

TPS25750 主机接口技术参考手册

Technical Reference Manual



Literature Number: ZHCU943A
NOVEMBER 2020 – REVISED JULY 2022



请先阅读.....	7
关于本手册.....	7
命名惯例.....	7
术语表.....	7
相关文档.....	7
支持资源.....	7
1 引言.....	9
1.1 引言.....	9
1.1.1 用途和范围.....	9
1.2 PD 控制器主机接口说明.....	10
1.2.1 概述.....	10
1.2.2 寄存器和字段表示法.....	10
1.3 唯一地址接口.....	10
1.3.1 唯一地址接口协议.....	10
1.3.2 唯一地址接口寄存器.....	12
2 唯一地址接口寄存器详细说明.....	15
2.1 0x03 MODE 寄存器.....	15
2.2 0x0D DEVICE_CAPABILITIES 寄存器.....	16
2.3 0x14 - 0x19 INT_EVENT、INT_MASK、INT_CLEAR 寄存器.....	17
2.4 0x1A STATUS 寄存器.....	19
2.5 0x26 POWER_PATH_STATUS 寄存器.....	21
2.6 0x29 PORT_CONTROL 寄存器.....	22
2.7 0x2D BOOT_STATUS 寄存器.....	23
2.8 0x30 RX_SOURCE_CAPS 寄存器.....	24
2.9 0x31 RX_SINK_CAPS 寄存器.....	25
2.10 0x32 TX_SOURCE_CAPS 寄存器.....	26
2.11 0x33 TX_SINK_CAPS 寄存器.....	28
2.12 0x34 ACTIVE_CONTRACT_PDO 寄存器.....	30
2.13 0x35 ACTIVE_CONTRACT_RDO 寄存器.....	31
2.14 0x3F POWER_STATUS 寄存器.....	32
2.15 0x40 PD_STATUS 寄存器.....	33
2.16 GPIO 事件.....	35
2.17 0x69 TYPEC_STATE 寄存器.....	37
2.18 0x70 SLEEP_CONFIG 寄存器.....	39
2.19 0x72 GPIO_STATUS 寄存器.....	40
3 4CC 任务详细说明.....	41
3.1 概述.....	41
3.2 PD 消息任务.....	42
3.2.1 'SWSk' - PD PR_Swap 到受电方.....	42
3.2.2 'SWSr' - PD PR_Swap 到供电方.....	42
3.2.3 'SWDF' - PD DR_Swap 到 DFP.....	43
3.2.4 'SWUF' - PD DR_Swap 到 UFP.....	43
3.2.5 'GSKC' - PD 获取受电方能力.....	44
3.2.6 'GSrC' - PD 获取供电方能力.....	44
3.2.7 'SSrC' - PD 发送供电方能力.....	45
3.3 补丁捆绑包更新任务.....	46

3.3.1 'PBMs' - 启动补丁突发模式下载序列.....	46
3.3.2 'PBMc' - 补丁突发模式下载完成.....	47
3.3.3 'PBMe' - 结束补丁突发模式下载序列.....	50
3.3.4 补丁突发模式示例.....	50
3.3.5 'GO2P' - 转到补丁模式.....	56
3.4 系统任务.....	57
3.4.1 'DBfg' - 清除电池电量耗尽标志.....	57
3.4.2 'I2Cr' - I2C 读取事务.....	58
3.4.3 'I2Cw' - I2C 写入事务.....	58
4 用户参考.....	59
4.1 PD 控制器应用程序自定义.....	59
4.2 加载补丁捆绑包.....	59
5 修订历史记录.....	62

插图清单

图 1-1. I2C 读取/写入协议关键字.....	10
图 1-2. I2C 唯一地址写入寄存器协议.....	11
图 1-3. I2C 唯一地址读取寄存器协议.....	11
图 4-1. 将补丁捆绑包通过 I2Cs 总线同时推送到多个 PD 控制器的流程.....	60
图 4-2. 分成两个事务时的补丁捆绑包突发数据的协议.....	61

表格清单

表 1-1. 唯一地址接口寄存器.....	12
表 1-2. 唯一地址接口任务.....	14
表 2-1. 0x03 MODE 寄存器.....	15
表 2-2. 0x03 MODE 寄存器位字段定义.....	15
表 2-3. 器件模式说明.....	15
表 2-4. 0x0D DEVICE_CAPABILITIES 寄存器.....	16
表 2-5. 0x0D DEVICE_CAPABILITIES 寄存器位字段定义.....	16
表 2-6. 0x14 - 0x19 INT_EVENTX、INT_MASKX、INT_CLEARX 寄存器.....	17
表 2-7. 0x14 - 0x19 INT_EVENTX、INT_MASKX、INT_CLEARX 寄存器位字段定义.....	17
表 2-8. 0x1A STATUS 寄存器.....	19
表 2-9. 0x1A STATUS 寄存器位字段定义.....	19
表 2-10. 0x26 POWER_PATH_STATUS 寄存器.....	21
表 2-11. 0x26 POWER_PATH_STATUS 寄存器位字段定义.....	21
表 2-12. 0x29 PORT_CONTROL 寄存器.....	22
表 2-13. 0x29 PORT_CONTROL 寄存器位字段定义.....	22
表 2-14. 0x2D BOOT_STATUS 寄存器.....	23
表 2-15. 0x2D BOOT_STATUS 寄存器位字段定义.....	23
表 2-16. 0x30 RX_SOURCE_CAPS 寄存器.....	24
表 2-17. 0x30 RX_SOURCE_CAPS 寄存器位字段定义.....	24
表 2-18. 0x31 RX_SINK_CAPS 寄存器.....	25
表 2-19. 0x31 RX_SINK_CAPS 寄存器位字段定义.....	25
表 2-20. 0x32 TX_SOURCE_CAPS 寄存器.....	26
表 2-21. 0x32 TX_SOURCE_CAPS 寄存器位字段定义.....	26
表 2-22. 第一个 PDO.....	27
表 2-23. 其他 PDO.....	27
表 2-24. 0x33 TX_SINK_CAPS 寄存器.....	28
表 2-25. 0x33 TX_SINK_CAPS 寄存器位字段定义.....	28
表 2-26. 第一个 PDO.....	28
表 2-27. 其他 PDO.....	29
表 2-28. 0x34 ACTIVE_CONTRACT_PDO 寄存器.....	30
表 2-29. 0x34 ACTIVE_CONTRACT_PDO 寄存器位字段定义.....	30
表 2-30. 0x35 ACTIVE_CONTRACT_RDO 寄存器.....	31
表 2-31. 0x35 ACTIVE_CONTRACT_RDO 寄存器位字段定义.....	31
表 2-32. 0x3F POWER_STATUS 寄存器.....	32

表 2-33. 0x3F POWER_STATUS 寄存器位字段定义.....	32
表 2-34. 0x40 PD_STATUS 寄存器.....	33
表 2-35. 0x40 PD_STATUS 寄存器位字段定义.....	33
表 2-36. GPIO 事件.....	35
表 2-37. 0x69 TYPEC_STATE 寄存器.....	37
表 2-38. 0x69 TYPEC_STATE 寄存器位字段定义.....	37
表 2-39. 0x70 SLEEP_CONFIG 寄存器.....	39
表 2-40. 0x70 SLEEP_CONFIG 寄存器位字段定义.....	39
表 2-41. 0x72 GPIO_STATUS 寄存器.....	40
表 2-42. 0x72 GPIO_STATUS 寄存器位字段定义.....	40
表 3-1. 标准任务响应.....	41
表 3-2. 'SWSk' - PD PR_Swap 到受电方.....	42
表 3-3. 'SWSr' - PD PR_Swap 到供电方.....	42
表 3-4. 'SWDF' - PD DR_Swap 到 DFP.....	43
表 3-5. 'SWUF' - PD DR_Swap 到 UFP.....	43
表 3-6. 'GSkC' - PD 获取受电方能力.....	44
表 3-7. 'GSrC' - PD 获取供电方能力.....	44
表 3-8. 'SSrC' - PD 发送供电方能力.....	45
表 3-9. 'PBMs' - 启动补丁突发下载序列.....	46
表 3-10. 'PBMc' - 补丁突发下载完成.....	47
表 3-11. 'PBMe' - 补丁突发模式退出.....	50
表 3-12. 'GO2P' - 强制 PD 控制器恢复到 'PTCH' 模式并等待 I2C 上的补丁.....	56
表 3-13. 'DBfg' - 清除电池电量耗尽标志.....	57
表 3-14. 'I2Cr' - 在 I2Cm 上执行 I2C 读取事务.....	58
表 3-15. 'I2Cw' - 在 I2Cm 上执行 I2C 写入事务.....	58
表 4-1. 在不同工作模式下使用从器件地址。.....	59

This page intentionally left blank.



关于本手册

本手册介绍了 TPS2575x 系列 USB Type-C® 和 PD 控制器器件支持的特性和外设。文中的内容涵盖 GPIO 事件、I2C 中断、主机接口 (4CC) 命令和寄存器读取/写入说明。本手册还介绍了通过 PBMx 命令对 PD 控制器配置进行编程的补丁突发模式过程。

命名惯例

本文档使用以下惯例。

- 十六进制数可以用后缀 **h** 或前缀 **0x** 显示。例如，以下数字是十六进制的 40 (十进制的 64) : 40h 或 0x40。
- 本文档中的寄存器如图所示，并在表中进行介绍。
 - 每个寄存器图都显示一个矩形，该矩形被划分为代表寄存器字段的字段。每个字段都标有其位名称、其起始位和结束位编号、其读/写属性及以下默认复位值。图例解释了用于属性的符号。
 - 寄存器图中的保留位可以有多种含义之一：
 - 未在器件上实现
 - 保留用于未来的器件扩展
 - 保留用于 TI 测试
 - 不支持的器件保留配置
 - 向保留位写入非默认值可能会导致意外行为，应避免此类行为。

术语表

TI 术语表 本术语表列出并解释了术语、首字母缩略词和定义。

相关文档

- Universal Serial Bus Specification, Revision 2.0, April 27, 2000 plus ECN and Errata. http://www.usb.org/developers/docs/usb20_docs/
- Battery Charging Specification, Revision 1.2, December 7, 2010 plus Errata.
- Universal Serial Bus 3.1 Specification, Revision 1.0, July 26, 2013 and ECNs approved through August 11, 2014. www.usb.org/developers/docs
- USB Power Delivery Specification Revision 3.0, Version 1.2, June 21, 2018 www.usb.org/developers/docs
- USB Type-C Cable and Connector Specification Revision 1.3, July 14, 2017. www.usb.org/developers/docs

支持资源

TI E2E™ 支持论坛是工程师的重要参考资料，可直接从专家获得快速、经过验证的解答和设计帮助。搜索现有解答或提出自己的问题可获得所需的快速设计帮助。

链接的内容由各个贡献者“按原样”提供。这些内容并不构成 TI 技术规范，并且不一定反映 TI 的观点；请参阅 TI 的《使用条款》。

商标

TI E2E™ is a trademark of Texas Instruments.

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。

This page intentionally left blank.



1.1 引言

1.1.1 用途和范围

备注

本节适用于高级用户，此处列出的特性仅为可选特性。本文档对于简单的电源应用而言不是必需的。您的系统中需要 **EC** 或主机来实现本文档中所述的某些特性。

本文档将介绍 TPS25750、Type-C 端口开关/电力输送 (PD) 控制器器件的主机接口。本文档适用于 TPS25750 的补丁 F509.04.02。

1.2 PD 控制器主机接口说明

1.2.1 概述

PD 控制器提供一个 I2C 从器件。I2Cs 从器件旨在连接到嵌入式控制器 (EC)。

主机接口定义了如何从 I2C 从器件端口和所有从器件地址访问寄存器。客户利用 PD 控制器上的 ADCIN1 和 ADCIN2 引脚来选择从器件地址 #1。另请参阅表 4-1，了解有关从器件地址的更多详细信息。

主机接口向这些 I2C 接口的主器件提供有关 PD 控制器的一般状态信息、控制 PD 控制器的能力、USB Type-C 端口的状态，以及与连接的器件（端口伙伴）和/或电缆插头之间通过 USB PD 消息进行的通信。所有使用唯一 I2C 地址进行通信的主机接口称为唯一地址接口。

PD 控制器支持基于寄存器的唯一地址接口。节 1.3.2 列出了唯一地址接口寄存器，章节 2 提供了唯一地址接口寄存器的详细说明。

协议图的关键字信息位于 SMBus 规范 2.0 版中，并在图 1-1 中部分重复。

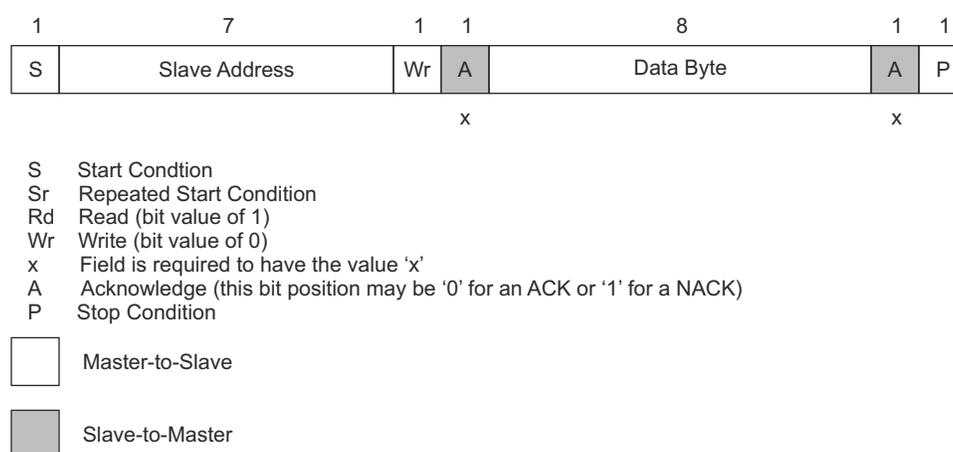


图 1-1. I2C 读取/写入协议关键字

1.2.2 寄存器和字段表示法

在本文档中，寄存器名称使用 ALL CAPS（全大写）表示法，字段名称使用 CamelBack（大小写混合）表示法。例如，TX_SOURCE_CAPS 表示寄存器 0x32，TX_SOURCE_CAPS.numValidPDos 表示该寄存器字节 1 中的特定字段。

