

USB Power Delivery Tester

Communication Protocol Guide

Model PM110 & PM125

Note: This document is only applicable to the latest firmware release of the USB Power Delivery Tester Model.

Document Edition: 1.16
Date: 1 April 2025
Web site: www.passmark.com

Table of Contents

| | |
|--|-------------------------------------|
| Introduction..... | 3 |
| System Requirements..... | 3 |
| Linking to the FTDI DLL | 4 |
| Search in D2XX devices | 4 |
| Open a device..... | 5 |
| Send and Receive data | 5 |
| Communication Protocol | 5 |
| Communication Format | 5 |
| Commands | 6 |
| Command Overview | 6 |
| Get Version | 6 |
| Get Connection Status..... | 7 |
| Get Capabilities..... | 8 |
| Get Statistics | 9 |
| Change Profile / Voltage..... | 9 |
| Set Current | 10 |
| Set Current (Fast)..... | 10 |
| Step Load Test | 11 |
| Set USB Internal Connection (PM125 only) | 12 |
| Get Hardware Sub-Revision (PM125 only)..... | 12 |
| Change Config (Persistent) | 13 |
| Change Config (Volatile)..... | 13 |
| Get Config..... | 14 |
| Set Backlight..... | 14 |
| Configuration Guide | 15 |
| Configuration Overview | 15 |
| Configuration Commands..... | Error! Bookmark not defined. |

Introduction

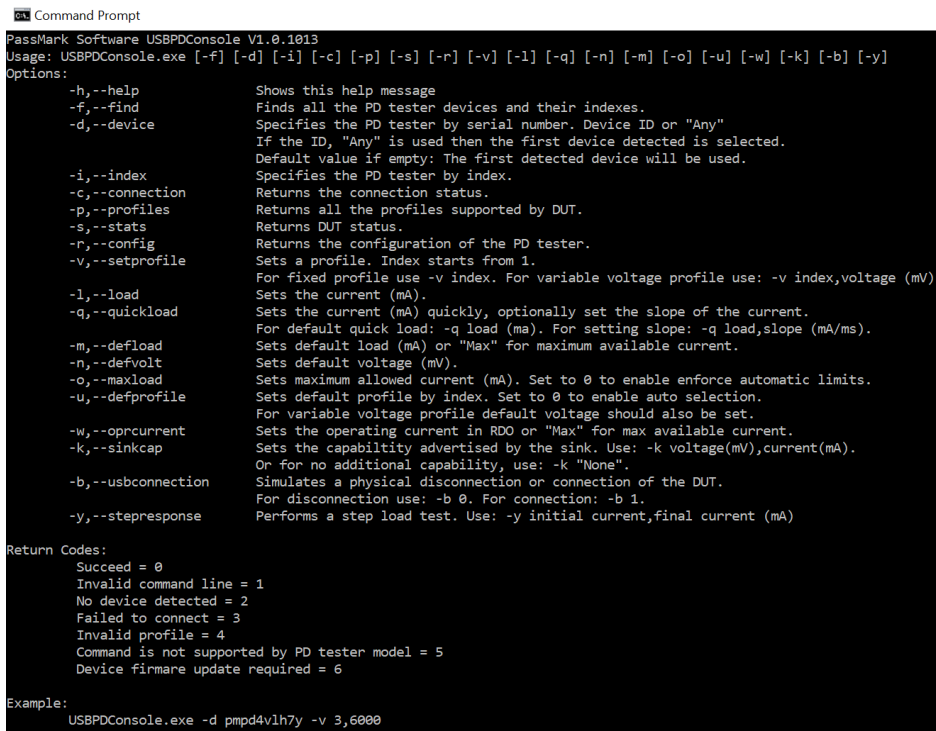
This document will demonstrate how to communicate with Passmark USB Power Delivery testers while using the [FTDI D2XX Interface](#). You may need to update device firmware as the information in this document only applies to firmware version 2.5 and later.

Details about the Passmark USB Power Delivery tester can be found here, <https://www.passmark.com.au/products/usb-power-delivery-tester.htm>

High level languages can access to the Passmark USB Power Delivery tester using the FTDI's proprietary "direct" driver interface. This interface, D2XX, is provided via a proprietary DLL (FTD2XX.DLL) and a static library (ftd2xx.lib). Please refer to the "D2XX Programmer's Guide" for detailed explanation of the functions available in the library.

