Application Report

TPS65988DH: Integrated Type-C PD Dual-Port Wall Socket With Power Balancing

Prajith Cheerakkoda

ABSTRACT

Higher power and wider compatibility with the battery powered devices makes a Type-C PD power port the preferred choice of today's gadget lovers. Consequently, integrated Type-C PD wall sockets are gaining more popularity than the Type-A integrated wall sockets at a faster rate. Why TI's highly configurable and integrated PD controller TPS65988DH is an integral part of your next Type-C/ PD design is the topic of this application note. With the highly flexible GPIO event feature, TI PD controller can implement the control logic required for power balancing between the ports and function as a complete standalone solution. This feature allows for an otherwise mandatory MCU (Microcontroller unit) to be removed from the system, alleviating firmware development and system complexity. This application note walks through how GPIO event feature has been utilized to implement power balancing between the ports. This feature is beneficial when designing end equipments like smart plugs, multi-port adapters, avionics seat backs, and other multi-port source-only applications using TI's highly configurable PD controller.

This application note also covers step by step instructions on GUI configuration for implementing the control logic required for power balancing. The example project file available with this application note can be used for testing the control logic on TPS65988EVM.

Table of Contents

List of Figures

Trademarks

USB Type-C® is a registered trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

In an effort to reduce e-waste, USB-IF redefined the role of a USB port as a universal interface for power too. Focus was given for faster and efficient power transfer to meet the growing needs of battery-powered applications. This approach was widely accepted in the USB world and is also gaining popularity among the non-USB OEMs as a preferred method of power delivery, such as electric shavers, portable speakers. Unlike Type-A ports on the wall socket, a Type-C PD port with higher power can serve a wider spectrum of consumer electronics and really help to reduce adapter clutter.

The intrinsic design challenges (restricted PCB area and thermal dissipation inside the gang-box) in a wall socket design make the Type-C PD integration more challenging. That being said, let's see how TI's highly integrated and configurable TPS665988DH PD controller can be of help to roll out your next Type-C PD wall socket, with a short development cycle to keep up market pace.

2 Integrated USB Wall Socket Design

An integrated USB wall socket accommodates USB charging ports on the face plate along with the AC outlet terminals.

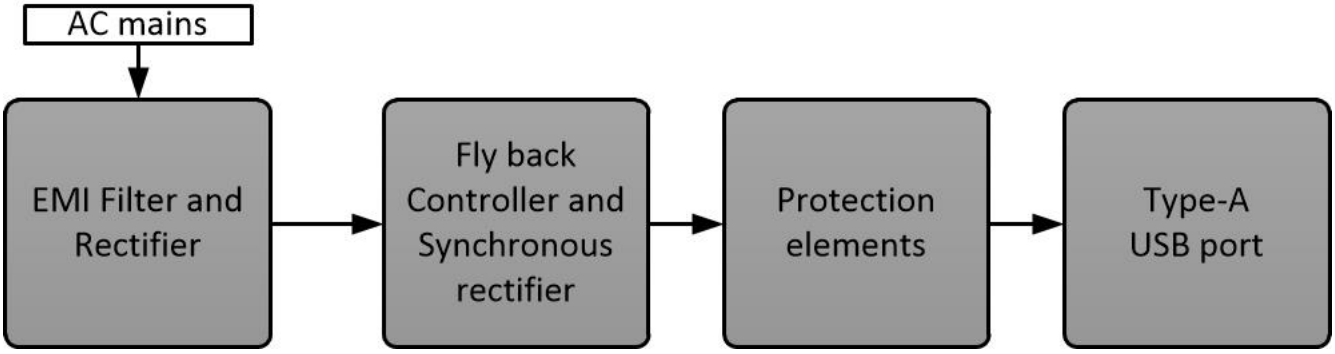

Figure 2-1. Integrated Type-A USB Charging Port Block Diagram

Figure 2-1 shows different blocks that constitute a conventional USB Type-A charging port on a wall socket. The EMI filter and rectifier block generate high voltage DC from AC mains input. The high frequency isolated fly back controller and synchronous rectifier block provides regulated DC for the charger port. The protection elements are meant for protecting the USB port from faults such as under voltage protection (UVP), over voltage protection (OVP), reverse current protection (RCP) and over current protection (OCP).

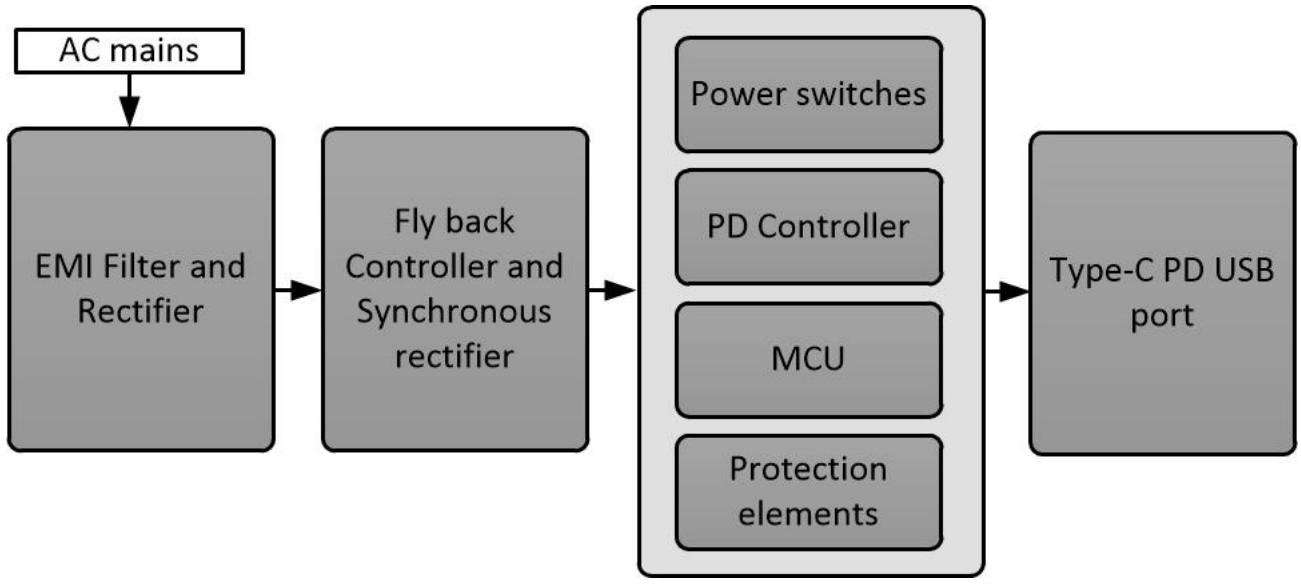

Figure 2-2. Integrated Type-C PD USB Charging Port Block Diagram

Now, as shown in Figure 2-2 a Type-C PD charging port requires few additional components compared to conventional Type-A port.

1. PD controller: To add Type-C PD functionality.
2. Power switch: to control the power delivery to the port as per PD specification.
3. Protection elements: to protect USB port from faults such as OVP, UVP, OCP, RCP, and so forth.
4. MCU: Required in fixed function PD controller to implement customized power policy (if required).

Manufacturers of plugs and sockets have to follow mechanical dimensions defined by the regulatory standards (Eg BS 1363). One electrical gang box is usually of 4”(L)x2.75”(W)x2.5”(D) in size. The actual available area for PCB inside the box will be smaller than this as the gang box behind the front panel has to accommodate thick wires connecting AC terminals on the face plate. Now, with these restrictions, finding additional space for the Type-C PD related components in the wall socket design can be a challenge for system designers.

Today, most of the PD controllers in the market require external power mosfets for controlling power delivery to the Type-C port. System designers using such PD controllers will have to build their own protection circuits around the external power switches with the expense of additional PCB area and BOM. As per Type-C/ PD specification OCP protection is mandatory for Source ports (Type-C PD port that delivers power). TI's highly integrated dual Port PD controller TPS665988DH comes with integrated power switches and built in protection features, avoiding the need for these additional external components.

