

Eason Zhou

**摘要**

本应用手册旨在介绍基于 MSPM0L130x 系列的 1 级测量仪表解决方案。它仅检测电压以直接计算片上系统 (SoC)。其中包含解决方案特性、硬件简介、GUI 简介、软件简介和评估指导。

本应用手册中讨论的工程配套资料可从以下 URL 下载：<https://www.ti.com/cn/lit/zip/slaaee3>。

**内容**

<b>1 引言</b>	<b>2</b>
<b>2 测量仪表硬件简介</b>	<b>4</b>
<b>3 测量仪表软件简介</b>	<b>5</b>
3.1 测量仪表算法简介	5
3.2 测量仪表 GUI 简介	6
<b>4 MSPM0 测量仪表评估步骤</b>	<b>8</b>
4.1 第 1 步：硬件准备	8
4.2 步骤 2：获取电池模型	8
4.3 步骤 3：输入自定义配置	11
4.4 步骤 4：评估	11
<b>5 MSPM0 测量仪表解决方案测试结果</b>	<b>15</b>
5.1 性能测试	15
5.2 电流消耗测试	16

**插图清单**

图 1-1. MSPM0 测量仪表硬件板	2
图 1-2. MSPM0 测量仪表软件工程	3
图 1-3. MSPM0 测量仪表 GUI 工程	3
图 2-1. MSPM0 测量仪表板方框图	4
图 2-2. 测量仪表板说明	4
图 3-1. MSPM0 测量仪表软件工程视图	5
图 3-2. 电池模型和 SoC-OCV 表	5
图 3-3. VGauge 软件流程	6
图 3-4. MCU COM Tool 功能	7
图 3-5. SM COM Tool 功能	7
图 4-1. 脉冲放电测试用例	9
图 4-2. 获取电池模型的硬件结构	9
图 4-3. 电池电路表生成	10
图 4-4. 电池电路表输入	10
图 4-5. tBattParamsConfig 结构	11
图 4-6. 测量仪表模式设置	12
图 4-7. 检测数据输入模式结构	12
图 4-8. 闪存数据输入模式结构	12
图 4-9. 电池运行文件生成	13
图 4-10. 电池运行文件复制	13
图 4-11. 更改时间阶跃的代码更改	13
图 4-12. 通信数据输入模式结构	14
图 4-13. 通信数据输入	14

图 5-1. 电池测试用例.....	15
图 5-2. 电池测试结果.....	15
图 5-3. 电流消耗测试.....	16

## 表格清单

表 1-1. MSPM0 测量仪表解决方案比较.....	2
表 4-1. 电池测试模式.....	8
表 4-2. MSPM0 L1 测量仪表 SOC-OCV 范围.....	10
表 4-3. MSPM0 L2 测量仪表 SOC-OCV 范围.....	11
表 4-4. 一般配置参数.....	11
表 4-5. VGauge 算法相关参数.....	11

## 商标

所有商标均为其各自所有者的财产。

## 1 引言

存在基于 MSPM0 的各种不同测量仪表解决方案。表 1-1 展示了各解决方案之间的快速比较结果，方便客户选择合适的解决方案。本文档重点介绍 MSPM0 L1 测量仪表解决方案。

**表 1-1. MSPM0 测量仪表解决方案比较**

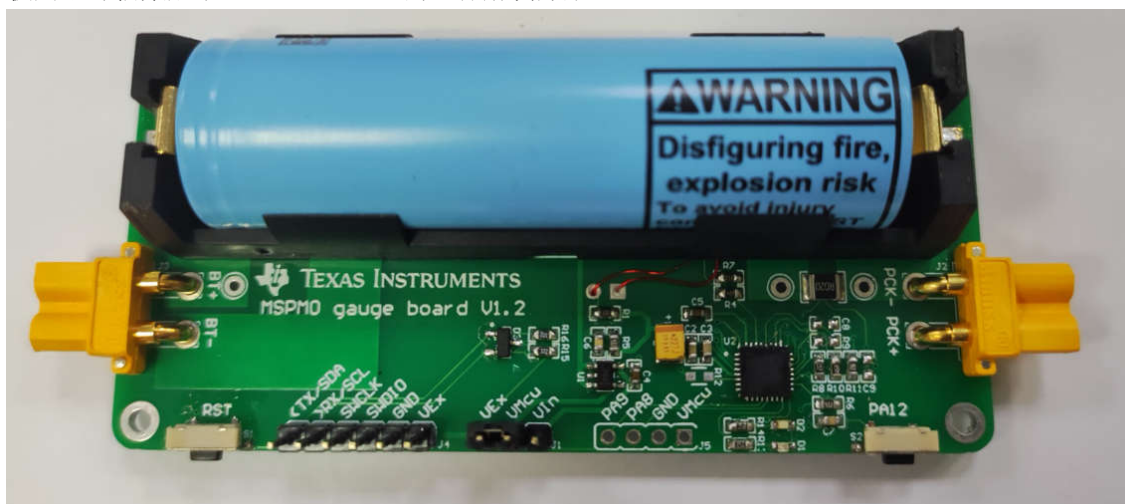
	MSPM0 L1 测量仪表	MSPM0 L2 测量仪表
检测的参数	电压、温度	电压、温度、电流
输出关键参数	SOC	SOC、SOH、剩余容量、周期
使用的方法	电压测量仪表	库仑计数 + 电压测量仪表 + 空载/满载补偿 + 容量学习
适用应用	具有低 SOC 精度的输出阶跃	具有高 SOC 精度的输出百分比

基于 MSPM0 的 1 级测量仪表解决方案的功能如下所示：

- 在 MCU 上电后工作，无需工厂校准或学习循环。
- 支持 SOC ( 充电状态 ) 和警告标志输出。
- 对电池化学参数输入的要求较低。
- 总体解决方案需要大约 6K 的闪存和 1.6K 的 SRAM。
- 无 UART 通信 ( NO\_OUTPUT 模式 ) 的电流消耗约为 3  $\mu$  A。

