Application Report
SLVA838–November 2016

GPIO Events

ABSTRACT

The Texas Instruments TPS6598x family of USB Type-C and USB PD controllers provides a set of GPIO events to achieve desired system behavior. A developer may program custom behavior triggered by GPIO to enable new functionality, and even load modified device configurations using GPIO events functionality. These firmware based GPIO events are simple to configure using the provided GUI software tools. The core TI PD controller firmware is unchanged when using GPIO events which ensures reliability, USB PD compliance, and also eases and speeds up development. This application report describes the procedure for configuring GPIO events on the TPS6598x family of devices and provides some concrete examples.

Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

The GPIO events feature of the TPS6598x family of devices allows users to tie specific events within the PD controller to trigger a signal in the system and also control the PD controller behavior by an external signal. These GPIO toggles in response to a defined PD or USB event can be used for customizing system behavior. The TPS6598x Configuration Tool is used to assign events to specific GPIO. The TPS6598x family of devices has a number of configurable GPIOs that can be used for this purpose and each GPIO behavior can be configured independently with such events depending on the system need.

The ability to configure independent GPIO events allows PD system designers to achieve variety of system behavior which helps TPS6598x users to implement unique applications and differentiate their end products with innovative system implementations. GPIO events are also available to trigger the loading of a modified device configuration setting in real-time based on the requirements of an application that require configuration change on-the-fly.

Unlike some PD controllers in the market that require firmware customization, Texas Instrument's PD controller can deliver the same custom behavior using the GPIO events feature keeping the core firmware same. This feature ensures that a fully tested and verified firmware can be used by all end users without having to modify PD controller internal firmware which helps speed up end product development cycle and ensures that overall system behavior is robust and reliable.

2 TPS6598x GPIO Event List

The firmware for the TPS6598x family of devices implements specific events that can be tied to GPIOs. These assigned events dictate the behavior of a system in response to a defined hardware or USB event. The TPS6598x Configuration Tool can be used to assign events to specific GPIOs. Table 1 lists all the GPIO events that are available in the TPS6598x family of devices and the event behavior.

Table 1. List of TPS6598x GPIO Events

Event Name	I/O	Active State	Behavior
ATTACHED_H (PLUG_EVENT)	Output	High	Asserted high when a Type-C electrical connection is made at either the CC1 or CC2 pin
			Asserted low when disconnected (opposite polarity of ATTACHED_L)
CC2_CONN (CABLE_ORIENTATION)	Output	High	Asserted high when an upside-down port connection is made (at the CC2 pin)
			Asserted low when port is disconnected or a right-side up port connection is made
PD_SOURCE_SINK_DISC (PROVIDER_CONSUMER_HIGH_Z)	Output	N/A (Tri-state)	Asserted high when USB PD contract negotiated as Source
			Asserted low when USB PD contract negotiated as Sink
			High-Z when port is disconnected or no PD contract is active (tri-state capable with equal value external pullup and pulldown resistors)
FAULT_CONDITION_L	Output	Low	Asserted low when an over-current fault condition occurs on any power path (PP_5V0, PP_HV, or PP_EXT) as a Source (USB Type-C or PD) or 5 V cannot be provided to VBUS on initial connection (short on contact)
			Asserted high during normal operation
DP_OR_USB3_H	Output	High	Asserted high when data connection is DisplayPort or USB3
			Asserted low if neither data mode is active or port is disconnected (opposite polarity of DP_OR_USB3_L)
DP_MODE_SELECTION	Output	High	Asserted high when data connection is DisplayPort (either 4-Lane mode or 2- Lane+USB3 mode)
			Asserted low when Type-C port is disconnected or DisplayPort mode is not active
SUPPLY_P5V	Output	High	Asserted high when PP_5V0 path is enabled
			Asserted low when PP_5V0 path is disabled (independent of other power paths)
SUPPLY_PHV	Output	High	Asserted high when PP_HV path is enabled
			Asserted low when PP_HV path is disabled (independent of other power paths)
SUPPLY_PHVE	Output	High	Asserted high when PP_EXT path is enabled
			Asserted low when PP_EXT path is disabled (independent of other power paths)
SUPPLY_PPCABLE	Output	High	Asserted high when PP_CABLE path is enabled and supplying VCONN to either CC1 or CC2, depending on connection orientation
			Asserted low when PP_CABLE path is disabled (independent of other power paths)
ATTACHED_L	Output	Low	Asserted l low when a Type-C electrical connection is made at either the CC1 or CC2 pin
			High when Type-C port is disconnected (opposite polarity of ATTACHED_H)
VBUS_DET	Output	High	Asserted high when voltage is present on VBUS and Power Status (USB Type-C or PD) is Sink
			Asserted low when port is disconnected and set low when connection is lost and VBUS approaches GND

Table 1. List of TPS6598x GPIO Events (continued)