一些寄存器具有相同的定义，但在不同的寄存器地址有多个实例。这些寄存器如下。

- INT_EVENT1 (0x14)
- INT_MASK1 (0x16)
- INT_CLEAR1 (0x18)
- CMD1 (0x08)
- DATA1 (0x09)

1.3 唯一地址接口

1.3.1 唯一地址接口协议

唯一地址接口允许 I2C 主器件和单个 PD 控制器之间进行复杂的交互。I2C 从器件唯一地址用于接收或响应主机接口协议命令。图 1-2 和图 1-3 分别展示了写入和读取协议。寄存器写入期间使用的字节数可能比实际写入的字节数长。换言之，主器件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，主器件可以在读取所有 N 个字节之前发出停止位。

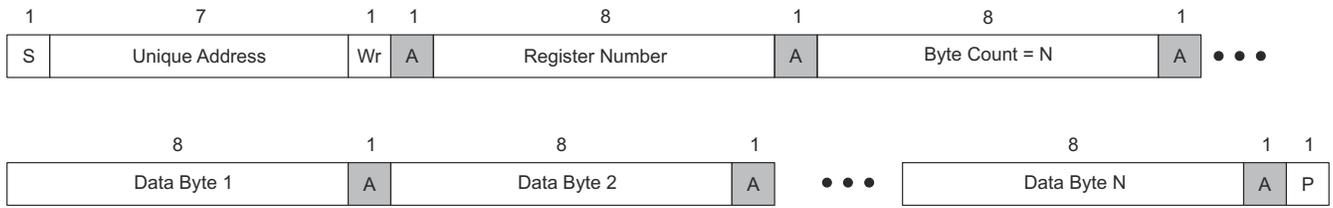


图 1-2. I2C 唯一地址写入寄存器协议

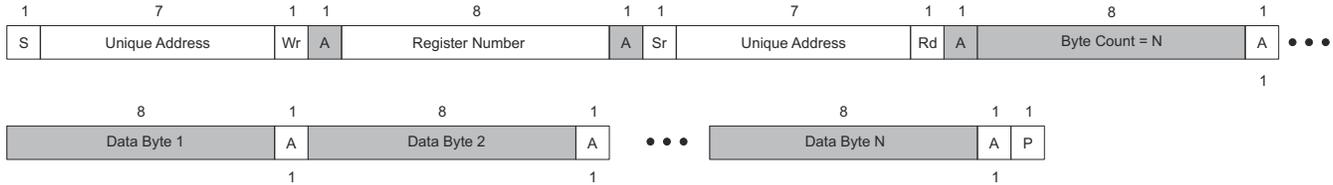


图 1-3. I2C 唯一地址读取寄存器协议

1.3.2 唯一地址接口寄存器

PD 控制器支持表 1-1 中提供的唯一地址接口寄存器（唯一地址寄存器）。除非另有说明，否则 2 或 4 字节寄存器是小端字节序（数据字节 1 中为最低有效字节）。此处定义了使用四字符代码（4CC）的寄存器，其中第一个字符对应于数据字节 1 的 ASCII 值，第二个字符对应于数据字节 2 的 ASCII 值，依此类推。任何少于 4 个字符的 4CC 代码均以空格（0x20）填充尾部。

对于标记为非唯一的寄存器，主机可以使用任一从器件地址读取该寄存器，并读回相同的值。对于标记为非唯一的寄存器，主机可以使用任一从器件地址写入该寄存器。

在下面的寄存器映射中，仅实现了显示的寄存器地址。所有其他寄存器地址都是必须忽略的只读保留寄存器。

表 1-1. 唯一地址接口寄存器

寄存器编号 (1)	寄存器名称	访问	数据字节数	每个端口均 唯一	说明
0x02	保留				
0x03	MODE	RO	4	否	指示端口的运行状态。在某些模式下，PD 控制器的功能受限。请参阅节 2.1
0x04	类型	RO	4	否	默认响应是 'I2C'（注意第 4 个字符是空格）。
0x06	CUSTUSE	RO	8	是	这 8 个字节分配给客户按需使用。PD 控制器不使用此寄存器。此寄存器可能会在应用程序自定义期间更改。
0x07	保留				
0x08	CMD1	RW	4	是	主命令接口的命令寄存器。在初始化期间和成功处理每个命令后，由 PD 控制器清除为 0x0000_0000。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 "ICMD"。
0x09	DATA1	RW	64	是	主命令接口 (CMD1) 的数据寄存器。
0x0A-0x0C	保留				
0x0D	DEVICE_CAPABILITIES	RO	4	否	所支持特性的说明。请参阅节 2.2
0x0E	保留				
0x0F	VERSION	RO	4	否	二进制编码的十进制版本号，引导加载程序/应用程序代码版本。表示为 VVVV.MM.RR，去掉前导 0。例如，65794d（十进制）-> 0x00010102 -> 0001.01.02 -> 1.1.2（版本）。版本信息以小端字节序格式返回，即字节 1 = RR，字节 2 = MM 等。16 位字段 RR 用作供电方能力扩展消息和受电方能力扩展消息中的固件版本。
0x10-0x13	保留				
0x14	INT_EVENT1	RO	11	是	I2Cs_IRQ 的中断事件位字段。如果此寄存器中的任何位为 1，则会将其 I2Cs_IRQ 引脚下拉至低电平。请参阅节 2.3
0x15	保留				
0x16	INT_MASK1	RW	11	是	INT_EVENT1 的中断屏蔽位字段。INT_EVENT1 中的某个位在此寄存器中清除后无法设置。请参阅节 2.3
0x17	保留				

表 1-1. 唯一地址接口寄存器 (continued)

寄存器编号 (1)	寄存器名称	访问	数据字节数	每个端口均 唯一	说明
0x18	INT_CLEAR1	RW	11	是	INT_EVENT1 的中断清除位字段。此寄存器中设置的位从 INT_EVENT1 中清除。请参阅节 2.3
0x19	保留				
0x1A	STATUS	RO	5	是	非中断事件的状态位字段。请参阅节 2.4
0x1B-0x25	保留				
0x26	POWER_PATH_STATUS	RO	5	否	电源路径状态。请参阅节 2.5
0x27-0x28	保留				
0x29	PORT_CONTROL	RW	4	是	影响系统策略的配置位。这些位在正常运行期间可能会更改，用于控制相应的端口。PD 控制器在写入时不会立即采取操作。对此寄存器所做的更改将在下次调用相应的策略时生效。由应用程序自定义进行初始化。请参阅节 2.6
0x2A-0x2C	保留				
0x2D	BOOT_STATUS	RO	5	否	引导过程的详细状态。此寄存器提供有关 PD 控制器引导标志、客户 OTP 配置和器件版本的详细信息。请参阅节 2.7
0x2E	BUILD_DESCRIPTION	RO	49	否	Build 说明。这是用于唯一标识自定义 build 信息的 ASCII 字符串。
0x2F	DEVICE_INFO	RO	40	否	器件信息。这是一个 ASCII 字符串，包含 PD 控制器的硬件和固件版本信息。
0x30	RX_SOURCE_CAPS	RO	29	是	收到的供电方能力。此寄存器存储通过 BMC 收到的最新供电方能力消息。请参阅节 2.8
0x31	RX_SINK_CAPS	RO	29	是	收到的受电方能力。此寄存器存储通过 BMC 收到的最新受电方能力消息。请参阅节 2.9
0x32	TX_SOURCE_CAPS	RW	31	是	要发送的供电方能力。此寄存器存储用于传出供电方能力 PD 消息的 PDO 和设置。由应用程序自定义进行初始化。请参阅节 2.10
0x33	TX_SINK_CAPS	RW	29	是	要发送的受电方能力。此寄存器存储用于传出受电方能力 USB PD 消息的 PDO。由应用程序自定义进行初始化。请参阅节 2.11
0x34	ACTIVE_CONTRACT_PDO	RO	6	是	有效合约的电源数据对象。此寄存器存储当前显式 USB PD 合约的 PDO 数据，如果没有合约，则为全零。请参阅节 2.12
0x35	ACTIVE_CONTRACT_RDO	RO	4	是	有效合约的电源数据对象。此寄存器存储当前显式 USB PD 合约的 RDO，如果没有合约，则为全零。请参阅节 2.13
0x36-0x3E	保留				
0x3F	POWER_STATUS	RO	2	是	有关连接功率的详细信息。此寄存器报告有关连接功率的状态。请参阅节 2.14
0x40	PD_STATUS	RO	4	是	PD 和 Type-C 状态机的状态。此寄存器包含有关 PD 消息状态和 Type-C 状态机的详细信息。请参阅节 2.15

表 1-1. 唯一地址接口寄存器 (continued)

寄存器编号 (1)	寄存器名称	访问	数据字节数	每个端口均 唯一	说明
0x41-0x68	保留				
0x69	TYPEC_STATE	RO	4	是	包含两个 CCn 引脚的当前状态。请参阅节 2.17
0x6A-0x71	保留				
0x72	GPIO_STATUS	RO	8	否	捕获所有 GPIO 引脚的状态和设置。请参阅节 2.19
0x73-0x7E	保留				

(1) 任何未显示的寄存器编号均保留用于专有用途。

PD 控制器会执行表 1-1 中定义的唯一地址接口任务。

表 1-2. 唯一地址接口任务

命令 4CC	类型	命令摘要	参考
PBMs	补丁捆绑包更新	启动补丁突发下载序列	请参阅节 3.3.1
PBMc	补丁捆绑包更新	补丁突发下载完成	请参阅节 3.3.2
PBMe	补丁捆绑包更新	补丁突发模式退出	请参阅节 3.3.3
GO2P	补丁捆绑包更新	强制 PD 控制器恢复到 'PTCH' 模式并等待 I2C 上的补丁。	请参阅节 3.3.5
GSkC	PD 消息	PD 获取受电方能力	请参阅节 3.2.5
GSrC	PD 消息	PD 获取供电方能力	请参阅节 3.2.6
SSrC	PD 消息	PD 发送供电方能力	请参阅节 3.2.7
SWDF	PD 消息	PD DR_Swap 到 DFP	请参阅节 3.2.3
SWSk	PD 消息	PD PR_Swap 到受电方	请参阅节 3.2.1
SWSr	PD 消息	PD PR_Swap 到供电方	请参阅节 3.2.2
SWUF	PD 消息	PD DR_Swap 到 UFP	请参阅节 3.2.4
DBfg	系统	清除电池电量耗尽标志	请参阅节 3.4.1
I2Cr	系统	在 I2Cm 上执行 I2C 读取事务。	请参阅节 3.4.2
I2Cw	系统	在 I2Cm 上执行 I2C 写入事务。	请参阅节 3.4.3



2.1 0x03 MODE 寄存器

MODE 寄存器是一个可返回 4 个 ASCII 字符的 32 位寄存器。

表 2-1. 0x03 MODE 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x03	MODE	RO	4	否	请参阅下表。

表 2-2. 0x03 MODE 寄存器位字段定义

位	名称	说明
31:0	mode	以 4 个 ASCII 字符描述的模式。更多详细信息，请参阅下表。

表 2-3. 器件模式说明

模式寄存器值	说明	I2Cs_IRQ 可生效	寄存器可用性
'APP'	PD 控制器在应用程序固件中完全正常工作。	是	所有寄存器完全可用。
'BOOT'	器件在电池电量耗尽的情况下引导。	否	受限 (请参阅此表结尾处的列表)
'PTCH'	器件处于补丁模式。	是	受限 (请参阅此表结尾处的列表)
任何其他值	PD 控制器功能受限。	否	否

当器件处于 'BOOT' 或 'PTCH' 模式时，主机不得读取或写入大多数寄存器。只有以下寄存器在 'BOOT' 和 'PTCH' 模式下可用：

- 4CC 补丁命令
- MODE (0x03)
- TYPE (0x04)、VERSION (0x0F)
- CMD1 (0x08)、DATA1 (0x09)
- DEVICE_CAPABILITIES (0x0D)
- INT_EVENT1 (0x14)、INT_MASK1 (0x16) 和 INT_CLEAR1 (0x18)
- BOOT_STATUS (0x2D)
- DEVICE_INFO (0x2F)

2.2 0x0D DEVICE_CAPABILITIES 寄存器

表 2-4. 0x0D DEVICE_CAPABILITIES 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x0D	DEVICE_CAPABILITIES	RO	4	否	

表 2-5. 0x0D DEVICE_CAPABILITIES 寄存器位字段定义

位	名称	说明
31:8	保留	
7	I2CmLevel	I2Cm 端口所需的上拉电压。
		0b 1.8V 或 3.3V
		1b 3.3V
6:5	BC1p2Supported	BC 1.2 支持能力。
		00b 不支持。
		01b 支持仅供电方。
		10b 保留
		11b
4:3	保留	
2	UsbPdCapability	USB 电力输送能力。
		0b 支持
		1b 不支持。
1:0	PowerRole	电源角色能力。
		00b 支持供电方和受电方这两种角色 (DRP)。
		01b 仅供电方。
		10b
		11b 仅供电方。

2.3 0x14 - 0x19 INT_EVENT、INT_MASK、INT_CLEAR 寄存器

此寄存器的字节 1 到 10 是特定于端口的字节，但字节 11 是 PD 控制器中所有端口通用的字节。

表 2-6. 0x14 - 0x19 INT_EVENTX、INT_MASKX、INT_CLEARX 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x14	INT_EVENT1	RO	11	是	0
0x15	保留				
0x16	INT_MASK1	RW	11	是	由应用程序配置进行初始化
0x17	保留				
0x18	INT_CLEAR1	RW	11	是	0
0x19	保留				

