



Mars Leung

摘要

本应用报告讨论了在嵌入式计量 (辅助计量) 应用中使用 MSP430i2040 的评估模块 (EVM) 设计。

内容

1 引言	4
1.1 安全性和预防措施	4
1.2 特性	4
2 设计详情	5
2.1 方框图	5
2.2 硬件设计	5
2.3 软件设计	9
3 校准技术	15
3.1 引言	15
3.2 校准技术	15
3.3 校准过程	16
4 硬件设置	20
4.1 EVM 的顶视图	20
4.2 EVM 的底视图	20
4.3 硬件设置过程	21
5 校准器软件	23
5.1 软件包内容	23
5.2 设置 PC 软件工具	23
5.3 仪器	25
6 操作 PC 软件工具	26
6.1 引言	26
6.2 开始使用 EVM	26
6.3 已知问题	31
7 串行通信命令	32
7.1 引言	32
7.2 通信协议	32
7.3 命令	33
8 固件和嵌入式计量库 API	43
8.1 引言	43
8.2 嵌入式计量库 API	43
9 示例应用程序代码	63
9.1 引言	63
9.2 准备要运行的应用程序代码	63
9.3 在没有 IAR 许可证的情况下下载	71
10 硬件设计文件	73
10.1 封装	73
10.2 原理图	74
11 EVM 规格和性能	75
11.1 EVM 规格	75
12 在 MSP430i2040 和 MSP430i2041 上运行	78
13 修订历史记录	82

插图清单

图 2-1. 系统方框图.....	5
图 2-2. 电压和电流传感器接口电路.....	6
图 2-3. 顶层布局.....	8
图 2-4. 底层布局.....	8
图 2-5. 系统功能图.....	12
图 2-6. 使用 SD24PRE 寄存器进行相位校正.....	13
图 2-7. 频率测量和周期跟踪.....	13
图 3-1. 前端接口模型.....	15
图 3-2. 校准设置.....	16
图 4-1. EVM 顶视图.....	20
图 4-2. EVM 底视图.....	20
图 5-1. “系统属性”窗口.....	24
图 5-2. “设备管理器”窗口.....	24
图 5-3. 编辑“calibration-config.xml”.....	25
图 6-1. 校准器软件启动窗口.....	27
图 6-2. 仪表状态窗口.....	28
图 6-3. “Meter calibration factors”窗口.....	29
图 6-4. “Meter features”窗口.....	30
图 6-5. “Meter error”窗口.....	31
图 7-1. 轮询模式数据帧格式.....	32
图 7-2. 轮询模式命令帧格式.....	33
图 7-3. 轮询模式响应帧格式.....	33
图 9-1. 打开工作区.....	64
图 9-2. 工程选项卡.....	65
图 9-3. 优化选项.....	65
图 9-4. 调试器选项.....	66
图 9-5. 下载选项.....	66
图 9-6. 编译应用程序.....	68
图 9-7. 警告.....	68
图 9-8. 连接 EVM 和 FET.....	69
图 9-9. 代码下载.....	70
图 9-10. 调试器屏幕.....	70
图 9-11. EVM 运行.....	71
图 9-12. 下载可执行文件.....	71
图 9-13. 选择要下载的文件.....	72
图 10-1. EVM 原理图.....	74
图 11-1. 功率百分比误差 % 负载电流 (A).....	76
图 11-2. 典型精度与温度间的关系.....	77
图 12-1. IAR 5.5 启动窗口.....	78
图 12-2. “Open Workspace”窗口.....	79
图 12-3. 工作区选项.....	79
图 12-4. “Options”窗口.....	80
图 12-5. 配置选项卡.....	80
图 12-6. 选择默认 XCL 文件.....	81
图 12-7. 完成设置更改.....	81
图 12-8. 保存设置更改.....	82

表格清单

表 6-1. LED 指示状态.....	26
表 7-1. HOST_CMD_GET_METER_NAME.....	33
表 7-2. HOST_CMD_GET_METER_VERSION.....	34
表 7-3. HOST_CMD_GET_METER_CONFIGURATION.....	34
表 7-4. 参数定义.....	35
表 7-5. HOST_CMD_GET_RTC.....	35
表 7-6. HOST_CMD_ALIGN_WITH_CALIBRATION_FACTORS.....	36
表 7-7. HOST_CMD_SET_PASSWORD.....	36

表 7-8. HOST_CMD_GET_READINGS_PHASE_N.....	37
表 7-9. HOST_CMD_GET_EXTRA_READINGS_PHASE_N.....	38
表 7-10. HOST_CMD_SUMCHECK_MEMORY.....	38
表 7-11. HOST_CMD_CLEAR_CALIBRATION_DATA.....	39
表 7-12. HOST_CMD_SET_CALIBRATION_PHASE_N.....	40
表 7-13. HOST_CMD_GET_CALIBRATION_PHASE_N.....	41
表 7-14. HOST_CMD_SET_CALIBRATION_EXTRAS.....	41
表 7-15. HOST_CMD_GET_CALIBRATION_EXTRAS.....	42
表 11-1. 220V、5A 下精度与温度间的关系.....	77

商标

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries, or both.

IAR Embedded Workbench® is a registered trademark of IAR Systems AB.

所有商标均为其各自所有者的财产。

1 引言

德州仪器 (TI) MSP430i2040 是一款超低功耗混合信号微控制器。该微控制器集成了四个独立的差分输入 24 位 Σ - Δ 模数转换器 (ADC)，带有一个可编程增益放大器、一个 16 位硬件乘法器，还包含一个支持通用异步接收器/发送器 (UART) 和串行外设接口 (SPI) 通信接口的 eUSCI_A0、一个支持 SPI 和集成电路总线 (I2C) 的 eUSCI_B0、两个 16 位计时器和 12 个通用输入/输出 GPIO 引脚 (采用 28 引脚 TSSOP 封装时) 或 16 个 GPIO 引脚 (采用 32 引脚 QFN 封装时)。外设集是用于嵌入式电能测量的良好组合。在此类应用领域中，电能测量器件嵌入在最终应用中，可提供有关电压、电流、功耗等的器件信息。此外，该 EVM 具有补偿线路电阻和 EMI 滤波器电容的功能。嵌入式计量 (辅助计量) 适用于许多领域，例如：

- 家用电器
- 服务器/PC 电源
- UPS
- 智能插头或电源板
- 光伏逆变器
- 电动汽车充电器
- 家庭监控、安全和自动化

本文档中的所有设置、操作、特性、行为、功能和接口说明均基于默认固件和原始 EVM 硬件设计。如果对硬件或固件进行了更改，则无法保证正常运行。本应用手册使用 [EVM430-12040S](#)。

备注

本文档引用了不再可用的旧电能测量库。但是，本文档中讨论的原则仍然适用。对于新设计，请考虑从 [MSP-EM-DESIGN-CENTER](#) 下载电能测量设计中心 (EMDC) 和软件库。

1.1 安全性和预防措施

本文档中讨论的设计应由接受过相应技术培训的专业人员操作。

WARNING

当电路板连接到火线时，可能会发生触电。该板应由专业人员小心处理。为安全起见，强烈建议使用具有过压和过流保护的隔离式设备。

1.2 特性

软件功能为：

- 测量均方根电压、均方根电流、有功功率、无功功率、视在功率、功率因数、交流频率、电压 THD、电流 THD、基波电压、基波电流、基波有功功率
- 读数每 4 个交流周期或 80ms (如果是直流输入) 更新一次
- 能够进行交流和直流测量
- 能够在交流和直流测量模式之间自动切换
- 能够进行 EMI 滤波电容器和导线电阻补偿
- 无需单独进行直流校准

2 设计详情

2.1 方框图

图 2-1 展示了该设计的系统方框图。来自交流电源的功率通过分流电阻器。分流电阻器上的电压被分接出来。交流电源电压由电阻梯分压为较小的一部分并被分接出来。然后这两个信号被传递到 MCU 上的两个 ADC 以供进一步处理。两个 SD24 ADC 以同步方式运行；硬件允许采样时间以精细可控的阶跃进行偏移。该功能用于校正由外部模拟电路引起的电压和电流信号之间的任何相移。

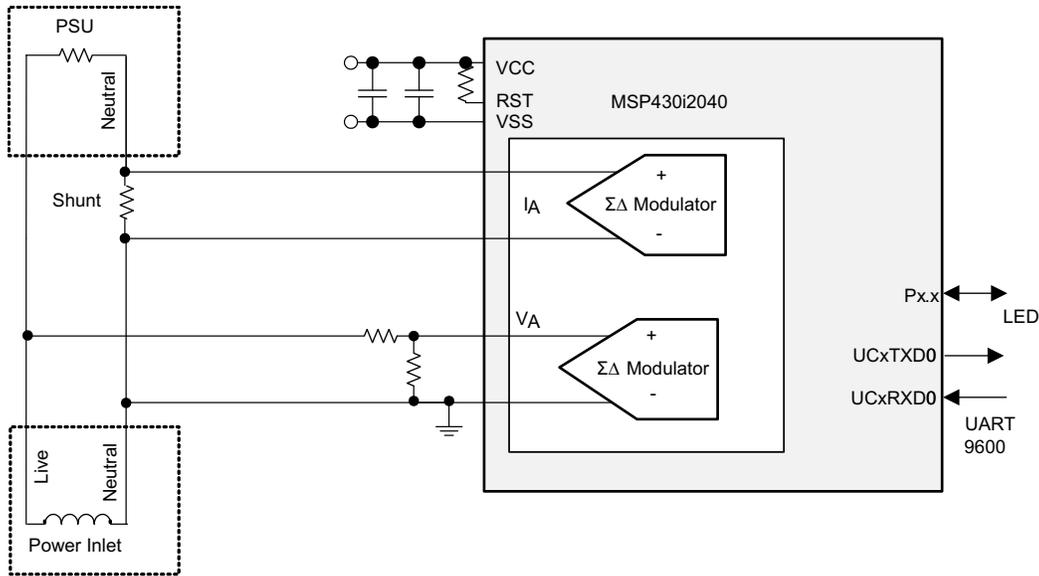


图 2-1. 系统方框图

2.2 硬件设计

2.2.1 接口电路

在计量电路的设计中，一个关键的成功因素是产生足够高的精度，这依赖于良好的采样电路。MSP430i2040 上的 24 位 Σ - Δ 模数转换器 (SD24) 与片上可编程增益放大器一起提供了从电流和电压传感器获得高质量模数转换结果的关键硬件。此外，接口电路、元件选择和电路布局对于成功的设计也起着至关重要的作用。

2.2.2

由于 Σ - Δ 模数转换器以及被测电压和电流信号的性质，需要一个外部接口电路来连接电压和电流，以使 SD24 ADC 保持正常工作并为相关频带提供适当的滤波。

接口电路的硬件很简单，除保护二极管 (D1、D8) 之外由无源器件组成。图 2-2 展示了设计的接口电路。顶部电路是电压传感器的接口，底部电路用于电流传感器 - 分流电阻器。

在电流传感器接口电路中，R6 为分流传感器，D1 为可选的保护二极管。除非使用高分流电阻值或电流互感器代替电流传感器，并且用户预计会出现明显的电流浪涌，从而导致传感器上的电压足够大而损坏 MCU，否则不需要使用保护二极管。

L1、R9 和 C9 构成了带宽为几 MHz 的低通滤波器。该滤波器的作用是抑制对 Σ - Δ ADC 的射频干扰。由 L2、R8 和 C8 构成的滤波器具有相同的特性，并针对差分信号进行了平衡。

由 R9、R8 和 C10 构成的滤波器是一个提供大约 10kHz 带宽的滤波器，它是适用于相关频带的滤波器。

务必注意应观察这些电容器的数量级；甚至电容的实际值也不是那么重要。

电压传感器接口是一个电阻梯。电阻梯中的电阻比允许测量高达交流 308V 和直流 420V 的电压。它与电流传感器滤波器非常相似，不同之处在于 R10 的值为 100 Ω ，而不是 1k Ω 。这是因为分压器电路具有与 R10 串联的等效源

电阻 $1.5k \times \frac{990k}{(1.5k + 990k)} \approx 1.5k$ 。为了针对差分信号平衡滤波器，使用了较小的 R10 值。组合效应的值与 R11 相似。尽管使用设计值存在轻微的不匹配，但对准确性没有明显影响。

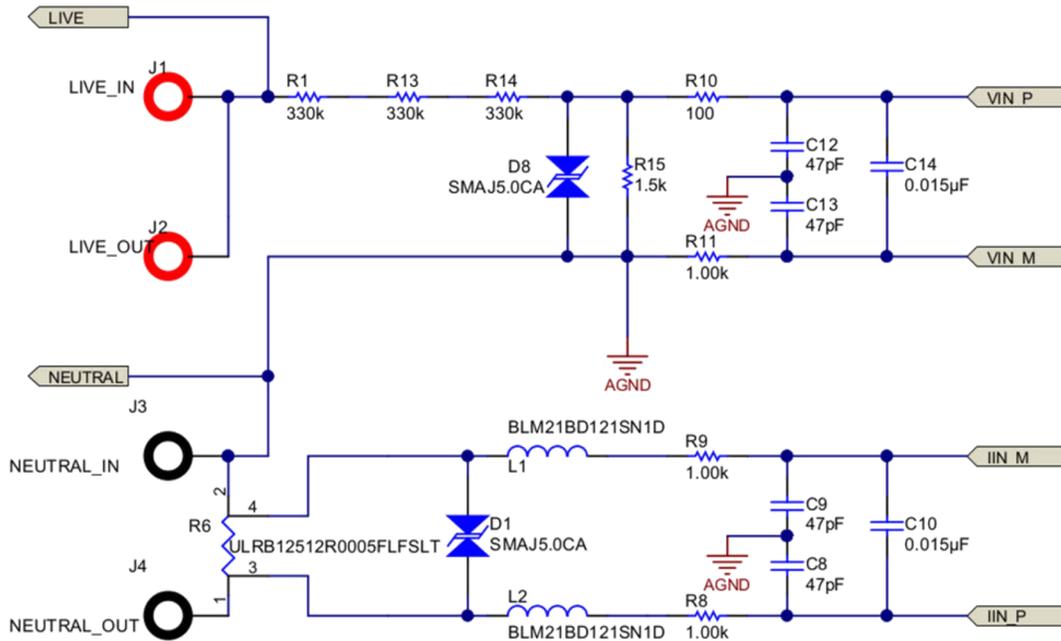


图 2-2. 电压和电流传感器接口电路

2.2.3 分流电阻器

为了实现精确测量，另一个重要因素是分流器的选择。一般而言，具有较小值的分流器是较好的选择。较小的分流器值有利于减小分流器上耗散的功率和相应的温度上升。较低的温度上升降低了对温度系数极低的分流电阻器（用于维持整个温度范围内的精度）的需求。然而，使用小值分流器也有缺点，即小值分流器的信噪比比大值分流器差。从这个意义上而言，可能需要值较高的分流器，例如在测量非常小的电流时，要测量的电流范围也处于一个小范围内。本设计考虑了几个因素：最大电流、电流动态范围、功率耗散。

在该应用中，每个插座的最大电流为 15A，所需的电流保持平坦误差百分比动态范围为 1000:1。分流电阻器上耗散的功率最好尽可能小。

为实现最佳精度，在测量最大电流时，希望提供给 SD24 ADC 的电流信号具有最大摆幅，因为 SD24 ADC 在 1 倍增益时的输入范围约为 900mV。使用 16 倍增益时，交流均方根的输入范围为：

$$\frac{\left(\frac{900}{16}\right)}{\sqrt{2}} = 39.77 \text{ mV}_{rms} \quad (1)$$

对于该数字，2mΩ 分流电阻应该适合此输入范围。然而，由于连接到分流电阻器的覆铜面积有限，因此最好使用值较小的分流器以使分流电阻器上的热积累较低；因此，选择了 0.5mΩ 的分流电阻。

除值之外，分流器的物理尺寸也是影响精度的一个重要因素。不是尺寸本身，而是电流流动时产生的热量。由于表面积有限，尺寸较小的分流器更容易升温。如果有很大的电流流过分流器，则建议使用尺寸足够大的分流器，并且 PCB 上也应该有足够的通风以防止热量积聚。在该设计中，选择了尺寸为 2512 的分流器。在阻值为 0.5mΩ 时，分流器在最大电流下的功率耗散为 $16^2 \times 0.5 = 0.128\text{W}$ ，约为 2512 尺寸 1W 分流电阻器额定功率的 1/8。

2.2.4 分压器电路

对于电压传感器电路，事情稍微简单一些。在该设计中，主要考虑因素是分压器应能够防止端子之间产生电弧。为了减小电阻器端子之间的电压差，该设计使用 4 个（而不是 2 个）电阻器构成分压器（R1、R13、R14 为 330kΩ，R15 为 1.5kΩ），因此对于 R1、R13 和 R14，每个电阻器两端获取大约 1/3 的交流电压。现在，在 220V 输入下，每个电阻器的压降约为 75V。R1、R13、R14 使用 2512 电阻器可在带电电压和 R15 之间提供更大的等效隔离距离。

在电压传感器电路中安装 D8 比在电流电路中安装 D1 更重要。原因是电压传感器电路上的电阻梯更有可能在 R15 上产生足以损坏 MCU 的电压浪涌。在其他情况下，如果 R15 开路或接触不良导致电阻更高，则电压通道 ADC 输入端的电压也会导致 MCU 损坏。

2.2.5 布局注意事项

虽然外部电路的电路设计非常简单，但在 PCB 布局和元件放置中有几点需要注意，以实现最佳精度。图 2-3 和图 2-4 分别展示了顶层和底层的布局。对于布局和放置，要记住的原则很简单：

- 关键电路是两组接口滤波器电路。
- 从传感器传输到 SD24 输入的信号在任何情况下都是差分信号。
- 将模拟接地和数字接地分开，在电源接地处重新连接在一起。
- 尽可能在模拟信号周围设置接地平面，以防止干扰。

现在回到布局，请注意图 2-3 中绿色矩形内是电压传感器和分流传感器的接口电路。该电路（包括用于电压传感器的 R10、R11、C12、C13、C14 以及 R8、R9、C8、C9 和 C10）的布置应与原理图上绘制的方式类似。这样做有助于在信号通过接口滤波器电路时保护其差分特性。

图 2-3 和图 2-4 中的黄色矩形内部显示离开接口滤波器的差分信号对应紧密并排布置并在它们一侧被模拟接地平面环绕（当然，这并不总是可能的）。

尽管 R15、R6、L1 有一个端子实质上连接到 AGND，但应将每个端子视为通过单点连接到 AGND 的单独信号（图 2-4 中以青色圈出的部分显示 R15 仅通过单条线路连接到 AGND）。实际上，这种布局方式也适用于 C12、C13、C8、C9 的连接。这有助于电路保持独立（电压信号和电流信号都没有干扰对方的路径）并且不受接地平面中流动的噪声的影响。

接地平面的设计使其环绕了大部分模拟电路。使用 i2040 很容易做到这一点，因为模拟信号引脚在封装的一侧组合在一起。此外，接地平面上增加了许多过孔，使顶部和底部接地平面紧密地连接在一起，从而提供更强的噪声保护。

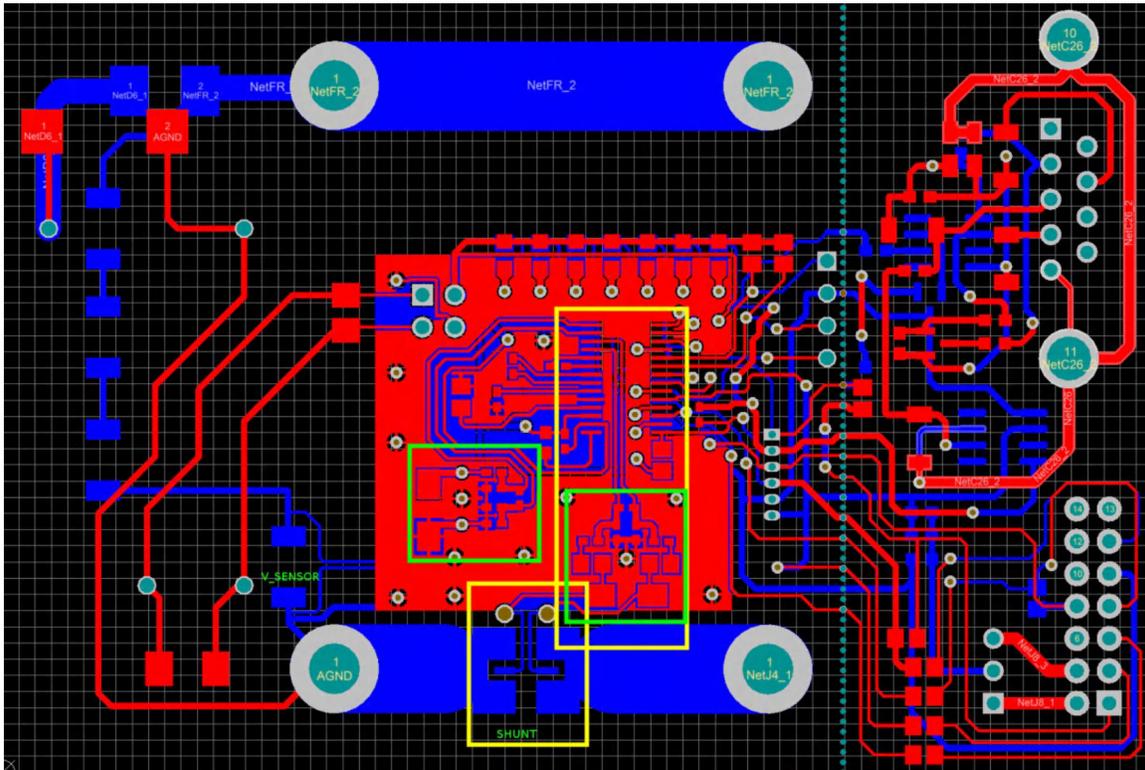


图 2-3. 顶层布局

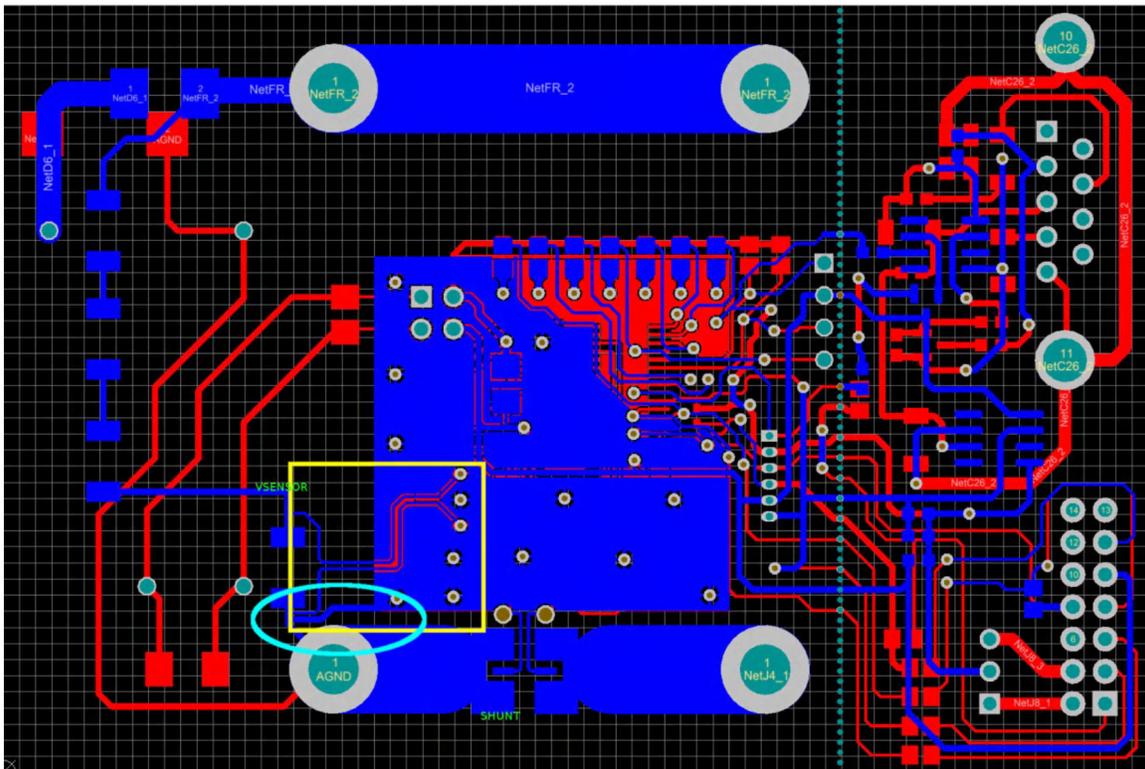


图 2-4. 底层布局

2.2.6 分流传感器焊盘设计

分流传感器焊盘的设计围绕两个原则进行：

- 大焊盘有助于散发分流传感器上产生的热量
- 分接点不受电流影响

在图 2-3 中，分流传感器标记为黄色矩形。在采用如图所示的设计形状的情况下，电流路径强制进入外部“[”“]”形状。分接焊盘（“□”形状内的小焊盘）与分流器的接触面积积极小，预计在该点没有电流流动。这有助于分接分流电压并消除焊料和覆铜的影响。

