
**面向STM32Cube的STM32 USB-PD（电力传输）
软件扩展**

前言

本文档介绍了面向STM32Cube的STM32 USB-PD（电力传输）软件扩展，参考X-CUBE-USB-PD。

USB Type-C™是最新的USB连接器生态系统，能够满足平台和设备不断变化的需求，同时保持USB的功能优势。

USB Power Delivery协议嵌入到了USB Type-C™连接器中，从而可以轻松连接/断开USB线缆，实现很多数据传输以外的功能。此协议能够传输超过常规5 V / 1.5 A的电压/电流，最大功率供应为100 W。

STM32 USB-PD包（X-CUBE-USB-PD）包含一个库（二进制格式）和以STM32F0设备作为USB Power Delivery微控制器的应用示例。可扩展到其他STM32系列。

该软件包包含的示例涵盖了大部分应用用例，允许用户基于USB-PD（供电方，受电方，和双角色端口）开发应用。

该库及相关示例可用于STM32F0设备以及专用P-NUCLEO-USB001扩展板（包含模拟前端和USB Type-C™连接器）。

本文档介绍了如何使用USB-PD库实现常规用途，以及创建自定义应用。它包含了以下专题，以简化此库的使用：

- USB-PD标准概述
- USB-PD库架构
- USB-PD堆栈使用描述
- 如何使用此库设置USB-PD应用。

目录

1	受电方概述	6
1.1	缩写和缩略语	6
1.2	参考	6
2	USB-C PD架构	7
2.1	架构概述	7
2.2	USB-PD层	8
2.3	消息流	9
2.4	数据流	11
3	USB-PD库描述	12
3.1	概述	12
3.2	特性	13
3.3	库结构	14
3.4	硬件关联组件	15
4	USB-PD库编程指南	18
4.1	库初始化	18
4.2	USB-PD库函数	19
4.3	USB-C PD库回调函数	20
5	示例说明	23
5.1	硬件说明	23
5.2	USB-PD供电方	24
5.2.1	示例设置	24
5.2.2	应用描述	24
5.3	USB-PD供电方（可支持CLI）	25
5.4	USB-PD供电方（可支持VDM）	25
5.5	USB-PD受电方	26
5.5.1	示例设置	26
5.5.2	应用描述	27
5.6	USB-PD受电方（可支持CLI）	27

5.7	USB-PD受电方（可支持VDM）	27
5.8	USB-PD受电方DRP	28
5.8.1	示例设置	28
5.8.2	应用描述	29
5.9	USB-PD双端口	30
5.9.1	示例设置	30
5.9.2	应用描述	30
6	内存占用	32
7	常见问题（FAQs）	33
8	版本历史	34

表格索引

表1.	缩略语列表	6
表2.	使用不同IP	15
表3.	Port0所用的GPIO	15
表4.	Port1所用的GPIO	16
表5.	中断优先级	17
表6.	DPM文件	18
表7.	USB-PD用户函数	19
表8.	USB-C PD库回调	20
表9.	USB-PD - 供电方内存占用	32
表10.	USB-PD - 受电方内存占用	32
表11.	USB-PD - 双重功能端口内存占用	32
表12.	文档版本历史	34
表13.	中文文档版本历史	34

图片索引

图1.	USB电力传输架构	7
图2.	USB-PD - 消息流	10
图3.	USB-PD堆栈架构	11
图4.	项目文件	12
图5.	USB-PD堆栈架构	14
图6.	STM32FO72RB Nucleo与P-Nucleo-USB001扩展板	23

1 受电方概述

1.1 缩写和缩略语

表1. 缩略语列表

术语	意义
API	应用编程接口
CAD	线缆检测模块
CLI	命令行接口
DFP	下行端口
DPM	设备策略管理器
DRP	双角色端口
FW	固件
HW	硬件
PD	供电
PE	策略引擎
PRL	协议层
UFP	上行端口
USB	通用串行总线
VDM	供应商定义信息

1.2 参考

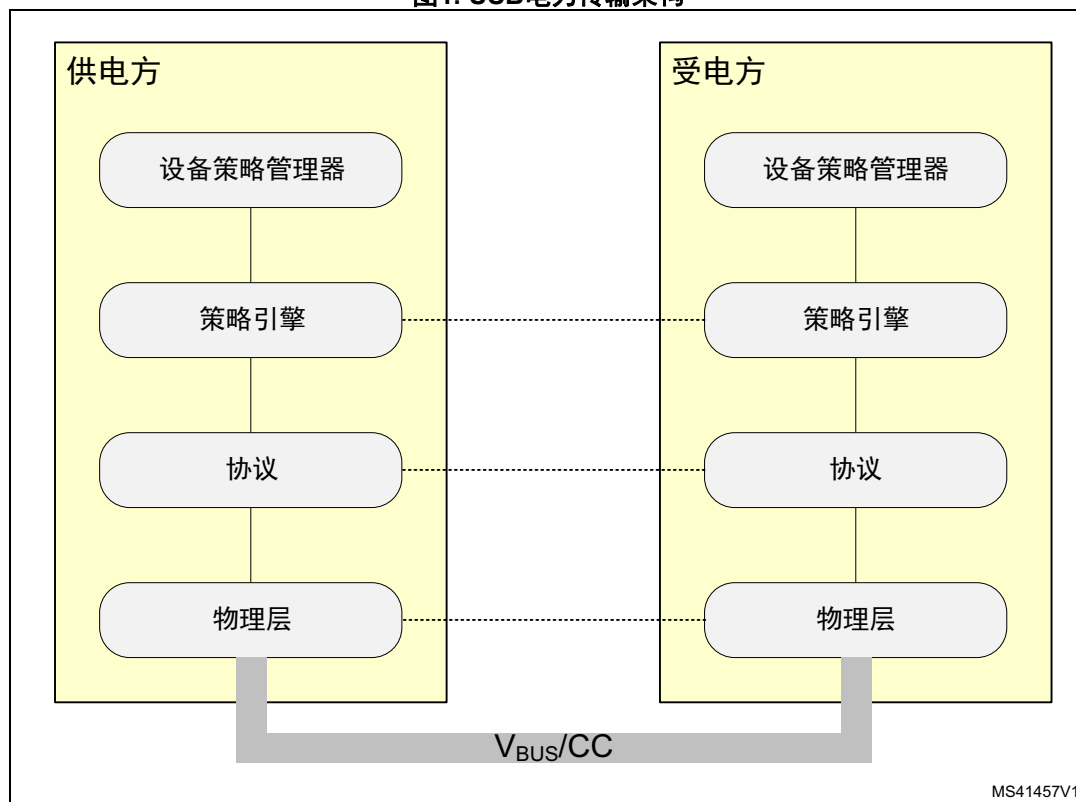
- 通用串行总线电力传输规范，版本2.0，2016年3月25日
- 通用串行总线Type-C线缆和连接器规范1.2，2016年3月25日

2 USB-C PD架构

2.1 架构概述

USB电力传输规范文档定义了PD设备（供电方或受电方）的通信层，如图1中所示。

图1. USB电力传输架构



假定一个具有PD功能的器件至少具有一个端口，可以是：

- 上行端口（UFP）
 - Sink电源（受电方）。
 - 可选source电源（受电方 / 供电方）。
 - 可选择通过USB进行通信
 - 利用SOP包通信。
- 下行端口（DFP）：
 - Source电源（供电方）。
 - 可选Sink电源（供电方 / 受电方）。
 - 可选择通过USB进行通信。
 - 利用SOP包通信
 - 可选择利用SOP包通信

在USB产品可支持USB Power Delivery协议的场合中，USB DFP最初是Source，而USB UFP最初是Sink，尽管USB-PD可以实现Source/Sink和DFP/UFP的功能交换。

注：在一对端口之间，每个PD通信中只有一个Source端口和一个Sink端口。

2.2 USB-PD层

- 设备策略管理器（DPM）
DPM的作用是管理USB Power Delivery端口所用的电源。在供电方应用中，它为受电方送电，而在受电方应用中，则向供电方要电。
一旦建立了Explicit Contact，DPM还允许交换VDM消息。Device Policy Manger是USB-C Power Delivery Stack的上一层级。
- 策略引擎（PE）层
Policy engine（PE）的作用是根据发送的消息及其预期响应来驱动消息序列。它支持协商电源，为电力交换建立Explicit Contract。
接受或拒绝请求取决于DPM对特定电源配置文件的响应。
PE还可以处理Vendor Defined Messages流，允许根据供电方和受电方所支持的功能，发现、进入或退出指定模式。
- 协议（PRL）层
PRL层的作用是驱动消息构建、发送和接收（与消息性质无关）。它允许对消息流进行检查，以检测通信错误。
PRL层是PE和PHY之间的包装层。
- PHY层
PHY层负责通过CC线发送和接收消息。它具有信号收发器，可在电线上叠加一个信号。
它负责管理线路上的数据并避免冲突，利用CRC检测消息中的错误。

