

**摘要**

本应用手册介绍了基于 MSPM0 的 2 级测量仪表实施方案。本文档通过采用数据融合方法，整合从库仑计（电流）和 OCV（电压）获取的 SOC 信息，提供了出色的长期充电状态 (SOC) 精度。该算法还可自动补偿电芯老化、温度和充放电率，同时可在多种工作条件下以百分比形式 (%) 提供准确的充电状态 (SOC)。

本应用手册内容包括实施方案特性、硬件简介、GUI 简介、软件简介和评估。本应用手册中详细介绍的工程配套资料可从 [MSPM0 L2 测量仪表开发包](#) 中下载。

内容

1 简介	3
2 算法介绍	4
2.1 电池基础知识简介	4
2.2 不同的 SOC 和所用技术	5
2.3 算法概述	12
3 测量仪表 GUI 简介	13
3.1 MCU COM 工具	15
3.2 SM COM 工具	15
3.3 数据分析工具	16
4 MSPM0 测量仪表评估步骤	17
4.1 第 1 步：硬件准备	17
4.2 第 2 步：获取电池模型	17
4.3 第 3 步：输入自定义配置	21
4.4 第 4 步：评估	24
4.5 第 5 步：测量仪表性能检查	26
5 MSPM0 测量仪表解决方案	29
5.1 MSPM0L1306 和 1 节 LiCO2 电池	29
5.2 MSPM0G3507、BQ76952 和 4 节 LiFePO4 电池	34
5.3 MSPM0L1306 和 BQ76905	39
6 总结	40
7 参考资料	40
8 修订历史记录	40

插图清单

图 2-1. 电池放电模式	4
图 2-2. 电池充电模式	5
图 2-3. SoC-OCV 表	6
图 2-4. OCV 校准和容量累积	6
图 2-5. 数据融合	7
图 2-6. 算法性能（通过仿真）	7
图 2-7. 电池模型	8
图 2-8. VGauge 软件流程	8
图 2-9. 容量到 SOC	9
图 2-10. CT 表示例	9
图 2-11. 电池内芯温度评估	10
图 2-12. CusRltSoc 至 SmoothRltSoc	11
图 2-13. 算法概述	12

图 3-1. 测量仪表 GUI 功能.....	14
图 3-2. MCU COM 工具功能.....	15
图 3-3. 数据分析工具.....	16
图 4-1. 脉冲放电测试用例.....	18
图 4-2. 用于获取电池模型的硬件结构.....	18
图 4-3. SmData 类型.....	19
图 4-4. 从 SMDData 生成参数文件.....	20
图 4-5. 电池电路表输入.....	20
图 4-6. 脉冲放电示例.....	21
图 4-7. Gauge_UserConfig.h 设置.....	21
图 4-8. 数据结构.....	22
图 4-9. tBattParamsConfig 结构.....	22
图 4-10. 测量仪表模式设置.....	24
图 4-11. 检测数据输入模式.....	24
图 4-12. 通信数据输入模式结构.....	25
图 4-13. McuData 类型.....	25
图 4-14. 通信数据输入步骤.....	26
图 4-15. 电池 3 次充放电.....	26
图 4-16. SmoothSoc 跳变.....	27
图 5-1. 电池 SOC.....	29
图 5-2. MSPM0 测量仪表硬件板.....	29
图 5-3. MSPM0 测量仪表板方框图.....	30
图 5-4. 测量仪表板说明.....	30
图 5-5. MSPM0L1306 LaunchPad UART 连接.....	31
图 5-6. MSPM0 测量仪表软件工程视图.....	31
图 5-7. 电池测试用例.....	32
图 5-8. 电池测试结果.....	33
图 5-9. 读取 Q 值.....	33
图 5-10. 电流消耗.....	34
图 5-11. MSPM0G3507 LaunchPad 和 BQ76952EVM.....	34
图 5-12. MSPM0G3507 和 BQ75952EVM 方框图.....	35
图 5-13. MSPM0 测量仪表软件工程视图.....	36
图 5-14. 电池测试用例.....	37
图 5-15. 电池电芯测试结果.....	37
图 5-16. 电池包测试结果.....	38
图 5-17. 电池测试用例.....	38
图 5-18. 电池测试结果.....	39
图 5-19. 电池测试结果.....	39
图 5-20. 硬件设置.....	40

表格清单

表 1-1. MSPM0 测量仪表实施比较表.....	3
表 2-1. IGauge 关键参数.....	12
表 2-2. 容量学习关键参数.....	13
表 2-3. 混合关键参数.....	13
表 4-1. 电池测试模式.....	17
表 4-2. 系统配置参数.....	22
表 4-3. 一般配置参数.....	22
表 4-4. 混合算法相关参数.....	23
表 4-5. 容量学习算法相关参数.....	23
表 4-6. VGauge 算法相关参数.....	24
表 4-7. IGauge 算法相关参数.....	24
表 4-8. 获得 SmoothSoc 精确值.....	27
表 5-1. 硬件连接.....	40

商标

LaunchPad™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

1 简介

市场上有基于 MSPM0 的不同测量仪表实施方案。表 1-1 所示为客户选型比较表。

表 1-1. MSPM0 测量仪表实施比较表

	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	库仑计数 + 电压测量 + 混合 + 容量学习
关键技术	电池模型	电池模型 + 数据融合 + 空载或满载补偿 + 内芯温度评估
应用	具有低 SOC 精度的输出阶跃数据	具有高 SOC 精度的输出百分比数据
电池类型	LiCO2、LiMn2O4	LiCO2、LiMn2O4、LiFePO4

MSPM0 L2 测量仪表是一款纯软件算法，以软件库形式提供，在 MCU 平台、AFE 或电池选型方面具有高灵活性。主要特性如下：

- 支持采用有限参数输入的 SOC、SOH、容量和警告标志输出
- 在 MCU 上电后工作，无需工厂校准或学习循环
- 支持残留 SOC 和满载 SOC 学习，可针对电芯老化、温度和充放电率进行补偿
- 利用数据融合方法减少 SOC 和 FullCap 测量误差
- 评估电池内芯温度以便处理低温放电
- 支持数据保存，便于重新加载

该 L2 测量仪表算法的 MCU 资源要求总结如下。

	单节电池	多节电池
闪存	优化级别 0：大约 14.8k 优化级别 0：大约 13.6k	优化级别 0：大约 14.8k 优化级别 0：大约 13.6k
RAM	1.54k	2 节电池：2.04k 3 节电池：2.54k N 节电池：1.04k + 0.5*Nk
基于 M0L 的算法运行时间	3ms	N*3ms
MSPM0 平台	MSPM0L、MSPM0C、MSPM0H、MSPM0G	
示例	MSPM0L1306 测量仪表板 + 1 节 LiCO2 电池 (节 5.1) MSPM0G3507 Launchpad + BQ76952 EVM + 4 节 LiFePO4 电池 (节 5.2) MSPM0L1306 测量仪表板 + BQ76905 EVM (节 5.3)	

该设计由三部分组成：硬件、软件和 GUI。这几个部分均可在 [MSPM0 L2 测量仪表开发包](#) 中找到。

- 硬件板用于检测电压、电流和温度，然后将其输入到算法中来计算 SOC。如需了解不同的硬件设置详情，请参阅 [节 5](#)。
- 该软件工程包括使用的测量仪表算法、MCU 控制和 AFE 通信。有关算法的说明，请参阅 [节 2](#)。有关典型的用例，请参阅 [节 5](#)。
- GUI 用 python 编写，可用于与测量仪表板通信、运行测试模式和执行数据分析。有关 GUI 介绍，请参阅 [节 3](#)。

2 算法介绍

下一节简要介绍了电池知识、所用算法以及如何获得所需 SOC，然后提供了软件说明和详细的算法说明。

2.1 电池基础知识简介

测量仪表算法主要用于告诉用户电池的工作状况，并在输出最大容量和延长电池寿命之间达到平衡。此处展示了这两种条件下的基本控制策略和电池性能。

图 2-1 展示了单节 LiCO₂ 电池的电池放电模式以及要介绍的相关概念。红线表示开路电压 (OCV)，即在没有电流流动且电极电势处于平衡状态时正电极 (PE) 和负电极 (NE) 之间的电势差。在将电池静置 1-2 个小时后，通常可以将 OCV 视为等于电池电压。蓝线表示运行时电芯电压 (V_{cell})。由于电池具有内部电阻，因此在恒定负载下 OCV 和 V_{cell} 之间会出现压降。

对于 LiCO₂ 电池，由于化学限制，需要设置完全充电电压阈值（例如 4.2V）和放电终止电压阈值（例如 3V），以避免对电池造成不可逆转的损坏。这意味着，对于不同的放电电流，用户可以从电池中获得不同的容量。由于当温度升高时 R_{cell} 会降低，输出容量也受到温度的影响。在此测量仪表设计中，不可更改的容量称为 NomFullCap，它表示电极结构中的可移动锂离子，其数量不会随电芯使用温度或 C 率的变化而变化。可变容量称为 CusFullCap，这表示最终用户的可用容量，并且受电池运行条件和阈值设置的影响。

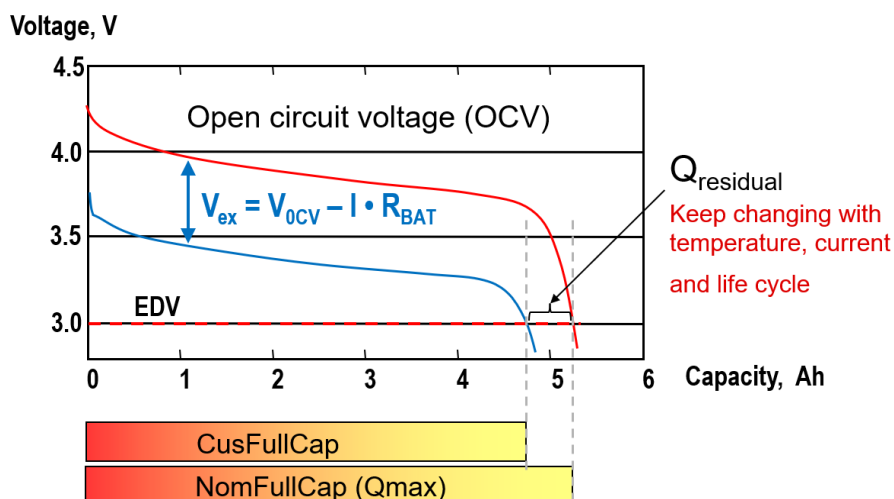


图 2-1. 电池放电模式

图 2-2 展示了正常单节电池的电池充电模式。对于充电情况，可以简化为恒流 (CC) 窗口和恒压 (CV) 窗口。在充电结束时，如果充电电压是恒定的，则电流将以指数级减少。如果电流变为 0，则获得 **NomFullCap**。为避免长时间等待，方案设置了终止充电电流，例如 1/20 容量 (1/20C)，与 **NomFullCap** 相比，这会使 **CusFullCap** 略有减少。

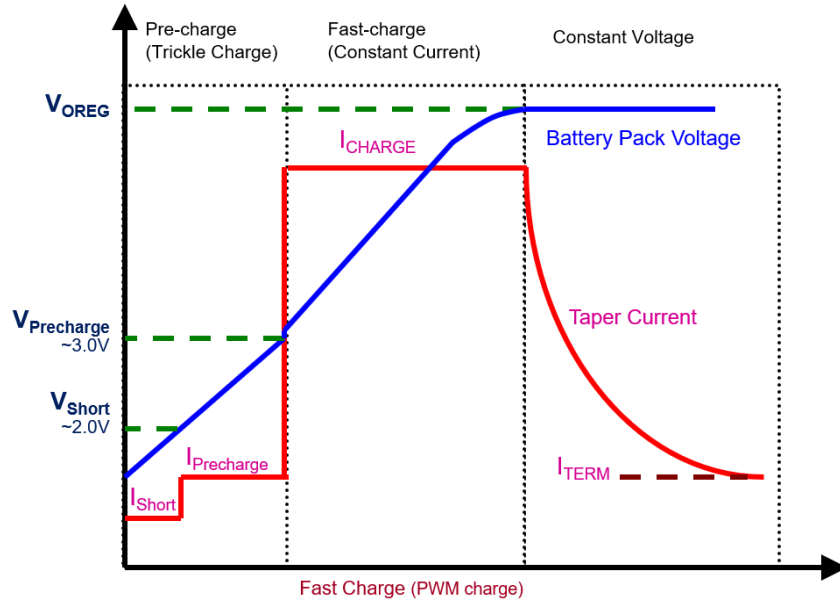


图 2-2. 电池充电模式

NomFullCap 是从一个 OCV 到另一个 OCV 之间测得的，而 **CusFullCap** 是从一个 **Vcell** 到另一个 **Vcell** 之间测得的。因此 **NomFullCap** 始终大于 **CusFullCap**。

在这个测量仪表算法中，**NomFullCap** 范围基于 SOC-OCV 表 (circuitParamsTable) 中每个保存的 OCV 范围。**CusFullCap** 范围基于 **MaxFullChgVoltThd** 和 **EmptyDhgVoltThd** 设置，并且在自学后也会发生变化。为了留出一些裕度，TI 建议将 **NomFullCap** 的 OCV 范围设置为略大于 **CusFullCap** 的电压范围。

2.2 不同的 SOC 和所用技术

上一节介绍了有关容量和 OCV 的基本理念。以下各节介绍最终用户最终需要的充电状态 (SOC)。还可以使用公式来帮助了解 SOC 的概念差异。有关 SOC 特性的更多信息，请参阅节 4.5。

2.2.1 NomAbsSoc 计算

如果用户使用 **NomFullCap** 作为可用容量相对于满电量电池容量的归一化因子，则将获得归一化的绝对充电状态 (**NomAbsSoc**)。这表示可移动锂离子在负电极固体颗粒中的剩余百分比。在此算法中，使用两种策略来计算 **NomAbsSoc**，下一节将对此进行详细介绍。第一种是具有 OCV 校准功能的库仑计。第二种是电池模型滤波器。使用数据融合方法可进一步减少 **NomAbsSoc** 评估误差。

2.2.1.1 具有 OCV 校准功能的库仑计

更新 **NomAbsSoc** 的常用方法是使用库仑计，如方程式 1 和方程式 2 所示。

$$Q_{use} = I(t) \cdot \Delta t \quad (1)$$

$$NomAbsSoc = \frac{NomFullCap - Q_{use}}{NomFullCap} \quad (2)$$

因为库仑计存在误差累积问题。**NomAbsSoc** 完全使用 OCV 进行校准，OCV 是在电池静置足够长时间后确定。图 2-3 展示了 OCV 到 SOC 搜索表示例。

$$NomAbsSoc = f(OCV) \quad (3)$$

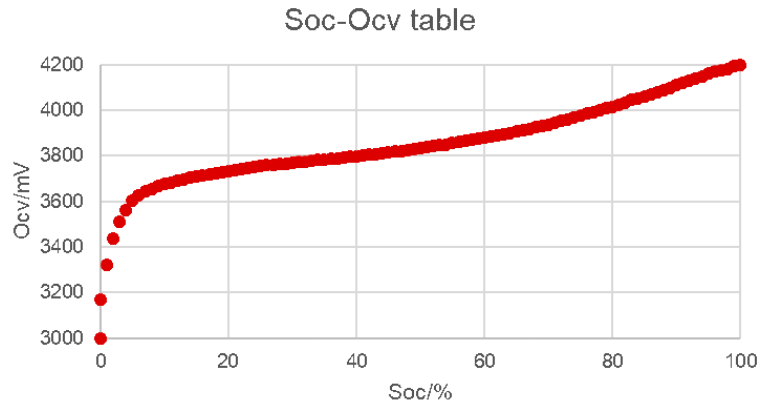


图 2-3. SoC-OCV 表

方程式 2 用于运行时输出 $NomAbsSoc$ ，图 2-1 用于定期校准 $NomAbsSoc$ 。经过两次校准之后，用户就可以得到差值容量和差值 $NomAbsSoc$ 。然后即可计算 $NomFullCap$ ，如方程式 4 所示。

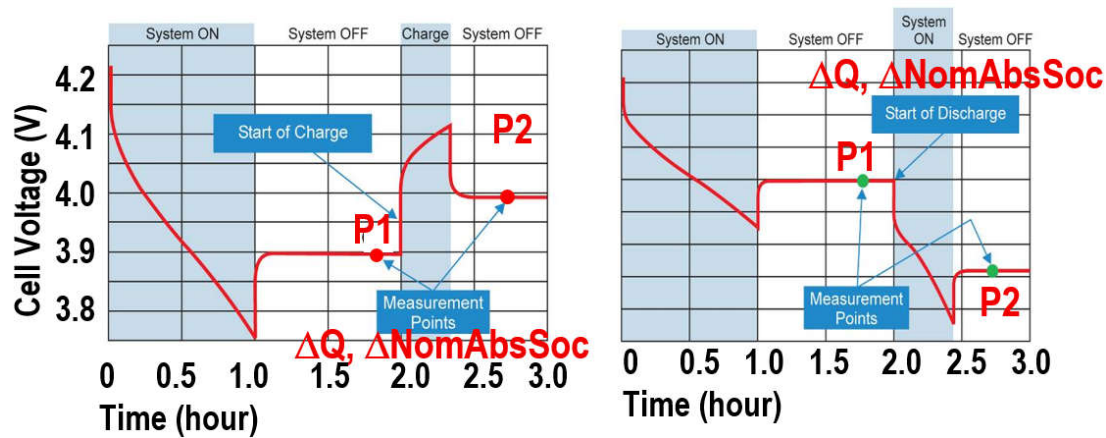


图 2-4. OCV 校准和容量累积

$$NomFullCap = \frac{\sum ABS(\Delta Q)}{\sum ABS(\Delta NomAbsSoc)} \quad (4)$$

对于真实的电池，由于电池老化， $NomFullCap$ 会略有减少。若要跟踪容量衰减问题，需定期校准 $NomFullCap$ 。方程式 5 使用名为运行状况 (SOH) 的指标来表示电池的容量衰减。然而在实际应用中，由于获得的 $NomFullCap$ 有误差，因此采用获得最大的 $NomFullCap$ 作为最大 $NomFullCap$ 。

$$SOH = \frac{NomFullCap[n]}{\text{Max}(NomFullCap[n])} \quad (5)$$

2.2.1.2 数据融合

如 [方程式 4](#) 所示，在实际应用中，校准后的 **NomAbsSoc** 会受到电压检测精度、电池静置时间和 **SOC-OCV** 表精度的影响，这对 **LiFePO4** 电池尤其重要。**DeltaQ** 受分流电阻器精度和 **ADC** 性能的影响。这意味着校准后的 **NomAbsSoc** 和 **NomFullCap** 均存在一些误差。2 级测量仪表使用数据融合方案，这与卡尔曼滤波器的设计理念相同。TI 对被测电流、测量电压、经评估的 **OCV** 等所有误差源进行了加权处理。此后，用户即可获得通过库仑计生成的 **SOC** 权重以及通过 **OCV** 校准生成的 **SOC** 权重。最后，用户将这两个 **SOC** 结合起来即可获得更准确的 **NomAbsSoc**，如 [图 2-5](#) 所示。

