

Application Note

MSPM0 L2 测量仪表解决方案指南

Eason Zhou

摘要

本应用手册介绍了基于 MSPM0 的 2 级测量仪表解决方案。它可检测电压、电流和温度，从而直接计算荷电状态 (SOC)。本应用手册内容包括解决方案特性、硬件简介、GUI 简介、软件简介和评估。

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/slaae5>。

内容

1 引言	3
2 算法介绍	3
2.1 电池基础知识简介.....	3
2.2 不同的 SOC 和使用的公式.....	5
2.3 算法概述.....	8
3 测量仪表 GUI 简介	11
3.1 MCU COM 工具.....	11
3.2 SM COM 工具.....	12
3.3 数据分析工具.....	12
4 MSPM0 测量仪表评估步骤	13
4.1 第 1 步：硬件准备.....	13
4.2 第 2 步：获取电池模型.....	13
4.3 第 3 步：输入自定义配置.....	17
4.4 第 4 步：评估.....	19
5 MSPM0 测量仪表解决方案	22
5.1 MSPM0L1306 + 1 节 LiCO2 电池.....	22
5.2 MSPM0G3507 + BQ76952 + 4 节 LiFePO4 电池.....	27
6 参考资料	32

插图清单

图 2-1. 电池放电模式.....	4
图 2-2. 电池充电模式.....	4
图 2-3. SoC-OCV 表.....	5
图 2-4. OCV 校准和容量累积.....	6
图 2-5. 电池模型.....	6
图 2-6. VGauge 软件流程.....	7
图 2-7. 容量到 SOC.....	7
图 2-8. CusRitSoc 至 SmoothRitSoc.....	8
图 2-9. 算法概述.....	8
图 2-10. 算法性能 (通过仿真)	9
图 2-11. CT 表示例.....	10
图 3-1. 测量仪表 GUI 功能.....	11
图 3-2. MCU COM 工具功能.....	12
图 3-3. 数据分析工具.....	13
图 4-1. 脉冲放电测试用例.....	14
图 4-2. 用于获取电池模型的硬件结构.....	15
图 4-3. SmData 类型.....	15
图 4-4. 从 SMDData 生成参数文件.....	16

图 4-5. 电池电路表输入.....	16
图 4-6. 脉冲放电示例.....	17
图 4-7. Gauge_UserConfig.h 设置.....	17
图 4-8. 数据结构.....	18
图 4-9. tBattParamsConfig 结构.....	18
图 4-10. 测量仪表模式设置.....	19
图 4-11. 检测数据输入模式.....	20
图 4-12. 通信数据输入模式结构.....	20
图 4-13. McuData 类型.....	21
图 4-14. 通信数据输入步骤.....	21
图 5-1. 电池 SOC.....	22
图 5-2. MSPM0 测量仪表硬件板.....	22
图 5-3. MSPM0 测量仪表板方框图.....	23
图 5-4. 测量仪表板说明.....	23
图 5-5. MSPM0L1306 LaunchPad UART 连接.....	24
图 5-6. MSPM0 测量仪表软件工程视图.....	24
图 5-7. 电池测试用例.....	25
图 5-8. 电池测试结果.....	26
图 5-9. 读取 Q 值.....	26
图 5-10. 电流消耗.....	27
图 5-11. MSPM0G3507+BQ75952EVM 方框图.....	28
图 5-12. MSPM0 测量仪表软件工程视图.....	29
图 5-13. 电池测试用例.....	30
图 5-14. 电池电芯测试结果.....	30
图 5-15. 电池包测试结果.....	31
图 5-16. 电池测试用例.....	31
图 5-17. 电池测试结果.....	32
图 5-18. 电池测试结果.....	32

表格清单

表 1-1. MSPM0 测量仪表解决方案比较.....	3
表 2-1. IGAUGE 关键参数.....	9
表 2-2. 容量学习关键参数.....	9
表 2-3. 混合关键参数.....	10
表 4-1. 电池测试模式.....	14
表 4-2. MSPM0 L2 测量仪表 SOC-OCV 范围.....	17
表 4-3. 系统配置参数.....	18
表 4-4. 一般配置参数.....	18
表 4-5. 混合算法相关参数.....	19
表 4-6. 容量学习算法相关参数.....	19
表 4-7. VGauge 算法相关参数.....	19
表 4-8. IGauge 算法相关参数.....	19

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 引言

这里有不同的基于 MSPM0 的测量仪表解决方案。表 1-1 快速比较了各个解决方案，方便客户选择合适的解决方案。本文档重点介绍具有不同设置的 MSPM0 测量仪表 L2 解决方案，包括

- 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)

表 1-1. MSPM0 测量仪表解决方案比较

	MSPM0 测量仪表 L1	MSPM0 测量仪表 L2
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 + 空/满补偿 + 容量学习
适用应用	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
适合的电池类型	LiCO ₂ /LiMn ₂ O ₄	LiCO ₂ /LiMn ₂ O ₄ /LiFePO ₄

请注意，MSPM0 L2 测量仪表是纯软件代码。MCU 平台、AFE 或电池仅用来显示此算法的功能。其功能如下所示：

- 在 MCU 上电后工作，无需工厂校准或学习循环。
- 支持 SOC、SOH、容量和警告标志输出。
- 对电池化学参数输入的要求较低。
- 高精度和高可靠性

该解决方案由三部分组成：硬件、软件和 GUI。所有这些都可以在 <https://www.ti.com/lit/zip/slaaef5> 找到。您还可以在 SDK 下面找到典型用例的 MCU 代码 (mspm0_sdk_xxx\examples\nortos\LP_MSPM0xxx\battery_gauge)。

- 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。有关典型的应用例，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

在以下部分中，在说明软件并详细介绍算法之前，先介绍了关于电池知识的基本概念、使用的算法以及如何获得所需的 SOC。

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。此处显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我们要介绍的相关概念。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 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，因此在恒定负载下可看到 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，您可以从电池中获得不同的容量。此外，因为当温度升高时 Rcell 会降低，输出容量也受到温度的影响。在这个测量仪表解决方案中，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。

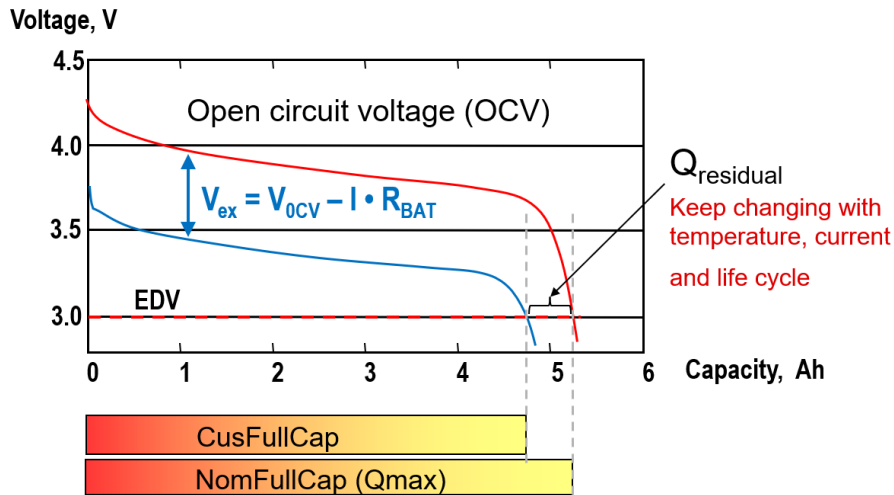


图 2-1. 电池放电模式

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，电流将以指数级减少。如果电流变为 0，则获得 NomFullCap。但是，为了避免长时间等待，实际上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。

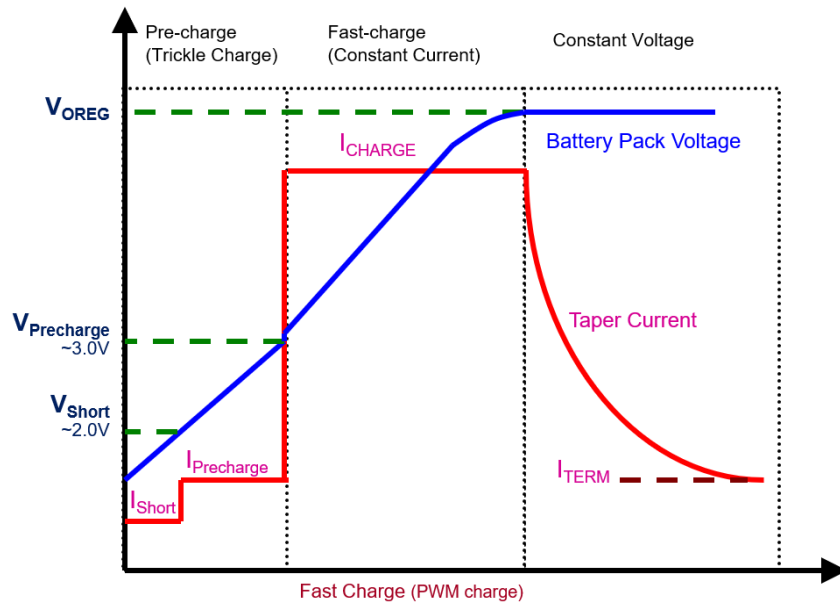


图 2-2. 电池充电模式

总之，NomFullCap 是从一个 OCV 到另一个 OCV 之间测得的，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。对于 CusFullCap 范围，它基于 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，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 和使用的公式

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。

2.2.1 NomAbsSoc 计算

如果您使用 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，下一节将对此进行讨论。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。

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。

$$Q_{use} = I(t) * \Delta t \tag{1}$$

$$NomAbsSoc = \frac{NomFullCap - Q_{use}}{NomFullCap} \tag{2}$$

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。

$$NomAbsSoc = f(OCV) \tag{3}$$

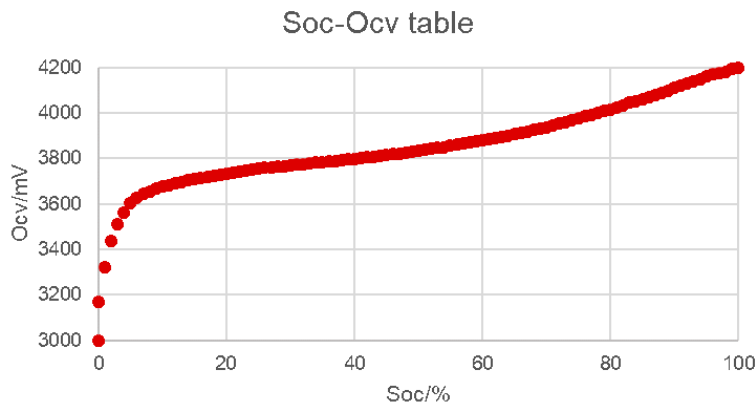


图 2-3. SoC-OCV 表

方程式 2 用于运行时输出 NomAbsSoc，图 2-1 用于定期校准 NomAbsSoc。再过两次校准之后，就可以得到差值容量和差值 NomAbsSoc。然后，可以计算 NomFullCap，如方程式 4 所示。实际上，在知道 NomFullCap 后，方程式 2 可以在可接受的精度下工作。

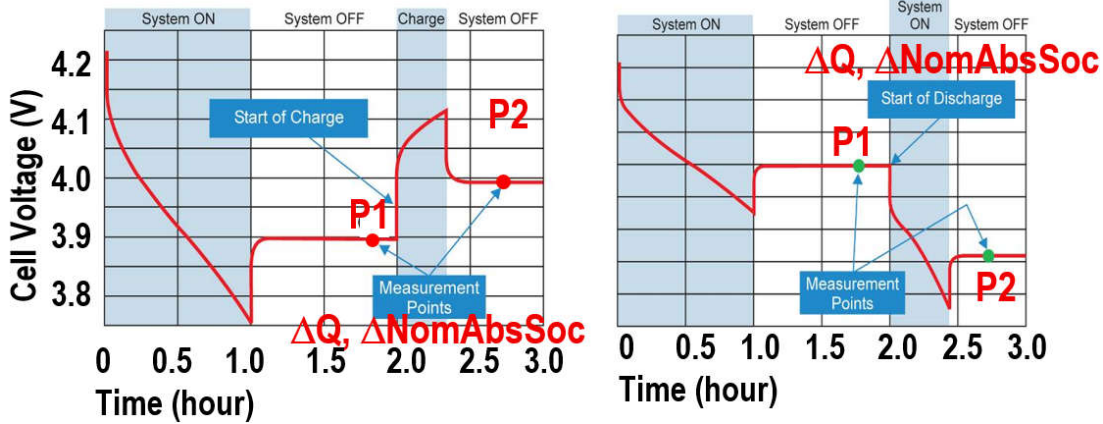


图 2-4. OCV 校准和容量累积

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

对于真实的电池，由于电池老化，其 $NomFullCap$ 会略有减少。为了跟踪容量下降问题，应定期校准 $NomFullCap$ 。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，首次获得的 $NomFullCap$ 用作最大 $NomFullCap$ 。

