

Application Note

MS9920T / MS9930T / MS9940T

Analog Front-End 3 ~ 15 Series Li-ion and Phosphate Cell Battery Monitor with I2C

9920/9930/9940 系列模拟前端 (AFE) 芯片，主要目标是用来测量电池组的电压、剩余电量，并在此基础上对电池提供保护和控制。9920 支持 3 到 5 组电池的串联，9930 支持 6 到 10 组电池的串联，9940 支持 11 到 15 组电池的串联。通过 I²C，主机控制器可以使用 99x0 来实现电池组管理功能，例如监控电池电压和电池组充放电电流以及电池组温度，提供过流以及短路的保护，另外芯片还内置电池均衡功能。

芯片内置两个 $\Sigma\Delta$ 转换器，它们相互独立。其中一个用来测量电池电压和芯片温度；另一个用来测量电量，又称为库伦计数器。

9920/9930/9940 内置过压保护 (OV)、欠压保护(UV)、和过流保护(OCD)、短路保护(SCD)。

9920/9930/9940 提供两个低侧场效应管(FET)驱动器，即充电 (CHG) 和放电 (DSG)，可用于直接操纵低侧功率 N 沟道 FET，或作为控制启用高侧 P 沟道或 N 沟道外部电路的场效应管。

9920/9930/9940 内部专用的警报管脚 ALERT 用作主微控制器的中断信号，快速通知微控制器芯片内部的状态变化。可用的 ALERT 引脚也可以由二级保护器从外部驱动，以提供禁用 CHG 和 DSG 信号和更高系统可见性的冗余方法。

9920/9930/9940 还提供 LDO 输出，供外部主控制器使用，省去外置的 LDO。

功能描述

1. 子系统

99x0 由三个主要子系统组成：测量、保护和控制。它们协同工作以确保基本电池组参数——电压、电流和温度——被准确捕获并易于主机控制器使用，同时确保硬件的基线或二级水平在主机控制器无法管理某些故障情况时提供保护。

注意：99x0 旨在作为芯片组系统解决方案的一部分用作模拟前端(AFE)：需要配套微控制器来监督和控制此 AFE。

- 测量子系统的核心职责是将电池电压、电池组电流（集成到通过的电荷计算中）、外部热敏电阻温度和内部芯片温度数字化。它还可以通过简单地将所有测量的电池电压相加来自动计算总电池组电压。

- 保护子系统提供基线或二级硬件保护，以便在主机失控或主机无法及时响应特定故障事件时更好地支持电池组的 FMEA 要求。集成保护包括电池组级故障，例如 OV、UV、OCD、SCD、外部二级保护器故障检测以及内部逻辑“看门狗”式芯片故障 (XREADY)。保护事件将触发 ALERT 引脚的切换，以及 DSG 和 CHG 的自动禁用（取决于故障）。从故障事件中恢复必须由主机微控制器处理。

- 控制子系统实现了一套有用的电池组功能，包括直接低侧 NCH FET 驱动器、电池均衡驱动器、ALERT 数字输出、外部 LDO 等。

以下部分更详细地描述了每个子系统，并解释了可用的各种电源状态。

1.1 测量子系统概述

监控子系统确保主机可以轻松测量所有电池电压、温度和电池组电流。所有 ADC 均由我司调整。

如果在同一事件中读取高位寄存器和低位寄存器（使用地址自动递增），则 ADC 和 CC 数据始终作为原子值返回。

1.1.1.1 到主机控制器的数据传输

99x0 具有全数字接口：所有信息都通过 I²C 传输，只需读取和/或写入存储相关数据的适当寄存器即可。块读取和写入，由每字节 8 位 CRC 代码缓冲，确保数据的快速和稳健传输。

1.1.1.2 14 位 ADC

每个 99x0 器件都使用 14 位 ADC 测量电池电压和温度。该 ADC 测量所有差分电池电压、热敏电阻和/或芯片温度，标称满量程无符号范围为 0~5.98V，LSB 为 354μV。

要启用 ADC，必须设置 SYS_CTRL1 寄存器中的 [ADC_EN] 位。只要器件进入 NORMAL 模式，该位就会自动置位。启用后，ADC 可确保集成的 OV 和 UV 保护功能正常工作。

对于每组连续五个单元（VC1 至 VC5、VC6 至 VC10），当该特定组中没有单元处于均衡状态时，每个单元都会在 50 毫秒的抽取窗口内进行测量，并且每 250 毫秒提供一次完整更新。在 9930 和 9940 中，五个电池上方的每组五个电池都是并行测量的。

50 ms 抽取有助于极大地消除嘈杂电机环境中存在的混叠效应。

当一组连续的 5 个电池中的任何电池被均衡时，这些受影响的电池会在减少的 12.5 毫秒抽取周期内进行测量，以允许电池均衡正常运行，而不会影响集成的 OV 和 UV 保护。由于电池均衡通常仅在电池组充电或空闲期间执行，

缩短抽取周期不应影响精度，因为在这些时间内系统噪声大大降低。这种减少的抽取周期仅适用于其中一个电池正在均衡的集合。

以下是 9920/9930/9940 器件的总结：

- VC1 至 VC5 测量均在 CELLBAL1 寄存器中的所有位为 0 时的 50 ms 抽取周期和 CELLBAL1 寄存器中的任何位为 1 时的 12.5 ms 抽取周期内进行。
- VC6 至 VC10 测量均在 CELLBAL2 寄存器中的所有位为 0 时的 50 ms 抽取周期内进行，以及 CELLBAL2 寄存器中的任何位为 1 时的 12.5 ms 抽取周期内进行。
- VC11 至 VC15 测量均在 CELLBAL3 寄存器中的所有位为 0 时的 50 ms 抽取周期和 CELLBAL3 寄存器中的任何位为 1 时的 12.5 ms 抽取周期内进行。
- 总更新间隔为 250 毫秒。

每个电池输入都针对增益或偏移进行了工厂微调，因此通过 I²C 得出的读数始终在器件之间保持一致，并且不需要额外的校准或校正因子应用程序。

需要启用 ADC 才能运行内部的过压(OV) 和欠压(UV)保护。

下面显示了如何将 14 位 ADC 读数转换为模拟电压。每个芯片都经过工厂校准，增益和偏移量存储在 MTP 中。

ADC 传递函数是一个线性方程，定义如下：

$$V(\text{cell}) = \text{GAIN} \times \text{ADC}(\text{cell}) + \text{OFFSET} (1)$$

GAIN 以 μV/LSB 为单位存储，而 OFFSET 以 mV 为单位存储。

下表提供了一些示例电池电压计算。出于说明目的，该示例使用 354μV/LSB 的假设增益 (ADCGAIN<4:0> = 0x0F)和 30mV 的 OFFSET(ADCOFFSET<7:0> = 0x1E)。

| 14 位 ADC 结果 | 14 位 ADC 结果 十进制 | GAIN(μV/LSB) | OFFSET(mV) | 电池电压(mV) |
|-------------|--------------------|--------------|------------|----------|
| 0x1800 | 6144 | 354 | 30 | 2205 |
| 0x1F10 | 7952 | 354 | 30 | 2845 |

注意，当从 SHIP 模式进入 NORMAL 模式时，在读取初始电池电压数据之前，请等待以下时间：

- 9920: 250 毫秒
- 9930: 400 毫秒
- 9940: 800 毫秒

1.1.2.1 使用主机微控制器的可选实时校准

14 位 ADC 测量的电池电压值的性能具有工厂校准精度。

虽然这适用于 99x0 AFE 系列旨在支持的大多数电池组保护和基本监控应用，但某些系统可能需要更高的精度性能。

为此，使用主机微控制器上可用的 ADC 通道和通用的输出端口与 99x0 配对。由两个精密电阻器 and 一个小信号场效应管 FET 组成的简单外部电路由主微控制器激活，以确定总堆栈电压 V_{STACK} 。然后将其与由 99x0 的内部 ADC 测量的单个电池电压的总和相比较。生成的传递函数系数 $GAIN_2$ 简单地应用于每个电池电压 ADC 值，以提高精度。

