



摘要

本文档介绍了德州仪器 (TI) Scalable PMICs 图形用户界面 (GUI) 工具的使用方法和功能。该 GUI 旨在与模拟 EVM 控制器 (AEVM) 一起使用以评估和配置 TPS6594-Q1、TPS6593-Q1、LP8764-Q1、LP8762-Q1 和 LP876242-Q1 系列器件。

内容

1 引言.....	3
2 支持的功能.....	3
3 修订.....	4
4 概述.....	5
5 入门.....	6
6 “Quick-start” (快速启动) 页面.....	13
7 “Register Map” (寄存器映射) 页面.....	22
8 NVM 配置页面.....	23
9 NVM 验证页面.....	55
10 看门狗页面.....	57
11 其他资源.....	62
12 附录 A：疑难解答.....	62
13 附录 B：高级主题.....	64
14 附录 C：已知限制.....	66
15 附录 D：迁移主题.....	67
16 修订历史记录.....	73

插图清单

图 4-1. GUI 概述.....	5
图 5-1. GUI Composer Gallery.....	6
图 5-2. 在“Gallery”中找到 PMIC GUI.....	7
图 5-3. GUI 软件下载选项.....	7
图 5-4. Gallery 中的 GUI 面板.....	8
图 5-5. PMIC GUI 桌面应用.....	8
图 5-6. 器件主页.....	9
图 5-7. 通过“Options” (选项) 选项卡显示的“Device Settings” (器件设置).....	10
图 5-8. 器件接口设置.....	10
图 5-9. SPI 多 PMIC 芯片选择.....	11
图 5-10. 器件设置栏.....	12
图 6-1. 快速启动扫描页面.....	13
图 6-2. 快速启动扫描页面结果.....	14
图 6-3. 快速启动页面重点内容.....	15
图 6-4. 系统信息.....	16
图 6-5. BUCK 配置.....	17
图 6-6. LDO 配置.....	18
图 6-7. GPIO 配置.....	19
图 6-8. 中断屏蔽和状态.....	20
图 6-9. 其他设置.....	21
图 7-1. 寄存器映射.....	22
图 8-1. 打开现有配置.....	23

图 8-2. 从空白模板开始.....	24
图 8-3. 配置开发流程.....	25
图 8-4. BUCK 静态配置.....	26
图 8-5. 静态配置之验证失败.....	26
图 8-6. 静态配置之验证通过.....	27
图 8-7. 功能安全假设检查.....	27
图 8-8. 未更新的静态配置值示例.....	28
图 8-9. 经常保存.....	28
图 8-10. PFSM 启动模板.....	29
图 8-11. PFSM 配置.....	29
图 8-12. PFSM 状态图面板.....	31
图 8-13. 完整的状态图.....	31
图 8-14. 全局设置.....	32
图 8-15. 目标状态设置为“ANY”.....	33
图 8-16. 自终止转换和触发关联.....	34
图 8-17. 如何添加序列.....	35
图 8-18. 电源序列命令窗口.....	35
图 8-19. WAIT 命令操作.....	40
图 8-20. 空子序列.....	41
图 8-21. 电源序列工具.....	42
图 8-22. 具有可变条件的导出序列.....	42
图 8-23. 触发条件：STANDBY (待机) 转 SAFE (安全)	43
图 8-24. 触发条件：STANDBY 转 SAFE (续)	44
图 8-25. 触发条件：SAFE (安全) 转 SAFE_RECOVERY (安全_恢复)	44
图 8-26. 将序列映射到触发条件.....	45
图 8-27. 已完成的触发条件设置.....	46
图 8-28. 触发条件优先级列表.....	47
图 8-29. PFSM 验证结果.....	48
图 8-30. NVM 编程.....	49
图 8-31. 跳至使用现有 NVM 配置进行编程.....	51
图 8-32. 使用现有配置进行编程.....	52
图 8-33. 编程过程中的接口更改.....	53
图 8-34. NVM 锁定选项.....	54
图 9-1. NVM 验证.....	55
图 10-1. 看门狗页面.....	57
图 10-2. 看门狗触发条件模式.....	58
图 10-3. 无效的触发条件看门狗输入.....	59
图 10-4. 看门狗 Q&A 模式.....	60
图 10-5. 有效的看门狗 Q&A 响应.....	60
图 10-6. 无效的看门狗 Q&A 响应.....	61
图 12-1. 硬件平台错误.....	62
图 12-2. PMIC 通信错误.....	63
图 13-1. 默认脚本编写窗口.....	64
图 13-2. 脚本编写，录制寄存器读取和写入.....	64
图 13-3. 脚本编写，运行录制的序列.....	65
图 15-1. 尝试将 LP8764-Q1 PG1.0 导入 GUI 版本 3.0.0 时出现错误消息.....	67
图 15-2. 需要标题 ***.....	68
图 15-3. 添加 PFSM_START 状态.....	69
图 15-4. 具有延迟时间的 PFSM 汇编程序验证错误.....	69
图 15-5. 迁移的延迟时间修复.....	70
图 15-6. 具有更新时间延迟的 PFSM 验证结果.....	70
图 15-7. 触发条件 SAFE_RECOVERY 的警告.....	71
图 15-8. 优先级列表更新.....	71
图 15-9. 立即触发功能.....	72
图 15-10. 解决触发警告后的 PFSM 验证结果.....	72

表格清单

表 3-1. GUI 版本.....	4
--------------------	---

表 5-1. AEMV 上的芯片选择表示.....	11
表 8-1. 用于编辑状态机的操作.....	30
表 8-2. 请求的延迟时间和实际的延迟时间示例.....	32
表 8-3. SPMI_LPM_EN、FORCE_EN_DRV_LOW 和 LPM_EN 示例汇编指令.....	38
表 14-1. GUI 限制及相关版本.....	66
表 14-2. GUI 限制说明.....	66

商标

SimpleLink™ and Launchpad™ are trademarks of Texas Instruments.

Chrome™ is a trademark of Google.

Firefox™ is a trademark of The Mozilla Foundation.

Microsoft Windows® is a registered trademark of Microsoft Corporation.

Linux® is a registered trademark of Linus Torvalds.

所有商标均为其各自所有者的财产。

1 引言

该工具基于 GUI Composer，需要使用 MSP432E401Y SimpleLink™ 以太网微控制器作为模拟 EVM 控制器 (AEMV)，该控制器集成在 PMIC EVM 中，也可以随 Launchpad™ 一同提供，请参阅节 11 中的 1。AEMV 提供用于连接主机个人计算机 (PC) 的 USB 接口以接收命令，然后使用 I²C 或 SPI 协议与 PMIC 进行通信。

GUI 通过单个可执行文件 (和单个 AEMV) 支持多个器件，从而在与多个器件一同工作时无需安装多个 GUI。多个器件采用主要/辅助配置，AEMV 通过选定的共享媒体与每个器件进行通信：I²C 或 SPI。

2 支持的功能

GUI 支持以下功能：

- 使用单个 GUI 连接多个 PMIC 器件
- “Quick-start” (快速启动) 和 “Register Map” (寄存器映射) 视图，用于在上电后对 PMIC 寄存器进行读取和写入
- 用于指示中断和 GPIO 状态的状态指示器
- 非易失性存储器 (NVM) 的编程和验证
- 看门狗配置和评估
- 多平台¹：Microsoft Windows®、Linux® (32 位和 64 位) 和 Mac OSX
- 提供基于网络的版本和独立版本
- 指向其他配套资料、支持论坛和常见问题解答的链接

¹ 请参阅 dev.ti.com 以了解与 GUI Composer 的版本兼容性。

3 修订

本节详细介绍了每个版本的 Scalable PMICs GUI 中添加的功能。节 14 还列出了每个版本的已知问题。

版本 1.0.0 是最初的预量产版本，其名称为 Programmable Processor PMIC's GUI。该版本的工具可用于评估和编程，但不提供用于创建 NVM 配置的机制。

版本 2.0.0 构建于版本 1.0.0 的特性之上，特别新增了 NVM 配置、NVM 验证和看门狗评估页面。

版本 3.0.0 是最新的版本，其名称为 Scalable PMICs GUI。本文档介绍了该最新版本。

表 3-1. GUI 版本

修订版本	发布日期	支持的器件	功能更新和添加
1.0.0	2019 年 12 月	<ul style="list-style-type: none"> TPS6594-Q1 版本 1.0 器件 	初始预量产版本 <ul style="list-style-type: none"> “Quick-start”（快速启动）页面 “Register Map”（寄存器映射）页面 NVM 编程
2.0.0	2020 年 11 月	<ul style="list-style-type: none"> TPS6594-Q1 LP8764-Q1 修订版 1.0 器件 TPS6593-Q1 TPS6594-Q1 版本 1.0 器件 	<ul style="list-style-type: none"> NVM 创建和验证 NVM 模板 脚本编写 LP8764-Q1 器件支持 提高了编程速度 “Watchdog Evaluation”（看门狗评估）页面 支持 SPI 和 CRC 的串行通信 自动定时调整以满足所需的延迟时间。
3.0.0	2022 年 6 月	<ul style="list-style-type: none"> TPS6594-Q1 LP8764-Q1 修订版 2.0 器件 TPS6593-Q1 LP8762-Q1 LP876242-Q1 	<ul style="list-style-type: none"> 更新了” NVM 创建，以使 NVM 设计更灵活。 更多反馈机制来识别潜在的 NVM 问题。 多 SPI 通信控制 SPI 总线上所有 PMIC 的芯片选择。 改进了页面之间的转换。 删除了自动定时调整以满足所需的延迟时间。 新增了验证错误和手动控制延迟时间的指示。

4 概述

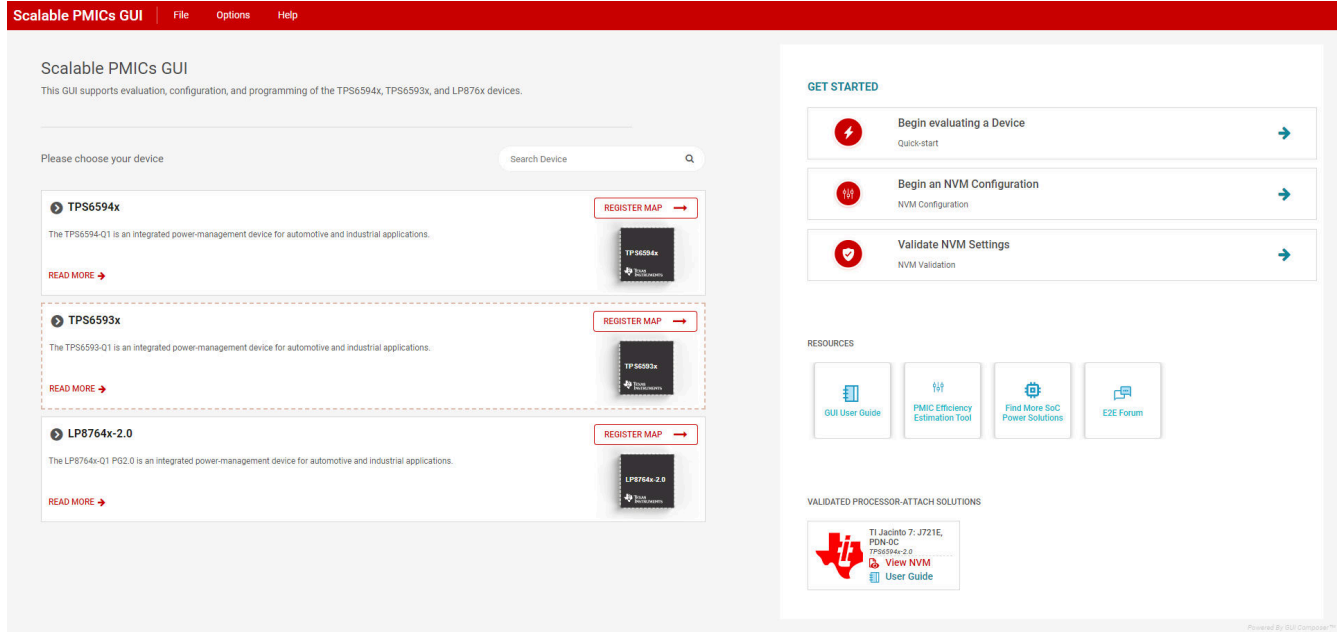


图 4-1. GUI 概述

GUI 提供了用于评估和配置 PMIC 的页面。对于评估，“Quick-start”（快速启动）、“Register Map”（寄存器映射）和“Watchdog”（看门狗）页面提供了用于通过 SPI 或 I²C 接口监视和控制 PMIC 的界面。可以通过“NVM Configuration”（NVM 配置）页面完成 PMIC NVM 配置，并且可以通过“NVM Validation”（NVM 验证）页面读取 PMIC 中的 NVM 内容。后续章节将提供有关“Quick-start”（快速启动）、“Register Map”（寄存器映射）、“NVM Configuration”（NVM 配置）、“NVM Validation”（NVM 验证）和“Watchdog”（看门狗）页面的更多信息。

此 GUI 还提供用户指南和数据表的模版和参考，以帮助完成开发过程。建议在适用时使用现有模板作为开发的起点。这些模板不仅提供 PMIC 操作的图形表示，而且还生成对 PMIC 进行编程所需的 NVM 文件。

器件选择

器件选择页面是 GUI 启动时显示的第一个页面。在该页面中，可以通过在给定器件窗口中选择“Register Map”（寄存器映射）来选择特定器件。也可以选择其他页面（与特定器件无关）。在这些页面中，需要使用器件接口设置窗口来选择器件和接口。

寄存器映射

“Register Map”（寄存器映射）页面是可用用户寄存器的列表视图，可以对这些寄存器进行读取和写入。

快速启动

建议将“Quick-start”（快速启动）页面用作评估器件的起点。“Quick-start”（快速启动）页面提供了与 PMIC 进行交互和配置器件寄存器的图形视图。除了该抽象视图之外，还通过“Register Map”（寄存器映射）页面提供了实际寄存器视图。

NVM 配置

“NVM Configuration”（NVM 配置）页面可用于配置非易失性存储器 (NVM) 设置以及对其进行编程。配置包括静态设置和预配置状态机 (PFSM) 设置。

NVM 验证

“[NVM Validation](#)”（[NVM 验证](#)）页面可读出 NVM 内容，可用于验证 NVM 的内容是否与通过“[NVM Configuration](#)”（[NVM 配置](#)）页面编程的内容相匹配。

看门狗评估

“[WatchDog](#)”（[看门狗](#)）页面是一个独特的页面，其中提供了一个图形界面，用于在选定的 PMIC 上配置和执行看门狗功能。

5 入门

入门涉及以下步骤：

1. 在“[Gallery](#)”中找到该 GUI。
2. 下载所需软件。
 - a. 通过网络浏览器运行该 GUI 的 GUI Composer Runtime
 - b. 该 GUI 的离线版本
3. 启动 GUI。

5.1 查找 GUI

PMIC GUI 基于与 Chrome™（46 以上版本）或 Firefox™（38 以上版本）兼容的 GUI Composer。建议使用 Chrome™ 网络浏览器，而本文档通篇使用该浏览器进行演示。可以通过 [TI DevTools](#) 页面中的 TI 开发工具找到该 GUI。从“[Explore](#)”选项卡导航到“[Gallery](#)”（在图 5-1 中以红色突出显示）是进入“[Gallery](#)”的一种方式。

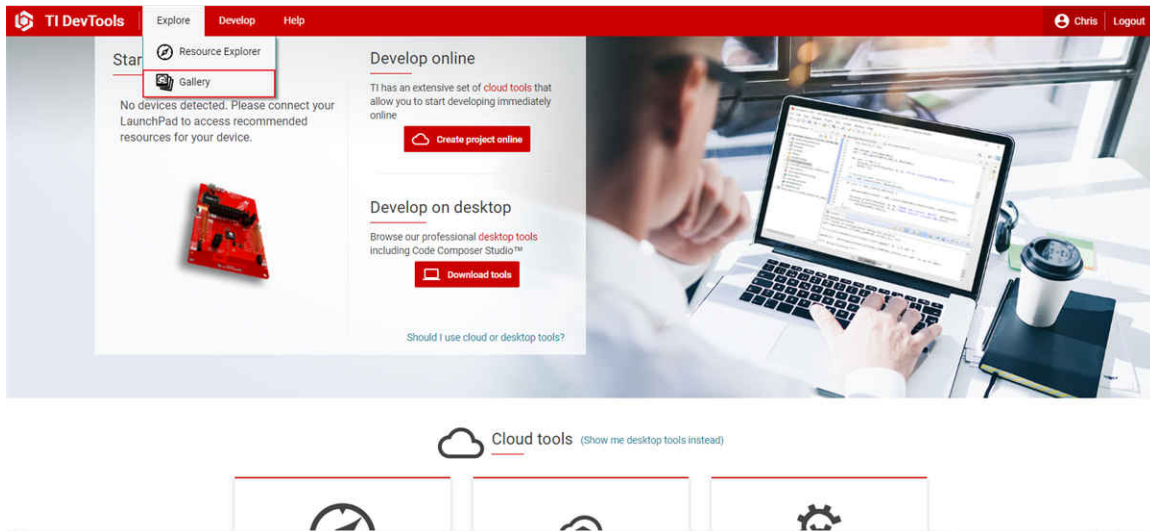


图 5-1. GUI Composer Gallery

进入 [Gallery](#) 后，找到图 5-2 中所示的“[Scalable PMICs GUI](#)”面板。如果主页中未显示此面板，则使用搜索栏，并在其中输入搜索词 *PMIC*。

备注

该 GUI 的名称已更改，因此针对 [Programmable-Processor-PMICs-GUI](#) 和 [Scalable-PMICs-GUI](#) 都提供了相应的面板。[Scalable-PMICs-GUI](#) 版本 3.0.0 是最新版本。

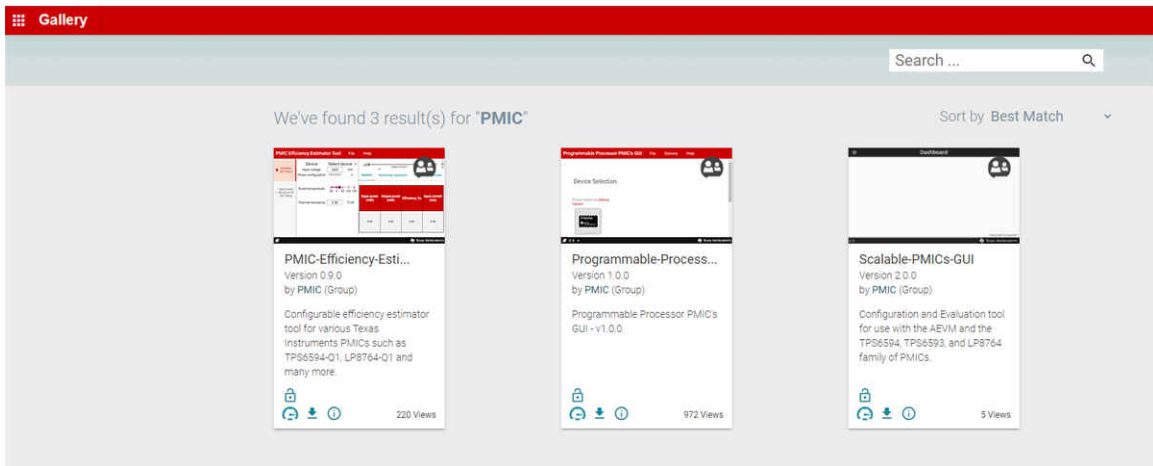


图 5-2. 在“Gallery”中找到 PMIC GUI

5.2 下载所需的软件

独立的 GUI 和 GUI Composer Runtime 都可通过“PMIC”面板获取。同样，GUI Composer Runtime 使 GUI 能够通过网络浏览器运行。这是一个很小的软件，但仍需要连接互联网才能运行 GUI。相比之下，独立 GUI 的容量要大得多，但不需要连接互联网。

如图 5-3 所示，在将光标放置在下载图标上时，弹出窗口中会显示下载选项。上部的三个选项适用于相应操作系统的独立下载，而下部的三个选项适用于 GUI Composer Runtime。

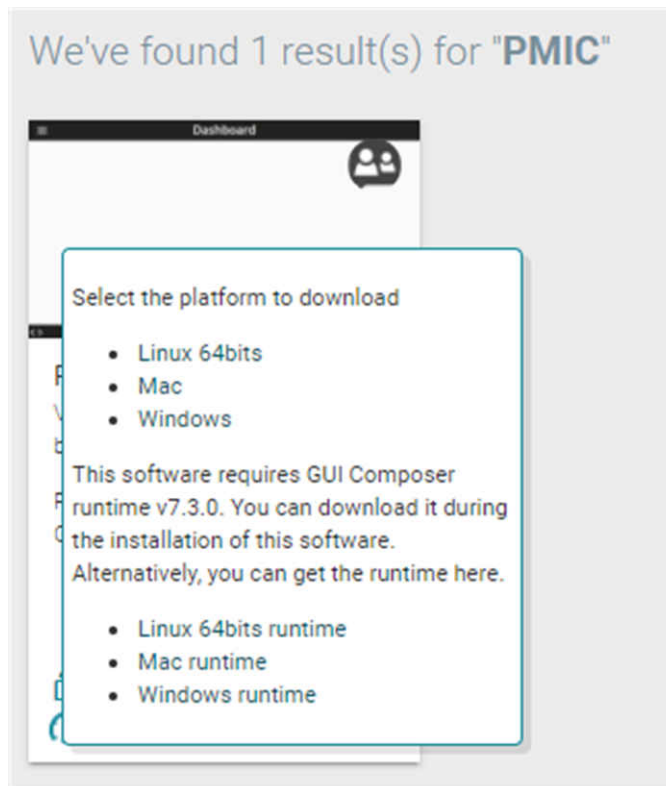


图 5-3. GUI 软件下载选项

5.3 启动 GUI

下载相应的软件后，可以通过 PC 应用在本地上启动 GUI，也可以使用 Gallery 通过 TI 云启动。若要使用 TI 云版本的 GUI，只需点击面板中与下载或信息图标无关的任意位置，如图 5-4 所示。

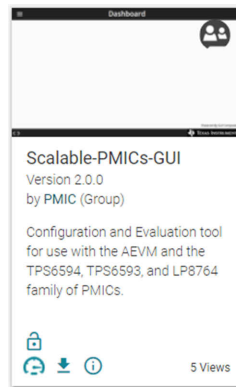


图 5-4. Gallery 中的 GUI 面板。

图 5-5 显示了 PC 应用的一个示例。

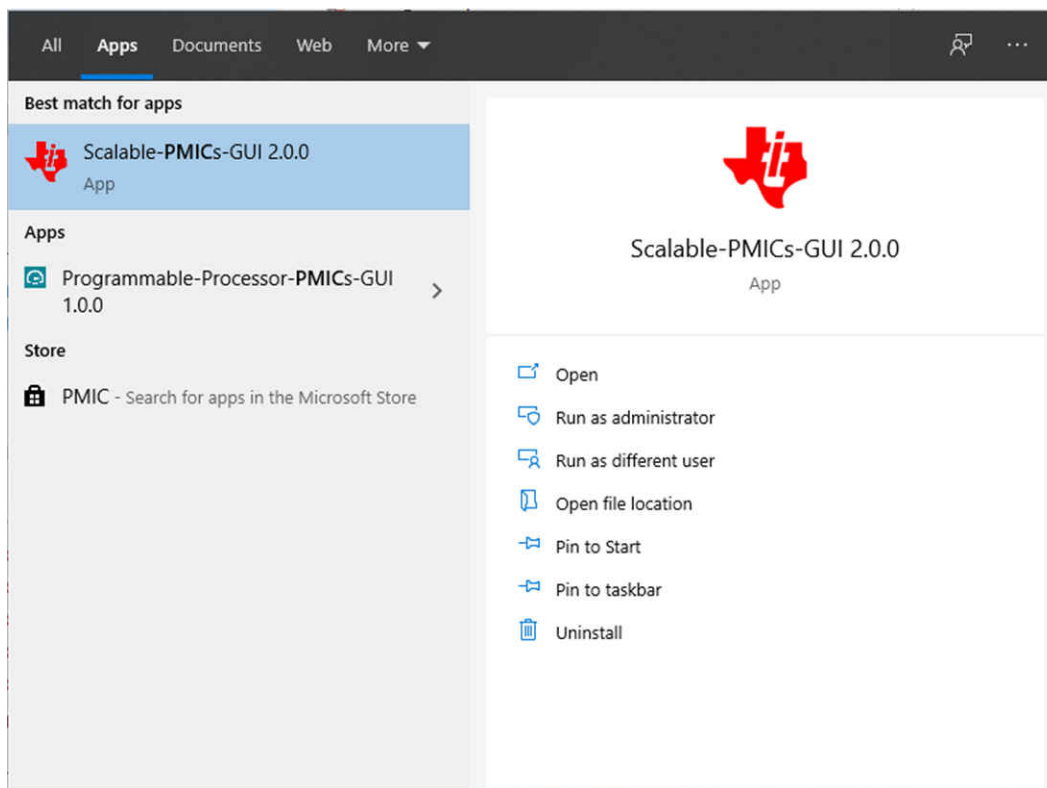


图 5-5. PMIC GUI 桌面应用

启动 GUI 会自动加载器件主页，如图 5-6 所示。

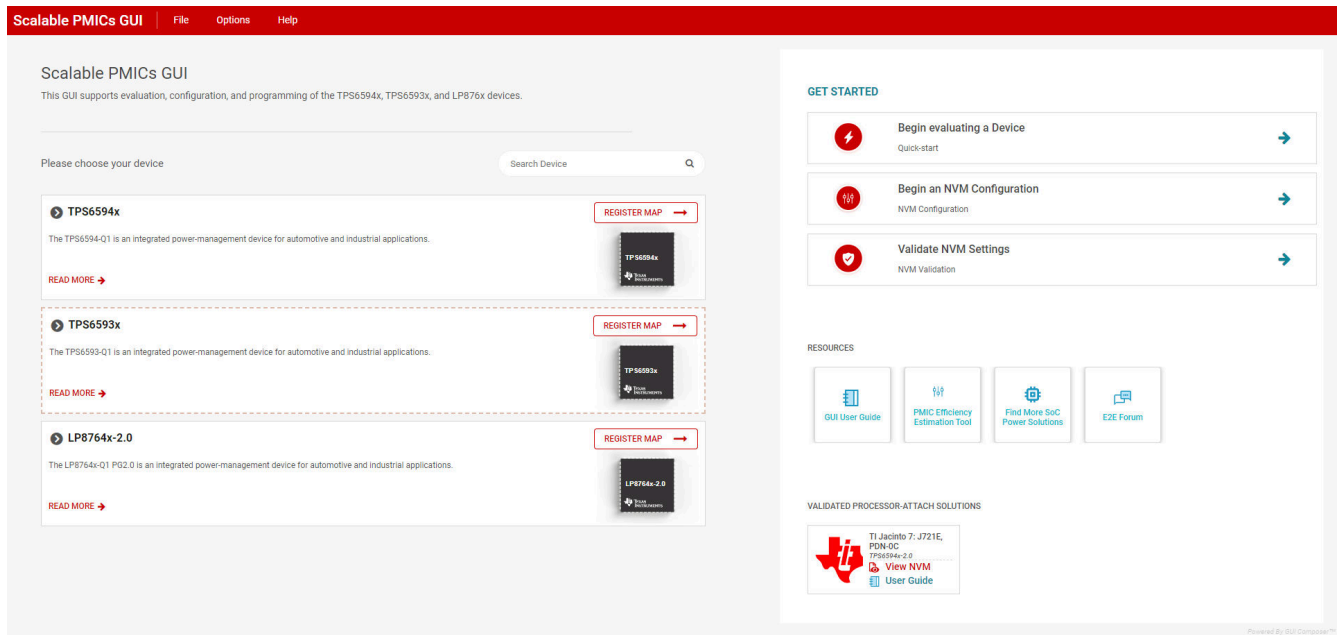


图 5-6. 器件主页

备注

除非进入“Register Map”（寄存器映射）或“Quick-start”（快速启动）页面，否则 GUI 不会尝试连接到 AEVM 控制器。GUI 将尝试自动检测并连接到正确的串行端口。建立此连接后，GUI 将尝试连接到默认的 I²C 地址 0x48。如果地址未被确认，GUI 将显示用于更新接口和地址的“Device Settings”（器件设置）窗口。如果 PMIC 使用 SPI 接口，则连接到 I²C 的初始尝试可能会影响 AEVM，并且需要从 GUI 中选择 SPI 接口重新启动 AEVM。有关解决连接问题的其他提示，请参阅节 12。

5.4 连接到 PMIC

GUI 和 AEVM 支持具有和不具有 CRC 功能的 I²C 和 SPI。如前所述，在进入“Register Map”（寄存器映射）和“Quick-start”（快速启动）页面时，GUI 将尝试连接到地址 0x48 处任何具有非 CRC I²C 接口的器件。如果需要备用地址或配置，则可以通过 *Device Settings*（器件设置）窗口来更改设置，如图 5-7 所示。也可以在 *Options*（选项）下方的下拉菜单中访问该窗口。

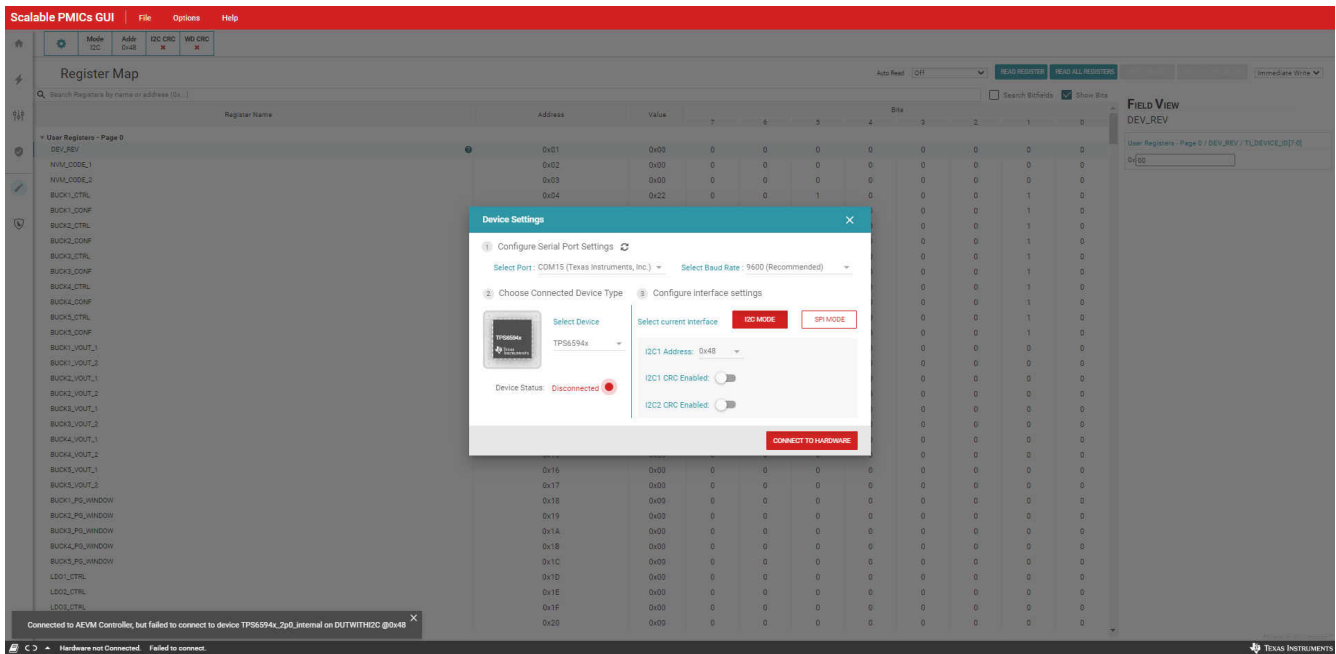


图 5-7. 通过“Options”（选项）选项卡显示的“Device Settings”（器件设置）。

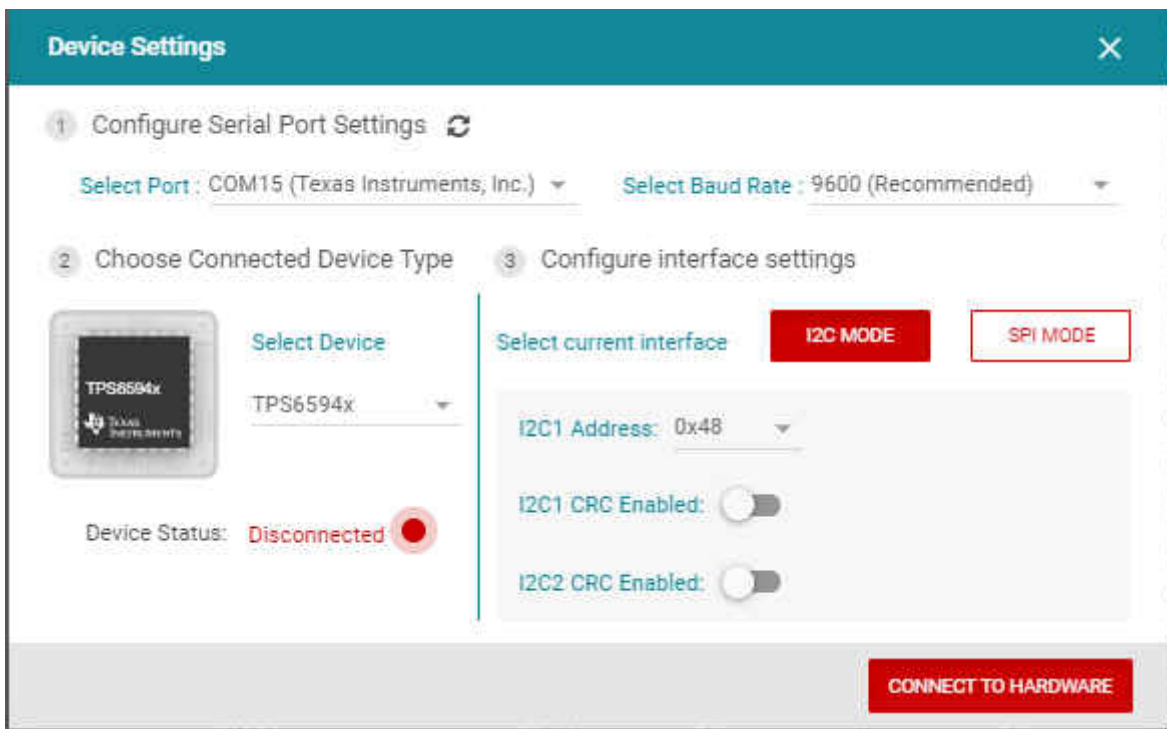


图 5-8. 器件接口设置

如图 5-8 所示，提供了用于选择器件和接口的选项。

选择器件

器件选择将决定寄存器如何解释写入 PMIC 和从 PMIC 中读取的数据。如果未能选择正确的器件，则可能导致错误的的数据被写入 PMIC 或从 PMIC 读取的数据被误解。

I²C

I²C1 地址选择仅限于 PMIC 的有效页面 0 地址。指定 I²C1 地址后，GUI 将自动确定页面 1-4 的地址，并确定是否要使用页面 4 的物理 I²C2 接口。

SPI

与 I²C 实现类似，GUI 会在通信期间自动更新页面信息。

最新的 AEVM 控制器固件将通过提供单个芯片选择 (CS) 控件来支持多 PMIC 操作。如图 5-9 中所示，在主要/辅助配置中最多可级联 6 个 PMIC，并且与每个器件的通信由芯片选择控制。表 5-1 显示了与 AEVM GPIO 的关系。