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

2.2.1.2 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 $CusRitSoc$ 。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 2-5 所示）来输出 $NomAbsSoc$ （仅带电压输入）。

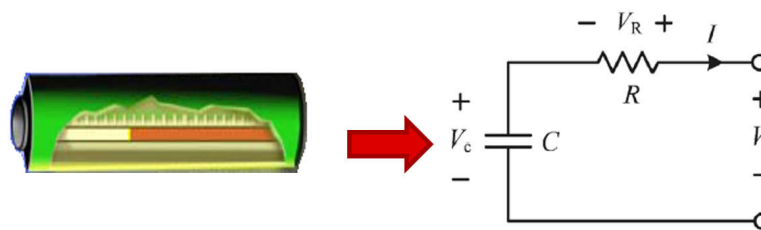


图 2-5. 电池模型

图 2-6 展示了 VGauge 的软件流程图。但是，在此解决方案中，SOC 查询表功能只需用于搜索此电池模型中的所需参数。对于 SOC 部分，它完全来自 IGauge。

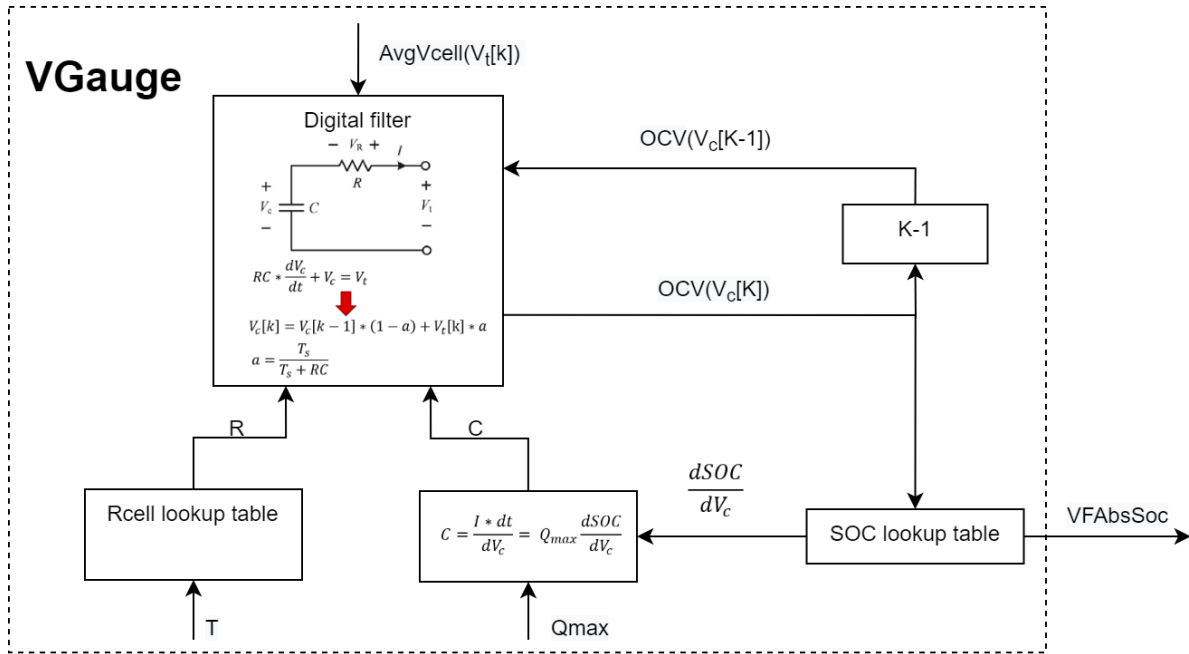


图 2-6. VGauge 软件流程

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](#)。

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。

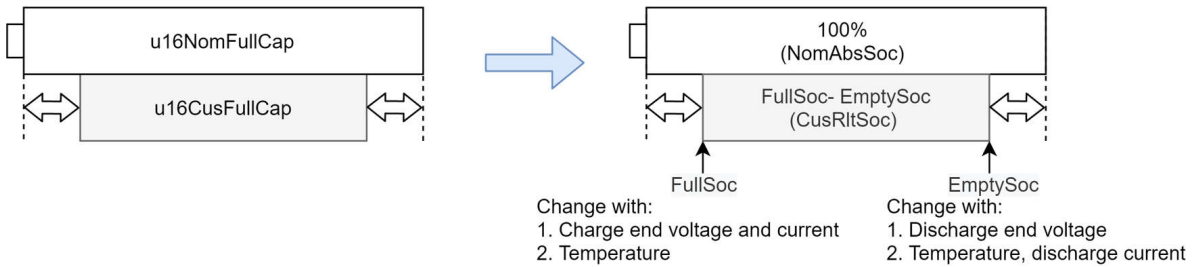


图 2-7. 容量到 SOC

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 2-7。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 用于表示相对 SOC，这是会随着电池状态而不断变化的归一化因子 (FullSoc-EmptySoc)。方程式 6 展示了用于计算 CusRltSoc 的公式。

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

在不同的条件下，获取 CusRltSoc 的关键点是获取 FullSoc 和 EmptySoc。在此测量仪表算法中，在不同电池条件下记录 FullSoc 和 EmptySoc 并更新到表格中。有关此部分的信息，请参阅节 2.3.4。

2.2.3 SmoothRltSoc 计算

图 2-8 中的蓝色线显示了电池充电和放电测试模式下的 CusRltSoc 反应。您可以看到，OCV 校准或 EmptySoc 变化导致出现一些数据跳变。

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。

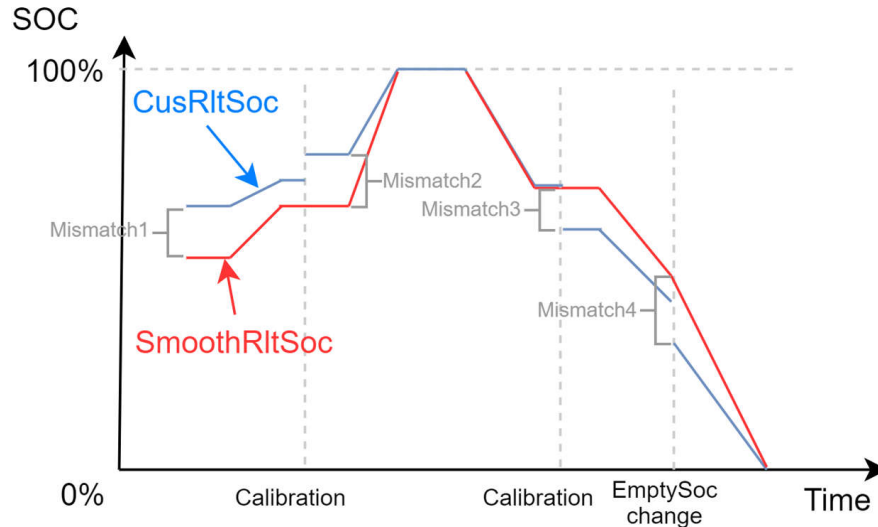


图 2-8. CusRltSoc 至 SmoothRltSoc

2.3 算法概述

本节概述了所介绍的测量仪表算法，如图 2-9 所示。这仅适用于电池电芯算法。对于电池包算法而言，只是电池电芯算法的组合。

此算法基于库仑计，可与前面介绍的其他方法搭配来解决其限制。它由四部分组成。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。

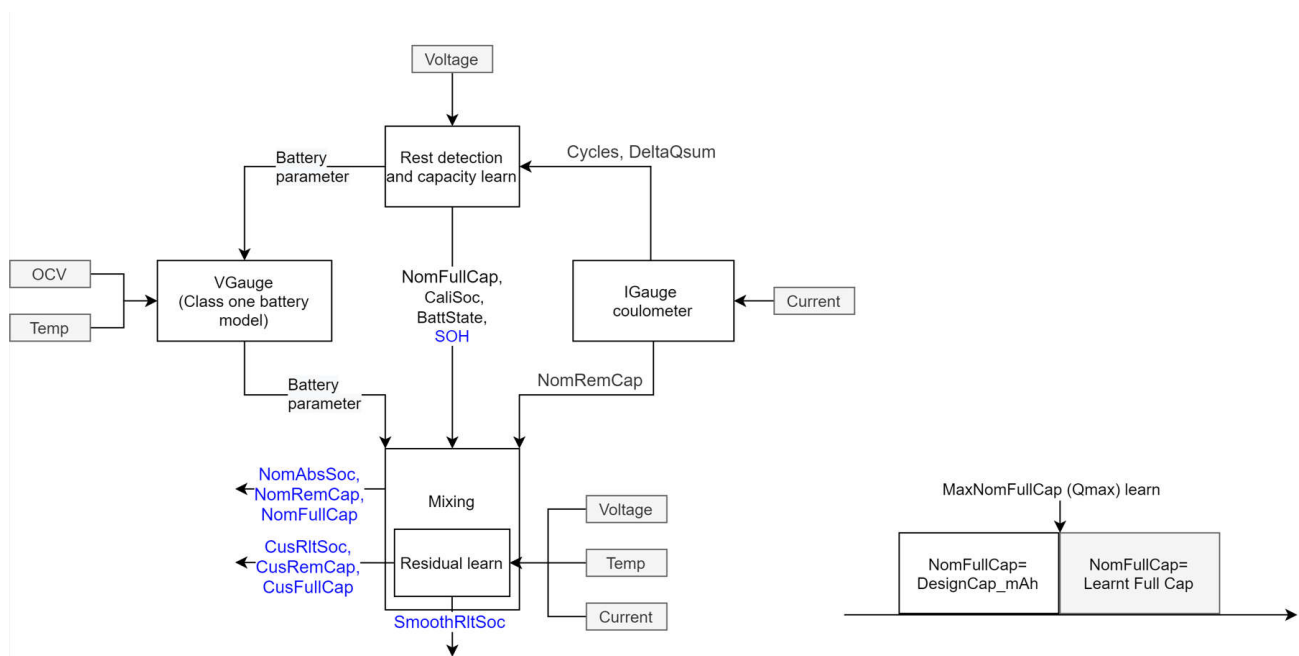


图 2-9. 算法概述

图 2-10 展示了基于 LiFePO4 电池仿真模型的算法性能，具有 3000 个随机 DHG/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。

请记住，结果只用于显示在有限条件下检测 NomAbsSoc 的算法能力，而不能在实际应用中检测 CusRltSoc 时保证算法的误差范围。

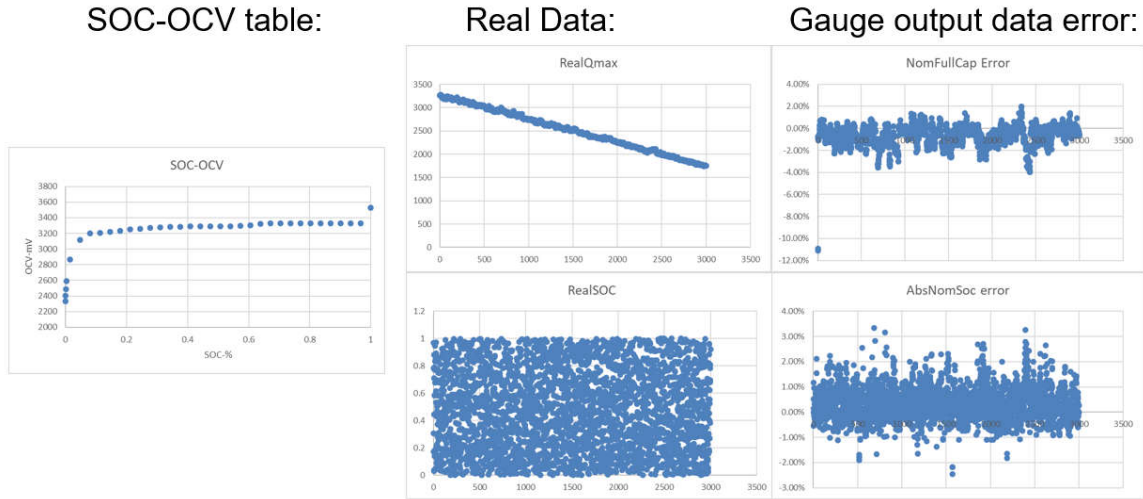


图 2-10. 算法性能 (通过仿真)

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

对于 VGauge，它仅用于其他功能方框图，以便从保存的电池模型中搜索电池参数：“circuitParamsTable”。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

