

ケースレー2450型ソースメータによる バッテリーの充放電サイクル・テスト

はじめに

充電（二次）バッテリー（これ以降、「バッテリー」と表記）は、ビデオ・ゲームのコントローラ、デジタル・カメラ、リモコンなどの電子デバイスの使い切り乾電池の代わりに広く使用されています。代表的なバッテリーの種類としては、リチウムイオン（Li-ion）、ニッケル水素（Ni-MH）、ニッカド（NiCd）などがあります。二次電池の特性評価では、一般に、放電と充電サイクルをテストします。サイクル・テストを行うことで、内部化学、容量、利用可能サイクル数、寿命などのバッテリーに関する情報が得られます。製造テストでの充放電サイクルは、バッテリーの仕様を検証し、欠陥がないことを確認するために行われます。

バッテリーの一般的な充放電テストでは、プログラマブル電源、電子負荷、電圧計、電流計が必要になります。2450型ソースメータは、ソース/シンク電流機能、電圧/電流の測定機能を兼ね備えた計測器であるため、バッテリー・テストを簡素化することができます。2450型を使用することで数多くの計測器をラックに組み込むことなく、一台の計測器をセットアップするだけで済みます。すなわち、電流印加によりバッテリーを充電し、電力を消費することでバッテリーを放電し、バッテリーの電圧と負荷電流をモニタすることができます。2450型の出力は、 $\pm 21V @ \pm 1.05A$ 、または $\pm 210V @ \pm 105mA$ です。2450型を使用したバッテリーの充放電サイクル・テストの一般的なシステム例を、図1に示します。



図1. 2450型ソースメータを使用したバッテリー・テストの例

バッテリーの充放電

一定の電流による充放電のレートは、バッテリー容量の単位として定義されており、これはバッテリーが保存できる電気量になります。この容量はミリアンペア時（mAH）で規定されており、放電、または負荷電流の単位として表わされます。1時間ですべてのバッテリーを放電しきる放電電流のレートは、Cレートとして知られています。例えば、1000mAHの定格のバッテリーは、1Cで放電した場合、1000mAを1時間出力します。500mAHのバッテリーを50mAで放電すると、0.1Cのレートで放電することになるため、50mAでは10時間供給できます。

テストの概要

充電、放電の両方のサイクルにおいて、2450型ソースメータは電圧を印加し、電流を測定するように設定できます。充電および放電サイクルの簡略回路を図2に示します。

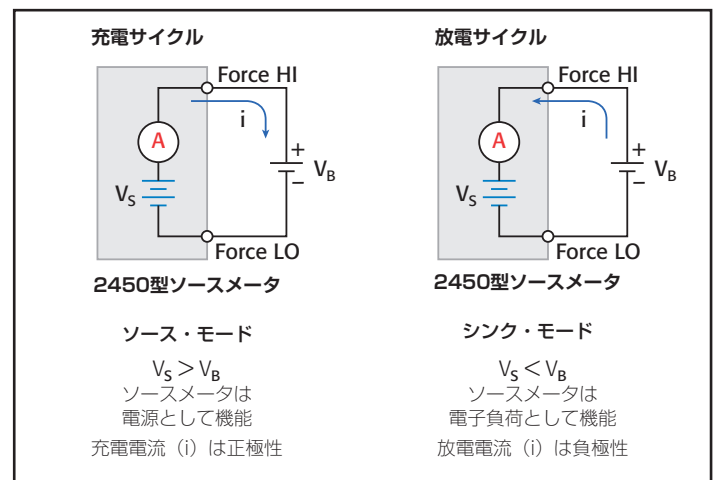


図2. 充放電の回路例

通常、バッテリーは一定の電流で充電します。2450型を使用する場合、電圧ソースをバッテリーの定格電圧に、充電電流を回路リミット値に設定することで実行できます。テスト開始時、バッテリーの電圧は2450型で設定した電圧出力以下の値になります。この電位差のために、電流値はユーザ設定の電流リミットにただちに制限されます。電流リミットが働いている間、2450型は設定された電圧レベルになるまで定電流源として機能します。バッテリーがフル充電に近くなると、電流はゼロまたはゼロ近くまで低下します。バッテリーへの損傷を防ぐため、バッテリーの過充電は十分に注意する必要があります。

