Configuration and Verification of TPS6598X-based USB PD Systems

GPIO Events

Contents

Table of Figures

Table of Tables

1 Introduction

GPIO Events feature of TPS6598x allow 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. TPS6598x Configuration Tool is used to assign events to specific GPIO. TPS6598x device has 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. This helps TPS6598x users to implement unique applications and differentiate their end products with innovative system implementations. There are also GPIO events available that can trigger loading a modified device configuration settings 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, TI PD controller can deliver the same custom behavior using GPIO events feature keeping the core firmware same. This ensures that a fully tested and verified firmware can be used by all end users without having to modify PD controller internal firmware. This helps speed up end product development cycle and ensures that overall system behavior is robust and reliable.

Below are the topics that would be covered in this chapter:

(1) Available GPIO events in TPS6598x

(2) Setting up and modifying examples of GPIO events capabilities

(3) Verifying correct behavior of event that is configured

(4) Analyzing the results of the PD trace data

(5) Modifying GPIO events using the Host Interface

2 TPS6598x GPIO Event List

TPS6598x firmware 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 Customization Tool can be used to assign events to specific GPIOs. Table 1 lists all the GPIO events that are available in TPS6598x and their behavior.

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; ● 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); ● 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; ● 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); ● High during normal operation
DP_OR_USB3_H	Output	High	● Asserted high when data connection is DisplayPort or USB3; ● 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); ● 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; ● Low when PP_5V0 path is disabled (independent of other power paths)
SUPPLY_PHV	Output	High	● Asserted high when PP_HV path is enabled; ● Low when PP_HV path is disabled (independent of other power paths)
SUPPLY_PHVE	Output	High	● Asserted high when PP_EXT path is enabled; ● 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; ● Low when PP_CABLE path is disabled (independent of other power paths)
ATTACHED_L	Output	Low	● Asserted 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; ● Low when port is disconnected and set low when connection is lost and VBUS approaches GND
P5V_OVERCURRENT	Output	Low	● Asserted low when over-current fault condition occurs on PP_5V0 path as a Source (USB Type-C or PD); ● Low during normal operation
PWR_SINK_SOURCE	Output	High	● Asserted high when Power Status is Sink (USB Type-C or PD); ● 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); ● 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; ● 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; ● 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; ● Low 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)	● High when 4-Lane DisplayPort mode; ● 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; ● 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; ● 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; ● 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; ● 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; ● Low when data role is DFP
DP_OR_USB3_L (HS_N_EN)	Output	Low	● Asserted low when data connection is DisplayPort or USB3; ● 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; ● 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); ● 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.
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.
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); ● 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; ● 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; ● 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

Table 1: List of TPS6598x GPIO Events

3 GPIO Events Register and Example settings

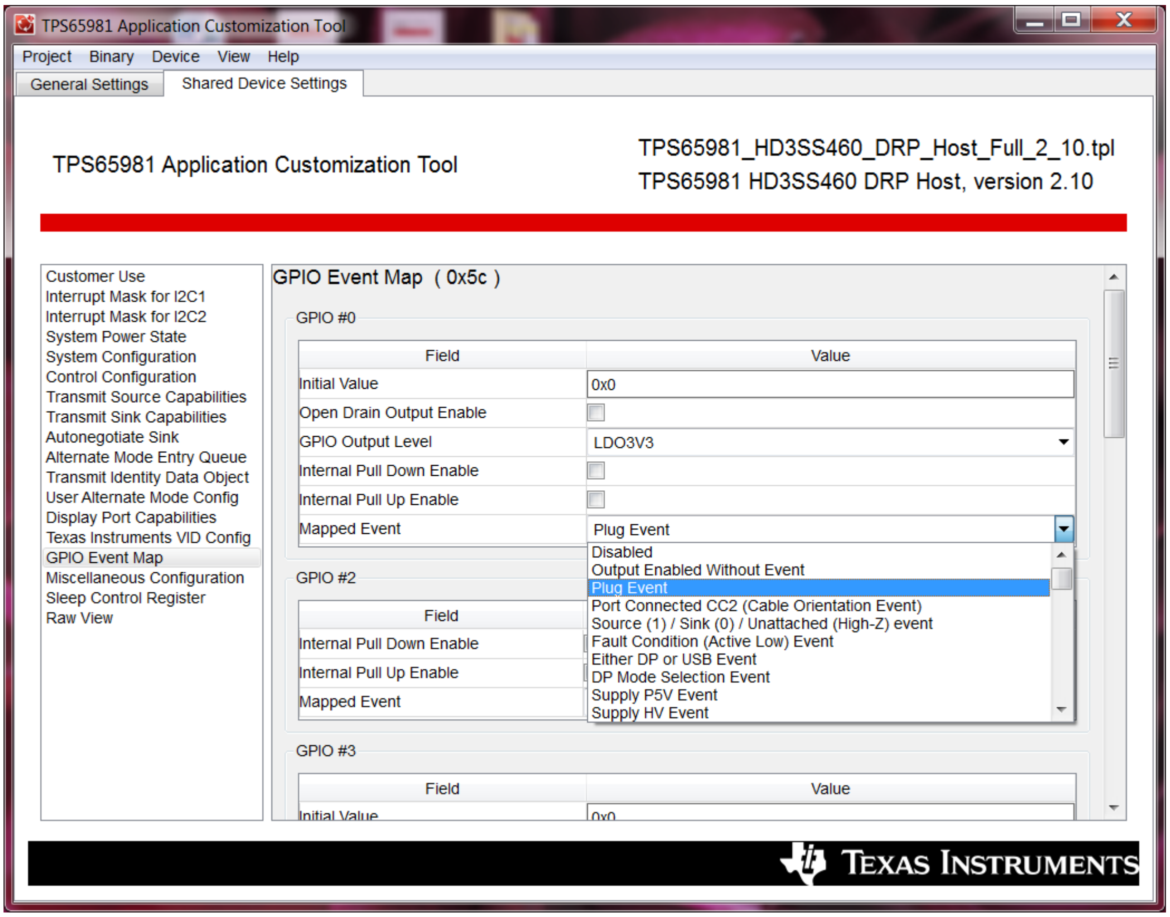
Configuration Registers

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

GPIO configuration registers of TPS6598x allows 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/pulldown resistors for each GPIO can also be configured using configuration register. Note that some of the GPIOs that are pre-configured in the firmware for specific event can't be changed using the Application Customizer tool.