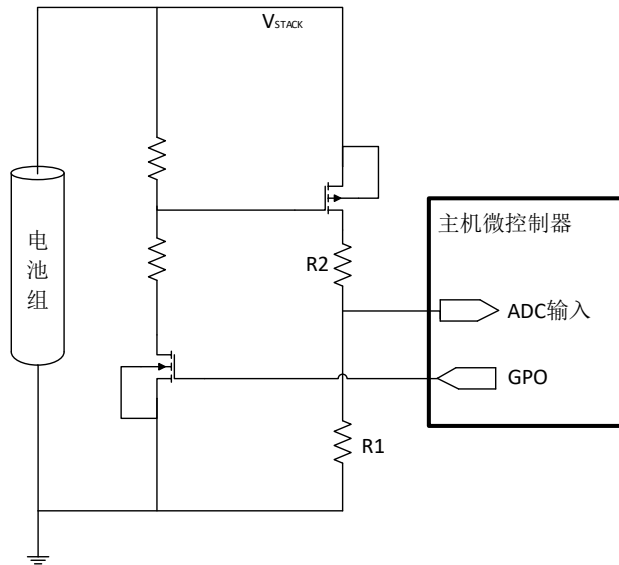


图 1. 针对内部 ADC 的外部实时校准电路

过程如下：

1. 定期测量 V_{STACK} 。

A. $V_{STACK} = V_{AD} \times (R1 + R2) / R1$

2. 从 99x0 读出所有 V_{CELL} ADC 读数并应用存储在 99x0 中标准的 $GAIN$ 和 $OFFSET$ 值

A. $V(1) = GAIN \times ADC1 + OFFSET$, $V(2) = GAIN \times ADC2 + OFFSET$, 依此类推

3. 将所有 V_{CELL} 值相加， V_{SUM} 。

A. $V_{SUM} = V(1) + V(2) + V(3) \dots$

4. 计算 $GAIN_2$

A. $GAIN_2 = V_{STACK} / V_{SUM}$

作为一般建议，当电池电压增加或降低超过 100 mV，应生成新的 $GAIN_2$ 函数。

使用 $GAIN_2$ ，每个电池电压计算变为：

$$V(\text{cell}) = GAIN_2 \times (GAIN \times \text{ADC}(\text{cell}) + \text{OFFSET})$$

对于不需要此附加使用中校准功能的系统， $GAIN_2$ 只是“1”。

1.1.3 16 位 CC

16 位 Σ - Δ ADC，通常称为库仑计数器(CC)，可测量电流检测电阻器上的累积电荷。此读数的积分周期为 250 毫秒。

CC 可以在两种模式之一下操作：始终开启模式 Always On 和单次模式 1-SHOT。

- 在 Always On 模式下，CC 以 100% 运行，每 250 毫秒收集一次新读数。每次读数结束时都会设置 CC_READY 位，这会将 ALERT 引脚切换为高电平以通知微控制器有新的读数可用。要启用 Always On 模式，请设置[CC_EN] = 1。
- 在 1-SHOT 模式下，CC 执行单个 250-ms 读取，并在完成时类似地设置 CC_READY 位。此模式适用于非计量用途，其中主机仅希望检查电池组电流。

要启用 1-SHOT 读数，请确保[CC_EN] = 0 并设置[CC_ONESHOT] = 1。

CC 的满量程范围为 $\pm 276.6\text{mV}$ ，最大推荐输入范围为 $\pm 200\text{mV}$ ，从而产生大约 $8.44\mu\text{V}$ 的 LSB。

以下等式显示了如果不执行板级校准，如何将 16 位 CC 读数转换为模拟电压：

CC 读数（以 μV 为单位）= [16 位 2 进制的补码值] \times ($8.44\mu\text{V}/\text{LSB}$)

| 16 位库仑计数器读数 | 对应十进制 | 对应的电压(μV) |
|-------------|--------|------------------------|
| 0x0001 | 1 | 8.44 |
| 0x2710 | 10000 | 84400 |
| 0x7D00 | 32000 | 270080 |
| 0x8300 | -32000 | -270080 |
| 0xC350 | -15536 | -131123.84 |
| 0xFFFF | -1 | -8.44 |

1.1.4 外部热敏电阻

要选择热敏电阻测量模式，请设置 [TEMP_SEL] = 1。

热敏电阻 TS1 连接在 TS1 和 VSS 之间；TS2 连接在 TS2 和 VC5 之间（仅限 9930 和 9940）；TS3 连接在 TS3 和 VC10X 之间（仅限 9940）。这些热敏电阻可以放置在电池组中的各个区域，以测量诸如局部电池温度、FET 加热等情况。

可以使用 TS1、TS2 和 TS3 寄存器中的 14 位 ADC 读数和 10k 内部上拉电阻计算热敏电阻阻抗，如下所示：

以下等式显示如何使用 TS1、TS2 和 TS3 中的 14 位 ADC 读数来确定外部 103AT 热敏电阻的电阻值：设内部上拉电阻为 $R1=10000\Omega$ ，外部下拉热敏电阻为 R_T ，内部 LDO 输出电压 3.3V ，ADC 读到 TS_x 端点电压为 V_{TS} ，则：

$$V_{\text{TS}} = (\text{十进制 ADC 读数}) \times 354 \mu\text{V}/\text{LSB} \quad (4)$$

由电阻分压得到：

$$R_T = (10,000 \times V_{\text{TS}}) \div (3.3 \times 10^{-6} - V_{\text{TS}}) \quad (5)$$

要将热敏电阻转换为温度，请参阅热敏电阻元件制造商的数据表。

1.1.5 芯片温度监控器

在外部和内部温度监控之间切换时，由于自然调度程序更新间隔，可能会产生 2 秒的延迟。

芯片温度模块产生与芯片温度成正比的电压，如果不使用封装热敏电阻或确保遵守芯片功耗要求，则提供一种减少组件数量的方法。使用与电池电压相同的 14 位 ADC 来测量芯片。

要选择内部芯片温度测量模式，请设置[TEMP_SEL] = 0。

对于 9930 和 9940，可进行多种芯片温度测量。这些存储在 TS2 和 TS3 中。

要将 DIETEMP 读数转换为温度，请参阅以下公式。如果需要从 DIETEMP 获得更准确的温度读数，则在生产校准期间应存储室温下的 DIETEMP 值。

以下等式显示了如何在 [TEMPSEL] = 0 时使用 TS1、TS2 和 TS3 中的 14 位 ADC 读数确定芯片内部温度：

$$V_{25^{\circ}\text{C}} = 1.26\text{V (标称)} \quad (6)$$

$$V_{\text{TS}} = (\text{十进制 ADC 读数}) \times 354 \mu\text{V/LSB} \quad (7)$$

$$\text{TEMPDIE} = 25^{\circ}\text{C} - ((V_{\text{TS}} \times 10^{-6} - V_{25^{\circ}\text{C}}) \div 0.0040) \quad (8)$$

1.1.6

以 4 节电池为例，转换为数字形式后，将每个电池电压相加，并将求和结果存储在 BAT 寄存器中。这个 16 位值的标称 LSB 为 1.46 mV。

下面显示了如何由 4 节电池构成的电池包电压对应的 14 位 ADC 读数转换为模拟电压。该值还使用存储在 MTP 中的 GAIN 和 OFFSET。

ADC 传递函数是一个线性方程，定义如下：

$$V(\text{BAT}) = 4 \times \text{GAIN} \times \text{ADC}(\text{cell}) + (\text{电池个数} \times \text{OFFSET}) \quad (9)$$

GAIN 以 $\mu\text{V/LSB}$ 为单位存储，而 OFFSET 以 mV 为单位存储。

1.1.7 系统调度器

主调度程序监督监控间隔，每 250 毫秒创建一次完整更新。每 2 秒进行一次温度测量。电池组电压每 250 毫秒计算一次。有关系统调度器的更多信息，请参见 99x0 应用报告的 Cell Battery Monitor 中的嵌入式调度器。

1.2 保护子系统

1.2.1 集成硬件保护

提供集成硬件保护作为额外的安全级别，旨在补充将纳入主机控制器固件的标准保护功能集。它们不应用作保护电池组的唯一手段，但可用于潜在的失效模式和效应分析；例如，如果主机微控制器无法对以下任何保护情况做出反应。在系统启动期间，主机微控制器应将所有硬件保护阈值和延迟加载到 AFE。AFE 还将默认为预定义的阈值和延迟设置，以防主微控制器无法或不想对保护设置进行编程。

放电过流(OCD)和放电短路(SCD)使用采样模拟比较器实现，这些比较器以 32 kHz 运行，并在器件处于正常模式时持续监控 (SRP-SRN)两端的电压。在检测到超过设定的 OCD 或 SCD 阈值的电压后，计数器开始计数，直至达到设定的延迟设置。如果计数器达到其目标值，则更新 SYS_STAT 寄存器以指示故障条件，更新 FET 状态，如表 1 所示，并且 ALERT 引脚被驱动为高电平以中断主机。

