

TPS6594/LP8764 PMIC 应用指南（四）

Lijia Zhu

摘要

随着新能源汽车的高速发展，汽车从电动化转向智能化，高阶辅助驾驶在各大 OEM 和 Tier1 的快速落地，TDA4 正在被广泛的使用在 ADAS 域控制器，车身域控制器和激光雷达等多种终端应用。以 TPS6594/LP8764 为代表的 TI 新一代 PMIC 家族，不仅是 TDA4 SOC 的最佳供电方案，也是其他型号 SOC 供电的良好选择。TPS6594/LP8764 具有高集成度，高可拓展性，支持高级别功能安全等特性，功能多且较复杂，应用起来会有不小的难度。本系列文章会从 TPS6594/LP8764 PMIC 的芯片主要机制介绍、系统设计注意点、常见问题定位思路、自定义 PMIC 固件（NVM）等方面进行分享。本文是系列文章的第四篇，对 PMIC 芯片常用的问题定位思路进行讨论。

目录

1	引言	3
2	问题初步排查	3
3	示波器测量波形定位故障	3
4	使用 PMIC EVM 定位故障	5
4.1	使用 EVM 读取故障板的 I2C 信息	5
4.1.1	EVM 硬件介绍	5
4.1.2	硬件连接	5
4.1.3	软件安装及简单使用	8
4.1.4	中断信息的读取与保存	10
5	更改 NVM 定位问题	12
5.1	使用 Scalable PMIC GUI 软件更新 NVM	12
5.2	实际应用案例	15
6	附录-寄存器读取脚本	15
7	参考文献	17

图

Figure 1.	PDN 0A 上电时序	4
Figure 2.	LDO 短路上电波形	4
Figure 3.	TPS6594 EVM 照片	5
Figure 4.	使用 EVM 读取待测板 PMIC 寄存器示意图	6
Figure 5.	I2C 路径电流倒灌机制示意图	7
Figure 6.	I2C2 未连接时软件报错信息	8
Figure 7.	软件主界面	8
Figure 8.	寄存器读取界面	9

Figure 9.	PMIC 切换地址弹窗	10
Figure 10.	Scripting 菜单	10
Figure 11.	Scripting 窗口 1.....	11
Figure 12.	Scripting 窗口 2.....	11
Figure 13.	Begin an NVM Configuration 按钮	12
Figure 14.	NVM Configuration 界面	12
Figure 15.	Program NVM 界面 1.....	13
Figure 16.	Program NVM 界面 2.....	14
Figure 17.	Program NVM 弹窗	14
Figure 18.	修改 NVM bin 文件	15

1 引言

TPS6594/LP8764 具有丰富的功能安全机制，这导致 PMIC 正常工作所需的外部条件比较严格，哪怕是现象最简单的无法启动问题，其根因可能是五花八门的，甚至一个问题会掩盖掉另一个问题，几个问题耦合在一起作用也是家常便饭，定位起来难上加难，需要工程师逐步抽丝剥茧的排除干扰项最终找到问题的根因。

根据 PMIC 的宏观行为，本文把遇到的问题总结为以下三类：

1. PMIC 无法上电，部分电源轨或没有任何电源轨尝试启动。
2. PMIC 正常上电后因为未知原因掉电。
3. PMIC 关机后无法再次启动。

本文介绍从硬件初步排查，寄存器信息辅助定位，更改 NVM 排除干扰因素直击根因，这几方面探讨下定位 PIMC 系统问题时的思路。

2 问题初步排查

TPS6594 具有 VSYS 控制机制，无论是哪种问题，首先确认 PMIC 的 VCCA 供电是否存在这一步非常关键，这可以通过测量系统上电后 TPS6594 的 VINTLDO（2 号 pin）和 LP8764 的 VINTLDO（20 号 pin）验证，如果 VINTLDO 存在稳定的 1.8V 电压信号，则说明 VSYS 和 VCCA 之间的 NMOS 正常开启且 PMIC 的内部数字状态机正常工作，可以按照后续章节的内容进行定位。

如果 VINT 管脚无电压，说明 NMOS 没有正常开启，一般来说三种可能性：

1. 在上电到掉电的整个过程中 VSYS 或 VCCA 触发了 OVP 且没有跌落到复位电平以下。
2. VCCA 因为反向漏电导致 Fail Short BIST 失败。
3. 芯片损坏、焊接问题或 NMOS 开路，可以通过示波器观察 VSYS，OVPDRV 和 VCCA 的关系确定损坏的部分。如果时 NMOS 损坏且临时找不到替换件，可以短接 MOS 的 DS 并将芯片 VSYS SENSE 接地来旁路 VSYS SENSE 机制。

3 示波器测量波形定位故障

在确定 VCCA 存在的情况下，可以读取寄存器来判断故障出现的位置，但是读取寄存器需要搭建软硬件环境，在搭建寄存器读取环境之前，可以使用示波器测量所有的电源轨输出来对故障进行初步判断，同时波形也是定位根因的一个很好的辅助。

针对第一种问题，测试输出不是按照 BUCK1-5，LDO1-4 这样逐个测过去，更省时间的做法是根据当前 PMIC 电源方案的上电时序的顺序从前往后逐个测量，因为上电顺序在前的电源轨出现故障会阻止后续电源轨的开启。图 1 所示的是 PDN-1A（TPS65941213+LP87611B4）的上电时序，可以在 TI 官网搜索“PDN 1A”关键字得到，其他 PDN 也可以搜索对应的名字获得。PDN 文件除了上下电时序外，整个 PFSM 的设置都写得非常清楚，定位问题前一定要仔细读一下。

这里就用 PDN 1A 来举例说明，使用这个 PDN 的板子出现无法启动故障时，可能测到的波形可能有以下几种情况：

1. 电源轨#1 和#2 完全无输出，始终为低电平，说明问题出现在上电时序之前，一个比较大的可能是 SPMI BIST 失败，可以看下 SPMI 的波形是否正常。
2. 电源轨#1 和#2 可以看到完整的 15 次重启过程，这是状态机 15 次进入 SAFE RECOVERY 最终 recover counter 到达上限停在 SAFE RECOVERY 状态的表现。波形上可以看到 15 个脉宽大于 1.7ms 的脉冲，电源轨#3 和#5 可以看到 15 次脉冲，虽然脉宽较短，但是脉冲顶部都达到了合理的预期的电压。而电源轨#4 没有达到预期的高度，图 2 是这样的一个例子，这里的 15 个脉冲都只到 200mV，说明是这个电压短路了，短路原因待进一步查明。

