Introduction

This document describes the STM32 USB-PD (Power Delivery) Expansion Package for STM32Cube, referenced as X-CUBE-USB-PD.

To discover all our MCU based solutions for USB Type-C™ and Power Delivery technology please go to our landing page https://www.st.com/content/st_com/en/stm32-usb-c.html.

X-CUBE-USB-PD is a USB-IF certified Expansion Package and consists of libraries, drivers, sources, APIs, and application examples running on STM32F0 Series microcontrollers acting as USB Type-C™ port manager (TCPM).

The provided example helps to develop applications based on USB-PD DRP (Dual Role Power).

The core of the stack is delivered in library format while the open-source format device part offers a high level of flexibility to match the design considerations.

This Expansion Package supports one hardware implementation:

- Standardized TCPM/TCPC solution for any STM32 microcontroller
  - Ideal solution to upgrade legacy design based-on any STM32 with USB-C™
  - Lowest memory footprint and easy porting within the Cortex®-M series
  - USB-PD 2.0/3.0+PPS compliant, multi-port
  - Tested with TCPC controller from On-SEMI FUSB307B. Refer to USB Type-C™ port manager (TCPM)/port controller (TCPC) evaluation board (DB3623).
## Contents

1. **Overview** ............................................................. 5  
   1.1 Acronyms and abbreviations ........................................ 5  
   1.2 References .......................................................... 6  

2. **USB-C™ PD architecture** ............................................. 7  
   2.1 Architecture overview .............................................. 7  
   2.2 Specific case of TCPM/TCPC architecture ......................... 8  

3. **USB-PD ST application setup** ...................................... 11  

4. **Examples description** ............................................... 12  
   4.1 USB-PD DRP (TCPM/TCPC architecture) ............................ 12  
      4.1.1 Example setup .................................................. 12  
      4.1.2 Application description ....................................... 12  
      4.1.3 Library initialization ......................................... 13  
      4.1.4 TCPM API description ......................................... 13  

5. Memory footprint of the EVAL_FUSB307_DRP example ............... 14  

6. **Frequently asked questions (FAQs)** .............................. 15  

7. **Revision history** .................................................... 16
List of tables

Table 1. List of acronyms ................................................................. 5
Table 2. USB-PD - Dual role port memory footprint (in bytes) ..................... 14
Table 3. Document revision history .................................................. 16
List of figures

Figure 1. USB Power Delivery architecture ............................................................... 7
Figure 2. USB Type-C™ port manager to USB Type-C™ port controller interface .......... 8
Figure 3. ST firmware architecture ........................................................................ 9
Figure 4. Project structure ...................................................................................... 12
1 Overview

This document describes how to use the USB-PD library supporting STM32 32-bit microcontrollers based on Arm® cores, for regular use and to create a customized application.

It covers the following topics to ease the use of the library:
- USB-PD standard overview
- USB-PD library architecture
- USB-PD stack usage description

1.1 Acronyms and abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application programming interface</td>
</tr>
<tr>
<td>CAD</td>
<td>Cable detection module</td>
</tr>
<tr>
<td>CC</td>
<td>Configuration channel</td>
</tr>
<tr>
<td>DFP</td>
<td>Downstream facing port</td>
</tr>
<tr>
<td>DPM</td>
<td>Device policy manager</td>
</tr>
<tr>
<td>DRP</td>
<td>Dual role port</td>
</tr>
<tr>
<td>FW</td>
<td>Firmware</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>PD</td>
<td>Power delivery</td>
</tr>
<tr>
<td>PE</td>
<td>Policy engine</td>
</tr>
<tr>
<td>PRL</td>
<td>Protocol layer</td>
</tr>
<tr>
<td>TCPC</td>
<td>USB Type-C™ port controller</td>
</tr>
<tr>
<td>TCPCI</td>
<td>USB Type-C™ port controller interface</td>
</tr>
<tr>
<td>TCPM</td>
<td>USB Type-C™ port manager</td>
</tr>
<tr>
<td>UFP</td>
<td>Upstream facing port</td>
</tr>
<tr>
<td>USB</td>
<td>Universal serial bus</td>
</tr>
<tr>
<td>VDM</td>
<td>Vendor defined messages</td>
</tr>
</tbody>
</table>

---

a. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
1.2 References

- Universal Serial Bus Power Delivery Specification, Revision 3.0, Version 2.0, August 29, 2019
- Universal Serial Bus Type-C™ Cable and Connector Specification 2.0, August 2019
- Universal Serial Bus Type-C™ Port Controller Interface Specification, Revision 1.0, Version 1.2, November 2016
- User manual *Managing USB power delivery systems with STM32 microcontrollers* (UM2552)
- Application note *USB Type-C™ Power Delivery using STM32xx Series MCUs and STM32xxx Series MPUs* (AN5225)
- ST wiki USBPD overview: https://wiki.st.com/stm32mcu/wiki/USB_Power_Delivery_overview
2  USB-C™ PD architecture

2.1  Architecture overview

The USB Power Delivery specification document defines the communicating layers of a PD device (either provider or consumer) as shown in Figure 1.