电流监测计就是一个简单的库仑计。它会累积用于不同功能的不同类型的容量。

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。
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。它还将获得 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。
- 第三是使用节 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。电流监测计部分也使用它在同一周期中重新计算 NomRemCap。

表 2-2. 容量学习关键参数 (续)

参数	注释
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。电流监测计部分也使用它来计算 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。

对于 EmptySoc, 给出了一个电流温度表 “AbsEmptySocMatrix”, 以便模拟电流和温度对 EmptySoc 的影响。图 2-11 展示了一个表示例。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。

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。为了进行简化, 获取的最大 EmptySoc 用于表示总块, 记录的 EmptySoc 和真实 EmptySoc 之间的差异会导致电池可放电容量缩减, 尤其是在最坏情况下。为了减少容量衰减, 客户可以将 CT 表调整得更加精细。支持的最大温度阈值数为 6。最大电流阈值数为 4。

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 2-11. CT 表示例

对于 FullSoc, 在通电后, 它使用 MaxFullChgVoltThd 作为 OCV 来首先获得默认 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。

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。但是, 如果 CusRltSoc 突然变为 0% 或 100%, SmoothRltSoc 会同时变为 0% 和 100%。此外, CusRltSoc 会高于 0% 和 100%。SmoothRltSoc 限制在 0% 至 100% 之间。

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。如果不进行初始化, 它会自主学习。

表 2-3. 混合关键参数 (续)

参数	注释
iq15AbsFullSocMatrix	在不同温度下记录不同 fullSoc 的矩阵，并且它会自动学习。

3 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，如图 3-1 所示。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 有三页。

- 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。
- 第三页是数据分析工具，用于生成电池参数并评估 SOC 误差。

请注意，如果您使用 GUI 执行文件来评估解决方案，由于 Python 语言限制，GUI 启动时间会有点长，大约需要 2-3 分钟。相反，使用 GUI 源文件将具有更快的 GUI 启动速度。

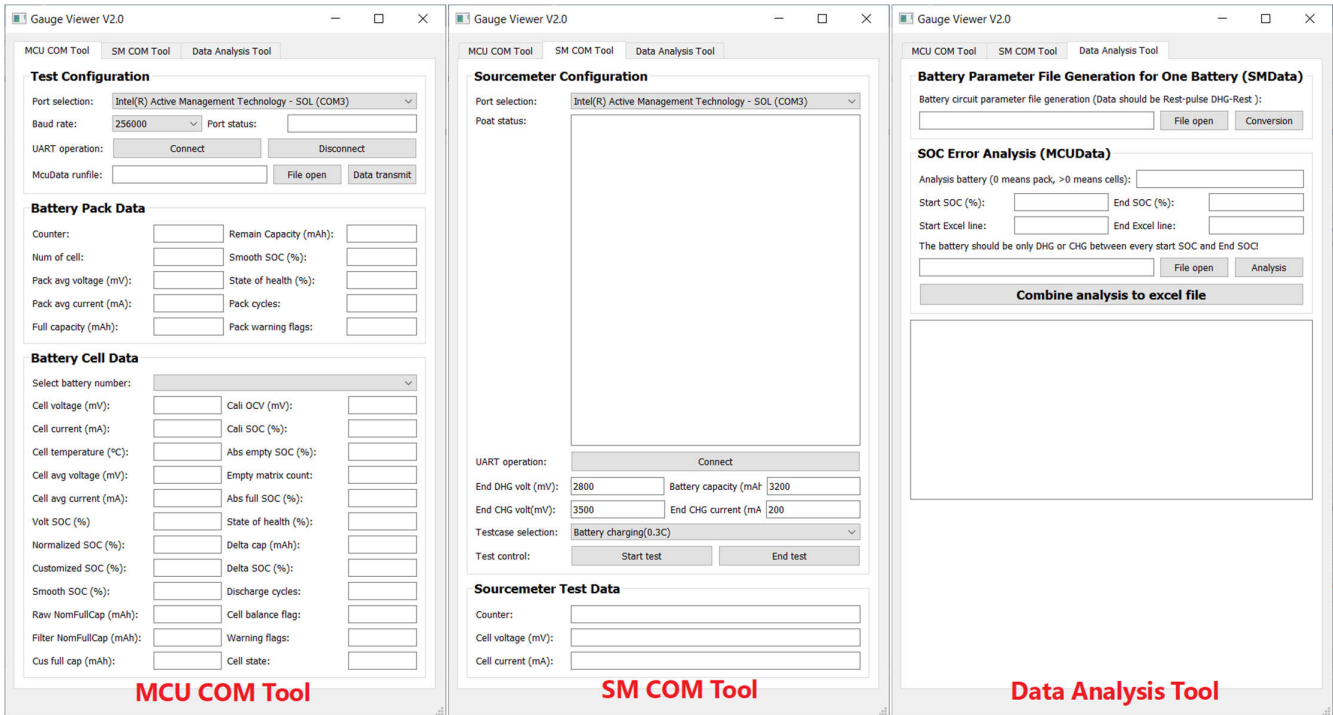


图 3-1. 测量仪表 GUI 功能

提供了一个测量仪表 GUI 执行文件。第一次打开的时间会很长，大约 5 至 10 分钟。您可以使用它进行评估，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，请参阅后续章节。

3.1 MCU COM 工具

最终用户普遍使用 MCU COM 工具 (如图 3-2 所示)。它有两个功能：

- 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。如果单元格编号大于 1，则可以按照图 3-2 中的步骤选择所需的编号。在测试完成或停止测试后，数据将自动保存到名为“time-McuData.xlsx”的 Excel 中。
- 第二个功能是加载所选的“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) 搭配用于运行算法。

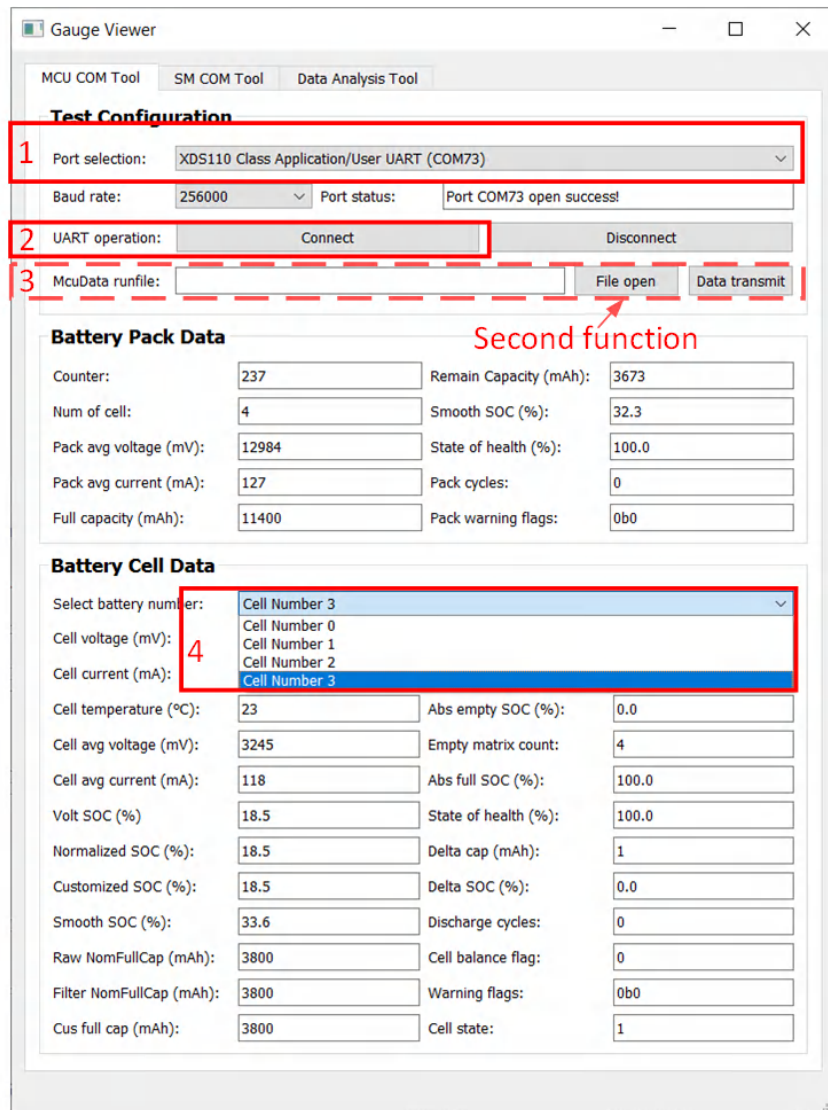


图 3-2. MCU COM 工具功能

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。如果要重新创建软件的这个部分, 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。

由于源表不是转为 BMS 制造的设备, 请注意它在 BMS 测试方面有局限性。例如, 它无法支持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一些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。它用于从脉冲放电数据 (SMDData 文件) 中提取电池电路信息。按照步骤操作, 您将获得电池参数文件, 有关详细信息, 请参阅节 4.2.1。

第二个功能是分析测量仪表精度性能。在真实测试后, 生成 MCUDData 文件。测量仪表在 Excel 行的开头输出 SOC, Excel 行的结尾是先验值。您输入的起始 SOC 和结束 SOC 是后验值。按下分析按钮后, 首先通过起始 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。这意味着您还需要在步骤 2 和步骤 3 中进行相同次数的输入操作。最后，您可以点击步骤 4，然后点击 GUI 输出最终结果。

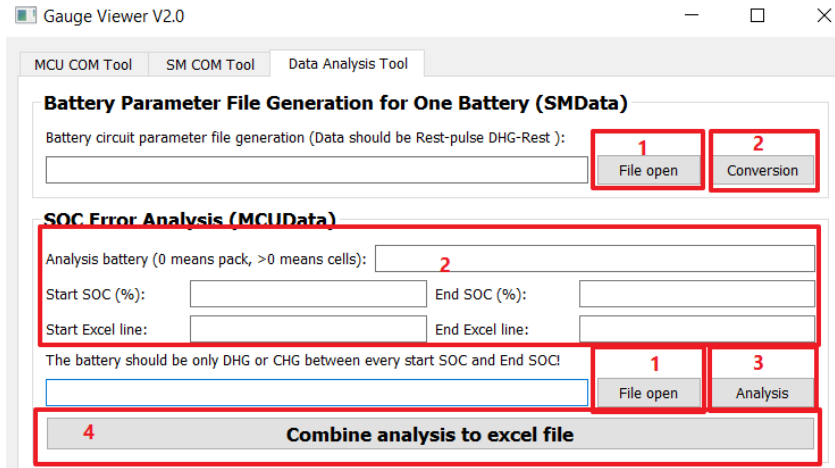


图 3-3. 数据分析工具

4 MSPM0 测量仪表评估步骤

4.1 第 1 步：硬件准备

硬件版：

如果要评估此整体解决方案，需要先进行硬件设置。如果您只想评估测量仪表软件，只需要一个 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。

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 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。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 SOC 校准精度取决于电池模型，所以强烈建议获取专用电池模型。

4.2.1 电池测试模式

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。每个脉冲后的静置时间应为 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 个数据点。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

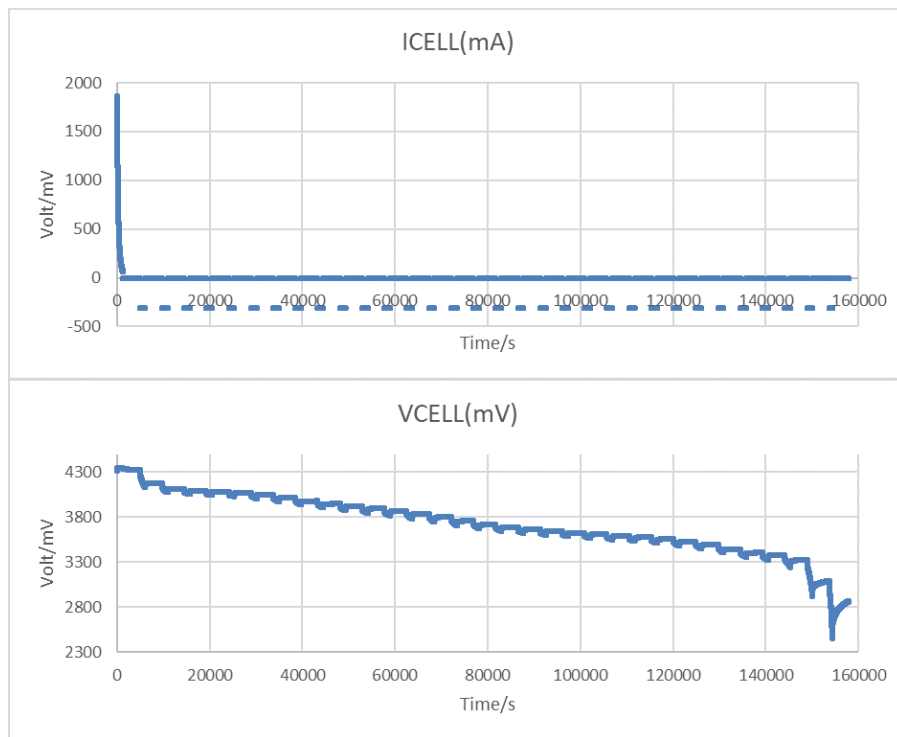
在进行电池测试时，应将 PCB 和电池插座的影响考虑在内。否则，测试的电阻器小于实际电路电阻器。

