

Application Note

使用 BQ25856-Q1 测量超级电容器的运行状况



Ethan Galloway, Michael Bradbourne

摘要

与常规电容器或电池相比，超级电容器可提供更高的功率密度。高功率密度与低 ESR 相结合，意味着超级电容器正变得越来越广泛。超级电容器用于为车载计算机、车门提供备用电源，或为服务器提供辅助电源。但是，超级电容器的电容会降低，等效串联电阻 (ESR) 会随着时间的推移而增加。了解电容器的电流 ESR 和电容，对于确保这些电容器能够在需要时正常工作至关重要。BQ2575X 和 BQ2585X-Q1 IC 系列可与微控制器配合使用，提供关于超级电容器的 ESR 和电容的近似测量值，以帮助估算电容器的使用寿命或 (EOL)。这些 IC 还可以限制充电和放电电流并精确控制输出电压，从而帮助延长超级电容器的使用寿命。本应用手册提供了概要和示例代码，说明了如何使用 BQ25856-Q1 和 MSPM0L2228 MCU 粗略测量超级电容器的 ESR 和电容。

内容

1 简介.....	2
2 如何延长超级电容器的使用寿命.....	3
2.1 电压的影响.....	3
2.2 电流的影响.....	3
2.3 温度的影响.....	3
3 超级电容器的 ESR 和电容.....	4
3.1 估算电容器的电容和 ESR.....	4
3.2 使用 BQ25856-Q1 测量超级电容器的运行状况.....	6
3.3 估算反向模式下的电容和 ESR.....	8
4 总结.....	9
5 附录 - BQ25858-Q1 的 MCU 代码示例.....	10
6 参考资料.....	14

商标

所有商标均为其各自所有者的财产。

1 简介

超级电容器作为备用系统变得越来越普遍。超级电容器可以为医疗设备、车门和车载计算机等终端设备提供备用电源。超级电容器还可以为服务器提供辅助电源。

与电池技术相比，超级电容器可提供更长的使用寿命和功率密度；但是，仍需要考虑老化的影响。随着超级电容器的使用年限增加，ESR 也会增加，而电容会减小。这两个因素共同作用，可降低超级电容器的总可用能量和功率。

这些电容器在实际应用中需要保障可靠且使用寿命长。本应用手册介绍了如何快速测量超级电容器的电容和 ESR。借助 ESR 和电容信息，可以在电容器达到使用寿命 (EOL) 之前更换超级电容器。

2 如何延长超级电容器的使用寿命

超级电容器的功能寿命取决于电压、充电电流、机械应力和工作温度。许多超级电容器制造商确定，当电容降至标称值的 80% 或等效串联电阻 (ESR) 升至标称值的 200% 时，其电容器就达到了 EOL。请咨询每个超级电容器的制造商，以确定超级电容器何时达到 EOL。

本应用手册主要介绍如何测量超级电容器的 ESR 和电容。尽管如此，仍存在若干会降低超级电容器整体使用寿命的因素，后文将介绍这些因素，并说明如何通过 BQ2585X-Q1 延缓其寿命衰减。

2.1 电压的影响

超级电容器若出现过压情况，会缩短超级电容器的总寿命。将超级电容器充电至较低的电压有助于延长超级电容器的工作寿命。BQ2575X 和 BQ2585X-Q1 充电控制器在反馈电阻上具有 0.5% 的电压精度，可将超级电容器精确充电至建议的满电电压。

2.2 电流的影响

大量电流灌入或拉取超级电容器会导致内部发热。灌电流或拉电流若超过超级电容器的规格，会缩短器件的使用寿命。BQ2575X 和 BQ2585X-Q1 系列具有放电电流限制，可通过 IBAT_REV 寄存器限制超级电容器的总拉电流。BQ2575X 和 BQ2585X-Q1 系列还具有充电电流设置，可准确调节进入超级电容器的电流。

