31/10/2023 19:27:04

Compare Results

Old File:		New File:
USB_PD_R3_1 V1.8 2023-04_Ch9.pdf	versus	**USB_PD_R3_2 V1.0 2023-10_Ch 9.pdf**
15 pages (1.01 MB)		**18 pages (883 KB)**
13/10/2023 19:37:35		31/10/2023 18:10:30

Total Changes

203

Text only comparison

Content

121 Replacements

38 Insertions

44 Deletions

Styling and Annotations

0 Styling

0 Annotations

Go to First Change (page 1)

9. States and Status Reporting

9.1 Overview

This chapter describes the Status reporting mechanisms for devices with data connections (e.g., D+/D- and or SSTx+/- and SSRx+/-). It also describes the corresponding USB state a device that supports USB PD ***Shall*** transition to as a result of changes to the USB PD state that the device is in.

This chapter does not define the System Policy or the System Policy Manager. That is defined in ***[UCSI]***. In addition, the Policies themselves are not described here; these are left to the implementers of the relevant products and systems to define.

All PD Capable USB (PDUSB) Devices ***Shall*** report themselves as self-powered devices (over USB) when plugged into a PD capable Port even if they are entirely powered from V_{BUS}. However, there are some differences between PD and ***[USB 2.0]*** / ***[USB 3.2]***; for example, the presence of V_{BUS} alone does not mean that the device (Consumer) moves from the USB Attached state to the USB Powered state. Similarly, the removal of V_{BUS} alone does not move the device (Consumer) from any of the USB states to the Attached state. See ***Section 9.1.2 "Mapping to USB Device States"*** for details.

PDUSB Devices ***Shall*** follow the PD requirements when it comes to suspend (see ***Section 6.4.1.2.2.2 "USB Suspend Supported"***), configured, and operational power. The PD requirements when the device is configured or operational are defined in this section (see ***Table 9.4 "PD Consumer Port Descriptor"***). Note that the power requirements reported in the PD Consumer Port descriptor of the device ***Shall*** override the power draw reported in the *bMaxPower* field in the configuration descriptor. A PDUSB Device ***Shall*** report zero in the *bMaxPower* field after successfully negotiating a mutually agreeable Contract and ***Shall*** disconnect and re-enumerate when it switches operation back to operating in standard ***[USB 2.0]***, ***[USB 3.2]***, ***[USB4]***, ***[USB Type-C 2.3]*** (USB Type-C®) or ***[USBBC 1.2]***. When operating in ***[USB 2.0]***, ***[USB 3.2]***, , ***[USB Type-C 2.3]*** or ***[USBBC 1.2]*** mode it ***Shall*** report its power draw via the *bMaxPower* field.

Each Provider and Consumer will have their own Local Policies which operate between directly connected ports. An example of a typical PD system is shown in ***Figure 9-1 "Example PD Topology"***. This example consists of a Provider, Consumer/Providers and Consumers connected together in a tree topology. Between directly connected devices there is both a flow of Power and also Communication consisting of both Status and Control information.

Figure 9-1 "Example PD Topology"