Event Name	I/O	Active State	Behavior
P5V_OVERCURRENT	Output	Low	Asserted low when over-current fault condition occurs on PP_5V0 path as a Source (USB Type-C or PD)
			Asserted low during normal operation
PWR_SINK_SOURCE	Output	High	Asserted high when Power Status is Sink (USB Type-C or PD)
			Asserted low when Power Status is Source (Type-C or USB PD) or port disconnected
USB3_H	Output	Hi-Z	High-Z when data connection requires USB3 (fixed open-drain configuration, requires pullup resistor for High state to operate correctly)
			Asserted low when USB3 data is not required or supported (for example, 4-Lane DisplayPort mode entered or USB3 support de-activated by firmware configuration)
USB2	Output	High	Asserted high when data connection is USB2
			Asserted low when Type-C port is disconnected or USB2 data is not required or supported
DPx2_MODE	Output	High	Asserted high when 2-Lane DisplayPort and USB3 mode is supported and entered
			Asserted low when Type-C port is disconnected, DisplayPort mode is not entered, or 4-Lane DisplayPort mode is entered
PD_SINK_SOURCE (CONSUMER_PROVIDER)	Output	High	Asserted high when USB PD contract negotiated as Sink
			Asserted lLow when USB PD contract negotiated as Source, no PD contract is active, or port is disconnected (opposite polarity of PD_SOURCE_SINK_DISC but not tri-state capable)
AMSEL	Output	N/A (Tri-state)	Asserted high when 4-Lane DisplayPort mode
			Asserted low when 2-Lane DisplayPort and USB3 mode is supported and entered
			High-Z when Type-C port is disconnected or USB3 data is required without DisplayPort mode entry (tri-state capable with equal value pullup and pulldown resistors)
SINK_LESS_12V	Output	High	Asserted high when in an active PD contract and sinking less than 12 V;
			Asserted low when any other sink or source PD contract is active, no PD contract is active, or port is disconnected
SINK_12V	Output	High	Asserted high when in an active PD contract and sinking 12 V
			Asserted low when any other sink or source PD contract is active, no PD contract is active, or port is disconnected
SINK_MORE_12V	Output	High	Asserted high when in an active PD contract and sinking more than 12 V
			Asserted low when any other sink or source PD contract is active, no PD contract is active, or port is disconnected
USB3_L (HS_SEL0)	Output	High	Asserted low when data connection is USB3
			Asserted high when USB3 data is not required or supported (opposite polarity of USB3_H)
UFP_DFP	Output	High	Asserted high when data role is UFP or no connection at Type-C port
			Asserted low when data role is DFP
DP_OR_USB3_L (HS_N_EN)	Output	Low	Asserted low when data connection is DisplayPort or USB3
			Asserted high if neither data mode is active or port is disconnected (opposite polarity of DP_OR_USB3_H)
AC_DETECT	Input	High	When signal is asserted high, CONSUMER_NO_AC is asserted low (indicating AC Adapter is present and external power is available)
			If low when TPS65982 becomes a Sink (Type-C or PD), then CONSUMER_NO_AC is asserted high
CONSUMER_NO_AC	Output	High	Asserted high when AC_DETECT is low as TPS65982 becomes a Sink
			Asserted low when AC_DETECT is asserted high or when AC_DETECT is low and TPS65982 becomes a Source
CC1_CONN	Output	High	Asserted high when a right-side up port connection is made (at the CC1 pin)
			Asserted low when port is disconnected or upside-down port connection is made
BARREL_JACK_DET	Input	High	Upon Rising Edge (Barrel Jack detected): • Clear Dead Battery Flag • Set Externally Powered = 1 • Swap to Source.
			Upon Falling Edge (Barrel Jack removed): • Set Externally Powered = 0 • Swap to Sink
PDIO_IN0 PDIO_IN1 PDIO_IN2, PDIO_IN3	Input	N/A	Input GPIO event for PDIO Alternate Mode (when supported by both port partners and mode is entered). A change in state of PDIO_INx will trigger a PDIO Alternate Mode message to be sent to the port partner. PDIO_OUTx will reflect the value of this signal after the PDIO Alternate Mode message is received by the port partner. These events do not have a pre- determined active state
PDIO_OUT0, PDIO_OUT1 PDIO_OUT2 PDIO_OUT3	Output	N/A	Output GPIO event for PDIO alternate mode. When PDIO Alternate Mode is supported by both port partners and entered, output follows GPIO pin mapped to PDIO_INx event on port partner.

Table 1. List of TPS6598x GPIO Events (continued)