2.3 温度的影响

超级电容器的使用寿命随着温度的升高而显著缩短。请咨询超级电容器制造商，了解超级电容器的温度要求。BQ2575X 和 BQ2585X-Q1 符合 JEITA 标准，可以与热敏电阻配合使用，以在超级电容器变热时降低充电电流或充电电压。

3 超级电容器的 ESR 和电容

3.1 估算电容器的电容和 ESR

超级电容器和电容器可以建模为电阻器和电容器，如 图 3-1 所示。

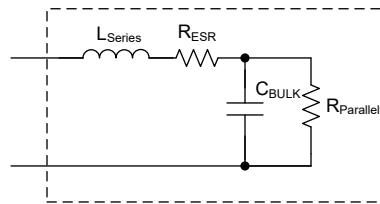


图 3-1. 电容器模型

由于电容和 ESR 是超级电容器运行状况的决定性特征，并且会随着时间的推移而变化，因此本应用手册重点介绍了如何测量这些值。

电容器中电流的定义公式如 方程式 1 所示：

$$i = C \frac{dV}{dt} \quad (1)$$

BQ2585X-Q1 和 BQ2575X 系列器件可以用作恒流源。使用恒流源意味着电压随时间的变化是线性的。方程式 1 可以重新排列成 方程式 2，以计算电容：BQ2585X-Q1 IC 可以通过板载 ADC 测量电压变化。测量充电开始后的初始电压和禁用充电之前的最终电压。可根据 方程式 3 确定电压变化：

$$C = \frac{i \Delta t}{\Delta V} \quad (2)$$

$$\Delta V = V_{Final} - V_{Initial} \quad (3)$$

可根据 方程式 4 计算电容：

$$C = \frac{i \Delta t}{V_{Final} - V_{Initial}} \quad (4)$$

通过使用已知电流来测量已知时间内的电压变化，上述公式可以计算出超级电容器的电容。可以跟踪电容随时间的变化，以确定电容器是否即将达到 EOL。

还需跟踪 ESR 随时间的变化。电容器 ESR 上的压降由欧姆定律定义，而电容器 ESR 的电阻可通过 \ 求解：

$$R_{ESR} = \frac{V_{ESR}}{i} \quad (5)$$

当器件提供电流为超级电容器充电时，从电容器测得的电压是两个电压之和：电容器电压和电容器 ESR 两端的压降 (V_{ESR})。可以在无灌电流或拉电流的情况下测量该电容器，以使 V_{ESR} 抵消。在这里，最终电压定义为充电结束前的电压。最终电压是超级电容器上无电流时的电压。最终电压没有电流；因此，ESR 上没有电压。

减去两个电压测量值会得到超级电容器 ESR 两端的压降：

$$V_{Final} - V_{Ending} = (V_{ESR} + V_{CAP}) - (V_{CAP}) = V_{ESR} \quad (6)$$

将 方程式 6 中的结果代入 方程式 5 并了解充电电流设置，可以确定超级电容器的 ESR。也可以在超级电容器的寿命内跟踪 ESR，以估算超级电容器是否已达到其使用寿命。

为了确保电流保持恒定而电容器在测试期间不会充满电，可使用以下公式计算测试后电容器的最终电压。

$$V_{Ending} = \frac{1}{C} \int_0^t i \, dt + V_{Start} \quad (7)$$

由于充电电流 i 是一个常数值，[方程式 7](#) 可重新排列为：

$$V_{Final} = \frac{i\Delta t}{C} + V_{Start} \tag{8}$$

此公式可用于确保 BQ2585X-Q1 IC 保持恒流模式，并且测试不会使电容器完全充电。BQ2585X-Q1 还具有状态寄存器，用于显示 IC 是恒压模式还是恒流模式。测试后可以读取状态寄存器，以确保 IC 保持在恒流模式。