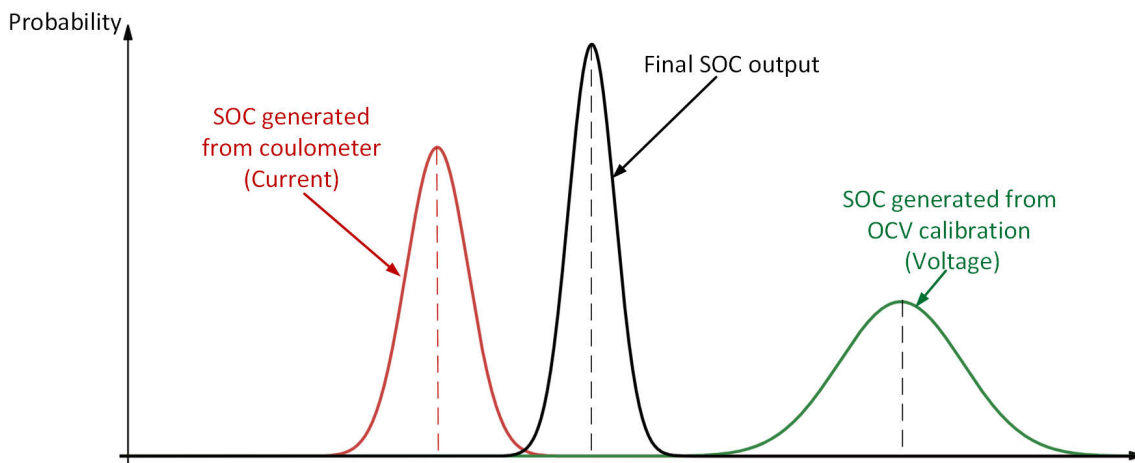


图 2-5. 数据融合

[图 2-6](#) 展示了基于 **LiFePO4** 电池仿真模型的数据融合算法性能，具有 3000 个随机 **DHG** 和 **CHG** 点，并考虑了检测误差。整个电池寿命期间的 **AbsNomSoc** 误差控制在 3.5% 以内，**NomFullCap** 误差控制在 4% 以内。请注意，结果只用于展示在有限条件下检测 **NomAbsSoc** 的算法能力，无法在实际应用中检测 **CusRltSoc** 时保证算法的误差范围。

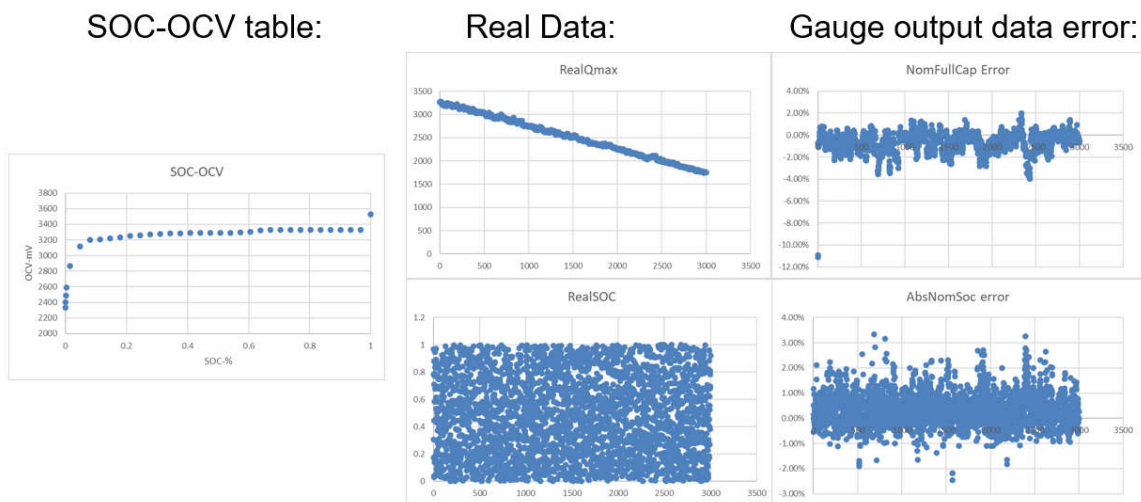


图 2-6. 算法性能 (通过仿真)

2.2.1.3 电池模型滤波器

在节 2.2.1.1 中，仅基于一个电池参数（电压或电流累加）来评估 **NomAbsSoc**。一种方法是利用 **OCV**（正电极和负电极平衡）与 **NomAbsSoc**（不同锂离子浓度）之间的关系。另一种方法是利用库仑计（电子数）和容量（可移动锂离子数）之间的关系。这种策略需要的计算资源更少，因为每个周期只需要使用库仑计即可工作。但需要等到生成 **NomFullCap** 后进行 1-2 个电池周期。

另外一种策略用来创建一个模型或滤波器，根据所有输入参数（如电流、电压、时间和温度）来评估 **NomAbsSoc**，甚至还可评估 **CusRltSoc**。常用的方法有等效电路模型、卡尔曼滤波器、神经网络等。**SOC** 输出的精度主要取决于模型精度。但是，模型越复杂，意味着需要更多 **MCU** 计算资源。

一种经济方法是使用最简单的等效电路模型（一阶 **RC** 模型，如图 2-7 所示）来输出 **NomAbsSoc**（仅带电压输入）。

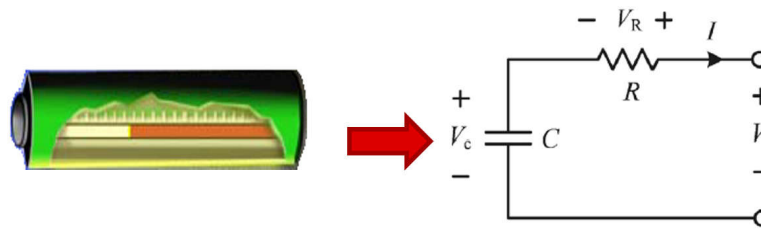


图 2-7. 电池模型

图 2-8 展示了 **VGauge** 的软件流程图。但在此设计中，**SOC** 查询表功能只需用于搜索此电池模型中的所需参数。**SOC** 部分由 **IGauge** 提供。

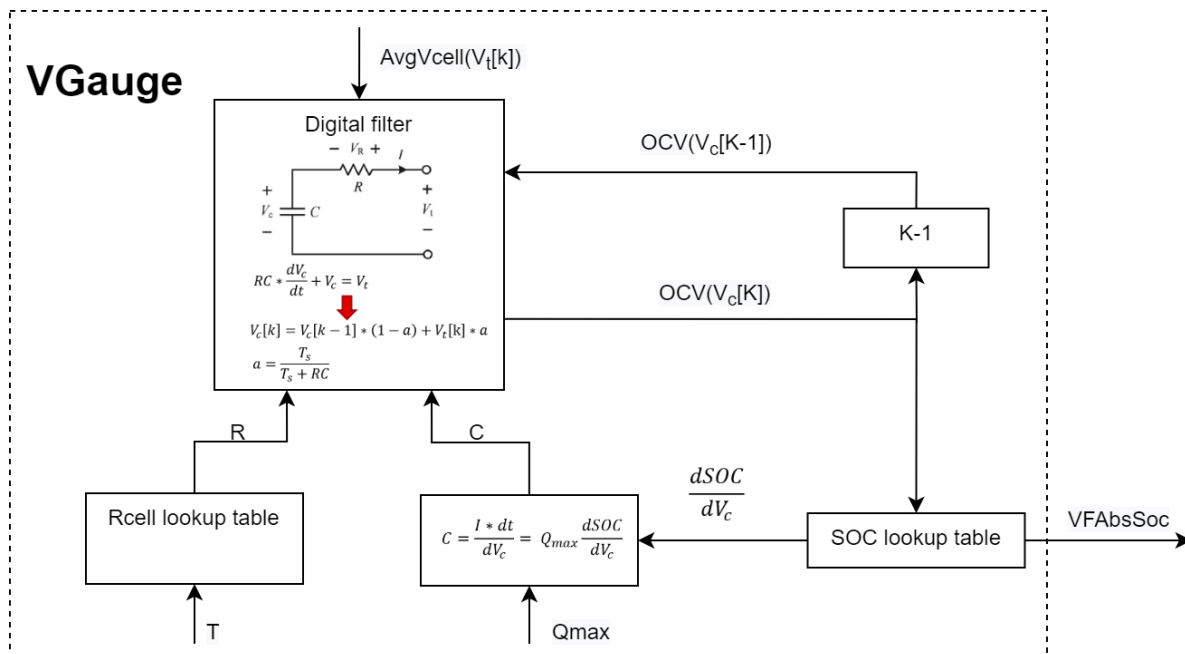


图 2-8. **VGauge** 软件流程

有关 **VGauge** 输出的 **NomAbsSoc** 精度的更多信息，请参阅 [MSPM0 L1 测量仪表解决方案指南](#)，[应用手册](#)。

2.2.2 CusRltSoc 计算

对于实际应用，客户所需的 SOC 不是 NomAbsSoc，因为当达到电池电量已满和电池电量耗尽状态时，无法将电流控制在 0，请参阅图 2-2。

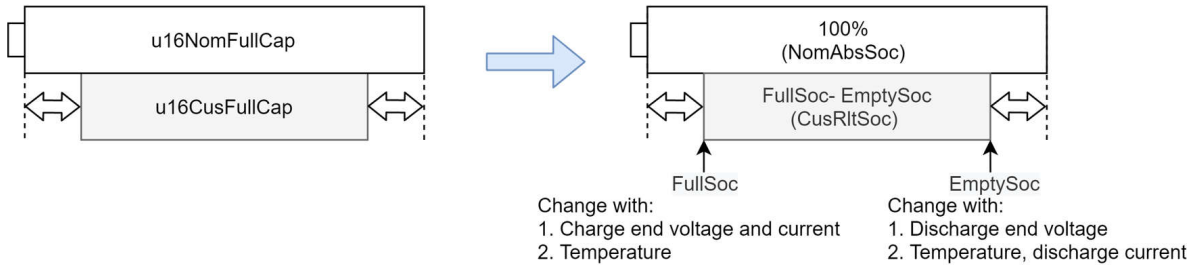


图 2-9. 容量到 SOC

基于归一化因子：NomFullCap，获得三个新 SOC 来表示不可更改的容量，请参阅图 2-9。FullSoc 用于表示未充电的 NomAbsSoc 阈值，而 EmptySoc 用于表示未放电的 NomAbsSoc 阈值。CusRltSoc 表示相对 SOC，通过基于电池状况对 FullSoc 和 EmptySoc 二者进行归一化得出。方程式 6 展示了用于计算 CusRltSoc 的公式。

$$\text{CusRltSoc} = \frac{\text{NomAbsSoc} - \text{EmptySoc}}{\text{FullSoc} - \text{EmptySoc}} \quad (6)$$

2.2.2.1 EmptySoc 和 FullSoc

要获得 CusRltSoc，需要先获取 FullSoc 和 EmptySoc。如图 2-1 中所示，FullSoc 和 EmptySoc (残留 SoC) 受电池电阻和电流的影响。这意味着在不同的条件下，FullSoc 和 EmptySoc 各不相同。

该测量仪表算法更新和补偿了在不同电池条件下的这两个 SOC，包括电芯老化、温度和充放电率。如图 2-10 所示为电流温度表 AbsEmptySocMatrix，以便模拟电流和温度对 EmptySoc 的影响。我们制作了 AbsFullSocMatrix 温度表来模拟温度对 FullSoc 的影响。

在 AbsEmptySocMatrix 中，当电池在 CT 表块范围内工作时，使用一个 EmptySoc 值来覆盖所有真实 EmptySoc。例如，如果 TempThd[1] < Tcell < TempThd[0]，且 CurtThd[0] < Icell < CurtThd[1]，则 EmptySoc[4] 用于表示此条件下的所有 EmptySoc。基于此设置，在一个块中，左下角的真实 EmptySoc 是最小值，右上角的真实 EmptySoc 是最大值。用户需要根据应用或测试结果调整 battParamsCfg 中的 TempThd[] 和 CurtThd[]。

Current-Temperature table example (3x3)

MinTemp			
	EmptySoc[6]	EmptySoc[7]	EmptySoc[8]
TempThd[1]	EmptySoc[3]	EmptySoc[4]	EmptySoc[5]
TempThd[0]	EmptySoc[0]	EmptySoc[1]	EmptySoc[2]
	MaxTemp/0	CurtThd[0]	CurtThd[1]
			MinCurt

图 2-10. CT 表示例

对于 FullSoc，当电池充满电时，校准后的 NomAbsSoc 会识别为新的 FullSoc，更新到不同温度下的电流温度表 AbsFullSocMatrix，与 EmptySoc 相同。

如果学习周期中需要较低的 **SmoothRltSoc** 误差，则用户可使用 **EmptyOcvMatrix[]** 和 **FullOcvMatrix[]**，以便设置 **AbsEmptySocMatrix** 和 **AbsFullSocMatrix** 的起始值。否则，**AbsEmptySocMatrix** 使用在 OCV 等于 **EmptyDhgVoltThd** 时的 SOC 作为默认值。**AbsFullSocMatrix** 使用在 OCV 等于 **MaxFullChgVoltThd** 时的 SOC 作为默认值。

备注

为了快速响应完全充电和完全放电阈值，测量仪表算法使用原始电流、电压和温度来生成 **EmptySoc** 和 **FullSoc** 输出结果。如果原始数据存在较大噪声，则用户需要考虑在电池信息输入端添加额外滤波器，如 IIR 滤波器。

2.2.2.2 内芯温度评估

在常见条件下，测得的温度为电池表面温度。然而内芯温度与电池电阻变化的联系更为密切。电池内芯温度和电池表面温度之间的温差会导致 **FullSoc** 和 **EmptySoc** 评估出现误差，尤其是在低温放电情况下。

为了获得更精确的 **EmptySoc** 和 **FullSoc**，采用了热模型来评估电池内芯温度。用户需要输入 **battModelUse** 参数和 **battWeight** 参数。支持的电池型号为 18650、21700、26650、软包和方形电池。图 2-11 所示为结果示例。

使用模块时，建议的 **iq15Tcell_C** 温度输入分辨率为 0.1°C 。在高放电电流下，采用 1°C 分辨率时，**iq15EvIcoreTemp_C** 误差约为 $\pm 1.5^{\circ}\text{C}$ 。未使用模块时，表面温度用作电池内芯温度。

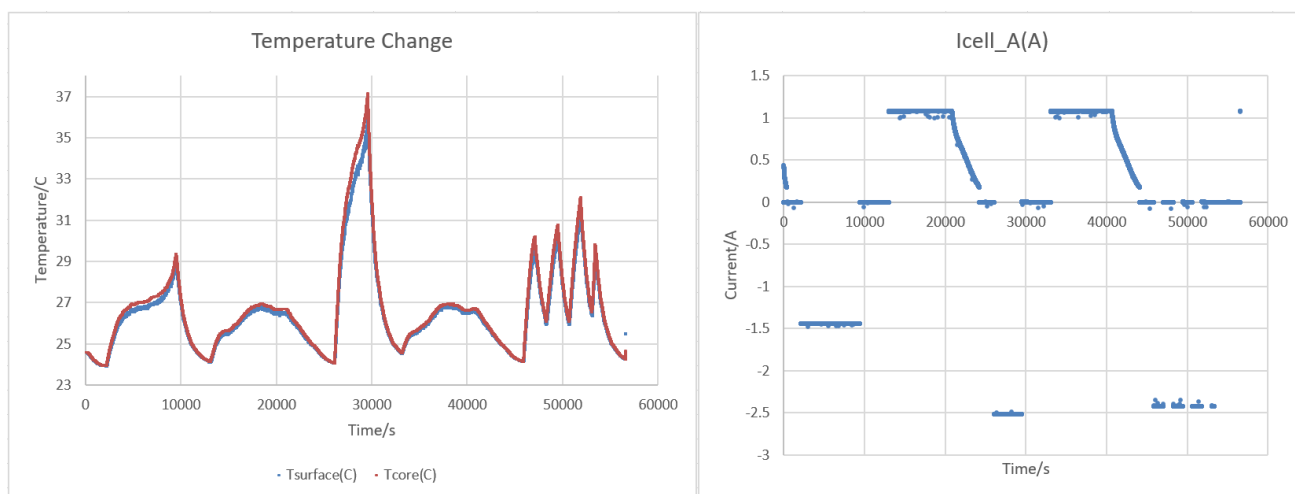


图 2-11. 电池内芯温度评估

2.2.3 SmoothRltSoc 计算

图 2-12 中的蓝色线显示了电池充电和放电测试模式下的 **CusRltSoc** 反应。可以看到由 OCV 校准或 **EmptySoc** 变化导致的数据跳变。

为了使输出 SOC 更容易被客户接受，创建 **SmoothRltSoc** 以使 SOC 输出保持恒定，没有突然的跳变。有关实现方法，请参阅节 2.3.4。

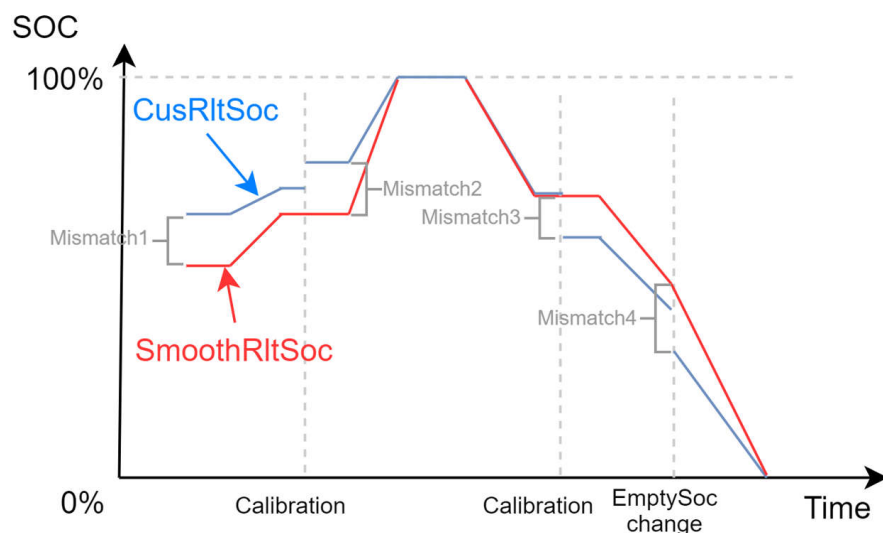


图 2-12. CusRltSoc 至 SmoothRltSoc

SmoothRltSoc 是通过跟踪 CusRltSoc 的变化来获得的，目标是 0% 或 100%，并且没有突然的变化。但如果 CusRltSoc 突然变为 0% 或 100%，SmoothRltSoc 也会同时变为 0% 和 100%。CusRltSoc 可高于 0% 和 100%。SmoothRltSoc 限制在 0% 至 100% 之间。