2.3 软件设计

EVM 随附的固件旨在使用分层方法来隔离计量细节和所涉及的相关计算，从而简化编程工作。

固件分为三个主要块。

- 该应用程序包含：
 - 系统复位和初始化
 - 主循环
 - 通信协议和命令处理
 - 非易失性参数预设和操作
- 打包到名为“emeter-metrology-i2041.r43”的库中的计量计算引擎包含：

备注

该库的源代码不包含在包中，如果需要源代码，请联系您所在地区的德州仪器 (TI) 销售团队。

- ADC 设置
- 参数初始化
- 基于样本的后台处理
- 基于报告周期的前台处理
- 读取应用程序接口
- 打包到名为“emeter-toolkit-i2041.r43”的库中的工具套件包含低级计算例程。

2.3.1 工具套件包

低级计算引擎为计量中的常见算术运算提供速度优化的处理。

- 功能包括：
 - 48 位累加
 - 24 位高通滤波直流偏移消除 (直流模式)
 - 16 位高通滤波直流偏移消除 (直流模式)
 - 基准纯正弦波生成
 - 48 位除以 16 位的除法
 - 16 位乘以 16 位、32 位乘以 16 位的乘法
 - 16 位、32 位、64 位平方根，整数平方根
 - 16 位乘以 16 位乘法累加成 48 位累加器
 - 16 位乘以 24 位乘法累加成 64 位累加器
 - Q1.15 定点数乘法
 - 16 位平方并累加到 48 位累加器
 - 24 位平方并累加到 64 位累加器

2.3.2 计量计算引擎

本设计使用 MCU 上的三个 Σ - Δ ADC：一个用于电压测量，一个用于电流测量，另一个连接到内置温度传感器进行温度测量。 Σ - Δ ADC 在硬件固定的大约 1.024MHz 调制器频率下运行，采样频率是通过将调制器频率除以 128 等于 8kHz 得出的，这为我们提供了 4kHz 的奈奎斯特带宽。该带宽足以覆盖 60Hz 交流的 66 次谐波和 50Hz 交流频率的 80 次谐波，因为由于被监控的开关电源的性质，服务器电源监控通常需要更宽的带宽。

后台处理在一个时间关键型循环中执行其操作，该循环由 ADC 以采样率触发。它在 ADC 的中断服务例程中运行并自动处理。它根据从电压和电流 ADC 通道收集的信息执行实际的样本计算。将时间要求不高的进程移交给前台处理。

前台处理是通过以报告和更新速率完成后台处理来触发的。后台进程在变量 `phase_state` 中设置标志 `PHASE_STATUS_NEW_LOG` 以指示有数据准备好由前台进程处理。然后，应用程序需要通过调用 `calculate_phase_readings ()` 来监视该标志以触发前台进程。

在实际计算中，计量计算采用以下公式：

$$V_{RMS} = VGAIN \times \sqrt{\frac{1}{N} \sum_{i=1}^N V_{samp}(i) \times V_{samp}(i)} \quad (2)$$

$$I_{RMS} = IGAIN \times \sqrt{\frac{1}{N} \sum_{i=1}^N I_{samp}(i) \times I_{samp}(i)} \quad (3)$$

$$P_{active} = PGAIN \times \frac{1}{N} \sum_{i=1}^N V_{samp}(i) \times I_{samp}(i) \quad (4)$$

$$P_{reactive} = PGAIN \times \frac{1}{N} \sum_{i=1}^N V_{samp,90}(i) \times I_{samp}(i) \quad (5)$$

$$P_{apparent} = V_{RMS} \times I_{RMS} \quad (6)$$

$$PF = \cos \varphi = \frac{P_{active}}{P_{apparent}} \quad (7)$$

$$V_{RMSfund} = VGAIN \times \sqrt{\frac{1}{N} \sum_{i=1}^N V_{samp}(i) \times V_{pure}(i)} \quad (8)$$

$$P_{activefund} = PGAIN \times \frac{1}{N} \sum_{i=1}^N I_{samp}(i) \times V_{pure}(i) \quad (9)$$

$$P_{reactivefund} = PGAIN \times \frac{1}{N} \sum_{i=1}^N I_{samp}(i) \times V_{pure(\pi/2)}(i) \quad (10)$$

$$V_{THD} = \frac{\sqrt{V_{RMS}^2 - V_{RMSfund}^2}}{V_{RMSfund}} \quad (11)$$

$$I_{RMSfund} = \frac{\sqrt{P_{activefund}^2 + P_{reactivefund}^2}}{V_{RMSfund}} \quad (12)$$

$$I_{THD} = \frac{\sqrt{I_{RMS}^2 - I_{RMSfund}^2}}{I_{RMSfund}} \quad (13)$$

2.3.3 后台进程

在后台执行基于样本的时间关键型进程。后台处理的流程很简单，该进程通过采样完成中断的触发以采样率开始。然后，后台进程：

- 从电压和电流 ADC 通道捕获数据
- 执行电压样本处理
- 电流样本处理
- 功率处理
- 线路频率处理
- 触发前台进程

图 2-5 展示了系统功能图，两个 SD24 ADC 通过 (校准和编程的) 分数延迟从电压和电流通道中采样。然后，样本通过直流去除滤波器。每个直流去除滤波器通过分析每个样本来动态跟踪 ADC 和信号的直流偏移，并在将样本传递到进一步处理之前去除直流 (来自信号和 ADC 直流偏移)。如果工作模式为直流模式，则直流去除滤波器仅去除 ADC 的直流偏移。

然后，电流样本和电压样本进行平方和累加得到均方根电流 (I_{rms}) 和电压 (V_{rms})。电流样本还与具有适当额外相位延迟 (如果需要) 的电压样本相乘并累加，来获得有功功率 (P_{active})；并与具有相应的 90° 相移的电压样本相乘并累加，来获得无功功率 ($P_{reactive}$)。

内部生成的与交流电压输入同步的纯正弦波振幅和相位连同 90° 相移版本用于与电流样本相乘并累加，来获得基波有功和基波无功功率。

周期检测机制通过检查预定义时间内的过零数来检测是否存在交流信号。如果过零数小于预期过零数减 1，则确定其处于直流模式，否则处于交流模式。当预定义的确切间隔设置为 80ms 时，这对于 25Hz 以上的频率非常适用。

在 4 个交流周期 (或直流模式下为 80ms) 之后，系统会设置一个标志，表明有需要处理的结果。然后，该标志触发系统执行缩放，并将这些累加值转换为具有适当单位 (V、A、W 等) 的值。

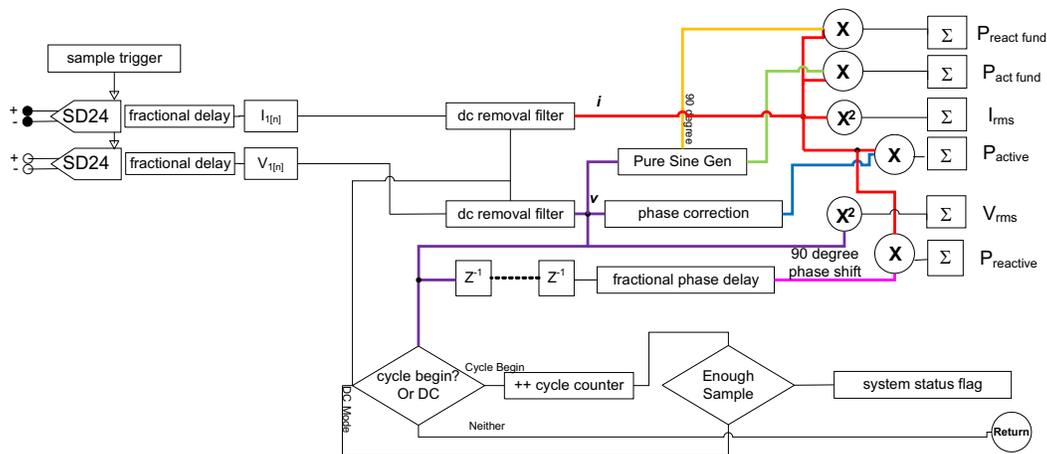


图 2-5. 系统功能图

2.3.4 相位校正

传感器和输入电路的无源器件在电流和电压信号之间引入了一个小但足够大的相位差。由于元件容差，这种相位不匹配因单元而异。这需要进行校正以实现高精度，因为即使是很小的相位误差也会破坏低功率因数下的功率测量精度。例如，如果功率因数为 0.5，则在有功功率测量中产生 0.1% 误差所需的相位误差仅为 0.034° 。

当 OSR 为 128 时，SD24 有一个内置电路，以 1/128 采样增量提供电压和电流采样时序之间的分数偏移。这相当于 50Hz 时的 0.034° 阶跃。通过将值加载到 SD24PREx 寄存器之一，可以随时调整相位校正。这会导致后续转换 (只有这一次转换) 被写入寄存器的 8 位量扩展。

图 2-6 说明了相位校正操作是如何完成的。

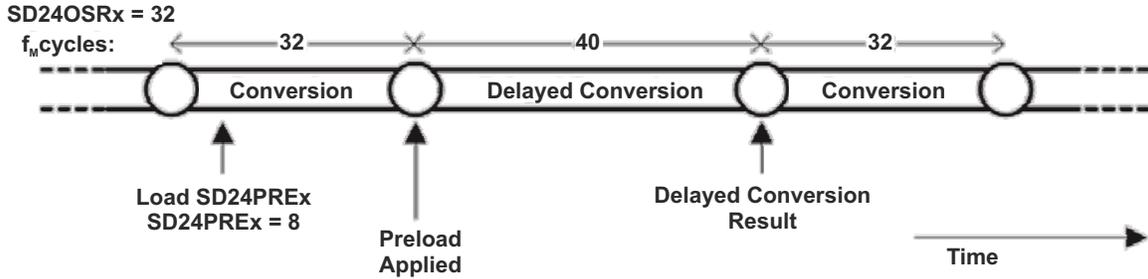


图 2-6. 使用 SD24PRE 寄存器进行相位校正

2.3.5 频率测量和周期跟踪

进行频率测量时，在多个周期中针对每个周期采集整数个样本并进行滤波，以产生具有高分辨率的电源周期值。但是，这可能需要很长时间才能稳定。为了缩短稳定时间，在过零的每一侧的电压样本之间进行直线内插。为了在电源轨上存在尖峰时使其更稳健，忽略估算中的不合理值。图 2-7 说明了该行为。

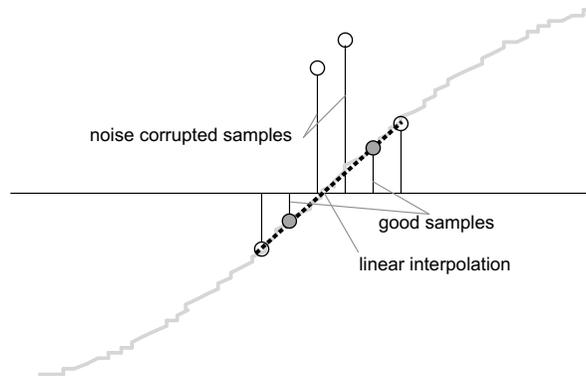


图 2-7. 频率测量和周期跟踪

由于噪声尖峰可能导致错误，因此使用变化率检查来滤除可能的错误信号，从而确保这两个点是从真正的过零点插入的。（例如，如果有两个负样本，那么噪声尖峰可能会使其中一个样本成为正样本，从而使该正负对看起来像是存在过零）。

由此产生的周期间时序会通过一个弱低通滤波器，从而进一步消除周期间变化。该结果是能够耐受噪声、稳定且准确的频率测量结果。

2.3.6 周期跟踪和前台进程触发

周期跟踪计数器和样本计数器与周期计数器一起跟踪累加了多少样本。当累加了 4 个周期的样本时，后台进程设置一个标志以通知前台进程，指示这是一个报告周期，应该产生一组新的缩放结果。

在实现中，有两组中间数据寄存器：一组使用后台作为中间数据，另一组使用前台计算读数。为了节省处理时间，当到达一个报告周期时，后台进程将一组中间寄存器传递给前台，并使用另一组累加下一组结果。

2.3.7 前台进程

后台收集到足够的样本后，后台设置标志 PHASE_STATUS_NEW_LOG。用户主循环检查该标志的状态并调用前台进程 calculate_phase_readings () 来执行剩余的运算以提供测量读数。前台进程包括：

- 功率处理：通过按样本数和功率比例因子进行缩放来计算有功功率、无功功率、基波有功功率、基波无功功率。
- 电压处理：通过按样本数进行缩放来计算 RMS 电压、基波 RMS 电压，计算平方根并乘以电压比例因子。
- 电流处理：
 - 通过按样本数进行缩放来计算 RMS 电流，计算平方根并乘以电流比例因子
 - 根据功率和电压计算基波 RMS 电流

- 通过将均方根电压和均方根电流相乘来计算视在功率
- 其他处理：
 - 计算功率因数
 - 计算交流线路的频率

3 校准技术

3.1 引言

为了尽可能地提高精度并补偿元件和制造容差，有必要执行校准过程。本节讨论校准技术、所需的仪器以及校准的过程和步骤。

3.2 校准技术

校准是基于前端接口模型定义的，如图 3-1 所示。

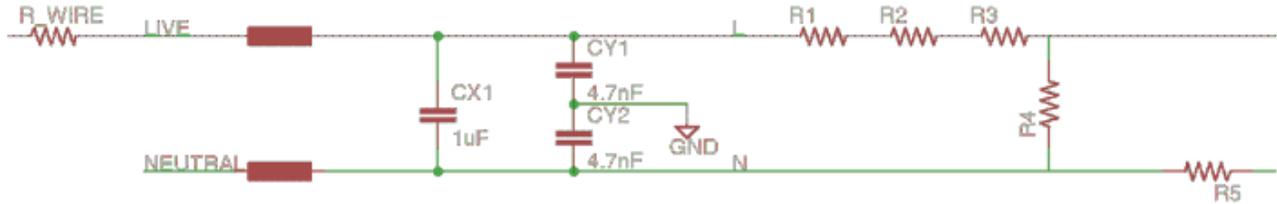


图 3-1. 前端接口模型

对于当前设计，需要执行两点校准和表征。VGAIN、IGAIN、PGAIN、CAP、RES、VDC_OFFSET、IDC_OFFSET 是在该过程中校准的参数；在设计和表征期间估算值被置于存储器中，以帮助加快校准。

备注

在校准之前，默认 CAP 和 RES 应设置为 0。

VAC_OFFSET、IAC_OFFSET、PHASE_CORRECT 是可能不需要校准的参数，除了 < 0.1% 的高精度外，表征对于嵌入式计量应用而言已经足够了。

备注

校准值被写入 EVM 的一个闪存页面中，因此，当需要写入一个新值时，整个页面被擦除。提供的 GUI 将自动执行读取、修改和写入操作。使用用户校准工具时，必须读取、备份整组校准值，更新修改的字段并将整组数据写回至 EVM。

3.2.1 校准设置

3.2.1.1 仪器

为了执行校准和测试，建议使用以下仪器列表：

- 能够输出足够功率以在额定频率 (50 - 60Hz) 和额定电压 (110V - 220V) 下驱动负载的交流电源，带有可变交流负载或 UUT。如果需要进行直流测量，还建议使用可变直流负载或 UUT。或者，也可以使用可以生成额定负载的交流测试装置。
- 一个可以根据交流仪表测试装置的电压、电流和相位设置提供交流参数读数的参考表

3.2.1.2 设置

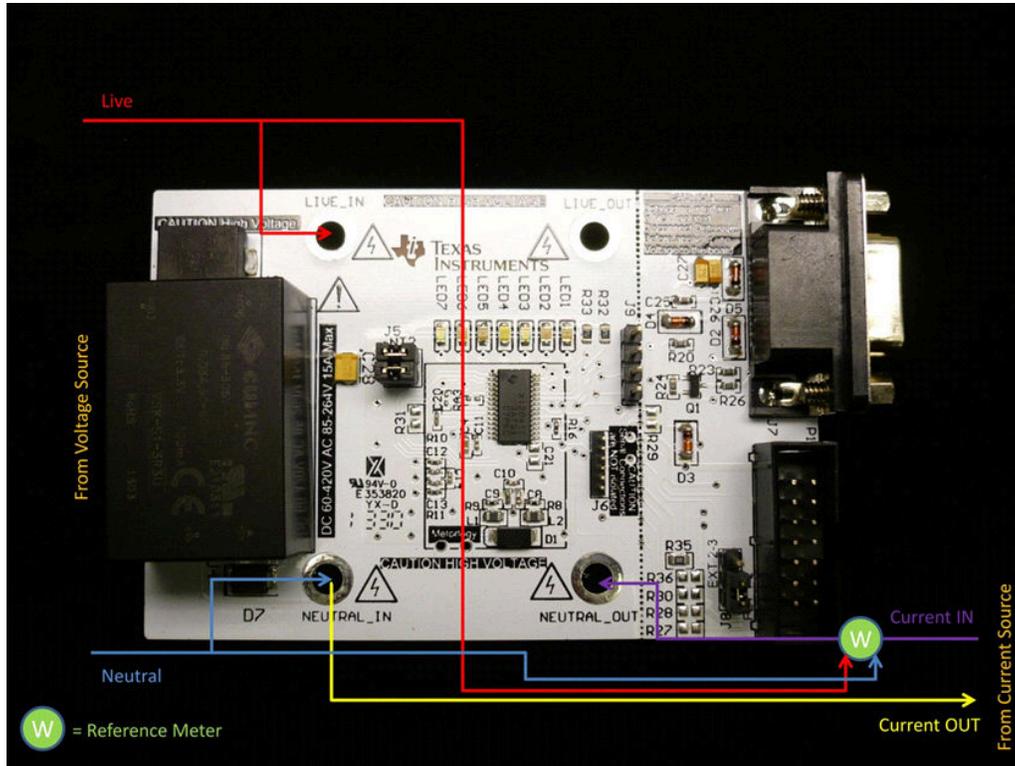


图 3-2. 校准设置

3.3 校准过程

3.3.1 交流和直流参数校准

按照以下步骤校准交流和直流参数。

校准 VGAIN :

1. 设置尽可能小的负载数。
2. 将 V_{IN} 设置为线电压。
3. 使用方程式 14 中的公式计算 VGAIN 的值：

$$VGAIN_{n+1} = \frac{V_{REF}}{V_{UUT}} \times VGAIN_n \quad (14)$$

其中，

$VGAIN_{n+1}$ 是新的电压校准系数

$VGAIN_n$ 是原始电压校准系数

V_{REF} 是为 VGAIN 校准设定的电压下的参考表电压读数

V_{UUT} 是为 VGAIN 校准设定的电压下的被测器件电压读数

或者，如果使用误差百分比（与提供的校准软件一样）：

$$\%Error = \frac{V_{UUT} - V_{REF}}{V_{REF}} \times 100\% \quad (15)$$

4. 写入并应用校准后的 VGAIN。

校准 IGAIN :

1. 将 VIN 设置为线电压。
2. 设置高负载和尽可能高的负载。
3. 使用方程式 16 中的公式计算 IGAIN 值：

$$IGAIN_{n+1} = \frac{I_{REF}}{I_{UUT}} \times IGAIN_n \quad (16)$$

其中，

IGAIN_{n+1} 是新的电压校准系数

IGAIN_n 是原始电压校准系数

I_{REF} 是为 IGAIN 校准设定的电压下的参考表电压读数

I_{UUT} 是为 IGAIN 校准设定的电压下的被测器件电压读数

或者，如果使用误差百分比（与提供的校准软件一样）：

$$\%Error = \frac{I_{UUT} - I_{REF}}{I_{REF}} \times 100\% \quad (17)$$

4. 写入并应用校准后的 IGAIN。

校准 PGAIN :

1. 与上述条件相同且功率因数 = 1。
2. 记下此时电压的误差百分比。
3. 计算 PGAIN 增益，使此时功率与电压具有相同的百分比误差。

$$PGAIN_{n+1} = \frac{P_{REF}}{P_{UUT}} \times PGAIN_n \times (1 - \%Error \text{ of } V_{UUT} \text{ at this load}) \quad (18)$$

其中，

PGAIN_{n+1} 是新的功率校准系数

PGAIN_n 是原始功率校准系数

P_{REF} 是为 PGAIN 校准设定的功率下的参考表电压读数

P_{UUT} 是为 PGAIN 校准设定的功率下的被测器件功率读数

或者，如果使用误差百分比（与提供的校准软件一样）：

$$\%Error = (1 + \%Error)(1 - \%Error \text{ of } V_{UUT} \text{ at this load}) \quad (19)$$

4. 写入并应用校准后的 PGAIN。

3.3.2 补偿电阻和电容校准

校准 VGAIN、IGAIN 和 PGAIN 后，按照以下步骤校准导线电阻和 EMI 电容的补偿。

校准 RES :

1. 使用方程式 20 计算 RES。

$$R_{WIRE} = \frac{V_{REF}(I_{max}) - V_{UUT}(I_{max})}{I_{max} - I_{min}} \approx \frac{V_{REF}(I_{max}) - V_{UUT}(I_{max})}{I_{max}} \quad (20)$$

其中，

R_{WIRE} 是估算电阻

$V_{REF(I_{max})}$ 是为 IGAIN 校准设定的电流下的参考表电压读数

$V_{UUT(I_{max})}$ 是为 IGAIN 校准设定的电流下的被测器件电压读数

I_{max} 是为 IGAIN 校准设定的电流

或者，如果使用误差百分比（与提供的校准软件一样）：

2. 写入并应用校准后的 RES（请注意电阻的单位为 $1/256 \Omega$ ）。

校准 CAP：

1. 设置尽可能小的负载数。
2. 将 VIN 设置为低线电压。
3. 使用方程式 21 中的公式计算 CAP：

$$C = \frac{1}{2\pi fV^2} \left(\sqrt{P_{APPERANT_REF}^2 - P_{ACTIVE}^2} - \sqrt{P_{APPERANT_UUT}^2 - P_{ACTIVE}^2} \right) \quad (21)$$

