

Application Note

使用 EC 的常见 TPS25751 用例和设置



Roy Chou

摘要

TPS25751 是一款高度集成的独立式 USB Type-C® 和电力输送 (PD) 控制器，针对支持 USB-C PD 电源的应用进行了优化。一般而言，TPS25751 会在设备通电时首先通过 EEPROM 加载补丁文件。如果没有 EEPROM，系统需要使用嵌入式控制器 (EC) 向 I2Ct 发出补丁文件。TPS25751 技术参考手册展示了通过 I2Ct 总线将补丁捆绑包同时推送到多个 PD 控制器的流程，但这部分内容可能较难理解。本应用手册展示了通过 I2Ct 使用 I2C 命令从 PTCH 模式切换到 APP 模式的分步说明，同时还展示了 TPS25751 中的常用功能。

内容

1 引言.....	2
2 ADCINX 设置.....	2
3 唯一地址接口协议.....	4
4 从 PTCH 模式切换至 APP 模式.....	5
4.1 从 PTCH 模式切换至 APP 模式的步骤.....	6
4.2 生成低区二进制文件的步骤.....	9
5 电池无电配置.....	10
6 I2Ct IRQ 的中断事件、屏蔽和清除.....	10
7 GPIOx 功能.....	11
8 4CC 命令.....	11
9 总结.....	12
10 参考资料.....	12

商标

USB Type-C® is a registered trademark of USB Implementers Forum.

所有商标均为其各自所有者的财产。

1 引言

本应用手册重点介绍了如何使用 EC 通过 I2Ct 向 TPS25751 发出 4CC 命令来从 PTCH 模式切换至 APP 模式。在此操作之前，本文介绍了与 TPS25751 初始配置相关的 ADCINX 配置和电池无电配置。在进入 App 模式后，本文介绍了一些 TPS25751 的常用功能，例如 I2Ct 设置和 GPIO 设置。

2 ADCINX 设置

设置适当的 ADCINX 值是正常使用 I2Ct 的第一种方法。建议参阅 TPS25751 数据表中的表 2-1。对于该应用，ADCINX 使用 EC，因此建议将 AlwaysEnableSink 或 NegotiateHighVoltage 用于电池无电配置。请注意，如果没有 EEPROM 而无法加载补丁文件，则使用 SafeMode。然后，设置与 I2C 总线上其他 I2C 地址不冲突的 I2C 地址索引。这里，ADCINX 为示例设置了 #1。

定义 ADCINX 解码值后，ADCINX 可以从表 2-2 获得 TPS25751 的唯一 I2C 地址。对于该设置，TPS25751 的唯一 I2C 地址为 0x20。

表 2-3 展示了用于设置所需 ADCINX 解码值的建议电阻。ADCINX 引脚必须在外部通过电阻分压器连接到 LDO_3V3 引脚，如图 2-1 所示。

表 2-1. 采用 ADCIN1 和 ADCIN2 的器件配置

ADCIN1 解码值	ADCIN2 解码值	I2C 地址索引	电池无电配置
7	5	#1	AlwaysEnableSink：无论连接的电源提供多少电流，该器件始终启用灌电流路径。在加载配置之前，USB PD 保持禁用状态。该配置与外部嵌入式控制器一起使用。当存在电池充电器时，嵌入式控制器负责管理系统中的电池充电器。
5	5	#2	
2	0	#3	
1	7	#4	
7	3	#1	NegotiateHighVoltage：无论连接的电源提供多少电流，该器件都会在初始隐式合约期间始终启用灌电流路径。PD 控制器进入 APP 模式，启用 USB PD PHY 并协商提供高达 20V 的最高功率合约。从 EEPROM 加载补丁时，无法使用该配置。对于能够从 5V 启动的系统，不建议使用此选项。此配置不适用于任何受支持的电池充电器。
3	3	#2	
4	0	#3	
3	7	#4	
7	0	#1	SafeMode：该器件未启用灌电流路径。在加载配置之前，USB PD 保持禁用状态。请注意，该配置可以将器件置于仅供电模式。当应用从 EEPROM 加载补丁时，建议使用此配置。当 PD 控制器负责管理电池充电器时，建议使用此配置。
0	0	#2	
6	0	#3	
5	7	#4	