バッテリーの放電では、2450型ソースメータは電力を消費するのではなく、電流を供給するので、電流源として機能します。2450型の電圧源は、バッテリー電圧よりも低いレベルに設定します。電流リミットは放電レートに設定します。出力をオンにすると、バッテリーからの電流は2450型のHI端子から流れ込みます。このため、電流の読み値は負の極性になります。放電電流は、バッテリーの電圧が2450型の電圧ソース設定に低下するまで一定を保ちます。

バッテリーの接続方法

2450型とバッテリーの接続例を、**図3**に示します。2450型とバッテリーを4線接続（またはリモート・センス）することにより、リード線の抵抗による影響を抑えます。これにより、バッテリーの電圧は、バッテリー端子近くでの測定値になります。

2450型のForce HIとSense HIの出力端子にはバッテリーの正（+）端子を、Sense LOとForce LO出力にはバッテリーの負（-）端子を接続します。

2450型ソースメータの出力がオフの場合、ハイ・インピーダンス（High Z）出力がオフに設定されていることを確認します。High Z出力がオフになっていると、出力リレーは開き、出力はオフになります。こうすることで、出力オフでバッテリーから間違えて流れ込まないようにすることができます。出力オフの状態からハイに設定するには、MENUボタンを押し、Source Settingsを選択します。次に、High Impedanceを選択し、HOMEボタンを押します。

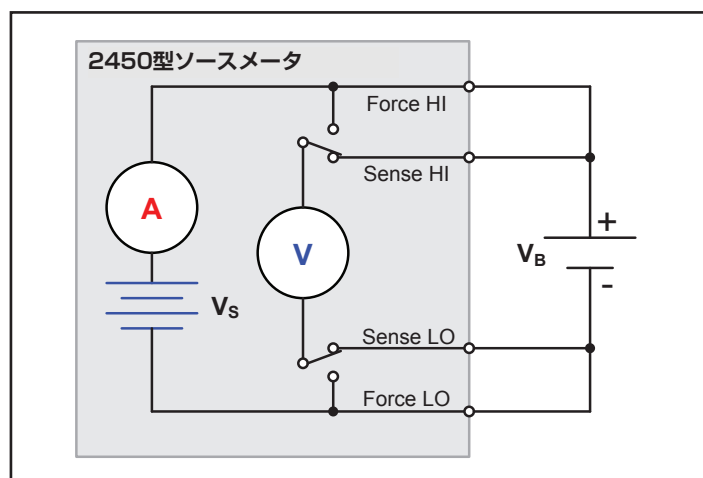


図3. 2450型とバッテリーの接続例

放電サイクルの自動化

充電／放電サイクルは数時間かかることがあるため、テストの自動化が重要になります。計測器がサポートしている通信インタフェース（GPIB、USB、またはEthernet）を使用してテストします。リモート通信インタフェースがある後部パネルを、**図4**に示します。

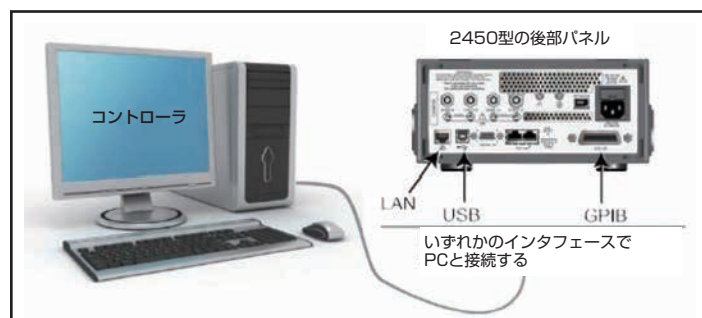


図4. 2450型のリモート・インタフェース

充電／放電では、以下の手順で2450型をプログラムします。

1. 4線式で測定するように設定します。
2. 2450型で電流を測定するように設定します。これにより、負荷電流をモニタできます。
3. Output Off StateでHi Impedanceを使用します。Hi Z出力がオフになっていると出力リレーは開き、出力はオフになります。こうすることで、出力オフでバッテリーから間違えて流れ込まないようにすることができます。
4. 2450型で電圧を出力するように設定します。電圧を出力するように設定されていますが、目的の電圧に達するまでは電流制限がかかるため、定電流で動作します。