表 4-1 展示了 LiCO₂/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	~25°C	
启动电压 (OCV)	4.3V 至 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 至 3.0V	确保剩余电压 (OCV) 不高于应用的最小放电电压
放电电流	0.05C 至 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	10 分钟至 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，进行 20 分钟的脉冲放电，并将电池静置 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。启动电压为 4322mV，终止电压为 2864mV，而不是 2450mV！


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。

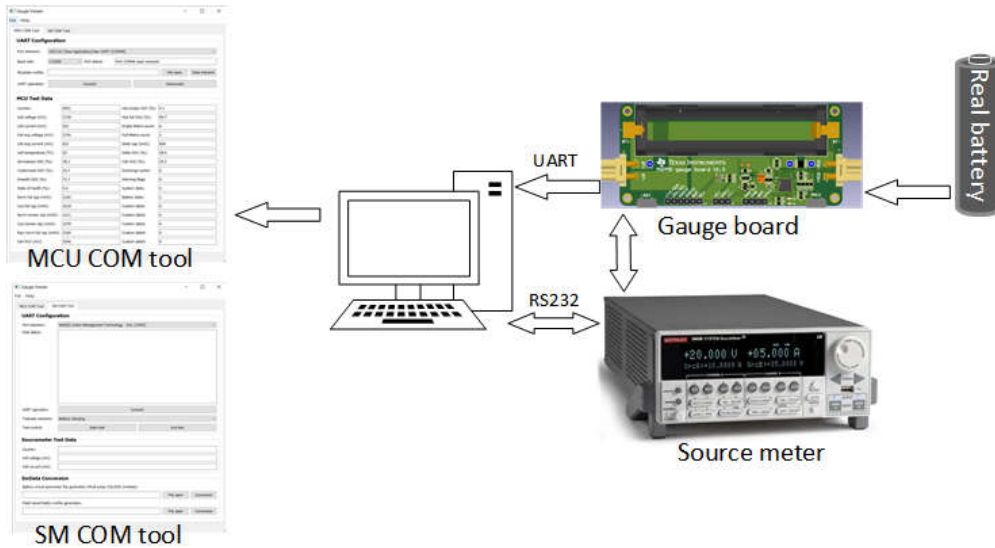


图 4-2. 用于获取电池模型的硬件结构

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.xlsx”结尾。

Time stamp	VCELL(mV)	ICELL(mA)
0	4200.3	1570.7
1	4200.3	1553.6
2	4200.3	1540.2
3	4200.3	1529.8
4	4200.3	1521.1
5	4200.3	1513.4
6	4200.3	1506.5
7	4200.3	1500.4
8	4200.3	1494.7
9	4200.3	1489.4
10	4200.3	1484.1

图 4-3. SmData 类型

4.2.2 电池模型生成

获得 SMDData 格式或 MCUData 格式的电池运行数据（名称也应该遵循它们的命名格式）后，即可以按照图 4-4 中的步骤，使用“Battery Parameter File Generation for One Battery”获取 Excel 和文本格式的电池模型（电池电路文件）。

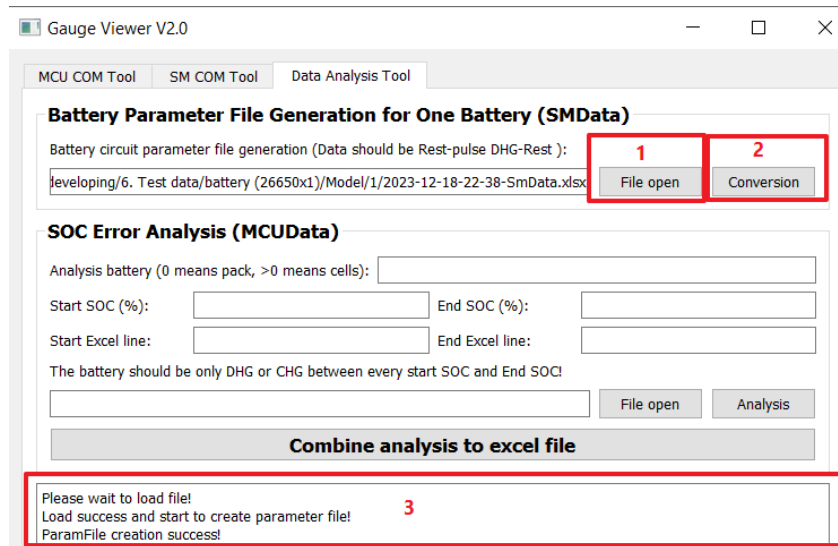


图 4-4. 从 SMDData 生成参数文件

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。然后您可以完成电池电路表输入。

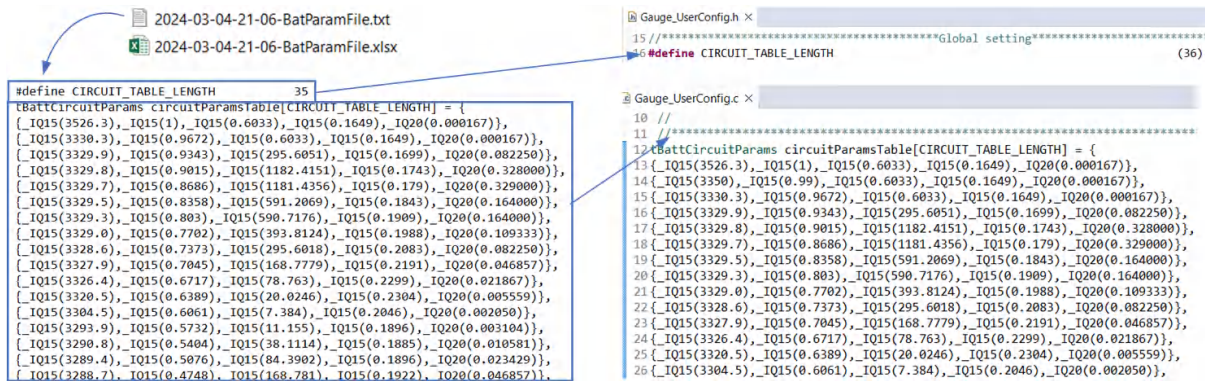


图 4-5. 电池电路表输入

电池电路表（电池模型）的元件有五种组合：

- 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。

对于 OCV 和 SOC，它会影响容量学习部分中使用的 SOC 校准。对于 Rcell 和电容系数，它将主要影响 VFabsSoc 精度。

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。测试后同时使用 Qmax 获得 SOC，使用方程式 2。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，它的值等于 dOcv(mV)/Current(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。

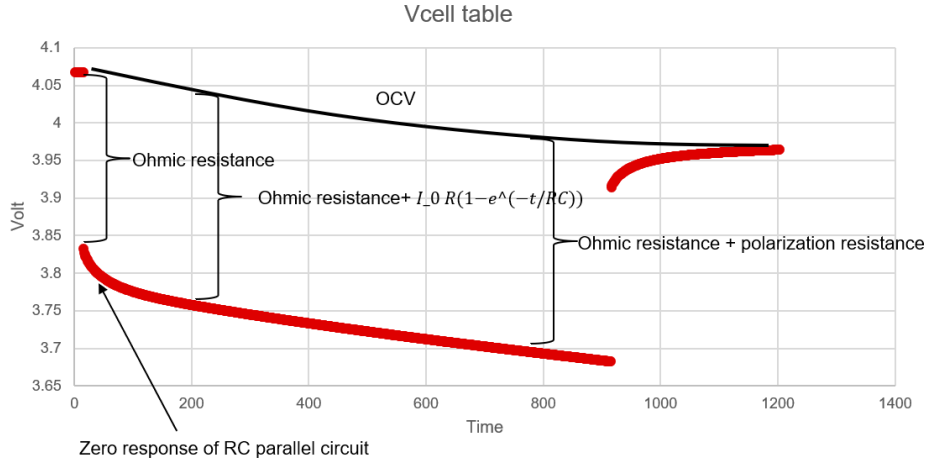


图 4-6. 脉冲放电示例

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 4-2 展示了不同放电条件的示例。请注意 OCV 与最大充电/放电电压之间的差异。

表 4-2. MSPM0 L2 测量仪表 SOC-OCV 范围

	电流 <0.01C	电流 <0.1C	电流 <0.5C
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 第 3 步：输入自定义配置

首先，您需要根据“Gauge_UserConfig.h”中的应用程序进行一些系统更改。图 4-7 展示了要更改的常见参数。

```

//*****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)

//*****Global setting*****//
#define CIRCUIT_TABLE_LENGTH (36) //OCV-SOC-Rcell table
#define CELL_NUMBER (4)
    
```

图 4-7. Gauge_UserConfig.h 设置

表 4-3 展示了这些参数的说明。

表 4-3. 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。
CELL_NUMBER	电池包的电芯数。
CIRCUIT_TABLE_LENGTH	circuitParamsTable 长度。

其次，您需要完成“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解决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都保存在“tBattPackParams”中。

“battGlobalParams_xx”代表电池包中的每个电池电芯。所有算法数据结构都在它下面。您需要创建与电池编号相同的“BattGlobalParams_xx”结构，并通过“battGlobalParamsArray”将其挂在“tGaugeApplication”下面。

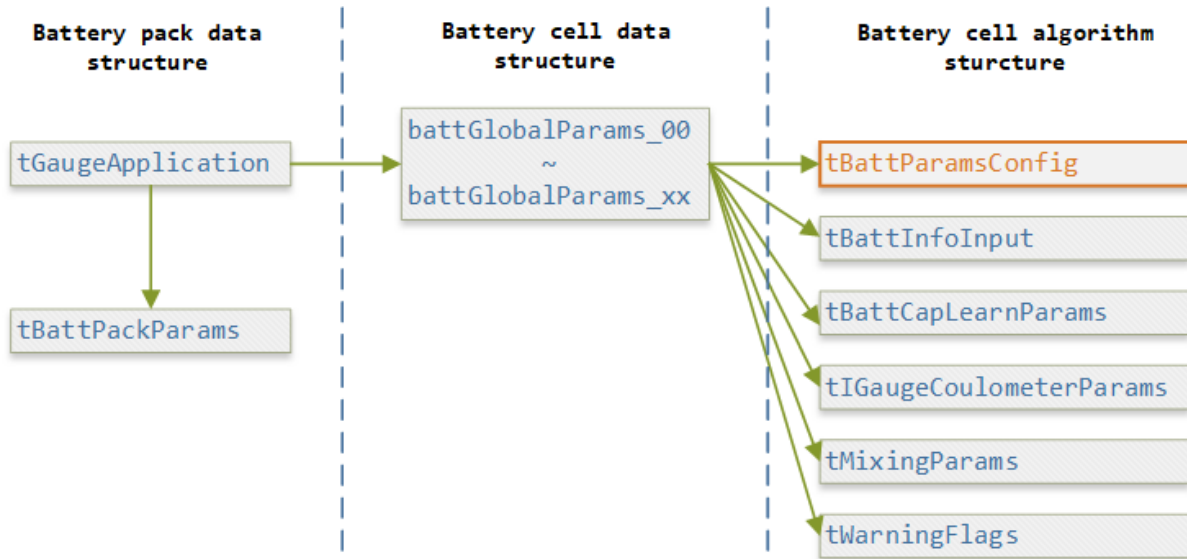


图 4-8. 数据结构

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。

```

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，您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 4-4. 一般配置参数

参数	注释
u16DesignCap_mAh	设计容量。只需输入电池的标准容量或通过电池参数生成测试进行测试的容量。

表 4-4. 一般配置参数 (续)

参数	注释
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。
u16MinFullChgVoltThd_mV u16MaxFullChgVoltThd_mV / i16FullChgCurtThd_mA	电池充满相关设置。电池充电电压应该在这个范围内。 u16MinFullChgVoltThd_mV 有助于判断电池电量何时充满。 u16MaxFullChgVoltThd_mV 将用作 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，我们将视为电池已充满。
u16EmptyDhgVoltThd_mV	当电压达到该值时，我们会将电池视为电量耗尽。
u8AvgBattParamsUpdateCount	在稳定周期后获得平均数据。

表 4-5. 混合算法相关参数

参数	注释
i8TempThdx_Ci16CurtThdx_mA	iq15AbsEmptySocMatrixInput[]、iq15AbsEmptySocMatrix[]、iq15AbsFullSocMatrix[] 的电流和温度阈值。仅对 emptySoc 提供输入。您可以根据节 2.3.4 使用所需的分频范围。
iq15AbsEmptySocMatrixInput[]	经过预先测试的 EmptySoc 矩阵输入。如果全部为 0，则使用 u16EmptyDhgVoltThd_mV 作为空 OCV 来计算相关的 EmptySoc。然后，它将在循环后自动学习。

表 4-6. 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，电池将被视为静置。 务必考虑电流检测的噪声。否则静置检测会有问题。
u8SOHCalcCycleThd	电池放电周期阈值，用以计算 SOH。
iq15DefaultSOH_DEC	电池默认 SOH 值。

表 4-7. 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。它根据经验进行，不会对性能产生太大影响。

表 4-8. IGauge 算法相关参数

参数	注释
i16AvgLeckageCurt_mA	内部漏电流补偿。
i16PassiveDhgCurt_mA	这是为无源放电电流状况而保留。

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。