Event Name	I/O	Active State	Behavior
SOURCE_PDO0_NEGOTIATED SOURCE_PDO1_NEGOTIATED SOURCE_PDO2_NEGOTIATED SOURCE_PDO3_NEGOTIATED	Output	High	Asserted high when the corresponding Source PDO # (Power Delivery Object) becomes the active contract (after Accept PD message is sent but before PS_Ready PD message is sent) Asserted low when no PD contract is active or one of the other 3 Source PDO events is active (these 4 GPIOs are mutually exclusive and only 1 can be active at any time)
SOURCE_PDO_NEGOTIATED_TT_BIT0 SOURCE_PDO_NEGOTIATED_TT_BIT1 SOURCE_PDO_NEGOTIATED_TT_BIT2	Output	High	These 3 Events combine to form a 3-bit truth table to allow digital outputs indicating the active state of up to 7 PDOs. Bit 2 is the most-significant bit (MSB) and Bit 0 is the least significant bit (LSB)
VBUS_UVP_QUICK_DETECT	Output	High	Asserted high when TPS65982 is a Sink and VBUS rises above the UVP threshold of the active Type-C connection or PD contract Asserted low when port is disconnected and set low immediately after VBUS falls below UVP threshold of the active Type-C connection or PD contract
LOAD_APPCONFIG_SET_1 LOAD_APPCONFIG_SET_2 LOAD_APPCONFIG_SET_3	Input	—	Upon Rising Edge: • App Config Set for GPIO = High will be loaded as the active configuration • 1st 4CC Data and Command is written to selected CMDX register (optional) • 2nd 4CC Data and Command (or PD Task) is written to selected CMDX register (optional) Upon Falling Edge: • App Config Set for GPIO = Low will be loaded as the active configuration • 1st 4CC Data and Command is written to selected CMDX register (optional) • 2nd 4CC Data and Command (or PD Task) is written to selected CMDX register (optional)
USBEP_ENABLE_EVENT	Input	High	When signal is asserted high, the Host Interface will be exposed through the USB2.0 Low Speed Endpoint. The TPS65982 Endpoint (EP) driver can be used to debug or to perform a FW update from a USB Host connected to the port
SINK_HVEXT	Output	High	Asserted high when either the PP_HV or PP_EXT switch is enabled as the Sink path (Type-C or PD, after Soft Start is complete Asserted low when port is disconnected or any switch is enabled a Source (PP_5V0, PP_HV, or PP_EXT)
THERM_PROT_EXT_SW_IN	Input	—	Configurable polarity (active-high or active-low) When this signal transitions to the active state it indicates an over temperature event for the external PP_EXT switch path and immediately opens the switch to stop the flow of current while keeping the connection or contract active.

3 GPIO Events Register and Example Settings

The configuration registers are listed as follows:

- 0x5C, GPIO configuration 1
- 0x5D, GPIO configuration 2

The GPIO configuration registers of the TPS6598x family of devices allow event mapping to available GPIOs. Each GPIO output can be configured as open drain or push-pull, and use either LDO_3V3 or VDDIO as the supply. Internal pullup and pulldown resistors for each GPIO can also be configured using a configuration register.

NOTE: Some of the GPIOs that are preconfigured in the firmware for specific event cannot be changed using the TPS6598x Application Customization Tool.

3.1 GPIO Event Example Settings

The TPS6598x Application Customization Tool can be used to set different GPIO event capabilities. Using the *GPIO Event Map* page of the tool, any event can be assigned to a GPIO as shown in Figure 1.

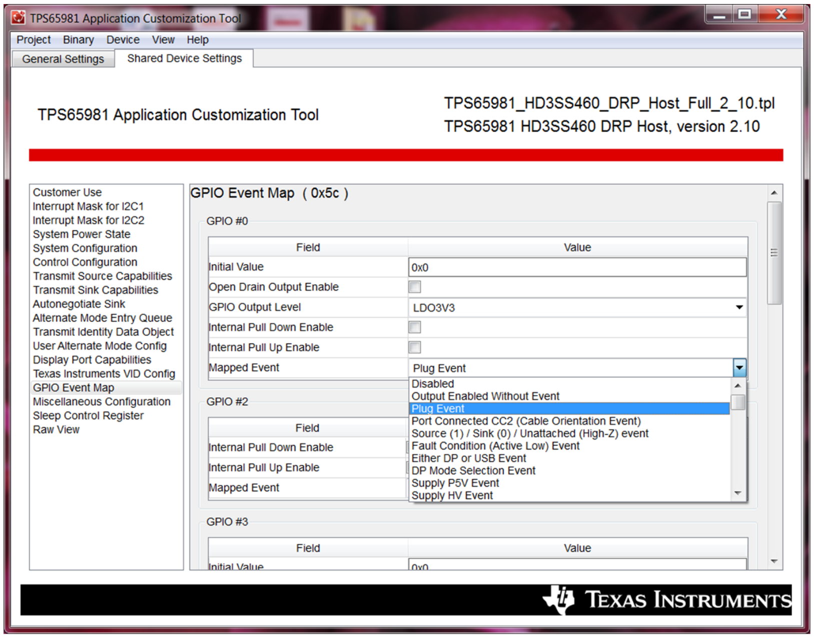

Figure 1. Mapping a GPIO Event Using Application Customization Tool

The TPS6598x Application Customization tool also contains example projects with different GPIO event capabilities already mapped depending on system need. The project template named *TPS65982_HD3SS460_DRP_Src_Full_2_10.tpl* demonstrates an example of how the GPIO Events are mapped for the TPS65981EVM. When the project template is loaded, all the relevant GPIO events that are configured can be seen from the *GPIO Event Map* page of the tool as shown in Figure 2.

