



Haury Zhou, Bing Huang, Seong Kim

摘要

TPS2576x-Q1 和 TPS2577x-Q1 是完全集成的 USB Type-C® 电力输送 (PD) 管理器件，适用于汽车 USB Type-C 充电应用。对于某些应用场景，客户可能需要更新器件的固件。本应用手册介绍了使用 MCU 等嵌入式器件的 TPS2576x-Q1 和 TPS2577x-Q1 固件更新过程。TPS2576x-Q1/TPS2577x-Q1 将引导和应用程序代码存储在其内部 ROM 中，并能够通过 I2C 接收来自外部主机或 EEPROM 的补丁捆绑包（包含应用程序配置和代码补丁）。

内容

1 用途和范围.....	2
2 固件引导代码简介.....	2
3 补丁捆绑包简介.....	2
4 固件更新.....	3
4.1 概述.....	3
4.2 EEPROM 固件更新.....	4
4.3 PD 控制器补丁捆绑包下载.....	8
附录 A：TVSP 引导配置设置.....	13
附录 B：使用 4CC 命令.....	14
修订历史记录.....	14

插图清单

图 4-1. 在 EEPROM 引导模式下使用主机对 EEPROM 进行编程.....	3
图 4-2. 使用 HUB/MCU 引导模式从主机直接引导.....	3
图 4-3. 区域 0（低位区域）和区域 1（高位区域）的 EEPROM 存储器组织方式.....	5
图 4-4. EEPROM 补丁捆绑包更新过程的流程图.....	7
图 4-5. 补丁捆绑包更新的流程图 - 突发模式下载.....	9
图 4-6. 补丁突发模式初始化.....	10
图 4-7. PBMs 任务命令.....	10
图 4-8. CMDx 输出 - 任务处理已完成.....	10
图 4-9. DATAx 输出 - 补丁初始化成功.....	11
图 4-10. 将补丁捆绑包生成为 C 数组.....	11
图 4-11. PBMc 任务命令.....	11
图 4-12. PBMe 任务命令.....	12
图 A-1. TVSP 设置.....	13

表格清单

表 4-1. 建议的 EEPROM.....	4
表 4-2. 4CC 任务命令集 - EEPROM 更新.....	6
表 4-3. 4CC 任务命令集 - 突发模式下载.....	8
表 A-1. TVSP 设置.....	13
表 B-1. 唯一地址接口寄存器.....	14

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 用途和范围

本应用手册详细介绍了 TPS2576x-Q1/TPS2577x-Q1 固件更新过程。TPS2576x-Q1/TPS2577x-Q1 固件包括引导代码和应用程序代码，存储在 ROM 中以执行器件功能。然后，器件可以通过 I2C 总线接受来自外部主机或 EEPROM 的补丁捆绑包（包含代码补丁和应用程序配置），并存储在 SRAM 中。可使用 TPS2576x-Q1/TPS2577x-Q1 应用自定义工具（TI 网站上的 [TPS257XX-Q1-GUI](#)）生成补丁捆绑包。要了解有关 GUI 的更多信息，请参阅 [TPS257XX-Q1-GUI 配置指南](#)。

2 固件引导代码简介

器件的固件引导代码有两个主要功能：

1. 器件初始化。
2. 补丁捆绑包加载和配置。

当器件上电并处于活动模式时，系统会启用 LDO_3V3 并发出上电复位 (POR) 信号。数字内核接收此复位信号，并从内部 ROM 加载引导代码，然后开始初始化器件设置。此初始化包括启用和复位内部寄存器、加载初始值和配置器件的 I2C 地址。引导代码还会测量 TVSP 引脚上的电阻并解码 TVSP 索引值。根据所选的引导模式，器件将接受来自外部 EEPROM 或 MCU 等主机控制器的补丁捆绑包。[附录 A](#) 详细说明了各种 TVSP 配置以及相应的器件引导顺序。

3 补丁捆绑包简介

固件补丁捆绑包有两个用途：

1. 提供代码补丁来替换应用 ROM 内的函数。
2. 提供器件的配置。

如前一节所述，固件补丁捆绑包可以在执行 POR 后使用器件的 I2C 主机接口从外部 EEPROM 或从主机加载。补丁捆绑包来自单一来源。如果由于通信错误而未成功加载配置和补丁数据，则引导代码将使用出厂器件配置。

4 固件更新

4.1 概述

固件更新是指通过外部 EEPROM 或外部主机使用补丁捆绑包来更新固件的应用程序配置和/或代码补丁。本应用手册介绍了一组 4CC ASCII 命令 (请参阅附录 B) , 当器件处于 EEPROM 引导模式时, 可以通过这些命令将补丁捆绑包编程到外部 EEPROM (请参阅图 4-1) , 或者在处于 HUB/MCU 引导模式时, 直接将补丁捆绑包下载到器件的 SRAM (请参阅图 4-2) 。