表 2-7. 0x14 - 0x19 INT_EVENTX、INT_MASKX、INT_CLEARX 寄存器位字段定义

位	名称	说明
字节 11：补丁状态（所有从端口通用）		
7:3	保留	
2	I2CMasterNACKed	I2C 主器件上的一个事务为 NACKed。
1	ReadyForPatch	器件准备好接收来自主机的补丁捆绑包。
0	PatchLoaded	补丁已加载到器件。
字节 9-10：		
15:2	保留	
1	TXMemBufferEmpty	发送存储器缓冲区为空。
字节 5-8：		
31:15	保留	
14	ErrorUnableToSource	供电方无法将电压提高到合约协商的电压。
13:12	保留	
11	PlugEarlyNotification	已检测到连接但未消除抖动。
10	SnkTransitionComplete	此事件仅在供电方模式下 (PD_STATUS.PresentPDRole = 1b) 发生。它发生在将接受消息发送给请求消息之后 tSrcTransition (ms)，即发送 PS_RDY 消息之前。
9	保留	
7	ErrorMessageData	收到错误消息。
6	ErrorProtocolError	收到来自伙伴器件的意外消息。
5	保留	
4	ErrorMissingGetCapMessage	伙伴器件未响应发送的 Get_Sink_Cap 或 Get_Source_Cap 消息。
3	ErrorPowerEventOccurred	VBUS 上发生了 OVP 或 ILIM 事件。或者发生了 TSD 事件。
2	ErrorCanProvideVoltageOrCurrentLater	USB PD 供电方可以提供可接受的电压和电流，但目前不会。已发送或接收“等待”消息。
1	ErrorCannotProvideVoltageOrCurrent	USB PD 供电方无法提供可接受的电压和/或电流。向受电方发送了拒绝消息或收到来自受电方的“能力不匹配”。
0	ErrorDeviceIncompatible	设置为 1 时，连接了规格版本不兼容的 USB PD 器件。或者伙伴器件不支持 USB PD。
字节 1-4：		
30	CMDComplete	每当 CMD 寄存器中的非零值设置为零或 !CMD 时置位。
29	保留	
28	保留	
27	PDStatusUpdate	每当 PD_STATUS 寄存器 (0x40) 的内容发生变化时置位。
26	StatusUpdate	每当 STATUS 寄存器 (0x1A) 的内容发生变化时置位。

表 2-7. 0x14 - 0x19 INT_EVENTX、INT_MASKX、INT_CLEARX 寄存器位字段定义 (continued)

位	名称	说明
25	保留	
24	PowerStatusUpdate	每当 POWER_STATUS 寄存器 (0x3F) 的内容发生变化时置位。
23	PPswitchChanged	每当 POWER_PATH_STATUS 寄存器 (0x26) 的内容发生变化时置位。
22	保留	
21	UsbHostPresentNoLonger	当 STATUS.UsbHostPresent 转换为 11b 以外的任何内容时置位。
20	UsbHostPresent	当 STATUS.UsbHostPresent 转换为 11b 时置位。
19	保留	
18	DRSwapRequested	端口伙伴请求进行 DR 交换。
17	PRSwapRequested	端口伙伴请求进行 PR 交换。
16	保留	
15	保留	
14	SourceCapMsgRcvd	收到来自端口伙伴的供电方能力消息时生效。
13	NewContractAsProv	已接受来自远端器件的 RDO 并且 PD 控制器是供电方。在发送 PS_RDY 消息后生效。请参阅 ACTIVE_CONTRACT_PDO 寄存器 (0x34) 和 ACTIVE_CONTRACT_RDO 寄存器 (0x35)，了解详细信息。
12	NewContractAsCons	远端供电方已接受 PD 控制器作为受电方发送的 RDO。请参阅 ACTIVE_CONTRACT_PDO 寄存器 (0x34) 和 ACTIVE_CONTRACT_RDO 寄存器 (0x35)，了解详细信息。
11:6	保留	
5	DRSwapComplete	已完成数据角色交换。请参阅 STATUS 寄存器 (0x1A) 和 PD_STATUS 寄存器 (0x40)，了解端口状态。
4	PRSwapComplete	已完成电源角色交换。请参阅 STATUS 寄存器 (0x1A) 和 PD_STATUS 寄存器 (0x40)，了解端口状态。
3	PlugInsertOrRemoval	USB 插头状态已更改。更多关于插头的详细信息，请参阅状态寄存器。
2	保留	
1	PDHardReset	已执行 PD 硬复位。更多信息，请参阅 PD_STATUS.HardResetDetails。
0	保留	

2.4 0x1A STATUS 寄存器

表 2-8. 0x1A STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x1A	STATUS	RO	5	是	尽管在断开和连接期间许多位都发生了变化，但切记不要完全复位。

表 2-9. 0x1A STATUS 寄存器位字段定义

位	名称	说明		
字节 5 :				
7:0	保留			
字节 1-4 :				
30:28	保留			
27	Bist	指示是否有 BIST 过程正在进行。		
		0b 没有正在进行的 BIST。		
		1b BIST 正在进行。也可通过 MODE 寄存器 (0x03) 读数 'BIST' 来指示该状态。		
26	保留			
25:24	ActingAsLegacy	指示 PD 控制器何时进入类似于传统 (非 PD) 器件的模式。PD 控制器可能需要大约 10 秒来确定其是连接到传统供电方还是受电方。		
		00b PD 控制器未处于传统 (非 PD) 模式		
		01b PD 控制器类似于传统受电方。它不会响应 USB PD 消息流量。		
		10b PD 控制器类似于传统供电方。它不会响应 USB PD 消息流量。		
23:22	UsbHostPresent	USB 主机连接状态。		
		00b 不存在主机。这意味着目前没有远端器件提供 VBUS，或者 PD 控制器的电源角色是供电方。		
		01b VBUS 由端口伙伴提供，该端口伙伴是不支持 USB 通信的 PD 器件。		
		10b VBUS 由端口伙伴提供，该端口伙伴不是 PD 器件。		
21:20	VbusStatus	指示 VBUS 的当前状态。		
		00b 处于 vSafe0V (小于 0.8V)。		
		01b 处于 vSafe5V (4.75V 至 5.5V)。		
		10b 在预期限值范围内。这些限值是根据 USB PD 协商值确定的。		
19:7	保留			
		6	DataRole	PD 控制器数据角色。仅在建立连接后有效。
		0b 上行端口 (UFP)		
		1b 下行端口 (DFP)		
5	PortRole	PD 控制器 CCx 终端的当前状态。也指示连接后的 PD 控制器电源角色。在 Unattached.* 状态转换期间不切换此位。		
		0b PD 控制器充当受电方角色。这意味着 CCx 下拉有效或端口被禁用/断开。		
4	PlugOrientation	1b PD 控制器为供电方 (CCx 上拉有效)。		
		插头方向指示器。在端口方向已知时指示该方向 (需要连接)。		
		0b 顶面朝上 (将 CC 插入 CC1)。也可能是未知方向或端口可能被禁用/断开。		
		1b 顶面朝下方向 (将 CC 插入 CC2)。		

表 2-9. 0x1A STATUS 寄存器位字段定义 (continued)

位	名称	说明
3:1	ConnState	已连接插头的详细信息。
		000b 无连接。
		001b 已禁用端口。
		010b 音频连接 (Ra/Ra)。
		011b 调试连接 (Rd/Rd)。
		100b 无连接，检测到 Ra (有 Ra 但没有 Rd)。
		101b 保留 (可用于 Rp/Rp 调试连接)。
		110b 存在连接，未检测到 Ra。可以是 Rd (但没有 Ra) 或检测到 Rp (先前未检测到 Ra)，包括以 Attached.SNK 状态连接的 PD 控制器。
		111b 存在连接，检测到 Ra。可以是检测到 Rd (和 Ra) 或检测到 Rp (先前检测到 Ra - 如果 PD 控制器作为供电方启动并随后切换为受电方)。
0	PlugPresent	插头状态
		0b 未连接插头。
		1b 已连接插头。

2.5 0x26 POWER_PATH_STATUS 寄存器

表 2-10. 0x26 POWER_PATH_STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x26	POWER_PATH_STATUS	RO	5	否	0

表 2-11. 0x26 POWER_PATH_STATUS 寄存器位字段定义

位	名称	说明
字节 4-5 : PP 和 PP_CABLE 过流		
15:14	PowerSource	指示当前 PD 控制器电源。注意：因为电池电量耗尽标志强制 PD 控制器由 VBUS 供电，所以在设置此标志后只有 10b 有效。任何其他设置都表示未设置电池电量耗尽标志。
		00b 保留。
		01b PD 控制器由 VIN_3V3 供电。
		10b PD 控制器由 VBUS 供电。设置电池电量耗尽标志。
	11b 保留。	
13:12	保留	保留。
10	PP_CABLE1_Overcurrent	PP_CABLE1 过流指示器。PP_CABLE1 (VCONN) 上存在过流情况时生效。
9:6	保留	保留。
4	PP1_Overcurrent	PP_5V1 过流指示器。PP1 开关 (PP_5V1) 上存在过流情况时生效。
3:0	保留	
字节 1-3 : PP 和 PP_CABLE 开关状态		
14:12	PP3switch	指示 PP3 (PP_EXT1) 的当前状态。
		0h PP3 开关已禁用。
		1h PP3 开关当前由于故障而被禁用。该开关是系统输出。
		2h 保留。
		3h PP3 开关已启用 (系统输入)。
	4h-7h 保留。	
8:6	PP1switch	指示 PP1 开关 (PP_5V1) 的当前状态。
		0h PP1 开关已禁用。
		1h PP1 开关当前由于故障而被禁用。该开关是系统输出。
		2h PP1 开关已启用 (系统输出)。
	3h-7h 保留。	
5:4	保留	
1:0	PP_CABLE1_switch	指示 PP_CABLE1 开关的当前状态。
		00b PP_CABLE1 开关已禁用。
		01b PP_CABLE1 开关当前已禁用。PD 控制器正在等待 PP5V 引脚变为高电平。
		10b PP_CABLE1 开关 CC1 已启用 (系统输出)。
	11b PP_CABLE1 开关 CC2 已启用 (系统输出)。	

2.6 0x29 PORT_CONTROL 寄存器

表 2-12. 0x29 PORT_CONTROL 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x29	PORT_CONTROL	RW	4	是	由应用程序配置进行初始化

表 2-13. 0x29 PORT_CONTROL 寄存器位字段定义

位	名称	说明
31:30	ChargerDetectEnable	配置要检测的传统充电器类型。
		00b 不检测任何传统充电器。
		01b 检测 BC 1.2 充电器。
		10b 保留，请勿使用
		11b 检测 BC 1.2 和专有传统充电器。
29	保留	
28:26	ChargerAdvertiseEnable	配置要模拟的传统充电器类型。
		0h 不模拟任何传统充电器。这意味着只有 SDP 模式将用于传统器件。
		1h 仅 BC 1.2 CDP。
		2h 仅 BC 1.2 DCP。
		3h 保留。
		4h 保留。
		5h DCP 自动 1 (2.7V 和 DCP)
		6h DCP 自动 2 (1.2V、2.7V 和 DCP)
7h 保留。		
25	保留	
24	Resistor15kPresent	配置 D+ 和 D- 终端。如果 D+ 和 D- 上存在 15kΩ 下拉电阻 (启用 USB2.0 Host Phy 下拉电阻)，则使此位生效。不得用于 DCP 模式和 DCP 自动模式。
		0b 系统没有 15kΩ 下拉电阻。PD 控制器将根据 BC 1.2 的需求应用 15kΩ 下拉电阻。
		1b 系统具有 15kΩ 下拉电阻。PD 控制器不会应用 15kΩ 下拉电阻。
23:2	保留	
1:0	TypeCCurrent	Type-C 电流广播。如果供电方角色未启用并激活，则会忽略此设置。在显式的 USB PD 合约期间 (此情况下 R _p 值用于根据 USB PD 规范的要求避免冲突) 也会忽略此设置。请注意，当 PP5V 为低电平时，固件将只使用默认的 Type-C 电流，而不管此字段中的值如何。
		00b USB 默认电流。
		01b 1.5A。
		10b 3.0A。
		11b 保留。

2.7 0x2D BOOT_STATUS 寄存器

表 2-14. 0x2D BOOT_STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x2D	BOOT_STATUS	RO	5	否	上下文相关 (从不复位)

表 2-15. 0x2D BOOT_STATUS 寄存器位字段定义

位	名称	说明
字节 5 : 修订版本 ID		
7:0	REV_ID	PD 控制器的修订版本 ID。
字节 1-4 : 引导标志 (视为 32 位小端字节序值)		
31:29	PatchConfigSource	补丁配置的来源。此字段表示已成功加载的配置补丁的来源。
		0h 未加载任何配置。
		1h-4h 保留。
		5h 已从 EEPROM 加载配置。
		6h 已从 I2C 加载配置。
		7h 保留。
28	保留	
27	保留	
24	保留	
19	MasterTSD	主热关断指示器。如果 PD 控制器在主热传感器引起复位后重新启动，此位将生效。
18	保留	
16:12	保留	
11	保留	
10	patchdownloaderr	发生补丁下载错误时生效。
9:4	保留	
3	I2cEepromPresent	EEPROM 存在指示器。当引导期间在 I2Cm 上发现 EEPROM 器件时，此位将生效。
2	DeadBatteryFlag	电池电量耗尽标志指示器。当 PD 控制器在电池电量耗尽模式下引导后，此位将生效。
1	保留	保留。
0	PatchHeaderErr	当补丁捆绑包头发生错误时生效。

2.8 0x30 RX_SOURCE_CAPS 寄存器

表 2-16. 0x30 RX_SOURCE_CAPS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x30	RX_SOURCE_CAPS	RO	29	是	断开或硬复位时清除。