OCD 和 SCD 保护的故障阈值和延迟设置通过 PROTECT1 和 PROTECT2 寄存器进行配置。有关支持的值的详细信息，请参阅寄存器映射。

通过将电池电压读数与 OV 和 UV 寄存器中的 8 位编程阈值进行比较，以数字方式处理过压(OV)和欠压(UV)保护。

OV 阈值存储在 OV_TRIP 寄存器中，是 14 位 ADC 读数的 8 位的直接映射，高 2 MSB 预设为“10”，低 4 LSB 预设为“1000”。也就是说，对应的 OV 映射到“10-XXXX-XXXX-1000”。OV 阈值的可编程范围约为 3.15 至 4.7 V，但由于用于映射 ADC 值的（GAIN、OFFSET）线性方程，这会发生变化。

UV 阈值存储在 UV_TRIP 寄存器中，是 14 位 ADC 读数的 8 位的直接映射，高 2 MSB 预设为“01”，低 4 LSB 预设为“0000”。也就是说，对应的 OV 映射到“01-XXXX-XXXX-0000”。UV 阈值的可编程范围约为 1.58 至 3.1 V，但由于用于映射 ADC 值的（增益、偏移）线性方程，这会发生变化。

表 1. OV 和 UV 寄存器的设置

| 保护类型 | 高 2 位 | 中间 8 位 | 低 4 位 |
|-------|-------|----------------|-------|
| 过压 OV | 10 | 设置 OV_TRIP 寄存器 | 1000 |
| 欠压 UV | 01 | 设置 UV_TRIP 寄存器 | 0000 |

注意：为了支持 9920、9930 和 9940 中的灵活单元配置，UV 忽略了任何电池电压读数低于 $UV_{MINQUAL}$ 的情形。这允许单元引脚在不需要所有电池的应用中短路（例如，6 节电池使用 9930）。

默认保护阈值和延迟显示在本数据表末尾的寄存器说明中。

当芯片进入正常模式时，它们被加载到芯片的数字寄存器(RAM)中。然后，这些 RAM 值可能会被主机控制器覆盖为任何其他值，这些值将保留到上电复位(POR)事件。建议主机控制器在其标准通电和/或重新初始化序列期间重新加载这些值。

要为芯片计算正确的 OV_TRIP 和 UV_TRIP 寄存器值，请使用以下过程：

1. 确定所需的 OV。
2. 从它们对应的寄存器中读出[ADCGAIN]和[ADCOFFSET]。请注意，ADCGAIN 存储单位为 $\mu\text{V}/\text{LSB}$ ，而 ADCOFFSET 以 mV 为单位存储。
3. 计算满足所需 OV 和 UV 跳闸阈值所需的完整 14 位 ADC 值，如下所示：
 - A. $OV_TRIP_FULL = (OV - ADCOFFSET) \div ADCGAIN$
 - B. $UV_TRIP_FULL = (UV - ADCOFFSET) \div ADCGAIN$
4. 从完整的 14 位值中去除高 2 位和低 4 位，只保留剩余的中间 8 位。这可以通过将 OV_TRIP_FULL 和 UV_TRIP_FULL 二进制值右移 4 位来完成并删除上面的高 2 位。
5. 将 OV_TRIP 和 UV_TRIP 写入其对应的寄存器。

OV 和 UV 保护都需要启用 ADC。确保[ADC_EN]位设置为 1。

1.2.2 缩短测试时间

SYS_CTRL2 寄存器中提供了一个特殊的调试和测试配置位，称为[DELAY_DIS]。设置[DELAY_DIS]绕过 OV/UV 保护故障计时器，并允许在应用此类故障条件后 200 ms 内记录故障条件。

1.3 控制子系统

1.3.1 FET 驱动（CHG 和 DSG）

每个 99x0 器件提供两个低侧 FET 驱动器 CHG 和 DSG，它们控制 N 沟道功率 FET 或可能用作启用各种其他电路（例如高侧 N 沟道电荷泵电路）的信号。

DSG 和 CHG 驱动器在启用时都具有快速上拉至标称 12 V 的电压。DSG 使用快速下拉。禁用时为 VSS，而禁用时 CHG 使用高阻抗（标称 800kΩ）下拉路径。额外的内部钳位电路可确保 CHG 引脚的最大电压不超过 20 V。

CHG 和 DSG 上拉电路的电源路径源自 REGSRC 引脚，而不是 BAT。

要启用 CHG FET，请将 [CHG_ON] 寄存器位设置为 1；要禁用，设置 [CHG_ON] = 0。放电 FET 可以通过 [DSG_ON] 寄存器位类似地进行控制。

某些故障条件或电源状态转换将清除 CHG/DSG FET 控制的状态。见表 2 显示响应各种系统事件对 [CHG_ON] 和 [DSG_ON] 采取的操作：

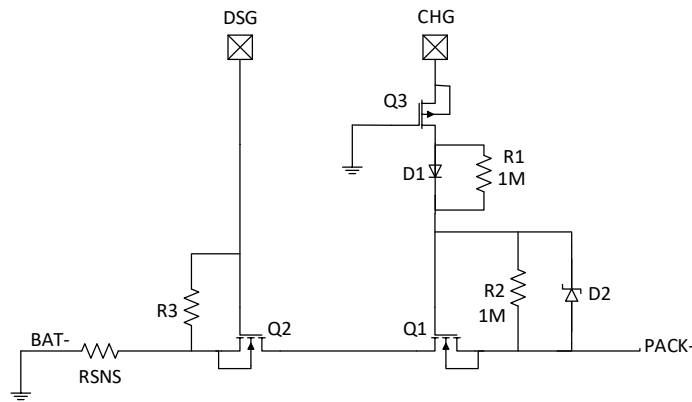


图 2. CHG 和 DSG 外部 FET 电路

说明：Q3 是低成本 P 沟道 FET，用于使 CHG 远离低于 VSS 的任何电压。当 CHG 没被拉高时，PACK- 被拉到低于 VSS 不会被 CHG 视为 Q2 没有打开。二极管 D1 允许 CHG 把 Q1 的栅极拉高。电阻 R1 的作用是当 PACK- 被拉高时降低电压，并且限制流入 CHG 脚的电流，R1 使流入 CHG 脚最大电流被限制为 $(V_{PACK} - V_{CHG_CLAMP})/R1$ 。钳位齐纳二极管 D2（可选）可用来防止 Q1 打开过快。当 CHG 关闭时，R2 可钳位 Q1。

表 2. 不同系统事件下 CHG,DSG 的响应

| 事件 | 寄存器 [CHG_ON] | 寄存器 [DSG_ON] |
|-------------------|--------------|--------------|
| OV | 设置为 0 | - |
| UV | - | 设置为 0 |
| OCD | - | 设置为 0 |
| SCD | - | 设置为 0 |
| ALERT 脚检测到故障 | 设置为 0 | 设置为 0 |
| DEVICE_XREADY 被置位 | 设置为 0 | 设置为 0 |
| 从正常模式进入到 SHIP 模式 | 设置为 0 | 设置为 0 |

注意：主机微控制器必须初始化所有保护设置。为了恢复故障之后的 FET 操作，主机微控制器必须首先清除掉 SYS_STAT 寄存器中相应的状态位，这也会使 ALERT 管脚清 0，然后手动重新使能 CHG 或/和 DSG 位。如果这种情况仍然存在，过压 OV 和欠压 UV 等故障，则可以立即重新触发。清除状态位的详细信息参考状态寄存器 SYS_STAT。

1.3.2 负载检测

CHG 引脚上有一个负载检测电路，当 CHG FET 被禁用([CHG_on]=0)时，该电路被激活。当高阻抗（约 800kΩ）下拉路径实际应将 CHG 引脚保持在 VSS 时，该电路检测 CHG 引脚是否被外部拉高，并有助于确定组件引脚（AFE 外部）是否保持在高电压下，例如，如果在功率 FET 关闭时存在负载。从 SYS_CTRL1 寄存器的[LOAD_PRESENT]位读取负载检测电路的状态。

发生 OCD 或 SCD 故障后，DSG FET 将被禁用（[DSG_ON] 清零），CHG FET 必须类似地明确禁用以激活负载检测电路。主机微控制器可以定期轮询[LOAD_PRESENT]位以确定 PACK- 引脚的状态并确定何时负载被去掉（[LOAD_PRESENT] = 0）。

1.3.3 电池均衡

9920 完全支持内部和外部无源电池均衡选项，而外部电池均衡建议对 9930 和 9940 进行。由主机控制器来确定确切的在任何给定系统中使用的均衡算法。每个 99x0 器件提供电池电压和均衡驱动程序以实现这一点。如果使用内部电池均衡驱动器，每个电池最多可均衡 50mA。如果使用外部电池均衡，则可以采用更高的均衡电流。

