Overview

The PD charger + battery example is a simple demonstration based on the MCUXpresso SDK PD stack.
The application simulate battery product (for example: laptop), it prints the battery percent continually.
The demo works as DRP. When connect, the board can be source or sink.

System Requirement

Hardware requirements

- One or two Type-C shield board

- One or two 9V DC power suppliers

- Type-C Cable

- One or two hardwares (Tower module/base board, and so on) for a specific device, for example: lpcxpresso54114 board

- Personal Computer

Software requirements

- The project files are in:

  <MCUXpresso_SDK_Install>/boards/<board>/usb_examples/usb_pd_charger_battery/<rtos>/<toolchain>.

  Note

   The <rtos> is Bare Metal or FreeRTOS OS.

- Terminal tool.

Getting Started

Hardware Settings

- Connect JP41 1-2, JP44 1-2, JP51 1-2, JP52 1-2, JP61 2-3, JP62 2-3, JP64 2-3, JP65 2-3, R523 2-3.

For detailed instructions, see the appropriate board User's Guide.

Note

   Set the hardware jumpers (Tower system/base module) to default settings.

Prepare the example

1. Download the program to the target board.

2. Power on Type-C shield board then power on development board.

Run the example

1. Connect the board to one charger or another shield board + development board (download this program too) with Type-C cable.

2. Connect the OpenSDA USB port to the PC and open terminal.

3. If running as source after connect.

   - If battery is (30%, 100%]: source caps are high power (5V/9V).

   - If battery is (0%, 30%]: source caps are low power (only 5V).

   - If partner request power role swap: accept.

   - prefer to work as sink.

- get partner source cap, and judge whether to swap (if partner is external powered and satisfy self request).
    * if get partner source cap fail, retry one more time.
    * if can swap, then start power role swap.
    * if cannot swap or get source cap fail. still work as source.
- do power role swap.
    * if fail, re-try one more time, if still fail, still work as source.
    * if success, please reference to sink description.
- battery will decrease (if battery decrease to [0%, 20%]):
    - get partner source cap, and judge whether to swap (source cap satisfy self request).
        * if get partner source cap fail, retry one more time.
        * if can swap, then start power role swap.
        * if cannot swap or get source cap fail, send low power source caps.
    - do power role swap.
        * if fail, re-try one more time, if still fail send low power source caps.
        * if success, please reference to sink description.
    - if swap fail, after send low power source caps to trigger source capabilities change.
        * retry swap with interval 10s.
    - if battery decrease to 5%:
        * stop provide vbus.
        * set Try.SNK

4. Running as sink after connect.

- when connect:
    - if battery is less than 100%: request high power (9V).
    - if battery is 100%: request low power (5V).
- when battery increase to 100%, request low power (5V).
- If partner request power role swap
    - if battery is (30%, 100%]: accept.
    - if battery is [0%, 30%]: reject.
- when battery increase to (30%, 100%]:
    - update self source caps as high power.