其中，

C 是估算的 EMI 滤波器电容

V 是为 C 校准设置的电压

$P_{APPERANT_REF}$ 是该校准设置下参考表的视在功率读数

$P_{APPERANT_UUT}$ 是该校准设置下被测器件的视在功率读数

P_{ACTIVE} 是参考表的有功功率读数

（假设已经校准为与 UUT 的有功功率读数相同）：

4. 写入并应用校准后的 CAP（请注意电容的单位为 $1/64 \mu F$ ）。

3.3.3 电流交流偏移校准

电流交流偏移是噪声拾取的结果，在分流电阻器电路上产生，会导致流过有限电流的错觉，而实际上没有电流流过分流器。虽然该噪声电流对功率读数的精度没有影响，但实际上会影响电流读数及其精度，尤其是在电流很小的时候。为了抵消这一点，EVM 固件具有将其从电流读数中移除的机制。移除该电流偏移的步骤为：

1. 施加标称电压以确保 EVM 正常工作。
2. 移除 EVM 上的所有负载。
3. 例如，取 100 个电流读数并求其平均值 I_NOISE （以 A 为单位）。
4. 使用方程式 22 计算电流交流偏移值。（请注意，即使只有几毫安的噪声，这也是一个很大的数字）。

$$I_AC_OFFSET = \text{int} \left(I_NOISE \left(\frac{1024 \times 10^6}{IGAIN} \right) \right)^2 \quad (22)$$

5. 写入并应用校准后的 I_AC_OFFSET

3.3.4 电压交流偏移校准

在大多数情况下，电压交流偏移对电压读数的影响很小，不需要校准。

3.3.5 相位校正校准

1. 将测试装置设置为产生额定电压并设置校准电流（例如 5A）。
2. 确保 PGAIN 校准已完成（PF = 1）。
3. 将测试装置设置为以功率因数 0.5 产生输出（此时 + 或 - 并不重要）。
4. 记下功率误差。
5. 将测试装置切换为以功率因数 0.5 和与步骤 3 相反的方向产生输出。

6. 记下功率误差。
7. 此时，两个功率误差应该具有相对于 $PF = 1$ 时的校准功率误差大致相同的偏差，但方向不同。
 - a. 例如，在 $PF = 1$ 时，校准误差为 0.1%。如果 $PF = 0.5$ ，则功率误差读数约为 0.5%。在 $PF = -0.5$ 时，功率误差读数应约为 -0.3%。
8. 使用相对于电流相位校正的时间偏差调整相位校正，以便将 $PF = \pm 0.5$ 时的功率误差降至最低。
 - a. 例如，如果电流相位校正为 $13 \mu s$ ，而需要 $11 \mu s$ ，则在手动校准窗口的相位校正框中输入 -2。

3.3.6 直流参数校准

EVM 的设计允许在校准交流电的同时自动校准直流测量参数。在校准的最后一步（电流交流偏移）中，直流测量参数会自动更新。

实际上，每当读取整组校准值时，该组值中也会包含最新的直流测量参数。当对其他参数进行更新时，直流测量参数也会更新为最新值。然而，当电流较低时该值最准确（完全没有电流时最好），因此建议在 `I_AC_OFFSET` 校准后执行直流测量参数更新。

备注

这可能不是一种用户友好的设置直流测量参数的方法。发布嵌入式计量定制校准软件后，修改并改进了实现最佳直流测量参数的过程。

4.3 硬件设置过程

4.3.1 设置 EVM 的电源

EVM 的设计允许通过内置电源为 EVM 供电，该电源接受线路输入以生成 EVM 所需的电源。该 EVM 还允许通过用户电源为 EVM 供电。

该 EVM 随附安装在 EVM 上的内置电源块。如果要通过外部电源供电，则必须断开内置 VSK-S1-3P3U 电源块。拆下 J5 上的两根跳线，并向 J5 上的 V_{DD} 和 V_{SS} 施加 3.3V 的电压，如图 4-1 所示。

WARNING

在硬件和软件设置完成之前，请勿为 EVM 供电。

4.3.2 设置串行通信接口

将 RS232 扩展电缆连接到 EVM 上的 DB-9 连接器和计算机上的标准 RS232 端口。

4.3.3 设置线路输入和负载输出

- 将交流电源火线或直流电源正极焊接到 LIVE_IN 连接器
- 将交流电源中性线或直流电源负极焊接到 NEUTRAL_IN 连接器
- 将负载的火线或正电源输入焊接到 LIVE_OUT 连接器
- 将负载的中性线或负电源输入焊接到 NEUTRAL_OUT 连接器

4.3.4 设置调试接口

Spy-bi-Wire 在该 EVM 上用作连接外部调试器的调试接口。

在连接到 MSP-FET430U1F 之前，如果要由 FET 为 i2040 供电（调试时），则将 J8 的 1 - 2（朝向电路板边缘）短接。如果要由 V_{DD} 引脚上的电压为 i2040 供电（调试时），则将 J8 的 2 - 3（朝向电路板中心）短接（请参阅图 4-1）。

WARNING

调试接口未隔离；请确保在连接交流或直流高电压时在 EVM 和用于调试的 PC 之间实现适当的隔离。

对于 EVM 运行，连接调试接口是可选的。EVM 可以在不连接调试器的情况下独立运行。

5 校准器软件

需要一个软件包来访问 EVM 的全部功能，可以从德州仪器 (TI) 网站 (TIDM-SERVER-PWR-MON) 下载该软件包。本节讨论了该软件包中的软件以及设置软件以与 EVM 配合使用的过程。

5.1 软件包内容

该 EVM 的 ZIP 文件包含多个软件包，其中包括：

- 在 PC 上运行以读取和校准 EVM 的软件工具。节 5.2 和节 6 讨论了该 PC 软件工具的设置和操作。
- 源代码和嵌入式计量库。节 8 和节 9 对此进行了讨论。
- 包含原理图、布局和物料清单的硬件设计文件，节 9.2 对此进行了讨论。

5.2 设置 PC 软件工具

5.2.1 最低系统要求

- 具有内置 RS232 端口或通过 USB 提供的 RS232 端口的 PC
- 运行 Microsoft® Windows® XP SP3 或 Windows 7 的 PC

软件工具已在安装了 1.25GB RAM 的相对较旧的 Pentium M 1.4GHz 机器上测试并正常运行，因此，假设当今大多数计算机都可以满足处理能力要求。

5.2.2 安装软件

只需将名为“calibrator-runtime.zip”的文件解压缩到任何文件夹中。包含运行软件工具所需文件的名为“calibrator-runtime”的文件夹将位于该文件夹中。名为“calibrator-20121120.exe”的文件是软件工具的可执行文件。名为“calibrator-config.xml”的文件包含软件工具的设置信息。需要先编辑该 XML 文件，然后才能启动 calibrator-20121120.exe。

5.2.3 配置软件

按照以下步骤设置 XML，以便 calibrator-20121120.exe 能够正常运行：

1. 右键点击“我的电脑”图标，然后在弹出菜单中选择“属性”。
2. 在“系统属性”窗口中选择“硬件”选项卡（请参阅图 5-1），然后点击“设备管理器”转到“设备管理器”窗口。

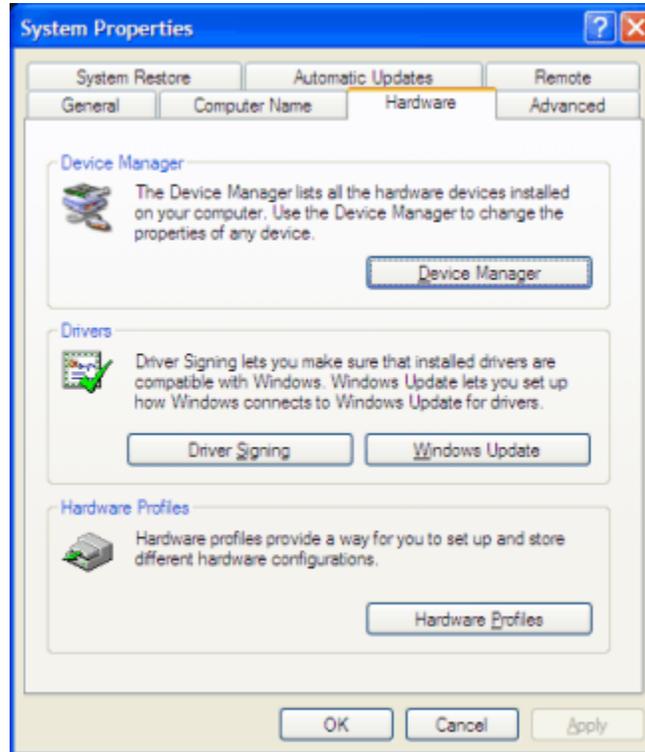


图 5-1. “系统属性”窗口

3. 检查连接 PC 和 EVM 的串行端口的 COM 端口号。

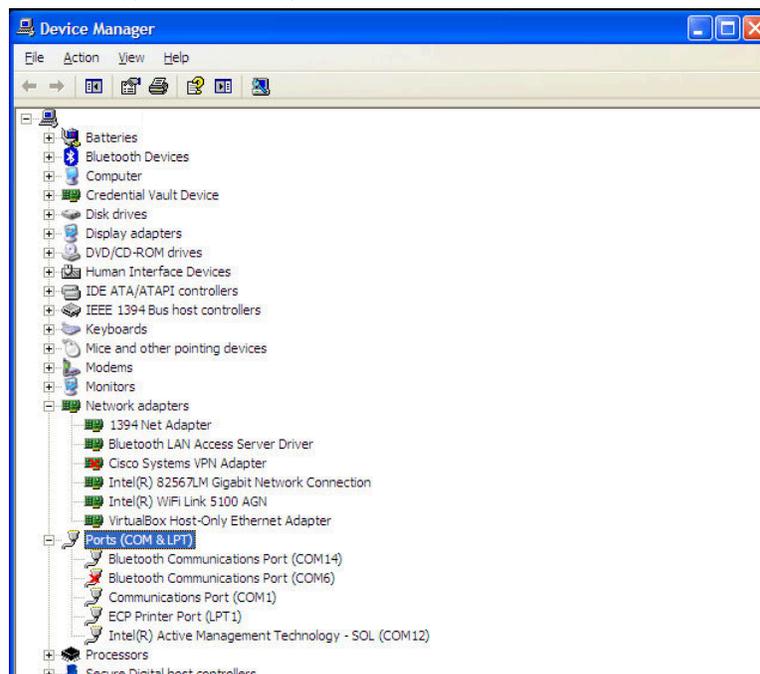


图 5-2. “设备管理器”窗口

4. 使用文本编辑器或 XML 编辑器打开“calibrator-runtime”文件夹中的“calibration-config.xml”。转到图 5-3 中显示的行，输入 COM 端口并保存该文件。

```

256 <step current="40.000" phase="0.0" gain="1.0"/>
257 <step current="45.000" phase="0.0" gain="1.0"/>
258 <step current="50.000" phase="0.0" gain="1.0"/>
259 <step current="55.000" phase="0.0" gain="1.0"/>
260 </correction>
261 </phase>
262 </temperature/>
263 </rtc/>
264 </cal-defaults>
265 <meter position="1">
266 <port name="com1" speed="9600"/>
267 </meter>
268 <reference-meter>
269 <port name="com1" speed="9600"/>
270 <type id="kaipo-p3001c"/>
271 <log requests="on" responses="on"/>
272 </reference-meter>
273 <generator>
274 <port name="com1" speed="9600"/>
275 <type id="kaipo-p3001c"/>
276 <log requests="on" responses="on"/>
277 </generator>
278 <idle voltage="230" current="5.0" phase="0" frequency="50.0"/>
279 <test-pattern>
280 <step type="idle" voltage="230" current="5.0" phase="0" frequency="50.0" duration="10"/>
281 <step type="sanity" voltage="230" current="5.0" phase="0" frequency="50.0"/>
282 <step type="check-active-leads" voltage="230" current="5.0" phase="0" frequency="50.0"/>
283 <step type="quick" voltage="230" current="5.0" phase="0" frequency="50.0"/>
284 <step type="quick" voltage="230" current="5.0" phase="60" frequency="50.0"/>
285 <step type="update-scalings" voltage="230" current="5.0" phase="0" frequency="50.0"/>
286 <step type="update-phase-correction" voltage="230" current="5.0" phase="0" frequency="50.0"/>
287 </test-pattern>
288 </calibration-config>
289
290

```

图 5-3. 编辑“calibration-config.xml”

5.3 仪器

示例代码中定义了一组校准系数，这些校准系数使读数的精度大致为百分之几。如果需要更准确的结果，则需要对 EVM 执行校准过程。有关校准过程的更多信息，请参阅节 3。为了执行校准，建议使用以下仪器列表：

- 在额定频率 (50 - 60Hz) 和额定电压 (110V - 220V) 下能够输出足够大的功率以驱动负载的交流电源，或能够产生额定频率和电压的交流测试装置。
- 可变交流负载或 UUT。如果需要进行直流测量，还建议使用可变直流负载或 UUT。或者，可以生成额定负载的交流测试装置。
- 可提供 V_RMS、I_RMS、P_ACTIVE 交流参数的参考表

以下链接是用于测试和校准设计的仪器的链接：

参考表：[http://www.hc.com.tw/portal_c1_cnt_page.php?](http://www.hc.com.tw/portal_c1_cnt_page.php?owner_num=c1_142363&button_num=c1&folder_id=17560&cnt_id=124863&search_field=&search_word=&search_field2=&search_word2=&search_field3=&search_word3=&bool1=&bool2=&search_type=1&up_page=1)

[owner_num=c1_142363&button_num=c1&folder_id=17560&cnt_id=124863&search_field=&search_word=&search_field2=&search_word2=&search_field3=&search_word3=&bool1=&bool2=&search_type=1&up_page=1](http://www.hc.com.tw/portal_c1_cnt_page.php?owner_num=c1_142363&button_num=c1&folder_id=17560&cnt_id=124863&search_field=&search_word=&search_field2=&search_word2=&search_field3=&search_word3=&bool1=&bool2=&search_type=1&up_page=1)

交流电源：<http://www.chromausa.com/acpowersources/61500lo-ac-source.php>

参考表：<http://www.chromausa.com/powermeters/66201-66202-digital-power-meters.php>

直流电子负载：<http://www.chromausa.com/dloads.php#6310a>

6 操作 PC 软件工具

6.1 引言

现在硬件和软件的设置和配置已经完成。此处讨论 EVM 和 PC 软件工具的操作。请务必先完成上一节所述的步骤，为 EVM 供电，并为负载施加线电压，然后再继续。

6.2 开始使用 EVM

当 EVM 通电后，七个 LED 中的一些 LED 会闪烁或亮起，来指示其运行状态。一些 LED 未使用；可以根据需要更改提供的源代码以使用所有 LED。表 6-1 展示了 LED 指示状态列表。

表 6-1. LED 指示状态

LED	状态	指示
LED1	打开	后台操作正在运行
LED1	OFF	后台操作已完成
LED2	打开	电压负半周期
LED2	OFF	电压正半周期
LED3	打开	未使用
LED3	OFF	未使用
LED4	打开	前台操作正在运行
LED4	OFF	前台操作已完成
LED5	打开	EVM 处于交流模式测量
LED5	OFF	EVM 未处于交流模式测量
LED6	打开	EVM 处于直流模式测量
LED6	OFF	EVM 未处于直流模式测量
LED7	打开	有功电能脉冲正在输出
LED7	OFF	有功电能脉冲空闲

EVM 现在已为运行做好准备，启动文件夹“calibration-runtime”中的软件“calibrator-20121120.exe”，开始与 EVM 通信。此时会显示如图 6-1 所示的窗口。按照 XML 文件“calibration-config.xml”中的定义，为仪表位置 1 分配了与 EVM 进行通信的串行端口，如果建立了从 PC 到 EVM 的通信，[Comms] 指示器会变为绿色，并且在通信发生时在红色和绿色之间闪烁。

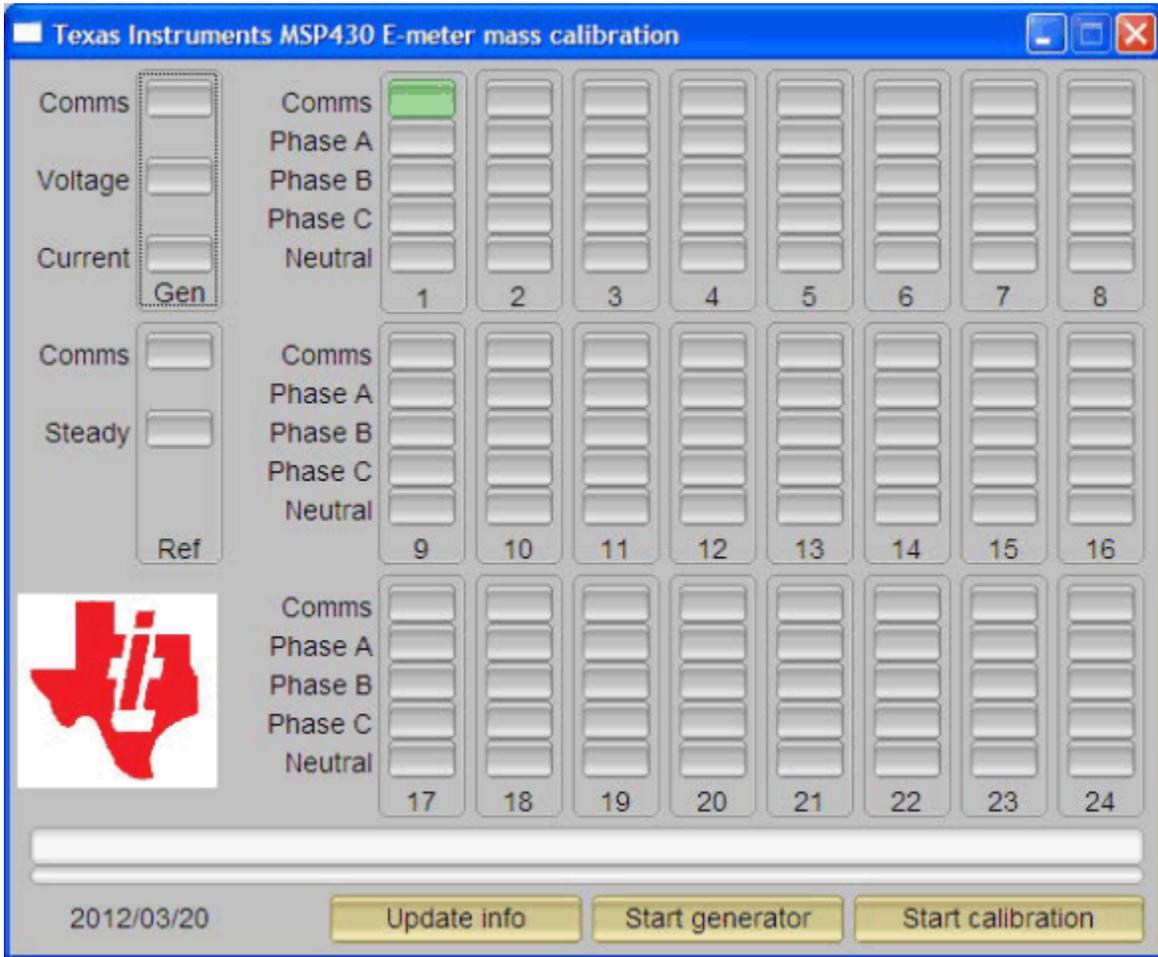


图 6-1. 校准器软件启动窗口

点击“Meter status”窗口中的 [Comms] 指示器，如图 6-2 所示。

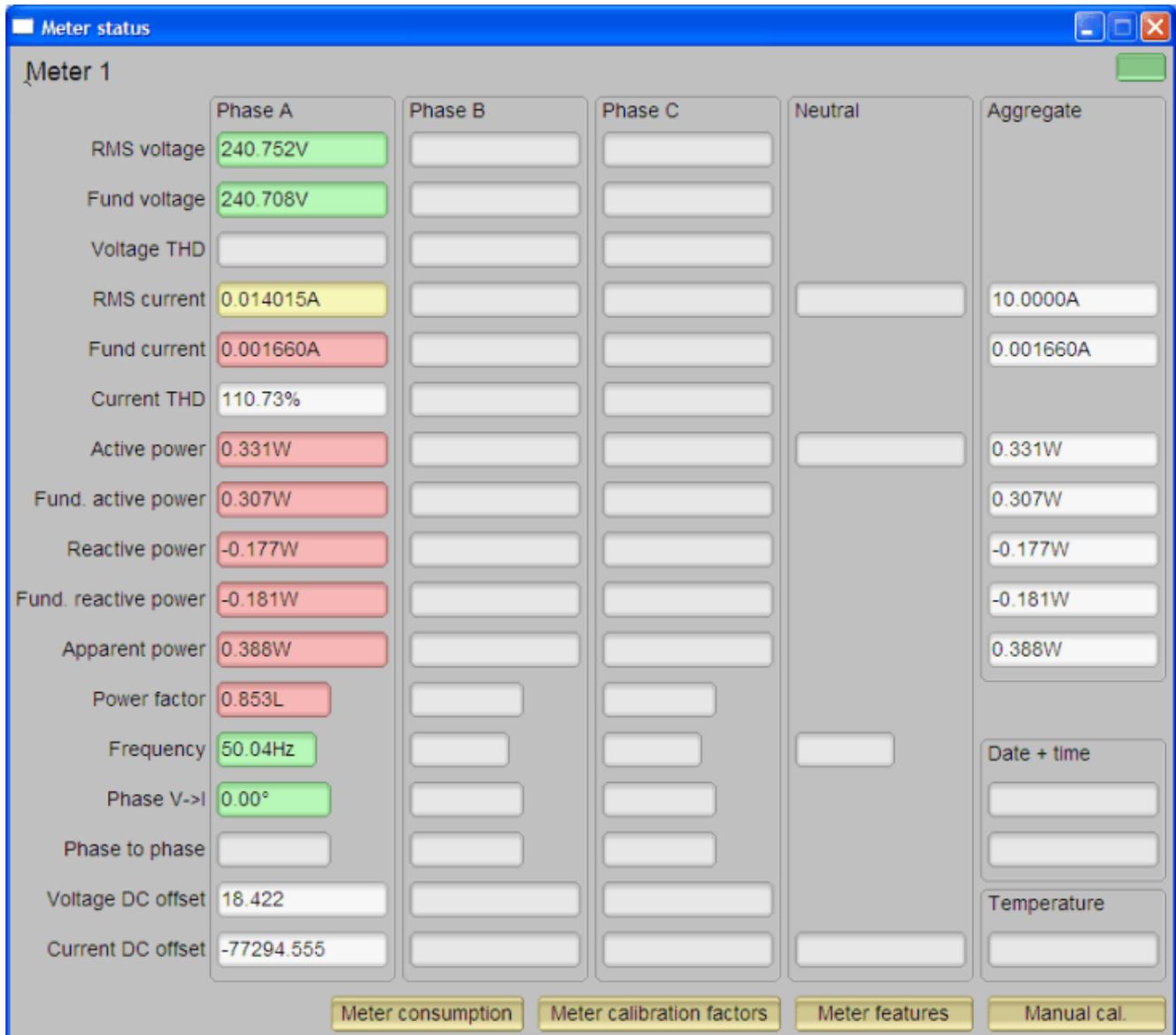


图 6-2. 仪表状态窗口

该窗口显示仪表的当前读数。如果 EVM 不支持特定的读数，则读数框的背景为灰色。如果来自 EVM 的读数与该框中的读数之间存在很大的差异，则该框变为红色。如果来自 EVM 的读数具有相当小的差异，则该框变为黄色。如果读数具有很小的差异，则该框变为绿色。请注意，PC 上的软件每秒读取一次 EVM，该软件还会对读取的数据求平均值，因此，更新速率比 EVM 的更新速率慢。