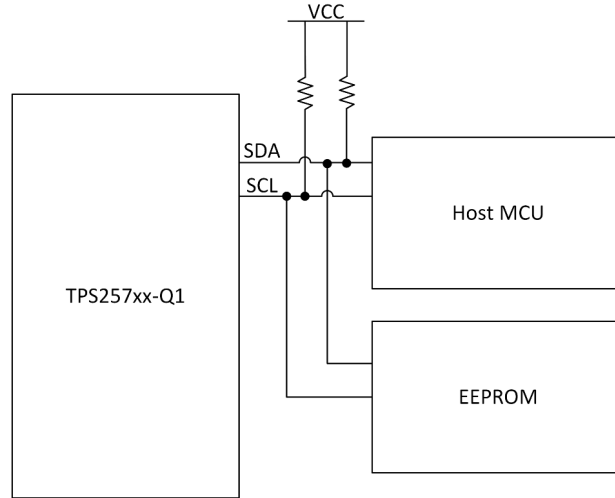


图 4-1. 在 EEPROM 引导模式下使用主机对 EEPROM 进行编程

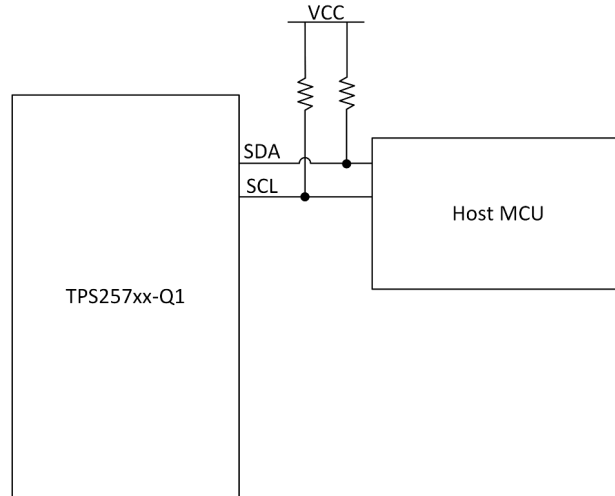


图 4-2. 使用 HUB/MCU 引导模式从主机直接引导

4.2 EEPROM 固件更新

如果 TVSP 设置为外部 EEPROM 补丁捆绑包加载，则器件上电后会尝试访问外部 EEPROM 的地址 (0x50)。如果检测到外部 EEPROM，则引导代码会首先尝试从 EEPROM 的低位区域读取补丁捆绑包。如果在读取过程的任何时候产生无效数据，器件会中止低位区域读取，并尝试从高位区域读取。如果这两个区域都包含无效的补丁捆绑包，则引导固件会继续进入补丁模式。当器件处于补丁模式时，大多数功能将受到限制，并且器件仅有几个功能可以正常工作。因此，当 TVSP 设置为 EEPROM 引导模式但未连接到 EEPROM 时，器件将进入补丁模式。

器件设计为通过其内部 LDO 稳压器 LDO_3V3 为外部 EEPROM 供电，因此用于 EEPROM 存储器的上拉电阻器必须连接到 LDO_3V3。请注意，LDO_3V3 可为外部电路供电，最大电流为 25mA。

I2C 端口支持标准模式、快速模式和快速模式增强版 I2C 接口。基本要求如下：

- 32kB (256kb)
- 7 位 I2C 地址 (0x50)
- 组织：32kB × 8 (总计 256kb)
- 活动固件映像存储在一个 16KB × 8 分区中。先前的固件映像保留在其他 16KB × 8 分区中，以确保可靠性。
- 页面大小缓冲区需要为 64 字节

表 4-1. 建议的 EEPROM

制造商	器件型号
ON-Semiconductor	CAV24C256
Microchip	24LC256
ST Micro	M24256
Rohm	BRA24T512FVT-3AM

4.2.1 EEPROM 存储器组织方式

器件的补丁捆绑包可以包含在 EEPROM NVM IC 的存储器中。本部分使用变量和相对位置来说明此数据在 EEPROM 中的组织方式。不同器件版本的存储器组织方式可能各不相同，应用程序开发人员在为器件开发 EEPROM 更新应用程序时应考虑这些差异。

本应用手册基于以下假设来说明 EEPROM 的存储器组织方式：EEPROM 专用于器件，且不与系统的其他 IC 共享。为了实现冗余，补丁捆绑包复制到两个区域：区域 0 (低位区域) 和区域 1 (高位区域)。如果直接读取 EEPROM 的整个存储器空间，将会得到包含区域头和区域的数据，这些数据会保存为 *eeeprom.bin* 文件。图 4-3 说明了器件的完整 *eeeprom.bin* EEPROM 存储器组织方式。