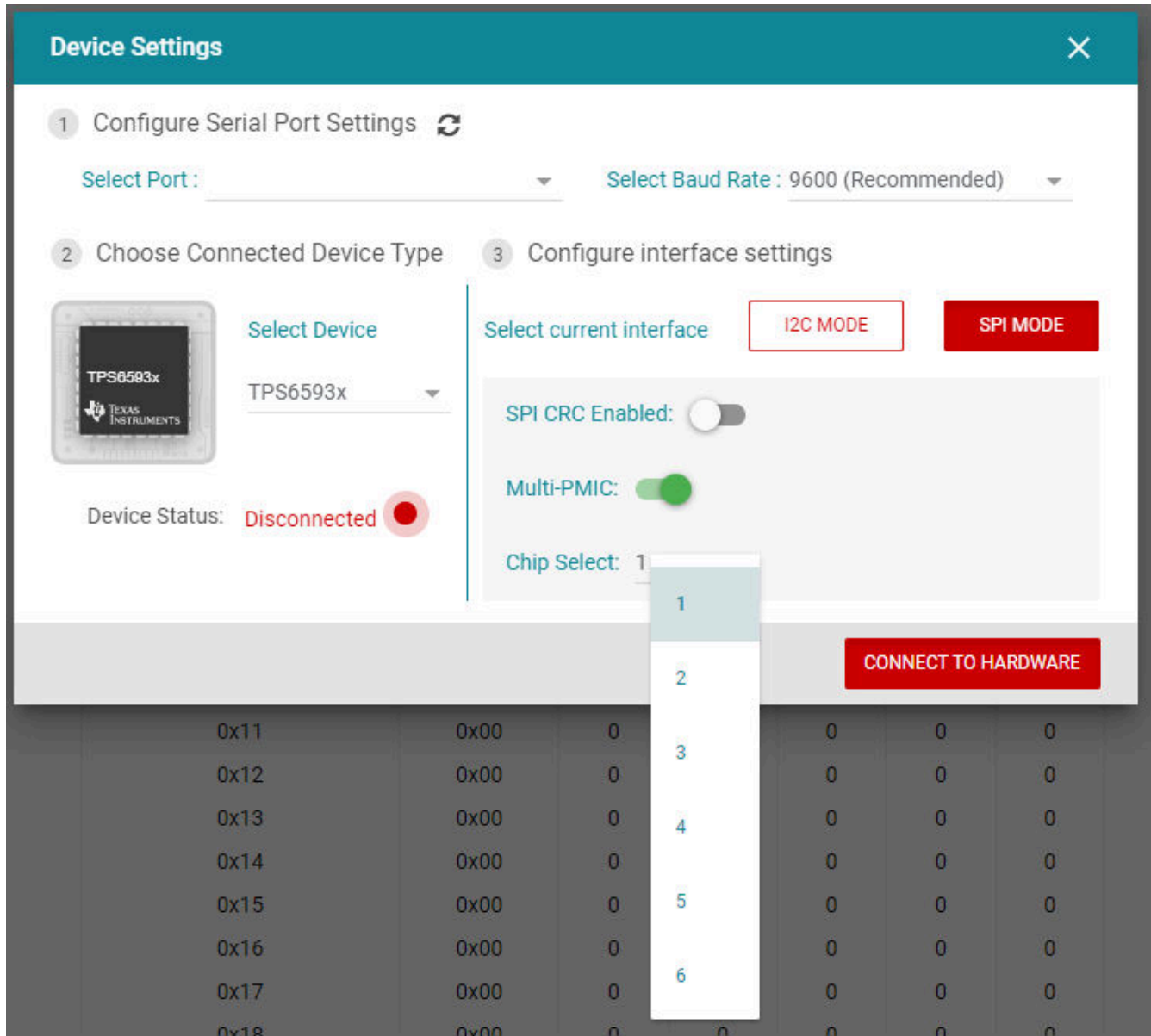


图 5-9. SPI 多 PMIC 芯片选择

表 5-1. AEVM 上的芯片选择表示

片选	AEVM 端口
未选择多 PMIC	PD2，使用 QSSI 模块 CS
1	PD2，使用 QSSI 模块 CS

表 5-1. AEVM 上的芯片选择表示 (continued)

片选	AEVM 端口
2	PD7
3	PC7
4	PC6
5	PC5
6	PC4

更新接口

除了 GUI 顶部的 *Option* (选项) 选项卡之外，在“Quick-start” (快速启动)、 “Register Map” (寄存器映射) 和 “Watchdog” (看门狗) 页面中，还有一个显示当前接口选择的器件设置栏。点击该栏中的齿轮图标也会打开 “Device Settings” (器件设置) 窗口。在 “NVM Configuration” (NVM 配置) 和 “NVM Validation” (NVM 验证) 页面中提供了接口选择。更多详细信息，请参阅这些部分。



图 5-10. 器件设置栏

备注

无需连接器件即可访问这些 GUI 页面。具体而言，无需连接器件即可创建 NVM 配置，如节 8 所述。

6 “Quick-start” (快速启动) 页面

在图 6-1 的初始快速启动页面中，提供了 *Detect Devices* (检测器件) 框，以自动检测连接到微控制器的 I²C 或 SPI 的所有 PMIC 器件² 总线。GUI 将尝试通过 I²C 连接到与器件关联的默认地址。位于 GUI 左下角的 *Hardware Connected* (硬件已连) 表示在该地址找到了一个器件。如果在默认地址找不到器件，则必须为连接到 GUI 的至少一个器件提供器件地址 (在 SPI 情境中则为配置) 才能使用 *DETECT DEVICES* (检测器件) 按钮。可以通过从页面顶部的 *Options* (选项) 下拉菜单中选择 *Device Settings* (器件设置) 来更改接口连接。

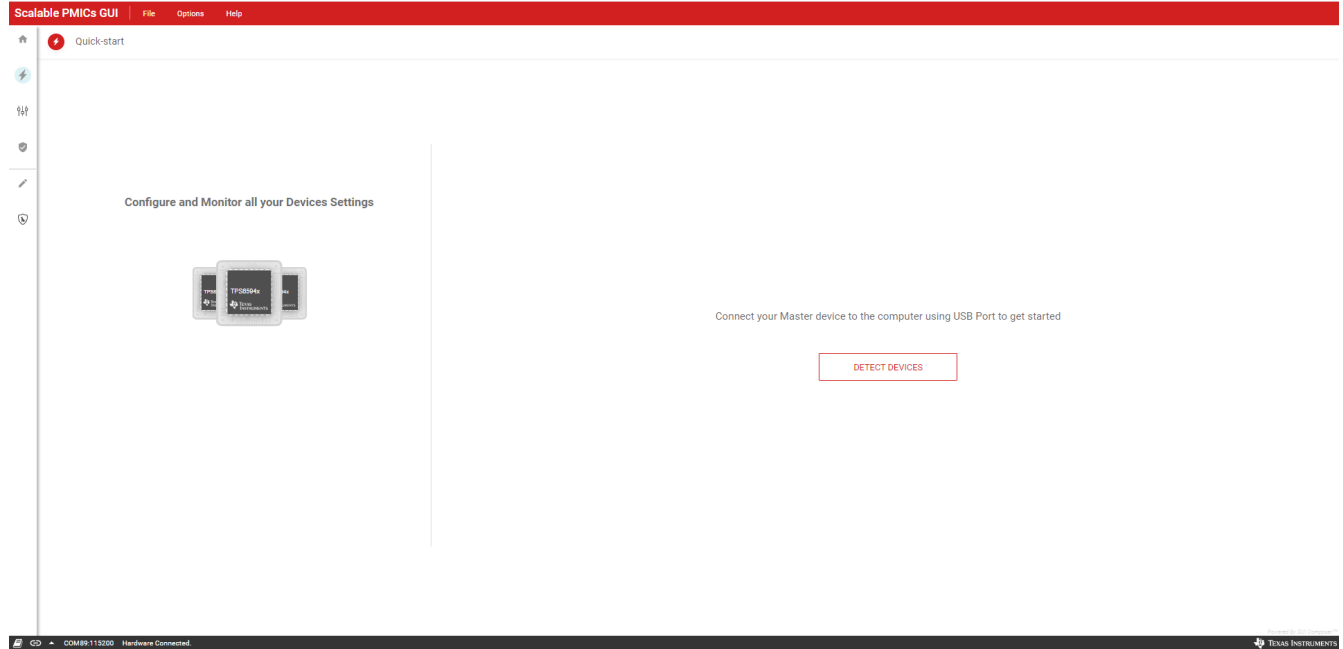


图 6-1. 快速启动扫描页面

² 快速启动页面不支持，也不显示同一 SPI 总线上的多个器件。必须单独选择每个器件。此操作可以手动完成，也可以使用器件设置中的芯片选择功能完成。

6.1 器件扫描结果

图 6-2 显示在本例中检测到两个 PMIC。一个 PMIC 配置为主器件，以带蓝色圆圈的 *M* 表示，另一个则配置为辅助器件。此外，还显示了有关每个器件的 BUCK 相位配置和 I²C ID 的更多信息。器件标签是一个可编辑字段，可以对其进行更新以提供自定义命名约定。通过点击 *Proceed* (继续) 按钮，“Quick-start” (快速启动) 页面将进入可以对所选器件寄存器进行写入或读取的界面窗口。

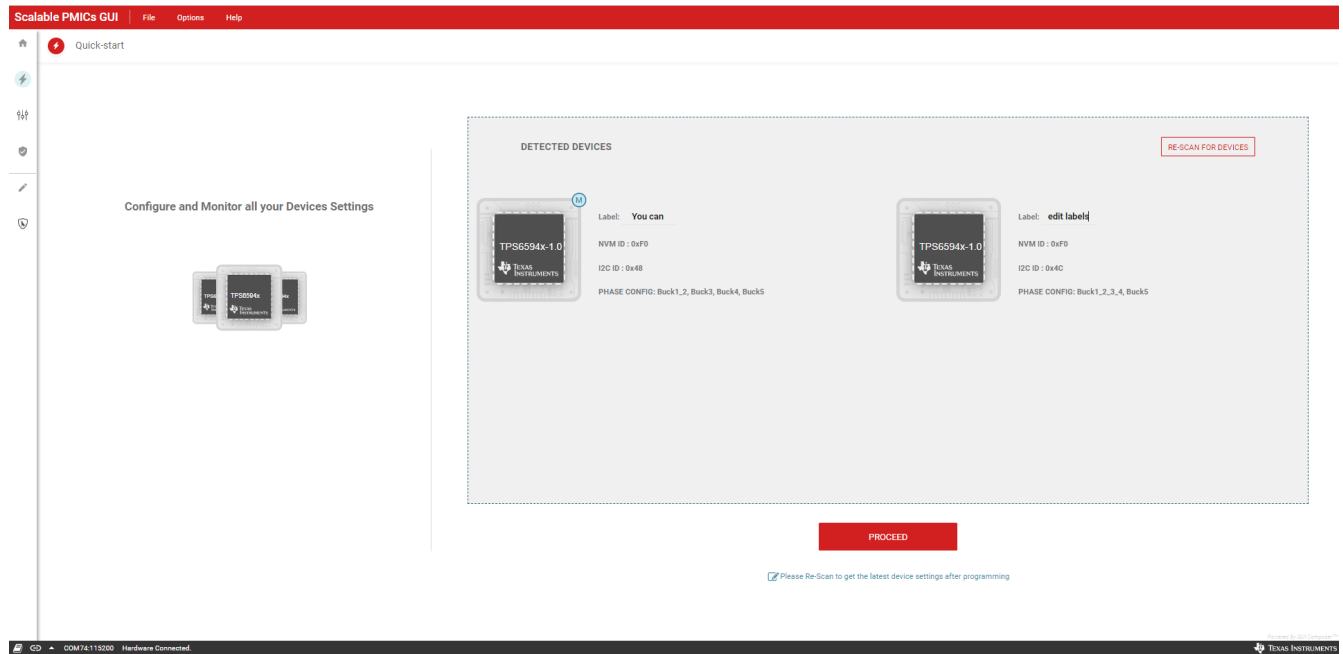


图 6-2. 快速启动扫描页面结果

6.2 配置和监视

在“Quick-start” (快速启动) 页面中，有六个水平对齐的选项卡用于编辑，四个垂直对齐的选项卡用于监视和访问高级特性。图 6-3 突出显示了这些选项卡。以下各节将介绍“Quick-start” (快速启动) 页面中的这些选项卡。务必注意，GUI 会不断轮询 PMIC 以更新“Interrupts” (中断)、“Resource Status” (资源状态) 和“GPIO Pin Status” (GPIO 引脚状态) 选项卡中显示的值。此外，每次对值进行更新或更改时都会引发与 PMIC 的通信，以更新相应的寄存器。可以通过下电上电来重置器件，以将寄存器设置恢复为 NVM 值。

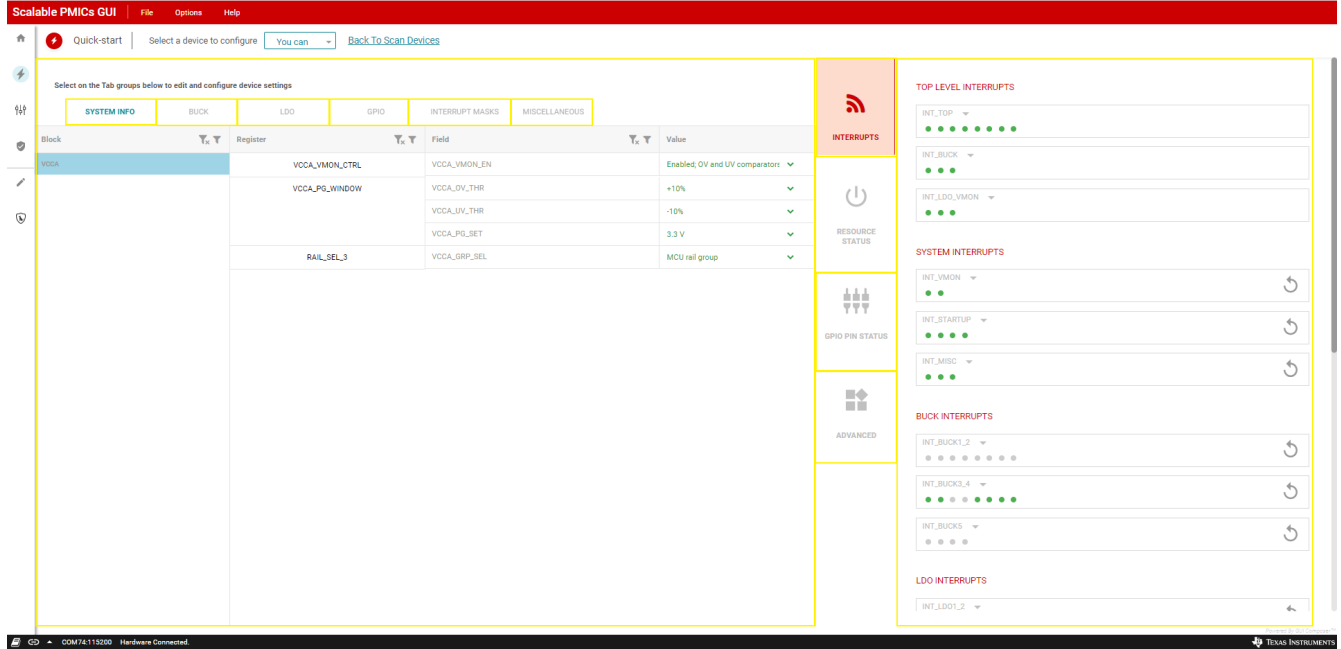


图 6-3. 快速启动页面重点内容

备注

“Quick-start” (快速启动) 页面顶部有一个下拉菜单，用于在连接了多个 PMIC 时选择器件。在图 6-2 中定义了该标签。

备注

所有寄存器字段都是对器件规格的直接引用。当字段中的值重复或者被保留时，GUI 将不会在下拉菜单选项中显示这些可能的值。

6.2.1 系统信息

如图 6-4 所示，“System Info”（系统信息）选项卡与 VCCA 输入电压相关，如果适用，还与附加电压监视器相关。提供了下拉菜单以显示每个字段的可能值。

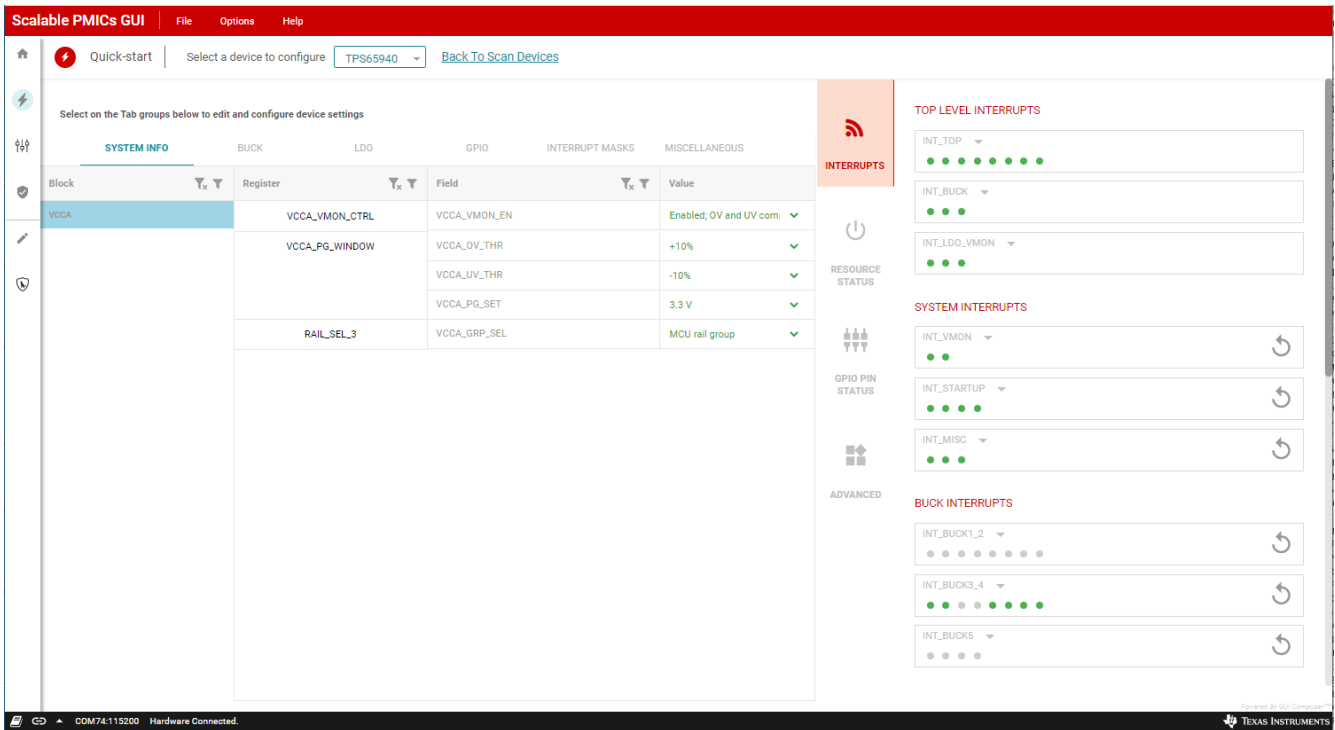


图 6-4. 系统信息

备注

无法通过“Quick-start”（快速启动）或“Register Map”（寄存器映射）页面更改相位配置。若要更改相位配置，必须重新配置器件 NVM，请参阅节 8。

6.2.2 BUCK

相位配置将决定可以通过“Quick-start”（快速启动）页面来编辑哪些 BUCK。在图 6-5 中，BUCK2 信息不可用，因为 BUCK2 与 BUCK1 构成了多相位。BUCK1 被视为主器件，BUCK2 被视为辅助器件，BUCK1 的所有属性都应用于 BUCK2。有关字段和相关操作的更详细说明，请参阅器件数据表。

Block	Register	Field	Value
Buck1_2	BUCK1_CTRL	BUCK1_EN	Disabled; BUCK1 regulatc
		BUCK1_FPWM	PWM operation only.
		BUCK1_FPWM_MP	Automatic phase adding
		BUCK1_VMON_EN	Disabled; OV and UV com
		BUCK1_VSEL	BUCK1_VOUT_1
		BUCK1_PLDN	Enabled; Pull-down resist
	BUCK1_CONF	BUCK1_SLEW_RATE	2.5 mV/us
		BUCK1_ILIM	5.5 A
	BUCK1_VOUT_1	BUCK1_VSET1	0.800 V
		BUCK1_VSET2	0.3 V
BUCK1_PG_WINDOW	BUCK1_OV_THR	+4% / +40 mV	
	BUCK1_UV_THR	-4% / -40 mV	
RAIL_SEL_1	BUCK1_GRP_SEL	SOC rail group	

Resource	Status
BUCK1_2	Disabled
BUCK3	Disabled
BUCK4	Enabled
BUCK5	Disabled
LDO1	Enabled
LDO2	Disabled
LDO3	Disabled
LDO4	Disabled

图 6-5. BUCK 配置

备注

频率选择寄存器 **FREQ_SEL** 受到保护，无法通过“Quick-start”（快速启动）页面进行更改。若要更改该字段，与相位配置类似，必须相应地更新 NVM。

6.2.3 LDO

可以在“LDO”选项卡中访问 LDO 配置。有关字段和相关操作的更详细说明，请参阅器件数据表。如果 PMIC 没有 LDO，“LDO”选项卡将被省略。

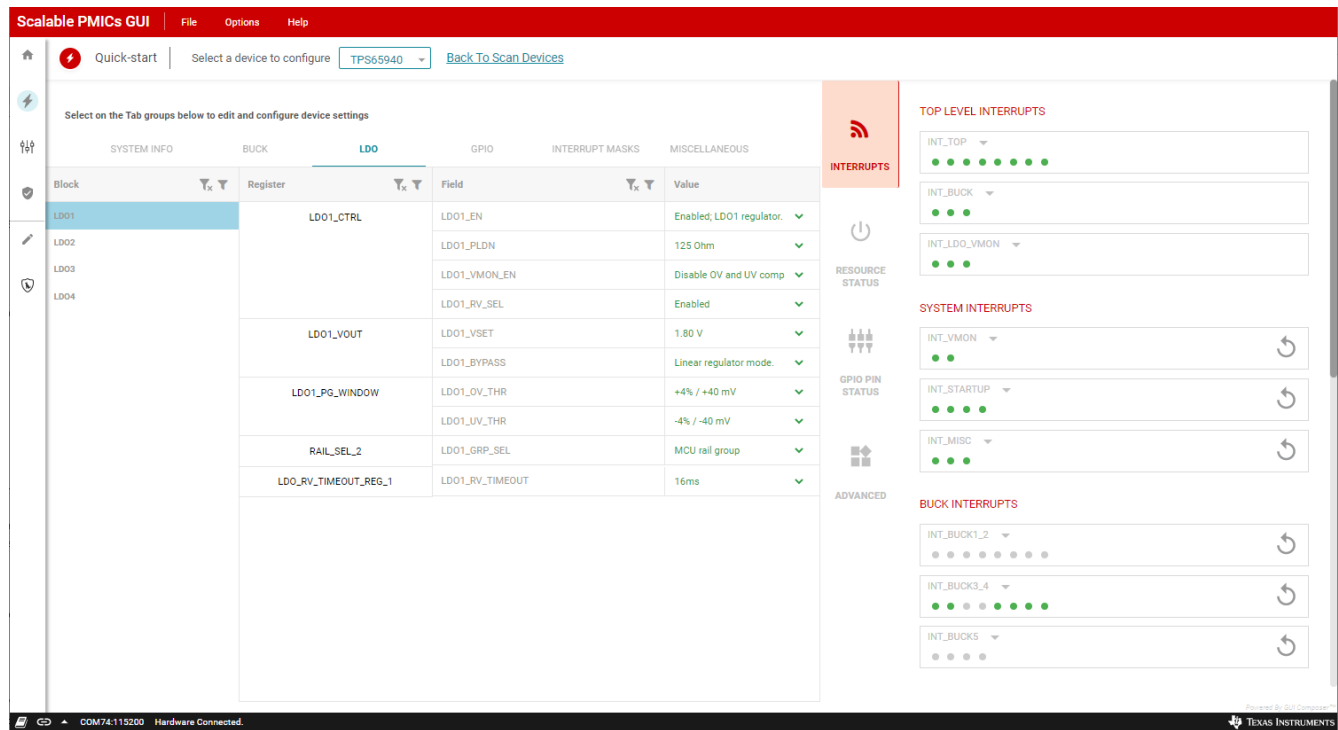


图 6-6. LDO 配置

6.2.4 GPIO

GUI 提供了用于配置 PMIC GPIO 的界面。当 GPIO 配置为输入时，GUI 提供了一种额外的机制来通过关联的 AEVM 引脚来驱动 GPIO 以及通过页面右侧的“GPIO Status”（GPIO 状态）垂直选项卡来查看状态变化。

PMIC GPIO 可配置为输入或输出，或映射到 PMIC 的内部功能。在本示例中，PMIC 将 GPIO1 和 GPIO2 用于第二个 I²C 实例。“GPIO PIN STATUS”（GPIO 引脚状态）窗格可用于确认每个引脚在配置为 GPIO 时的功能和电平。当将 GPIO 方向设置为输入时会显示 CONFIGURE GPIO LEVEL（配置 GPIO 电平）框（请参阅图 6-7），用于对 AEVM 输出电平进行初始设置。请确保将硬件平台配置为支持预期的 GPIO 操作。

备注

在评估多 PMIC 解决方案时，AEVM 输出控制仅用于连接到 GUI 的 AEVM（通过 USB 端口连接到 PC）。

GPIO 引脚状态（请参阅图 6-7）指示 GPIO 的功能以及未配置为特殊功能的 GPIO 的当前状态（高电平或低电平）。颜色为灰色、红色和绿色。除了颜色之外，还在每个指示器的右侧提供了文字说明。

The screenshot displays the Scalable PMICs GUI for device TPS65940. The main configuration table is as follows:

Block	Register	Field	Value
GPIO 1	GPIO1_CONF	GPIO1_OD	Open-drain output
		GPIO1_DIR	Input
		GPIO1_SEL	SCL_I2C2/CS_SPI
		GPIO1_PU_SEL	Pull-down resistor selects
		GPIO1_PU_PD_EN	Disabled; Pull-up/pull-dov
		GPIO1_DEGLITCH_EN	No deglitch, only synchro
		GPIO1_OUT	Low
GPIO 2-11	GPIO_OUT_1	GPIO1_OUT	Low
		GPIO2	
		GPIO3	
		GPIO4	
GPIO 5-11			
NPWRON and nRSTOUT			

The 'GPIO PIN STATUS' panel on the right shows the following status for pins 1-11:

- GPIO1: SCL_I2C2/CS_SPI
- GPIO2: SDA_I2C2/SDO_SPI
- GPIO3: GPIO3 (Red)
- GPIO4: GPIO4 (Green)
- GPIO5: SCLK_SPMI
- GPIO6: SDATA_SPMI
- GPIO7: GPIO7 (Green)
- GPIO8: GPIO8 (Red)
- GPIO9: GPIO9 (Green)
- GPIO10: GPIO10 (Green)
- GPIO11: GPIO11 (Red)

图 6-7. GPIO 配置

6.2.5 中断

在“Interrupts”（中断）选项卡中，用户可以决定屏蔽或监视各种中断源：状态机、GPIO、BUCK 等。

图 6-8 右侧所示的中断状态可用于监视中断事件。中断根据功能进行分组，并且可以将其展开以查看每个单独的中断源。**TOP LEVEL INTERRUPTS**（顶层中断）是只读的，无法清除。其他中断可以在寄存器或位级别进行清除，如复位符号所示。带有红点的 **ERROR**（错误）状态表示发生了中断，而带有绿点的 **NORMAL**（正常）状态表示没有发生中断或中断已被清除。通常，灰色表示被屏蔽的中断。如果某个中断具有带灰点的 **NORMAL**（正常）状态，则表明该中断不适用于指定的相位配置。GUI 将忽略任何取消屏蔽或生成不适用于器件相位配置的中断的尝试。

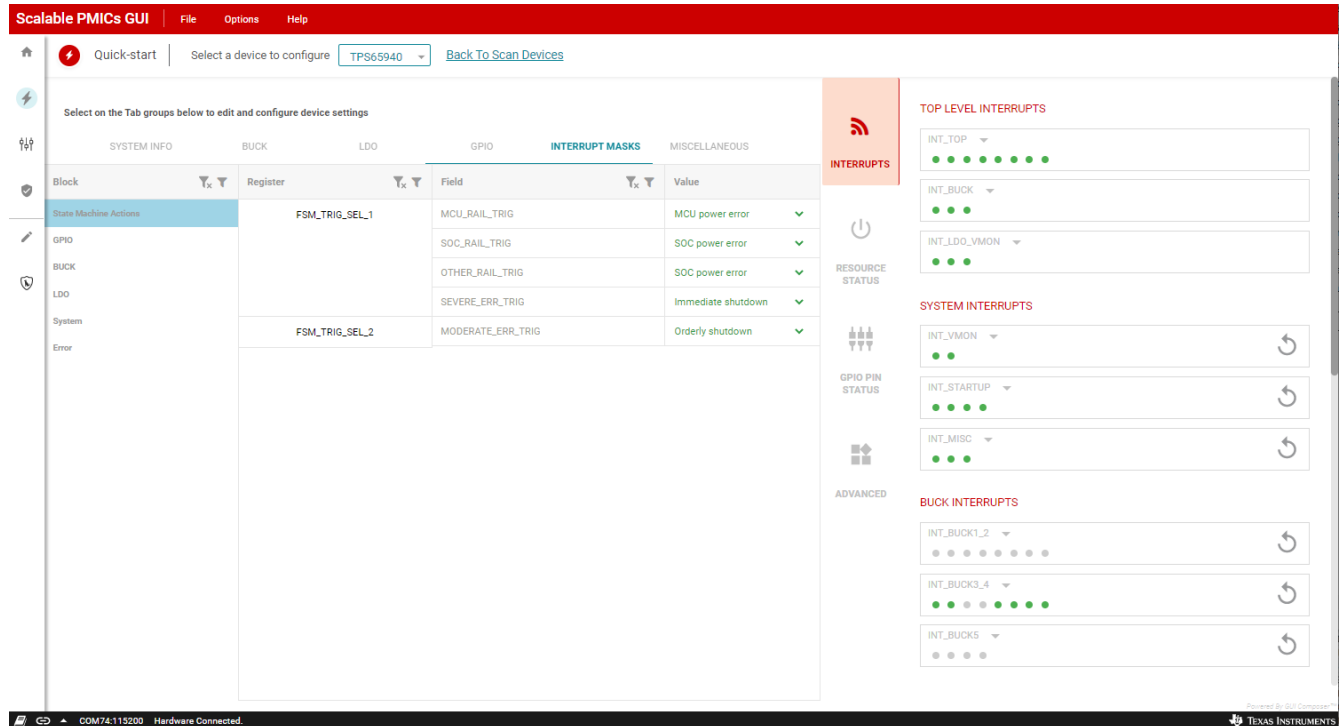


图 6-8. 中断屏蔽和状态

6.2.6 其他设置

“MISCELLANEOUS” (其他) 选项卡包括设置电源正常 (PGOOD)、展频配置和 PMIC 的其他配置。

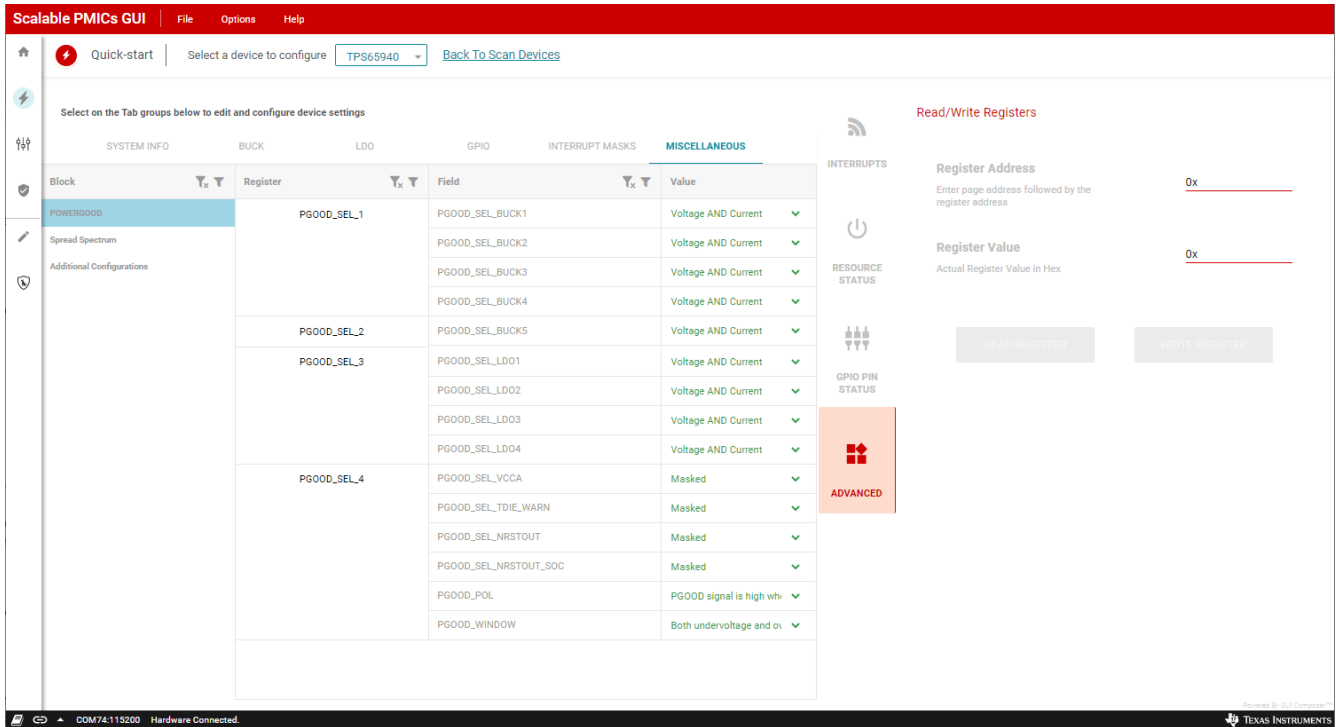


图 6-9. 其他设置

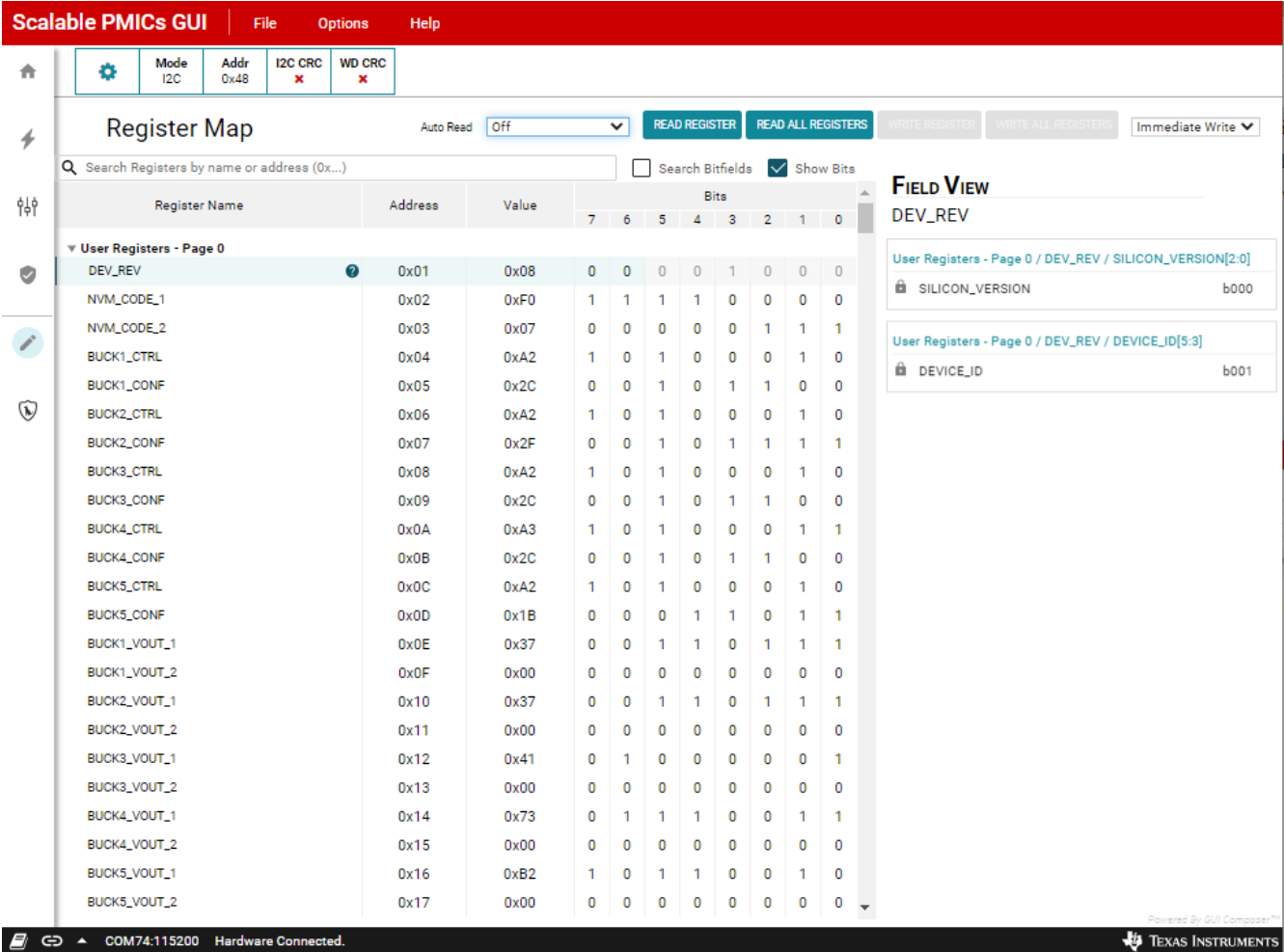
6.2.7 高级选项卡

“Advanced” (高级) 选项卡提供对寄存器的直接写入和读取访问。访问格式为 0xabc，其中 a 为页码，bc 为寄存器地址。有关给定器件寄存器的页码和寄存器地址，请参阅器件数据表。