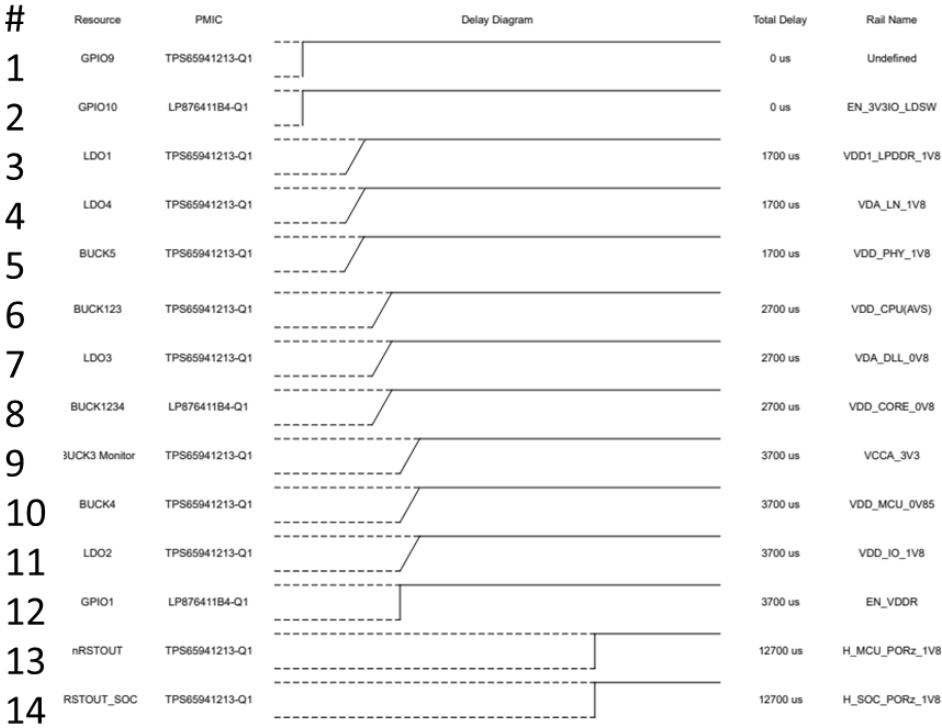


Figure 1. PDN 0A 上电时序

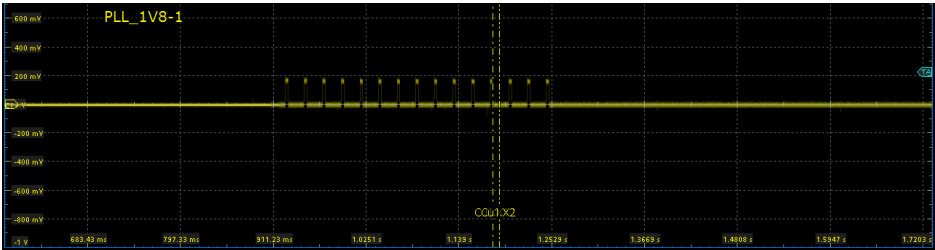


Figure 2. LDO 短路上电波形

3. 第二点里面只是一个例子，其他电源轨出问题的情况也是可能的，思路在于从前往后依次向后排查，直到看到没有正常启动的电源轨。

对于第二种问题，建议优先看寄存器，基于寄存器的结果确定出问题的电源轨或其他机制。示波器逐个排查会比较费时间。

对于第三种问题，也是和第一种一样按照上电时序先后去逐个监控，看是哪个电源轨阻塞了重启。一般是某一路电源轨上面有残压。

4 使用 PMIC EVM 定位故障

4.1 使用 EVM 读取故障板的 I2C 信息

获取 PMIC 的寄存器信息在调试过程中非常关键，然而，因为 PMIC 具有大量的功能安全机制，导致 PMIC 非常容易触发保护而关闭输出，此时板载的 SOC 没有供电，无法通过板载的 SOC 来获取 PMIC 的寄存器信息。可以使用独立的 USB-I2C 协议转换芯片（比如 FT2232），配合自有的上位机软件读取 PMIC 的寄存器信息，但是需要额外的研发成本去调试这套软硬件工具。一个更简单的方法是使用 TPS6594/LP8764 PMIC 的 EVM 配合 Scalable PMIC GUI 这个上位机软件读取 PMIC 的寄存器。

4.1.1 EVM 硬件介绍

TPS6594 EVM 的照片如图 3，板上最关键的两个器件是正面的 PMIC 和反面的 MSP432 MCU，MCU 和 PMIC 的 I2C 互联在一起，MCU 作为 I2C 主设备，PMIC 作为 I2C 从设备，PC 通过 USB 接口和 MCU 通信，MCU 转发 PC 发来的命令到 I2C 接口完成 PC 和板上 PMIC 的通信。MCU 可以读取 EVM 上面 PMIC 的 I2C，也可以通过跳线的方式读取外部待测板的 PMIC I2C 信息。