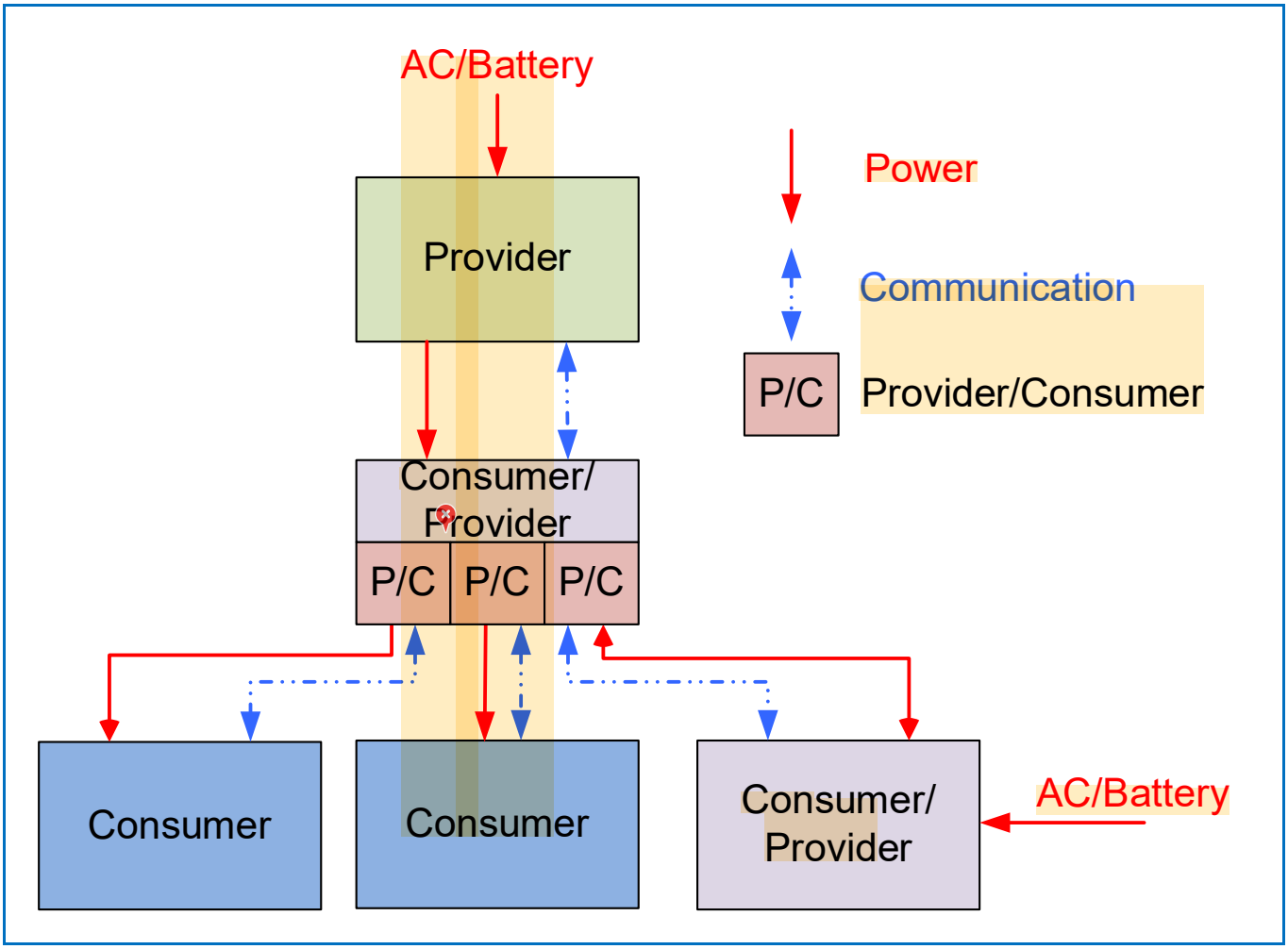
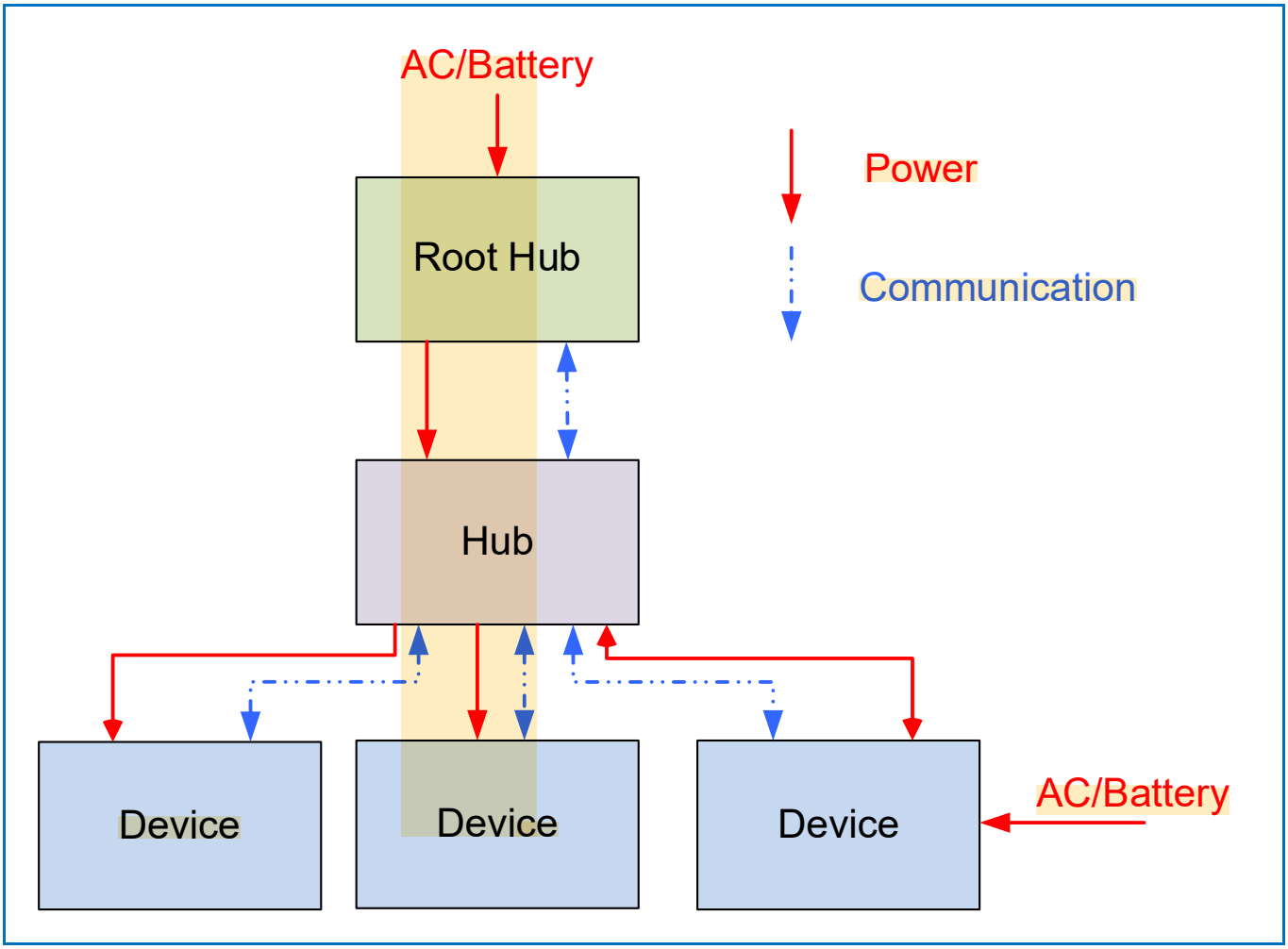


Figure 9-2 "Mapping of PD Topology to USB" shows how this same topology can be mapped to USB. In a USB based system, policy is managed by the host and communication of system level policy information is via standard USB data line communication. This is a separate mechanism to the USB Power Delivery V_{BUS} protocol which is used to manage Local Policy. When USB data line communication is used, status information and control requests are passed directly between the System Policy Manager (SPM) on the host and the Provider or Consumer.

Status information comes from a Provider or Consumer to the SPM so it can better manage the resources on the host and provide feedback to the end user.

Real systems will be a mixture of devices which in terms of power management support might have implemented PD, *[USB 2.0]*, *[USB 3.2]*, *[USB4]*, *[USB Type-C 2.3]* or *[USBBC 1.2]* or they might even just be non-compliant Power Sucking Devices. The level of communication of system status to the SPM will therefore not necessarily be comprehensive. The aim of the status mechanisms described here is to provide a mechanism whereby each connected entity in the system provides as much information as possible on the status of itself.

Figure 9-2 "Mapping of PD Topology to USB"



Information described in this section that is communicated to the SPM is as follows:

- Versions of USB Type-C® Current, PD and BC supported.
- Capabilities as a Provider/Consumer.
- Current operational state of each Port e.g. Standard, USB Type-C® Current, BC, PD and negotiated power level.
- Status of AC or Battery Power for each PDUSB Device in the system.

The SPM can negotiate with Providers or Consumers in the system in order to request a different Local Policy, or to request the amount of power to be delivered by the Provider to the Consumer. Any change in Local Policy could trigger a renegotiation of the Contract, using USB Power Delivery protocols, between a directly connected Provider and Consumer. A change in how much power is to be delivered will, for example, cause a renegotiation.

9.1.1 PDUSB Device and Hub Requirements

All PDUSB Devices ***Shall*** return all relevant descriptors mentioned in this chapter. PDUSB Hubs ***Shall*** also support a PD bridge as defined in ***[UCSI]***.