3.1 GPIO Event Example Settings

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


TPS65981 Application Customization Tool
Project Binary Device View Help
General Settings
Shared Device Settings

TPS65981 Application Customization Tool

TPS65981_HD3SS460_DRP_Host_Full_2_10.tpl
TPS65981 HD3SS460 DRP Host, version 2.10

Customer Use
Interrupt Mask for I2C1
Interrupt Mask for I2C2
System Power State
System Configuration
Control Configuration
Transmit Source Capabilities
Transmit Sink Capabilities
Autonegotiate Sink
Alternate Mode Entry Queue
Transmit Identity Data Object
User Alternate Mode Config
Display Port Capabilities
Texas Instruments VID Config
GPIO Event Map
Miscellaneous Configuration
Sleep Control Register
Raw View

GPIO Event Map (0x5c)

GPIO #0

Field	Value
Initial Value	0x0
Open Drain Output Enable	☐
GPIO Output Level	LDO3V3
Internal Pull Down Enable	☐
Internal Pull Up Enable	☐
Mapped Event	Plug Event

Disabled
Output Enabled Without Event
Plug Event
Port Connected CC2 (Cable Orientation Event)
Source (1) / Sink (0) / Unattached (High-Z) event
Fault Condition (Active Low) Event
Either DP or USB Event
DP Mode Selection Event
Supply P5V Event
Supply HV Event

GPIO #2

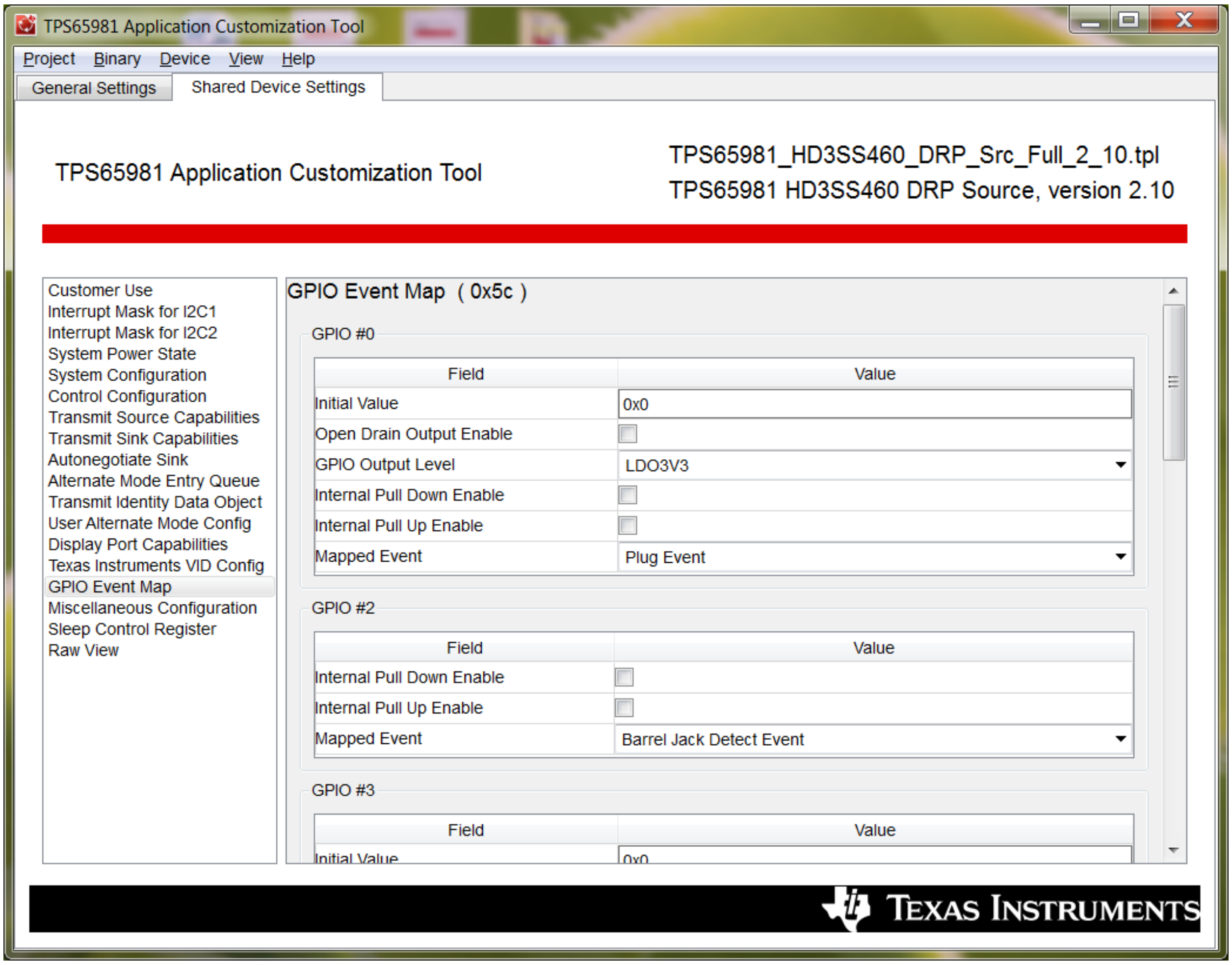
Field	Value
Internal Pull Down Enable	
Internal Pull Up Enable	
Mapped Event	

GPIO #3

Field	Value
Initial Value	0x0

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 TPS65981EVM. Once the project template is loaded all the relevant GPIO Events that are configured can be seen from "GPIO Event Map" page of the tool as shown in Figure 2.


TPS65981 Application Customization Tool
Project Binary Device View Help
General Settings
Shared Device Settings

TPS65981 Application Customization Tool

TPS65981_HD3SS460_DRP_Src_Full_2_10.tpl
TPS65981 HD3SS460 DRP Source, version 2.10

Customer Use
Interrupt Mask for I2C1
Interrupt Mask for I2C2
System Power State
System Configuration
Control Configuration
Transmit Source Capabilities
Transmit Sink Capabilities
Autonegotiate Sink
Alternate Mode Entry Queue
Transmit Identity Data Object
User Alternate Mode Config
Display Port Capabilities
Texas Instruments VID Config
GPIO Event Map
Miscellaneous Configuration
Sleep Control Register
Raw View

GPIO Event Map (0x5c)

GPIO #0

Field	Value
Initial Value	0x0
Open Drain Output Enable	☐
GPIO Output Level	LDO3V3
Internal Pull Down Enable	☐
Internal Pull Up Enable	☐
Mapped Event	Plug Event

GPIO #2

Field	Value
Internal Pull Down Enable	☐
Internal Pull Up Enable	☐
Mapped Event	Barrel Jack Detect Event