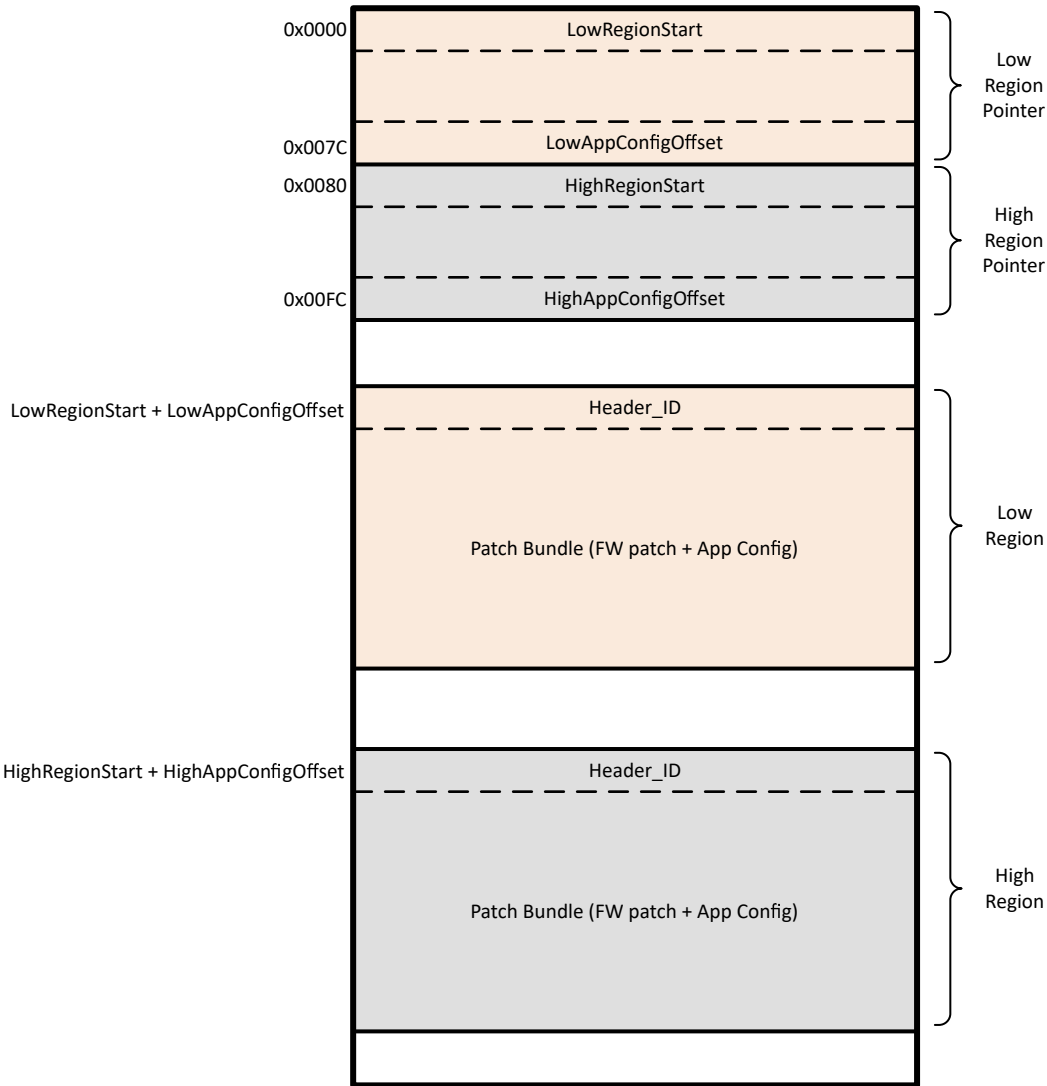


图 4-3. 区域 0 (低位区域) 和区域 1 (高位区域) 的 EEPROM 存储器组织方式

首次为平台上电时，应使用完整的 *eeeprom.bin* 对外部 EEPROM 进行编程，以便正确设置区域头。随后的 EEPROM 更新可由外部主机通过执行后续章节中详述的序列来执行。可以使用器件的应用程序自定义工具 *TPS257XX-Q1-GUI* 来生成完整的 *eeeprom.bin*。使用 GUI 生成完整的 *eeeprom.bin* 后，系统会自动设置区域 0 和区域 1 的区域偏移量。

通过使用 4CC 命令，我们可以从 MCU 等主机控制器更新 EEPROM。在图 4-3 中，前两个块是区域 0 和区域 1 的标头。每个标头需要 128B 的空间。标头用于指示补丁捆绑包的起始地址，它包含地址基址和地址偏移。标头的前四个字节是地址基址，标头的后四个字节是地址偏移。因此，补丁捆绑包的起始地址是 *地址基址加上地址偏移*。

补丁捆绑包由补丁捆绑包标头、自定义配置数据和补丁代码组成。区域 0 和区域 1 的补丁捆绑包与前面提到的完全相同。

4.2.2 EEPROM 更新 - 4CC 任务命令集

将补丁捆绑包从主机写入 EEPROM 时，需要使用表 4-2 中列出的 4CC ASCII 命令。

表 4-2. 4CC 任务命令集 - EEPROM 更新

4CC 命令的名称	ASCII	输入 DataX 长度 (以字节为单位)	输出 DataX 长度 (以字节为单位)	说明
安全刷写更新启动命令	SFWi	无	3	SFWi 使器件准备好接收即将到来的数据包。调用此任务时，PD 控制器应处于 FWUP 模式。在 FWUP 模式下，PD 控制器不应执行任何 PD 操作。
安全刷写更新数据命令	SFWd	64	3	SFWd 任务是固件更新流中的主要步骤。SFWd 为 PD 控制器提供下一个要刷写到 I2C EEPROM 中的 64 字节。
安全固件更新完成	SFWs	64	3	SFWs 任务是固件更新流中的最后一步，前提是已使用之前的 SFWx 命令为安全刷写配置了 PD 控制器。SFWs 将图像签名信息传递给 PD 控制器，以便验证之前通过 SFWd 任务接收到的数据。
无符号固件更新完成	SFWu	无	3	如果还没有为安全刷写配置 PD 控制器，则 SFWu 任务是固件更新流中的最后一步。SFWu 通知 PD 控制器，固件更新过程已完成，并使 PD 控制器在假设所有检查都通过的情况下验证映像和更改活动区域。

要执行 4CC 任务，主机应用程序应遵循以下顺序：

1. 如果 4CC 任务需要输入，应用程序应首先将输入数据写入 DATAx (如果使用 I2C1，则为 0x09；如果使用 I2C2，则为 0x11) 寄存器。
2. 应用程序随后应将 4CC 任务字符写入相应的 CMDx (如果使用 I2C1，则为 0x08；如果使用 I2C2，则为 0x10) 寄存器。
3. 应用程序应先等待，直至 CMDx 寄存器的四个字节读取以下内容：
 - 表示命令执行成功的 0x00。
 - 或表示命令执行失败的 !CMD。