9.1.2 Mapping to USB Device States

As mentioned in *Section 9.1 "Overview"* a PDUSB Device reports itself as a self-powered device. However, the device ***Shall*** determine whether or not it is in the USB Attached or USB Powered states as described in *Figure 9-3 "USB Attached to USB Powered State Transition"*, *Figure 9-4 "Any USB State to USB Attached State Transition (When operating as a Consumer)"* and *Figure 9-5 "Any USB State to USB Attached State Transition (When operating as a Provider)"*. All other USB states of the PDUSB Device ***Shall*** be as described in Chapter 9 of *[USB 2.0]* and *[USB 3.2]*.

Figure 9-3 "USB Attached to USB Powered State Transition" shows how a PDUSB Device determines when to transition from the USB Attached to the USB Powered state. USB Type-C® Dead Battery operation does not require special handling since the default state at Attach or after a Hard Reset is that the USB Device is a Sink.

Figure 9-3 "USB Attached to USB Powered State Transition"

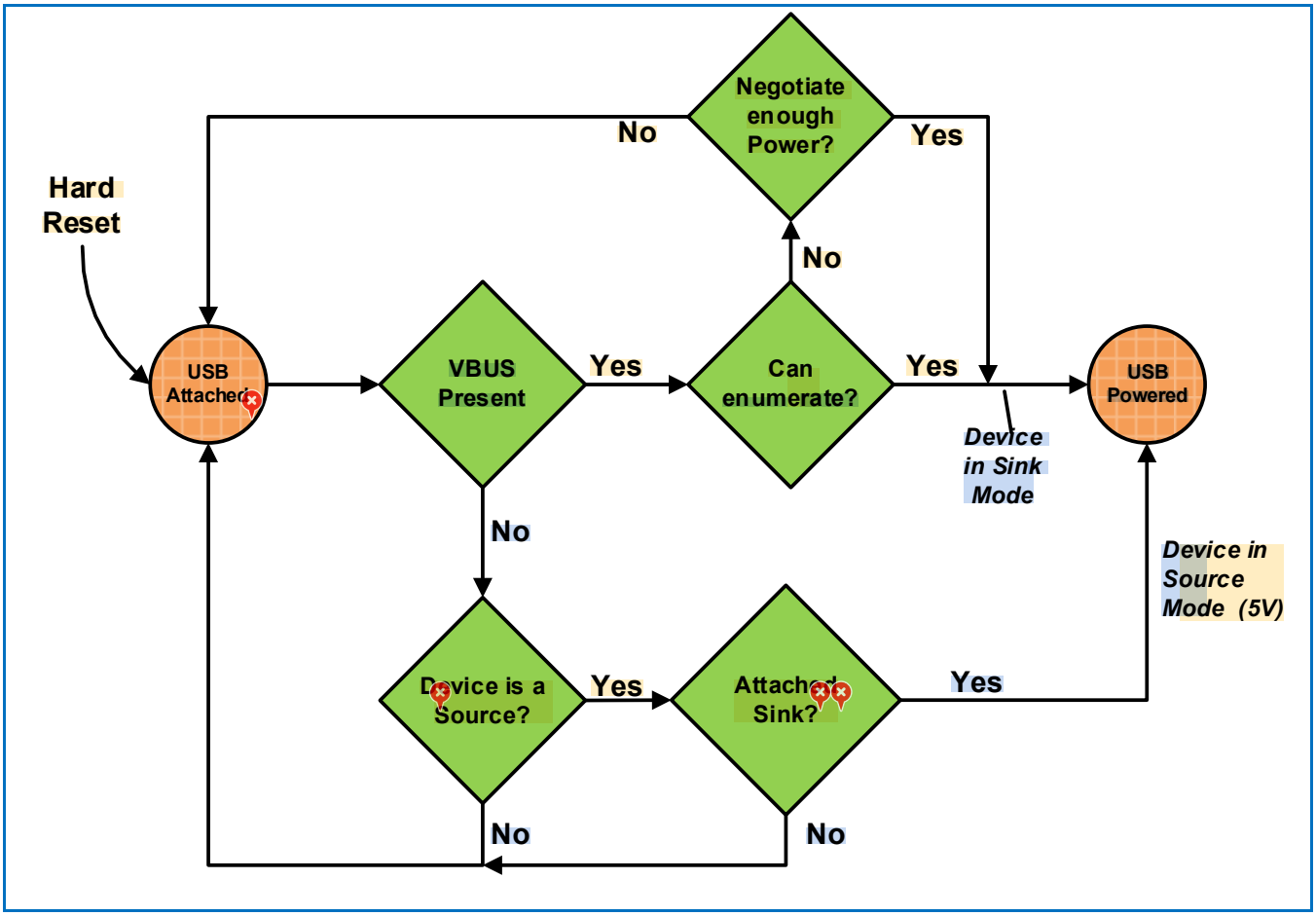


Figure 9-4 "Any USB State to USB Attached State Transition (When operating as a Consumer)" shows how a PDUSB Device determines when to transition from the USB Powered state to the USB Attached state when the device is a Consumer. A PDUSB Device determines that it is performing a Power Role Swap as described in *Section 8.3.3.20.3 "Policy Engine in Source to Sink Power Role Swap State Diagram"* and *Section 8.3.3.20.4 "Policy Engine in Sink to Source Power Role Swap State Diagram"*. See *Section 7.1.5 "Response to Hard Resets"* for additional information on device behavior during Hard Resets.

Figure 9-4 "Any USB State to USB Attached State Transition (When operating as a Consumer)"