以下流程图显示了如何使用 BQ25856-Q1 测量超级电容器的运行状况：

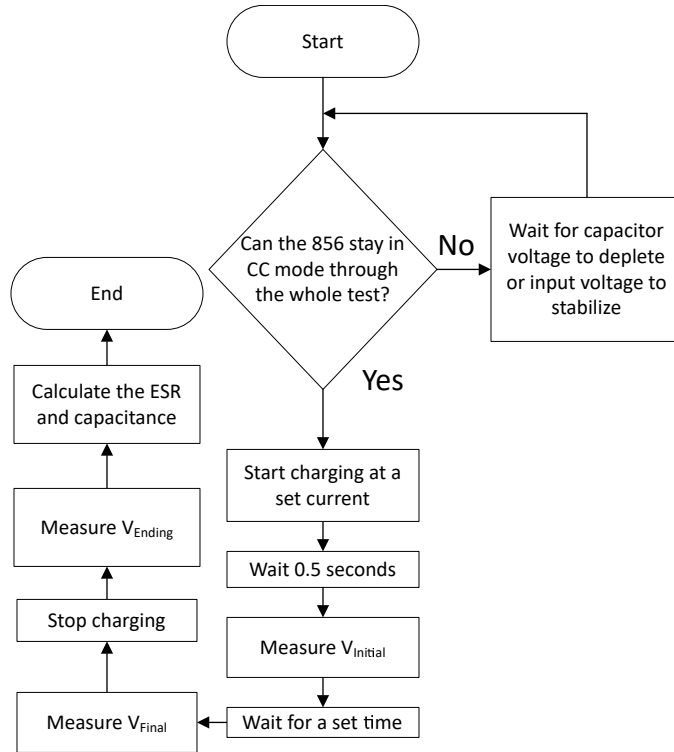


图 3-2. 超级电容器运行状况测量流程图

3.2 使用 BQ25856-Q1 测量超级电容器的运行状况

在以下示例中，BQ25856-Q1 IC 与 MSPM0L2228 MCU 一起测量八个串联超级电容器的电容和 ESR。BQ25856-Q1 与 MCU 主机控制器可以通过以下步骤测量 ESR 和电容：

1. 测量电容器的电压
2. 计算总充电时间和充电电流，以确保 BQ25856-Q1 在整个充电过程中保持恒流模式
3. 开始对电容器充电
4. 充电开始后测量 $V_{Initial}$
5. 充电结束前测量 V_{Ending}
6. 停止为电容器充电
7. 当没有充电电流时，测量 V_{Final}
8. 使用 $V_{Initial}$ 、 V_{Ending} 和两次测量之间的时间来计算电容
9. 使用充电电流、 V_{Ending} 和 V_{Final} 计算 ESR。

以下测试结果是在下列条件下测试得出的：

- $V_{AC} = 10V$
- $I_{CHG} = 1A$
- $t_{pulse} = 1s$
- $C_{OUT_nominal} = 10F \times 8S1P$
- $R_{ESR_nominal} = 35 m\Omega \times 8S1P$