```
Gauge_UserConfig.h
7 //*****Algorithm detection mode selection*****
8 #define DETECTION_MODE (COMMUNICATION_DATA_INPUT)
9 //define DETECTION_MODE (DETECTION_DATA_INPUT)
10
11 //*****Algorithm communication selection*****
12 //define OUTPUT_MODE (NO_OUTPUT)
13 #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/发送器 (UART) 实现数据输出。然后，您可以通过 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。对于检测数据输入模式，这是常用模式。所有算法输入均采用 AFE 测试的真实数据。对于通信数据输入模式，其输入数据来自 GUI，一个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，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，需要将测量仪表代码下载到 LaunchPad。

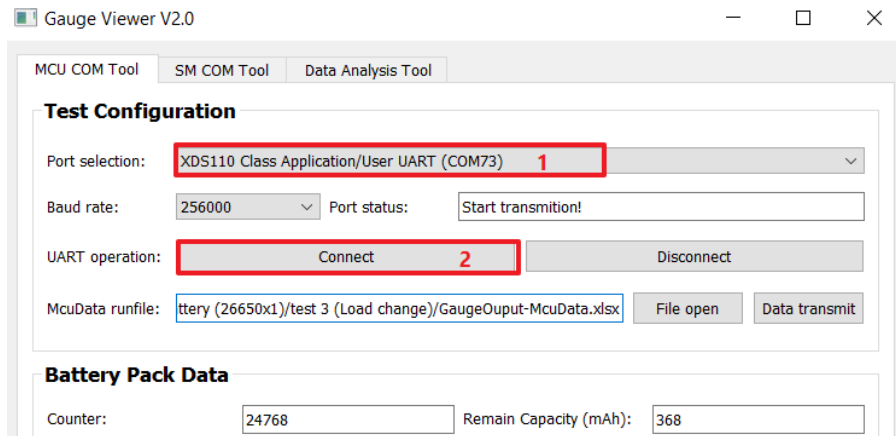


图 4-11. 检测数据输入模式

对于 GUI 控制，您只需选择名为 XDS110 Class Application/User UART 的正确端口，然后点击连接按钮。如果 MCU 已经正常工作，您将在 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 Excel 格式保存在 GUI 的相同地址下面，其名称为“YYYY-MM-DD-HH-MM-McuData.xlsx”。

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。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。

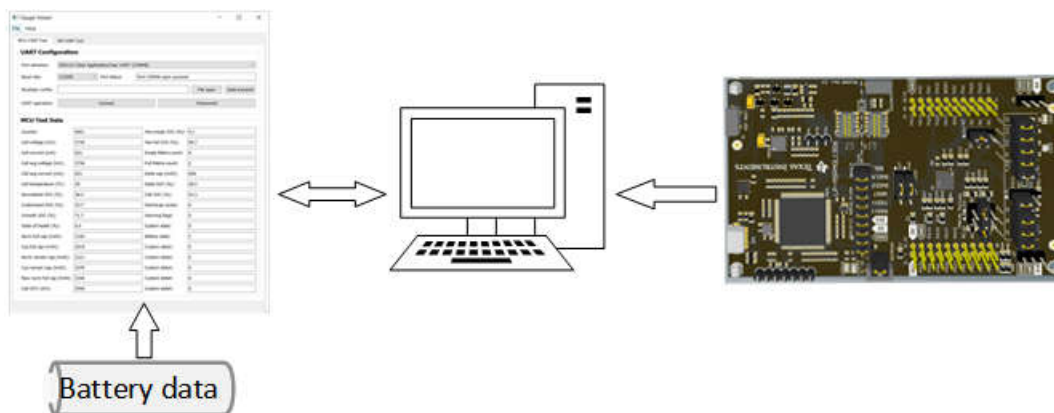


图 4-12. 通信数据输入模式结构

1. 首先，要实现此方法，您只需要使用 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。

- 第三，您需要有一个 **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。您需要在 **B** 列中输入电芯编号。然后将每个电池的 **Vcell(mV)**、**Icell(mA)** 和 **Tcell(°C)** 输入到同一个 **McuData** 文件中的相同列。为此，您可以先生成一个 **McuData** 文件，并参考该文件进行传输。最后，将文件命名为以 **“-McuData.xlsx”** 结尾。

	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

- 第四，按照图 4-14 连接 UART COM 端口，并通过点击打开文件按钮在 MCU COM 工具中加载 **MCuData** 运行文件。点击数据传送按钮后，您需要等待端口状态变为 **“Start transmission!”**。如果文件非常大，数据加载时间和 **Excel** 保存时间将会很长。此时间将为 **5~10** 分钟。

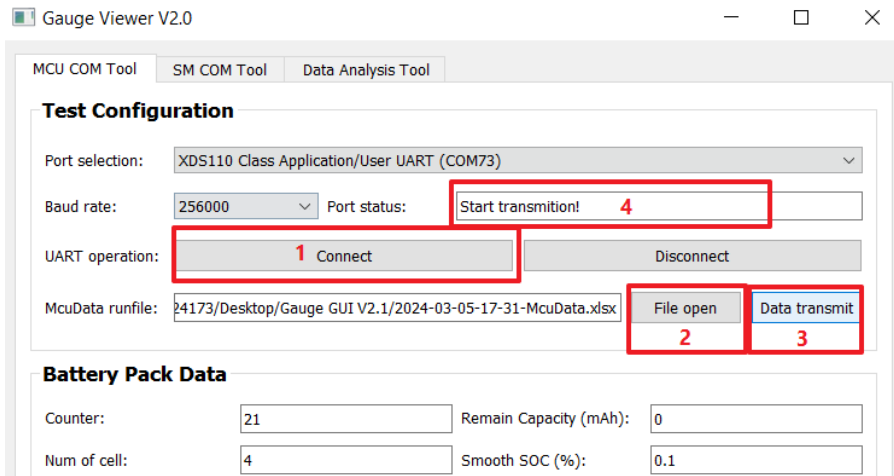


图 4-14. 通信数据输入步骤

您将从 **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。

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

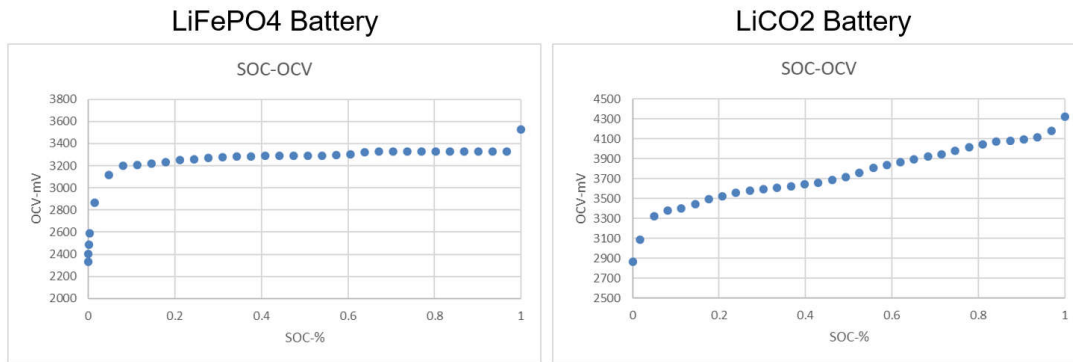


图 5-1. 电池 SOC

还展示了 MSPM0 硬件特性，因此它可以适用于 BMS 解决方案，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 + 1 节 LiCO2 电池

解决方案特性：

- 总体解决方案需要大约 15K 的闪存和 1.3K 的 SRAM。
- 没有通用异步接收器/发送器 (UART) 通信 (NO_OUTPUT 模式) 时的电流消耗约为 9uA。

解决方案优势：

- 具有内部模拟外设的纯单芯片解决方案
- 误差约为 1% 的可自校准电流检测
- 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电量监测计解决方案。



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。该解决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。

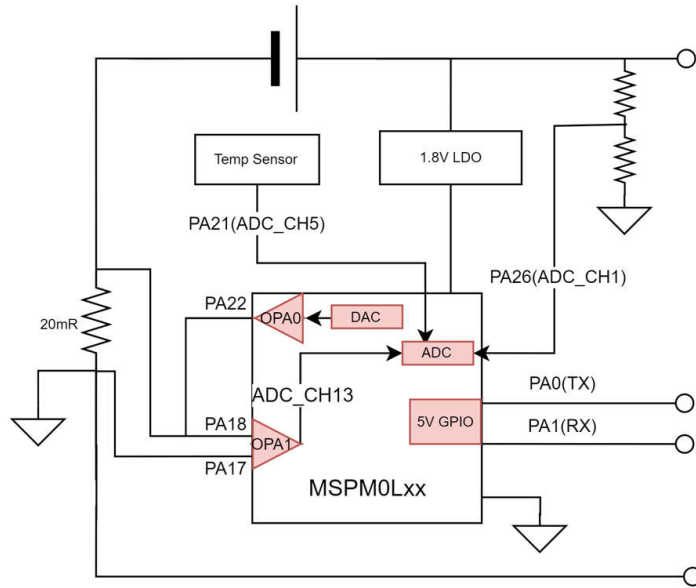


图 5-3. MSPM0 测量仪表板方框图

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 $\pm 0.25\%$ 。有关硬件介绍及其性能的更多信息，请参阅 [基于 MSPM0 的自校准电流检测解决方案](#)。

图 5-4 中显示了测量仪表板说明。注意 MCU 电源开关供电跳线。若要下载，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。若要进行评估，请将 VMCU 连接到 Vin，然后为 MCU 提供 1.8V LDO。这样可以确保出色的模拟性能。此外，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，电流应为 0。否则会有恒定的电流失调。