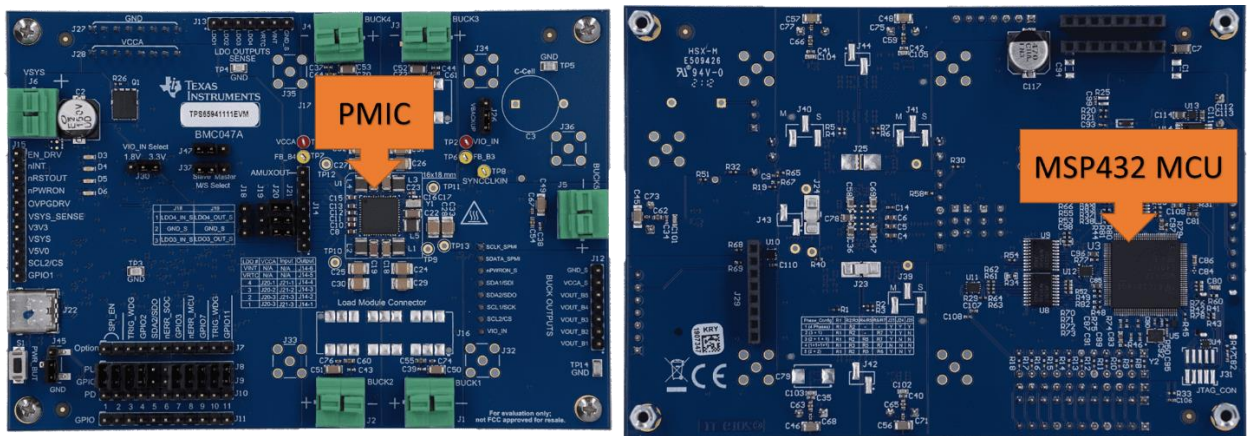


Figure 3. TPS6594 EVM 照片

4.1.2 硬件连接

通过 EVM 访问外部待测板的系统如图 4 所示，搭建方式在本节后面内容详细介绍。

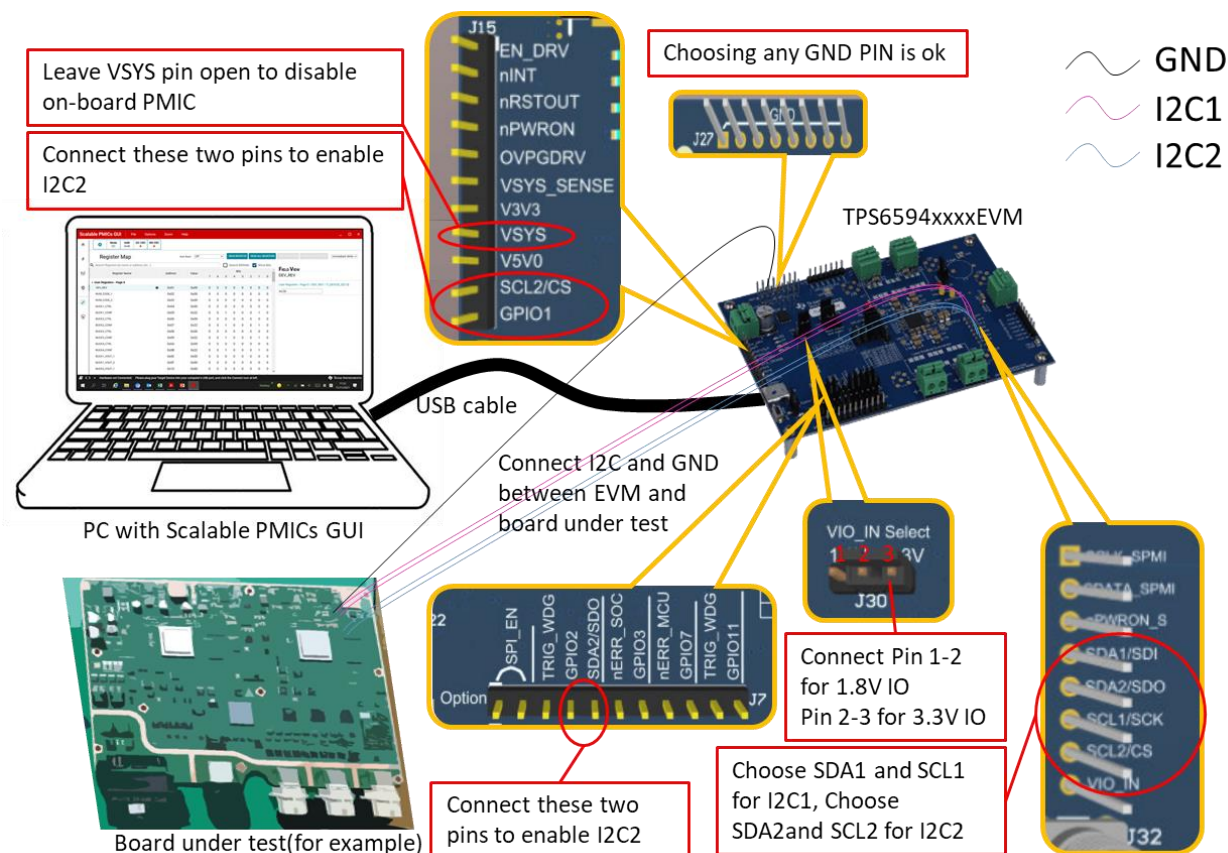


Figure 4. 使用 EVM 读取待测板 PMIC 寄存器示意图

1. 为避免 EVM 板上已有的 PMIC 芯片 I2C 地址和外部待测板子上的 PMIC I2C 地址发生冲突（使用同一个地址），需要确保 EVM 上面 J15 的 VSYS 针脚是悬空状态。若 VSYS 和 V3V3 短接（对于 5V Vin 的 PMIC 型号，则是 VSYS 和 V5V0 短接），则会使能 EVM 上面的 PMIC，造成 I2C 地址冲突读取到错误的信息。
2. 使用跳线连接 EVM 的 SDA1/SDI 和 SCL2/SCK 针脚到待测板 I2C1 的 SDA 和 SCL。使用第三根跳线使两个板子共地。
3. EVM 上的 I2C 自带 1k 上拉电阻，有 1.8V 和 3.3V 两个上拉电压值可供选择，通过切换 J30 的跳帽状态来切换。如果待测板可以正常启动，可以悬空 J30 的 2 号针脚，此时使用待测板上的 I2C 上拉电阻提供上拉电压。如果待测板无法启动或者无法确定状态（概率性启动），此时待测板无法提供所需的 I2C 上拉电压，需要插上 J30 上的跳帽，使 EVM 输出的上拉电压值和待测板设计的 I2C 上拉电压的一致，这一步电压选错可能导致损坏硬件。定位问题更多遇到的是第二种情况，因此这一步建议插上 J30 跳帽的方式，通用性会更好一点。

4. PC 和 EVM 通过 USB A to C 数据线连接, USB 为 EVM 供电, 除此之外 EVM 不需要额外的供电。一般而言, 需要先给待测板上电, 再给 EVM 供电, 否则系统大概率会起不来。原因分析如图 5 所示, 分为 (a) 和 (b) 两种情况讨论, 对于情况 (a), 待测板的 I2C 上拉到 PMIC 控制的外部开关输出, 当 EVM 先上电时, I2C 信号管腿先获的了电压, 通过外置 load switch 的寄生体二极管, 如红色虚线箭头所示的路径, 倒灌回了 PMIC 的输入端, 导致 Fail Short BIST 失败, NMOS 不开启, 待测芯片处于关机状态, 此时无法通过 I2C 访问待测板上 PMIC。对于情况 (b), 待测板的 I2C 上拉到 PMIC 的某一路输出, 如果此时 EVM 先上电, 电压就会经过红色虚线箭头的路径倒灌到 PMIC 的输出端, 阻塞残压检测机制导致无法启动, 此时待测板上 I2C 是可以工作的, 但是 PMIC 是没有输出的 (可能待定位的故障现象宏观上也是这样), 此时会因为读取到错误的寄存器信息 (比如当前连接到 I2C 上拉的那一路报 SC 故障) 而带偏问题的定位方向, 因此千万要注意!

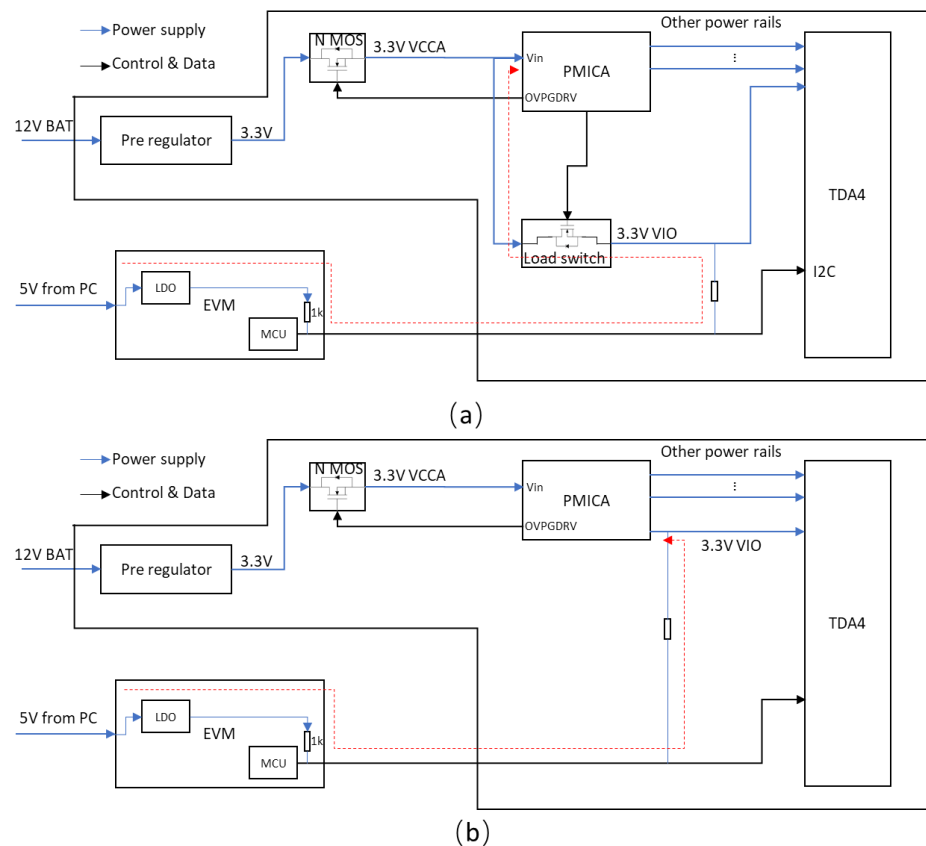


Figure 5. I2C 路径电流倒灌机制示意图

在实际定位问题过程中, 还有多次重启待测板以复现概率性问题的需求, 在这个过程中每次插拔 EVM 板是比较麻烦的事情, 此时可以摘掉待测板上 SCL 和 SDA 的上拉电阻的方式阻断反向漏电路径。