2.3 算法概述

本节概述了所介绍的测量仪表算法，如 图 2-13 所示。这仅适用于电池电芯算法。电池包算法仅是电池电芯算法的组合。

此算法基于库仑计，可搭配使用前面介绍的其他方法来解决其限制。算法由四个部分组成。容量学习部分用于检测电池静置、执行 OCV 校准和计算 SOH。VGauge 部分用于从保存的一级电池模型输出相关参数。IGauge 是一种用于累积容量的库仑计。混合部分用来计算 NomAbsSoc、CusRltSoc 和 SmoothRltSoc 并输出给客户。

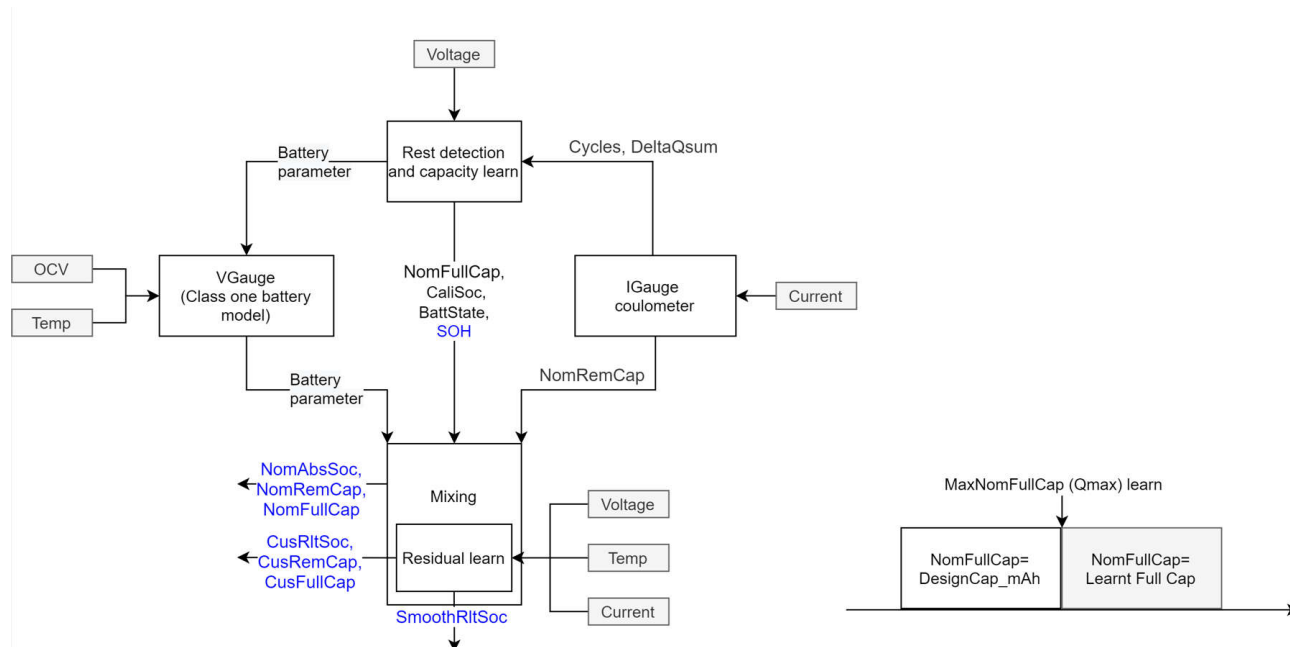


图 2-13. 算法概述

下一节介绍了每个算法部分及其输出到 GUI 或其他功能的关键参数。

2.3.1 电压监测计简介

VGauge 仅用于其他功能方框图，以便从保存的电池模型 *circuitParamsTable* 中搜索电池参数。有关电池模型的更多详细信息，请参阅节 4.2.2。

2.3.2 电流监测计简介

电流测量仪表是一款简单的库仑计，可累积用于不同功能的不同类型容量。

功能的关键输出参数如表 2-1 所示。

表 2-1. IGauge 关键参数

参数	注释
u16DhgCycles	用于告诉用户电池经历了多少次循环。这仅在电池放电时累积电流。当累积容量高于 NomFullCap 时，该值会增加。
iq15DeltaQsum	用于 CapacityLearn 算法部分中的 NomFullCap (Qmax) 计算。
16IF_NomRemCap_mAh	运行时根据检测到的电流计算和报告 NomRemCap。该参数还将在获得 OCV 后进行校准。

2.3.3 容量学习简介

容量学习部分有三个功能：

- 首先是更新电池状态，用于指示何时进行 OCV 或 SOC 校准。
- 其次是校准 OCV 和 SOC。这会在电池静置一段时间且压降达到可接受的范围时进行校准。
- 最后是使用节 2.2.1.1 中所示的方法获得 NomFullCap 和 SOH。

表 2-2 中列出了功能的关键输出参数。

表 2-2. 容量学习关键参数

参数	注释
iGaugeDominationFlg	设置此标志后会生成 u16MaxNomFullCap_mAh，且 SOC 精度达到可接受的精度
iq15CaliSoc_DEC	在设置 ocvCaliFinishFlg 后，容量学习部分使用此参数作为已校准 NomAbsSoc 的记录，以便进一步进行 NomFullCap 计算。电流测量仪表部分也使用该参数在同一周期中重新计算 NomRemCap。
u16CaliOcv_mV	用于在设置 ocvCaliFinishFlg 后校准 NomAbsSoc。
u16NomFullCap_mAh	容量学习部分使用此参数来计算 SOH。电流测量仪表部分也使用该参数来计算 NomRemCap 和放电或充电周期。
u16MaxNomFullCap_mAh	获取第一个 u16NomFullCap 并用于计算 SOH 之后，此参数就会固定不变。
iq15SOH_DEC	用于显示容量下降情况。

2.3.4 混合简介

混合算法用于计算不同类型的 SOC 和容量。

NomAbsSoc 和 NomFullCap 直接来自 IGauge 或 CapacityLearn。

表 2-3 中列出了功能的关键输出参数。

表 2-3. 混合关键参数

参数	注释
iq15CusRltSoc	相对 SOC 根据 NomAbsSoc 紧跟 EmptySoc 和 FullSoc 的变化。
SmoothRltSoc	用于清除 CusRltSoc 的突然跳变。
i16CusRemCap_mAh	客户可以在当前负载和温度条件下使用剩余容量。
u16CusFullCap_mAh	客户可以在当前负载和温度条件下使用全部容量。
i16NomRemCap_mAh	根据 circuitParamsTable 范围计算电池剩余容量。
u16NomFullCap_mAh	根据 circuitParamsTable 范围计算电池的全部容量。
iq15AbsEmptySocMatrix	记录不同温度和电流条件下不同 emptySoc 的矩阵。如果未初始化，则参数会自动学习。
iq15AbsFullSocMatrix	在不同温度下记录不同 fullSoc 的矩阵，支持自动学习。

3 测量仪表 GUI 简介

测量仪表 GUI 也是此实施方案的一个重要部分，如 图 3-1 所示。GUI 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 有三页。

- 第一页是 MCU COM 工具，用于与 MSPM0 进行通信并记录 MCU 传输的电池运行数据。
- 第二页是 SM COM 工具，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。
- 第三页是数据分析工具，用于生成电池参数并评估 SOC 误差。

使用 GUI 执行文件评估实施方案时请多加注意。由于 Python 的限制，GUI 需要 2-3 分钟才能启动。相比之下，GUI 源文件具有更快的 GUI 启动速度。如果用户要在 SM COM 工具下自定义电池测试用例，TI 建议使用源代码。有关如何使用 GUI 的更多详细信息，请参阅后续章节。

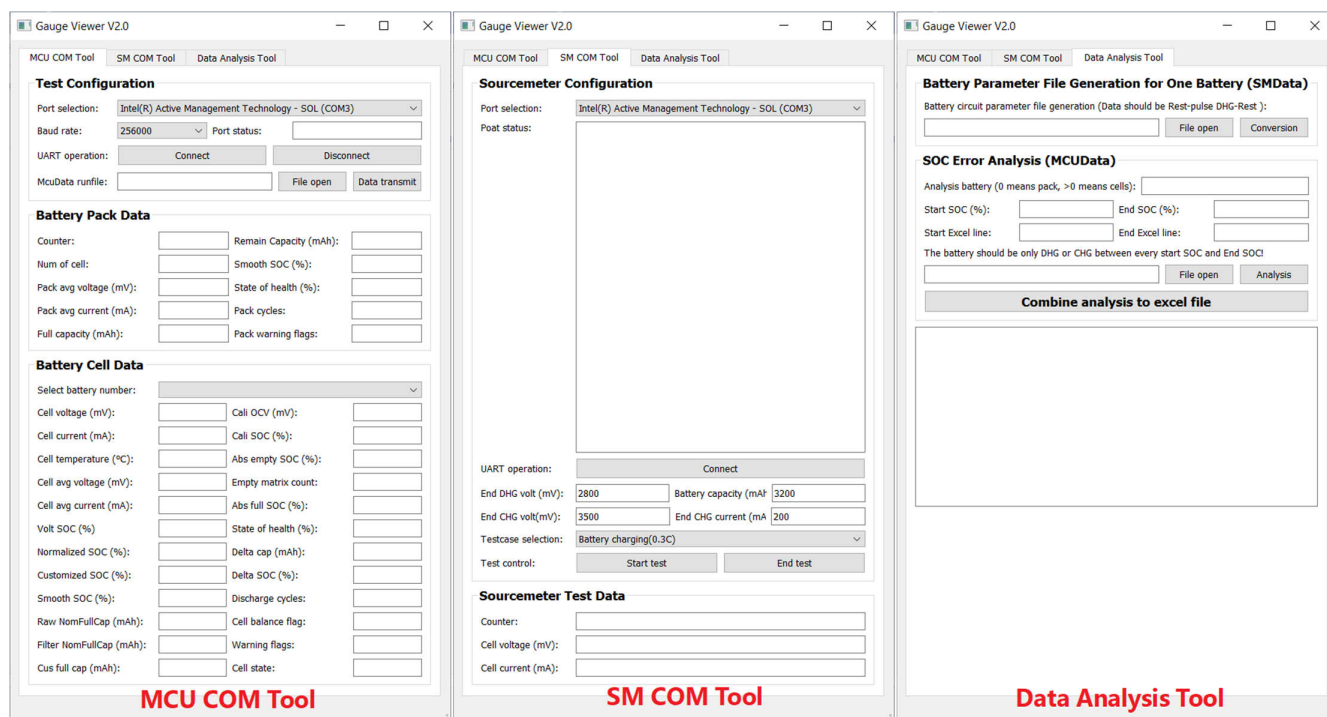


图 3-1. 测量仪表 GUI 功能

3.1 MCU COM 工具

终端用户普遍使用 MCU COM 工具，如 图 3-2 所示。该工具有两个功能：

- 默认功能是从 MCU 接收电池运行数据。默认显示的数据是电池包数据和电芯 1 数据。如果电芯编号大于 1，则可以按照 图 3-2 中的步骤选择所需编号。在测试完成或停止测试后，数据将自动保存到名为 *time-McuData.csv* 的 CSV 文件中。
- 第二个功能是加载所选的 *time-McuData.csv* CSV 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，以便在启用相关测量仪表模式（通信数据输入模式）时支持算法运行。

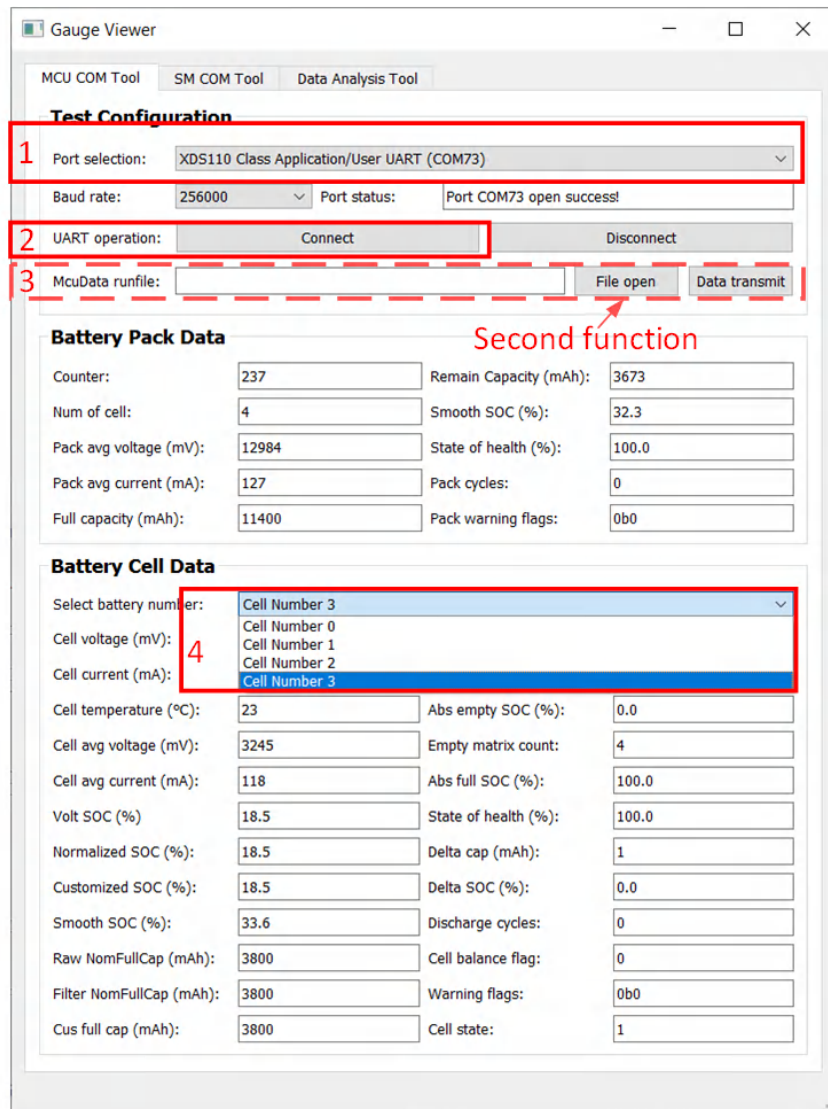


图 3-2. MCU COM 工具功能

3.2 SM COM 工具

SM COM 工具（如 图 3-1 所示）用于控制源表以运行电池测试用例，并显示源表测得的数据。记录数据保存在名为 *time-SmData.csv* 的 CSV 文件中。如果用户要重新创建软件的这一部分，则需安装 NI_VISA。对于硬件，用户需要一个 USB 转 rs232 导线和 Keithley 2602A 源表。

由于源表不是专为 BMS 所设计的设备，因此在 BMS 测试中需考虑其局限性。例如，源表无法对高压电池进行大电流控制。因为电池电压会发生变化，超过 AFE 中设置的阈值，这会触发 AFE 进入具有恒流控制的保护模式。

3.3 数据分析工具

数据分析工具用于执行数据转换和数据分析工作。

第一个功能是电池参数文件生成，如 图 3-3 所示。该功能用于从脉冲放电数据（SMDData 文件）中提取电池电路信息。请按照以下步骤获取电池参数文件。有关更多详细信息，请参阅 节 4.2.1。

第二个功能是分析测量仪表精度性能。在实际测试后，生成 MCUDData 文件。SOC 由测量仪表输出在 Excel 行的开头，该行末尾则保存了先验值。用户输入的起始 SOC 和结束 SOC 是后验值。使用分析按钮后，通过起始 SOC 和结束 SOC 输入来计算 Qmax，然后为每个 Excel 行生成后验值。在一个测试模式中，可能会存在多个 DHG 和 CHG。这意味着用户还需要在步骤 2 和步骤 3 中进行相同次数的输入操作。最后单击第 4 步，GUI 即可输出最终结果。

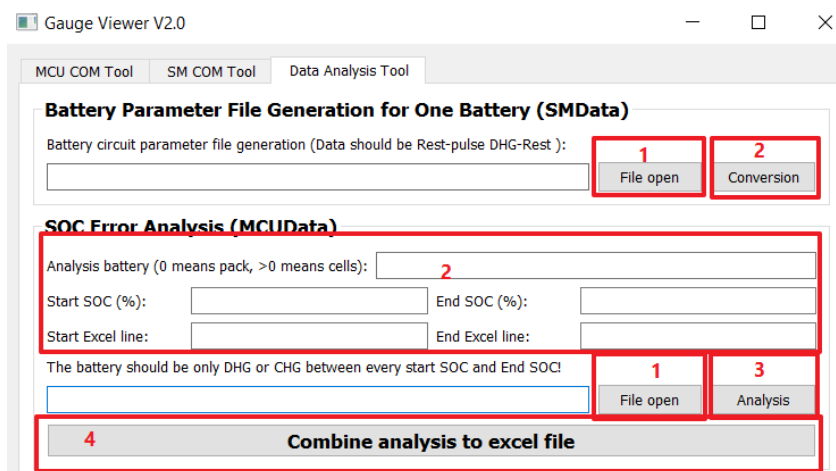


图 3-3. 数据分析工具

4 MSPM0 测量仪表评估步骤

4.1 第 1 步：硬件准备

硬件版：

如果用户要评估此实施方案，需要先进行硬件设置。如果只想要评估测量仪表软件，则仅需一个 LaunchPad™，并将准备好的电压和温度数据输入到 MSPM0 测量仪表算法中即可。

测试设置：

为了测试和评估 MSPM0 测量仪表性能，用户需要准备源表或其他电池测试机来控制电池的充电和放电。如果用户要使用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 第 2 步：获取电池模型