![Figure 1. USB Power Delivery architecture](image)

A PD 3.0-capable device is assumed to be made up of at least one port, which can
- sink power (a consumer), source power (a provider), or toggle between the two roles (dual role power)
- optionally communicate via USB
- communicate using SOP Packets
- optionally communicate using SOP Prime packets.

Where USB products support USB Power Delivery protocols a USB DFP is initially a Source and a USB UFP is initially a Sink, although USB-PD enables the Source/Sink and the DFP/UFP roles to be swapped.

Note: There is only one Source port and one Sink port in each PD communication between port partners.
2.2 Specific case of TCPM/TCPC architecture

Implementation of USB-PD function may also be achieved in a TCPM/TCPC architecture, with a standardized interface between USB Type-C™ port manager (TCPM) and a simple USB Type-C™ port controller (TCPC) with the goal of easing USB Type-C™ Power Delivery implementations.

![Figure 2. USB Type-C™ port manager to USB Type-C™ port controller interface](image)

TCPC interface (TCPCi) is using an I²C link and an alert pin. USB Type-C™ port controller interface Specification (TCPCI) defines the interface between the TCPM and the TCPC(s). In this architecture, one TCPM may be used to drive multiple TCPCs.

Protocol (PRL) functionalities are shared between TCPM and TCPC. TCPC must manage:
- GoodCRC management
- Retransmission

ST firmware architecture is described in Figure 3:
Device Policy Manager (DPM)
DPM’s role is to manage all the ports present in the application and to share the power between them according to the capability of the connected port partners. Once the negotiation of power has ended, an explicit contract is concluded between them. DPM can manage VDM exchanges, to manage the alternate modes, only once an explicit contract is established. This layer is the high level of the USB C power stack and the place where the user defines the power strategy of its application.

Policy Engine (PE) layer
The Policy Engine (PE) role is to drive the message sequences according to the sent message and to its expected response. It allows negotiating power, establishing an Explicit Contract for the power exchange. The acceptance or the refusal of a request depends on the response of the DPM towards a specific power profile. The PE also handles the Vendor Defined Messages flow, allowing to discover, enter or
exit specified modes according to those supported by both provider and consumer sides.

- **Protocol layer high (PRL layer high)**
  This layer’s role is to construct, transmit, and receive messages from/to TCPC layers to/from the PE layer.
  It interacts with the TCPM layer to transmit messages to TCPC through the TCPCi interface.

- **USB Type-C™ protocol management (TCPM)**
  This layer must handle messages exchanged between PRL and TCPC layers. It means the management of alerts raised by TCPC, wrap communications, and others.

- **USB Type-C™ protocol controller (TCPC) component**
  This layer is used to define all the registers accesses linked to TCPC component used for USB Type-C™.

- **USB Type-C™ protocol controller interface (TCPCi)**
  This layer manages all the I²C transactions between TCPM and TCPC and also interruptions linked to alerts.

- **TCPC hardware**
  This layer is responsible for sending and receiving messages across the CC wires.
  The TCPC hardware implements the entire USB-PD PHY layer with BMC encoding.
  The TCPC implements a portion of the protocol layer transmission state diagram.
  - CRCReceiveTimer
  - RetryCounter
  - Messages that are received are passed to the TCPM via I²C
  - BIST handling

**Note:** For more information about the USB Power Delivery protocol, refer to the official specification document mentioned in Section 1.2.
3 USB-PD ST application setup

The USB-C™ Power Delivery library is provided in binary format, comes on top of the STM32Cube HAL driver, and offers all the APIs required to develop a USB-PD application.

This section describes the USB-PD library middleware integration and illustrates how users can develop their own power-delivery applications using this library.

The USB-PD library is developed following the Universal Serial Bus Power Delivery Specification and Universal Serial Bus Type-C™ Cable and Connector Specification, refer to Section 1.2, and this library is officially certified.