5. 部分 PMIC 型号使能了 I2C2，此时 page 4 的寄存器访问权限从 I2C1 切换到 I2C2，仅仅连接 I2C1 的话，软件读取全部寄存器会有如图 6 所示的报错信息。Page4 的寄存器是看门狗相关的设置，若待定位问题和看门狗无关，可以忽略这个告警。若要对这些寄存器进行访问，需要短接 J15 上的 SCL2 和 GPIO1 针脚，J7 上的 GPIO2 和 SDA2 针脚。



Figure 6. I2C2 未连接时软件报错信息

4.1.3 软件安装及简单使用

Scalable PMIC GUI 软件下载地址：<https://dev.ti.com/gallery/view/PMIC/Scalable-PMICs-GUI/ver/3.0.0/>。共有软件本体和 runtime 两个文件需要下载，注意根据自己电脑操作系统下载正确的版本，安装时先安装 runtime 再安装软件本体。

1. 硬件完成连接后，打开软件后界面如图 7 所示，可以直接点击对应 PMIC 型号条目右上角的 register map 按钮进入寄存器详情页。

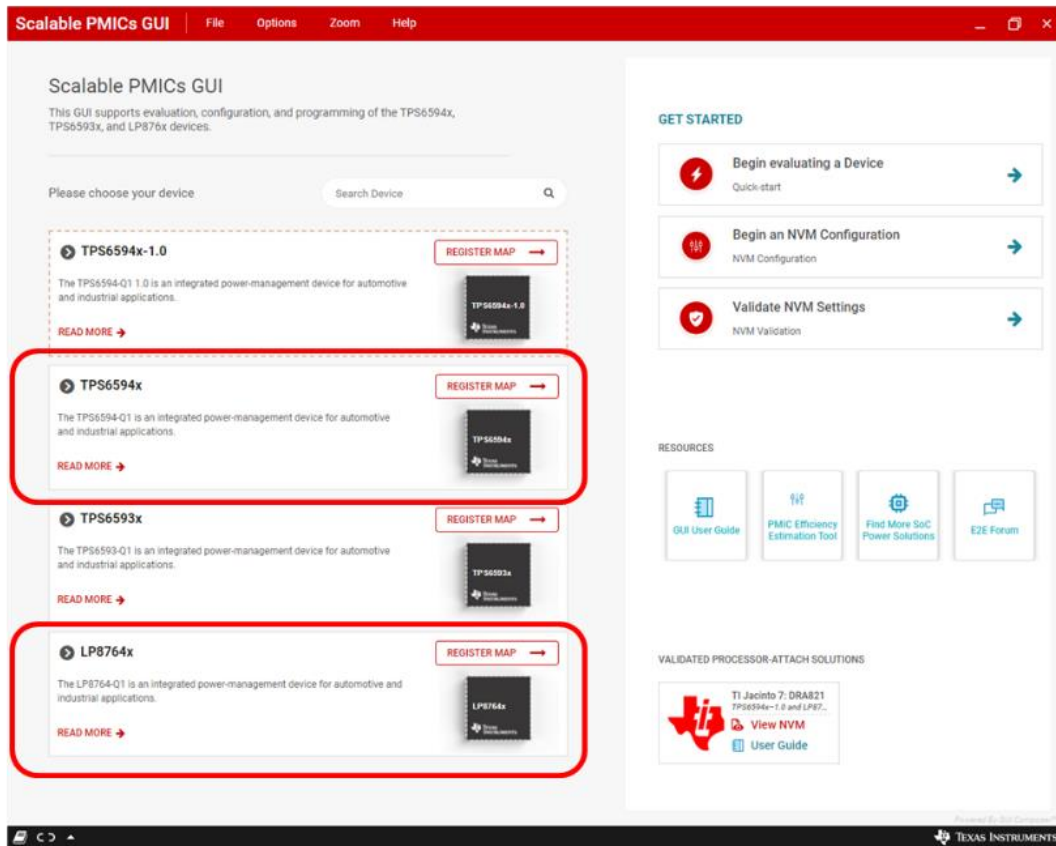


Figure 7. 软件主界面

- 进入寄存器读取页面后，软件会尝试连接 0x48 地址的 PMIC，连接成功会在下面状态栏显示：“Hardware connected”，如图 8 所示。I2C 通信建立后，可以点击 READ ALL REGISTERS 按钮手动刷新所有的寄存器数据。如果显示 “Connect to AEVM Controller but failed to connect to device xxxxx（器件型号） on DUT with I2C @ 0x48”，那么说明 MCU 和 PC 通信正常，但是 MCU 无法建立和 0x48 这个地址的 PMIC 的 I2C 通信，可能当前系统中确实不存在地址为 0x48 的 PMIC 或者硬件连接不稳定没连上 0x48 地址的 PMIC。如果需要切换 I2C 地址，可以点击页面左上角的齿轮标记，打开设置弹窗进行更改，弹窗内容如图 9 所示。需要注意，同一个板子上的多个 PMIC 具有不同的 I2C 地址和各自不同的器件型号，需要 select device 和 I2C1 Address 两个下拉框的选项到正确的设置。如果显示 “Hardware not connected, please plug your target device into your computer’s USB port and click the connect icon at left”，说明 PC 和 EVM 间的连接有问题，需要查一下 EVM 是否损坏或者软件驱动是否安装正常，可以重启电脑试下。