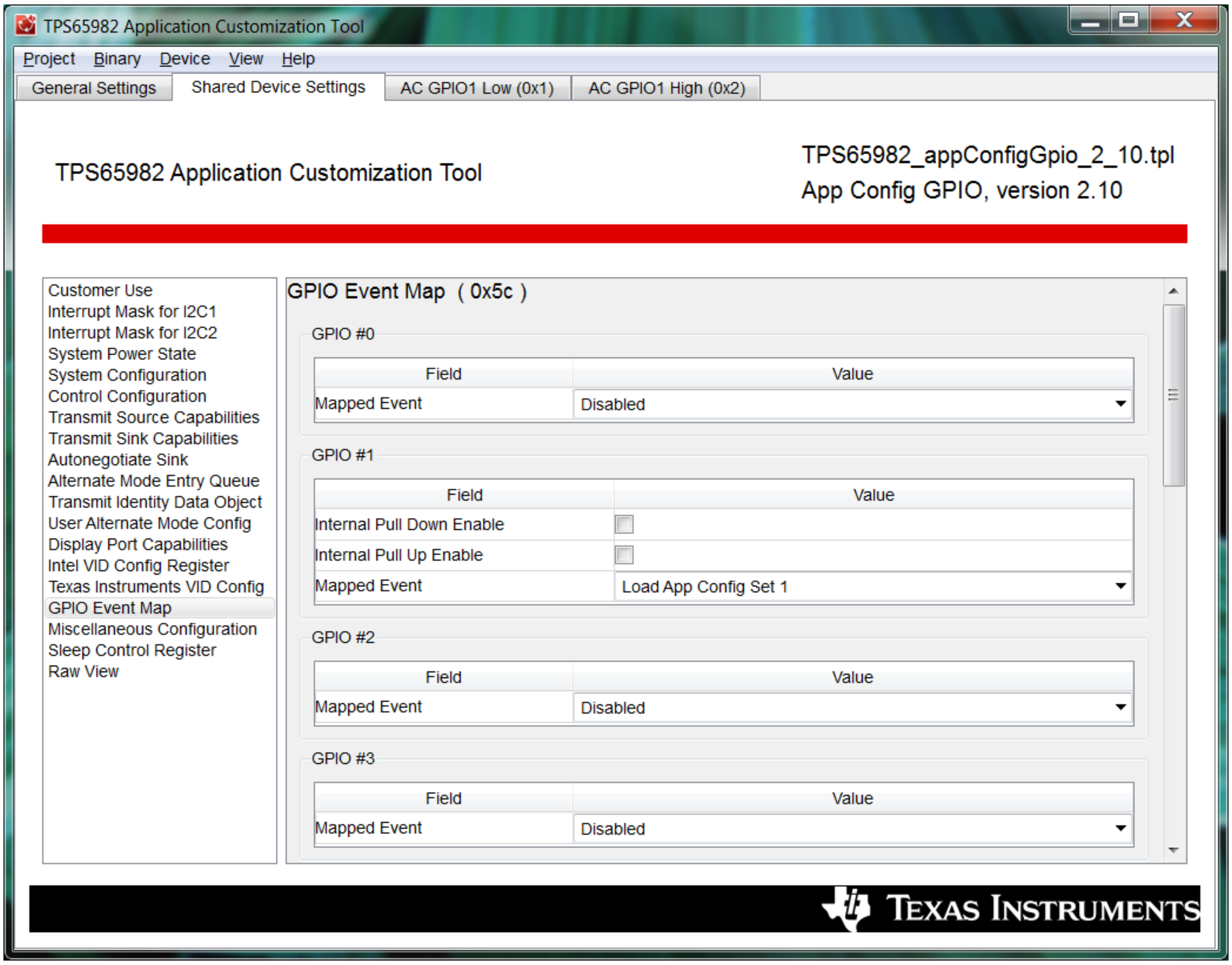
GPIO #3

Field	Value
Initial Value	0x0

Figure 2: Template with GPIO Events mapped for TPS65981EVM

3.2 App Config GPIO Event Settings

There are advanced GPIO events that 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*" demonstrates an example of how to use TPS6598x App Config GPIO feature. Once this project template is loaded, Figure 3 show "GPIO Event Map" page of the GUI which indicates that Load App Config Set 1 event has been mapped to GPIO #1.


TPS65982 Application Customization Tool
Project Binary Device View Help
General Settings
Shared Device Settings
AC GPIO1 Low (0x1)
AC GPIO1 High (0x2)
TPS65982 Application Customization Tool
TPS65982_appConfigGpio_2_10.tpl
App Config GPIO, version 2.10
Customer Use
Interrupt Mask for I2C1
Interrupt Mask for I2C2
System Power State
System Configuration
Control Configuration
Transmit Source Capabilities
Transmit Sink Capabilities
Autonegotiate Sink
Alternate Mode Entry Queue
Transmit Identity Data Object
User Alternate Mode Config
Display Port Capabilities
Intel VID Config Register
Texas Instruments VID Config
GPIO Event Map
Miscellaneous Configuration
Sleep Control Register
Raw View
GPIO Event Map (0x5c)
TEXAS INSTRUMENTS

GPIO #0

Field	Value
Mapped Event	Disabled

GPIO #1

Field	Value
Internal Pull Down Enable	☐
Internal Pull Up Enable	☐
Mapped Event	Load App Config Set 1

