

USB Power Delivery Tester

Communication Protocol Guide

Model PM110 & PM125

Document Edition: 1.9
Date: 3 November 2021
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.....	6
Get Capabilities.....	7
Get Statistics	8
Change Profile / Voltage.....	9
Set Default Voltage.....	9
Set Default Load	10
Set Default Profile Index	10
Set Load	11
Set Load (Fast).....	11
Change Config	12
Get Config.....	13
Set Backlight.....	13

Introduction

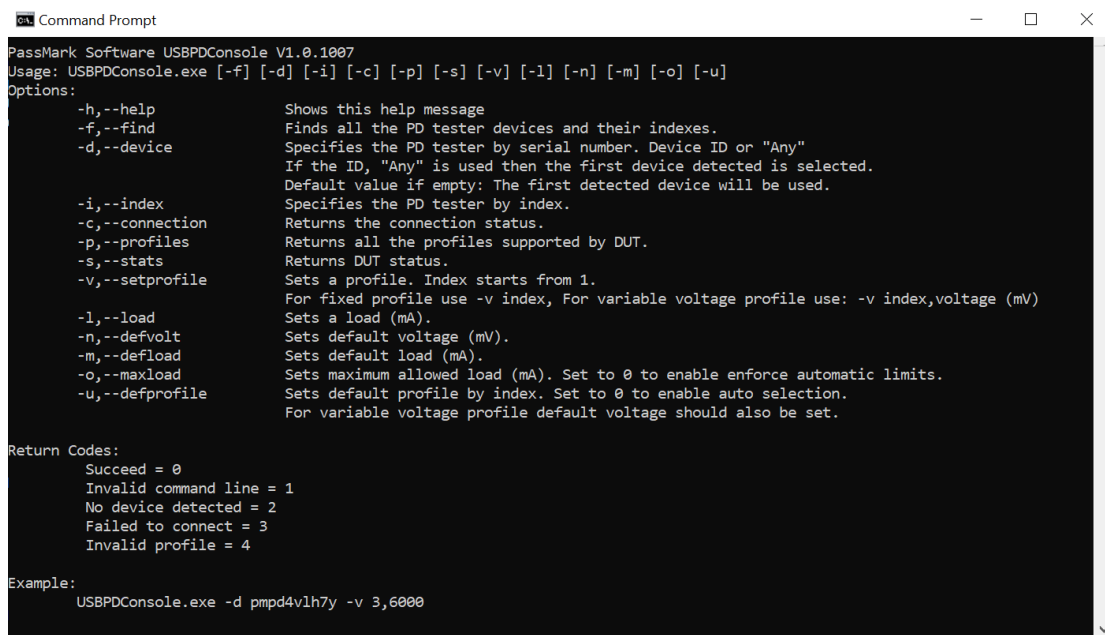
This document will demonstrate how to communicate with Passmark USB Power Delivery testers while using the [FTDI D2XX Interface](#). You may need to update device firmware as the information in this document only applies to firmware version 2.5 and later.

