

利用吉时利2450型触摸屏数字源表，实现可充电电池的充电/放电循环

概述

可充电电池或二次电池常用来取代电子设备（如视频游戏控制器、数码相机和遥控器）中使用的一次性电池。常见的可充电电池种类包括锂离子电池、镍氢电池和镍镉电池。通常利用放电和充电循环来对二次电池特征进行测试。循环测试提供电池的有关信息，如内部化学物质、容量、可用周期数及寿命。在生产测试中，往往通过充电/放电循环来验证电池的性能指标，并确保其没有缺陷。

典型的电池充电/放电测试设置通常包括可编程电源、电子负载、电压表和电流表。使用单一仪器——吉时利2450型触摸屏数字源表可以简化电池测试，该仪器具有灵活的源/阱电流以及测量电压和电流。利用2450型触摸屏数字源表，用户只需设置单一设备而非整个机架设备。因此，2450型仪器可以通过源电流为电池充电，通过消耗功率为电池放电，并监控电池的电压和负载电流。2450型仪器可以输出 $\pm 21V @ \pm 1.05A$ 或 $\pm 210V @ \pm 105mA$ 。图1给出使用2450型仪器的典型充电/放电循环系统。



图1 利用2450型触摸屏数字源表进行电池测试

电池充电/放电

恒定电流充电和放电的速率定义为电池容量，即电池可以存储的电荷数量。电池容量由可用毫安小时(mAH)来说明，应当用放电或负载、电流来表示。在1小时内对整个电池进行放电时的放电电流称为C速率。例如，额定1000mAH的电池，如果以1C速率放电，那么它在1小时的输出即为1000mA。如果500mAH电池在50mA时放电，然后以C/10(0.1C)速率放电，那么它可以作为50mA源供电10小时。

测试说明

对于充电和放电循环，2450型触摸屏数字源表可以配置为源电压并测量电流。图2给出简化的充电和放电循环电路示意图。

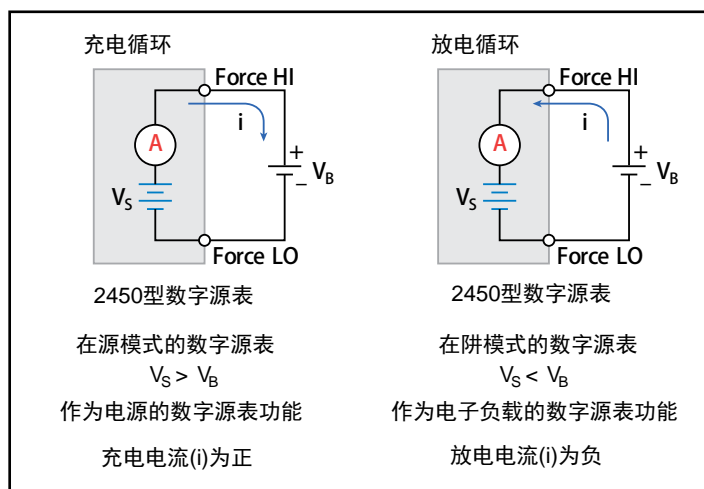


图2 充电放电电路示意图

电池充电通常使用恒定电流。利用2450型触摸屏数字源表可以实现充电，即把它作为电压源，设置电池额定电压，并设置期望的充电电流作为电流限幅。测试开始时，电池电压低于2450型仪器的电压输出设置。因此，在这个压差作用下，电流立刻达到用户定义的电流限幅。当达到电流限幅时，2450型仪器就作为恒流源，直达到达设定的电压电平。随着电池完全充电，电流将下降，直达到达零或接近零。为了预防安全隐患或损坏电池，必须注意不要过度充电。

当电池放电时，2450型触摸屏数字源表作为阱，因为它消耗电能而不是提供电源。将2450型仪器电压源设置为低于电池电压，电流限幅设置为放电速率。当输出启用时，电流从电池流入2450型仪器的高电平端口。因此，电流读数将是负数。放电电流应当保持恒定，直到电池电压下降至2450型仪器的电压源设定值。

与电池连接

为了建立测试，需将2450型触摸屏数字源表与电池进行连接，如图3所示。将2450型仪器端口与电池进行4线或程控检测连接，以可以消除引线电阻的影响。这样，测得的电池电压尽可能接近其端口电压。

2450型仪器的Force HI和Sense HI输出端子与电池正极(+)端子相连，其Sense LO和Force LO输出端子与电池负极(-)端子相连。

当2450型触摸屏数字源表输出关闭后，确保其设置在高阻抗(High Z)输出关闭状态。选定高阻抗(High Z)输出关闭状态，当输出关闭时，将开启输出继电器。这将防止输出关闭后电池出现泄漏。要将输出关闭状态设置为高阻抗，首先按压MENU按键，然后选择源设置，在选择高阻抗，最后按压HOME按键。