7 “Register Map” (寄存器映射) 页面

“Register Map” (寄存器映射) 页面列出了可用于配置的不同寄存器。与“Quick-start” (快速启动) 页面不同, 如果存在多个 PMIC, 这里没有器件标签可供选择。在 I²C 的情况下, 在 GUI 顶部的 *Options* (选项卡) 下的 (器件设置) 中选择器件地址。对于 SPI, 片选引脚的硬件连接将决定 GUI 与哪个 PMIC 进行通信, 并将其内容显示在“Register Map” (寄存器映射) 页面中。

“Register Map” (寄存器映射) 页面用于直接对 PMIC 寄存器进行读取和写入。读取可以分别完成, 也可以一次性全部完成。类似地, 寄存器写入也可以一次全部完成或分别完成。在 *Immediate Write* (立即写入) 模式 (位于页面右上角的选项) 下, 只能在每次更改“Field View” (字段视图) (位更改或十六进制值更改) 后, 立即对各个寄存器分别进行写入。在 *Deferred Write* (延迟写入) 模式下, 选择 **WRITE REGISTER** (写入寄存器) 或 **WRITE ALL REGISTERS** (写入所有寄存器) 按钮后, 单个寄存器或所有寄存器才会执行写入。



Register Name	Address	Value	Bits							
			7	6	5	4	3	2	1	0
▼ User Registers - Page 0										
DEV_REV	0x01	0x08	0	0	0	0	1	0	0	0
NVM_CODE_1	0x02	0xF0	1	1	1	1	0	0	0	0
NVM_CODE_2	0x03	0x07	0	0	0	0	0	1	1	1
BUCK1_CTRL	0x04	0xA2	1	0	1	0	0	0	1	0
BUCK1_CONF	0x05	0x2C	0	0	1	0	1	1	0	0
BUCK2_CTRL	0x06	0xA2	1	0	1	0	0	0	1	0
BUCK2_CONF	0x07	0x2F	0	0	1	0	1	1	1	1
BUCK3_CTRL	0x08	0xA2	1	0	1	0	0	0	1	0
BUCK3_CONF	0x09	0x2C	0	0	1	0	1	1	0	0
BUCK4_CTRL	0x0A	0xA3	1	0	1	0	0	0	1	1
BUCK4_CONF	0x0B	0x2C	0	0	1	0	1	1	0	0
BUCK5_CTRL	0x0C	0xA2	1	0	1	0	0	0	1	0
BUCK5_CONF	0x0D	0x1B	0	0	0	1	1	0	1	1
BUCK1_VOUT_1	0x0E	0x37	0	0	1	1	0	1	1	1
BUCK1_VOUT_2	0x0F	0x00	0	0	0	0	0	0	0	0
BUCK2_VOUT_1	0x10	0x37	0	0	1	1	0	1	1	1
BUCK2_VOUT_2	0x11	0x00	0	0	0	0	0	0	0	0
BUCK3_VOUT_1	0x12	0x41	0	1	0	0	0	0	0	1
BUCK3_VOUT_2	0x13	0x00	0	0	0	0	0	0	0	0
BUCK4_VOUT_1	0x14	0x73	0	1	1	1	0	0	1	1
BUCK4_VOUT_2	0x15	0x00	0	0	0	0	0	0	0	0
BUCK5_VOUT_1	0x16	0xB2	1	0	1	1	0	0	1	0
BUCK5_VOUT_2	0x17	0x00	0	0	0	0	0	0	0	0

图 7-1. 寄存器映射

备注

尽管所有寄存器都显示在“Register Map” (寄存器映射) 页面中, 但并非所有寄存器都可以通过该页面进行编辑。具体来说, 无法更改接口配置, 并且与“Quick-start” (快速启动) 页面类似, 无法更改 Buck 频率。不过, 不会报告任何错误, 每次写入都与一次读取相关联。读取将更新显示内容, 因此, 对受保护字段的写入可以准确地反映出该写入是不成功的。

8 NVM 配置页面

“NVM Configuration”（NVM 配置）页面是 GUI 的主要功能，突出了 PMIC 的可配置性。配置由静态和动态 (PFSM) 设置组成，可针对广泛的应用进行自定义。“NVM Configuration”（NVM 配置）页面还提供了界面，可用于将配置下载到目标器件的 NVM 中。可以在创建自定义配置后执行下载，也可以使用现有的 NVM 配置执行下载。

8.1 创建自定义配置

“NVM 配置”页面无需硬件来开发 NVM 配置。只有在尝试将配置上传到目标器件中时才需要连接实际器件。

可以通过三种机制进行开发。第一种是使用屏幕顶部“File”（文件）选项卡下方的 *Open Configuration*（打开配置）功能，在同一个“File”（文件）选项卡下方的 *Save Configuration*（保存配置）功能打开先前保存的配置。其次，也以模板的形式提供标准配置，可以在 *Select a template to start with*（选择要开始的模板）下方进行选择。图 8-1 显示了初始“Select”（选择）视图，其中包含所选择的上传和模板机制。

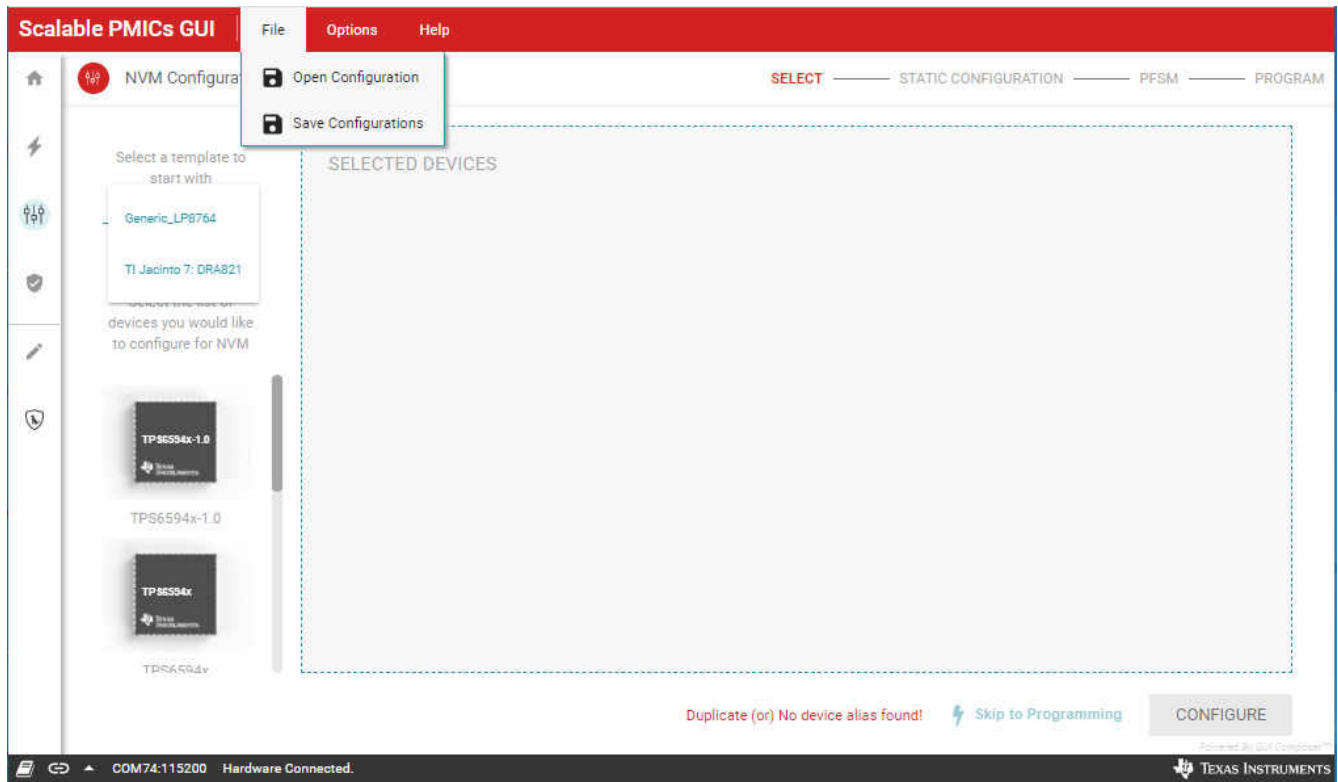


图 8-1. 打开现有配置

最后，左侧提供了器件图标，可以选择这些图标来创建单个或多个 PMIC 应用，如图 8-2 所示。添加器件后，可以编辑器件名称，并选择主器件/次器件。GUI 要求使用唯一的器件名称并且只有一个主器件。

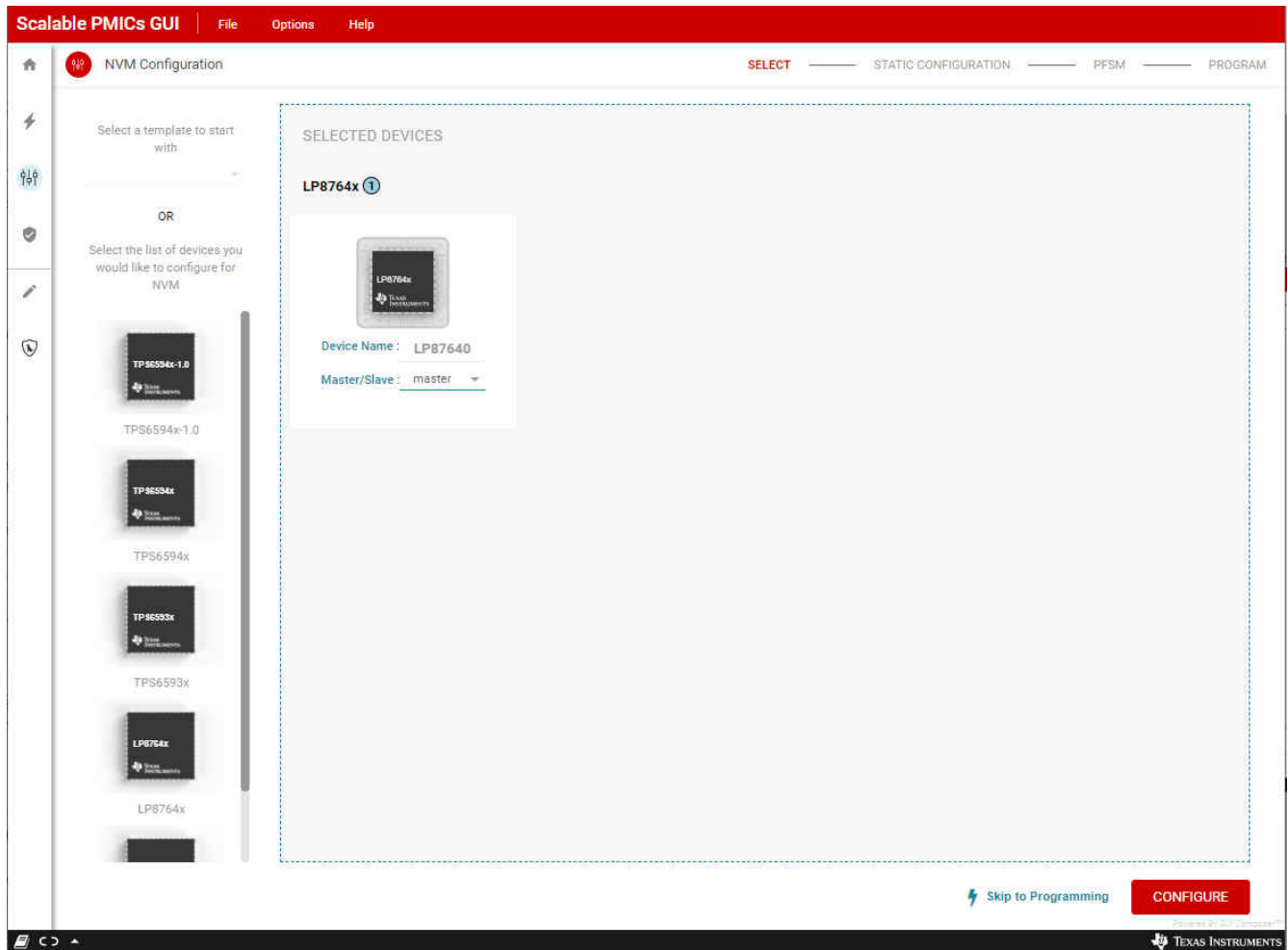


图 8-2. 从空白模板开始

上传配置或选择用于应用的器件后，开发流程将转至“Static Configuration”（静态配置）和“PFSM”透视图，然后最终到达“Program”（编程）透视图，图 8-3 突出显示了这些透视图。节 8.2.1 将介绍如何使用图 8-3 底部的 *Skip to Programming*（跳至编程选项）选项对现有 NVM 配置进行编程。

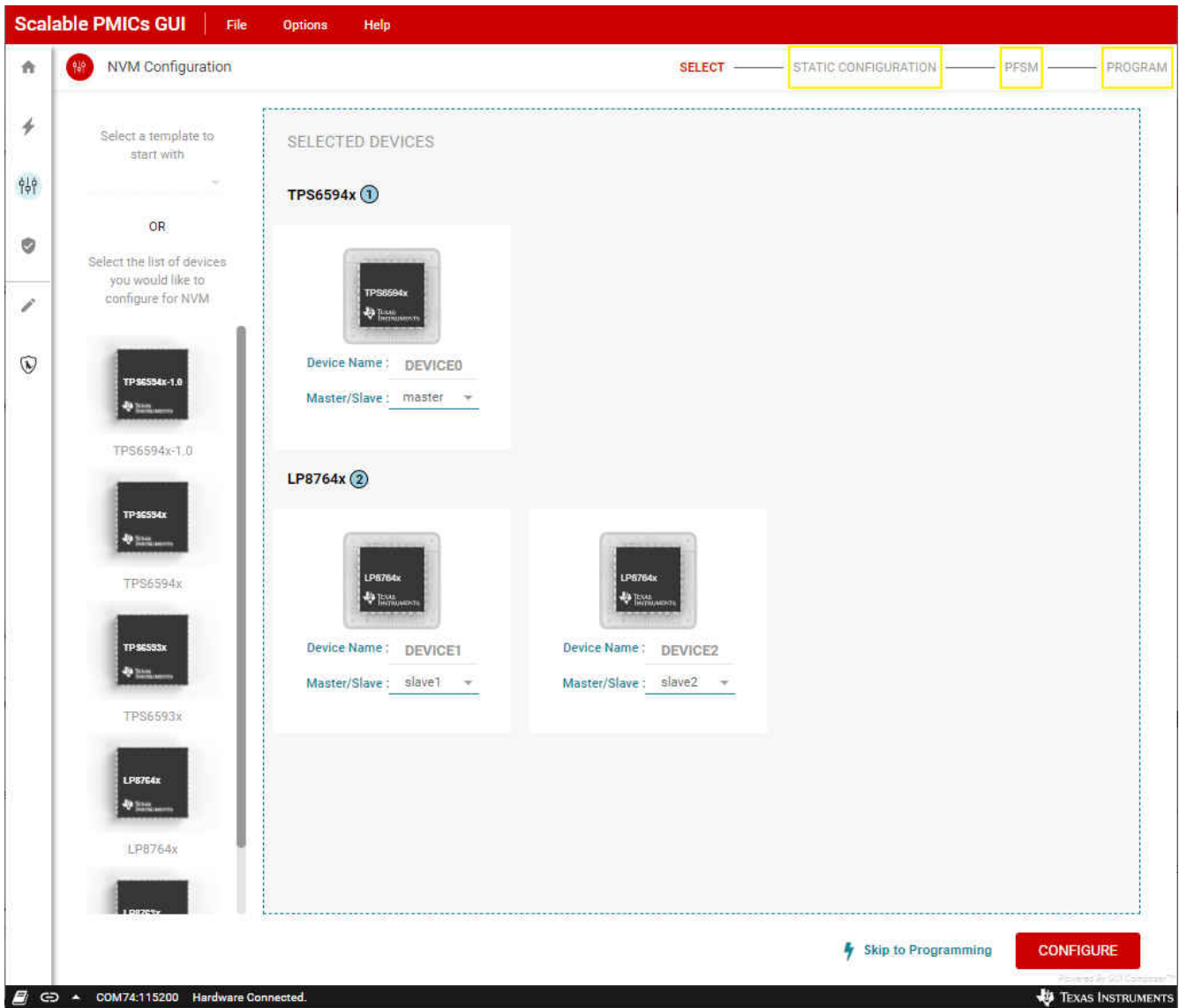


图 8-3. 配置开发流程

备注

不建议更改或编辑现有配置的选定器件（器件名称除外）。否则可能会破坏 PFSM 中的依赖关系。如果在开发过程中发现器件的数量或类型定义错误，则需要通过空白模板或现有模板重新启动新配置。

8.1.1 静态配置

“Static Configuration”（静态配置）透视图提供了与“Quick-start”（快速启动）页面中的内容相似的界面。建议的流程是从最左侧的“System Info”（系统信息）选项卡开始，更新“System Info”（系统信息）选项卡中的每个块，然后进入下一个配置选项卡。在多器件系统中，只需从页面左上角的“Select a device to configure”（选择一个要配置的器件）下拉菜单中选择要配置的器件。

每个块中都有一个寄存器和字段列表，它们直接与器件寄存器和字段名称相匹配。建议根据数据表规格来理解并正确设置给定应用的字段值。BUCK 块中还有一个图形选择工具（请参阅图 8-4），用于将寄存器设置抽象成器件数据表中也介绍过的用例。该图形选择工具将显示在页面的最右一列，提供配置用例名称以及建议的电感器值。

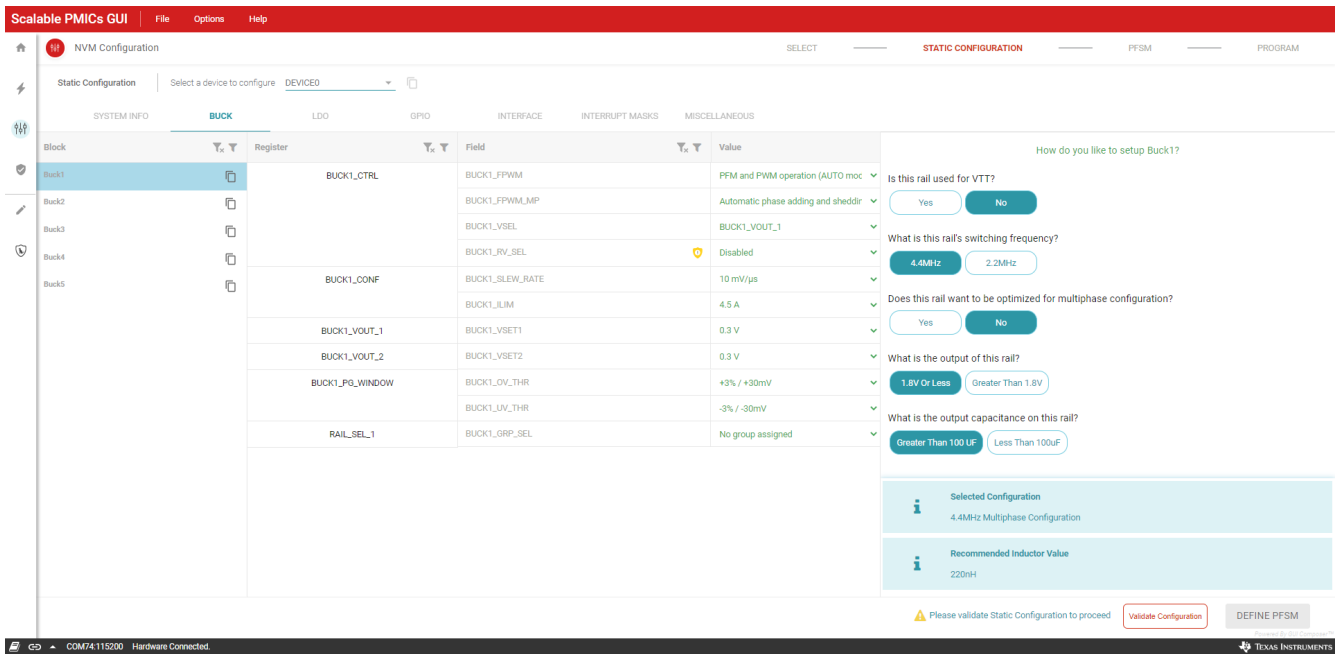


图 8-4. BUCK 静态配置

备注

一次仅显示一个器件。确保应用中的所有器件都已定义。

在“Static Configuration”（静态配置）透视图图中，GUI 监视配置的有效性。具体来说，对于多器件解决方案，GUI 确保电源（VCCA 和 nPWRON）和接口选择相匹配。点击透视图底部的 **Validation Failed**（验证失败）文本后会显示一个弹出窗口（如图 8-5 所示），该窗口描述所有无效并阻止进入开发下一个阶段的问题。

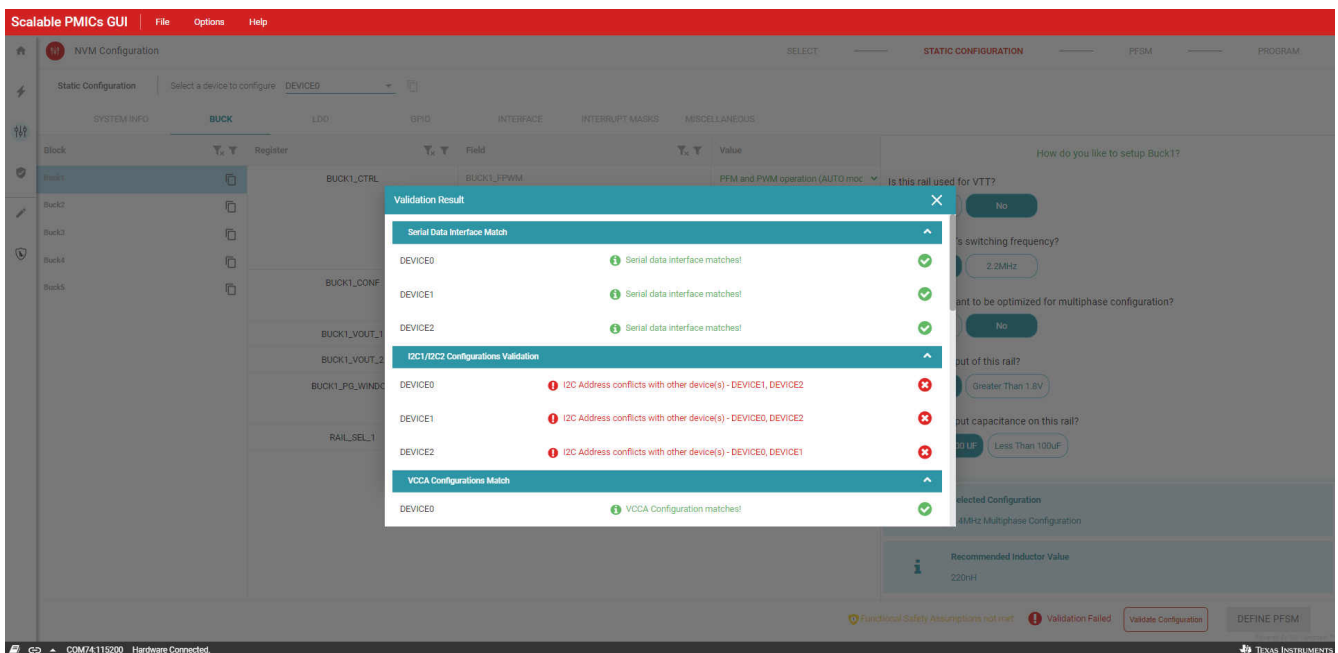


图 8-5. 静态配置之验证失败

在本示例中，全部三个器件具有相同的 I²C 地址。将每个器件更新为唯一的地址后，GUI 现在显示“Validation Success”（验证成功），并且“Define PFSM”（定义 PFSM）按钮现在处于有效状态。

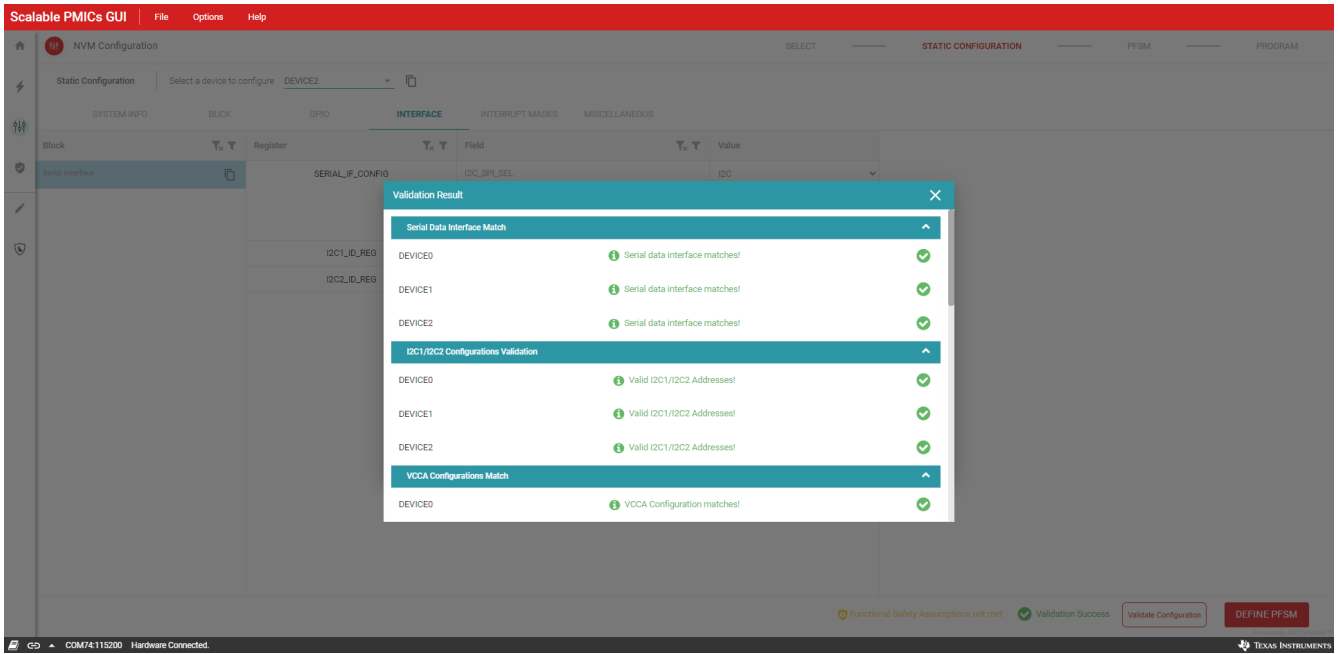


图 8-6. 静态配置之验证通过

除了检查静态设置外，在支持功能安全的器件中还要检查其他安全相关功能。点击 **Functional Safety Assumptions not met** (功能安全假设未满足) 文本后，会显示一个弹出窗口，如图 8-7 所示。该窗口列出了与器件的功能安全假设相关的参数。请参阅器件的功能安全手册和应用的功能安全目标，以确认所选设置符合其功能安全目标。

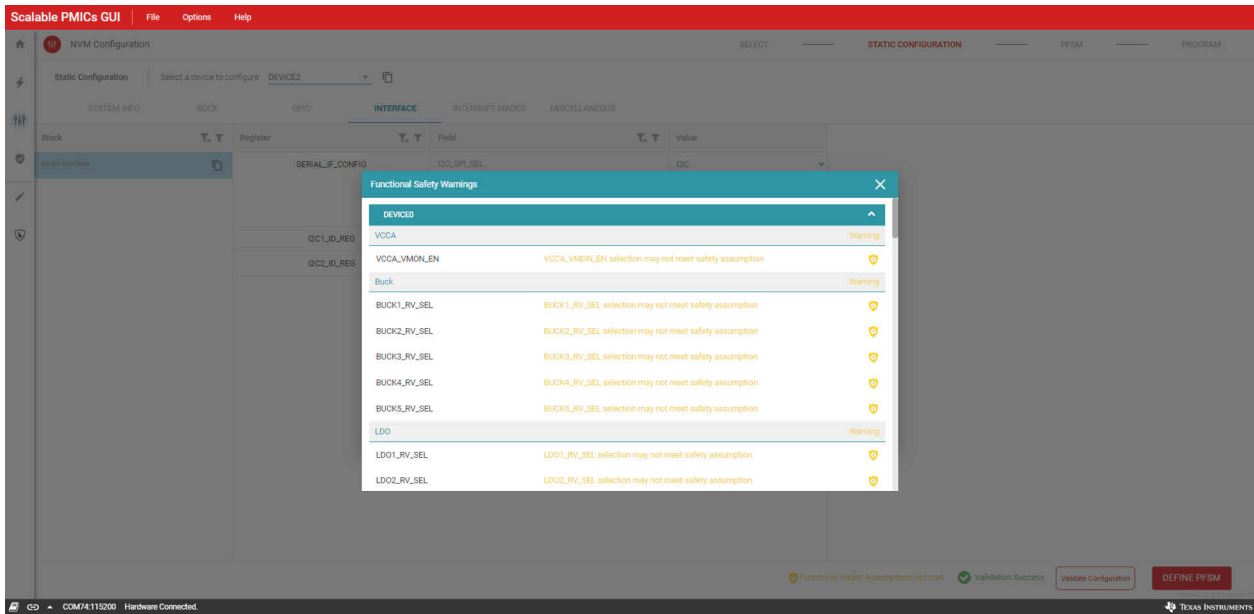


图 8-7. 功能安全假设检查

在从静态设置切换到“PFSSM”透视图时，如果有任何值未经过更新，则会显示警告消息。对于未更新的值，将在其周围以黄色边框突出显示。在本示例中，仅更新了 I²C 地址。德州仪器 (TI) 建议查看并确认所有突出显示为未更改的设置。

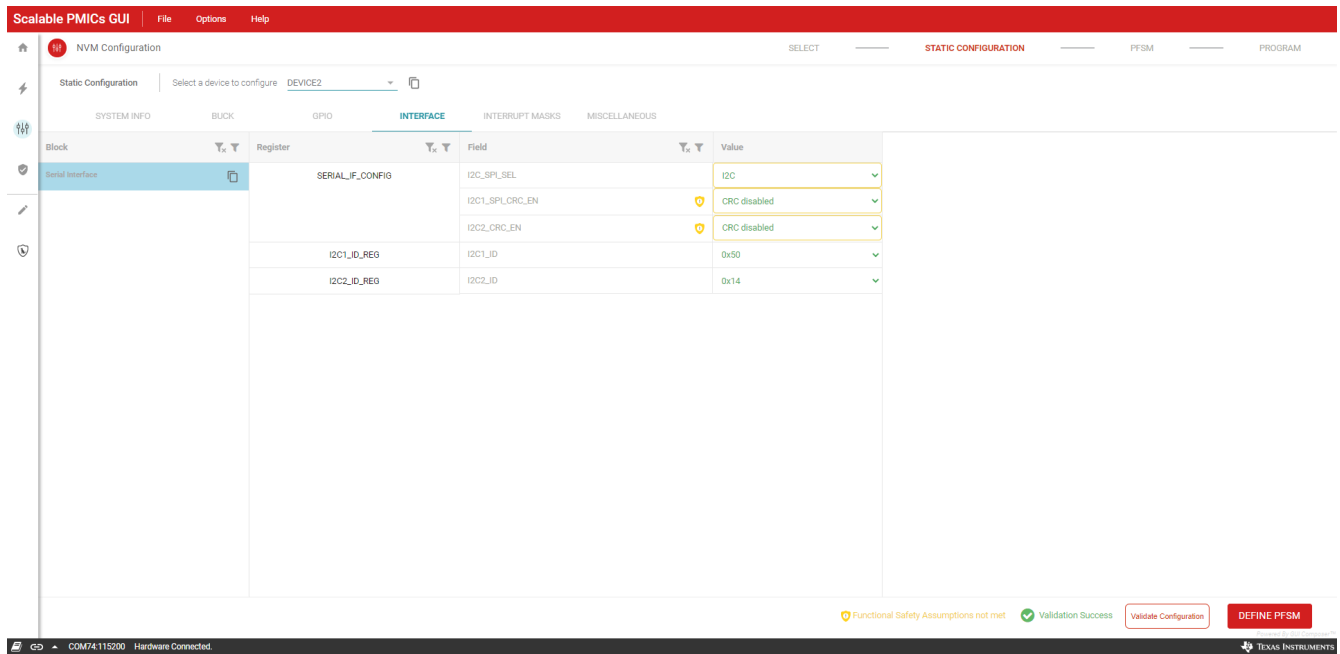


图 8-8. 未更新的静态配置值示例

备注

GUI 不提供自动保存功能。应经常使用“File”（文件）选项卡项下的“Save Configurations”（保存配置）进行保存，如图 8-9 所示。



图 8-9. 经常保存

8.1.2 可预配置的任务状态 (PFSM)

虽然静态配置是针对每个器件执行的，但在“PFSM”透视图中该配置是针对一同工作的所有器件执行的。会为每个器件创建单独的命令，但这些命令在系统解决方案的状态和转换的上下文中进行分组。