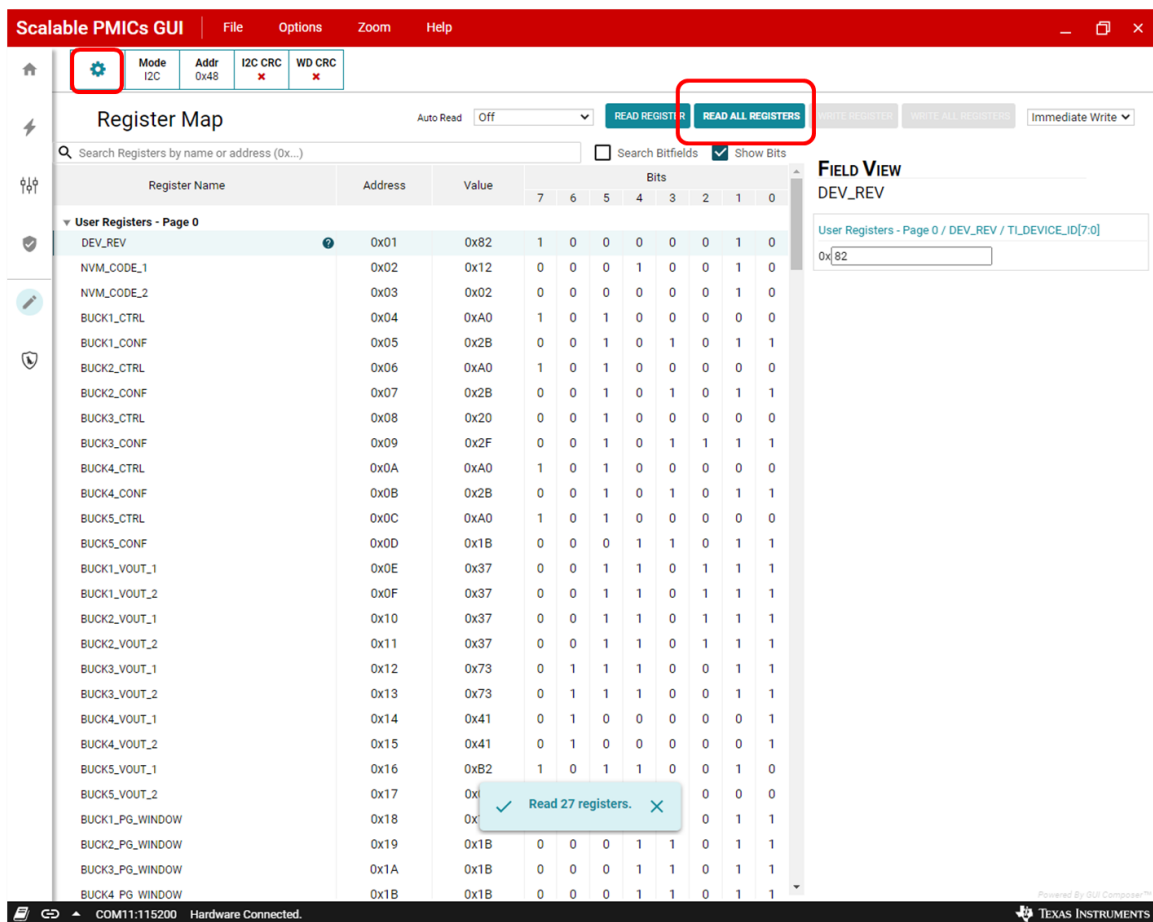


Figure 8. 寄存器读取界面

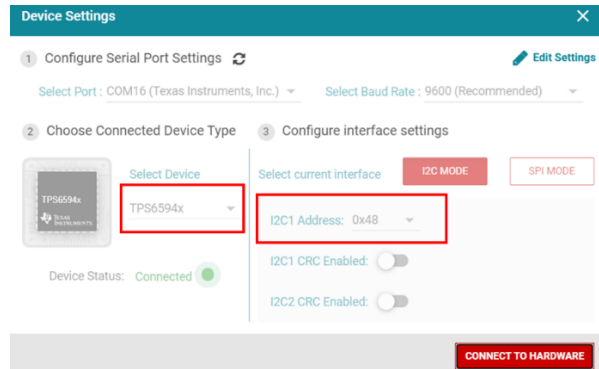


Figure 9. PMIC 切换地址弹窗

GUI 软件的更多使用方法可以参考 Scalable PMIC's GUI User's Guide 文档。本文仅讨论定位问题需要用到的部分功能。

4.1.4 中断信息的读取与保存

中断信息储存在 0x5A 到 0x6C 的寄存器地址范围内，在软件内单击每个对应的 bit 后可以在右边的侧边栏看到该 bit 的详细解释，告警信息的分析参考系列文章的第二篇。为了方便后续进一步分析探讨，把告警信息存下来十分有必要，除了把对应的部分寄存器的软件通过截图的方式保存外，还可以使用脚本的方式保存为纯文本形式。

如图 10 所示，在任意界面点击菜单栏的 option，选择 Scripting 选项，出现一个脚本弹窗，如图 11 所示，这个弹窗的内容仅仅是一个例程且处于写保护模式无法编辑，需要把当前的例程保存为脚本文件后再次打开才可以编辑。点击窗口右上角书本状的图标，在当前窗口的下方会出现 log 窗口，读到的寄存器信息都会显示在 log 窗口中。

点击任务栏图标回到 Scalable PMIC GUI 主窗口（不需要关闭 scripting 窗口），根据软件操作步骤 2 连接上待测 PMIC 的 I2C。复制附录的脚本到 scripting 窗口中，点击绿色三角形图标运行脚本，log 窗口会显示读到的寄存器信息，如图 12 所示，点击保存按钮，数据会保存为一个 CSV 文件。使用脚本还可以读取一些隐藏寄存器供定位问题使用，附录中额外增加了一部分 SPMI、PFSM 和 FSM 相关的内部寄存器，他们在定位中有一定参考价值，这部分寄存器内容解析请咨询 FAE 团队。