应用程序可以轮询或设置并使用 *CMDxComplete* I2C 事件 (对于本应用手册，由于尚未下载补丁捆绑包，主机可以轮询 CMDx 寄存器的状态)。

如果任务成功执行，则在相关任务有输出值的情况下，主机可以继续读取 DATAx 寄存器中包含输出数据的 3 个字节内容。

4.2.3 EEPROM 补丁捆绑包更新过程

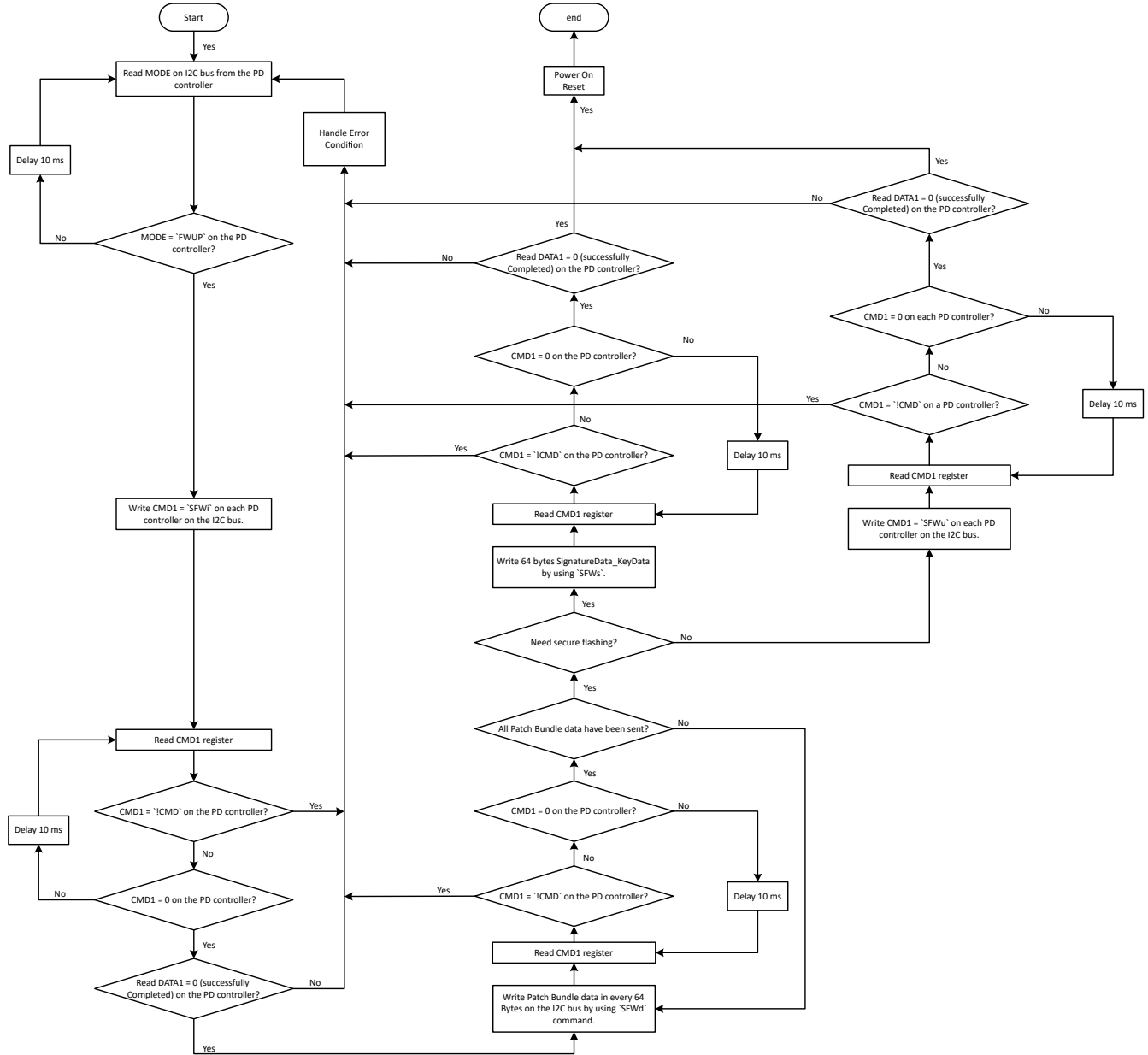


图 4-4. EEPROM 补丁捆绑包更新过程的流程图

图 4-4 展示了 EEPROM 更新过程。对于 EEPROM 固件更新，整个过程应根据 EEPROM 存储器组织方式执行两次：第一次执行是更新一个区域，第二次执行是更新另一个区域。固件会自动确定需要先更新哪个区域（区域 0 或区域 1）。以下是从主机更新 EEPROM 补丁捆绑包的过程。

1. 当器件在 FWUP 模式（外部 EEPROM 或 USB 端点引导模式）下完全正常运行时，主机可以启动闪存更新过程，并且在更新过程中应禁用端口。
2. 主机将获取有关两个区域中哪个区域是当前活动区域的信息 - 活动区域是外部 EEPROM 上器件在当前引导期间成功加载了补丁捆绑包的区域，它可以是区域 0 或区域 1。在 FWUP 模式下，默认情况下可能将活动区域设置为区域 0。
3. 主机应先尝试更新非活动区域的内容。仅当成功更新非活动区域后，主机才应尝试更新活动区域的内容。此流程确保在外部 EEPROM 上提供补丁捆绑包的冗余和有效副本，以便器件始终成功引导，从而使 EEPROM 更新过程进入失效防护。
4. 主机应实施以下序列来更新补丁捆绑包：

- 执行 **SFWI** 命令以获取有关待更新区域的信息。
- 执行 **SFWd** 命令，通过 PD 控制器将补丁捆绑包数据刷写到外部 EEPROM。补丁捆绑包数据分为多个数据页，每页有 64 字节的数据。
- 在发送补丁捆绑包数据后执行 **SFWs** 或 **SFWu** 命令，以便完成整个 EEPROM 更新操作。如果需要安全刷写，请使用 **SFWs** 来写入签名数据和密钥数据（总共 64 字节）。否则，使用 **SFWu** 来结束 EEPROM 更新。如果这是第一次执行 EEPROM 固件更新，则这两条命令都会将区域标头交换到另一个未更新的区域。

4.3 PD 控制器补丁捆绑包下载

如果 TVSP 设置为 HUB /MCU 引导模式，则器件将在执行 POR 后等待大约 1 秒，以便主机启动补丁捆绑包下载。4CC 任务命令可用于执行补丁捆绑包下载，例如突发模式下载（**PBMx** 命令，**x** 是特定命令对应的字母）。

在突发模式下载中，补丁捆绑包分为 256 字节扇区，并在一个数据传输阶段发送到 PD 控制器。

当 PD 控制器在外部 HUB/MCU 引导模式下上电或复位时，器件将在大约 850ms 后进入补丁模式（MODE (0x03) 寄存器中的“PTCH”状态），并通过将 GPIO9 拉至低电平进入 I2C 中断模式来通知主机。I2C 中断指示器件已成功进入 ReadyForPatch 状态，并且正在等待主机发送补丁捆绑包数据。因此，突发模式下载的初始步骤为：

1. 在 HUB/MCU 引导模式下将 PD 控制器上电
2. 通过读取 INT_EVENT1 寄存器 (0x14) 来确认器件是否准备好进行补丁更新。如果 INT_EVENT1 寄存器中的任何位为 1，则器件已准备好进行补丁更新。另一种方法是等到在 I2C 总线上将中断置为有效并将 GPIO9 拉至低电平。
3. 读取存储在 MODE 寄存器中的 PD 控制器模式。如果模式显示“PTCH”，则意味着器件已进入补丁就绪状态。
4. 执行与突发模式下载相对应的 4CC 任务命令。