The STM32 USB-PD package contains:
- The USB-PD core stack and the device drivers
- An example with USB-PD DRP (TCPM/TCPC architecture) with ON Semiconductor FUSB307 evaluation board

More details on the stack library can be found in the user manual regarding the management of USB Power Delivery systems with STM32 MCUs and the wiki. For both links, refer to Section 1.2.
4 Examples description

4.1 USB-PD DRP (TCPM/TCPC architecture)

This project implements a USB-PD DRP application, suitable for the ON Semiconductor FUSB307 evaluation board, based on the use of USB-PD libraries delivered in the X-CUBE-USB-PD Expansion Package, and highlighting the TCPM/TCPC architecture.

4.1.1 Example setup

The USB-PD TCPM/TCPC designed for FUSB307 evaluation board application can be found under Projects\STM32F072RB-Nucleo\Applications\USB_PD\EVAL_FUSB307_DRP

This application provides an example of USB Power Delivery implementation based on TCPM/TCPC architecture. The application and USB-PD core stack are located on the TCPM side and are running on the STM32F0xx Nucleo device on the ON Semiconductor FUSB307 board. The application is driving the TCPC controller through the I²C link.

4.1.2 Application description

When the application starts, USB-PD has the capability of operating as either a Source or Sink.
When connecting to a USB-PD device (source or sink), the application must detect the type of connected device, and adopt a corresponding suitable role, in order to trigger the power negotiation:

- User must plug the USB-C™ cable on the dedicated connector.
- When the STM32 MCU side behaves as a Consumer (Sink mode), i.e. when connected to a Source device, it waits for Power Capabilities message from the attached provider. When a Source Capabilities message is received, the STM32 starts the evaluation of the received capabilities and check if one of the received power objects can meet its power requirement. The STM32 sends the Request message to request the new power level from the offered Source Capabilities. Once the PS_RDY message is received, the Explicit Contract is established.
- When the STM32 MCU side behaves as a Provider (Source mode), i.e. when connected to a Sink device, it exchanges Power profiles with the connected device and waits for Power Request message from the attached consumer. If the requested power can be met, the STM32 MCU sends the Accept message followed by the PS_RDY message. Explicit Contract is then considered as established.

4.1.3 Library initialization

Refer to section 3.3 Application initialization of the user manual *Managing USB power delivery systems with STM32 microcontrollers* (UM2552).

4.1.4 TCPM API description

For a complete description of the USB-PD core stack callbacks to be implemented on the DPM user side, refer to the CHM file provided in the *Documentation* directory of the X-CUBE-USB-PD Expansion Package *(STM32F072xB_USBPD_CORE_RELEASE_User_Manual.chm)*, in particular to *USBPD_PE_Callbacks Struct* reference description. For a complete description of the USBPD core TCPM stack callbacks to be implemented on the TCPC component side, refer to the CHM file provided in the *Documentation* directory of the X-CUBE-USB-PD Expansion Package *(STM32F072xB_USBPD_CORE_TCPM_RELEASE user manual)*, in particular to *TCPC_DrvTypeDef Reference description*
5 Memory footprint of the EVAL_FUSB307_DRP example

The values in Table 2 are calculated according to the following configuration:

- Compiler: IAR™ Embedded Workbench® for Arm®, Version 8.32.3
- Optimization: high size
- MCU: STM32F072RB
- ON-FUSB3-STM32 board

<table>
<thead>
<tr>
<th>Software</th>
<th>Flash memory</th>
<th>RAM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI</td>
<td>7904</td>
<td>700</td>
<td>Responder for STM32CubeMonitor-UCDPD (STM32CubeMonUCPD) graphical user interface</td>
</tr>
<tr>
<td>Application</td>
<td>5286</td>
<td>352</td>
<td>Application</td>
</tr>
<tr>
<td>USB-PD library</td>
<td>25592</td>
<td>72</td>
<td>PD3 Config1 stack</td>
</tr>
<tr>
<td>DRV</td>
<td>7334</td>
<td>364</td>
<td>TCPCi drivers for FUSB305</td>
</tr>
<tr>
<td>HAL</td>
<td>3912</td>
<td>40</td>
<td>STM32 HAL/LL drivers (Like Flash, I²C, USART for debugging)</td>
</tr>
<tr>
<td>Debug</td>
<td>1550</td>
<td>1044</td>
<td>For Tracer debugging (STM32CubeMonitor-UCPD)</td>
</tr>
<tr>
<td>FreeRTOS™</td>
<td>4332</td>
<td>7028</td>
<td>FreeRTOS™ allocation and services</td>
</tr>
<tr>
<td>sideLib</td>
<td>2654</td>
<td>2904</td>
<td>IAR™ libraries</td>
</tr>
<tr>
<td>Device</td>
<td>960</td>
<td>76</td>
<td>For usbpd_pwr_if.c</td>
</tr>
<tr>
<td>Total</td>
<td>59554</td>
<td>12580</td>
<td>-</td>
</tr>
</tbody>
</table>
6 Frequently asked questions (FAQs)

How can I get the STM32 USB-PD library?

