EZ-PD™ PMG1-S3 Dock user guide

About this document

This user guide serves as an easy-to-use dock design reference manual for two USB-C Dock solutions using EZ-PD™ PMG1-S3:

- EZ-PD™ PMG1-S3 USB-C Dock
- Griffin Creek Thunderbolt Dock

This document also introduces the EZ-PD™ PMG1-S3 Dock SDK and EZ-PD™ Dock Configuration Utility and enables you to evaluate the Infineon EZ-PD™ PMG1-S3 based USB-C Dock reference hardware.

Scope and purpose

This document familiarizes the readers with EZ-PD™ PMG1-S3 USB-C Dock hardware, the EZ-PD™ Dock Configuration Utility, and Griffin Creek Dock hardware. It showcases use-case examples for EZ-PD™ PMG1-S3 USB-C Dock hardware, which can also be applied to Griffin Creek Dock hardware.

The document explains examples on how to modify the PMG1-S3 Dock SDK for the EZ-PD™ PMG1-S3 USB-C Dock solution. These steps can be extended to the PMG1S3 TBT Griffin Creek Dock Solution for Griffin Creek Dock hardware.

Intended audience

The document is intended for Original Device Manufacturer (ODM) or Original Equipment Manufacturer (OEM) designing docks based on EZ-PD™ PMG1-S3 USB-C Dock hardware or Griffin Creek Dock hardware.

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **4**
1.1 EZ-PD™ PMG1 high-voltage microcontrollers 4
1.2 Dock solutions 4
1.2.1 EZ-PD™ PMG1-S3 USB-C Dock 4
1.2.2 Griffin Creek hardware 4
1.3 Development tools used with the Docks 5
1.3.1 EZ-PD™ PMG1-S3 Dock SDK 5
1.3.2 Programming and debugging 5
1.3.3 ModusToolbox™ software 5
1.3.4 EZ-PD™ Dock configuration tool 5
1.3.5 EZ-PD™ Configuration Utility 5
1.3.6 EZ-PD™ Analyzer tool 5
2 EZ-PD™ PMG1-S3 USB-C Dock **6**
2.1 EZ-PD™ PMG1-S3 USB-C Dock hardware details 6
2.2 USB-C hardware system block diagram 12
2.3 Features 13
3 EZ-PD™ PMG1-S3 USB-C Dock application examples **14**
3.1 Prerequisite 14
3.2 Charging a notebook from Dock's upstream port [SPR example] 14
3.3 EPR source example 15
3.4 USB data and display mirroring or extension using downstream ports 16
3.5 Charging on DS1 and DS3 without US connection 16
4 Griffin Creek Thunderbolt Dock **18**
4.1 Griffin Creek dock hardware 18
4.2 Connectors and Jumper settings 19
4.3 Features 19
4.4 Griffin Creek Dock's PMG1-S3 SWD programming 20
5 CCG7SC **21**
5.1 I2C slave address 21
5.2 Data Mux control 21
6 Customizing the EZ-USB™ HX3 Hub **22**
7 EZ-PD™ Dock Configuration Utility **23**
7.1 Prerequisite 23
7.2 Modifying PMG1-S3 VID and PID 24
7.3 Enable/Disable Smart Power 25
7.4 Adding a new devices component or removal of a component for firmware update support 26
7.5 Enabling signed firmware update 28
8 Dock firmware update **29**
8.1 Prerequisite 30
8.2 Dock status 30
8.3 Dock Composite bin creation 31
8.4 Generating and updating the signature 32
8.4.1 Generating signature 32
8.4.2 Updating signature using Dock Configuration tool 33

Table of contents

8.5 Firmware update 33
8.6 Switching from unsigned firmware update to signed firmware update 35
9 PMG1-S3 Dock SDK customization 36
9.1 Prerequisite 36
9.2 Customizing the US port buck boost controller 36
9.2.1 Makefile changes 36
9.2.2 Creating new buck boost support header and source files 37
9.2.3 Modifications to new controller header file cy_app_new_buck_boost.h 38
9.2.4 Modification to cy_app_buck_boost.h 38
9.2.5 Modifications to new controller source files cy_app_new_buck_boost.c and main.c 38
9.3 Customizing the Data Mux for USB-C Dock 40
9.4 USB and DP hub controllers modifications for custom designs 41
9.5 Enabling the custom ALT mode support 41
References 45
Glossary 46
Revision history 48
Disclaimer 49

1 Introduction

1.1 EZ-PD™ PMG1 high-voltage microcontrollers

EZ-PD™ PMG1 (Power Delivery microcontroller Gen 1) is the industry's first family of high-voltage microcontrollers (MCUs) with USB-C Power Delivery (PD). These chips include an Arm® Cortex®-M0/M0+ CPU and a USB-C PD controller, along with analog and digital peripherals. It is targeted at any embedded system that provides or consumes power to or from a high-voltage USB-C PD port and leverages the microcontroller to provide additional control capability.

In addition to the EZ-PD™ PMG1-S3, which is used as a PD controller and a Dock Management Controller (DMC) in dock solutions, the following devices from Infineon can also be used as PD controllers in dock solutions:

- EZ-PD™ CCG7SC
- EZ-PD™ CCG7DC

For more details on other Type-C controllers from Infineon and for feature comparisons, see the USB-C Power Delivery controllers.

1.2 Dock solutions

The EZ-PD™ PMG1-S3 Dock user guide serves as an easy-to-use dock design reference manual for the following listed Dock solutions, which use the EZ-PD™ PMG1-S3 as a Dock management controller with USB-C Power Delivery.

- EZ-PD™ PMG1-S3 USB-C Dock
- Griffin Creek Dock

This document also introduces the EZ-PD™ PMG1-S3 Dock SDK and EZ-PD™ Dock Configuration Utility and enables customers to evaluate the Infineon EZ-PD™ PMG1-S3-based USB-C Dock solutions.