GPIO #2

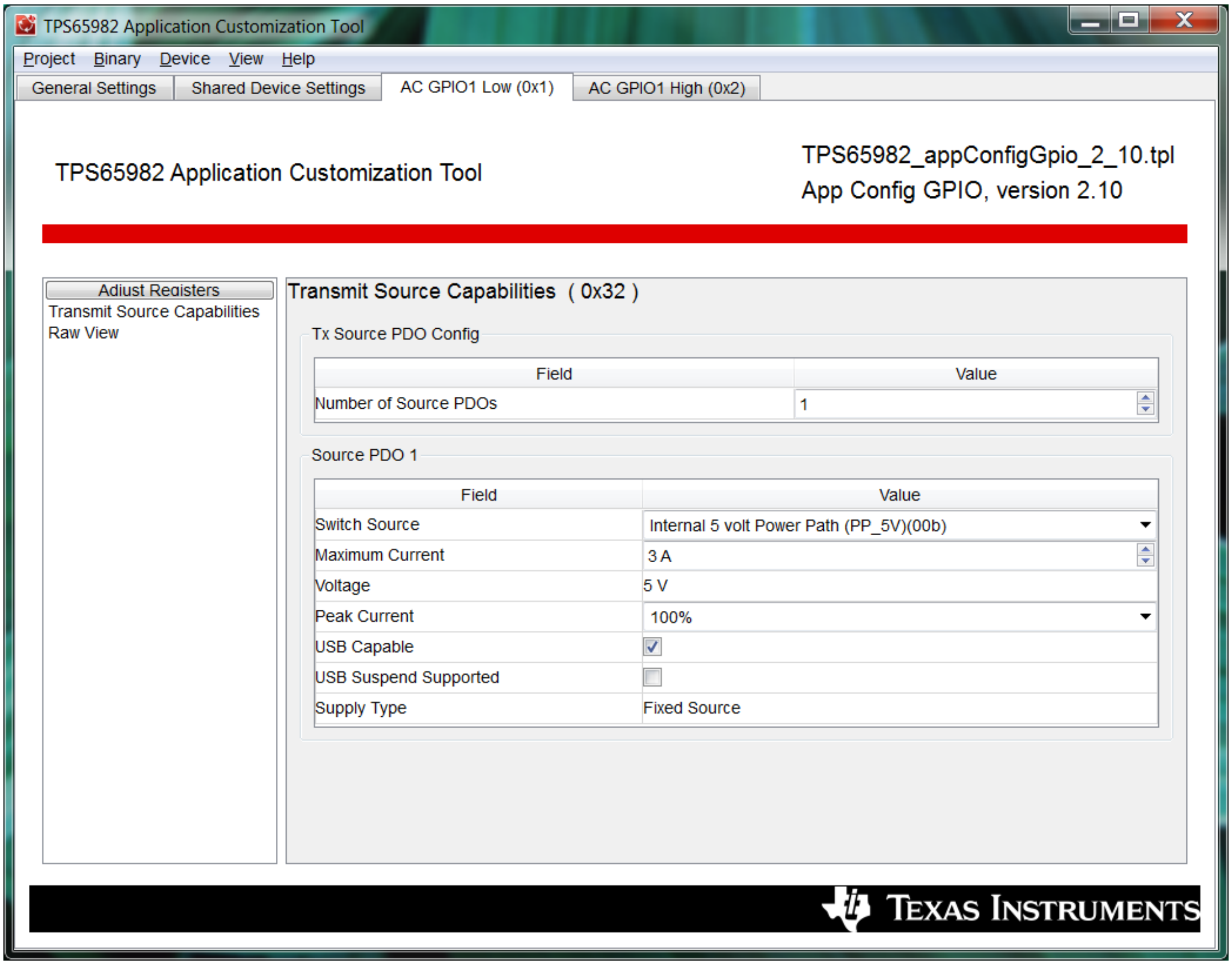
Field	Value
Mapped Event	Disabled

GPIO #3

Field	Value
Mapped Event	Disabled

Figure 3: Mapping App Config Set to GPIO Event

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


TPS65982 Application Customization Tool
Project Binary Device View Help
General Settings
Shared Device Settings
AC GPIO1 Low (0x1)
AC GPIO1 High (0x2)
TPS65982 Application Customization Tool
TPS65982_appConfigGpio_2_10.tpl
App Config GPIO, version 2.10
Adjust Registers
Transmit Source Capabilities
Raw View
TEXAS INSTRUMENTS

Transmit Source Capabilities (0x32)

Tx Source PDO Config

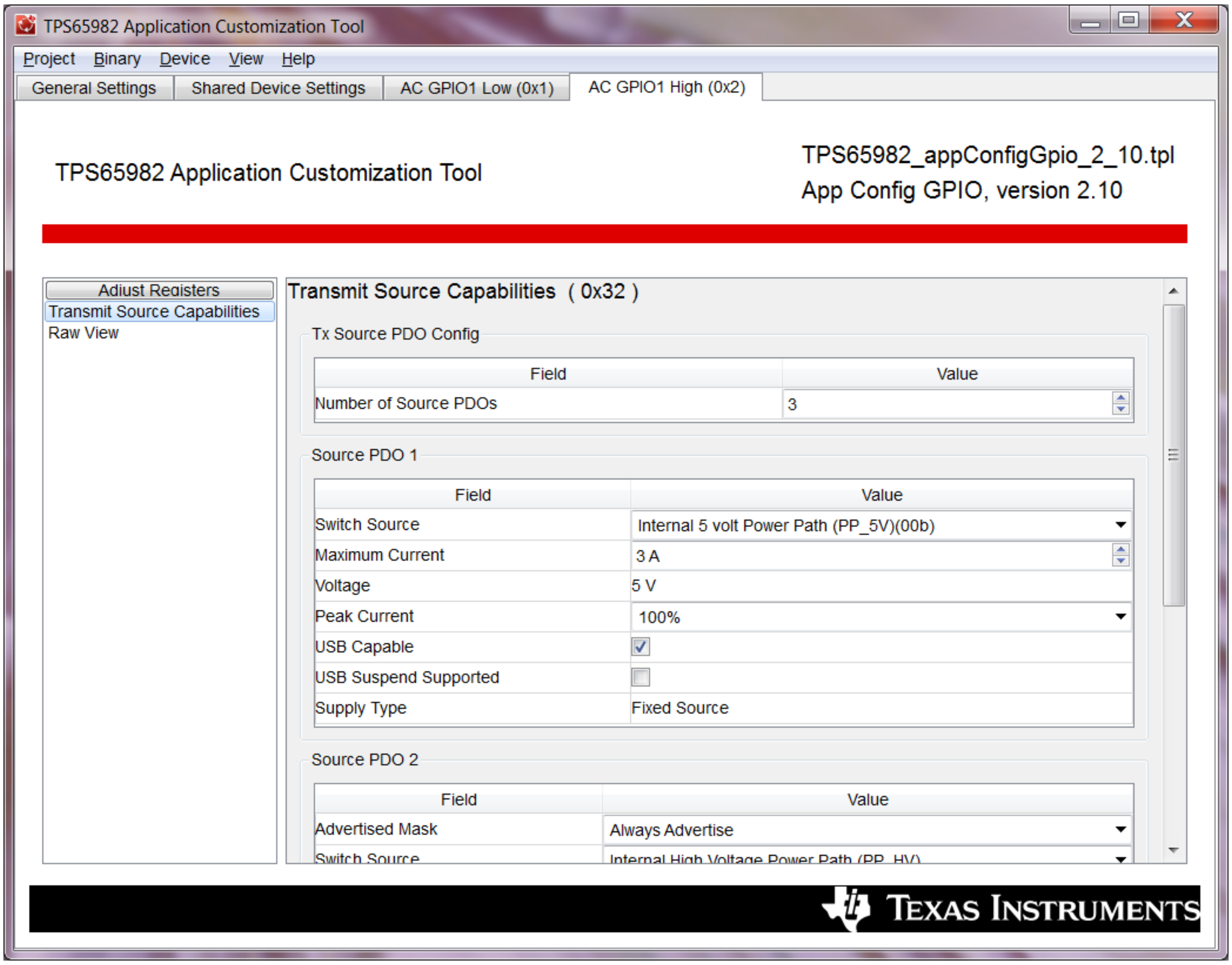
Field	Value
Number of Source PDOs	1

Source PDO 1

Field	Value
Switch Source	Internal 5 volt Power Path (PP_5V)(00b)
Maximum Current	3 A
Voltage	5 V
Peak Current	100%
USB Capable	✓
USB Suspend Supported	☐
Supply Type	Fixed Source

Figure 4: App Config GPIO Set Event, GPIO Low settings example

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


TPS65982 Application Customization Tool
Project Binary Device View Help
General Settings
Shared Device Settings
AC GPIO1 Low (0x1)
AC GPIO1 High (0x2)
TPS65982 Application Customization Tool
TPS65982_appConfigGpio_2_10.tpl
App Config GPIO, version 2.10
Adjust Registers
Transmit Source Capabilities
Raw View
TEXAS INSTRUMENTS

Transmit Source Capabilities (0x32)

Tx Source PDO Config

Field	Value
Number of Source PDOs	3

Source PDO 1

Field	Value
Switch Source	Internal 5 volt Power Path (PP_5V)(00b)
Maximum Current	3 A
Voltage	5 V
Peak Current	100%
USB Capable	✓
USB Suspend Supported	☐
Supply Type	Fixed Source

Source PDO 2

Field	Value
Advertised Mask	Always Advertise
Switch Source	Internal High Voltage Power Path (PP_HV)