该窗口的底部有四个按钮：

1. 点击“Meter consumption”按钮，会打开“Meter consumption”窗口，但该 EVM 不支持此功能。“Meter consumption”不提供任何有用的信息。
2. 点击“Meter calibration factors”按钮，会打开“Meter calibration factors”窗口（请参阅图 6-3）。该窗口中显示当前校准系数值。

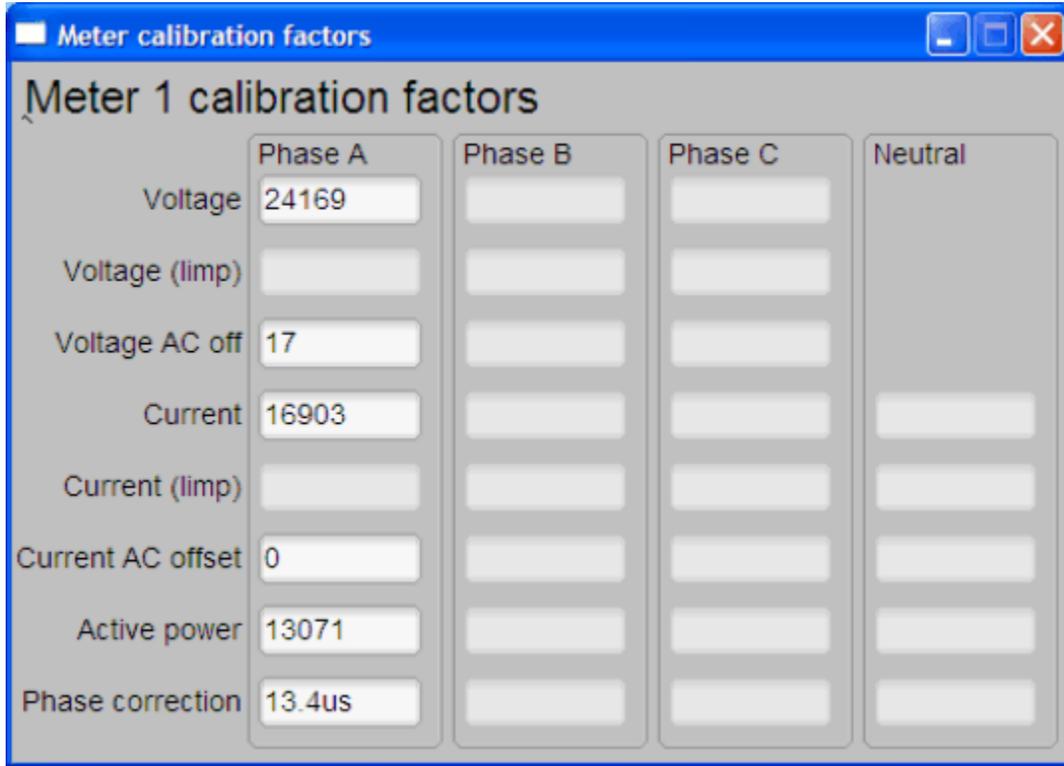


图 6-3. “Meter calibration factors” 窗口

3. 点击“Meter features”按钮，会打开“Meter features”窗口（请参阅图 6-4）。“Meter features”窗口中报告 EVM 的支持功能。图 6-4 仅展示了仪表功能窗口的外观，但这并不是 EVM 的确切功能。



图 6-4. “Meter features” 窗口

4. 点击“Manual cal.”按钮，会打开“Meter error”窗口（请参阅图 6-5）。在该窗口中，可以通过输入 EVM 读数相对于参考表读数的百分比误差来调整校准系数。有关执行校准的技术和过程的更多详细信息，请参阅节 3。

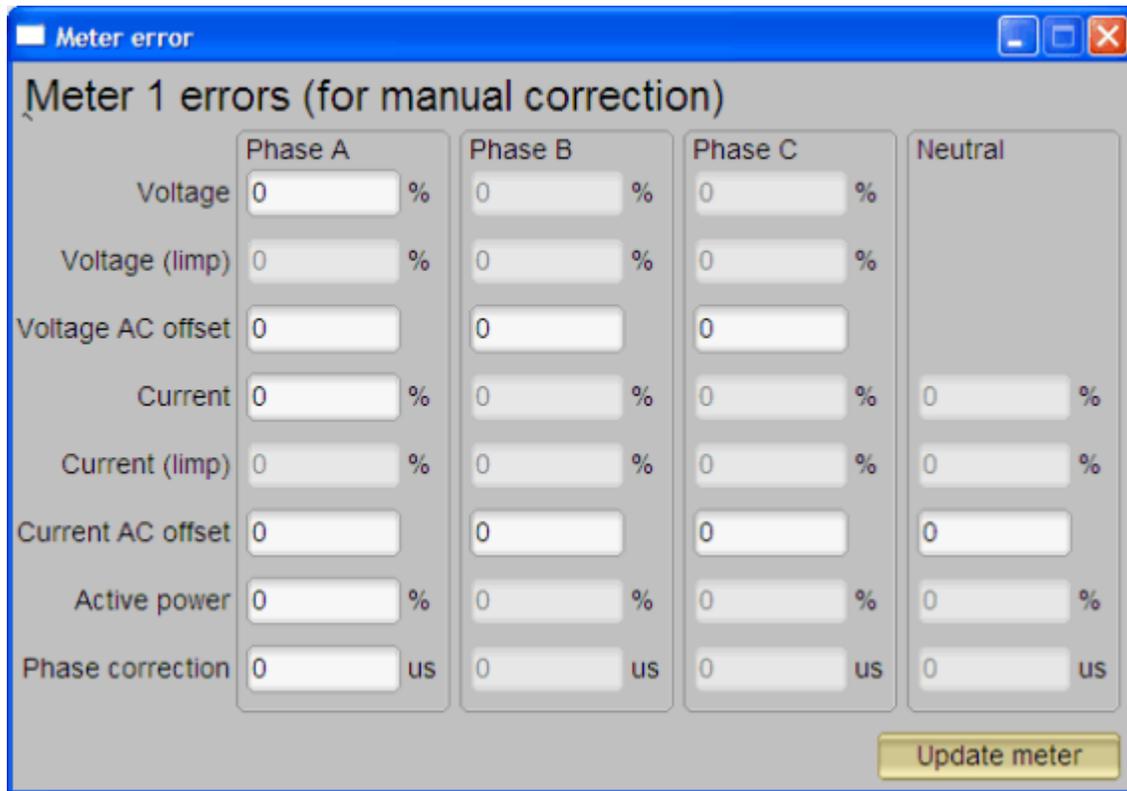


图 6-5. “Meter error” 窗口

要修改校准系数，需要在百分比误差中输入校正值。方程式 23 展示了百分比误差的计算方法：

$$\%Error = \frac{EVM \text{ Reading} - \text{Reference Meter Reading}}{\text{Reference Meter Reading}} \times 100\% \quad (23)$$

在“Meter error”窗口的相应框中输入百分比误差（请参阅图 6-5）。点击“Update meter”，将计算出更新后的校准值，这些值会被写入 EVM。相应的值将反映在“Meter calibration factor”窗口中。

6.3 已知问题

校准器软件是与未针对嵌入式计量完全定制的公用事业仪表一起运行的旧版软件。以下是现有软件的已知问题列表，这些问题将在为嵌入式计量定制的下一版校准器软件发布时得到修复：

- 中性线监控在“Meters Features”窗口中显示为支持的功能；这是错误的指示。启用导线电阻补偿或入口电容器补偿后，校准器被解释为支持中性线监控。
- 无法使用校准器软件对导线电阻补偿的电阻值和入口电容器补偿的电容值进行编程。
- 无法使用校准器软件写入电压交流偏移值。输入到电压交流偏移框中的值实际上被写入到电流交流偏移中。输入到电流交流偏移框中的值无效。
- 无法通过校准软件将直流偏移值写入 EVM，电流校准软件在每次任何校准值更新时获取电流和电压直流偏移值。
- *Meter Status* 窗口中的聚合电流始终显示为 10.0000A。

7 串行通信命令

7.1 引言

要成为一个有用的工具，必须具有外部接口。EVM 使用其连接到 RS232 DB-9 连接器的 UART 端口与外部进行通信。本节讨论 EVM 用于发送读数和命令的协议。

备注

可以更改提供的源代码以执行自定义通信协议、命令、响应和功能。

7.2 通信协议

在运行期间，EVM 使用以下参数通过串行通信端口与外部主机通信：

- 9600bps、无奇偶校验、8 个数据位、1 个停止位

EVM 支持轮询模式通信，可以访问全部功能，读取整组测量结果，读取和写入校准系数。

7.2.1 轮询模式

当 EVM 运行时，它等待主机通过串行端口发出的命令。当接收到命令帧并通过帧检查后，EVM 会解释该命令并执行请求的操作。EVM 使用数据帧格式与主机通信，如图 7-1 所示。

0	1	2	3	4	5	6	7	8	9	
F_Start	Address						F_Start	C_cod	Length	
0x68	0x99	0x99	0x99	0x99	0x99	0x99	0x68	0x23	len	
0..9			10..len+9				len+10	len+11		
Header			Data				CS	0x16		

图 7-1. 轮询模式数据帧格式

帧以 9 字节的标头开始，后跟 0 至 255 字节的数据字段，然后是校验和字节和帧结束字节。

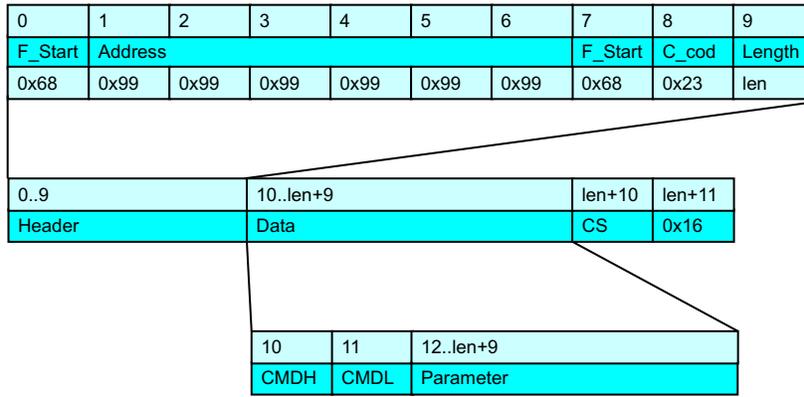
标头以 0x68 开头，后跟一个地址字段，该字段固定为 0x999999999999。可以修改提供的通信协议源代码 (emeter-dlt645.c) 以使 EVM 响应不同的地址。

0x68 和 0x23 是用于指示数据字段中的字节数的长度字节，后跟固定的分隔符。帧的末尾是一个校验和字节，它是从标头开始到数据字段结束、后跟帧结束标记 0x16 的每个字节的字节总和对 256 取模的结果。

7.2.1.1 命令和响应帧

图 7-2 和图 7-3 展示了轮询模式下命令和响应帧的结构。命令和响应帧的标头和帧末尾具有相同的结构，区别在于数据字段。命令的数据字段中的前 2 个字节是 CMDH 和 CMDL，这两个字节定义命令和后面的参数。从主机接收到有效的命令帧后，EVM 以 RSPH = CMDH 和 RSPL = CMDL | 0x80 (因此 CMDL 无法使用大于 0x7F 的值) 的响应帧来响应主机。响应数据由接收到的命令定义。

下一节列出了命令的定义和相应的响应。



A. 如上所示，响应放置在数据字段中：RSPH = CMDH，RSPL = CMDL | 0x80。

图 7-2. 轮询模式命令帧格式

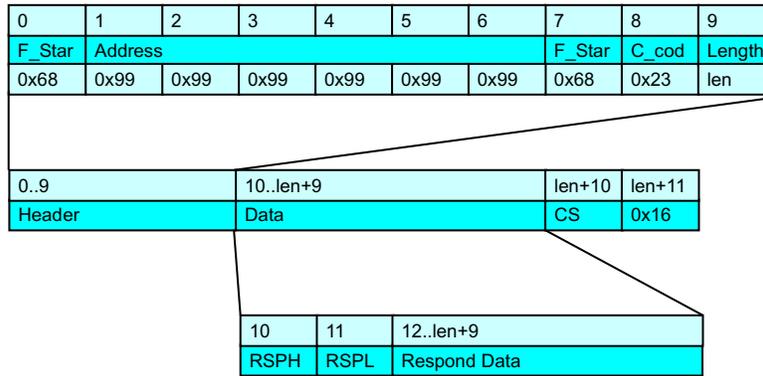


图 7-3. 轮询模式响应帧格式

备注

在写入自定义协议时，请确保数据缓冲区足以保存来自主机的数据字节。

7.3 命令

7.3.1 HOST_CMD_GET_METER_NAME

读取由“metrology-calibration-template.h”中的 #define METER_NAME 定义的 32 字节仪表名称字符串。

7.3.1.1 命令格式

表 7-1. HOST_CMD_GET_METER_NAME

命令	命令			响应		
LEN	2			34		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x52	0	U8	0x52
CMDL	1	U8	0x00	1	U8	0x80
				2	U8(32)	32 字节仪表名称

7.3.2 HOST_CMD_GET_METER_VER

读取由以下代码定义的 4 x 32 位版本号：

- #define METER_SOFTWARE_VERSION
- #define METER_HARDWARE_VERSION
- #define METER_METROLOGY_VERSION

- #define METER_PROTOCOL_VERSION

可以在“metrology-calibration-template.h”中找到这些代码。

7.3.2.1 命令格式

表 7-2. HOST_CMD_GET_METER_VERSION

	命令			响应		
LEN	2			18		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x53	0	U8	0x53
CMDL	1	U8	0x00	1	U8	0x80
				2	U8(4)	4 字节软件版本
				6	U8(4)	4 字节硬件版本
				10	U8(4)	4 字节计量版本
				14	U8(4)	4 字节协议版本

7.3.3 HOST_CMD_GET_METER_CONFIGURATION

这命令 EMV 返回 EVM 支持的参数和功能。

7.3.3.1 命令格式

表 7-3. HOST_CMD_GET_METER_CONFIGURATION

	命令			响应		
LEN	2			20		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x56	0	U8	0x56
CMDL	1	U8	0x00	1	U8	0x80
				2	U8	相位数
				3	U8	功能 0*
				4	U8	功能 1*
				5	U8	功能 2*
				6	U8	功能 3*
				7	U8	0x00
				8	U16	电源标称频率
				10	U16	电源标称电压
12				12	U16	电源基本电流
				14	U16	电源最大电流
				16	U32	采样率的 100 倍

7.3.3.2 参数定义

表 7-4. 参数定义

功能 0		功能 2	
位 7	保留	位 7	测量正交无功功率
位 6	保留	位 6	测量电源频率
位 5	保留	位 5	测量功率因数
位 4	保留	位 4	测量 I _{RMS}
位 3	保留	位 3	测量 V _{RMS}
位 2	导线电阻补偿支持	位 2	测量视在功率
位 1	入口电容器补偿支持	位 1	测量三角无功功率
位 0	中性线监控支架	位 0	估算有功功率
功能 1		功能 3	
位 7	多速率支持	位 7	测量骤降和骤升
位 6	未定义	位 6	测量电流 THD
位 5	温度支持	位 5	测量电压 THD
位 4	校正的 RTC 支持	位 4	测量基波 I _{RMS}
位 3	实时时钟 (RTC) 支持	位 3	测量基波 V _{RMS}
位 2	动态相位校正支持	位 2	测量基波有功功率
位 1	自动报告支持	位 1	测量基波无功功率
位 0	跛行模式支持	位 0	未定义

7.3.4 HOST_CMD_GET_RTC

由于 EVM 中没有 RTC，因此该命令不会读取任何与 RTC 相关的内容，但通过该命令从仪表读取的数据之一是温度。因此，该命令被视为使用 MSP430i2040 上的内部温度传感器读取 EVM 温度的命令。

7.3.4.1 命令格式

表 7-5. HOST_CMD_GET_RTC

命令	偏移	宽度	数据	响应	偏移	宽度	数据
LEN	2			10			
CMDH	0	U8	0x59	0	U8	0x59	
CMDL	1	U8	0x00	1	U8	0x80	
				2	U8(6)		6 字节虚拟数据
				8	S16		以 0.01°C 为单位的温度

7.3.5 HOST_CMD_ALIGN_WITH_CALIBRATION_FACTORS

这命令 EVM 将校准系数从闪存重新加载到其测量活动的操作中。

7.3.5.1 命令格式

表 7-6. HOST_CMD_ALIGN_WITH_CALIBRATION_FACTORS

	命令			响应		
LEN	2			2		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x5A	0	U8	0x5A
CMDL	1	U8	0x00	1	U8	0x80

7.3.6 HOST_CMD_SET_PASSWORD

该命令将密码从主机传递到 EVM 以启用校准模式，从而允许执行校准和其他功能。如果启用了自动报告模式，该命令还会禁用自动报告模式并将 EVM 设置为轮询模式。在示例代码中，密码默认为密码 1 = 0x1234，密码 2 = 0x5678，密码 3 = 0x9ABC，密码 3 = 0xDEF0。（可以通过修改 `emeter-template.h` 中的项，然后重新编译来更改密码）。

7.3.6.1 命令格式

表 7-7. HOST_CMD_SET_PASSWORD

	命令			响应		
LEN	10			2		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x5A	0	U8	0x60
CMDL	1	U8	0x00	1	U8	0x80
	2	U16	密码 1			
	4	U16	密码 2			
	6	U16	密码 3			
	8	U16	密码 4			

7.3.7 HOST_CMD_GET_READINGS_PHASE_N

该命令从 EVM 读取最新的测量值。如果 CMDL 被指定为 0x00 以外的值，则 CMDL 值将优先确定相数。

备注

对于该 EVM，仅支持 CMDH = 0x61。

7.3.7.1 命令格式

表 7-8. HOST_CMD_GET_READINGS_PHASE_N

	命令			响应		
LEN	2			34		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x6n (n = 1、2、3)	0	U8	0x6n (n = 1、2、3)
CMDL	1	U8	0xdp (在该 EVM 中始终为 0x00)	1	U8	0x80 0xdp , d = 器件 [0..7] , p = 相位 [1..15]
				2	S32	以 mV 为单位的电压
				6	S32	以 μ A 为单位的电流
				10	S32	以 mW 为单位的有功功率
				14	S32	以 mW 为单位的无功功率
				18	S32	以 mW 为单位的视在功率
				22	S16	以 0.001 为单位的功率因数
				24	S16	以 0.01Hz 为单位的频率
				26	S32	电压通道直流偏移
				30	S32	电流通道直流偏移

7.3.8 HOST_CMD_GET_EXTRA_READINGS_PHASE_N

该命令从 EVM 读取最新的额外测量值。如果 CMDL 被指定为 0x00 以外的值，则 CMDL 值将优先确定相数。

备注

对于该 EVM，仅支持 CMDH = 0x69。

7.3.8.1 命令格式

表 7-9. HOST_CMD_GET_EXTRA_READINGS_PHASE_N

命令	命令	宽度	数据	响应	宽度	数据
LEN	2			34		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x68 + n (n = 1、2、3)	0	U8	0x68 + n (n = 1、2、3)
CMDL	1	U8	0xdp (在该 EVM 中始终为 0x00)	1	U8	0x80 0xdp , d = 器件 [0..7] , p = 相位 [1..15]
				2	S32	以 mW 为单位的基波有功功率
				6	S32	以 mW 为单位的基波无功功率
				10	S32	以 mV 为单位的基波电压
				14	S32	以 μ A 为单位的基波电流
				18	U16	以 0.01% 为单位的电压 THD
				20	U16	以 0.01% 为单位的电流 THD
				22		保留接下来的 12 个字节

7.3.9 HOST_CMD_SUMCHECK_MEMORY

该命令请求 EVM 计算并返回从起始闪存地址到结束闪存地址 (含) 的 16 位校验和。

7.3.9.1 命令格式

表 7-10. HOST_CMD_SUMCHECK_MEMORY

命令	命令	宽度	数据	响应	宽度	数据
LEN	10			4		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0x75	0	U8	0x75
CMDL	1	U8	0x00	1	U8	0x80
	2	U32	起始闪存地址	2	U16	校验和
	6	U32	结束闪存地址			

7.3.10 HOST_CMD_CLEAR_CALIBRATION_DATA

由于校准数据存储在闪存页面中，因此需要首先清除闪存页面，然后才能写入任何校准数据。该命令请求 EVM 擦除包含校准数据的闪存。

备注

在执行该命令之前将完整的校准数据集读回并保存在主机中，否则校准数据将无法再次检索。

7.3.10.1 命令格式

表 7-11. HOST_CMD_CLEAR_CALIBRATION_DATA

	命令			响应		
LEN	2			2		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0xD0	0	U8	0xD0
CMDL	1	U8	0x00	1	U8	0x80

7.3.11 HOST_CMD_SET_CALIBRATION_PHASE_N

这命令 EVM 使用列出的值设置校准值。在执行该命令之前，请确保已读取、保存和擦除包含校准值的闪存页面。

备注

对于该 EVM，仅支持 CMDH = 0xD1。

7.3.11.1 命令格式

表 7-12. HOST_CMD_SET_CALIBRATION_PHASE_N

	命令			响应		
LEN	30			2		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0xD0 + n	0	U8	0xD0 (n = 1、2、3)
CMDL	1	U8	0xdp (在该 EVM 中始终为 0x00)	1	U8	0x80 0xdp , d = 器件 [0..7] , p = 相位 [1..15]
	2	S16	0xdp (在该 EVM 中始终为 0x00)			
	4	U16	以 1/64 μ F 为单位的入口电容			
	6	S32	电流通道直流偏移			
	10	U32	电压通道交流偏移			
	14	U32	电流通道交流偏移			
	18	S16	以 1/1024Ts 为单位的相位校正			
	20	U16	V _{rms} 比例因子			
	22	U16	以 1/256 Ω 为单位的导线电阻			
	24	U16	I _{rms} 比例因子			
	26	U16	保留 0x0000			
	28	U16	功率比例因子			

7.3.12 HOST_CMD_GET_CALIBRATION_PHASE_N

该命令从 EVM 读取校准值。

备注

对于该 EVM，仅支持 CMDH = 0xD6。

7.3.12.1 命令格式

表 7-13. HOST_CMD_GET_CALIBRATION_PHASE_N

	命令			响应		
LEN	2			30		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0xD5 + n	0	U8	0xD5+n (n = 1、2、3)
CMDL	1	U8	0xdp (在该 EVM 中始终为 0x00)	1	U8	0x80 0xdp , d = 器件 [0..7] , p = 相位 [1..15]
				2	S16	电压通道直流偏移
				4	U16	以 1/64μF 为单位的入口电容
				6	S32	电流通道直流偏移
				10	U32	电压通道交流偏移
				14	U32	电流通道交流偏移
				18	S16	以 1/1024Ts 为单位的相位校正
				20	U16	V _{rms} 比例因子
				22	U16	以 1/256 Ω 为单位的导线电阻
				24	U16	I _{rms} 比例因子
				26	U16	保留
				28	U16	功率比例因子

7.3.13 HOST_CMD_SET_CALIBRATION_EXTRAS

这命令 EVM 使用列出的值设置额外的校准值。在执行该命令之前，请确保已读取、保存和擦除包含校准值的闪存页面。

7.3.13.1 命令格式

表 7-14. HOST_CMD_SET_CALIBRATION_EXTRAS

	命令			响应		
LEN	10			2		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0xD5	0	U8	0xD5
CMDL	1	U8	0x00	1	U8	0x80
	2	U16	校准状态			
	4	U16	截距温度			
	6	U16	截距温度			
	8	U16	温度斜率/度			