1.2.1 EZ-PD™ PMG1-S3 USB-C Dock

EZ-PD™ PMG1-S3 USB-C Dock is a self-powered dock that supports one upstream (US) Type-C Power Delivery (PD) port, two downstream (DS1, DS3) Type-C PD ports, and one Type-A port. The US port's power is controlled by an MPS4247 or RT6190 buck boost converter. The dock provides power connectors so that either of the buck boost converters can be connected to the hardware. The Dock also supports one HDMI and one DP port for extension or mirroring of displays. USB-C Dock hardware (HW) can be used as a reference platform by ODM's for designing custom USB-C docking stations.

1.2.2 Griffin Creek hardware

Griffin Creek is a self-powered TBT and USB4 reference design from Intel. It uses Infineon's EZ-PD™ PMG1-S3 to control the upstream Type-C and downstream Type-C ports. EZ-PD™ PMG1-S3 is also used to exchange host capabilities with the Goshen Ridge controller and downstream (DS) Type-C ports. Intel's Goshen Ridge controller controls USB SuperSpeed data and display data.

Customers can contact Intel for Griffin Creek Dock hardware and other hardware design files.

1.3 Development tools used with the Docks

1.3.1 EZ-PD™ PMG1-S3 Dock SDK

The PMG1-S3 Dock SDK is a collection of software libraries and reference projects hosted on GitHub repositories and made available through the ModusToolbox™ software development environment.

The PMG1-S3 Dock SDK supports the following reference projects:

- PMG1-S3 based Griffin Creek Dock Solution supports DP, USB4, and TBT alternate modes.
- PMG1-S3 based USB-C Dock Solution supports DP alternate mode.

Refer to the EZ-PD™ PMG1-S3 SDK user guide for details about the EZ-PD™ PMG1-S3 Dock SDK.

1.3.2 Programming and debugging

Install the ModusToolbox™ Programming tools package on the host development platform to program PMG1-S3 and CCG7SC using Miniprog4 Programmer/Debugger. After installation, the tool is available as a mtb-programmer on the host PC. Miniprog4 can also be used for debugging using the Eclipse-based development environment ModusToolbox™.

Refer to the Miniprog4 webpage for programmer usage and how to obtain a programmer.

1.3.3 ModusToolbox™ software

ModusToolbox™ Software is a modern, extensible development ecosystem supporting a wide range of Infineon microcontroller devices, including PSoC™ Arm® Cortex® Microcontrollers, TRAVEO™ T2G Arm® Cortex® Microcontroller, XMC™ Industrial Microcontrollers, AIROC™ Wi-Fi devices, AIROC™ Bluetooth® devices, and USB-C Power Delivery Microcontrollers.
The ModusToolbox™ software environment provides template applications, a set of tools for configuring the device, setting up peripherals, and complementing projects with world-class middleware.

1.3.4 EZ-PD™ Dock configuration tool

The EZ-PD™ Dock Configuration Utility is a GUI application that supports PMG1-S3-based dock designs. Refer to the EZ-PD™ Dock Configuration Utility user guide for tool usage details.

1.3.5 EZ-PD™ Configuration Utility

The EZ-PD™ Configuration Utility helps customers configure and program Infineon's CCGx devices. The tool can be used to configure the CCG7SC firmware and the source power delivery object (PDO) with the pre-built binaries.

1.3.6 EZ-PD™ Analyzer tool

Infineon uses the CY4500 kit and the CY4500 EZ-PD™ Protocol Analyzer Utility for capturing the CC traffic between Type-C port partners for analysis of USB PD packets. The CY4500 kit and EZ-PD™ Protocol Analyzer Utility are used to showcase the examples of USB PD traffic in this document.

2 EZ-PD™ PMG1-S3 USB-C Dock

The EZ-PD™ PMG1-S3 USB-C Dock is a reference design from Infineon. The reference design uses the dual-port EZ-PD™ PMG1-S3 MCU, the EZ-USB™ HX3 USB 5 Gbps USB hub, and the EZ-PD™ CCG7SC.

The reference hardware supports one upstream (US) Type-C port, two downstream (DS) Type-C ports, one Type-A port, one HDMI, and a DP port to demonstrate port expandability capabilities over a single Type-C cable.

The EZ-PD™ PMG1-S3 controls upstream and one of the downstream Type-C ports. The other downstream Type-C PD port is controlled by EZ-PD™ CCG7SC. EZ-PD™ CCG7SC is a single-port USB Type-C Power Delivery solution with an integrated buck-boost controller.

The EZ-USB™ HX3 USB 5 Gbps hub controller supports USB SuperSpeed (5 Gbps), USB2 speeds on the US port, and four downstream Type-C/Type-A ports.

The hardware design files for the EZ-PD™ PMG1-S3 USB-C Dock are available at PMG1-S3 Dock webpage.