Figure 5: App Config GPIO Set Event, GPIO High settings example

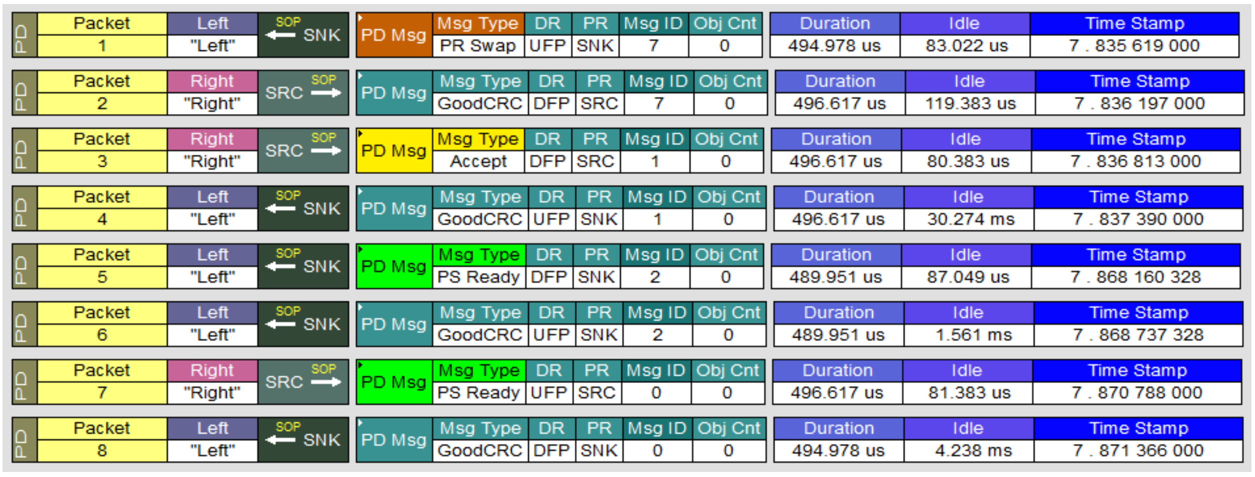
4 PD Controller customization by GPIO Events

This section is targeted to show how TI PD controller GPIO events can be used in a system to alter system behavior keeping the core firmware same. We would use Barrel Jack Event as example to show how a docking application can initiate 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

Actual PD trace of this example Barrel Jack Event implementation in a system is shown below. This event can be used in a docking application when external power becomes available to the docking station. Rising edge on the GPIO that has been assigned for Barrel Jack Event initiates the required PD message flow for power role swap.

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

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

Packet 1 → EVM-DCK is UFP/SNK and sends "PR Swap" message to the EVM-LPT which is DFP/SRC.

Packet 2 → DFP/SRC sends "GoodCRC" acknowledgement response for "PR Swap" message.

Packet 3 → DFP/SRC sends "Accept" message to signal that it is willing to do a Power Role Swap and has begun the Power Role Swap sequence.

Packet 4 → UFP/SNK sends "GoodCRC" acknowledgement response.

Packet 5 → EVM-LPT changes role to DFP/**SNK** and sends “PS Ready” message. It is important to note that the initial Source Port is now setting the “Port Power Role” field to **Sink** in the “PS Ready” message indicating that the initial Source’s power supply is turned off.

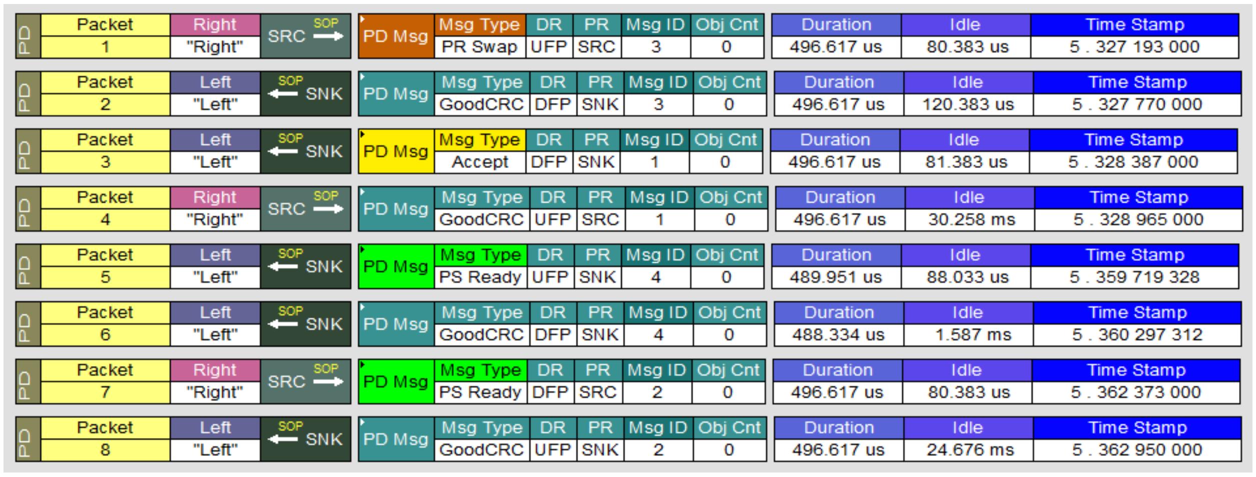
Packet 6 → EVM-DCK sends “GoodCRC” acknowledgement response for “PS Ready” message. Note that the GoodCRC Message sent by the initial Sink in response to the “PS Ready” message from the initial Source will have its Port Power Role field set to Sink since this is initiated by the Protocol Layer.

Packet 7 → EVM-DCK changes role to UFP/**SRC** and sends “PS Ready” message.

Packet 8 → EVM-LPT which is now DFP/SNK sends “GoodCRC” acknowledgement response.

4.2 Barrel Jack Removal Event PD Flow

Once power is removed from the EVM-DCK, falling edge generated on the GPIO would initiate the reverse process so that EVM-LPT can become the Power Source again. Actual PD trace of the removal event is shown in Figure 7.



