

Automated Battery hot swap configuration

The “[AUTO Clear DBfg] TPS65988 EVM dual sink test (ADVANCED).pj” project file, is intended for use with the “windows_64bit_TPS65981_2_6_7_8_application_customization_6.1.3” release of the TI Application Customization GUI.

The above project file incorporates the following configurations:

1. PortA (J2A on the EVM/Port 1 in the GUI) advertises two sink Power Data Objects (PDOs):

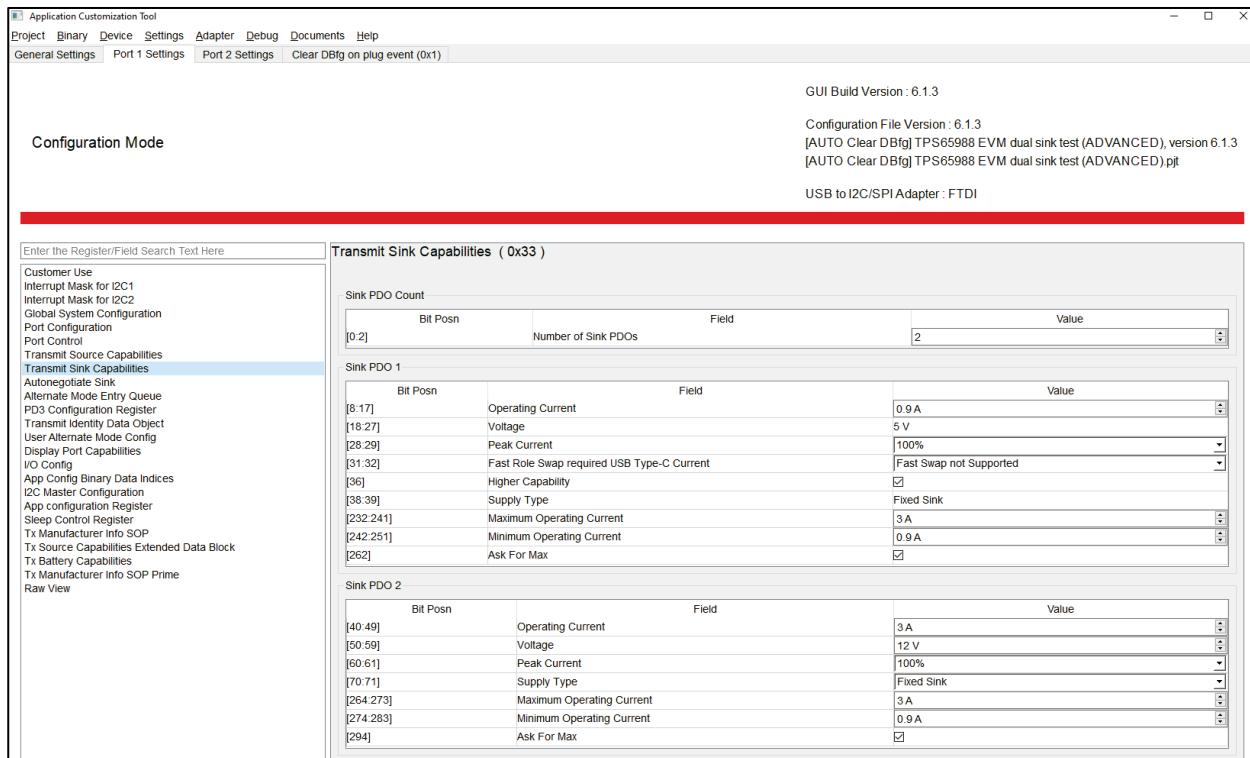


Figure 1: PortA Sink capabilities

*Note1: For USB-PD compliance, a 5V PDO must always be offered.

*Note2: For this Proof Of Concept (PoC), PortA will simulate the nofio battery pack.