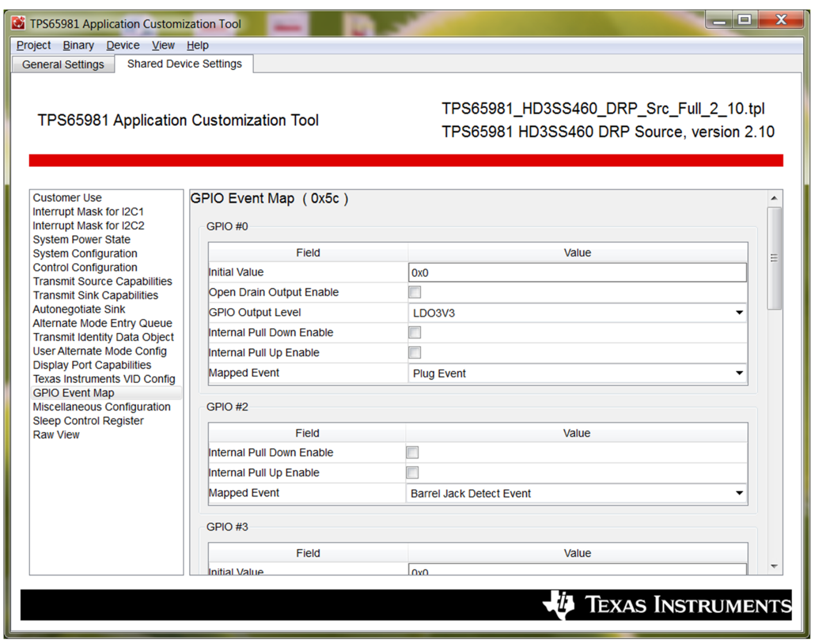

Figure 2. Template With GPIO Events Mapped for TPS65981EVM

3.2 App Config GPIO Event Settings

Advanced GPIO events can be used to load modified configurations to device at run-time. The TPS6598x Application Customization tool has a project template named *TPS65982_appConfigGpio_2_10.tpl* which demonstrates an example of how to use TPS6598x App Config GPIO feature. Figure 3 shows the *GPIO Event Map* page of the GUI which indicates that the *Load App Config Set 1* event has been mapped to GPIO #1 after loading this project template.

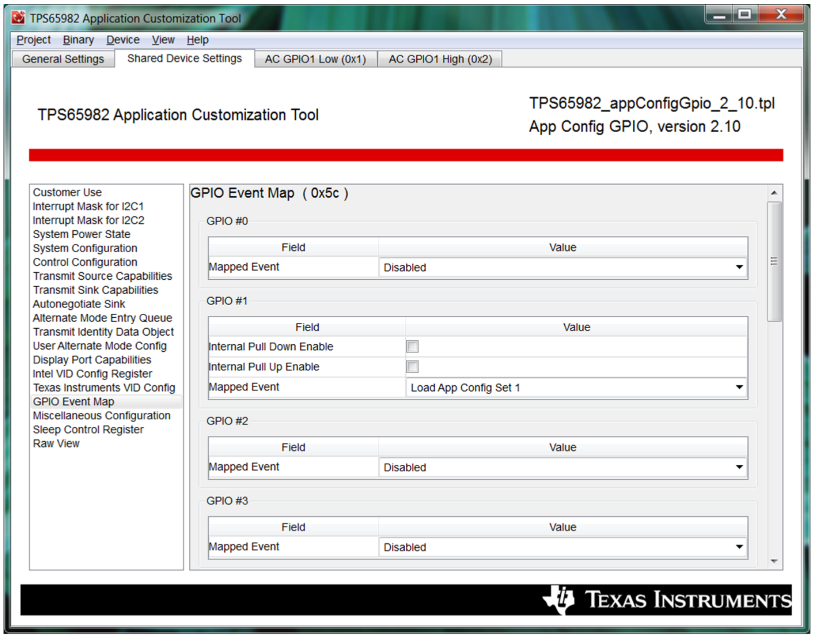

Figure 3. Mapping App Config Set to GPIO Event

In the set configuration for the App Config GPIO, an external hardware event can trigger the PD controller to change configurations. In this example template, setting GPIO#1 to a high-to-low transition would configure the Transmit Source Capabilities resister (0x32) with one PDO as shown in Figure 4.

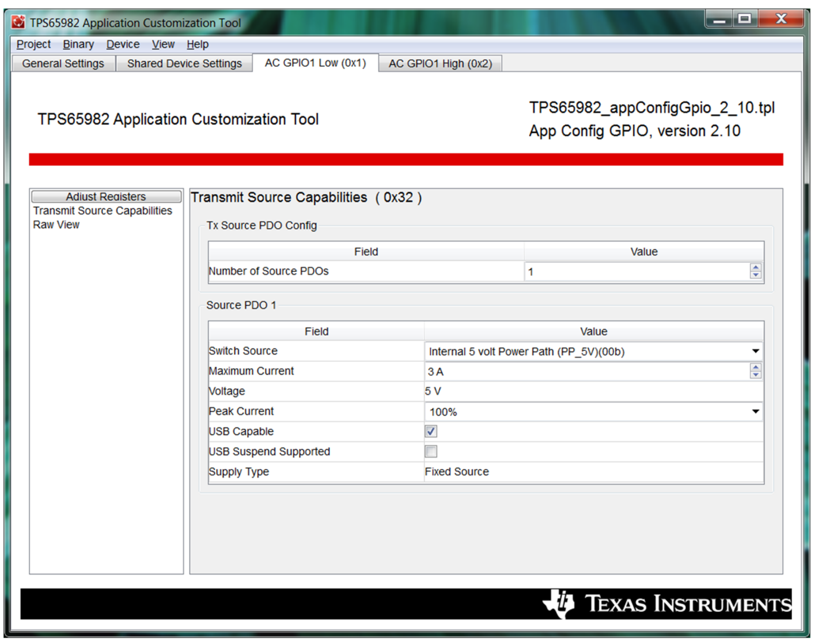

Figure 4. App Config GPIO Set Event, GPIO Low Settings Example

In the *TPS65982_appConfigGpio_2_10.tpl* template, the *AC GPIO1 High* settings, as shown in Figure 5, are applied to the Transmit Source Capabilities resister (0x32) of the device when a high-to-low transition on GPIO#1 occurs. This setting dynamically reconfigures the device to advertise 3 PDOs and changes the system behavior without any need for a custom firmware.