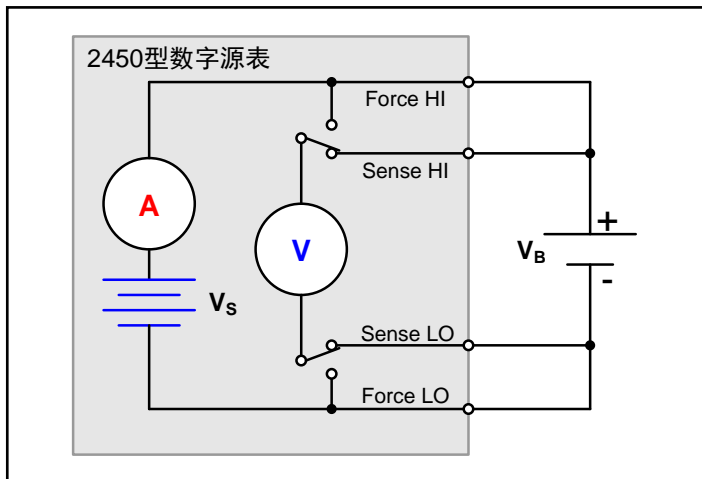


图3 2450型触摸屏数字源表与电池连接示意图

自动放电循环

电池的充电和放电循环通常需要几个小时，因此自动化测试非常重要。利用2450型仪器的任何通信接口(GPIB、USB或以太网)，都可执行测试。程控通信接口的后面板连接位置如图4所示。

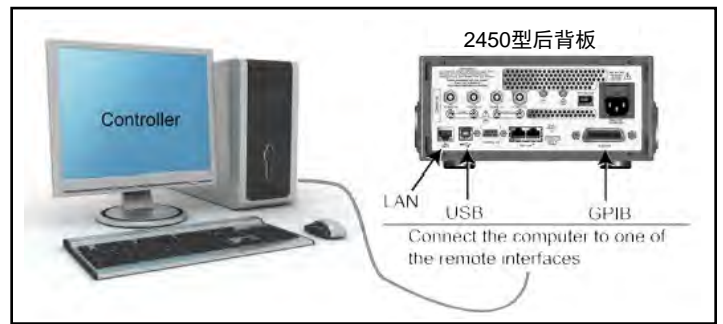


图4 2450型触摸屏数字源表程控接口连接

为实现充电和放电，应当对2450型仪器进行设置，执行以下步骤：

1. 设置4线配置测量。
2. 设置2450型仪器测量电流，使得用户可以监视负载电流。
3. 使用高阻抗输出关闭。当Output（输出）关闭后，这个输出关闭状态将开启输出继电器。这将防止电池连接后输出关闭时电池出现泄漏。
4. 设置2450型一起的输出电压。即使仪器设置为输出电压，它仍将在恒流模式运行，因为在达到期望电压之前，它将处于电流限幅之内。
对于电池充电， $V_s > V_B$ 。
对于电池放电， $V_s < V_B$ 。
5. 开启电压源回读功能，使得电池充电或放电时，2450型电压表能够测量电池电压。
6. 将电流限幅（或极限）设置为电池充电或放电的电流水平。这是测试的负载电流。
7. 回读负载电流、源回读电压以及相应的时间戳。
8. 监视电压，直到电池电压达到期望的电压电平，并停止测试。

使用2450型触摸屏数字源表对2300mAH AA电池进行放电

以2300mAH AA (1.2V)电池为例，说明如何利用2450型触摸屏数字源表进行电池放电。根据上面列举的步骤对仪器进行编程(4线、源电压、测量电流等)。对于本次测试，使用460mA负载电流，以0.2C速率对2300mAH AA电池进行放电。每隔10秒钟采集一次电池电压、负载电流和相对时间读数，直到电池电压达到指定电平(1V)。2300mAH电池放电特性测量结果如图5所示。

除了通过总线监视读数，在测试过程中，2450型触摸屏数字源表还能同时通过用户界面，显示负载电流、电池电压以及经历的测试时间。请看在2450型仪器显示屏上呈现的大字体、易读测量结果，如图6所示。

附录给出生成电池放电特性所用的代码。虽然2450型触摸屏数字源表的灵活性使得用户可以选择SCPI或TSP®(测试脚本处理器)指令进行编程，但这个代码是通过Test Script Builder软件工具、使用TSP指令编写的。这是与2450型仪器一起提供的免费软件工具，可帮助用户创建和修改TSP代码和脚本。它具有即时仪表控制台，可向仪器发送指令，从仪器接收数据。

结束语

2450型触摸屏数字源表是对可充电电池的充电和放电循环进行测试的理想工具，因为它具有精确地4象限、高功率输出，能够精确地测量电流和电压。利用单一仪器进行电池测试，不仅简化测试设置，缩短编程时间，而且节省机架空间。