2. PortB (J2B on the EVM/Port 2 in the GUI) advertises a “Variable Sink” PDO of 12-20V:

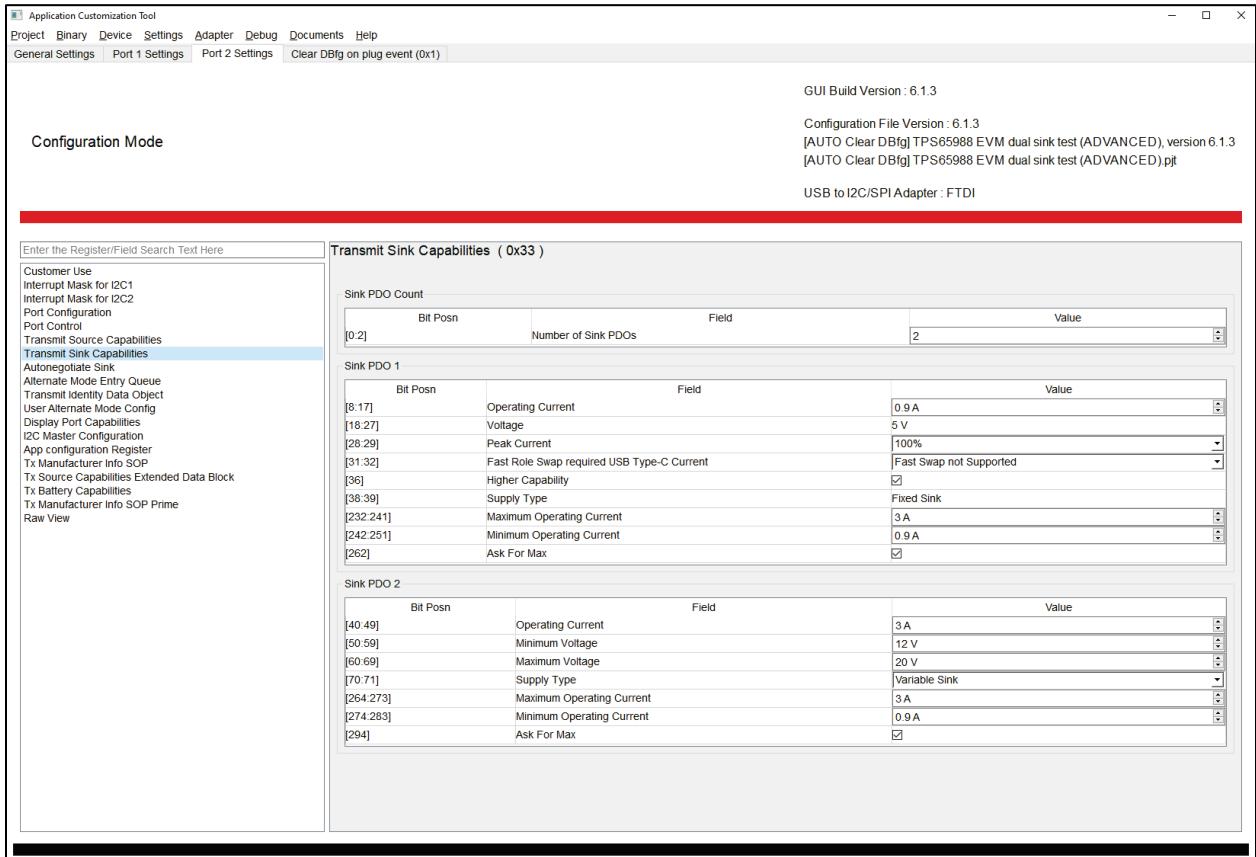


Figure 2:PortB Sink capabilities

*Note3: For this Proof Of Concept (PoC), PortB will simulate the user battery pack.

3. The TPS65988EVM device is configured as an Upward Facing Port (UFP) ONLY device and will reject all requests for role swapping; TPS65988EVM power-paths PP1/2 (“PP1/2 Switch Config” in the GUI), are configured as “**Sink Only**”. External power-paths PP3/4 are unused, and therefore disabled.

To afford the controller simultaneous negotiation of multiple PDOs, the “**Multipoint Sink Policy (0x27)_[96:97]**” register is set to close both PP1&2.

Reverse Current Protection (RCP) will handle which VBUS is currently supplying the SYSPWR rail on the EVM.

Global System Configuration (0x27)			
Bit Posn	Field	Value	
[0:1]	PP Cable 1 Switch Config	PP Cable Switch as Output, Guaranteed 4.5-5.5V	
[2:3]	PP Cable 2 Switch Config	PP Cable Switch as Output, Guaranteed 4.5-5.5V	
[16:18]	PP 1 Switch Config	PP Switch as Sink Only (Input)	
[19:21]	PP 2 Switch Config	PP Switch as Sink Only (Input)	
[32:34]	PP 3 Switch Config	PP Switch Disabled	
[35:37]	PP 4 Switch Config	PP Switch Disabled	
[55]	87 Emulation Mode	<input type="checkbox"/>	
[56]	PP1 Switch to VBUS Map	VBUS 1	
[57]	PP2 Switch to VBUS Map	VBUS 2	
[58]	PP3 Switch to VBUS Map	VBUS 1	
[59]	PP4 Switch to VBUS Map	VBUS 2	
[81]	I2C1 Enable as Master	<input type="checkbox"/>	
[82]	I2C3 Enable as Master	<input type="checkbox"/>	
[96:97]	Multipoint Sink Policy	Both sink paths will close switch	
[105:106]	Sink Policy Non-overlap Time	1 mSec	
[98:99]	Multipoint Alternate Mode Policy	Both ports can enter the same alternate mode	
[100:101]	External Processor	Default	
[102]	Ice Lake Enable One UFP Policy	<input type="checkbox"/>	
[103]	Ice Lake Enable Simple Source Policy Manager	<input type="checkbox"/>	
[107]	TBT Controller I2C Port	I2C1	
[108:110]	I2C Timeout	1 S	
[111]	SPI Read Only	<input type="checkbox"/>	

Figure 3: Port mapping & Role definition

4. For both PortA and PortB, the Over Voltage Protection (OVP) has been configured to disconnect the offending VBUS if a value of **≥24VDC** is sensed; For the event that a remaining connected battery has a lower voltage PDO which is attempting to close onto a capacitively charged SYSPWR of higher potential, setting the OVP in this way will prevent potential SYSPWR interruptions.