USB PD device applications often require customizations in order to achieve desired system behavior. For instance, customization is required in the dual port wall socket design (to be discussed later in this application note) for live power balancing between the ports. Any customization in the core firmware affects reliability, USB PD compliance and also slows down the development cycle. Typically, systems using fixed function PD controllers are required to have an on board microcontroller in this regard, with the TI PD controller being an exception. The GPIO event features of TPS65988DH allow PD system designers to customize the PD controller behavior using GUI software tool without affecting the core PD firmware.

2.1 Thermal Considerations

Face plate is the only means for dissipating the heat generated inside a gang box. Hence, focus should be given in selecting components with less power dissipation. In integrated wall socket designs, the maximum output power of USB ports are capped by the efficiency of ACDC and DCDC regulators. The power dissipated in TPS665988DH's internal power switch with Rdson of 28m Ω (at Junction temperature of 84℃) is very less compared to the heat contributed by the aforementioned components on the power path. The junction temperature calculation at ambient temperature of 65℃ shows that TPS665988DH is well below the max junction temperature of 150℃.

- Ambient temperature inside the Gang box (mainly contributed by ACDC transformer) (Ta) = 65℃
- Junction-to-ambient thermal resistance of TPS65988DH ($R_{\theta Ja}$) = 36.4℃/W
- Power dissipation on internal power paths due to IPPHV(continuous current through power path from PP_HVx to VBUSx) of 3A = Rdson * Iout * Iout = 28 m Ω * 3 A * 3A = 252 mW.
- Total power dissipation from both power paths = 504 mW
- Total power dissipation (considering GPIO leakage and active power I_{VIN_3V3}) P ~= 520mW
- Junction temperature = Ta + ($R_{\theta Ja}$ * P) = 65 ℃ + (36.4℃/W * 0.52W) = 83.9℃

The Junction temperature can be brought down to a lower value by careful PCB design as recommended in the data sheet.

The above thermal calculation is considering the architecture (shown in Figure 2-3), which uses internal power paths of the TI PD controller. In this design, the Buck-Boost regulator is fed from the output of ACDC regulator. Therefore, the ACDC regulator is required to maintain the minimum voltage required for the Buck-Boost regulator to operate. The I2C master of PD controller can be programmed to configure the regulators to supply fixed DC voltage based on the PD negotiation on the respective ports. More details on configuring TPS665988DH's I2C master to trigger slave write on specific PD events can be found Using I2C Master in TPS65987D and TPS65988 PD Controllers application report.

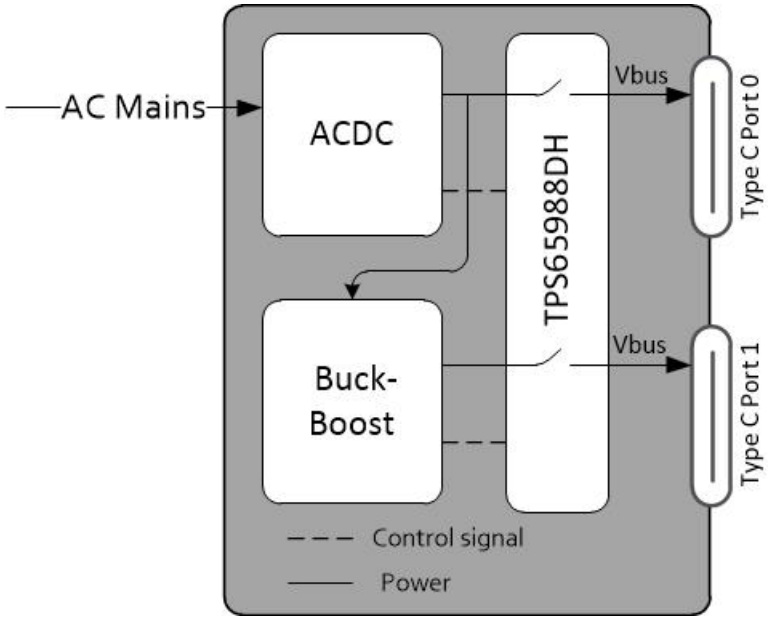

Figure 2-3. Dual Port Integrated Type-C PD Wall Socket Architecture Using TPS665988DH's Internal Power Switch

2.2 Dynamic Port Power Balancing

Higher level of integration available in TPS665988DH PD controller makes it the best suited part for power socket design. Protection features such as over current, over voltage, under voltage, and reverse current protections are highly desirable in any kind of power supply designs and most of the time system designers have to add these protection elements externally. The 100W capable internal integrated power paths come with all these protection features built in it.

For effective use of system power, the design is required to dynamically share the available power between the ports based on the port connection as listed below.

- When no ports are connected, the total power shall be made available to the first device, getting connected to either of the ports.
- When the second device gets connected, the total power shall be split equally b/w the two ports.

For the discussion, let's assume that the maximum system power available is 60W.

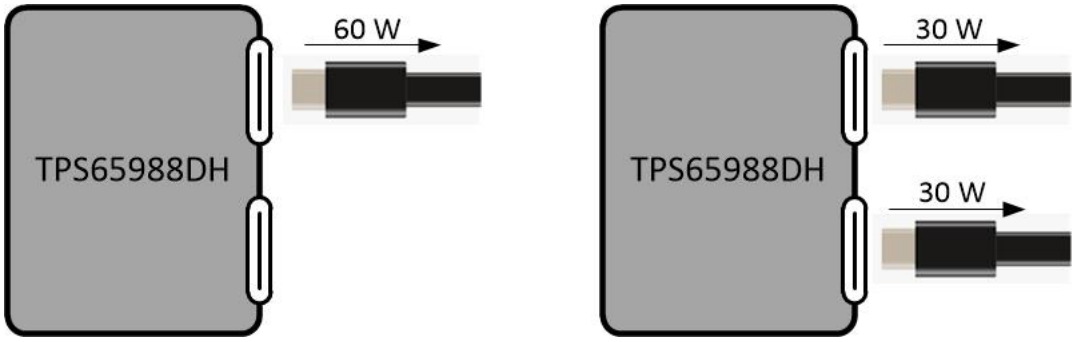

Figure 2-4. Dual Port Power Socket Design With Power Balancing

Without firmware modification, this can be achieved using real-time loading of modified PD Port configuration, which usually requires an external MCU. The same design can be made possible with TPS665988DH using GPIO events, without any firmware modification and without the need for an external MCU.

Each port can be configured to have two port power configurations and select them dynamically based on the control logic.

- **Configuration-1:** *100% Power PDO* [1] (advertise total power available to the system, 60W in this example, when only one port is connected)
- **Configuration-2:** *50% Power PDO* (50% of the total power, 30W in this example, when both ports are connected).

Note

Power Data Object (PDO): Data Object used to expose a Source Port's power capabilities as part of a Source_Capabilities message.

GPIO events are used to dynamically load the appropriate configuration (100% or 50% PDOs) at run time for a particular port, based on Plug Event from the other port. The following section introduces the GPIO events and its usage for this particular design as an example.

2.3 GPIO Events

GPIO events feature enables users to map a PD or USB Type-C event inside the PD controller to a GPIO (see Figure 2-5).

- An output GPIO event asserts/de-asserts a GPIO pin based on the user selected polarity, when the mapped PD/ Type-C event occurs. For example, *Plug Event* can be used to reflect Type-C plug/unplug on a GPIO.
- Similarly, an Input GPIO event (rising or falling edge on a TPS665988DH GPIO pin, from an external signal in the system) can be used to trigger a specific PD or USB Type-C event inside the PD controller. For example, the *Fault Input Event* allows external devices to trigger error recovery on a given port.[2].

Using the Configuration Tool, a predefined event picked from the drop-down menu can be mapped to any of the GPIOs. A list of all the GPIO events can be found in the *TPS65987DDH and TPS65988DH Host Interface Technical Reference Manual* of the PD controller.