该解决方案由三部分组成。这几个部分均可在 [MSPM0 L1 测量仪表开发包](#) 中找到。

1. 硬件板用于评估集成了 MSPM0L130x 的整体解决方案。



**图 1-1. MSPM0 测量仪表硬件板**

2. 基于 MSPM0L130x 的软件工程，包括所使用的测量仪表算法。

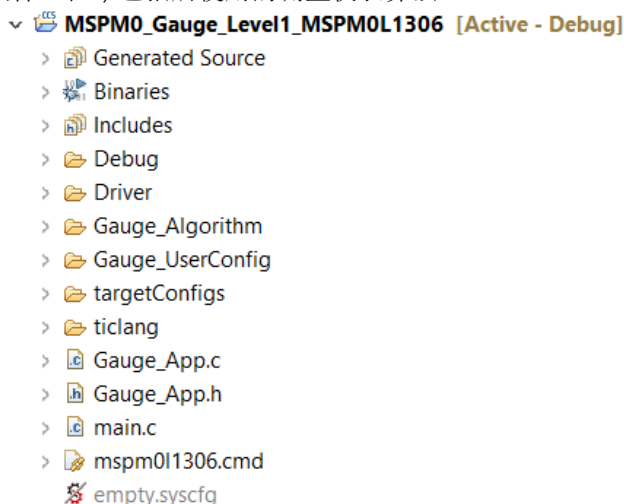


图 1-2. MSPM0 测量仪表软件工程

3. GUI 用 python 编写，可用于与测量仪表板通信，通过控制源表来运行测试模式并生成电池参数。

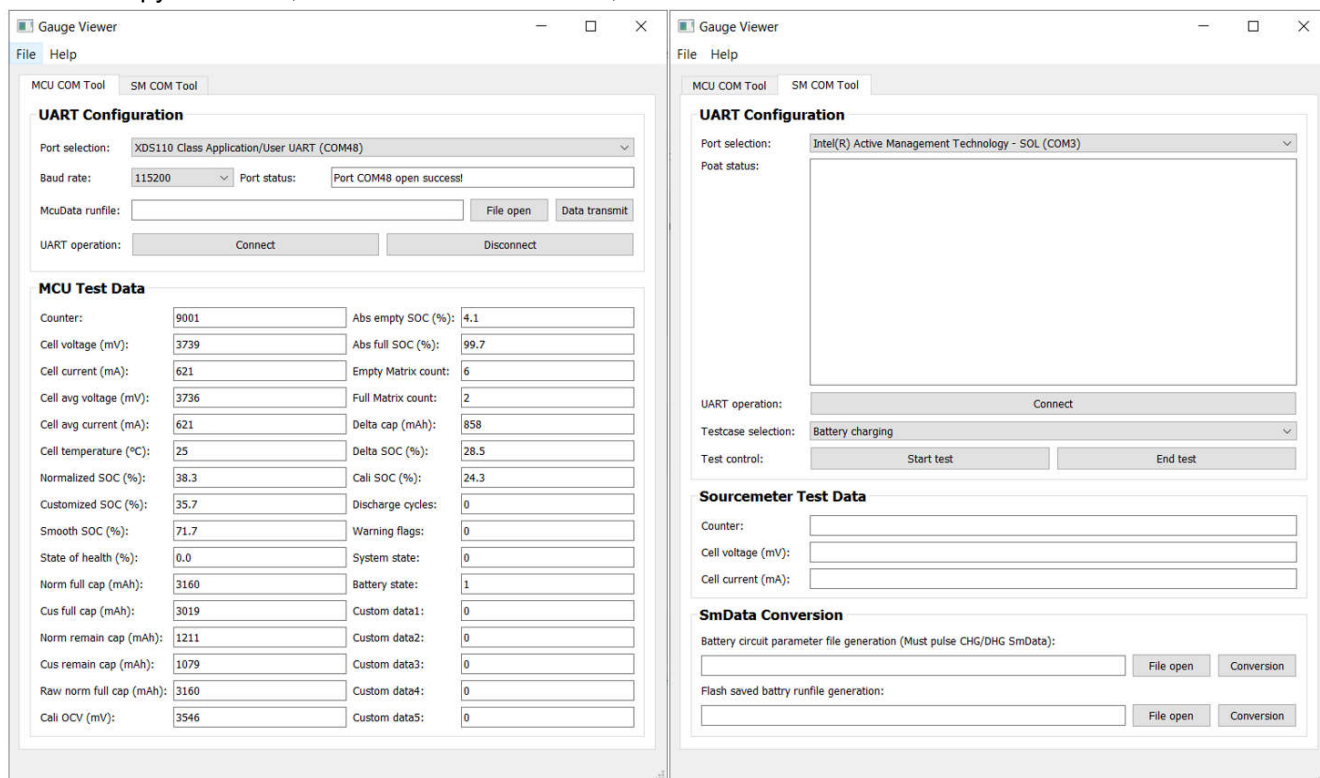


图 1-3. MSPM0 测量仪表 GUI 工程

## 2 测量仪表硬件简介

图 2-1 展示了硬件简要方框图。输入参数仅是从 ADC 通道 1 和 ADC 通道 5 测试的电压和温度。

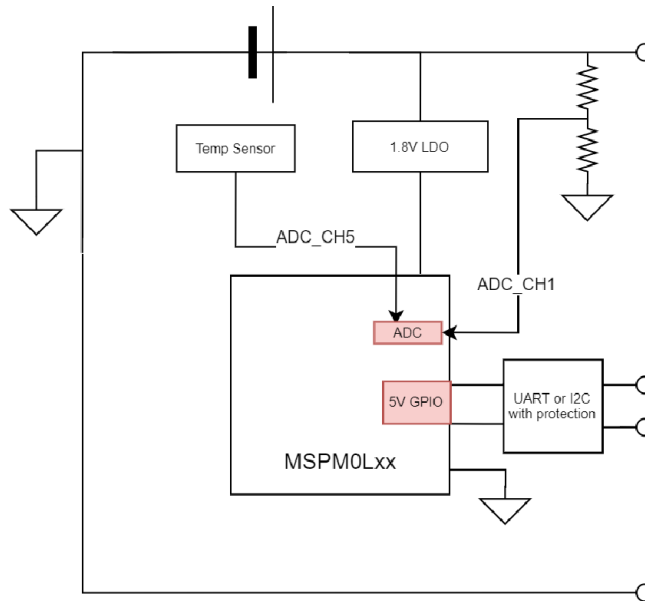


图 2-1. MSPM0 测量仪表板方框图

此设计可以使用以下 MSPM0 特性：

- 用于温度和电压检测的高精度 12 位 ADC
- 具有 UART 或 I2C 功能的 5V 容限开漏 I/O，可在不同电源轨下与主器件通信
- 低至 1.62V 的工作电压，支持单节电池全电压范围
- 低至 1.1  $\mu$ A 待机电流，具有 SRAM 保持功能，适用于电池应用

在这里，我们将简要介绍硬件板及其使用方法。对于电池，您可以将其插入默认插座或连接到备用电池电源输入。调试和 UART COM 端口用于连接到可下载代码或与 GUI 通信的 PC。

注意 MCU 电源开关供电跳线。为了进行下载，请将 VMCU 连接到 VEx，然后为 MCU 提供 3.3V 电压，这样可以确保电压与调试器匹配。为了进行评估，请将 VMCU 连接到 Vin，然后将为 MCU 提供 1.8V LDO。它可以确保出色的模拟性能。

