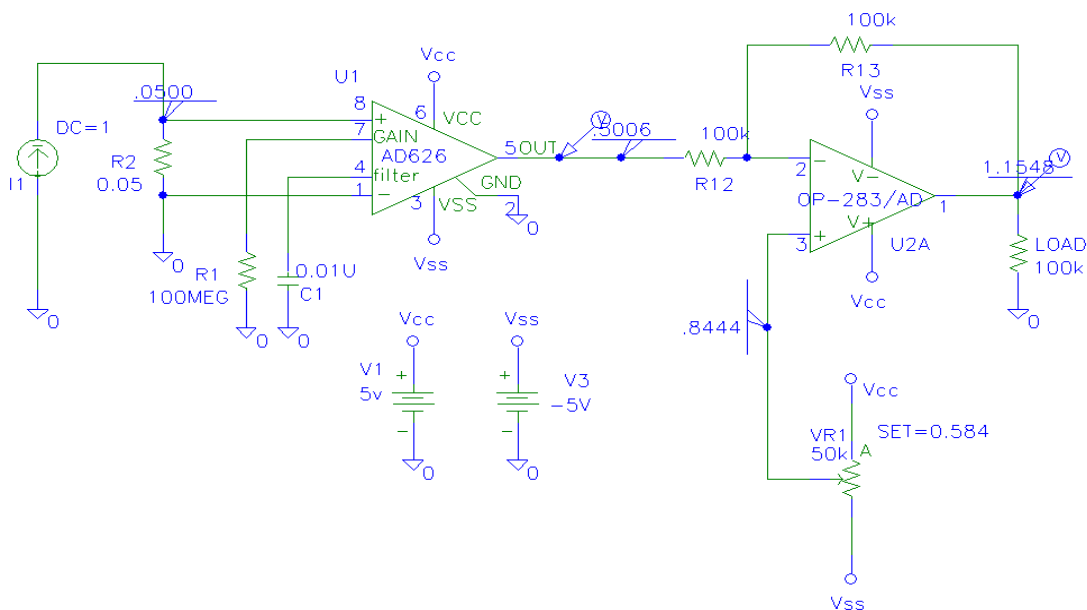


図 3-10: 電流測定回路

ただし、後段のオペアンプは反転回路なので、図 3-11 に示すように測定電流と測定値の傾きは逆になるが、この辺はソフトで修正する。

AD626 の GAIN 設定は 10/100 倍を選択できるが今回は 10 倍固定にした。

測定範囲は充電(-2A)～放電(2A)の範囲とし、AD コンバータへの入力は 2.0V～0.64V の範囲となる。

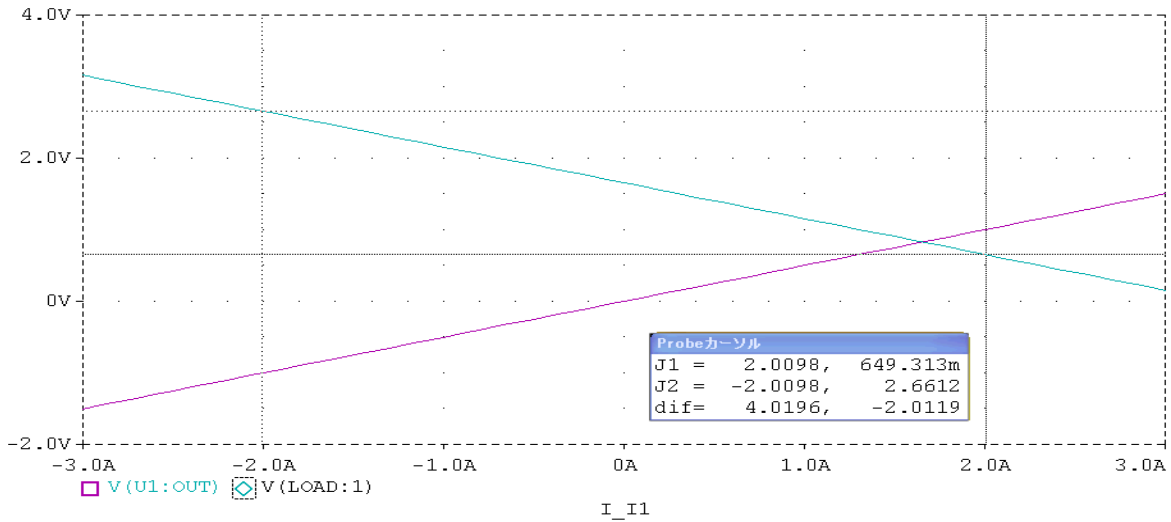


図 3-11: 電流測定回路特性

3.3.4 電圧測定回路

電圧測定レンジは今回の使用ではリチウム 1 セルまでの対応としたため 4.6V 程度を MAX 値とした。また、マイコンの AD コンバータのリファレンス電圧を安定化するため 3.3V の電源電圧とした。14bit ΔΣタイプ of AD コンバータで測定範囲は $1/2VCC \pm \text{Bandgap}(1.3V)$ なので $1.65V \pm 1.3V$ となり 0.35V~2.95V であり 100k と 56kΩ の分圧抵抗を用いることで測定電圧範囲に収まるようにした。そのため測定電圧範囲は 0.546V~4.6V となる。

3.3.5 温度センサー

これは主にニッケル水素電池充電時の発熱反応をモニタするためのものである。温度センサーには 10k@25°C のサーミスタを用いることにする。4.7kΩ の分圧抵抗を用いて S/W で電圧値を温度値に読み替える。

センサーは 2 系統用意した。1つは写真 3-1 に示すように電池フォルダに装着した。

これは単三型のニッケル水素電池は満充電近くになると発熱することから、過充電防止のためである。もう1つは温度センサーとして独立で設置できるようにした。もともとこのセンサーはリチウムイオン電池の充放電の温度監視で使おうと思っていたが、リチウム電池はその性質上あまり発熱しないことが判ったので、実質使用しないですみそうである。

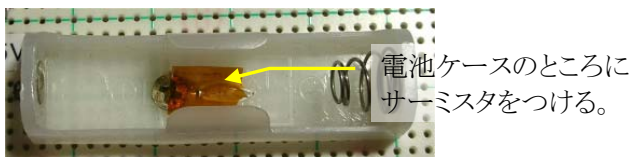


写真 3-1: サーマスタを一体化した電池ソケット

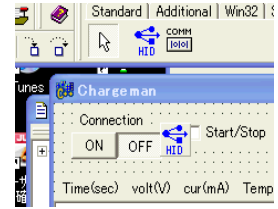
3.4 PSoC S/W

3.4.1 PC Interface(HID)

PC との IF は UART が一般的に使われる。最近ではシラボの CP2102 のように UART on USB のような IFIC もありお手軽になってきた。このような接続方式は PC で仮想 UART として認識さ

れるため、USBを刺すポートやハブによって認識されるポート番号が変更されたりして面倒なことも最近多い。ということで、USBのHIDデバイスを使用し、ベンダーIDとプロダクトIDを固定することでPCアプリケーションから安定した接続とした。

また、USBではパケットデータの交換となり、HOSTとDEVICEそれぞれにデータバッファのENDPOINTレジスタをもつためデータ処理がやりやすいというメリットもあり、最近これになるべく使うようにしている。PC側の開発はBorlandのC++ Builder 6.0を使用している。これは以前HID IFをこの環境のコンポーネントライブラリとして作成しており、かなり使いやすく仕上がっているのでお気に入りではある。(こんな感じ→)



INPUT(デバイス→HOST)