<http://www.ftdichip.com/Support/Documents/ProgramGuides.htm>

The executable file for the example project is located under "Release" directory. To find a short description of all command line options run the executable with -h option. The below screenshot is taken from the USBPDConsole application version V1.0.1013.



```
Command Prompt
PassMark Software USBPDConsole V1.0.1013
Usage: USBPDConsole.exe [-f] [-d] [-i] [-c] [-p] [-s] [-r] [-v] [-l] [-q] [-n] [-m] [-o] [-u] [-w] [-k] [-b] [-y]
Options:
-h,--help                Shows this help message
-f,--find                Finds all the PD tester devices and their indexes.
-d,--device              Specifies the PD tester by serial number. Device ID or "Any"
                        If the ID, "Any" is used then the first device detected is selected.
                        Default value if empty: The first detected device will be used.
-i,--index               Specifies the PD tester by index.
-c,--connection          Returns the connection status.
-p,--profiles            Returns all the profiles supported by DUT.
-s,--stats               Returns DUT status.
-r,--config              Returns the configuration of the PD tester.
-v,--setprofile           Sets a profile. Index starts from 1.
                        For fixed profile use -v index. For variable voltage profile use: -v index,voltage (mV)
-l,--load                Sets the current (mA).
-q,--quickload            Sets the current (mA) quickly, optionally set the slope of the current.
                        For default quick load: -q load (ma). For setting slope: -q load,slope (mA/ms).
-m,--defload             Sets default load (mA) or "Max" for maximum available current.
-n,--defvoltage           Sets default voltage (mV).
-o,--maxload              Sets maximum allowed current (mA). Set to 0 to enable enforce automatic limits.
-u,--defprofile           Sets default profile by index. Set to 0 to enable auto selection.
                        For variable voltage profile default voltage should also be set.
-w,--oprcurrent           Sets the operating current in RDO or "Max" for max available current.
-k,--sinkcap              Sets the capability advertised by the sink. Use: -k voltage(mV),current(mA).
                        Or for no additional capability, use: -k "None".
-b,--usbconnection        Simulates a physical disconnection or connection of the DUT.
                        For disconnection use: -b 0. For connection: -b 1.
-y,--stepresponse         Performs a step load test. Use: -y initial current,final current (mA)

Return Codes:
Succeed = 0
Invalid command line = 1
No device detected = 2
Failed to connect = 3
Invalid profile = 4
Command is not supported by PD tester model = 5
Device firmware update required = 6

Example:
USBPDConsole.exe -d pmpd4v1h7y -v 3,6000
```

System Requirements

To compile the example, you will need;

Microsoft Visual Studio 2019

FTDI D2XX driver (2.12.28 or later) which can be downloaded from:

<http://www.ftdichip.com/Drivers/D2XX.htm>

Linking to the FTDI DLL

When using Visual Studio “VC++ Directories” need to be edited to point the location where the FTDI library is copied. The FTDI library is inside the driver zip file.

Include Directories

Need to add “Driver folder”

Library Directories

Need to add “Driver folder\Static\i386”

Or

for 64 bit projects “Driver folder\Static\amd64”

In the **Linker** settings for the project you will need to add “ftd2xx.lib” to the **Additional Dependencies**

In the project’s Property Pages, expand Configuration Properties > C/C++ > Preprocessor, add FTD2XX_STATIC to the Preprocessor Definitions

Search in D2XX devices

D2XX driver allows you to get a list of all FTDI devices connected. Below is an example code for searching in D2XX devices to find the Passmark USB Power Delivery testers:

```
DWORD    devcount = 0;
UCHAR    serial[50];
FT_STATUS ftStatus;

ftStatus = FT_ListDevices(&devcount, NULL, FT_LIST_NUMBER_ONLY);
for (DWORD curDevice = 0; curDevice < devcount; curDevice++)
{
    ftStatus = FT_ListDevices((PVOID)curDevice, serial, FT_LIST_BY_INDEX
| FT_OPEN_BY_SERIAL_NUMBER);
    if(ftStatus == FT_OK)
    {
        if(!strncmp((char*)serial, “PMPD” , 4))
        {
            // Add this serial number to the list of testers.
        }
    }
}
```