在图 3-3 中，BQ25856-Q1 需要大约三秒的时间来为电容器充电以测量电容和 ESR。如果需要，可在更短的时间内完成测量。

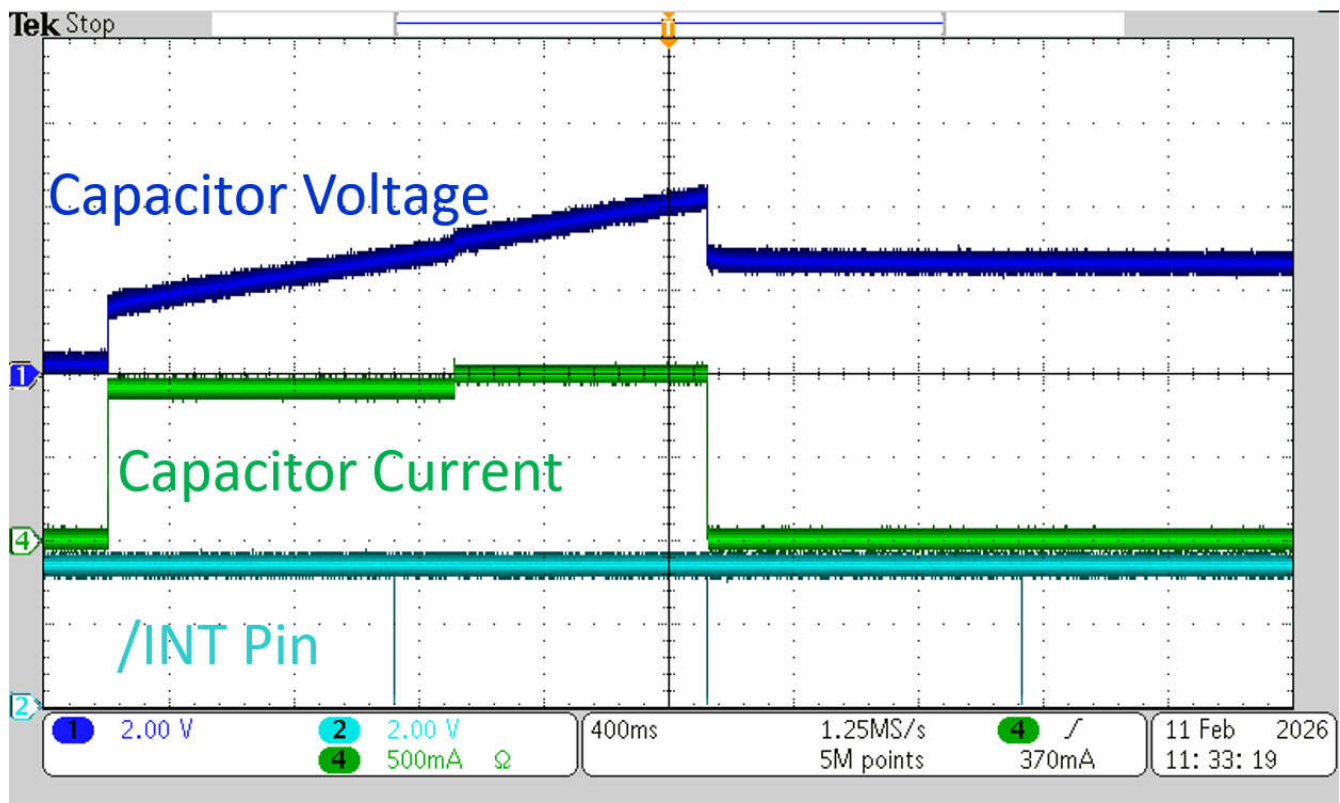


图 3-3. 示例测量波形

图 3-3 展示了一个演示测量代码实现方案的示例波形。器件开始使用在通道 4 上探测到的 I_{CHG} 电流为电容器充电。在通道 1 上探测 IC 两端的电容器电压。每次进行 ADC 测量时，器件的 /INT 引脚都会使中断置为有效。在通道 2 上监控 /INT 引脚，以演示代码时序。通道 2 上的第一个脉冲显示了用于确定 $V_{Initial}$ 的第一个 ADC 测量值。在此之后，计时器开始计时，充电会持续 1 秒。通道 2 上的第二个脉冲显示了用于确定 V_{Ending} 的第二个 ADC 测量值。在 ADC 测量之后会立即禁用充电，并且还有另一个等待期来使电容器电压稳定。通道 2 上的第三个脉冲显示了用于确定 V_{Ending} 的最后一个 ADC 测量值。使用 [方程式 5](#) 和 [方程式 4](#)，可以通过 BQ25856-Q1 的 ADC 测量值、BQ25856-Q1 设置的已知充电电流以及 MCU 计时器的已知充电时间来确定 ESR 和电容。