需要注意，即使 I2C 未连接，脚本也会正常运行，只是读取到的寄存器值是全 0，需要特别注意避免保存了无用的数据。

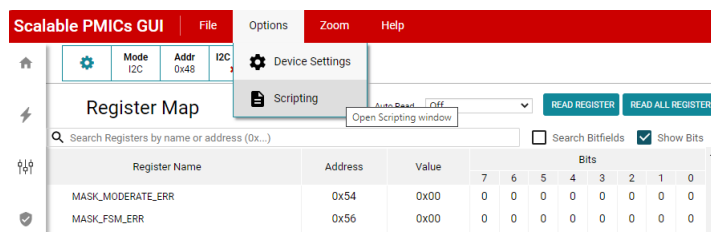


Figure 10. Scripting 菜单

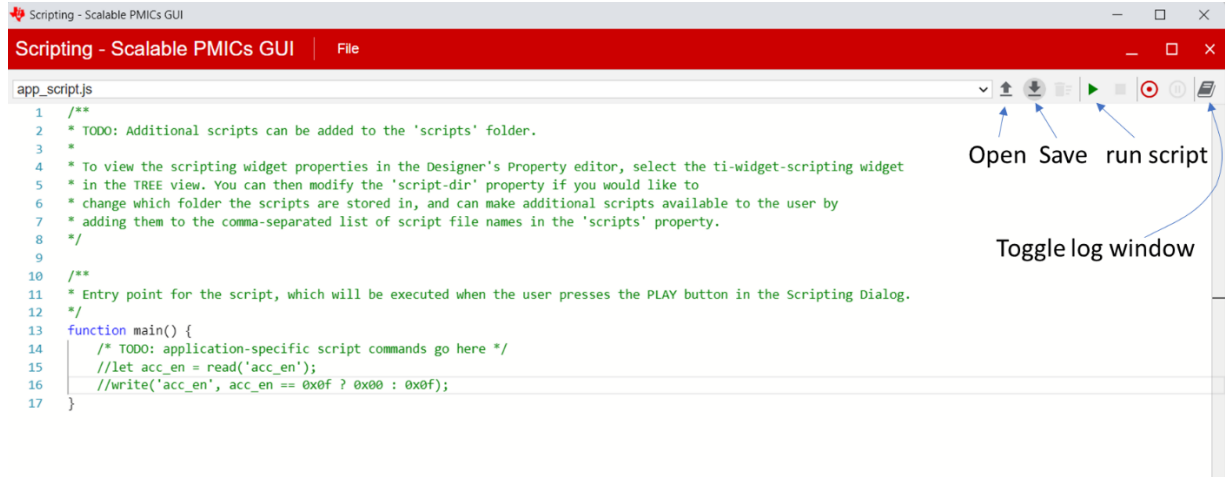


Figure 11. Scripting 窗口 1

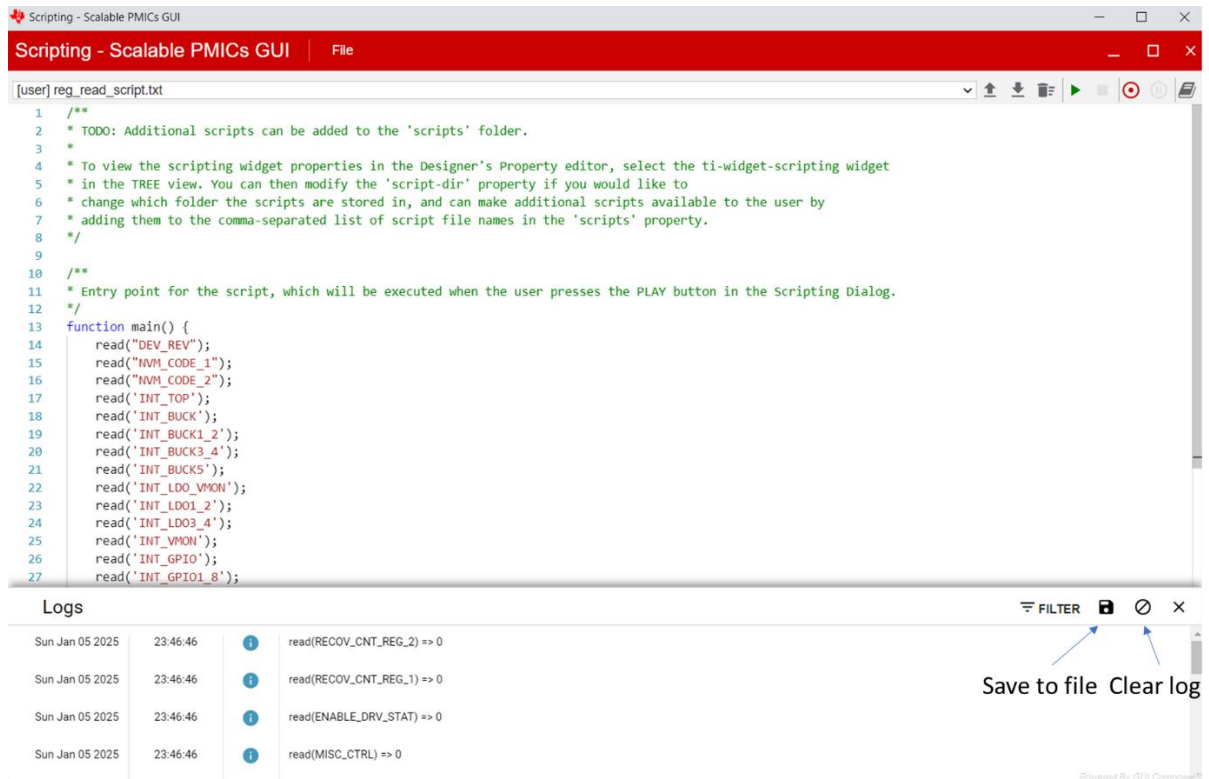


Figure 12. Scripting 窗口 2

5 更改 NVM 定位问题

前面的几个方法都是通过“问诊”的方式去观察问题导致的现象从而定位故障，有时候通过“问诊”可能会看到多个相互耦合的故障作用的最终结果，为了定位根因，可以通过临时更改 NVM 的方式，屏蔽一部分病因来确定最终的根因，这好比是给板子“做手术”，设计好的“手术”方案可以加速问题的定位。

5.1 使用 Scalable PMIC GUI 软件更新 NVM

1. Scalable PMIC GUI 软件提供了更新 NVM 的工具，如图 13 所示，在软件主界面点击右上角的 Begin an NVM Configuration 按钮，进入如图 14 所示的 NVM 编辑界面。在此界面左边选择对应的器件型号，此时右下角 skip to programming 亮起，点击 skip to programming 按钮。

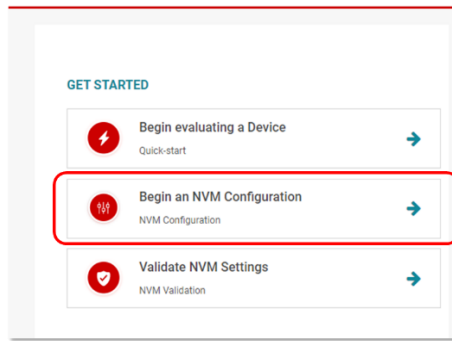


Figure 13. Begin an NVM Configuration 按钮

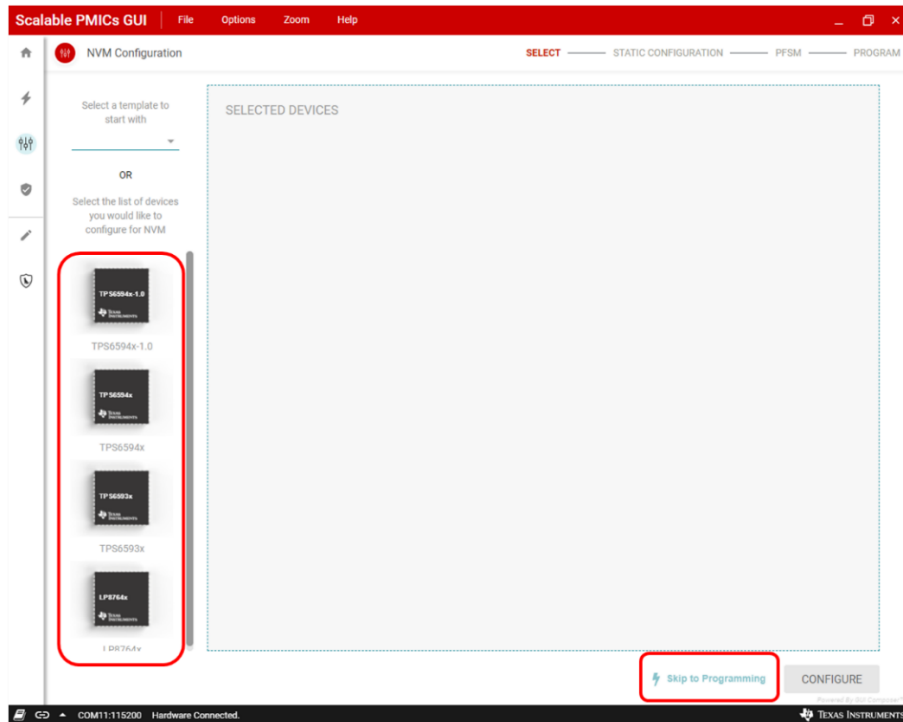


Figure 14. NVM Configuration 界面

2. 之后进入如图 15 所示的 program NVM 界面，点击左侧 Select from your computer 按钮，载入需要烧录的 NVM 固件，这个固件可以是器件原始的 NVM（用以恢复产品原状），或者是基于原始 NVM 修改的 NVM（用以定位问题），或者是自己生成的 NVM（满足特殊用途），自己生成 NVM 的话题将在这个系列的第五篇中讨论。注意右上角的信息确认芯片是否处于连接状态，未连接时的界面如绿色框中所示，需要选对 I2C 地址后点击 connect to hardware 连接 PMIC。
3. 载入 NVM 文件后的界面如图 16 所示，如果需要更换固件可以点击红框所示的图标重选 NVM 文件。最后从上自下点击绿色框的按钮，查看连接状态并开始烧录。若看到图示的那样红字显示 cannot connect to WD address 的提示也是正常的，原因参考硬件部分步骤 5 的说明。
4. 此后会依次跳出三个弹窗，如图 17 所示，前两个务必点击红色的按钮，第三个弹窗根据需求进行选择。至此 PMIC 中的 NVM 已经被更新成所需的固件。需要注意，在 program NVM 界面连续两次 NVM 更新操作（同一个芯片两次更新或换个芯片进行 NVM 更新）时，第二次会出现烧录失败的提示，此时只需要重选文件，点击图 16 中绿色框所示的图标，然后再次 program 一次即可，这是 scalable PMIC GUI 软件的已知 bug。