4.3.1 补丁捆绑包下载 - 4CC 任务命令集

将补丁捆绑包直接写入 TPS257x2-Q1 时，应使用表 4-3 中列出的 4CC ASCII 命令。

表 4-3. 4CC 任务命令集 - 突发模式下载

4CC 命令的名称	ASCII	输入 DataX 长度 (以字节为单位)	输出 DataX 长度 (以字节为单位)	说明
启动补丁突发下载序列	PBMs	6	1	PBMs 任务启动补丁加载序列。 PBMs 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。
补丁突发下载完成	PBMc	无	40	PBMc 任务结束补丁加载序列。在所有补丁数据传输完毕后发送 PBMc 任务。
结束补丁突发模式下载序列	PBMe	无	1	PBMe 任务结束补丁加载序列。 PBMe 任务指示 PD 控制器完成补丁加载过程。

4.3.2 突发模式补丁下载过程

以下是详细说明了突发模式下载过程的流程图。

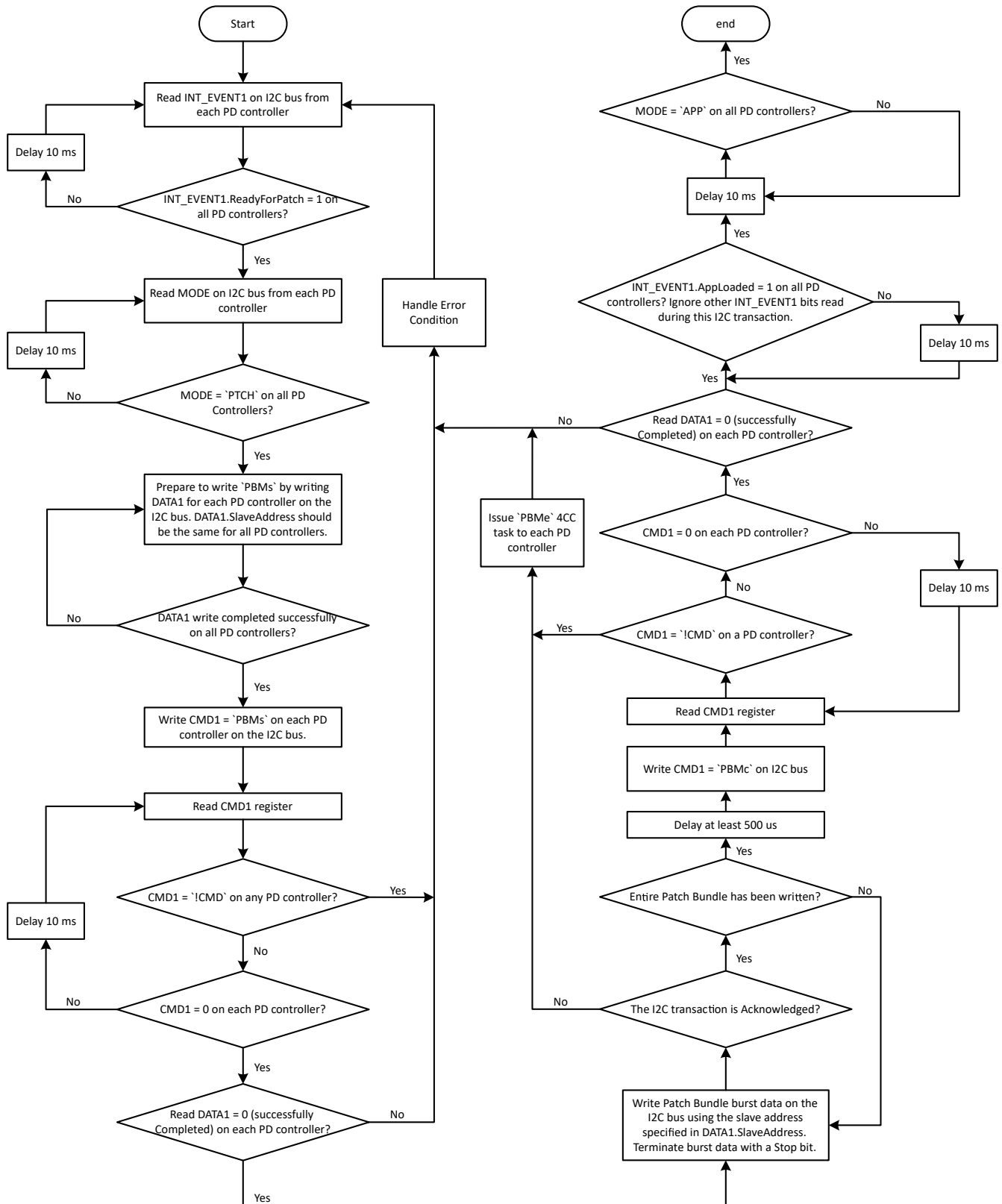


图 4-5. 补丁捆绑包更新的流程图 - 突发模式下载

图 4-5 展示了突发模式下载的过程。在主机确认器件已进入 *PTCH* 模式后，主机应实施以下序列来下载补丁捆绑包：

1. 主机初始化固件，通过向 *DATAx* (如果使用 *I2C1*，则为 *0x09*；如果使用 *I2C2*，则为 *0x11*) 寄存器写入 6 个字节的数据，为 *PBMx* 加载序列做好准备，并确定补丁捆绑包的内容：
 - a. 首先发送 *0x06*，告诉 PD 控制器写入了一个 6 字节的有效载荷。
 - b. 字节 0 至 3 为补丁捆绑包大小。在图 4-6 中，补丁捆绑包大小为 *0x3500* 或 13568 字节。字节 4 是要为数据传输分配的 *DATAx.SlaveAddress*。PD 控制器的 *0x00* 或器件的 *I2C* 从器件地址 (*0x22/0x26* 或 *0x23/0x27*) 无效。图 4-6 展示了已选择和使用的随机地址 *0x35*。
 - c. 字节 5 是突发模式超时值 (LSB 为 100ms)。必须使用非零值，并建议始终使用 *0x32*，这样会提供 5 秒的窗口来完成突发模式补丁更新。