オクテット	内 容
1	各状態 bit0:動作状態 0:停止中、1:動作中 bit1:電池種別 0:ニッケル水素、1:リチウムイオン電池 bit2:充電モード 0:放電、1:充電 bit3~bit7: Reserved
2	測定経過時間(1/8 秒単位) トータル 32bit の 31bit~24bit
3	測定経過時間(1/8 秒単位) トータル 32bit の 23bit~16bit
4	測定経過時間(1/8 秒単位) トータル 32bit の 15bit~8bit
5	測定経過時間(1/8 秒単位) トータル 32bit の 7bit~0bit
6	bit5~bit0:電圧測定値 $\Sigma \Delta AD$ コン出力 14bit の上位 6bit
7	bit7~bit0:電圧測定値 $\Sigma \Delta AD$ コン出力 14bit の下位 8bit
8	bit5~bit0:電流測定値 $\Sigma \Delta AD$ コン出力 14bit の上位 6bit
9	bit7~bit0:電流測定値 $\Sigma \Delta AD$ コン出力 14bit の下位 8bit
10	bit5~bit0:温度測定値 1 $\Sigma \Delta AD$ コン出力 14bit の上位 6bit
11	bit7~bit0:温度測定値 1 $\Sigma \Delta AD$ コン出力 14bit の下位 8bit
12	bit5~bit0:温度測定値 2 $\Sigma \Delta AD$ コン出力 14bit の上位 6bit
13	bit7~bit0:温度測定値 2 $\Sigma \Delta AD$ コン出力 14bit の下位 8bit
14	積算電流(mAh)値単精度 IEEE 754 形式(上位から 1byte 目)
15	積算電流(mAh)値単精度 IEEE 754 形式(上位から 2byte 目)
16	積算電流(mAh)値単精度 IEEE 754 形式(上位から 3byte 目)
17	積算電流(mAh)値単精度 IEEE 754 形式(上位から 4byte 目)
18	積算電力(mWh)単精度 IEEE 754 形式(上位から 1byte 目)
19	積算電力(mWh)単精度 IEEE 754 形式(上位から 2byte 目)
20	積算電力(mWh)単精度 IEEE 754 形式(上位から 3byte 目)
21	積算電力(mWh)単精度 IEEE 754 形式(上位から 4byte 目)

22	設定放電電流値(mA) IEEE 754 形式(上位から 1byte 目)
23	設定放電電流値(mA) IEEE 754 形式(上位から 2byte 目)
24	設定放電電流値(mA) IEEE 754 形式(上位から 3byte 目)
25	設定放電電流値(mA) IEEE 754 形式(上位から 4byte 目)
26~29:reserved	

OUTPUT(HOST→デバイス)

オクテット	内 容
1	各状態設定 bit0:動作状態 0:停止中、1:動作中 bit1:電池種別 0:ニッケル水素、1:リチウムイオン電池 bit2:充電モード 0:放電、 1:充電 bit3~bit7: Reserved
2	充電時の電流を設定。目標電流値(0~255) PWM は 8bit なので 8bit に丸める。(電流値→PWM の変換は PC 側でおこなっておく)
3	放電設定値(DAC9bit の値) 上位(bit8) (電流値→PWM の変換は PC 側でおこなっておく)
4	放電設定値(DAC9bit の値) 下位(bit7~bit0) (電流値→PWM の変換は PC 側でおこなっておく)
5~16 まで RESERVED	

INPUT(デバイス→HOST)の 14 から 25 オクテットの IEEE754(単精度)の形式で渡すとき、4byte のデータの並びが逆になっていた。どちらが正しいかわからないが、とりあえず読み方を逆にすることで解決した。

PSoC(CY8C24794)の USB デバイスのディスクリプタは下記のような感じ。PC 側(HOST)で特別なドライバーを必要としない(ドライバーが標準に組み込まれている)ため、HID デバイスとして使用している。

USB Setup Wizard

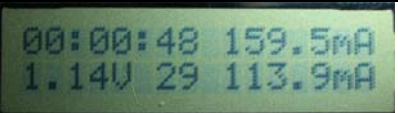
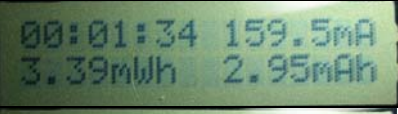
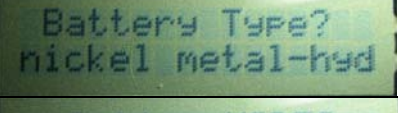
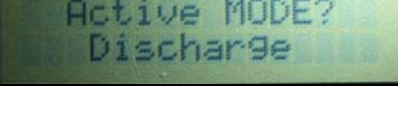
Descriptor	Data	Operations
USB User Module Descriptor Root	USBFS	Add Device
Device Descriptor	Device	Remove Add Configuration
Device Attributes		
Vendor ID	4242	
Product ID	EE02	
Device Release (bcdDevice)	0000	
Device Class	Defined in Interface Descriptor	
Subclass	No Subclass	
Manufacturer String	RamenKikaku	
Product String	ShimmyDevice	
Serial Number String	No String	
Configuration Descriptor	Configuration	Remove Add Interface
Configuration Attributes		
Configuration String	No String	
Max Power (mA)	500	
Device Power	Bus Powered	
Remote Wakeup	Disabled	
Interface Descriptor	Interface	Remove Add Endpoint
Interface Attributes		
Interface String	SimpleIO	
Class	HID	
Subclass	No Subclass	
HID Class Descriptor		Remove
Descriptor Type	Report	
Country Code	Not Supported	
HID Report	SimpleIO	
Endpoint Descriptor	Endpoint	Remove
Endpoint Attributes		
Endpoint Number	EP1	
Direction	IN	
Transfer Type	INT	
Interval	10	
Max Packet Size	29	
Endpoint Descriptor	Endpoint	Remove
Endpoint Attributes		
Endpoint Number	EP2	
Direction	OUT	
Transfer Type	INT	
Interval	10	
Max Packet Size	16	

String/LANGID	Data	Operations
String Descriptors	USBFS	Add String
LANGID	English (United States)	
String	RamenKikaku	Remove
String	LoadTester	Remove
String	ShimmyDevice	Remove
String	SimpleIO	Remove

Descriptor	Data	Operations
HID Report Descriptor Root	USBFS	Add HID Report Import HID Re
HID Report Descriptor	SimpleIO	Insert HID Item Delete HID Re
Usage Page	Usage Page 06 A0 FF	Insert HID Item Remove Move
Usage	Usage 09 A5	Insert HID Item Remove Move
Collection	Collection (Application 01)	Insert HID Item Remove Move
Usage	Usage 09 A7	Insert HID Item Remove Move
Logical Minimum	Logical Minimum 15 80	Insert HID Item Remove Move
Logical Maximum	Logical Maximum 25 7F	Insert HID Item Remove Move
Report Size	Report Size 75 08	Insert HID Item Remove Move
Report Count	Report Count 95 1D	Insert HID Item Remove Move
Input	Input (Data, Variable, Absolute 02)	Insert HID Item Remove Move
Usage	Usage 09 A9	Insert HID Item Remove Move
Logical Minimum	Logical Minimum 15 08	Insert HID Item Remove Move
Logical Maximum	Logical Maximum 25 7F	Insert HID Item Remove Move
Report Size	Report Size 75 08	Insert HID Item Remove Move
Report Count	Report Count 95 10	Insert HID Item Remove Move
Output	Output (Data, Variable, Absolute 02)	Insert HID Item Remove Move
End Collection	End Collection C0	Insert HID Item Move Up Mov

図 3-12: Cypress(デバイス側)の USB ディスクリプタの設定内容。

3.4.2 各種設定画面

画面 No	表示	表示内容
1		【経過時間】 【設定電流値】 【測定電圧値】 【測定電流値】 (電源投入時の最初の画面)
2		【経過時間】 【設定電流値】 【積算電力値】 【積算電流値】
3		ニッケル水素とリチウムイオン電池の切り替え 放電終止電圧がその電池にレベル設定される。それぞれ 0.9V と 2.9V
4		充電モードと放電モードの切り替え 充電方式は電池の種類によって変わる

3.4.3 システムタイマー

積算電力を求めるためある程度正確な時間が必要である。

CY8C24794 は 32kHz のクロックが内蔵されており、これを分周することで1秒周期の時間を取得してみようと思った。その精度を測定してみたところ

実時間:23940 秒(6 時間 39 分)に対して 24289 秒だった

つまり、 $(24289-23940)/23940=1.45\%$ (約 5 分 48 秒の誤差となった)これは常温での精度であるが、温めたり冷やしたりするとかかなり周波数が変わったことから温度偏差が大きいらしいことが判った。水晶振動子で発振させているわけではないのでこの手の周波数精度はあまりアテにならない。

そこで 32.768kHz の水晶発振器を用い、絶対周波数の精度を上げ、温度による偏差をすくなくした。図 3-13 にシステムタイマーの構成を示すが、内部ソフトの処理はUSBIFの受信処理以外、基本的に 125msec の周期で行う。

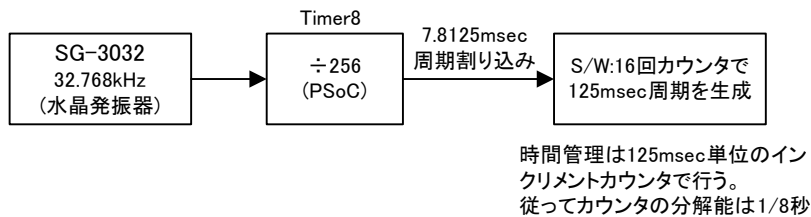


図 3-13:システムタイマーの構成

試しにクロック精度を測定してみる。IO ポートにタイミング信号を取り出し、その周期を測定する。

この測定値からすると精度は

$$(124.99966-125)/125 = -2.72\text{ppm}$$

の誤差となる。これは約1ヶ月動作させて7秒程度の誤差に相当する。

ということで、精度的には問題無いレベル。環境温度に対しても問題ない。(リアルタイムクロックの源振に使用されるのだから当然だけどd(^-^)ネ!