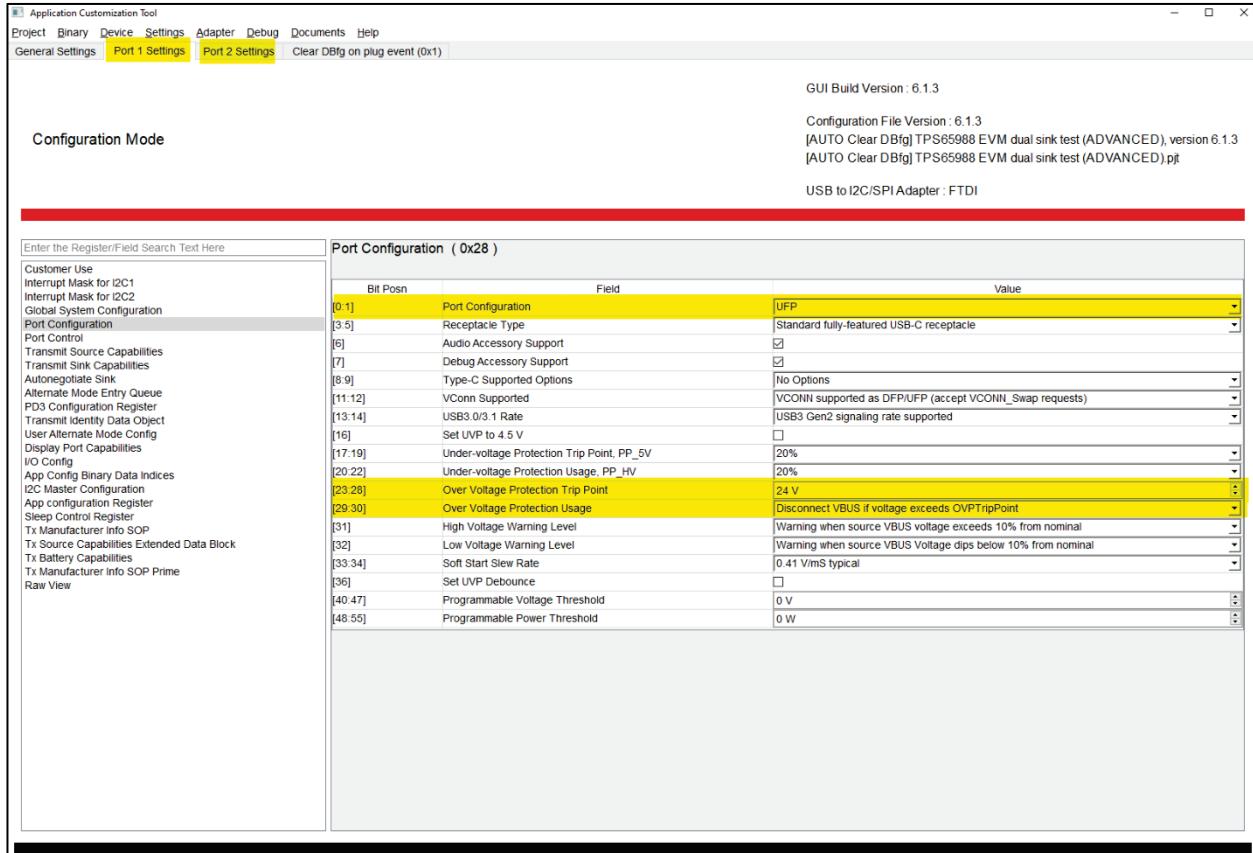


Figure 4: OVP Config

5. The TPS65988 USB-PD controller can be supplied from either an external 3V3 LDO, or from either VBUS-A/B. When the controller boots with VBUS supply only, the “**DBfg**” (Dead Battery Flag) system flag is set, indicating normal supply is unavailable. This carries several implications, one of which directly impedes the case for a battery hot-swap function.

- As an example.

- With no external supply present on SYSPWR, a battery is plugged into PortA, consequently the controller will be supplied from VBUS-A and set “**DBfg**” high.
- A second battery is then plugged into PortB. Now a second controller supply is available on VBUS-B (unused).
- If the battery in PortA is then removed, no mechanism exists for the TPS65988 controller to swap its supply from VBUS-A to VBUS-B without power cycling.
- **The solution to this problem (*within the context of this battery hot-swap functionality*) is to clear the “**DBfg**”, forcing the controller to swap source supplies to the 3V3 LDO.**

With RCP gating SYSPWR (which feeds the 3V3 LDO), if two batteries are plugged and one is removed, the RCP closes the remaining VBUS supply back onto SYSPWR and the 3V3 LDO is sustained.

To achieve this autonomously, the TPS65988 features event triggered interrupts which can be multiplexed to high/low output transitions of GPIO pins. These are then physically shorted to separate GPIO pin inputs, and tied (in software) to GUI “Configuration Sets”:

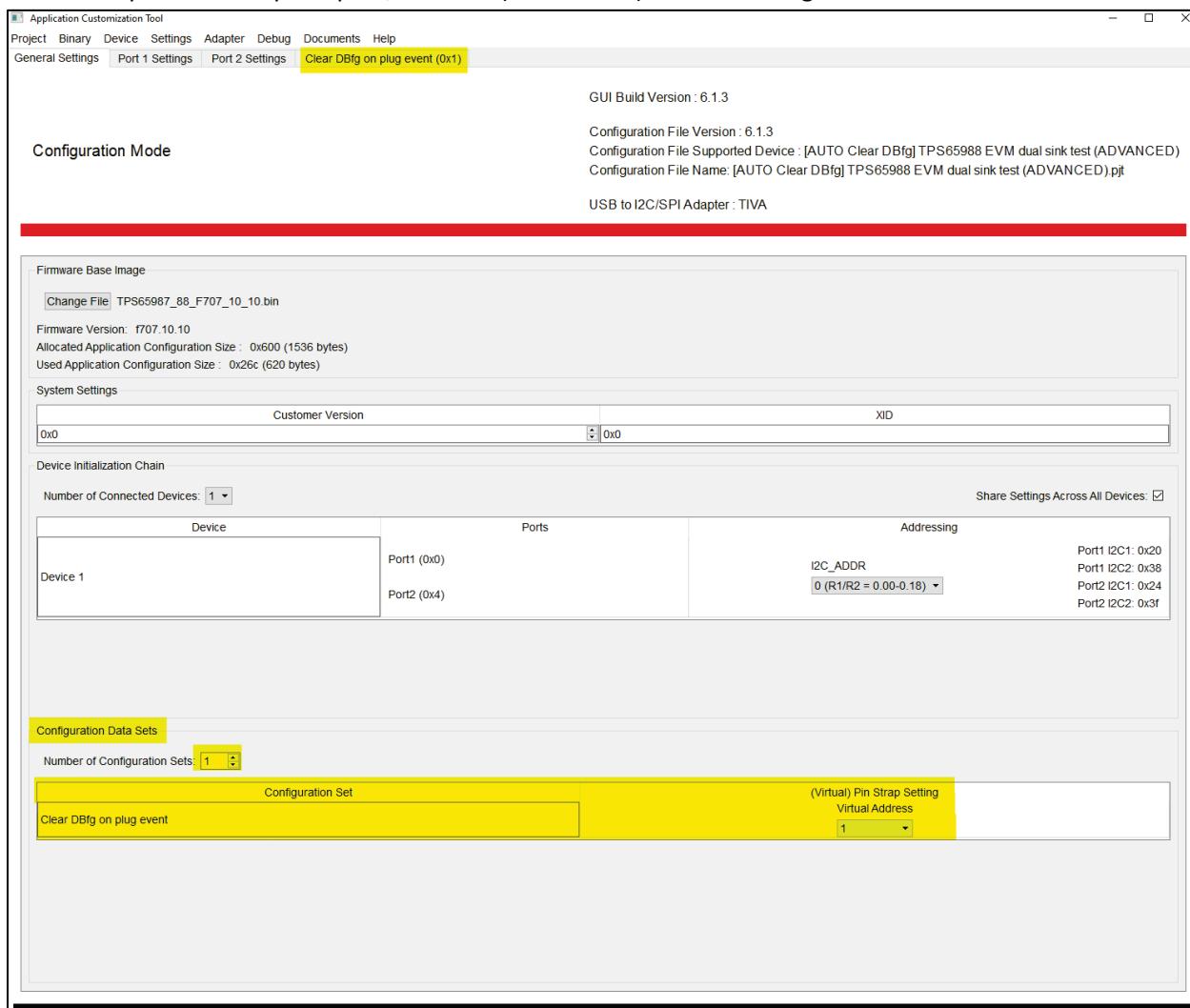


Figure 5: Instantiating Configuration Sets

The “Configuration Sets” can be used to load alternate device functionalities at run-time (such as different Sink/Source transmit capabilities), though is only used in this example as a means of clearing the “**DBfg**”:

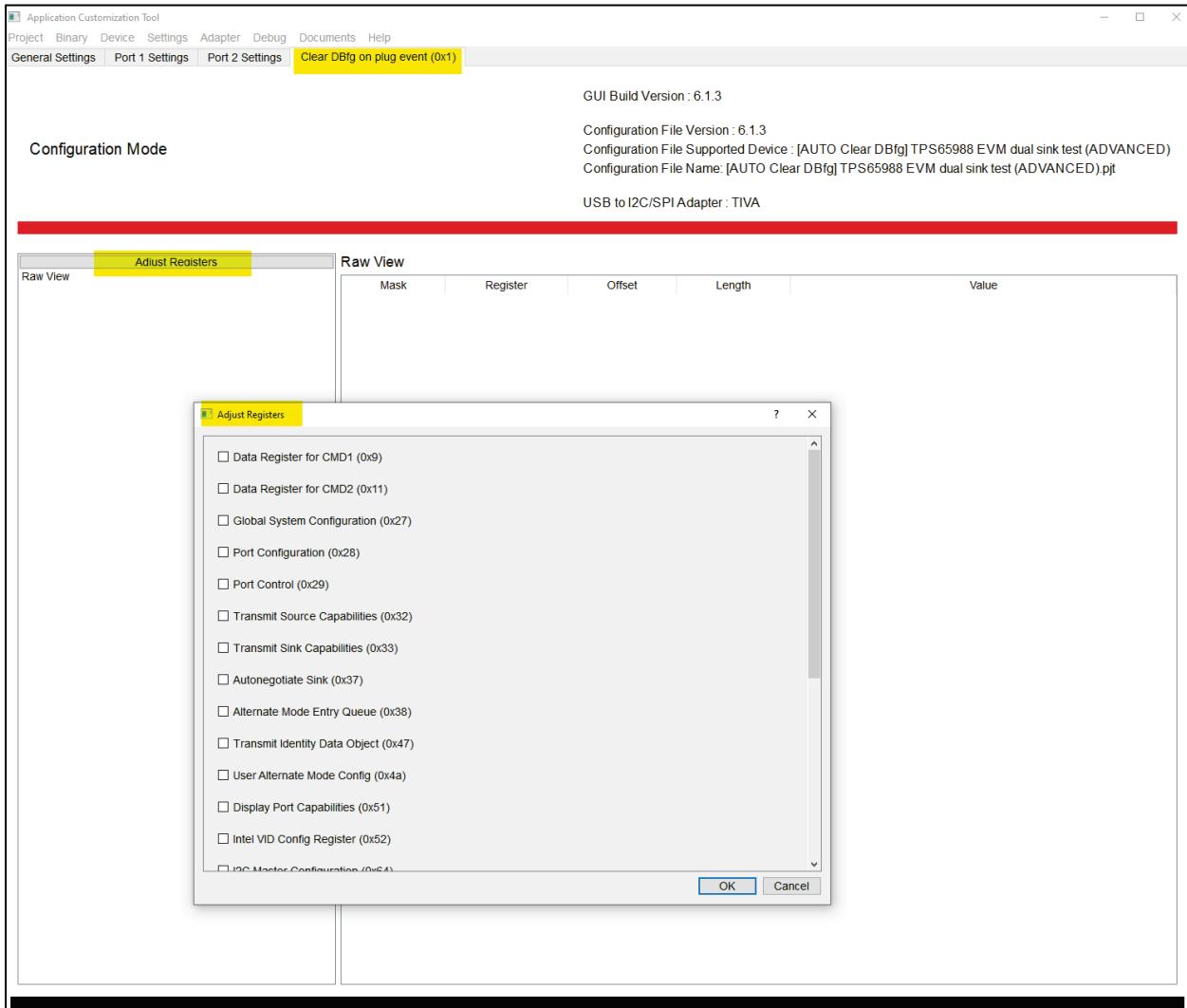


Figure 6: Configuration Set unused functionality

The “Port 0 Plug Event” and “Port 1 Plug Event”, have been mapped through the GUI, as the triggering events for a “DBfg” clear:

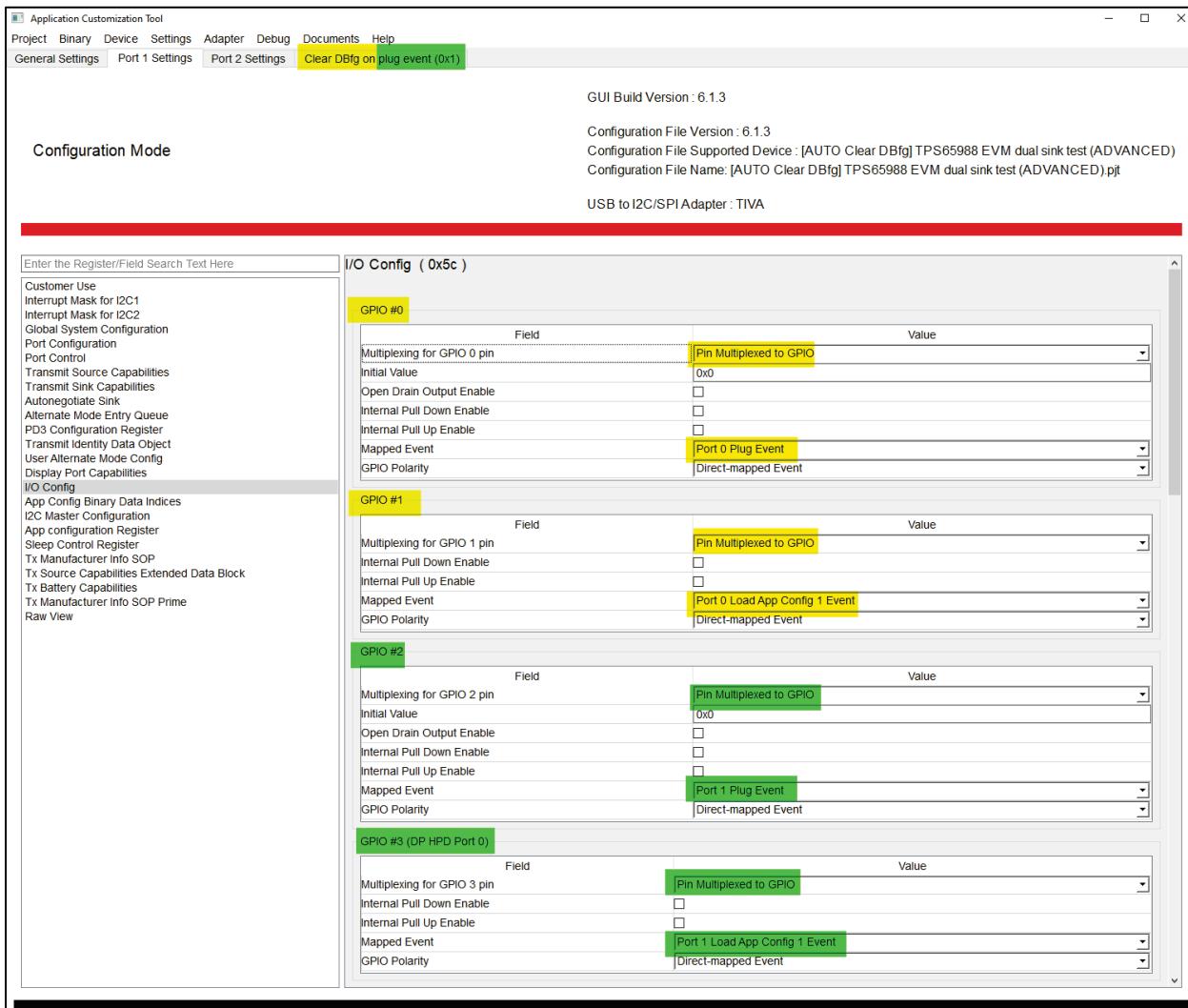


Figure 7: GPIO Mapping

As seen above in Figure 7, the input GPIO has been mapped to “App Config 1 Event”. The function of which, is defined (per port) in the “**App configuration Register**”:

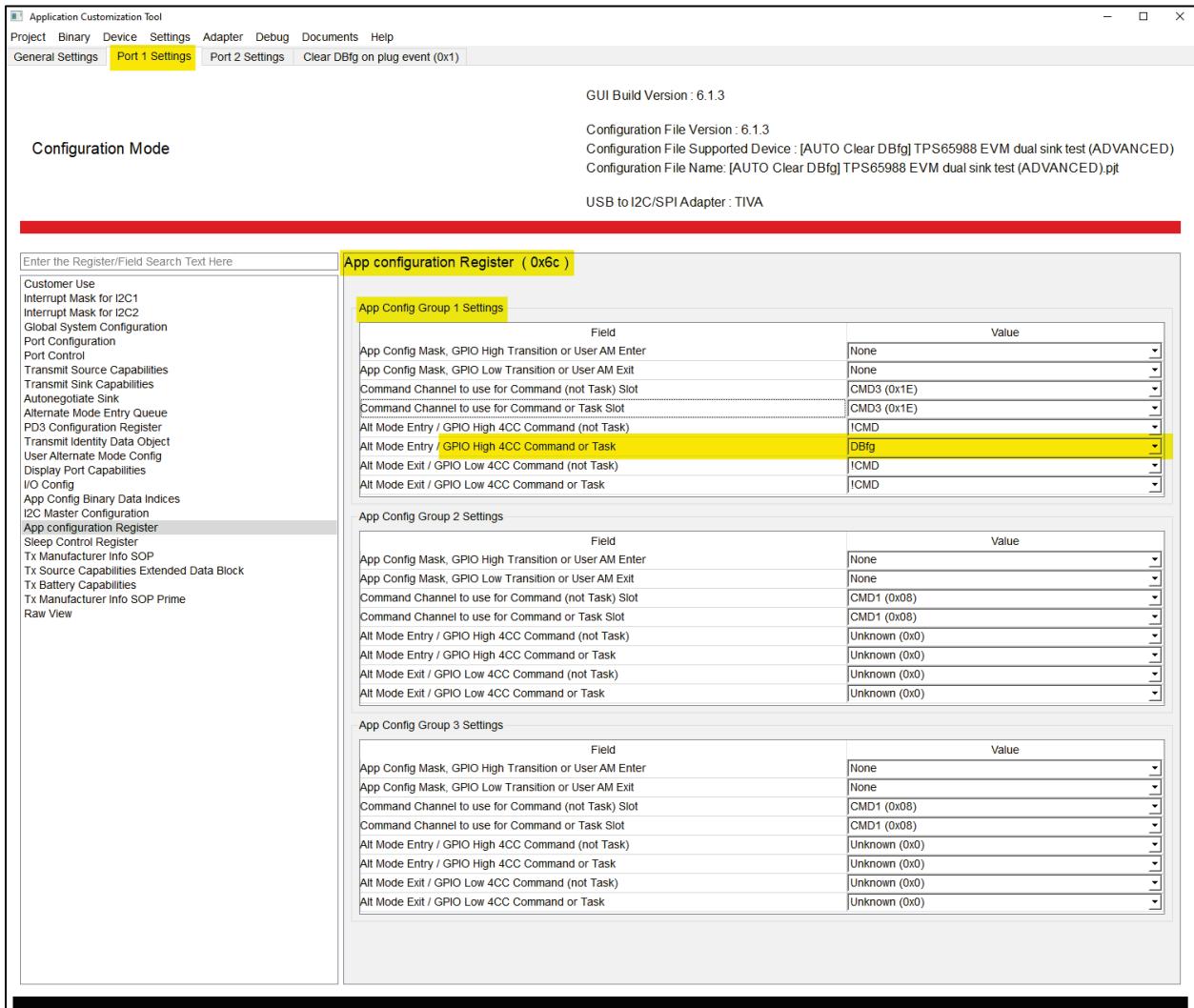


Figure 8: Defining the clearing of “DBfg” behaviour

As delineated in Figure 8, upon detection of a rising-edge the “**DBfg**” 4CC (Four Character Command) is executed. The other edge-detections are mapped as “**ICMD**” which signifies a “Do Nothing” case. The chosen “Command Channel” is of no significance. This configuration MUST be completed in the (0x6c) register of both “Port 1&2 Settings”.

6. Plug in the TPS65988EVM with the 19.5V DELL laptop PSU, and connect the EVM's FTDI chip to your PC via USB-Micro-C:

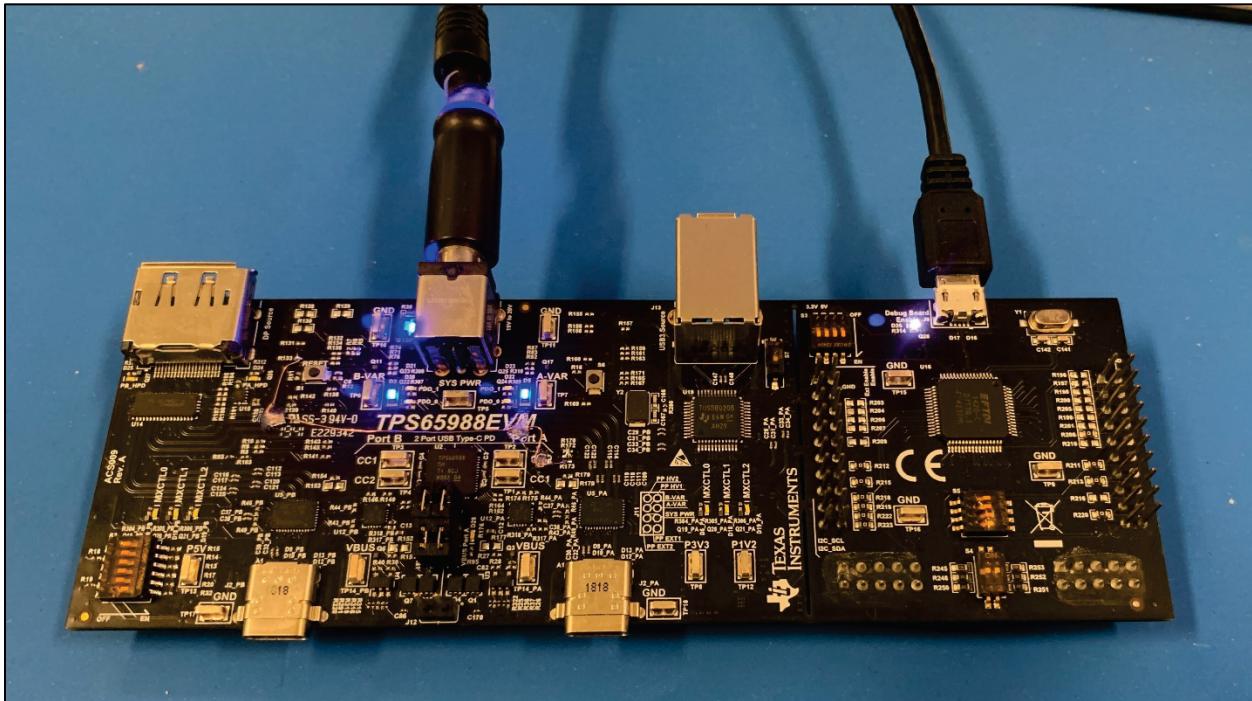


Figure 9: EVM connection

To flash the GUI config to the TPS65988EVM, the binaries can be flash directly from the open project to the device:

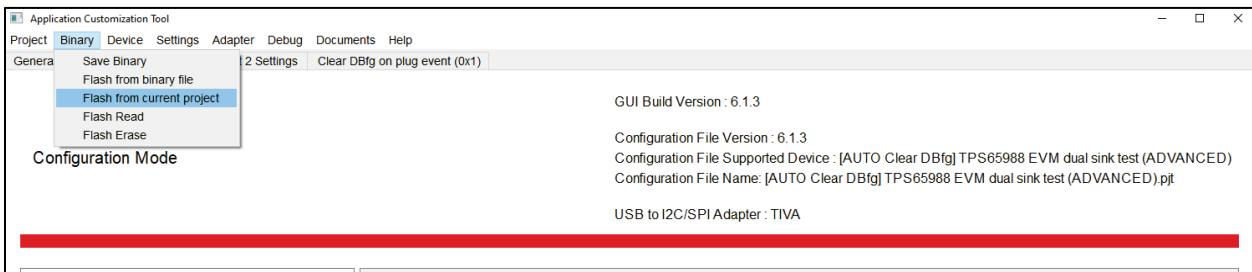


Figure 10: Flash from current project

Select “FTDI” as the SPI Adapter and click “Read Current Region Offsets” to automatically point to the correct device memory location, then click “OK”, and wait while the device is programmed:

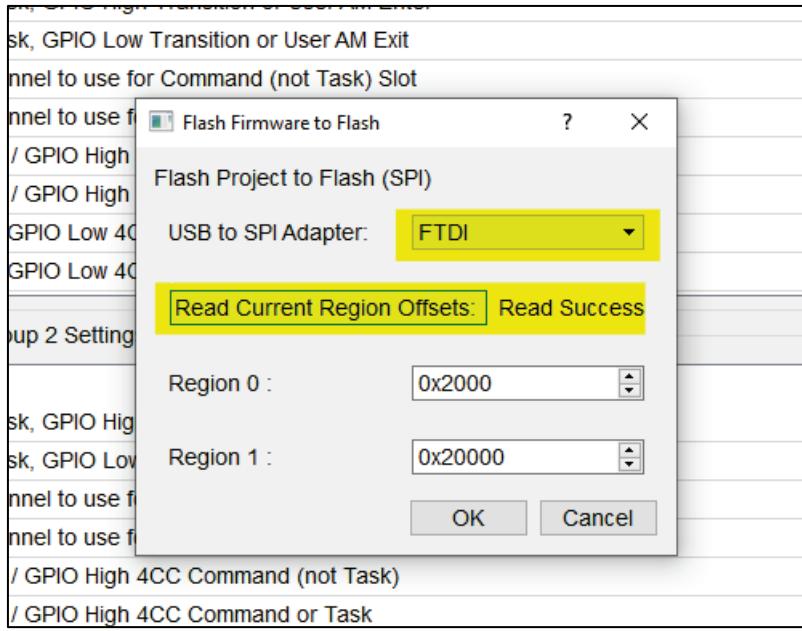


Figure 11: Flashing Current Project

*Note4: The TPS65988EVM requires a power-cycle after a successful flashing, to load the new configuration.

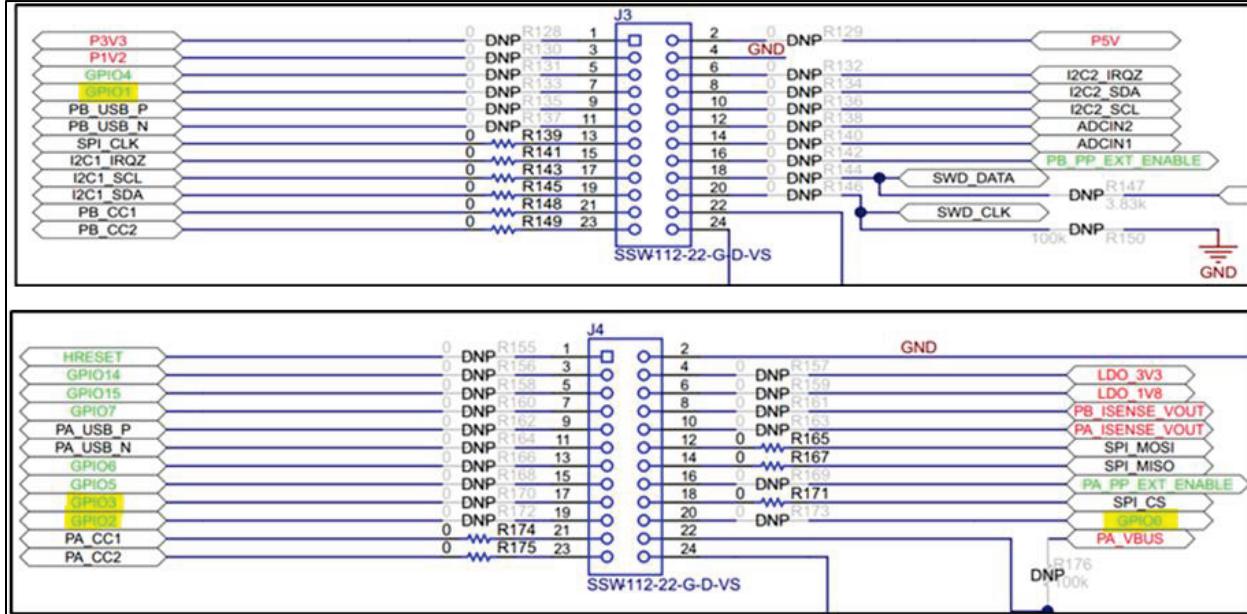


Figure 12: HEADER - GPIO Connection

Populate **R133/170/172/173** and connect GPIO 0->1 & 2->3 via **J3/4** as illustrated in Figure 12 above.