7.3.14 HOST_CMD_GET_CALIBRATION_EXTRAS

该命令从 EVM 读取额外的校准值。

7.3.14.1 命令格式

表 7-15. HOST_CMD_GET_CALIBRATION_EXTRAS

	命令			响应		
LEN	2			10		
	偏移	宽度	数据	偏移	宽度	数据
CMDH	0	U8	0xDA	0	U8	0xDA
CMDL	1	U8	0x00	1	U8	0x80
				2	U16	校准状态
				4	U16	截距温度
				6	U16	温度截距
				8	U16	温度斜率/度

8 固件和嵌入式计量库 API

8.1 引言

有一个经预先测试的固件可使 EVM 正常运行。提供了源代码和嵌入式计量库（作为示例应用程序）。本节汇总了嵌入式计量库的 API。

8.2 嵌入式计量库 API

本节介绍嵌入式计量库的 API。

计量库利用函数调用、回调和应用程序级校准函数与应用程序通信。函数调用是用于访问嵌入式计量提供的功能和读数的实际调用。回调是从嵌入式计量库调用到用户应用程序的预定义函数。应用程序级校准函数是用户应用程序中用于访问校准参数和定义默认校准参数的函数。

8.2.1 嵌入式计量库函数调用

“metrology-readings.h”和“metrology-foreground.h”中定义了用于连接计量库的函数调用，可访问以下功能：

- 计量引擎控制，包括：
 - 初始化计量库和 ADC 硬件
 - 使用默认校准参数初始化计量库
- 使用后台处理收集的数据执行计量计算
- 读取计量计算的结果

8.2.1.1 用于计量引擎控制的函数

8.2.1.1.1 用于计量引擎控制的函数

int metrology_init (void)

说明

检查校准参数存储器中的 PGAIN 是否为 0xFFFF，如果为真，则将默认校准参数复制到校准参数存储器中。否则跳过复制。

参数

无

返回

如果使用默认校准参数初始化校准参数存储器，则为 0xFFFF，否则为 0x0000。

int metrology_init_from_nv_data (void)

说明

使用加载到适当存储器变量的校准参数初始化计量引擎的直流滤波器，初始化导线电阻补偿的比例因子和 EMI 电容器补偿的比例因子。

参数

无

返回

始终为 0x0000。

void align_metrology_with_calibration_data (void)

说明

使用适当的相位校正重新初始化与计量相关的 ADC 通道。

参数

无

返回

无。

void metrology_switch_to_normal_mode (void)

说明

通过适当的相位校正初始化与计量相关的 ADC 通道，将系统设置为在正常模式下运行。

参数

无

返回

无。

void metrology_init_analog_front_end_normal_mode (void)

说明

初始化所有与计量相关的 ADC 通道的硬件。

参数

无

返回

无。

void metrology_disable_analog_front_end (void)

说明

禁用所有与计量相关的 ADC 通道的硬件。

参数

无

返回

无。

8.2.1.2 计量引擎初始化过程

- metrology_init () - 使用默认值初始化未初始化的校准参数
- metrology_disable_analog_front_end () - 防止计量 ADC 在初始化期间生成中断
- metrology_init_from_nv_data () - 初始化直流滤波器
- 执行需要禁用 ADC 中断的其他设置操作
- metrology_switch_to_normal_mode () - 切换至正常模式，初始化计量相关 ADC

8.2.1.3 用于计算和读取读数的函数

8.2.1.3.1 用于计算和读取读数的函数

power_t calculate_phase_readings (void)

说明

当接收到指示时，应用程序必须显式调用该函数，以确保读数正确。该函数使用在后台进程中收集的数据执行非时间关键型计量计算。返回后，会更新所有计量读数。

参数

无

返回

有功功率读数。

power_t active_power (int ph)**说明**

获取相位 ph 的最新有功功率读数。

参数

ph = 1

返回

相位 ph 的最新有功功率。

power_t reactive_power (int ph)**说明**

返回相位 ph 的最新无功功率读数。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

相位 ph 的最新有功功率

power_t apparent_power (int ph)**说明**

返回最新计算的相位 ph 的视在功率读数。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

相位 ph 的最新有功功率

power_t fundamental_active_power(int ph)**说明**

返回最新计算的相位 ph 的基波有功功率读数。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

最新的基波有功功率。

power_t fundamental_reactive_power(int ph)**说明**

返回最新计算的相位 ph 的基波无功功率读数。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

最新的基波无功功率。

power_factor_t power_factor (int ph)**说明**

返回相位 ph 的最新功率因数读数。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

相位 ph 的最新功率因数。

rms_voltage_t rms_voltage (int ph)**说明**

返回相位 ph 的最新均方根电压。

参数

ph = 1

返回

在相位 ph 上测得的最新均方根电压。

rms_voltage_t fundamental_rms_voltage(int ph)**说明**

返回相位 ph 的最新基波均方根电压。

参数

ph = 1

返回

在相位 ph 上测得的最新基波均方根电压。

thd_t voltage_thd(int ph)**说明**

返回相位 ph 的电压的最新 THD。

参数

ph = 1

返回

相位的电压的最新 THD。

rms_current_t rms_current (int ph)**说明**

返回相位 ph 的最新均方根电流。

参数

ph = 1

返回

相位 ph 的最新均方根电流。

rms_current_t fundamental_rms_current(int ph)**说明**

返回相位 ph 的最新基波均方根电流。

参数

ph = 1

返回

相位 ph 的最新基波均方根电流。

thd_t current_thd(int ph)**说明**

返回相位 ph 的电流的最新 THD。

参数

ph = 1

返回

相位的电流的最新 THD。

int16_t mains_frequency (int ph)**说明**

返回相位 ph 的测量交流频率。当 EVM 在直流测量模式下运行时，该函数的返回值无效。

参数

ph = 1

返回

在相位 ph 上测得的最新交流频率。

uint16_t phase_status (int ph)**说明**

返回相位 ph 的状态字。

参数

ph = 1

返回

相位 ph 的状态字。

8.2.2 嵌入式计量库回调

嵌入式计量库使用回调来调用应用程序中定义的函数作为事件指示。回调在“[emeter-metrology.h](#)”中声明，必须在应用程序代码中提供函数实现（即使是空函数也是可以接受的，以防不需要处理事件）。在提供的应用程序示例中，回调函数在“[emeter-main.c](#)”中实现。

- **void BACKGROUND_PROCESS_ON (void)**
后台进程启动时调用
- **void BACKGROUND_PROCESS_OFF (void)**
后台进程完成时调用
- **void FOREGROUND_PROCESS_ON (void)**
前台进程启动时调用
- **void FOREGROUND_PROCESS_OFF (void)**
前台进程完成时调用
- **void ZERO_CROSS_ON (void)**
电压信号从正到负过零时调用
- **void ZERO_CROSS_OFF (void)**
电压信号从负到正过零时调用
- **void AC_MODE_ON (void)**
进入交流测量模式时调用
- **void AC_MODE_OFF (void)**
离开交流测量模式时调用
- **void DC_MODE_ON (void)**
进入直流测量模式时调用
- **void DC_MODE_OFF (void)**
离开直流测量模式时调用
- **void active_energy_pulse_start (void)**
能量脉冲输出从空闲状态变为脉冲状态时调用。
- **void active_energy_pulse_end (void)**
能量脉冲输出从脉冲变为空闲时调用

8.2.3 应用程序级校准函数

8.2.3.1 用于读取和写入校准参数的函数

8.2.3.1.1 用于读取和写入校准参数的函数

int get_calibration_status (void)

说明

返回仪表的校准状态。

嵌入式计量库未定义返回值。解释将由应用程序定义。

void set_calibration_status (int value)

说明

将校准状态设置为特定的值并保存到校准参数存储器中。

嵌入式计量库未定义参数值。解释将由应用程序定义。

int clear_calibration_data (void)**说明**

清除校准参数存储器中的所有校准数据

备注

必须在写入任何校准数据之前执行该操作，因为所有校准数据都存储在同一个闪存页面中。在清除校准数据之前，应读取并备份所有校准参数。写入任何校准数据时，应将修改后的值与所有其他已备份且未修改的值一起写入。

int16_t get_temperature_intercept (void)**说明**

获取温度截距的校准值。

参数

无。

返回

校准参数存储器中校准温度的校准 ADC 读数。

int16_t get_temperature_slope (void)**说明**

获取温度斜率的校准值。

参数

无。

返回

校准参数存储器中每 1°C 增量的校准 ADC 读数。

void set_temperature_parameters (int16_t temperature_at_calibration, int16_t temperature_sensor_intercept, int16_t temperature_sensor_slope)**说明**

将与温度相关的校准值写入校准参数存储器。

- temperature_at_calibration：校准完成时的温度，以 0.01°C 为单位。
- temperature_sensor_intercept：校准温度下的 ADC 读数。
- temperature_sensor_slope：每 1°C 的 ADC 读数增量。

参数

无。

calibration_scaling_factor_t get_P_scaling (int phx)**说明**

获取相位 phx 的功率比例因子。

参数

phx = 1

返回

校准参数存储器中的功率比例因子 (有关 `calibration_scaling_factor_t` 的类型定义 , 请参阅 `metrology-types.h`) 。

void set_P_scaling (int phx, calibration_scaling_factor_t value)

说明

使用 `value` 写入相位 `phx` 的功率比例因子。

参数

phx = 1

值

要写入校准参数存储器中功率比例因子的值。

返回

无。

calibration_scaling_factor_t get_V_rms_scaling (int phx)**说明**

获取相位 phx 的电压比例因子。

参数

phx = 1

返回

校准参数存储器中的电压比例因子 (有关 calibration_scaling_factor_t 的类型定义, 请参阅 metrology-types.h) 。

void set_V_rms_scaling (int phx, calibration_scaling_factor_t value)**说明**

使用 value 写入相位 phx 的电压比例因子。

参数

phx = 1

值

要写入校准参数存储器中电压比例因子的值。

返回

无。

int16_t get_v_dc_estimate (int phx)**说明**

获取相位 phx 的有效 (动态) 电压通道 ADC 直流偏移。

参数

phx = 1

返回

相位 phx 的动态电压通道直流偏移, 以 ADC 计数表示。

int16_t get_initial_v_dc_estimate (int phx)**说明**

获取相位 phx 的初始电压通道 ADC 直流偏移。

参数

phx = 1

返回

存储在校准参数存储器中的相位 phx 电压通道直流偏移, 以 ADC 计数表示。

void set_v_dc_estimate (int phx, int16_t value)

说明

使用 value 写入相位 phx 的电压通道 ADC 直流偏移。

参数

phx = 1

值

要写入校准参数存储器中相位 phx 的电压通道直流偏移的值，以 ADC 计数表示。

返回

无。

int32_t get_v_ac_offset (int phx)

说明

获取相位 phx 的电压通道交流偏移。

参数

phx = 1

返回

要存储在校准参数存储器中的相位 phx 的电压通道交流偏移，以 ADC 计数的平方表示。

```
void set_v_ac_offset (int phx, int32_t value)
```

说明

使用 `value` 写入相位 `phx` 的电压通道交流偏移。应按照 [方程式 24](#) 所示计算值：

$$V_AC_OFFSET = \text{int} \left(V_NOISE \left(\frac{1024 \times 10^3}{VGAIN} \right) \right)^2 \quad (24)$$

参数

`phx = 1`

值

要写入校准参数存储器中相位 `phx` 的电压通道交流偏移的值，以 ADC 计数的平方表示。

返回

无。

```
calibration_scaling_factor_t get_i_rms_scaling(int phx);
```

说明

获取相位 `phx` 的电流比例因子。

参数

`phx = 1`

返回

校准参数存储器中的电流比例因子（有关 `calibration_scaling_factor_t` 的类型定义，请参阅 `metrology-types.h`）。

```
void set_i_rms_scaling(int phx, calibration_scaling_factor_t value);
```

说明

使用 `value` 写入相位 `phx` 的电流比例因子。

参数

`phx = 1`

值

要写入校准参数存储器中电流比例因子的值。

返回

无。

```
int32_t get_i_dc_estimate(int phx);
```

说明

获取相位 `phx` 的有效（动态）电流通道 ADC 直流偏移。

参数

phx = 1

返回

相位 phx 的动态电流通道直流偏移，以 ADC 计数表示。

int32_t get_initial_i_dc_estimate(int phx)**说明**

获取相位 phx 的初始电流通道 ADC 直流偏移。

参数

phx = 1

返回

存储在校准参数存储器中的相位 phx 电流通道直流偏移，以 ADC 计数表示。

```
void set_i_dc_estimate(int phx, int32_t value);
```

说明

使用 value 写入相位 phx 的电流通道 ADC 直流偏移

参数

phx = 1

值

要写入校准参数存储器中相位 phx 的电流通道直流偏移的值，以 ADC 计数表示。

返回

无。

```
int32_t get_i_ac_estimate(int phx);
```

说明

获取相位 phx 的电流通道交流偏移。

参数

phx = 1

返回

要存储在校准参数存储器中的相位 phx 的电流通道交流偏移，以 ADC 计数的平方表示。

```
void set_i_ac_offset (int phx, int32_t value)
```

说明

使用 value 写入相位 phx 的电流通道交流偏移。应按照[方程式 25](#) 所示计算值：

$$I_AC_OFFSET = \text{int} \left(I_NOISE \left(\frac{1024 \times 10^6}{IGAIN} \right) \right)^2 \quad (25)$$

参数

phx = 1

值

要写入校准参数存储器中相位 phx 的电流通道交流偏移的值，以 ADC 计数的平方表示。

返回

无。

```
uint16_t get_compensate_capacitor_value (int phx)
```

说明

获取相位 phx 的 EMI 滤波电容器值。

参数

phx = 1

返回

校准参数存储器中要补偿的电容 (以 $1/64\mu\text{F}$ 为单位) 。

void set_compensate_capacitor_value (int phx, uint16_t value)

说明

使用 value 写入相位 phx 的 EMI 滤波电容器值。

参数

phx = 1

值

要写入校准参数存储器的电容 (以 $1/64\mu\text{F}$ 为单位) 。

返回

无。

uint16_t get_compensate_resistance (int phx)**说明**

获取相 phx 的导线电阻值。

参数

phx = 1

返回

校准参数存储器中要补偿的导线电阻 (以 $1/256\ \Omega$ 为单位) 。

void set_compensate_resistance (int phx, uint16_t value)**说明**

使用 value 写入相位 phx 的导线电阻值。

参数

phx = 1

值

要写入校准参数存储器的导线电阻 (以 $1/256\ \Omega$ 为单位) 。

返回

无。

int16_t get_phase_corr (int phx)**说明**

获取 phx 的相位校正。

参数

phx = 1

返回

校准参数存储器中的相位校正 (以 $0.9765625\ \mu\text{s}$ 为单位) 。

void set_phase_corr (int phx, int16_t value)**说明**

使用 value 写入相位 phx 的相位校正。

参数

phx = 1

值

要写入校准参数存储器的相位校正 (以 $0.9765625\ \mu\text{s}$ 为单位) 。

返回

无。

8.2.4 设置默认校准参数

作为计量的一部分，但不是计量本身，校准参数默认值读取和操作函数作为应用程序级源代码提供。用户定义的默认参数在“metrology-calibration-template.h”和“metrology-capacitor-compensation.h”中定义。

备注

文件“metrology-calibration-defaults.c”为要置于闪存中的默认值提供源代码级支持。除非完全理解代码，否则不要修改该文件中的代码。

- **DEFAULT_TEMPERATURE_INTERCEPT**

校准温度下的 ADC 读数，采用 16 位标度

- **DEFAULT_TEMPERATURE_SLOPE**

每 1°C 增量的 ADC 计数

- **DEFAULT_ROOM_TEMPERATURE**

以 0.01°C 为单位的校准温度

- **DEFAULT_BASE_PHASE_A_CORRECTION**

由硬件引起的电压和电流通道之间的相位校正。

- **DEFAULT_P_SCALE_FACTOR_A**

功率比例因子

- **DEFAULT_V_RMS_SCALE_FACTOR_A**

电压比例因子

- **DEFAULT_V_DC_ESTIMATE_A**

电压通道 ADC 的直流偏移

- **DEFAULT_V_AC_OFFSET_A**

电压通道 ADC 上估算噪声水平的平方（很小而难以察觉，通常为 0）。

- **DEFAULT_V_RMS_SCALE_FACTOR_A**

电流比例因子

- **DEFAULT_I_DC_ESTIMATE_A**

电流通道 ADC 的直流偏移

- **DEFAULT_I_AC_OFFSET_A**

电压通道 ADC 上估算噪声水平的平方（通常设置为 0），可通过以下公式进行计算：

$$\text{DEFAULT_I_AC_OFFSET} = \left(\frac{1024 \times 10^6}{\text{Current Scaling Factor}} \times \text{Estimated noise level} \right)^2 \quad (26)$$

- **DEFAULT_EMI_FILTER_CAP_UF_A**

以 1/64μF 为单位的 EMI 滤波器电容，最大为 1023 (15.984375μF)

如果将值设置为大于 0x8000，则不执行任何 EMI 滤波器补偿。

- **DEFAULT_WIRE_RESISTANCE_A**

以 1/256 Ω 为单位的估算导线电阻，最大为 255 (0.99609375 Ω)

- **METER_NAME**

定义由 HOST_CMD_GET_METE_NAME 报告的仪表名称，在提供的示例代码“MSP430I2040SUBMETEREVM”中定义。

- **METER_SOFTWARE_VERSION**

定义由 HOST_CMD_GET_METE_VER 报告的仪表软件版本。

- **METER_METROLOGY_VERSION**

定义由 HOST_CMD_GET_METE_VER 报告的仪表计量库版本。

- **METER_PROTOCOL_VERSION**

定义由 HOST_CMD_GET_METE_VER 报告的仪表协议版本。

9 示例应用程序代码

9.1 引言

工程结构：

- `emeter-communication.c` - 低级 UART 通信例程的源代码，包括 UART 端口设置、通过 UART 进行写入和读取、按字节发送和接收的中断服务例程
- `emeter-dlt645.c` - 轮询模式协议实现的源代码
- `emeter-main.c` - 用于系统初始化、主循环、回调函数实现和中断矢量放置的源代码。
- `emeter-metrology-i2041.r43` - 嵌入式计量库目标代码
- `emeter-setup.c` - 用于低级系统初始化的源代码
- `emeter-template.h` - 用于配置的源代码
- `metrology-calibration-default.c` - 用于将用户定义的默认校准参数放入适当的数据结构的源代码
- `metrology-calibration-template.h` - 用户定义的默认校准参数的源代码
- `emeter-autoreport.c` - 用于执行自动报告的源代码。

备注

要启用自动报告支持，请取消注释 `emeter-template.h` 中的 `#define AUTOREPORT_SUPPORT` 定义。

9.2 准备要运行的应用程序代码

1. 启动 IAR5.5 Embedded Workbench IDE，然后点击“File”→“Open”→“Workspace”。
2. 当提示打开工作区时，选择“`emeters.eww`”。

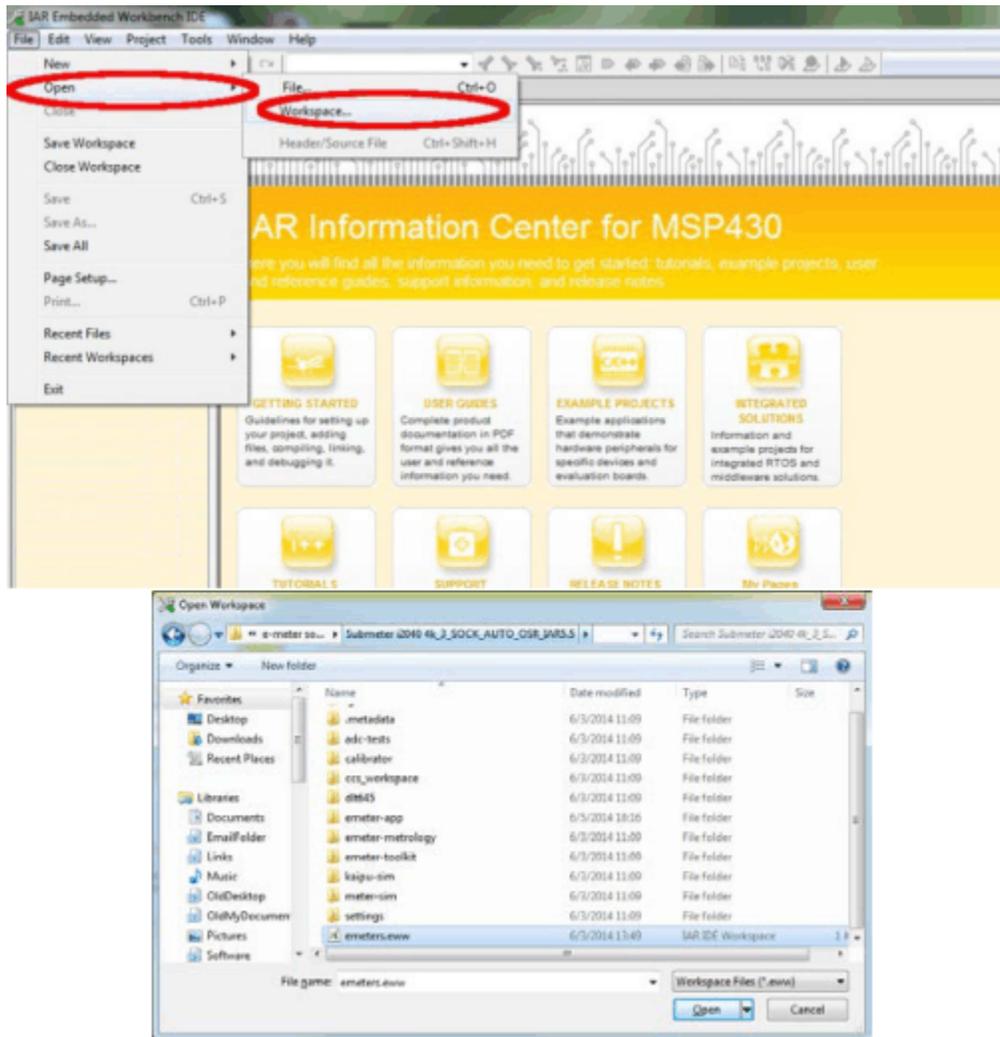


图 9-1. 打开工作区

3. 选择“Workspace”窗口底部的“emeter-app-i2041”工程选项卡。
4. 通过右键点击工程名称并从弹出菜单中选择“Options…”来检查工程选项，如图 9-2 所示。