通过脉冲放电测试用例的电池参数计算来获取电池模型。然而，对于实际应用中具有低放电电流的 MSPM0 L1 测量仪表，用户并不需要进行测试。用户可以重复使用代码中的默认模型，也可使用与电池化学成分相关的模型。对于更高级别的 MSPM0 测量仪表实施方案，由于 SOC 校准精度取决于电池模型，TI 建议获取专用电池模型。

4.2.1 电池测试模式

对于测试机，用户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，用户需要以低电流对电池放电（例如以 0.1C 持续 20 分钟）。每个脉冲后的静置时间需要 1-2 小时。随后用户可使用 Vcell 作为 OCV。最后基于该设置，用户会获得大约 30 个数据点，这是 SOC-OCV 表的最低数据量。TI 建议在开始和结束时减小放电电流和缩短放电时间，以便捕捉到电压的快速变化并提高精度，尤其是对于 LiFePO4 电池。

备注

在进行电池测试时，需要将 PCB 和电池插座的影响考虑在内。否则，测试的电阻器小于实际电路电阻器。

表 4-1 展示了用于 LiCO2 和 LiMn2O4 的建议测试模式。对于 LiFePO4，也可以参考该图。

表 4-1. 电池测试模式

参数	值	注释
测试温度	约为 25°C	
启动电压	约为 4.3-4.4V	确保启动电压不低于应用的最大充电电压
终止电压	约为 2.5-3.0V	确保静置电压不高于应用的最小放电电压
放电电流	约为 0.05C-0.1C (容量)	使用低电流可增加数据点。对于开始和最后 5% 的容量，建议使用 0.05C。
放电时间	约为 10-20 分钟	放电时间短可增加数据点。对于开始和最后 5% 的容量，建议使用 10 分钟。
静置时间	1-2 小时	越长越好。但 1 小时就足够了

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)，静置 1 小时后，电压降至 4322mV。然后，电池进行 20 分钟的脉冲放电，并静置 1 小时，以便得到不同 SOC 下的 OCV。此测试在 2450mV 时终止。静置 1 小时后，电压升至 2864mV。因此，SoC-OCV 表的 OCV 范围为 4322mV 至 2864mV。

备注

对于电池测试模式，请确保测试的 OCV 范围足够宽裕，以避免校准后的 OCV 在实际应用中超出 SOC-OCV 表范围。最简单的方法是让 SOC-OCV 表的 OCV 范围覆盖电池工作范围 (MaxFullChgVoltThd 至 EmptyDhgVoltThd)。例如，SOC-OCV 表 (OCV 范围为 4322mV 至 2864mV) 非常适合在 4200mV 至 3000mV 之间运行的电池。

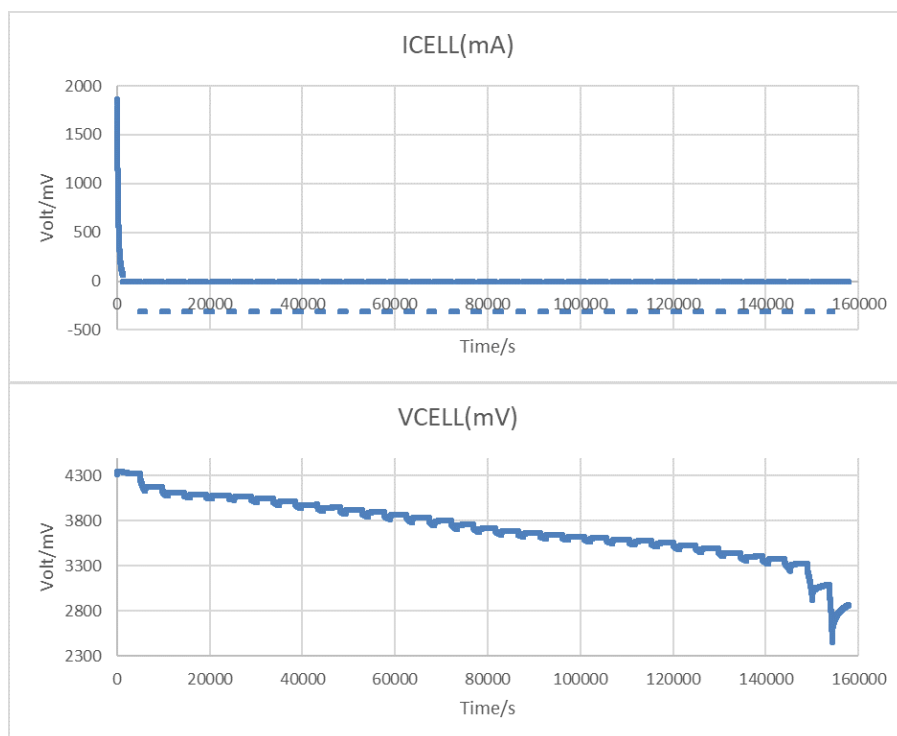


图 4-1. 脉冲放电测试用例

如果您使用 GUI 和建议的源表进行电池测试，请记住在四线模式下使用源表，这样可以减少线路电阻导致的电压检测误差。建议设置如图 4-2 所示。MCU COM 工具用于获取电池运行数据。SM COM 工具用于控制源表进行脉冲式电池充电，并收集电压和电流数据，从而后续生成电池参数。

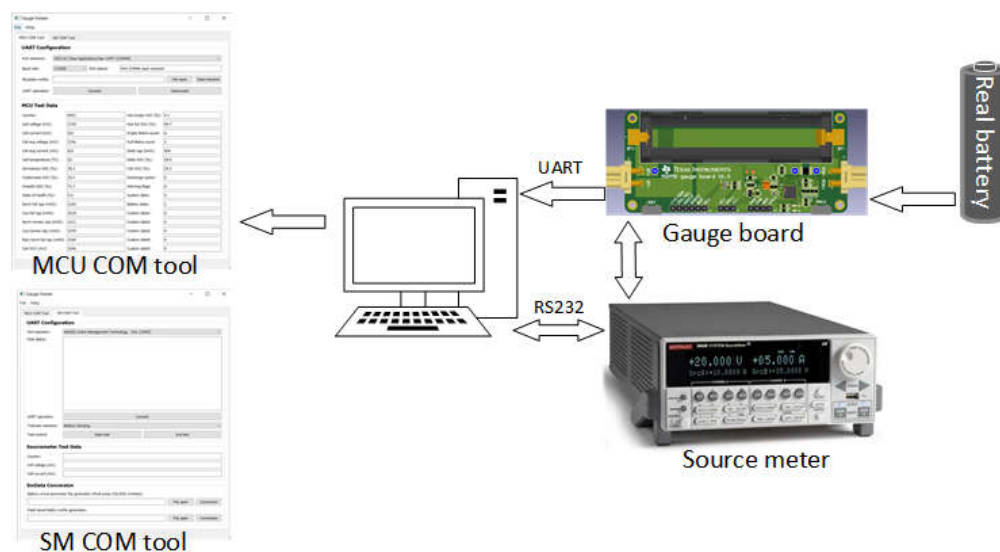


图 4-2. 用于获取电池模型的硬件结构

如果您使用自己的测试机进行测试，则可以根据 SMDData 格式构造测试数据，稍后使用 SM COM 工具来生成电池模型。这是 SMDData 格式。您需要在第 2 行的 B 列和 C 列中输入测试的 Vcell 和 Icell 数据。然后将文件命名为以 "-SmData.csv" 结尾。

2025-04-08-11-17-SmData.csv

	A	B	C	D	E	F	G
1	Time stamp	VCELL(mV)	ICELL(mA)				
2	0	4321.9	1199.8				
3	1	4323.6	1199.8				
4	2	4325.2	1199.8				
5	3	4326.5	1199.8				
6	4	4327.6	1199.8				
7	5	4328.7	1199.8				
8	6	4329.7	1199.8				

图 4-3. SmData 类型

4.2.2 电池模型生成

当电池运行 SMDData 格式或 MCUDData 格式的数据（名称也应遵循命名格式）后，用户即可按照图 4-4 中的步骤，使用 *Battery Parameter File Generation for One Battery* 获取 csv 格式和文本格式的电池模型（电池电路文件）。

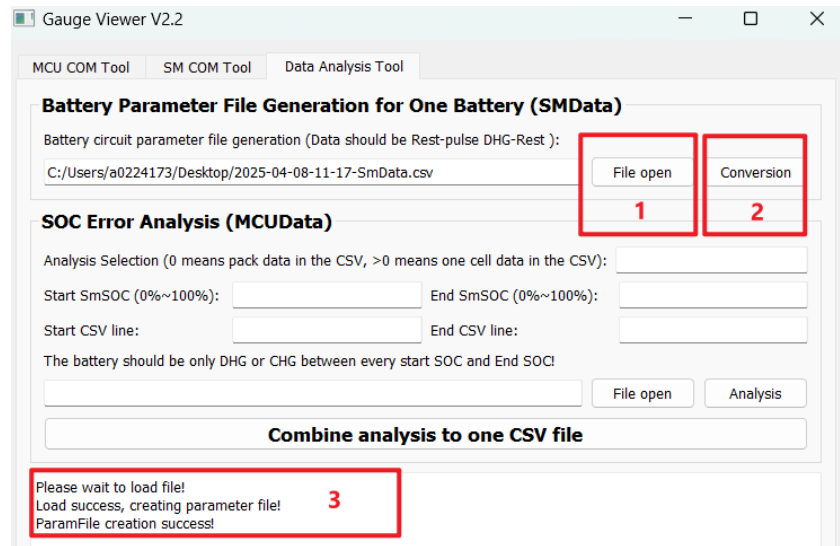


图 4-4. 从 SMDData 生成参数文件

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。随后即可完成电池电路表输入。

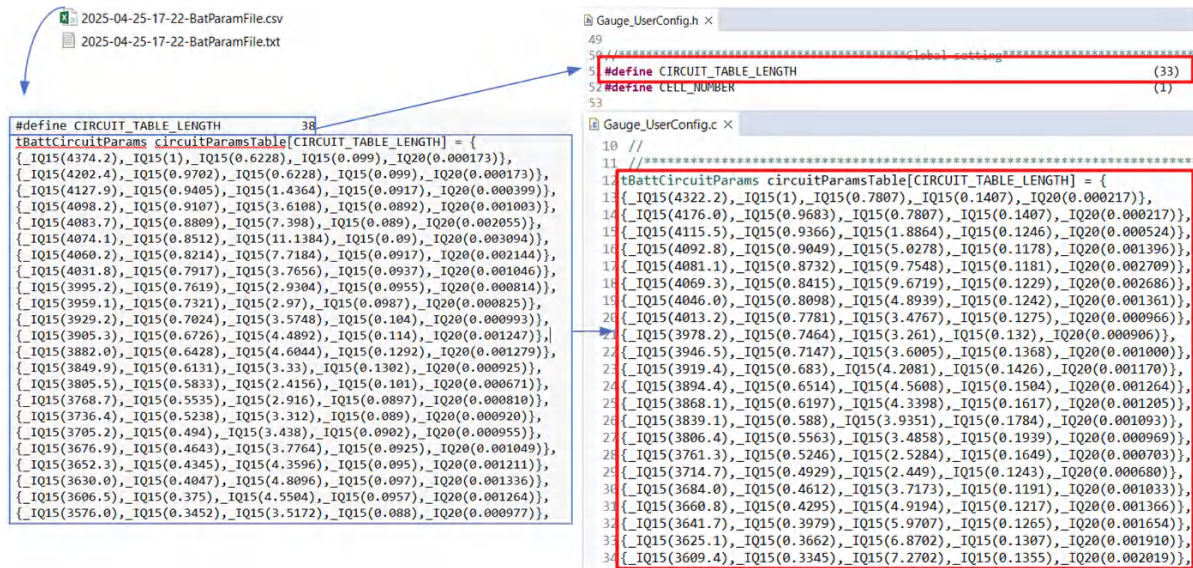


图 4-5. 电池电路表输入

电池电路表（电池模型）的元素有五种组合：

- 第一种情况是 OCV (mV)。
- 第二种情况是 SOC。
- 第三种情况是 Rcell (Ω)。
- 第四种情况是电容系数。
- 第五种情况是斜率。

下面简要介绍了如何生成这些参数。如图 4-6 所示，OCV 等于放电前的最终 Vcell。测试后使用 方程式 2 一同计算得出 Qmax 和 SOC。Rcell 等于图 4-6 中所示的欧姆电阻。一秒钟内的电压变化视为 Rcell 的影响，其值等于 dOcv(mV)/电流 (mA)。电容系数等于 dSOC(%) / dOCV(mV) * Qmax(As) 或 dSOC(%) / dOCV(V) * 3.6 * Qmax(mAh)。斜率等于 dSOC(Dec) / dOCV(mV)。有关详细的参数生成方法，请参阅本文档开发包中分享的 Python 源代码。

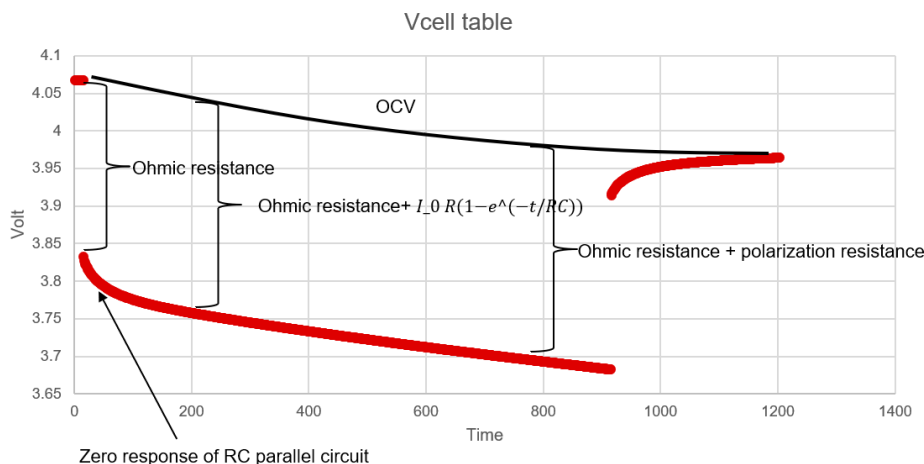


图 4-6. 脉冲放电示例

4.3 第 3 步：输入自定义配置

首先，用户需要根据 Gauge_UserConfig.h 中的应用程序进行系统更改。图 4-7 展示了通常需更改的参数。

```

//*****Algorithm detection mode selection*****//
#define DETECTION_MODE (COMMUNICATION_DATA_INPUT)
//define DETECTION_MODE (DETECTION_DATA_INPUT)

//*****Algorithm communication selection*****//
//define OUTPUT_MODE (NO_OUTPUT)
#define OUTPUT_MODE (UART_OUTPUT)

//*****Data recover *****//
#define PACK_DATA_RECOVER

//*****Global setting*****//
#define CIRCUIT_TABLE_LENGTH (35) //OCV-SOC-Rcell table
#define CELL_NUMBER (4)

```

图 4-7. Gauge_UserConfig.h 设置

表 4-2 展示了这些参数的说明。

表 4-2. 系统配置参数

参数	注释
DETECTION_MODE	影响算法数据输入源，如 节 4.4 中所述。
OUTPUT_MODE	控制是否将测试的数据输出到 GUI，如 节 4.4 中所述。
PACK_DATA_RECOVER	如果定义了该参数，则启用电池包数据恢复功能。这是为满足 MCU 关断时的数据保存要求而保留的。对于数据保存和数据加载功能，用户则需要进行开发。
CELL_NUMBER	电池包的电芯数。
CIRCUIT_TABLE_LENGTH	circuitParamsTable 长度。

其次，用户需要完成 *Gauge_UserConfig.c* 中的数据结构配置。图 4-8 中简要介绍了此测量仪表实施方案中使用的数据结构。

tGaugeApplication 代表电池包。所有与电池包相关的结果均保存在 *tBattPackParams* 中。*tbattGlobalParams[]* 代表电池包中的每个电芯。各算法数据结构如下所示。