2.1 EZ-PD™ PMG1-S3 USB-C Dock hardware details

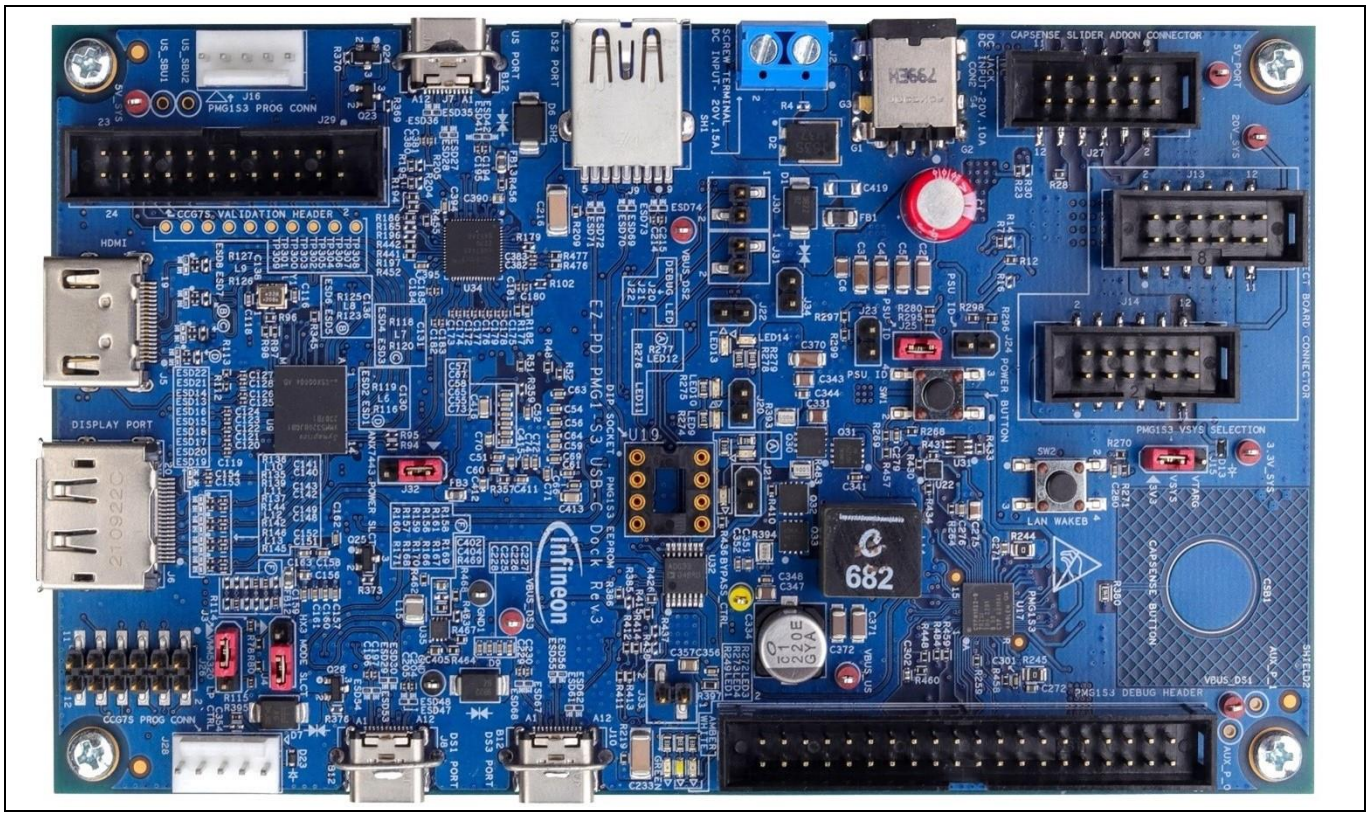


Figure 1 USB-C Dock hardware top view

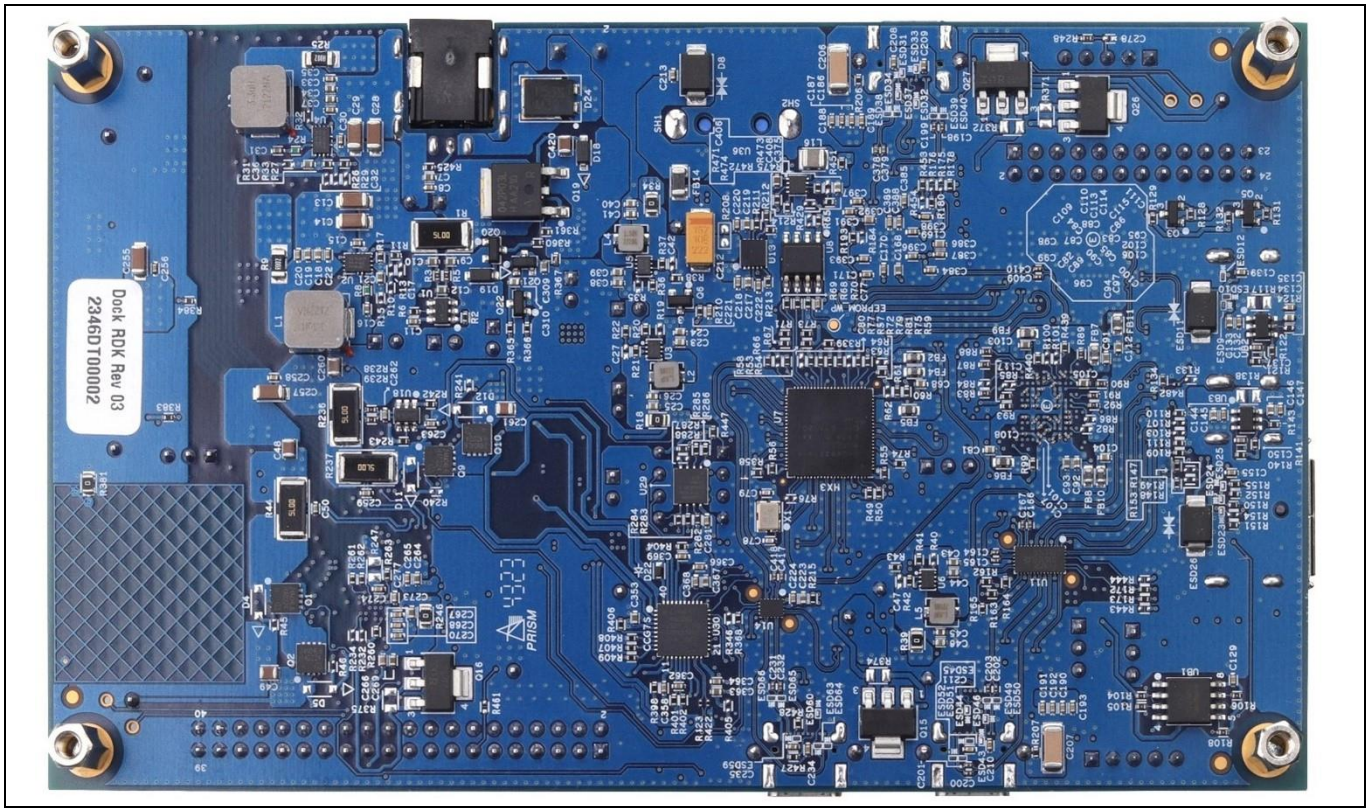


Figure 2 USB-C Dock hardware bottom view

Table 1 describes the default jumper settings to be used with EZ-PD™ PMG1-S3 USB-C Dock hardware.