注：关于USB Power Delivery协议的更多信息，请参考第 1.2 节中的官方规范文档。

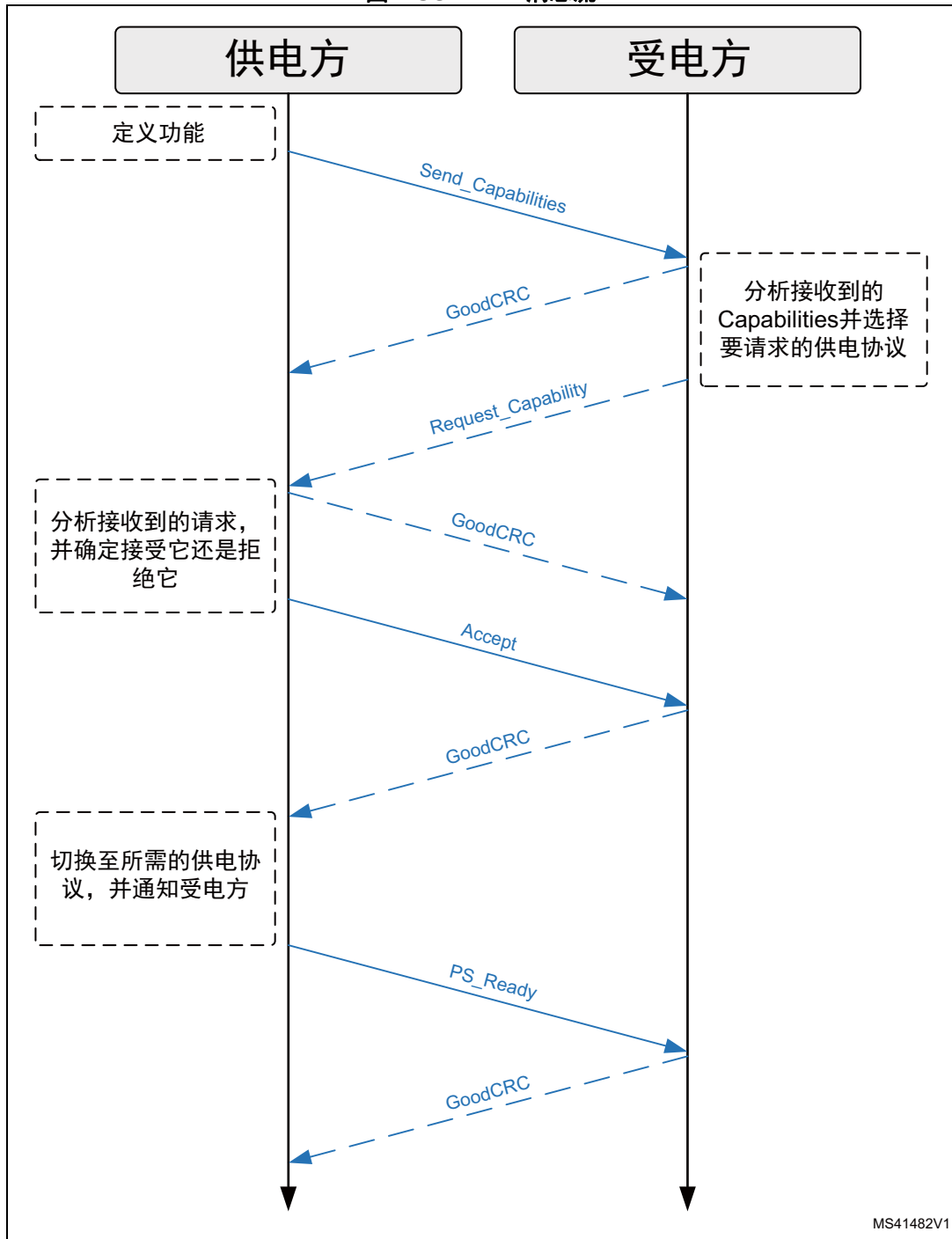
2.3 消息流

从设备发送的每个消息都应该由GoodCRC消息应答。这使得接收方可向发送方确认：其消息已被正确接收，并将由上层进行处理。

接收到的错误消息应被忽略，在持续发生通信错误的情况下，软件复位可重置协议参数，以重新建立通信。如果错误仍然存在，则执行系统的硬件复位。

正常情况下，协商供电合同应按照图 2中所示的顺序进行。

图2. USB-PD - 消息流

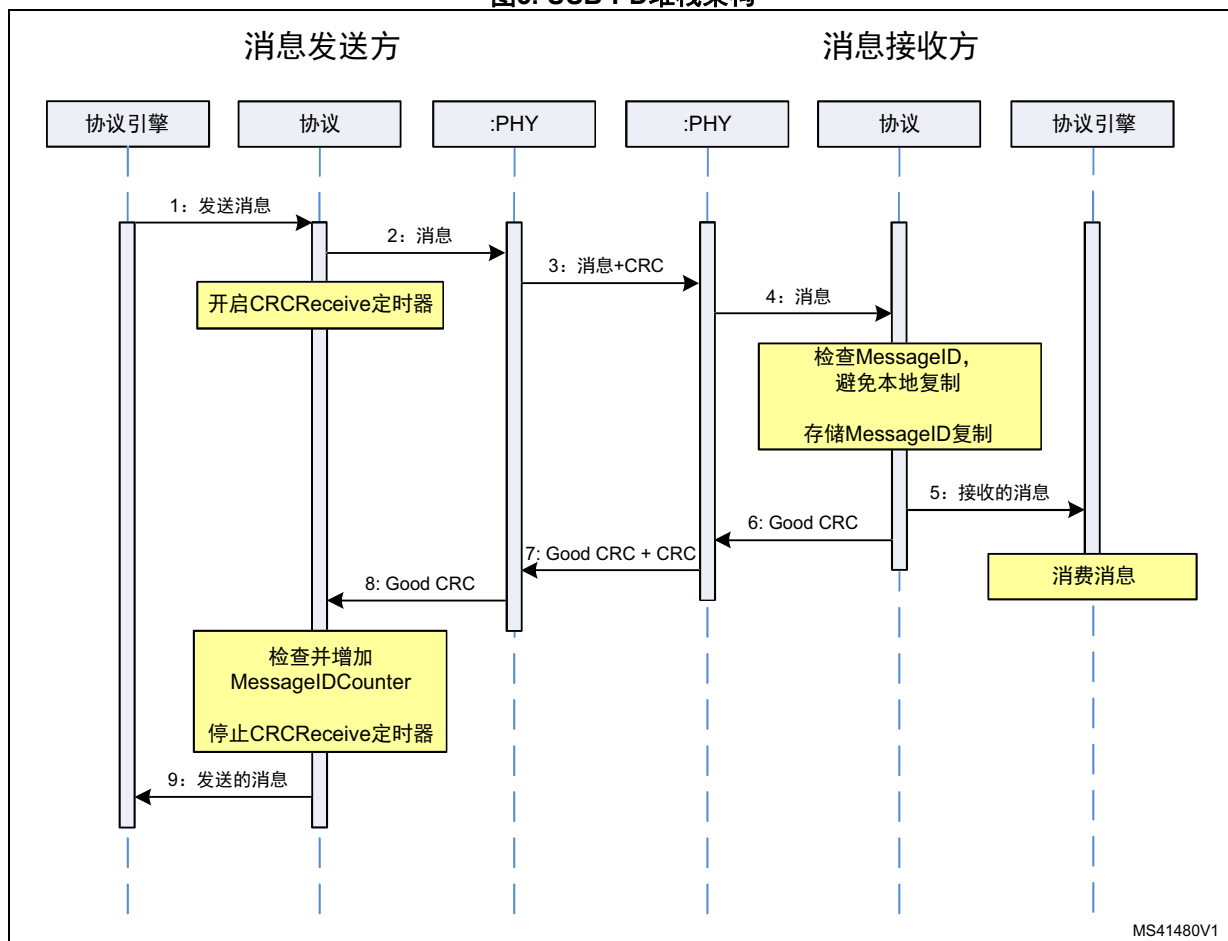


2.4 数据流

正常情况下，每条USB-PD基本消息都应从PE层向下传递到PHY层。

接收到的每条消息都应由GoodCRC消息来应答，通知发送方已正确接收（见 图 3）。BIST消息是一个例外，因为它们一旦被接收到，就不会再传递到PE层，并且只能应答为GoodCRC。