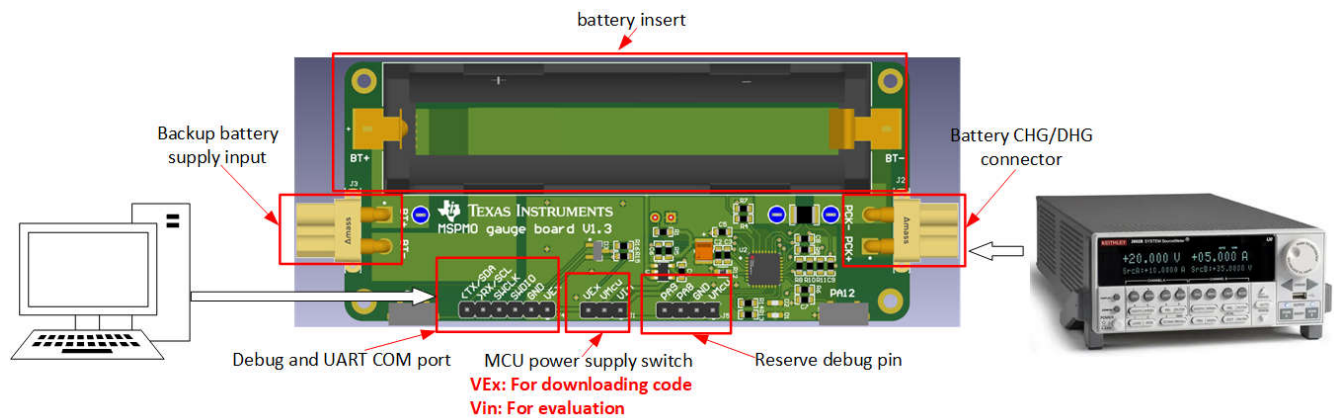


图 2-2. 测量仪表板说明

### 3 测量仪表软件简介

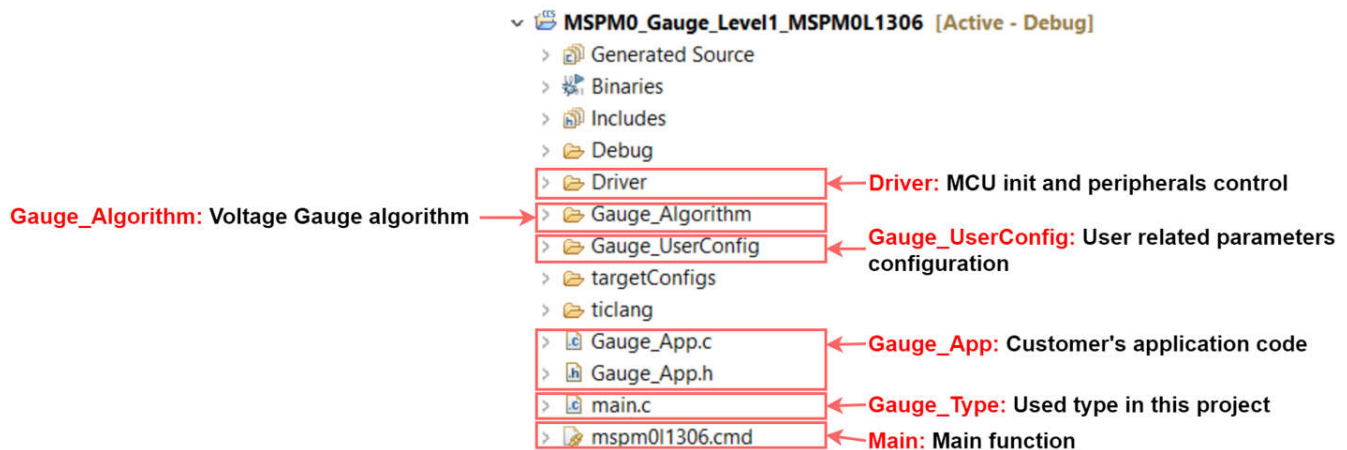


图 3-1. MSPM0 测量仪表软件工程视图

软件工程如上图所示。与测量仪表算法相关的工程和文件包含 6 个部分。所有 MSPM0 工程的其他文件都是相同的。

对于 Gauge\_UserConfig 部分，包含节 4。

Gauge\_Algorithm 部分在本节的最后介绍。

Driver 部分包含所有与 MCU 相关的外设控制。它将 Vcell 和 Tcell 数据准备到 Gauge\_Algorithm 中。

Gauge\_App 部分包含简要测量仪表算法调用。客户可以在此部分定制自己的功能。

Gauge\_Type 部分包含此工程中使用的所有结构。您还可以在其中找到一些详细注释。

Main 部分包含最高的系统功能代码。

#### 3.1 测量仪表算法简介

对于 VGauge，它将电池视为一阶 RC 模型。然后，它将使用 RC 模型和 SoC 开路电压 (OCV) 表生成 VF\_SoC。由于它使用低阶电路模型来模拟电池，因此 VF\_SoC 的精度不是很高。但是，在一开始没有检测到电池电流或不知道电池满容量 (AbsFullCap) 时，如果您了解电池 SoC，会有所帮助。在软件代码中，RC 模型和 SoC-OCV 表保存在“circuitParamsTable”中。

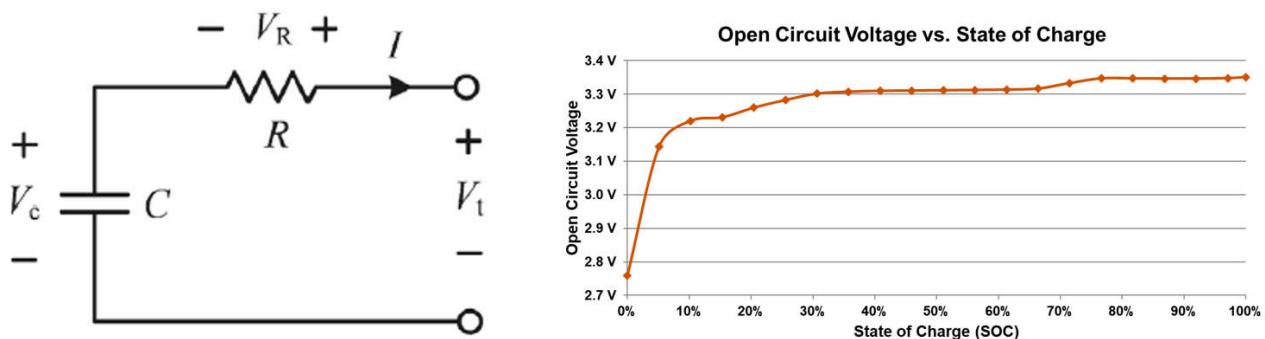


图 3-2. 电池模型和 SoC-OCV 表

图 3-3 展示了 VGauge 功能的软件流程图。circuitParamsTable ( Rcell 查找表和 SOC-OCV 查找表 ) 和 QMax 在每次测量或计算后都会保存。当 MCU 开始工作时，它将第一个 AvgVcell 视为 OCV[K-1]，然后查看 SOC-OCV

表以查找 SoC。计算 Rcell 和 Ccell 并将它们输入模型。使用 AvgVcell 输入，计算出新的 OCV[K]，该 OCV[K] 被视为在下一个计算周期在模型中输入的新 OCV[K-1]。