Open a device

The below code opens a device by its serial number and returns a handle that will be used for subsequent accesses.

```
FT_HANDLE devHandle;
FT_STATUS ftStatus;

ftStatus = FT_OpenEx(serial, FT_OPEN_BY_SERIAL_NUMBER, &devHandle);
if(ftStatus == FT_OK)
{
    FT_SetBaudRate(devHandle, 115200);
    FT_SetDataCharacteristics(devHandle, FT_BITS_8, FT_STOP_BITS_1,
    FT_PARITY_NONE);
    FT_SetTimeouts(devHandle, 1, 100);
}
```

Send and Receive data

```
FT_STATUS ftStatus;
ftStatus = FT_Write(devHandle, buf, len, &NumBytesWritten);
ftStatus = FT_Read(devHandle, buf, 1, &NumRecvBytes);
```

Communication Protocol

Communication Format

Following is the frame structure for the commands sent by the host and response received from the tester.

| | | | | | |
|--------|--------|---------|------|----------|--------------|
| Header | Length | Command | Data | Checksum | Closing Flag |
|--------|--------|---------|------|----------|--------------|

Opening Flag: This is a single byte that indicates the beginning of the frame.

Length: This byte indicates the total number of bytes between Length and Checksum.

Command: This byte is used to instruct the tester which operation to perform. For the response packets, this indicates which command's response.

Data [Payload]: Data bytes are the parameters of a command or response. The least significant byte is always sent and received first (LSB First).

Checksum: This byte is used on the host as well as the tester to check the validity of the packet and to trap any data corruption. This is calculated by XORing all the bytes except the Checksum byte.

Closing Flag: This is a single byte that indicates the end of the frame.

Commands

Command Overview

Table below summarizes the command list of USB Power Delivery Tester. Note that some config parameters can only be set on the PM125 model (See **Configuration Overview**).

| Command | Description |
|---------|--|
| 0x01 | Get Version |
| 0x0A | Get Connection Status |
| 0x0B | Get Capabilities |
| 0x0C | Get Statistics |
| 0x0D | Change Profile / Voltage |
| 0x10 | Set Current |
| 0x11 | Set Current (Fast) |
| 0x13 | Step Load Test |
| 0x14 | Set USB Internal Connection (PM125 only) |
| 0xD1 | Get Hardware Sub-Revision (PM125 only) |
| 0xE0 | Change Config (Persistent) |
| 0xE6 | Change Config (Volatile) |
| 0xE1 | Get Config |
| 0xE2 | Set Backlight |
| 0xE7 | Set Protocol Analyzer |

Get Version

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
| 0x02 | 0x01 | 0x01 | - | 0x01 | 0x03 |

Response:

| Header | Length (bytes) | |
|------------------|----------------|------------------|
| Length | 1 | 0x03 |
| Command | 1 | 0x01 |
| Hardware Version | 1 | Hardware version |
| Firmware Version | 1 | Firmware version |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example:

> Request: 02 01 01 01 03

> Response: 02 03 01 0a 19 10 03 (Hardware Ver 1.0, Firmware Ver: 2.5)

Get Connection Status

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
| 0x02 | 0x01 | 0x0A | - | 0x0A | 0x03 |

Response:

| | | | | | | | | |
|-------------------|----------------|---|------------------|------------|-------------|------------|----------|------------|
| Header | Length (bytes) | 0x02 | | | | | | |
| Length | 1 | 0x0D | | | | | | |
| Command | 1 | 0x0A | | | | | | |
| Connection Status | 1 | 0x00: Not Connected 0x01: Connected | | | | | | |
| Profile Index | 1 | Profile Index (index starts from 0) | | | | | | |
| Profile Type | 1 | 0x00: Legacy 0x01: Proprietary Charger 0x02: Battery Charging 0x03: Quick Charge (only available on hardware rev 2) 0x04: Type-C (without PD support) 0x05: Type-C (with PD support) | | | | | | |
| Profile Sub Type | 1 | | | | | | | |
| | | | Profile Sub Type | | | | | |
| | | Profile Type | 0x00 | 0x01 | 0x02 | 0x03 | 0x04 | 0x05 |
| | | 0x00 | | | | | | |
| | | 0x01 | Apple 0.5A | Apple 1A | Apple 2.1A | Apple 2.4A | Apple 3A | Samsung 2A |
| | | 0x02 | SDP | CDP | DCP | | | |
| | | 0x03 | QC1 | QC2 | QC3 | | | |
| | | 0x04 | USBC 1.5A | USBC 3A | | | | |
| | | 0x05 | PD Fix | PD Battery | PD Variable | PD APDO | | |
| Requested Voltage | 2 | Requested Voltage in millivolts | | | | | | |
| Max Current | 2 | Maximum current for the selected voltage | | | | | | |
| Max Power | 4 | Maximum power for the selected voltage | | | | | | |
| Checksum | 1 | To be calculated | | | | | | |
| Flag | 1 | 0x03 | | | | | | |

Example:

> Request: 02 01 0a 0a 03

> Response: 02 0d 0a 01 00 05 00 88 13 60 09 e0 2e 00 00 3e 03

Port Type = Type-C (with PD support), Profile Index = 0, Profile Type = Type-C (with PD support), Profile Sub Type = PD Fix, Requested Voltage = 5000 millivolts, Max Current = 2400mA, Max Power = 12000 milliwatts

Get Capabilities

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
| 0x02 | 0x01 | 0x0B | - | 0x0B | 0x03 |

Note: If there are more than 7 Power Data Objects, then a second message with the same structure will be sent shortly after the first, containing the remaining profiles.

Response:

| | | |
|--------------------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x3B |
| Command | 1 | 0x0B |
| Num Power Profiles | 2 | Number of power profiles |
| Power Data Objects | 56 | Profile Index (4 bits) Profile Type (4 bits) Profile Sub Type (3 bits) PDO Index (4 bits) Selectable (1 bit) Min Voltage in millivolts (2 bytes) Max Voltage in millivolts (2 bytes) Max Current in milliamperes (2 bytes) . Next Profile Data Object The values for Profile Type and Profile Sub Type fields are explained in Get Connection Status command |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

For SPR AVS profiles the structure is instead:

- Min Voltage \rightarrow Current limit @ 15V – 20V
- Max Current \rightarrow Current limit @ 9V – 15V

Example:

> Request: 02 01 0b 0b 03

[illegible]

Num Power Profiles = 4, Profile Index = 0, Profile Type = 4 (Type-C without PD support), Profile Sub Type = 0 (USB-C 1.5A), Min Voltage = 5000mV, Max Voltage

= 5000mV, Max Current = 3000mA, Profile Index = 1, Profile Type = 3 (QC), Profile Sub Type = 2 (QC3), Min Voltage = 3600mV, Max Voltage = 5000mV, Max Current = 1500mA, Profile Index = 2, Profile Type = 3 (QC), Profile Sub Type = 2 (QC3), Min Voltage = 3600mV, Max Voltage = 9000mV, Max Current = 1500mA, Profile Index = 3, Profile Type = 3 (QC), Profile Sub Type = 2 (QC3), Min Voltage = 3600mV, Max Voltage = 12000mV, Max Current = 1500mA,

Get Statistics

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
| 0x02 | 0x01 | 0x0C | - | 0x0C | 0x03 |

Response:

| Header | Length (bytes) | Data |
|----------|----------------|---|
| Header | 1 | 0x02 |
| Length | 1 | 0x0b |
| Command | 1 | 0x0C |
| Data | 8 | Internal Temperature in Celsius (1 byte) = 0x00 Reserved Heatsink Temperature in Celsius (1 byte) Voltage in millivolts (2bytes) Set Current in milliamperes (2bytes) Current in milliamperes (2bytes) Loopback Current in milliamperes (2bytes) |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example:

> Request: 02 01 0c 0c 03

> Response: 02 0b 0c 00 1c 4d 14 00 00 00 00 00 00 54 03

Internal Temperature = 0, Heatsink Temperature = 26, Voltage = 5197 millivolts, Set Current = 0mA, Current = 0mA

Change Profile / Voltage

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|--|---------------------|------|
| 0x02 | 0x04 | 0x0D | Profile Index (1 byte) + Voltage (2bytes) | To be calculated | 0x03 |

Profile Indexes starts from 0 and can be optioned by sending “Get Capabilities” command.