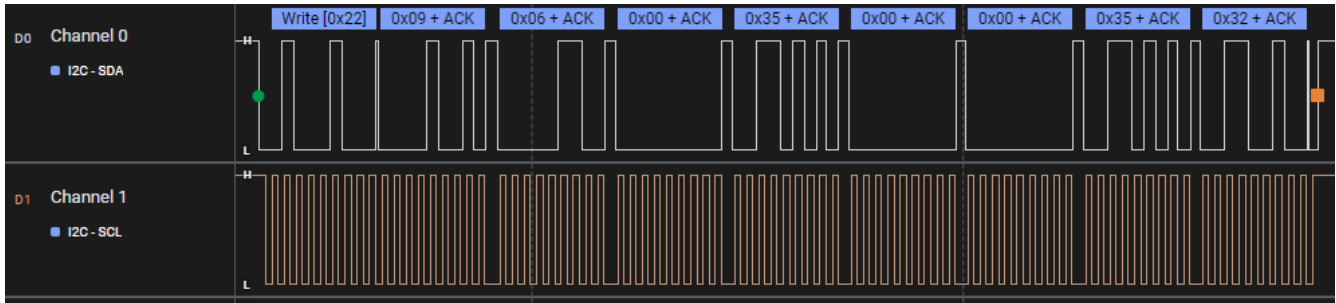


图 4-6. 补丁突发模式初始化

2. 主机应通过向 *CMDx* (如果使用 *I2C1*，则为 *0x08*；如果使用 *I2C2*，则为 *0x10*) 寄存器发送 4CC ASCII *PBMx* 任务命令来启动突发模式补丁下载过程。首先发送 *0x04*，告诉 PD 控制器总共将写入 4 个字节。

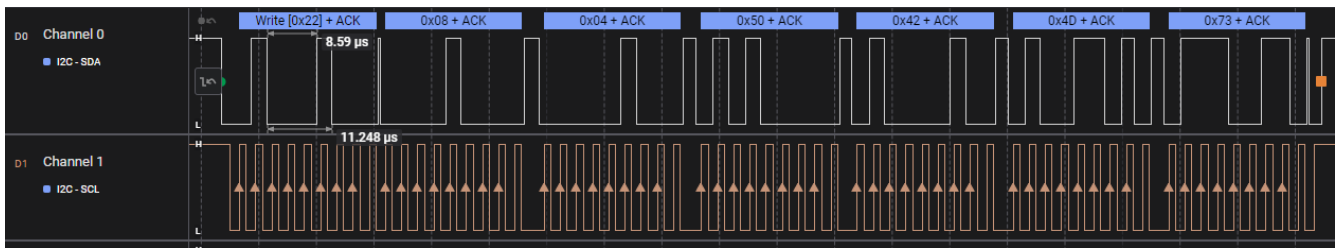


图 4-7. *PBMx* 任务命令

3. 读取并轮询 *CMDx* 寄存器，直到字节 1 为 *0x00*，指示任务处理已完成。

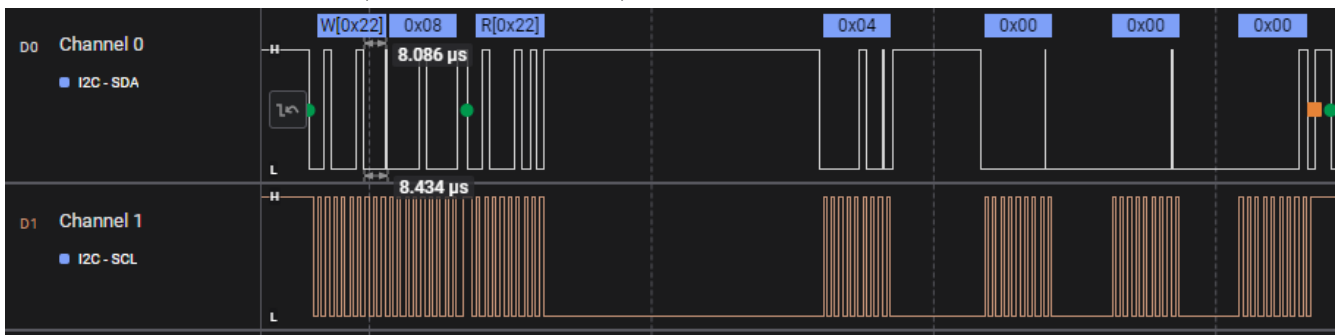


图 4-8. *CMDx* 输出 - 任务处理已完成

4. 读取并轮询 *DATAx* 寄存器，直至字节 1 为 *0x00*，指示步骤 1 的补丁初始化成功。

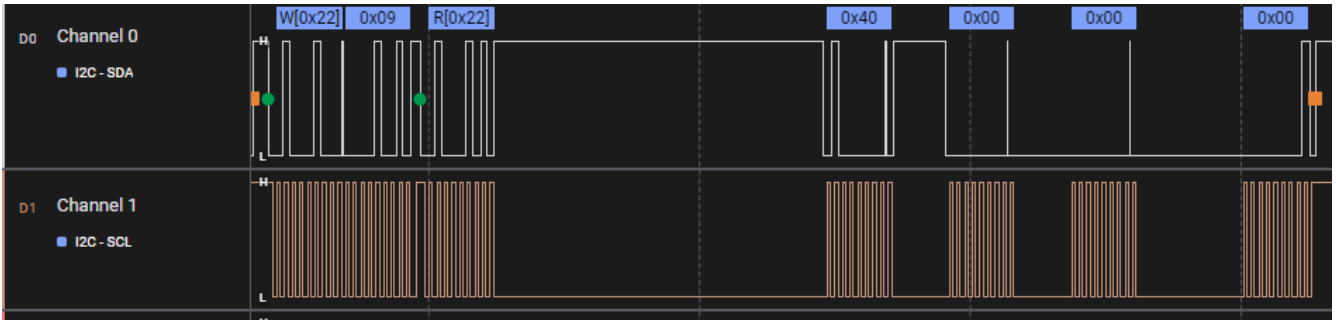


图 4-9. DATAx 输出 - 补丁初始化成功

- 以大小为 256 字节的数据包，将补丁捆绑包数据发送到在步骤 1 中配置的 DATAx.SlaveAddress。可以使用 [TPS257XX-Q1-GUI](#) (从 v1.2.0 开始提供) 将补丁捆绑包生成为 C 样式数组。从“Build GUI FLASH Image”菜单中选择“SAVE LOW REGION BINARY”。然后选择 C 头文件源文件作为格式，并点击“SAVE”。此文件还将包含补丁捆绑包大小。