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

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

Status Register

- 0x72, GPIO Status

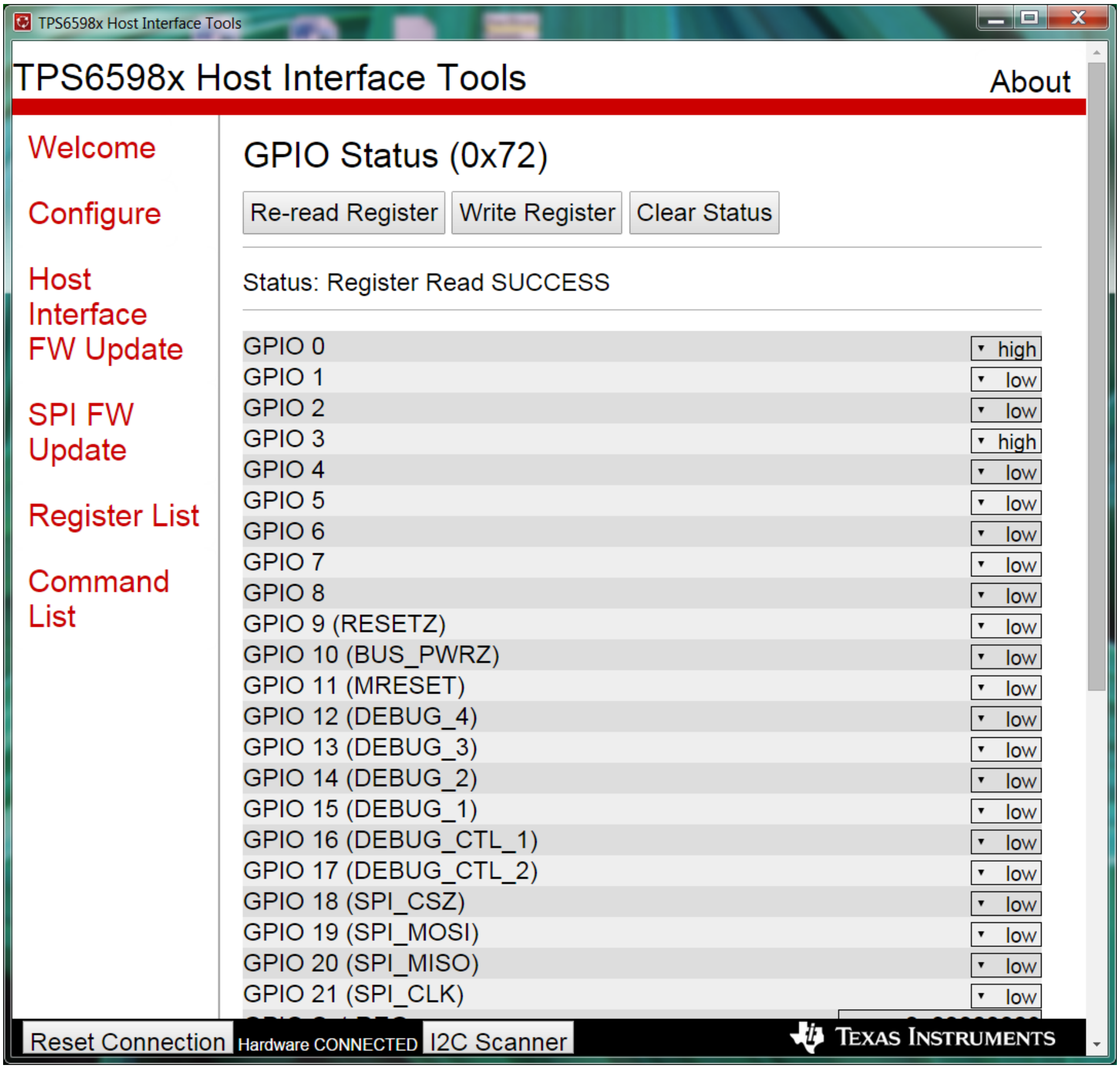
4CC Commands

- ‘*GPie’*, GPIO Input Enable
- ‘*GPoe’*, GPIO Output Enable
- ‘*GPsh’*, GPIO Set Output High
- ‘*GPsl’*, GPIO Set Output Low

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

5.1 GPIO Status monitoring

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 5V, 9V, 15V and 20V variable supplies, 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 PD contract is done for 5V. Once an explicit PD contract is negotiated for 20V supply, both GPIO7 and GPIO8 are driven High by the PD controller as indicated in Figure 9.

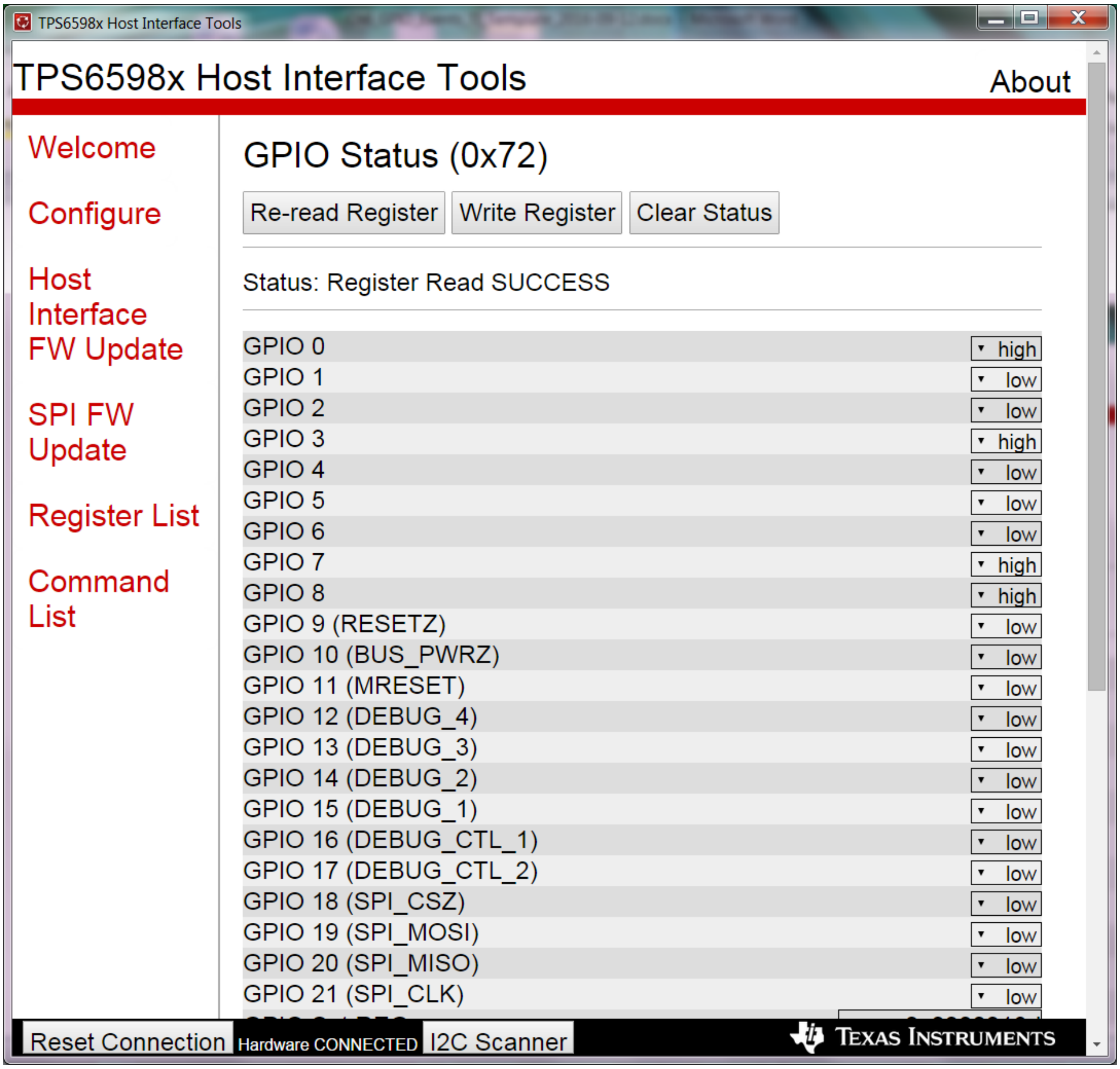


GPIO Status (0x72)

Status: Register Read SUCCESS

GPIO 0	high
GPIO 1	low
GPIO 2	low
GPIO 3	high
GPIO 4	low
GPIO 5	low
GPIO 6	low
GPIO 7	low
GPIO 8	low
GPIO 9 (RESETZ)	low
GPIO 10 (BUS_PWRZ)	low
GPIO 11 (MRESET)	low
GPIO 12 (DEBUG_4)	low
GPIO 13 (DEBUG_3)	low
GPIO 14 (DEBUG_2)	low
GPIO 15 (DEBUG_1)	low
GPIO 16 (DEBUG_CTL_1)	low
GPIO 17 (DEBUG_CTL_2)	low
GPIO 18 (SPI_CSZ)	low
GPIO 19 (SPI_MOSI)	low
GPIO 20 (SPI_MISO)	low
GPIO 21 (SPI_CLK)	low