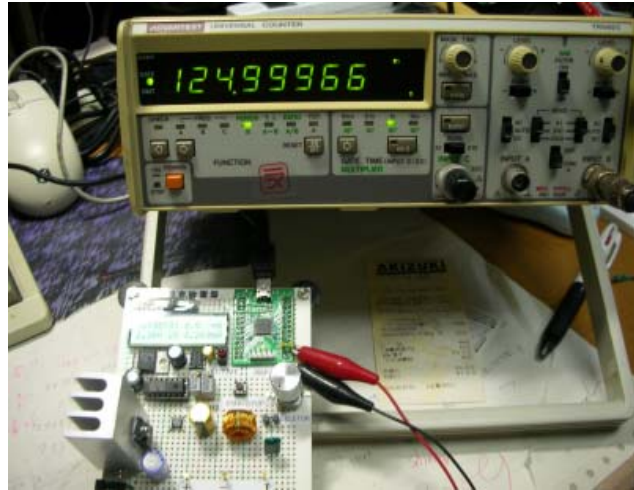


写真 3-2:システムタイマーの精度を測定

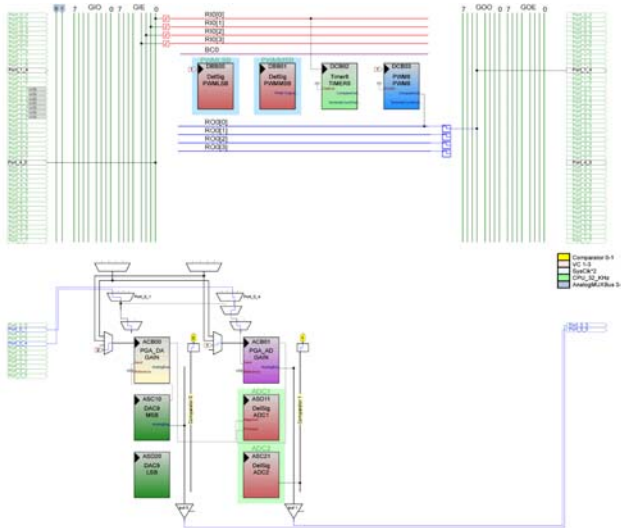


図 3-14: PSoC 内部のリソースの配置

Cy8C24794 は USB IF を特徴とするマイコンである。内部リソース的にも上記のように PSoC のシリーズの中では少ない方である。ROM: 16kbyte, RAM: 1kbyte のメモリ容量であるため、それほど複雑なことはできない。特に今回 ROM の使用率は 98% でありカツカツって感じ。

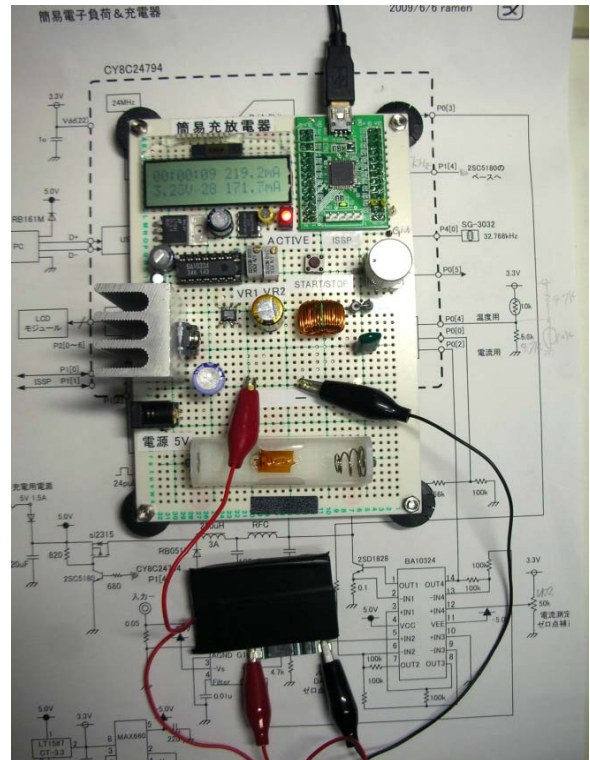


写真 3-3: 簡易充放電器の外観

3.4.4 AD値の扱いについて

今回、電流測定において充電時は”マイナス電流値”、放電時は”プラス電流値”で電流の方向を定めた。従って、無負荷(無充電)時に PSoC に入力される電流相当の電圧値(0V)は $1/2V_{CC}$ であり、AD コンバータの測定範囲は $1/2V_{CC} \pm \text{Bandgap}(1.3V)$ 。変換結果はコンプリメンタルバイナリとしているので、 $1/2V_{CC}$ は 0digit となり $0.35V \rightarrow -8192$ 、 $2.95V \rightarrow 8191$ となる。従って、デバイスから HOST(PC)に

USB HID 経由でデータ転送するときには”符号無し”として扱っているので、PCで受けとったデータはオフセット値を加減する必要がある。

3.4.5 温度補正

サーミスタは温度によって抵抗値が大きく変化する部品である。その特性は温度係数に対して負の特性となる。つまり、温度が上昇すると抵抗値は下がる。この特性は一般的に以下のように表される。

$$R_T = R_0 \cdot \exp\left(B \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right)\right) \quad \dots\dots \text{式①}$$

ここで、サーミスタのそのときの抵抗値を $R_T(\Omega)$ 、絶対温度を $T(K)$ とすると R_T は図中に示すような関係にあります。ここで $T_0(K)$ は基準温度 R_0 は基準温度 T_0 での抵抗値、 $B(K)$ は B 定数といわれるものです。この特性パラメータである B 定数はデバイスにより変わる固有値です。

今回使用したサーミスタは $B=3370$ 。

図 3-15はサーミスタにシリーズに $4.7k\Omega$ の抵抗を入れることでより直線に近いリアリティとなる分圧電圧値が出力されるようにする。グラフ 3-1は式①の特性のサーミスタを図 3-15の回路で取り出し、ADコンバータで変換した時を計算で求めたモノである。ADコンバータの値に対する温度の値は近似式ではほぼ相関が1となるので、AD値→温度(°C)はこの三次近似式を用いることにする。

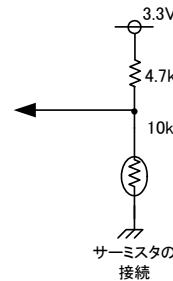
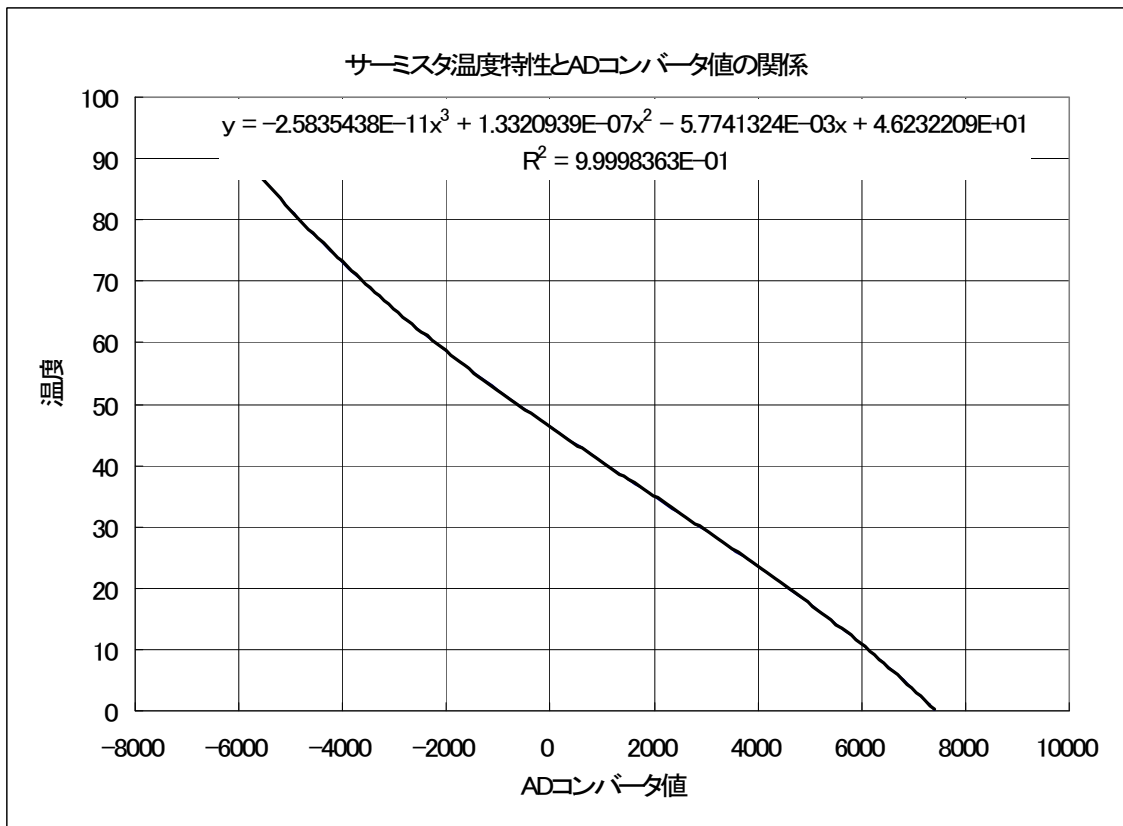