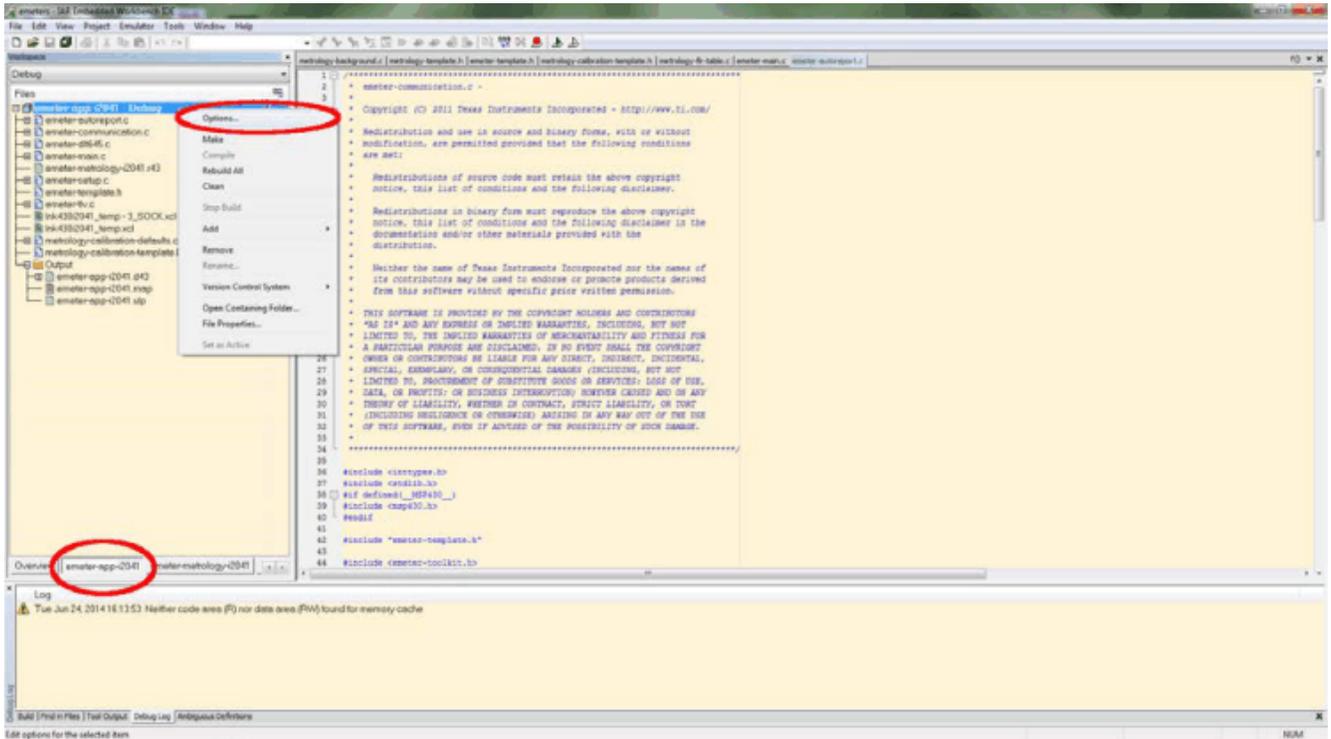


图 9-2. 工程选项卡

- 显示选项后，选择左侧列中的“C/C++ Compiler”，然后选择右侧的“Optimizations”选项卡，检查优化设置是否如图 9-3 所示。

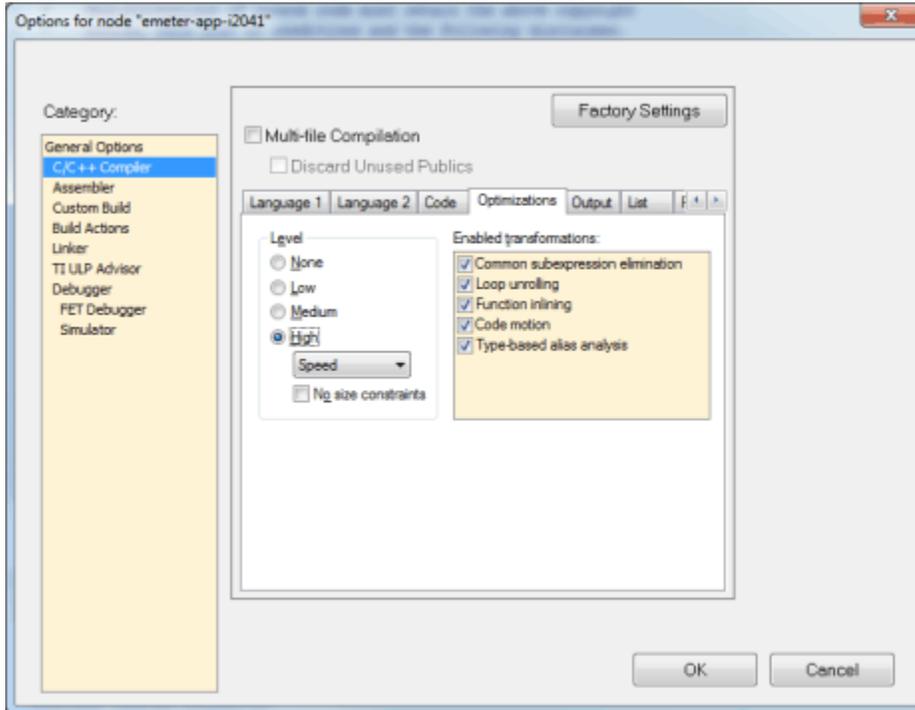


图 9-3. 优化选项

6. 在“Setup”选项卡的左侧列中选择“FET Debugger”。EVM 使用 Spy-Bi-Wire 进行代码下载和调试。检查以确保选项如图 9-4 所示。

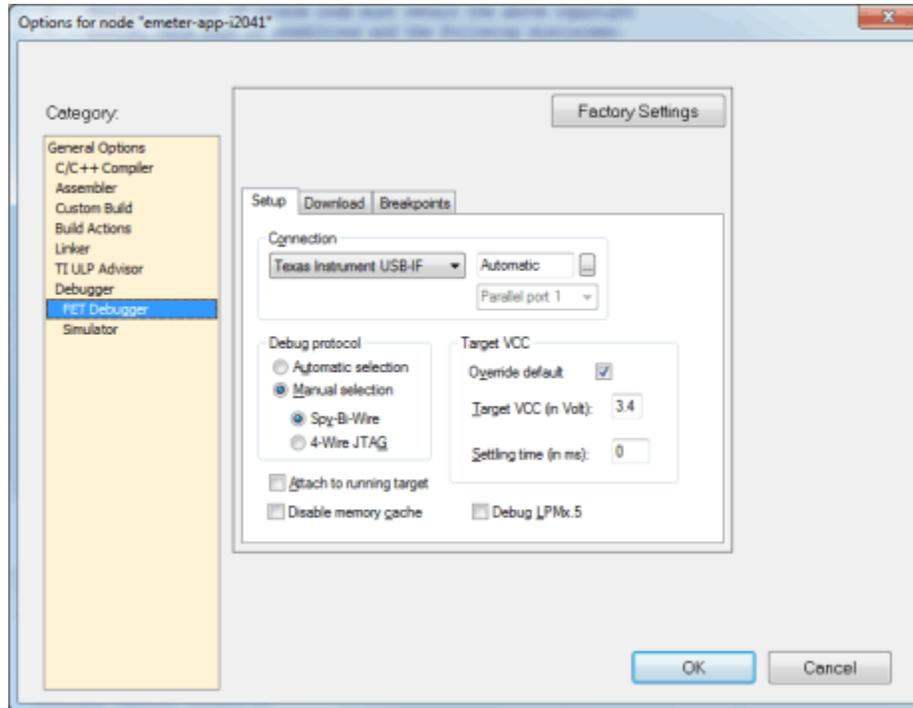


图 9-4. 调试器选项

7. 选择“Download”选项卡。在“Flash erase”下，请勿选择“Erase main and Information memory”；该选项会擦除两组数据，无法恢复。相反，请选择“Erase main memory”作为下载选项以保留以下出厂设置参数：系统时钟校准、ADC 校准和内部基准校准（图 9-5）。但是，存储在主存储器中的计量校准（例如 VGAIN、IGAIN、PGAIN 等）始终会在下载后被擦除。选择“Erase main memory”作为下载选项以保留出厂设置参数。

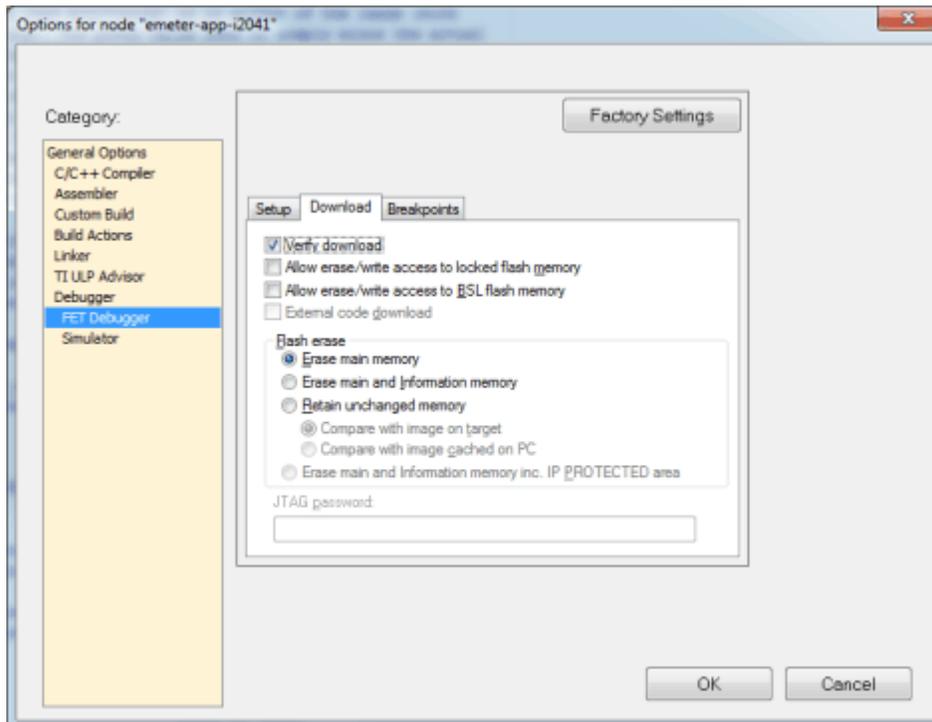


图 9-5. 下载选项

8. 完成所有更改后，点击“OK”。

- 通过右键单击工程并从弹出菜单中选择“Rebuild All”来重新编译工程（请参阅图 9-6）。在重新编译过程中会报告三条警告（请参阅图 9-6），可以放心地忽略这三条警告。要打开工程工作区，修改代码，编译和下载代码，必须使用有效的许可证安装 IAR Embedded Workbench® 5.5。如果没有可用的有效许可证，仍然可以下载目标代码。有关下载过程，请参阅节 9.3。

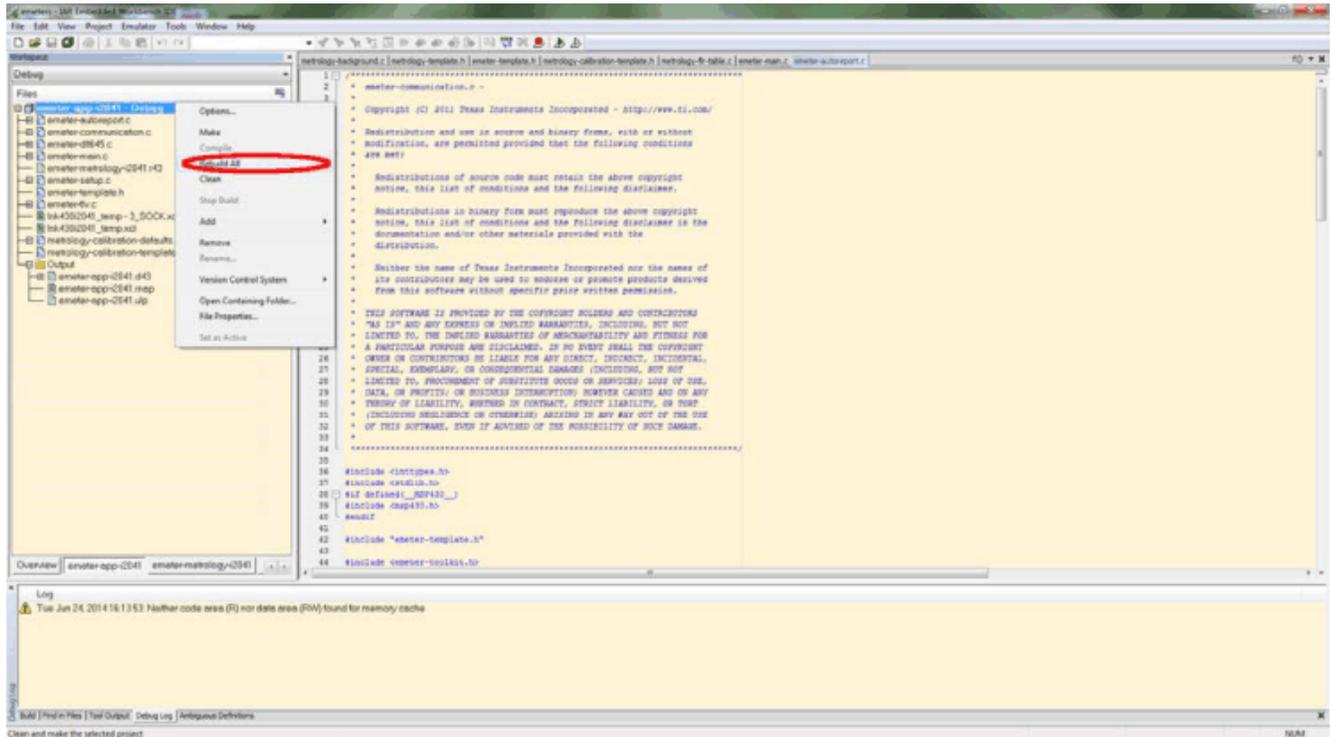


图 9-6. 编译应用程序

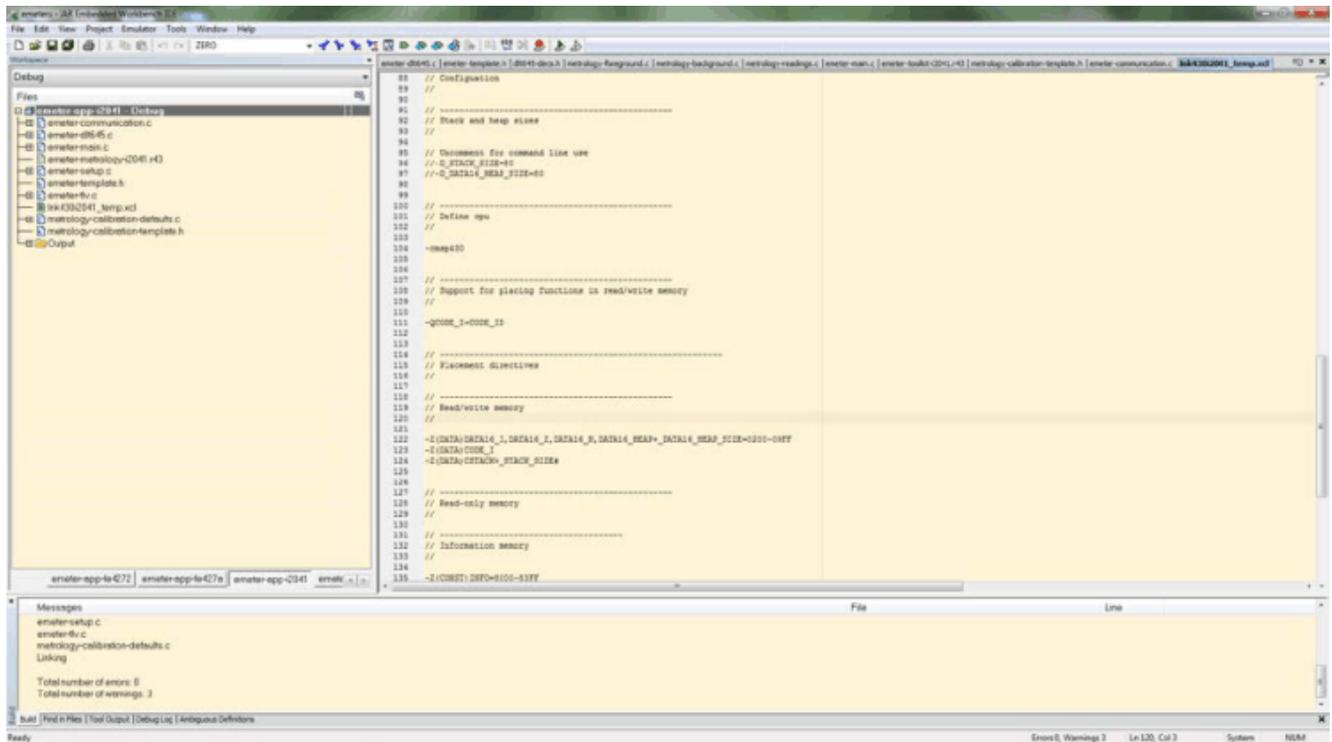


图 9-7. 警告

10. 确保 J8 上的跳线正确短接。通过一条扁平电缆将 14 引脚连接器 P1 连接到 MSP-FET430UIF，如图 9-8 所示。

WARNING

调试接口未隔离；请确保在连接交流或直流高电压时在 EVM 和用于调试的 PC 之间实现适当的隔离。

对于 EVM 运行，连接调试接口是可选的。EVM 可以在不连接调试器的情况下独立运行。

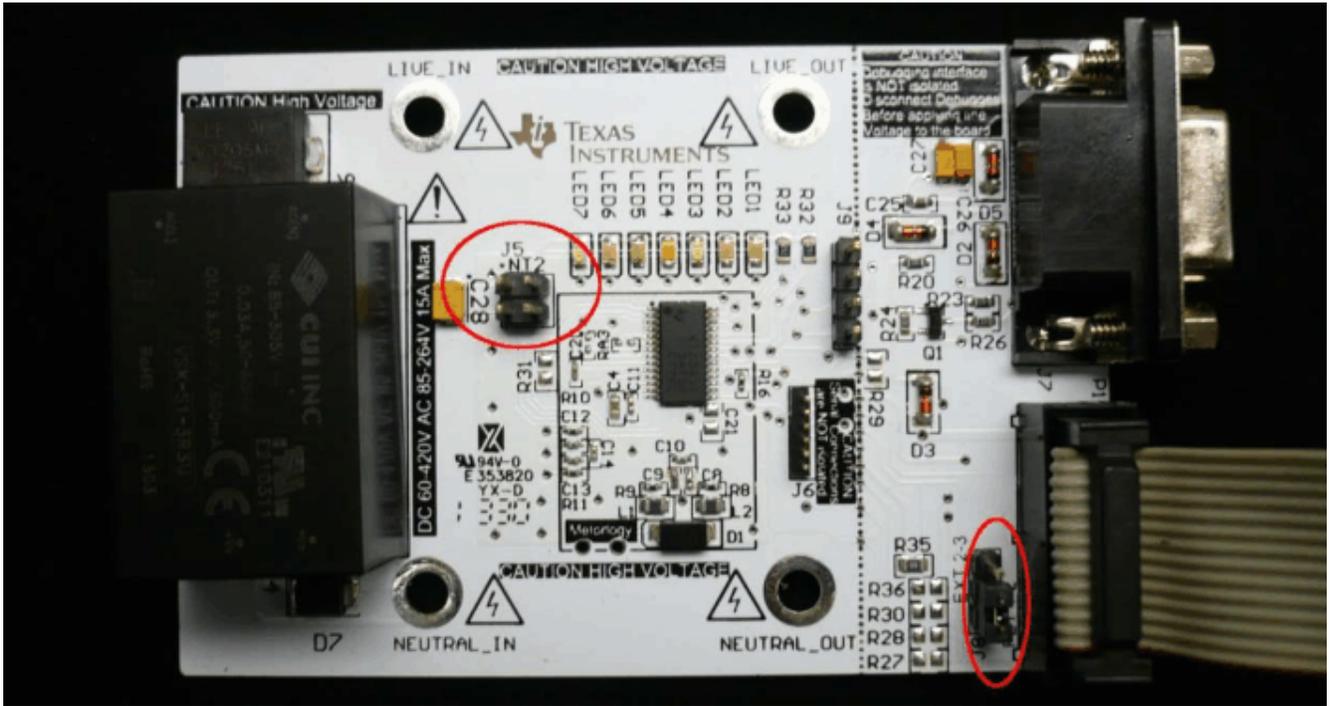


图 9-8. 连接 EVM 和 FET

11. 点击“Download and Debug”按钮开始下载和调试，如图 9-9 所示。

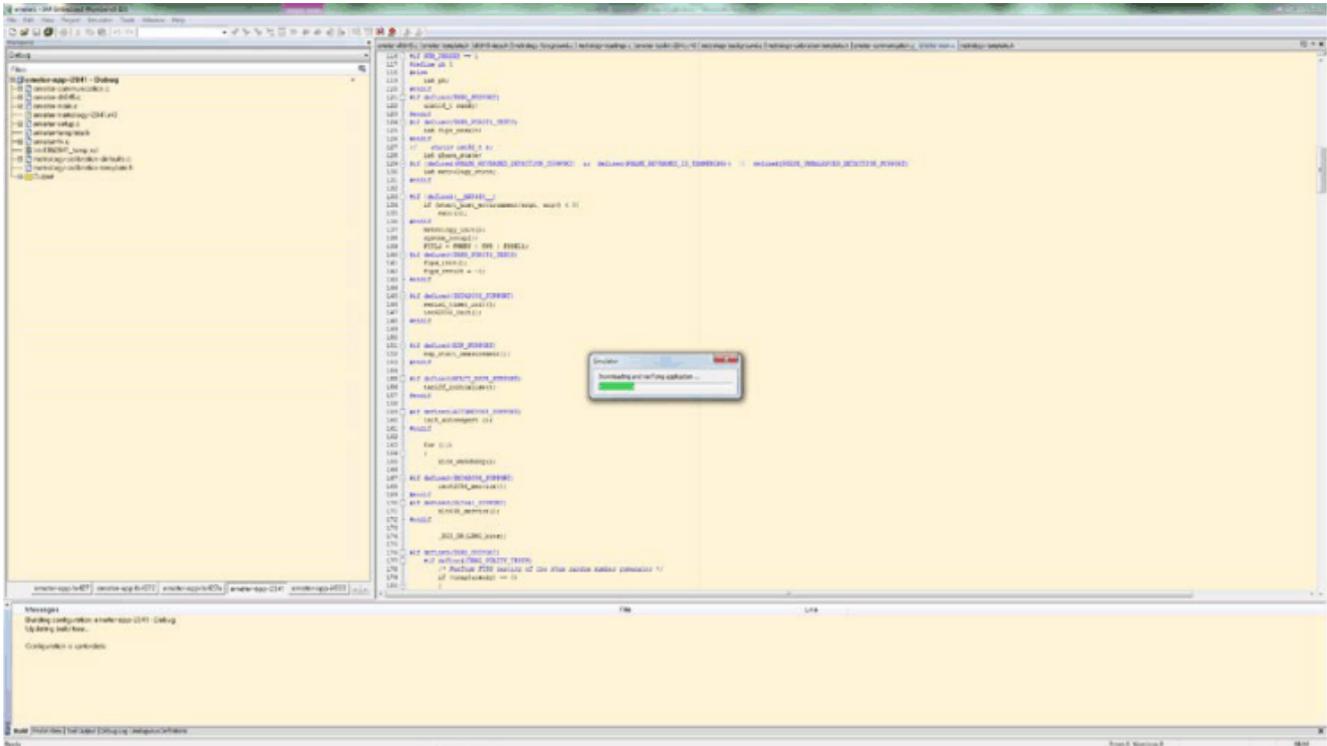


图 9-9. 代码下载

12. 下载完成并成功之后，将显示如图 9-10 所示的屏幕。点击“Go”开始运行应用程序。

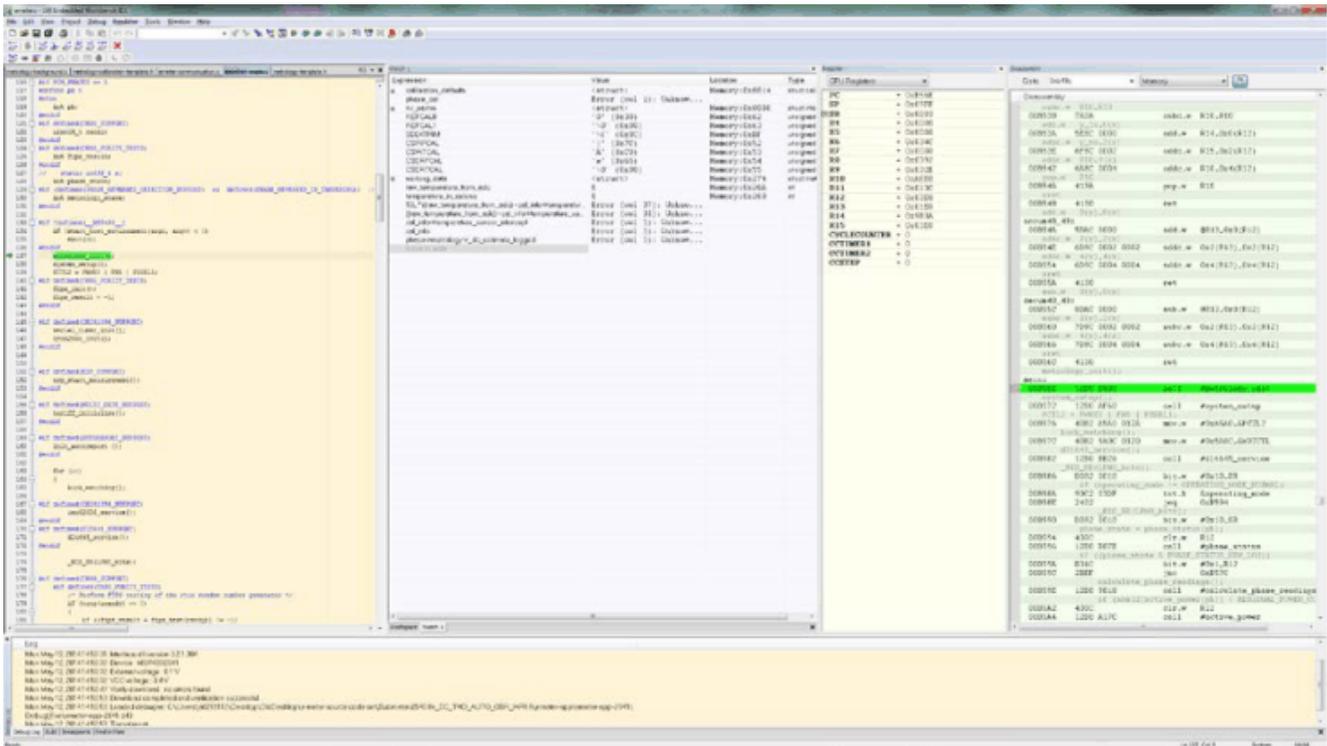


图 9-10. 调试器屏幕

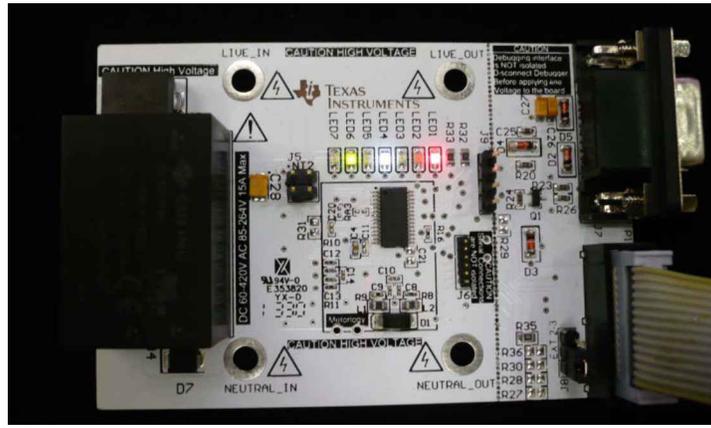


图 9-11. EVM 运行

9.3 在没有 IAR 许可证的情况下下载

按照以下步骤使用安装的 IAR Embedded Workbench 5.5 (如果没有有效的 IAR Embedded Workbench 5.5 许可证) 下载可执行代码：