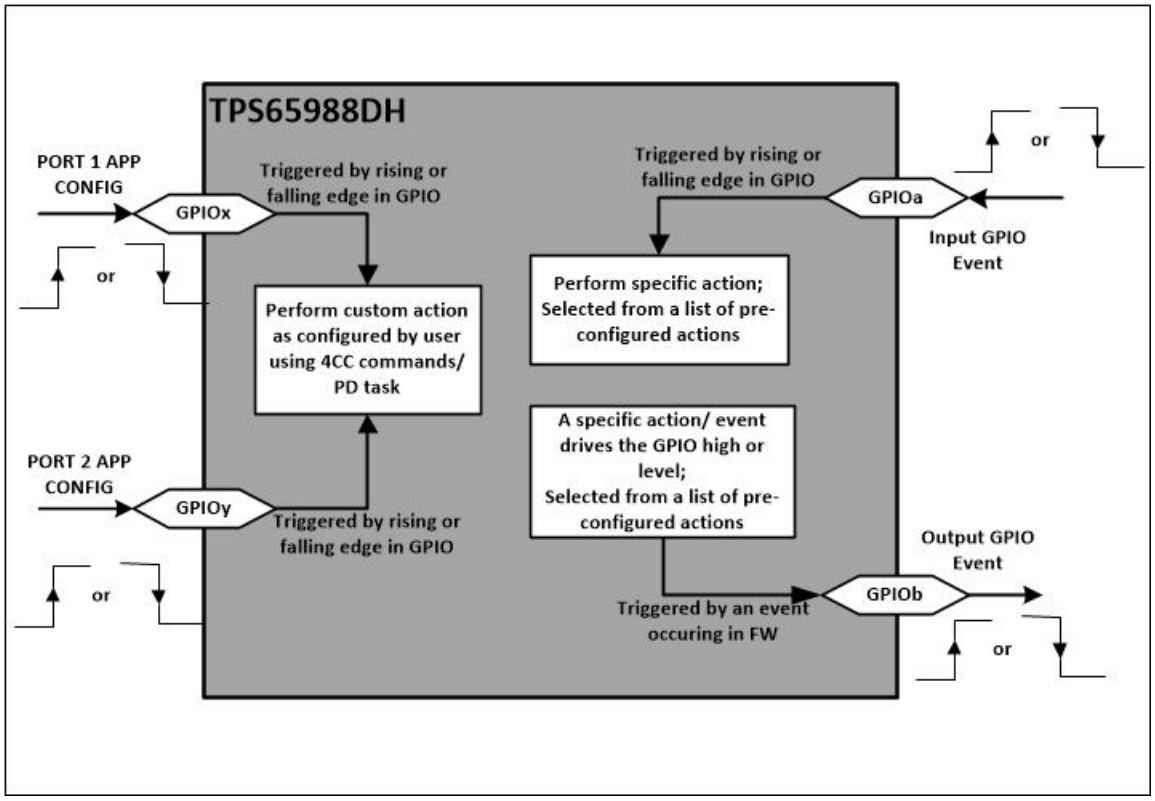

Figure 2-5. GPIO Event

- Unlike input GPIO event that triggers a predefined PD or USB Type-C® event, an Application Configuration GPIO event provides more flexibility in the design to trigger custom actions using an input GPIO (see Figure 2-5). Custom actions could be configured by the user using 4CC commands or PD tasks in Configuration

Tool, which otherwise needs an EC for issuing the same. In addition, an input GPIO mapped to an Application configuration event can be used to load a modified configuration to the PD controller at run-time.

Note

- For details on 4CC commands/PD tasks, see the *TPS65987DDH and TPS65988DH Host Interface Technical Reference Manual*.
- Error recovery is a Type-C state to recover from any fault state. During error recovery state, a Type-C port removes CC termination for a defined amount of time and puts it back. This is to simulate a detach event and to make a new connection.

2.4 Control Logic Implementation with GPIO Events

Figure 2-6 shows pictorial representation of Events and GPIOs used in the Dual port power socket design with power balancing.

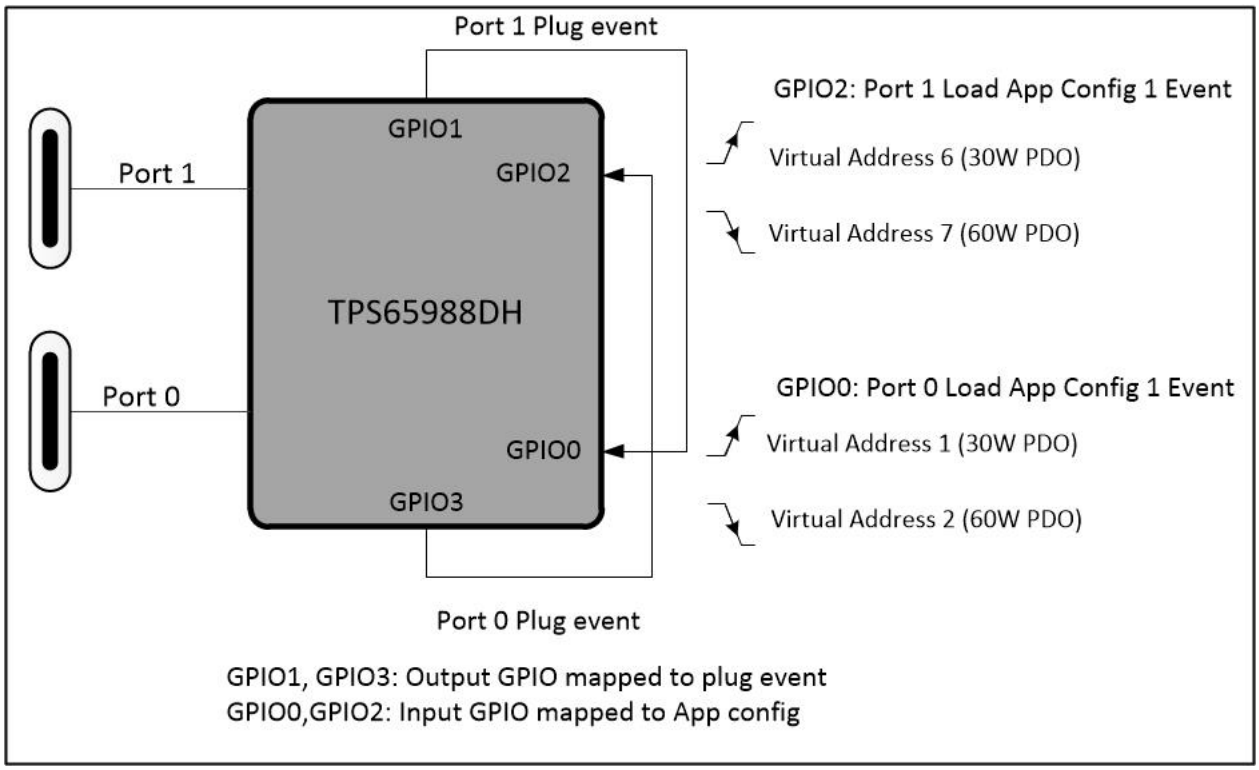

Figure 2-6. GPIO Events Usage in Dual Port Power Socket Design with Power Balancing

- Output Plug event GPIOs: GPIO1 (Port-1 plug event) and GPIO3 (Port-0 plug event) are configured to assert on *PLUG_EVENT* event. These output GPIOs should assert high when a device is connected to the corresponding Port.
- Application Configuration event GPIOs: GPIO2 (Port-1 App Config) and GPIO0 (Port-0 App Config) are input GPIOs configured to load the appropriate configuration (100% or 50% PDOs) based on the input signal and also execute a PD message task (to send SSrC - PD Send Source Capabilities).

The output Plug event GPIOs are connected to the input App Config GPIOs of the alternate port. When the input GPIO of a particular port is high (implies plug event on the other port), input GPIO event shall load 50% Power PDO configuration. Otherwise, when the input GPIO of a particular port is low (implies no *PLUG_EVENT* on the other port, for example, when the other port in disconnected state), input GPIO event shall load *100% Power PDO* configuration. This implementation can be tested on TPS65988D EVM using the project file available on the E2E Design Support forum.

2.5 GUI Configuration

1. The first step is to create a new DFP only (Source only) project by loading the default template *TPS65988DDH_advanced_v6_1_2.tpl*.
   a. Open TPS6598x Configuration Tool and click the *Project* tab.
   b. Select TPS65988DDH→Advanced→Downstream Facing Port (DFP) only.