要激活特定的电池均衡通道，只需在 CELLBAL1, CELLBAL2 或 CELLBAL3 寄存器中设置该电池的相应位。例如，通过设置[CB1]启用 VC1-VC0，而设置 VC12-VC11 通过[CB12]。

多个电池可以同时均衡。由用户自行决定理想数量的电池同时均衡。相邻电池不应同时均衡。这可能会导致电池引脚超过其绝对最大条件，这也不建议用于外部均衡。此外，如果使用内部均衡，应注意避免超过封装功耗评级。

注意：

主机控制器必须确保在以下每组中没有两个相邻的电池同时均衡：

- VC1-VC5
- VC6-VC10
- VC11-VC15

用于均衡的总占空比约为每 250 毫秒 70%。这是因为 250 m 中有一部分被分配给通过 ADC 进行的正常电池电压测量。如果[ADC_EN]=1，则 OV 和 UV 保护不受电池均衡的影响，因为电池均衡是暂时的，每 250 毫秒暂停一小段时间，在此期间获取电池电压读数。这确保 OV 和 UV 保护不会意外触发，或错过电池上的实际 OV/UV 条件，同时均衡已启用。

注意：

CELLBAL1、CELLBAL2 和 CELLBAL3 中的所有电池均衡控制位在以下事件下自动清除，并且必须在清除事件后由主机微控制器明确重写：

- DEVICE_XREADY 被置位
- 从 SHIP 模式进入正常模式

1.3.4 警报

ALERT 引脚用作高电平有效数字中断信号，可连接到主机的微控制器端口 GPIO。该信号是 SYS_STAT 寄存器中所有位的或。

为了清除 ALERT 信号，必须首先通过写“1”来清除 SYS_STAT 寄存器中的相应的源位信号。一旦所有位都被清除，这将导致 ALERT 引脚自动清除。

ALERT 引脚也可由外部源驱动；例如，该电池包可能包括一个辅助过压保护 IC。当 ALERT 引脚在低电平时从外部强制为高电平，芯片将识别此作为 OVRD_ALERT 故障并设置 [OVRD_ALERT] 位。这会触发 CHG 和 DSG FET 驱动器。当芯片已经强制执行时，当芯片已经从另一个条件强制 ALERT 信号高时，芯片无法识别 ALERT 信号输入高。

ALERT 引脚没有内部去抖动支持，因此应注意保护引脚免受噪声或其他寄生瞬变。

注意：

强烈建议在 ALERT 脚对 VSS 放置一个 500 kΩ-1 MΩ 的外部下拉电阻，且尽可能靠近 IC。其他建议包括：

- a) 保持 IC 和连接至 ALERT 引脚的组件之间的所有走线通道尽量短。
- b) 在连接到 ALERT 脚的周围包括一个防护环。

1.3.5 输出 LDO

提供可调输出电压调节器 LDO，作为向电池组中的附加组件（如主机微控制器或 LED）供电的简单方法。LDO 在生产测试过程中在 MTP 中配置，默认出厂 3.3V，可提供 2.5V。

使用 9930 和 9940，必须在 BAT 和 REGSRC 之间的外部路径中添加共源共栅小信号 FET。这有助于降低封装外部的大部分功耗，并降低封装功耗。

1.4 通信子系统

AFE 实现一个标准的 100 kHz I²C 接口，并充当从芯片。I²C 芯片地址为 7 位，在工厂编程。写入事件如图 3 所示。通过在停止之前发送额外的数据字节，可以进行块写入。I²C 块将自动增加每个数据字节后的寄存器地址。

启用时，CRC 的计算如下：

- 在单字节写入事件中，CRC 通过从机地址、寄存器地址和数据进行计算。
- 在块写入事件中，第一个数据字节的 CRC 通过从机地址、寄存器地址和数据进行计算。后续数据字节的 CRC 仅在数据字节上计算。

CRC 多项式为 x^8+x^2+x+1 ，初始值为 0。

当从芯片检测到坏 CRC 时，I²C 从芯片将不对该 CRC 响应，这将导致 I²C 从芯片进入空闲状态。

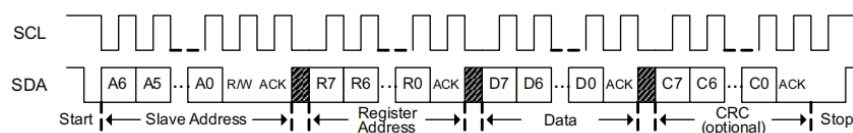


图 3. I²C 写入

图 4 显示了使用重复启动的读取事件

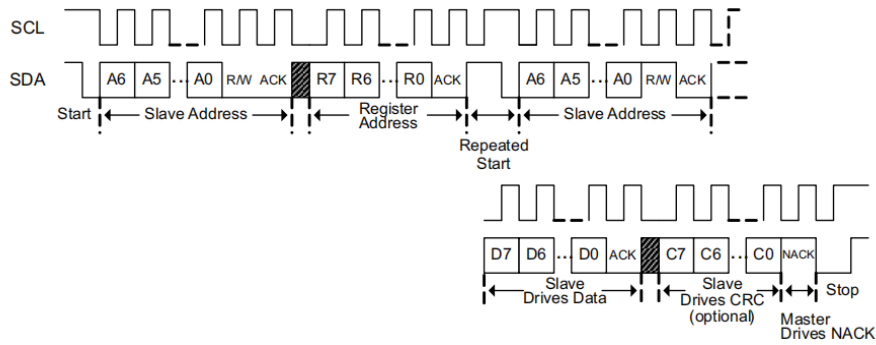


图 4. 使用重复启动的读取事件

图 5 显示了一个不使用重复启动的读取事件，例如在硬件不可用的情况下。对于块读取，主 ACK 对除最后一个数据字节之外的每个数据字节进行同步，并继续对接口进行时钟设置。I2C 块将自动增加每个数据字节后的寄存器地址。

启用时，读取事件的 CRC 计算如下：

- 在单字节读取事件中，CRC 在第二次启动后计算，并使用从属地址和数据字节。
- 在块读取事件中，第一个数据字节的 CRC 在第二次启动后计算，并使用从属地址和数据字节。后续数据字节的 CRC 仅在数据字节上计算。

CRC 多项式为 x^8+x^2+x+1 ，初始值为 0。

当主芯片检测到坏 CRC 时，I2C 主芯片将不对该 CRC 响应，这将导致 I2C 从芯片进入空闲状态。

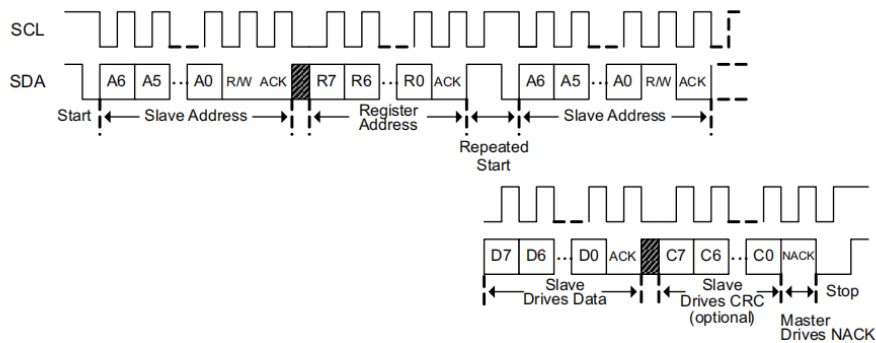


图 5. 不使用重复启动的读取事件

芯片功能模式

每个 99x0 芯片支持以下操作模式。

| 模式 | 描述 |
|---------|---|
| 正常模式 | 完全运行状态。ADC 和 CC 都可以由主机微控制器打开或禁用。如果 ADC 打开，则 OV 和 UV 保护启用。OCD 和 SCD 启用。可禁用 ADC 和 CC 以降低功耗，CC 可在“单次”模式下运行，以实现灵活的节能。 |
| SHIP 模式 | 最低可能的电源状态，用于电池组组装和/或长期电池组存储。必须在 TS1 引脚上看到 BOOT 信号 (VBOOT>1V) 才能从 SHIP 模式进入正常模式。请注意，芯片在 POR 时始终进入 SHIP 模式。 |

1. 正常模式

正常模式表示完全运行模式，在该模式下，所有模块均已启用，且芯片的电流消耗最高。在此模式下，某些模块/功能可被禁用以节省电源，包括 ADC 和 CC。只要 ADC 启用，OV 和 UV 就会持续运行。在此模式下，OCD 和 SCD 比较器可能无法禁用。