Table 1 Jumper Settings

Reference	Connector name	Description
J15	1x3 jumper	3-pin header to select the PMG1-S3 VSYS input. 1 – 2: Select VSYS input from an external 3.3 V regulator. Install 1 – 2, when power to the hardware is supplied externally. 2 – 3: Supply VSYS from the VTARG pin of the PMG1-S3 Serial Wire Debug (SWD) programming header. **Default jumper setting**: Short 1 – 2.
J32	1x3 jumper	3-pin header to choose, 1.2 V supply to upstream data mux. 1 – 2: Upstream data mux generates 1.2 V from its internal LDO. 2 – 3: 1.2 V is supplied from an external regulator. **Default jumper setting**: Short 1 – 2.
J3	1x3 jumper	3-pin header to choose whether display controller low power mode is enabled or disabled. 1 – 2: Low-power mode is disabled. Install it only while programming the display controller firmware. 2 – 3: Low-power mode is enabled. Install it for display mirroring or extension. **Default jumper setting**: Short 2 – 3.
J4	1x3 jumper	3-pin header to select the HX3 USB hub MODE_SEL [1] pin setting. 1 – 2: HX3 boots from the internal ROM. Use this option to program an external EEPROM using the Blaster Plus tool. 2 – 3: HX3 boots or loads settings from an external EEPROM. **Default jumper setting**: Short 2 – 3.
J34	1x2 jumper	2-pin header to isolate the Host Processor Interface (HPI) and HPI Interrupt GPIO between CCG7SC and PMG1-S3. **Default jumper setting**: uninstall
J23	1x2 jumper	2-pin headers for setting the Power Adaptor size. Based on the power input to the CON2/J2 screw terminal, the J23, J24, and J25 headers are to be set as per Table 2. PMG1-S3 Dock SDK uses the power input setting as input to limit the maximum PDO's advertised on the upstream port and for smart power feature implementation. See Table 2 for power adaptor settings supported by USB-C Dock hardware. Customers are expected to set the jumper settings as per the adaptor availability correctly; otherwise, the smart power feature will not work. *Note: To advertise SPR range PDO's, set the power adaptor selection jumper setting to 120 W/135 W.* *Note: For EPR range PDO's, set the power adaptor selection jumper settings to >135 W.* For more details on firmware implementation and the smart power feature, refer to [1].
J24	1x2 jumper	
J25	1x2 jumper	

Table 2 describes the J23, J24, and J25 jumper settings to be used for different power adaptor wattages.

Table 2 Power adaptor selection

Power adaptor/screw terminal wattage	J23	J24	J25
120 W	OPEN	OPEN	CLOSE
135 W	OPEN	CLOSE	OPEN
150 W	OPEN	CLOSE	CLOSE
180 W	CLOSE	OPEN	OPEN
200 W	CLOSE	OPEN	CLOSE
230 W	CLOSE	CLOSE	OPEN
280 W	CLOSE	CLOSE	CLOSE

Table 3 describes the different Infineon's components used in the EZ-PD™ PMG1-S3 USB-C Dock.

Table 3 Infineon's components used in EZ-PD™ PMG1-S3 USB-C Dock

Reference	Component	Description
U17	EZ-PD™ PMG1-S3	EZ-PD™ PMG1-S3 (CYPM1322-97BZXI) dual-port USB Type-C PD high-voltage MCU
U30	EZ-PD™ CCG7SC	Single-port USB Type-C PD solution with DC-DC controller
U7	EZ-USB™ HX3	CYUSB3314-88LTXC, 4-port USB hub controller
U29	QUAD SPI flash	IC flash 64M SPI 108 MHz 8USON, S25FL064LABNFI041
Q1, Q2, Q9, Q10	BSZ024N04LS6ATMA1	MOSFET N-CH 40 V 24 A/40 A TSDSON
Q3, Q5, Q6, Q20, Q21, Q22, Q23, Q24, Q25, Q28	2N7002H	MOSFET N-CH 60 V 300 MA SOT23-3
Q15, Q16	IRFL024ZTRPBF	MOSFET N-CH 55 V 5.1 A SOT223
Q26, Q27	IRLL014NTRPBF	MOSFET N-CH 55 V 2 A SOT223
Q19	IPD042P03L3GATMA1	MOSFET P-CH 30 V 70 A TO252-3
Q30, Q31, Q32, Q33	BSZ063N04LS6ATMA1	MOSFET N-CH 40 V 15 A/40 A TSDSON
ESD1-8, ESD13-20, ESD35-42, ESD47-ESD54, ESD61-ESD68, ESD71-ESD74	ESD150-B1-W0201	3.8 V (Typ) Clamp 3 A (8/20 µs) Ipp Tvs Diode Surface Mount PG-WLL-2-3
ESD9, ESD10, ESD12, ESD21-ESD25, ESD27-ESD34, ESD43-ESD46, ESD55, ESD56, ESD59, ESD60, ESD69, ESD70	ESD245B1W0201E6327XTSA1	7.5 V Clamp 5.5 A (8/20 µs) Ipp Tvs Diode Surface Mount WLL-2-3

Refer to the bill of material available at PMG1-S3 Dock webpage.

Table 4 describes the different LED indications used in the EZ-PD™ PMG1-S3 USB-C Dock.

Table 4 LED indication

Reference	Color	Description
LED1	Green	Unused in EZ-PD™ PMG1-S3 USB-C Dock reference hardware.
LED3	AMBER	
LED4	White	User LED On: When the dock is powered on. When the SW1 power button press is detected, the LED state changes to on. Blink: While downloading the composite bin to SPI flash and while updating individual components. Off: When the SW1 power button release is detected, the LED state changes to off.
LED3, LED9, LED10, LED11, LED12, LED13, LED14	Orange	Unused in EZ-PD™ PMG1-S3 USB-C Dock reference hardware. Connect these LEDs to any free GPIO's of the PMG1-S3 available on the EZ-PD™ PMG1-S3 USB-C Dock reference hardware.

Table 5 describes the connectors present on the EZ-PD™ PMG1-S3 USB-C Dock.