表 2-17. 0x30 RX_SOURCE_CAPS 寄存器位字段定义

位	名称	说明
字节 26-29 : PDO #7 (视为 32 位小端字节序值)		
31:0	SourcePdo7	收到的第七个供电方能力 PDO。
字节 22-25 : PDO #6 (视为 32 位小端字节序值)		
31:0	SourcePdo6	收到的第六个供电方能力 PDO。
字节 18-21 : PDO #5 (视为 32 位小端字节序值)		
31:0	SourcePdo5	收到的第五个供电方能力 PDO。
字节 14-17 : PDO #4 (视为 32 位小端字节序值)		
31:0	SourcePdo4	收到的第四个供电方能力 PDO。
字节 10-13 : PDO #3 (视为 32 位小端字节序值)		
31:0	SourcePdo3	收到的第三个供电方能力 PDO。
字节 6-9 : PDO #2 (视为 32 位小端字节序值)		
31:0	SourcePdo2	收到的第二个供电方能力 PDO。
字节 2-5 : PDO #1 (视为 32 位小端字节序值)		
31:0	SourcePdo1	收到的第一个供电方能力 PDO。
字节 1 : 标头		
7:3	保留	
2:0	numValidPDos	此寄存器中有效 PDO 的数量。每个 PDO 为 4 个字节 (最多 7 个)。

2.9 0x31 RX_SINK_CAPS 寄存器

表 2-18. 0x31 RX_SINK_CAPS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x31	RX_SINK_CAPS	RO	29	是	断开或硬复位时清除。

表 2-19. 0x31 RX_SINK_CAPS 寄存器位字段定义

位	名称	说明
字节 26-29 : PDO #7 (视为 32 位小端字节序值)		
31:0	SinkPdo7	收到的第七个受电方能力 PDO。
字节 22-25 : PDO #6 (视为 32 位小端字节序值)		
31:0	SinkPdo6	收到的第六个受电方能力 PDO。
字节 18-21 : PDO #5 (视为 32 位小端字节序值)		
31:0	SinkPdo5	收到的第五个受电方能力 PDO。
字节 14-17 : PDO #4 (视为 32 位小端字节序值)		
31:0	SinkPdo4	收到的第四个受电方能力 PDO。
字节 10-13 : PDO #3 (视为 32 位小端字节序值)		
31:0	SinkPdo3	收到的第三个受电方能力 PDO。
字节 6-9 : PDO #2 (视为 32 位小端字节序值)		
31:0	SinkPdo2	收到的第二个受电方能力 PDO。
字节 2-5 : PDO #1 (视为 32 位小端字节序值)		
31:0	SinkPdo1	收到的第一个受电方能力 PDO。
字节 1 : 标头		
7:3	保留	
2:0	numValidPDos	此寄存器中有效 PDO 的数量。每个 PDO 为 4 个字节 (最多 7 个)。

2.10 0x32 TX_SOURCE_CAPS 寄存器

PD 控制器将发送写入此寄存器的供电方能力，而不对其进行验证（但会限制电流，具体请参阅下文）。用户负责按照 USB PD 要求正确写入此寄存器。PD 控制器将只使用第一个 TXSourceNumPDOs PDO，主机可以在配置期间写入多个 PDO，然后动态地写入 TXSourceNumPDOs 以更改广播的 PDO。如果此寄存器已更改，主机随后必须发出 4CC 命令 'SSrC'。这将导致 PD 控制器重新加载此 TX 供电方能力寄存器。

PD 控制器将读取电缆的载流能力，并限制每个 PDO 中的最大电流，确保符合电缆的 VBUS 电流处理能力。

表 2-20. 0x32 TX_SOURCE_CAPS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x32	TX_SOURCE_CAPS	RW	31	是	由应用程序配置进行初始化

表 2-21. 0x32 TX_SOURCE_CAPS 寄存器位字段定义

位	名称	说明
字节 28-31 : PDO #7 (视为 32 位小端字节序值)		
31:0	TXSourcePDO7	第七个供电方能力 PDO 的内容。请参阅表 2-23。
字节 24-27 : PDO #6 (视为 32 位小端字节序值)		
31:0	TXSourcePDO6	第六个供电方能力 PDO 的内容。请参阅表 2-23。
字节 20-23 : PDO #5 (视为 32 位小端字节序值)		
31:0	TXSourcePDO5	第五个供电方能力 PDO 的内容。请参阅表 2-23。
字节 16-19 : PDO #4 (视为 32 位小端字节序值)		
31:0	TXSourcePDO4	第四个供电方能力 PDO 的内容。请参阅表 2-23。
字节 12-15 : PDO #3 (视为 32 位小端字节序值)		
31:0	TXSourcePDO3	第三个供电方能力 PDO 的内容。请参阅表 2-23。
字节 8-11 : PDO #2 (视为 32 位小端字节序值)		
31:0	TXSourcePDO2	第二个供电方能力 PDO 的内容。请参阅表 2-23。
字节 4-7 : PDO #1 (视为 32 位小端字节序值)		
31:0	TXSourcePDO1	第一个供电方能力 PDO 的内容。请参阅表 2-22。
字节 2-3 : 每个 PDO 的电源路径配置。		
15:14	保留	
13:12	PowerPathForPDO7	配置用于 PDO7 的 PP。与 PowerPathForPDO2 格式相同。
11:10	PowerPathForPDO6	配置用于 PDO6 的 PP。与 PowerPathForPDO2 格式相同。
9:8	PowerPathForPDO5	配置用于 PDO5 的 PP。与 PowerPathForPDO2 格式相同。
7:6	PowerPathForPDO4	配置用于 PDO4 的 PP。与 PowerPathForPDO2 格式相同。
5:4	PowerPathForPDO3	配置用于 PDO3 的 PP。与 PowerPathForPDO2 格式相同。
3:2	PowerPathForPDO2	配置用于 PDO2 的 PP。
		00b 保留。
		01b 保留。
1:0	PowerPathForPDO1	配置用于 PDO1 的 PP。
		00b 对此 PDO 使用 PP_5V1。
		10b 对此 PDO 使用 PP_EXT1。
字节 1 : 标头		
7:3	保留	
2:0	numValidPDOs	此寄存器中有效 PDO 的数量。每个 PDO 为 4 个字节 (最多 7 个)。

此寄存器中的 PDO 遵循 USB PD 规范中的定义。为方便起见，此处对其进行了复制，但如需了解有关每个字段的更多详细信息，请参阅 USB PD 规范。

表 2-22. 第一个 PDO

位	说明
31:30	电源类型，应始终设置为 00b (固定电源)。
29	双角色电源，这将由 PORT_CONTRL 寄存器中的 ProcessSwapToSink、ProcessSwapToSource、InitiateSwapToSink 和 InitiateSwapToSource 字段的逻辑“或”结果覆盖。
28	支持 USB 挂起模式。
27:26	保留
25	双角色数据，这将由 PORT_CONTRL 寄存器中的 ProcessSwapToUFP、ProcessSwapToDFP、InitiateSwapToUFP 和 InitiateSwapToDFP 字段的逻辑“或”结果覆盖。
24	支持未分块的扩展消息。
23:22	保留。
21:20	峰值电流。
19:10	电压。
9:0	最大电流。

表 2-23. 其他 PDO。

位	说明		
	固定电源	可变电源	电池电源
31:30	00b	01b	10b
29:20	保留。	最大电压	最大电压
19:10	电压	最小电压	最小电压
9:0	最大电流	最大电流	最大允许功率

2.11 0x33 TX_SINK_CAPS 寄存器

PD 控制器在接收到 Get_Sink_Cap 消息后将此寄存器的内容作为 Sink_Capabilities 消息发送 (除非其配置或 USB PD 规则在上下文中要求不同的响应)。

备注

对此寄存器的写入不会立即生效。每当必须发送受电方能力消息时 PD 控制器都会更新并使用此寄存器。

表 2-24. 0x33 TX_SINK_CAPS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x33	TX_SINK_CAPS	RW	29	是	由应用程序配置进行初始化

表 2-25. 0x33 TX_SINK_CAPS 寄存器位字段定义

位	名称	说明
字节 26-29 : PDO #7 (视为 32 位小端字节序值)		
31:0	TXSinkPDO7	第七个受电方能力 PDO 的内容。请参阅表 2-27。
字节 22-25 : PDO #6 (视为 32 位小端字节序值)		
31:0	TXSinkPDO6	第六个受电方能力 PDO 的内容。请参阅表 2-27。
字节 18-21 : PDO #5 (视为 32 位小端字节序值)		
31:0	TXSinkPDO5	第五个受电方能力 PDO 的内容。请参阅表 2-27。
字节 14-17 : PDO #4 (视为 32 位小端字节序值)		
31:0	TXSinkPDO4	第四个受电方能力 PDO 的内容。请参阅表 2-27。
字节 10-13 : PDO #3 (视为 32 位小端字节序值)		
31:0	TXSinkPDO3	第三个受电方能力 PDO 的内容。请参阅表 2-27。
字节 6-9 : PDO #2 (视为 32 位小端字节序值)		
31:0	TXSinkPDO2	第二个受电方能力 PDO 的内容。请参阅表 2-27。
字节 2-5 : PDO #1 (视为 32 位小端字节序值)		
31:0	TXSinkPDO1	第一个受电方能力 PDO 的内容。请参阅表 2-26。
字节 1 : 标头		
7:3	保留	
2:0	numValidPDOS	

此 TX_SINK_CAPS 寄存器中的每个 PDO 都遵循 USB PD 规范中的定义, 为方便起见, 在下面对其进行了复制。更多有关每个字段的含义的详细信息, 请参阅 USB PD 规范。

表 2-26. 第一个 PDO

位	说明
31:30	电源类型, 应始终设置为 00b (固定电源)
29	双角色电源, 这将由 PORT_CONTRL 寄存器中的 ProcessSwapToSink、ProcessSwapToSource、InitiateSwapToSink 和 InitiateSwapToSource 字段的逻辑“或”结果覆盖。
28	更高的能力
27:26	保留
25	双角色数据, 这将由 PORT_CONTRL 寄存器中的 ProcessSwapToUFP、ProcessSwapToDFP、InitiateSwapToUFP 和 InitiateSwapToDFP 字段的逻辑“或”结果覆盖。
24:20	保留
19:10	电压

表 2-26. 第一个 PDO (continued)

位	说明
9:0	工作电流

表 2-27. 其他 PDO。

位	说明			
	固定电源	可变电源	电池电源	APDO (PPS)
31:30	00b	01b	10b	11b
29:28	保留。	最大电压	最大电压	00b
27:25				保留
24:20				MaxPpsVoltage
19:17	电压	最小电压	最小电压	保留
16				MinPpsVoltage
15:10				保留
9:8	工作电流	工作电流	工作功率	保留
7				MaxPpsCurrent
6:0				

2.12 0x34 ACTIVE_CONTRACT_PDO 寄存器

表 2-28. 0x34 ACTIVE_CONTRACT_PDO 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x34	ACTIVE_CONTRACT_PDO	RO	6	是	断开、硬复位或 PR_Swap 时清除。

表 2-29. 0x34 ACTIVE_CONTRACT_PDO 寄存器位字段定义

位	名称	说明
字节 5-6 : 供电方属性		
15:10	保留	
9:0	firstPDOControlBits	包含第一个 PDO 的位 29:20。选择了哪个 PDO 无关紧要，始终从第一个 PDO 提取此字段。
字节 1-4 : 合约 PDO (视为 32 位小端字节序值)		
31:0	ActivePDO	电源数据对象。此字段包含由 PD 控制器作为受电方进行请求并被供电方接受的 PDO 的内容。

2.13 0x35 ACTIVE_CONTRACT_RDO 寄存器

表 2-30. 0x35 ACTIVE_CONTRACT_RDO 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x35	ACTIVE_CONTRACT_RDO	RO	4	是	断开、硬复位或 PR_Swap 时清除。

表 2-31. 0x35 ACTIVE_CONTRACT_RDO 寄存器位字段定义

位	名称	说明
字节 1-4 : 合约 RDO (视为 32 位小端字节序值)		
31	保留	
30:28	ObjectPosition	由 USB PD 定义。
27	GiveBackFlag	由 USB PD 定义。
26	CapabilityMismatch	由 USB PD 定义。
25	USBCommCapable	由 USB PD 定义。
24	NoUSBSuspend	由 USB PD 定义。
23	UnchunkedSupported	由 USB PD 定义。
22:20	保留	
19:10	OperatingX	由 USB PD 定义。
9:0	MaxMinOperatingX	由 USB PD 定义。

2.14 0x3F POWER_STATUS 寄存器

表 2-32. 0x3F POWER_STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x3F	POWER_STATUS	RO	2	是	断开会清除。

表 2-33. 0x3F POWER_STATUS 寄存器位字段定义

位	名称	说明
15:10	保留	保留。
9:8	ChargerAdvertiseStatus ^{(1) (2)}	充电器广播状态。
		00b 禁用或不运行充电器广播。
		01b 充电器广播正在进行中。
		10b 充电器广播完成。
		11b 保留。
7:4	ChargerDetectStatus ^{(1) (2)}	0h 禁用或不运行充电器检测。
		1h 充电器检测正在进行中。
		2h 充电器检测完成，未检测到。
		3h 充电器检测完成，检测到 SDP。
		4h 充电器检测完成，检测到 BC 1.2 CDP。
		5h 充电器检测完成，检测到 BC 1.2 DCP。
		6h 充电器检测完成，检测到 Divider1 DCP。
		7h 充电器检测完成，检测到 Divider2 DCP。
		8h 充电器检测完成，检测到 Divider3 DCP。
		9h 充电器检测完成，检测到 1.2V DCP。
		Ah-Fh
3:2	TypeCCurrent	当 SourceSink 为 1b 时，此字段与寄存器 0x40 中的 PD_STATUS.CCPullUp 重复。当 SourceSink 为 0b 时，此字段与寄存器 0x29 中的 PORT_CONTROL.TypeCCurrent 重复。未来可能会删除这个多余的字段。此字段适用于 Type-C 受电方工作模式。如果端口作为供电方进行连接，则此字段仅在初始连接时更新。
		00b USB 默认电流。
		01b 1.5A。
		10b 3.0A。
		11b 由显式 PD 合约设置电流。已协商 PD 合约（更多详细信息，请参阅其他 PD 寄存器）。
1	SourceSink	供电方/受电方指示器。此位相当于寄存器 0x40 中的 PresentPDRole。为方便起见，在此寄存器中对其进行了复制。未来可能会删除这个多余的位。
		0b 连接请求电源。PD 控制器是供电方。
		1b 连接提供电源（PD 控制器充当受电方）。
0	PowerConnection	在有连接时生效。当 PlugPresent 为 TRUE 且 ConnState 大于 5h 时，此位将生效。因此，它与寄存器 0x1A 中的信息重复。为方便起见，在此寄存器中对其进行了复制。未来可能会删除这个多余的位。
		0b 无连接。此寄存器中的其余位无效。
		1b 存在连接。更多详细信息，请参阅此寄存器中的其他位。