1. 打开工程工作区，如节 9.2 中所述：步骤 - 。将电路板连接至 MSP-FET430UIF，如步骤 (节 9.2) 所述。
2. 然后从菜单中选择“Project”→“Download”→“Download File...”，如图 9-12 所示。

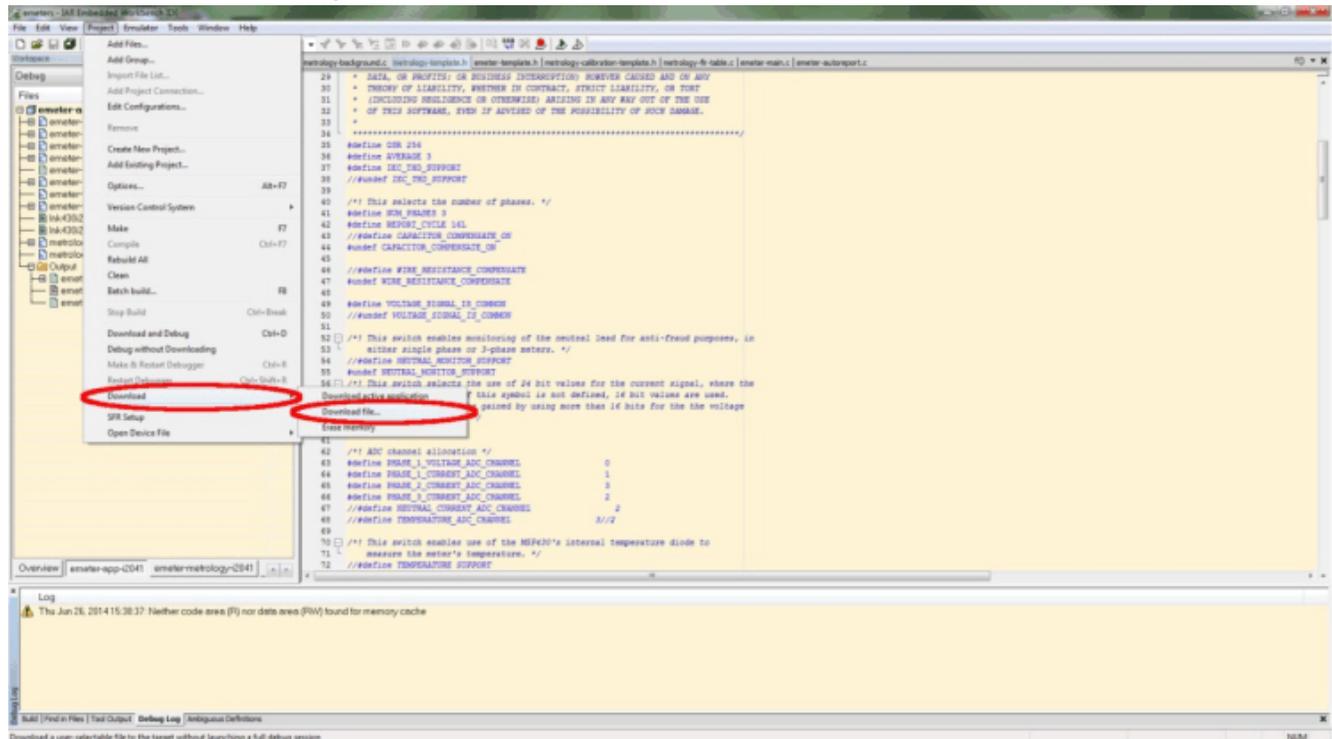


图 9-12. 下载可执行文件

- 当系统提示选择文件时，从文件夹 `Submeter i2040 4k_3 SOCK_AUTO_OSR_IAR5.5\emeter-app\emeter-app-i2041\Debug\Exe` 中进行选择。该文件名为“emeter-app-i2041.d43”，如图 9-13 所示。

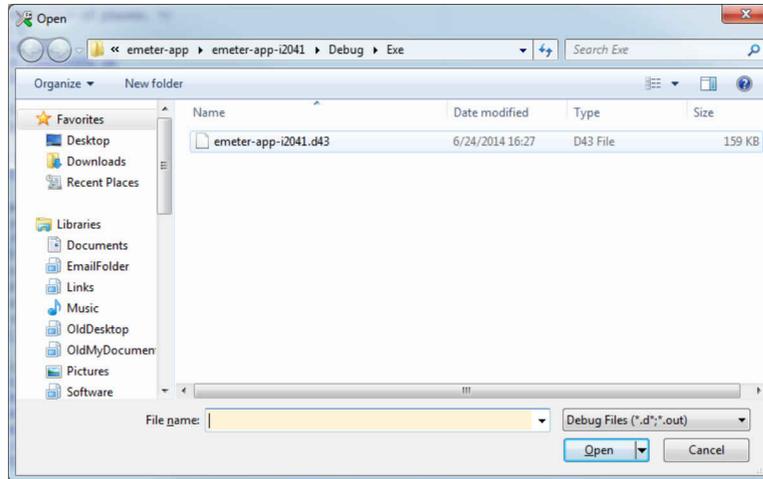


图 9-13. 选择要下载的文件

- 可执行代码将下载到电路板上。

10 硬件设计文件

10.1 封装

与硬件相关的文件包含在一个可下载软件包中，该软件包是一个名为“hardware.zip”的压缩文件，可解压缩为名为“Hardware”的文件夹。包括以下文件：

- PDF 格式的原理图
- Altium Designer 格式的原理图 CAD 文件
- Altium Designer 格式的 PCB 布局 CAD 文件
- 物料清单
- 光绘文件格式的 PCB 布局文件

10.2 原理图

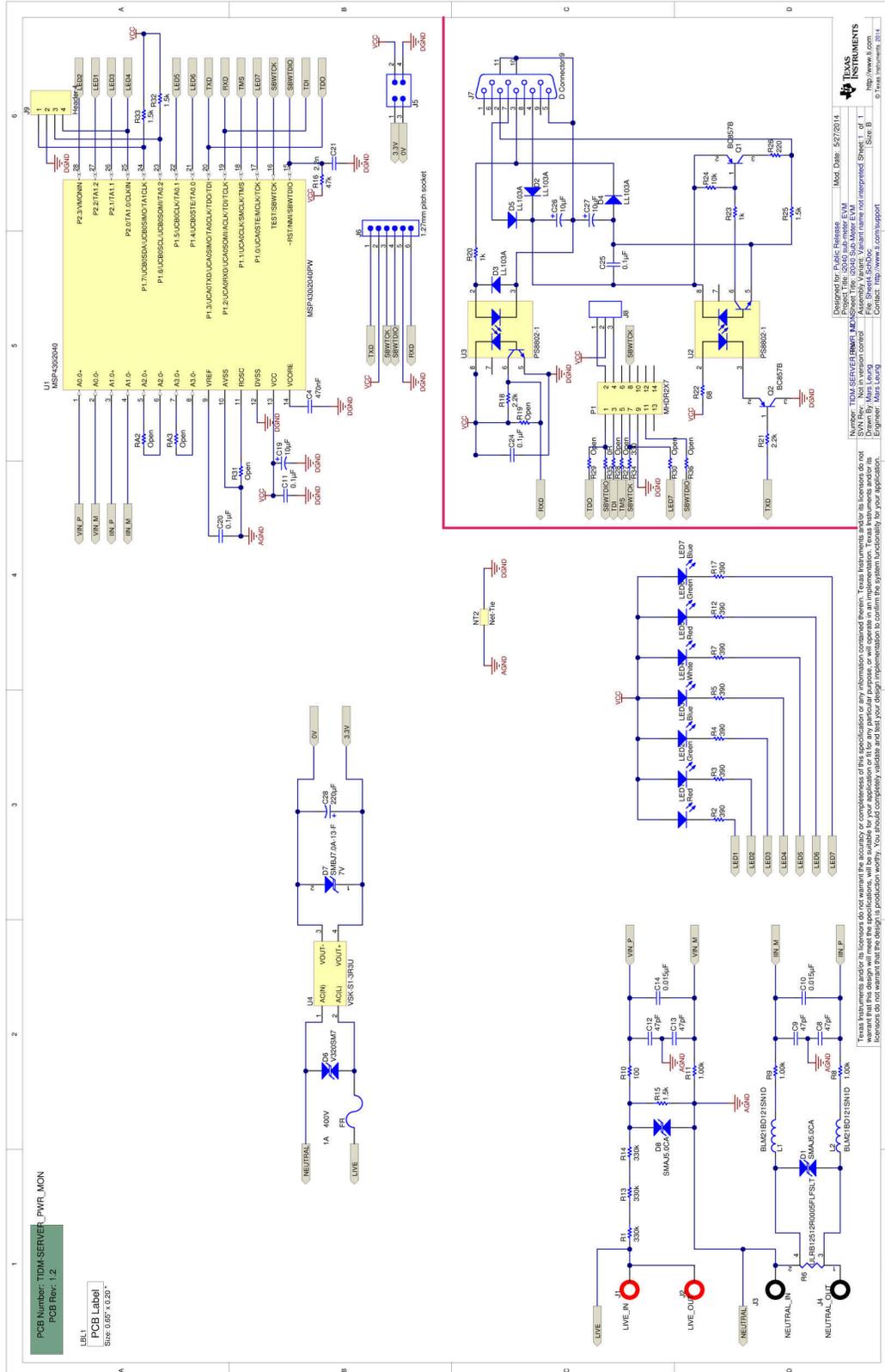


图 10-1. EVM 原理图

11 EVM 规格和性能

11.1 EVM 规格

- 电压工作范围 (使用提供的电源) : 85 - 265VAC、120 - 380VDC
- 采样率 : 8000 Hz
- 采样位深度 : 24 位
- 支持轮询报告
- 更新速率 : 4 个交流周期 (交流模式)、80ms (直流模式)
- UART 通信数据速率 : 9600bps
- 交流/直流测量模式开关 : 4 个交流周期 (交流转直流)、80ms (直流转交流)
- 测量范围 : RMS 电压、RMS 电流、有功功率、无功功率、视在功率、功率因数、线路频率、温度
- 测量电压范围 : 0V - 265V_{rms} 交流、0 - ±380VDC (使用参考设计电路和元件值)
- 测量电流范围 : 0 - 15Arms 交流、0 - ±22.5ADC (使用参考设计电路和 0.5mΩ 分流器阻值)
- 电压分辨率 : 1mV
- 电流分辨率 : 1μA
- 有功功率分辨率 : 1mW
- 无功功率和视在功率分辨率 : 1mW
- 功率因数分辨率 : 0.001
- EVM 性能
- 室温下的功率精度

PF = 1		在 7.50212A、1650.7W 下校准					
电流基准	电流读数	电流误差	电流 %	功率基准	功率读数	功率误差	功率 %
0.014555	0.014760	0.000205	1.408%	3.202	3.213	0.011	0.355%
0.029637	0.029737	0.000100	0.339%	6.517	6.515	-0.002	-0.029%
0.074836	0.074950	0.000114	0.152%	16.579	16.615	0.036	0.217%
0.145378	0.145391	0.000013	0.009%	31.984	31.992	0.009	0.027%
0.295964	0.295992	0.000028	0.009%	65.114	65.168	0.054	0.083%
0.747332	0.747569	0.000237	0.032%	165.104	165.256	0.152	0.092%
1.500050	1.500000	-0.000050	-0.003%	330.060	330.314	0.254	0.077%
2.988460	2.989000	0.000540	0.018%	662.136	662.586	0.450	0.068%
7.502390	7.500000	-0.002390	-0.032%	1650.380	1650.779	0.399	0.024%
14.350000	14.320000	-0.030000	-0.209%	3090.240	3084.544	-5.696	-0.184%
19.321400	19.242000	-0.079400	-0.411%	4162.290	4143.000	-19.290	-0.463%

PF = 0.5L		在 7.50212A、1650.7W 下校准					
电流基准	电流读数	电流误差	电流 %	功率基准	功率读数	功率误差	功率 %
0.014558	0.014654	0.000096	0.659%	1.555	1.559	0.004	0.257%
0.029628	0.029774	0.000146	0.492%	3.173	3.168	-0.005	-0.158%
0.074823	0.074824	0.000001	0.001%	8.037	8.055	0.018	0.218%
0.145166	0.145141	-0.000025	-0.017%	13.287	13.298	0.011	0.083%
0.295495	0.295481	-0.000014	-0.005%	27.587	27.589	0.002	0.007%
0.747391	0.747393	0.000002	0.000%	71.399	71.492	0.093	0.130%
1.500100	1.500000	-0.000100	-0.007%	165.134	165.379	0.245	0.148%
2.988390	2.989000	0.000610	0.020%	319.346	319.745	0.399	0.125%
7.402320	7.400000	-0.002320	-0.031%	596.562	596.954	0.392	0.066%
14.371200	14.354000	-0.017200	-0.120%	890.268	889.965	-0.303	-0.034%
20.021600	19.914000	-0.107600	-0.537%	2197.230	2187.520	-9.710	-0.442%

PF = 0.5C		在 7.50212A、1650.7W 下校准					
电流基准	电流读数	电流误差	电流 %	功率基准	功率读数	功率误差	功率 %
0.014543	0.014671	0.000128	0.879%	1.659	1.652	-0.007	-0.422%
0.029624	0.029638	0.000014	0.046%	3.388	3.390	0.002	0.056%
0.074827	0.074848	0.000021	0.029%	8.525	8.537	0.012	0.141%
0.145314	0.145267	-0.000047	-0.032%	18.491	18.503	0.012	0.068%
0.295834	0.295752	-0.000082	-0.028%	37.381	37.391	0.010	0.027%
0.747077	0.747067	-0.000010	-0.001%	94.044	94.030	-0.014	-0.015%
1.481450	1.481000	-0.000450	-0.030%	171.561	171.526	-0.035	-0.020%
2.988510	2.989000	0.000490	0.016%	345.256	345.311	0.055	0.016%
7.401750	7.399000	-0.002750	-0.037%	1025.610	1025.054	-0.556	-0.054%
14.334700	14.318000	-0.016700	-0.117%	2196.370	2190.308	-6.062	-0.276%
19.304000	19.241000	-0.063000	-0.326%	2965.780	2951.400	-14.380	-0.485%

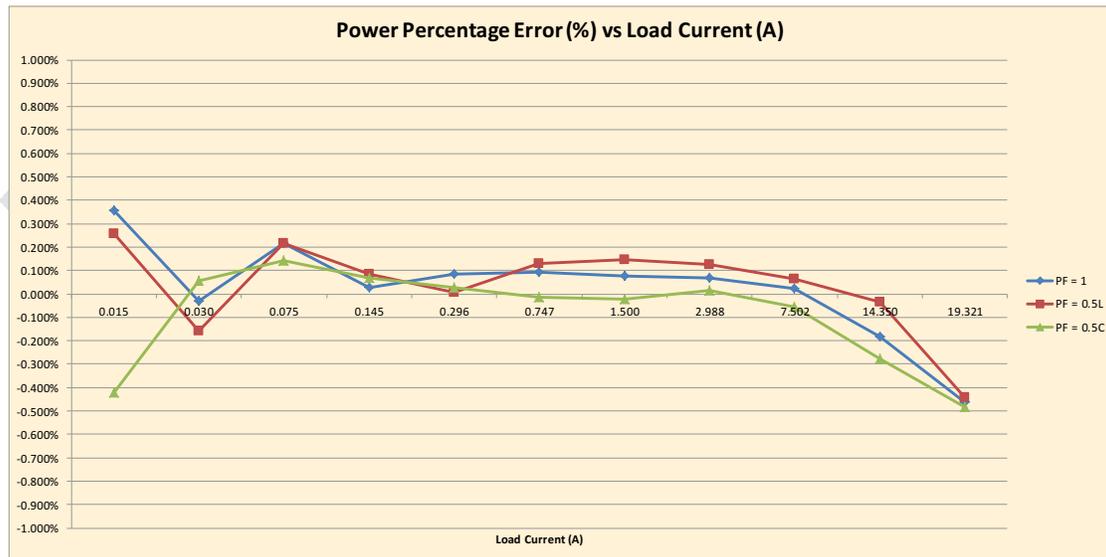


图 11-1. 功率百分比误差 % 负载电流 (A)

备注

在图 11-1 中，图形表明在较高的电流下误差开始变为负值。这种现象是分流器被流经板上引线的电流加热所致。在低电流下，较高的误差主要是由导致读数不稳定的噪声引起的。

表 11-1. 220V、5A 下精度与温度间的关系

温度/C	电压 (V)			电流 (A)			功率 (W)		
	V	Vref	V%	I	Iref	I%	P	Pref	P%
0	219.786	219.991	-0.093%	5.005	5.000	0.097%	1100.730	1099.940	0.072%
5	219.728	219.978	-0.114%	5.005	5.000	0.088%	1099.470	1099.860	-0.035%
10	219.801	219.992	-0.087%	5.005	5.000	0.082%	1100.050	1100.040	0.001%
15	219.839	219.997	-0.072%	5.004	5.000	0.072%	1100.110	1100.020	0.008%
20	219.896	220.013	-0.053%	5.003	5.000	0.053%	1100.120	1099.920	0.018%
25	219.883	219.954	-0.032%	5.002	5.000	0.025%	1100.050	1099.830	0.020%
30	219.951	220.021	-0.032%	4.998	5.000	-0.040%	1099.680	1099.930	-0.023%
35	219.990	220.060	-0.032%	4.996	5.000	-0.088%	1099.290	1099.960	-0.061%
40	220.012	220.021	-0.004%	4.993	5.000	-0.142%	1098.910	1100.060	-0.105%
45	220.060	220.054	0.003%	4.991	5.000	-0.193%	1098.330	1099.840	-0.137%
50	220.054	220.053	0.000%	4.988	5.000	-0.256%	1097.580	1099.920	-0.213%
55	220.026	219.996	0.014%	4.983	5.000	-0.343%	1096.890	1099.910	-0.275%
60	220.050	220.027	0.010%	4.979	5.000	-0.427%	1095.800	1099.890	-0.372%
65	220.009	220.051	-0.019%	4.974	5.000	-0.522%	1094.700	1100.020	-0.484%
70	219.968	219.995	-0.012%	4.969	5.000	-0.630%	1093.210	1099.940	-0.612%
75	219.896	219.973	-0.035%	4.969	5.000	-0.629%	1093.060	1100.030	-0.634%
80	219.892	219.952	-0.027%	4.969	5.000	-0.637%	1092.600	1099.960	-0.669%
85	219.895	219.957	-0.028%	4.968	5.004	-0.704%	1092.840	1099.910	-0.643%