Response:

| Header | Length (bytes) | Data |
|---------|----------------|-------------|
| Header | 1 | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0x0D |
| Data | 1 | Status Byte |

| | | |
|----------|---|----------------------------------|
| | | 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example - Select Profile #3:

> Request: 02 04 0d 03 20 4e 65 03

> Response: 02 02 0d 00 0e 03

Status = Successful

Set Current

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|--------------------------------------|------------------|------|
| 0x02 | 0x03 | 0x10 | Current in milliamperes (2 bytes) | To be calculated | 0x03 |

Response:

| | | |
|----------|-------------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0x10 |
| Data | 1 | Status 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Set Current 1000mA:

> Request: 02 03 10 e8 03 f9 03

> Response: 02 02 10 00 13 03

Status = Successful

Set Current (Fast)

This command can be used with an optional argument that controls the slope of the current as it changes. Using this argument, the time for the load to change can be calculated as:

$$\text{Time in milliseconds} = \frac{\text{Final voltage} - \text{Initial voltage}}{\text{Slope value}}$$

Command (Default):

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|--------------------------------------|------------------|------|
| 0x02 | 0x03 | 0x11 | Current in milliamperes (2 bytes) | To be calculated | 0x03 |

Command (Set slope):

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
|--------|--------|---------|------|----------|------|

| | | | | | |
|------|------|------|--|---------------------|------|
| 0x02 | 0x05 | 0x11 | Current in milliamperes (2 bytes) + Slope in milliamperes per millisecond (2 bytes) | To be calculated | 0x03 |
|------|------|------|--|---------------------|------|

Response:

| | | |
|----------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0x11 |
| Data | 1 | Status 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Quickly Set Current 1000mA:

```
> Request: 02 03 11 e8 03 f8 03
```

```
> Response: 02 02 11 00 12 03
```

Status = Successful

Step Load Test

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|--|---------------------|------|
| 0x02 | 0x03 | 0x13 | Initial current (2 bytes) + final current (2 bytes) | To be calculated | 0x03 |

Note: Three responses are sent consecutively with the same structure, the measurements are to be combined to form a list containing 75 measurements. The first measurement is the voltage when the step load is applied, then the following value is the voltage measured after *Sample time*.

Response:

| | | |
|----------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x34 |
| Command | 1 | 0x13 |
| Data | 51 | Sample time in microseconds (1 byte) Measurements in millivolts (2 bytes each x 25) |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Step load test from 0mA to 3000mA:

```
> Request: 02 05 13 00 00 b8 0b a4 03
```

```
> Response 1: 02 34 13 64 0a 15 ba 14 b3 13 1 13 a5 12 80 12 7a 12 7a 12 7a 12 74
12 7a 12 7a 12 74 12 7a 12 74 12 7a 12 74 12 7a 12 74 12 74 12 74 12 7a 12 7a
12 74 12 02 03
```

> Response 2: 02 34 13 64 74 12 7a 12 74 12 74 12 7a 12 74 12 7a 12 7a 12 74 12 74
12 74 12 7a 12 6d 12 7a 12 74 12 74 12 74 12 7a 12 7a 12 74 12 7a 12 74 12 74 12 7a
12 7a 12 33 03

> Response 3: 02 34 13 64 7a 12 74 12 7a 12 74 12 74 12 7a 12 74 12 7a 12 7a 12 7a
12 74 12 7a 12 7a 12 74 12 7a 12 7a 12 7a 12 7a 12 74 12 74 12 74 12 7a 12 7a 12 74
12 74 12 24 03

Sample Time = 100 microseconds

Set USB Internal Connection (PM125 only)

This command can be used to simulate a physical disconnection and reconnection of the device under test.