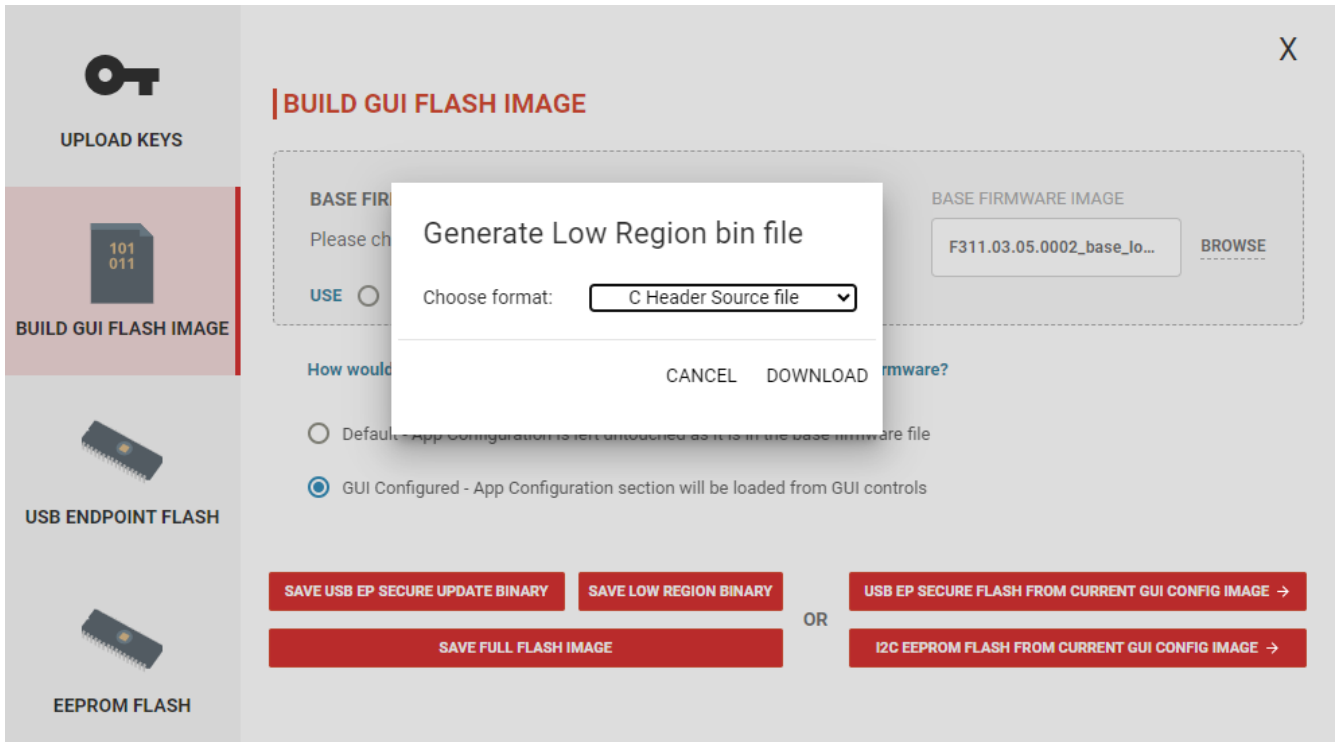


图 4-10. 将补丁捆绑包生成为 C 数组

- 成功写入补丁捆绑包数据后，等待 500us，然后将 PBMc 任务写入 CMDx 寄存器，以便完成补丁加载序列。在此步骤之后，确保返回来写入到 PD 控制器的原始 I2C 从器件地址。

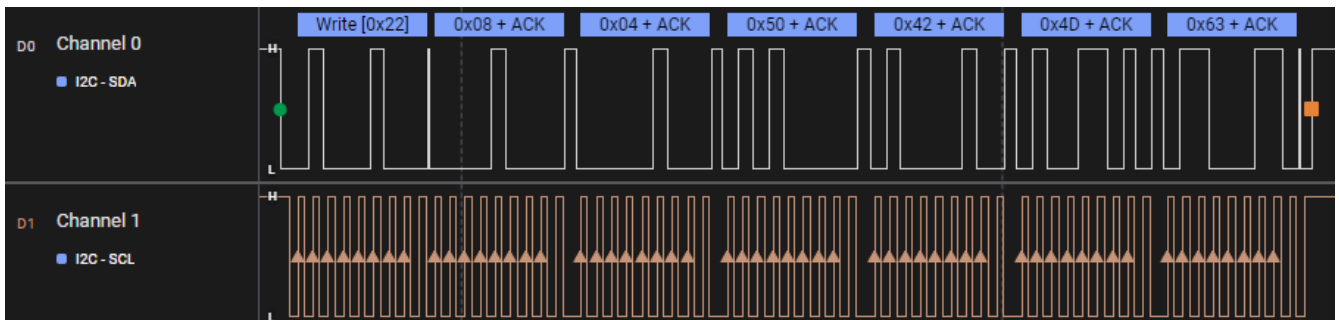


图 4-11. PBMc 任务命令

- 读取并轮询 CMDx 寄存器，直到字节 1 为 0x00，指示任务处理已完成。

8. 读取并轮询 DATAx 寄存器，直至字节 1 为 0x00，指示器件补丁捆绑包下载成功。I2C 中断 (即 GPIO9) 将在大约 110ms 后释放。
9. 将 PBM_e 任务写入 CMDx 寄存器，以便结束补丁加载序列并进入 “APP” 模式 (可选) 。