The library is provided for free download in a binary format, from www.st.com. The general application note USB Type-C™ Power Delivery using STM32xx Series MCUs and STM32xxx Series MPUs (AN5225) describes how to do power delivery with the STM32. This firmware package for X-CUBE-USB-PD is located also on the github server: https://github.com/STMicroelectronics/x-cube-usb-pd.

I want to use only the USB-C™ feature (cable detachment attachment and cable orientation). Is this possible?

Yes, this is possible since the CAD (Cable attachment and detachment) module and the PD communication are driven by two separate processes. You can call only the CAD process to ensure cable detection.

Does the X-CUBE-USB-PD Expansion Package work on platforms different from STM32F0?

The core stack is device-independent. The STM32F0 and STM32F4 are supported on the device part in this delivery.

More information can be found in https://wiki.st.com/stm32mcu/wiki/USB_Power_Delivery_overview
7 Revision history

Table 3. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>08-Jun-2016</td>
<td>1</td>
<td>Initial release.</td>
</tr>
<tr>
<td>23-Jan-2017</td>
<td>2</td>
<td>Updated Section 1.2: References, Section 3.1: Overview, Section 3.2: Features, Section 3.3: Library architecture, Section 3.4: Hardware related components (for P-NUCLEO-USB001), Section 4: USB-PD library programming guidelines, Section 4.2: Library initialization, Section 4.3: USB-PD core stack library functions, Section 4.4: USB-PD core stack, Section 6.1: Hardware description, Section 6.2: USB-PD provider, Section 6.5: USB-PD consumer, Section 6.8: USB-PD consumer DRP and Section 8: Frequently asked questions (FAQs). Added Section 6.3: USB-PD provider (with CLI support), Section 6.4: USB-PD provider (with VDM support), Section 6.6: USB-PD consumer (with CLI support), Section 6.7: USB-PD consumer (with VDM support) and Section 6.9: USB-PD Dual Port. Updated Table 1: List of acronyms, Table 2: Use of different IPs, Table 3: GPIOs used by Port0, Table 4: GPIOs used by Port1, Table 6: DPM files, Table 7: USB-PD user functions, Table 8: USB-C™ PD callbacks, Table 9: USB-PD - Provider memory footprint (in Bytes), Table 10: USB-PD - Consumer memory footprint and Table 11: USB-PD - Dual role port memory footprint. Updated Figure 6: Project files.</td>
</tr>
<tr>
<td>09-May-2018</td>
<td>3</td>
<td>Updated Introduction, Section 1.2: References, Section 2.1: Architecture overview, Section 3.1: Overview, Section 3.2: Features, Section 3.3: Library architecture, Section 3.4: Hardware related components (for P-NUCLEO-USB001), Section 4: USB-PD library programming guidelines, Section 4.2: Library initialization, Section 4.4: USB-PD core stack, Section 6.5.1: Example setup, Section 6.9.1: Example setup, Section 7: Memory footprint and Section 8: Frequently asked questions (FAQs). Added Section 4.5: USB-PD device components. Updated Figure 1: USB power delivery architecture, Figure 6: Project files and Figure 7: USB-PD stack architecture (for P-NUCLEO-USB001). Updated Table 6: USB-PD - Provider memory footprint (in Bytes), Table 7: USB-PD - Consumer memory footprint (in Bytes) and Table 8: USB-PD - Dual role port memory footprint (in Bytes). Removed former Table 6: DPM files, Table 7: USB-PD user functions and Table 8: USB-C™ PD callbacks.</td>
</tr>
<tr>
<td>30-May-2018</td>
<td>4</td>
<td>Updated Introduction. Added Chapter 2.2: Specific case of TCPM/TCPC architecture, Chapter 4.1: Description of available configurations, Chapter 5: Atomic message sequencing and Chapter 6.10: USB-PD DRP (TCPM/TCPC architecture)</td>
</tr>
</tbody>
</table>
### Table 3. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
</table>
| 10-Apr-2020   | 5        | Updated *Introduction*  
Former section 3 moved to *Section 4.1.4: TCPM API description*  
Removed P-NUCLEO-USB001 and P-NUCLEO-USB002. The first is obsolete, and the second is downloadable from another page. |
| 16-Jun-2020   | 6        | Updated:  
– *Figure 4: Project structure* simplified,  
– *Section 4.1.3: Library initialization* referring to the corresponding UM2552 section. |
IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2020 STMicroelectronics – All rights reserved