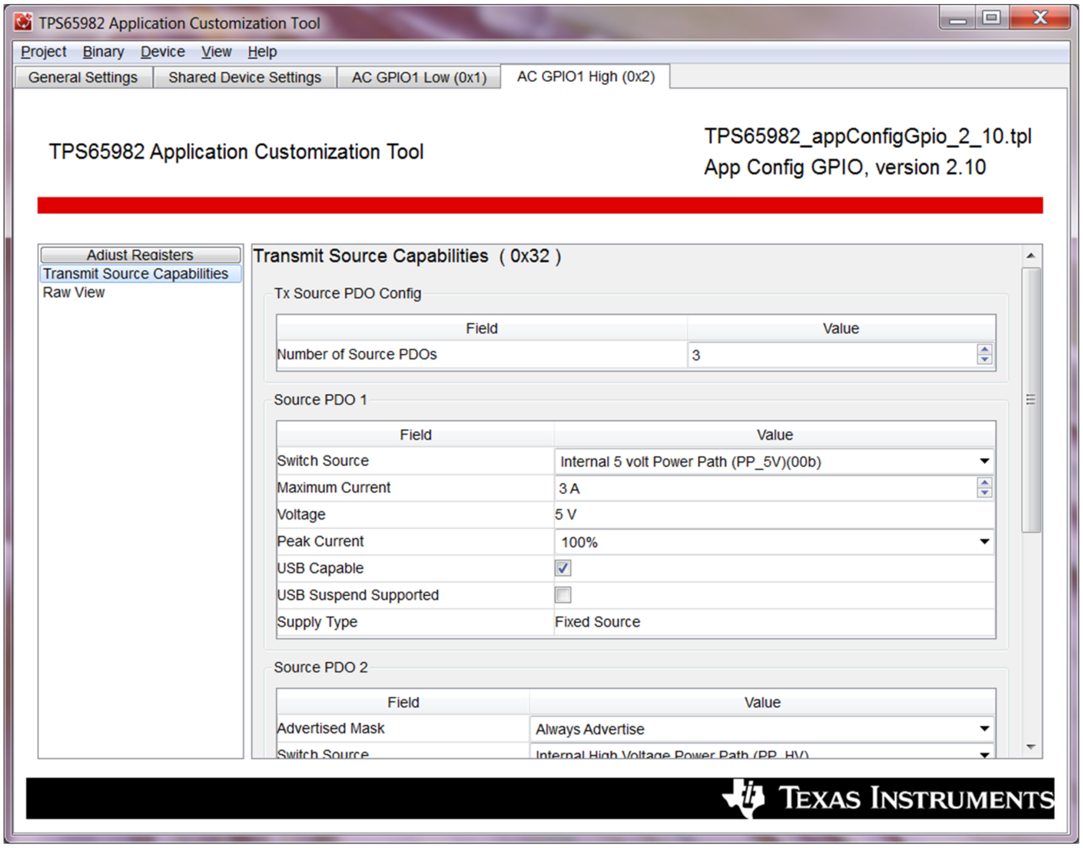

Figure 5. App Config GPIO Set Event, GPIO High Settings Example

4 PD Controller customization by GPIO Events

This section is targeted to show how the GPIO events of the TI PD controller can be used in a system to alter system behavior keeping the core firmware same. The Barrel Jack Event is used as example to show how a docking application can initiate a power-role (PR) swap when external power is connected to the system. Removal of the external power would generate PD traffic to reverse the PR swap and put the system back to original state.

4.1 Barrel Jack Connect Event PD Flow

Figure 6 shows the actual PD trace of this example for Barrel Jack Event implementation in a system. This event can be used in a docking application when external power becomes available to the docking station. The rising edge on the GPIO that has been assigned for the Barrel Jack Event to initiate the required PD message flow for PR swap.

The PD message trace shown in Figure 6 was taken with a Teledyne LeCroy PD analyzer between two TPS6598x EVMs, one loaded with a binary created from the example template *TPS65982_HD3SS460_DRP_Src_Full_2_8.tpl* which represents the settings of a docking station and would be referred as EVM-DCK from now on. The other EVM is loaded with example template *TPS65981_HD3SS460_DRP_Host_Full_2_10.tpl* which represents the configurations of a laptop, and would be referred as EVM-LPT.