(1) TPS25750_F509.04.02 不支持该特性。

(2) TPS25750_F509.04.02 不支持该特性。

2.15 0x40 PD_STATUS 寄存器

表 2-34. 0x40 PD_STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x40	PD_STATUS	RO	4	是	连接时清除。

表 2-35. 0x40 PD_STATUS 寄存器位字段定义

位	名称	说明
31	保留	保留。
30:28	DataResetDetails	数据复位原因。
	0h	复位值：无数据复位。
	1h	从端口伙伴收到的数据复位消息。
	2h-7h	保留
27:22	ErrorRecoveryDetails	错误恢复原因。
	00h	复位值：无错误恢复。
	01h	系统：过热关断。
	02h	系统：PP5V 意外变为低电平。
	03h	系统：故障输入 GPIO 已生效。GPIO 可以是：Fault_Input_Event。
	04h	系统：在 VBUS 引脚上检测到过压。
	05h	保留。
	06h	系统：PP_5V 上的 ILIM。
	07h	系统：PP_CABLE 上的 ILIM。
	08h	系统：在 CC 上检测到 OVP。
	9h-Fh	保留。
	10h	协议错误：无效的 DR_Swap。
	11h	协议错误：PR_Swap 序列期间没有 Good_CRC。如果受电方没有及时关闭，则会发生这种情况。
	12h-14h	保留。
	15h	策略引擎：NoResponse 计时器超时。
	16h	策略引擎：PSSourceOffTimer 在 PR_Swap 期间超时。
	17h	策略引擎：PSSourceOnTimer 在 PR_Swap 期间超时。
	18h-21h	保留
	22h	HI：在电池电量耗尽期间发生交换错误。在主机清除电池电量耗尽标志并且该配置需要作为供电方运行后，PD 控制器无法切换到受电方。
	23h-2Fh	保留。
	30h	Type-C：Attached.SRC 状态出错。
	31h	Type-C：VCONN 放电失败。
	32h-3Fh	保留。
21:16	HardResetDetails	硬复位原因
	00h	复位值，无硬复位。
	01h	从端口伙伴处收到。
	02h	由主机请求。
	03h	活动模式期间的无效 DR_Swap 请求
	04h	策略引擎需要：DischargeFailed。
	05h	策略引擎需要：NoResponseTimeOut。
	06h	策略引擎需要：SendSoftReset。

表 2-35. 0x40 PD_STATUS 寄存器位字段定义 (continued)

位	名称	说明
		07h 策略引擎需要：Sink_SelectCapability。
		08h 策略引擎需要：Sink_TransitionSink。
		09h 策略引擎需要：Sink_WaitForCapabilities。
		0Ah 策略引擎需要：SoftReset。
		0Bh 策略引擎需要：SourceOnTimeout。
		0Ch 策略引擎需要：Source_CapabilityResponse。
		0Dh 策略引擎需要：Source_SendCapabilities。
		0Eh 策略引擎需要：SourcingFault。
		0Fh 策略引擎需要：UnableToSource。
		10h 保留
		11h 策略引擎需要：意外消息
		12h 策略引擎需要：无法在 PP5V 上升沿后 200ms 内完成 VCONN 恢复序列。
		13h-3Fh 保留。
15:13	保留	
12:8	SoftResetDetails	软复位原因。
		0h 复位值，无软复位。
		1h 从端口伙伴收到的软复位。
		2h 保留。
		3h 保留。
		4h 收到的供电方能力消息无效。
		5h 消息重试次数已用尽。
		6h 意外收到一条接受消息。
		7h 意外收到一条控制消息。
		8h 意外收到一条 GetSinkCap 消息。
		9h 意外收到一条 GetSourceCap 消息。
		Ah 意外收到一条 GotoMin 消息。
		Bh 意外收到一条 PS_RDY 消息。
		Ch 意外收到一条 Ping 消息。
		Dh 意外收到一条拒绝消息。
		Eh 意外收到一条请求消息。
		Fh 意外收到一条受电方能力消息。
		10h 意外收到一条供电方能力消息。
		11h 意外收到一条交换消息。
		12h 意外收到一条等待能力消息。
		13h 收到一条未知控制消息。
		14h 收到一条未知数据消息。
		15h 在插头中初始化 SOP' 控制器。
		16h 在插头中初始化 SOP" 控制器。
		17h 意外收到一条扩展消息。
		18h 收到一条未知扩展消息。
		19h 意外收到一条数据消息。
		1Ah 意外收到一条不受支持消息。
		1Bh 意外收到一条 Get_Status 消息。
		1Ch-1Fh 保留。

表 2-35. 0x40 PD_STATUS 寄存器位字段定义 (continued)

位	名称	说明
7	保留	
6	PresentPDRole	当前的 PD 电源角色。PD 控制器在此 PD 电源角色下运行。
		0b 受电方。
		1b 供电方。
5:4	PortType	当前的 Type-C 电源角色。PD 控制器在此 Type-C 电源角色下运行。
		00b 受电方/供电方。
		01b 受电方。
		10b 供电方。
3:2	CCPullUp	CC 上拉值。PD 控制器在 CC 下拉模式下检测到的上拉值。
		00b 未处于 CC 下拉模式/未检测到 CC 上拉。
		01b USB 默认电流。
		10b 1.5A (SinkTxNG)。
1:0	保留	3.0A (SinkTxOK)。

2.16 GPIO 事件

表 2-36. GPIO 事件

事件编号	事件名称	I/O	说明
76	PdNegotiationInProgress	输出	在供电方模式下，此 GPIO 在收到请求消息后、发送接受消息之前生效。此 GPIO 在发送 PS_RDY 消息后失效。在受电方模式下，此 GPIO 在发送请求消息之前生效，并在收到 PS_RDY 消息后失效。在任一模式下，当发生断开时，此 GPIO 失效。
75	AttachedAsSink	输出	当 PD 控制器有一个端口连接到供电方时，此 GPIO 将生效。仅当 PD 控制器中的端口均未连接到供电方时，此 GPIO 才会在断开、硬复位、电源角色交换期间失效。
73	EnableSource	输出	当 PD 控制器发送一条接受消息以根据高功率合约开始在上提供 VBUS 时，此 GPIO 生效为高电平。只要高功率合约有效，便会保持高电平。如果从高功率合约转换到低功率合约，则在发送 PS_Rdy 消息后，此 GPIO 将发生从高到低的转换。
71	保留		
69	MRESET	输入	在此 GPIO 上出现上升沿时，PD 控制器将在延迟后驱动 RESETZ GPIO 上的上升沿。在此 GPIO 上出现下降沿时，PD 控制器将在延迟后驱动 RESETZ GPIO 上的下降沿。
68	RESETZ	输出	与 MRESET 搭配使用。
67	Fault_Condition_Active_Low_Global	输出	在端口上发生过流事件时生效为低电平。
61	Dp_Dm_Mux_Enable_Event	输出	必须使用此 GPIO 启用/禁用 USB 2.0 D+/D- 多路复用器。此 GPIO 在连接时驱动至高电平，断开时驱动至低电平。
52	VCONN_On_Event	输出	启用 PP_CABLE1 后，此 GPIO 将生效。
50	Debug_Accessory_Mode_Event	输出	输出：当调试附件连接后，此 GPIO 生效为高电平。
48	Audio_Mode_Event	输出	输出：当音频附件 (Ra/Ra) 连接后，此 GPIO 生效为高电平。
45	Prevent_DRSwap_To_UFP_Event	输入	当 GPIO 为高电平时，PD 控制器将拒绝任何端口伙伴请求将数据角色从 DFP 更改为 UFP 的 DR_Swap 消息。
44	UFP_Indicator_Event	输出	当 PD 控制器中任何端口的数据角色为 UFP 时，此 GPIO 将驱动至高电平。

表 2-36. GPIO 事件 (continued)

事件编号	事件名称	I/O	说明
43	Barrel_Jack_Event	输入	当此 GPIO 为高电平时, PD 控制器将其解释为连接了筒形插孔适配器并且系统已通电。因此, 当触发此事件时, 它将清除电池电量耗尽标志并尝试执行电源角色交换以转换为供电方。
42:36	保留		
35	Fault_Condition_Active_Low_Event	输出	发生过流事件时生效为低电平。
33	Fault_Input_Event	输入	由系统设置为低电平时, 端口 1 进入 Type-C 错误恢复状态。设置为高电平时, 不执行任何操作。
32:30	保留		
29	UFP_DFP_Event	输出	输出: 当端口 1 作为 UFP 运行时, 生效为高电平。当端口作为 DFP 运行时, 生效为低电平。
28:14	保留		
13	SourcePDOContractBit2	输出	输出: 二进制编码输出的位 2, 指示供电方 PDO1 到 PDO7 的协商时间 (已发送接受消息并且 tSrcTransition 计时器已到期)。
12	SourcePDOContractBit1	输出	输出: 二进制编码输出的位 1, 指示供电方 PDO1 到 PDO7 的协商时间 (已发送接受消息并且 tSrcTransition 计时器已到期)。
11	SourcePDOContractBit0	输出	输出: 二进制编码输出的位 0, 指示供电方 PDO1 到 PDO7 的协商时间 (已发送接受消息并且 tSrcTransition 计时器已到期)。
10	SourcePDO4Contract	输出	输出: 已协商供电方 PDO4 (已发送接受消息并且 tSrcTransition 计时器已到期) 时生效为高电平。已协商除 PDO2 之外的 PDO 时失效。
9	SourcePDO3Contract	输出	输出: 已协商供电方 PDO3 (已发送接受消息并且 tSrcTransition 计时器已到期) 时生效为高电平。已协商除 PDO2 之外的 PDO 时失效。
8	SourcePDO2Contract	输出	输出: 已协商供电方 PDO2 (已发送接受消息并且 tSrcTransition 计时器已到期) 时生效为高电平。已协商除 PDO2 之外的 PDO 时失效。
7	SourcePDO1Contract	输出	输出: 已协商供电方 PDO1 (已发送接受消息并且 tSrcTransition 计时器已到期) 时生效为高电平。已协商除 PDO1 之外的 PDO 时失效。
6:4	保留		
3	Cable_Orientation_Event	输出	输出: 指示插头方向。插头顶面朝上连接 (CC1 连接到电缆中的 CC) 或断开时为低电平。插头顶面朝下连接 (CC2 连接到电缆中的 CC) 时为高电平。
1	PlugEvent	输出	输出: 发生插头事件时 (已连接状态) 生效为高电平, 否则为低电平。
0	NullEvent	不适用	没有与此 GPIO 关联的事件。

2.17 0x69 TYPEC_STATE 寄存器

表 2-37. 0x69 TYPEC_STATE 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x69	TYPEC_STATE	RO	4	是	0

表 2-38. 0x69 TYPEC_STATE 寄存器位字段定义

位	名称	说明
31:24	TypeCPortState	Type-C 状态机的当前状态。
		00h 已禁用。
		01h-04h 保留。
		05h ErrorRecovery。
		06h-23h 保留。
		24h Unattached.Accessory。
		25h-2Ah 保留。
		2Bh AttachWait.Accessory。
		2Ch-44h 保留。
		45h Try.SRC。
		46h-4Dh 保留。
		4Eh TryWait.SNK。
		4Fh Try.SNK。
		50h TryWait.SRC。
		51-5Fh 保留。
		60h Attached.SRC。
		61h Attached.SNK。
		62h AudioAccessory。
		63h DebugAccessory。
		64h AttachWait.SRC。
65h AttachWait.SNK。		
66h Unattached.SNK。		
67h Unattached.SRC。		
68h-FFh 保留。		
23:16	Cc2PinState	CC2 引脚的状态
		00h 未连接。
		01h 检测到 Ra (仅供电方) 。
		02h 检测到 Rd (仅供电方)
		03h 检测到 USB 默认广播 (仅受电方) 。
		04h 检测到 1.5A 广播 (仅受电方) 。
		05h 检测到 3.0A 广播 (仅受电方) 。
		06h-FFh 保留。

表 2-38. 0x69 TYPEC_STATE 寄存器位字段定义 (continued)

位	名称	说明	
15:8	Cc1PinState	CC1 引脚的状态	
		00h	未连接。
		01h	检测到 Ra (仅供电方) 。
		02h	检测到 Rd (仅供电方) 。
		03h	检测到 USB 默认广播 (仅受电方) 。
		04h	检测到 1.5A 广播 (仅受电方) 。
		05h	检测到 3.0A 广播 (仅受电方) 。
		06h-FFh	保留。
7:0	CcPinForPd	用于 PD 通信的 CC 引脚。	
		00h	未连接。
		01h	CC1 用于 USB PD 通信。
		02h	CC2 用于 USB PD 通信。
		03h-FFh	保留。

2.18 0x70 SLEEP_CONFIG 寄存器

表 2-39. 0x70 SLEEP_CONFIG 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x70	SLEEP_CONFIG	RW	1	否	由应用程序配置进行初始化