図 3-15:サーミス回路



グラフ 3-1:サーミスタの温度特性と AD 値

3.4.6 充電制御

3.4.6.1 ニッケル水素電池の場合

ニッケル水素電池は2.1章で説明したように基本的に定電流方式となる。このときの電流値は電池の容量によって異なる。急速充電を行う場合はメーカーによってその電流値は異なるようだが、ここでは1Cくらいを目安にする。つまり2000mAhの容量をもつ電池であれば充電電流は2Aということなる。中には「いたわり充電」といって0.1C(2000mAhの電池であれば200mA)で充電する装置もあるようだけど、こんなのかつたるくてやられてられない。一方超急速充電と称して15分くらいで充電するものもあるみたいだ。そうすると4C程度の電流が必要で、 $2000\text{mA} \times 4 = 8\text{A}$ の電流を流すことになる。これはこれで「大丈夫かな」っていった具合だ。

最近 SANYO とか SONY から出ているようですね。

<http://wince.goo.ne.jp/snap/cnBoard.asp?PID=1155&DIR=P>

決して15分の充電自体が難しい話ではない。難しいのはどこで充電をやめるかということ。ニッケル水素系電池の過充電は劣化の要因ともなるのでなるべく速やかに充電を終了する必要がある。一般的にニッケル系の電池は定電流で充電しているため、充電終了の判断はリチウムイオン電池より難しい。この充電終了方法として大きく3つある。

1. タイマー方式、2. 充電末期の電圧低下を検知する方式(- ΔV 方式)、3. 電池の温度上昇を計る方式。

タイマー方式は「いたわり充電」などでもよく使われますが、電池が空になっていることを条件で「電流値×時間」で決められた時間で終了するもの。確かに確実ではありますが、電池の状態を把握しておく必要があるので、ちょっと管理が面倒です。

温度上昇を検出して充電を停止する急速充電方式は、ノートパソコンのNiCdバッテリー、コードレスドライバーのバッテリーなどによく採用されているようです。電池に+,-の端子だけでなく、温度検出用のサーミスターから引き出した端子もあります。電池の内部にサーミスターがあるため、温度検出が正確にできるようになっています。

電池を充電していくと、次第に電圧が上がっていき、満タンになる前に急激に電圧が上がります。満タンになった後は、充電電流は熱に変わるため、電池が発熱します。発熱によって電池自身の温度が上がるため、電圧が低下し始めます。ここでピーク電圧低下を利用するのが- ΔV 方式である。

しかし、ニッケル水素系の電池ではこの電圧変化が捕らえにくくなっています。

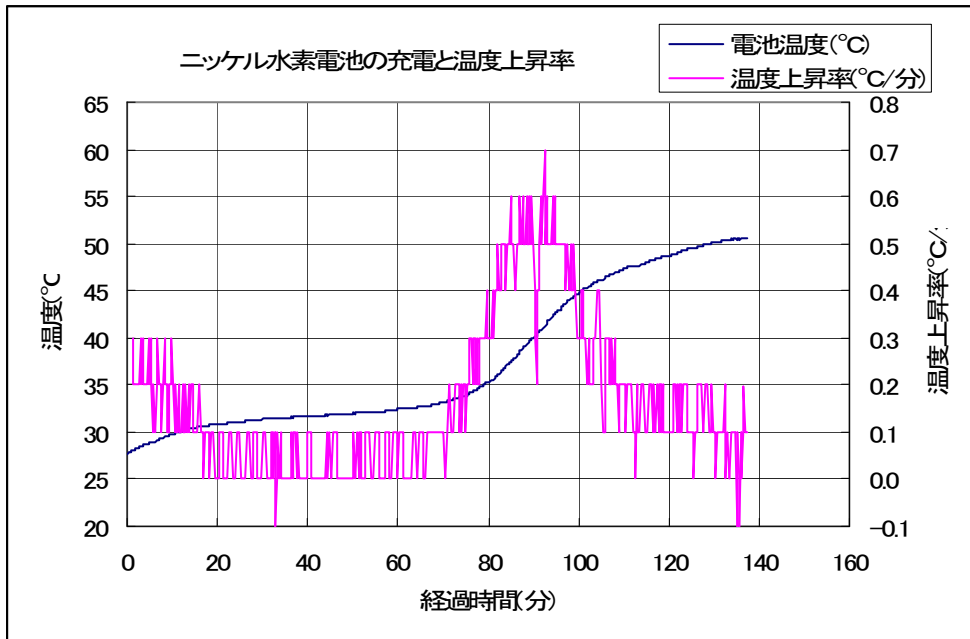
そこで今回、超急速充電時の終止判断にも使える「温度監視」による方法を採用しました。

これであれば満充電状態に近いモノであればすぐに発熱モードになるので確実かと思った次第です。しかし、「温度監視」には欠点が1つあります。それは電池の温度を安定して測定できるかということです。今回のものは単三型の電池のみを扱う対象としましたが、通常この手の電池には温度センサーは付いていません。従って、写真 3-1(P20)のように電池ケースの下になるべく電池に接触する形でサーミスタをつけることにしました。ただ、それでもセンサーと電池との接触具合によって2,3°C低めに測定されてしまうことがありました。最初、52°C以上の温度上昇を検出すると充電を終了していましたが、グラフ 5-2(P34)のように電圧変化だけをみると明らかに充電終了と判断しているものですが、52°C以上の温度を検出できなかったために充電動作をやめることはありませんでし

た。これではちょっと危険だと感じたため、絶対温度の閾値で判断するのではなく、温度上昇率を充電の終了と判断するようにしてみました¹。

当初、**グラフ 3-2**に充電状態と電池の温度上昇率とを重ねたグラフを表す。これによると温度上昇率 $0.4^{\circ}\text{C}/\text{分}$ を3回(3分)以上続いたとき満充電と判断してもいいだろう。それと、これまでの「 52°C 以上になったら充電終了」という条件とのORを取るようにする。

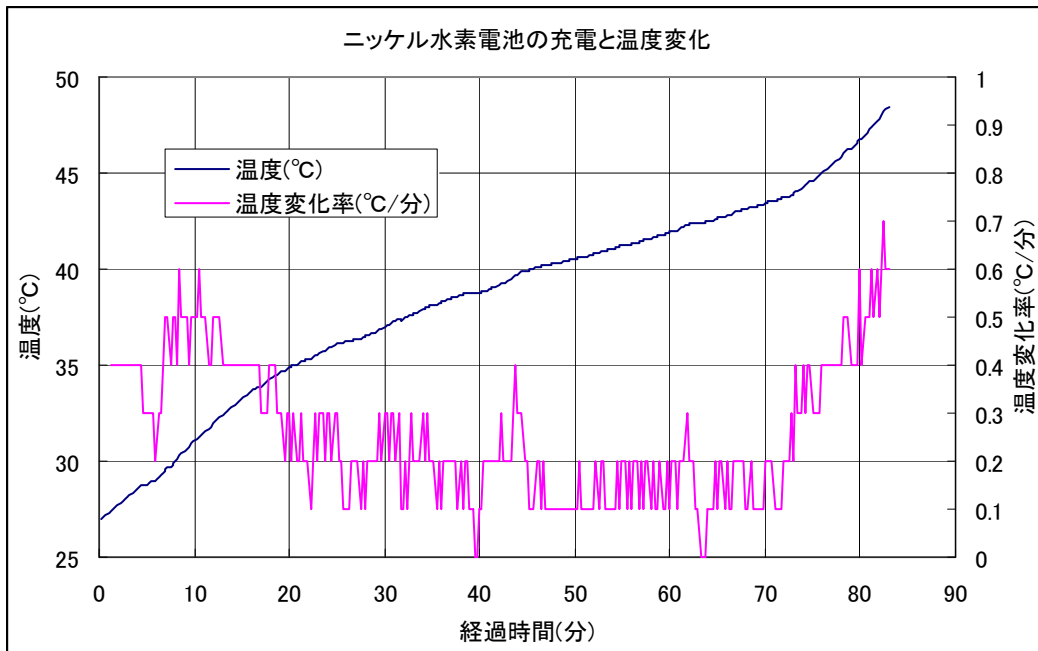
しかしこの判定では最初の充電開始時に温度が割と急勾配で上昇するところもあり、その部分で誤判定をしてしまう恐れもあったため、終了判定のための温度上昇係数を大きくした。