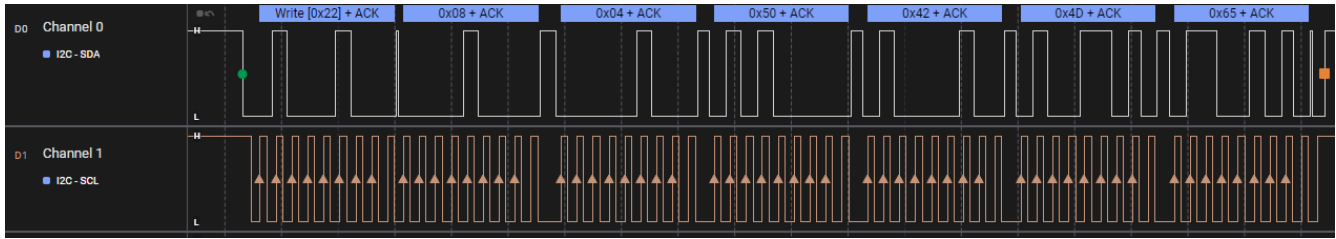


图 4-12. PBM_e 任务命令

10. 读取 MODE 寄存器以检查器件是否处于 APP 模式，这表明 PD 控制器收到所有补丁和应用程序配置数据并在应用程序固件中完全正常运行。

附录 A : TVSP 引导配置设置

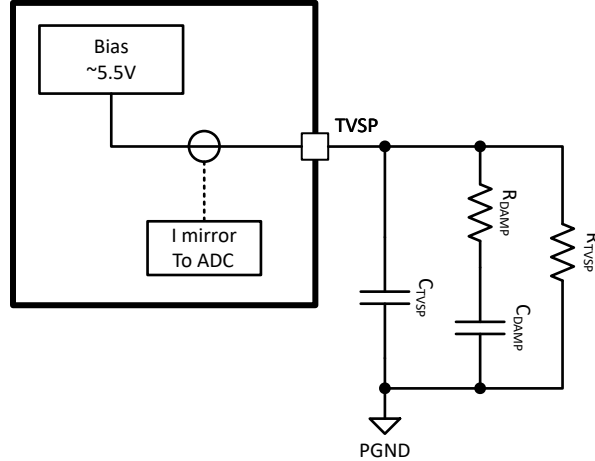


图 A-1. TVSP 设置

TVSP 引脚是多功能引脚，但对于引导过程， R_{TVSP} 是确定引导行为的电阻器。上电时，TVSP 引脚和 PGND 之间的电阻决定了引导方法、USB PD 端口 I2C 地址和 I2C 逻辑阈值。请参阅 R_{TVSP} 配置设置。图 A-1 展示了引脚硬件连接。在器件初始化和引导期间，通常在上电后的 4 秒内，输入电压必须高于 7.6V，以便确保 TVSP 引脚适当偏置至 5.5V。引导完成后，器件可以在整个 VIN 范围内运行。下面的表 A-1 展示了 R_{TVSP} 的设置。有关 TVSP 引脚的更多详细信息可在数据表中找到。

表 A-1. TVSP 设置

$R_{TVSP}(k\Omega)$	TVSP 索引	ADC 值	I2C 从器件端口地址 (A/B)	I2C 逻辑 (V_{DD})	引导模式
开路	0	$\leq 10(0 \times 0A)$	$0 \times 22 0 \times 26$	3.3V	EEPROM
93.1	1	$\leq 24(0 \times 18)$	$0 \times 23 0 \times 27$	3.3V	外部 HUB MCU
47.5	2	$\leq 42(0 \times 2A)$	$0 \times 22 0 \times 26$	1.8V	EEPROM
29.4	3	$\leq 63(0 \times 3F)$	$0 \times 23 0 \times 27$	1.8V	外部 HUB MCU
20.0	4	$\leq 89(0 \times 59)$	$0 \times 23 0 \times 27$	3.3V	EEPROM
14.7	5	$\leq 119(0 \times 77)$	$0 \times 22 0 \times 26$	3.3V	外部 HUB MCU
11.0	6	$\leq 156(0 \times 9C)$	$0 \times 23 0 \times 27$	1.8V	EEPROM
8.45	7	$\leq 201(0 \times C9)$	$0 \times 22 0 \times 26$	1.8V	外部 HUB MCU
6.65	8	$\leq 255(0 \times FF)$	$0 \times 22 0 \times 26$	3.3V	USB 端点

附录 B：使用 4CC 命令

4CC (4 字节字符代码) 命令是一组命令，可简化 PD 控制器常用功能的使用。它让用户可以发送单个任务，用于管理更复杂的子例程以及为其进行特定功能的寄存器写入。

4CC 命令结构与软件函数类似，其中包含输入参数 (输入 DATAx)、函数调用 (将 4CC 命令写入 CMDx 寄存器) 和返回的输出 (输出 DATAx)。

通过将 4 字符命令转换为 ASCII 来获得写入 CMDx 寄存器的 4CC 命令。您可以使用 ASCII 转换器来帮助转换代码 (例如，4CC 命令 *PBMs* 转换为 *50 42 4D 73*)。请记住，这些命令区分大小写。

CMD1 (0x08) 寄存器通过 I2C1 总线向其写入 4CC 命令。任何数据 (输入 DATAx、输出 DATAx) 都写入 DATA1 (0x09) 寄存器或从中读取。I2C2 总线还有第二组寄存器，分别位于 0x10 (对于 CMD2) 和 0x11 (对于 DATA2)。表 B-1 提供了详细的唯一地址接口说明。

表 B-1. 唯一地址接口寄存器

寄存器地址	寄存器名称	访问	说明
0x03	MODE	RO	指示端口的运行状态。
0x08	CMD1	RW	主命令接口的命令寄存器。如果将无法识别的命令写入此寄存器，则会将其替换为 4CC 值 "!CMD"。
0x09	DATA1	RW	主命令接口 (CMD1) 的数据寄存器。
0x10	CMD2	RW	辅助命令接口的命令寄存器。如果将无法识别的命令写入此寄存器，则会将其替换为 4CC 值 "!CMD"。
0x11	DATA2	RW	用于辅助命令接口 (CMD2) 的数据寄存器。
0x14	INT_EVENT1	RO	I2C_EC_IRQ 的中断事件位字段。如果此寄存器中的任何位为 1，则会将 I2C_EC_IRQ 引脚下拉至低电平。
0x15	INT_EVENT2	RO	I2C2s_IRQ 的中断事件位字段。如果此寄存器中的任何位为 1，则会将 I2C2s_IRQ 引脚下拉至低电平。

修订历史记录

Changes from Revision * (December 2022) to Revision A (August 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

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