表 2-40. 0x70 SLEEP_CONFIG 寄存器位字段定义

位	名称	说明
7:1	保留	
0	SleepModeAllowed	如果此位生效，PD 控制器将在器件空闲一段时间后进入睡眠模式。

2.19 0x72 GPIO_STATUS 寄存器

查看特定于器件的数据表以获取可用的 GPIO，因为它可能因器件类型而异。

表 2-41. 0x72 GPIO_STATUS 寄存器

地址	名称	访问	长度	每个端口均唯一	上电默认值
0x72	GPIO_STATUS	RO	8	否	不适用

表 2-42. 0x72 GPIO_STATUS 寄存器位字段定义

位	名称	说明
字节 5-8：方向配置		
31:9	保留	
8	GPIO8Dir ⁽¹⁾	当该 GPIO 配置为输出时，此位将生效。
7	GPIO7Dir	当该 GPIO 配置为输出时，此位将生效。
6	GPIO6Dir	当该 GPIO 配置为输出时，此位将生效。
5	GPIO5Dir	当该 GPIO 配置为输出时，此位将生效。
4	GPIO4Dir	当该 GPIO 配置为输出时，此位将生效。
3	GPIO3Dir	当该 GPIO 配置为输出时，此位将生效。
2	GPIO2Dir	当该 GPIO 配置为输出时，此位将生效。
1	GPIO1Dir	当该 GPIO 配置为输出时，此位将生效。
0	GPIO0Dir	当该 GPIO 配置为输出时，此位将生效。
字节 1-4：输出数据		
31:13	保留	
12	GPIO12Data	在 GPIO 上检测到逻辑高电平时生效。
11	保留	
10	保留	
9	保留	
8	GPIO8Data ⁽¹⁾	在 GPIO 上检测到逻辑高电平时生效。
7	GPIO7Data	在 GPIO 上检测到逻辑高电平时生效。
6	GPIO6Data	在 GPIO 上检测到逻辑高电平时生效。
5	GPIO5Data	在 GPIO 上检测到逻辑高电平时生效。
4	GPIO4Data	在 GPIO 上检测到逻辑高电平时生效。
3	GPIO3Data	在 GPIO 上检测到逻辑高电平时生效。
2	GPIO2Data	在 GPIO 上检测到逻辑高电平时生效。
1	GPIO1Data	在 GPIO 上检测到逻辑高电平时生效。
0	GPIO0Data	在 GPIO 上检测到逻辑高电平时生效。

(1) TPS25750_F509.05.02 不支持该特性。



3.1 概述

备注

本节适用于高级用户，此处列出的特性仅为可选特性。您的系统中需要 EC 或主机来实现以下部分中所述的特性。

本节介绍了由 PD 控制器主机接口定义的 4CC 任务。在本节中，这些任务分为不同的子组。所有使用 DATA 寄存器来返回数据的任务始终会确保在将 CMD 寄存器设置为 0 以指示任务完成之前将正确的输出数据加载到这些寄存器中。DATA 在 CMD 更改为 0 后始终不会被 PD 控制器修改，这是为了确保主机可以从先前执行的任务中检索数据，并确保主机可以为将来的任务加载这些寄存器而没有覆盖风险。请注意，其他寄存器可能会在任务完成后继续更新，因为任务可能有其他副作用。

许多任务在 DATA 寄存器的第一个字节中返回一个状态代码。标准任务响应字节在表 3-1 中定义。其余 DATA 字节可由每个任务自行使用。

表 3-1. 标准任务响应

说明	任务是一种在 DATA 寄存器第一个字节中返回一个状态码的特殊形式的任务。			
	位	名称	说明	
Output DATA	字节 1：任务返回代码			
	7:4	保留	保留用于标准任务。可由某些任务用于特定于任务的返回代码。如果 TaskResult 为 0x0，成功返回代码可以使用此字节。	
	3:0	TaskResult	标准任务返回代码。	
			0x0	任务成功完成。
			0x1	任务超时或被 'ABRT' 请求中止。
			0x2	保留。
			0x3	任务被拒绝。
			0x4	由于锁定了 Rx 缓冲区，任务被拒绝。适用于需要 PD 控制器使用 Rx 缓冲区的任务。
0x5-0xF	保留用于标准任务。可由某些任务用于特定于任务的错误代码。遇到时视为错误。			

3.2 PD 消息任务

3.2.1 'SWSk' - PD PR_Swap 到受电方

表 3-2. 'SWSk' - PD PR_Swap 到受电方

说明	'SWSk' 任务指示 PD 控制器在第一时间通过 PR_Swap 尝试成为受电方，同时保持策略引擎的合规性。
INPUT DATAx	无。
OUTPUT DATAx	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'SWSk' 任务在 PR_Swap 完成或因其他原因确定为不可能或失败时完成。此任务可能会由于供电方正在发送等待消息而继续运行。如果出现以下情况，则应将 'SWSk' 任务视为被拒绝：</p> <ul style="list-style-type: none"> 供电方通过供电方能力消息指示其不支持双角色电源。 PR_Swap 被拒绝。 <p>如果出现以下情况，则应将 'SWSk' 任务视为已超时：</p> <ul style="list-style-type: none"> PR_Swap 被接受，但未能按照 PD 规范完成。 <p>如果出现以下情况，则应将 'SWSk' 任务视为成功：</p> <ul style="list-style-type: none"> PD 控制器已处于受电方电源角色。 PR_Swap 被接受并正常完成。
副作用	当 'SWSk' 任务成功完成时，PD 控制器将转换为受电方电源角色，这会影响其他寄存器。如果在发送接受消息后 PR_Swap 失败，则可能会根据 PD 规范要求发生软复位和/或硬复位。
其他信息	无。

3.2.2 'SWSr' - PD PR_Swap 到供电方

表 3-3. 'SWSr' - PD PR_Swap 到供电方

说明	'SWSr' 任务指示 PD 控制器在第一时间通过 PR_Swap 尝试成为供电方，同时保持策略引擎的合规性。
INPUT DATAx	无。
OUTPUT DATAx	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'SWSr' 任务在 PR_Swap 完成或因其他原因确定为不可能或失败时完成。此任务可能会由于受电方正在发送等待消息而继续运行。如果出现以下情况，则应将 'SWSr' 任务视为被拒绝：</p> <ul style="list-style-type: none"> 受电方通过受电方能力或供电方能力消息指示其不支持双角色电源。 PR_Swap 被拒绝。 <p>如果出现以下情况，则应将 'SWSr' 任务视为已超时：</p> <ul style="list-style-type: none"> PR_Swap 被接受，但未能按照 PD 规范完成。 <p>如果出现以下情况，则应将 'SWSr' 任务视为成功：</p> <ul style="list-style-type: none"> PD 控制器已处于供电方电源角色。 PR_Swap 被接受并正常完成。
副作用	当 'SWSr' 任务成功完成时，PD 控制器将转换为供电方电源角色，这会影响其他寄存器。如果在发送接受消息后 PR_Swap 失败，则可能会根据 PD 规范要求发生软复位和/或硬复位。
其他信息	无。

3.2.3 'SWDF' - PD DR_Swap 到 DFP

表 3-4. 'SWDF' - PD DR_Swap 到 DFP

说明	'SWDF' 任务指示 PD 控制器在第一时间通过 DR_Swap 尝试成为 DFP，同时保持策略引擎的合规性。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'SWDF' 任务在 DR_Swap 完成或因其他原因确定为不可能或失败时完成。此任务可能会由于 DFP 正在发送等待消息而继续运行。如果出现以下情况，则应将 'SWDF' 任务视为被拒绝：</p> <ul style="list-style-type: none"> • UFP 通过供电方能力或受电方能力消息指示其不支持数据角色交换。 • DR_Swap 被拒绝。 <p>如果出现以下情况，则应将 'SWDF' 任务视为成功：</p> <ul style="list-style-type: none"> • PD 控制器已处于 DFP 数据角色。 • DR_Swap 被接受并正常完成。
副作用	当 'SWDF' 任务成功完成时，PD 控制器将转换为 DFP 数据角色，这会影响其他寄存器。如果在发送接受消息后 DR_Swap 失败，则可能会根据 PD 规范要求发生软复位和/或硬复位。
其他信息	无。

3.2.4 'SWUF' - PD DR_Swap 到 UFP

表 3-5. 'SWUF' - PD DR_Swap 到 UFP

说明	'SWUF' 任务指示 PD 控制器在第一时间通过 DR_Swap 尝试成为 UFP，同时保持策略引擎的合规性。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'SWUF' 任务在 DR_Swap 完成或因其他原因确定为不可能或失败时完成。此任务可能会由于 UFP 正在发送等待消息而继续运行。如果出现以下情况，则应将 'SWUF' 任务视为被拒绝：</p> <ul style="list-style-type: none"> • DFP 通过供电方能力或受电方能力消息指示其不支持数据角色交换。 • DR_Swap 被拒绝。 <p>如果出现以下情况，则应将 'SWUF' 任务视为成功：</p> <ul style="list-style-type: none"> • PD 控制器已处于 UFP 数据角色。 • DR_Swap 被接受并正常完成。
副作用	当 'SWDF' 任务成功完成时，PD 控制器将转换为 UFP 数据角色，这会影响其他寄存器。如果在发送接受消息后 DR_Swap 失败，则可能会根据 PD 规范要求发生软复位和/或硬复位。
其他信息	无。

3.2.5 'GSKC' - PD 获取受电方能力

表 3-6. 'GSKC' - PD 获取受电方能力

说明	'GSKC' 任务指示 PD 控制器在第一时间向端口伙伴发出 Get_Sink_Cap 消息，同时保持策略引擎的合规性。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'GSKC' 任务在收到受电方能力消息或任务因其他原因失败时完成。</p> <ul style="list-style-type: none"> 端口伙伴是供电方，并且表明其不是双角色电源。 端口伙伴以 Reject 或 Not_Supported 消息来响应 Get_Sink_Cap 消息。 <p>如果出现以下情况，则应将 'GSKC' 任务视为已超时：</p> <ul style="list-style-type: none"> 端口伙伴未能在 PD 规范要求的时间内做出响应。 <p>如果出现以下情况，则应将 'GSKC' 任务视为成功：</p> <ul style="list-style-type: none"> 已发送 Get_Sink_Cap 消息，已收到 GoodCRC，并且已收到并处理了受电方能力响应。
副作用	当 'GSKC' 任务成功完成时，RX_SINK_CAPS 寄存器 (0x31) 将被更新。
其他信息	无。

3.2.6 'GSrC' - PD 获取供电方能力

表 3-7. 'GSrC' - PD 获取供电方能力

说明	'GSrC' 任务指示 PD 控制器在第一时间向端口伙伴发出 Get_Source_Cap 消息，同时保持策略引擎的合规性。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'GSrC' 任务在收到供电方能力消息或任务因其他原因失败时完成。如果出现以下情况，则应将 'GSrC' 任务视为被拒绝：</p> <ul style="list-style-type: none"> 端口伙伴是受电方，并且（通过先前的供电方或受电方能力）表明其不是双角色电源。 端口伙伴以 Reject 或 Not_Supported 消息来响应 Get_Source_Cap 消息。 <p>如果出现以下情况，则应将 'GSrC' 任务视为已超时：</p> <ul style="list-style-type: none"> 端口伙伴未能在 PD 规范要求的时间内做出响应。 <p>如果出现以下情况，则应将 'GSrC' 任务视为成功：</p> <ul style="list-style-type: none"> 已发送 Get_Source_Cap 消息，已收到 GoodCRC，并且收到并处理了供电方能力响应。
副作用	当 'GSrC' 任务成功完成时，RX_SOURCE_CAPS 寄存器 (0x30) 将被更新。
其他信息	无。

3.2.7 'SSrC' - PD 发送供电方能力

表 3-8. 'SSrC' - PD 发送供电方能力

说明	'SSrC' 任务指示 PD 控制器在第一时间发送供电方能力消息，同时保持策略引擎的合规性。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>'SSrC' 任务在收到受电方能力消息 GoodCRC 或任务因其他原因失败时完成。如果出现以下情况，则应将 'SSrC' 任务视为被拒绝：</p> <ul style="list-style-type: none"> PD 控制器未处于供电方角色。 <p>如果出现以下情况，则应将 'SSrC' 任务视为已超时：</p> <ul style="list-style-type: none"> 已发送供电方能力消息，但未收到 GoodCRC。 <p>如果出现以下情况，则应将 'SSrC' 任务视为成功：</p> <ul style="list-style-type: none"> 已发送供电方能力消息，并收到 GoodCRC。
副作用	由于存在以新的供电方能力消息开始的合约协商，其他寄存器可能会发生变化。
其他信息	无。

3.3 补丁捆绑包更新任务

下面列出的 4CC 命令和序列从外部主机通过 I2C 以突发模式将补丁捆绑包加载到 PD 控制器。

以下任务用于更新补丁捆绑包。

3.3.1 'PBMs' - 启动补丁突发模式下载序列

表 3-9. 'PBMs' - 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会包含的内容。		
INPUT DATAx	位	名称	说明
	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，建议在此字段中始终使用 0x32 (5 秒) (LSB 为 100ms) 。
	字节 5 : 用于下载补丁的 I2C 从器件。		
	7	保留	
	6:0	I2C 从器件地址	以下从器件地址无效 : <ul style="list-style-type: none"> • 0x00。 • 由 ADCInx 引脚选择的任何端口的 I2C1s 从器件地址。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		
OUTPUT DATAx	位	名称	说明
	7:0	PatchStartStatus	补丁启动状态。
			0x00 补丁启动成功。
			0x04 无效的捆绑包大小。
			0x05 无效的从器件地址。
0x06 无效的超时值。			
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', 则将拒绝此任务。		
副作用	当 'PBMs' 成功时, 第二个从器件地址将设置为输入值。		
其他信息	主机只能向 PD 控制器的 I2Cs 端口发出 '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), 则它将拒绝 'PBMs' 任务。		