グラフ 3-2: ニッケル水素電池の充電と温度変化(ケース1)

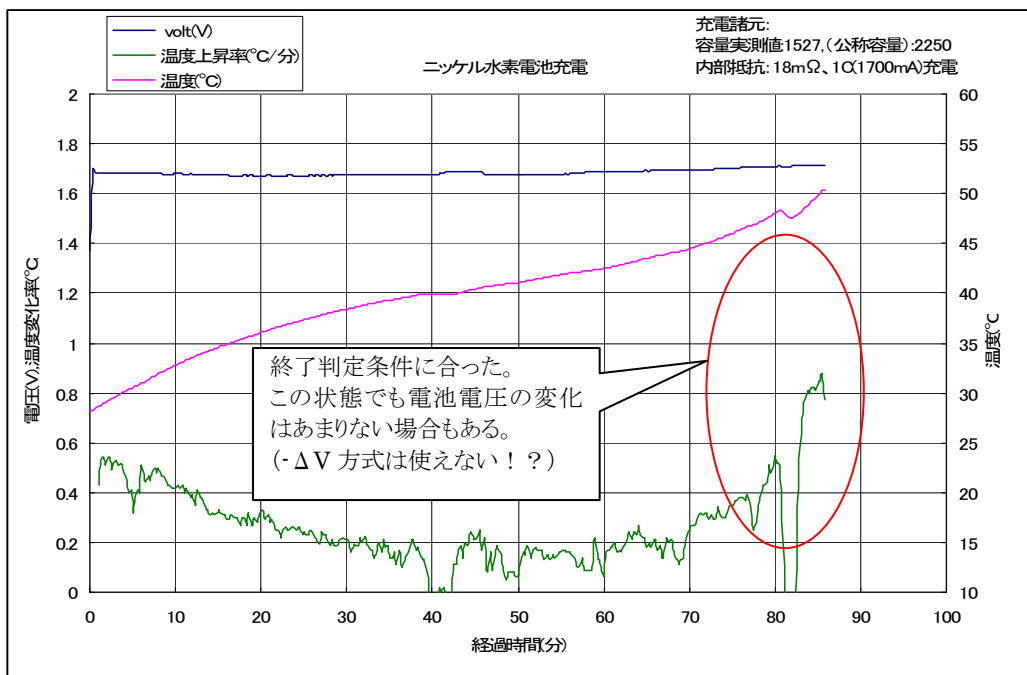
充電終了条件は「温度変化率(上昇率) $0.5^{\circ}\text{C}/\text{分}$ が4回連続で超えたとき、あるいは測定温度が 52°C を超えたとき。」**グラフ 3-3**では前記の条件で充電終了となっている。(充電電流条件: $1\text{C}(1.7\text{A})$)

¹実はマイコン ROM が少なくなってきたのであまり追加機能は入れられなかったのがこのようになっていました



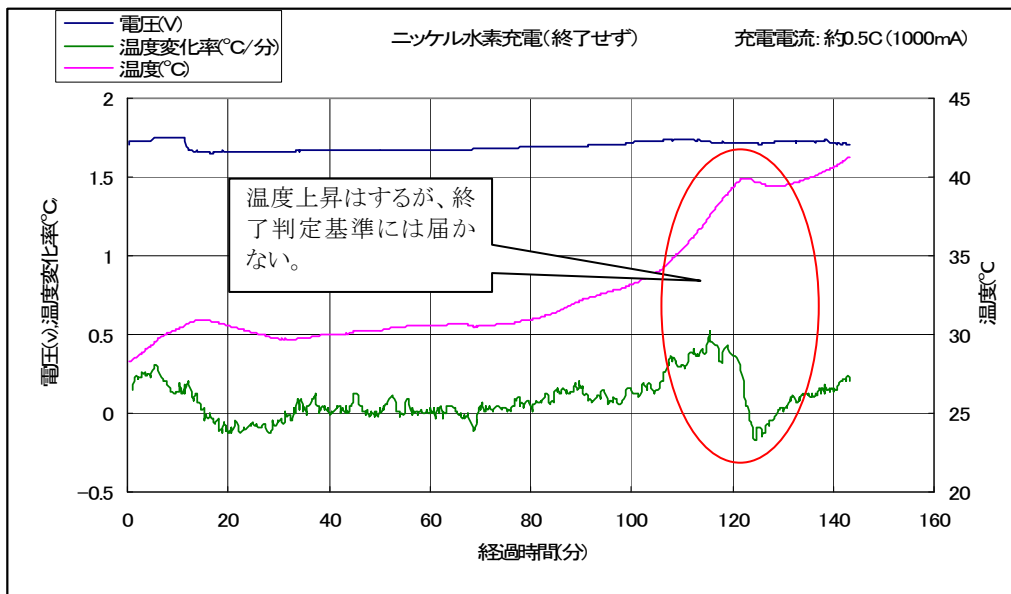
グラフ 3-3:ニッケル水素電池の充電と温度変化(ケース 2)

「温度変化率(上昇率) 0.5°C/分が 4 回連続で超えたとき、あるいは測定温度が 52°Cを超えたとき。」を充電終止条件にしたとき、**グラフ 3-4**では充電終了しているが、**グラフ 3-5**では最終的に 2300mAh分のエネルギーを流したにもかかわらず充電終了しない。

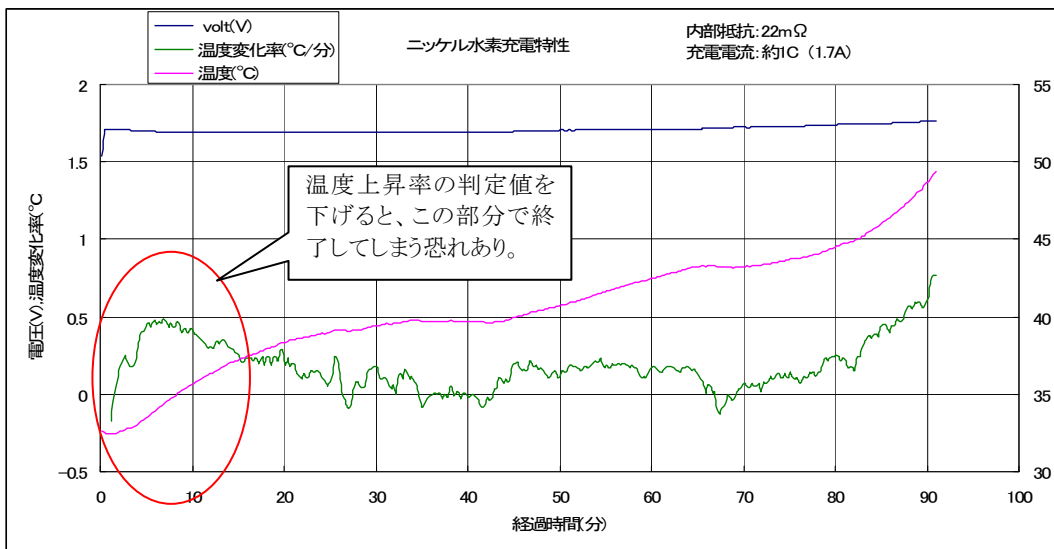


グラフ 3-4:ニッケル水素電池の充電例(ケース3)

この違いは充電電流の差によるところが大きい。グラフ 3-4では約 1C充電を行っているのに対しグラフ 3-5では約 0.5Cである。急速充電のように大きな電流を加えると満充電付近での温度上昇率が大きくなり、今回の判定基準で充電終了した。グラフ 3-5でも温度上昇の傾向はわかるが、この条件での温度上昇で充電終了判定とした場合、グラフ 3-6に示すように、電源開始時にも充電終了時と似た温度上昇があり、1C程度の急速充電では満充電になる手前で終了してしまうおそれがある。この判定基準に対しては充電電流との関連があり、充電終了判定を充電電流の関数にしたところで、今度は電池の容量によって温度上昇率がことなるので、一筋縄では行かない難しさがある。