Table 5 Connectors

Reference	Connector name	Description
CON2	DC Jack Input 20 V – 10 A	Power to the board can be supplied by connecting a 19.5 V and 10 A power adaptor to the connector. The power will be supplied through either CON2 or J2.
J2	Screw Terminal 20 V – 15 A	Power to the board can be supplied through either CON2 or J2. When supplied through J2, connect a bench power supply with a 20 V and up to 15 A rating.
J16	PMG1S3 PROG CONN	5-pin SWD header to program the EZ-PD™ PMG1-S3 MCU using MiniProg4.
J28	CCG7S PROG CONN	5-pin SWD header to program CCG7SC using MiniProg4.
J13, J14	INTER-CONNECT Board Connector	An interconnect board connector is used to connect the buck-boost controller, which controls the power to the upstream port.
J7	US PORT	Upstream Facing Port (UFP) of the Dock, which can be connected to the notebook to charge the notebook and extend the ports.
J6	DS1 PORT	USB PD Downstream Facing Port (DFP). The maximum power rating of the port is 5 V at 3 A. A user can connect a Type-C monitor to mirror or extend the notebook, which is connected to the US port. Or the user can connect any USB PD device.
J9	DS2 Port	Type-A downstream port with USB 3.1 capabilities, which are controlled by HX3. The maximum power rating of the port is 5 V at 1 A.

Reference	Connector name	Description
J10	DS3 Port	USB DFP port, which is controlled by CCG7SC. Supported PDOs are 5 V at 3 A and 9 V at 3 A. The port can support USB 3.1 data, which is controlled by HX3.
J5	HDMI	Connect an external monitor that supports the HDMI port to observe the mirroring/extension of the notebook connected to the US port (J7).
J6	DISPLAY PORT	Connect an external monitor that supports the display port to observe the mirroring/extension of the notebook connected to the US port (J7).
J17	PMG1S3 Debug Header	Customers can connect a digital oscilloscope to debug various signals.
J26	PMG1S3 Debug Header2	Customers can connect a digital oscilloscope to debug digital signals.
J29	CCG7SC Validation header	The developer can connect a digital oscilloscope to debug various CCG7SC controllers' signals.
J27	CAPSENSE™ Slider Addon	Customers can connect a CAPSENSE™ slider to this header.
J30, J31, J33	DEBUG CONNECTORS	These connectors contain a few GPIO's from the CCG7SC made available for probing.

2.2 USB-C hardware system block diagram

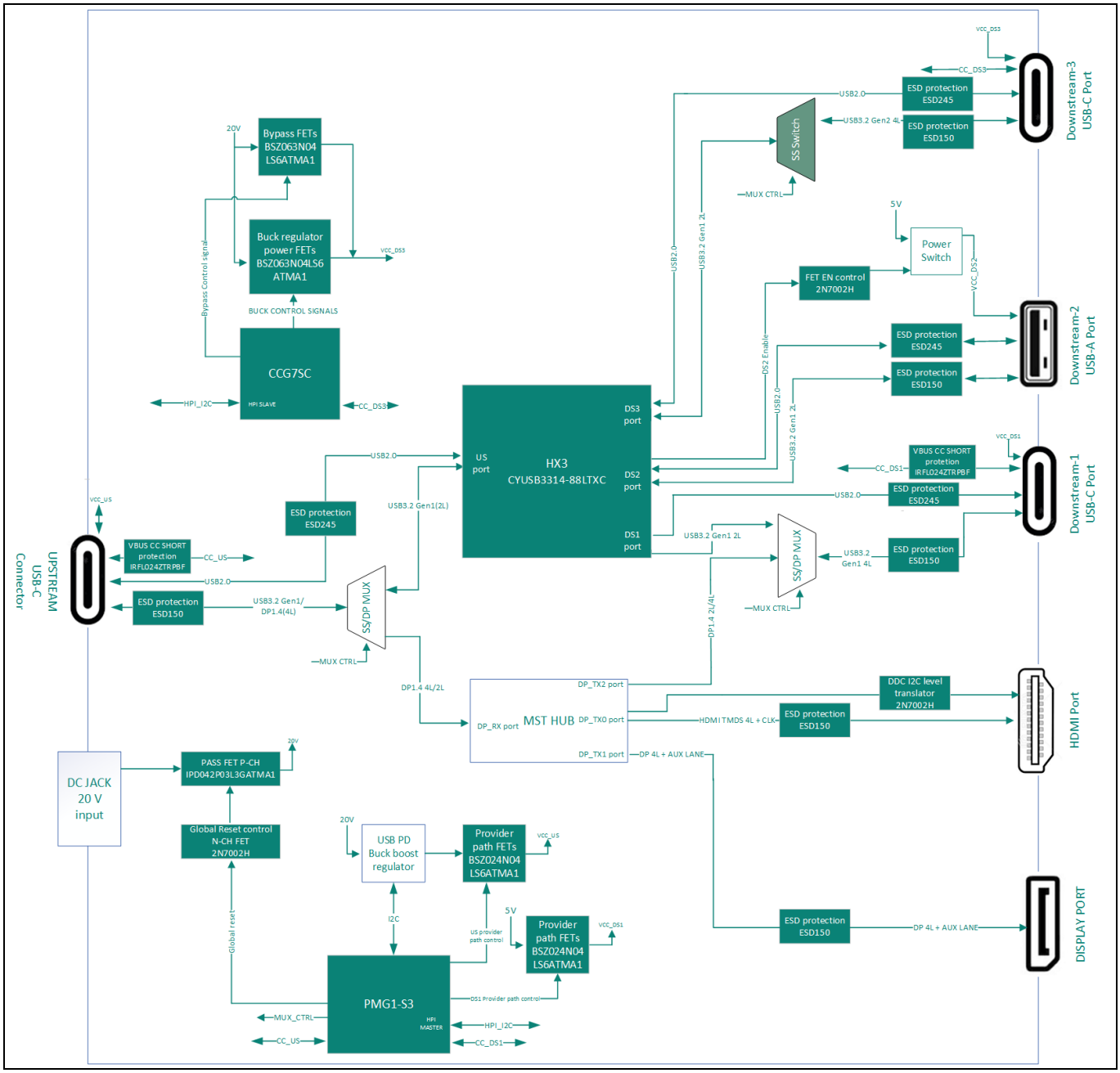


Figure 3 USB-C hardware system block diagram

2.3 Features

The following is a list of features supported by the EZ-PD™ PMG1-S3 USB-C Dock reference design hardware:

Upstream Type-C port:

- Controlled by EZ-PD™ PMG1-S3
- Controls Analogix's ANX7443 US Mux, based on Type-C polarity
- Supports USB SuperSpeed (5 Gbps) and USB2 data using the EZ-USB™ HX3 USB 5 Gbps hub controller
- Supports DP ALT Mode over Type-C for DP Sink applications
- US current measurement using the current sensor
- Capable of advertising source PDO's up to 28 V at 5 A
- Uses MPS4247 as a buck-boost converter

DS1 Type-C port:

- Controlled by EZ-PD™ PMG1-S3
- Controls Pericom's PI3USB31532 Mux, based on Type-C polarity
- Support USB SuperSpeed (5 Gbps) and USB2 data using the EZ-USB™ HX3 USB 5 Gbps hub controller
- Supports DP ALT Mode over Type-C for DP source applications. DP data is controlled by the Synaptics VMM5320 DP hub controller
- Charging ability when US is not attached to the notebook, with 5 V at 3 A source PDO

DS2 Type-A port:

- Configured to charge devices up to 1 A, power is controlled by the EZ-USB™ HX3 USB 5 Gbps hub controller
- Support USB SuperSpeed (5 Gbps) and USB2 data using the EZ-USB™ HX3 USB 5 Gbps hub controller

DS3 Type-C port:

- Controlled by EZ-PD™ CCG7SC
- External Mux control based on Type-C polarity
- Support USB SuperSpeed (5 Gbps) and USB2 data using the EZ-USB™ HX3 USB 5 Gbps hub controller
- Ability to charge irrespective of US connection, with 5 V at 3 A and 9 V at 3 A source PDOs

EZ-USB™ HX3's DS4 port:

- EZ-PD™ PMG1-S3 enumerates as a USB2 device:
  - Connected as a non-removable device to the DS4 of the EZ-USB™ HX3 USB 5 Gbps hub controller
  - Presents a USB Billboard interface to notify the host PC of ALT Mode failures

HDMI and DP:

- HDMI and DP ports, which are controlled by the Synaptics VMM5320 DP hub controller

System:

- SW1 and SW2 push-button switches
- Current sensor to measure US and total dock current
- Dock add-on connectors for plug and play of a variety of buck-boost converters

3 EZ-PD™ PMG1-S3 USB-C Dock application examples

3.1 Prerequisite

Follow these steps for the use-case examples provided in the following subsequent sections.

1. Download the pre-built binaries for the PMG1-S3 USB-C Dock solution.
2. Set the default jumper setting provided as per Table 1.
3. Set the J23, J24, and J25 jumper settings as per Table 2, and supply power through CON2 or J2 accordingly.
4. Refer to [1], which covers the steps to be followed for SWD programming PMG1-S3. The same steps can be followed for SWD programming CCG7SC. However, customers are required to choose Platform as CCGx from the mtb-programmer tool. See Programming and debugging section for more details.

3.2 Charging a notebook from Dock's upstream port [SPR example]

This use case is an example of advertising SPR PDO's on the US port:

1. Follow the steps provided in the Prerequisite section.
2. Set J23, J24, and J25 settings to 120 W power adaptor.
3. Supply a 120 W power supply to CON2 or J2.
4. Connect a 3 A or 5 A capable Type-C to Type-C cable between the EZ-PD™ PMG1-S3 USB-C Dock J7 and the notebook's Type-C PD port.