图3. USB-PD堆栈架构



3 USB-PD库描述

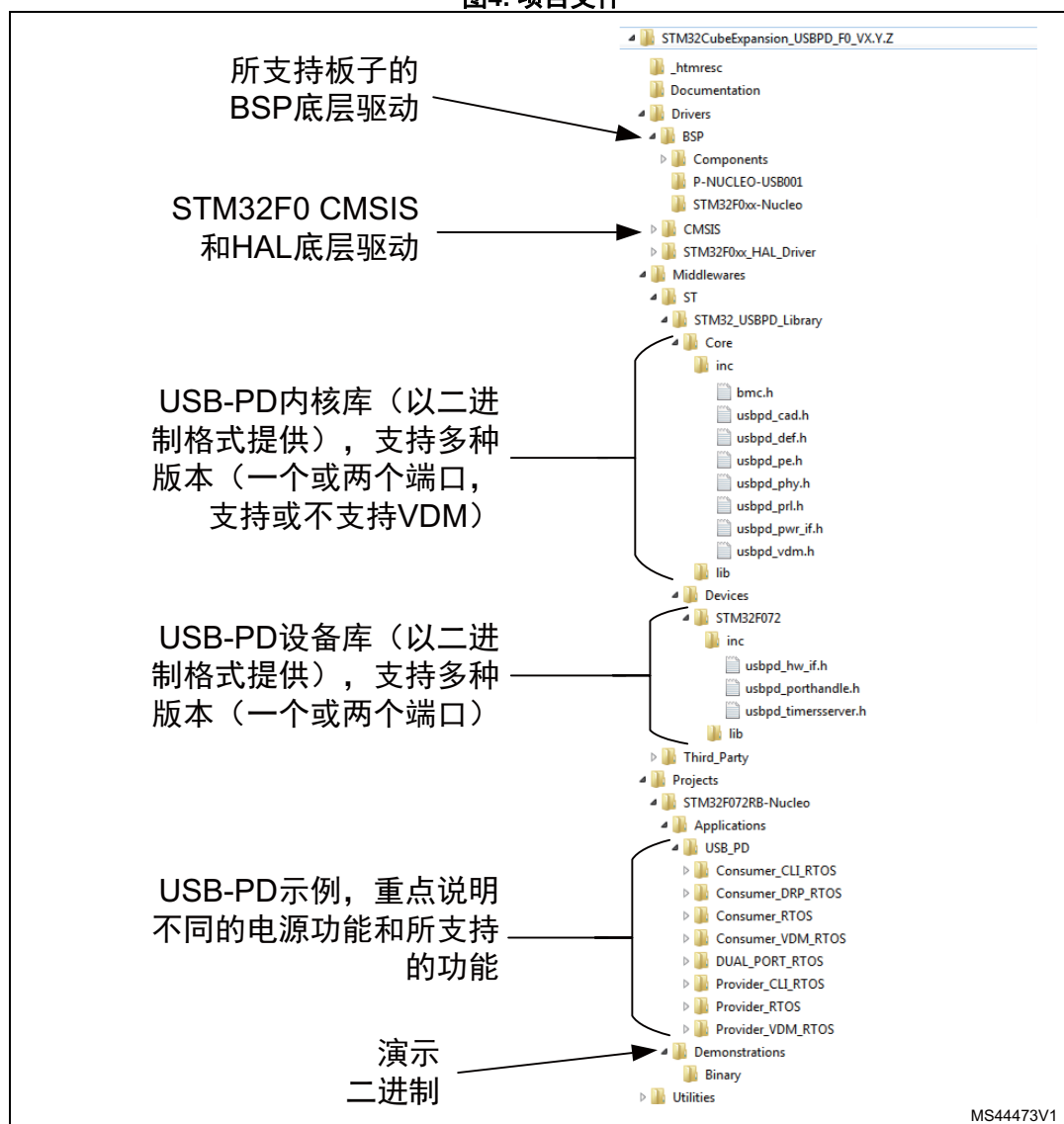
3.1 概述

意法半导体提供了USB-C Power Delivery Library，可支持STM32F0xRB微控制器。该库采用二进制格式，再加上STM32Cube HAL底层驱动，提供了开发USB PD应用所需的所有API。

本节介绍USB-PD库中间件模块，并说明用户如何使用此库开发自己的电力传输应用。

USB-PD库是根据通用串行总线电力传输规范版本2.0、V1.2（2016年3月25日）和通用串行总线type-C线缆和连接器规范，版本1.2（2016年3月25日）来开发的。它已成功通过官方认证。

图4. 项目文件



STM32 USB-PD包包括：

- USB-PD内核堆栈和设备底层驱动
- 一组示例，说明如何使用USB-PD驱动器来开发电力传输应用。这组示例包括一些STM32 Power Delivery供电方、受电方，双角色电源和双端口示例。

VDM（Vendor Defined Messages）功能的使用在供电方和受电方用例下均作了说明。

3.2 特性

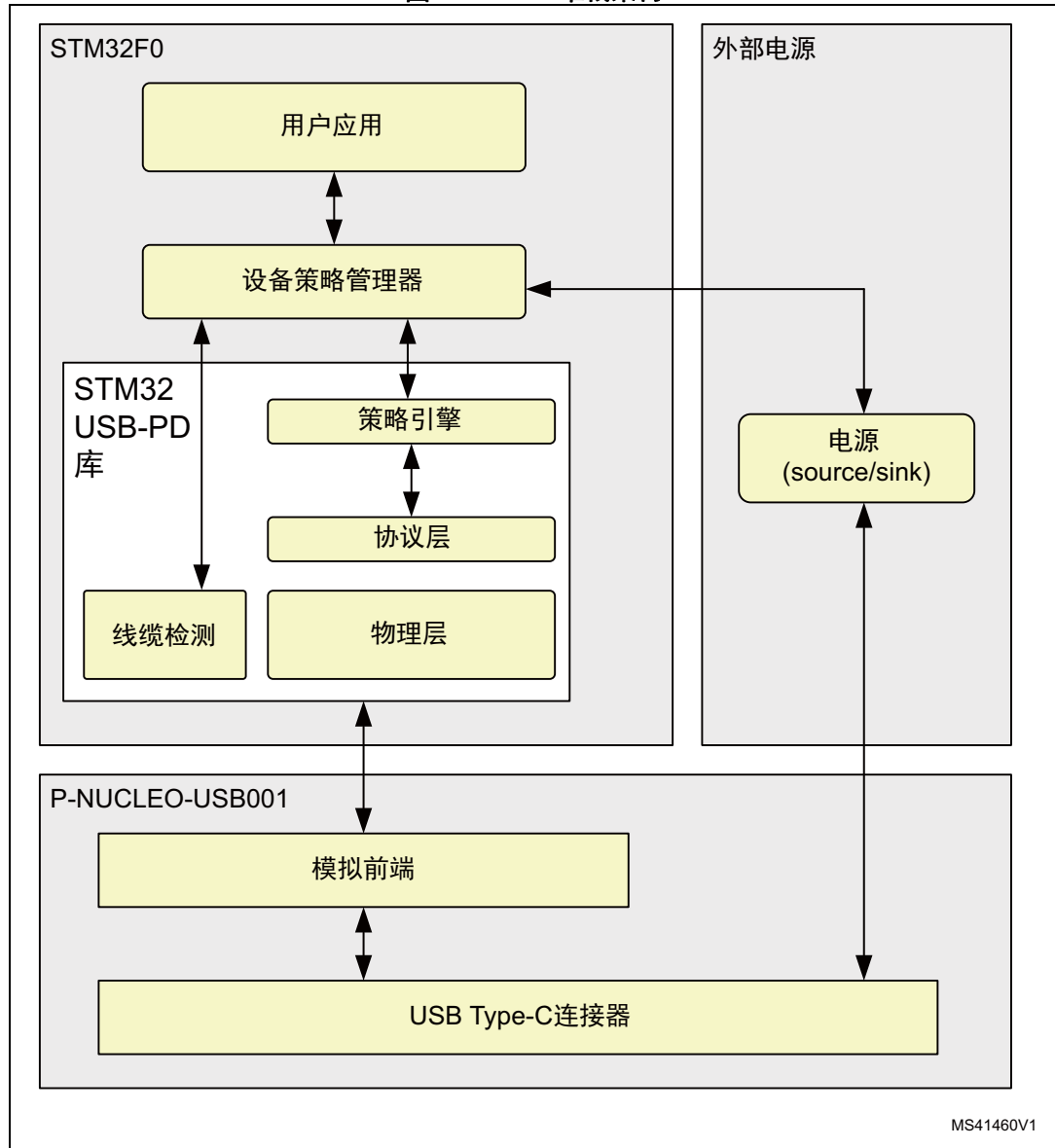
USB-PD软件扩展包的主要特性如下：

- 可兼容USB Type-C 1.2和USB Power Delivery 2.0标准
- 可支持供电方、受电方和双角色功能模式（DRP）
- 支持双端口
- 支持线缆两端的PD通信
- 线缆检测（检测USB-C线缆插入和拔除）
- 线缆定向（检测方向，允许用户选择通信线路CC1/CC2）
- USB-PD消息发送/接收：通过CC线的通信，以及消息交换，利用BMC编码/解码和5b4b编码
- 驱动VCONN开关
- BIST测试模式支持：BIST测试模式可在运行时对平台进行测试
- 结构化VDM支持可允许Alternate模式和扩展实现