表 2-2. I2Ct_SCL 的 I2C 默认目标地址，SDA

I2C 地址索引 (从 ADCIN1 和 ADCIN2 解 码)	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0	启动期间可用
#1	0	1	0	0	0	0	0	R/W	是
#2	0	1	0	0	0	0	1	R/W	是
#3	0	1	0	0	0	1	0	R/W	是
#4	0	1	0	0	0	1	1	R/W	是

表 2-3. ADCIN1 和 ADCIN2 引脚的解码

DIV = Rdown/(Rup + Rdown)			不使用 RUP 或 RDOWN	ADCINX 解码值
最小值	目标	最大值		
0	0.0114	0.0228	连接至 GND	0
0.0229	0.0475	0.0722	不适用	1
0.0723	0.1074	0.1425	不适用	2
0.1425	0.1899	0.2372	不适用	3
0.2373	0.3022	0.3671	不适用	4

表 2-3. ADCIN1 和 ADCIN2 引脚的解码 (续)

DIV = Rdown/(Rup + Rdown)			不使用 RUP 或 RDOWN	ADCINX 解码值
最小值	目标	最大值		
0.3672	0.5368	0.7064	连接至 LDO_1V5	5
0.7065	0.8062	0.9060	不适用	6
0.9061	0.9530	1.0	连接至 LDO_3V3	7

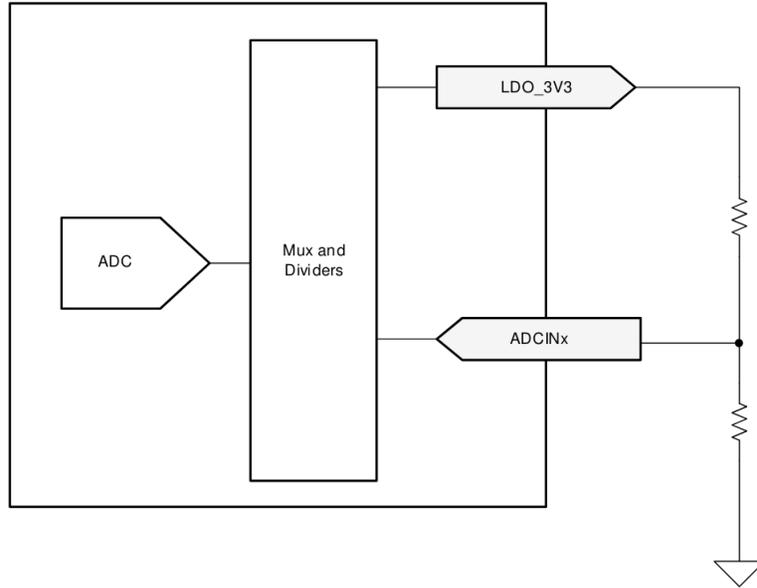


图 2-1. ADCINX 电阻分压器

3 唯一地址接口协议

唯一地址接口允许 I2C 控制器和单个 PD 控制器之间进行复杂的交互。I2C 目标唯一地址用于接收或响应主机接口协议命令。图 3-1 和图 3-2 分别展示了写入和读取协议。寄存器写入期间使用的字节计数可以大于实际写入的字节数，换句话说，控制器可以发出停止位而不写入 N 个字节。同样地，在寄存器读取期间，控制器可以在读取所有 N 个字节之前发出停止位。N 个字节是指要读取或写入的字节数。