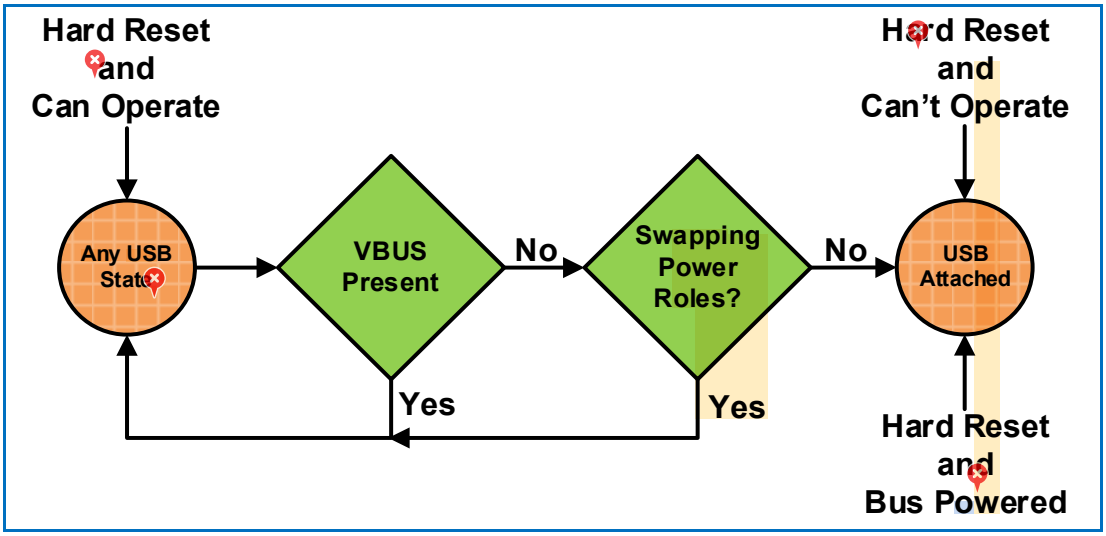


Figure 9-5 "Any USB State to USB Attached State Transition (When operating as a Provider)" shows how a PDUSB Device determines when to transition from the USB Powered state to the USB Attached state when the device is a Provider.

Figure 9-5 "Any USB State to USB Attached State Transition (When operating as a Provider)"

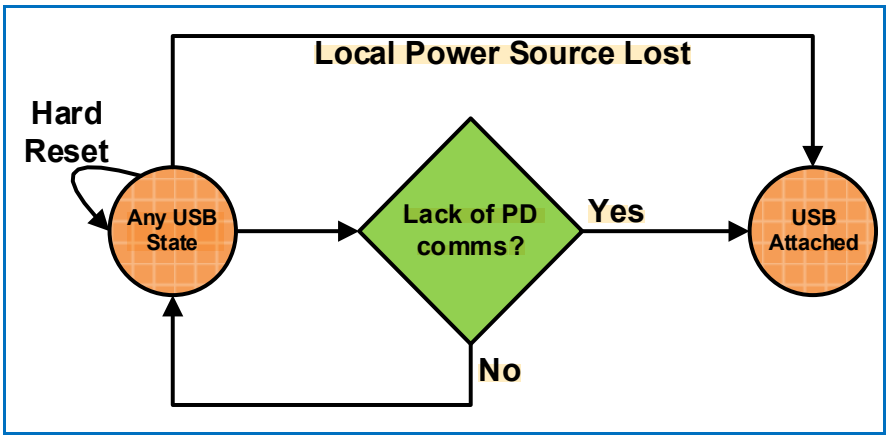
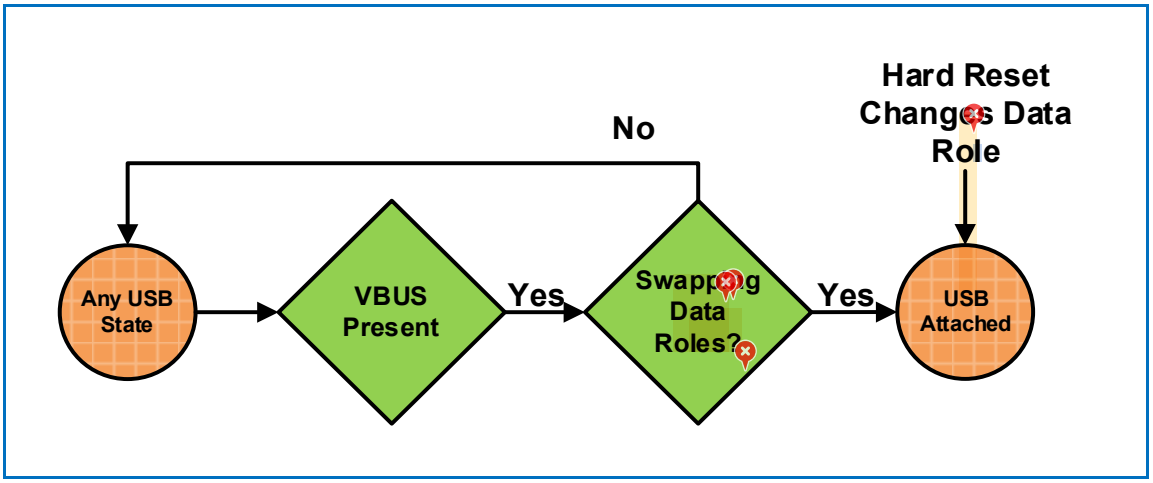


Figure 9-6 "Any USB State to USB Attached State Transition (After a USB Type-C® Data Role Swap)" shows how a PDUSB Device using the USB Type-C® connector determines when to transition from the USB Powered state to the USB Attached state after a Data Role Swap has been performed i.e., it has just changed from operation as a PDUSB Host to operation as a PDUSB Device. The Data Role Swap is described in *Section 6.3.9 "DR_Swap Message"*. A Hard Reset will also return a Sink acting as a PDUSB Host to PDUSB Device operation as described in *Section 6.8.3 "Hard Reset"*. See *Section 7.1.5 "Response to Hard Resets"* for additional information on device behavior during Hard Resets.