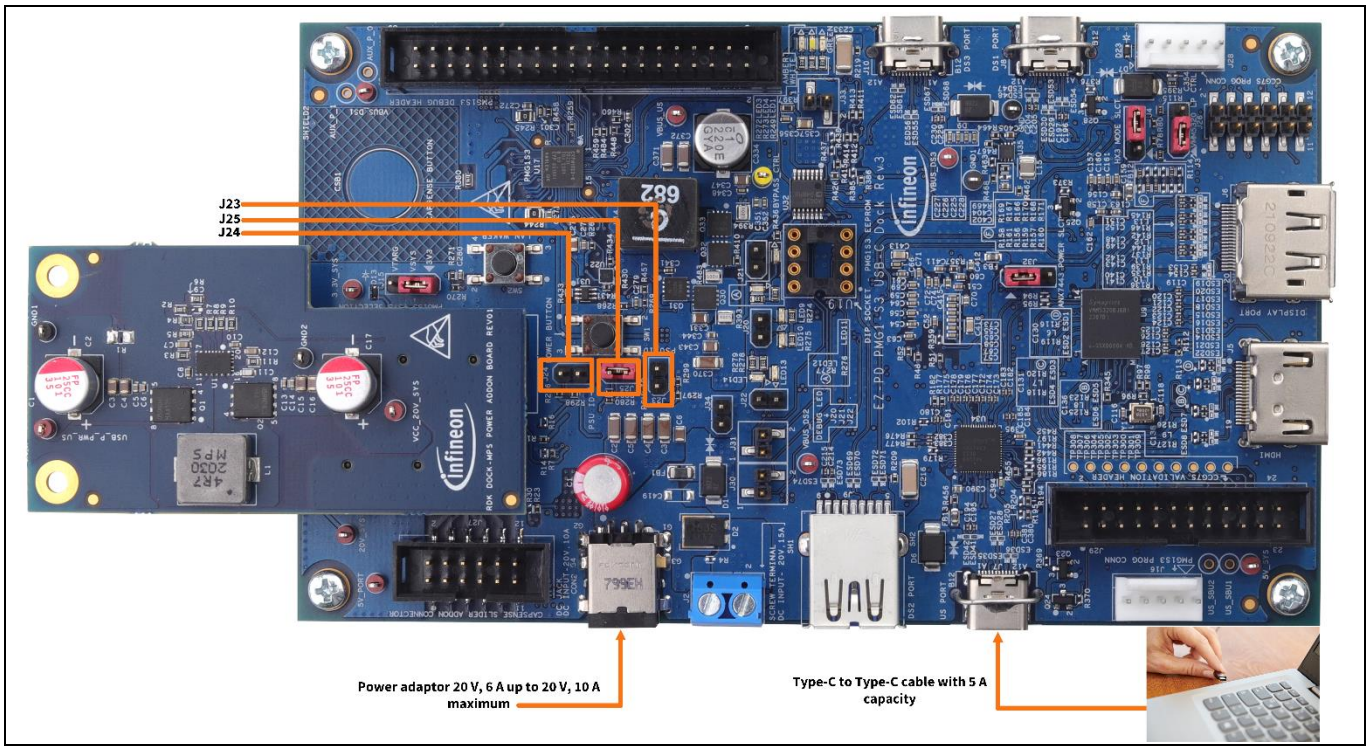


Figure 4 EZ-PD™ PMG1-S3 USB-C Dock US port connected to notebook

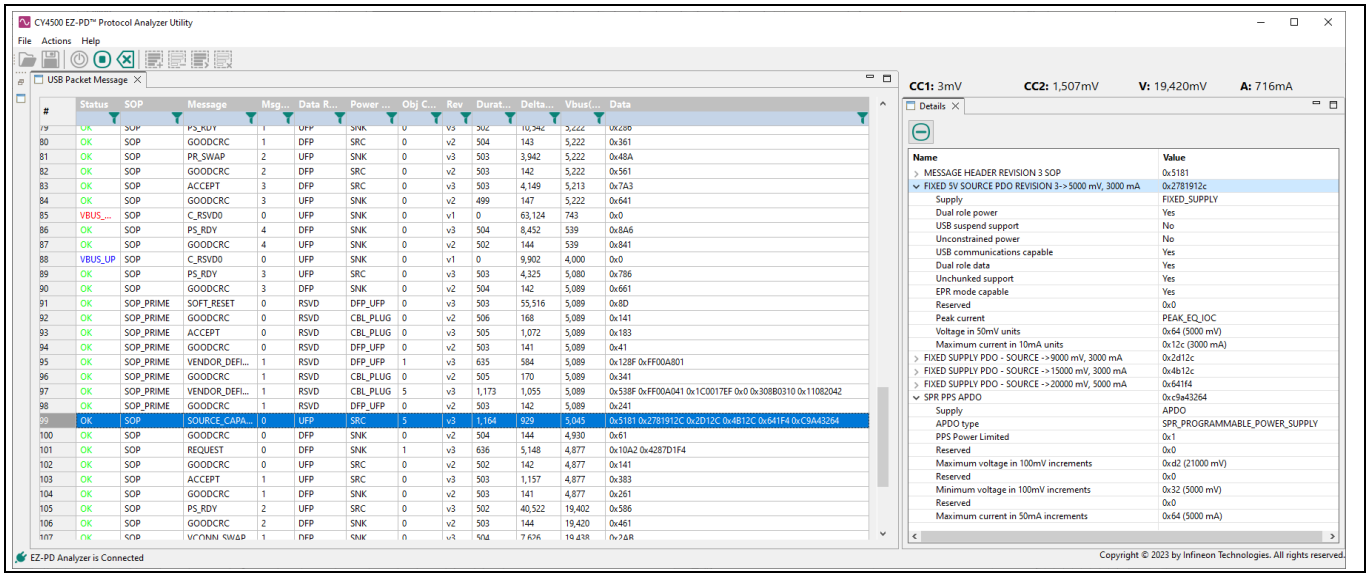


Figure 5 Source PDO's being advertised by US port with 5 A cable and with a 120 W power adaptor setting

3.3 EPR source example

1. Follow the steps provided in the Prerequisite section.
2. Make sure the power jumper settings J23, J24, and J25 are set to greater or equal to 150 W power input as per Table 2.
3. Connect the power supply to CON2 or J2 accordingly.
4. Power on the USB-C Dock hardware.
5. Connect an EPR-capable sink device or EPR-capable notebook to the US port of the USB-C Dock using an EPR-capable Type-C to Type-C cable.