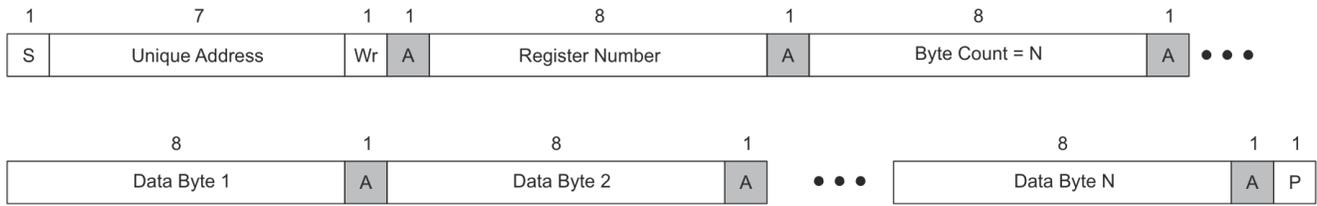


图 3-1. I2C 唯一地址写入寄存器协议

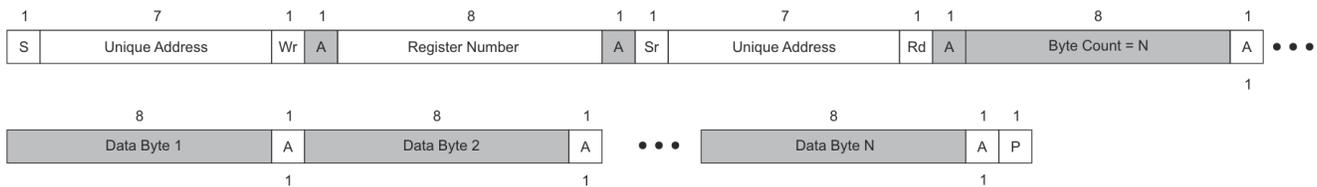


图 3-2. I2C 唯一地址读取寄存器协议

4.1 从 PTCH 模式切换至 APP 模式的步骤

第 1 步：对 TPS25751 施加 VIN_3V3 后，主机可以读取 I2C1 寄存器位 [81] (偏移 = 14h) 的中断事件，以确定器件是否已准备好接收主机的补丁捆绑包。以下是该命令及预期结果的示例。

[0x20] + ACK (唯一地址/WR/A)

0x14 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x0B (字节计数)

0x00 0x02 (MSB)

第 2 步：读取模式 (偏移 = 3h)，确保 TPS25751 在 PTCH 模式下运行。以下是该命令及预期结果的示例。

[0x20] + ACK (唯一地址/WR/A)

0x03 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x50 0x54 0x43 0x48 (以 4 个 ASCII 字符表示的 PTCH)

第 3 步：然后，通过针对 I2Ct 总线上的每个 PD 控制器写入 DATA1(9h) 来准备写入 PBM。所有 PD 控制器的 DATA1.TargetAddress 需要相同。对于 4CC 命令，检查 DATA1 是需要写入相应值，还是仅写入 CMD1(8h) 以执行 4CC 命令。对于 PBMs 4CC 命令，需要首先写入 DATA1(9h) 并在 CMD1 中写入 PBMs 4CC 命令。

[0x20] + ACK (唯一地址/WR/A)

0x09 + ACK (寄存器编号/A)

0x06 (字节计数)

0x80 0x2C 0x00 0x00 0x30 0x32 (捆绑包大小的字节 1/2/3/4，I2C 目标地址，超时值)

捆绑包大小可以参见下一节。

表 4-1. PBMs 任务 - 启动补丁突发下载序列

说明	PBMs 任务启动补丁加载序列。此任务初始化固件以准备补丁捆绑包加载序列并指示补丁捆绑包中包含的内容		
	位	名称	说明
INPUT DATAx	字节 6：突发模式超时		
	7:6	保留	
	5:0	超时值	此任务的超时值。必须使用非零值，此字段中始终使用 0x32 (5 秒) (LSB 为 100ms)。
	字节 5：用于下载补丁的 I2C 目标器件。		
	7	保留	
	6:0	I2C 目标地址	以下目标地址无效： <ul style="list-style-type: none"> • 0x00 • 由 ADCINx 引脚选择的任何端口的 I2Ct 目标地址。请参阅数据表。
	字节 0-3：低区二进制捆绑包大小 (以字节为单位)：[字节 4, 字节 3, 字节 2, 字节 1]		
	39:32	捆绑包大小的字节 4	
	31:24	捆绑包大小的字节 3	
	23:16	捆绑包大小的字节 2	
15:8	捆绑包大小的字节 1		