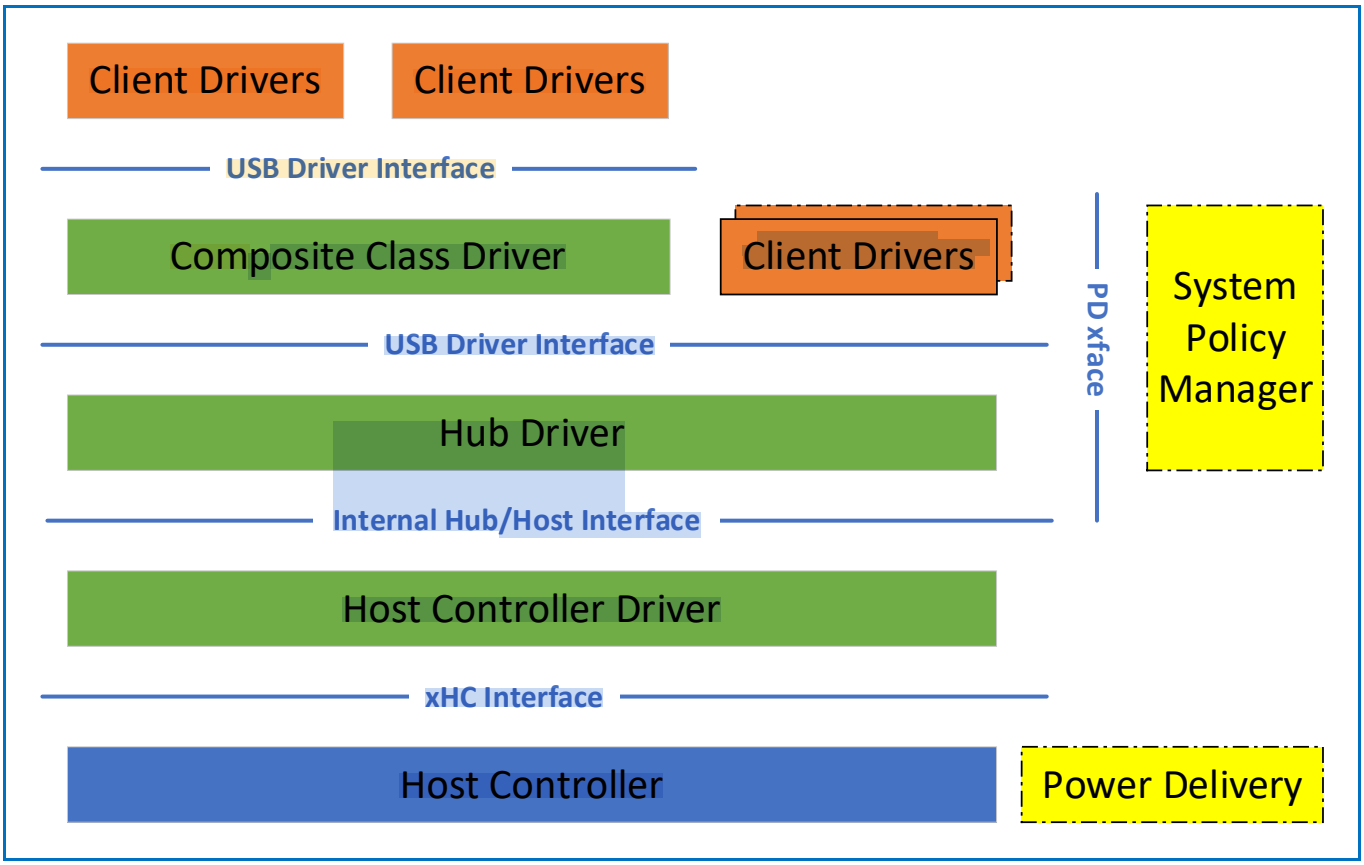
Figure 9-6 "Any USB State to USB Attached State Transition (After a USB Type-C® Data Role Swap)"



9.1.3 PD Software Stack

Figure 9-7 "Software stack on a PD aware OS" gives an example of the software stack on a PD aware OS. In this stack we are using the example of a system with an xHCI based controller. The USB Power Delivery hardware ***May*** or ***May Not*** be a part of the xHC.

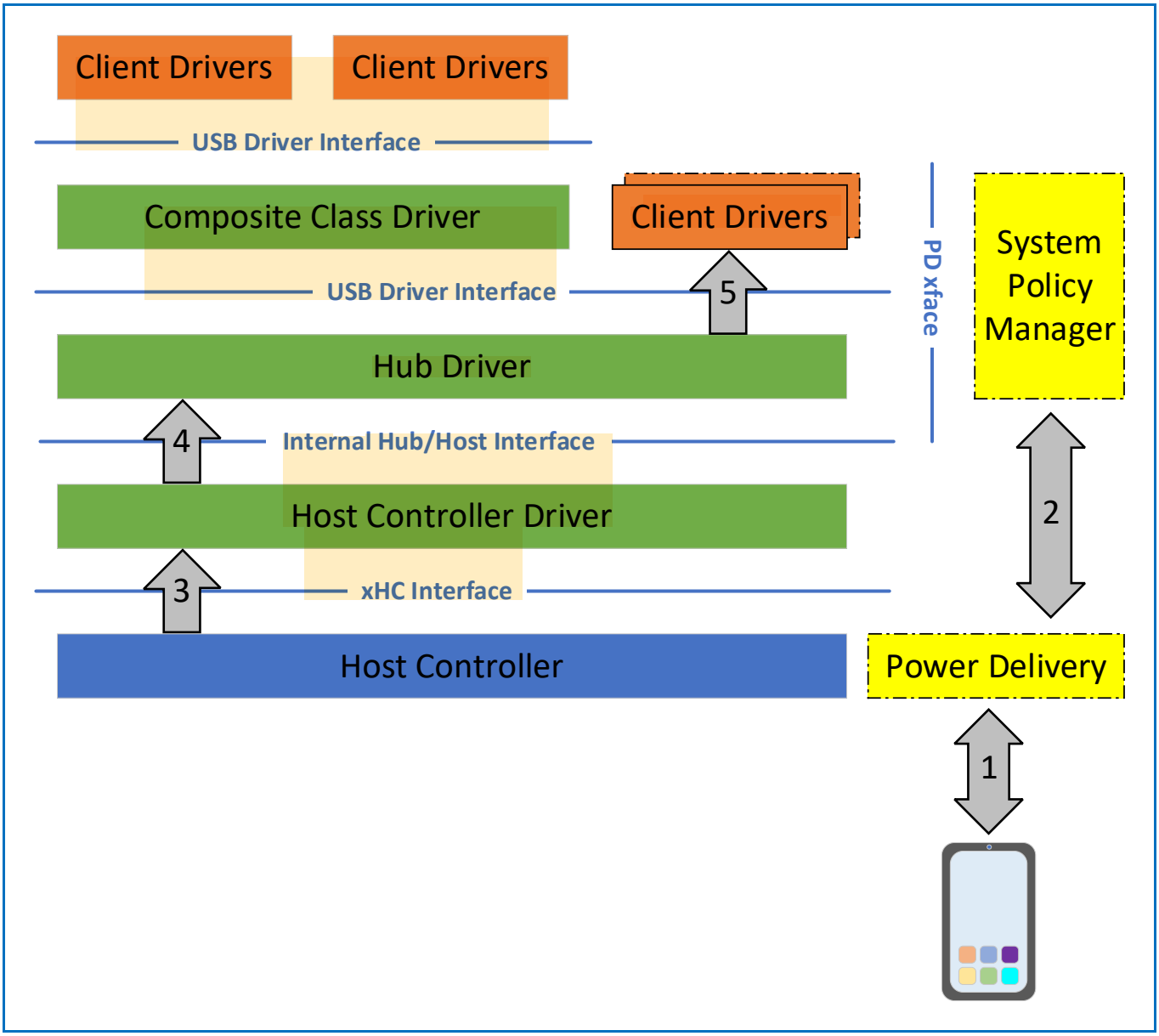
Figure 9-7 "Software stack on a PD aware OS"