当进入 PFSM 透视图且未从 SELECT 透视图选择模板时，将提供一个启动模板以帮助开发 PFSM。图 8-10 显示了此模板。

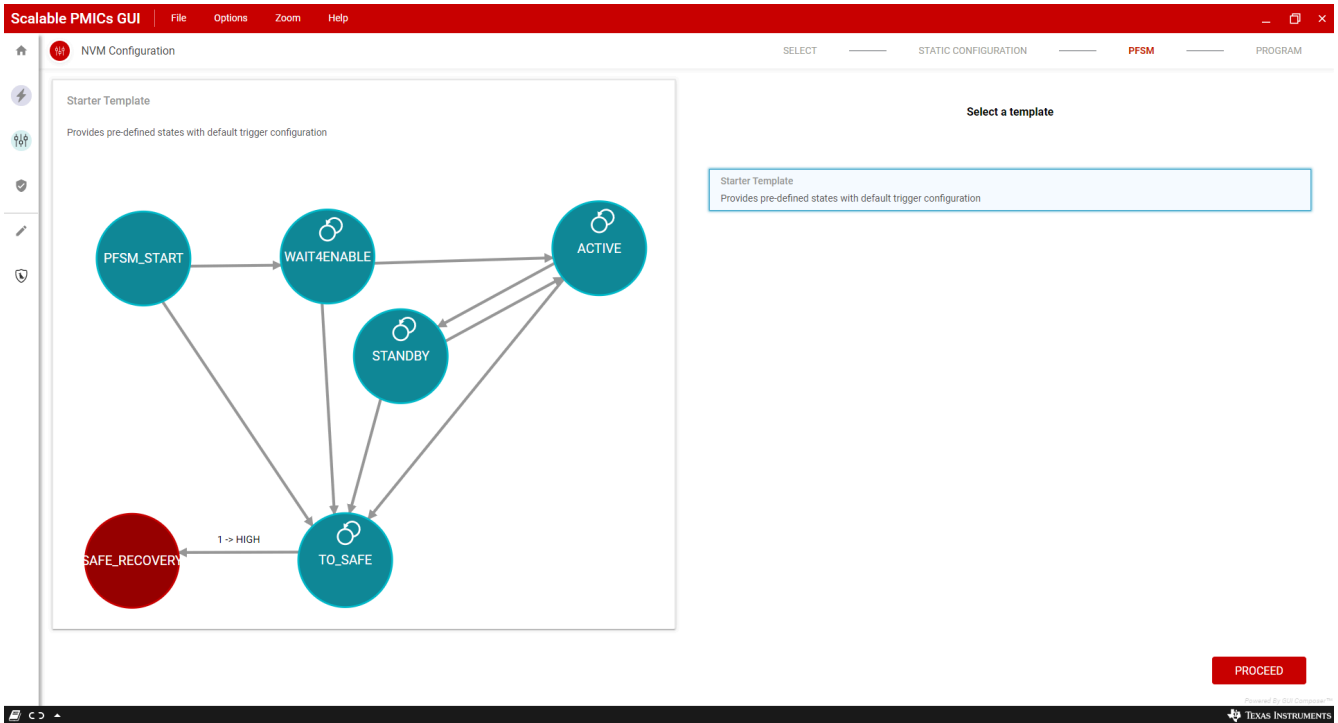


图 8-10. PFSM 启动模板

选择模板后，GUI 将转换到“PFSM”透视图。

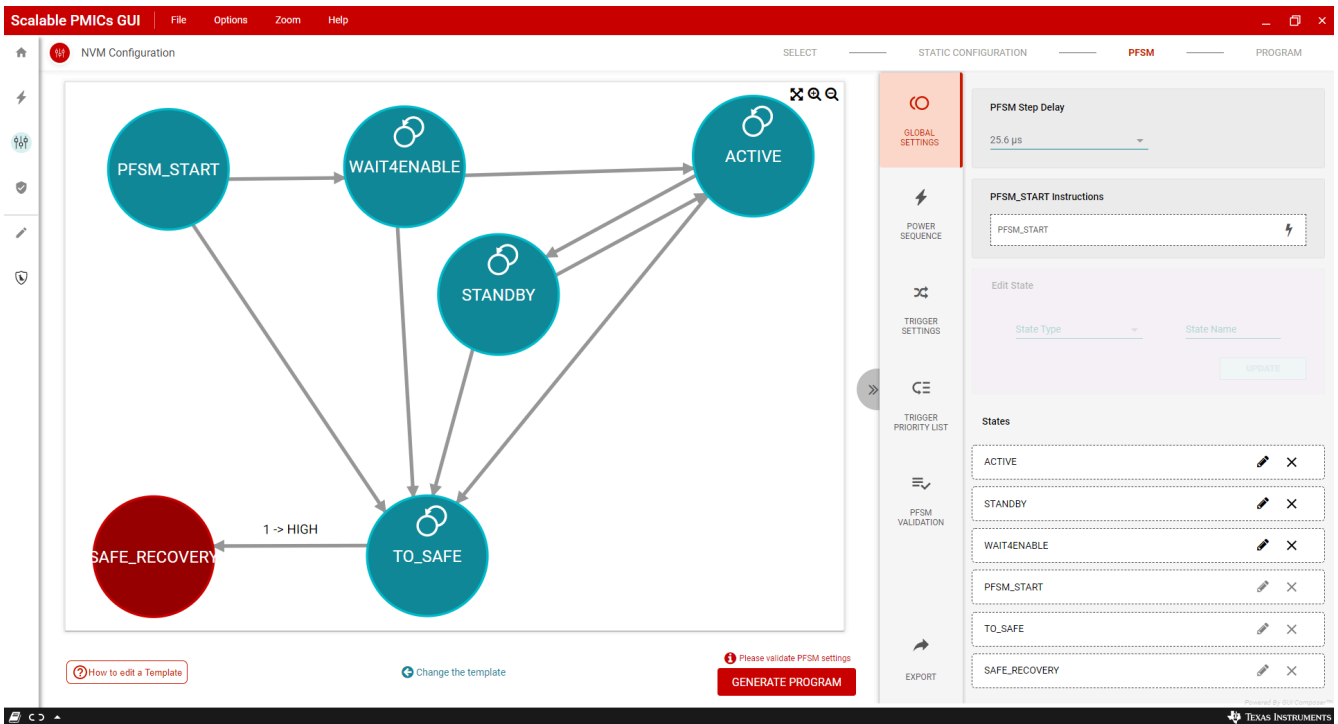


图 8-11. PFSM 配置

“PFSM”透视图在左侧提供了一个工作区来绘制 PFSM。表 8-1 以及左下角的 *How to edit a Template* (如何编辑模板) 弹出窗口说明了如何添加和删除状态和转换。

表 8-1. 用于编辑状态机的操作

操作	接口
创建一个新状态	Shift + 点击鼠标左键
创建转换	Shift + 点击鼠标左键 + 从源状态内拖动到目标状态
创建循环	在状态内鼠标左键点击循环图标
移动状态	在状态内点击鼠标左键并拖动
删除	鼠标左键点击状态、转换或循环，然后按 Backspace (Delete) (退格键) (删除)
缩放	将鼠标放在绘图区域内并滚动
平移	在工作区内的任何空白区域中点击鼠标左键 + 拖动

备注

GUI 不提供自动保存功能。应经常使用“File” (文件) 选项卡项下的 *Save Configurations* (保存配置) 进行保存，如图 8-9 所示。

开发 PFSM 可以是一个迭代的非线性过程。此处列出的线性过程仅用于展示功能和基本工作流程。可以随时返回到先前的步骤 (步骤 1-5) 进行更改。

1. 创建状态图
 - a. 添加状态
 - b. 添加转换 (触发条件)
2. 全局设置
3. 电源序列
4. 触发条件设置
5. 触发条件优先级列表
6. PFSM 验证

8.1.2.1 创建状态图

在“PFSM”面板中开发的状态图包含用户定义的任务状态和三种硬件状态，即 **SAFE_RECOVERY**、**LP_STANDBY** 和 **RUNTIME_BIST**。要在硬件状态和任务状态之间建立联系，需要用户定义的任务状态 **PFSM_START**。**TO_SAFE** 状态作为是任务状态和硬件状态 **SAFE_RECOVERY** 之间的转换是必需的。如图 8-12 所示，该面板重点显示任务状态以及 **SAFE_RECOVERY**、**LP_STANDBY** 和 **RUNTIME_BIST** 硬件状态，可以通过 PFSM 从任务状态进行访问。

WARNING

PFSM_START 状态、TO_SAFE 状态和 SAFE_RECOVERY 状态是必需的。

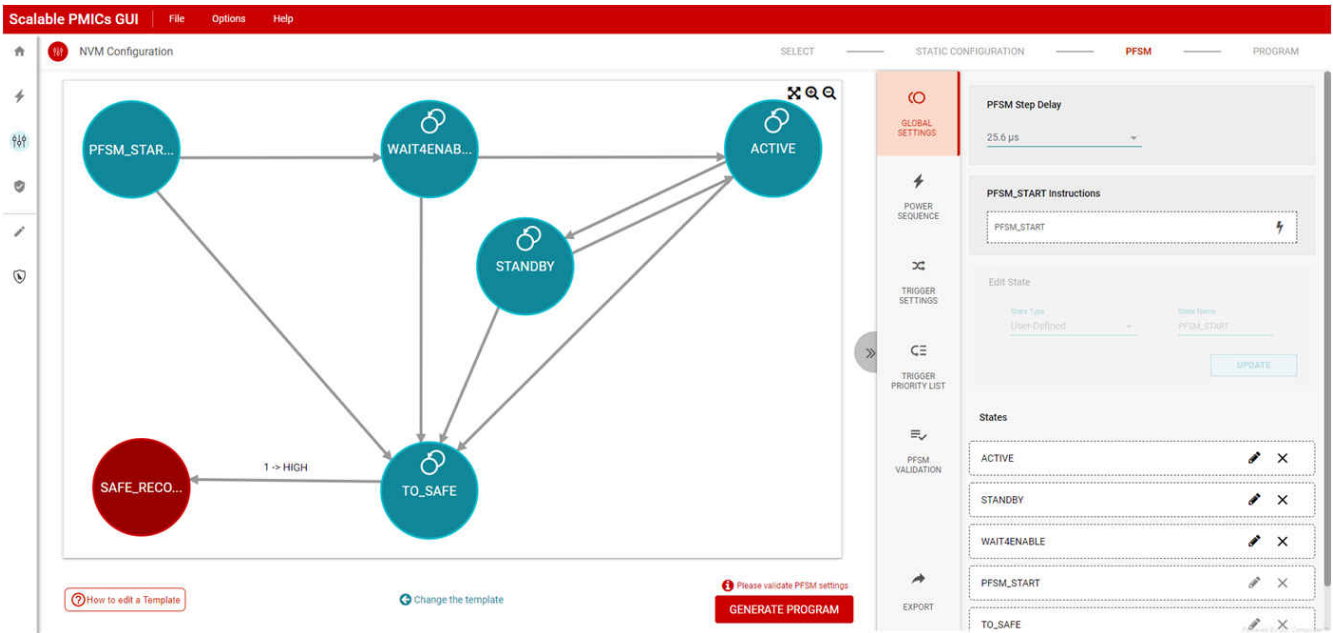


图 8-12. PFSM 状态图面板

图 8-13 显示了完整的状态机以及硬件和任务状态之间相关的其他转换。可以通过选择右下角的导出图标（位于“GENERATE PROGRAM”（生成程序）旁边）来查看完整的状态机。有关状态和转换的更详细的说明，请参阅器件规格。

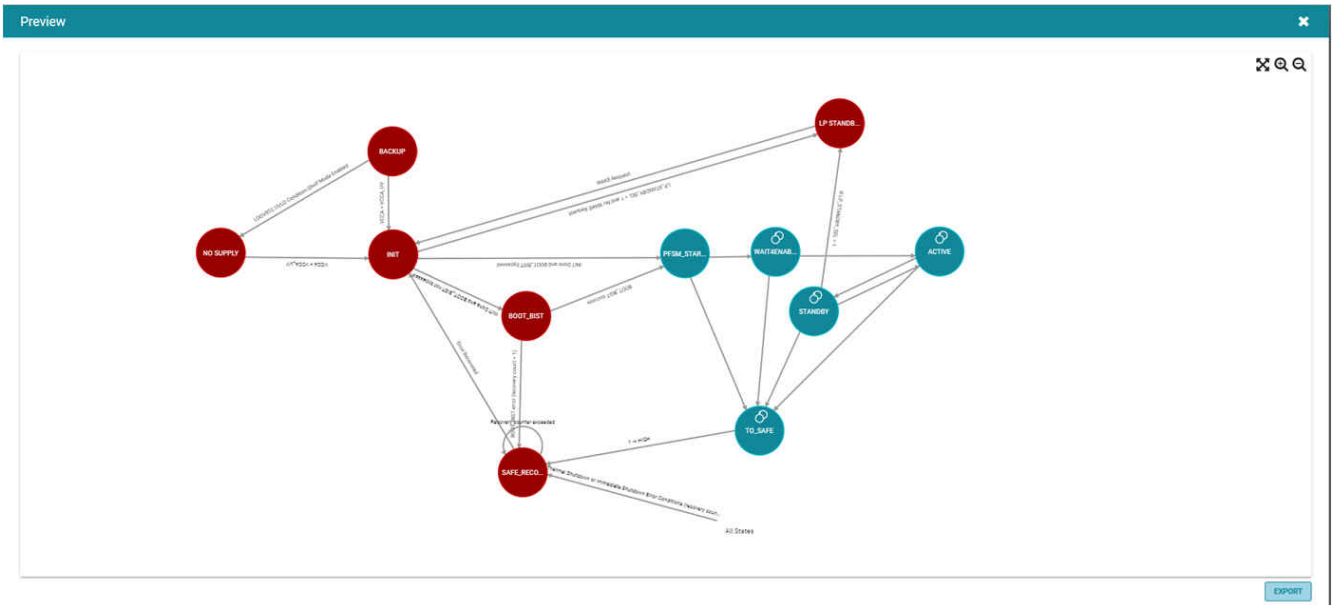


图 8-13. 完整的状态图

8.1.2.2 全局设置

添加状态后，它们将显示在 **GLOBAL SETTINGS**（全局设置）面板中，如图 8-14 所示。状态的名称是可配置的，但状态类型仅限于用户定义的状态或硬件状态。已经在 PMIC 的有限状态机中定义了硬件状态，根据定义，除了 LP_STANDBY 之外，没有与转换到硬件状态相关的电源序列，也不能从硬件状态定义转换。

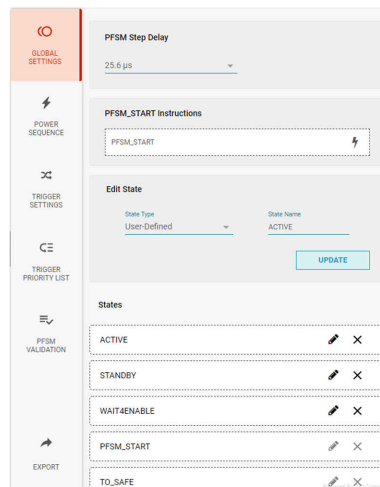


图 8-14. 全局设置

PFSM 将始终从 PFSM_START 状态开始。此状态包括所有的 TRIG_SET 定义以及初始 TRIG_MASK。默认情况下，在 PFSM_START 中找到的 TRIG_MASK 由在 GUI 中退出 PFSM_START 状态的箭头定义。不能对 PFSM_START 状态定义任何箭头。在 GLOBAL SETTINGS (全局设置) 中，用户可以编辑 PFSM_START 状态下的 TRIG_MASK，还可以在最后一条 TRIG_SET 指令之后添加将附加到 PFSM_START 序列的指令。

PFSM Step Delay (PFSM 步进延迟) 设置也是 GLOBAL SETTINGS (全局设置) 的一部分。“PFSM Step Delay” (PFSM 步进延迟) 设置将确定 GUI 将使用哪个时间间隔来尝试满足在电源序列中发现的所需延迟。实际延迟是所需延迟、所使用的指令以及 PFSM 步进延迟的函数。指令延迟限制为步进延迟的 6 位或 8 位倍数。如果 GUI 无法通过现有的步进延迟达到所需的延迟时间，或者如果步进实际上大于所需的延迟，GUI 将在 PFSM 验证期间产生错误。表 8-2 举例说明了实际延迟 (PFSM 步进延迟的函数) 与请求的延迟时间之间的关系。

备注

如果选择的 PFSM 步进延迟是电源时序中所需的大多数延迟的公因数，则可以优化存储器在器件中的使用。

表 8-2. 请求的延迟时间和实际的延迟时间示例

PFSM 步进延迟 (us)	请求的延迟 (us)	延迟指令	实际延迟 ³ (us)
25.6	2500	DELAY_IMM (8 位)	2483.2
		REG_WRITE_VCTRL_IMM (6 位)	2457.6
204.8	40000	DELAY_IMM (8 位)	39936
		REG_WRITE_VCTRL_IMM (6 位)	39321
409.6	300000	DELAY_IMM (8 位)	299827
		REG_WRITE_VCTRL_IMM (6 位)	294912

8.1.2.3 电源序列

电源序列取决于目标状态定义，因此需要在创建电源序列之前在全局设置中定义该状态。大多数序列适用于到特定状态的转换。可以使某些序列与目标状态无关，以便可以在多个状态中重复使用相同的序列。这些序列要求选择的目标状态是“ANY”，并且只映射到自终止的转换，如图 8-15 和图 8-16 中所示。

³

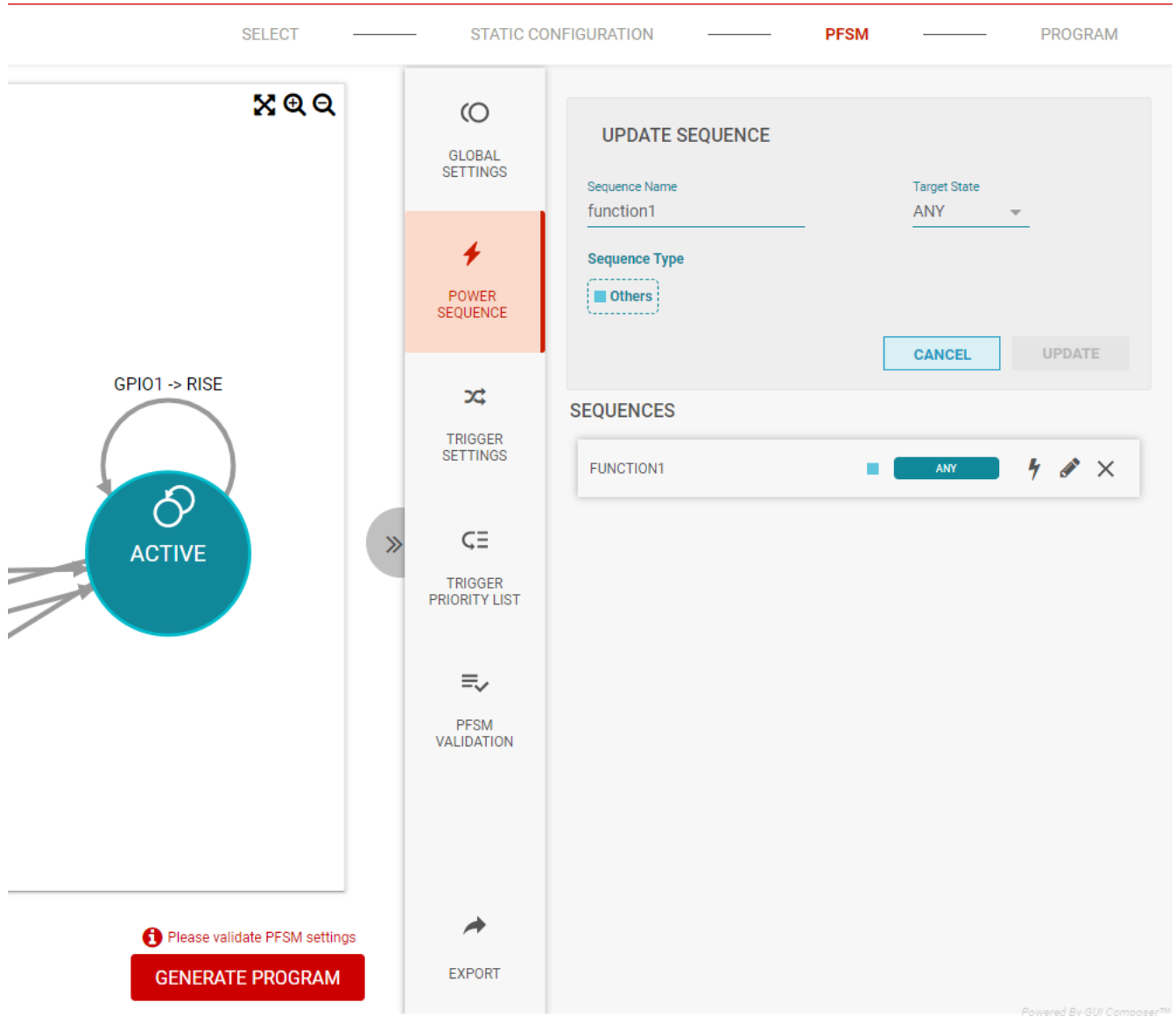
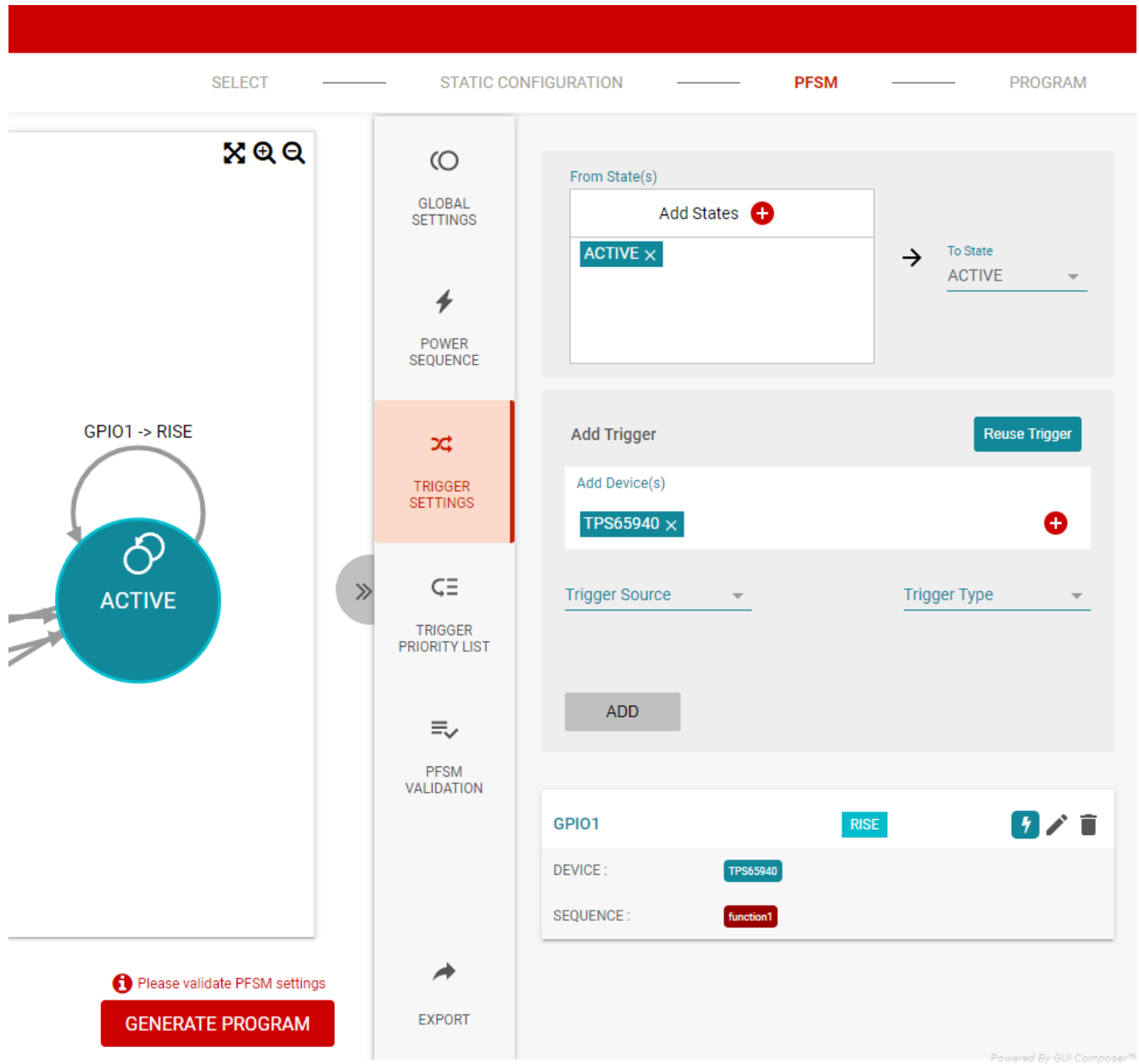


图 8-15. 目标状态设置为“ANY”



The screenshot displays the PFSM configuration page with the following components:

- Navigation Bar:** SELECT, STATIC CONFIGURATION, **PFSM**, PROGRAM.
- Left Panel:**
 - Search icon (magnifying glass).
 - Diagram showing a self-loop on the **ACTIVE** state labeled "GPIO1 -> RISE".
 - Message: "Please validate PFSM settings".
 - GENERATE PROGRAM** button.
- Central Menu:**
 - GLOBAL SETTINGS
 - POWER SEQUENCE
 - TRIGGER SETTINGS** (highlighted)
 - TRIGGER PRIORITY LIST
 - PFSM VALIDATION
 - EXPORT
- Main Configuration Area:**
 - From State(s):** Add States (+) button, list containing **ACTIVE** (x), To State dropdown set to **ACTIVE**.
 - Add Trigger:** Reuse Trigger button, Add Device(s) list containing **TPS65940** (x) (+), Trigger Source and Trigger Type dropdowns, ADD button.
 - GPIO1 Configuration:**
 - GPIO1: **RISE** (with lightning bolt, edit, and delete icons)
 - DEVICE: **TPS65940**
 - SEQUENCE: **function1**

图 8-16. 自终止转换和触发关联

除了目标状态，还有用于选择序列类型的选项。序列类型 *Power Up* (上电)、*Power Down* (断电) 和 *Reset* (复位) 根据序列预先填充了 **ACTIVATE** 和/或 **DEACTIVATE** 命令。不建议在多 PMIC 应用中使用 *Power Down* (断电) 和 *Reset* (复位) 类型。有关如何正确使多个 PMIC 断电的讨论，请参阅 **DEACTIVATE** 说明 16。所有序列类型都预先填充了 **TRIG_MASK**。节 8.1.2.3.1 介绍了这些命令。

To Safe (安全) 序列类型是一种新方法，用于识别哪些序列会导致 **SAFE_RECOVERY** 状态。从功能上讲，此选择没有影响，但它确实会提醒 PFSM 验证例程在进入 **SAFE_RECOVERY** 之前检查所需的指令序列。

创建新序列后，序列名称将添加到序列列表中 (请参阅图 8-17)，并且可以编辑 (闪电)、更新 (铅笔) 或删除 (“X”) 该序列。仅允许更新序列的名称以及目标状态。使用编辑图标会打开一个新窗口，如图 8-18 所示。

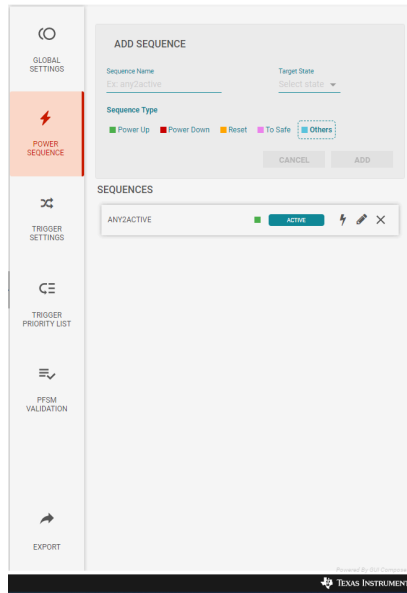


图 8-17. 如何添加序列

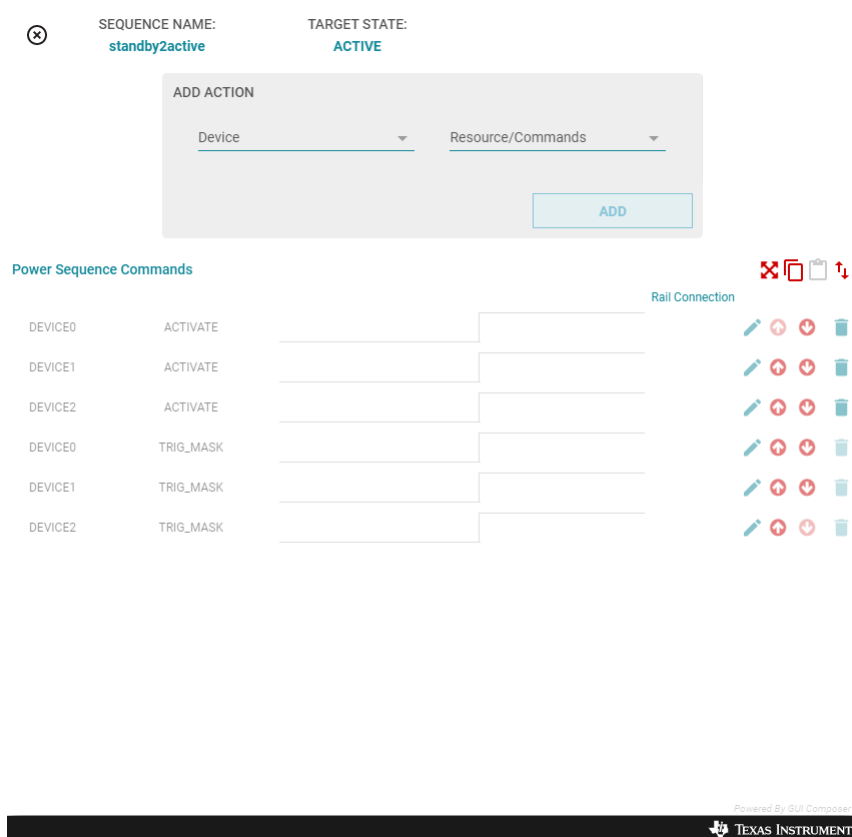


图 8-18. 电源序列命令窗口

在该窗口中，可以为每个 PMIC 添加命令，并显示每条命令之间的时序关系。提供的箭头可用于在序列内移动命令。

备注

ADD ACTION (添加操作) 或 *UPDATE ACTION* (更新操作) 窗口中的延迟是相对于前一个操作而言的。“Power Sequence Commands” (电源序列命令) 列表中显示的延迟是相对于序列开始而言的。

具有电源轨连接的命令和不具有电源轨连接的命令是有差别的。具有电源轨连接的命令通常表示器件和电压监视器的物理输出。电源轨连接将显示在电源序列图中，如节 8.1.2.3.3 所述。电源轨连接的名称是可编辑的，可以更新为与应用相关的更有意义的名称。

最初会提供每个器件的触发条件掩码。在添加其他命令时，它们将始终放置在触发条件掩码之上。可以移动触发条件掩码以使其在序列中具有不同的顺序，但它们通常是顺序中最后的命令。电源序列末尾的触发条件掩码是根据目标状态的可用触发条件自动生成的。可以编辑该掩码以阻止触发条件触发电源序列和状态更改。在编辑掩码之前，建议先定义触发条件，如节 8.1.2.4 所述。

8.1.2.3.1 电源序列资源和命令

本节列出了可用的不同命令和资源。本节内容参考了器件数据表中介绍的 PMIC 指令集。

资源和命令

并非所有命令都适用于所有器件。请参阅器件数据表。

1. BUCKx

BUCK 资源是汇编指令 `REG_WRITE_VCTRL_IMM` 和 `REG_WRITE_VOUT_IMM` 的抽象概念。在“Update action”（更新操作）窗口中选择“VCTRL”或“VOUT”选项卡将确定要应用的指令。每个选项卡中都包含每条指令的各种参数。

BUCKx 命令可用于每个可用的 BUCK。如果 BUCK 是多相的，则分组反映在资源名称中（例如 BUCK1_2_3_4），但“Parameter”窗口中仅显示主 BUCK 信息。在生成的程序中，主 BUCK 将是指令中唯一反映的压降。

2. BUCKx Monitor

BUCK Monitor 是 `REG_WRITE_VCTRL_IMM` 和 `REG_WRITE_VOUT_IMM` 指令的特殊子集。未用于该资源的参数是不可选的。“VIN”选项卡相同，但表示监视器电压设置。

3. VMONx

与 BUCKx 监视器类似，VMONx 是 LP876x-Q1 系列器件上的专用电压监视器。此资源仅在 GPIO 函数配置为 VMONx 函数时才可用。VMONx 是一个抽象的 `REG_WRITE_MASK_IMM`，用于寄存器地址 `0x2B` (`VCCA_VMON_CTRL`)。

4. LDOx

与“BUCK”和“BUCK Monitor”资源类似，这是汇编指令 `REG_WRITE_VCTRL_IMM` 和 `REG_WRITE_VOUT_IMM` 的抽象概念。在“Update action”（更新操作）窗口中选择“VCTRL”或“VOUT”选项卡将确定要应用的指令。每个选项卡中包含每条指令的参数。

5. nRSTOUT

nRSTOUT 命令写入或清除 `MISC_CTRL` 寄存器中的 nRSTOUT 位。该命令是应用于寄存器 `MISC_CTRL`（地址为 `0x81`）的 `REG_WRITE_MASK_IMM` 汇编指令的抽象概念。通过选择“unchanged”（未改变）、“high”（高）或“low”（低）来确定数据和掩码。选项 *unchanged*（未改变）不会产生任何影响，仅作为一个指令周期的延迟。

6. nRSTOUT_SOC

nRSTOUT_SOC 命令写入或清除 `MISC_CTRL` 寄存器中的 nRSTOUT_SOC 位。该命令是应用于寄存器 `MISC_CTRL`（地址为 `0x81`）的 `REG_WRITE_MASK_IMM` 汇编指令的抽象概念。通过选择“unchanged”（未改变）、“high”（高）或“low”（低）来确定数据和掩码。选项 *unchanged*（未改变）不会产生任何影响，仅作为一个指令周期的延迟。

备注

nRSTOUT 和 nRSTOUT_SOC 都可以在 `MISC_CTRL` 寄存器中找到。可以使用一个 `REG_WRITE_MASK_IMM` 命令更有效地设置和清除这两个位，而不是使用两个单独的命令 nRSTOUT 和 nRSTOUT_SOC。

7. WAIT

WAIT 命令在指令集中提供条件分支，类似于 *if* 或 *while* 语句。当提供超时，WAIT 条件实际上是一条 *if* 语句，如果条件为真，则继续执行下一条指令，如果条件为假，则跳转到目标。如果超时不为零，则 PMIC 将等

待，直到条件为真，然后执行下一条指令，或直到超时，然后跳转到目标。目标必须始终位于 WAIT 命令之后，因为 WAIT 指令的跳过计数始终为正。在版本 3.0.0 中，超时值必须可以通过电流 PFSM_DELAY_STEP 来实现。如果无法实现超时，GUI 将在 PFSM 验证中返回错误。使用 PFSM_DELAY_STEP 命令以更新 PFSM_DELAY_STEP。

备注

不建议对多个器件使用超时，因为如果在超时之前满足条件，则其他器件不知道会等待多长时间。

8. JUMP

JUMP 命令是 WAIT 命令的特殊实现，其超时时间为 0，并且条件始终为假。目标必须位于 JUMP 命令之后。

9. RESET_BUCKs

RESET_BUCKs 命令直接写入地址 0x87 的 BUCK_RESET_REG 寄存器中。在该命令中，复位是针对每个 BUCK 进行的，即使 BUCK 是多相的，也必须对每个 BUCK 进行配置。RESET_BUCKs 命令被转换为 REG_WRITE_MASK_IMM 命令以寻址 0x87，以清除或设置位 0 至 4 (表示 BUCKS 1-5)。

CAUTION

RESET_BUCK 命令将使降压稳压器停止开关。只应在断电序列中使用该命令。

10. GO_TO_LP_STANDBY

GO_TO_LP_STANDBY 命令直接将 1 写入 LDOINT_CTRL 寄存器中的 LDOINT 禁用位，地址为 0x21。LDOINT 是一个自行清除位。GO_TO_LP_STANDBY 命令被转换为 REG_WRITE_MASK_IMM 命令以寻址 0x21，数据值为 0x01，掩码为 0xFE。

11. SET_WD_LONGWINDOW

将 SET_WD_LONGWINDOW 命令直接写入 WD_LONGWIN_CFG 寄存器中的 WD_LONGWIN 字段，地址为 0x405。该字段用于对看门狗长窗口的持续时间进行编程。SET_WD_LONGWINDOW 命令被转换为 REG_WRITE_MASK_IMM 命令以寻址 0x405，数据值为 0x00 至 0xFF，掩码为 0x00。0x00 的值约为 100ms，而 0xFF 的值约为 12 分钟。

12. GO_TO_LONGWIN

GO_TO_LONGWIN 命令直接将 1 写入 WD_MODE_REG 寄存器中的 WD_RETURN_LONGWIN 位，地址为 0x406。此命令将使看门狗在当前看门狗序列完成后返回长窗口。GO_TO_LONGWIN 命令被转换为 REG_WRITE_MASK_IMM 命令以寻址 0x406，数据值为 0x01，掩码为 0xFE。

13. FIRST_STARTUP_DONE

FIRST_STARTUP_DONE 命令直接将 1 写入 RTC_CTRL_2 寄存器中的 FIRST_STARTUP_DONE 位，地址为 0xC3。FIRST_STARTUP_DONE 命令被转换为 REG_WRITE_MASK_IMM 命令以寻址 0xC3，数据值为 0x80，掩码为 0x7F。

14. INCREASE_RECOVERY_COUNT

INCREASE_RECOVERY_COUNT 命令直接将 1 写入 RECOV_CNT_PFSM_INCR 中的 INCREASE_RECOVERY_COUNT 位，地址为 0xA5。该位会自行清除，因此每条命令都会使恢复计数器递增。INCREASE_RECOVERY_COUNT 命令被转换为 REG_WRITE_MASK_IMM 指令以寻址 0xA5，数据值为 0x01，掩码为 0xFE。