总而言之，该模型用于根据电池参数和 AvgVcell 输入评估 OCV。可通过搜索 SoC-OCV 表来获取 SoC。

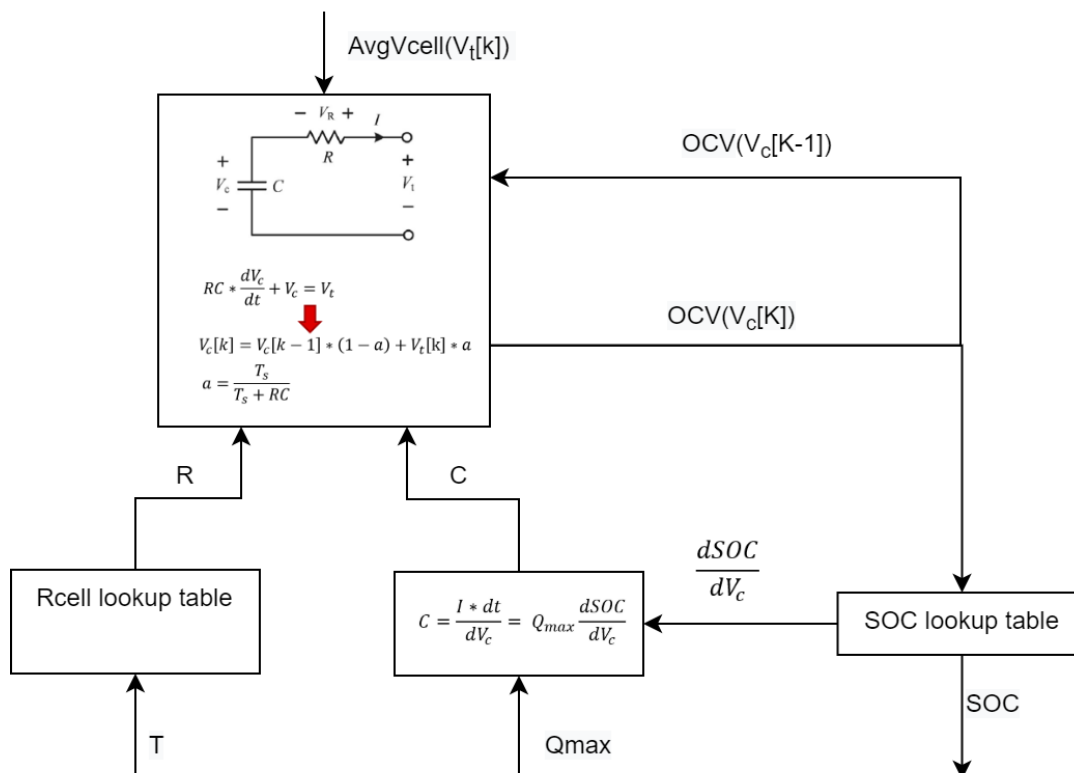


图 3-3. VGauge 软件流程

有关 VGauge™ 的更多详细信息，请参阅[利用滤波端电压的锂离子电池电流无传感器荷电状态估计算法](#)。

### 3.2 测量仪表 GUI 简介

测量仪表 GUI 也是此解决方案的一个重要部分。它可用于记录 MCU 数据、运行电池测试用例和执行数据转换。此 GUI 有两页。第一页是 MCU COM Tool，用于与 MSPM0 进行通信并记录 MCU 传送的电池运行数据。第二页是 SM COM Tool，用于与源表进行通信、运行电池测试用例并记录源表发出的测试数据。此页中也会完成数据转换，以便与不同的测量仪表工作模式配对。

首先，请查看 MCU COM Tool，如图 3-4 所示。它有两个功能。首先是从 MCU 接收电池运行数据。在测试完成或停止测试后，数据将自动保存到名为“time-McuData.xlsx”的 Excel 中。

其次是加载所选的“time-McuData.xlsx”Excel 文件，并将此文件中的电芯电流、电芯电压和电芯温度数据传输到 MCU，以便与相关测量仪表模式（通信数据输入模式）配对运行算法。



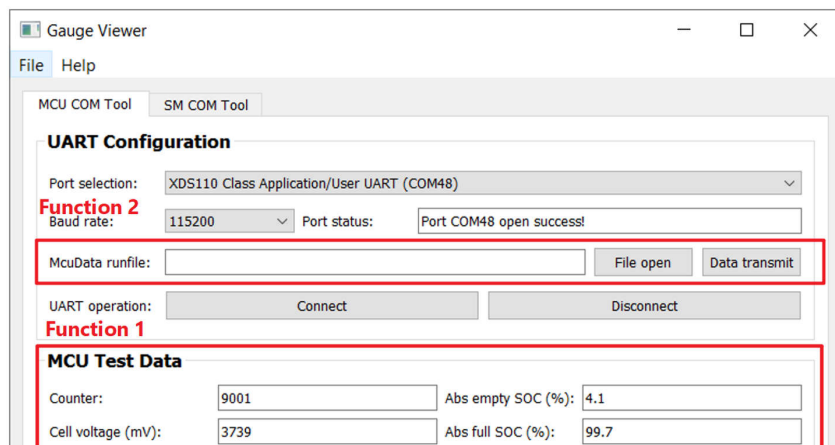


图 3-4. MCU COM Tool 功能

对于 SM COM Tool，它还有两个功能，如图 3-5 所示。功能 1 用于控制源表来运行电池测试用例。然后显示并记录源表测量的数据。记录数据保存在名为“time-SmData.xlsx”的 Excel 中。如果要重新创建此部分，对于软件，至少需要安装 NI\_VISA。对于硬件，需要购买一个 USB 转 RS232 导线和 Keithley 2602A 源表。

功能 2 有助于将记录数据聚合到 C 文件中。对于“Battery circuit parameter file generation”，它用于从脉冲 CHG/DHG 文件中提取电池参数，包括 SOC、OCV 和 Rcell，以生成“circuitParamsTable”。对于“Flash saved battery runfile generation”，它用于将记录文件聚合到 C 文件中。然后，可以将该文件保存到 MCU 中，与相关的测量仪表模式（闪存数据输入模式）配对运行。

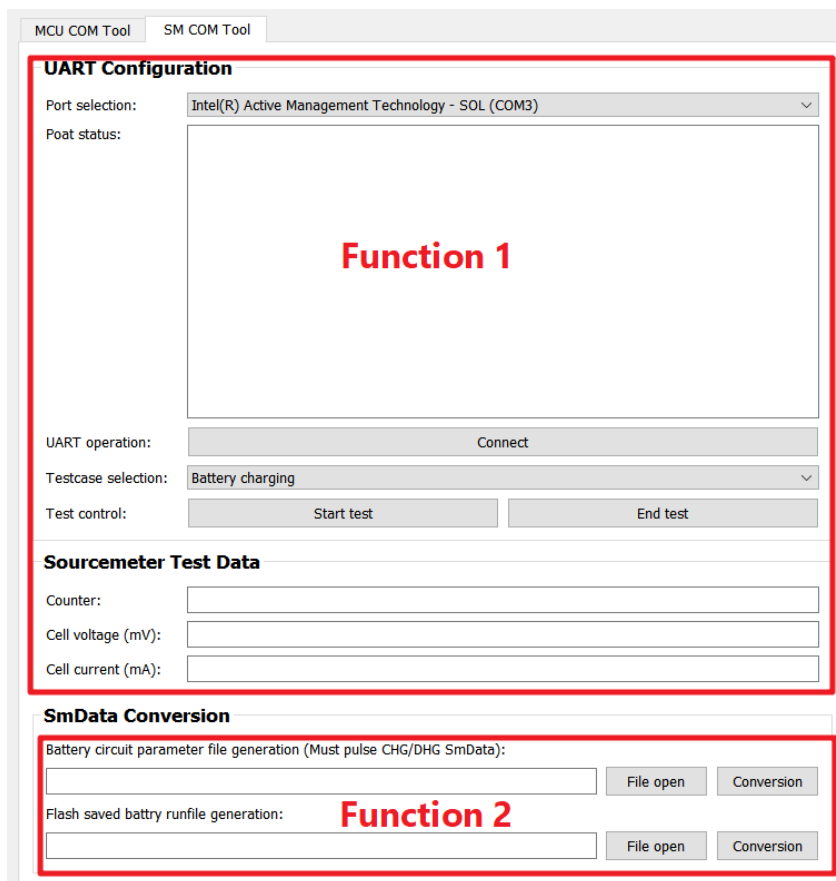


图 3-5. SM COM Tool 功能

提供了 Gauge GUI 执行文件，可以使用它进行评估，而无需安装 python。但是，如果要在 SM COM Tool 下自定义电池测试用例，建议使用源代码。有关如何使用 GUI 的更多详细信息，请参阅下一节。

## 4 MSPM0 测量仪表评估步骤

### 4.1 第 1 步：硬件准备

#### 硬件版：

如果要评估此整体解决方案，需要先设计测量仪表板。如果您只想评估测量仪表软件，只需要一个 MSPM0L1306 LaunchPad 并将准备好的电压和温度数据输入到 MSPM0 测量仪表算法中。