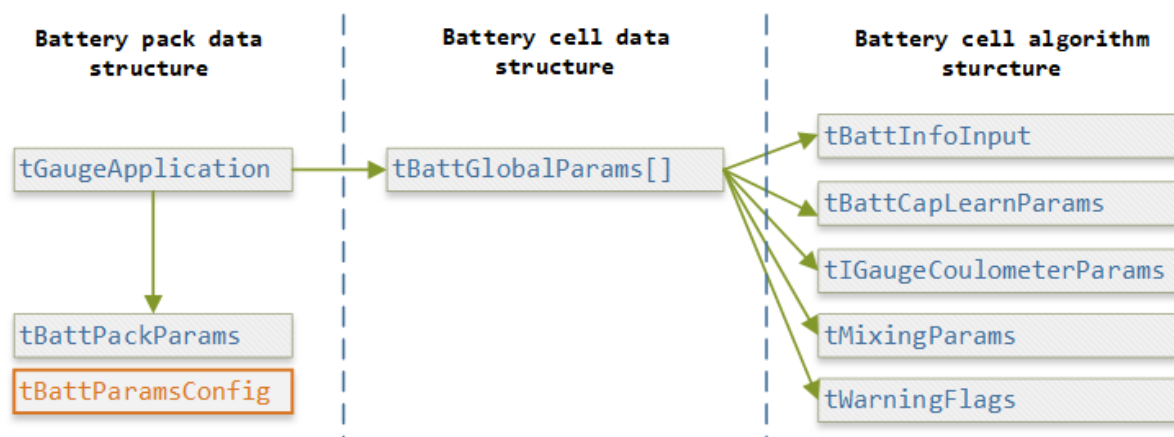


图 4-8. 数据结构

最重要的数据结构是 *tBattParamsConfig*，如 图 4-9 所示。它包含所有电池参数设置和算法设置。

```

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 4-9. tBattParamsConfig 结构

为便于评估，用户只需更改一般配置参数。这些参数分为五个部分。这里简要描述了所有这些相关参数。

表 4-3. 一般配置参数

参数	注释
battModelUse	这使得热模型能够评估内芯温度 iq15EvlCoreTemp_C，该模型仅用于更新和搜索 emptySoc 以及 fullSoc。使用模块时，建议采用的 iq15Tcell_C 温度分辨率为 0.1℃。使用 1℃ 分辨率时，iq15EvlCoreTemp_C 误差会增加约 ±1.5℃。未使用模块时，表面温度用作电池内芯温度。
u16battWeight_g	用于内芯温度评估的电池重量。例如，18650 : 49g ; 21700 : 60g ; 26650 : 96g。

表 4-3. 一般配置参数（续）

参数	注释
u16DesignCap_mAh	设计容量。输入电池标准容量或通过电池参数生成的测试电池容量。
u16MinBattVoltThd_mV / u16MaxBattVoltThd_mV i16MaxChgCurtThd_mA / i16MinDhgCurtThd_mA i8MaxChgTempThd_C / i8MinChgTempThd_C i8MaxDhgTempThd_C / i8MinDhgTempThd_C	电池 Vcell、Icell 和 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。这些不会影响测量仪表的性能。
u16MinFullChgVoltThd_mV u16MaxFullChgVoltThd_mV i16FullChgCurtThd_mA	电池充满相关设置。电池充电电压应在此范围内。u16MinFullChgVoltThd_mV 有助于判断电池电量何时充满。u16MaxFullChgVoltThd_mV 可用作 MCU 上电后的默认充满 OCV。当电流低于 i16FullChgCurtThd_mA 且电压高于 u16MinFullChgVoltThd_mV 时，电池视为已充满。
u16EmptyDhgVoltThd_mV	当电压达到该值时，电池视为电量耗尽。
u8AvgBattParamsUpdateCount	在稳定周期后获得平均数据。

表 4-4. 混合算法相关参数

参数	注释
u16ConvergeStartVolt_mV	当运行时电压低于此参数时，收敛算法开始发挥作用以补偿 emptySoc。
i8TempThd_C[] i16CurtThd_mA[]	用于根据电流和温度阈值找到 iq15AbsEmptySocMatrix[] 和 iq15AbsFullSocMatrix[] 中的理想 emptySoc 和 fullSoc。有关更多详细信息，请参阅节 2.3.4。
u16EmptyOcvMatrix[]	用于计算 iq15AbsEmptySocMatrix[] 的相关 emptySoc。如果全部为 0，则使用 u16EmptyDhgVoltThd_mV 作为空 OCV 来计算相关 emptySoc。系统会在循环后自动学习该操作。 如果用户希望在学习周期中提高测量仪表性能，则用户可以测试相关范围内的 OCV 并将其输入到矩阵中。
u16FullOcvMatrix[]	用于计算 iq15AbsFullSocMatrix[] 的相关 fullSoc。如果全部为 0，则使用 u16MaxFullChgVoltThd_mV 作为满载 OCV 来计算相关 fullSoc。系统会在循环后自动学习该操作。 如果用户希望在学习周期中提高测量仪表性能，则用户可以测试相关范围内的 OCV 并将其输入到矩阵中。

表 4-5. 容量学习算法相关参数

参数	注释
i16UnloadCurtLowThd_mA i16UnloadCurtHighThd_mA	如果电流在此范围内，电池将视为静置。 务必考虑电流检测的噪声。否则静置检测会出现问题。
u8SOHCalcCycleThd	电池放电周期阈值，用以计算 SOH。
iq15DefaultSOH_DEC	电池默认 SOH 值。
u8NomFullCapIIRLevel	用于控制 NomFullCap 的 IIR 滤波器电平。freq_cut = Freq_sample / (2*pi*2^ui8Beta)。

表 4-6. VGauge 算法相关参数

参数	注释
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。这不会对性能产生太大影响。用户可使其保持不变。

表 4-7. IGauge 算法相关参数

参数	注释
i16AvgLeakageCurt_mA	内部漏电流补偿。这意味着无法通过分流电阻器测量的评估电流，是由 MCU 消耗或电池内部漏电流导致的。
i16EqualizationCurt_mA	这意味着要实现均衡电流，用户需要自行处理电池均衡。

4.4 第 4 步：评估

图 4-10 展示了在 Gauge_UserConfig.h 中选择的用于不同条件的不同评估模式。