グラフ 3-5 ニッケル水素電池の充電例(ケース4) 自動終了せず



グラフ 3-6: ニッケル水素電池の充電例(ケース5)

その電池の定格容量に対する充電値(C)値対して温度上昇率がことなるので、市販の充電器はその対象となる充電器以外の電池に対して正確な充電終了判定ができないであろう。

できるだけ電池の定格容量のそろったモノを使うと充電器は共通のものを使える可能性がある。

ということで、この充電器モードでは「**充電電流設定値を 1C に設定する。**」という使い方になる。

3.4.6.2 リチウムイオン電池の場合

リチウムイオン電池は充電電圧の上限さえ守ればよいのでそれほど難しい充電方式ではない。定電流にするのは充電開始時のみ大きな電流が流れることもあるので、それを適当に制限する目的だ。だいたい 1C 程度の電流リミットがあればいいことになる。定電流(電流制限)の動作している時間はそれほど長くない、ほとんどの充電区間は定電圧充電である。従って満充電に近くなると自然と充電電流は小さくなる。

逆に充電電圧の最大値がほぼ 4.2V なので急速充電は基本的にできないことになるが、「§ 2.2.2 充電方法(p10)」にもあるように 120 分で容量の 90% は充電されているとみなせる。これは電池のトータル容量には依存しないので時間だけを気にしておけばよく、そういった意味ではわかりやすいのかもしれない。

この装置においてはリチウム電池の場合放っておいても自動的に充電電流は流れなくなるので特に装置側で充電を終わることはしていない。つまり充電電流を見て、数十 mA オーダーまで下がっているようであればほぼ充電は完了していると見なせる。

3.4.7 放電制御

3.4.7.1 ニッケル水素電池の場合

ここで放電時に気をつけることは終止電圧で放電をやめることである。過放電も電池にとっては大敵です(詳細は § 2.1.5(P6)参照)。ということで、今回ニッケル水素系電池の放電終止電圧は **0.9V**とした。この電圧を一瞬でも下回ると放電終了と見なして制御を終える。

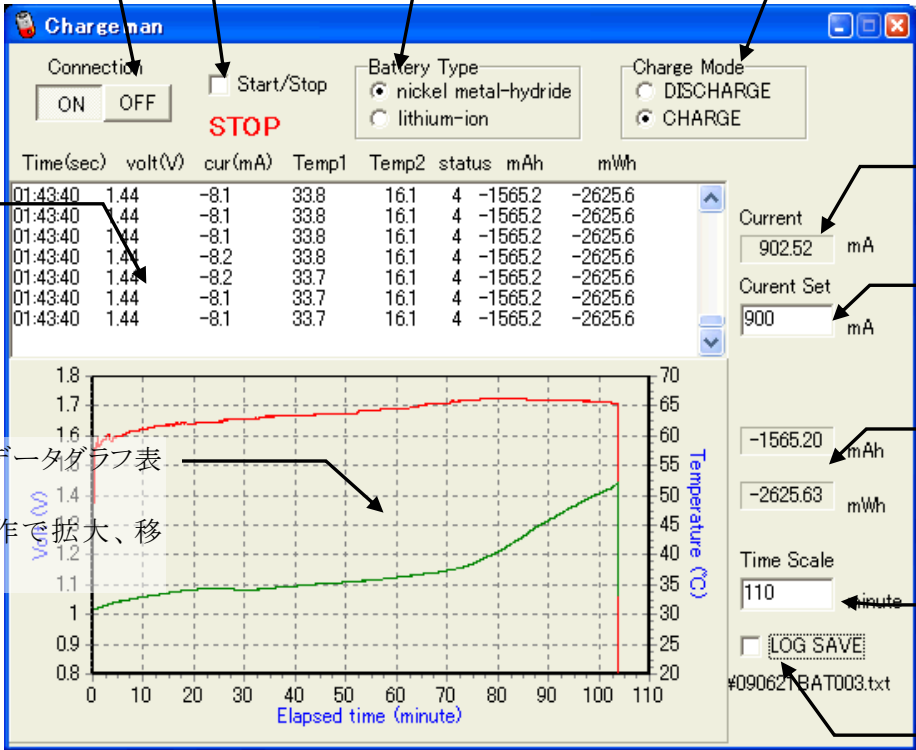
3.4.7.2 リチウムイオン電池の場合

リチウムイオン電池の場合もニッケル水素系電池と同じように過放電には弱い。放電電圧下限は **2.9V**とする。

この電圧を一瞬でも下回ると放電終了と見なす。

4 PC TOOL

簡易充放電器の簡単な制御だけでなく、装置からのデータをロギングしてデータをファイルとして保存したり、グラフ表示したりする。



測定開始・停止チェック

USBIF の接続/接続断

ニッケル水素電池かリチウムイオン電池の選択

充電か放電の選択

実際に測定した電流値

充電時の電流値の設定

積算電流・積算電力値
符号がマイナスの時は充電時

グラフ表示の時間軸スケールの設定

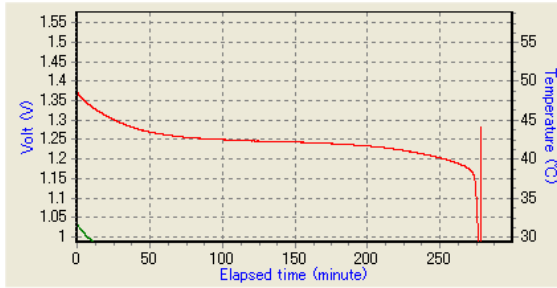
測定データ保存時にチェックを入れる。

取得したデータグラフ表示エリア。
マウス操作で拡大、移動可能

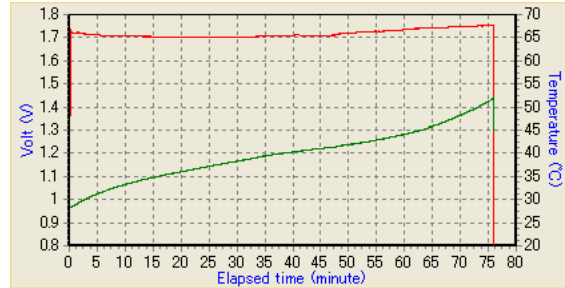
測定データの数値表示エリア

Time(sec)	volt(V)	cur(mA)	Temp1	Temp2	status	mAh	mWh
01:43:40	1.44	-8.1	33.8	16.1	4	-1565.2	-2625.6
01:43:40	1.44	-8.1	33.8	16.1	4	-1565.2	-2625.6
01:43:40	1.44	-8.1	33.8	16.1	4	-1565.2	-2625.6
01:43:40	1.44	-8.2	33.8	16.1	4	-1565.2	-2625.6
01:43:40	1.44	-8.1	33.7	16.1	4	-1565.2	-2625.6
01:43:40	1.44	-8.1	33.7	16.1	4	-1565.2	-2625.6