#### 测试设置：

为了测试和评估 MSPM0 测量仪表性能，需要准备源表或其他电池测试机来控制电池充电和放电。如果您有热流来评估测量仪表在不同温度下的性能，也将很有帮助。

### 4.2 步骤 2：获取电池模型

电池模型来自脉冲放电测试用例。为工程获取电池模型始终是个不错的选择。然而，对于实际应用中具有低放电电流的 MSPM0 L1 测量仪表，您实际上并不需要进行测试。您可以重复使用代码中的默认模型，也可以从网络上获取与电池化学成分相关的模型。对于更高级别的 MSPM0 测量仪表解决方案，由于精度取决于电池模型，强烈建议获取专用电池模型。

对于测试机，您可以使用能为电池充电和放电并能记录测试数据的任何机器。与提供的 GUI 配对的测试机是 Keithley 2602A 源表，它通过 USB 转 RS232 导线进行控制，与 NI\_VISA 配对使用。

要获得更精确的模型，您需要以低电流对电池放电，例如以 0.1C 持续 20 分钟。每个脉冲后的休息时间应为 1-2 小时，然后您可以将 VCell 作为 OCV。最后，使用此设置，您将获得大约 30 点。

表 4-1 展示了建议的测试模式。

表 4-1. 电池测试模式

参数	值	注释
启动电压 (OCV)	4.3V~4.4V	确保启动电压不低于应用的最大充电电压
终止电压 (OCV)	2.5V~3.0V	确保休息电压 (OCV) 不高于应用的最小放电电压
放电电流	0.05°C ~ 0.1°C	低电流意味着点更多
放电时间	20 分钟	低放电时间意味着点更多
休息时间	1-2 小时	越长越好

图 4-1 展示了一个电池模型示例测试用例。它将电池充满电 (4350mV)，休息 1 小时后，电压降至 4322mV。然后，它会进行脉冲放电并休息，以获取不同 SoC 下的 OCV。此测试以 2450mV 终止。休息 1 小时后，电压升至 2864mV。因此，SoC-OCV 表的 OCV 范围为 2864mV 至 4322mV。在 OCV 和电池电压之差下，启动电压为 4322mV，终止电压为 2864mV。



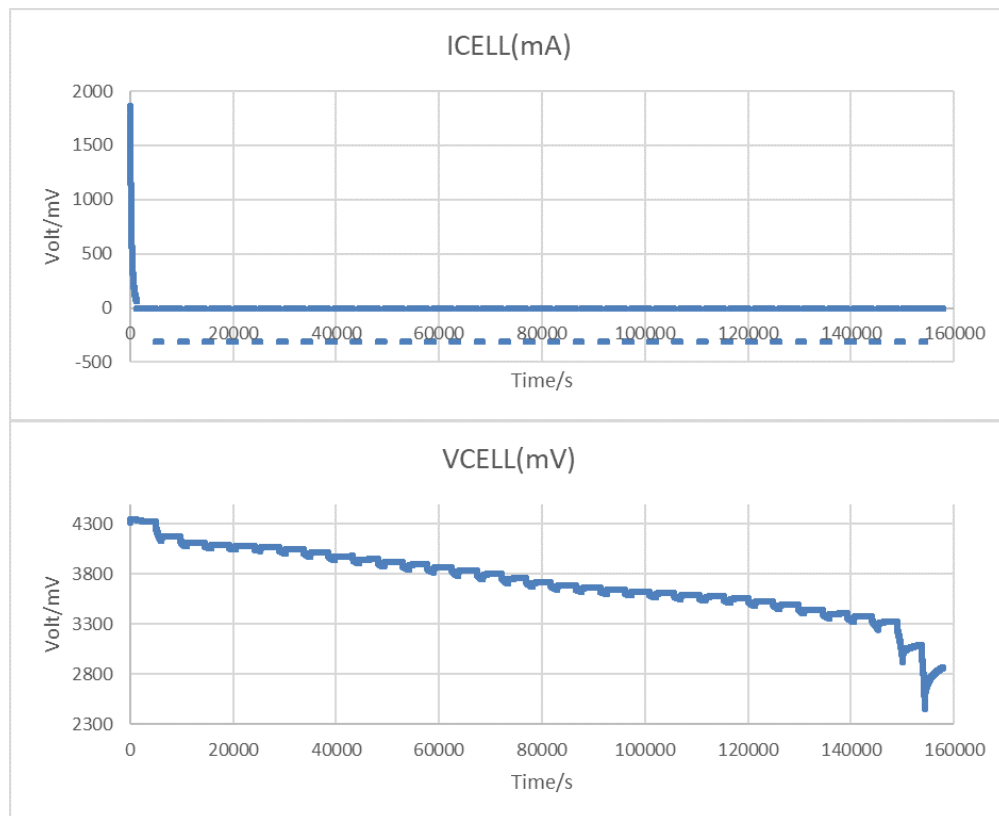


图 4-1. 脉冲放电测试用例

建议的设置如下所示。**MCU COM** 工具用于获取电池运行数据。**SM COM** 工具用于控制源表来生成电池脉冲充电，并收集电压和电流数据，以便稍后生成电池参数。

注意以四线模式连接源表，这样可以减少线路电阻引起的电压检测误差。在实际应用板下测试电池，因为应用板也会影响电池参数，尤其是电池电阻。

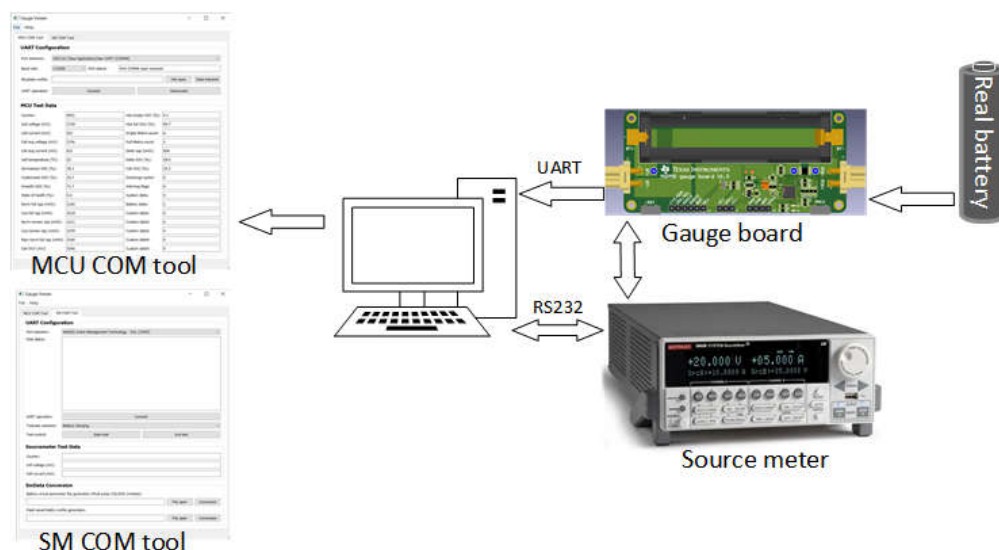


图 4-2. 获取电池模型的硬件结构

请记住根据您的应用更改 python 源代码中的参数，例如放电电流、终止电压等。获得电池运行数据后，您可以使用“Battery circuit parameter file generation”在 Excel 和文本中获取电池电路文件，如图 4-3 所示。输入文件可以是 SMDData 和 MCUDData。