```

1 //*****Algorithm detection mode selection*****
2 #define DETECTION_MODE (COMMUNICATION_DATA_INPUT)
3 //define DETECTION_MODE (DETECTION_DATA_INPUT)
4
5 //*****Algorithm communication selection*****
6 #define OUTPUT_MODE (NO_OUTPUT)
7 #define OUTPUT_MODE (UART_OUTPUT)

```

图 4-10. 测量仪表模式设置

对于不同的输出模式，UART_OUTPUT 是指通过通用异步接收器或发送器 (UART) 实现数据输出。然后，用户可以通过 USB 连接工具，在 GUI 上观察电池运行参数。“NO_OUTPUT”意味着终止 UART 数据输出。

下一节将详细介绍不同的检测模式。检测数据输入模式是常用模式。所有算法输入均采用 AFE 测试的真实数据。通信数据输入模式的输入数据来自 GUI，单周期的时间限制取决于 UART 通信速度。

4.4.1 检测数据输入模式

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

对于软件设置，在将 Gauge_UserConfig.h 中的检测模式更改为 DETECTION_DATA_INPUT 后，用户需要将测量仪表代码下载到 LaunchPad。

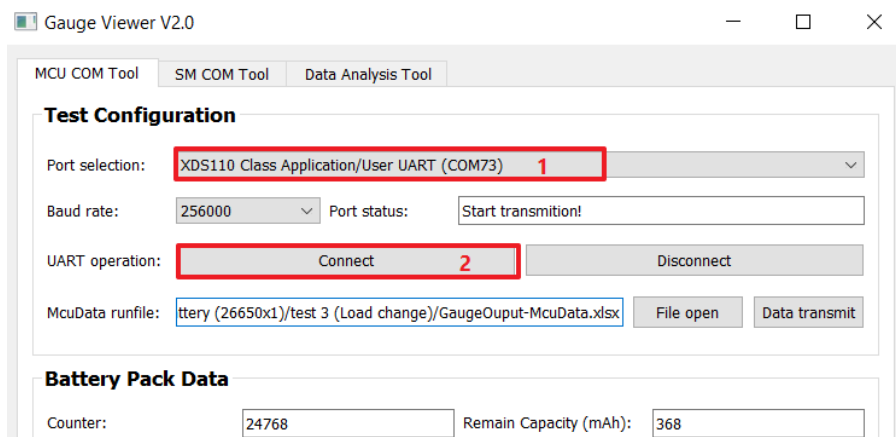


图 4-11. 检测数据输入模式

对于 GUI 控制，用户只需选中名为 XDS110 Class Application 或 User UART 的正确端口，然后点击连接按钮即可。如果 MCU 已经正常工作，用户将在 MCU 测试数据块中看到测试数据。点击断开连接按钮后，数据将以 CSV 格式保存在 GUI 的相同地址下面，其名称为 YYYY-MM-DD-HH-MM-McuData.csv。

备注

如果 Vcell、Tcell 和 Icell 上存在高频噪声，TI 建议用户在算法输入端添加额外的 IIR 滤波器。

4.4.2 通信数据输入模式

对于此模式，电池运行数据通过 GUI 输入。用户由此能够仅使用 LaunchPad 即可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。

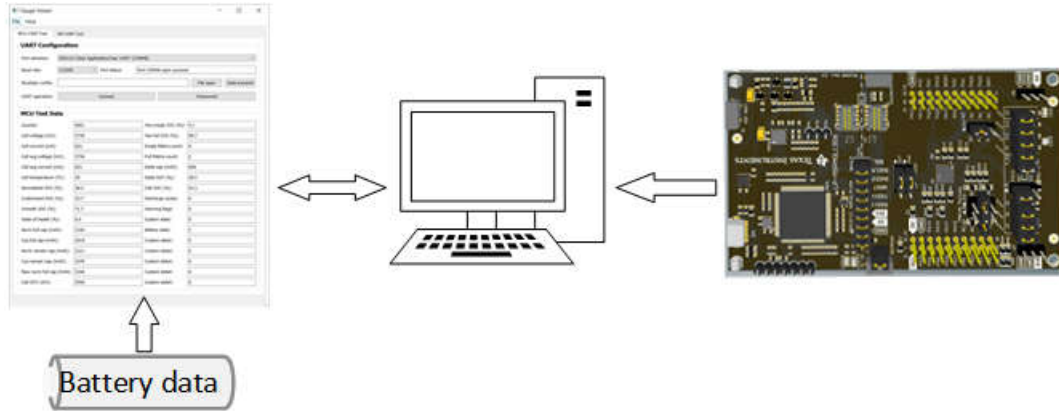


图 4-12. 通信数据输入模式结构

1. 首先，要实现此方法，用户只需要使用 LaunchPad 并进行适当的硬件设置。
2. 其次，在更改 Gauge_UserConfig.h 中的 COMMUNICATION_DATA_INPUT 检测模式后，将测量仪表代码下载到 LaunchPad。
3. 然后，用户需要一个 MCUData 文件。其中说明如何将测试数据传输到 GUI 可识别的文件，尤其是对于那些不是通过 GUI 生成测试文件的用户，这个说明会很有用。用户需要在 B 列中输入电芯编号。然后将电池的各个 Vcell(mV)、Icell(mA) 和 Tcell(°C) 输入到同一个 McuData 文件中的相同列中。为此，用户可先生成一个 McuData 文件，并参考该文件进行传输。最后，使用 -McuData.csv 命名文件。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Time start	CellNum									Vcell	ICELL(mA)	1_TCELL(C)
2	0	4									3394.88	998.7	23
3	1	4									3398.43	998.7	23
4	2	4									3401.33	998.7	23
5	3	4									3403.65	998.7	23
6	4	4									3405.78	998.7	23
7	5	4									3407.68	998.7	23
8	6	4									3409.45	998.7	23
9	7	4									3411.13	998.7	23
10	8	4									3412.73	998.7	23
11	9	4									3414.25	998.7	23
12	10	4									3415.7	998.7	23

图 4-13. McuData 类型

4. 最后，按照 图 4-14 连接 UART COM 端口，点击打开文件按钮在 MCU COM 工具中加载 MCUData 运行文件。点击数据传输按钮后，等待端口状态变为 Start transmit!。

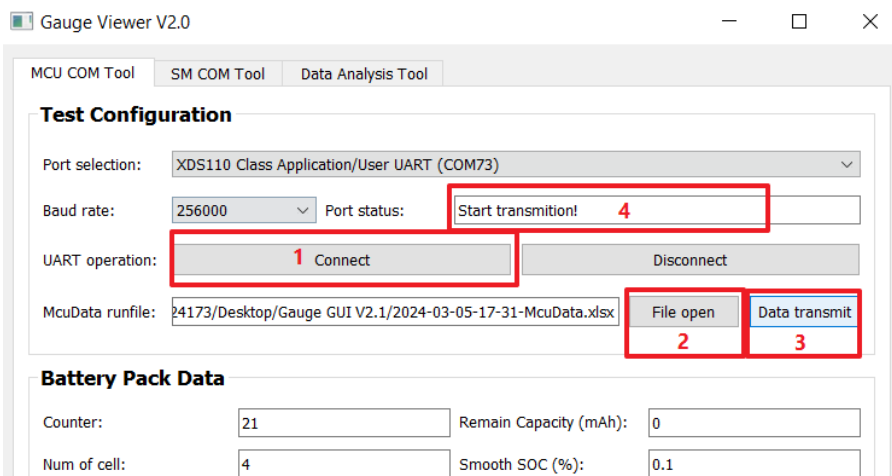


图 4-14. 通信数据输入步骤

用户将从 MCU 接收电池运行数据，如 MCU 测试数据块中所示。完成传输后，GUI 会自动将接收到的数据保存在 GUI 地址下。

4.5 第 5 步：测量仪表性能检查

4.5.1 学习周期

在评估测量仪表性能之前，用户需要了解该算法的学习周期。该算法采用三个参数，通过代码运行学习而来。第一个参数是 **NomFullCap** 和 **MaxNomFullCap**。第二个参数是 **FullSoc**，最后第三个参数是 **EmptySoc**。所有参数均会影响 **SmoothSoc** 输出性能和精度。下文将通过一个电池测试用例来展示此算法的学习能力，如 图 4-11 所示。

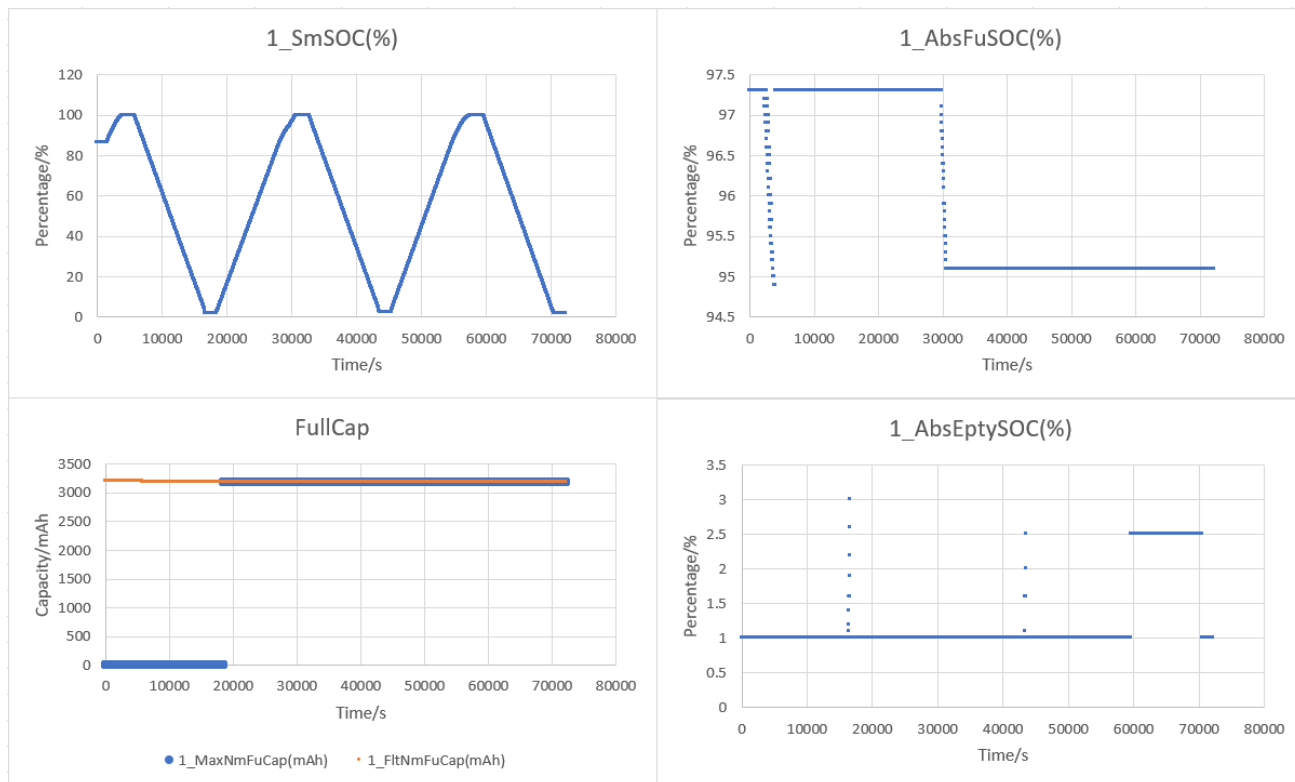


图 4-15. 电池 3 次充放电

设置 iGaugeDominationFlg 后，MaxNomFullCap 从 0 变为非零值。此时，这意味着测量仪表算法会计算 NomFullCap 误差，进而 NomFullSoc 误差会减小到可接受的范围。通常此过程需要一次完全放电或充电。

FullSoc 和 EmptySoc 与 CusSoc 及 SmoothSoc 相对应。这些参数会在电池充满电或电量耗尽时进行调整。但由于 NomFullSoc 误差较大，一开始并不会保存学习结果。在图 4-15 中，FullSoc 会在约 4000 秒时调整到 94.9%，然后在 30000 秒时再次调整。EmptySoc 会在 16000 秒时进行调整，并在 43000 秒时再次调整。在 60000 秒时，由于负载相同，EmptySoc 会加载保存在 EmptySocMatrix 中的值。

当 EmptySoc 和 FullSoc 学习时，SmoothSoc 可能会有较大的跳变，尤其是当 MaxNomFullCap 尚未完成学习且 NomAbsSoc 在学习周期中存在较大误差时。电池将持续从 100% 放电到 0%。图 4-16 展示了一个示例。这在 LiFePO4 电池上很常见。要减少 SmoothSoc 误差，用户可使用 PackRecordLoad 函数为算法提供一个 NomAbsSoc 起始值，并在 battParamsCfg 中设置 EmptyOcvMatrix 和 FullOcvMatrix。如果用户只是想要避免 SmoothSoc 出现大幅跳变，建议在 SmoothSoc 输出端添加一个具有出色频率抑制效果的 IIR 滤波器。

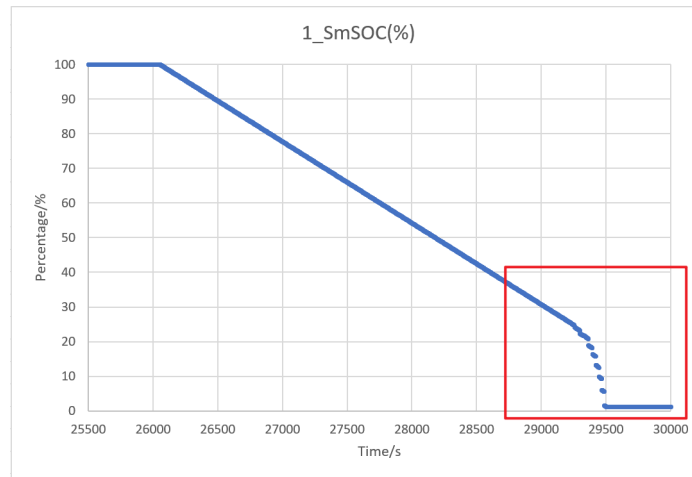


图 4-16. SmoothSoc 跳变

4.5.2 SOC 和 SOH 精度评估

如果用户想要评估 SOC 精度，请参阅 SmoothSoc 而非 NomSoc 或 CusSoc。原因在于从终端用户角度来看，所需的 SOC 是相对 SOC，其只能从 0% 变为 100%，并且在 SOC 输出中不会出现 SOC 跳变。

由于 SmoothSoc 是一个相对 SOC，因此无法通过搜索 SOC-OCV 表来获得精确值。SmoothSoc 达到 100% 和 0% 时的数据点清晰可见。若要获得除 0% 和 100% 以外的 SmoothSoc 精确值，用户首先需要构建 SmoothSoc 和 CusRemCap 之间的对应关系，方法是将电池充电至 100% 并放电至 0%。然后，在相同的负载和温度下重新进行测试。用户之后即可使用 CusRemCap 获得 SmoothSoc 精确值。有关进一步说明，请参阅表 4-8。

表 4-8. 获得 SmoothSoc 精确值

SmoothSoc 精确值	条件
100%	当电流降至 FullChgCurtThd 且电压介于 MinFullChgVoltThd 和 MaxFullChgVoltThd 之间时。
0%	当电压降至 EmptyDhgVoltThd 时。