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 V1.0.1007.



```
Command Prompt
PassMark Software USBPDConsole V1.0.1007
Usage: USBPDConsole.exe [-f] [-d] [-i] [-c] [-p] [-s] [-v] [-l] [-n] [-m] [-o] [-u]
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.
    -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 a load (mA).
    -n,--defvolt              Sets default voltage (mV).
    -m,--defload              Sets default load (mA).
    -o,--maxload              Sets maximum allowed load (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.

Return Codes:
    Succeed = 0
    Invalid command line = 1
    No device detected = 2
    Failed to connect = 3
    Invalid profile = 4

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.

Command	Description
0x01	Get Version
0x0A	Get Connection Status
0x0B	Get Capabilities
0x0C	Get Statistics
0x0D	Change Profile / Voltage
0x0E	Set Default Voltage
0x0F	Set Default Load
0x12	Set Default Profile Index
0x10	Set Load
0x11	Set Load (Fast)
0xE0	Change Config
0xE1	Get Config
0xE2	Set Backlight

Get Version

Command:

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

Response:

Header	Length (bytes)	0x02
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	0x00: Apple 0.5A, SDP, QC1, USBC 1.5A, PD Fix 0x01: Apple 1A, CDP, QC2, USBC 3A, PD Battery 0x02: Apple 2.1A, DCP, QC3, PD Variable 0x03: Apple 2.4A, 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

Response:

Header	Length (bytes)	0x02
Length	1	0x3B
Command	1	0x0B

Num Power Profiles	2	Number of power profiles
Power Data Objects	35	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

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 (USBC 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)	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 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)	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 Default Voltage

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	0x02	0x0E	Voltage in millivolts (2 bytes)	To be calculated	0x03

Response:

Header	Length (bytes)	0x02
Length	1	0x02

Command	1	0x0E
Data	1	Status Byte 0x00: Successful 0x01: Failed
Checksum	1	To be calculated
Flag	1	0x03

Example – Set Default Voltage to 15V:

> Request: 02 03 0e 98 3a ae 03

> Response: 02 02 0e 00 0d 03

Status = Successful

Set Default Load

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	0x02	0x0F	Load in milliamperes (2 bytes)	To be calculated	0x03

Response:

Header	Length (bytes)	0x02
Length	1	0x02
Command	1	0x0F
Data	1	Status Byte 0x00: Successful 0x01: Failed
Checksum	1	To be calculated
Flag	1	0x03

Example – Set Default Load to 1000mA:

> Request: 02 03 0f e8 03 e6 03

> Response: 02 02 0f 00 0c 03

Status = Successful

Set Default Profile Index

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	0x02	0x12	Profile Index (1 byte): index starts from 0 0xFF: Automatic profile selection based on default voltage and load	To be calculated	0x03

Response:

Header	Length (bytes)	0x02
Length	1	0x02
Command	1	0x0E
Data	1	Status Byte 0x00: Successful 0x01: Failed
Checksum	1	To be calculated
Flag	1	0x03

Example – Set Default Profile Index to 3:

> Request: 02 02 12 02 13 03

> Response: 02 02 12 00 11 03

Status = Successful

Set Load

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	0x03	0x10	Load 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 Load 1000mA:

> Request: 02 03 10 e8 03 f9 03

> Response: 02 02 10 00 13 03

Status = Successful

Set Load (Fast)

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	0x03	0x11	Load in milliamperes (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 – Set Load 1000mA:

> Request: 02 03 11 e8 03 f8 03

> Response: 02 02 11 00 12 03

Status = Successful

Change Config

Command:

Header	Length	Command	Data	Checksum	Flag
0x02	Depends on the parameter	0xE0	See Table below	To be calculated	0x03

Parameter	Data Bytes
Loopback Port	Disable: 0x00 0x00 Enable: 0x00 0x01
Current Limit	Enforce Limits: 0x01 0x00 Allow 20% over current: 0x01 0x01 Force Limit: 0x01 0x02 MaxCurrent in milliamperes (2 Bytes) Example: Set maximum current to 3000mA 0x01 0x02 0xB8 0x0B
Max SDP Current	0x02 MaxCurrent in milliamperes (2 Bytes) Example: Set maximum current to 900mA 0x02 0x84 0x03
Upstream VBUS Estimation	Disable: 0x03 0x00 Enable: 0x03 0x01
Cable Resistance	0x04 CableResistance in ohms (2 Bytes) Example: Set cable resistance to 80 ohms 0x04 0x50 0x00 Note: Cable Resistance is effective when “Upstream VBUS Estimation” is enabled.

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

Status = Successful

Get Config

Command

Header	Length	Command	Data	Checksum	Flag
0x02	0x02	0xE1	See Table below	To be calculated	0x03

Parameter	Data Bytes
Loopback Port	0x00
Max SDP Current	0x02
Upstream VBUS Estimation	0x03
Cable Resistance	0x04

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