9.1.4 PDUSB Device Enumeration

As described earlier, a PDUSB Device acts as a self-powered device with some caveats with respect to how it transitions from the USB Attached state to USB Powered state. *Figure 9-8 "Enumeration of a PDUSB Device"* gives a high-level overview of the enumeration steps involved due to this change. A PDUSB Device will first (Step1) interact with the Power Delivery hardware and the Local Policy manager to determine whether or not it can get sufficient power to enumerate/operate. Note PD is likely to have established a Contract prior to enumeration. The SPM will be notified (Step 2) of the result of this negotiation between the Power Delivery hardware and the PDUSB Device. After successfully negotiating a mutually agreeable Contract the device will signal a connect to the xHC. The standard USB enumeration process (Steps 3, 4 and 5) is then followed to load the appropriate driver for the function(s) that the PDUSB Device exposes.

Figure 9-8 "Enumeration of a PDUSB Device"



If a PDUSB Device cannot perform its intended function with the amount of power that it can get from the Port, it is connected to then the host system ***Should*** display a Message (on a PD aware OS) about the failure to provide sufficient power to the device. In addition, the device ***Shall*** follow the requirements listed in ***Section 8.2.5.2.1 "Local device handling of mismatch"***.

9.2 PD Specific Descriptors

A PDUSB Device ***Shall*** return all relevant descriptors mentioned in this section.

The device ***Shall*** return its capability descriptors as part of the device's Binary Object Store (BOS) descriptor set. *Table 9.1 "USB Power Delivery Type Codes"* lists the type of PD device capabilities.

Table 9.1 "USB Power Delivery Type Codes"

Capability Code	Value	Description
POWER_DELIVERY_CAPABILITY	06H	Defines the various PD Capabilities of this device
BATTERY_INFO_CAPABILITY	07H	Provides information on each Battery supported by the device
PD_CONSUMER_PORT_CAPABILITY	08H	The Consumer characteristics of a Port on the device
PD_PROVIDER_PORT_CAPABILITY	09H	The Provider characteristics of a Port on the device

9.2.1 USB Power Delivery Capability Descriptor

Table 9.2 USB Power Delivery Capability Descriptor details the fields in the USB Power Delivery Capability Descriptor.

Table 9.2 USB Power Delivery Capability Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of descriptor
1	bDescriptorType	1	Constant	DEVICE CAPABILITY Descriptor type
2	bDevCapabilityType	1	Constant	Capability type: *POWER_DELIVERY_CAPABILITY*
3	b***Reserved***	1	***Reserved***	***Shall*** be set to zero.

<table>
<tr><td>4</td><td>bmAttributes</td><td>4</td><td>Bitmap</td><td>Bitmap encoding of supported device level features. A value of one in a bit location indicates a feature is supported; a value of zero indicates it is not supported. Encodings are:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Reserved. Shall be set to zero.</td></tr>
<tr><td>1</td><td>Battery Charging. This bit Shall be set to one to indicate this device supports the Battery Charging Specification as per the value reported in the bcdBCVersion field.</td></tr>
<tr><td>2</td><td>USB Power Delivery. This bit Shall be set to one to indicate this device supports the USB Power Delivery Specification as per the value reported in the bcdPDVersion field.</td></tr>
<tr><td>3</td><td>Provider. This bit Shall be set to one to indicate this device is capable of providing power. This field is only Valid if Bit 2 is set to one.</td></tr>
<tr><td>4</td><td>Consumer. This bit Shall be set to one to indicate that this device is a consumer of power. This field is only Valid if Bit 2 is set to one.</td></tr>
<tr><td>5</td><td>This bit Shall be set to 1 to indicate that this device supports the feature CHARGING_POLICY. Note that supporting the CHARGING_POLICY feature does not require a BC or PD mechanism to be implemented.</td></tr>
<tr><td>6</td><td>USB Type-C® Current. This bit Shall be set to one to indicate this device supports power capabilities defined in the USB Type-C® Specification as per the value reported in the bcdUSBTypeCVersion field</td></tr>
<tr><td>7</td><td>Reserved. Shall be set to zero.</td></tr>
<tr><td>15:8</td><td>bmPowerSource. At least one of the following bits 8, 9 and 14 Shall be set to indicate which power sources are supported.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>8</td><td>AC Supply</td></tr>
<tr><td>9</td><td>Battery</td></tr>
<tr><td>10</td><td>Other</td></tr>
<tr><td>13:11</td><td>NumBatteries. This field Shall only be Valid when the Battery field is set to one and Shall be used to report the number of batteries in the device.</td></tr>
<tr><td>14</td><td>Uses V_{BUS}</td></tr>
<tr><td>15</td><td>Reserved and Shall be set to zero.</td></tr>
</table></td></tr>
<tr><td>31:16</td><td>Reserved. Shall be set to zero</td></tr>
</table></td></tr>
</table>

8	bcdBCVersion	2	BCD	Battery Charging Specification Release Number in Binary-Coded Decimal (e.g., V1.20 is 120H). This field ***Shall*** only be ***Valid*** if the device indicates that it supports BC in the *bmAttributes* field.
10	bcdPDVersion	2	BCD	USB Power Delivery Specification Release Number in Binary-Coded Decimal. This field ***Shall*** only be ***Valid*** if the device indicates that it supports PD in the *bmAttributes* field.
12	bcdUSBTypeCVersion	2	BCD	USB Type-C® Specification Release Number in Binary-Coded Decimal. This field ***Shall*** only be ***Valid*** if the device indicates that it supports USB Type-C® in the *bmAttributes* field.

9.2.2 Battery Info Capability Descriptor

A PDUSB Device ***Shall*** support the capability descriptor shown in ***Table 9.3 "Battery Info Capability Descriptor"*** if it reported that one of its power sources was a Battery in the *bmPowerSource* field in its Power Deliver Capability Descriptor. It ***Shall*** return one Battery Info Descriptor per Battery it supports.