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|---|------------------|------|
| 0x02 | 0x02 | 0x14 | (1 byte) 0x00: USB Disconnected 0x01: USB Connected | To be calculated | 0x03 |

Response:

| | | |
|----------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0x14 |
| Data | 1 | Status 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Disconnect Device Under Test

> Request: 02 02 14 00 17 03

> Response: 02 02 14 00 17 03

Status = Successful

Get Hardware Sub-Revision (PM125 only)

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
| 0x02 | 0x01 | 0xD1 | - | 0xD1 | 0x03 |

Response:

| | | |
|---------|----------------|---|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0xD1 |
| Data | 1 | Sub-Revision 0x00: Hardware revision 2.0 0x01: Hardware revision 2.1 0x02: Hardware revision 2.2 |

| | | |
|----------|---|------------------|
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Hardware revision 2.0

> Request: 02 01 D1 D1 03

> Response: 02 02 D1 00 D2 03

Status = Successful

Change Config (Persistent)

Warning: Performing this operation too many times can burn out the flash memory, as all configuration parameters are written to flash memory after the change operation. For this reason, it is recommended to use the volatile option when possible.

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------------------------|---------|---|------------------|------|
| 0x02 | Depends on the parameter | 0xE0 | See Configuration Guide Section. | To be calculated | 0x03 |

Response:

| | | |
|----------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0xE0 |
| Data | 1 | Status 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Enable Loopback Port:

> Request: 02 03 e0 00 01 e3 03

> Response: 02 02 e0 00 e3 03

Status = Successful

Change Config (Volatile)

Warning: Configuration change will not persist after power loss, since no write to flash memory is performed.

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------------------------|---------|---|------------------|------|
| 0x02 | Depends on the parameter | 0xE6 | See Configuration Guide Section. | To be calculated | 0x03 |

Response:

| | | |
|----------|----------------|--|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0xE6 |
| Data | 1 | Status 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Enable Loopback Port:

> Request: 02 03 e6 00 01 e5 03

> Response: 02 02 e6 00 e5 03

Status = Successful

Get Config

Command

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|---|------------------|------|
| 0x02 | 0x02 | 0xE1 | See Configuration Guide Section. | To be calculated | 0x03 |

Response:

| | | |
|----------|--------------------------|---|
| Header | Length (bytes) | 0x02 |
| Length | 1 | Depends on the parameter |
| Command | 1 | 0xE1 |
| Data | Depends on the parameter | Loopback Port: 0x00: Disabled 0x01: Enables Max SDP Current: 2 bytes LSB First |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Loopback Port:

> Request: 02 02 e1 00 e2 03

> Response: 02 02 e1 01 e3 03

Loopback Port = Enabled

Set Backlight

Command:

| Header | Length | Command | Data | Checksum | Flag |
|--------|--------|---------|------|----------|------|
|--------|--------|---------|------|----------|------|

| | | | | | |
|------|------|------|--|------------------|------|
| 0x02 | 0x02 | 0xE2 | On/Off (1 byte) 0x00: Off 0x01: On | To be calculated | 0x03 |
|------|------|------|--|------------------|------|

Response:

| | | |
|----------|----------------|---|
| Header | Length (bytes) | 0x02 |
| Length | 1 | 0x02 |
| Command | 1 | 0xE2 |
| Data | 1 | Status Byte 0x00: Successful 0x01: Failed |
| Checksum | 1 | To be calculated |
| Flag | 1 | 0x03 |

Example – Turn backlight off:

> Request: 02 02 e2 00 e1 03

> Response: 02 02 e2 00 e1 03

Status = Successful

Configuration Guide

Configuration Overview

| Parameter | Description | Supported | |
|--------------------------|---|-----------|-------|
| | | PM125 | PM110 |
| Loopback Port | Connect loopback port to power and data lines from device under test. | ✓ | ✓ |
| Current Limit | Maximum current that can be drawn from device under test. | ✓ | ✓ |
| Max SDP Current | The maximum current for standard downstream ports can be defined by this option. | ✓ | ✓ |
| Upstream VBUS Estimation | When enabled, the displayed voltage is the upstream voltage from the device under test (estimated based on set cable resistance). When disabled, displayed voltage is voltage measured downstream at test unit. | ✓ | ✓ |
| Cable Resistance | Resistance of the cable connecting the device under test to the test unit. Used for estimating the upstream voltage from the device under test. | ✓ | ✓ |
| Default Profile Index | The initial profile index selected when the device under test is connected. In “Auto mode”, selects best matching profile to the default voltage. | ✓ | ✗ |