表 3-1. ADC 电压测量

ADC 读取值	电压 (V)
初始电压	2.586
峰值电压	4.000
最终电压	2.454

表 3-2. 超级电容器参数计算

参数	结果
八个电容器串联在一起	0.71F
八个串联电容器的 ESR	1.55 Ω
每个电容器的平均电容	5.68F
每个电容器的平均 ESR	194m Ω

测量完成后，整个充电周期可以恢复，直到电容器充电并准备好提供备用电源。可以跟踪计算出的电容和 ESR 随时间的变化，以确定电容器何时达到使用寿命。在此测试中，电容器是旧的且磨损的，并且已经达到使用寿命，因为 ESR 远大于标称值，并且电容已降至原始值的 70%。当计算得出的电容降至标称值的 80% 以下或 ESR 超过标称值的 200% 时，表明电容器已达到 EOL。

3.3 估算反向模式下的电容和 ESR

只要满足某些条件，也可以在反向模式下使用 BQ25856-Q1 测量电容和 ESR。BQ25856-Q1 在反向工作模式下，需要在 VAC 端接入负载，以便从电容中泄放电流。此外，BQ25856-Q1 在反向模式下主要充当恒压源。这意味着超级电容器在向外供电时将没有线性电流。

有几种方法可以测量来自电容器的总电流。首先，如果系统能够测量来自电容器的平均源电流，则平均电流乘以时间可以替换为的积分 $\int_0^t i \, dt$ 。这将得出以库仑为单位的值。电容器与电荷的关系公式为：

$$C = \frac{q}{V} \quad (9)$$

电容器公式可以重新排列成：

$$\frac{\int_0^t i \, dt}{V_{Ending} - V_{Start}} = C \quad (10)$$

方程式 10 未考虑电容器 ESR 上的压降。TI 建议确保 V_{Ending} 和 V_{Start} 处的电容器电流近似相等。

可以通过测量两个不同电容器电流在已知时间之间的电压差来计算 ESR：

$$\frac{V_{C2} - V_{C1} - \frac{1}{C} \int_0^t i \, dt}{(i_2 - i_1)} = R_{ESR} \quad (11)$$

在 **方程式 11** 中， V_{C1} 和 V_{C2} 是在 i_1 和 i_2 的电容器源电流下测得的电容器电压。 t 是 V_{C1} 和 V_{C2} 测量之间的时间间隔。可通过 **方程式 10** 计算电容 C 。

或者，如果 MCU 无法计算平均电流，则可以使用 IBAT_REV 寄存器将流出电容器的电流限制为 5A、10A、15A 或 20A。在电容器的源电流受限的情况下，电流将是线性的，可以使用 **节 3.1** 中的公式计算电容和 ESR。确保 VAC 上的负载加载足够大的输入，从而使 IBAT_REV 变为有效状态并限制放电电流。

4 总结

与电解电容器或电池相比，超级电容器具有很大的优势。与电池相比，超级电容器具有更高的功率密度和更长的使用寿命。此外，超级电容器可以存储比电解电容器更多的能量。但是，超级电容器仍会老化，可用功率和能量会随着时间的推移而降低。BQ2585X-Q1 和 BQ2575X 系列 IC 可以估算超级电容器的 ESR 和电容。这些信息有助于跟踪超级电容器的运行状况，并可用于估算电容器的使用寿命。