表 4-1. PBMs 任务 - 启动补丁突发下载序列 (续)

说明	PBMs 任务启动补丁加载序列。此任务初始化固件以准备补丁捆绑包加载序列并指示补丁捆绑包中包含的内容			
	位	名称	说明	
OUTPUT DATAx	7:0	PatchStartStatus	补丁启动状态。	
			0x00	补丁启动成功
			0x04	捆绑包大小无效
			0x05	目标地址无效
			0x06	超时值无效
任务完成	当输出具有有效的 PatchStartStatus 后, PBMs 任务完成。如果 MODE 寄存器 (0x03) 等于 APP, 则可以拒绝此任务。			
副作用	当 'PBMs' 成功时, 第二个目标地址可以设置为输入值。			
其他信息	主机只能向 PD 控制器的 I2Ct 端口发出一个 PBMs 任务。如果主机第二次发出 PBMs, 则 PD 控制器会忽略 DATAx 输入, 重新启动突发模式计时器, 并将指针重置为 RAM 中补丁空间的开头。如果 MODE 寄存器为 APP (表明 PD 控制器处于 APP 模式), 则主机可以拒绝 PBMs 任务。			

第 4 步: 发出 'PBMs' DATA1 后, 在 I2Ct 总线上的每个 PD 控制器上写入 CMD1 = 'PBMs'。

[0x20] + ACK (唯一地址/WR/A)

0x08 + ACK (寄存器编号/A)

0x04 (字节计数)

0x50 0x42 0x4D 0x73 (以 4 个 ASCII 字符表示的 PBMs)

第 5 步: 然后读取 CMD1 寄存器, 预期结果如下所示。

[0x20] + ACK (唯一地址/WR/A)

0x08 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x00 0x00 0x00 0x00 (可以全为 “0x00” 。)

第 6 步: 然后在每个 PD 控制器上读取 DATA1 = 0 (成功完成)。

[0x20] + ACK (唯一地址/WR/A)

0x09 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x00 0x00 0x00 0x00 (可以全为 “0x00” 。)

第 7 步: 使用 DATA1.TargetAddress 中指定的目标地址在 I2Ct 总线上写入补丁捆绑包突发数据。使用停止位终止突发数据。

[0x30] + ACK (唯一地址/WR/A)

0x01 + ACK (寄存器编号/A)

写入补丁捆绑包突发数据。下一节展示了如何使用 GUI 来生成该数据。

第 8 步: 延迟至少 500us 并在 I2Ct 总线上写入 CMD1 = 'PBMc'

[0x20] + ACK (唯一地址/WR/A)

0x08 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x50 0x42 0x4D 0x63 (以 4 个 ASCII 字符表示的 PBMc)

第 9 步 : 读取 CMD1 寄存器

[0x20] + ACK (唯一地址/WR/A)

0x08 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x00 0x00 0x00 0x00 (可以全为 “0x00” 。)

第 10 步 : 然后在每个 PD 控制器上读取 DATA1 = 0 (成功完成) 。

[0x20] + ACK (唯一地址/WR/A)

0x09 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x00 0x00 0x00 0x00 (可以全为 “0x00” 。)

第 11 步 : 主机可以读取 I2C1 寄存器位 [80] (偏移 = 14h) 的中断事件 , 以确定补丁是否已加载到器件中。

第 12 步 : 确认所有 PD 控制器上的 MODE = 'APP'