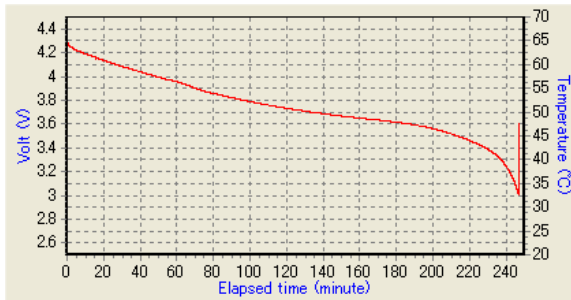
グラフ部は各動作モードによって異なる



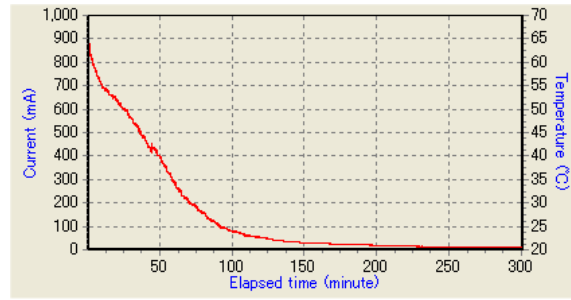
ニッケル水素電池放電



ニッケル水素電池充電



リチウムイオン電池放電

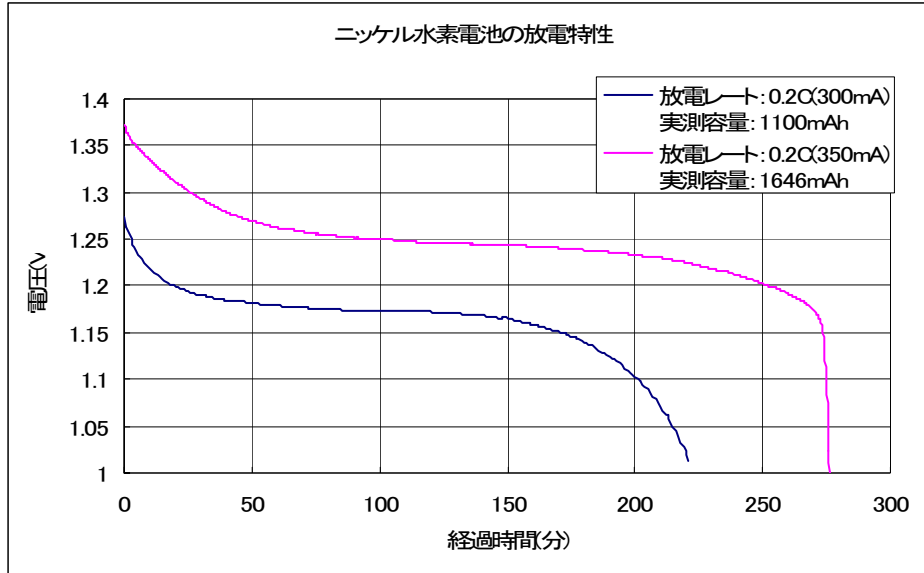


リチウムイオン電池充電

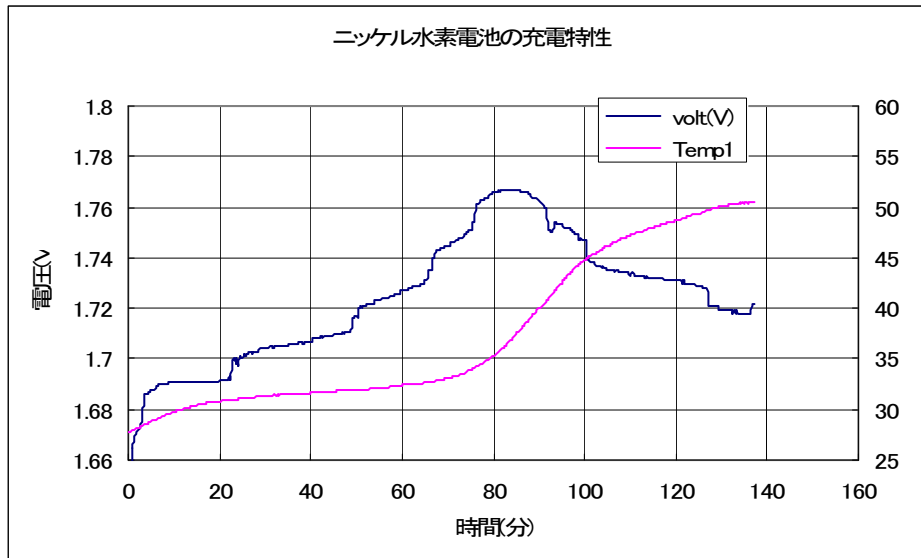
5 電池評価

5.1 ニッケル水素電池

放電特性



グラフ 5-1:ニッケル水素電池の放電実測特性



グラフ 5-2:ニッケル水素電池の充電実測特性

5.2 リチウムイオン電池

5.2.1 充放電特性

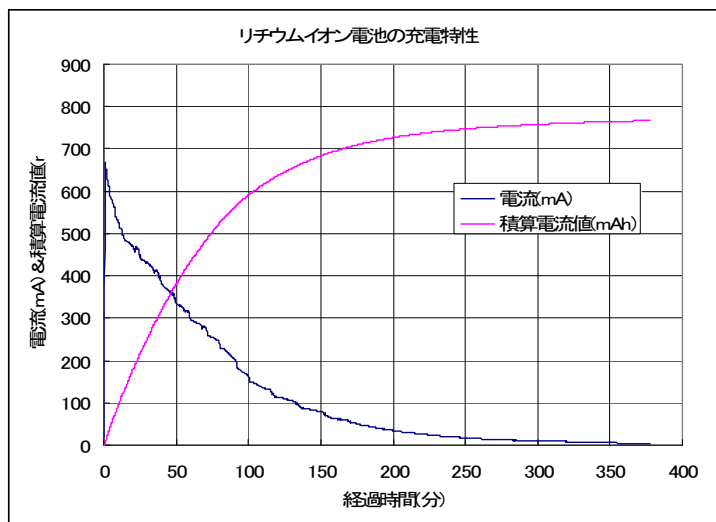
まず評価対象の電池はお手軽な携帯のリチウムイオン電池とした。ノート PC とかカメラも最近リチウムイオン電池を使っているが、1 セル(3.7V)で使用しているケースは少なく、今回製作した簡易充放電器で1セル以上のリチウム電池を測定することができないのだ。(ああ～なんてショボイのだろう(。`))

ちなみにこれらの評価試験の環境温度は常温(約 25°C～30°C)である。恒温層なんてないから温度試験はできないよ～(。`)

充電は基本的に定電圧充電。

充電し始めるとき大きな電流が流れることがあるので電流制限をかける程度。使用した電池の公称容量は780mAh。

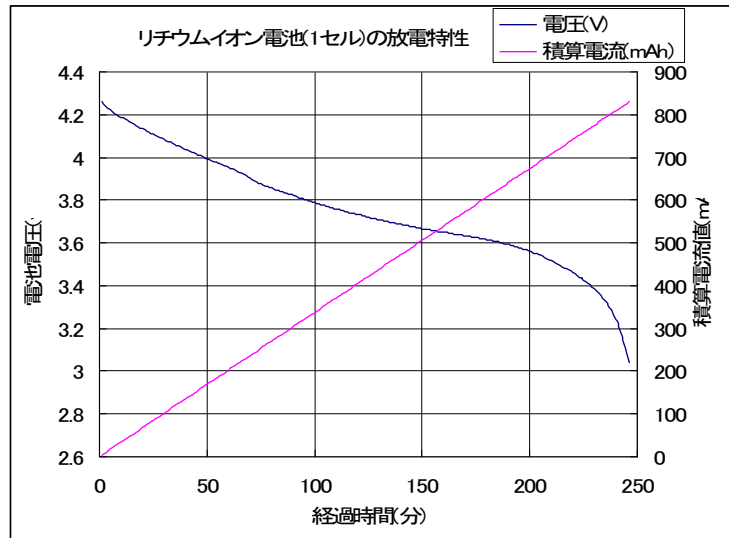
約 160 分で公称容量の 90%の充電ができる。定電圧(4.2V)で充電する限り、この時間は電池によらずほぼ同じくらい。(グラフ 5-3)



グラフ 5-3:リチウムイオン電池の充電特性(1セル)

グラフ 5-4は 1 セルリチウムイオン電池の放電特性を測定した代表的なものを示す。

公称:780mAh の電池ではあるが、この放電試験を行ったとき 800mAh を超えていた。これはこの前に充電したわけだが、充電 S/W のバグのため、ちょっと多めに充電してしまった影響だと考えられる。過充電には気をつけましょう。(。)



グラフ 5-4:リチウムイオン電池(1セル)放電特性

5.2.2 大容量リチウム電池を作ってみる

ここからはよい子はマネをしない方がいいかもしれない。自己責任でやってくださいね。

最近ノート PC をはじめ多種多様のモバイル機器が増えてきている。いつも課題となるのは電池(電力)源問題だ。ちょうどそこに携帯の電池がいくつか余っていた。ただ、携帯のものを使用するというだけに電力容量がちょっと足りない気もした。そこで、この電池をまとめればある程度大容量のバッテリーとして使用できるかもしれないと思い、まとめたときに本当に容量が増えるかどうか測定してみた。



写真 5-1:携帯の電池をまとめる

その前に注意しなくてはならないのは電池の特性を揃えるということである。電池にも製造ばらつきがあり、劣化したものを混ぜるとその電池に負荷が集中して危険な状態になるらしい。