图 4-3. 电池电路表生成

将生成的文本表复制到 Gauge\_UserConfig.c 中，将表长度复制到 Gauge\_UserConfig.h 中。然后您可以完成电池电路表输入。电容系数等于  $dSoc(\%)/dOcv(mV) \cdot Qmax(As)$  或  $dSoc(\%)/dOcv(V) \cdot 3.6 \cdot Qmax(mAh)$ 。对于其他参数生成方法，请查看 python 源代码。

图 4-4. 电池电路表输入

**注意：**对于 MSPM0 L1 测量仪表，它会计算静态 SoC，这不会考虑剩余 SoC 或电池老化。这意味着，您需要通过将 SOC-OCV 表的最小 OCV 设置为高于应用的最小放电电压来添加一些缓冲。表 4-2 展示了不同放电条件下建议的最小 OCV 的示例。

表 4-2. MSPM0 L1 测量仪表 SOC-OCV 范围

	<0.01°C 时的电流	<0.1°C 时的电流	<0.5°C 时的电流
表最大 OCV	4.2V	4.2V	4.2V

表 4-2. MSPM0 L1 测量仪表 SOC-OCV 范围 (continued)

	<0.01°C 时的电流	<0.1°C 时的电流	<0.5°C 时的电流
表最小 OCV	3V	3.2V	3.4V
应用的最大充电电压	4.2V	4.2V	4.2V
应用的最小放电电压	2.8V	2.8V	2.8V

对于高级 MSPM0 测量仪表解决方案，考虑到剩余 SoC 或电池老化，最好使电路表大于应用的电压范围，以保留一些缓冲。表 4-3 展示了不同放电条件的示例。

表 4-3. MSPM0 L2 测量仪表 SOC-OCV 范围

	<0.01°C 时的电流	<0.1°C 时的电流	<0.5°C 时的电流
表最大 OCV	4.3V	4.3V	4.3V
表最小 OCV	2.6V	2.6V	2.6V
应用的最大充电电压	4.2V	4.2V	4.2V
应用的最小放电电压	2.8V	2.8V	2.8V

### 4.3 步骤 3：输入自定义配置

您需要完成“Gauge\_UserConfig.c”中“tBattParamsConfig”结构的配置。为便于评估，您只需更改一般配置参数。

```
Gauge_UserConfig.c
49};
50
51 const tBattParamsConfig battParamsCfg = {
52 //*****General configuration parameters**
53     .pBattCircuitParams = circuitParamsTable,
54     .u16DesignCap_mAh = 3200,
55     .u16MinBattVoltThd_mV = 2500, //Need to ensure the battery
56     .u16MaxBattVoltThd_mV = 4300, //Need to ensure the battery
57     .u16MinFullChgVoltThd_mV = 4100, //We advise to set the value
```

图 4-5. tBattParamsConfig 结构

将这些参数分为两部分。有关所有这些相关参数的简要说明，请参阅表 4-4。

表 4-4. 一般配置参数

参数	备注
u16DesignCap_mAh	只需输入电池的标准容量或通过电池参数生成的测试电池容量即可。
u16MinBattVoltThd_mV u16MaxBattVoltThd_mV i8MaxTempThd_C i8MinTempThd_C	电池 Vcell、Tcell 阈值。它们被保留以在电池情况高于这些参数时控制警告标志。
u8AvgBattParamsUpdateCount	它告知平均数据是在稳定周期之后获得的。
u8SysTikShift sysTikFreq	选择算法运行频率。

表 4-5. VGauge 算法相关参数

参数	备注
u8CircuitTableLength	电路表长度
u8CircuitTableTestTemp_C iq15RcellNegTshift_R iq15RcellPosTshift_R	这些参数用于评估不同温度下的 Rcell。它根据经验进行，不会对性能产生太大影响。

### 4.4 步骤 4：评估

开始之前，图 4-6 展示了 Gauge\_UserConfig.h 中与评估相关的一些设置。

```

Gauge_UserConfig.h ×
7 //*****Algorithm detection mode selection****
8 //#define DETECTION_MODE (FLASH_DATA_INPUT)
9 //#define DETECTION_MODE (COMMUNICATION_DATA_INPUT)
10 #define DETECTION_MODE (DETECTION_DATA_INPUT)
11
12 //*****Algorithm data output mode selection**
13 //#define OUTPUT_MODE (NO_OUTPUT)
14 #define OUTPUT_MODE (UART_OUTPUT)
15

```

图 4-6. 测量仪表模式设置

对于不同的输出模式，UART\_OUTPUT 意味着通过 UART1 实现数据输出。然后，您可以在 GUI 上观察电池运行参数。NO\_OUTPUT 意味着终止 UART 数据输出。这是调试算法的好方法，可以在短时间内运行多个周期。

下一节将详细介绍不同的检测模式。

#### 4.4.1 检测数据输入模式

在此模式下，需要 MSPM0 测量仪表板和真实电池进行测试。检测数据 (Vcell、Icell 和 Tcell) 来自 MSPM0 模拟外设。GUI 可以帮助记录电池运行数据，以便进一步分析。

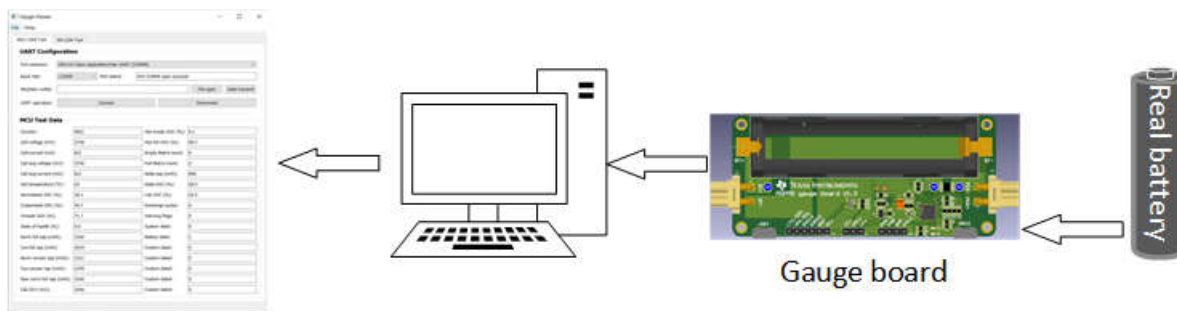


图 4-7. 检测数据输入模式结构

#### 4.4.2 闪存数据输入模式

此模式意味着电池运行数据 (Vcell、Icell、Tcell) 将保存到 MCU 中。此方法无需使用硬件并可增加算法运行频率。由于不需要 UART 通信，因此运行频率最快。

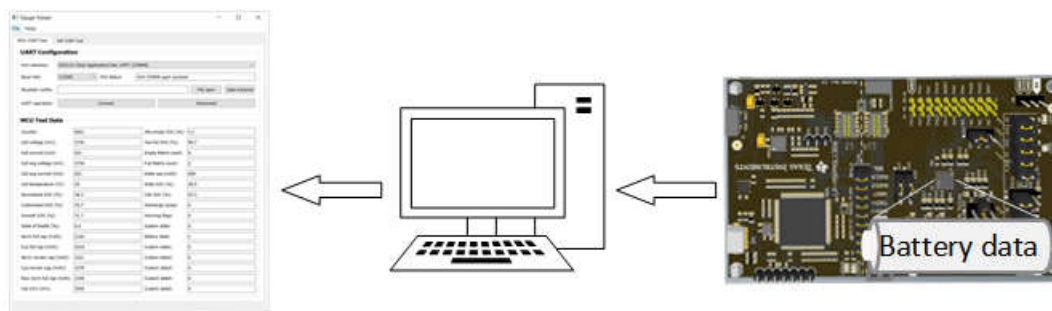


图 4-8. 闪存数据输入模式结构

为了实现这种方法，需要使用“Flash saved battery runfile generation”功能将 SMDData 文件或 McuData 文件转换为 C 代码。