Packet	Direction	Role		Msg Type	DR	PR	Msg ID	Obj Cnt	Duration	Idle	Time Stamp
1	Left "Left"	SOP SNK	PD Msg	PR Swap	UFP	SNK	7	0	494.978 us	83.022 us	7 . 835 619 000
2	Right "Right"	SRC SOP	PD Msg	GoodCRC	DFP	SRC	7	0	496.617 us	119.383 us	7 . 836 197 000
3	Right "Right"	SRC SOP	PD Msg	Accept	DFP	SRC	1	0	496.617 us	80.383 us	7 . 836 813 000
4	Left "Left"	SOP SNK	PD Msg	GoodCRC	UFP	SNK	1	0	496.617 us	30.274 ms	7 . 837 390 000
5	Left "Left"	SOP SNK	PD Msg	PS Ready	DFP	SNK	2	0	489.951 us	87.049 us	7 . 868 160 328
6	Left "Left"	SOP SNK	PD Msg	GoodCRC	UFP	SNK	2	0	489.951 us	1.561 ms	7 . 868 737 328
7	Right "Right"	SRC SOP	PD Msg	PS Ready	UFP	SRC	0	0	496.617 us	81.383 us	7 . 870 788 000
8	Left "Left"	SOP SNK	PD Msg	GoodCRC	DFP	SNK	0	0	494.978 us	4.238 ms	7 . 871 366 000

Figure 6. PD Trace of Barrel Jack Connect Event

The messages in Figure 6 represent PD traffic flow when the Barrel Jack adapter supplying 20 V is connected to the EVM-DCK configured with settings appropriate for a docking station.

1. Packet 1: EVM-DCK is UFP/SNK and sends *PR Swap* message to the EVM-LPT which is DFP/SRC.
2. Packet 2: DFP/SRC sends *GoodCRC* acknowledgement response for *PR Swap* message.
3. Packet 3: DFP/SRC sends *Accept* message to signal that it is willing to do a PR swap and has begun the PR Swap sequence.
4. Packet 4: UFP/SNK sends *GoodCRC* acknowledgement response.
5. Packet 5: EVM-LPT changes role to DFP/SNK and sends *PS Ready* message. Note that the initial source port is now setting the *Port Power Role* field to sink (SNK) in the *PS Ready* message indicating that the power supply of the initial source is turned off.
6. Packet 6: EVM-DCK sends *GoodCRC* acknowledgement response for *PS Ready* message. Note that the *GoodCRC* message sent by the initial sink in response to the *PS Ready* message from the initial source will have the Port Power Role field set to sink because this is initiated by the Protocol Layer.
7. Packet 7: EVM-DCK changes role to UFP/SRC and sends *PS Ready* message.
8. Packet 8: EVM-LPT which is now DFP/SNK sends *GoodCRC* acknowledgement response.

4.2 Barrel Jack Removal Event PD Flow

When power is removed from the EVM-DCK, the falling edge generated on the GPIO would initiate the reverse process so that EVM-LPT can become the power source again. Figure 7 shows the actual PD trace of the removal event.

Packet	Direction	Role		Msg Type	DR	PR	Msg ID	Obj Cnt	Duration	Idle	Time Stamp
1	Right "Right"	SRC SOP	PD Msg	PR Swap	UFP	SRC	3	0	496.617 us	80.383 us	5 . 327 193 000
2	Left "Left"	SOP SNK	PD Msg	GoodCRC	DFP	SNK	3	0	496.617 us	120.383 us	5 . 327 770 000
3	Left "Left"	SOP SNK	PD Msg	Accept	DFP	SNK	1	0	496.617 us	81.383 us	5 . 328 387 000
4	Right "Right"	SRC SOP	PD Msg	GoodCRC	UFP	SRC	1	0	496.617 us	30.258 ms	5 . 328 965 000
5	Left "Left"	SOP SNK	PD Msg	PS Ready	UFP	SNK	4	0	489.951 us	88.033 us	5 . 359 719 328
6	Left "Left"	SOP SNK	PD Msg	GoodCRC	DFP	SNK	4	0	488.334 us	1.587 ms	5 . 360 297 312
7	Right "Right"	SRC SOP	PD Msg	PS Ready	DFP	SRC	2	0	496.617 us	80.383 us	5 . 362 373 000
8	Left "Left"	SOP SNK	PD Msg	GoodCRC	UFP	SNK	2	0	496.617 us	24.676 ms	5 . 362 950 000

Figure 7. PD Trace of Barrel Jack Removal Event

5 Status Register and 4CC Commands

The GPIO status can be monitored by reading a register and system controller can take appropriate actions based on that. GPIO-related 4CC commands are also available that can be used by system controller to alter GPIO behavior.

- Status register:
  - 0x72, GPIO Status
- 4CC commands:
  - GPie, GPIO Input Enable
  - GPoe, GPIO Output Enable
  - GPsh, GPIO Set Output High
  - GPsl, GPIO Set Output Low

The Status register and 4CC command capabilities of TPS6598X Host Interface Utility Tool provides a way to test and modify GPIO configurations of a real system. Using the TPS6598X Host Interface Utility Tool, GPIO configurations can be changed on-the-fly over an I²C bus to try new settings quickly. When the expected system behavior is confirmed, appropriate GPIO configurations can be implemented through the system controller processor.

5.1 GPIO Status Monitoring

The GPIO status register can be used to monitor various GPIOs that are configured to achieve desired system behavior. For example, to support PD power rules with 5-V, 9-V, 15-V, and 20-V variable supplies, the TPS65981EVM is designed to use PDO GPIO events that trigger the power supply circuit and generate the desired voltage output. In this case GPIO7 and GPIO8 are assigned with appropriate PDO events to achieve the variable DC-DC supply. Figure 8 shows that both GPIO7 and GPIO8 are set low indicating that the PD contract is done for 5 V. When an explicit PD contract is negotiated for a 20-V supply, both GPIO7 and GPIO8 are driven high by the PD controller as indicated in Figure 9.