Figure 8: Variable DC/DC GPIO status for 5V supply



GPIO Status (0x72)

Status: Register Read SUCCESS

GPIO 0	high
GPIO 1	low
GPIO 2	low
GPIO 3	high
GPIO 4	low
GPIO 5	low
GPIO 6	low
GPIO 7	high
GPIO 8	high
GPIO 9 (RESETZ)	low
GPIO 10 (BUS_PWRZ)	low
GPIO 11 (MRESET)	low
GPIO 12 (DEBUG_4)	low
GPIO 13 (DEBUG_3)	low
GPIO 14 (DEBUG_2)	low
GPIO 15 (DEBUG_1)	low
GPIO 16 (DEBUG_CTL_1)	low
GPIO 17 (DEBUG_CTL_2)	low
GPIO 18 (SPI_CSZ)	low
GPIO 19 (SPI_MOSI)	low
GPIO 20 (SPI_MISO)	low
GPIO 21 (SPI_CLK)	low

Figure 9: Variable DC/DC GPIO status for 20V supply

5.2 Using 4CC GPIO Commands

TPS6598x Host Interface Utility tool can be used to exercise the GPIO related 4CC commands and observe, 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.

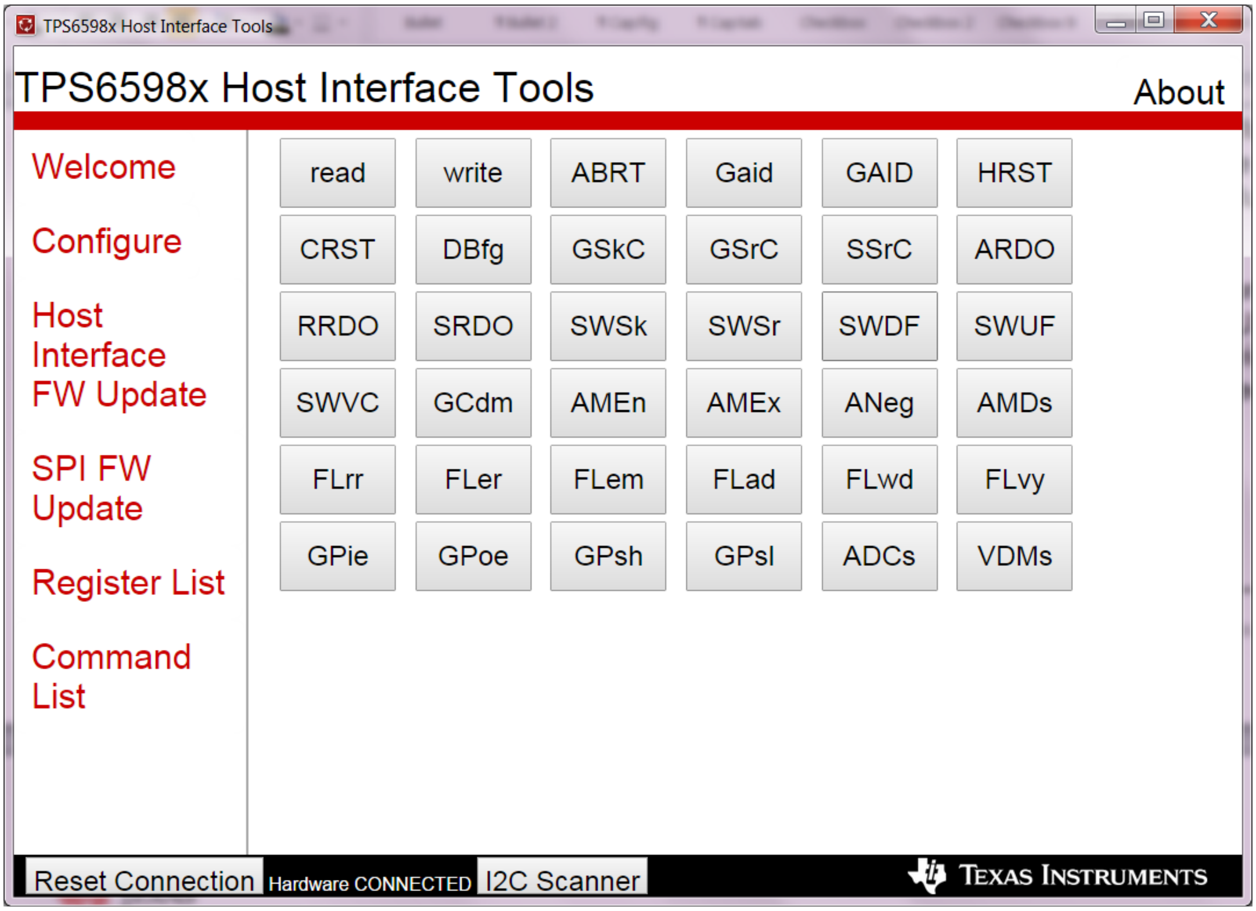
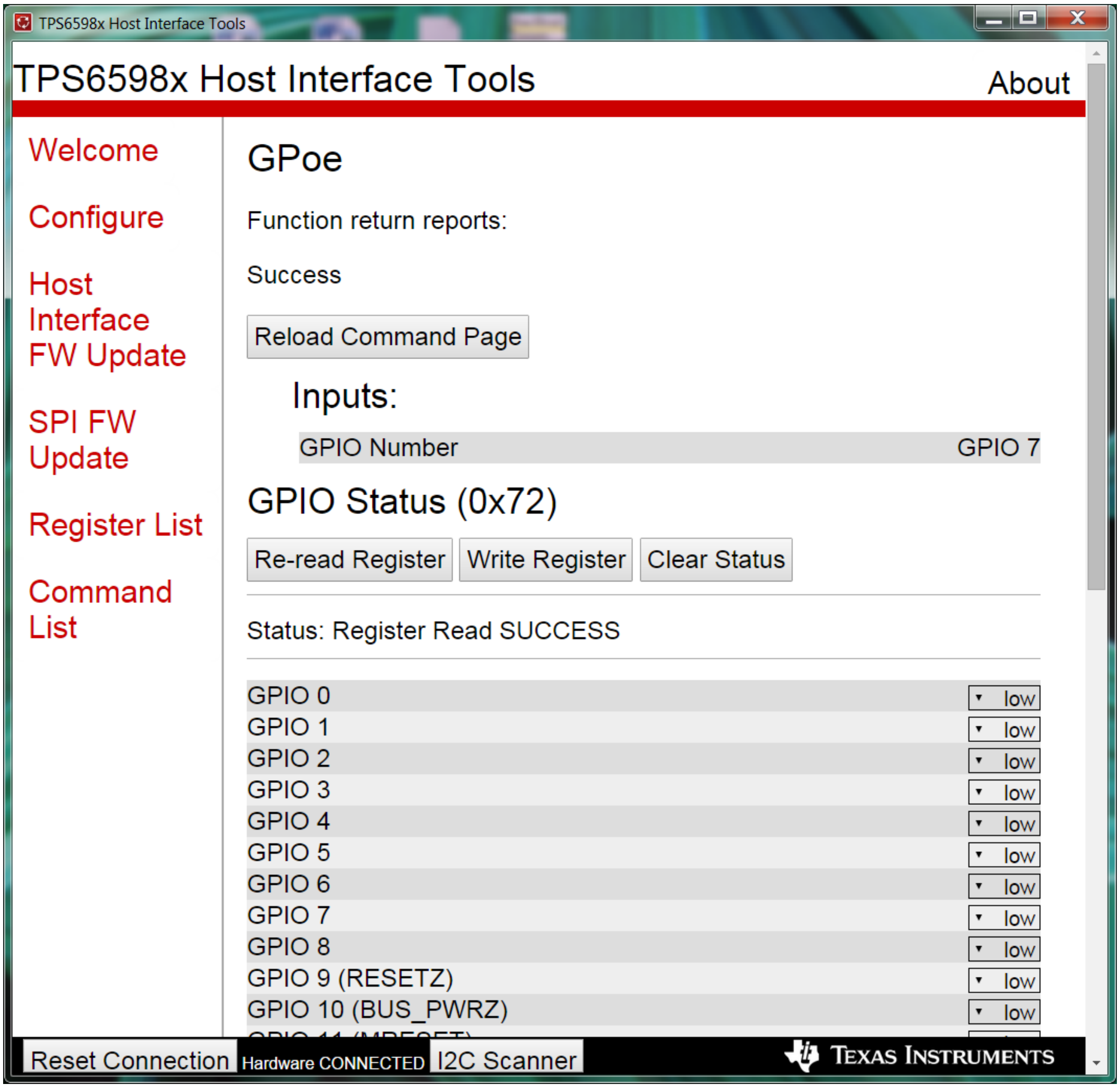