3.3 库结构

USB-PD库使用USB-CPD规范架构作为构建库的参考。包含模拟前端和USB-C连接器的硬件扩展板（P-NUCLEO-USB001扩展板）与Nucleo板一起使用，以确保通信。

图5. USB-PD堆栈架构



USB-PD库涵盖两个主要部分：

- USB-PD内核堆栈（通用组件，独立于HW）：
 - 协议层
 - 策略引擎层
 - 设备策略管理器
 - 应用层
- STM32F072组件（具体）：
 - 物理层
 - 电缆检测模块（CAD）

3.4 硬件关联组件

物理层和线缆检测与HW直接相关，并需要使用一组IP（见表2）。

表2. 使用不同IP

IP	用途
CRC	Good CRC 计算
ADC1	线缆插入/拔除检测
COMP1、COMP2	数据接收
DMA1（通道2、3、4和5）	缓冲数据
SPI1、SPI2	数据发送
TIM1、TIM3、TIM14、TIM15、TIM16、TIM17	Power Delivery定时器计算 为通信用途提供时间基准

使用这些IP时，一些GPIO（见表3和4）是为库而保留的，不能用于其他用途。

表3. Port0所用的GPIO

GPIO	引脚	IP	用途
GPIOA	0	COMP1-ADC1	CC1数据接收
	1	ADC1	CC参考电压值
	3	ADC1	用于Port0的ADC通道引脚
	5	COMP1-ADC1	CC2数据接收
GPIOB	1	TIM14	数据发送
	8	-	P-NUCLEO-USB001扩展板控制
	9	-	
	12	-	
	13	SPI2	数据发送
	14	SPI2	CC1数据发送引脚

表3. Port0所用的GPIO (续)

GPIO	引脚	IP	用途
GPIOC	0	ADC1	Port0作为ADC的通道引脚
	2	SPI2	CC2数据发送引脚
	3	-	P-NUCLEO-USB001扩展板控制
	4	ADC1	Port0作为ADC的通道引脚
	8	-	P-NUCLEO-USB001扩展板控制
GIOD	2	-	P-NUCLEO-USB001扩展板控制

表4. Port1所用的GPIO

GPIO	引脚	IP	用途
GPIOA	1	ADC1	CC参考电压值
	2	COMP2-ADC1	CC1数据接收
	4	COMP2-ADC1	CC2数据接收
	6	SPI1	CC2数据发送引脚
	7	ADC1	Port0作为ADC的通道引脚
	15	-	P-NUCLEO-USB001扩展板控制
GPIOB	2	-	P-NUCLEO-USB001扩展板控制
	3	SPI1	数据发送
	4	SPI1	CC1数据发送引脚
	5	-	P-NUCLEO-USB001扩展板控制
	15	TIM15	数据发送
GPIOC	0	ADC1	Port0作为ADC的通道引脚
	5	ADC1	Port0作为ADC的通道引脚
	6	-	P-NUCLEO-USB001扩展板控制
	7	-	
	14	-	

注: 关于P-NUCLEO-USB001扩展板的更多信息和STM32所需连接, 请参考用户手册UM2050, 可从 www.st.com 上获取。

USB-C PD库定义并使用三组中断优先级，如表 5所列。

表5. 中断优先级

优先级	制程	优先值
最高	发送	0
高	接收	2
低/中	其它	3

关键和高优先级中断用于处理USBPD发送和接收过程。因此，用户应用不应定义优先级等于或低于2的中断，以免干扰PD/VDM通信。优先值高于2的中断可以随时使用。

4 USB-PD库编程指南

X-CUBE-USB-PD包包括二进制格式的库（USB-PD内核堆栈和设备组件），以及描述库API的相应头文件“.h”。

4.1 库初始化

要使用并初始化库，请按照下列步骤。

1. 使您的项目包含STM32 USB-C PD库。根据您的工具链和需要支持的功能（1个或2个端口，是否支持VDM）选择库类型。在 `Middlewares\ST\STM32_USBPD_Library\Core\lib\` 目录中选择一个库，用于USB-PD内核堆栈组件。
根据所用设备在 `Middlewares\ST\STM32_USBPD_Library\Devices\` 目录中选择第二个库，用于USB-PD组件。
对于STM32F072特殊用例，请从 `.Middlewares\ST\STM32_USBPD_Library\Devices\STM32F072\lib\` 选择设备库。
2. 在应用中包含DPM文件。一些DPM示例文件可以在提供的项目工程示例的固件中找到，如表 6 中所示。
3. 配置MCU时钟，执行一个模板，该模板可用于每个项目示例的main.c文件的 `systemClock_Config()` 函数。
4. 调用 `USBPD_HW_IF_GlobalHwInit()` 函数来初始化硬件接口层。
5. 调用 `USBPD_DPM_Init()` 函数来初始化Device Policy Manager层和所有通信层（PE, VDM, PRL, PHY）。
6. 启动RTOS Kernel（任务调度程序）。

表6. DPM文件

应用	文件位于 <code>.\Projects\STM32F072RB-Nucleo\Applications\USB_PD</code>
供电方	<code>Provider_RTOS\Inclusbpd_dpm.h</code>
	<code>Provider_RTOS\Src\usbpd_dpm.c</code>
供电方，带有 Command Line Interface	<code>Provider_CLI_RTOS\Inclusbpd_dpm.h</code>
	<code>Provider_CLI_RTOS\Src\usbpd_dpm.c</code>
供电方，可支持VDM	<code>Provider_VDM_RTOS\Inclusbpd_dpm.h</code>
	<code>Provider_VDM_RTOS\Src\usbpd_dpm.c</code>
受电方	<code>Consumer_RTOS\Inclusbpd_dpm.h</code>
	<code>Consumer_RTOS\Src\usbpd_dpm.c</code>
受电方，带有 Command Line Interface	<code>Consumer_CLI_RTOS\Inclusbpd_dpm.h</code>
	<code>Consumer_CLI_RTOS\Src\usbpd_dpm.c</code>
受电方，可支持VDM	<code>Provider_VDM_RTOS\Inclusbpd_dpm.h</code>
	<code>Provider_VDM_RTOS\Src\usbpd_dpm.c</code>
双端口，带有 Command Line Interface	<code>DUAL_PORT_RTOS\Inclusbpd_dpm.h</code>
	<code>DUAL_PORT_RTOS\Src\usbpd_dpm.c</code>