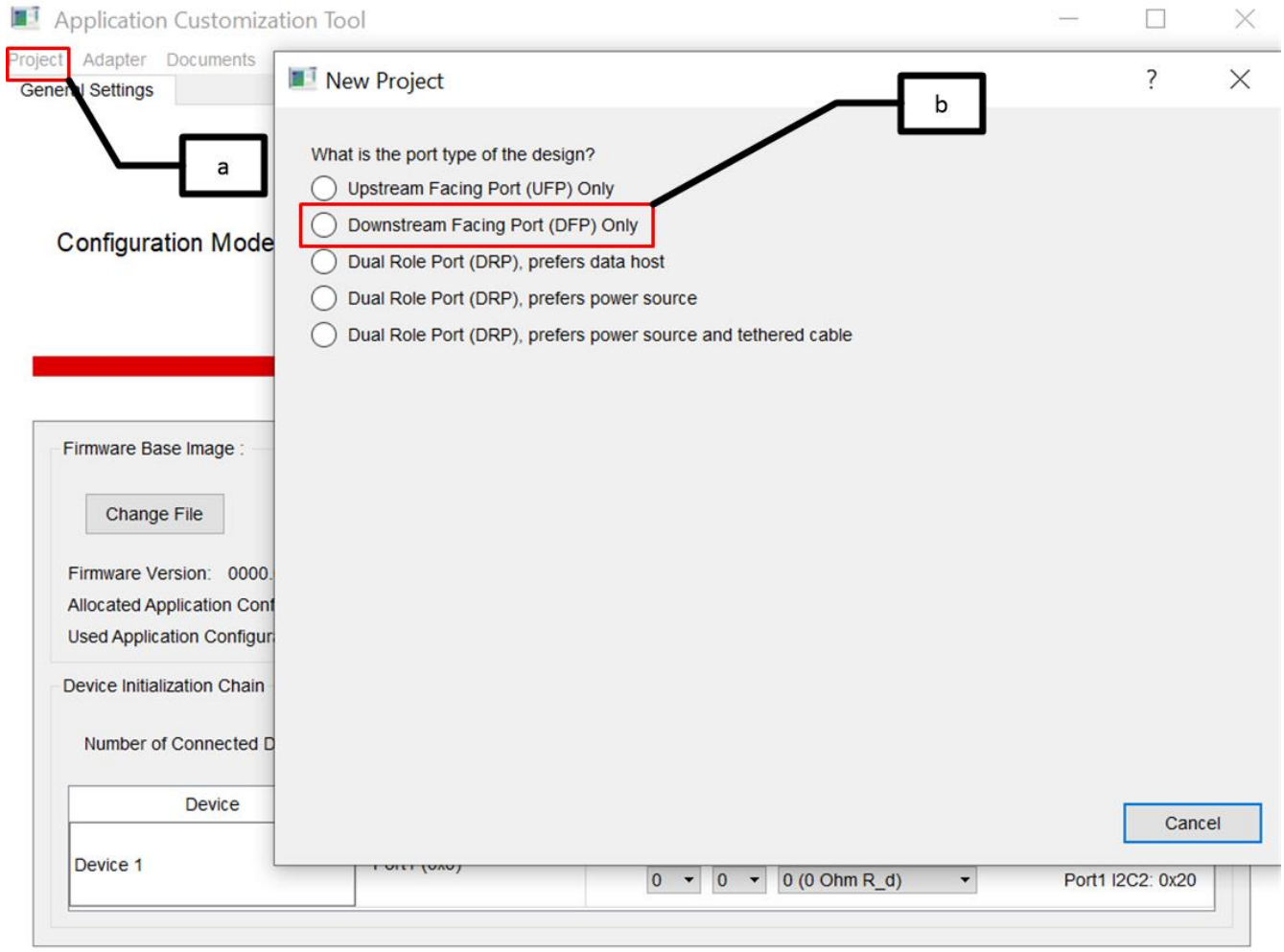

Figure 2-7. Load Template File

2. As shown in Figure 2-6, the application requires two sets of power profiles per port: corresponding 30W and 60W PDOs. In this step, generate these configurations and map them to virtual addresses.
   a. Change Number of Configuration Sets in General Settings to four.
   b. Rename Configuration Set as shown below for convenience.
      i. Virtual Address 1 -> Port-0 30W
      ii. Virtual Address 2 -> Port-0 60W
      iii. Virtual Address 3 -> Port-1 30W
      iv. Virtual Address 4 -> Port-1 60W
   c. Now, map Configuration sets to *Virtual Address* as shown in Figure 2-8.

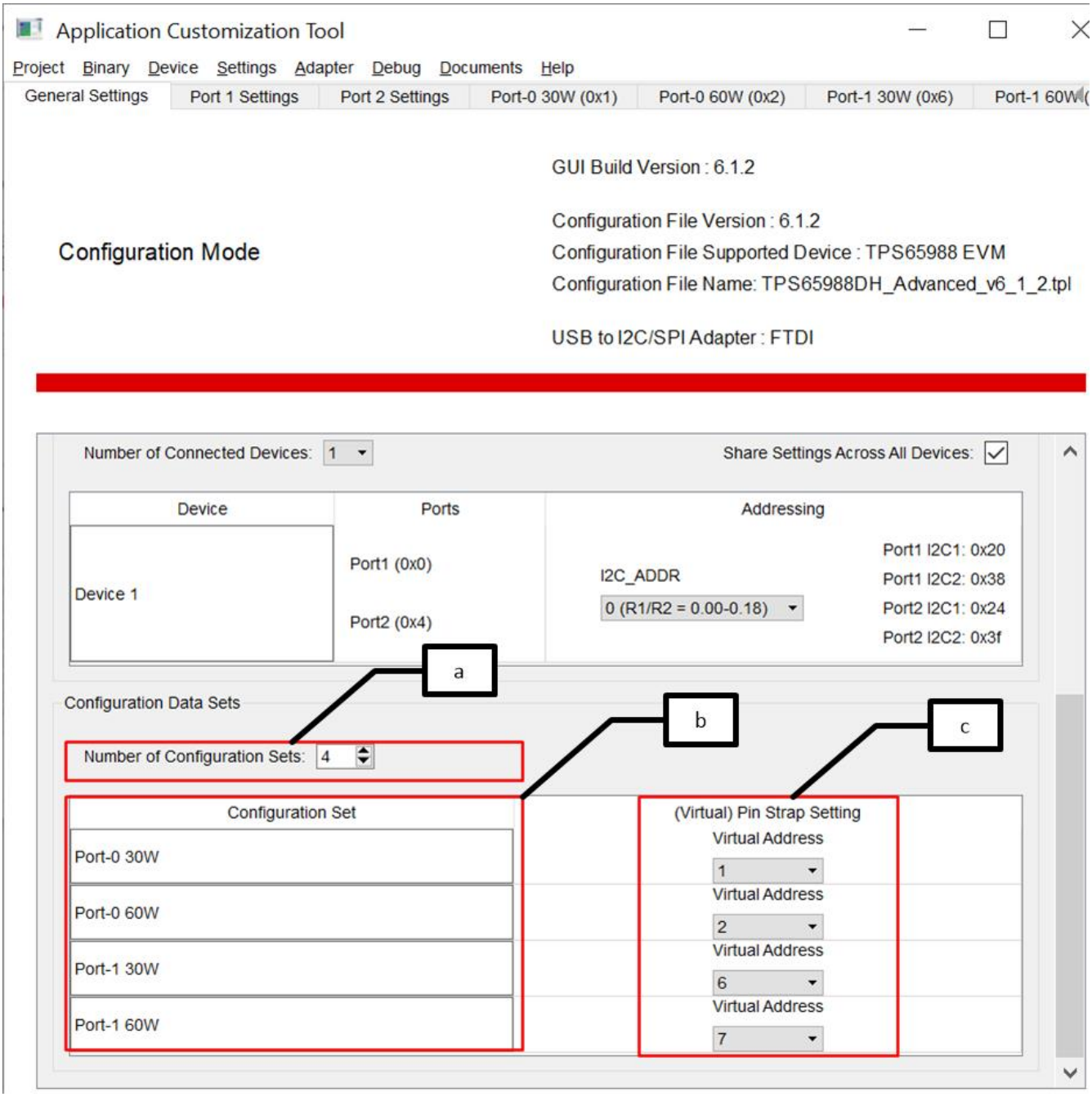

Figure 2-8. Generate Configuration Sets and Map to Virtual Address

Note

Do not use address 0x00 and 0x04 as they are reserved for port configurations.