15. ACTIVATE

ACTIVATE 命令是与上电序列相关的若干命令的组合。这些命令包括：

- a. 应用于 ENABLE_DRV_STAT 的 REG_WRITE_MASK_IMM，用于清除 SPMI_LPM_EN。
- b. 应用于 VCCA_VMON_CTRL 的 REG_WRITE_MASK_IMM，用于清除或设置 VCCA_VMON_EN，具体取决于从“ACTIVATE”命令窗口中选择的值。
- c. 应用于 MISC_CTRL 的 REG_WRITE_MASK_IMM，用于清除或设置 AMUXOUT_EN/REFOUT_EN 和 CLKMEN_EN，具体取决于从“ACTIVATE”命令窗口中选择的值。该指令包含 LPM_EN 清除。

16. DEACTIVATE

DEACTIVATE 命令是与断电序列相关的几个命令的组合。不建议在多 PMIC 应用中使用此命令。这些命令包括：

- a. 应用于 ENABLE_DRV_STAT 的 REG_WRITE_MASK_IMM，用于设置 SPMI_LPM_EN。
- b. 应用于 VCCA_VMON_CTRL 的 REG_WRITE_MASK_IMM，用于清除或设置 VCCA_VMON_EN，具体取决于从“DEACTIVATE”命令窗口中选择的值。
- c. 应用于 MISC_CTRL 的 REG_WRITE_MASK_IMM，用于清除或设置 AMUXOUT_EN/REFOUT_EN，具体取决于从“ACTIVATE”命令窗口中选择的值。该指令包含 LPM_EN 设置。

不建议将 Deactivate 命令用于多 PMIC 应用。SPMI_LPM_EN、VCCA_VMON_EN、AMUXOUT_EN/REFOUT_EN、CLKMON_EN 和 LPM_EN 等参数的控制由 ACTIVATE 和 DEACTIVATE 命令抽象。

SPMI_LPM_EN 将 SPMI 设置为停止 SPMI WD（总线检测信号）的低功耗模式。在多 PMIC 应用中，必须同时处理 SPMI_LPM_EN 以防止 SPMI WD 故障。因此，为了缓解时钟变化，必须在每个 PMIC 序列中尽早设置和清除 SPMI_LPM_EN。

LPM_EN 参数将 PMIC 置于低功耗模式。预期的用例是 PFSM 在进入低功耗状态时设置 LPM_EN。最终目标是禁用数字振荡器以减少功耗。有关设置 LPM_EN 时禁用的函数，请参阅数据表。

表 8-3. SPMI_LPM_EN、FORCE_EN_DRV_LOW 和 LPM_EN 示例汇编指令

TO_SAFE	TO_STANDBY	TO_RETENTION	TO_ACTIVE
SPMI_LPM_EN=1, FORCE_EN_DRV_LOW=1	SPMI_LPM_EN=1, FORCE_EN_DRV_LOW=1	nRSTOUT = 0, nRSTOUT_SOC=0	LPM_EN=0, AMUXOUT_EN = 1, CLKMON_EN = 1
nRSTOUT = 0, nRSTOUT_SOC=0	nRSTOUT = 0, nRSTOUT_SOC=0	SPMI_LPM_EN=1, FORCE_EN_DRV_LOW =1	SPMI_LPM_EN = 0
...
LPM_EN=1, AMUXOUT_EN = 0, CLKMON_EN = 0	LPM_EN=1, AMUXOUT_EN = 0, CLKMON_EN = 0	LPM_EN=1, AMUXOUT_EN = 0, CLKMON_EN = 0	FORCE_EN_DRV_LOW =0 nRSTOUT = 1, nRSTOUT_SOC=1

备注

EN_DRV、nRSTOUT 和 nRSTOUT_SOC 引脚可能不会在应用中使用，但出于完整性考虑在此处显示。在这些示例中，VCCA_VMON_EN 保持为“1”。

多 PMIC 解决方案的关键要求是 SPMI_LPM_EN 应首先出现，而 LPM_EN 应在序列最后出现。因为 DEACTIVATE 命令将 LPM_EN 和 SPMI_LPM_EN 封装在一组执行中，因此，在开头或结束时放置 DEACTIVATE 将违反其中一个要求。

17. ENDRV

ENDRV 命令是直接写入 ENABLE_DRV_STAT 寄存器中的 FORCE_EN_DRV_LOW 位，地址 0x82。ENDRV 命令被转换为 REG_WRITE_MASK_IMM 命令，地址为 0x82，数据值为 0x08 或 0x00，掩码为 0xF7。

18. DELAY_IMM

DELAY_IMM 命令是 DELAY_IMM 指令的直接表示。该命令中指定的延迟应用于所有器件。在版本 3.0.0 中，如果现有 PFSM_STEP_SIZE 无法实现 DELAY_IMM 的大小，GUI 将不再自动调整 PFSM_STEP_SIZE。用户负责管理所需延迟的 PFSM_STEP_SIZE。这也适用于所有具有定时元件的指令。如果无法实现延迟，会在 PFSM 验证中提供错误。

备注

编译器将对 DELAY_IMM 指令进行优化并与以下指令相结合（如果该指令包含计时参数）。

19. REG_WRITE_MASK_IMM

REG_WRITE_MASK_IMM 命令是 REG_WRITE_MASK_IMM 指令的直接表示。REG_WRITE_MASK_IMM 命令包含一个掩码，用于在不影响寄存器中的其他位的情况下写入或清除特定的位。

20. TRIG_MASK

TRIG_MASK 命令是 TRIG_MASK 指令的直接表示。触发条件掩码将决定要启用和禁用的中断。如果使用自动触发条件，则会根据“TARGET STATE”中的触发条件设置来设置触发条件。

21. REG_WRITE_IMM

REG_WRITE_IMM 命令是 REG_WRITE_IMM 指令的直接表示。REG_WRITE_IMM 命令覆盖指定的寄存器中的所有位。

22. REG_WRITE_MASK_SREG

REG_WRITE_MASK_SREG 命令是 REG_WRITE_MASK_SREG 指令的直接表示。

REG_WRITE_MASK_SREG 命令包含一个掩码，用于在不影响寄存器中的其他位的情况下写入或清除特定位。数据来自指定的擦除寄存器。

23. SREG_READ_REG

SREG_READ_REG 命令是 SREG_READ_REG 指令的直接表示。该命令将寄存器的内容复制到指定的擦除寄存器。

24. SREG_WRITE_IMM

SREG_WRITE_IMM 命令是 SREG_WRITE_IMM 指令的直接表示。

25. DELAY_SREG

DELAY_SREG 命令是 DELAY_SREG 指令的直接表示。该延迟不同于资源和命令中的其他延迟。该延迟仅应用于指定的器件。

26. PFSM_DELAY_STEP

PFSM_DELAY_STEP 是写入 PFSM_DELAY_STEP 寄存器的 SET_DELAY 指令的直接表示。

27. END

END 命令是 END 指令的直接表示。这可以用于终止在 PFSM 序列中使用 JUMP 和 WAIT 命令创建分支。

8.1.2.3.2 子序列

子序列是电源序列中的命令组，与 JUMP 或 WAIT 指令相关联。从功能上而言，子序列只是要跳转到的 JUMP 或 WAIT 语句的目标标签。从图形上看，子序列可用于对命令进行分组，然后可以在电源序列内移动整个组。在图 8-19 中，WAIT 指令用于测试 GPIO1，如果 GPIO1 为高电平，则发生超时，执行跳转到 SKIPBUCK5 标签并继续进行。具体而言，如果 GPIO1 为低电平，则启用 BUCK5，如果 GPIO1 为高电平，则不启用该稳压器。图 8-20 显示子序列中没有指令。放置在子序列之内或之后的指令在时间上是等效的。同样，将指令放置在子序列中的主要好处是对命令进行逻辑分组，然后在需要时将它们作为一个组进行移动。

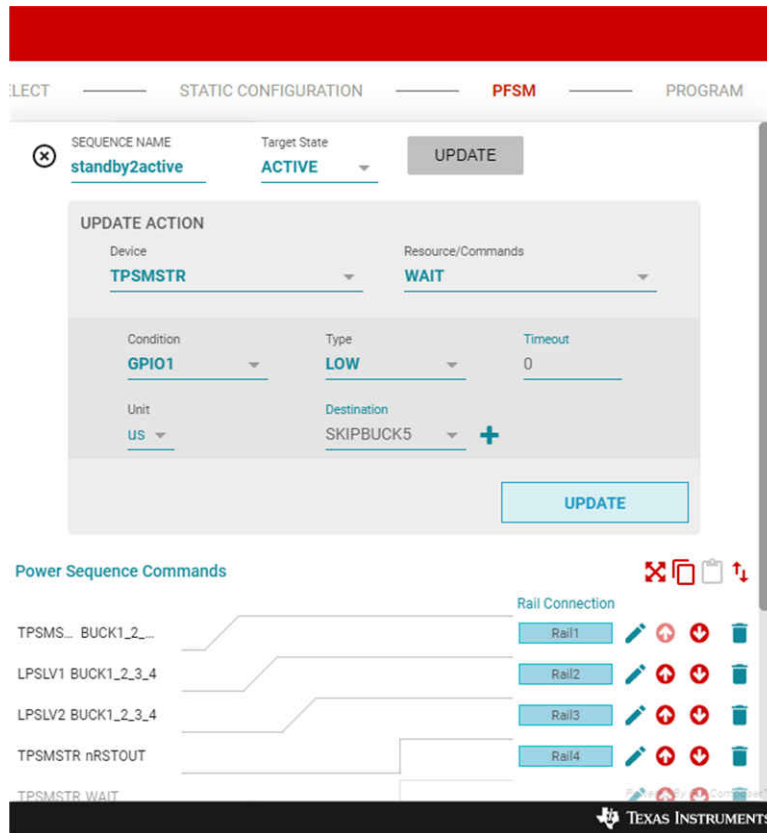


图 8-19. WAIT 命令操作

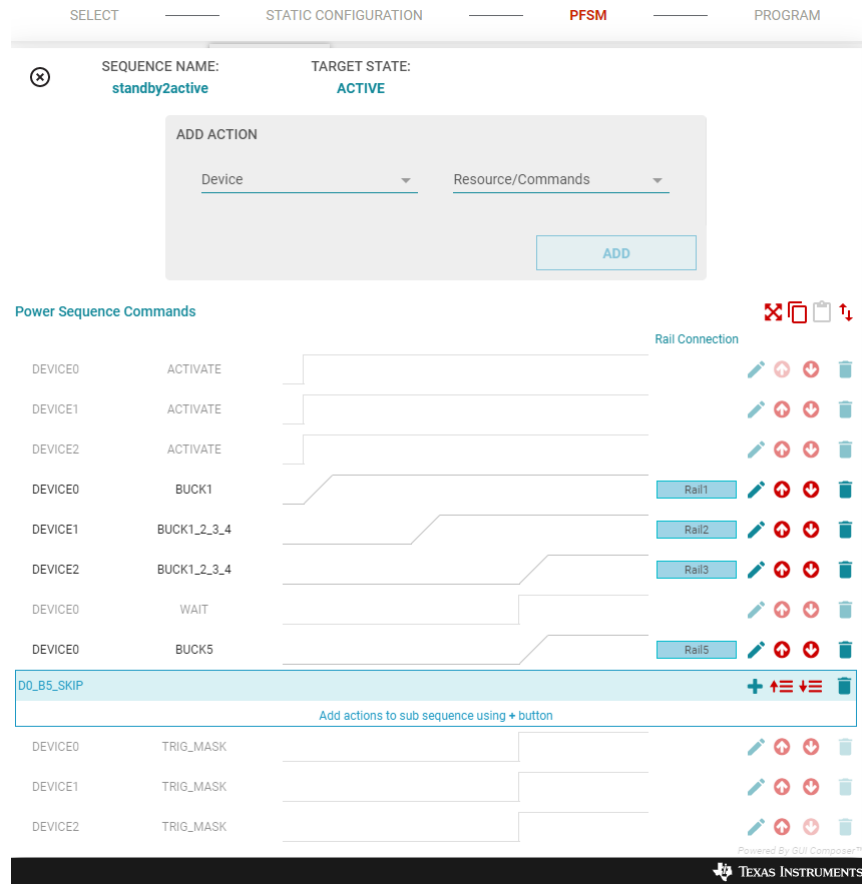


图 8-20. 空子序列

备注

WAIT 和 JUMP 语句只能在命令序列中向前跳转。目标永远不能放置在 JUMP 或 WAIT 语句之前。

8.1.2.3.3 电源序列编辑工具

除了每条命令的四个编辑工具（更新、上移、下移、删除）之外，还有四个工具可用于全面地编辑电源序列。按照从左到右的顺序，这些工具分别为导出电源序列、复制序列、粘贴序列和反向序列（请参阅图 8-21）。复制、粘贴和反向序列工具创建了一种便捷的方式来复制电源序列（例如上电序列）并将其粘贴到新序列中，然后进行反向以创建断电序列。

备注

反向操作不适用于 TRIG_MASKS，因为它们始终位于序列的末尾。反向只涉及命令的顺序，而不是时序关系或极性；启用不会变为禁用。

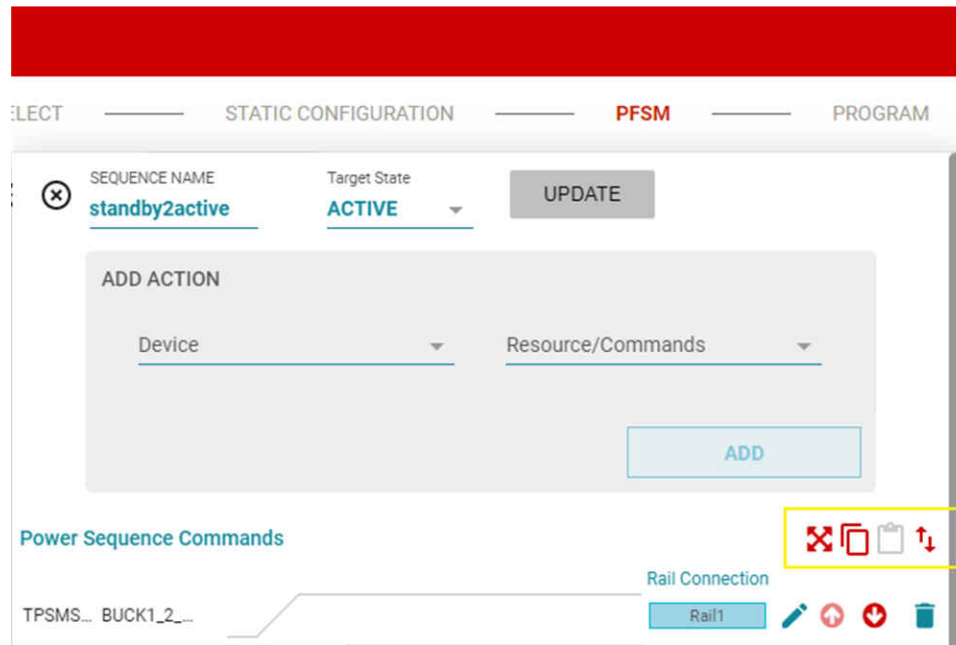


图 8-21. 电源序列工具

导出电源序列提供了一个更完整的并可以导出的图形视图。在该视图中，用户能够为条件 WAIT 命令提供输入，并为不同的条件生成时序图。如图 8-22 所示，既可以选择超时为真的条件，也可以相应地更新时序图。

备注

德州仪器 (TI) 建议导出电源序列以确认时序是否符合系统要求。

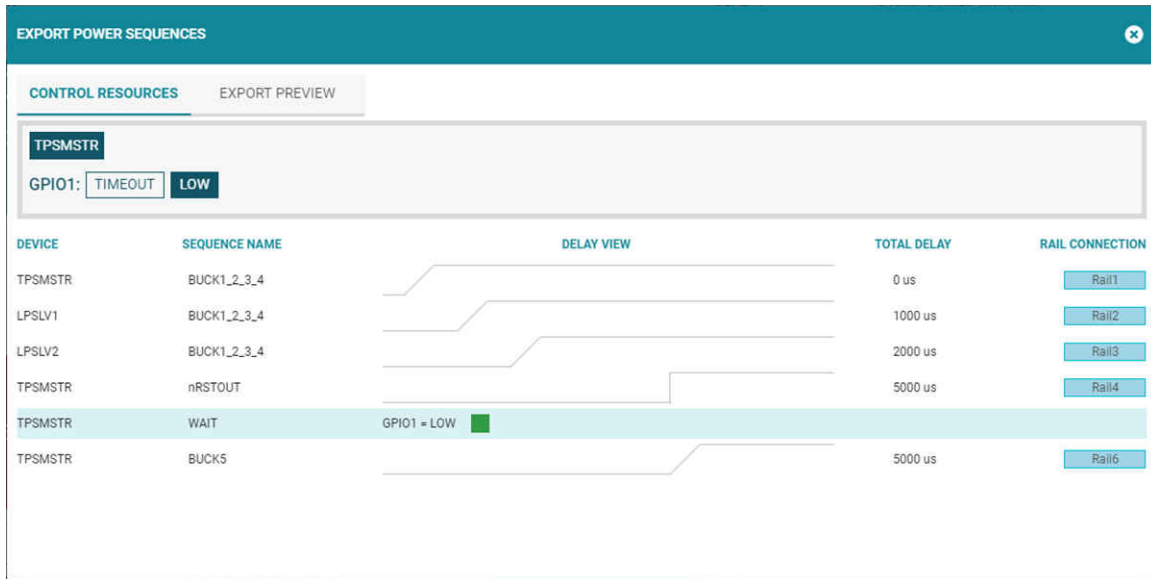


图 8-22. 具有可变条件的导出序列

8.1.2.4 触发条件设置

最大的可用触发条件数为 28。有关保留触发条件的使用方法，请参阅器件规格。GUI 将确保不超过最大触发条件数，并管理触发条件在多 PMIC 应用中的跨器件使用。

触发条件设置确定哪些触发条件会将 PMIC 操作从一种状态转换至另一种状态。在 GUI 情境中，状态之间的箭头（或循环回同一状态的箭头）必须具有至少一个触发条件定义。下面通过一个示例来概述配置触发条件设置的步

骤。在该示例中，任何器件上的 **IMMEDIATE_SHUTDOWN** (立即_停止) 位于高位都将触发所有器件执行断电序列，然后切换到 **SAFE** (安全) 状态，这进而又会自动将所有器件 (无电源序列) 转换到硬件状态 **SAFE_RECOVERY** (安全_恢复)，以重置器件。以下步骤说明了如何设置该示例。

1. 通过点击图中的转换箭头，选择 **STANDBY** (待机) 和 **SAFE** (安全) 状态之间的转换。

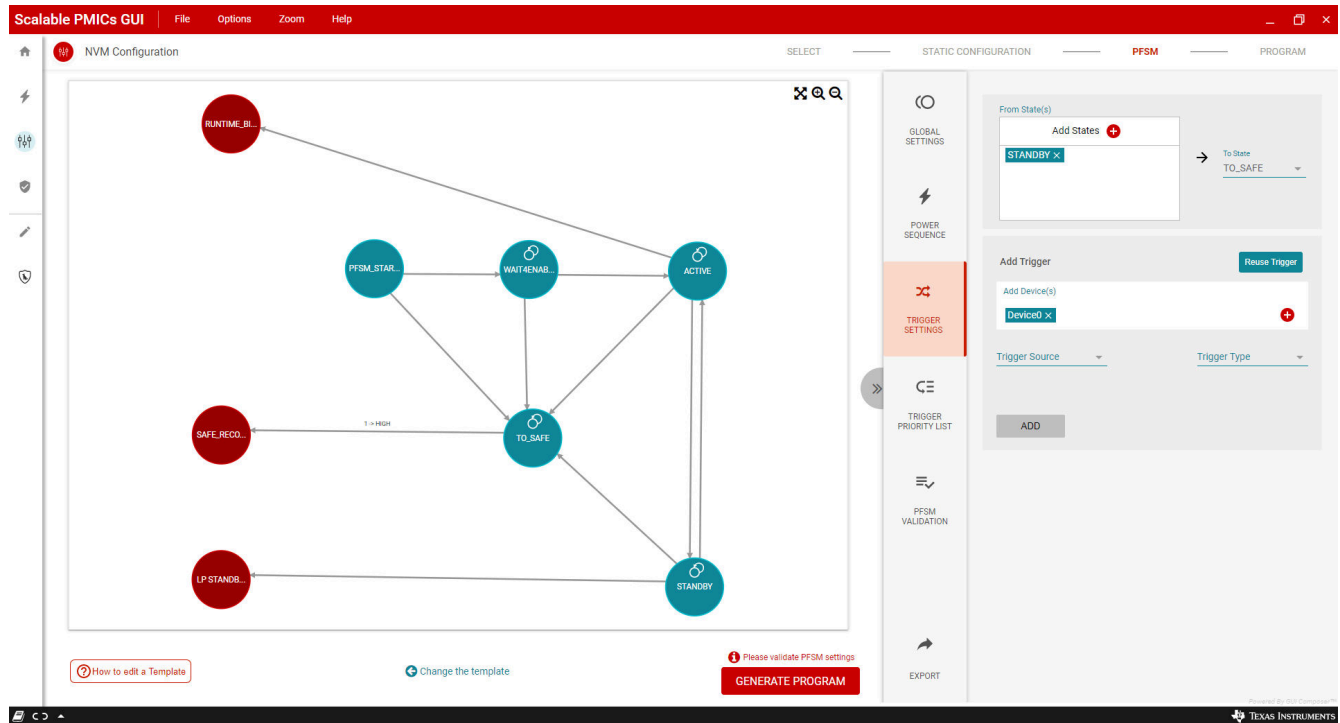


图 8-23. 触发条件：STANDBY (待机) 转 SAFE (安全)

2. 在 *From States* 中，添加 **ACTIVE**、**PFSM_START** 和 **WAIT4ENABLE**。⁴
3. 在 “Add Devices” (添加器件) 中选择 *Any* (任意)，以便所有器件都显示在窗口中⁵。
4. 在 *Trigger Source* (触发条件来源) 下拉菜单中选择 **IMMEDIATE_SHUTDOWN** (立即_停止)。
5. 在 *Trigger Type* (触发条件类型) 下拉菜单中选择 **HIGH** (高)。
6. 选中 *Immediate* (立即) 复选框。这意味着触发条件可以立即发生，而不会等待当前序列完成。

⁴ “Reuse Trigger” (重用触发条件) 按钮是一个替代选项。步骤 2 省略。步骤 7 之后，选择其他到 **安全 (SAFE)** 状态的转换。此时选择重复使用触发器按钮，然后选择用于 **待机 (STANDBY)** 到 **安全 (SAFE)** 的同一触发条件。

⁵ 作为一个示例，德州仪器 (TI) 建议所有器件中都存在 **IMMEDIATE_SHUTDOWN** (立即_停止) 触发条件。对于其他触发条件，触发条件仅存在于一个器件中是可以接受的。

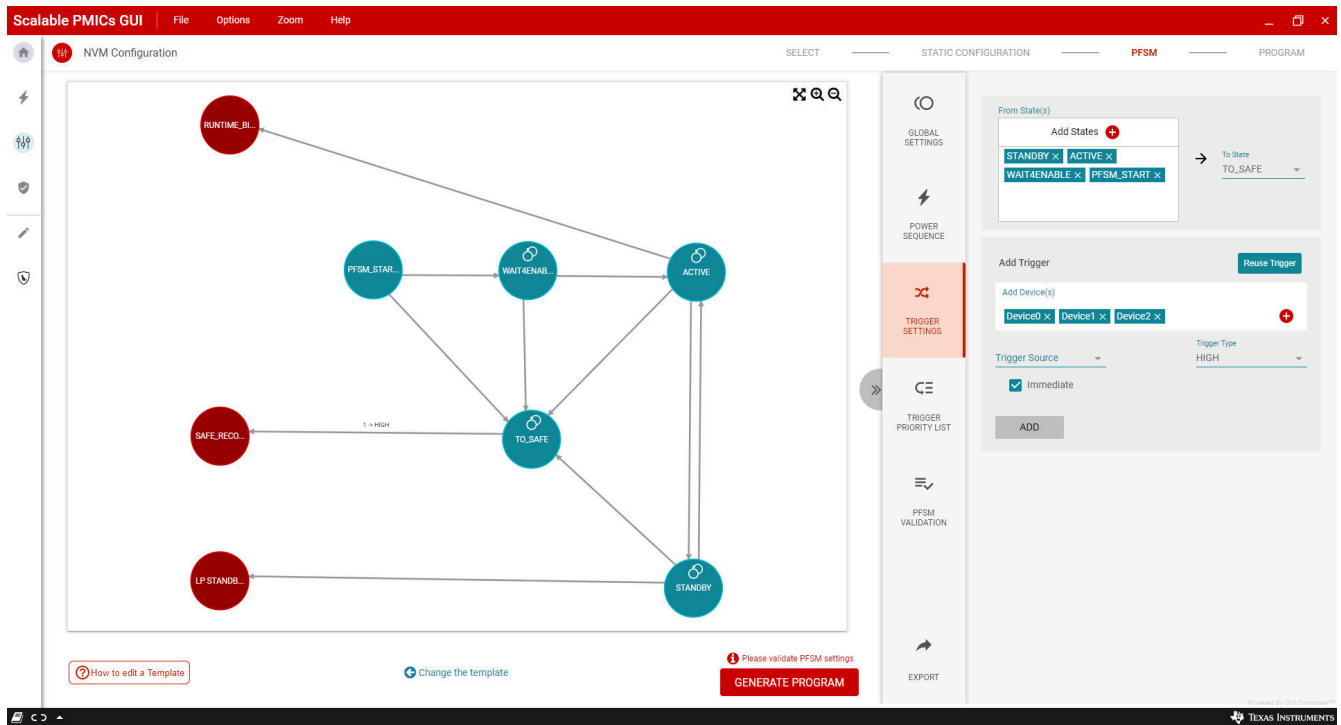


图 8-24. 触发条件：STANDBY 转 SAFE (续)

7. 点击 **ADD** (添加)。
8. 现在点击 **SAFE** (安全) 和 **SAFE_RECOVERY** (安全_恢复) 之间的转换箭头。无需执行任何操作，因为会自动填充此触发条件。

备注

由于 **SAFE_RECOVERY** (安全_恢复) 是硬件状态，因此触发条件不会具有关联的电源序列。这是由触发条件描述中的“EXT”属性指示的。同样，**RUNTIME_BIST** 也将具有“EXT”属性。

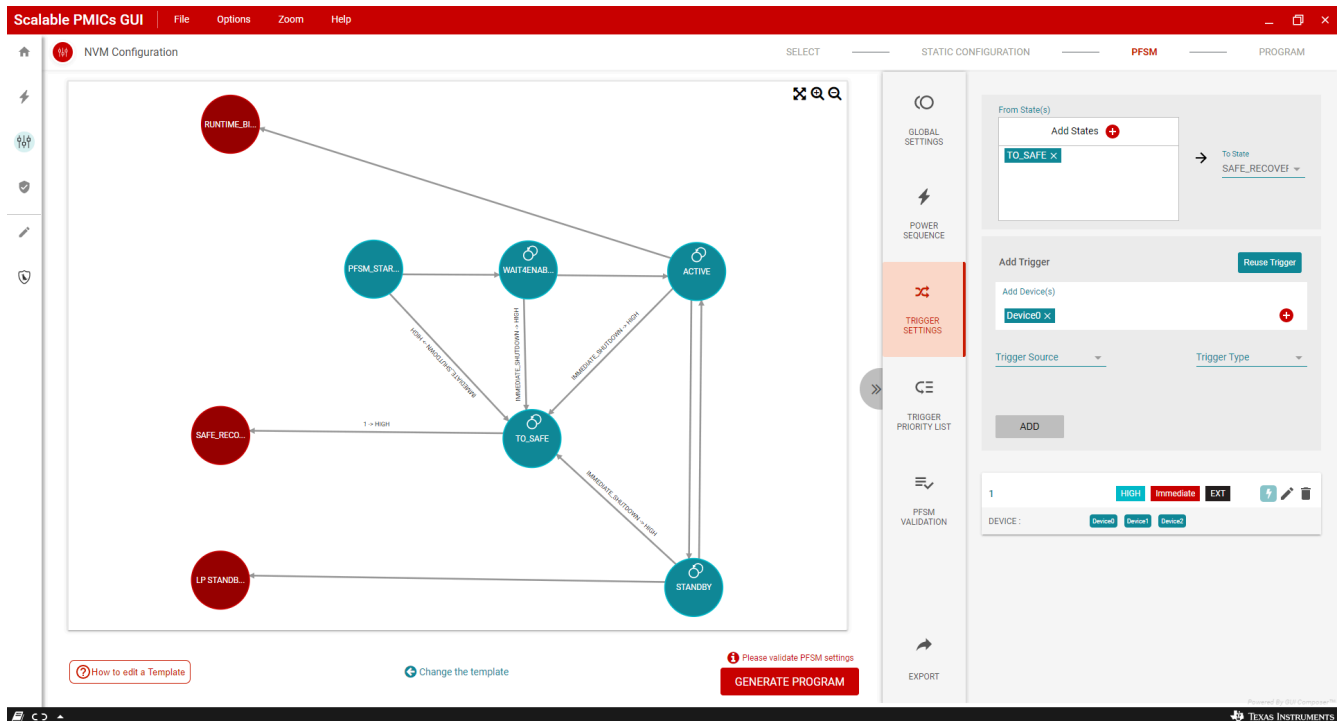


图 8-25. 触发条件：SAFE (安全) 转 SAFE_RECOVERY (安全_恢复)

“TRIGGER SETTINGS”（触发条件设置）底部汇总了与某个转换相关联的触发条件（图 8-25 中的突出显示部分）。提供了一个滚动条来查看窗格的底部。最后一步是将电源序列与转换相关联。由于 **SAFE_RECOVERY**（安全_恢复）是一种硬件状态，因此设置了 **EXT** 标志，并且在转换到 **SAFE_RECOVERY**（安全_恢复）时不需要任何顺序。不过，到 **SAFE**（安全）状态的转换确实需要一个序列。以下内容说明了如何在触发条件和序列之间建立联系。

1. 点击 **STANDBY**（待机）和 **SAFE**（安全）（或 **ACTIVE**（活动）和 **SAFE**（安全））之间的转换箭头。
2. 点击闪电图标。窗口显示具有相同目标或目标状态的所有序列。在本例中仅有一个序列，即 **any2safe**。

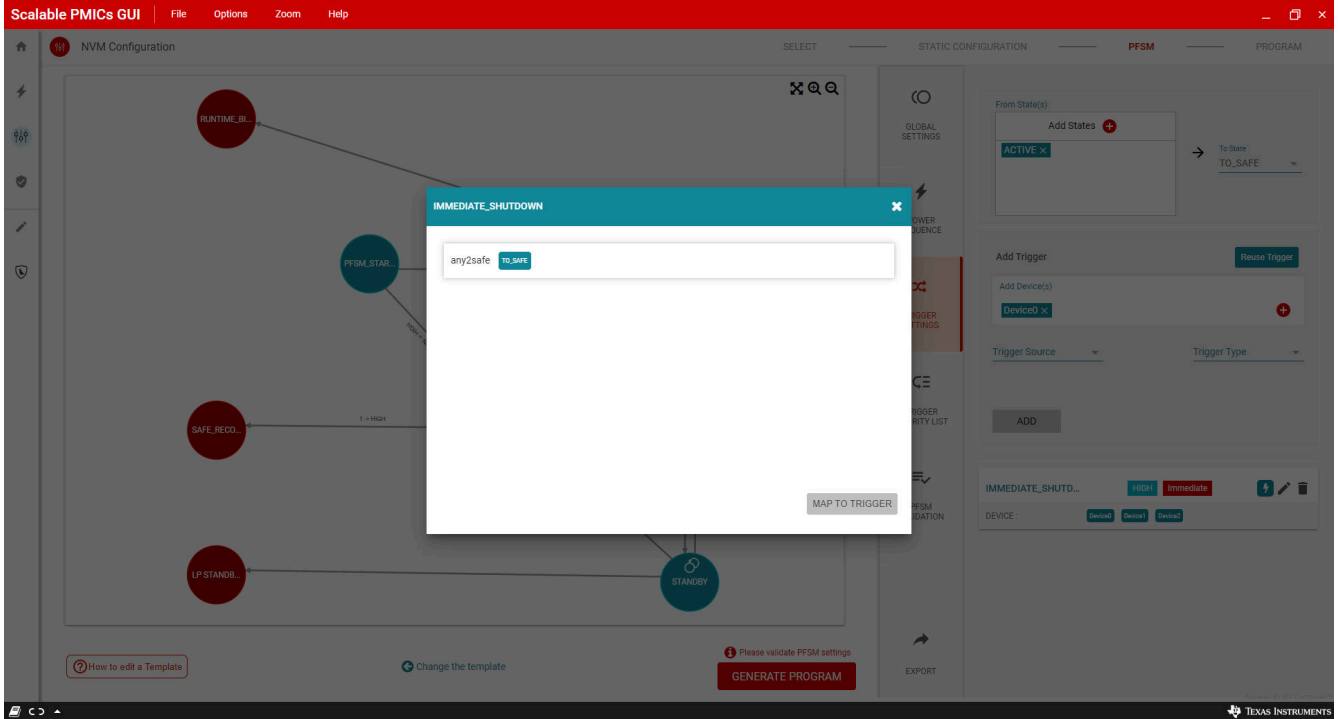


图 8-26. 将序列映射到触发条件

3. 选择该序列，然后点击 **MAP TO TRIGGER**（映射到触发条件）。针对该转换的触发条件设置现在已完成。

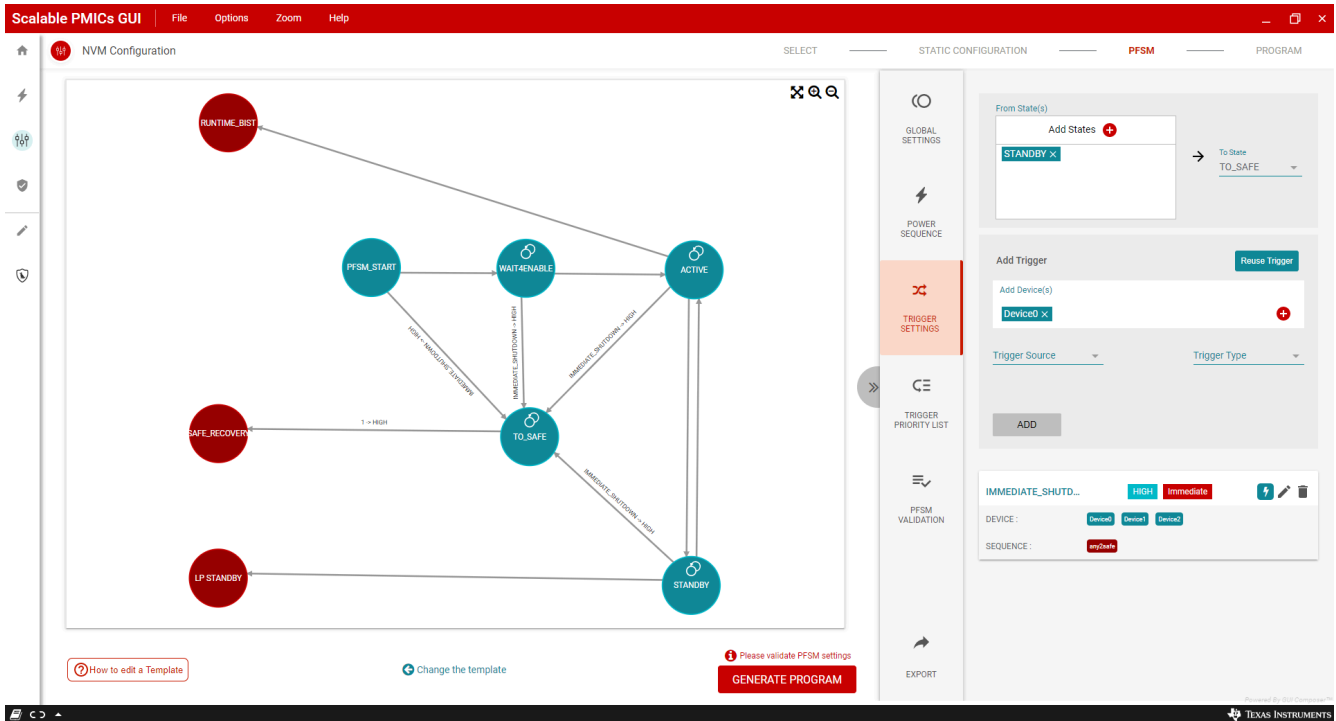


图 8-27. 已完成的触发条件设置

为所有转换分配触发条件并且所有触发条件都与电源序列相关联后，电源序列中的默认触发条件掩码将被更新。请注意到目前为止讨论的各种组件和关系：

1. 状态

状态是硬件状态或任务状态。

2. 转换

转换具有源状态和目标状态。转换可以有多个触发条件，但至少需要一个触发条件。到同一目标状态的转换可以共享同一个触发条件。

3. 触发条件

器件规格中定义了可用的触发条件。每个触发条件只能有一个与之关联的电源序列。多个触发条件可以共享同一个电源序列。在转换目标状态为硬件状态的特殊情况下，触发类型为 EXT，并且没有电源序列。

4. 电源序列

电源序列是根据目标状态定义的，因此可能与多个触发条件或转换相关联。触发条件掩码位于电源序列内。自动触发条件掩码来自目标状态的所有转换以及这些转换中定义的触发条件进行定义。手动触发条件掩码可用于创建自定义触发条件掩码。在 TRIG_MASK 命令中有一个用于选择自动或手动触发条件掩码的选项。

8.1.2.5 触发条件优先级列表

最初根据数据表中定义的每个触发条件的 TRIG_SEL 值来确定触发条件的优先级。较低的值具有较高的优先级。务必要确认“TRIGGER PRIORITY LIST”（触发条件优先级列表）中的优先级是否与应用所需的优先级相匹配。图 8-28 显示了上一节所示简单示例的优先级列表。点击列表中的箭头可以按优先级上下移动触发条件。

The screenshot displays the Scalable PMICs GUI interface. The main area shows a state transition diagram with the following states and transitions:

- LP STANDBY** (red circle) and **RUNTIME_BI...** (red circle) are connected to **STANDBY** (teal circle).
- STANDBY** and **ACTIVE** (teal circle) are connected by bidirectional arrows.
- STANDBY** transitions to **SAFE** (teal circle) on the event **IMMEDIATE_SHUTDOWN -> HIGH**.
- ACTIVE** transitions to **SAFE** on the event **IMMEDIATE_SHUTDOWN -> HIGH**.
- SAFE** transitions to **SAFE_REC...** (red circle) on the event **1 -> HIGH**.