注： USB-C PD库使用FreeRTOS来管理高级层。

4.2 USB-PD库函数

DPM层从PE和VDM层调用函数来触发PD合同建立及使用Vendor Defined Messages。

表7. USB-PD用户函数

功能	说明
void USBPD_CAD_PortEnable (uint8_t PortNum, USBPD_CAD_activation State);	使能或禁用CAD端口。
USBPD_CAD_StatusTypeDef USBPD_CAD_Init(uint8_t PortNum, USBPD_CAD_Callbacks CallbackFunctions);	初始化指定端口的CAD模块。
void USBPD_CAD_AssertRd(uint8_t PortNum);	将该端口设置为Sink。
void USBPD_CAD_AssertRp(uint8_t PortNum);	将该端口设置为Source。
USBPD_StatusTypeDef USBPD_PE_Init(uint8_t PortNum, USBPD_PortPowerRole_TypeDef功能, USBPD_PE_Callbacks pcallbacks);	针对指定功能的端口初始化其Policy Engine层。
USBPD_StatusTypeDef USBPD_PE_DeInit(uint8_t PortNum);	反向初始化Policy Engine层。
void USBPD_PE_SRCProcess(uint8_t PortNum);	面向Source应用的Policy Engine处理函数。
void USBPD_PE_SNKProcess(uint8_t PortNum);	面向Sink应用的Policy Engine处理函数。
void USBPD_PE_DRPProcess(uint8_t PortNum);	面向Dual Role应用的Policy Engine处理函数。
USBPD_StatusTypeDef USBPD_PE_RequestNewPowerProfile(uint8_t PortNum, uint8_t PDOIndex);	由DPM调用，评估从Source端口接收到的Capabilities Message，并准备Request消息。
USBPD_StatusTypeDef USBPD_PE_RequestPowerRoleSwap(uint8_t PortNum);	由DPM调用，请求PE来执行Power Role Swap。
USBPD_StatusTypeDef USBPD_PE_GetReceivedPowerProfile(uint8_t PortNum, uint32_t *pPDO, uint32_t *pNbPDO);	由DPM调用，检索从source端口合作伙伴所接收到的电源配置文件。
USBPD_StatusTypeDef USBPD_PE_IsCableConnected(uint8_t PortNum, uint8_t IsConnected);	由DPM调用，设置所连接或断开的线缆状态。
USBPD_StatusTypeDef USBPD_VDM_Init(uint8_t PortNum, USBPD_VDM_Callbacks cbs);	初始化VDM层。
USBPD_StatusTypeDef USBPD_PE_SVDM_RequestIdentity(uint8_t PortNum, USBPD_SOPTType_TypeDef SOPTType);	由DPM调用，请求PE来执行VDM识别请求。

表7. USB-PD用户函数（续）

功能	说明
USBPD_StatusTypeDef USBPD_PE_SVDM_RequestModeExit(uint8_t PortNum, USBPD_SOPTType_TypeDef SOPT);	由DPM调用，请求PE来执行VDM模式退出。
USBPD_StatusTypeDef USBPD_PE_SVDM_RequestAttention(uint8_t PortNum);	由DPM调用，请求PE来执行VDM Attention。

4.3 USB-C PD库回调函数

Policy Engine或VDM层需要通知DPM一个通知或一个请求。这通过使用回调结构来使能。

表8. USB-C PD库回调

功能	说明
void (*USBPD_CAD_CallbackEvent)(uint8_t PortNum, USBPD_CAD_STATE State, CCxPin_TypeDef Cc)	由CAD调用，通知线缆插入/拔除，以及CC线。
void (*USBPD_PE_RequestSetupNewPower)(uint8_t PortNum);	请求DPM来设置新的电源水平。
uint32_t (*USBPD_PE_HardReset)(uint8_t PortNum);	请求DPM执行Hard Reset。
USBPD_StatusTypeDef (*USBPD_PE_EvaluatPRSwap)(uint8_t PortNum);	从DPM处获取PRswap请求的评估。
void (*USBPD_PE_TurnOffPower)(uint8_t PortNum, USBPD_PortPowerRole_TypeDef Role);	请求DPM关闭电源。
void (*USBPD_PE_TurnOnPower)(uint8_t PortNum, USBPD_PortPowerRole_TypeDef Role);	请求DPM接通电源。
void (*USBPD_PE_AssertRd)(uint8_t PortNum);	请求DPM声明Rd。
void (*USBPD_PE_AssertRp)(uint8_t PortNum);	请求DPM声明Rp。
void (*USBPD_PE_ExplicitContractDone)(uint8_t PortNum);	通知DPM已经建立了Explicit contract。
void (*USBPD_PE_GetDataInfo)(uint8_t PortNum, USBPD_CORE_DataInfoType_TypeDef DataId, uint32_t *Ptr, uint32_t *Size);	允许PE检索来自DPM/PWR_IF的信息。
void (*USBPD_PE_SetDataInfo)(uint8_t PortNum, USBPD_CORE_DataInfoType_TypeDef DataId, uint32_t *Ptr, uint32_t Size);	允许PE更新来自DPM/PWR_IF的信息。
USBPD_StatusTypeDef (*USBPD_PE_EvaluateRequest)(uint8_t PortNum);	允许PE请求对从Sink接收到的Request的DPM进行评估。

表8. USB-C PD库回调 (续)

功能	说明
USBPD_StatusTypeDef (*USBPD_PE_EvaluateCapabilities)(uint8_t PortNum);	允许PE请求对从Source接收到的Source Capabilities的DPM进行评估。
USBPD_StatusTypeDef (*USBPD_VDM_DiscoverIdentity)(uint8_t Port, USBPD_DiscoveryIdentity_TypeDef **pIdentity);	VDM Discovery识别回调 (Discover Identity消息的应答)。
USBPD_StatusTypeDef (*USBPD_VDM_DiscoverSVIDs)(uint8_t Port, USBPD_SVIDInfo_TypeDef **p_SVID_Info);	VDM Discover SVID回调 (检索设备支持的SVID, 以应答Discovery模式)。
USBPD_StatusTypeDef (*USBPD_VDM_DiscoverModes)(uint8_t Port, uint16_t SVID, USBPD_ModelInfo_TypeDef **p_ModelInfo);	VDM Discover Mode回调 (报告SVID所支持的所有模式)。
USBPD_StatusTypeDef (*USBPD_VDM_ModeEnter)(uint8_t Port, uint16_t SVID, uint32_t ModeIndex);	向DPM报告Mode Entry的结果。
USBPD_StatusTypeDef (*USBPD_VDM_ModeExit)(uint8_t Port, uint16_t SVID, uint32_t ModeIndex);	向DPM报告Mode Exit的结果。
USBPD_StatusTypeDef (*USBPD_VDM_InformIdentity)(uint8_t Port, USBPD_SOPTType_TypeDef SOPTType, USBPD_DiscoveryIdentity_TypeDef *Identity);	通知识别回调 (Discovery识别应答中接收到的Identity信息)。
USBPD_StatusTypeDef (*USBPD_VDM_InformSVID)(uint8_t Port, USBPD_SOPTType_TypeDef SOPTType, uint16_t SVID);	通知SVID回调。
USBPD_StatusTypeDef (*USBPD_VDM_InformMode)(uint8_t Port, USBPD_SOPTType_TypeDef SOPTType, USBPD_ModelInfo_TypeDef *ModesInfo);	通知Mode回调。
USBPD_StatusTypeDef (*USBPD_VDM_InformModeEnter)(uint8_t Port, uint16_t SVID, uint32_t ModeIndex);	通知Mode enter回调。
USBPD_StatusTypeDef (*USBPD_VDM_InformModeExit)(uint8_t Port, uint16_t SVID, uint32_t ModeIndex);	通知Mode exit回调。
uint16_t (*USBPD_VDM_GetSVID)(uint8_t Port);	得到用于Discovery Mode回调的SVID。
uint32_t (*USBPD_VDM_GetModeIndex)(uint8_t Port, uint16_t SVID);	得到SVID (Enter模式) 回调的VDO索引。
uint16_t (*USBPD_VDM_GetSVID)(uint8_t Port);	得到用于Discovery Mode回调的SVID。

表8. USB-C PD库回调 (续)

功能	说明
<code>USBPD_StatusTypeDef (*USBPD_VDM_Attention)(uint8_t Port, USBPD_AttentionInfo_TypeDef **pAttentionInfo);</code>	向VDM报告已处理attention命令。
<code>uint32_t (*USBPD_VDM_HardReset)(uint8_t Port);</code>	请求DPM执行VDM Hard Reset。