5 附录 - BQ25858-Q1 的 MCU 代码示例

```
extern volatile bool timer_exp;

int main(void)
{
    SYSCFG_DL_init(); // Initialize the I2C and the timer via SuperCapHealth.sysconfig

    __NVIC_EnableIRQ(I2C_0_INST_INT_IRQN); // Initialize the I2C interrupt routine for I2C communication

    __NVIC_EnableIRQ(TIMER_0_INST_INT_IRQN); // Initialize the timer interrupt routine.
    timer_exp = 0; // Initialize the timer flag.

    // I2C Settings
    uint8_t DIS_WD[1] = {0x05}; // Disable Watchdog timer
    uint8_t EN_CHG[1] = {0xC9}; // 0x09 for BQ25858-Q1
    uint8_t DIS_CHG[1] = {0xC8}; // 0x08 for BQ25858-Q1
    uint8_t DIS_PRECHG[1] = {0x06}; // Disable Precharge and Termination
    uint8_t EN_ADC[1] = {0xE0}; // One shot conversion mode to control when ADC Reads
    uint8_t VOUT_ADC[1] = {0xEF}; // Set only the VOUT ADC to read (shortens conversion time)
    uint8_t DIS_INT_MASK[3] = {0x7F, 0xFF, 0xFF}; // Set only the ADC interrupt

    // Current Parameter
    uint8_t CHG_CURRENT[2] = {0x50, 0x00}; // 1 A IOUT setting
    volatile uint16_t ICHG = 0x0000; // 16 bit variable for charge current
    volatile double_t current_conv = 0.05; // 50 mA per bit on BQ2585X-Q1 Register
    volatile double_t current = 0.0; // Current value for ESR and CAP calculations
    // Determine Charge Current in AMPERES
    ICHG = (((uint16_t)CHG_CURRENT[1] << 8) | (uint16_t)CHG_CURRENT[0]) >> 2;
    current = (double_t)ICHG * current_conv;

    // Voltage Parameter
    uint8_t VBAT[2] = {0x00, 0x00}; // Variable to read ADC Conversion
    volatile unsigned int adc_result0 = 0; // 16 bit variable for ADC Conversion result
    volatile unsigned int adc_result1 = 0;
    volatile unsigned int adc_result2 = 0;
    volatile double_t volt_conv = 0.002; // 2 mV per bit on BQ2585X-Q1 ADC
    volatile double_t voltage0 = 0; // Voltage value for ESR and CAP calculations
    volatile double_t voltage1 = 0;
    volatile double_t voltage2 = 0;

    // Time parameter
    volatile double_t charge_time = 1.0; // 1000 ms charge time

    //Results
    volatile double_t ESR = 0.0; // ESR Result
    volatile double_t CAP = 0.0; // Capacitance result
}
```

```
// Disable Watchdog to avoid registers being overwritten
I2C_Write(BQ2585X_TIMER_CONTROL, DIS_WD, 1);

// Disable Charge to set proper settings
I2C_Write(BQ2585X_CHARGER_CONTROL, DIS_CHG, 1);

// Set 1 A IOU for measurement calculations
I2C_Write(BQ2585X_CHARGE_CURRENT_LIMIT_LSB, CHG_CURRENT, 2);

// Disable Precharge and Termination
I2C_Write(BQ2585X_PRECHARGE_AND_TERMINATION_CONTROL, DIS_PRECHG, 1);

// Set only VOUT_ADC for fast ADC operation
I2C_Write(BQ2585X_ADC_CHANNEL_CONTROL, VOUT_ADC, 1);

// Set only ADC interrupts
I2C_Write(BQ2585X_MASK_1, DIS_INT_MASK, 3);

while (1) {
    // Enable the charge sequence
    I2C_Write(BQ2585X_CHARGER_CONTROL, EN_CHG, 1);
    // Start timer to remove EN_CHG delay
    DL_TimerG_startCounter(TIMER_0_INST);
    // Wait for the 1s timer to finish
    while (!timer_exp);
    // Reset timer flag for next timer cycle
    timer_exp = 0;

    // Perform ADC conversion for initial voltage
    I2C_Write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
    // Wait for ADC conversion to finish
    while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
    // Read ADC conversion results
    I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
    // Store ADC conversion results for later calculations
    adc_result0 = (VBAT[1] << 8) | (VBAT[0]);

    // Start the timer
    DL_TimerG_startCounter(TIMER_0_INST);
}
```

```

// Wait for the 1 s timer to finish
while (!timer_exp);

// Perform ADC conversion for max voltage
I2C_Write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
// Wait for ADC conversion to finish
while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
// Disable charge
I2C_Write(BQ2585X_CHARGER_CONTROL, DIS_CHG, 1);
// Read ADC conversion results
I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
// Store ADC conversion results for later calculations
adc_result1 = (VBAT[1] << 8) | VBAT[0];
// Reset timer flag for next timer cycle
timer_exp = 0;

// Start timer to let output voltage settle
DL_TimerG_startCounter(TIMER_0_INST);
// Wait for the 1 s timer to finish
while (!timer_exp);
// Reset timer flag for next timer cycle
timer_exp = 0;

// Perform ADC conversion for final voltage
I2C_Write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
// Wait for ADC conversion to finish
while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
// Read ADC conversion results
I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
// Store ADC conversion results for later calculations
adc_result2 = (VBAT[1] << 8) | (VBAT[0]);

// Convert ADC values to a voltage
voltage0 = (double_t)adc_result0 * volt_conv;
voltage1 = (double_t)adc_result1 * volt_conv;
voltage2 = (double_t)adc_result2 * volt_conv;

// Calculations for Capacitance and ESR

CAP = (current * charge_time) / (voltage1 - voltage0);
ESR = (voltage1 - voltage2) / current;
}
}