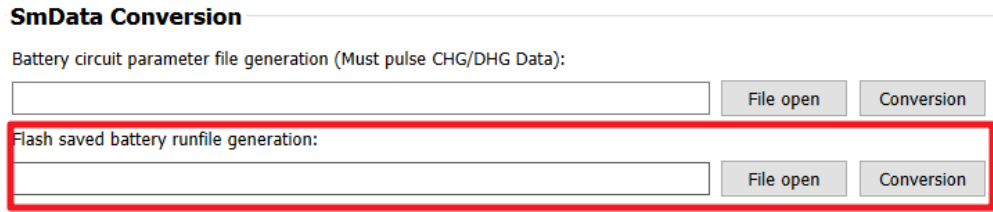


图 4-9. 电池运行文件生成

然后，可以将代码从 txt 复制到 C 文件中。更改检测模式的定义后，可使用单个 LaunchPad 运行算法。

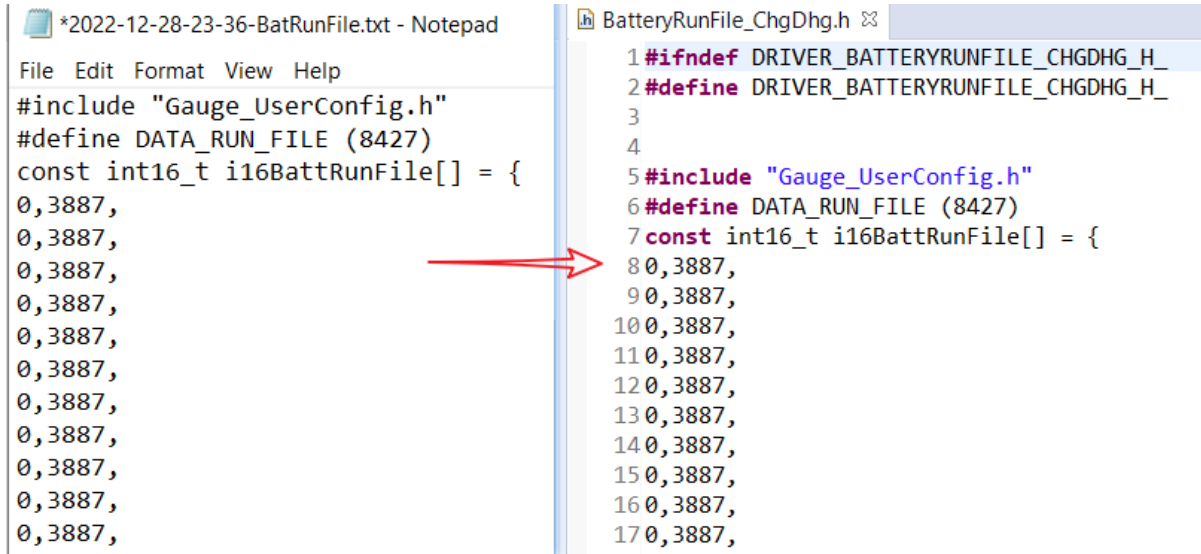


图 4-10. 电池运行文件复制

记住，由于 MCU 存储器大小有限制，输入到 MCU 中的电池运行文件不能没有限制。如果希望运行较长电池周期的测试用例，则需要更改 python 代码和 C 代码中的时间阶跃。可能还需要同时减小 u8AvgVcellDetectPeriod。

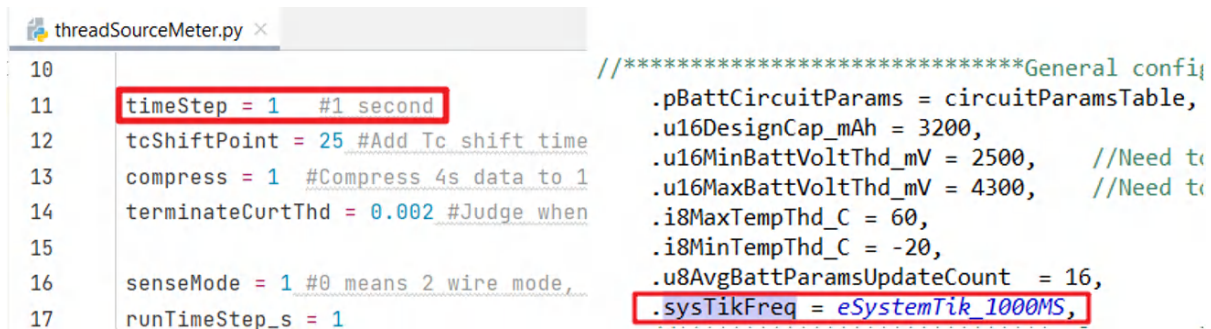


图 4-11. 更改时间阶跃的代码更改



### 4.4.3 通信数据输入模式

对于此模式，电池运行数据通过 GUI 输入。它使您能够仅使用 LaunchPad 运行实际测试用例或评估 MSPM0 测量仪表。该方法无需使用硬件，可增加算法运行频率，并且对电池运行数据的长度没有限制。

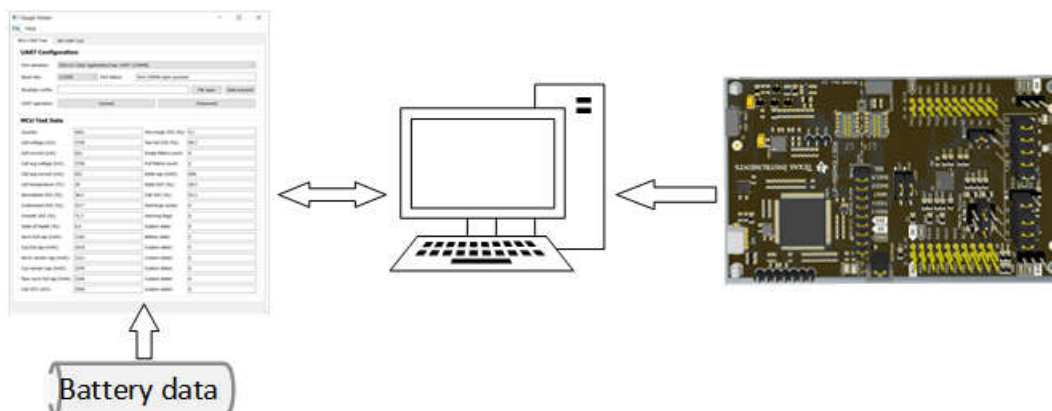


图 4-12. 通信数据输入模式结构

要实现此方法，首先需要连接 UART COM 端口并在 MCU COM 工具中加载 MCUData 运行文件。点击数据传送按钮后，需要等待端口状态变为“Start transmission!”，如图 4-13 所示。如果文件非常大，数据加载时间和 Excel 保存时间将会很长。将为 5~10 分钟。