5 示例说明

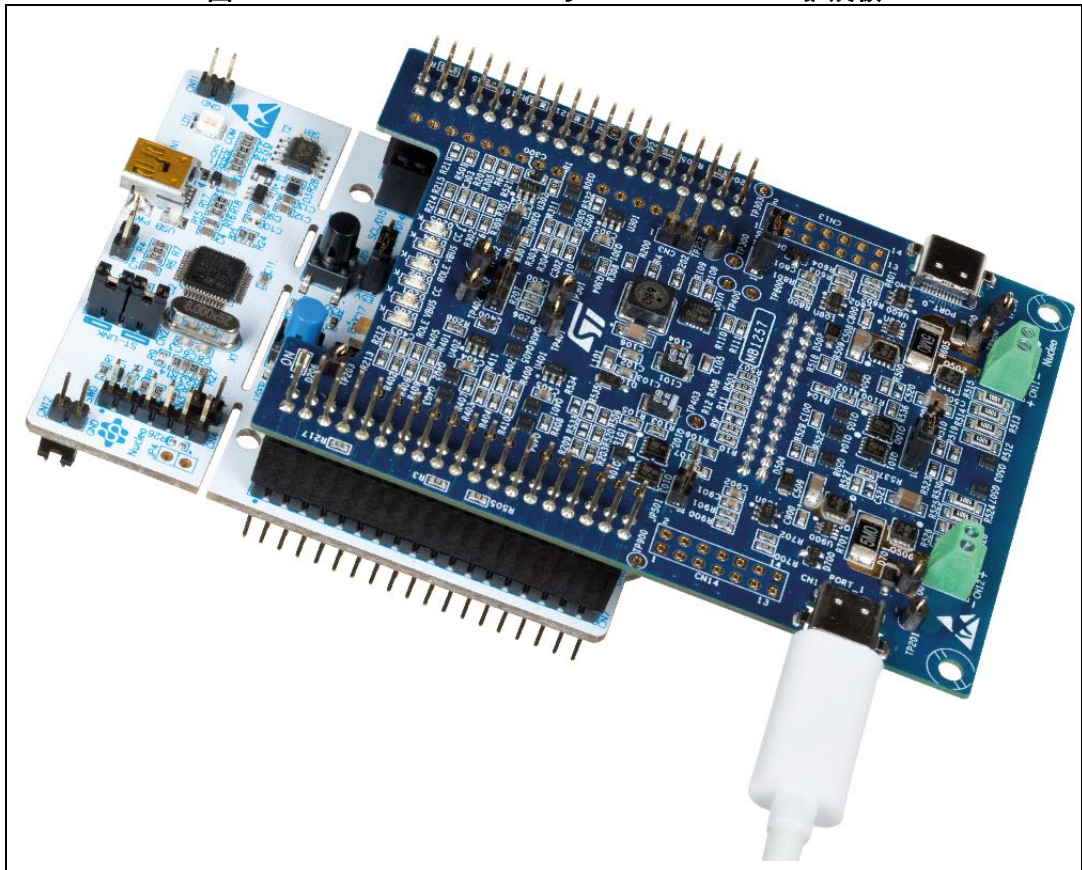
5.1 硬件说明

为了使用该库，需要P-NUCLEO-USB001扩展板作为STM32F072RB Nucleo的扩展板。P-NUCLEO-USB001屏蔽是一个扩展板，带有两个USB Type-C连接器，用来管理两个端口，具有以下特性：

- 两个双角色端口
- 专用电源连接器连接外部供电板，提供不同配置文件（可达20 V、5 A）和 V_{CONN}
- 板上电源管理能够从 V_{BUS} 提供内部所需的电压
- 六个调试LED
- 一个端口上可提供USB 2.0接口功能
- 与STM32 Nucleo板兼容
- 配有ST morpho连接器

P-NUCLEO-USB001扩展板必须连接到Nucleo板的CN7和CN10连接器。

图6. STM32F072RB Nucleo与P-Nucleo-USB001扩展板



对于提供的所有示例，用户必须

1. 打开其首选工具链
2. 重新编译所有文件，并将二进制文件加载到目标内存中
3. 运行应用。

USB-PD应用可以根据其具体供电功能（供电方，受电方，DRP），与任意具有PD功能的USB-C设备一起使用。

5.2 USB-PD供电方

此项目实现USB-PD供电方专用应用，适用于USB Type-C和Power Delivery（STM32F072 Nucleo板和USB-C PD扩展板MB1257）的STM32 Nucleo包，基于X-CUBE-USB-PD扩展包中提供的USB-PD库使用。

5.2.1 示例设置

USB-PD供电方应用可以在以下目录中找到：

Projects\STM32F072RB-Nucleo\Applications\USB_PD\Provider_RTOS

可用两种不同供电选项来管理供电方，对应两种配置设置。

1. 供电方由板上STM32F072RB-Nucleo RevC稳压器供电，利用USB Type-A到Mini-B的线缆连接到CN1连接器，然后连接到PC。

- a) 在STM32F072RB-Nucleo RevC板上，确定跳线JP1断开、JP5（PWR）闭合于U5V上（安装引脚1-2），并且JP6（IDD）闭合。
- b) 在MB1257扩展板上，闭合与供电方端口相连的跳线（对于PORT_0端口为J500，对于PORT_1为JP501）。

此设置允许从STM32F072RB-Nucleo RevC USB PWR电压（CN1连接器）开始管理所选端口上的 V_{BUS} 。

2. 供电方通过电源连接器CN4配备外部板：

- a) 在STM32F072RB-Nucleo RevC板上，必须保证以下跳线设置：JP1闭合，JP5（PWR）闭合于E5V上（安装引脚2-3），并且JP6（IDD）闭合。
- b) 在P-NUCLEO-USB001扩展板上，跳线JP100、J500和JP501必须保持断开。

此设置配置允许由外部供电板为整个系统供电，以及为端口的 V_{BUS} 提供电压（特别是对于USB PD应用）。

该应用可以通过一个连接器（通过X-NUCLEO扩展板中连接到CN0的USB-C Power Delivery线缆）到USB-C Power Delivery受电方设备来进行测试。

装载了Consumer_RTOS应用程序的第二块板可用来作为连接的受电方设备。

5.2.2 应用描述

此应用提供了一个Port0作为供电方专用端口的示例。

当应用启动时，连接一个USB-C Power Delivery受电方设备（sink模式）来触发电源协商：

- 开始时，Role LED（LED D203）将会闪烁，表示Port角色（只闪烁一下表示供电方功能）。
- 用户应将USB-C线缆插入专用连接器。连接时，如果连接到CC1，则CC LEDs（D205）将会闪烁一下，如果连接到CC2则会闪烁两下。
- 蓝色LED（D203）将会闪烁一下，表示设备作为供电方。
- STM32 MCU作为供电方（source模式），它会与连接的设备交换电源配置文件，并等待来自所连接受电方的Power Request消息。
- 连接之后，且在建立Explicit Contract之前，V_{BUS} LED（D204）将会闪烁。
- 如果达到了所请求的电源，STM32 MCU将会发送Accept消息，后跟PS_RDY消息。
- 一旦建立了Explicit Contract，V_{BUS} LED（D204）就会亮起，表示Power Contract已建立。

5.3 USB-PD供电方（可支持CLI）

此应用类似于USB-PD供电方应用，因此第 5.2 节中提供的说明也适用于此项目，区别在于本应用可支持命令行界面（CLI）。

该应用嵌入了命令行界面（Command Line interface, CLI）功能，使用户能够获取Port0上运行的Power Delivery应用的状态，并可通过串行通信链路对其进行操作。

详情请参考UM2051“面向USB Type-C™和Power Delivery的STM32 Nucleo包入门”，可从www.st.com上获取：

- CLI应用的板配置
- HyperTerminal配置（PC端）
- 可用的CLI命令行和参数。

在此应用中，只有一个端口（Port0）被处理并随后由CLI命令访问。

USB-PD供电方（可支持CLI）应用可在 Projects\STM32F072RB-Nucleo\Applications\USB_PD\Provider_CLI_RTOS下找到

5.4 USB-PD供电方（可支持VDM）

此应用类似于USB-PD供电方应用，因此第 5.2 节中提供的说明也适用于此项目，区别在于本应用可支持Vendor Defined Messages。

除了将Port0作为供电方专用端口之外，此应用还提供了一个VDM消息示例，用来实现设备之间USB-PD相关消息以外的信息交换。执行供电方应用来产生SVDM消息。提供了VDM回调作为示例。

这些回调允许用户模拟进入DP模式（一旦允许进入模式时，所有LED都会闪烁）。

为了测试此应用的Power Delivery部分，Consumer_RTOS应用可作为第二块板上的受电方。

连接到对应Port0性能的设备时可对Vendor Defined Messages进行测试（该示例的完整定义在所提供的VIF文件中），或使用支持VDM功能的USB PD一致性测试仪。

支持Vendor Defined Messages应用的USB-PD供电方可在Projects\STM32F072RB-Nucleo\Applications\USB_PD\Provider_VDM_RTOS下找到

5.5 USB-PD受电方

此项目基于扩展包提供的USB-PD库，实现了USB-PD受电方专用应用，适用于面向USB Type-C和Power Delivery（STM32F072 Nucleo板和USB-C PD扩展板MB1257）的STM32 Nucleo包，基于X-CUBE-USB-PD扩展包提供的USB-PD库使用。

5.5.1 示例设置

USB-PD受电方应用可在以下目录中找到：

Projects\STM32F072RB-Nucleo\Applications\USB_PD\Provider_VDM_RTOS

系统可以管理受电方配置的两种供电选项。第一种由NUCLEO-F072RB供电，而第二种则实现了USB PD解决方案的特定功能（比如，当受电方由供电方通过其 V_{BUS} 来供电时）。