7. To check config functionality, enter “Debug Mode” in the GUI, and look for the “**Dead Battery Flag**” under the “Boot Flags (0x2d)” debug register:

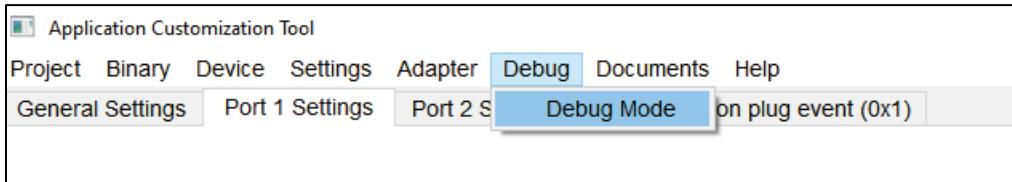


Figure 13: Debug Mode

The screenshot shows the Application Customization Tool interface with the Debug Registers tab selected. The title bar indicates "connected FTDI, 0x20 (I2C1)". On the left, a search bar says "Enter the Register/Field Search Text Here" and a list of register names. The "Boot Flags (0x2d)" section is highlighted with a yellow box. The table shows the following data:

Field	Value
Patch Header Error	False
Dead Battery Flag	False
SPI Flash Present	True
Region 0	True
Region 1	False
Region 0 Invalid	False
Region 1 Invalid	False
Region 0 Flash Error	False
Region 1 Flash Error	False
Patch Download Error	False
Region 0 CRC Fail	False
Region 1 CRC Fail	False
Customer OTP Invalid	False
PP1 Switch	False
PP2 Switch	False
PP3 Switch	False
PP4 Switch	False
Customer OTP Valid	Unknown (0x2)
OTP Customer Dead Battery	Bus Power From ADCIN
OTP I2C Address [6:5] for I2C2	0x2
OTP Customer TBT Present	False
OTP Customer I2C Threshold	True
OTP Customer TBT Controller Type	0x2
OTP Customer Disable PD	True
Revision ID Metal	0x0
Revision ID Base	TPS65988

Figure 14: Successfully Cleared "**DBfg**"

*Note5: The GUI's Debug menu quite frequently bugs out. If you see all flags marked as true, or other nonsensical values, power cycle the device until the FTDI chip starts behaving itself.

10. For ease of use, fly-wires have been soldered between GPIO pins 0&1, and between 2&3. Please acknowledge these connections before commencing future prototyping:

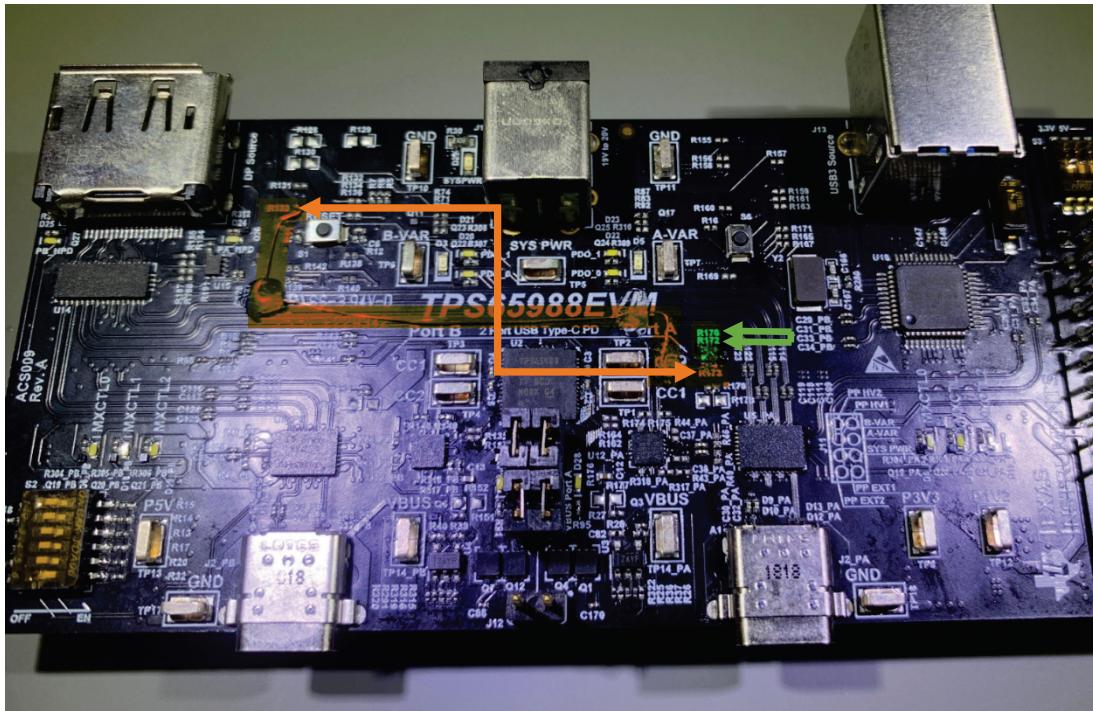


Figure 15: GPIO_0/1 -> R133&173 || GPIO_2/3 -> R170&172