3. The App configuration register (0x6C) holds the index pointing to the application configuration that needs to be loaded when the GPIO assert/ de-assert and also holds the command channel interface for executing the PD message task. In this step, modify Port-0 *App Configuration Register* to point to the application configuration index *Virtual Address 1* on GPIO high transition (corresponding to application configuration Port-0 30W) and point to *Virtual Address 2* on GPIO high transition (corresponding to application configuration Port-0 60W) and then execute SSrC task using CMD1 channel.
   a. Click the *Port 1 Settings* tab.
   b. Click the *App Configuration Register*.

c. The index of configurations that need to be loaded on GPIO low/high transition are selected from the drop-down menu below *Value* tab.
d. The default command channel CMD1 (0x08) can be used for executing PD message task.
e. Select PD message task *SSrC* from the drop-down menu to transmit the Source Capability configured in the application configuration space indexed by Virtual Address 1 or Virtual Address 2. Enter *!CMD* in *not Task* field to indicate no action.

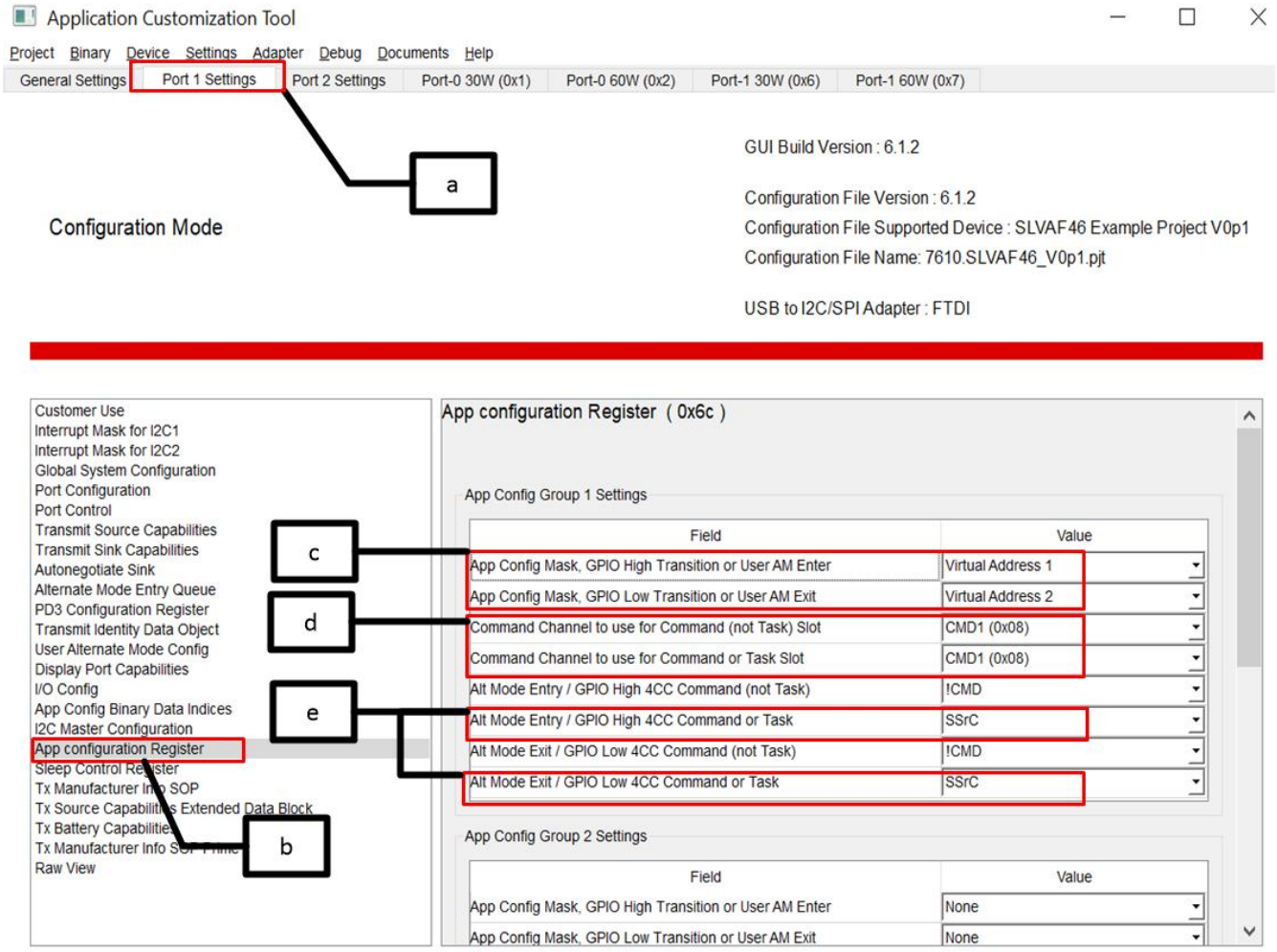

Figure 2-9. App Configuration Register Entry

4. Similarly, make entry for Port-*1 App Configuration Register* by referring the pjt file attached.
5. Now, configure Transmit Source Capabilities register (0x32) pointed by Virtual Address indexes. Virtual address 1 is the index corresponding to 30W PDO.
   a. Click *Port-0 30W (0x1)* and then click *Adjust Registers*.
   b. From the pop-up box, select *Transmit Source Capabilities (0x32) Register*.
   c. Click *OK*.
   d. Click *Transmit Source Capabilities* and make entry for 30W PDO as shown in Figure 2-11.