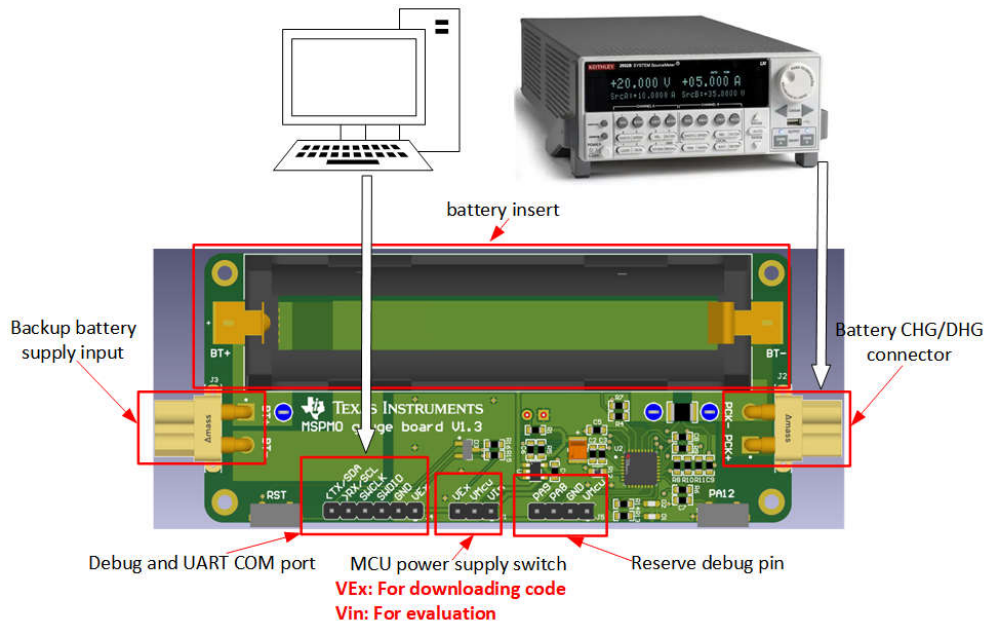


图 5-4. 测量仪表板说明

如果在通信数据输入模式下使用 MSPM0L1306 Launchpad，除了进行软件更改外，还需要按如下方式连接 UART 引脚。

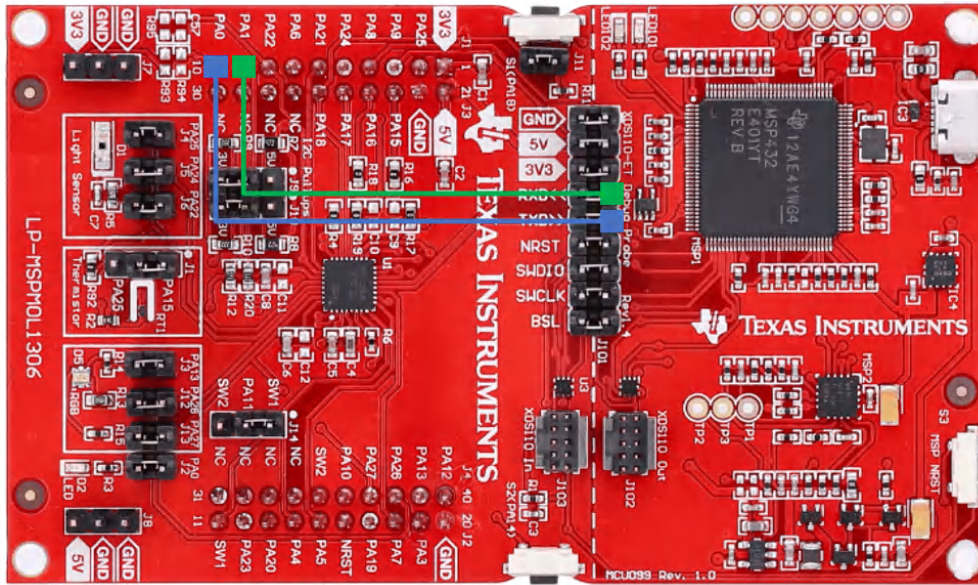


图 5-5. MSPM0L1306 LaunchPad UART 连接

5.1.2 软件和评估简介

在开始软件开发和评估之前，建议先参阅 [MSPM0 设计流程指南](#)，以便对 MSPM0 生态系统有一个基本的了解。它将帮助您快速跟上 MSPM0 的发展进程。

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 工程的其他文件都是相同的。

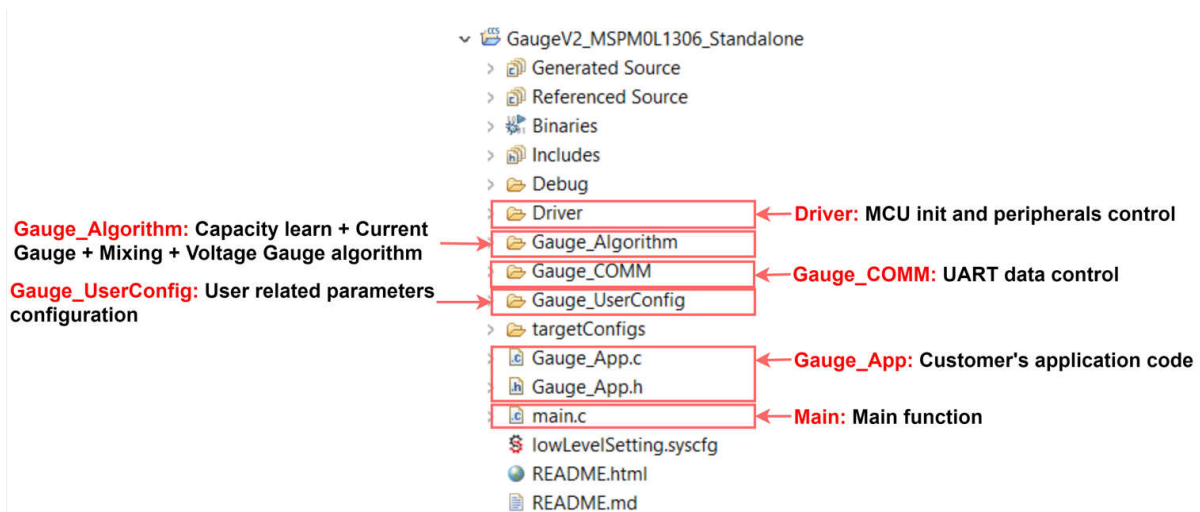


图 5-6. MSPM0 测量仪表软件工程视图

对于 Gauge_UserConfig 部分，您可以在 [节 4.3](#) 中找到说明。[节 2](#) 简要介绍了 Gauge_Algorithm 部分。Driver 部分包括所有与 MCU 相关的控制。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 部分包括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。Main 部分包括最高的系统功能代码。

记得按照 [节 4](#) 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
- 根据您的自己的电池电芯来更新 `battGlobalParams_xx` 和 `battGlobalParamsArray`。
- 更新检测模式、通信模式、电路表长度和电芯编号

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。

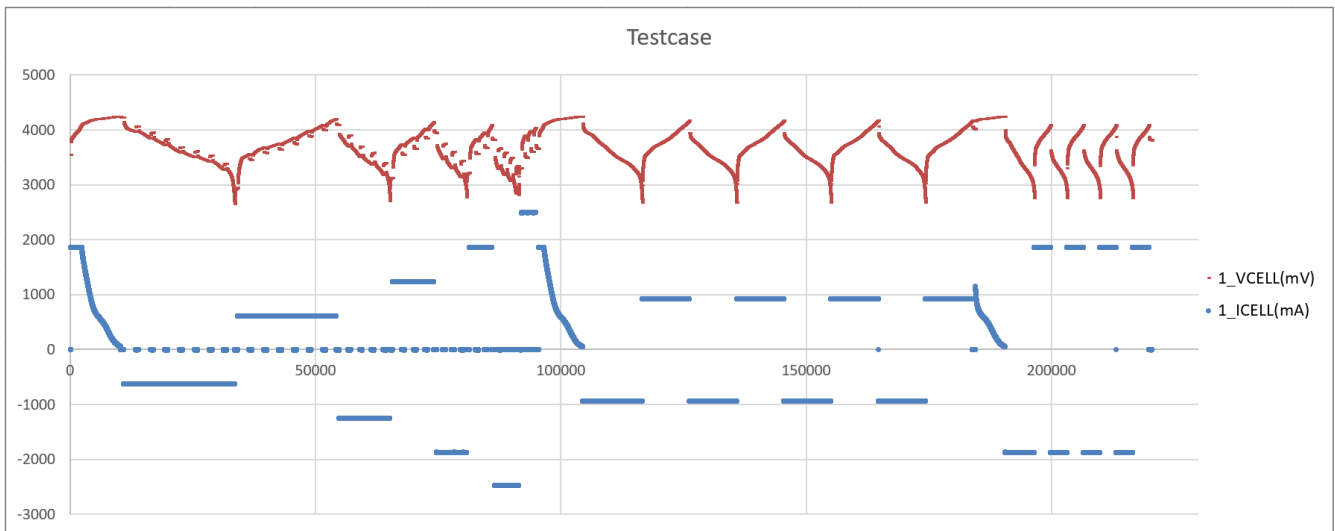


图 5-7. 电池测试用例

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。

您会发现，在放电结束时 CusSOC 有一些差异，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 将变得越来越准确。在 MaxNomFullCap 从 0 变为某个值后，意味着输出 NomFullCap 具有可接受的精度。

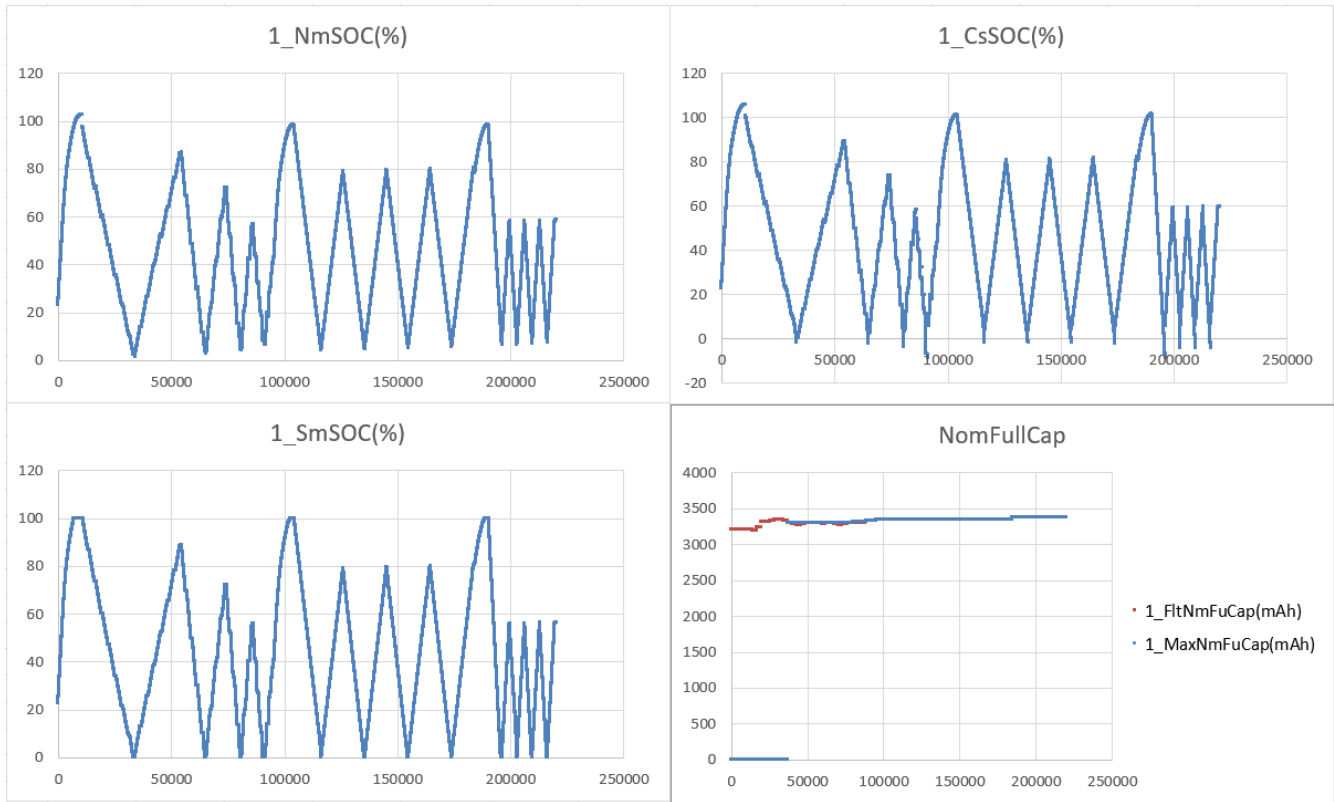


图 5-8. 电池测试结果

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。

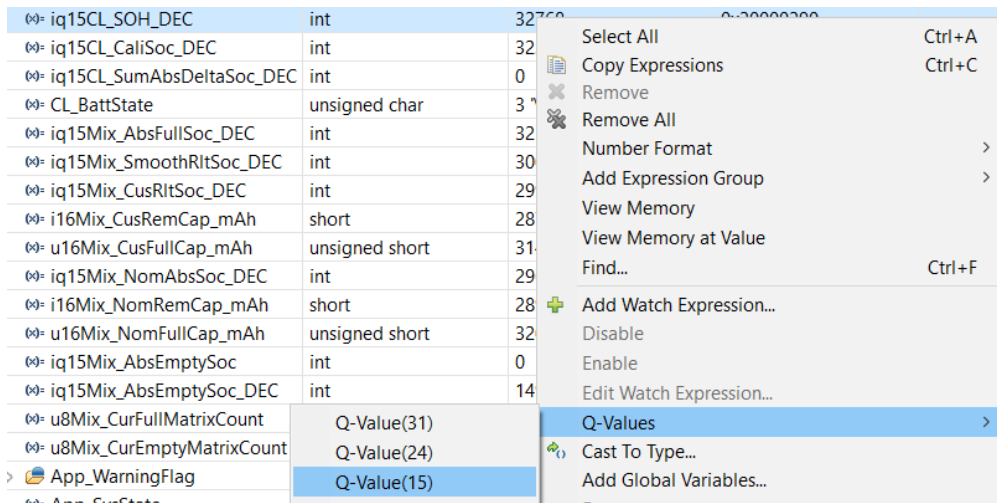


图 5-9. 读取 Q 值

5.1.3.2 电流消耗测试

因为 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。基于测量仪表板测试的电流会有点高。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。