两种配置对应两种不同的设置：

- 如果受电方由NUCLEO-F072RB稳压器供电，则系统设置如下：
 - 在NUCLEO-F072RB板上，确定跳线JP1断开，JP5（PWR）闭合于U5V上（安装引脚1-2），并且JP6（IDD）闭合。
 - 在P-NUCLEO-USB001扩展板上，断开跳线JP100、J500、JP501。
- 如果受电方由 V_{BUS} （由USB Type-C线连接的供电方提供）供电，那么系统设置如下：
 - 在NUCLEO-F072RB板上，JP1必须是闭合的，JP5（PWR）闭合于E5V上（安装引脚2-3），并且JP6（IDD）闭合。
 - 在P-NUCLEO-USB001扩展板上，当跳线J500、JP501断开时，跳线JP100必须根据为系统供电所选的端口来设置（对于PORT_0配置为2-3，对于PORT_1则配置为1-2）。

该应用可以通过一个连接器（通过X-NUCLEO扩展板中连接到CN0的USB-C Power Delivery线缆）到USB-C Power Delivery受电方设备来进行测试。

装载了Provider_RTOS应用程序的第二块板可用来作为连接的供电方设备。

5.5.2 应用描述

当应用启动时，连接一个USB-C Power Delivery供电方设备（source模式）来触发电源协商：

- 开始时，Role LED（LED D203）将会闪烁，表示Port角色（闪烁两下表示受电方角色）。
- 用户应将USB-C线缆插入专用连接器。
- 连接时，如果连接到CC1，则CC LED（D205）将会闪烁一下，如果连接到CC2则会闪烁两下。
- 蓝色LED（D203）每次会闪烁两下，表示设备作为受电方。
- STM32MCU作为受电方（sink模式），会等待来自所连接供电方的PowerCapabilities消息。当接收到Source Capabilities消息时，STM32开始评估接收性能，并检查所接收的电源对象中是否有一个能满足供电要求。
- 通信时，V_{BUS} LED（D204）会闪烁。
- STM32将发送一条消息，向所提供的Source Capabilities请求新的电源。
- 一旦建立了ExplicitContract（接收到PS_Ready消息），V_{BUS}LED（D204）就会亮起，表示Power Contract已建立。

5.6 USB-PD受电方（可支持CLI）

此应用类似于USB-PD受电方应用，因此第 5.5节中提供的说明也适用于此项目，区别在于本应用可支持命令行界面（CLI）。

该应用嵌入了命令行界面（CLI）功能，使用户能够获取Port0上运行的Power Delivery应用的状态，并可通过串行通信链路对其进行操作。

详情请参考UM2051“面向USB Type-C™和Power Delivery的STM32 Nucleo包入门”，可从www.st.com上获取：

- CLI应用的板配置
- HyperTerminal配置（PC端）
- 可用的CLI命令行和参数。

在此应用中，只有一个端口（Port0）被处理并随后由CLI命令访问。

USB-PD受电方（可支持CLI）应用可在 Projects\STM32F072RB-Nucleo\Applications\USB_PD\Consumer_CLI_RTOS下找到

5.7 USB-PD受电方（可支持VDM）

此应用类似于USB-PD受电方应用，因此第 5.5节中提供的说明也适用于此项目，区别在于本项目中可以支持Vendor Defined Messages。

除了将Port0作为受电方专用端口之外，此应用还提供了一个VDM消息示例，可以实现设备之间对USB-PD相关消息以外的信息交换。执行受电方应用程序来应答SVDM消息。提供了VDM回调作为示例。

这些回调允许用户模拟进入DP模式（一旦允许进入模式时，所有LED都应闪烁）。

为了测试此应用的Power Delivery部分，Provider_RTOS应用可作为第二块板上的供电方（Source）。

连接一个与Port0性能对应的设备（示例的完整定义在所提供的VIF文件中），可以测试Vendor Defined Messages，或者使用支持VDM功能的USB PD一致性测试仪来对其进行测试。

支持Vendor Defined Messages应用的USB-PD受电方可在Projects\STM32F072RB-Nucleo\Applications\USB_PD\Consumer_VDM_RTOS下找到

5.8 USB-PD受电方DRP

该项目实现了USB-PD Dual Role Port（DRP）应用，适用于面向USB Type-C和Power Delivery的STM32 Nucleo包（STM32F072 Nucleo板和USB-C PD扩展板MB1257），基于X-CUBE-USB-PD扩展板提供的USB-PD库。

5.8.1 示例设置

USB-PD Dual Role Port（DRP）应用可在 Projects\STM32F072RB-Nucleo\Applications\USB_PD\Consumer_DRP_RTOS下找到

根据Port0连接另一个设备（供电方或受电方）之后的角色功能，系统可以为Port0提供两个供电选项。

- 系统由板上STM32F072RB-Nucleo RevC稳压器供电，利用USB Type-A到Mini-B的线缆连接到CN1连接器，然后连接到PC。
 - 在STM32F072RB-Nucleo RevC板上，确定跳线JP1断开、JP5（PWR）闭合于U5V上（安装引脚1-2），并且JP6（IDD）闭合。
 - 在MB1257扩展板上，闭合与供电方端口相连的跳线（对于PORT_0端口为J500）。此设置允许从STM32F072RB-Nucleo RevC USB PWR电压（CN1连接器）开始管理所选端口上的 V_{BUS} 。
- 系统通过电源连接器CN4配备外部板：
 - 在STM32F072RB-Nucleo RevC板上，必须保证以下跳线设置：JP1闭合，JP5（PWR）闭合于E5V上（安装引脚2-3），并且JP6（IDD）闭合。
 - 在P-NUCLEO-USB001扩展板上，当跳线J500、JP501断开时，跳线JP100必须根据为系统供电所选的端口来设置（对于PORT_0配置为2-3，对于PORT_1则配置为1-2）。

作为供电方的情况下，此设置配置允许由外部供电板为整个系统供电，以及为端口的 V_{BUS} 提供电压（特别是对于USB PD应用）。

作为受电方的情况下，受电方由 V_{BUS} （由USB Type-C线连接的供电方提供）供电。

该应用可以通过一个连接器（通过X-NUCLEO扩展板中连接到CN0的USB-C Power Delivery电缆）到USB-C Power Delivery设备，作为受电方、供电方或DRP设备来进行测试。

可以使用另一块装载了相应应用程序的板。

5.8.2 应用描述

该应用程序提供了一个示例，根据所连接的设备，将Port0配置为具有双角色功能，可以作为供电方或受电方。连接之后，还可支持Power Role交换：

- 连接到USB-C供电方专用设备（source模式）时，两块板之间不能进行电源功能交换，DRP板将自动成为sink。
- 连接到USB-C受电方专用设备（sink模式）时，两块板之间不能进行电源功能交换，DRP板将自动成为source。
- 使用DRP连接到USB-C时，每次按下用户按钮，都会进行电源功能交换。
- 应用程序启动时，Port0可以作为Source或Sink来工作。在Port0上连接USB-PD设备（source或sink）时，应用程序能够检测出所连接设备的类型，并采取相应合适的功能来触发电源协商：
- 开始时，Role LED（LED D203）将会闪烁，表示Port功能（闪烁三下表示DRP）。
- 用户应将USB-C线缆插入专用连接器。
- 当STM32 MCU作为受电方（sink模式），即连接到source设备时，会等待来自所连接供电方的Power Capabilities消息。当接收到Source Capabilities消息时，STM32开始评估接收性能，并检查所接收的电源对象中是否有一个能满足供电要求。STM32将发送一条Request消息，向所提供的Source Capabilities请求新的电源。一旦建立了ExplicitContract（接收到PS_Ready消息）， V_{BUS} LED（D204）就会亮起，表示Power Contract已建立。
- 当STM32 MCU作为供电方（source模式），即连接到Sink设备时，它会与连接的设备交换电源配置文件，并等待来自所连接受电方的Power Request消息。如果达到了所请求的电源，STM32 MCU将会发送Accept消息，后跟PS_RDY消息。一旦建立了Explicit Contract， V_{BUS} LED（D204）就会亮起，表示Power Contract已建立。
- 连接之后，且在建立Explicit Contract之前， V_{BUS} LED（D204）将会闪烁。
- 连接时，如果连接到CC1，则CC LED（D205）将会闪烁一下，如果连接到CC2则会闪烁两下。
- 如果设备作为受电方，则Role（蓝色）LED（D203）每次会闪烁两下，如果设备作为供电方，则它每次闪烁一下。

5.9 USB-PD双端口

该项目实现了USB-PD Dual Role Port应用，适用于面向USB Type-C和Power Delivery的STM32 Nucleo包（STM32F072 Nucleo板和USB-C PD扩展板MB1257），基于X-CUBE-USB-PD扩展包中提供的USB-PD库。