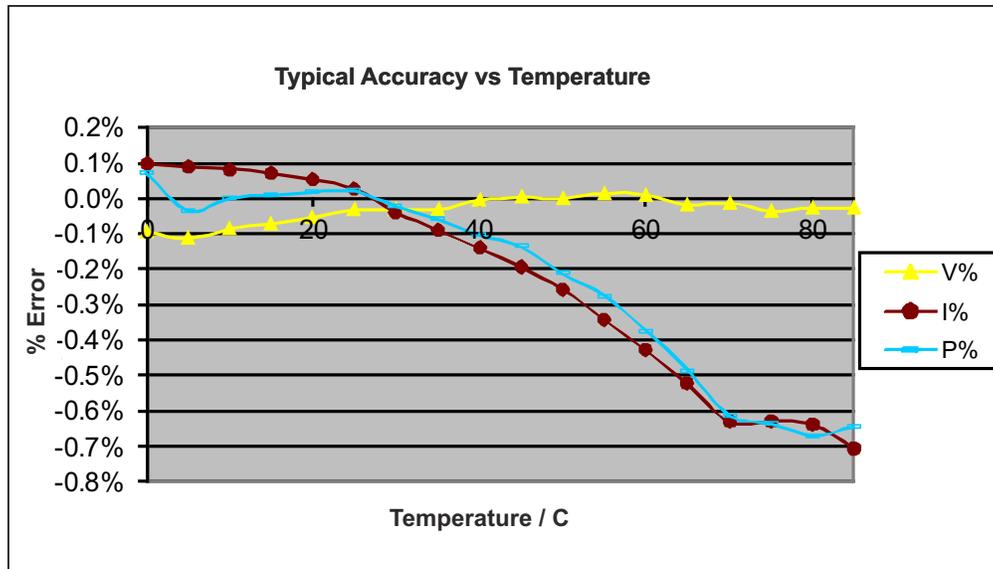


图 11-2. 典型精度与温度间的关系

备注

在图 11-2 中，图形表明在较高的温度下误差开始变为负值。这种现象是板温度升高以及整个板的组合温度系数生效所致。

12 在 MSP430i2040 和 MSP430i2041 上运行

12.1

针对 MSP430i2041 配置了示例 IAR 工程。要在 MSP430i2040 上运行，请将以下步骤执行一次：

1. 打开工作区。启动 IAR 5.5 后，选择 **File** → **Open** → **Workspace**。

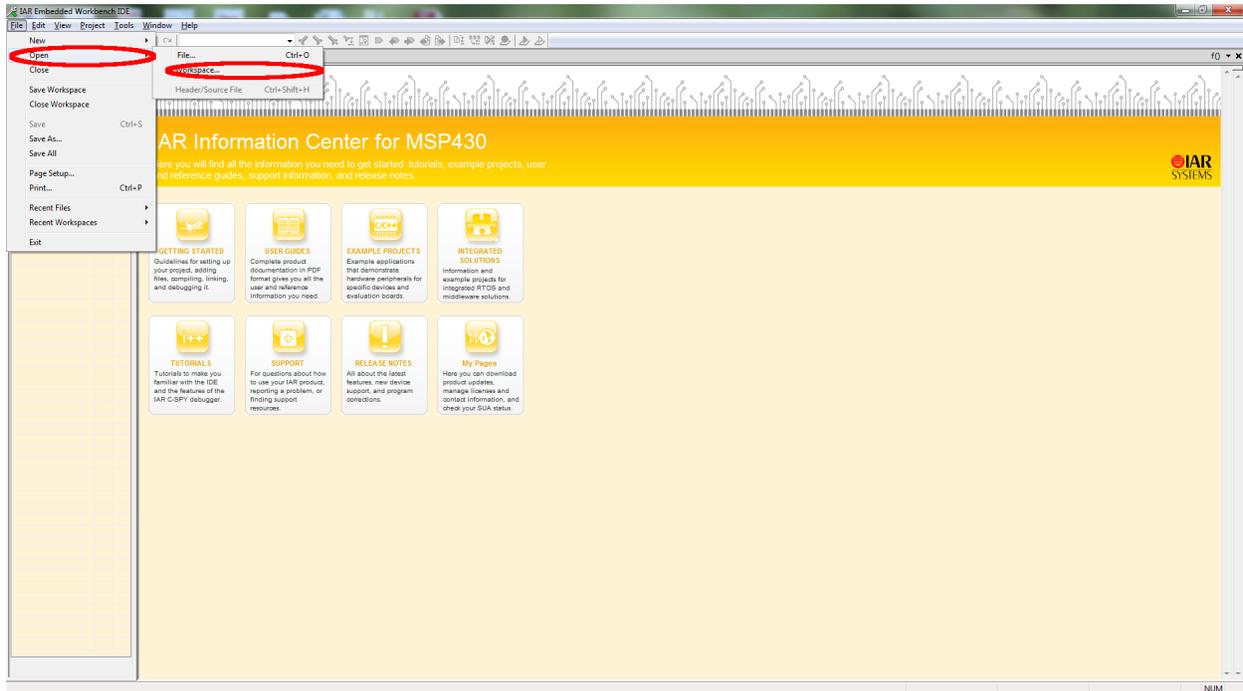


图 12-1. IAR 5.5 启动窗口

当系统提示打开工程时，从示例代码的工程目录中选择 *emeters.eww*。

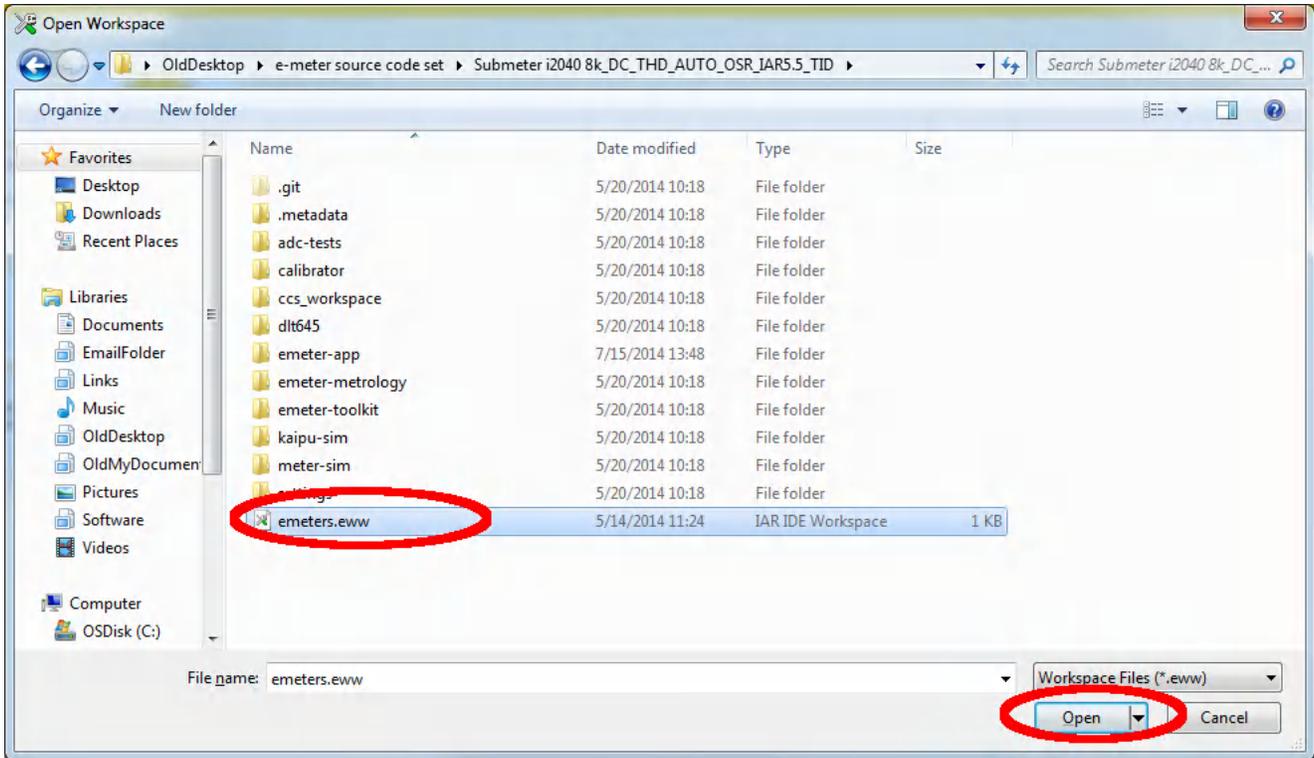


图 12-2. “Open Workspace” 窗口

2. 设置选项。右键单击工程 *emeter-app-i2041* 并选择 *Options...*。

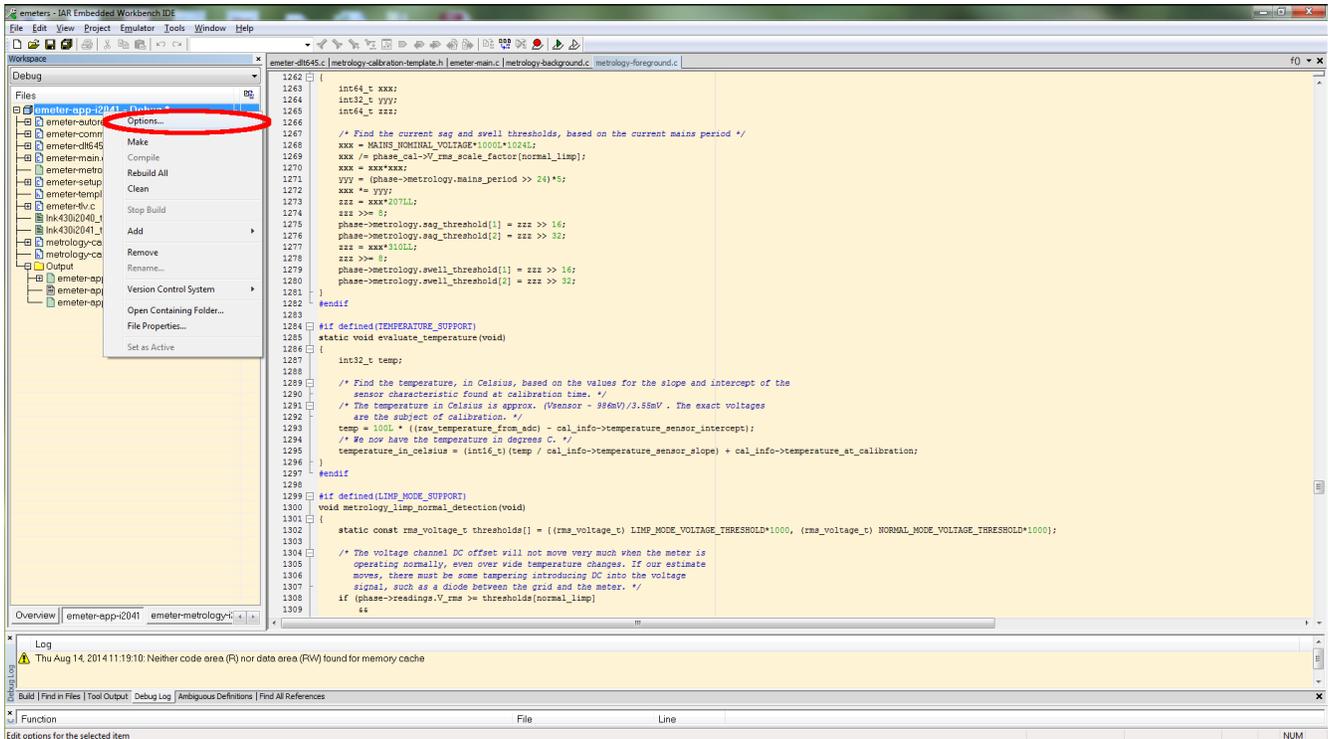


图 12-3. 工作区选项

在 **Options** 窗口中，点击左侧列中的 **General Options**，然后选择 **Target** 选项卡。然后，在 **Device** 部分中，点击输入框右侧的按钮并选择 **MSP430Ixxxx Family** → **MSP430I2040**。

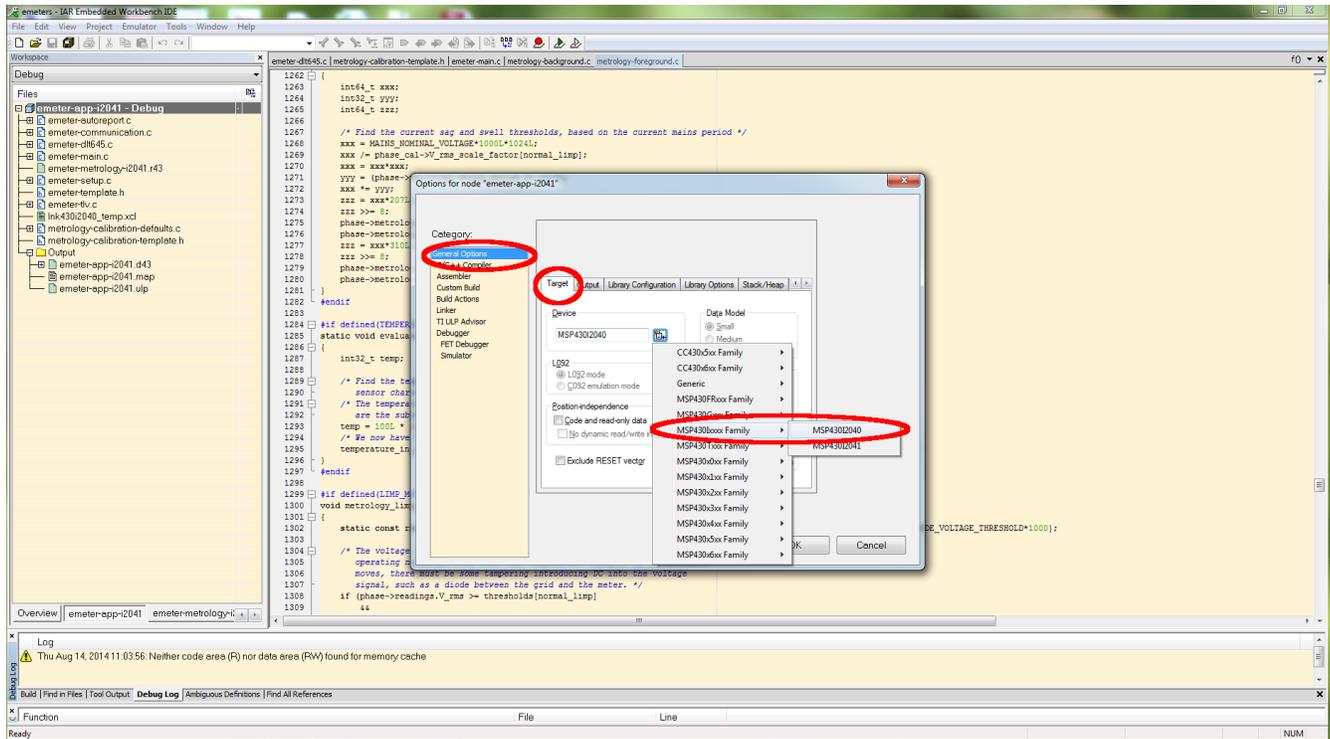


图 12-4. “Options” 窗口

点击左侧列中的 **Linker**，然后选择 **Config** 选项卡。然后点击 **Linker configuration file** 框中右侧的按钮。

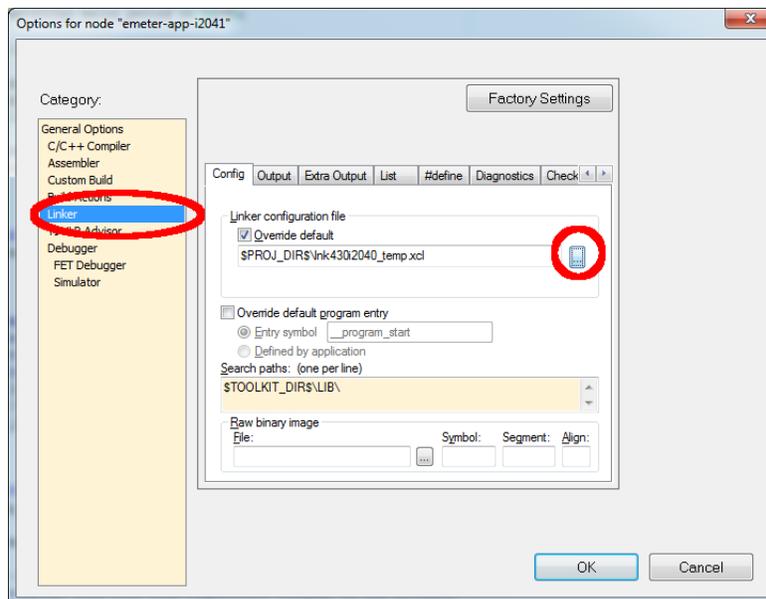


图 12-5. 配置选项卡

系统提示时选择 *Ink430i2040_temp.xcl*，然后点击 *Open*。

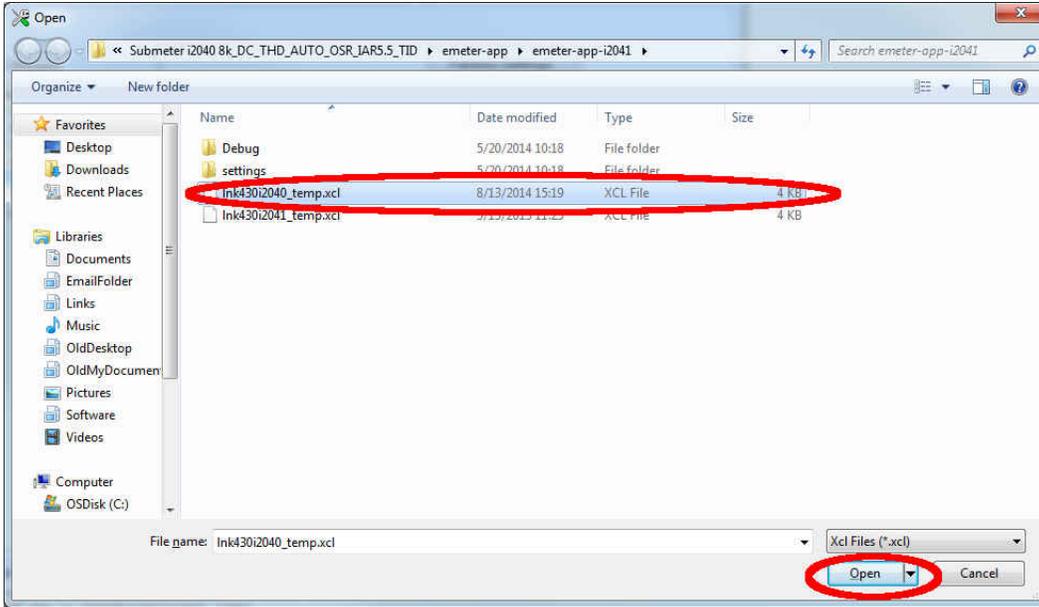


图 12-6. 选择默认 XCL 文件

返回到 *Options* 窗口后，点击 *OK*。

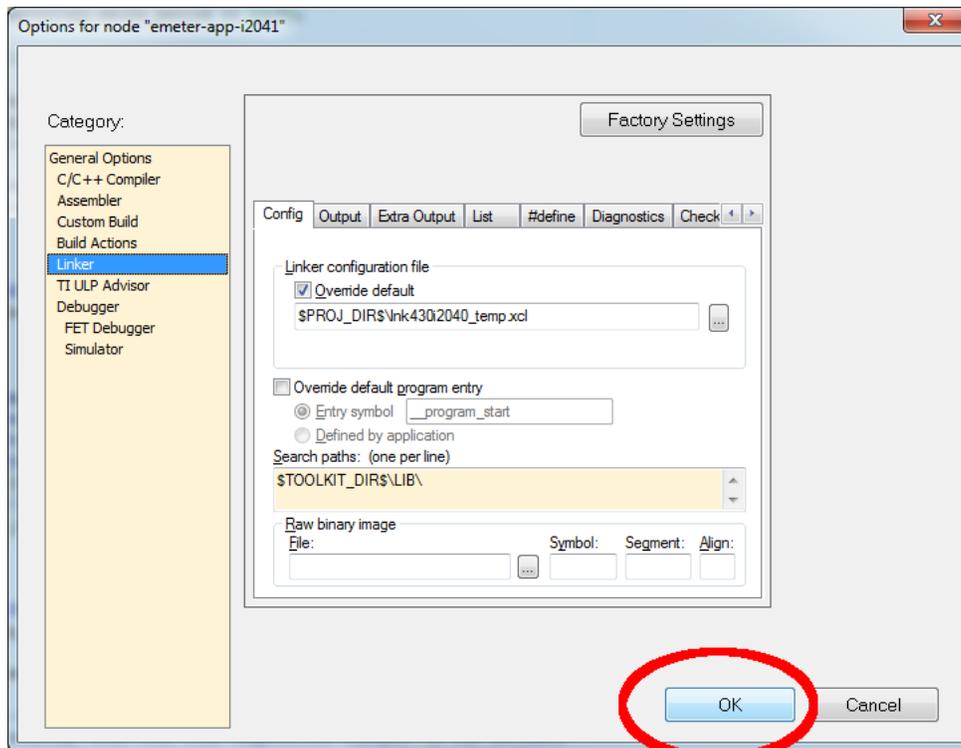


图 12-7. 完成设置更改

返回到主屏幕后，选择 **File** → **Save** 保存设置。

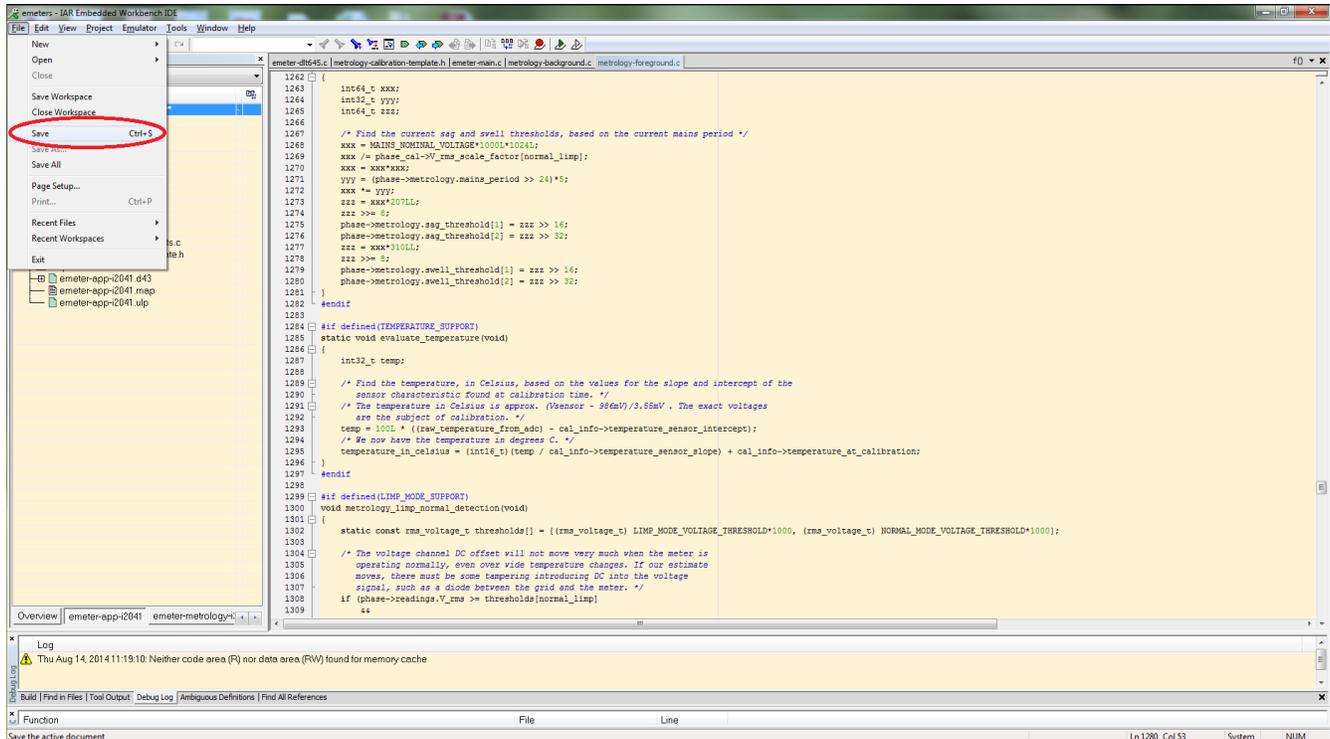


图 12-8. 保存设置更改

现在已针对在 MSP430i2040 上运行设置了该工程。要使该工程重新在 MSP430i2041 上运行，请按照相同的步骤操作，不过在“Options”窗口中选择“MSP430xxxx Family” → “MSP430i2041”，然后在系统提示时在 *Linker configuration file* 框中选择 *Ink430i2041_temp.xcl*。

13 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 * (August 2014) to Revision A (February 2023)

Page

- 更新了整个文档中的表格、图和交叉参考的编号格式。..... 1
- 删除了之前的“仪器链接”主题；存储在存储库中..... 16

重要声明和免责声明

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

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

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

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

TI 反对并拒绝您可能提出的任何其他或不同的条款。

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

Copyright © 2023，德州仪器 (TI) 公司