主机也会启动从正常模式到 SHIP 模式的转换，并且需要连续写入 SYS_CTRL1 寄存器中的两位。

2. SHIP 模式

SHIP 模式是 99x0 支持的基本和最低功率模式。SHIP 模式在初始组件组装期间和每次 POR 事件之后自动进入。当芯片处于正常模式时，主机控制器可通过特定的 I²C 命令序列进入 SHIP。

在 SHIP 模式下，仅打开最少的电路模块，包括 VSTUP 电源和原始 BOOT 检测器。从 SHIP 模式唤醒到正常模式需要拉动 TS1 引脚，使其大于 VBOOT，从而触发芯片启动序列。

要从 NORMAL 模式进入 SHIP 模式，SYS_CTRL1 寄存器中的 [SHUT_A] 和 [SHUT_B] 位必须在两次连续写入中使用特定模式写入：

- 写入#1: [SHUT_A] = 0, [SHUT_B] = 1
- 写入#2: [SHUT_A] = 1, [SHUT_B] = 0

注意，[SHUT_A] 和 [SHUT_B] 在执行上述 shutdown 命令之前应分别处于 0 状态。如果在芯片中输入此特定序列，芯片将转换为 SHIP 模式。如果任何其他序列被写入 [SHUT_A] 和 [SHUT_B] 位，或者如果两种模式中的任何一种没有正确输入，芯片将不会进入 SHIP 模式。

警告

请勿操作低于 POR 的芯片。使用 9940 进行设计时，中间电压 (BAT-VC10X)、(VC10X-VC5X) 和 (VC5X-VSS) 必须分别永远不会低于 VSHUT。万一发生这种情况，必须关闭所有三个中间电压 (BAT-VC10X)、(VC10X-VC5X) 和 (VC5X-VSS)，使它们低于 VSHUT 并通过应用适当的 VBOOT 信号重新启动 TS1 引脚，以此启动完整的芯片初始化。使用 9930 进行设计时，中间电压 (BAT-VC5X) 和 (VC5X-VSS) 都绝不能低于 VSHUT。如果发生这种情况，必须关闭两个中间电压 (BAT-VC5X) 和 (VC5X-VSS)，使它们低于 VSHUT 并通过应用适当的 VBOOT 信号重新启动 TS1 引脚，以此启动完整的芯片初始化。

在 POR 事件期间，芯片也将进入 SHIP 模式；但是，这不是建议的 SHIP 模式输入方法。如果任何电源侧电压降到 VSHUT 以下，然后备份到 VPORA 以上，则芯片默认进入 SHIP 模式状态。这类类似于初始组件组装条件。为了退出 SHIP 模式进入正常模式，芯片必须遵循标准 BOOT 顺序，在 TS1 引脚上施加大于 VBOOT 阈值的电压。

寄存器表

| 名称 | 地址 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|-------------------------|------|----------|------|-------------------|----------------|----|----|-----|-----|
| SYS_STAT | 0x00 | CC_READY | RSVD | DECIVE_XR EADY | OVRD_ ALERT | UV | OV | SCD | OCD |
| CELLBAL1 | 0x01 | RSVD | | | CB<5:1> | | | | |
| CELLBAL2 ⁽¹⁾ | 0x02 | RSVD | | | CB<10:6> | | | | |
| CELLBAL3 ⁽²⁾ | 0x03 | RSVD | | | CB<15:11> | | | | |

| | | | | | | | | | |
|------------------------|------|------------------|----------------|------------------|----------------|-----------------|-----------------|------------|------------|
| SYS_CTRL1 | 0x04 | LOAD_ PRESENT | RSVD | | ADC_EN | TEMP_ SEL | RSVD | SHUT_ A | SHUT_ B |
| SYS_CTRL2 | 0x05 | DELAY_DIS | CC_EN | CC_ ONESHOT | RSVD | | | DSG_ ON | CHG_ ON |
| PROTECT1 | 0x06 | RSNS | RSVD | | SCD_DELAY<1:0> | | SCD_THRESH<2:0> | | |
| PROTECT2 | 0x07 | RSVD | OCD_DELAY<2:0> | | | OCD_THRESH<3:0> | | | |
| PROTECT3 | 0x08 | UV_DELAY<1:0> | | OV_DELAY<1:0> | | RSVD | | | |
| OV_TRIP | 0x09 | OV_THRESH<7:0> | | | | | | | |
| UV_TRIP | 0x0A | UV_THRESH<7:0> | | | | | | | |
| CC_CFG | 0x0B | RSVD | | CC_CFG<5:0> | | | | | |
| VC1_HI | 0x0C | RSVD | | VC1_VALUE<13:8> | | | | | |
| VC1_LO | 0x0D | VC1_VALUE<7:0> | | | | | | | |
| VC2_HI | 0x0E | RSVD | | VC2_VALUE<13:8> | | | | | |
| VC2_LO | 0x0F | VC2_VALUE<7:0> | | | | | | | |
| VC3_HI | 0x10 | RSVD | | VC3_VALUE<13:8> | | | | | |
| VC3_LO | 0x11 | VC3_VALUE<7:0> | | | | | | | |
| VC4_HI | 0x12 | RSVD | | VC4_VALUE<13:8> | | | | | |
| VC4_LO | 0x13 | VC4_VALUE<7:0> | | | | | | | |
| VC5_HI | 0x14 | RSVD | | VC5_VALUE<13:8> | | | | | |
| VC5_LO | 0x15 | VC5_VALUE<7:0> | | | | | | | |
| VC6_HI ⁽¹⁾ | 0x16 | RSVD | | VC6_VALUE<13:8> | | | | | |
| VC6_LO ⁽¹⁾ | 0x17 | VC6_VALUE<7:0> | | | | | | | |
| VC7_HI ⁽¹⁾ | 0x18 | RSVD | | VC7_VALUE<13:8> | | | | | |
| VC7_LO ⁽¹⁾ | 0x19 | VC7_VALUE<7:0> | | | | | | | |
| VC8_HI ⁽¹⁾ | 0x1A | RSVD | | VC8_VALUE<13:8> | | | | | |
| VC8_LO ⁽¹⁾ | 0x1B | VC8_VALUE<7:0> | | | | | | | |
| VC9_HI ⁽¹⁾ | 0x1C | RSVD | | VC9_VALUE<13:8> | | | | | |
| VC9_LO ⁽¹⁾ | 0x1D | VC9_VALUE<7:0> | | | | | | | |
| VC10_HI ⁽¹⁾ | 0x1E | RSVD | | VC10_VALUE<13:8> | | | | | |
| VC10_LO ⁽¹⁾ | 0x1F | VC10_VALUE<7:0> | | | | | | | |
| VC11_HI ⁽²⁾ | 0x20 | RSVD | | VC11_VALUE<13:8> | | | | | |
| VC11_LO ⁽²⁾ | 0x21 | VC11_VALUE<7:0> | | | | | | | |
| VC12_HI ⁽²⁾ | 0x22 | RSVD | | VC12_VALUE<13:8> | | | | | |
| VC12_LO ⁽²⁾ | 0x23 | VC12_VALUE<7:0> | | | | | | | |

| | | | | |
|------------------------|------|-----------------|------------------|------|
| VC13_HI ⁽²⁾ | 0x24 | RSVD | VC13_VALUE<13:8> | |
| VC13_LO ⁽²⁾ | 0x25 | VC13_VALUE<7:0> | | |
| VC14_HI ⁽²⁾ | 0x26 | RSVD | VC14_VALUE<13:8> | |
| VC14_LO ⁽²⁾ | 0x27 | VC14_VALUE<7:0> | | |
| VC15_HI ⁽²⁾ | 0x28 | RSVD | VC15_VALUE<13:8> | |
| VC15_LO ⁽²⁾ | 0x29 | VC15_VALUE<7:0> | | |
| BAT_HI | 0x2A | BAT_VALUE<15:8> | | |
| BAT_LO | 0x2B | BAT_VALUE<7:0> | | |
| TS1_HI | 0x2C | RSVD | TS1_VALUE<13:8> | |
| TS1_LO | 0x2D | TS1_VALUE<7:0> | | |
| TS2_HI ⁽¹⁾ | 0x2E | RSVD | TS2_VALUE<13:8> | |
| TS2_LO ⁽¹⁾ | 0x2F | TS2_VALUE<7:0> | | |
| TS3_HI ⁽²⁾ | 0x30 | RSVD | TS3_VALUE<13:8> | |
| TS3_LO ⁽²⁾ | 0x31 | TS3_VALUE<7:0> | | |
| CC_HI | 0x32 | CC_VALUE<15:8> | | |
| CC_LO | 0x33 | CC_VALUE<7:0> | | |
| ASCGAIN1 | 0x50 | RSVD | ADCGAIN<4:3> | RSVD |
| ASCOFFSET | 0x51 | ASCOFFSET<7:0> | | |
| ASCGAIN2 | 0x59 | ADCGAIN<2:0> | RSVD | |