| | | | |
|--------------------------------|--|---|---|
| Default Voltage | The initial voltage set when the device under test is connected. | ✓ | ✓ |
| Default Current | The initial current set when the device under test is connected. | ✓ | ✓ |
| Operating Current | The value advertised by the test unit as the maximum current it will accept. If not required, set to “Max”. See Section 6.4.2.8 <i>Operating Current</i> from the <i>USB Power Delivery Specification Revision 3.1</i> for more details. | ✓ | ✗ |
| Sink Capability | The second capability of the sink advertised, alongside the first which is always 5V, 5A. If not required, set to “None”. This value is used when replying to a “Get Sink Capabilities” message, see Section 6.4.3.1 <i>Sink Capabilities Message</i> from the <i>USB Power Delivery Specification Revision 3.1</i> for more details | ✓ | ✗ |
| Profile Limit | The maximum current that can be set for profiles of corresponding type (Cannot exceed limit from specifications). Can also be set to “Disabled” to turn off detection of the Profile type. | ✓ | ✗ |
| Hold Load (on voltage changes) | When enabled, the current set will be maintained during all voltage and profile changes. If disabled, the current will go to 0 when the voltage or selected profile is changed. | ✓ | ✗ |

Configuration Commands

| Parameter | Data Bytes | | Example Set Command Data |
|--------------------------|------------|---|---|
| | Read | Set / Response data from read | |
| Loopback Port | 0x00 | Disable: 0x00 0x00 Enable: 0x00 0x01 | |
| Current Limit | N/A | Enforce Limits: 0x01 0x00 Allow 20% over current: 0x01 0x01 Force Limit: 0x01 0x02 <i>MaxCurrent</i> in milliamperes (2 Bytes) | Set maximum current to 3000mA: 0x01 0x02 0xB8 0x0B |
| Max SDP Current | 0x02 | 0x02 <i>MaxCurrent</i> in milliamperes (2 Bytes) | Set maximum current to 900mA 0x02 0x84 0x03 |
| Upstream VBUS Estimation | 0x03 | Disable: 0x03 0x00 Enable: 0x03 0x01 | |

| | | | |
|---|------|---|---|
| Cable Resistance | 0x04 | 0x04 <i>CableResistance</i> in ohms (2 Bytes) | Set cable resistance to 80 ohms: 0x04 0x50 0x00 |
| Default Profile Index | 0x05 | 0x05 <i>DefaultProfileIndex</i> (1 Byte) Or for Auto mode: 0x05 0xFF | Set default profile index to 2: 0x05 0x02 |
| Default Voltage | 0x06 | 0x06 <i>DefaultVoltage</i> in millivolts (2 Bytes) | Set default voltage to 2000mV: 0x06 0xD0 0x07 |
| Default Current | 0x07 | 0x07 <i>DefaultCurrent</i> in milliamperes (2 Bytes) | Set default current to 1500mA: 0x07 0xDC 0x05 |
| Operating Current | 0x08 | 0x08 <i>MaxOperatingCurrent</i> in milliamperes (2 Bytes) Or for Max: 0x08 0xFF 0xFF | Set maximum operating current to 1500mA: 0x08 0xDC 0x05 |
| Sink Capability | 0x09 | 0x09 <i>Voltage</i> in millivolts (2 Bytes) <i>Current</i> in milliamperes (2 Bytes) Or for None: 0x09 0xFF 0xFF 0xFF 0xFF | Set sink capability to 15V, 3A: 0x09 0x98 0x3A 0xB8 0x0B |
| Hold Load | 0x16 | Disable: 0x16 0x00 Enable: 0x16 0x01 | |
| Enable PD-PPS / PD-AVS profiles | 0x17 | Disable: 0x17 0x00 Enable: 0x17 0x01 | |
| Apply default configuration on capability changes | 0x18 | Disable: 0x18 0x00 Enable: 0x18 0x01 | |
| Fail if Default Voltage is not supported | 0x19 | When Default Voltage fails: Set closest lower voltage: 0x19 0x00 Set 5V: 0x19 0x01 | |
| Fail if Default Current is not supported | 0x1A | When Default Current fails: Set max available: 0x20 0x00 Set 0A: 0x20 0x01 | |