On the right side, the **Trigger Priority List** configuration panel is visible, showing two entries:

Priority	Event	Action
0	IMMEDIATE_SHUTDOWN_HIGH	STANDBY -> SAFE, ACTIVE -> SAFE
1	1_HIGH	SAFE -> SAFE_RECOVERY

At the bottom of the diagram area, there is a red button labeled **GENERATE PROGRAM** and a warning message: **Please validate PFSM settings**.

图 8-28. 触发条件优先级列表

8.1.2.6 PFSM 验证

“TRIGGER PRIORITY LIST”（触发条件优先级列表）是构建 PFSM 配置的最后一个组件。配置完成后，PFSM 验证将对 PFSM 内容进行检查和验证。在“PFSM Validation”（PFSM 验证）视图中，点击 **Validate PFSM**（验证 PFSM）按钮。

在“PFSM”视图中，将显示结果列表。任何错误或警告都将附有说明和建议。错误和警告不会阻止程序生成，但表明应用中的器件性能存在潜在风险。建议更正所有错误，否则会妨碍正确编译或文件生成。更正所有错误后，点击 **GENERATE PROGRAM**（生成程序）按钮转到“PROGRAM”（程序）透视图。

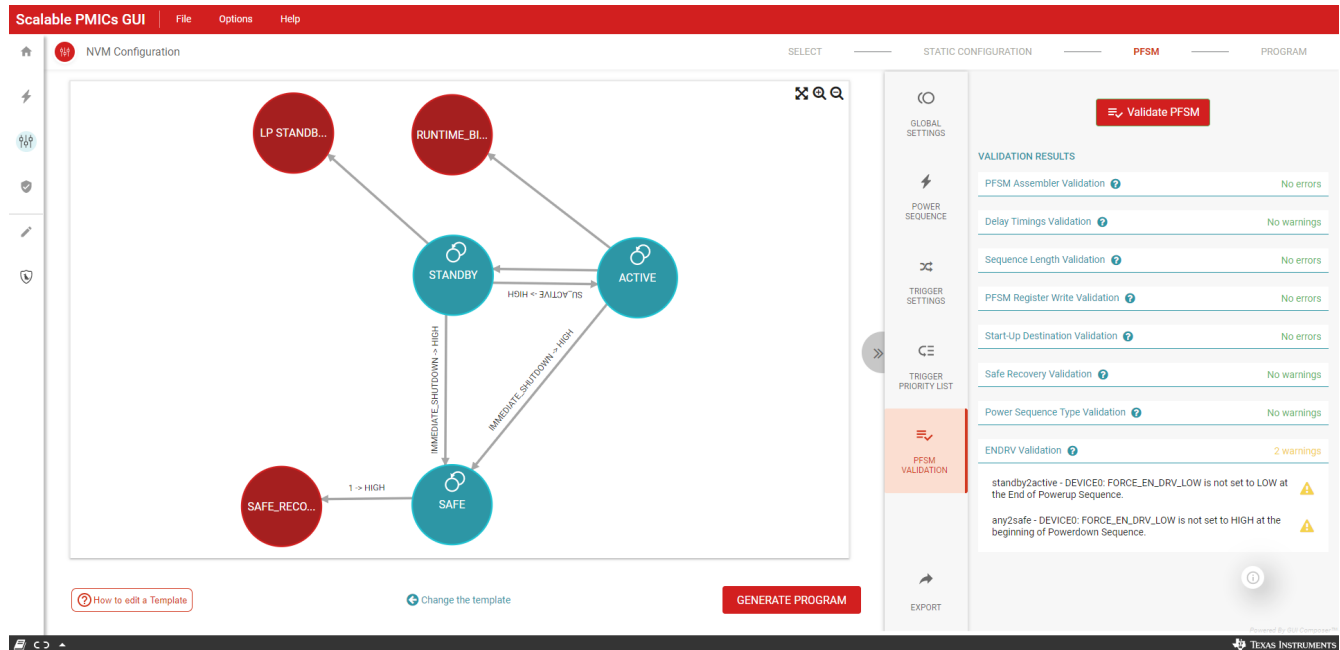


图 8-29. PFSM 验证结果

8.2 程序

备注

在编程之前最好先保存配置，请参阅图 8-9。

程序页面的 **Generated Program** (生成的程序) 选项卡中会显示生成程序的结果，如图 8-30 的左侧所示。这是一个文本格式文件，可以通过滚动条来查看整个内容。右侧是编程的控制机制。

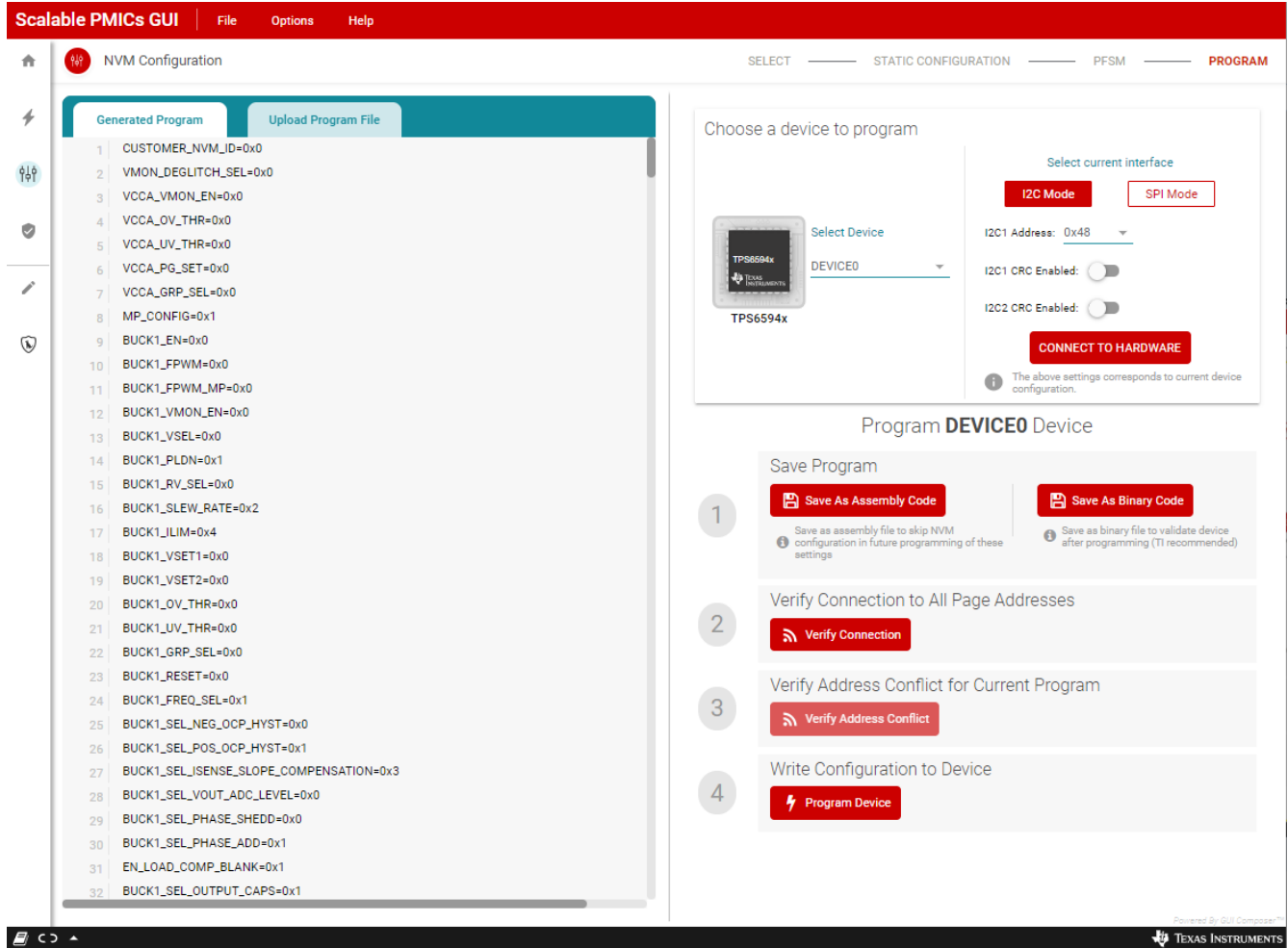


图 8-30. NVM 编程

如果配置了多个 PMIC，则会提供 **Select Device** (选择器件) 下拉菜单。选择器件后，相关程序会显示在“Generated Program” (生成的程序) 选项卡中。这也会确定将保存和编写哪个程序。

Select current interface (选择当前接口) 将确定使用哪个物理地址来验证连接和程序。

提供了两个用于保存程序的选项，即 **Save as Assembly Code** (另存为汇编代码) 和 **Save as Binary Code** (另存为二进制代码)。**Save as Assembly Code** (另存为汇编代码) 的格式与 **Generate Program** (生成程序) 选项卡中显示的格式相同。**Save as Binary Code** (另存为二进制代码) 格式是寄存器地址和这些地址处的十六进制值。这两种格式都可以上传到程序页面并编程到选定的器件中，而无需使用配置步骤，如节 8.2.1 所述。二进制格式也可在“NVM Validation” (NVM 验证) 页面中用于验证器件的 NVM 内容。验证物理连接之后，可以对器件进行编程。

备注

德州仪器 (TI) 建议保存全部三种类型的文件：配置、汇编和二进制。

WARNING

德州仪器 (TI) 不支持手动编辑的汇编或二进制文件。

8.2.1 对现有 NVM 配置进行编程

“NVM Configuration”（NVM 配置）页面也可用于使用现有 NVM 配置以汇编或二进制格式对器件进行编程。该过程从页面的开头开始，然后通过点击器件图标来选择要编程的器件类型，如图 8-31 所示。选择器件后，点击 *Skip to Programming*（跳至编程）按钮直接进入编程页面。

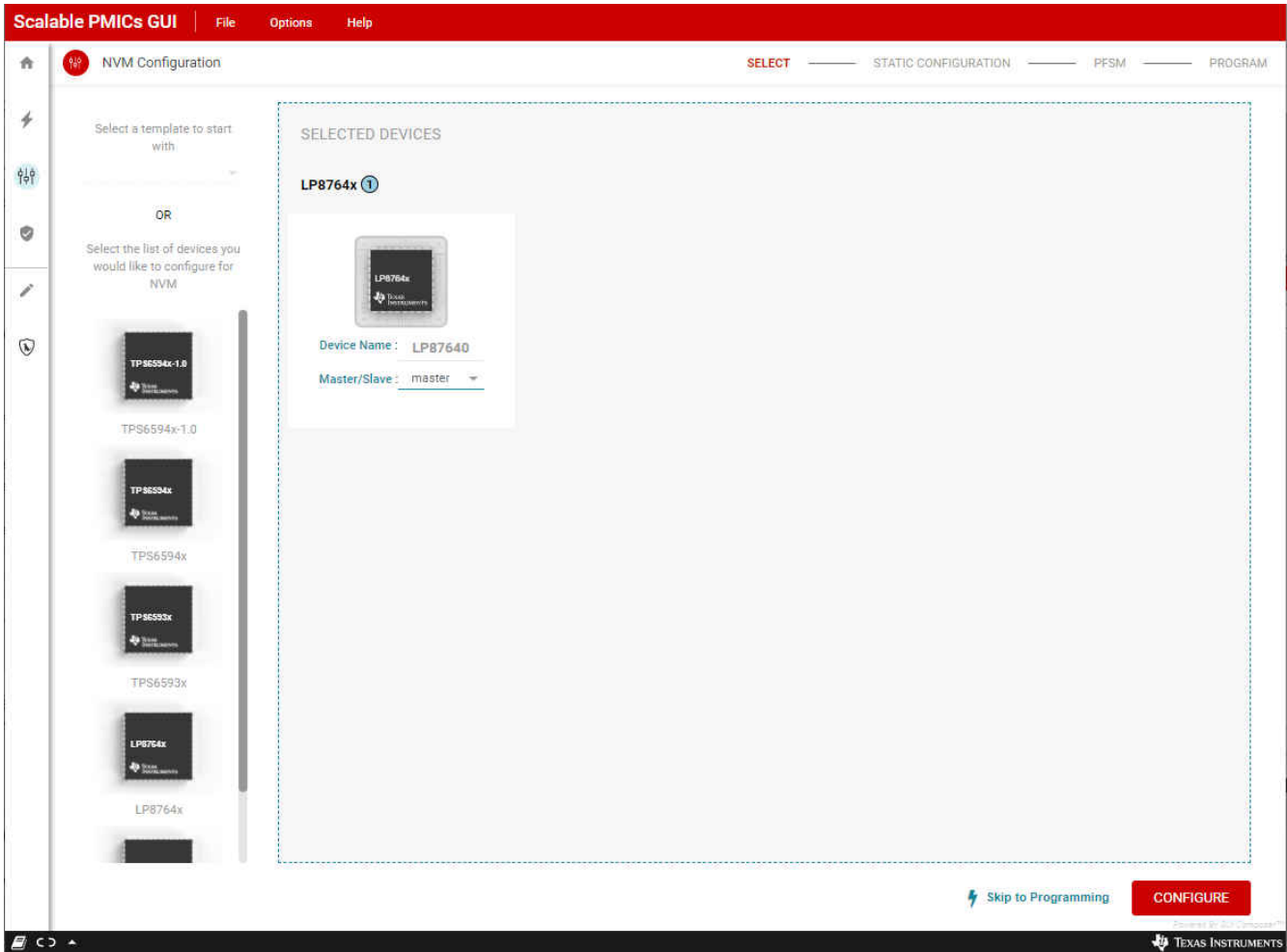


图 8-31. 跳至使用现有 NVM 配置进行编程

编程透视图与节 8.2 中所述的内容略有不同。如图 8-32 所示，仅 *Uploaded Program File*（上传的编程文件）选项卡可用，保存选项被禁用，*Select Device*（选择器件）反映了开始时的器件选择。接口选择应与正在对其进行编程的器件相匹配。如果接口设置不正确，则屏幕底部的连接指示器将反映 *Hardware not Connected*（硬件未连接）。将显示 *Failed to connect*（连接失败），任何 *Verify Connection*（验证连接）尝试都将失败。连接正确的⁶ 器件并选择要编程的文件后，点击 *Verify Connection*（验证连接），然后点击 *Program Device*（编程器件）。

⁶ 当多个器件连接到 AEVM 时，确保对于 I²C，地址是要编程的器件的地址。对于 SPI，确保片选信号连接到要编程的器件。

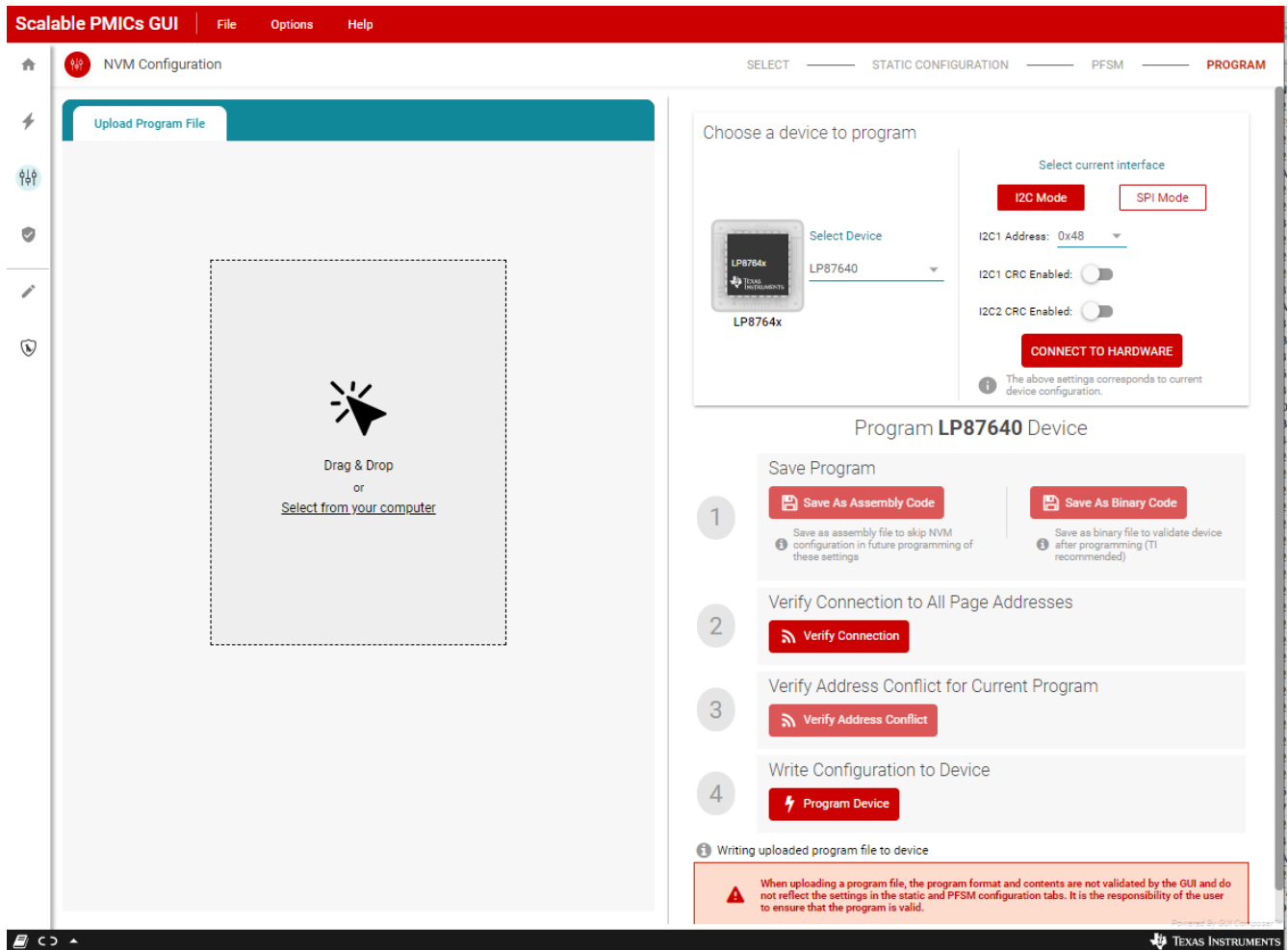


图 8-32. 使用现有配置进行编程

8.2.2 NVM 配置特殊用例：更改通信接口

该 GUI 工具使用识别的接口来写入寄存器设置，然后使用批量编程方法将寄存器设置传输到 NVM。如果 NVM 映像使用不同的接口，则被编程的器件应支持这两种接口，这一点很重要。当 GUI 写入寄存器设置以更改接口（SPI 改为 I²C 或 I²C 改为 SPI）时，GUI 将暂停并显示提示（请参阅图 8-33），从而更改硬件配置以启用新接口。该消息专门针对 EVM，但可以推广到任何进行了更改（从 I²C 转换为 SPI 或从 SPI 转换为 I²C）的硬件配置。

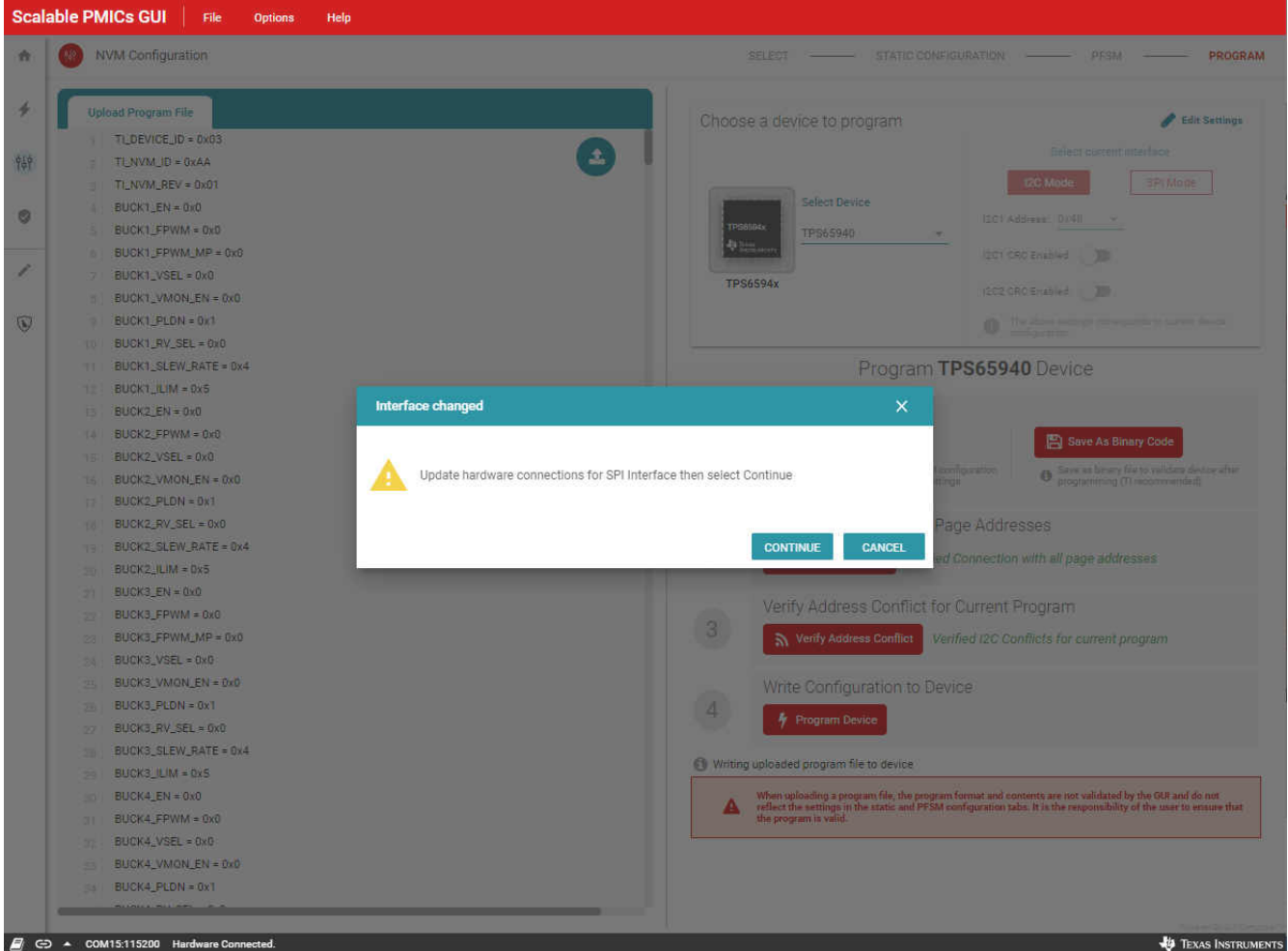


图 8-33. 编程过程中的接口更改

8.2.3 NVM 编程期间的锁定选项

可以将 PMIC NVM 永久锁定，以防止对 NVM 配置进行任何更改。虽然在开发过程中不建议这样做，但 GUI 支持此功能。在编程过程中，会显示一个对话框窗口（请参阅图 8-34），其中包含用于锁定 NVM 的选项。完成该操作之后，就无法更改 PMIC NVM 了。

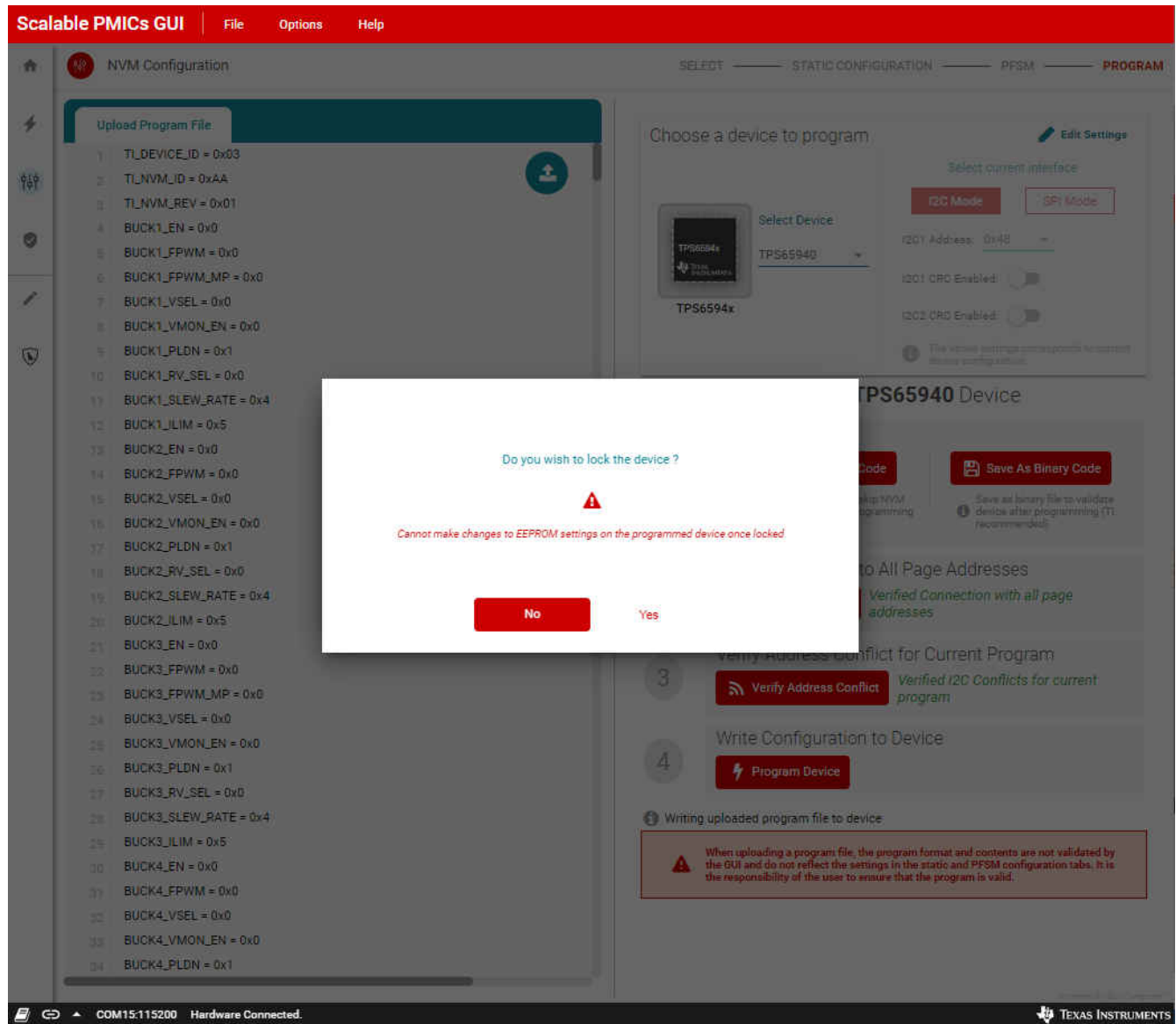


图 8-34. NVM 锁定选项

CAUTION

会永久性锁定器件。

9 NVM 验证页面

备注

德州仪器 (TI) 建议在“NVM Configuration” (NVM 配置) 页面中进行编程后, 务必使用 NVM 验证来确认器件的配置是否正确, 这是最佳做法。

“NVM Validation” (NVM 验证) 页面用于将当前寄存器或 NVM 设置下载到主机 PC 中, 或将该设置与主机 PC 中的文件进行比较。该页面的一个重要作用是, 通过覆盖当前寄存器设置来访问 NVM 设置。如果选择了 **NVM Settings** (NVM 设置) 按钮 (如图 9-1 所示), 那么选择 **DOWNLOAD CONFIGURATION** (下载配置) 按钮后 GUI 会立即向 PMIC 发出一组命令, 以使用 NVM 中的内容覆盖所有现有寄存器设置。这将覆盖通过“Quick-start” (快速启动) 或“Register Map” (寄存器映射) 页面对器件进行的任何设置或配置。完成覆盖后, 可通过通信接口读出寄存器内容。

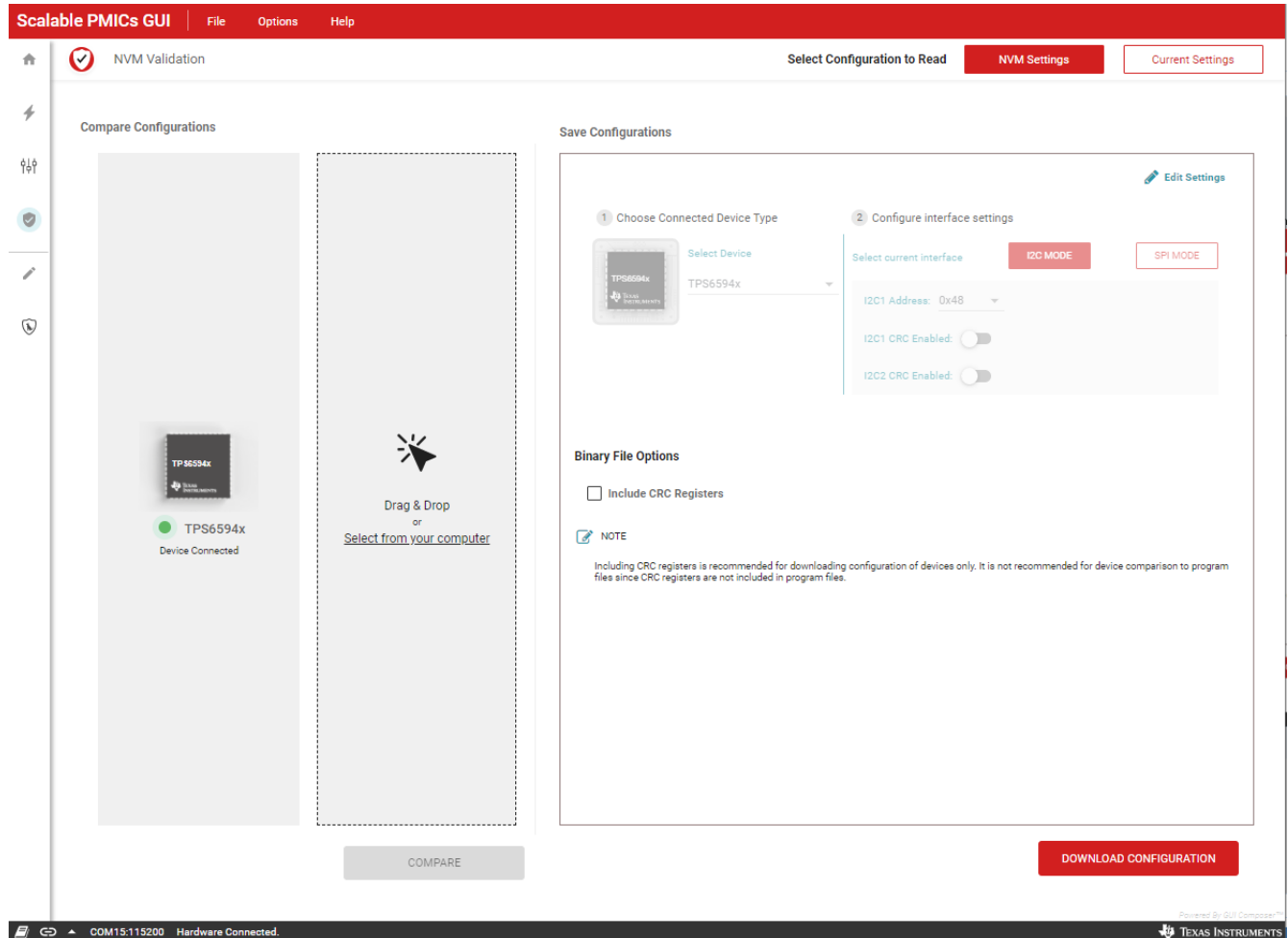


图 9-1. NVM 验证

页面的右上角指示正在读取 **Current Settings** (当前设置) 还是正在读取 **NVM Settings** (NVM 设置)。利用页面左侧的界面, 可以选择一个已知文件以将读取的内容与该文件进行比较。这提供了一个快速的可视化通过/失败响应来评估读取的内容。页面右侧提供了 **DOWNLOAD CONFIGURATION** (下载配置) 选项。

将当前或 NVM 设置与现有二进制文件进行比较。

1. 确认已连接了正确的器件。
 - a. 选择正确的器件类型。
 - b. 选择正确的通信接口。
2. 从主机 PC 中选择二进制文件以将器件 NVM 与其进行比较。
 - a. 使用拖放功能或文件导航器。
 - b. 选择有效文件后，**COMPARE** (比较) 按钮将变为有效状态。
3. 按下 **COMPARE** (比较) 按钮。

将当前或 NVM 设置下载到一个二进制文件中。

1. 确认已连接了正确的器件。
 - a. 选择正确的器件类型。
 - b. 选择正确的通信接口。
2. 选择 **Current Settings** (当前设置) 或 **NVM Settings** (NVM 设置) 按钮。
3. 按 **DOWNLOAD CONFIGURATION** (下载配置) 按钮。

10 看门狗页面

“Watchdog”（看门狗）页面是一个交互式评估工具，用于在“TRIGGER”和“Q&A”模式下执行看门狗功能。该工具使用 AEVM 上的 MSP432E 微控制器为正确和不正确的用例创建看门狗激励。进行状态监视以显示 PMIC 响应。