1. 寄存器描述

1.1 系统状态寄存器(0x00h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------|------|---------------|------------|------|------|------|------|
| CC_READY | RSVD | DEVICE_XREADY | OVRD_ALERT | UV | OV | SCD | OCD |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

注：系统状态寄存器中的可以通过写入“1”来清除相应位，写入“0”不改变相应位的状态。

CC_READY - 指示有新的库伦计数器值可用。若该位不是在两个相邻的库伦计数器读数之间被清零，则该位保持 1。该位只能被主机清零，而不是设置。

0 - 新的库伦计数器读数还不可用或被主机清零。

1 - 新的库伦计数器读数可用。在被主机清零前一直保持高电平。

DEVICE_XREADY - 芯片内部故障指示。当该位被设置为 1 时，应由主机清零。可能由于过多的系统瞬态而被设置。该位只能被主机清零，而不是设置。

0 - 未检测到故障，器件正常。

1 - 检测到故障，建议主机在等待一段时间后清除该位。在被主机清零前一直保持高电平。

OVRD_ALERT- ALERT 引脚上外部上拉指示。仅在 ALERT 引脚尚未被芯片本身拉高时有效。

0- 未检测到外部覆盖。

1- 检测到外部覆盖。在被主机清零前一直保持高电平。

UV - 欠压故障指示。

0- 未检测到欠压故障。

1- 检测到欠压故障。在被主机清零前一直保持高电平。

OV - 过压故障指示。

0- 未检测到过压故障。

1- 检测到过压故障。在被主机清零前一直保持高电平。

SCD - 放电短路故障指示。

0- 未检测到放电短路故障。

1- 检测到放电短路故障。在被主机清零前一直保持高电平。

OCD - 放电过流故障指示。

0- 未检测到放电过流故障。

1- 检测到放电过流故障。在被主机清零前一直保持高电平。

1.2 电池均衡 1 寄存器(0x01h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|------|------|---------|------|------|------|------|
| RSVD | | | CB<5:1> | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | rw | rw | rw | rw | rw |

CB<5:1> - 用于 9920、9930、9940 电池均衡。

0- 关闭电池 x 的电池均衡。

1- 开启电池 x 的电池均衡。

1.3 电池均衡 2 寄存器(0x02h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|------|------|----------|------|------|------|------|
| RSVD | | | CB<10:6> | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | rw | rw | rw | rw | rw |

CB<10:6> - 用于 9930、9940 电池均衡。

0- 关闭电池 x 的电池均衡。

1- 开启电池 x 的电池均衡。

1.4 电池均衡 3 寄存器(0x03h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|------|------|-----------|------|------|------|------|
| RSVD | | | CB<15:11> | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | rw | rw | rw | rw | rw |

CB<15:11> - 用于 9940 电池均衡。

- 0 - 关闭电池 x 的电池均衡。
- 1 - 开启电池 x 的电池均衡。

1.5 系统控制 1 寄存器(0x04h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|--------------|------|------|--------|----------|------|--------|--------|
| LOAD_PRESENT | RSVD | | ADC_EN | TEMP_SEL | RSVD | SHUT_A | SHUT_B |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| r | r | r | rw | rw | rw | rw | rw |

LOAD_PRESENT - 仅当 CHG_ON = 0 时有效。当 CHG_ON = 0 时，若 CHG 引脚检测到超过 VLOAD_DETECT，则该位变高，表示存在外部负载。该位只读，且在负载移除时自动清零。

- 0 - CHG 引脚小于 VLOAD_DETECT，或 CHG_ON = 1。
- 1 - CHG 引脚大于 VLOAD_DETECT，当 CHG_ON = 0。

ADC_EN - ADC 使能。

- 0 - 关闭电压和温度 ADC 读数（同时关闭过压保护）。
- 1 - 开启电压和温度 ADC 读数（同时开启过压保护）。

TEMP_SEL - 选择 TSx_HI 和 TSx_LO 的温度来源。

- 0 - 选择芯片内部温度。
- 1 - 选择热敏电阻（所以热敏电阻端口）

SHUT_A/B - 关闭指令。必须以特定的顺序写入：

- SHUT_A = 0, SHUT_B = 0;
 - SHUT_A = 0, SHUT_B = 1;
 - SHUT_A = 1, SHUT_B = 0。
- 忽略其他任何写入指令。

1.6 系统控制 2 寄存器(0x05h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|-----------|-------|------------|------|------|------|--------|--------|
| DELAY_DIS | CC_EN | CC_ONESHOT | RSVD | | | DSG_ON | CHG_ON |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

DELAY_DIS - 关闭 OV, UV, OCD 和 SCD 延时。

- 0 - 正常延时设置。
- 1 - 关闭延时电路，零延时（大约 250ms）。

CC_EN - 库伦计数器连续操作使能。若设置为 1，则忽略 CC_ONESHOT。

- 0 - 关闭库伦计数器连续读。
- 1 - 开启库伦计数器连续读，且忽略 CC_ONESHOT。

CC_ONESHOT - 库伦计数器单次读。若设置为 1，库伦计数器开启一次 250ms 读数，然后关闭。在连续读数时 CC_ONESHOT 会被清零，CC_READY 位会被置 1。

- 0 - 无操作。
- 1 - 开启库伦计数器单次读（仅当 CC_EN = 0，且 CC_READY = 0 时有效）。

DSG_ON - 放电 FET 驱动或放电信号控制。

0 - 关闭 DSG。

1 - 开启 DSG。

CHG_ON - 放电 FET 驱动或放电信号控制。

0 - 关闭 CHG。

1 - 开启 CHG。

1.7 保护 1 寄存器(0x06h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|------|------|----------------|------|-----------------|------|------|
| RSNS | RSVD | | SCD_DELAY<1:0> | | SCD_THRESH<2:0> | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

RSNS - 允许同时将 OCD 和 SCD 阈值加倍。

0 - 较低输入范围的 OCD 和 SCD 阈值。

1 - 较高输入范围的 OCD 和 SCD 阈值。

SCD_DELAY<1:0> - 放电短路延时设置。在使用最大的电池测量输入电阻 R_c 为 $1k\Omega$ 时（对应于最小内部电池均衡电流或外部电池均衡配置），推荐设置 $400\mu s$ 。

| 码 | 时间(μs) |
|-----|---------------|
| 0x0 | 70 |
| 0x1 | 100 |
| 0x2 | 200 |
| 0x3 | 400 |

SCD_THRESH<2:0> - 放电短路阈值设置。

| 码 | RSNS = 1 (mV) | RSNS = 0 (mV) |
|-----|---------------|---------------|
| 0x0 | 37 | 18.5 |
| 0x1 | 59 | 29 |
| 0x2 | 78.5 | 39 |
| 0x3 | 98 | 49 |
| 0x4 | 119 | 59 |
| 0x5 | 139 | 69.5 |
| 0x6 | 159 | 79.5 |
| 0x7 | 181.5 | 91 |

1.8 保护 2 寄存器(0x07h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|----------------|------|------|-----------------|------|------|------|
| RSVD | OCD_DELAY<2:0> | | | OCD_THRESH<3:0> | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

OCD_DELAY<2:0> - 放电过流延时设置。

| 码 | 时间 (ms) |
|-----|---------|
| 0x0 | 8 |
| 0x1 | 20 |
| 0x2 | 40 |
| 0x3 | 80 |
| 0x4 | 160 |
| 0x5 | 320 |
| 0x6 | 640 |
| 0x7 | 1280 |

OCD_THRESH<3:0> - 放电过流阈值设置。

| 码 | RSNS = 1 (mV) | RSNS = 0 (mV) |
|-----|---------------|---------------|
| 0x0 | 18.5 | 9.5 |
| 0x1 | 22.5 | 11.5 |
| 0x2 | 28.5 | 14.5 |
| 0x3 | 34.5 | 17.5 |
| 0x4 | 38.5 | 19.5 |
| 0x5 | 44.5 | 22.5 |
| 0x6 | 49.5 | 25 |
| 0x7 | 56.5 | 28 |
| 0x8 | 62.5 | 31.5 |
| 0x9 | 65 | 33 |
| 0xA | 72.5 | 36.5 |
| 0xB | 80.5 | 40.5 |
| 0xC | 83.5 | 42 |
| 0xD | 88.5 | 44.5 |
| 0xE | 90.5 | 46 |
| 0xF | 94 | 48 |