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 + 4 节 LiFePO4 电池

解决方案特性：

- 总体解决方案需要大约 15K 的闪存和 2.9K 的 SRAM

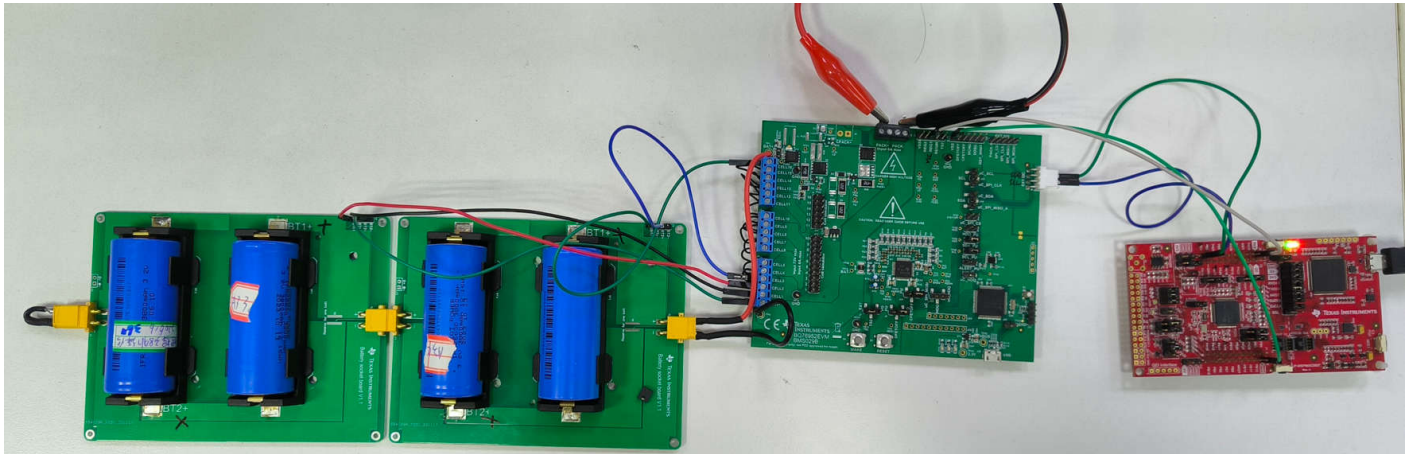
解决方案优势：

- 集成了 CAN 的高性能 MCU
- TI 的 BMS 组合解决方案
- 高性能测量仪表算法

5.2.1 硬件设置介绍

硬件板基于 MSPM0G3507 Launchpad 和 BQ76952EVM 构建。

MSPM0G3507 LP + BQ76952EVM



有关硬件连接的信息，请参阅图 5-11。您只需在 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。

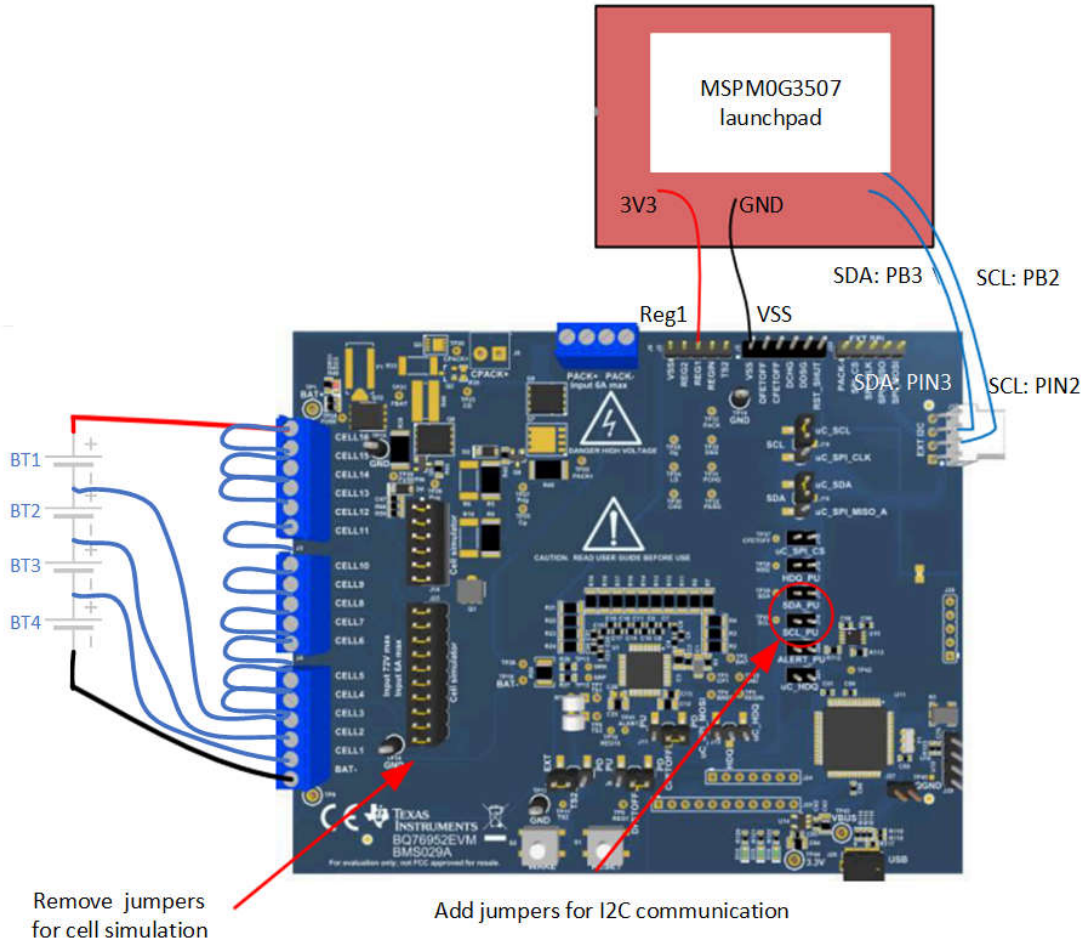


图 5-11. MSPM0G3507+BQ75952EVM 方框图

如果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，建议先参阅 [MSPM0 设计流程指南](#)，以便对 MSPM0 生态系统有一个基本的了解。它将帮助您快速跟上 MSPM0 的发展进程。

图 5-12 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 工程的其他文件都是相同的。

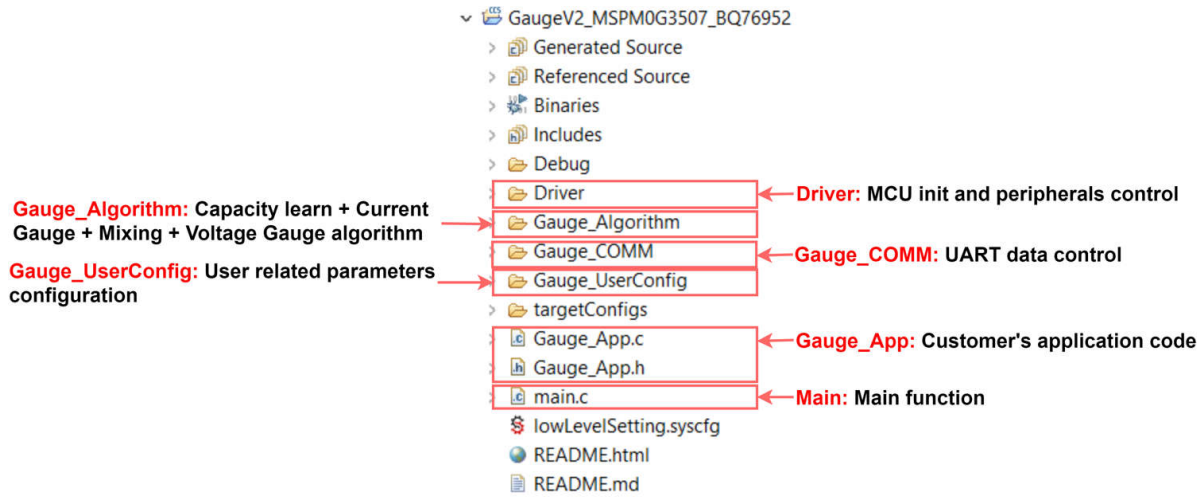


图 5-12. MSPM0 测量仪表软件工程视图

对于 Gauge_UserConfig 部分，您可以在 节 4.3 中找到说明。节 2 简要介绍了 Gauge_Algorithm 部分。Driver 部分包括所有与 MCU 相关的控制。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 部分包括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。Main 部分包括最高的系统功能代码。

记得按照 节 4 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
- 根据您的自己的电池电芯来更新 battGlobalParams_xx 和 battGlobalParamsArray。在此解决方案中，共有四个电芯。
- 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 5-13 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。

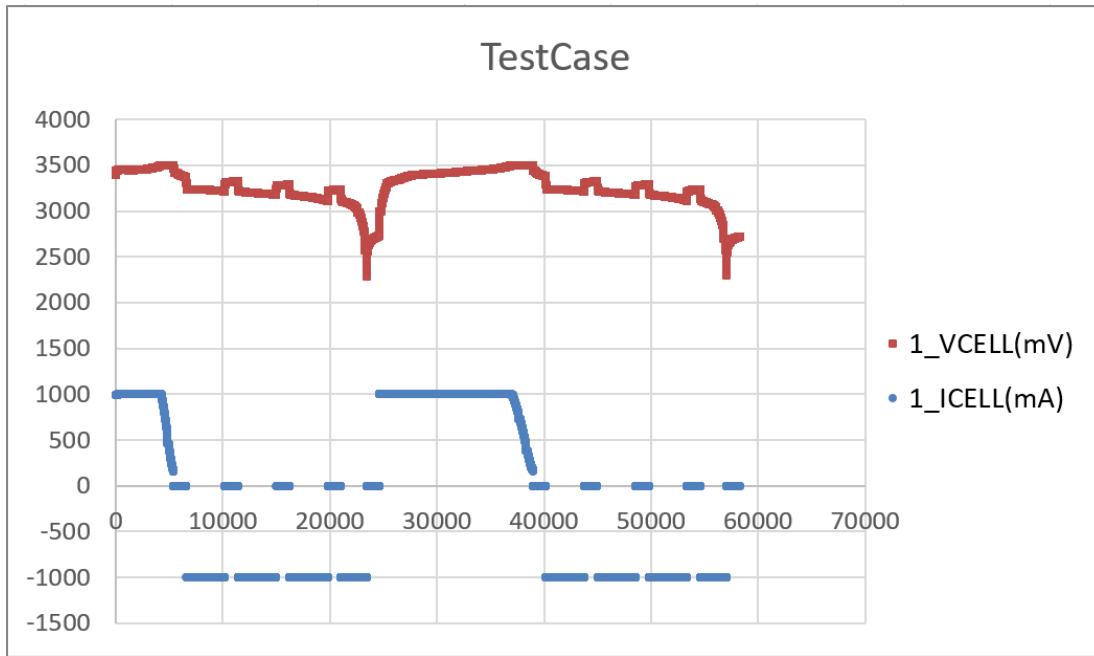


图 5-13. 电池测试用例

在开始时，您会发现 NomSOC、CusSOC 和 SmoothSOC 存在明显的差距。请参阅图 5-13 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，您可以发现，即使在第一个放电周期内，NomSOC 的变化也与实际情况非常吻合。但是，由于电流负载仅为 1A，在 CusSOC 中看不到 EmptySOC 有多大的影响。