バッテリーの充電では、 $V_S > V_B$

バッテリーの放電では、 $V_S < V_B$

5. 電圧源のリードバック機能をオンにします。これにより、2450型の電圧計は充電または放電でもバッテリー電圧を測定することができます。
6. 電流リミット（コンプライアンス）を、バッテリーの充放電における電流レベルに設定します。これがテストにおける負荷電流になります。
7. 負荷電流、電圧源のリードバック電圧、相対タイムスタンプを読み込みます。
8. バッテリー電圧が所定の電圧レベルになるまでモニタし、テストを停止します。

2450型ソースメータを使用した2300mAH AA（単三形）バッテリーの放電

2300mAHのAA（単三形）、1.2Vのバッテリーを例に、2450型ソースメータでバッテリーを放電する方法について説明します。ソースメータは、先の項で説明した手順（4線式、ソース電圧、電流測定など）でプログラムしています。このテストでは、2300mAHの単三バッテリーを、460mAの負荷電流、0.2Cのレートで放電します。バッテリー電圧が規定のレベルである1Vになるまで、10秒ごとにバッテリー電圧、負荷電流、相対タイムスタンプを読み取ります。このように測定した2300mAHのバッテリーの放電特性を、**図5**に示します。

パス上で読み値が確認できるだけでなく、2450型はユーザ・インタフェース上に、負荷電流、バッテリー電圧、テスト実行の経過時間も同度に表示することができます。2450型の大型ディスプレイにわかりやすく表示される、単三バッテリーの測定値の例を図6に示します。

放電特性実行で使用するコードの例を、付録に記します。2450型では、SCPIまたはTSP (Test Script Processor) コマンドのどちらでもプログラムを作成できますが、この例では、ソフトウェア・ツールのTest Script Builderを使用したTSPコマンドで書かれています。このソフトウェア・ツールは2450型に付属しており、TSPのコード、スクリプトの作成、編集が行えます。さらに、コマンドを送ったり、2450型からデータを受け取るなど、迅速な制御のためのコンソールなども備えています。

まとめ

2450型は、正確な4象限、ハイパワー出力、電流/電圧の正確な同時測定などの機能があるため、バッテリーの充放電テストに理想的なツールです。1台の計測器でバッテリーがテストできるため、テストのセットアップが簡素化され、プログラムに要する時間が短縮でき、ラック・スペースも節約できます。