1.9 保护 3 寄存器(0x08h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|---------------|------|---------------|------|------|------|------|------|
| UV_DELAY<1:0> | | OV_DELAY<1:0> | | RSVD | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

UV_DELAY<2:0> - 欠压延时设置。

| 码 | 时间 (s) |
|-----|--------|
| 0x0 | 1 |
| 0x1 | 4 |
| 0x2 | 8 |
| 0x3 | 16 |

OV_DELAY<2:0> - 过压延时设置。

| 码 | 时间 (s) |
|-----|--------|
| 0x0 | 1 |
| 0x1 | 2 |
| 0x2 | 4 |
| 0x3 | 8 |

1.10 过压阈值寄存器(0x09h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|------|------|------|------|------|------|
| OV_THRESH<7:0> | | | | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| rw | rw | rw | rw | rw | rw | rw | rw |

OV_THRESH<7:0> - 所需过压保护阈值中 ADC 对应的中间 8 位，默认高 2 位设置为 10 且低 4 位设置为 1000，等效的 OV 阈值为： $10 - OV_THRESH<7:0> - 1000$ 。OV_THRESH<7:0>默认设置为 0xAC。OV_THRESH<7:0>基于 ADC 电压，这需要使用 ADCGAIN<4:0>和 ADCOFFSET<7:0>进行反推计算。

1.11 欠压阈值寄存器(0x0Ah)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|------|------|------|------|------|------|
| UV_THRESH<7:0> | | | | | | | |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| rw | rw | rw | rw | rw | rw | rw | rw |

UV_THRESH<7:0> - 所需欠压保护阈值中 ADC 对应的中间 8 位，默认高 2 位设置为 01 且低 4 位设置为 0000，等效的 UV 阈值为： $01 - UV_THRESH<7:0> - 0000$ 。UV_THRESH<7:0>默认设置为 0x97。UV_THRESH<7:0>基于 ADC 电压，这需要使用 ADCGAIN<4:0>和 ADCOFFSET<7:0>进行反推计算。

1.12 库伦计数器配置寄存器(0x0Bh)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|------|------|-------------|------|------|------|------|------|
| RSVD | | CC_CFG<5:0> | | | | | |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| rw | rw | rw | rw | rw | rw | rw | rw |

CC_CFG<5:0> - 为获得最佳性能，该寄存器应在设备启动后立即设置为 0x19。

1.13 电池电压寄存器(0x0Ch - 0x29h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|-----------------|------|------|------|------|------|
| RSVD | | VCx_VALUE<13:8> | | | | | |
| VCx_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

VC1_VALUE<13:8> - 电池 x 的 ADC 结果值高 6 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

VC1_VALUE<7:0> - 电池 x 的 ADC 结果值低 8 位。

1.14 BAT 寄存器(0x2Ah - 0x2Bh)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|-----------------|------|------|------|------|------|------|------|
| BAT_VALUE<15:8> | | | | | | | |
| BAT_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

BAT_VALUE<15:8> - 基于电池单元 1-15 相加的 BAT 计算结果值高 8 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

BAT_VALUE<7:0> - 基于电池单元 1-15 相加的 BAT 计算结果值低 8 位。

1.15 TS1 寄存器(0x2Ch - 0x2Dh)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|-----------------|------|------|------|------|------|
| RSVD | | TS1_VALUE<13:8> | | | | | |
| TS1_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

TS1_VALUE<13:8> - TS1 或芯片温度的 ADC 结果值高 6 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

TS1_VALUE<7:0> - TS1 或芯片温度的 ADC 结果值低 8 位。

1.16 TS2 寄存器(0x2Eh - 0x2Fh)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|-----------------|------|------|------|------|------|
| RSVD | | TS2_VALUE<13:8> | | | | | |
| TS2_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

TS2_VALUE<13:8> - TS2 的 ADC 结果值高 6 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

TS2_VALUE<7:0> - TS2 的 ADC 结果值低 8 位。

1.17 TS3 寄存器(0x30h - 0x31h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|-----------------|------|------|------|------|------|
| RSVD | | TS3_VALUE<13:8> | | | | | |
| TS3_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

TS3_VALUE<13:8> - TS3 的 ADC 结果值高 6 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

TS3_VALUE<7:0> - TS3 的 ADC 结果值低 8 位。

1.18 库伦计数器寄存器(0x32h - 0x33h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|------|------|------|------|------|------|
| CC_VALUE<15:8> | | | | | | | |
| CC_VALUE<7:0> | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| r | r | r | r | r | r | r | r |

CC_VALUE<13:8> - 库伦计数器结果值高 6 位。如果高位和低位寄存器在同一个通信中读取（使用地址自增），总是作为原子值返回。

CC_VALUE<7:0> - 库伦计数器结果值低 8 位。

1.19 ADC 增益偏移寄存器(0x50h, 0x59h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|--------------|------|------|------|--------------|------|------|------|
| RSVD | | | | ADCGAIN<4:3> | | RSVD | |
| ADCGAIN<2:0> | | | RSVD | | | | |
| - | - | - | - | - | - | - | - |
| r | r | r | r | r | r | r | r |

ADCGAIN<4:3> - ADC 增益偏移高 2 位。

ADCGAIN<2:0> - ADC 增益偏移低 3 位。

ADCGAIN<4:0>是 ADC 转换的生产调整值，单位为 $\mu\text{V}/\text{LSB}$ 。范围为 $344\mu\text{V}/\text{LSB}$ 至 $375\mu\text{V}/\text{LSB}$ ，根据下面公式计算： $\text{GAIN} = 344\mu\text{V}/\text{LSB} + \text{ADCGAIN}<4:0> \times (1\mu\text{V}/\text{LSB})$ 。或者根据下面的转换表格

| 码 | ADC 增益 ($\mu\text{V}/\text{LSB}$) | 码 | ADC 增益 ($\mu\text{V}/\text{LSB}$) |
|------|-------------------------------------|------|-------------------------------------|
| 0x00 | 344 | 0x10 | 360 |
| 0x01 | 345 | 0x11 | 361 |
| 0x02 | 346 | 0x12 | 362 |
| 0x03 | 347 | 0x13 | 363 |
| 0x04 | 348 | 0x14 | 364 |
| 0x05 | 349 | 0x15 | 365 |
| 0x06 | 350 | 0x16 | 366 |
| 0x07 | 351 | 0x17 | 367 |
| 0x08 | 352 | 0x18 | 368 |
| 0x09 | 353 | 0x19 | 369 |
| 0x0A | 354 | 0x1A | 370 |
| 0x0B | 355 | 0x1B | 371 |
| 0x0C | 356 | 0x1C | 372 |
| 0x0D | 357 | 0x1D | 373 |
| 0x0E | 358 | 0x1E | 374 |
| 0x0F | 359 | 0x1F | 375 |

1.20 ADC 失调寄存器(0x51h)

| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
|----------------|------|------|------|------|------|------|------|
| ADCOFFSET<7:0> | | | | | | | |
| - | - | - | - | - | - | - | - |
| r | r | r | r | r | r | r | r |

ADCOFFSET<7:0> - ADC 失调，以 2 的补码格式存储，单位为 mV。正满量程对应 0x7Fh，负满量程对应 0x80h。满量程输入范围为-128mV 至 127mV，LSB 为 1mV。

| 码 | 失调(mV) |
|------|--------|
| 0x00 | 0 |
| 0x01 | 1 |
| 0x7F | 127 |
| 0x80 | -128 |
| 0x81 | -127 |
| 0xFF | -1 |

配置串联电池节数

每个 99x0 IC 系列支持多种电池方案。下表根据电池组中的电池数量提供了使用哪种芯片和哪种输入引脚的方案。

表 3. 9920 的电池连接

| 电池输入 | 3 节电池 | 4 节电池 | 5 节电池 |
|-----------|-------|-------|-------|
| VC5 - VC4 | 第 3 节 | 第 4 节 | 第 5 节 |
| VC4 - VC3 | 短路 | 短路 | 第 4 节 |
| VC3 - VC2 | 短路 | 第 3 节 | 第 3 节 |
| VC2 - VC1 | 第 2 节 | 第 2 节 | 第 2 节 |
| VC1 - VC0 | 第 1 节 | 第 1 节 | 第 1 节 |