其他值	首先获取 SmoothSoc-CusRemCap 表，然后在相同条件下重测，以使 EmptySoc 和 FullSoc 一致。然后，用户可使用 CusRemCap 获得准确的 smoothSoc 值。

如节 4.5.1 所示，在至少一次放电或充电后，可获得 MaxNomFullCap，同时算法开始学习 EmptySoc 和 FullSoc 并保存到矩阵中。为了获得更精确的 SmoothSoc，TI 建议用户首先完成两个电池周期。第一个周期用于学习 MaxNomFullCap。第二个周期用于学习和保存 SmoothSoc 及 FullSoc。随后，测量仪表算法即可发挥出理想性能。

SOH 等同于 $\text{MaxNomFullCap}/\text{NomFullCap}$ 。在代码中，SOH 输出需增加 2% 的裕度，以避免由于 NomFullCap 波动而在第一个电池周期中低于 100%。如果用户需要检测 SOH 精度，TI 建议改为检测 NomFullCap 精度。若要在多个周期后评估 NomFullCap 精度，建议流程如下：

1. 让电池静置 1 小时，测量电芯电压，并使用 SOC-OCV 表将电压作为 COV 映射到 SOC1。
2. 在一个充电或放电周期后，电池静置 1 小时，再次测量 OCV 并映射到 SOC2。
3. 计算 NomFullCap： $\text{NomFullCap} = Q_{\text{use}} / (\text{SOC2} - \text{SOC1})$ 。

5 MSPM0 测量仪表解决方案

本节使用两个不同的测试用例来展示 MSPM0 L2 测量仪表的能力：

- 能够从单节电芯检测轻松切换到多节电芯检测
- 能够处理不同的电池类型，尤其是对于 LiFePO4，其 SOC-OCV 表更为平坦

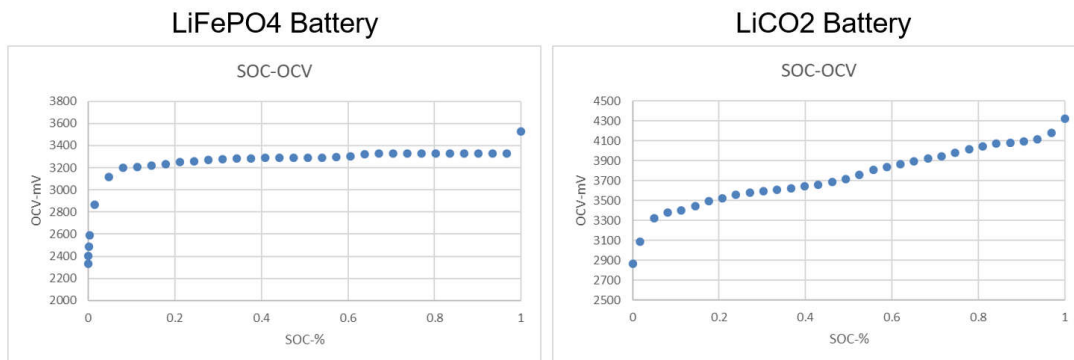


图 5-1. 电池 SOC

本节还展示了 MSPM0 硬件特性，因此其尤为适用于 BMS 实施方案，包括内部高精度模拟外设和多种通信方法（CAN/UART/串行外设接口 (SPI)/内部集成电路 (I2C)）。

5.1 MSPM0L1306 和 1 节 LiCO2 电池

实施方案特性：

- 总体实施方案需要大约 15K 的闪存和 1.3K 的 SRAM
- 无通用异步接收器或发送器 (UART) 通信（NO_OUTPUT 模式）时的电流消耗约为 9uA。

实施方案优势：

- 具有内部模拟外设的单芯片实施方案
- 支持自校准电流检测，误差约为 1%
- 高性能测量仪表算法

5.1.1 硬件设置介绍

硬件板通常用于评估单节电池测量仪表的实施方案。

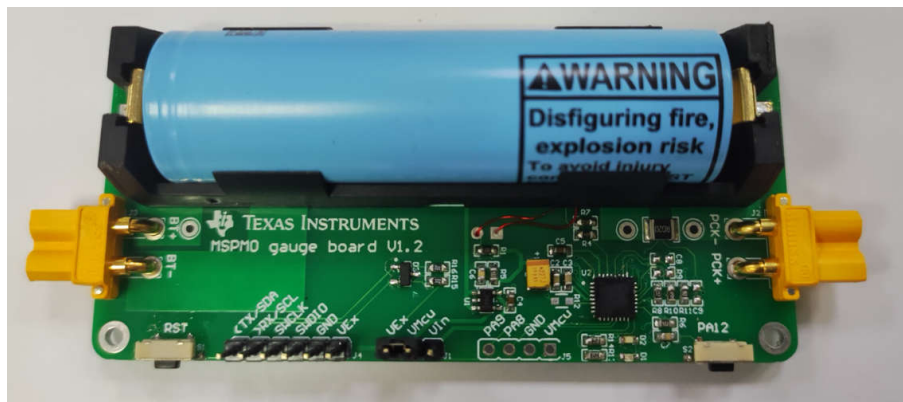


图 5-2. MSPM0 测量仪表硬件板

图 5-3 展示了硬件简要方框图，其中显示了此演示使用的所有引脚。该实施方案测试模数控制器 (ADC) 通道 13 的电流、ADC 通道 5 的温度和 ADC 通道 1 的电压。

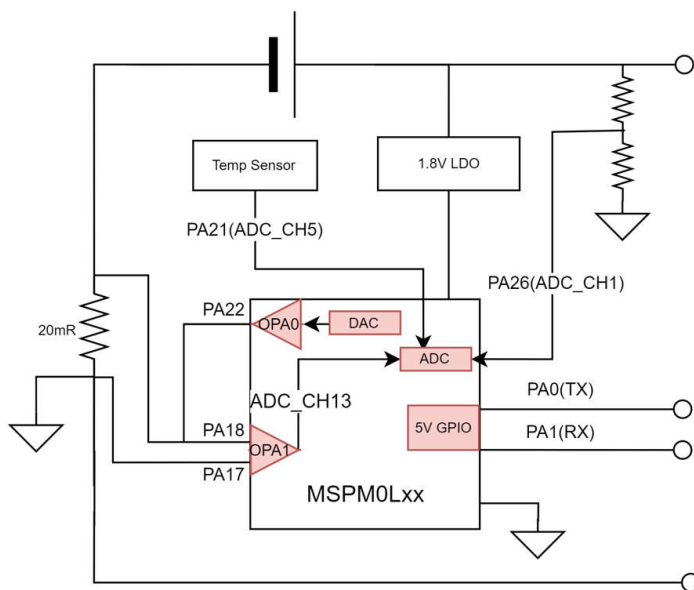


图 5-3. MSPM0 测量仪表板方框图

借助用于电流检测的内部 OPA，在 $\pm 2A$ 负载下，其在室温下的检测误差可达到 $\pm 0.25\%$ 。有关硬件介绍及性能的更多信息，请参阅基于 MSPM0 应用手册的自校准电流检测解决方案。

测量仪表板说明如图 5-4 所示。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，然后为 MCU 提供 1.8V LDO。这可确保实现出色的模拟性能。当 MCU 在大约 500ms 内通电时，MCU 会校准 ADC+OPA 以进行电流检测。此时，电流为 0。否则说明存在恒定的电流失调。

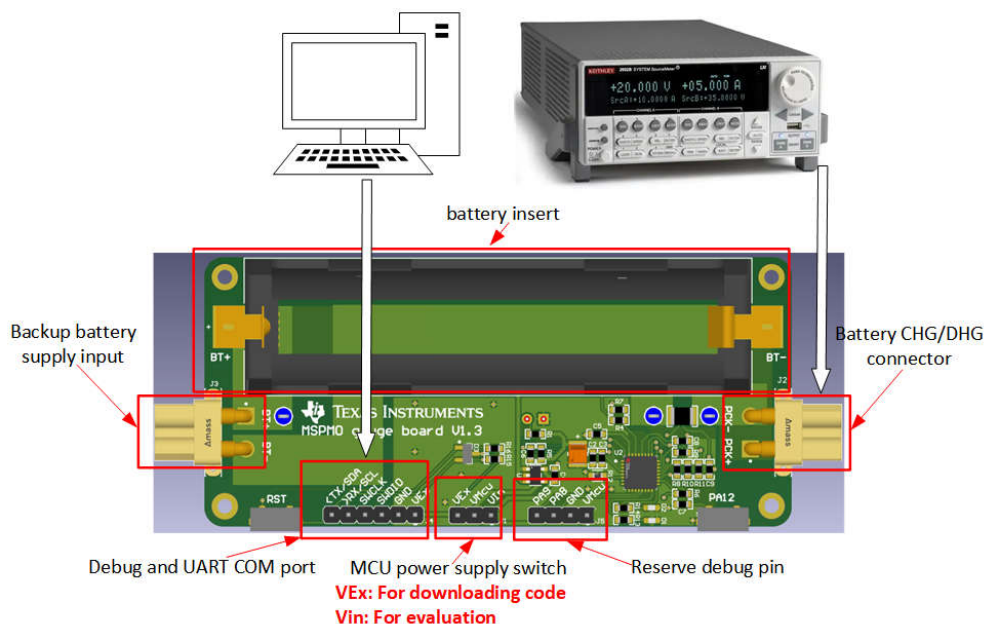


图 5-4. 测量仪表板说明

如果用户在通信数据输入模式下使用 MSPM0L1306 LaunchPad，则按如下方式连接 UART 引脚。这同样适用于软件更改。

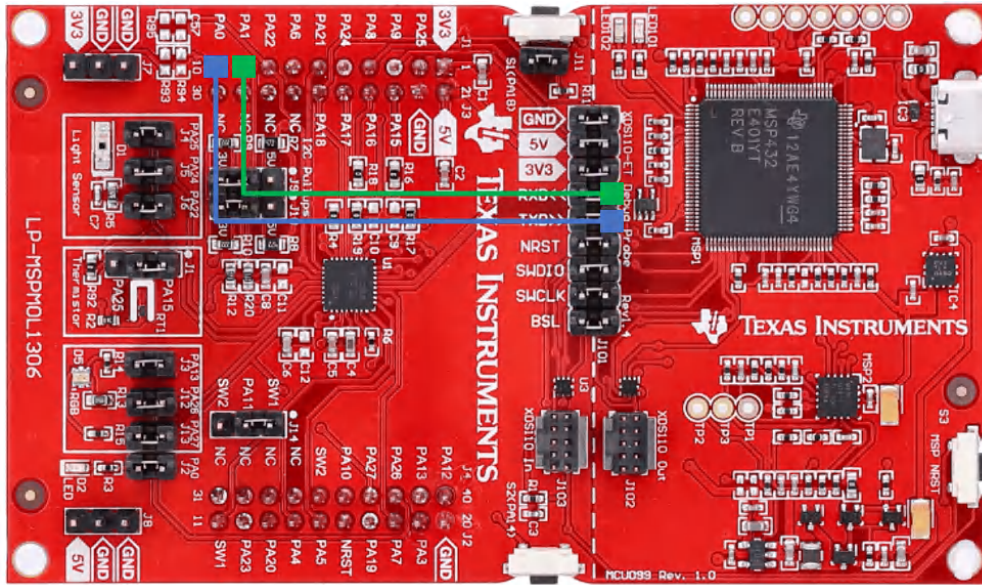


图 5-5. MSPM0L1306 LaunchPad UART 连接

5.1.2 软件和评估简介

在开始软件开发和评估之前，TI 建议参阅 [MSPM0 设计流程指南用户指南](#)，以便对 MSPM0 生态系统有基本的了解。这可帮助用户了解 MSPM0 的开发。

图 5-6 演示了该软件工程。与测量仪表算法相关的工程和文件包含五个部分。所有 MSPM0 工程的其他文件都相同。

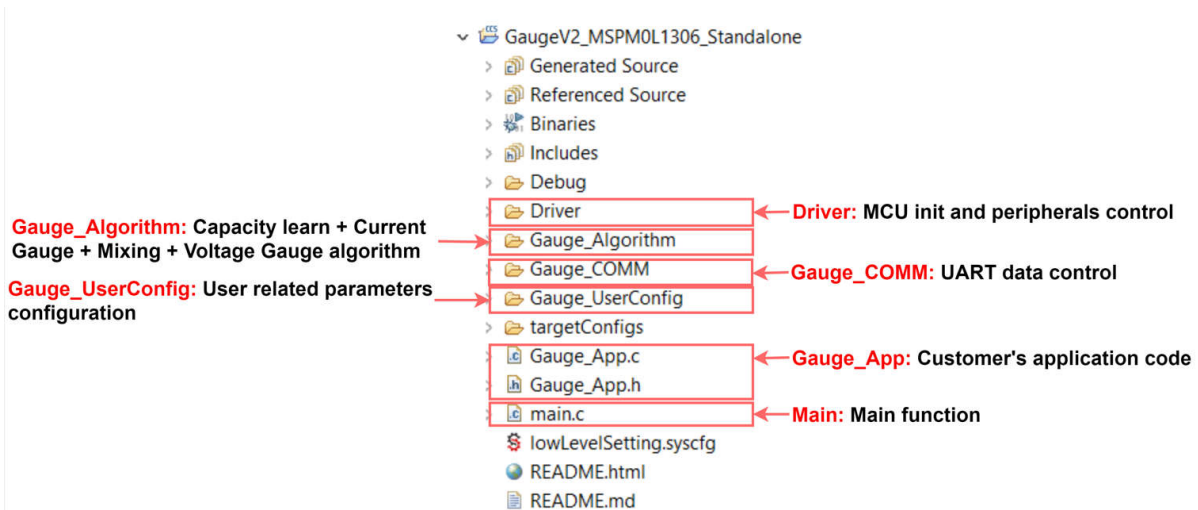


图 5-6. MSPM0 测量仪表软件工程视图

对于 Gauge_UserConfig 部分，相关说明请参阅 [节 4.3](#)。[节 2](#) 简要介绍了 Gauge_Algorithm 部分。Driver 部分包含所有与 MCU 相关的控制设计。它将 Icell、Vcell 和 Tcell 数据准备到 Gauge_Algorithm 中。Gauge_COMM 部分处理所有 UART 协议。Gauge_App 部分包含对高级测量仪表算法的调用。客户可以在此部分定制功能。Main 部分包含最高的系统功能代码。

请按照 [节 4](#) 中步骤更新 Gauge_UserConfig 文件夹中的配置：

- 生成电池模型或使用默认电池模型
- 更新 `tBattParamsConfig` 结构中的配置
- 根据电池电芯来更新 `battGlobalParams_xx` 和 `battGlobalParamsArray`。
- 更新检测模式、通信模式、电路表长度和电芯编号

用户对 MSPM0 至 XDS110 进行编程后，可以使用 GUI 检查并记录结果。

5.1.3 电池测试用例

5.1.3.1 性能测试

这是基于 25°C 下的 3200mAh LiCO2 电池进行的测试。u16MaxFullChgVoltThd 设置为 4200mV。EmptyDhgVoltThd 设置为 3000mV。

备注

确保在 MCU 上电之前电池处于稳定状态，且在测试之前电池处于静止状态。否则在达到第一个 SOC 输出时会出现错误。

测试模式如下：

- 在不同负载下执行脉冲放电和脉冲充电。
- 在不同负载下，持续充电和放电，无需静置 4 个周期。

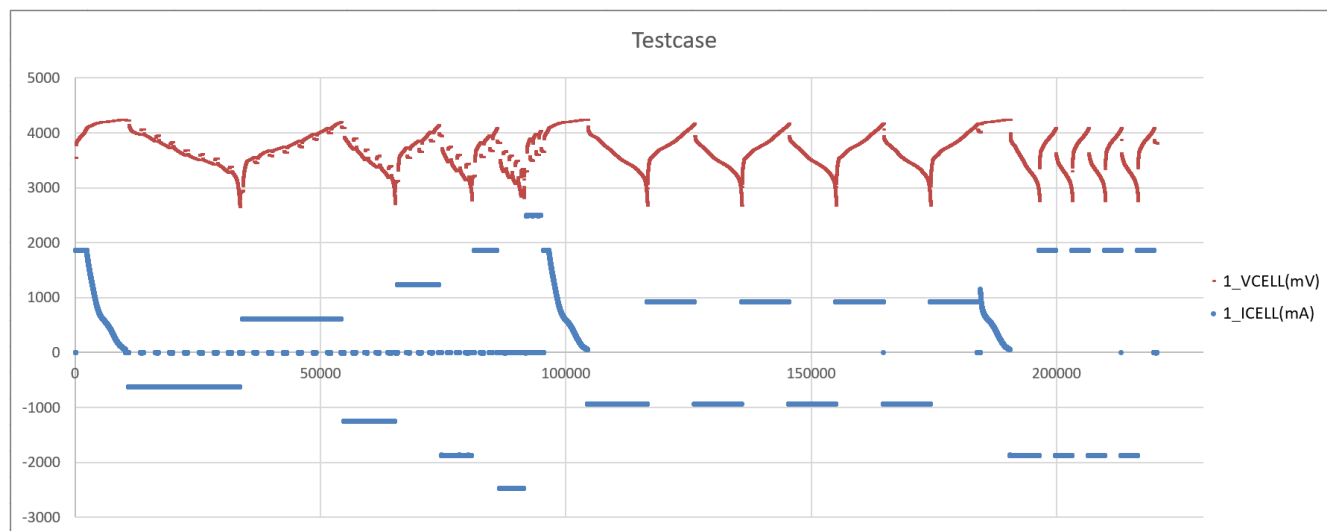


图 5-7. 电池测试用例

在开始时，NomSOC、CusSOC 和 SmoothSOC 存在明显的差距。请参阅图 5-8 中的测试结果。这是由第一次 OCV 校准误差引起的。

在放电终止时，CusSOC 会存在一些差异，这是因为受到 EmptySOC 的影响。在电池静置时，SmoothSOC 是平稳的，没有跳变。所有数据均控制在 0% 至 100% 范围内。

对于不同的 NomFullCap，几乎每次静置后都会更新 FltNomFullCap。借助数字滤波器，NomFullCap 变得越来越准确。MaxNomFullCap 从 0 变为非零值，这意味着输出 NomFullCap 具有可接受的精度。

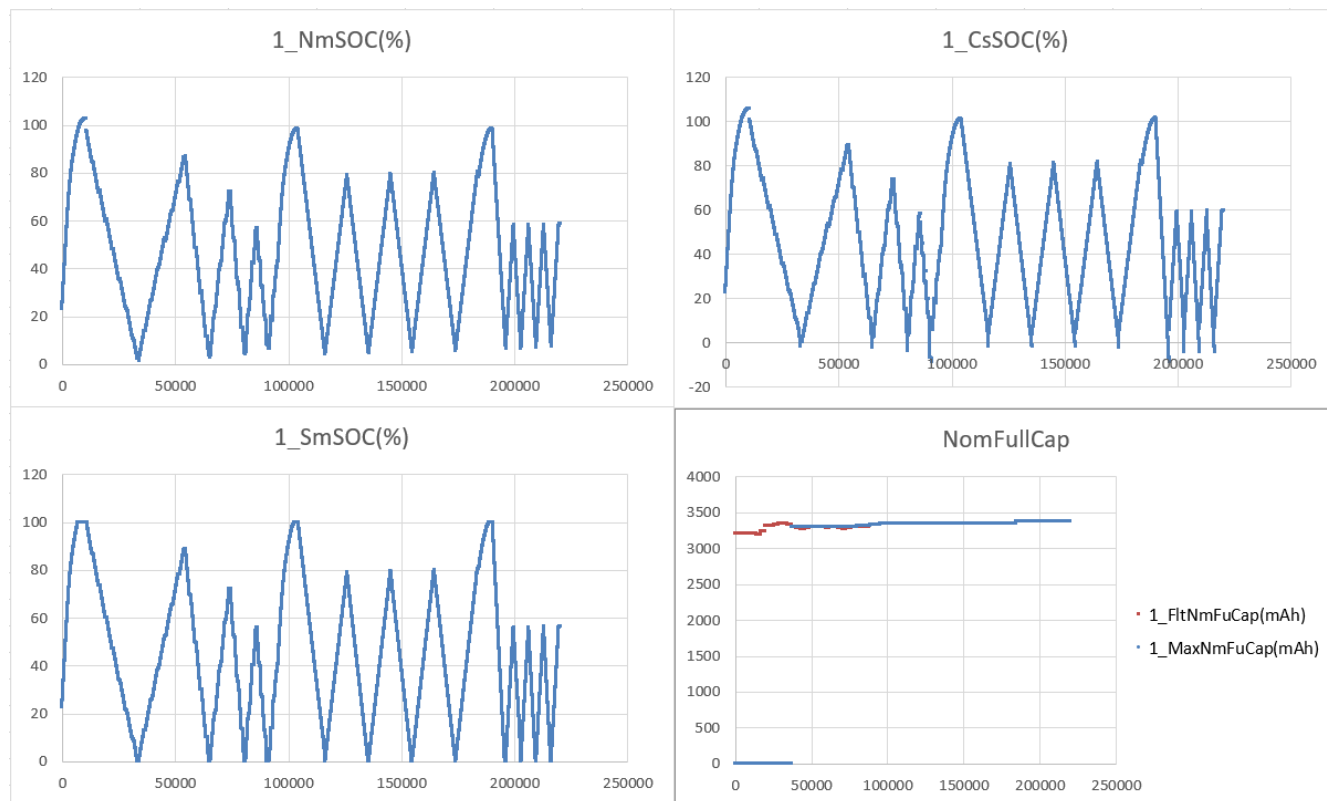


图 5-8. 电池测试结果

如果用户要在调试模式下以 Q 格式检查更多参数，则需要使用右键点击该值并选择相关的 Q 值格式。

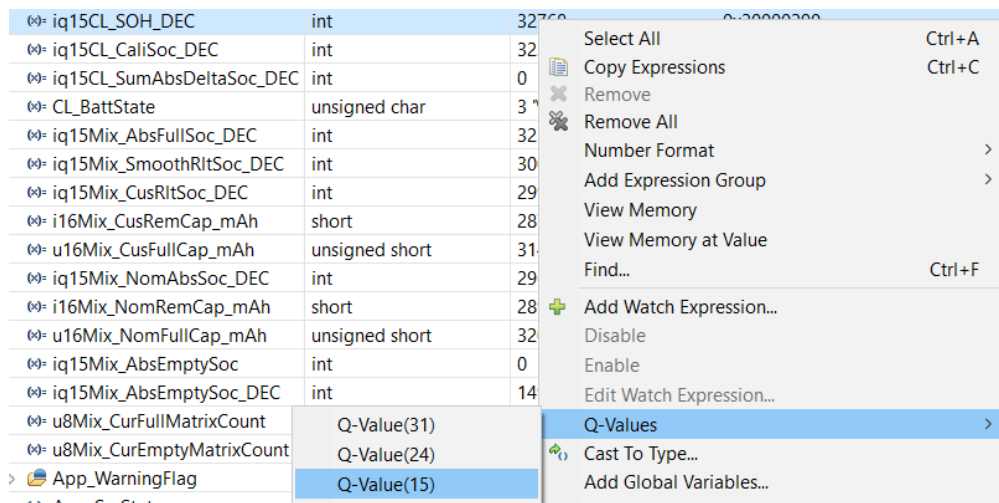


图 5-9. 读取 Q 值

5.1.3.2 电流消耗测试

MSPM0 测量仪表板主要侧重于评估功能，而不考虑如何优化功耗。基于测量仪表板测得的电流会较高。为了进行优化，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。

下面是在 NO_OUTPUT 模式下移除钽电容器、温度传感器和分压器电阻器后的电流测试结果。

EnergyTrace™ Profile	
Name	Live
▼ System	
Time	56 sec
Energy	1.368 mJ
▼ Power	
Mean	0.0275 mW
Min	0.0073 mW
Max	6.4136 mW
▼ Voltage	
Mean	3.3000 V
▼ Current	
Mean	0.0083 mA
Min	0.0022 mA
Max	1.9435 mA
Battery Life	CR2032: 3 year 30 day (est.)

图 5-10. 电流消耗

5.2 MSPM0G3507、BQ76952 和 4 节 LiFePO4 电池

实施方案特性：

- 总体实施方案需要大约 15K 的闪存和 2.9K 的 SRAM