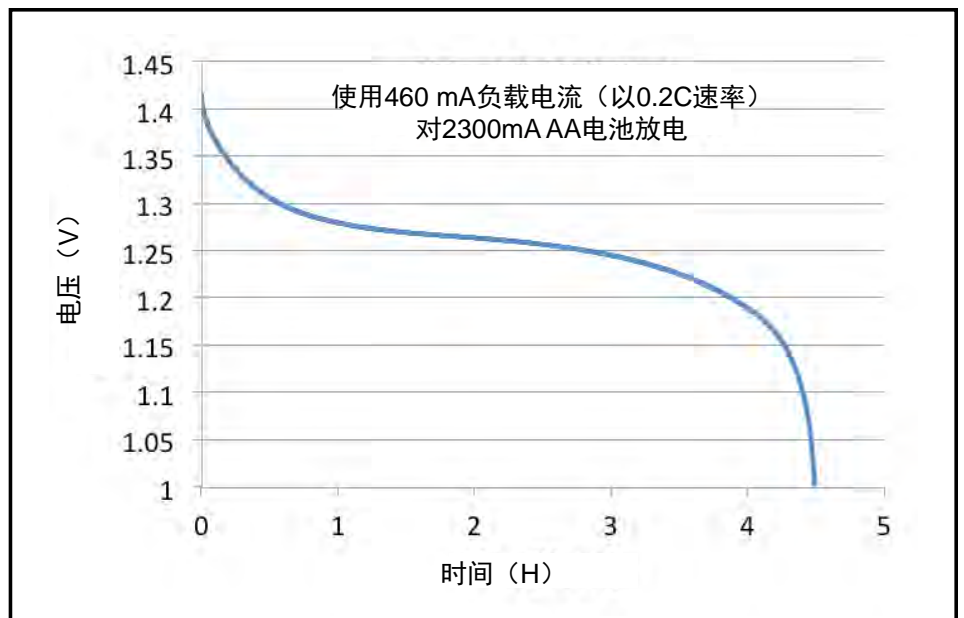


图5 2300mAH AA电池放电特性

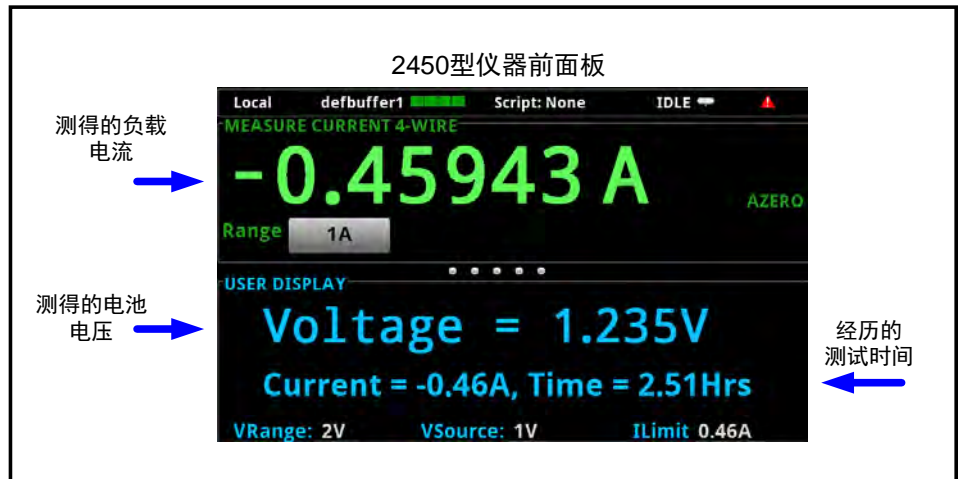


图6 2450型触摸屏数字源表显示AA电池放电时的负载电流、电池电压和经历的测试时间

附录：用于生成2300mAH电池放电特性的TSP代码

```
-- 仪器复位，并清除缓冲区
reset()

-- 源设置
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

-- 测量设置
smu.measure.func = smu.FUNC_DC_CURRENT
smu.measure.range = 460e-3
smu.measure.sense = smu.SENSE_4WIRE

-- 设置电池停止放电电压限幅
-- 设置变量循环次数
voltLimit = 1.0
iteration = 1

-- 开启源输出
smu.source.output = smu.ON

-- 将显示屏变更到用户屏幕
display.changescreen(display.SCREEN_USER_SWIPE)

-- 在循环中保持读数，直到测量电压
-- 等于电压限幅
while true do

    -- 读取读数，并得到电流、电压和时间
    curr = smu.measure.read(defbuffer1)
    volt = defbuffer1.sourcevalues[iteration]
    time = defbuffer1.relativetimestamps[iteration]
    hours = time/3600

    -- 打印完整的周期、电压和循环时间数值。
    -- 在前面板显示信息
    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))

    -- 增加循环次数，并等待10秒
    -- 对测量电压和电压限幅进行比较Compare the measured voltage to the voltage limit
    -- 退出循环，如果满足
    if volt <= voltLimit then
        break
    end

    iteration = iteration + 1
    delay(10)
end

-- 当达到电压限幅时，关闭输出
smu.source.output = smu.OFF

-- 以4栏格式打印测量值
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
```