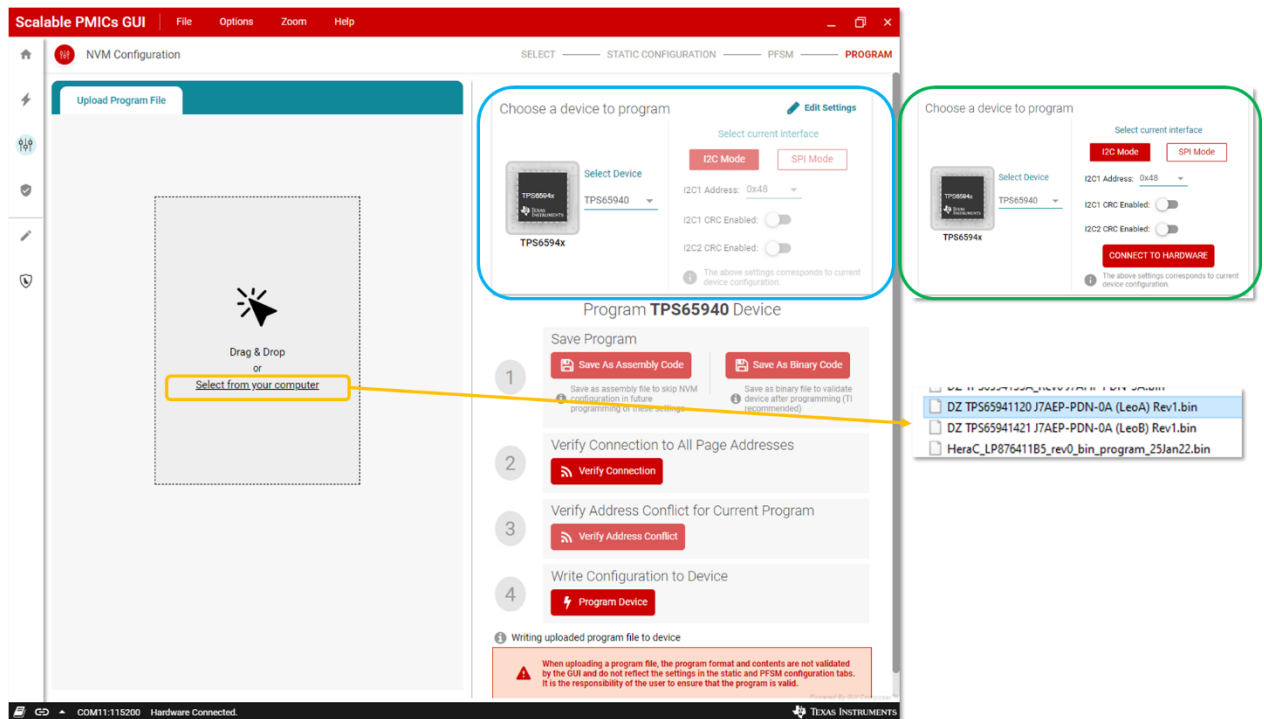


Figure 15. Program NVM 界面 1

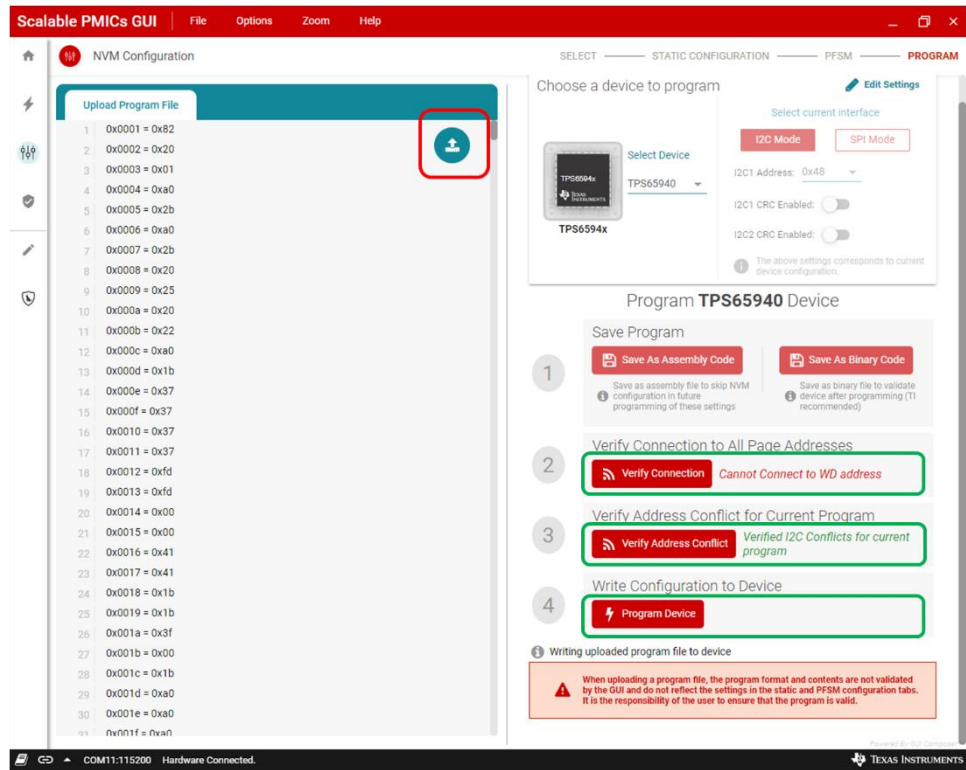


Figure 16. Program NVM 界面 2

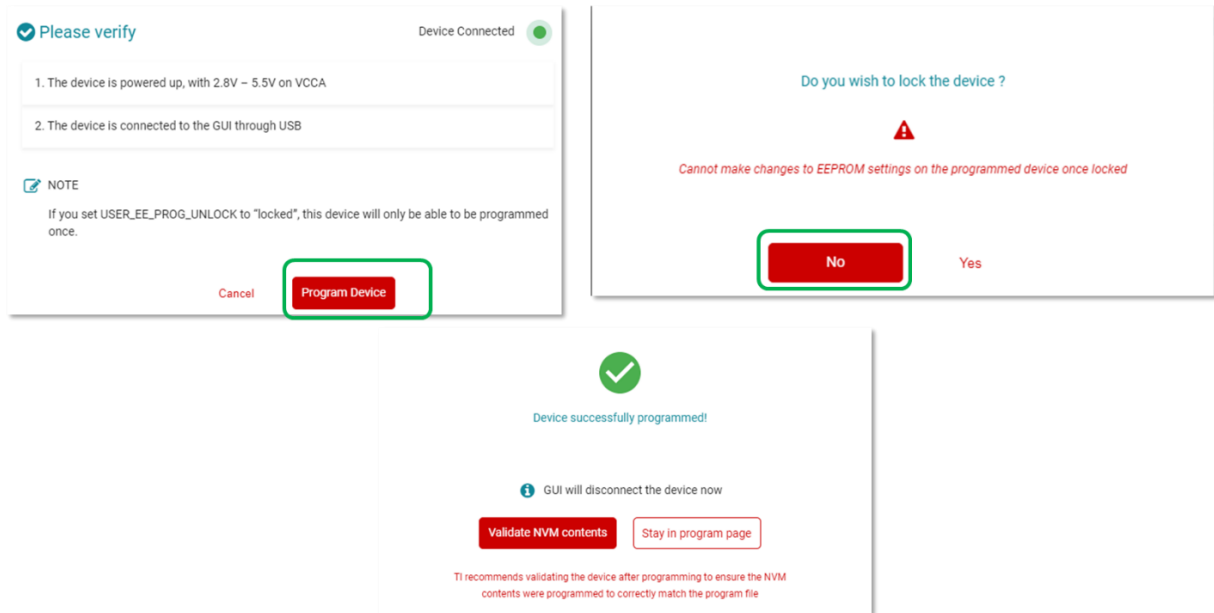
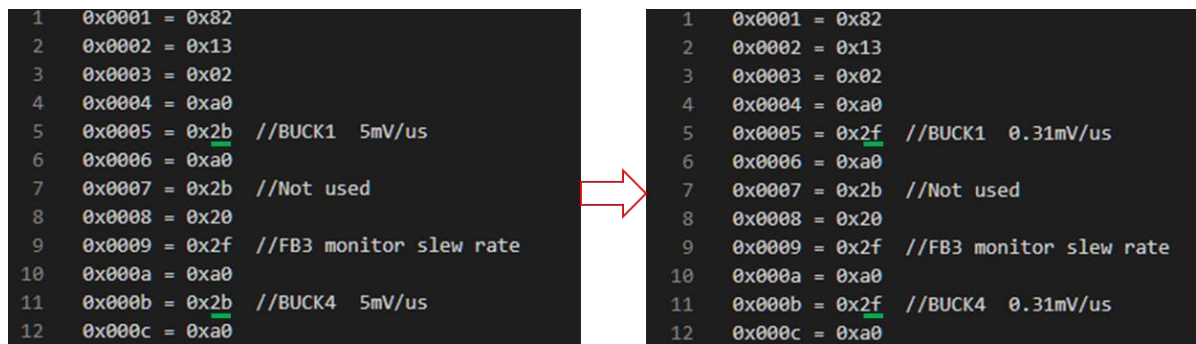


Figure 17. Program NVM 弹窗

5.2 实际应用案例

为了定位个别单板高温下启动后短时间掉电的故障，需要想办法获取寄存器信息。然而故障单板关机过程中因为 VCCA 冲高触发了 VCCA OVP（本系列第三篇介绍的案例，此案例的规避方案是 SOC 在上电后配置 BUCK 的 slew rate 到较低的速率，但是掉电发生在上电后的很短时间，SOC 来不及配置寄存器），寄存器信息因为 NMOS 的关闭而丢失。

通过示波器波形，明确了 VCCA OVP 这个故障是因为 PMIC 掉电行为导致的而不是它导致了掉电本身之后，我们可以通过更新 NVM 固件的方式，将 BUCK 上下电的斜率更改到一个不会触发 VCCA OVP 的水平，如图 18 所示，使用文本编辑器修改原始 NVM bin 文件中对应的寄存器 bit 后另存为新的 bin 文件。使用这个新的 NVM，我们成功读取到的寄存器中断信息，定位到了触发掉电行为的那一路故障电源轨。



Line	Original Value	Original Comment	Modified Value	Modified Comment
1	0x0001 = 0x82		0x0001 = 0x82	
2	0x0002 = 0x13		0x0002 = 0x13	
3	0x0003 = 0x02		0x0003 = 0x02	
4	0x0004 = 0xa0		0x0004 = 0xa0	
5	0x0005 = 0x2b	//BUCK1 5mV/us	0x0005 = 0x2f	//BUCK1 0.31mV/us
6	0x0006 = 0xa0		0x0006 = 0xa0	
7	0x0007 = 0x2b	//Not used	0x0007 = 0x2b	//Not used
8	0x0008 = 0x20		0x0008 = 0x20	
9	0x0009 = 0x2f	//FB3 monitor slew rate	0x0009 = 0x2f	//FB3 monitor slew rate
10	0x000a = 0xa0		0x000a = 0xa0	
11	0x000b = 0x2b	//BUCK4 5mV/us	0x000b = 0x2f	//BUCK4 0.31mV/us
12	0x000c = 0xa0		0x000c = 0xa0	

Figure 18. 修改 NVM bin 文件

6 附录-寄存器读取脚本

```
function main() {
    read("DEV_REV"); //check device type TPS6594 or LP8764
    read("NVM_CODE_1"); //check OPN
    read("NVM_CODE_2"); //check OPN
    read("INT_TOP"); //INT
    read("INT_BUCK");
    read("INT_BUCK1_2");
    read("INT_BUCK3_4");
    read("INT_BUCK5");
    read("INT_LDO_VMON");