5.9.1 示例设置

USB-PD Dual Port应用可以在以下目录中找到：
Projects\STM32F072RB-Nucleo\Applications\USB_PD\DUAL_PORT_RTOS

根据Port0和Port1连接另一个设备（供电方或受电方）之后的角色功能，系统可以为Port0和Port1提供两个供电选项。

- 系统由板上STM32F072RB-Nucleo RevC稳压器供电，利用USB Type-A到Mini-B的线缆连接到CN1连接器，然后连接到PC。
 - 在STM32F072RB-Nucleo RevC板上，确定跳线JP1断开、JP5（PWR）闭合于U5V上（安装引脚1-2），并且JP6（IDD）闭合。
 - 在MB1257扩展板上，闭合与供电方端口相连的跳线（对于PORT_0端口为J500）。此设置允许从STM32F072RB-Nucleo RevC USB PWR电压（CN1连接器）开始管理所选端口上的 V_{BUS} 。
- 系统通过电源连接器CN4配备外部板：
 - 在STM32F072RB-Nucleo RevC板上，必须保证以下跳线设置：JP1闭合，JP5（PWR）闭合于E5V上（安装引脚2-3），并且JP6（IDD）闭合。
 - 在P-NUCLEO-USB001扩展板上，当跳线J500、JP501断开时，跳线JP100必须根据为系统供电所选的端口来设置（对于PORT_0配置为2-3，对于PORT_1则配置为1-2）。

作为供电方的情况下，此设置配置允许由外部供电板为整个系统供电，以及为端口的 V_{BUS} 提供电压（特别是对于USB PD应用）。

作为受电方的情况下，受电方由VBUS（由USB Type-C线连接的供电方提供）供电。

该应用可以通过一个连接器（通过X-NUCLEO扩展板中连接到CN0的USB-C Power Delivery电缆）到USB-C Power Delivery设备，作为受电方、供电方或DRP设备来进行测试。

可以使用另一块装载了相应应用程序的板。

5.9.2 应用描述

此应用提供了同时管理Port0和Port1的示例。两个端口均为DRP，并且独立运行。用户应将USB-C线插入专用连接器（Port0：CN0，或Port1：CN1）。

应用程序启动时，每个端口都可以作为Source或Sink来工作。在给定端口上连接USB-C Power Delivery设备（source或sink）时，应用程序能够检测出所连接设备的类型，并采取相应合适的功能来触发电源协商：

- 开始时，RoleLED（对于Port0为LEDD203，对于Port1为LEDD200）将会闪烁，表示端口角色（闪烁三次表示DRP Role）。
- 用户应将USB-C线插入专用连接器（对于Port0为CN0，对于Port1为CN1）。
- 当端口作为受电方（sink模式），即连接到Source设备时，会等待来自所连接供电方的Power Capabilities消息。
- 当接收到Source Capabilities消息时，STM32开始评估接收性能，并检查所接收的电源对象中是否有一个能满足供电要求。STM32随后会发送一条Request消息，向所提供的Source Capabilities请求新的电源。一旦建立了Explicit Contract（接收到PS_Ready消息），V_{BUS}LED（对于Port0为LEDD204，对于Port1为LEDD201）就会亮起，表示Power Contract已建立。
- 当端口作为供电方（source模式），即连接到Sink设备时，它会与连接的设备交换电源配置文件，并等待来自所连接受电方的Power Request消息。如果达到了所请求的电源，STM32 MCU将会发送Accept消息，后跟PS_RDY消息。一旦建立了Explicit Contract，V_{BUS} LED（对于Port0为LED D204，对于Port1为LED D201）就会亮起，表示Power Contract已建立。
- 连接之后，且在建立Explicit Contract之前，V_{BUS} LED（对于Port0为D204，对于Port1为LED D201）将会闪烁。
- 连接时，如果连接到CC1，则CC LED（对于Port0为D205，对于Port1为LED D202）将会闪烁一下，如果连接到CC2则会闪烁两下。
- 如果设备作为受电方，则Role（蓝色）LED（对于Port0为LEDD203，对于Port1为LED D200）每次会闪烁两下，如果设备作为供电方，则它每次闪烁一下。

该应用还嵌入了命令行界面（CLI）功能，使用户能够获取Port0和Port1上运行的Power Delivery应用的状态，并可通过串行通信与应用进行交互。

详情请参考UM2051“面向USB Type-C™和Power Delivery的STM32 Nucleo包入门”，可从www.st.com上获取：

- CLI应用的板配置
- HyperTerminal配置（PC端）
- 可用的CLI命令行和参数。

本应用中，两个端口（Port0和Port1）都可通过CLI命令访问。

6 内存占用

表9至11中的值根据以下配置计算得出：

- 编译器：Embedded Workbench[®] for ARM[®]，版本7.50.1
- 优化：高速
- MCU：STM32F072RB
- 扩展板：P-NUCLEO-USB001扩展板

表9. USB-PD - 供电方内存占用

Project	供电方（基于RTOS）		说明
	闪存（字节）	RAM（字节）	
USB-PD库	15014	2329	USB-PD库所需要的内存。
应用层	18489	3863	用户应用程序，HAL驱动程序，实用程序，FreeRTOS，LED管理及其他。
总计	33053	6192	总内存。

表10. USB-PD - 受电方内存占用

Project	受电方（基于RTOS）		说明
	闪存（字节）	RAM（字节）	
USB-PD库	14815	2389	USB-PD库所需要的内存。
应用层	18529	3811	用户应用程序，HAL驱动程序，实用程序，FreeRTOS，LED管理及其他。
总计	33344	6200	总内存。

表11. USB-PD - 双重功能端口内存占用

Project	DRP（基于RTOS）		说明
	闪存（字节）	RAM（字节）	
USB-PD库	16028	2389	USB-PD库所需要的内存。
应用层	18768	3811	用户应用程序，HAL驱动程序，实用程序，FreeRTOS，LED管理及其他。
总计	34796	6200	总内存。

注：DRP值对于供电方DRP和受电方DRP都有效。

7 常见问题（FAQs）

如何获取STM32 USB-PD库？

该库为二进制格式，可以从www.st.com上免费下载。

该库是否支持USB数据通信？

此库仅支持PD通信，但是第一个端口可以携带USB数据。可以添加STM32 USB库，以便通过第一个端口实现USB通信。

我只想使用USB-C功能（线缆插入/拔除和线缆定向）。是否可以？

可以，因为CAD（线缆插入和拔除）模块和PD通信由两个独立的过程来驱动。您可以只调用CAD过程来实现线缆检测。

X-CUBE-USB-PD扩展包是否适用于与STM32F0不同的平台？

内核堆栈与设备独立，但是在此交付产品中，设备部分仅支持STM32F0平台。未来会增强软件包，使其能够支持其他STM32微控制器。

使用供电方，如何为需要5 V以上电压值的受电方端口供电？

要提供更高的电压值（高达20 V），必须通过连接器CN4为P-NUCLEO-USB001扩展板连接外部供电板（详情请参考UM2050）。

8 版本历史

表12. 文档版本历史

日期	版本	变更
2016年6月8日	1	初始版本。
2017年1月23日	2	<p>更新了第 1.2 节: 参考, 第 3.1 节: 概述, 第 3.2 节: 特性, 第 3.3 节: 库结构, 第 3.4 节: 硬件关联组件, 第 4 节: USB-PD库编程指南, 第 4.1 节: 库初始化, 第 4.2 节: USB-PD库函数, 第 4.3 节: USB-CPD库回调函数, 第 5.1 节: 硬件说明, 第 5.2 节: USB-PD供电方, 第 5.5 节: USB-PD受电方, 第 5.8 节: USB-PD受电方DRP和第 7 节: 常见问题 (FAQs)。</p> <p>增加了第 5.3 节: USB-PD供电方 (可支持CLI), 第 5.4 节: USB-PD供电方 (可支持VDM), 第 5.6 节: USB-PD受电方 (可支持CLI), 第 5.7 节: USB-PD受电方 (可支持VDM) 和第 5.9 节: USB-PD双端口。</p> <p>更新了表 1: 缩略语列表, 表 2: 使用不同IP, 表 3: Port0所用的GPIO, 表 4: Port1所用的GPIO, 表 6: DPM文件, 表 7: USB-PD用户函数, 表 8: USB-C PD库回调, 表 9: USB-PD - 供电方内存占用, 表 10: USB-PD-受电方内存占用和表 11: USB-PD-双重功能端口内存占用。</p> <p>更新了图 4: 项目文件。</p>

表13. 中文文档版本历史

日期	版本	变更
2017年9月29日	1	中文初始版本。

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利