Table 9.3 "Battery Info Capability Descriptor"

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of descriptor
1	bDescriptorType	1	Constant	DEVICE CAPABILITY Descriptor type
2	bDevCapabilityType	1	Constant	Capability type: ***BATTERY_INFO_CAPABILITY***
3	iBattery	1	Index	Index of string descriptor ***Shall*** contain the user-friendly name for this Battery.
4	iSerial	1	Index	Index of string descriptor ***Shall*** contain the Serial Number String for this Battery.
5	iManufacturer	1	Index	Index of string descriptor ***Shall*** contain the name of the Manufacturer for this Battery.
6	bBatteryId	1	Number	Value ***Shall*** be used to uniquely identify this Battery in status Messages.
7	bReserved	1	Number	***Reserved*** and ***Shall*** be set to zero.
8	dwChargedThreshold	4	mWh	***Shall*** contain the Battery Charge value above which this Battery is considered to be fully charged but not necessarily "topped off."
12	dwWeakThreshold	4	mWh	***Shall*** contain the minimum charge level of this Battery such that above this threshold, a device can be assured of being able to power up successfully (see Battery Charging 1.2).
16	dwBatteryDesignCapacity	4	mWh	***Shall*** contain the design capacity of the Battery.
20	dwBatteryLastFullchargeCapacity	4	mWh	***Shall*** contain the maximum capacity of the Battery when fully charged.

9.2.3 PD Consumer Port Capability Descriptor

A PDUSB Device ***Shall*** support the capability descriptor shown in *Table 9.4 "PD Consumer Port Descriptor"* if it is a Consumer.

Table 9.4 "PD Consumer Port Descriptor"

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of descriptor
1	bDescriptorType	1	Constant	DEVICE CAPABILITY Descriptor type
2	bDevCapabilityType	1	Constant	Capability type: *PD_CONSUMER_PORT_CAPABILITY*
3	bReserved	1	Number	***Reserved*** and ***Shall*** be set to zero.
4	bmCapabilities	2	Bitmap	Capability: This field ***Shall*** indicate the specification the Consumer Port will operate under. Bit 0: Battery Charging (BC) Bit 1: USB Power Delivery (PD) Bit 2: USB Type-C® Current Bit 15:3: ***Reserved*** and ***Shall*** be set to zero.
6	wMinVoltage	2	Number	***Shall*** contain the minimum Voltage in 50mV units that this Consumer is capable of operating at.
8	wMaxVoltage	2	Number	***Shall*** contain the maximum Voltage in 50mV units that this Consumer is capable of operating at.
10	wReserved	2	Number	***Reserved*** and ***Shall*** be set to zero.
12	dwMaxOperatingPower	4	Number	***Shall*** contain the maximum power in 10mW units this Consumer can draw when it is in a steady state operating mode.
16	dwMaxPeakPower	4	Number	***Shall*** contain the maximum power in 10mW units this Consumer can draw for a short duration of time *(dwMaxPeakPowerTime)* before it falls back into a steady state.
20	dwMaxPeakPowerTime	4	Number	***Shall*** contain the time in 100ms units that this Consumer can draw peak current. A device ***Shall*** set this field to 0xFFFF if this value is unknown.

9.2.4 PD Provider Port Capability Descriptor

A PDUSB Device ***Shall*** support the capability descriptor shown in *Table 9.5 "PD Provider Port Descriptor"* if it is a Provider.

Table 9.5 "PD Provider Port Descriptor"

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of descriptor
1	bDescriptorType	1	Constant	DEVICE CAPABILITY Descriptor type
2	bDevCapabilityType	1	Constant	Capability type: *PD_PROVIDER_PORT_CAPABILITY*
3	bReserved	1	Number	***Reserved*** and ***Shall*** be set to zero.
4	bmCapabilities	2	Bitmap	This field ***Shall*** indicate the specification the Provider Port will operation under. Bit 0: Battery Charging (BC) Bit 1: USB Power Delivery (PD) Bit 2: USB Type-C® Current Bit 15:3: ***Reserved***. ***Shall*** be set to zero.
6	bNumOfPDObjects	1	Number	***Shall*** indicate the number of Power Data Objects.
7	bReserved	1	Number	***Reserved*** and ***Shall*** be set to zero.
8	wPowerDataObject1	4	Bitmap	***Shall*** contain the first Power Data Object supported by this Provider Port. See *Section 6.4.1 "Capabilities Message"* for details of the Power Data Objects.
…	…	…	…	…
4*(N+1)	wPowerDataObjectN	4	Bitmap	***Shall*** contain the 2nd and subsequent Power Data Objects supported by this Provider Port. See *Section 6.4.1 "Capabilities Message"* for details of the Power Data Objects.

9.3 PD Specific Requests and Events

A PDUSB Device that is compliant to this specification ***Shall*** support the Battery related requests if it has a battery.

A PDUSB Hub that is compliant to this specification ***Shall*** support a USB PD Bridge as described in ***[UCSI]*** irrespective of whether the PDUSB Hub is a Provider, a Consumer, or both.

9.3.1 PD Specific Requests

PD defines requests to which PDUSB Devices ***Shall*** respond as outlined in ***Table 9.6 "PD Requests"***. All ***Valid*** requests in ***Table 9.6 "PD Requests"*** ***Shall*** be implemented by PDUSB Devices.

Table 9.6 "PD Requests"

Request	bmRequestType	bRequest	wValue	wIndex	wLength	Data
GetBatteryStatus	10000000B	*Get_Battery_Status*	Zero	Battery ID	Eight	Battery Status
SetPDFeature	00000000B	set_feature	Feature Selector	Feature Specific	Zero	None

Table 9.7 "PD Request Codes" gives the bRequest values for commands that are not listed in the hub/device framework chapters of ***[USB 2.0]***, ***[USB 3.2]***.

Table 9.7 "PD Request Codes"

bRequest	Value
GET_BATTERY_STATUS	21

Table 9.8 "PD Feature Selectors" gives the ***Valid*** feature selectors for the PD class. Refer to ***Section 9.4.2.1 "BATTERY_WAKE_MASK Feature Selector"***, and ***Section 9.4.2.2 "CHARGING_POLICY Feature Selector"*** for a description of the features.

Table 9.8 "PD Feature Selectors"

Feature Selector	Recipient	Value
BATTERY_WAKE_MASK	Device	40
CHARGING_POLICY	Device	54

9.4 PDUSB Hub and PDUSB Peripheral Device Requests

9.4.1 GetBatteryStatus

The request shown in ***Table 9.9 "Get Battery Status Request"*** returns the current status of the Battery in a PDUSB Hub/Peripheral, with Battery Status information as shown in ***Table 9.10 "Battery Status Structure"***.