您将收到来自 MCU 的电池运行数据。此方法的主要好处是，可以反复加载电池运行数据，以改进算法和参数设置。

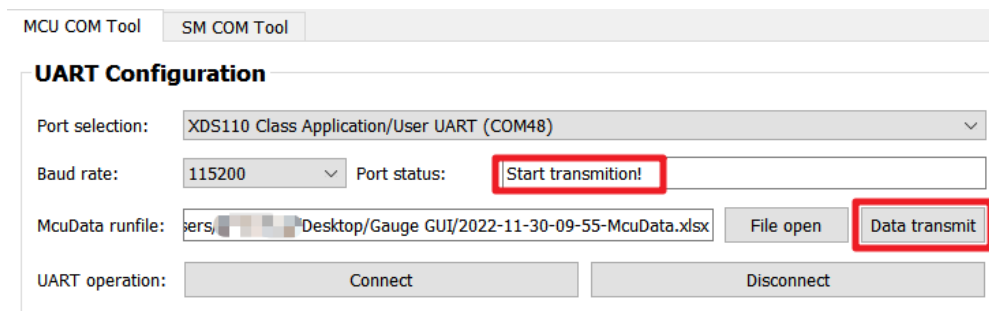


图 4-13. 通信数据输入



## 5 MSPM0 测量仪表解决方案测试结果

### 5.1 性能测试

请参阅 MSPM0 L1 测量仪表的性能。以下测试基于 3100mAh 锂离子电池、在 25°C 下进行。

测试模式如下：

1. 将电池充满电 (4.25V) 并休息 1 小时。
2. 将电池放电至电量耗尽 (2.5V) 并休息 1 小时，温度为 0.5°C/0.3°C/0.1°C。
3. 0.3°C 下的脉冲放电电池。

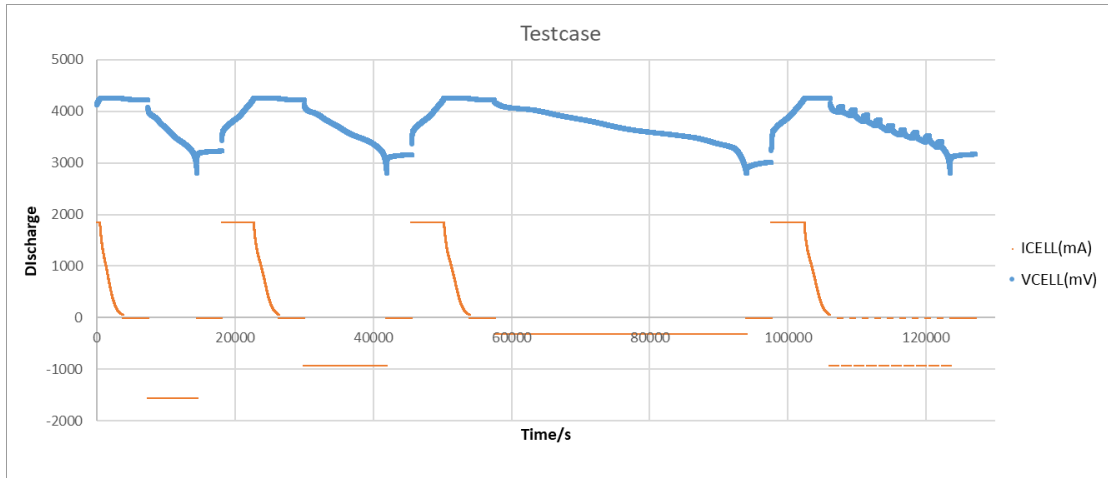


图 5-1. 电池测试用例

电池满容量是根据休息 1 小时后获得的增量 SoC 和累积容量计算的。然后，它将生成测试结果。测试结果中显示了结论。

- 该解决方案适用于恒定低电流放电。您可以发现，在 0.1°C 放电条件下，SoC 误差在 2% 以内。当转向脉冲放电时，SoC 误差会增加。
- 由于电池模型延迟，当电流较高时，SoC 误差会增加。对于 0.5°C 放电，最大误差约为 9%。对于 0.3°C 放电，最大误差约为 4%。对于 0.1°C 放电，最大误差约为 2%。
- 该解决方案非常适合向最终用户输出阶跃而不是百分比。由于它不会测试电流，因此 SoC 在电池停止工作时仍可能发生变化。

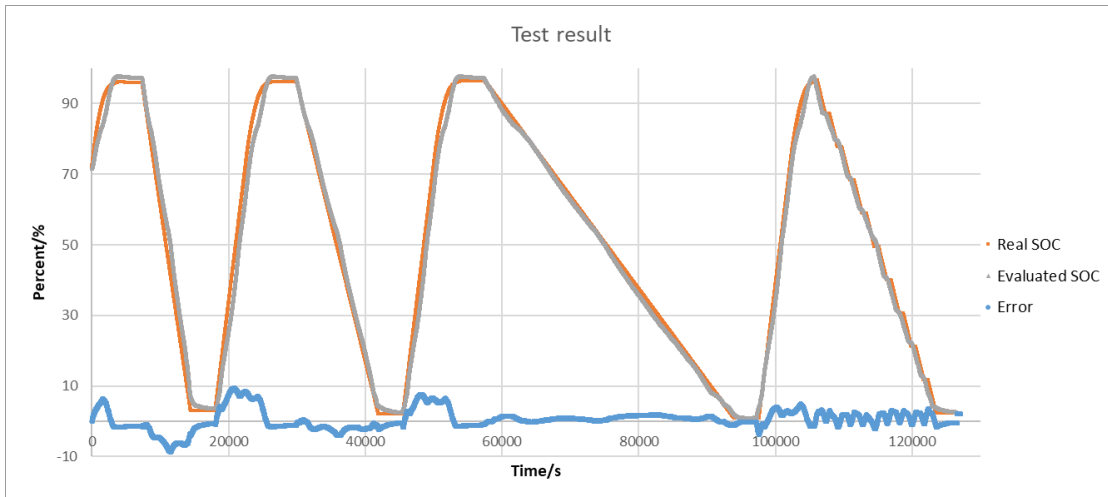


图 5-2. 电池测试结果

此解决方案的优点是：

- 硬件设置简单，仅需检测电压。这意味着平均电流较低
- 软件代码量小
- 与库仑计相比，无累积误差

由于此解决方案主要取决于电池模型的精度，因此温度或老化也是 SoC 计算的两个重要因素，此解决方案不会考虑这些因素。为了在更多条件下提高此解决方案的精度，建议记录受温度或老化影响的 SoC 参数趋势，并在电池模型计算中添加因素。

## 5.2 电流消耗测试

由于 MSPM0 测量仪表板主要用于评估功能，因此基于测量仪表板的电流测试结果有点高。为了进行优化，您需要移除钽电容器，将温度传感器连接到 GPIO 作为 GND，并增大分压器电阻。为了进一步改善电流消耗，可以首先减少 ADC 采样和平均时间。其次，缩短 Vcell 平均时间 (u8AvgBattParamsUpdateCount) 并降低 CPU 唤醒频率 (sysTikFreq)。

电流测试结果如下，平均电流大约为 3μA，采用 NO\_OUTPUT 模式，并移除钽电容器、温度传感器和分压器电阻器。仅用于展示 MSPM0 的功耗。

Name	Live
Time	10 sec
Energy	0.094 mJ
▼ Power	
Mean	0.0093 mW
Min	0.0050 mW
Max	0.0129 mW
▼ Voltage	
Mean	3.3000 V
▼ Current	
Mean	0.0028 mA
Min	0.0015 mA
Max	0.0039 mA

图 5-3. 电流消耗测试

## 重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2023，德州仪器 (TI) 公司