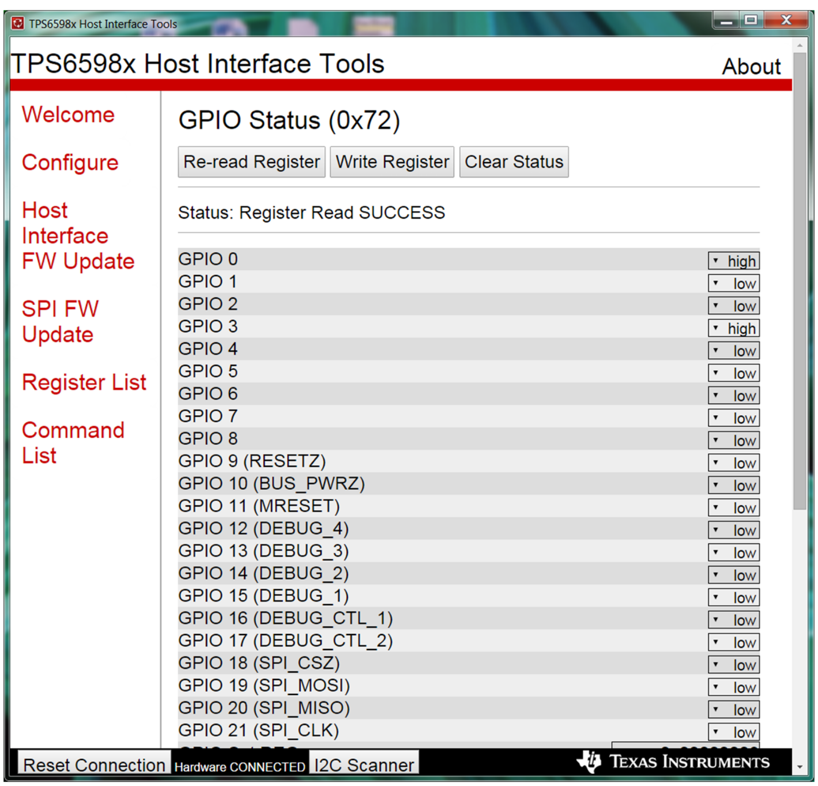

Figure 8. Variable DC-DC GPIO Status for 5-V Supply

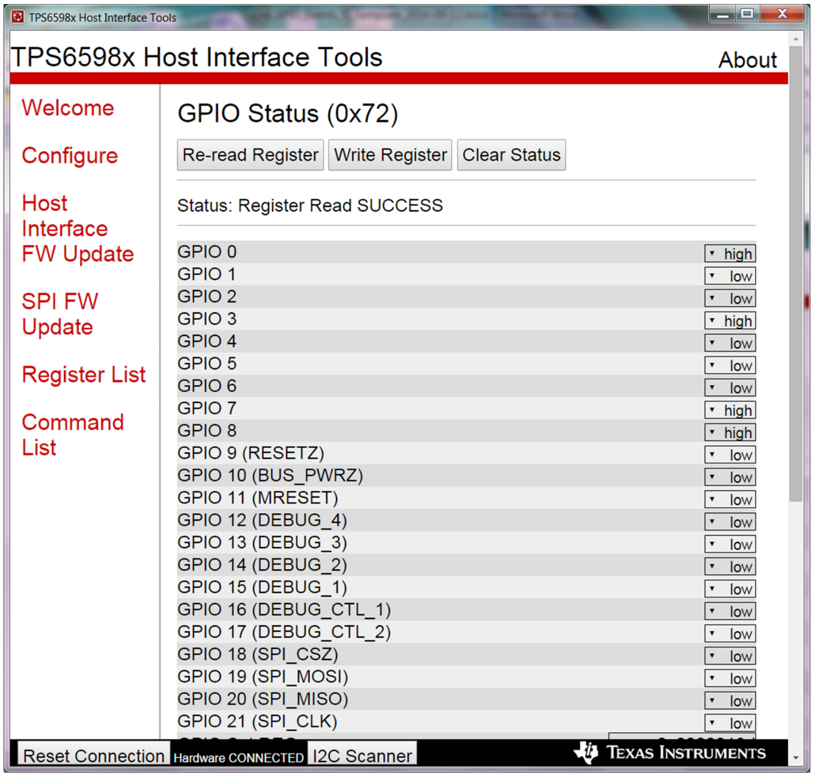

Figure 9. Variable DC-DC GPIO Status for 20-V Supply

5.2 *Using 4CC GPIO Commands*

The TPS6598x Host Interface Utility tool can be used to exercise the GPIO related 4CC commands, and observe and develop system behavior before system controller implements the desired driver software. Figure 10 shows the commands list page of the tool that can be used to exercise the GPxx 4CC commands.