```

```

int main(void) {
    SYSCFG_DL_init(); // Initialize the I2C and the timer via SuperCapHealth.sysconfig
    __NVIC_EnableIRQ(I2C_0_INST_INT_IRQN); // Initialize the I2C interrupt routine for I2C
communication
    __NVIC_EnableIRQ(TIMER_0_INST_INT_IRQN); // Initialize the timer interrupt routine.
    timer_exp = 0; // Initialize the timer flag.
    // I2C Settings
    uint8_t DIS_WD[1] = {0x05}; // Disable watchdog timer
    uint8_t EN_CHG[1] = {0xc9}; // 0x09 for BQ25858-Q1
    uint8_t DIS_CHG[1] = {0xc8}; // 0x08 for BQ25858-Q1
    uint8_t DIS_PRECHG[1] = {0x06}; // Disable Precharge and Termination
    uint8_t EN_ADC[1] = {0xe0}; // One shot conversion mode to control when ADC Reads
    uint8_t VOUT_ADC[1] = {0xef}; // Set only the VOUT ADC to read (shortens conversion time)
    uint8_t DIS_INT_MASK[3] = {0x7f, 0xff, 0xff}; // Set only the ADC interrupt for the best results
    // Current Parameter
    uint8_t CHG_CURRENT[2] = {0x50, 0x00}; // 1 A IOUT setting
    volatile uint16_t ICHG = 0x0000; // 16 bit variable for charge current volatile
    double_t current_conv = 0.05; // 50 mA per bit on BQ2585X-Q1 Register
    volatile double_t current = 0.0; // Current value for ESR and CAP calculations
    // Determine Charge Current in AMPERES
    ICHG = (((uint16_t)CHG_CURRENT[1] << 8) | (uint16_t)CHG_CURRENT[0]) >> 2;
    current = (double_t)ICHG * current_conv;
    // Voltage Parameter
    uint8_t VBAT[2] = {0x00, 0x00};
    // Variable to read ADC Conversion
    volatile unsigned int adc_result0 = 0; // 16 bit variable for ADC Conversion result
    volatile unsigned int adc_result1 = 0;
}