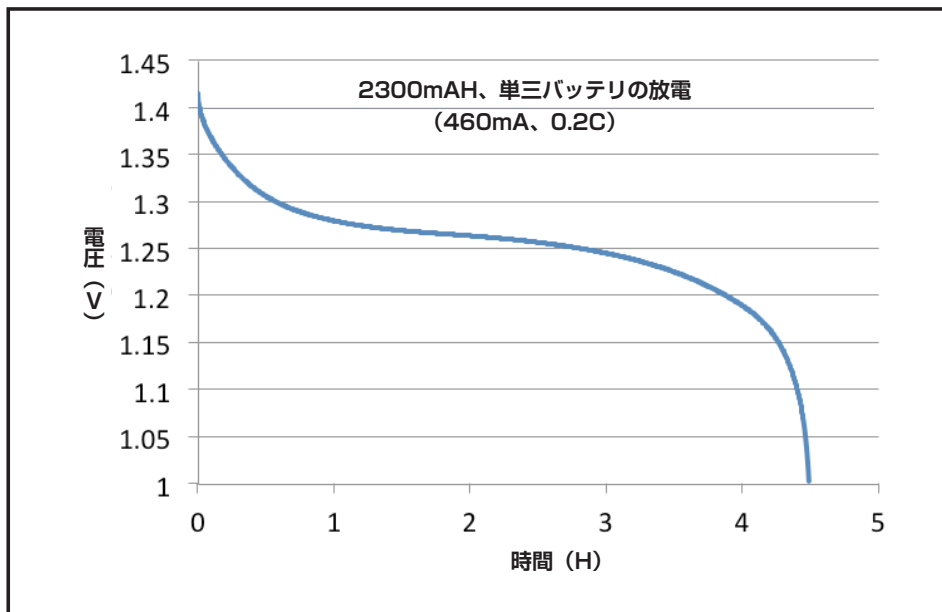


図5. 2300mAh単三バッテリーの放電特性の例

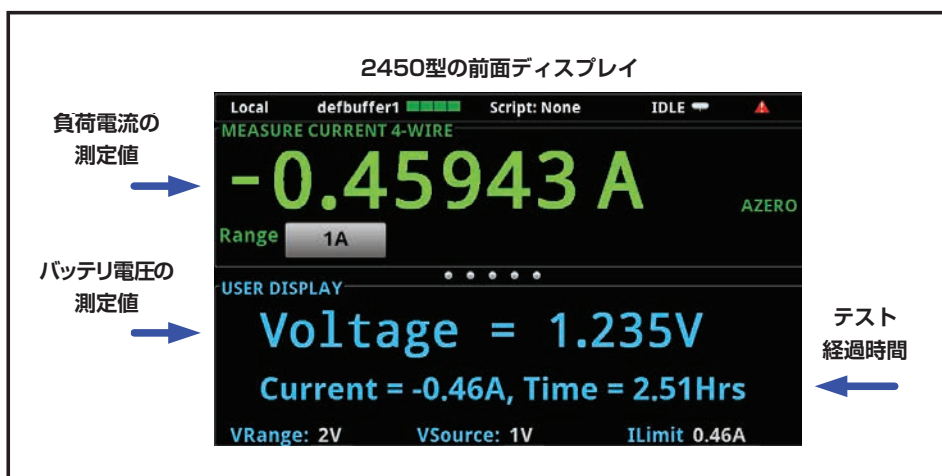


図6. 2450型ソースメータのディスプレイに表示された、単三バッテリー放電における負荷電流、バッテリー電圧、テスト経過時間の例

付録：2300mAhのバッテリー放電特性のためのTSPコード例

```
--Reset the instrument and clear the buffer
reset()

--Source Settings
smu.source.func = smu.FUNC_DC_VOLTAGE
smu.source.offmode = smu.OFFMODE_HIGHZ
smu.source.level = 1
smu.source.range = 2
smu.source.readback = smu.ON
smu.source.ilimit.level = 460e-3

--Measurement Settings
smu.measure.func = smu.FUNC_DC_CURRENT
smu.measure.range = 460e-3
smu.measure.sense = smu.SENSE_4WIRE

--Set the voltage limit for the battery to stop discharging
--Set the variable for number of iterations
voltLimit = 1.0
iteration = 1

--Turn on the source output
smu.source.output = smu.ON

--Change display to user screen
display.changescreen(display.SCREEN_USER_SWIPE)

--Keep taking readings in the while loop until the measured voltage
--is equal to the voltage limit
while true do

    --Take a reading and get the current, voltage and time
    curr = smu.measure.read(defbuffer1)
    volt = defbuffer1.sourcevalues[iteration]
    time = defbuffer1.relativetimestamps[iteration]
    hours = time/3600

    --Print the # of completed cycles, the voltage and the time for
    --the iteration. Display information on front panel
    print("Completed Cycles: ", iteration, "Voltage: ", volt, "Time: ", time)

    display.settext(display.TEXT1, string.format("Voltage = %.4fV", volt))
    display.settext(display.TEXT2, string.format("Current = %.2fA, Time = %.2fHrs", curr, hours))

    --Increment the number of iterations and wait 10 seconds
    --Compare the measured voltage to the voltage limit
    --Exit the loop if it is
    if volt <= voltLimit then
        break
    end

    iteration = iteration + 1
    delay(10)
end

--Turn the output off when the voltage limit is reached
smu.source.output = smu.OFF

--Print out the measured values in a 4 column format
print("\nIteration:\tCurrent:\tVoltage:\tTime:\n")
for i = 1, defbuffer1.n do
    print(i, '\t', defbuffer1[i], '\t', defbuffer1.sourcevalues[i], '\t', defbuffer1.relativetimestamps[i])
end
```

KEITHLEY

A Tektronix Company

www.keithley.jp

テクトロニクス/ケースレーインストルメンツお客様コールセンター

TEL : 0120-441-046 電話受付時間/9:00~12:00・13:00~18:00(土・日・祝・弊社休業日を除く)

〒108-6106 東京都港区港南2-15-2 品川インターシティ B棟6階

記載内容は予告なく変更することがありますので、あらかじめご了承ください。

Copyright © Keithley Instruments. All rights reserved. 記載された製品名はすべて各社の商標あるいは登録商標です。

Number 3221 2013年8月