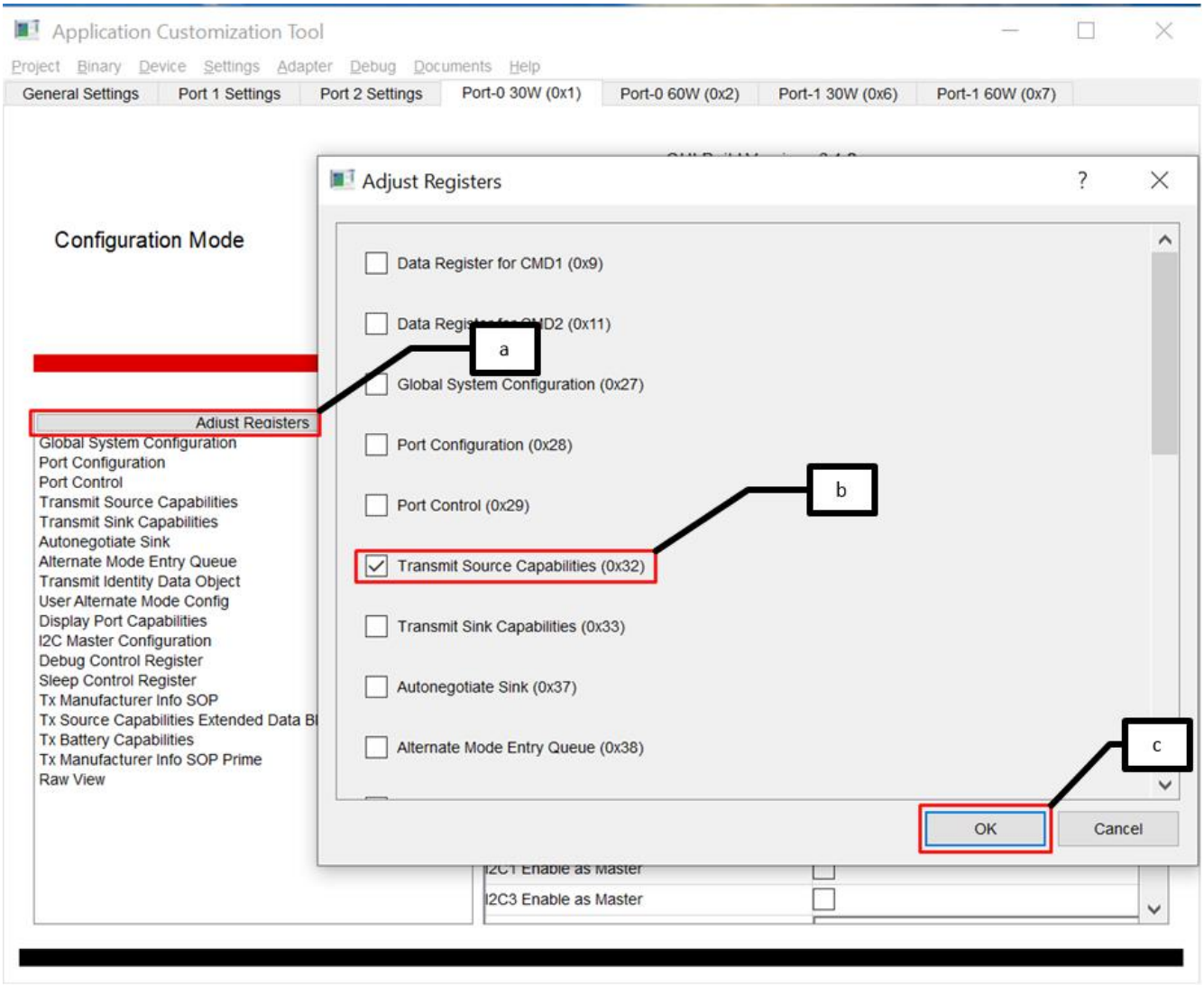

Figure 2-10. Select Transmit Source Capabilities Register

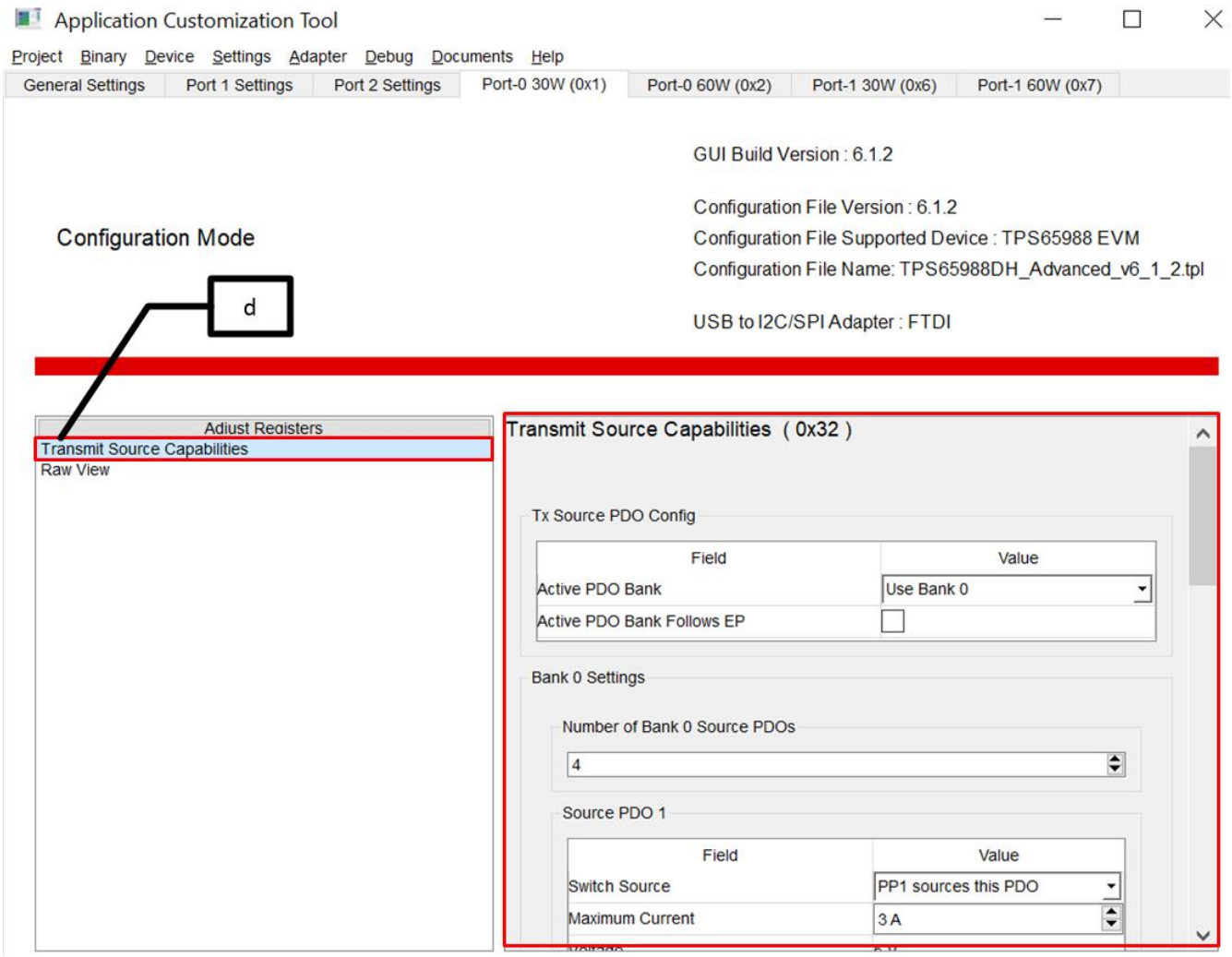

Figure 2-11. Configure Transmit Source Capabilities

6. Similarly, make entries for remaining Virtual Addresses referring the pjt file attached with this application.
7. Map App Config and GPIO events to respective GPIOs as shown in Figure 2-12.
   a. Select *IO Config* Register.
   b. Map events for each GPIO. Refer to project file for GPIO 2 and 3.