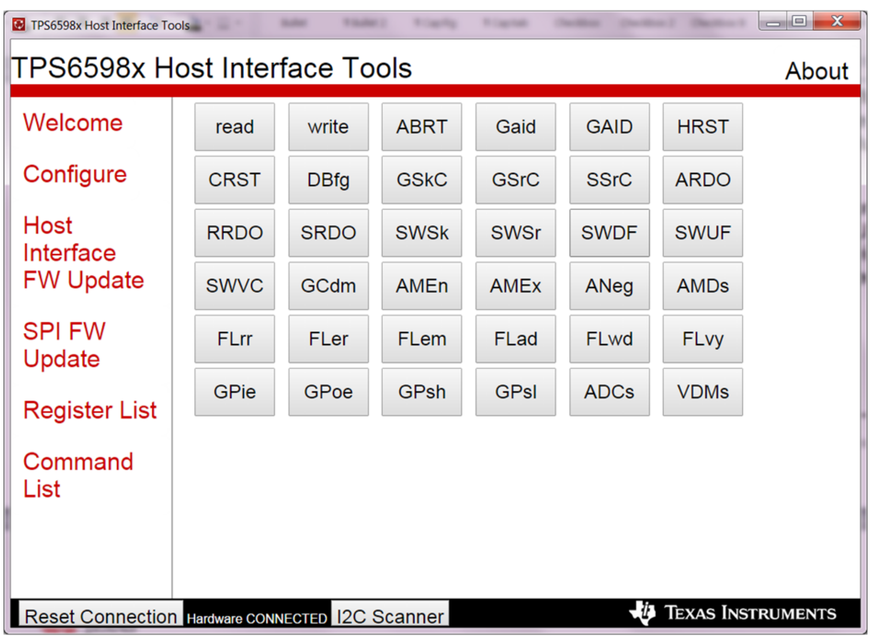

Figure 10. 4CC Commands in Utilities Tool

For example, to set the GPIO7 to high, use the steps that follow:

Step 1. Send GPoe 4CC command as shown in Figure 11.

Step 2. Send GPsh 4CC command as shown in Figure 12.

In the GPIO Status (0x72), GPIO7 is now set to high.

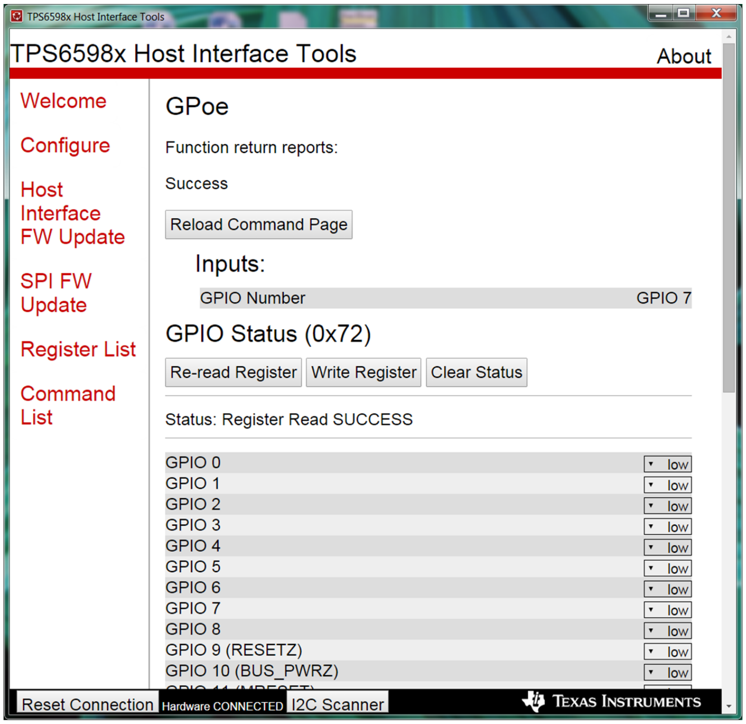

Figure 11. Using GPoe 4CC Command

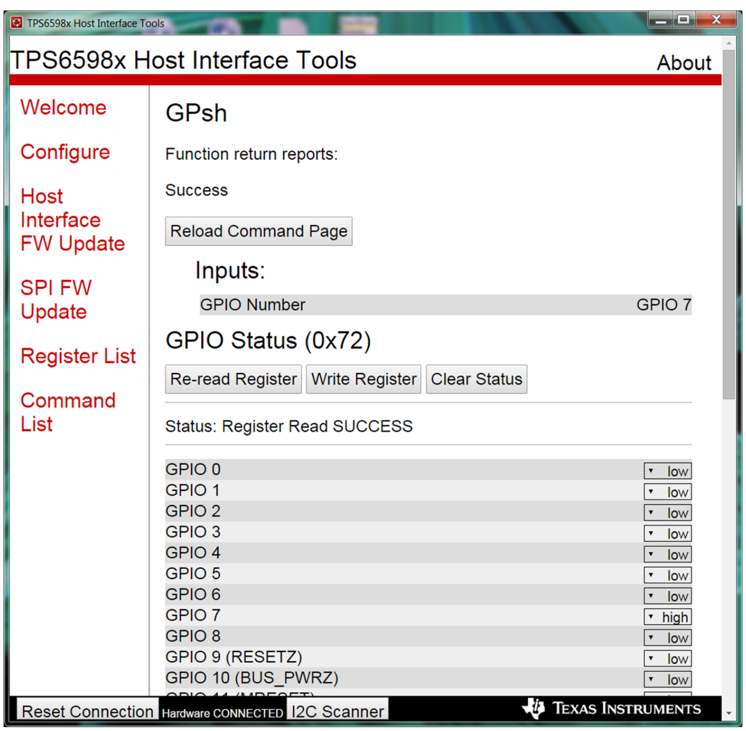

Figure 12. Using GPsh 4CC Command