Table 9.9 "Get Battery Status Request"

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000000B	***Get_Battery_Status***	Zero	Battery ID	Eight	Battery Status

The PDUSB Hub/Peripheral ***Shall*** return the Battery Status of the Battery identified by the value of *wIndex* field.

Every PDUSB Device that has a Battery ***Shall*** return its Battery Status when queried with this request. For Providers or Consumers with multiple batteries, the status of each Battery ***Shall*** be reported per Battery.

Table 9.10 "Battery Status Structure"

Offset	Field	Size	Value	Description
0	bBatteryAttributes	1	Number	***Shall*** indicate whether a Battery is installed and whether this is charging or discharging. **Value** / **Description** 0 – There is no Battery 1 – The Battery is charging 2 – The Battery is discharging 3 – The Battery is neither discharging nor charging 255-4 – ***Reserved*** and ***Shall Not*** be used
1	bBatterySOC	1	Number	***Shall*** indicate the Battery State of Charge given as percentage value from Battery Remaining Capacity.
2	bBatteryStatus	1	Number	If a Battery is present ***Shall*** indicate the present status of the Battery. **Value** / **Meaning** 0 – No error 1 – Battery required and not present 2 – Battery non-chargeable/wrong chemistry 3 – Over-temp shutdown 4 – Over-Voltage shutdown 5 – Over-current shutdown 6 – Fatigued Battery 7 – Unspecified error 255-8 – ***Reserved*** and ***Shall Not*** be used

<table>
<tr><td>3</td><td>bRemoteWakeCapStatus</td><td>1</td><td>Bitmap</td><td>If the device supports remote wake, then the device Shall support Battery Remote wake events. The default value for the Remote wake events Shall be turned off (set to zero) and can be enable/disabled by the host as required. If set to one the device Shall generate a wake event when a change of status occurs. See Section 9.4.2 “SetPDFeature” for more details.<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Battery present event</td></tr><tr><td>1</td><td>Charging flow</td></tr><tr><td>2</td><td>Battery error</td></tr><tr><td>7:3</td><td>Reserved and Shall be set to zero</td></tr></table></td></tr>
<tr><td>4</td><td>wRemainingOperatingTime</td><td>2</td><td>Number</td><td>Shall contain the operating time (in minutes) until the Weak Battery threshold is reached, based on Present Battery Strength and the device’s present operational power needs.
Note: this value Shall exclude any additional power received from charging.
A Battery that is not capable of returning this information Shall return a value of 0xFFFF.</td></tr>
<tr><td>6</td><td>wRemainingChargeTime</td><td>2</td><td>Number</td><td>Shall contain the remaining time (in minutes) until the Charged Battery threshold is reached based on Present Battery Strength, charging power and the device’s present operational power needs. Value Shall only be Valid if the Charging Flow is “Charging”.
A Battery that is not capable of returning this information Shall return a value of 0xFFFF.</td></tr>
</table>

If *wValue* or *wLength* are not as specified above, then the behavior of the PDUSB Device is not specified.

If *wIndex* refers to a Battery that does not exist, then the PDUSB Device ***Shall*** respond with a Request Error.

If the PDUSB Device is not configured, the PDUSB Hub's response to this request is undefined.

If the PDUSB Hub is not configured, the PDUSB Hub's response to this request is undefined.

9.4.2 SetPDFeature

The request shown in *Table 9.11 “Set PD Feature”* sets the value requested in the PDUSB Hub/Peripheral.

Table 9.11 “Set PD Feature”

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B	set_ feature	Feature Selector	Feature Specific	Zero	None

Setting a feature enables that feature or starts a process associated with that feature; see *Table 9.8 “PD Feature Selectors”* for the feature selector definitions. Features that ***May*** be set with this request are:

- *BATTERY_WAKE_MASK.*
- *CHARGING_POLICY.*

9.4.2.1 BATTERY_WAKE_MASK Feature Selector

When the feature selector is set to ***BATTERY_WAKE_MASK***, then the *wIndex* field is structured as shown in ***Table 9.12 "Battery Wake Mask"***.

Table 9.12 "Battery Wake Mask"

Bit	Description
0	**Battery Present**: When this bit is set then the PDUSB Device ***Shall*** generate a wake event if it detects that a Battery has been Attached.
1	**Charging Flow**: When this bit is set then the PDUSB Device ***Shall*** generate a wake event if it detects that a Battery switched from charging to discharging or vice versa.
2	**Battery Error**: When this bit is set then the PDUSB Device ***Shall*** generate a wake event if the Battery has detected an error condition.
15:3	***Reserved*** and ***Shall Not*** be used.

The SPM ***May*** Enable or Disable the wake events associated with one or more of the above events by using this feature.

If the PDUSB Hub is not configured, the PDUSB Hub's response to this request is undefined.

9.4.2.2 CHARGING_POLICY Feature Selector

When the feature selector is set to ***CHARGING_POLICY***, the wIndex field ***Shall*** be set to one of the values defined in ***Table 9.13 "Charging Policy Encoding"***. If the device is using USB Type-C® Current above the default value or is using PD then this feature setting has no effect and the rules for power levels specified in the ***[USB Type-C 2.3]*** or USB PD specifications ***Shall*** apply.

Table 9.13 "Charging Policy Encoding"

Value	Description
00H	The device ***Shall*** follow the default current limits as defined in the USB 2.0 or USB 3.1 specification, or as negotiated through other USB mechanisms such as BC. This is the default value.
01H	The Device ***May*** draw additional power during the unconfigured and suspend states for the purposes of charging. For charging the device itself, the device ***Shall*** limit its current draw to the higher of these two values: ICCHPF as defined in the USB 2.0 or USB 3.1 specification, regardless of its USB state. Current limit as negotiated through other USB mechanisms such as BC.
02H	The Device ***May*** draw additional power during the unconfigured and suspend states for the purposes of charging. For charging the device itself, the device ***Shall*** limit its current draw to the higher of these two values: ICCLPF as defined in the USB 2.0 or USB 3.1 specification, regardless of its USB state. Current limit as negotiated through other USB mechanisms such as BC.
03H	The device ***Shall Not*** consume any current for charging the device itself regardless of its USB state.
04H-FFFFH	***Reserved*** and ***Shall Not*** be used

This is a ***Valid*** command for the PDUSB Hub/Peripheral in the Address or Configured USB states. Further, it is only ***Valid*** if the device reports a USB PD capability descriptor in its BOS descriptor and Bit 5 of the bmAttributes in that descriptor is set to 1. The device will go back to the wIndex default value of 0 whenever it is reset.