そこで写真 5-2に示すバッテリーテスター(HIOKI 3555)で電池内部抵抗がほぼそろったモノをパラにまとめることにする。



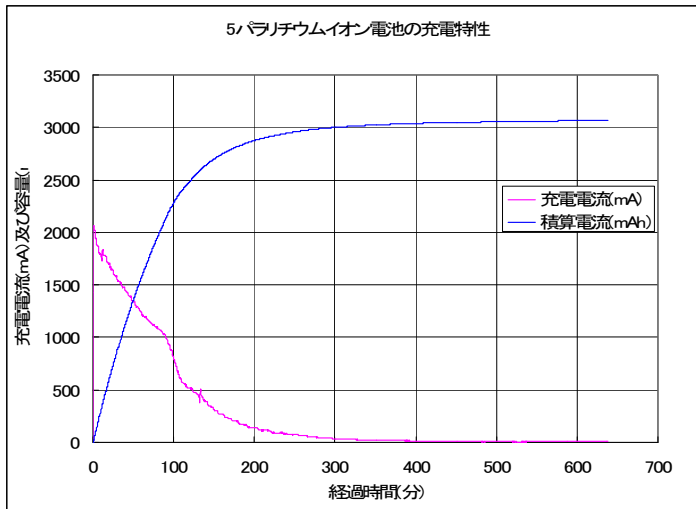
写真 5-2:電池テスター

ちなみにリチウム電池 1 セルあたりの内部抵抗は 160mΩくらいで、ニッケル水素は約 20mΩ程度。ということで、ニッケル水素電池の方が瞬間パワーは引き出せることになる。電池の劣化と共に容量が少なくなるのはもちろん内部抵抗も大きくなってくる。

最悪 10 倍くらいに抵抗値があがることもあるが、そのような電池は寿命だという判断していいだろう。

このリチウム電池は 1 セル 1 個あたり 160mΩでありパラに接続するとオームの法則通り全体として内部抵抗が小さくなる。今回 5 個パラで接続したので $160/5 \div 32\text{m}\Omega$ であり、実測もこの程度になった。そうなるとニッケル水素電池の内部抵抗に近くなり瞬間パワーも増す。

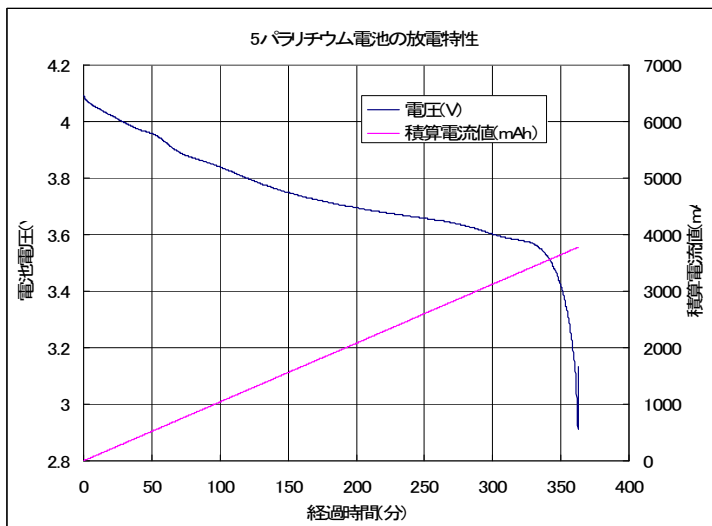
この携帯のリチウムイオン電池は 1 個あたり公称:780mAh であるから 5 個パラ接続で容量的には $780\text{mA} \times 5 = 3900\text{mAh}$ となるはず。まず充電特性から



グラフ 5-5:リチウムイオン電池を5パラにしたときの充電特性

グラフ 5-5に充電したときの特性を示す。期待値公容量値 3900mAhに対して約 3070mAhなので 78%の容量となった。まあ、それでも大きな部類にはいるのでこんなものかもしれない・・・ってここでは妥協してみる。(^^ゞ
90%充電で 120 分くらい要する。この辺は他のリチウム電池の充電特性と同じだ。

次にこの電池の放電特性を測定してみる。



グラフ 5-6:リチウムイオン電池を5パラにしたときの放電特性

全容量値を 3000mAh としたとき 0.2C である約 600mA を放電電流値として設定する。
グラフ 5-6に放電したときの特性を示す。充電時には 3000mAhと見ていたが、放電させてみると 3800mAhとなりほぼ期待容量値が得られた。おそらく充電開始時に残っていた容量があったと思われる。



さらなる大容量の電池を作ってみる。

$$780\text{mAh} \times 13 \text{ 個} = 10.14\text{Ah}$$

それを 2 つ使うことで

$$10.14\text{Ah} \times 2 \times 7.2\text{V} = 146\text{Wh}$$

の電池ができあがり！

これなら十分にビデオの電源として使えるかもです。(。)

6 まとめ

7 課題と対策

今回、なるべくインタフェースや構成を簡単にするために PSoC の USB IF 付き CY8C24794 を使用した。ただし、内部のデジタル・アナログブロックは十分とは言えないモノであった。

その具体的な問題点として

①4つあるデジタルブロックのうち $\Delta \Sigma$ A/D コンバータに2つ割り当てられてしまうため、残りは 8bit のシステムタイマーと 8bit の PWM 生成ブロックに割り当てた。

この PWM は充電時の電圧を設定するときを使用しているが決して分解能が高くない。従って収束させたい電圧(あるいは電流)にするために、収束地点が安定せず電圧(電流)がドリフトを繰り返すことも多い。長周期的に見ればほぼ定められた値になるのだが、短期間での電圧(電流)変動は電池にとって必ずしもいいことではない。この充電出力を制御する PWM としては少なくとも 16bit は必要と思われる。欲を言えば 24bit あれば高負荷時のこまかい設定ターゲットに対して安定した電力供給ができと思われる。

②USB を刺さずに動作開始してしまった後で USB を刺しても PC にデータを取り込むことができない。これは PSoC(CY8C24794)の仕様で USB IF の初期化を電源投入時におこなっているからで、相手がないとその初期化部分をパスさせており、後で認識させることが難しい。(できなくもないが、修正するコード領域が足りないかもしれない。)

③電源入力系統には必ずレギュレータを入れること。

今回 PSoC を 3.3V で動作させた。これは 5V 入力に対して 3.3V のレギュレータ出力にして A/D コンバータのリファレンス電圧を安定させるためである。この部分に関しては特に問題は無かったが、アナログ部、特に OPAMP の電源は 5V 直から取っている。そのため大きく影響擦るわけではないが、オフセット調整の値が若干変わることもある。この辺は電源電圧を高くしてアナログ電源系にもレギュレータで電圧を安定化させたものを使用するのが好ましいと思われる。

今回製作して簡易充放電器は USB 電源でも動作することを前提としていたので 5V を全体の電源とした。

④電圧測定用回路は単に入力電圧を抵抗で分圧した値を PSoC の A/D コンバータへ入力した。この A/D コンバータの入力範囲は $1/2v_{dd} \pm \text{bandGap}(1.3V)$ ということを入力範囲は

$1.65 \pm 1.3 \rightarrow 0.35V \sim 2.95V$ となり、下限 0.35V は必ずしも小さくはない。今回オフセット調整するためのオペアンプが無かったためこのままの範囲を使用することにしたが、次回はちゃんと 0V から測定できるようにしたいものである。

⑤放電電流値は PSoC の DAC の出力値をリファレンスとしている。この DAC は設定電圧の低いところでの直線性があまりよくないようで、DAC 設定値の小さいところでは実際の動作値と誤差が大きくなっている。PSoC の DAC ブロックはスイッチドキャパシタを利用しているのでこの辺の動作誤差が影響しているものと思われる。これを静的な R・2R ラダー回路を使用したものを使えばもう少し精度が向上すると思われる。

次回の「充放電器」に対する仕様は次の通り。

メイン CPU	M16C/M30627FJPG (ROM512kbyte, RAM31kbyte)
サブ CPU	PSoC CY8C24794 (USB HID IF 用)
CPLD	EPM240T100C5N (PWM 生成用)
OS	Smalght OS (μ iTron のサブセット)
表示	モノクロドットマトリクス表示(128×64) or キャラクタ LCD(20×4 行)
放電部	
被測定数	8 本(個)
放電モード	低電流モード/定抵抗負荷モード(8 セット独立設定可能)
測定範囲	0V~20V, 0A~10A /セット
放電終止電圧監視	ニッケル水素、リチウムイオン電池共に使用セル数に応じた設定を する。(可変)
充電部	
供給能力	0V~20V, 0A~10A /セット
電池充電方式	ニッケル水素: ΔV 、温度監視ハイブリッド型 リチウムイオン: 定電流・定電圧(値は設定可能)方式
その他システム	
ストレージ	SD カードメモリにデータログを保存
温度センサー	10°C~80°C ×8
外部 IF	USB (HID)
電圧測定精度	0.01V
電流測定精度	1mA