3.3.2 'PBMc' - 补丁突发模式下下载完成

表 3-10. 'PBMc' - 补丁突发下载完成

说明	'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此任务。此任务将启动对已传输的二进制补丁数据进行的 CRC 校验，如果 CRC 成功，则会执行补丁中包含的 patch_init 函数。			
INPUT DATAx	无			
OUTPUT DATAx	位	名称	说明	
	319:288	acCalculatedCRC	在固件中为配置数据计算的 CRC。	
	287:256	acTransferredCRC	随配置数据一同传输的 CRC。	
	255:240	保留	读数为 0。	
	239:224	acIndicatedDataSize	传输的配置数据中指示的 DataSize。	
	223:216	acHeaderVersion	传输的配置数据中指示的标头版本。	
	215:208	acFailCode	如果应用程序配置数据未能应用，则会显示此错误代码，指示未能应用的原因。	
			0x00	AC_FAIL_NONE：无失败。
			0x01	AC_FAIL_WRONG_HEADER_VERSION：标头版本应为 1，但不是 1。
			0x02	AC_FAIL_TOO_MUCH_DATA：DataSize 字段表明您正在尝试加载已分配了 SRAM 的更多配置数据。
	207:200	acState	AppConfig 状态机的当前内部状态。	
			0x00	AC_NODATA：尚未找到配置数据，因为我们尚未开始查找。
			0x01	AC_LOADING_DEFAULT：尝试从出厂默认值加载配置数据。
			0x02	AC_LOADING_SRAM：尝试从 SRAM 加载配置数据。
			0x03	AC_LOADING_FLASH：尝试从闪存加载配置数据。
			0x04	AC_LOADING_I2C：尝试从 I2C 加载配置数据。
			0x05	AC_LOADING_DONE：配置数据加载完成，我们找到了有效数据。
			0x06	AC_ERROR：通用错误状态。
			0x07	AC_DONE_SUCCESS：完全完成了应用程序自定义过程，并成功应用了记录。
	0x08	AC_DONE_FAIL：完全完成了应用程序自定义过程，但未应用记录。		
199:192	configBundleGood	如果顶级状态机找到有效的配置捆绑包，则为 1，否则为 0。		
191:160	rpRomVersionExpected	传输的捆绑包补丁标头中的 romVersionExpected		
159:144	rpBundleTotalSize	传输的捆绑包补丁标头中的 bundleTotalSize。		
143:128	rpBundleFlags	传输的捆绑包补丁标头中的 bundleFlags。		
127:96	rpPatchBodyCrc	传输的捆绑包补丁标头中的 patchBodyCrc。		
95:64	rpPatchHeaderCrc	传输的捆绑包补丁标头中的 patchHeaderCrc。		

表 3-10. 'PBMc' - 补丁突发下载完成 (continued)

说明	'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此任务。此任务将启动对已传输的二进制补丁数据进行的 CRC 校验，如果 CRC 成功，则会执行补丁中包含的 patch_init 函数。			
OUTPUT DATAx	位	名称	说明	
	55:48	rpBundleSignature	传输的捆绑包补丁标头中的 bundleSignature。	
	47:40	rpState	RomPatch 状态机的当前内部状态。	
			0x00	RP_NOPATCH：未加载任何补丁。
			0x01	RP_LOADING：在加载补丁数据的过程中。
			0x02	RP_LOADINGDONE：已收到所有补丁数据。
			0x03	RP_RUNNING：一个补丁已加载并正在运行。也可能指示有一个 NULL 补丁处于活动状态。
			0x04	RP_EARLYLOAD_SKIPPED：指示早期引导过程不需要等待 I2C 上的补丁。
			0x05	RP_UARTBOOTED：检查 RAM 中的补丁。
	0x06	RP_ERROR：通用错误状态。		
	39:32	patchBundleGood	如果顶级状态机找到良好的 ROM 补丁，则为 0x01，否则为 0x00。	
	31:24	AppConfigPatchCompleteStatus	0x00	
			0x40	警告。
			0x80	失败。
	23:16	DevicePatchCompleteStatus	一个指示 RomPatch 状态机是否成功执行的返回代码。此值始终有效，并且反映了 RomPatch 机制的内部状态，但仅当传输的捆绑包确实包含补丁数据时必须纳入考量。	
			0x00	成功。
			0x20	未就绪。
			0x40	不是补丁。
			0x41	补丁标头校验和不匹配。
			0x42	补丁与此版本的 ROM 不兼容。
			0x43	补丁代码校验和不匹配。
			0x44	收到 null 补丁。
	0x45	收到错误补丁。		
15:8	cpReturn	始终返回成功，无法使其失败。		
字节 1：返回代码				
7:4	rpReturnIndicator	rpReturn 值的最高有效半字节。		
		0x0	成功。	
		0x2	参考。	
		0x4	警告。	
		0x8	错误。	
3:0	acReturnIndicator	acReturn 值的最高有效半字节。		
		0x0	成功。	
		0x2	参考。	
		0x4	警告。	
		0x8	错误。	
任务完成	当输出具有有效的 DevicePatchCompleteStatus 和 AppConfigPatchCompleteStatus 时，'PBMc' 任务完成。如果 DATAx 输入不包含总补丁大小，则此任务被拒绝。如果 MODE 寄存器 (0x03) 等于 'APP'，则此任务将被拒绝。			
副作用	在此任务完成之前，它会将 I2C 从器件地址从补丁地址更改回正常值。成功完成此任务后，PD 控制器会将 MODE 寄存器 (0x03) 更改为 'APP' 并进入应用程序模式。			

表 3-10. 'PBMc' - 补丁突发下载完成 (continued)

说明	'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此任务。此任务将启动对已传输的二进制补丁数据进行的 CRC 校验，如果 CRC 成功，则会执行补丁中包含的 patch_init 函数。
其他信息	当 CMDx 寄存器变为 0 时，应检查 Output DATAx 寄存器的状态。如果 MODE 寄存器为 'APP' (表明 PD 控制器处于 APP 模式)，则它将拒绝 'PBMc' 任务。

3.3.3 'PBMe' - 结束补丁突发模式下载序列

表 3-11. 'PBMe' - 补丁突发模式退出

说明	'PBMe' 任务结束补丁加载序列。此任务指示 PD 控制器完成补丁加载过程。
INPUT DATAx	.
OUTPUT DATAx	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	'PBMe' 任务在结束补丁加载序列时完成。如果 MODE 寄存器 (0x03) 等于 'APP'，则将拒绝此任务。
副作用	当 'PBMe' 成功时，第二个从器件地址将恢复为 ADCINx 引脚配置的值。PD 控制器将 MODE 寄存器 (0x03) 保留为 'PTCH'，并会等待修补过程重新启动。
其他信息	如果 MODE 寄存器为 'APP' (表明 PD 控制器处于 APP 模式)，则它将拒绝 'PBMe' 任务。

3.3.4 补丁突发模式示例

命令

- **PBMs**：'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。
- **PBMc**：'PBMc' 任务结束补丁加载序列。此命令应在所有补丁数据传输完毕后执行。此任务启动对已传输的二进制补丁数据进行的 CRC 校验，如果 CRC 成功，则执行补丁中包含的 patch_init 函数。如果此任务在 'PBMs' 启动任务之前发送，则会向 PD 控制器指示没有可用的补丁并绕过补丁过程。
- **PBMe**：'PBMe' 任务结束补丁加载序列。此任务指示 PD 控制器完成补丁加载过程。
- **PTCr**：此任务将补丁固件重置为无补丁状态。

1. 器件生成一个 I2C 中断 INT_EVENT.ReadyForPatch，表明其已准备好进行修补。主机只有在收到器件发出的此通知后才会启动补丁下载过程。

2. 主机应通过向器件发送 PBMs 命令来启动补丁下载过程。

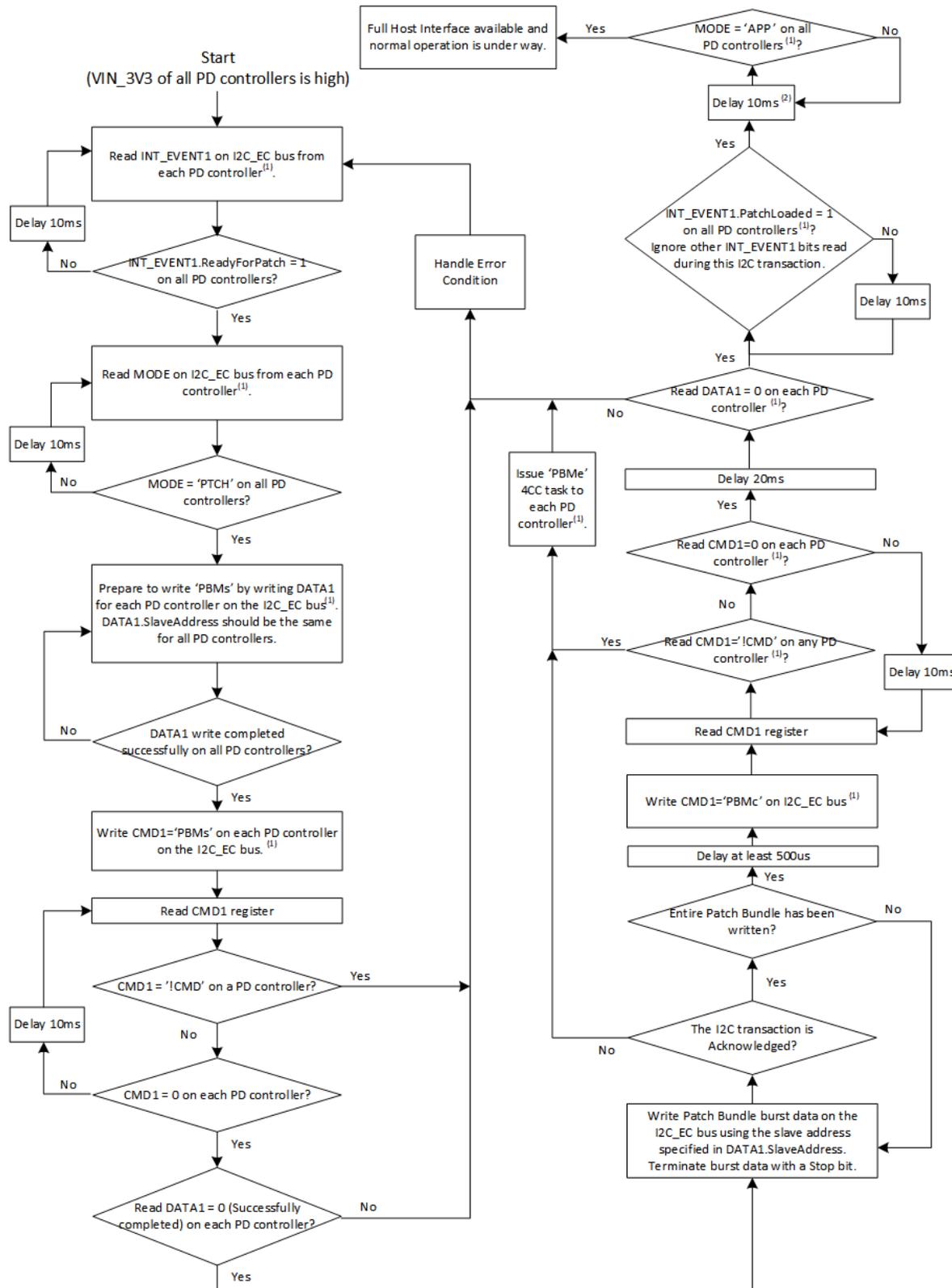
3. 主机随后应发送整个补丁捆绑包数据。主机可以在单个 I2C 事务中发送整个补丁捆绑包，也可以将其拆分为多个事务。PD 控制器会递增其补丁存储器空间的指针，并在补丁从器件地址（在 'PBMs' 4CC 任务中进行配置）上接收每个字节。

4. 主机随后应发送 PBMc 命令以指示 PD 控制器补丁加载序列已完成。成功完成引导序列后，PD 控制器开始执行应用程序。

若要执行 4CC 命令，主机应用程序应遵循以下顺序：

- 如果 4CC 命令需要输入，应用程序应首先将输入数据写入 DataX (0x09 或 0x11) 寄存器。
- 应用程序随后应将 4CC 命令字符写入相应的 CmdX (0x08 或 0x10) 寄存器。
- 应用程序可以注册并等待 CmdXComplete 事件或重复读取 CmdX 寄存器的四字节内容，直至读取到：
 - 0x00'，表示命令执行成功。
 - 或 '!CMD'，表示命令执行失败。
- 如果命令成功执行，应用程序将继续读取包含输出的 DataX 寄存器 'n' 字节内容。

执行流程



⁽¹⁾ Use the fundamental I2C slave address of each PD controller.
⁽²⁾ This delay before reading the MODE register, is optional but recommended.