表 4. 9930 的电池连接

| 电池输入 | 6 节电池 | 7 节电池 | 8 节电池 | 9 节电池 | 10 节电池 |
|------------|-------|-------|-------|-------|--------|
| VC10 - VC9 | 第 6 节 | 第 7 节 | 第 8 节 | 第 9 节 | 第 10 节 |
| VC9 - VC8 | 短路 | 短路 | 短路 | 短路 | 第 9 节 |
| VC8 - VC7 | 短路 | 短路 | 第 7 节 | 第 8 节 | 第 8 节 |
| VC7 - VC6 | 第 5 节 | 第 6 节 | 第 6 节 | 第 7 节 | 第 7 节 |
| VC6 - VC5b | 第 4 节 | 第 5 节 | 第 5 节 | 第 6 节 | 第 6 节 |
| VC5 - VC4 | 第 3 节 | 第 4 节 | 第 4 节 | 第 5 节 | 第 5 节 |
| VC4 - VC3 | 短路 | 短路 | 短路 | 第 4 节 | 第 4 节 |
| VC3 - VC2 | 短路 | 第 3 节 | 第 3 节 | 第 3 节 | 第 3 节 |
| VC2 - VC1 | 第 2 节 | 第 2 节 | 第 2 节 | 第 2 节 | 第 2 节 |
| VC1 - VC0 | 第 1 节 | 第 1 节 | 第 1 节 | 第 1 节 | 第 1 节 |

表 5. 9940 的电池连接

| 电池输入 | 11 节 电池 | 12 节 电池 | 13 节 电池 | 14 节 电池 | 15 节 电池 |
|------------|------------|------------|------------|------------|------------|
| VC15-VC14 | 第 11 节 | 第 12 节 | 第 13 节 | 第 14 节 | 第 15 节 |
| VC14-VC13 | 短路 | 短路 | 短路 | 短路 | 第 14 节 |
| VC13-VC12 | 短路 | 第 11 节 | 第 12 节 | 第 13 节 | 第 13 节 |
| VC12-VC11 | 第 10 节 | 第 10 节 | 第 11 节 | 第 12 节 | 第 12 节 |
| VC11-VC10b | 第 9 节 | 第 9 节 | 第 10 节 | 第 11 节 | 第 11 节 |
| VC10-VC9 | 第 8 节 | 第 8 节 | 第 9 节 | 第 10 节 | 第 10 节 |

| 电池输入 | 11 节 电池 | 12 节 电池 | 13 节 电池 | 14 节 电池 | 15 节 电池 |
|----------|------------|------------|------------|------------|------------|
| VC9-VC8 | 短路 | 短路 | 短路 | 第 9 节 | 第 9 节 |
| VC8-VC7 | 第 7 节 | 第 7 节 | 第 8 节 | 第 8 节 | 第 8 节 |
| VC7-VC6 | 第 6 节 | 第 6 节 | 第 7 节 | 第 7 节 | 第 7 节 |
| VC6-VC5b | 第 5 节 | 第 5 节 | 第 6 节 | 第 6 节 | 第 6 节 |
| VC5-VC4 | 第 4 节 | 第 4 节 | 第 5 节 | 第 5 节 | 第 5 节 |
| VC4-VC3 | 短路 | 短路 | 第 4 节 | 第 4 节 | 第 4 节 |
| VC3-VC2 | 第 3 节 | 第 3 节 | 第 3 节 | 第 3 节 | 第 3 节 |
| VC2-VC1 | 第 2 节 | 第 2 节 | 第 2 节 | 第 2 节 | 第 2 节 |
| VC1-VC0 | 第 1 节 | 第 1 节 | 第 1 节 | 第 1 节 | 第 1 节 |

1.1 设计要求

表 6. 99x0 设计要求

| 设计参数 | TA=25°C 时的示例值 |
|------------|---------------|
| 最小系统工作电压 | 24 V |
| 电池最低工作电压 | 3.0 V |
| 串联电池数量 | 8 |
| 充电电压 | 33.6 V |
| 最大充电电流 | 3.0 A |
| 峰值放电电流 | 10.0 A |
| 过压保护阈值 | 4.30 V |
| 过压保护延迟 | 2s |
| 欠压保护阈值 | 2.5 V |
| 欠压保护延迟 | 4 s |
| 最大过流保护阈值 | 15 A |
| 过流保护延迟时间 | 320 ms |
| 最大短路保护电流阈值 | 25 A |
| 短路保护延迟时间 | 100 μ s |

1.2 详细设计步骤

要开始设计过程，需要一些关键步骤来选择组件和配置保护。

1.2.1 分步设计程序

- 确定串联电池的数量。
 - 该值取决于电池化学和系统的负载要求。例如，要使用电池最低电压为 3.0V 的 Li-CO₂ 型电池支持 24 V 的最低电池电压，至少需要 8 节电池。
- 选择正确的 99x0 器件。
 - 对于 8 节电池，需要 9930。
 - 有关正确的电池连接，请参见表 4。
- 选择正确的保护场效应管 FET（CHG 和 DSG 驱动的 FET）。
 - 9930 使用适用于 N 沟道场效应管(FET)的低压侧驱动器。
 - 这些 FET 的最大额定值应为：
 - 每个串联电池的电压应约为直流 5 V，峰值 10 V。
 - 电流，应根据最大直流电流和具有一定冗余的最大瞬态电流进行计算：例如，30 A。
 - 功耗，这是选择 FET 以及设计 PCB 时需要考虑的因素，假如所选 FET 的导通电阻是 5mΩ，那么最大 30A 的电流对应的功耗是 4.5W，所选 FET 额定功率不能低于该值。
- 选择正确的感测电阻器。
 - 应选择电阻值，以最大限度地利用库仑计数器量程 CC_{RANGE} 的输入范围，但不得超过绝对最大额定值。
 - 使用正常最大放电电流， $R_{SNS}=200\text{mV}/10.0\text{ A}=20\text{m}\Omega$ 。
 - 但是，考虑到 I_{SCD} 为 25 A，假设最大输入 SRP-SRN =-300mV， $R_{SNS} = 300\text{mV}/25\text{ A}=7.5\text{m}\Omega$ 。
 - 还应考虑系统的最大工作电流，因为其应低于最大 OCD 阈值， R_{SNS} 为 7.5mΩ 时，最大 OCD 电流设置为 13.3 A。
 - 还应考虑进一步容差分析（温度变化等）和 PCB 设计裕度，因此 5mΩ 的 R_{SNS} 适用于 75ppm 的温度系数和 5 W 的额定功率。
- 选择 9930，因此 REGSRC 引脚需要通过源极跟随器电路供电，其中 FET 用于从蓄电池正极向 REGSRC 提供电流，同时将电压降低至 IC 的适当值。
 - FET 也会耗散 IC 外部负载电流和压降产生的功率，应注意确保所选 FET 指定了正确的耗散额定值。
- 通过 PROTECT1 和 PROTECT2 配置基于当前的保护设置：
 - 理想 SCD 阈值= $25\text{A}\times 5\text{m}\Omega=125\text{mV}$ 。
 - 然而，最接近的选项是分别提供 22.2 A 和 26.6 A 的 111mV(0x03)和 133mV(0x04)。两个选项的 RSNS 位均为 1。
 - 0x03 (22.2 A)将在本例中使用。
 - 100μs 延迟的 SCD 延迟阈值设置为 0x01。
 - 因此，应使用 0x8C 对 PROTECT1 进行编程。

- 理想 OCD 阈值= $15\text{A}\times 5\text{m}\Omega=75\text{mV}$ 。
- 然而,最接近的选项是 $72\text{mV}(0x0A)$ 和 $78\text{mV}(0x0B)$, 分别提供 14.4A 和 15.6A 。两个选项的 RSNS 位均为 1。
- $0x0A(14,4\text{A})$ 将在本例中使用。
- 320 毫秒延迟的 OCD 延迟阈值设置为 $0x05$, 因此, 应使用 $0x5B$ 对 PROTECT2 进行编程。

注意:

确定 OV_TRIP 和 UV_TRIP 的设置时应小心, 因为这是 ADC 输出, 与电池电压的相关性也需要考虑 ADC 增益和 ADC 偏移寄存器。更多具体细节可在保护子系统找到。

- 通过 OV_TRIP 、 UV_TRIP 和 PROTECT3 配置基于电压的保护设置:
 - 选择的 OV 阈值为 4.30V 。
 - 因此, OV_TRIP 应使用 $0xC9$ 编程。
 - 选择的 UV 阈值为 2.5V 。
 - 因此, UV_TRIP 应使用 $0x1A$ 编程。
 - 选择的 OV 延迟为 2 秒, 选择的 UV 延迟为 4 秒, 因此, 应使用 $0x50$ 对 PROTECT3 进行编程。