```

```

volatile unsigned int adc_result2 = 0;
volatile double_t volt_conv = 0.002; // 2 mv per bit on BQ2585X-Q1 ADC
volatile double_t voltage0 = 0; // Voltage value for ESR and CAP calculations
volatile double_t voltage1 = 0;
volatile double_t voltage2 = 0;
// Time parameter
volatile double_t charge_time = 1.0; // 1000 ms charge time
//Results
volatile double_t ESR = 0.0; // ESR Result
volatile double_t CAP = 0.0; // Capacitance result
// Disable watchdog to avoid registers being overwritten
I2C_write(BQ2585X_TIMER_CONTROL, DIS_WD, 1);
// Disable Charge to set proper settings
I2C_write(BQ2585X_CHARGER_CONTROL, DIS_CHG, 1);
// Set 1 A IOUT for measurement calculations
I2C_write(BQ2585X_CHARGE_CURRENT_LIMIT_LSB, CHG_CURRENT, 2);
// Disable Precharge and Termination
I2C_write(BQ2585X_PRECHARGE_AND_TERMINATION_CONTROL, DIS_PRECHG, 1);
// Set only VOUT_ADC for best operation
I2C_write(BQ2585X_ADC_CHANNEL_CONTROL, VOUT_ADC, 1);
// Set only ADC interrupts for best operation
I2C_write(BQ2585X_MASK_1, DIS_INT_MASK, 3);
while (1) {
    // Enable the charge sequence
    I2C_write(BQ2585X_CHARGER_CONTROL, EN_CHG, 1);
    // Start timer to remove EN_CHG delay
    DL_TimerG_startCounter(TIMER_0_INST);
    // wait for the 1 s timer to finish
    while (!timer_exp);
    // Reset timer flag for next timer cycle
    timer_exp = 0;
    // Perform ADC conversion for initial voltage
    I2C_write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
    // wait for ADC conversion to finish
    while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
    // Read ADC conversion results
    I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
    // Store ADC conversion results for later calculations
    adc_result0 = (VBAT[1] << 8) | (VBAT[0]);
    // Start the timer
    DL_TimerG_startCounter(TIMER_0_INST);
    // wait for the 1 s timer to finish
    while (!timer_exp);
    // Perform ADC conversion for max voltage
    I2C_write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
    // wait for ADC conversion to finish
    while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
    // Disable charge
    I2C_write(BQ2585X_CHARGER_CONTROL, DIS_CHG, 1);
    // Read ADC conversion results
    I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
    // Store ADC conversion results for later calculations
    adc_result1 = (VBAT[1] << 8) | VBAT[0];
    // Reset timer flag for next timer cycle
    timer_exp = 0;
    // Start timer to let output voltage settle
    DL_TimerG_startCounter(TIMER_0_INST);
    // wait for the 1 s timer to finish
    while (!timer_exp);
    // Reset timer flag for next timer cycle
    timer_exp = 0;
    // Perform ADC conversion for final voltage
    I2C_write(BQ2585X_ADC_CONTROL, EN_ADC, 1);
    // wait for ADC conversion to finish
    while (DL_GPIO_readPins(BQ_INT_PORT, BQ_INT_PIN_0_PIN));
    // Read ADC conversion results
    I2C_Read(BQ2585X_VBAT_ADC_LSB, VBAT, 2);
    // Store ADC conversion results for later calculations
    adc_result2 = (VBAT[1] << 8) | (VBAT[0]);
    // Convert ADC values to a voltage
    voltage0 = (double_t)adc_result0 * volt_conv;
    voltage1 = (double_t)adc_result1 * volt_conv;
    voltage2 = (double_t)adc_result2 * volt_conv;
    // Calculations for Capacitance and ESR
    CAP = (current * charge_time) / (voltage1 - voltage0);
    ESR = (voltage1 - voltage2) / current;
}
}

```

6 参考资料

1. Wiley Advanced , “ *The Many Deaths of Supercapacitors: Degradation, Aging, and Performance Fading* ” , 网页
2. 德州仪器 (TI) , [BQ25856-Q1 : 车用独立/I2C 控制、1 节至 14 节电池、双向降压/升压电池充电控制器](#) , 数据表。
3. 德州仪器 (TI) , [BQ25856-Q1 电容器备用电路](#) , 应用手册。

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、与某特定用途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安全、安保法规或其他要求。

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。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 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。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月