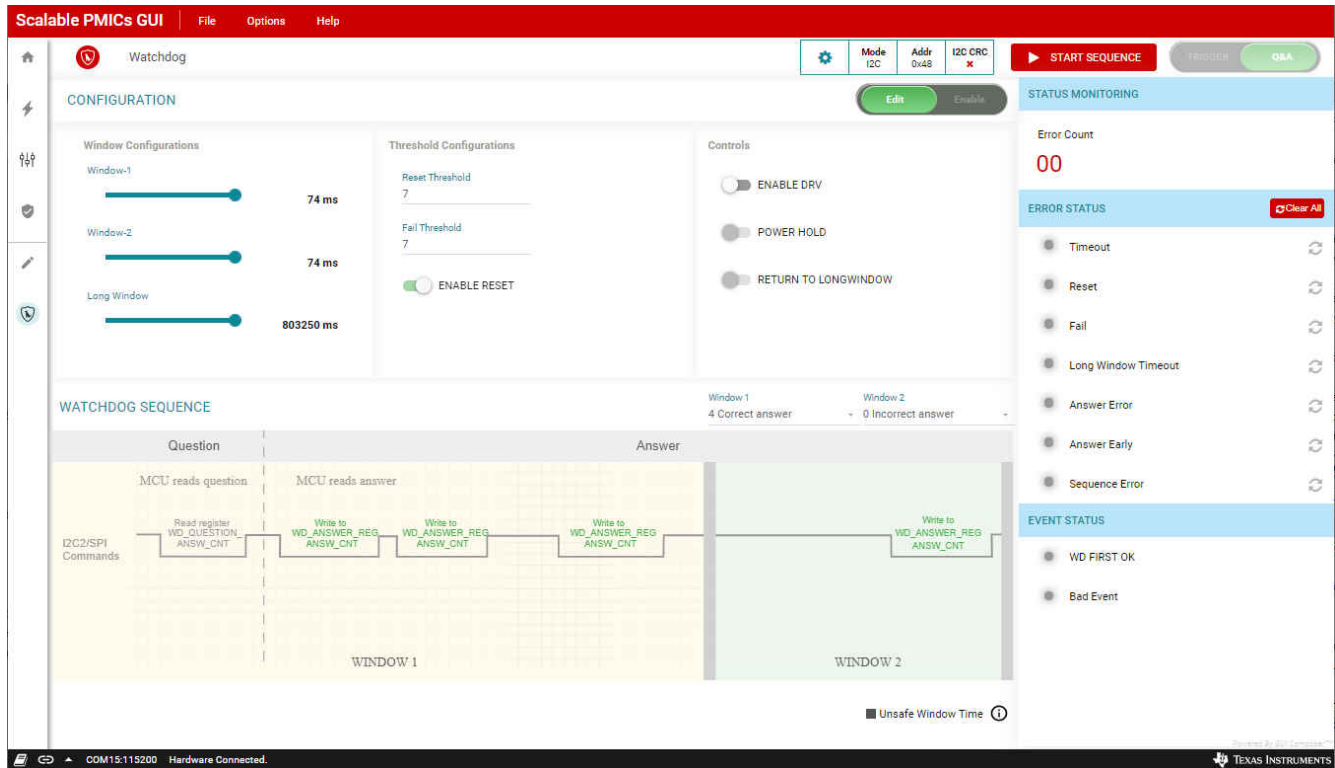


图 10-1. 看门狗页面

若要运行看门狗模块，请确保 PMIC 满足以下条件：

1. 器件支持看门狗功能。
2. PMIC 处于任务状态。**SAFE RECOVERY**（安全恢复）等硬件状态会阻止看门狗运行。
3. NRSTOUT（在寄存器 MISC_CTRL 中）必须为 1。
4. WD_EN（寄存器 WD_THR_CFG）必须为 1。
5. 任何配置为 DISABLE_WDOG 的 GPIO 都应处于逻辑低电平（如果适用于器件）。
6. WD_PWRHOLD 必须为 0。
7. 为了进行评估，建议将 WD_RST_EN 设置为 0，以便在评估期间器件不会进入 **WARM RESET** 状态。

触发条件模式

“Watchdog”（看门狗）页面将反映“Controls”（控件）部分中的设置。要从 QA 模式更改为 TRIGGER（触发条件）模式，应将 **Edit/Enable**（编辑/启用）指示器（在图 10-1 中以黄色突出显示）切换为 **Enable**（启用）。进入“TRIGGER”（触发条件）模式后，切换回 **Edit**（编辑）模式，并将默认的 Window-1 和 Window-2 值调整为未超过 AEVM 限值的时间范围。

使用有效的时序窗口，可以重新启用看门狗并启动看门狗序列。

启动序列：

1. 启用看门狗，选择 **Enable**（启用）
2. 选择模式，**TRIGGER**（触发条件）模式或 **Q&A 模式**
3. 编辑 AEVM 触发条件波形，选择 **编辑**
4. 调整 **Window-1** 和 **Window-2**

5. 选择 **Sequence Configuration** (序列配置)
6. 启用看门狗, 选择**启用**
7. 选择 **START SEQUENCE** (启动序列)

在“EVENT STATUS” (事件状态) 视图中, “WD FIRST OK” 呈绿色亮起 (请参阅图 10-2), 表示已正确接收第一个看门狗触发序列。在操作过程中会不断轮询 PMIC, 并更新“ERROR STATUS” (错误状态) 视图。

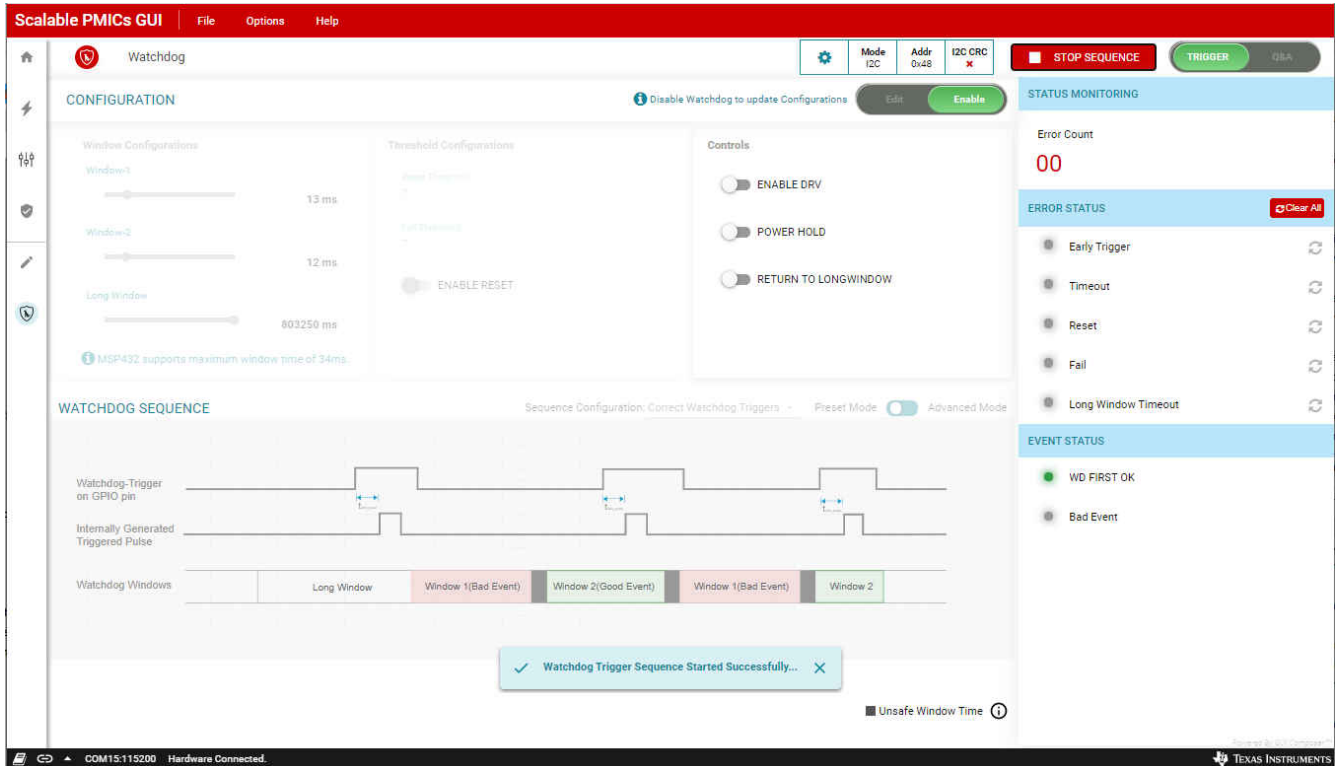


图 10-2. 看门狗触发条件模式

若要停止评估, 请首先使用控件设置“RETURN TO LONGWINDOW” (返回 LONGWINDOW), 再从 **Enable** (启用) 模式切换到 **Edit** (编辑) 模式, 然后点击 **STOP SEQUENCE** (停止序列)。

停止序列:

1. 选择 **RETURN TO LONGWINDOW** (返回 LONGWINDOW)
2. 选择 **Edit** (编辑)
3. 选择 **STOP SEQUENCE** (停止序列)

可以使用下拉菜单 **Sequence Configuration** (序列配置) 修改序列配置以故意提供不正确的序列。图 10-3 显示了不正确的时序和随之而来的糟糕事件。若要评估此配置, 首先从 **Edit** (编辑) 模式切换到 **Enable** (启用) 模式, 再禁用 **RETURN TO LONGWINDOW** (返回 LONGWINDOW), 然后点击 **START SEQUENCE** (启动序列)。

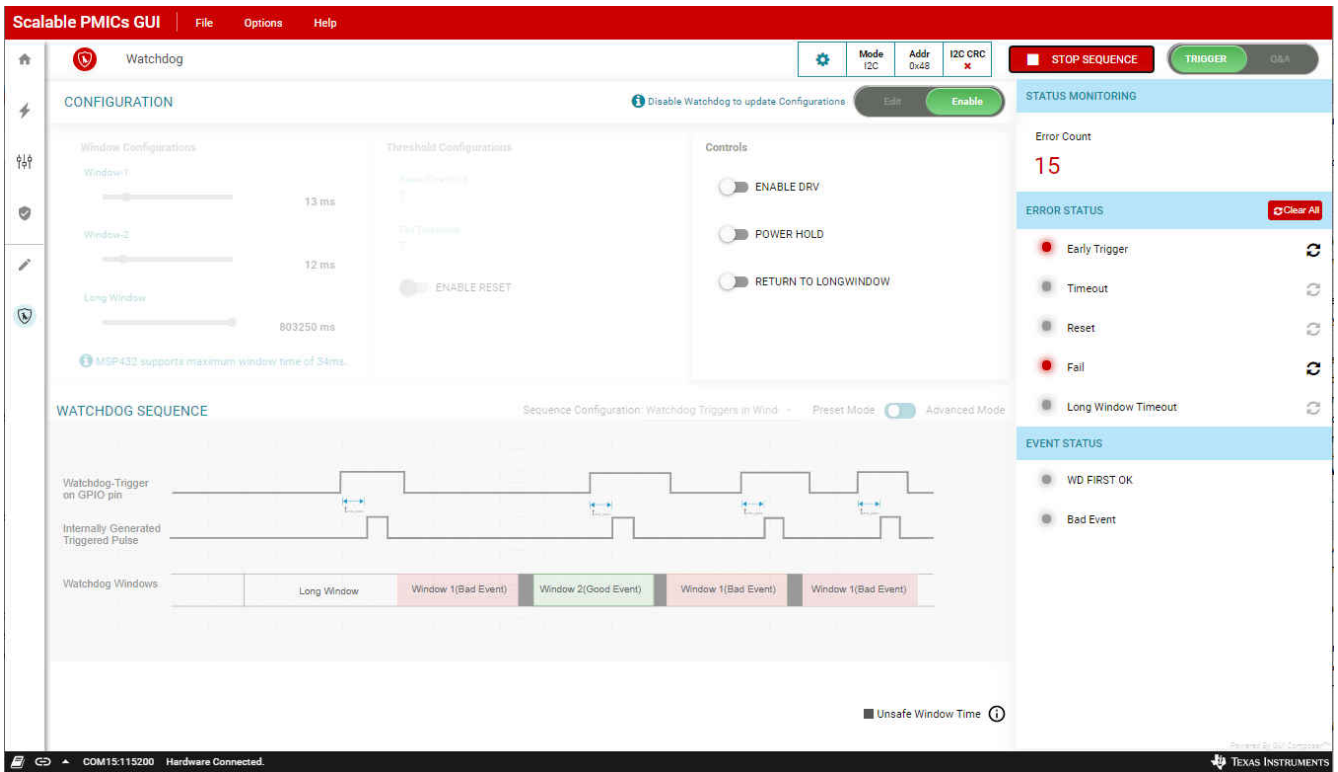


图 10-3. 无效的触发条件看门狗输入

可以使用每个错误右侧的图标或通过点击 *Clear All* (清除全部) 来清除错误。

Q&A 模式

如前所述，在 *Edit/Enable* (编辑/启用) 指示器移至 *Enable* (启用) 时，可以选择“Q&A”模式。

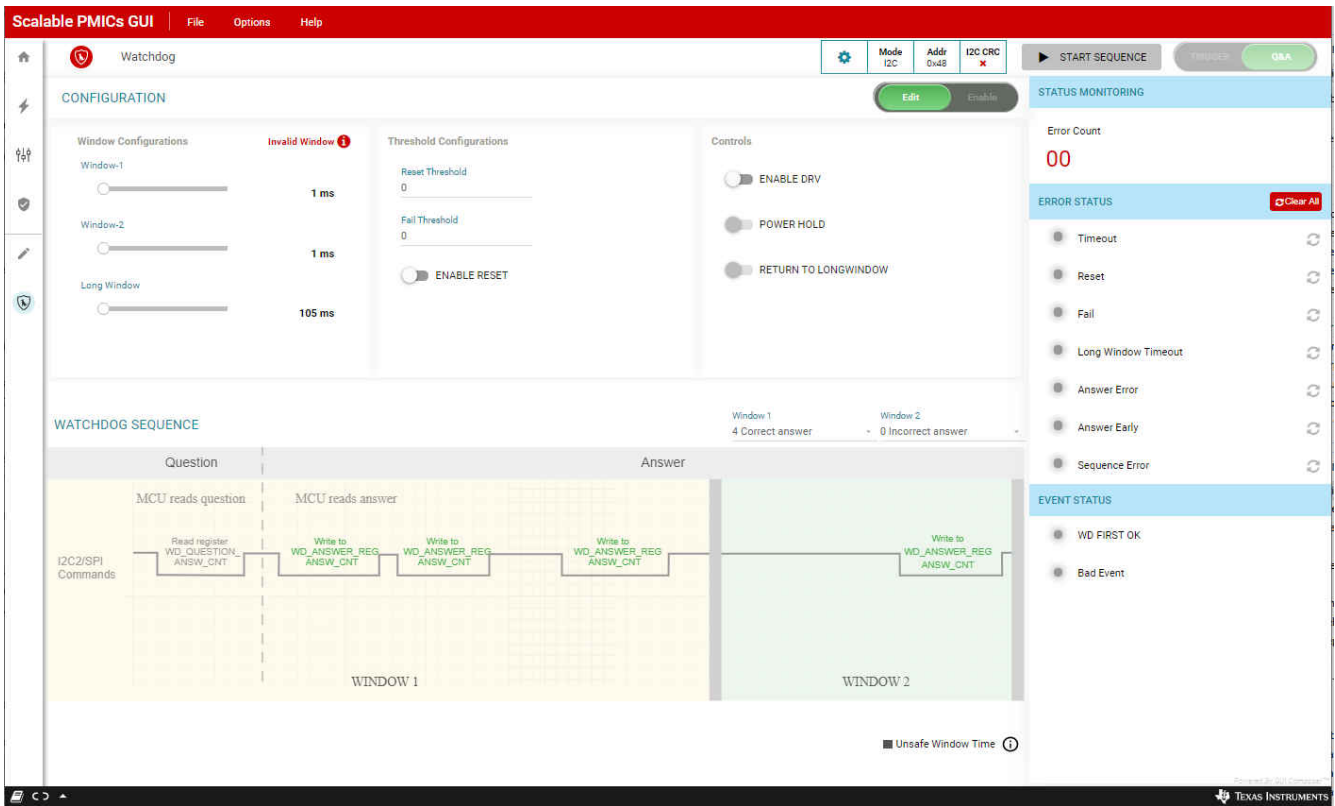


图 10-4. 看门狗 Q&A 模式

针对 Q&A 更新窗口配置后，可以使用类似的过程来评估“Q&A”模式。图 10-5 显示了问题的正确答案序列，而图 10-6 显示了错误答案。

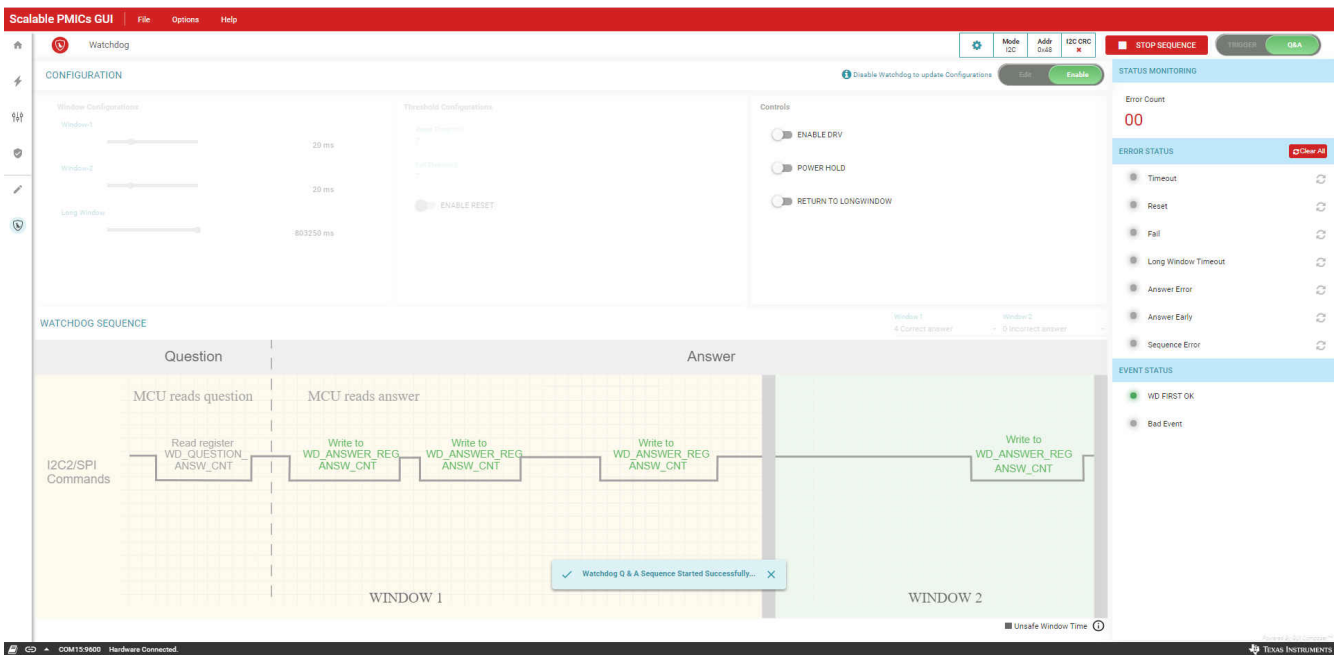


图 10-5. 有效的看门狗 Q&A 响应

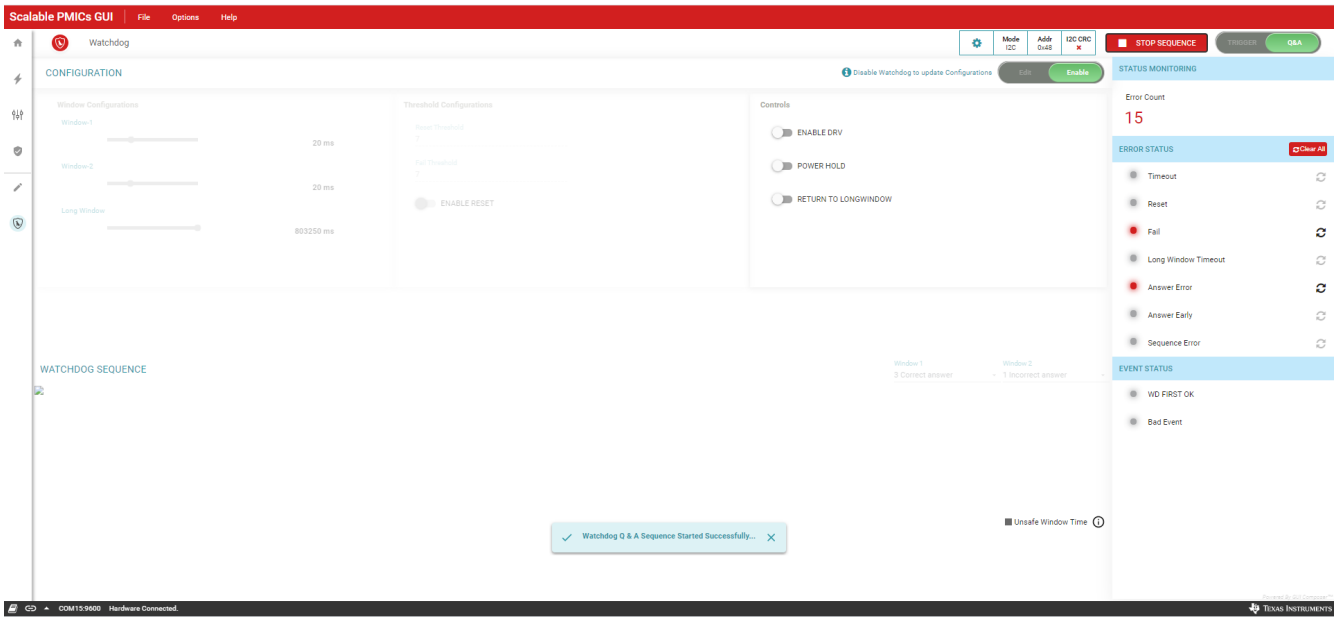


图 10-6. 无效的看门狗 Q&A 响应

11 其他资源

1. [SimpleLink™ 以太网 MSP432E401Y MCU Launchpad™ 开发套件](#)
2. [GUI Composer 用户指南](#)
3. [E2E 支持论坛](#)

12 附录 A：疑难解答

12.1 无法识别硬件平台

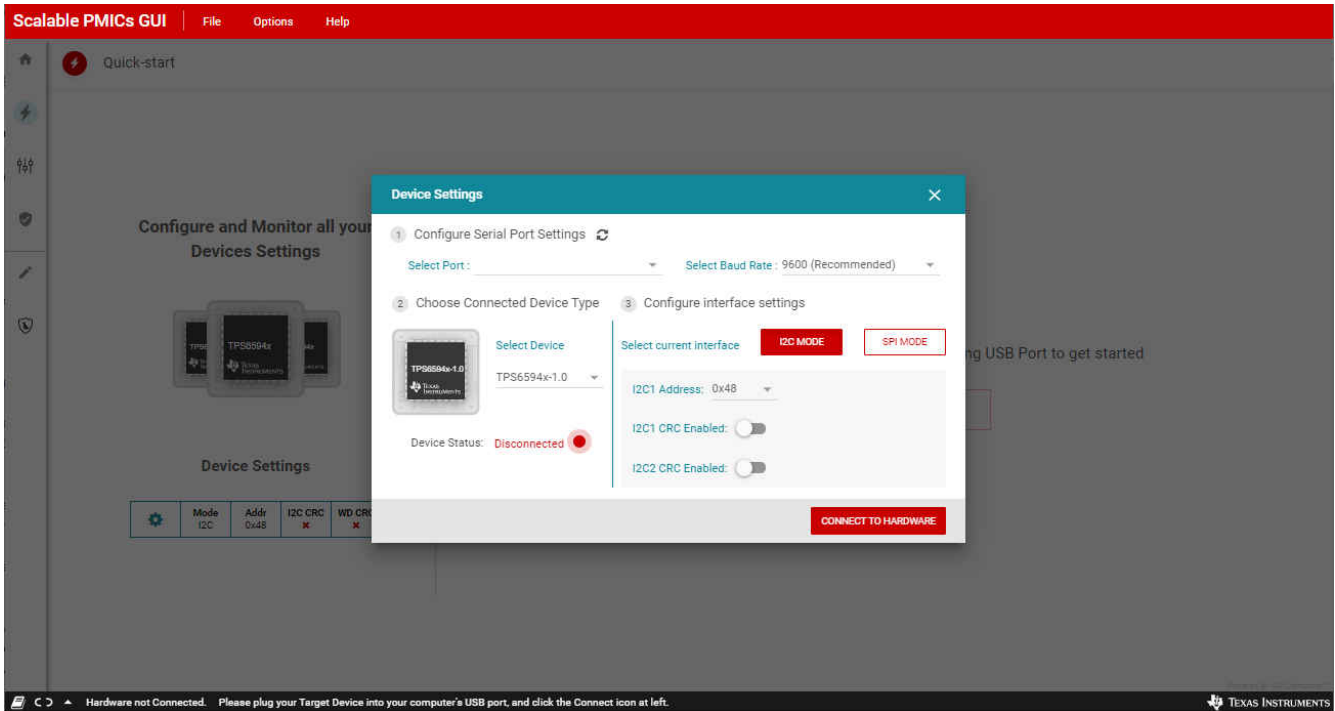


图 12-1. 硬件平台错误

GUI 将自动连接到 AEVM (具有模拟 EVM 控制器固件的微控制器)，然后连接到 PMIC。通常，如果 GUI 无法找到硬件平台，则是 USB 连接错误或 GUI 试图连接到错误的通信 (COM) 端口导致的。硬件平台将枚举为三类器件：两个 CDC 类和一个 DFU 类。用户可以通过另一个程序 (如 Windows 中的设备管理器) 来验证哪个 COM 端口是 ACctrl，哪个 COM 端口是 ACctrl 控制台。在 GUI 选项中，用户应选择 ACctrl (而非 ACctrl 控制台) 的 COM 端口号。AEVM 支持 115200 的波特率。

连接到 PC 的其他器件也可以枚举为 CDC 类器件，GUI 可能会尝试与这些器件进行通信。如果器件没有做出响应，那么在当前器件响应之前 GUI 可能不会尝试连接其他器件。

一次只能将一个 AEVM 连接到主机 (PC 的 MAC)。GUI 无法处理和响应连接到主机的多个 AEVM 器件。

12.2 未找到 PMIC 器件

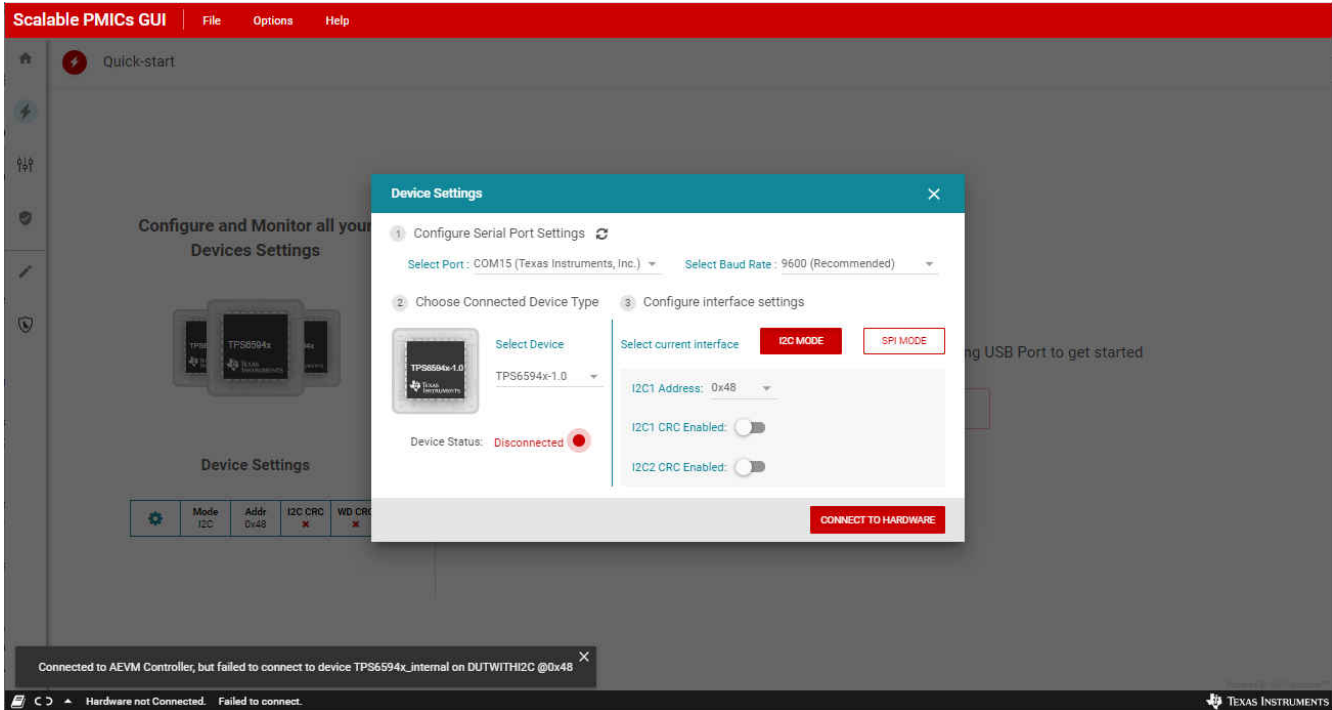


图 12-2. PMIC 通信错误

如果已连接硬件平台但找不到 PMIC 器件，则 GUI 会报告有关地址的错误：*Connected to AEVM Controller, but failed to connect to device TPS6594x_external on I2C @xx* (已连接到 AEVM 控制器，但未能连接到 I2C@xx 上的器件 TPS6594x_external)。对于 I²C，这意味着地址未被确认 (NACK)。执行相应的 I²C 总线检查：检查相应的上拉电阻器、验证是否没有器件将时钟保持在低电平（时钟拉伸），等等。验证发送的 I²C 地址是否正确。GUI 使用的默认地址可能不正确，需要使用 *Options* (选项) 选项卡下的 *Device Settings* (器件设置) 来更新地址。

与 SPI 类似，请确保硬件连接正确，并且在使用多个 PMIC 时将片选信号连接到相应的 PMIC。

12.3 I2C2 已配置但未连接

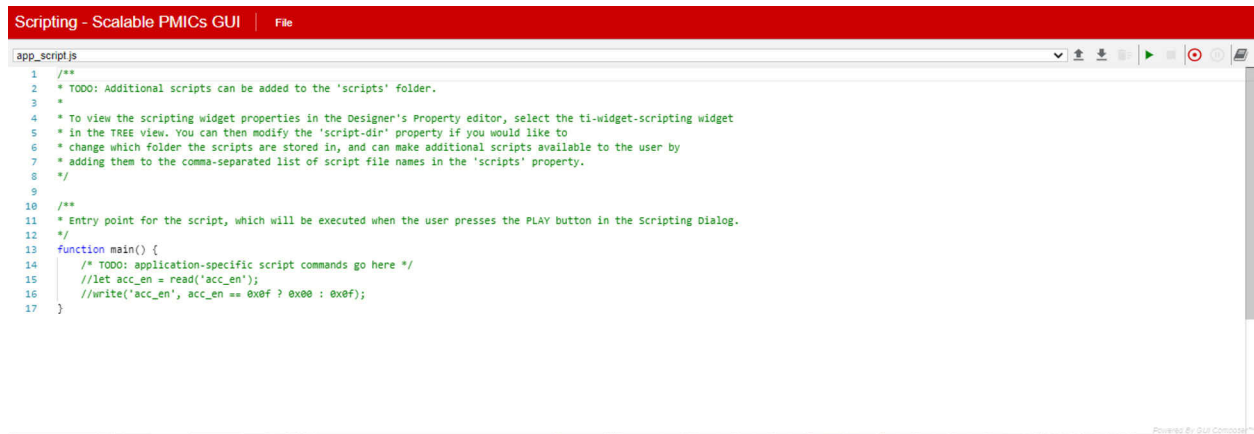
PMIC 支持专用 I²C 接口 I2C2，用于访问第 4 页的寄存器空间。只有看门狗配置需要此接口，并且仅在适当配置了 GPIO 时启用。GUI 将自动询问 PMIC 并通过 AEVM 使用 I2C2。如果 I2C2 没有物理连接但 GPIO 配置为用于 I2C2 操作，GUI 将在尝试访问第 4 页的寄存器空间时报告错误。在 GUI 版本 2.0.0 中，当用户尝试从第 4 页读取寄存器时会出现此错误；*READ REGISTER* (读取寄存器) 按钮。在 GUI 版本 3.0.0 中，当 PMIC 在连接后更新所有寄存器以及使用 *READ ALL REGISTERS* (读取所有寄存器) 按钮时，会立即显示此错误。在 GUI 版本 2.0.0 和 3.0.0 中，此错误都报告为 *I2C 读取错误。状态 = 4294966526*

如果不需要第 4 页的信息，可以忽略此错误。此外，建立 I2C2 连接将消除此错误。

13 附录 B : 高级主题

13.1 脚本编写窗口

脚本编写是向 PMIC 器件寄存器发送一系列读取或写入命令（而不是与“Quick-start”（快速启动）或“Register Map”（寄存器映射）页面视图中的参数更新相关联的各条命令）的便捷方式。*Scripting*（脚本编写）位于 GUI 顶部的“Options”（选项）选项卡项下。打开“Scripting”（脚本编写）窗口的同时会打开一个新窗口，而 GUI 窗口仍将处于有效状态，如下一段所示。



```

1  /**
2  * TODO: Additional scripts can be added to the 'scripts' folder.
3  *
4  * To view the scripting widget properties in the Designer's Property editor, select the ti-widget-scripting widget
5  * in the TREE view. You can then modify the 'script-dir' property if you would like to
6  * change which folder the scripts are stored in, and can make additional scripts available to the user by
7  * adding them to the comma-separated list of script file names in the 'scripts' property.
8  */
9
10 /**
11 * Entry point for the script, which will be executed when the user presses the PLAY button in the Scripting Dialog.
12 */
13 function main() {
14     /* TODO: application-specific script commands go here */
15     //let acc_en = read('acc_en');
16     //write('acc_en', acc_en == 0x00 ? 0x00 : 0x0f);
17 }
  
```

图 13-1. 默认脚本编写窗口

图 13-1 显示了初始“Scripting”（脚本编写）窗口和提供的默认文本。可以编辑该文件以提供所需的命令序列。开始使用“Scripting”（脚本编写）窗口的一种快速方法是使用录制功能来捕获一系列命令。录制图标位于右上角。将光标悬停在该图标上会显示 *Start Recording*（开始录制）帮助框。在图 13-2 和图 13-3 所示的示例中，录制已开始，然后当返回 GUI 窗口时，“Register Map”（寄存器映射）页面用于读取 DEV_REV 和 NVM_CODE_2，然后写入 BUCK1_CTRL 寄存器。这些操作序列完成后，返回到“Scripting”（脚本编写）窗口将显示录制的命令。此时可以停止录制，并且可以针对各种寄存器扩展和重复这些命令。完成该序列后，点击 *Run*（运行）图标执行该序列。