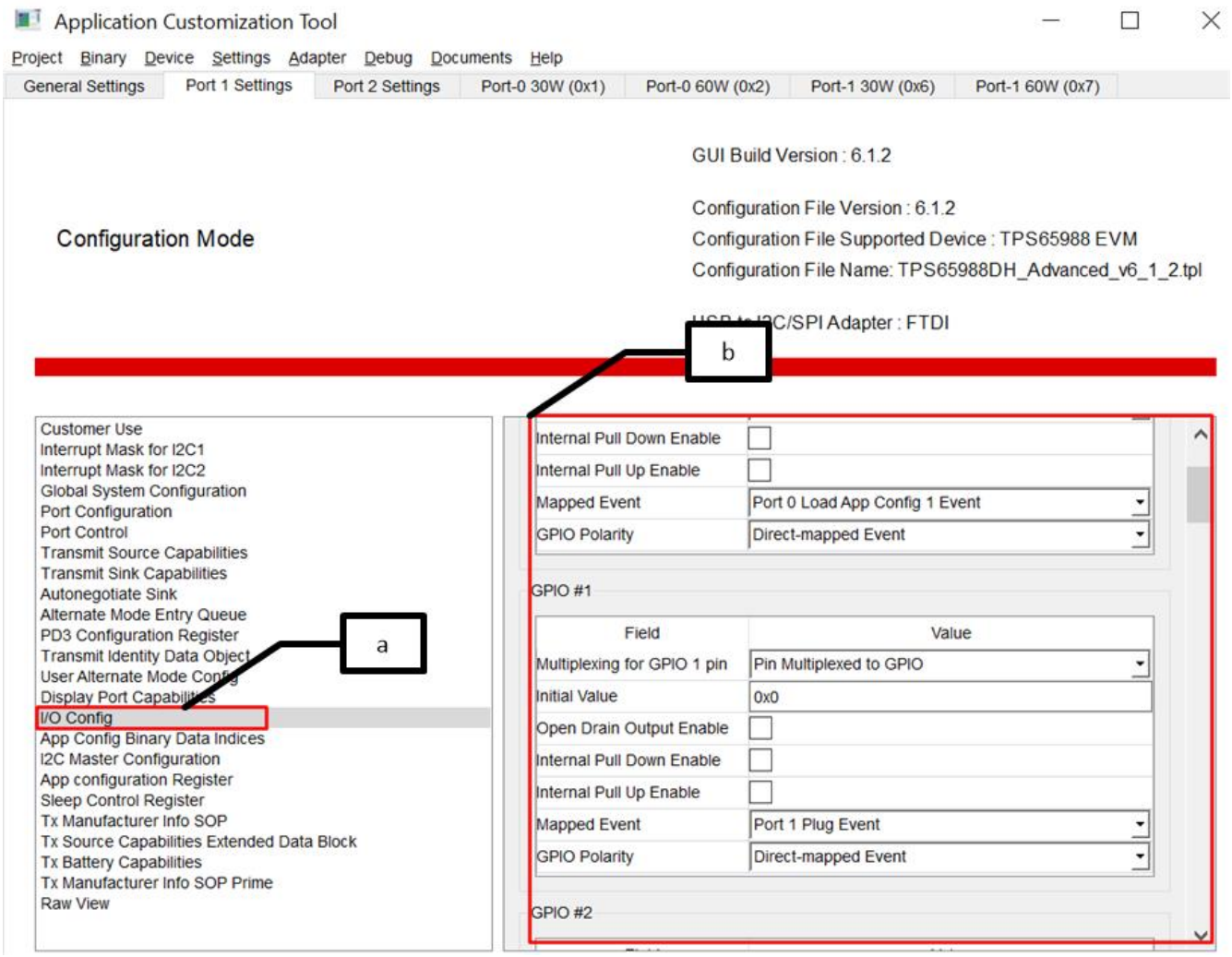

Figure 2-12. GPIO Mapping

2.6 Test Setup and Results

The control logic to balance port power (as shown in Figure 2-6) can be tested on TPS65988EVM using the project file. TPS65988EVM has two Buck regulators (LM3489) corresponding to Port A and Port B as shown in Figure 2-13. The GPIO pins GPIO12 and GPIO13 are mapped to output GPIO events *Port 0 Source PDO Negotiated TT1* and *Port 0 Source PDO Negotiated TT2* events, respectively. These GPIOs toggle based on the negotiated PDO and set the feedback voltage of LM3489 accordingly, to generate the negotiated PDO voltage. Similarly, voltage on Port B (Port 1) is controlled by GPIO14 and GPIO15. In the actual application, Buck regulators shall be replaced by ACDC and DCDC regulators as shown in Figure 2-13. TPS65988DH includes three I2C ports out of which one I2C port can work as both master and slave, one I2C port can work as a I2C master only and one can work as a I2C slave only. The I2C masters allow the PD controller to control various kinds of slaves directly based on events inside the PD controller. I2C master events can be used to configure ACDC/DCDC with I2C interface.

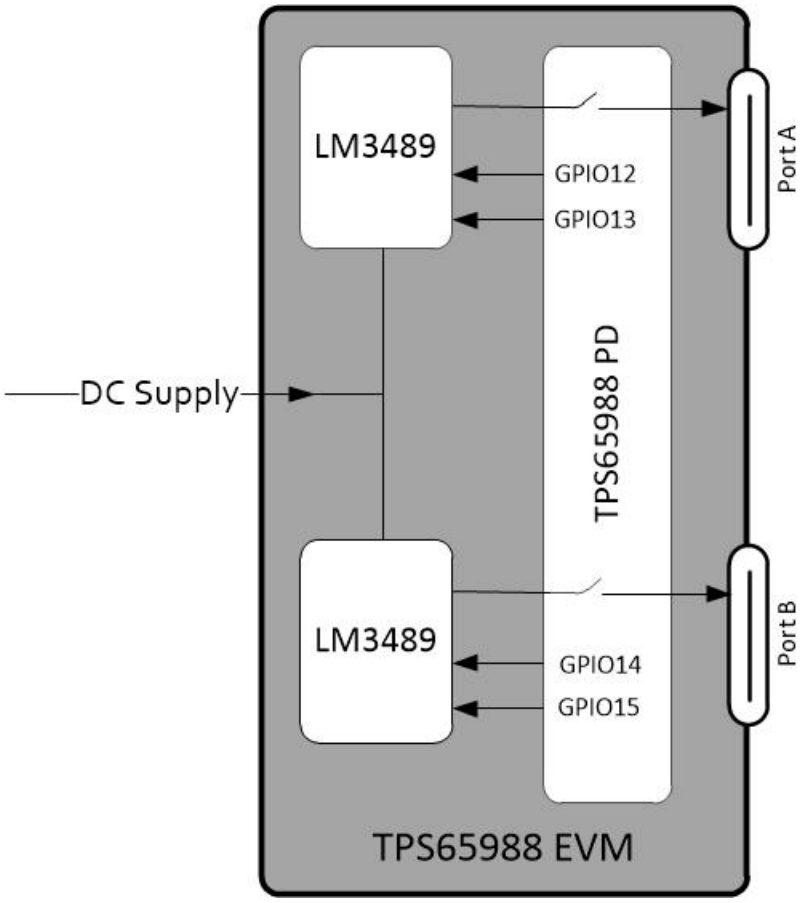

Figure 2-13. EVM Block Diagram

Following are the steps to prepare EVM and then test the port power balancing control logic.

1. Populate 0E resistors R170-R173 and R133 to make GPIO0-GPIO3 signals available on signal headers J3 and J4.
2. Short GPIO0 (on J4) to GPIO1 (on J3) and GPIO2 (on J4) to GPIO3 (on J4) using jumper wires.

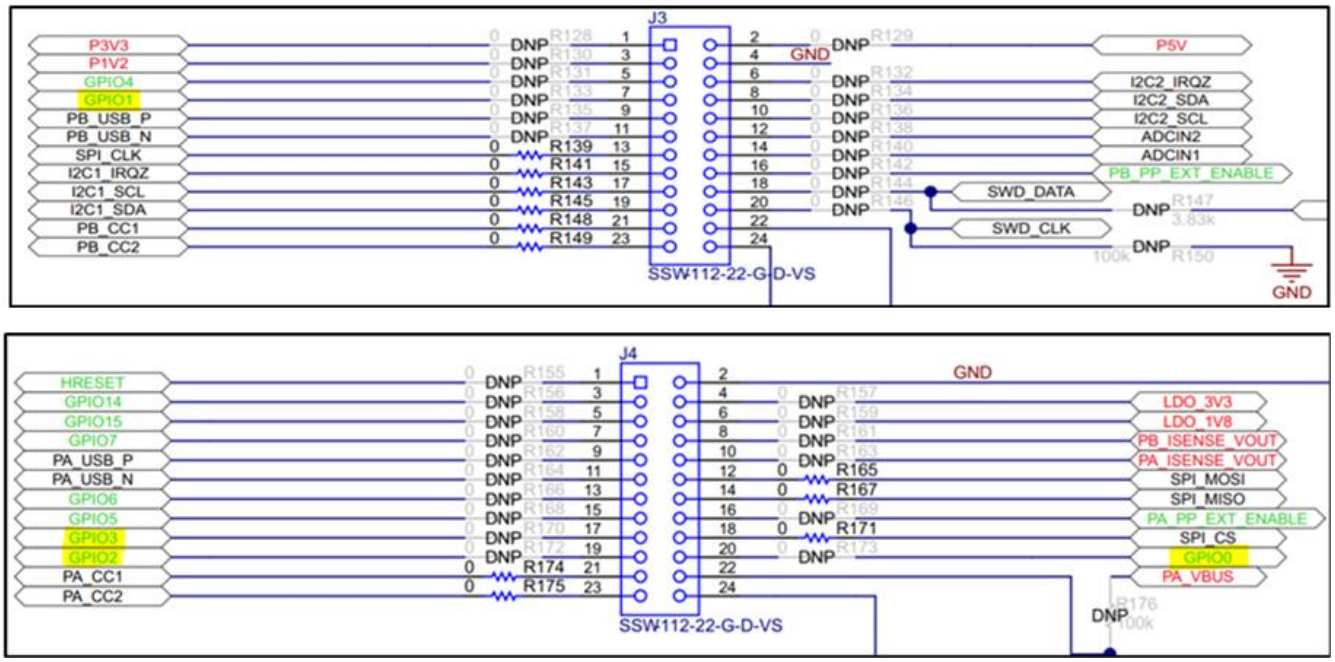

Figure 2-14. GPIO Setting on EVM

3. Populate PPHV jumpers to connect buck regulators' outputs to the system side of internal power switches. Short pins 5-7 and 6-8 of J11.
4. Power up the EVM by connecting the Power Adapter to barrel Jack connector J1.
5. Load the project file in GUI Software tool and load the firmware patch in PD controller.
6. Power cycle the EVM by unplugging and plugging the Power Adapter.
7. Connect a PD capable Sink device to one of the ports through a PD analyzer. The TPS665988DH Source port must advertise 60W capability; ensure the same by checking the Source Capability message.
8. Connect another PD device to the second port and verify that the already connected port advertises modified 30W Source Capability message.