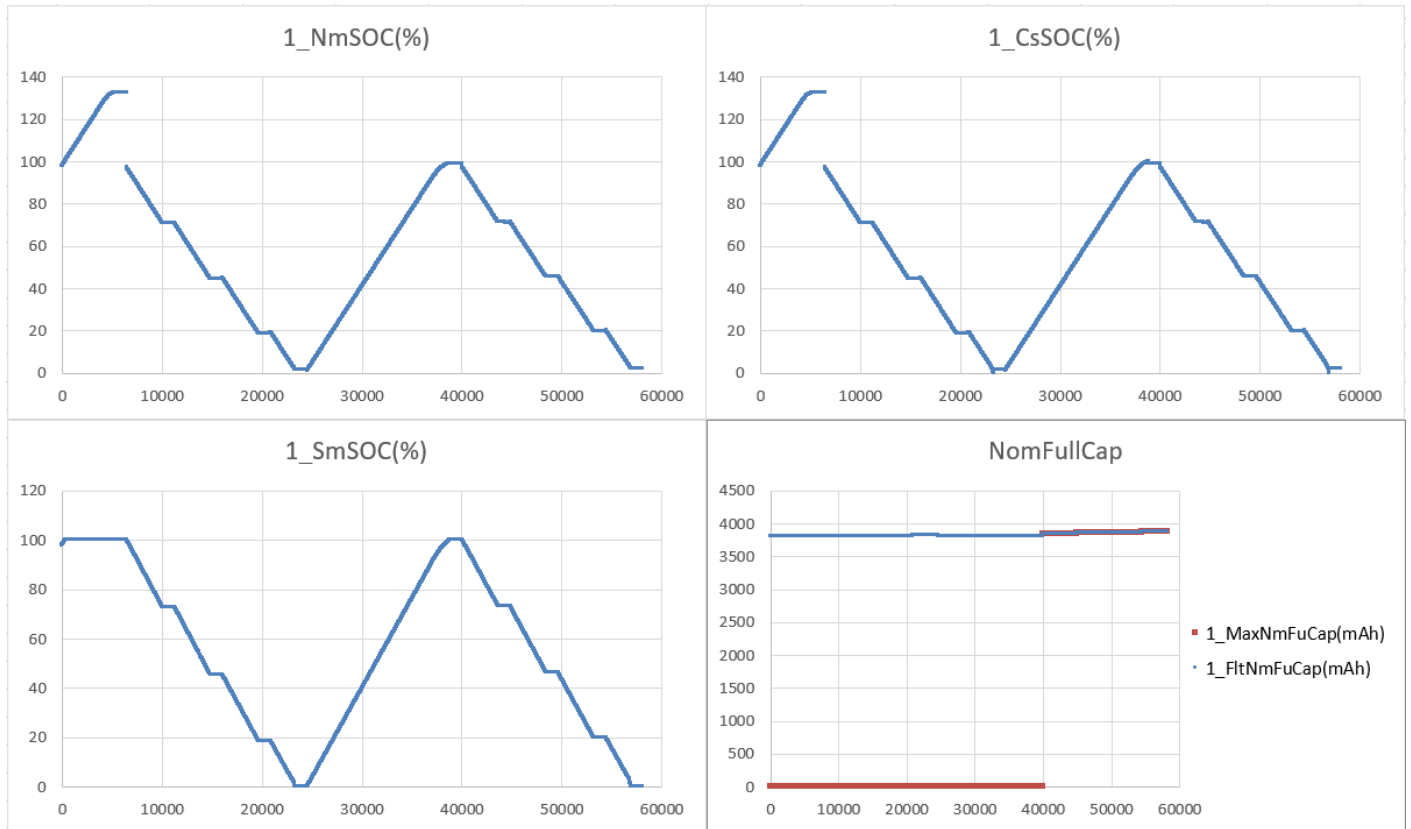


图 5-14. 电池电芯测试结果

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。对于 PackFullCap，它是所有 CusFullCap 的组合，受 EmptySOC 和 FullSOC 的影响。所以您会在 PackFullCap 图表中看到数据跳跃。此外，PackRemCap 的情况也一样，它是所有 CusRemCap 的组合。

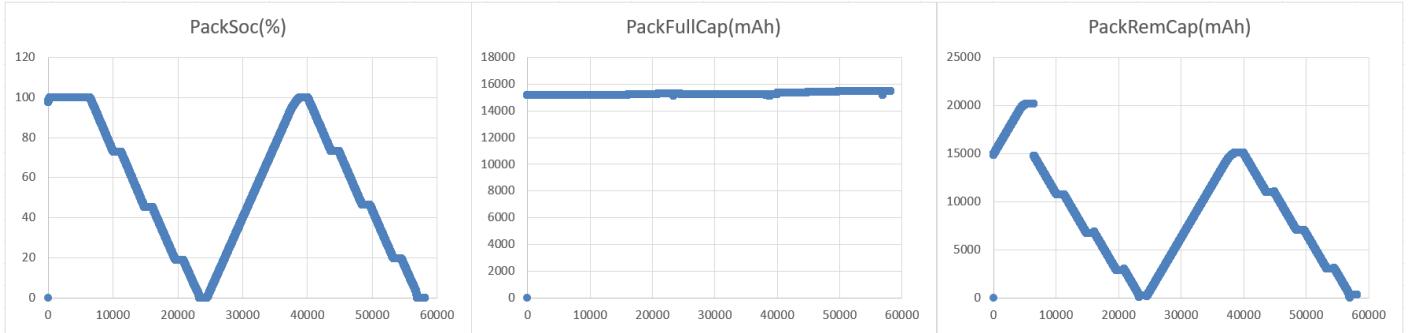


图 5-15. 电池包测试结果

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：进行恒流放电 2 次，然后改变负载。图 5-16 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。

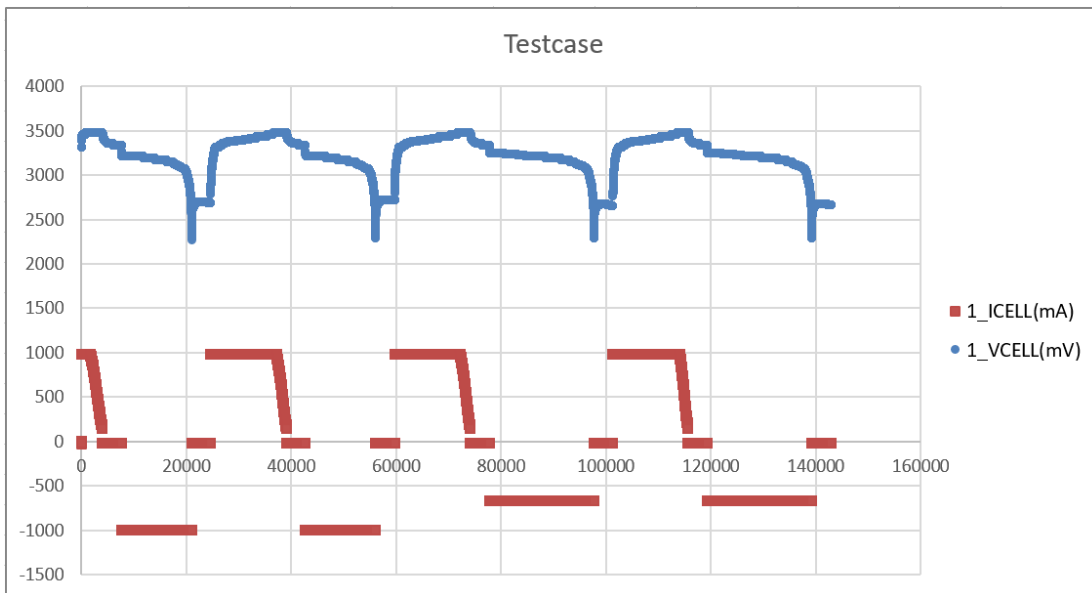


图 5-16. 电池测试用例

请参阅图 5-16 中的测试结果，您会发现，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，您可以看到，当电压达到放电结束电压 (2300mV) 时，SmoothSOC 可以在 0% 至 100% 之间表现得非常理想。同时要记住，EmptySOC 需要学习周期，这意味着如果您没有输入 iq15AbsEmptySocMatrixInput，则当电池达到其第一次放电电压结束时，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 变为某个值后，意味着输出 NomFullCap 具有可接受的精度。

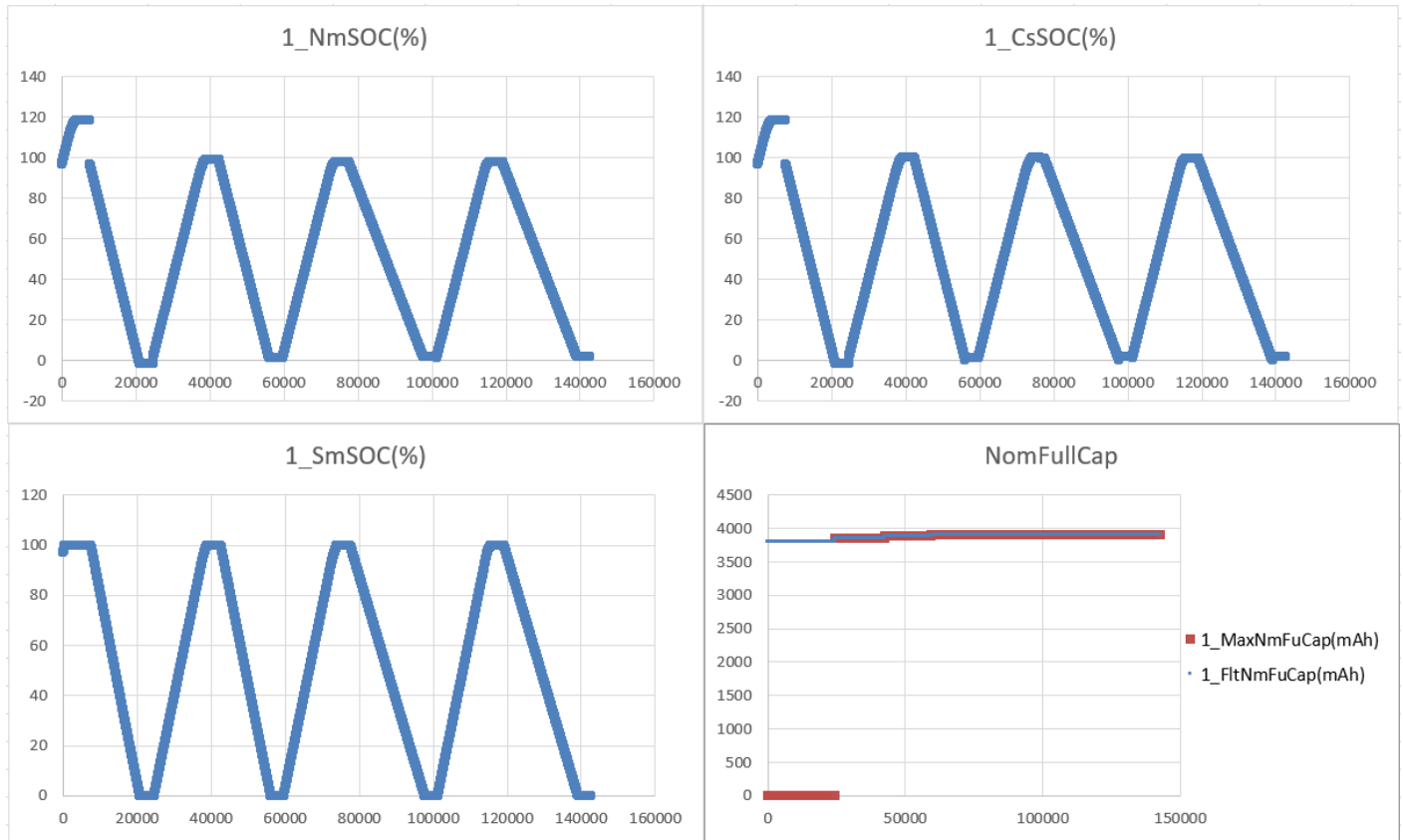


图 5-17. 电池测试结果

图 5-18 显示电池包的结果。

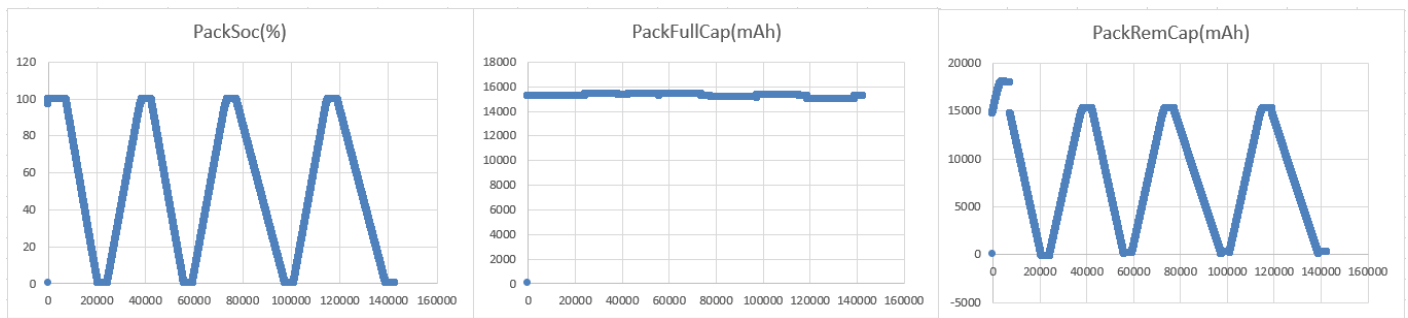


图 5-18. 电池测试结果

6 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 设计流程指南](#)

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 © 2024，德州仪器 (TI) 公司