    read("INT_LDO1_2");
    read("INT_LDO3_4");
}
```

```
    read('INT_VMON');
    read('INT_GPIO');
    read('INT_GPIO1_8');
    read('INT_STARTUP');
    read('INT_MISC');
    read('INT_MODERATE_ERR');
    read('INT_SEVERE_ERR');
    read('INT_FSM_ERR');
    read('INT_COMM_ERR');
    read('INT_READBACK_ERR');
    read('INT_ESM');
    read('MISC_CTRL');
    read('ENABLE_DRV_STAT');
    read('RECOV_CNT_REG_1'); //RECOVCNT status
    read('RECOV_CNT_REG_2');
    read('TEST_SPMI_1'); //SPMI related internal registers
    read('TEST_SPMI_2');
    read('TEST_SPMI_5');
    read('TEST_SPMI_6');
    read('TEST_SPMI_7');
    read('TEST_PFSM_0'); //PFSM status
    read('TEST_PFSM_1');
    read('TEST_PFSM_2');
    read('TEST_PFSM_3');
    read('TEST_FFSM_1'); //FSM status
    read('TEST_FFSM_2');
}
```

7 参考文献

1. Datasheet “[TPS6594-Q1 Power Management IC \(PMIC\) with 5 BUCKs and 4 LDOs for Safety-Relevant Automotive Applications](#)”
2. Datasheet “[LP8764-Q1 Four-Phase, 20-A Buck Converter With Integrated Switches](#)”
3. Application Note “[Scalable PMIC NVM Update Guide](#)”
4. User's Guide “[Scalable PMIC's GUI User's Guide](#)”
5. User Guide “[TPS65941213-Q1 and LP876411B4-Q1 PMIC User Guide for J721E, PDN-1A](#)”
6. User Guide “[TPS6594x-Q1 Evaluation Module user's guide](#)”

重要通知和免责声明

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

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

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

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

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
版权所有 © 2025，德州仪器 (TI) 公司