示例代码

1.

```

void AsyncEvtHandlerP11()
{
    s_AppContext *const pCtx = &gAppCtx;
    I2C_IF_TPS6598xIntDisable(pCtx->i2cPort);
    ProcessEvent();
}

static int32_t ProcessEvent()
{
    s_AppContext *const pCtx = &gAppCtx;
    s_AppData *const pData = &gAppData[pCtx->i2cPort];
    uint8_t outdata[MAX_BUF_BSIZE] = {0};
    uint8_t indata[MAX_BUF_BSIZE] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF};
    s_TPS_IntEvent *pSetEvent = NULL;
    s_TPS_IntEvent *pClrEvent = NULL;
    int32_t retVal = -1;
#ifdef WORKAROUND_WAIT_AFTER_GAID
    if(0 == wait_after_gaid)
    {
        wait_after_gaid += 1;
        /*
        * The 100ms delay here is to allow the device to ACK the I2C transactions
        * after a cold start
        */
        Board_IF_Delay(100000);
    }
#endif /* WORKAROUND_WAIT_AFTER_GAID */
    retVal = ReadReg(pData->event_reg, REG_LEN_INTEVENT1, &outdata[0]);
    RETURN_ON_ERROR(retVal);
    /*
    * output[0] = Register's size in bytes
    * output[1..REG_LEN_INTEVENT1] = Register's content, and hence '&outdata[1]' below.!
    */
    pSetEvent = (s_TPS_IntEvent *)&outdata[1];
    pClrEvent = (s_TPS_IntEvent *)&indata[0];
    if(0 != pSetEvent->cmd1complete)

```

```

{
pData->cmdExeInProgress = 0;
pClrEvent->cmd1complete = 1;
}
if(0 != pSetEvent->readyforpatch)
{
pClrEvent->readyforpatch = 1;
SignalEvent(APP_EVENT_READY_FOR_PATCH);
}
if(0 != pSetEvent->patchloaded)
{
pClrEvent->patchloaded = 1;
SignalEvent(APP_EVENT_PATCH_LOADED);
}
/* App events are triggered - The corresponding events of the device can now be cleared */ retVal =
WriteReg(pData->clear_reg, REG_LEN_INTCLEAR1, &indata[0]); RETURN_ON_ERROR(retVal);
return 0;
}

```

2.

```

static int32_t StartPatchDownloadBurstMode()
{
uint8_t outdata[MAX_BUF_BSIZE] = {0};
s_PBMs_InData pbmsInData = {0};
s_PBMs_OutData *p_pbmsOutData = NULL;
int32_t retVal = -1;
UART_PRINT("Starting Patch Download in Burst Mode\n\r");
retVal = ReadMode();
RETURN_ON_ERROR(retVal);
retVal = ReadVersion(); RETURN_ON_ERROR(retVal); /* If the device is in APP mode, don't allow burst-mode
patch download */
if(isDeviceInAppMode())
{
UART_PRINT("Burst mode patching not allowed in APP mode\n\r");
retVal = -1;
goto error;
} /* Wait for CmdComplete event when executing the 4CC commands */

```

```

retVal = MaskCmdComplete(1);
RETURN_ON_ERROR(retVal); /* Fill the input, and execute 'PBMs' */
pbmsInData.lrbinsize = gSizeLowregionArray;
pbmsInData.slaveAddress = SLAVE_ADDR_FOR_PATCH_DATA;
pbmsInData.timeout = BURST_MODE_TIMEOUT;
retVal = ExecCmd(PBMs, sizeof(pbmsInData), (uint8_t *)&pbmsInData, \ PBMs_OUTPUT_SIZE, &outdata[0]);
RETURN_ON_ERROR(retVal); /* 'outdata' will contain the command's output after the successful execution of
PBMs. */
p_pbmsOutData = (s_PBMs_OutData *)&outdata[1];
if(PBMs_SUCCESS != p_pbmsOutData->patch_start_status)
{
ERR_PRINT(p_pbmsOutData->patch_start_status);
retVal = -1; goto error;
}
error: return retVal;
}

```

3.

可使用应用程序自定义 GUI 生成低区二进制 'C' 数组格式。以下代码段中引用的 `tps6598x_lowregion_array` 和 `gSizeLowregionArray` 取自这个生成的文件。

```

static int32_t DownloadDataBurstMode()
{
int32_t retVal = 0;
#ifdef WORKAROUND_IGNORE_CMDxCMPLT_BURST_TX
/*
* Workaround: Device was erroneously sending an unexpected CmdXComplete
* event occasionally while in the middle of the patch download process,
* causing the SM to go for a toss.!
*/
MaskCmdComplete(0);
#endif /* WORKAROUND_IGNORE_CMDxCMPLT_BURST_TX */
/*
* tps6598x_lowregion_array and gSizeLowregionArray are defined
* in the 'C' array file that can be created using the
* Application customization GUI
*/
UART_PRINT(".");

```

```

SendDataToSlave(SLAVE_ADDR_FOR_PATCH_DATA, gSizeLowregionArray, (uint8_t
*)&tps6598x_lowregion_array[0]);
/* Proceed to the next step after the patch is successfully downloaded */
UART_PRINT("\n\r");
SignalEvent(APP_EVENT_PATCH_DOWNLOADED);
return retVal;
}

```

4.

```

static int32_t PatchDownloadCompleteBurstMode()
{
s_AppContext *const pCtx = &gAppCtx;
/* PBMc return structure and statuses are exactly same as PTCc */
s_PTCc_OutData *p_pbmcOutData = NULL;
uint8_t outdata[MAX_BUF_BSIZE] = {0};
int32_t retVal = -1;
UART_PRINT("Patch Download Complete\n\r");
#ifdef WORKAROUND_IGNORE_CMDxCMPLT_BURST_TX
MaskCmdComplete(1);
#endif /* WORKAROUND_IGNORE_CMDxCMPLT_BURST_TX */
/* Execute PBMc to indicate to the device that the patch is successfully downloaded */
retVal = ExecCmd(PBMc, 0, NULL, PTCc_OUTPUT_SIZE, &outdata[0]);
RETURN_ON_ERROR(retVal);
/* 'outdata' will contain the command's output after the successful execution of PBMc.*/
p_pbmcOutData = (s_PTCc_OutData *)&outdata[3];
if(p_pbmcOutData->patch_complete_status != PTCc_SUCCESS)
{
ERR_PRINT(p_pbmcOutData->patch_complete_status);
retVal = -1;
goto error;
}
/* Expecting 'PatchLoaded' event */
retVal = I2C_IF_TPS6598xIntEnable(pCtx->i2cPort);
RETURN_ON_ERROR(retVal);
error: return retVal;
}

```

3.3.5 'GO2P' - 转到补丁模式

表 3-12. 'GO2P' - 强制 PD 控制器恢复到 'PTCH' 模式并等待 I2C 上的补丁

说明	'GO2P' 任务使 PD 控制器重新进入补丁模式 (MODE = 'PTCH')。
INPUT DATA	无。
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。
任务完成	<p>在 PD 控制器重新进入补丁模式后，'GO2P' 任务完成。</p> <ul style="list-style-type: none"> 如果 PD 控制器已重新进入补丁模式，MODE 寄存器的读数为 'PTCH'。 <p>如果出现以下情况，则应将 'GO2P' 任务视为被拒绝：</p> <ul style="list-style-type: none"> PD 控制器未进入 'APP' 模式，没有通过 I2C 收到补丁。 BOOT_STATUS.PatchConfigSource 的读数不是 3h 或 4h。
副作用	当 'GO2P' 任务成功完成时，MODE 寄存器的读数将是 'PTCH'，并将禁用 USB PD PHY。PD 控制器可能会暂时对 I2C 事务执行 NAK。主机必须等待 IRQ 信号生效（因为 INT_EVENT1.ReadyForPatch 已生效），然后尽快推送补丁。
其他信息	仅当使用 ADCINx 配置选项 NegotiateHighVoltage 时才能使用 'GO2P' 任务。

3.4 系统任务

3.4.1 'DBfg' - 清除电池电量耗尽标志

表 3-13. 'DBfg' - 清除电池电量耗尽标志

说明	'DBfg' 任务用于清除电池电量耗尽标志。此任务不会禁用 在电池电量耗尽后运行期间可能已启用的 PP_EXT 输入开关。
INPUT DATAx	无。
OUTPUT DATAx	无。
任务完成	当清除电池电量耗尽标志的作用结束后，'DBfg' 任务完成。
副作用	电池电量耗尽标志会使 PD 控制器执行特定操作，因此清除此标志会产生副作用。在清除电池电量耗尽标志之前，PD 控制器的电源输入被强制为 VBUS，因此执行此任务将更改 PD 控制器的电源输入。
其他信息	无。

在电池电量耗尽标志生效 (PowerPathStatus.PowerSource = 10b) 期间，PD 控制器存在几个限制。

- 在受电方角色中不会发送硬复位 (在任一端口上)。
- 即使存在 3.3V 输入，仍会选择 VBUS 作为 PD 控制器的电源。
- PD 控制器将拒绝申请成为供电方的 PR_Swap 请求 (在任一端口上)。
- PD 控制器中未连接的第二个端口作为供电方连接时，只会提供 USB Type-C 默认 Rp (忽略 PortControl.TypeCCurrent)。
- 无论配置如何，连接到供电方的端口都只会充当 Type-C 受电方。
- 如果在引导过程完成后没有收到供电方能力消息 (Status.ActingAsLegacy=11b)，则即使 SinkWaitCapTimer 到期，PD 控制器也不会发送硬复位，直到清除电池电量耗尽标志。

3.4.2 'I2Cr' - I2C 读取事务

表 3-14. 'I2Cr' - 在 I2Cm 上执行 I2C 读取事务

说明	'I2Cr' 任务可用于使 PD 控制器使用 I ² C 读取事务通过 I2Cm_SDA 和 I2Cm_SCL 引脚从指定的从器件地址和寄存器偏移中执行读取。		
INPUT DATA	位	名称	说明
	字节 3：要在从器件中读取的字节数。		
	7:0	NumBytes。	
	字节 2：要在 I2C 读取事务中使用的寄存器偏移。		
	7:0	RegisterOffset。	
	字节 1：从器件地址。		
	7	保留。	
6:0	要用于此事务的从器件。		
OUTPUT DATA	位	名称	说明
	字节 2-65：要在从器件中读取（按照接收顺序）的数据字节。		
	511:0	数据。	
字节 1：标准任务返回代码。另请参阅表 3-1。			
任务完成	PD 控制器在成功读取指定的字节数后或 I ² C 事务因其他原因终止后完成。		
副作用	此任务将导致 PD 控制器在 I2Cm 端口上发出一条命令。可能会导致 INT_EVENT。I2CMasterNACKed 生效。		
其他信息	无。		

3.4.3 'I2Cw' - I2C 写入事务

表 3-15. 'I2Cw' - 在 I2Cm 上执行 I2C 写入事务

说明	'I2Cw' 任务可用于使 PD 控制器写入一个特殊的 I ² C 事务（使用 I2Cm_SDA 和 I2Cm_SCL）。		
INPUT DATA	位	名称	说明
	字节 5-14：I2C 事务的有效载荷。		
	字节 4：I2C 事务的寄存器偏移。		
	7:0	寄存器偏移。	
	字节 2-3：长度。		
	15:8	保留。	
	7:0	事务有效载荷中的字节数。	
	字节 1：从器件地址。		
	7	保留。	
6:0	要用于此事务的从器件。		
OUTPUT DATA	字节 1：标准任务返回代码。另请参阅表 3-1。		
任务完成	PD 控制器维护着一个要在 I2Cm 端口上发送的事务队列。如果 PD 控制器已配置为在发生某些事件时发送事务，则在收到 'I2Cw' 任务时队列中可能存在事务。在这种情况下，该任务将在事务插入队列后成功完成。如果 PD 控制器因任何原因未能将任务插入队列，则该任务被拒绝。所以，当此任务成功完成时，并不能确保 I2C 事务已完成。如果可能，主机必须使用 'I2Cr' 4CC 任务来确认写入成功。		
副作用	成功后，此任务将导致 PD 控制器在 I2Cm 端口上发出一条命令。这可能导致 INT_EVENT.I2CMasterNACKed 生效。		
其他信息	如果写入 DATA 寄存器的字节数超过 14，则会忽略字节 14 以外的所有字节。PD 控制器对 I ² C 写入事务的最大长度有限制。		



4.1 PD 控制器应用程序自定义

PD 控制器应用程序二进制文件可以使用 I2Cs 端口通过 I2C 进行推送，或者 PD 控制器可以按 I2Cm 端口上的从器件地址 0x50 从外部 EEPROM 中进行读取。PD 控制器应用程序二进制文件提供了一种自定义和初始化 PD 控制器设置的方法。它允许在 PD 控制器应用程序开始正常运行之前更改通过主机接口访问的任何寄存器位，以便配置在做出任何应用程序决策之前必须保证正确的系统相关设置。TI 提供了一个 GUI 工具来创建 PD 控制器应用程序二进制文件。

4.2 加载补丁捆绑包

补丁捆绑包可能包含应用程序自定义数据以及用于修改 PD 控制器中默认应用固件的补丁二进制文件。本节介绍了主机如何加载补丁捆绑包。主机使用 I2Cs 总线来处理与加载补丁捆绑包相关的所有事务。如下图所示，I2C 从器件地址根据 PD 控制器所处的模式而有所不同。补丁突发模式允许主机同时将补丁捆绑包推送到多个 PD 控制器。

以下流程图演示了成功加载补丁的过程。根据特定系统遇到的错误的性质，可能还需要其他错误处理步骤。EC 可以通过发出 'PBMe' 4CC 任务来重置和重新启动补丁过程。

表 4-1. 在不同工作模式下使用从器件地址。

MODE 寄存器回读值	I2Cs
	从器件地址 #1
'BOOT'	由端口 A 的 ADCINx 引脚进行配置。这是“基本”I2C 从器件地址。
'PTCH' ⁽¹⁾	
'APP' ⁽²⁾	

(1) 成功完成 'PBMs' 任务会将 PD 控制器置于 'PTCH' 模式。

(2) 成功完成 'PBMc' 任务会将 PD 控制器置于 'APP' 模式。

主机在写入补丁捆绑包突发数据时，必须遵循下图中的 I2C 协议。主机可以在单个 I2C 事务中发送整个补丁捆绑包，也可以将其分解为多个事务。PD 控制器会递增其补丁存储器空间的指针，并在补丁从器件地址（在 'PBMs' 4CC 任务中由 DATA1.SlaveAddress 进行配置）上接收每个字节。EC 可以重新发出 'PBMs' 4CC 任务或者可以发出 'PBMe' 4CC 任务来重置指针。

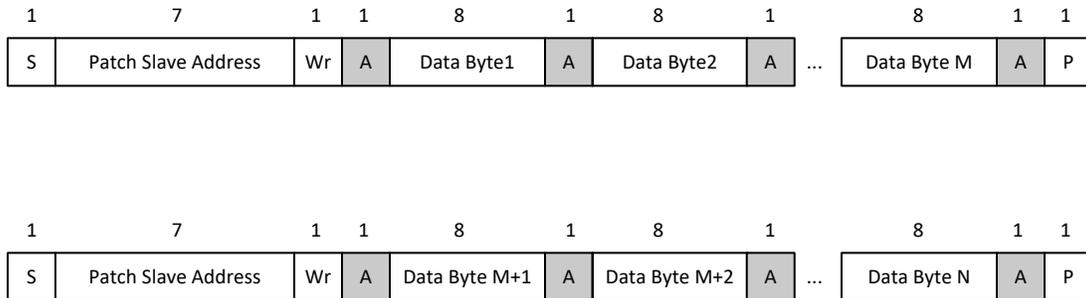


图 4-2. 分成两个事务时的补丁捆绑包突发数据的协议

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 * (November 2020) to Revision A (July 2022)	Page
• 添加了序文部分.....	7
• 添加了 PBMx 方法和示例.....	9

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