[0x20] + ACK (唯一地址/WR/A)

0x03 + ACK (寄存器编号/A)

[0x20] + ACK (唯一地址/R/A)

0x04 (字节计数)

0x41 0x50 0x50 0x20 (以 4 个 ASCII 字符表示的 APP)

4.2 生成低区二进制文件的步骤

使用 **USBCPD 应用程序自定义工具** 可以轻松地生成低区二进制文件。按照问卷选择系统配置和条件。然后导出低区二进制文件，如图 4-2 所示。生成低区二进制文件后，主机可以看到捆绑包大小如图 4-3 所示，这与节 4.1 的第 3 步相对应。

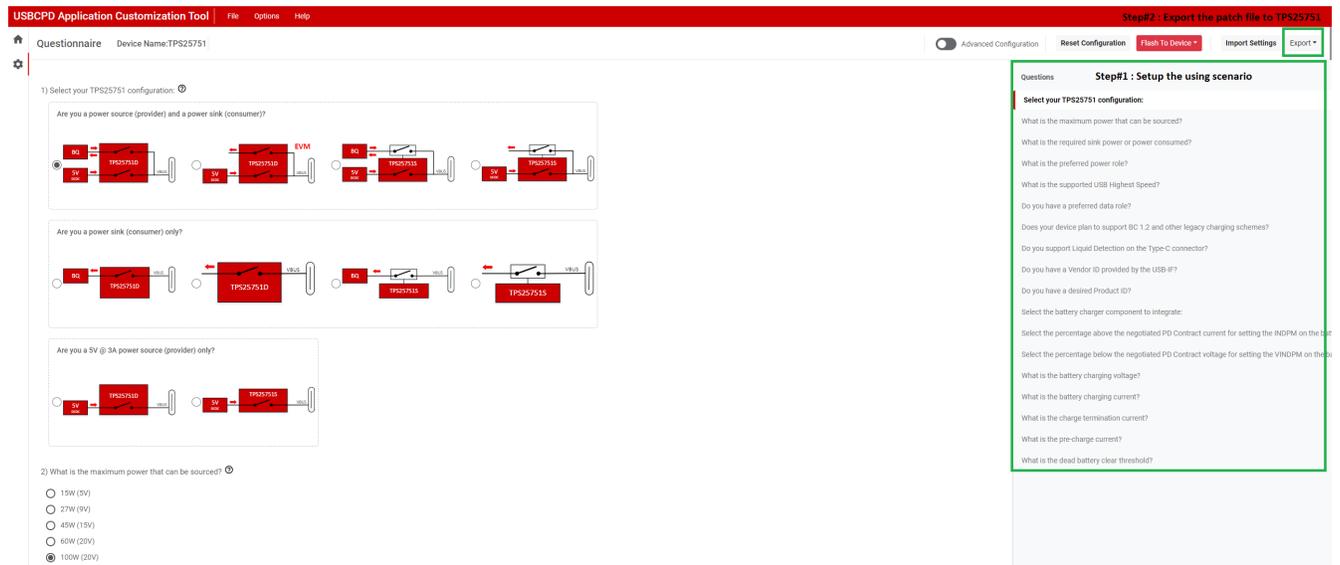


图 4-2. 用于生成低区二进制文件的设置

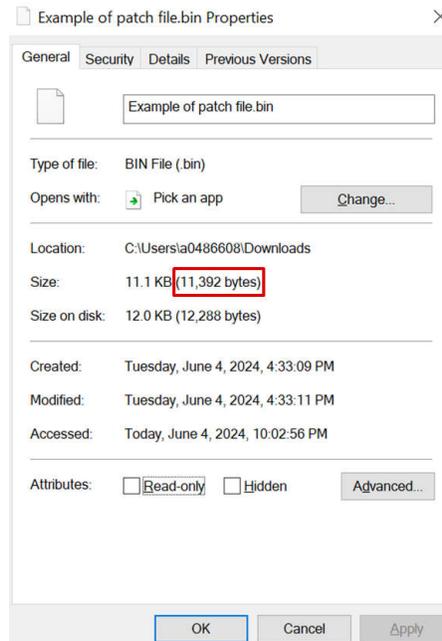


图 4-3. 捆绑包大小示例

5 电池无电配置

电池无电模式进入规则是 TPS25751 仅由 VBUS 供电，这意味着在该模式下可通过 VBUS 引脚打开 TPS25751。MODE 寄存器 (偏移 = 3h) 可以显示 BOOT，指示 PD 控制器在电池无电的情况下引导。将 EC 与 TPS25751 搭配使用时，建议将电池无电配置设置为 AlwaysEnableSink 或 NegotiateHighVoltage。

如果系统要在 TPS25751D 上存在 VIN_3V3 和 PP5V 后使 TPS25751 退出电池无电模式，请通过 I2C 使用 4CC DBfg 命令来清除 PD 的电池无电标志，让 TPS25751D 依赖 VIN_3V3 和 PP5V，而不是 VBUS。在清除电池无电标志后，PD 可以从 VIN_3V3 供电。PD 现在可以从 PP5V 供电，但无法更改端口上的活跃 PD 协议。除非收到命令，否则 PD 无法自动切换为供电方电源角色。

如果 TPS25751 首先具有 VIN_3V3 和 PP5V，则 TPS25751 不会进入电池无电模式，也不需要清除电池无电标志。