实施方案优势：

- 集成了 CAN 的高性能 MCU
- TI 的 BMS 组合实施方案
- 高性能测量仪表算法

5.2.1 硬件设置介绍

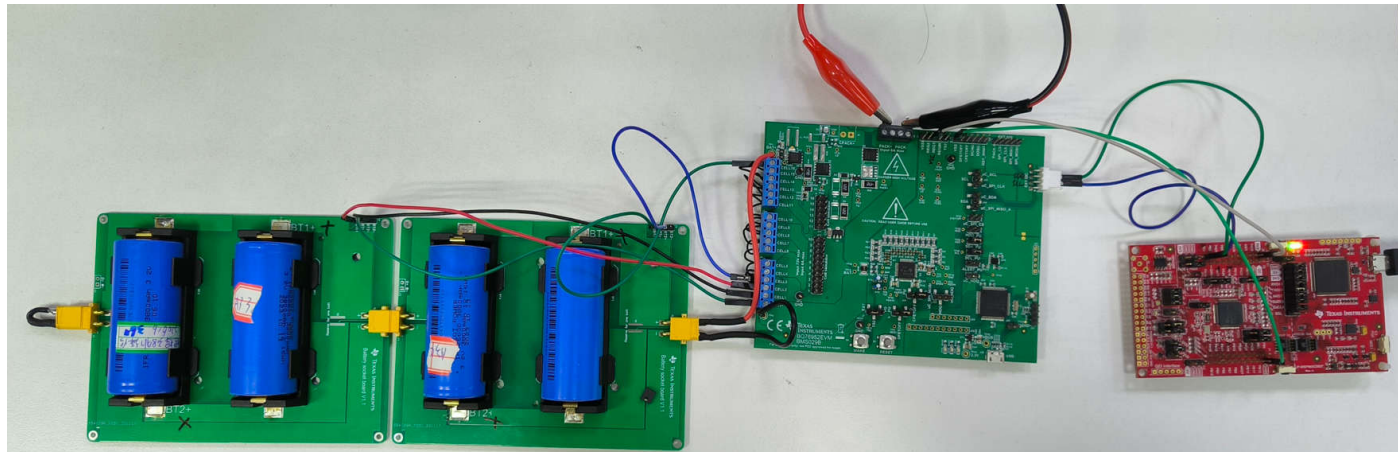


图 5-11. MSPM0G3507 LaunchPad 和 BQ76952EVM

硬件板基于 MSPM0G3507 LaunchPad 和 BQ76952EVM 构建。

有关硬件连接，请参阅图 5-12。在 MSPM0 LaunchPad (SDA : PB3 , SCL : PB2) 和 BQ76952EVM (SDA: J17 PIN3, SCL: J17:PIN2) 之间连接电源和 I2C。记得要移除电芯仿真跳线，并添加用于 I2C 上拉的跳线。

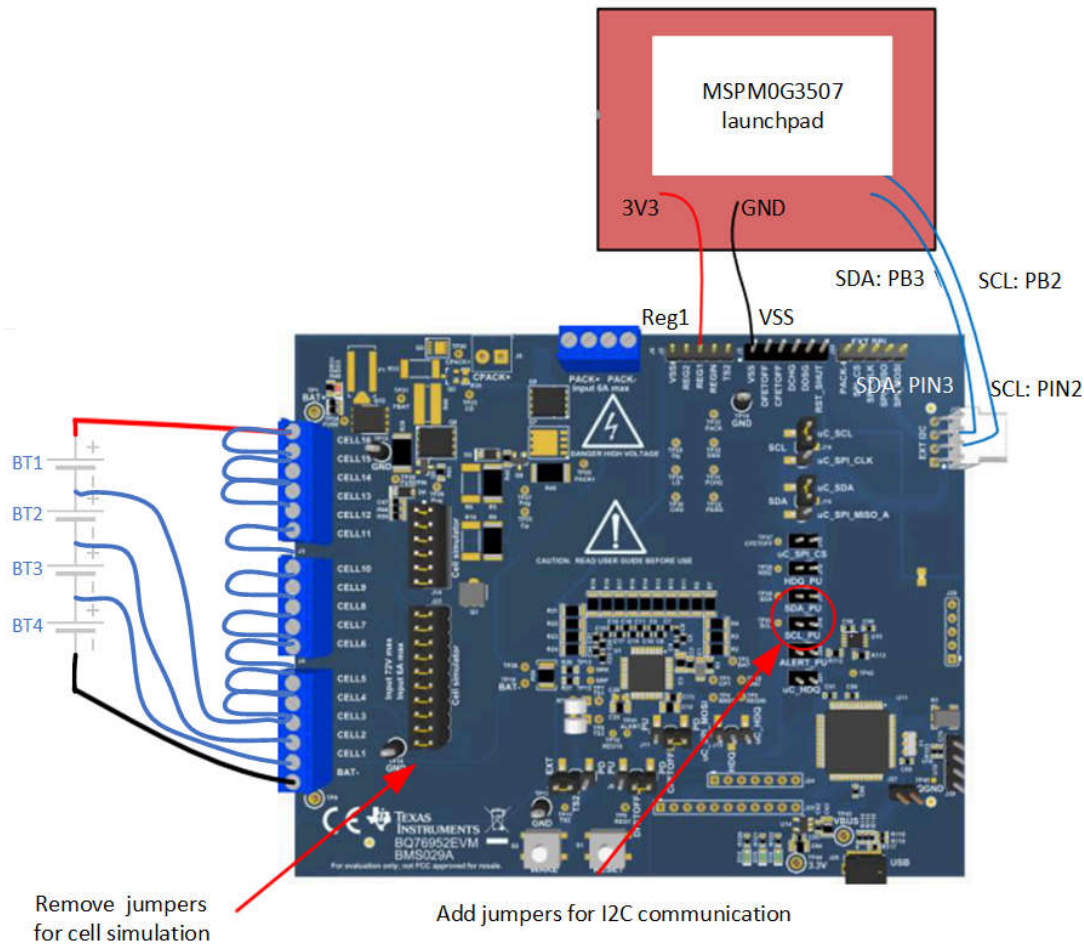


图 5-12. MSPM0G3507 和 BQ75952EVM 方框图

如果用户要在通信数据输入模式下使用 MSPM0G3507 LaunchPad，则与默认的 MSPM0G3507 LaunchPad 硬件设置相比，不需要其他硬件更改。

5.2.2 软件和评估简介

在开始软件开发和评估之前，TI 建议参阅 [MSPM0 设计流程指南用户指南](#)，以便对 MSPM0 生态系统有基本的了解。这可帮助用户快速了解 MSPM0 的开发。

图 5-13 展示了软件工程。与测量仪表算法相关的工程和文件包含五个部分。所有 MSPM0 工程的其他文件都相同。

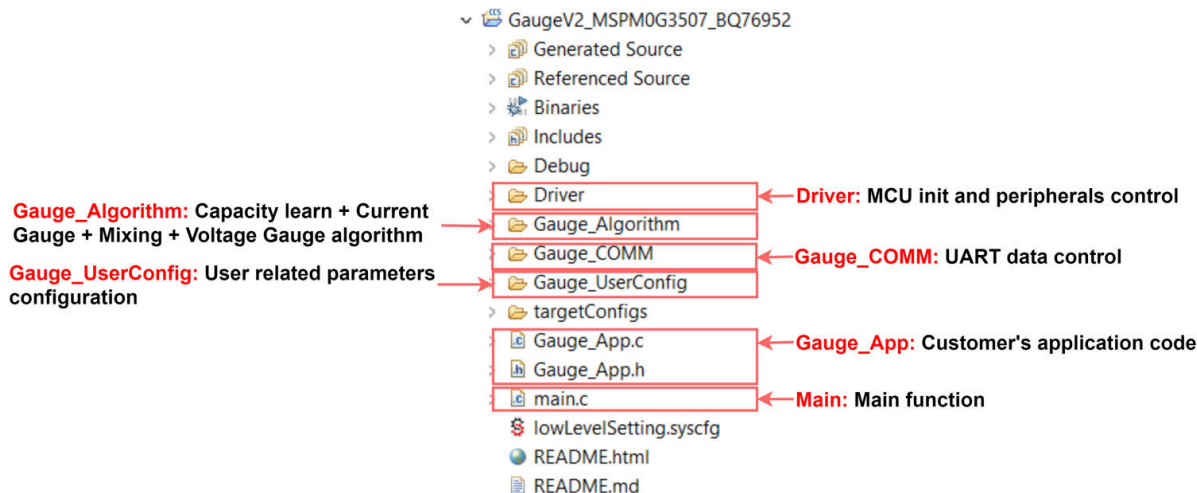


图 5-13. MSPM0 测量仪表软件工程视图

对于 Gauge_UserConfig 部分，相关说明请参阅 节 4.3。节 2 简要介绍了 Gauge_Algorithm 部分。Driver 部分包含所有与 MCU 相关的控制设计。它将 Icell、Vcell 和 Tcell 数据准备到 Gauge_Algorithm 中。Gauge_COMM 部分处理所有 UART 协议。Gauge_App 部分包含对高级测量仪表算法的调用。客户可以在此部分定制功能。Main 部分包含最高的系统功能代码。

请按照 节 4 中步骤更新 Gauge_UserConfig 文件夹中的配置：

- 生成电池模型或使用默认设置
- 更新 tBattParamsConfig 结构中的配置
- 根据电池电芯来更新 battGlobalParams_xx 和 battGlobalParamsArray。该实施方案由四个电芯组成。
- 更新检测模式、通信模式、电路表长度和电芯编号

对 MSPM0 至 XDS110 进行编程后，可以使用 GUI 检查并记录结果。

5.2.3 电池测试用例

5.2.3.1 性能测试 1 (脉冲放电)

这是基于 25°C 下的 3800mAh LiFePO4 电池进行的测试。u16MaxFullChgVoltThd 设置为 3800mV。EmptyDhgVoltThd 设置为 2300mV。

备注

确保在 MCU 上电之前电池处于稳定状态，且在测试之前电池处于静止状态，否则在达到第一个 SOC 输出时会出现错误。

测试模式如下：执行脉冲放电 2 次。图 5-14 展示了电池包中电池电芯的状况。由于数字源表存在功率限制，仅运行简单的测试。

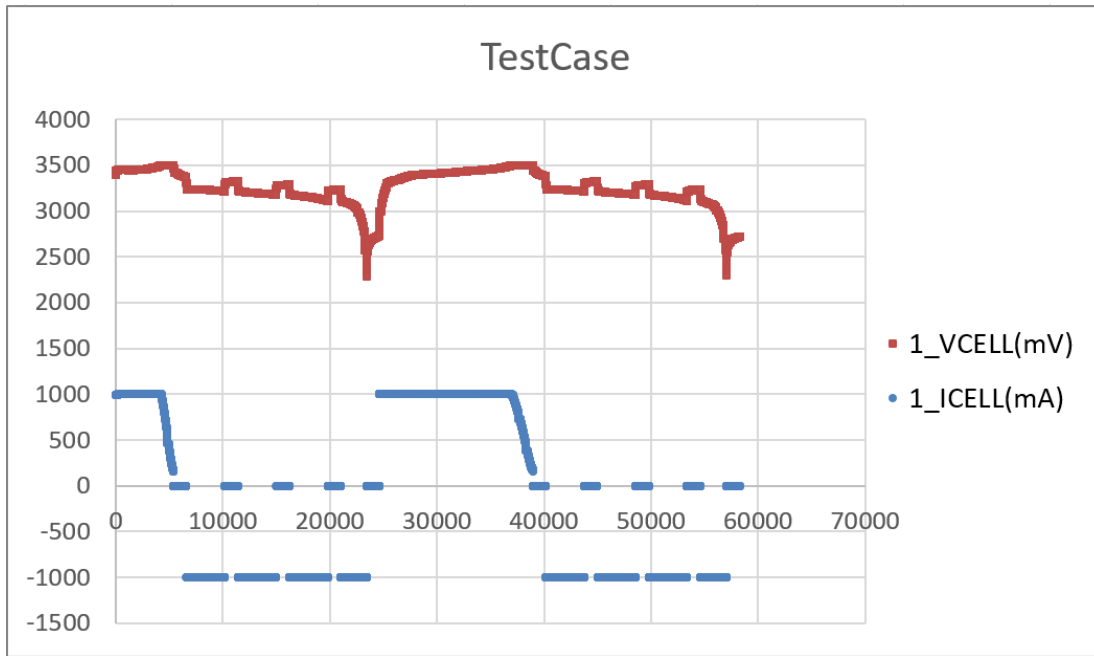


图 5-14. 电池测试用例

在开始时，NomSOC、CusSOC 和 SmoothSOC 存在明显的差距。请参阅图 5-14 中的测试结果。这是由第一次 OCV 校准误差引起的。之所以这么高是因为校准点位于 LiFePO4 SOC-OCV 平坦区域下面。

以下是单节电池的结果。借助集成式数字滤波器，即使在第一个放电周期内，NomSOC 的变化也与实际情况吻合。但由于电流负载仅为 1A，在 CusSOC 中看不到 EmptySOC 有多大的影响。

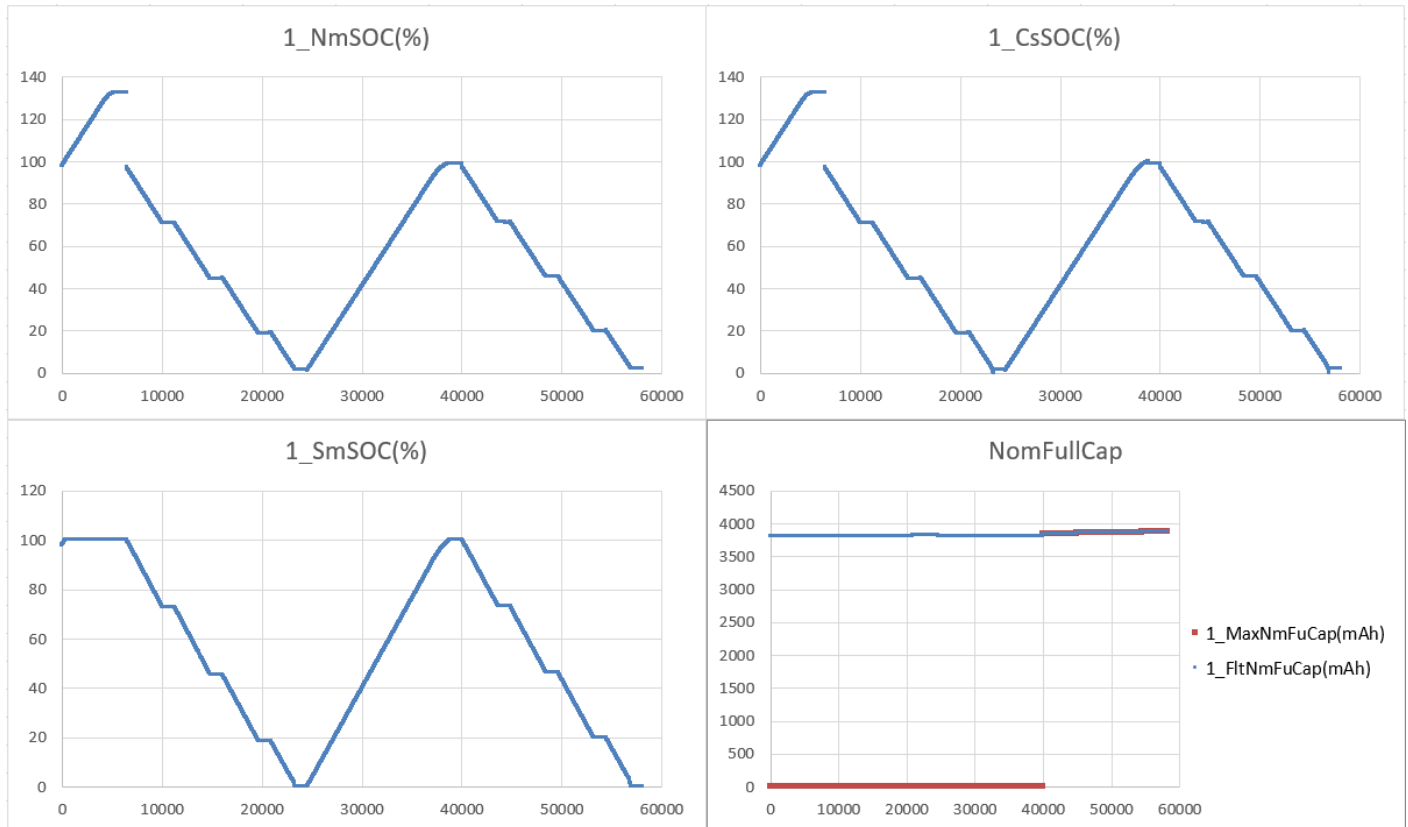


图 5-15. 电池电芯测试结果

以下是电池包的结果 PackSOC，该结果是电池包中所有电芯的最小 SmoothSOC。PackFullCap 是所有 CusFullCap 值的组合，受 EmptySOC 和 FullSOC 的影响。因此 PackFullCap 图表中会出现数据跳变。PackRemCap 的情况也是如此，它是所有 CusRemCap 的组合。

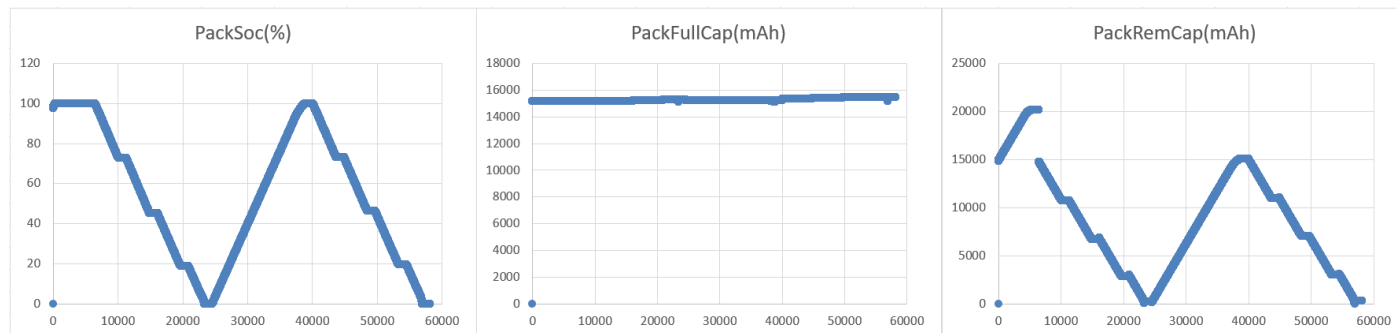


图 5-16. 电池包测试结果

5.2.3.2 性能测试 2 (负载变化)

这是基于 25°C 下的 3800mAh LiFePO4 电池进行的测试。u16MaxFullChgVoltThd 设置为 3800mV。EmptyDhgVoltThd 设置为 2300mV。

备注

确保在 MCU 上电之前电池处于稳定状态，且在测试之前电池处于静止状态，否则在达到第一个 SOC 输出时会出现错误。

测试模式如下：执行恒定放电 2 次，然后改变负载。图 5-17 展示了电池包中电池电芯的状况。由于数字源表存在功率限制，仅运行简单的测试。

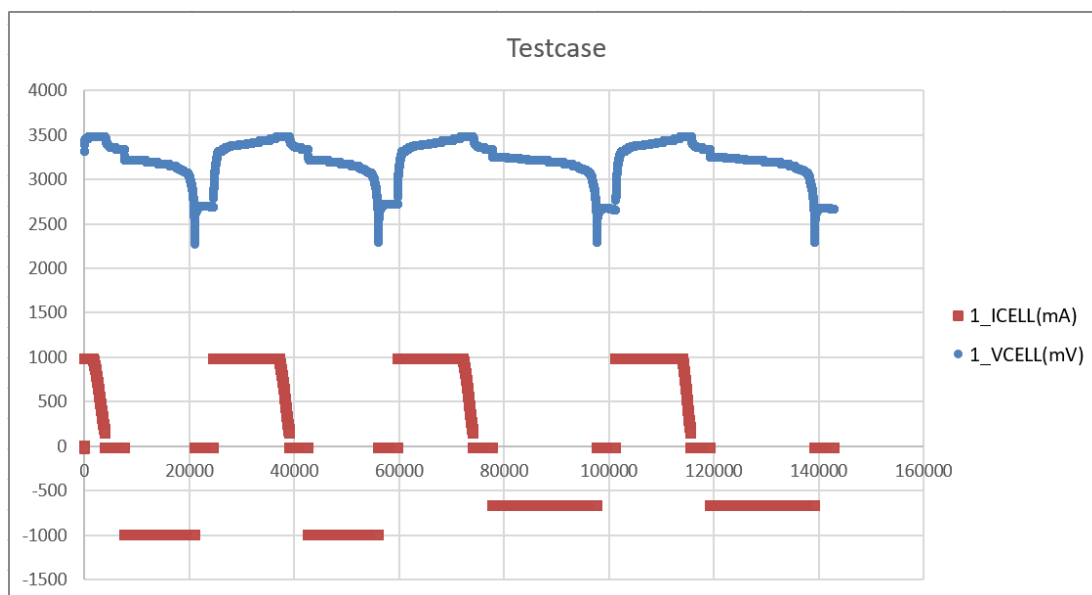


图 5-17. 电池测试用例

请参阅图 5-17 中的测试结果。在开始时，NomSOC、CusSOC 和 SmoothSOC 存在明显的差距。这是由第一次 OCV 校准误差引起的。

由于残留学习算法的作用，当电压达到放电终止电压 (2300mV) 时，SmoothSOC 可在 0% 至 100% 之间正常运行。同时需注意，EmptySOC 需要学习周期，这意味着如果用户没有输入 iq15AbsEmptySocMatrixInput，则当电池达到其第一次放电终止电压时，SmoothSOC 误差会很大。

对于不同的 NomFullCap，几乎每次静置后都会更新 FitNomFullCap。借助数字滤波器，NomFullCap 变得越来越准确。在 MaxNomFullCap 从 0 变为非零值后，意味着输出 NomFullCap 具有可接受的精度。

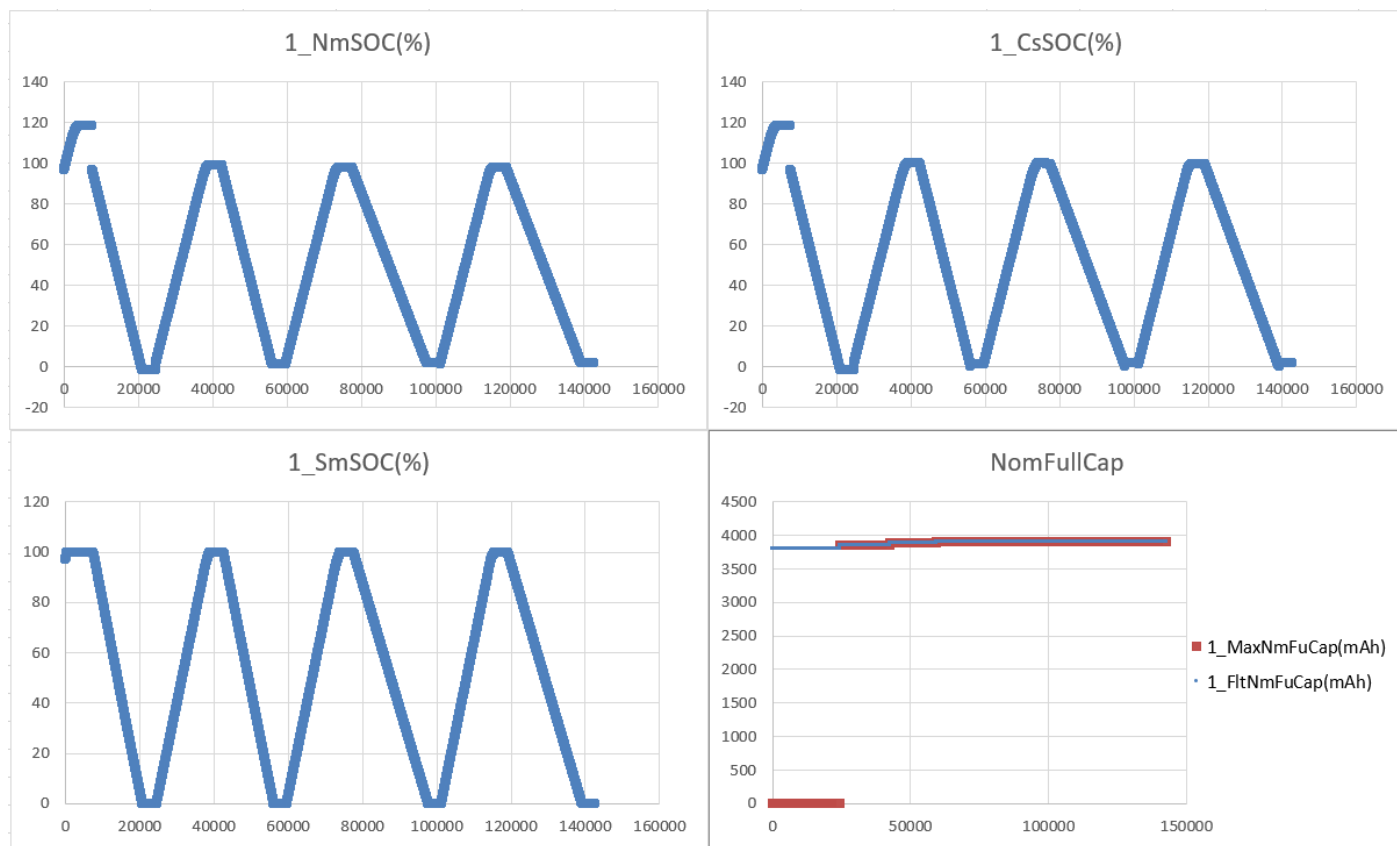


图 5-18. 电池测试结果

图 5-19 显示电池包的结果。

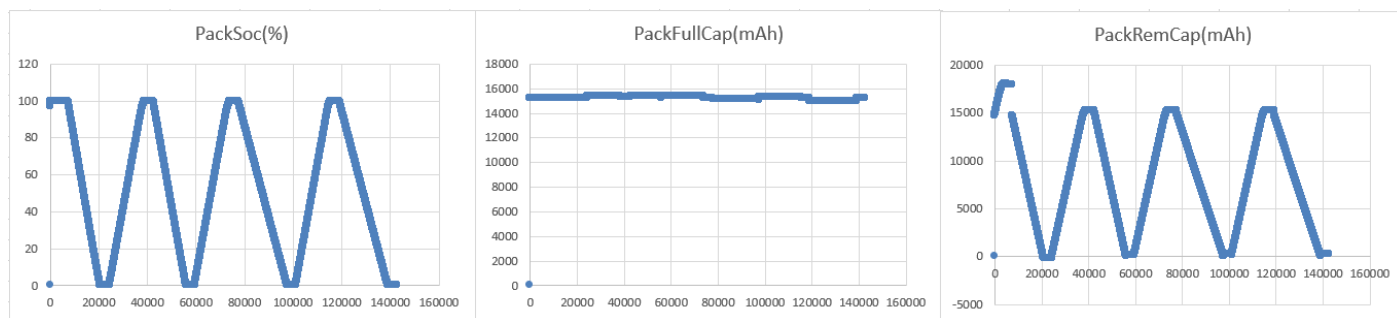


图 5-19. 电池测试结果

5.3 MSPM0L1306 和 BQ76905

此实施方案用于展示 MSPM0L1306 和 BQ76905 之间的设置，其中包含 5 节电芯，旨在提高典型 BQ 器件的评估速度。与 BQ76952 不同，BQ76905 不具有无源均衡功能，该功能需要用户自行实现。

以下是硬件连接：

表 5-1. 硬件连接

	LP-MSPM0L1306	BQ76905 EVM
GND	J7-1	J4-1
SCL	PA1	J4-2
SDA	PA0	J4-3

这是实际设置画面。请注意，在进行评估前请将 CELL5 短接至 CELL4，并将 CELL3 短接至 CELL2 电芯输入端子。

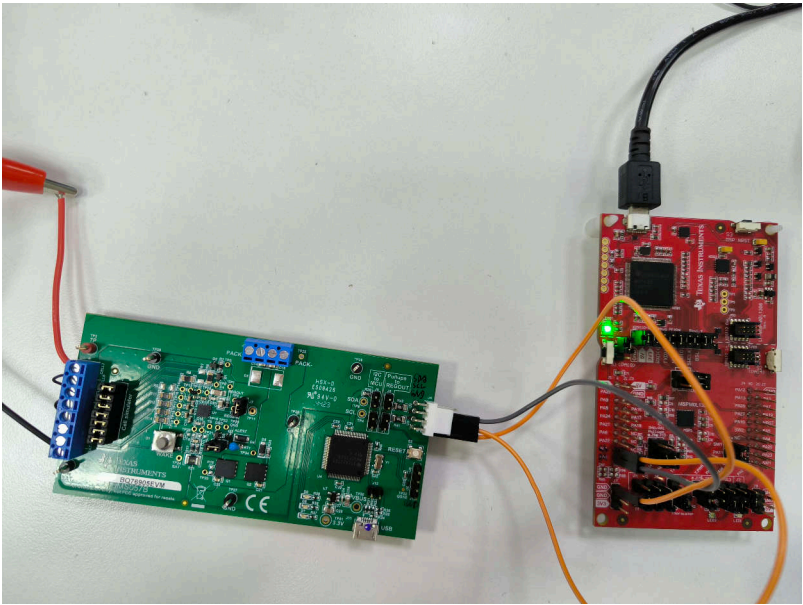


图 5-20. 硬件设置

6 总结

本文档概述了采用 MSPM0 微控制器的 2 级 BMS 测量仪表实施方案。该解决方案通过集成双源数据融合算法，获得了长期充电状态 (SOC) 精度 (典型条件下为 $\pm 1.5\%$)。用户可按照本文档完成电池参数设置、算法评估和性能分析。该解决方案仅需调整少量参数，即可用于各类锂离子电池，并满足用户的特性和精度要求。

7 参考资料

- 德州仪器 (TI) : [MSPM0 L1 测量仪表解决方案指南](#) ,
- 德州仪器 (TI) : [基于 MSPM0 的自校准电流检测解决方案](#)
- 德州仪器 (TI) : [MSPM0 MCU 开发指南](#)

8 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision A (June 2025) to Revision B (June 2025)	Page
• 更新了本文档的“摘要”中的.....	1
• 更新了 节 1 中的开发套件链接。.....	3

重要通知和免责声明

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 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。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
版权所有 © 2025，德州仪器 (TI) 公司