Figure 2-15 shows the PD trace captured on Port 0. At t1 – only Port 0 is connected and therefore the source capability message shows 60W. At t2 – Port 1 is also connected and therefore Port 0 scaled down its power capability to 30W. The modified 30W source cap message transmitted from Port 0 at t2 is also show in the figure.

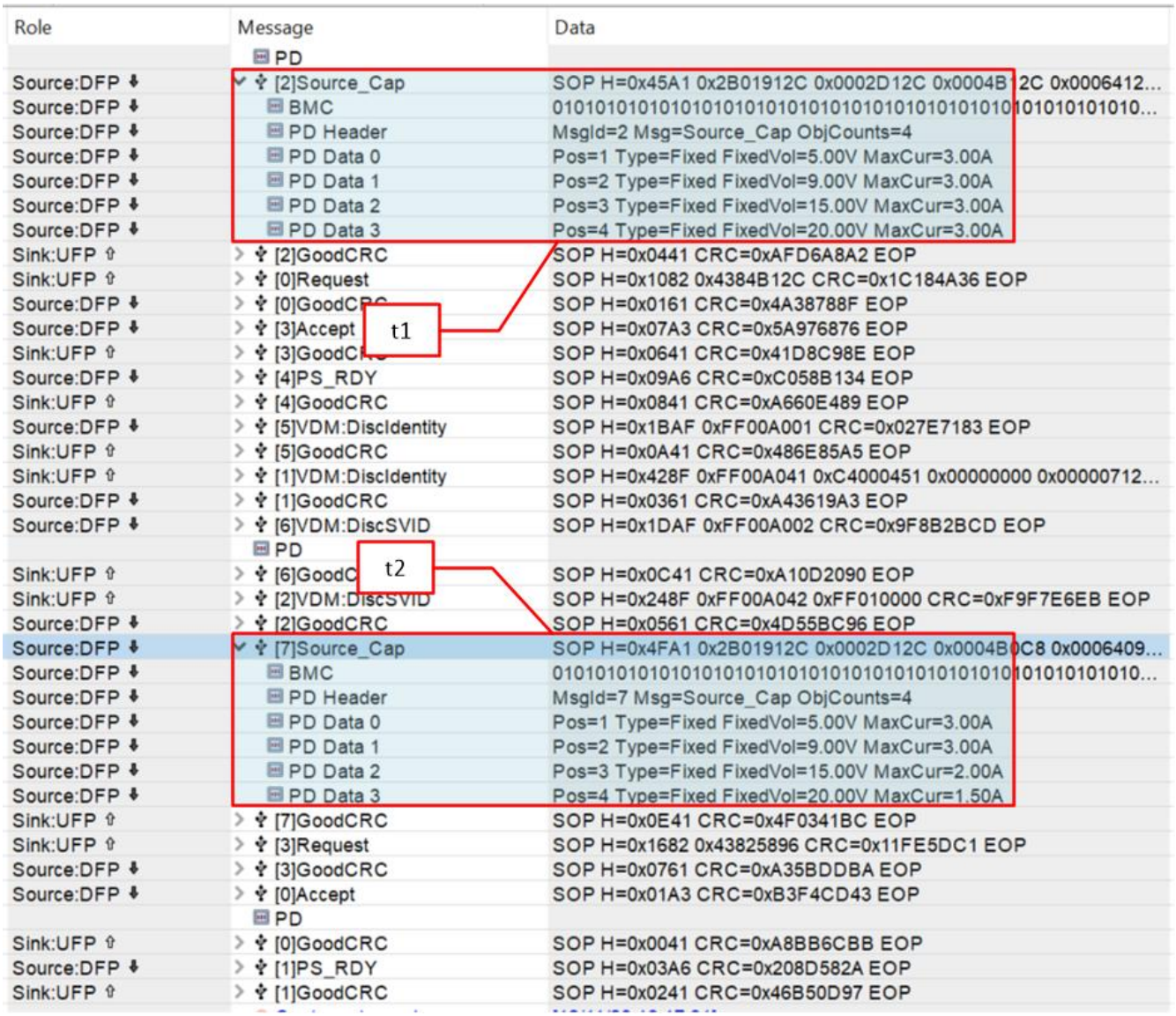

Figure 2-15. PD Trace Showing PDOs Corresponding to Single and Dual Port Connections

Figure 2-16 shows Port-0 (channel 1) and Port-1 (Channel 2) Vbus voltage transitions captured in parallel with the PD trace. Channel 2 shows Port-1 Vbus transition to 20 V.

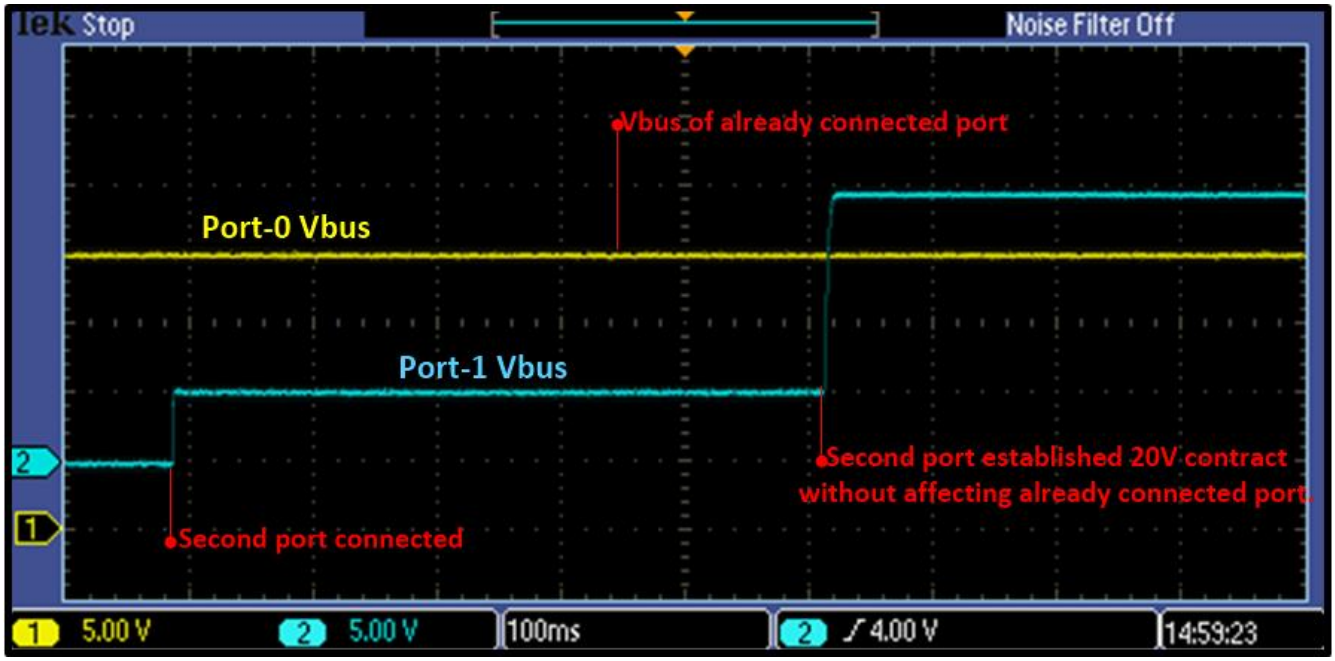

Figure 2-16. Scope Capture of Vbus Transitions

3 References

- Texas Instruments, TPS65987D GPIO Events application report.
- Texas Instruments, Using I2C Master in TPS65987D and TPS65988 PD Controllers application report.
- Texas Instruments, Dual Port USB Type-C™ and USB PD Controller with Integrated Power Switches data sheet.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265