6 I2Ct IRQ 的中断事件、屏蔽和清除

对于中断事件、中断屏蔽和中断清除，请参阅 0x14h、0x16h 和 0x18h。对于 I2Ct_，请使用 0x16 中断屏蔽来使 I2Ct_ 在发生要求事件时报告低电平。例如，如果设置 0x16[3] = 1，则意味着我们屏蔽了 *插头插入或移除* 事件。如果 Type-C 端口发生 *插头插入或移除* 事件，I2Ct_ 可以报告低电平。如果系统想要将 I2Ct_ 状态恢复为高电平，则需要清除 0x18h 中的相应位。

通常，如果用户不清楚哪个中断操作对应哪个事件，可以屏蔽 0x16h 中的所有位，但通常建议 EC 仅屏蔽所需的事件，或仅屏蔽在 IRQ 置为低电平时检查的事件。当 I2Ct_ 报告低电平时，请检查 0x14h 中断事件。

7 GPIOx 功能

GPIOx 是用户常用的功能。GPIO 配置可以参考 0x5C[12:0] 寄存器。例如，如果 0x5C[12:0] = 0110000001101b，则可以从下表中查看 GPIO 类型。但在 TPS25751 中，GPIO8 和 GPIO9 不可用。如果 GPIOx 配置为输出类型，则可以使用 0x72[7:0] 和 0x72[12] 来读取 GPIOx 状态。当 GPIOx 配置为输入类型时，则无法读取状态。

表 7-1. GPIOx 配置复位值 0x5C[12:0]

GPIOx	12	11	10	9	8	7	6	5	4	3	2	1	0
数据	0	1	1	不适用	不适用	0	0	0	0	1	1	0	1
类型	I	O	O	不适用	不适用	I	I	I	I	O	O	I	O

GPIOx 可以配置为输入类型和输出类型，并具有相应的映射事件。[USBCPB 应用程序自定义工具](#)可以轻松地在低区二进制文件中生成自定义的 GPIOx 设置。例如，如果 GPIOx 需要将 GPIO0 配置为输出类型，并将事件设置为 plugevent_port1 且初始值为 0。当该事件发生时，GPIO0 会报告 1。完成设置并导出低区二进制文件后，在从 PTCH 模式切换至 APP 模式时发出该文件。