Figure 10: 4CC Commands in Utilities Tool

For example, to set the GPIO7 to High

- First send 'GPoe' 4CC command as shown in Figure 11
- Then send 'GPsh' 4CC command as shown in Figure 12
- In the GPIO Status (0x72) it can be seen that GPIO7 has been set to High



GPoe

Function return reports:

Success

Inputs:

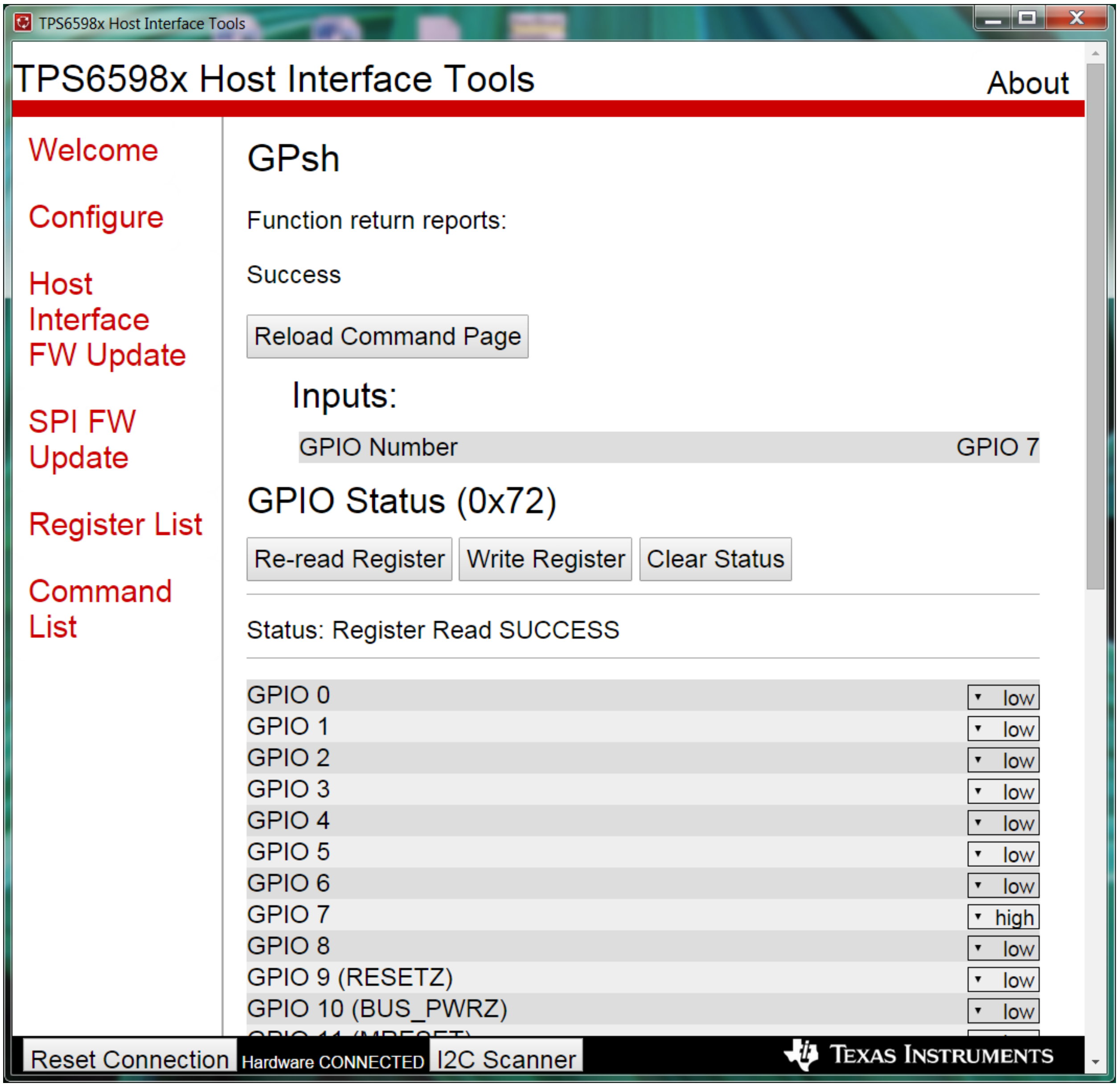
GPIO Number	GPIO 7

GPIO Status (0x72)

Status: Register Read SUCCESS

GPIO 0	low
GPIO 1	low
GPIO 2	low
GPIO 3	low
GPIO 4	low
GPIO 5	low
GPIO 6	low
GPIO 7	low
GPIO 8	low
GPIO 9 (RESETZ)	low
GPIO 10 (BUS_PWRZ)	low

Figure 11: Using ‘GPoe’ 4CC command



GPsh

Function return reports:

Success

Inputs:

GPIO Number	GPIO 7

GPIO Status (0x72)

Status: Register Read SUCCESS

GPIO 0	low
GPIO 1	low
GPIO 2	low
GPIO 3	low
GPIO 4	low
GPIO 5	low
GPIO 6	low
GPIO 7	high
GPIO 8	low
GPIO 9 (RESETZ)	low
GPIO 10 (BUS_PWRZ)	low

Hardware CONNECTED

Figure 12: Using ‘GPsh’ 4CC command