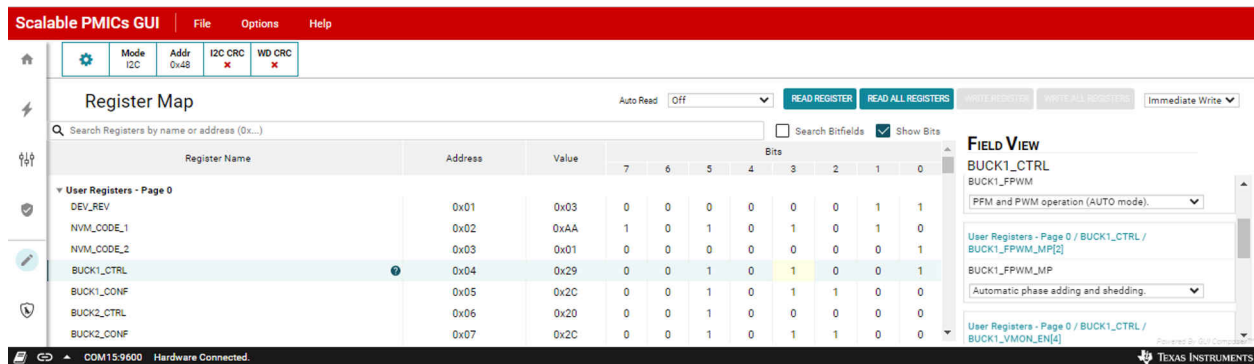


图 13-2. 脚本编写，录制寄存器读取和写入



图 13-3. 脚本编写，运行录制的序列

14 附录 C : 已知限制

本节包含 GUI 的已知限制，以及这些限制所适用的版本。如您发现以下列表中未包含的任何问题或限制，请在支持论坛中说明。

表 14-1. GUI 限制及相关版本

数量	1.0.0	2.0.0	3.0.0
1	X	X	X
2	X	X	X
3	X	X	X
4	X	X	
5	X	X	X
6	X	X	X
7	X	X	
8	X	X	X
9	X	X	X
10			X
11			X
12		X	X

表 14-2. GUI 限制说明

编号	说明	权变措施
1	当包含寄存器 CRC 时，NVM 验证可以报告与地址 0x0F4 (CRC_5)、0x0F5 (CRC_6)、0x0FE (CRC_15) 和 0x0FF (CRC_16) 相关的差异。	无。GUI 将报告 NVM 文件不匹配。检查寄存器并忽略与这些寄存器相关的差异。每个器件在 CRC 中都可以具有与生产信息相关的唯一值。
2	不建议在创建 PFSM 后修改器件类型及其配置（例如 Primary/Secondary、Phase Configuration 等），因为这可能不会产生预期的结果	如果修改相位配置，则应在尝试更改相位配置之前删除 PFSM 中的所有引用。更改器件类型需要重新模板或空白模板开始。
3	可能会错误地检测状态栏中显示的 SPI 硬件连接状态。	使用“Register map”（寄存器映射）页面确认是否可以从器件中读取除 0x00 和 0xFF 之外的值。此时会出现一条对话框消息以解决相同的问题。
4	GUI 不提供对多个 SPI 片选输出的控制。	无。在多器件 SPI 配置中，需要从 AEVM 手动将芯片选择信号单独移至每个 PMIC。
5	与蓝牙器件关联的 COM 端口的 USB 连接问题。	禁用或删除在主机 PC 上枚举为 COM 端口的器件。
6	枚举（将 USB 电缆连接到 AEVM）将尝试使用以前或默认的设置连接到器件，并且可能无法连接。	使用“Device Settings”（器件设置）并点击 Connect to Hardware （连接到硬件），将 GUI 更新为正确的连接设置。
7	启用寄存器 CRC 的编程可能会导致错误的 CRC 值。一些用户寄存器不被 NVM 支持，但却是 CRC 计算的一部分。	对 NVM 进行两次编程。
8	具有 SPI 接口的目标应在 GUI 配置为 SPI 模式后连接到 PC。	在 GUI 中选择 SPI 后为 AEVM 重新上电。
9	对 LP876x-Q1 系列器件进行编程，并在 nRSTOUT 和其他一些函数之间更改 GPIO 功能会导致串行接口暂时丢失。通常，这会导致 I2C 中的 NACK 和 SPI 中的帧错误。NACK 将被 GUI 视为错误，并且编程序列将停止。	使用寄存器映射页面更改 GPIO 设置，以匹配待编程的 NVM 中的设置。在第一次尝试失败后重新编程（不重启 PMIC）也会产生相同的影响，因为 GPIO 设置的用户寄存器已在初始尝试编程时更新。
10	如果在序列结束时未恢复 PFSM 延迟步骤，PFSM 验证将在“延迟时间验证”中报告警告，即使 PMIC 在序列执行后将切换到 SAFE_RECOVERY 状态也是如此。	无。当 PMIC 转换到 SAFE_RECOVERY 状态时，PFSM 延迟步骤会自动恢复为默认值。只要 PMIC 转换为 SAFE_RECOVERY，用户则可以忽略此警告。J721E_TPS6594 模板中可以看到此警告。
11	如果使用 SREG_DELAY 指令在主辅 PMIC 之间创建所需的时序关系，PFSM 验证将在“序列长度验证”中报告错误。	无。SREG_DELAY 指令是一个独特的延迟指令，其可以在运行期间改变值，具体取决于加载到 SREG 中的值。如果了解了延迟的使用并保留时序关系，用户就可以忽略此错误。这只适用于多 PMIC 应用。J721E_TPS6594 模板中可以看到此错误。
12	BUCK 电压选择 BUCKx_VOUT1 和 BUCKx_VOUT2 不受 BUCK 用例限制。	用户必须确保所选的输出电压在给定的配置用例的数据表中指定的范围内。

15 附录 D : 迁移主题

版本 3.0.0 的相关更改将导致使用版本 2.0.0 创建的 json 文件不兼容，需要进行以下更新。

1. 从 LP8764-Q1 器件版本 PG1.0 迁移到器件版本 PG2.0。
2. 更新 PFSM 以包含 pfsm_start。
3. 更新所有手动实施的时间延迟。
4. 更新触发优先级和设置。

15.1 从 LP8764-Q1 PG1.0 迁移到 PG2.0

LP8764-Q1 PG1.0 和 LP8764-Q1 PG2.0 在 GUI 上的区别与格式有着密切的关系，会导致如图 15-1 中所示的错误。可以通过 e2e 论坛 3 提出申请以更新格式。或者，用户可以选择修改 json 文件。这些修改很小，仅与 json 文件中的器件名称有关。此示例使用 *Generic_LP8764* 模板作为从 PG1.0 迁移到 PG2.0 的示例。

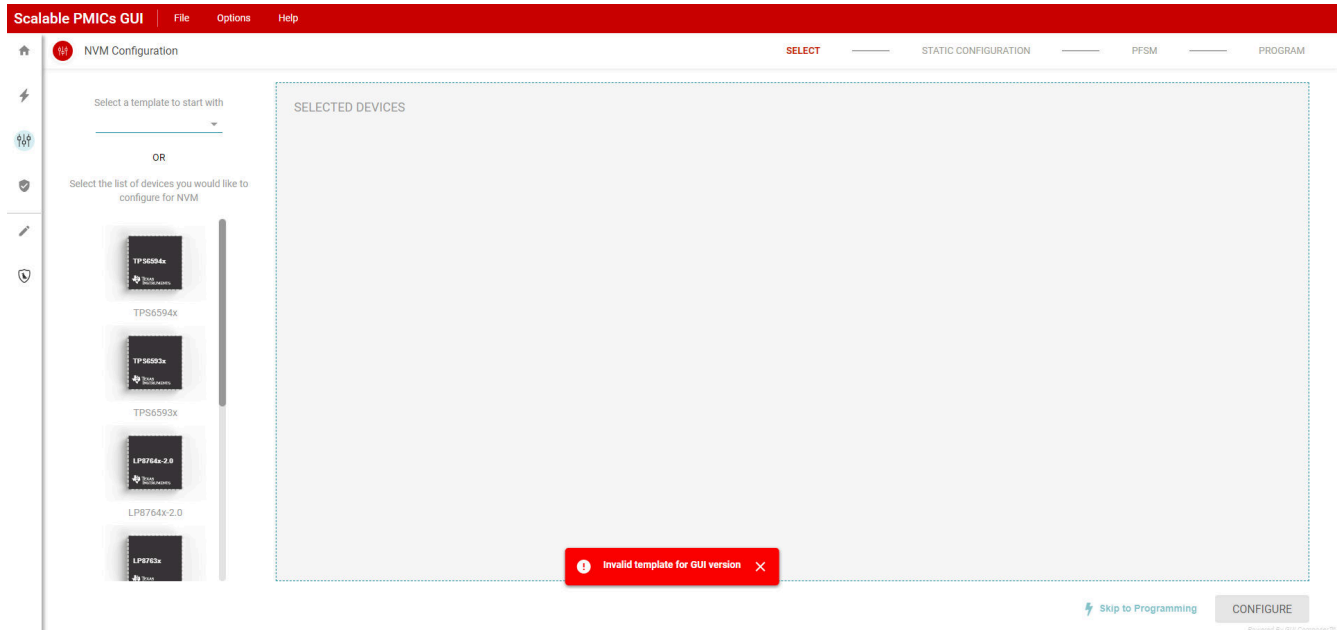
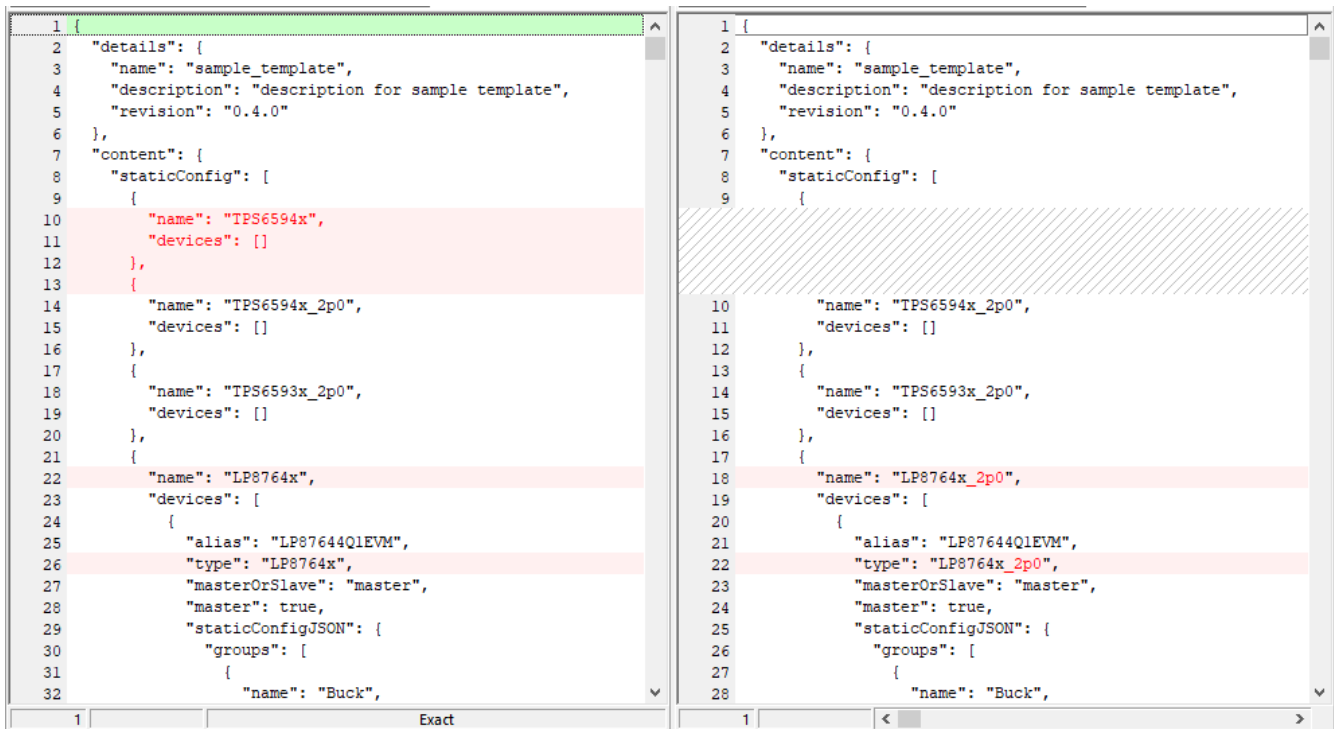


图 15-1. 尝试将 LP8764-Q1 PG1.0 导入 GUI 版本 3.0.0 时出现错误消息

如图 15-2 中所示，必须删除 TPS6594-Q1 PG1.0，并且 LP8764-Q1 必须更新为 PG2.0。左侧是原始 json 文件的左侧和，右侧是为使 json 文件与 GUI 版本 3.0.0 兼容所需的更改。



```

1 {
2   "details": {
3     "name": "sample_template",
4     "description": "description for sample template",
5     "revision": "0.4.0"
6   },
7   "content": {
8     "staticConfig": [
9       {
10        "name": "TPS6594x",
11        "devices": []
12      },
13      {
14        "name": "TPS6594x_2p0",
15        "devices": []
16      },
17      {
18        "name": "TPS6593x_2p0",
19        "devices": []
20      },
21      {
22        "name": "LP8764x",
23        "devices": [
24          {
25            "alias": "LP87644Q1EVM",
26            "type": "LP8764x",
27            "masterOrSlave": "master",
28            "master": true,
29            "staticConfigJSON": {
30              "groups": [
31                {
32                  "name": "Buck",
    
```

图 15-2. 需要标题 ***

备注

不支持从发布前的器件 TPS6594-Q1 PG1.0 迁移到发布后的器件 TPS6594-Q1 PG2.0。GUI 版本 2.0.0 和 3.0.0 均支持 TPS6594-Q1 PG2.0。

15.2 更新 PFSM 以包含 PFSM_START 状态

此迁移示例继续使用 *Generic_LP8764* 并对 json 文件进行必要的更改，如节 15.1 中所述。

如节 8.1.2.1 中所述，PFSM_START 状态是必须的，并且必须添加此状态。在 *Generic_LP8764* 示例中，可以从 ACTIVE 或 MCU_ONLY 状态进入 STANDBY (待机) 状态，因此必须添加一个单独的 PFSM_START 状态，如图 15-3 中所示。如果 STANDBY (待机) 状态没有来自其他状态的输入并被指定为开始状态，可以简单地将 STANDBY (待机) 状态重命名为 PFSM_START。

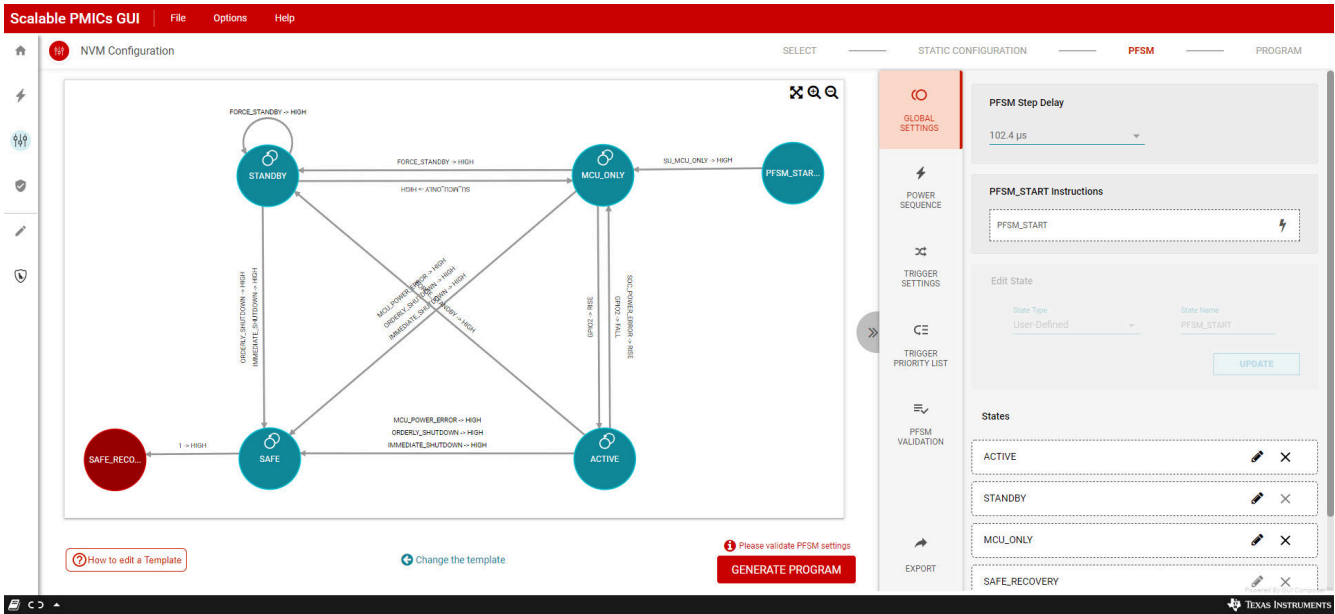


图 15-3. 添加 PFSM_START 状态

15.3 更新时间延迟

从 GUI 版本 3.0.0 中删除自动时间管理需要手动更新时间。需要更新的地方在 PFSM 验证中显示。如图 15-4 中所示，DELAY_IMM 500 ms 命令需要在执行该命令之前更新 PFSM_STEP_DELAY。DELAY_IMM 500 ms 和相关修复位于 *immediate_pd_shutdown* 电源序列中，如图 15-5 中所示。

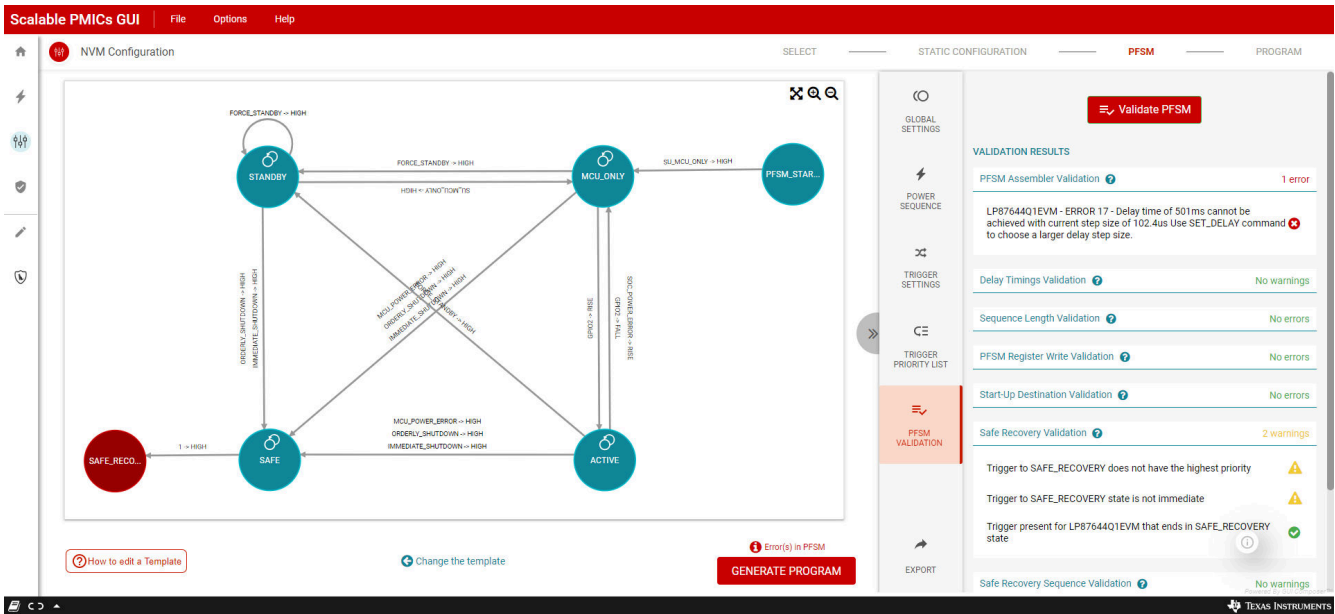


图 15-4. 具有延迟时间的 PFSM 汇编程序验证错误

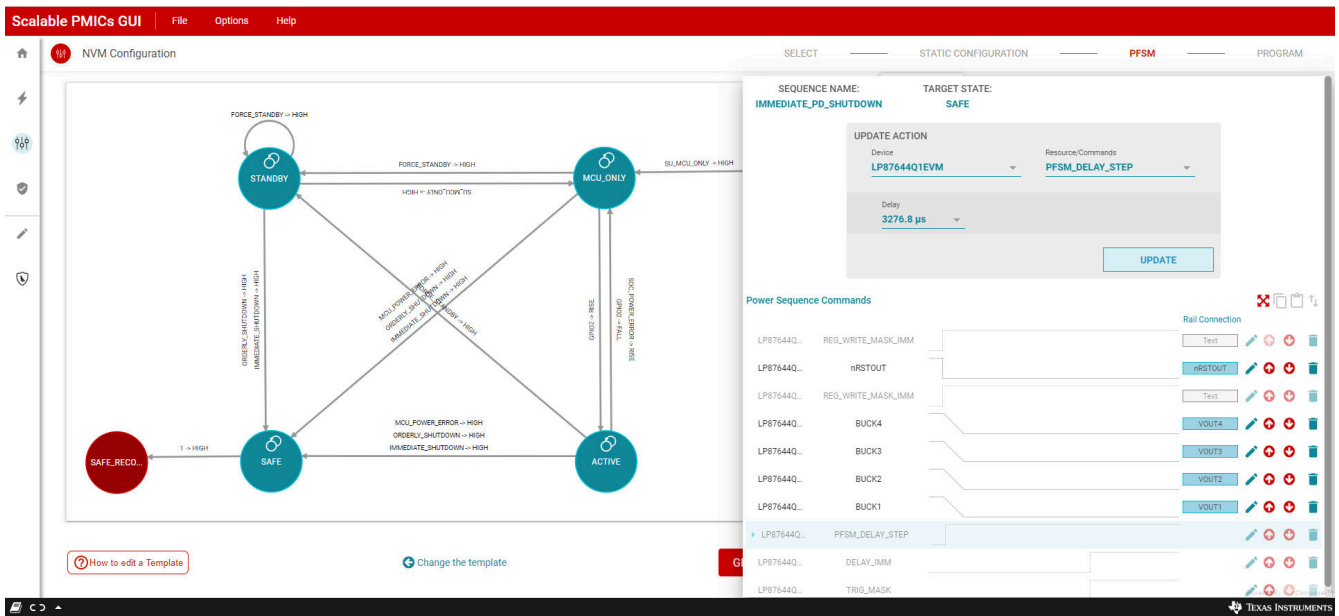


图 15-5. 迁移的延迟时间修复

一旦更新序列，错误将得到解决，如图 15-6 中所示。该警告是此示例有意为之的结果。在大多数情况下，必须将 PFSM_DELAY_STEP 恢复为全局设置。在这种特殊情况下，在转换到 SAFE（安全）状态之前执行该序列，然后自动转换到硬件 SAFE_RECOVERY 状态。在这种情况下，当 PMIC 从 SAFE_RECOVERY 状态返回到任务状态时，PFSM_DELAY_STEP 会自动恢复到全局设置，并且可以忽略警告消息。

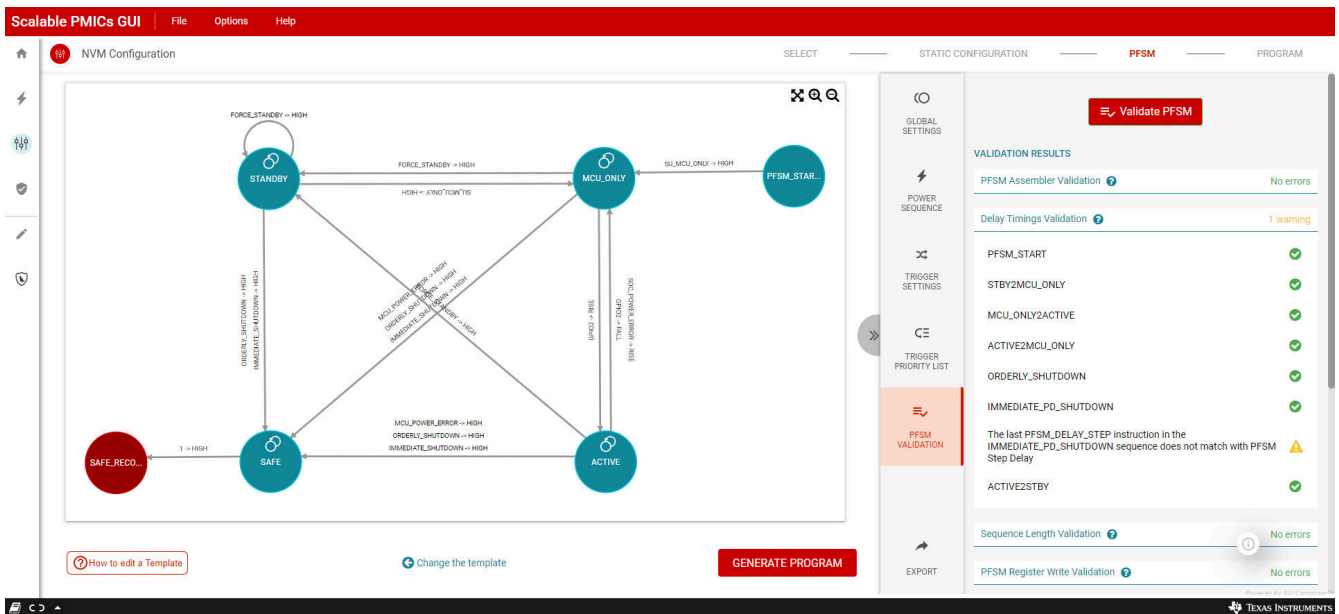


图 15-6. 具有更新时间延迟的 PFSM 验证结果

15.4 更新触发条件优先级和设置

继续使用节 15.3 中的示例，还有与触发条件相关的其他警告。图 15-7 显示了与触发条件 SAFE_RECOVERY 相关的警告。

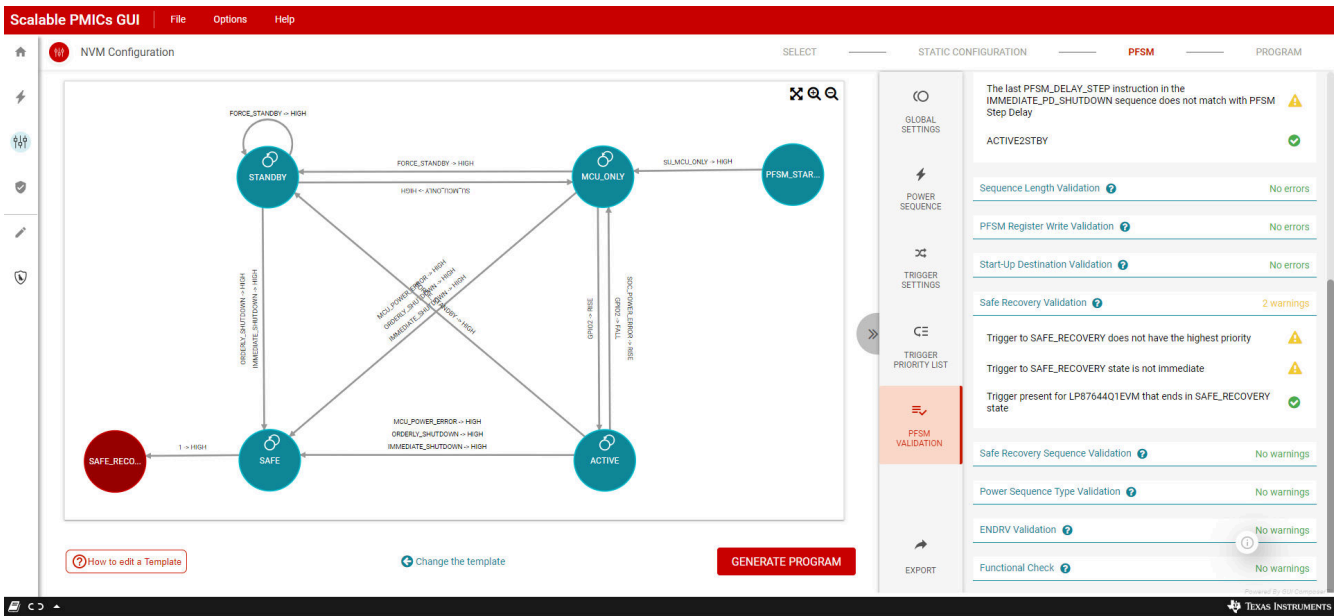


图 15-7. 触发条件 SAFE_RECOVERY 的警告

可以在 *Trigger Priority List* (触发条件优先级列表) 中更新触发条件优先级, 如图 15-8 中所示。图 15-9 显示了 *TRIGGER SETTINGS* (触发条件设置) 中找到的 *Update Trigger* (更新触发条件) 中指示的立即触发功能。

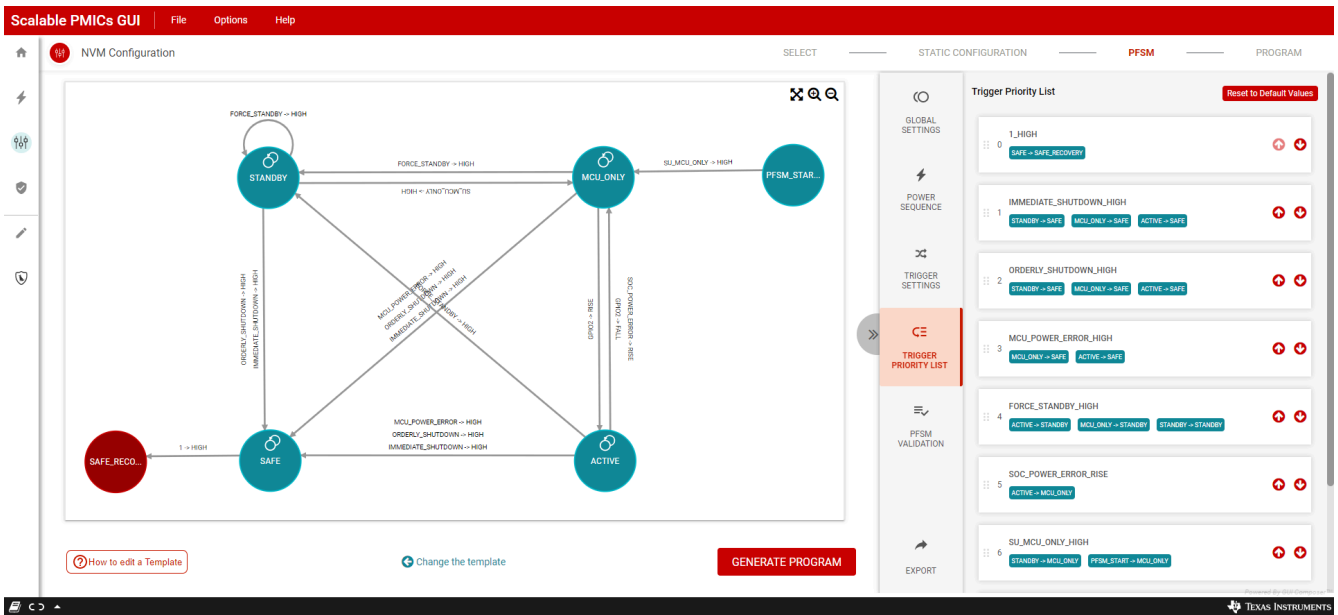


图 15-8. 优先级列表更新

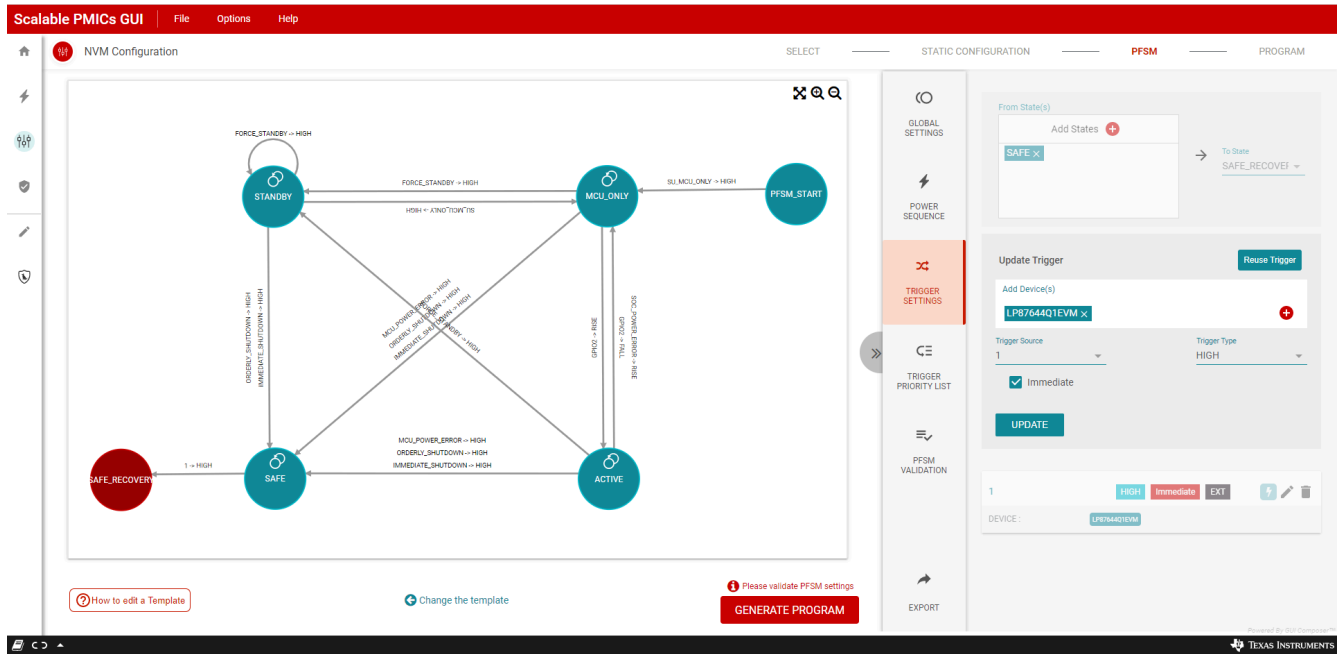


图 15-9. 立即触发功能

图 15-10 显示了通过这些修复，PFSM 验证只有一个警告，在节 15.3 中进行了讨论。

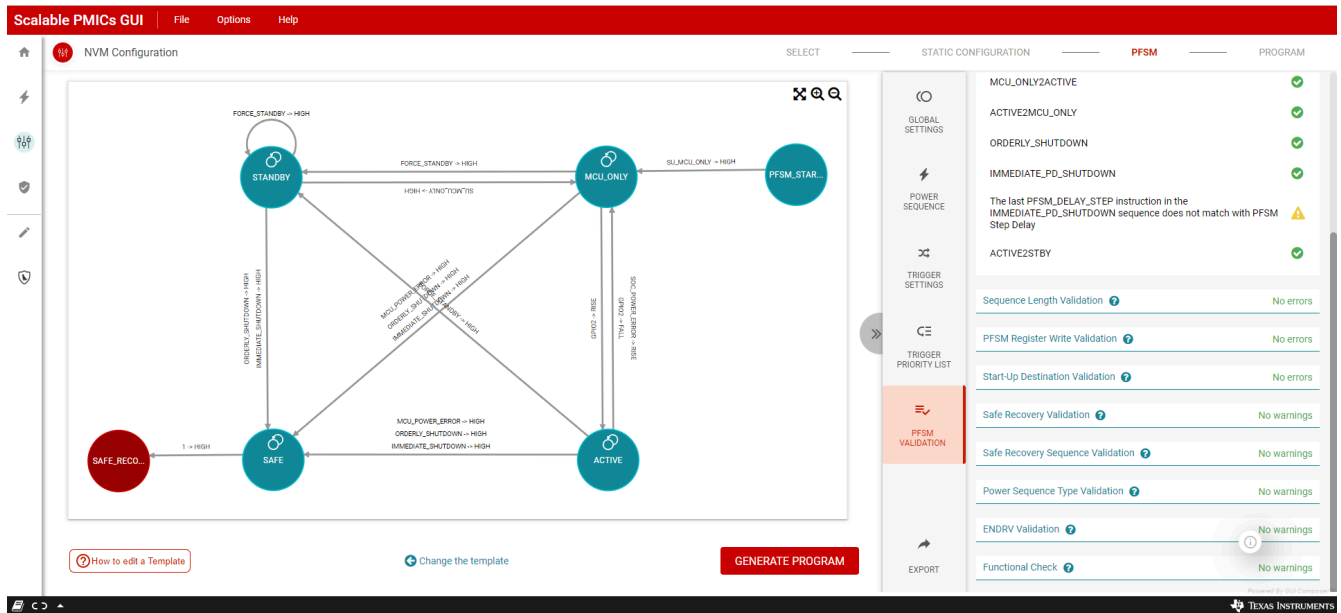


图 15-10. 解决触发警告后的 PFSM 验证结果

16 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 A (November 2020) to Revision B (June 2022)	Page
• 新增其他器件系列。.....	1
• 更新了包容性语言。.....	3
• 新增了版本 3.0.0 的摘要。.....	4
• 更新了版本 3.0.0 的图像.....	5
• 将版本更新至 3.0.0.....	6
• 新增了有关最初连接到配置有 SPI 串行接口的 PMIC 的说明.....	7
• 新增多 PMIC SPI 选项的信息.....	9
• 更新了 SPI 信息.....	13
• 删除了注释.....	23
• 更新了版本 3.0.0 的图像。.....	28
• 新增了 PFSM_START 状态。.....	30
• 更新了版本 3.0.0 的全局设置.....	31
• 更新了版本 3.0.0 的电源序列窗口.....	32
• 新增了指令：nRSTOUT_SOC、VMONx、PFSM_DELAY_STEP 和 END 以及说明。更新了 DEACTIVATE 说明：不建议用于多 PMIC 解决方案。.....	36
• 更新了版本 3.0.0 的图像.....	42
• 添加了表以按照版本对问题进行分类，并新增了新限制。.....	66
• 新增了附录 D.....	67
Changes from Revision * (December 2019) to Revision A (November 2020)	Page
• 在整个文档中针对版本 2.0.0 进行了更新.....	3

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 © 2022，德州仪器 (TI) 公司