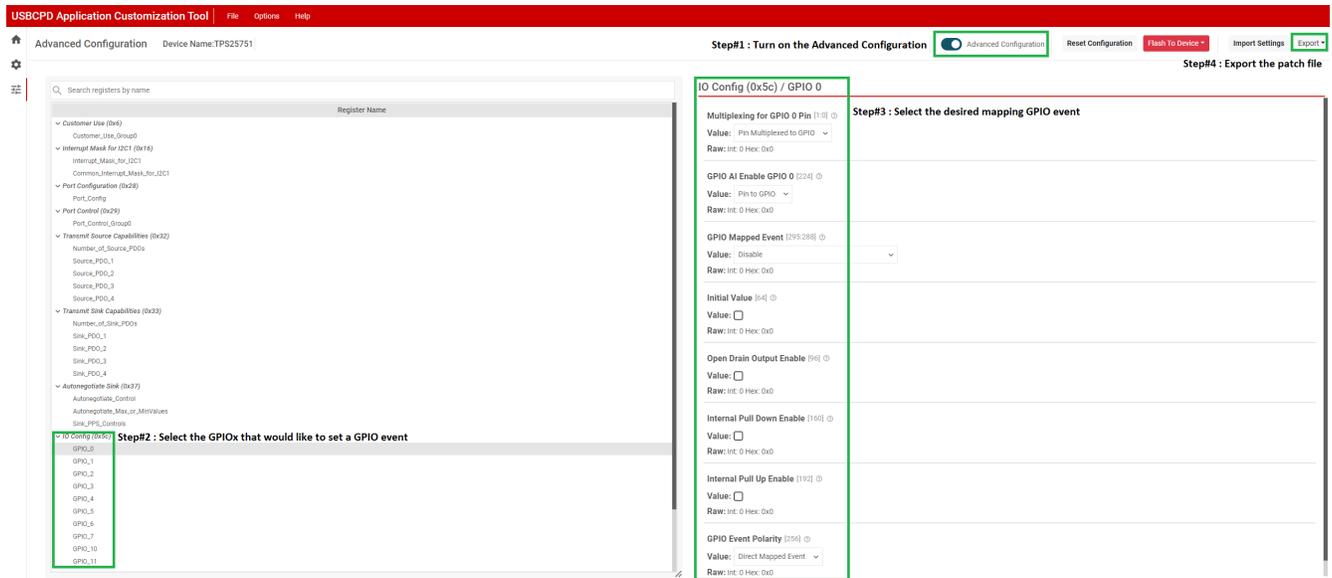


图 7-1. 使用 USBPCB 应用程序自定义工具设置 GPIOx

对于 GPIOx 类型，目前这些值由中断事件固定，无法更改类型，这意味着 TPS25751 不支持将 GPIO 用作通用输入或输出 GPIO 并读取其状态。相反，根据配置的 GPIO 事件，PD 可以自动将引脚的 GPIO 设置更改为输入或输出。

8 4CC 命令

4CC 的全名是四字符命令。要使用这些 4CC 命令，需要将 ASCII 命令转换为十六进制并写入寄存器 0x08。写入命令后，命令可以在寄存器 0x08 清除时完成。然后，用户可以检查寄存器 0x09 的 *Output DataX*，以确认任务是否成功完成。TPS25750 TRM 提供了详细的 *DataX*，通过查看任务返回代码，可判断任务是否已成功写入。

9 总结

本应用手册介绍了如何在早期启动 TPS25751。

10 参考资料

- 德州仪器 (GPIO), [GPIO](#), E2E™ 设计支持论坛。
- 德州仪器 (TI), [电池无电](#), E2E™ 设计支持论坛。
- 德州仪器 (PTCH), [从 PTCH 到 APP](#), E2E™ 设计支持论坛。
- 德州仪器 (TI), [Salae 代码](#), E2E™ 设计支持论坛。
- 德州仪器 (TI), [屏蔽](#), E2E™ 设计支持论坛。

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 © 2024，德州仪器 (TI) 公司