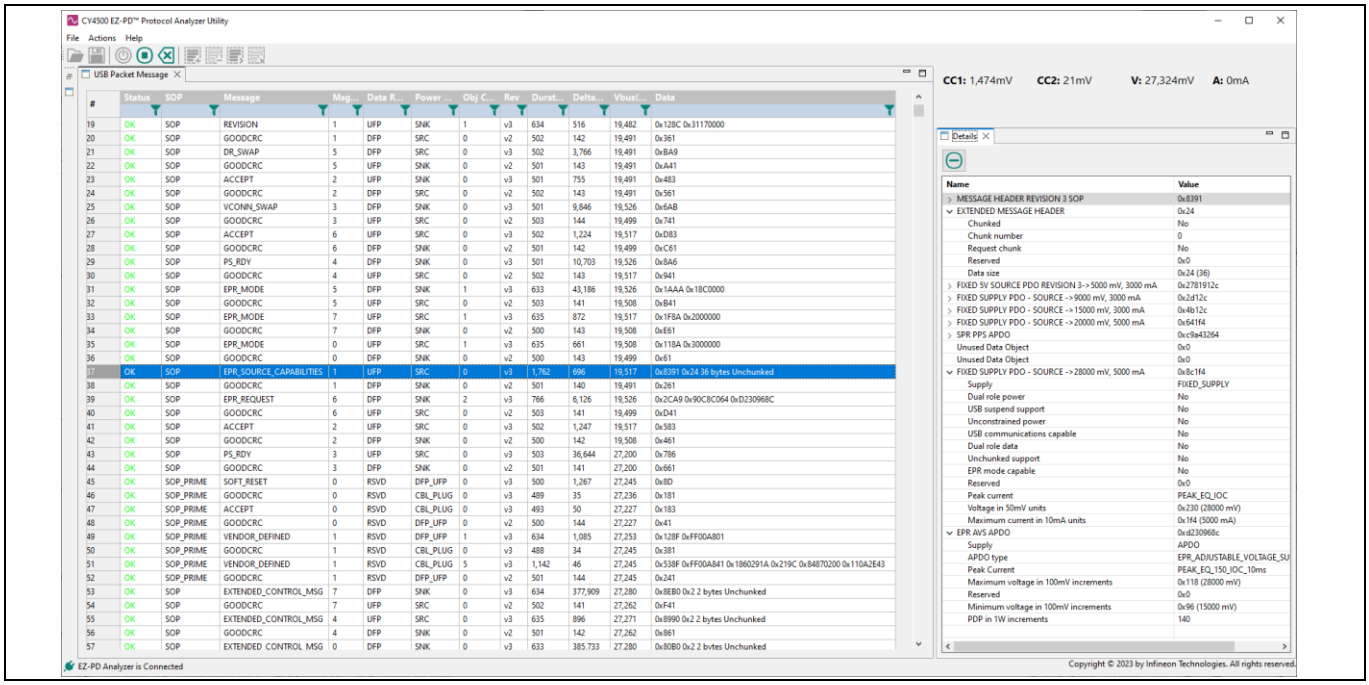


Figure 6 EZ-PD™ analyzer log showing EPR PDO's and contract made at 28 V when CY7113 EZ-PD™ PMG1-S3 prototyping kit is connected

3.4 USB data and display mirroring or extension using downstream ports

1. Follow the steps provided in the Prerequisite section.
2. Connect PMG1-S3 US port J7 to a notebook PC using a Type-C to Type-C cable.
3. Connect Type-C, HDMI, and DP monitors to DS1, HDMI, and DP ports, respectively.
4. Connect a Type-A-based SSD drive or pen drive to the Type-A port and a Type-C-based mobile device or tablet to the DS3 port.

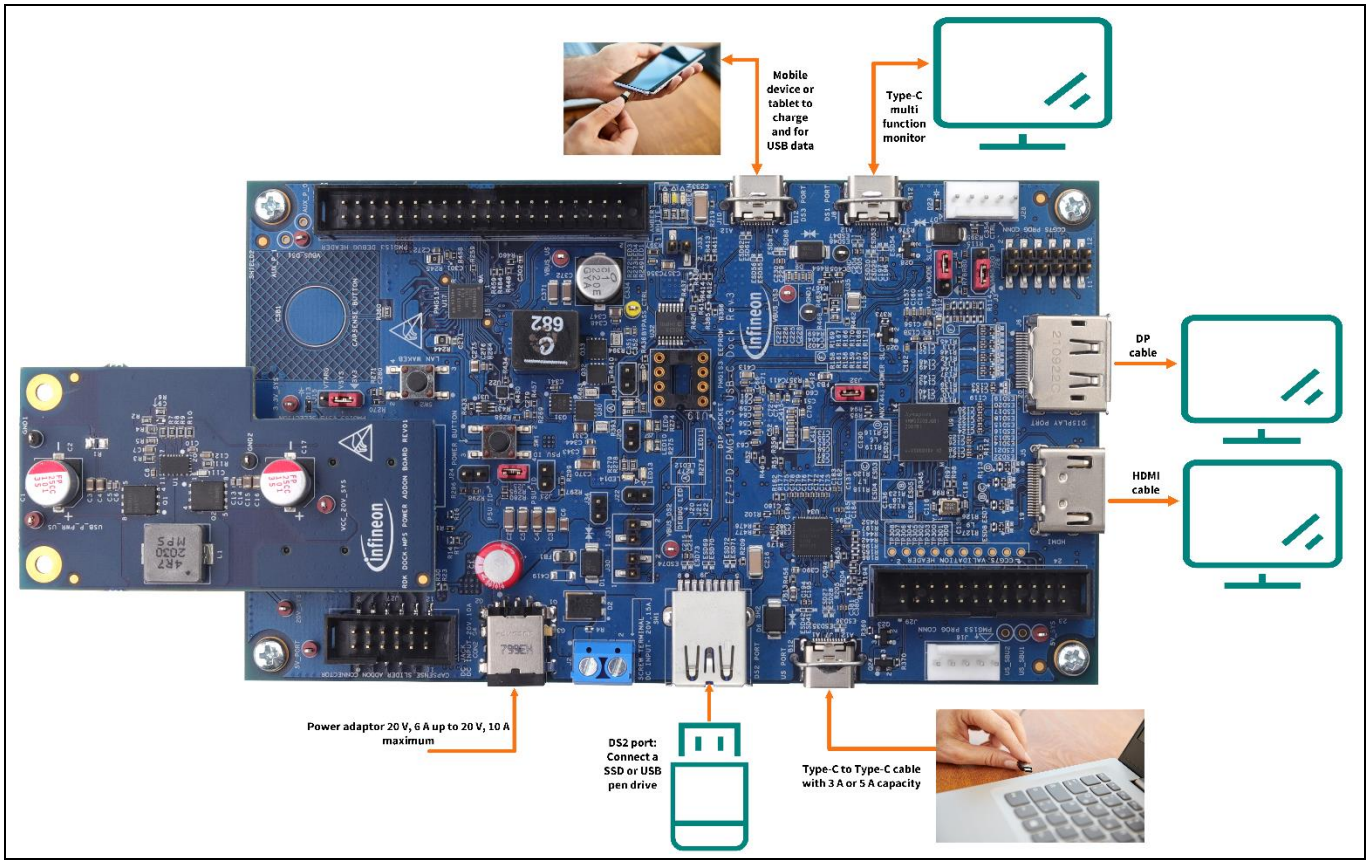


Figure 7 Dock with all downstream and upstream ports connected

3.5 Charging on DS1 and DS3 without US connection

1. Follow the steps provided in the Prerequisite section.
2. Connect the mobile device to the DS1 and/or DS3 ports.
3. The user can observe that DS3 (controlled by CCG7SC) can advertise up to 27 W, whereas DS1 (controlled by PMG1-S3) can advertise 15 W.

EZ-PD™ PMG1-S3 USB-C Dock application examples

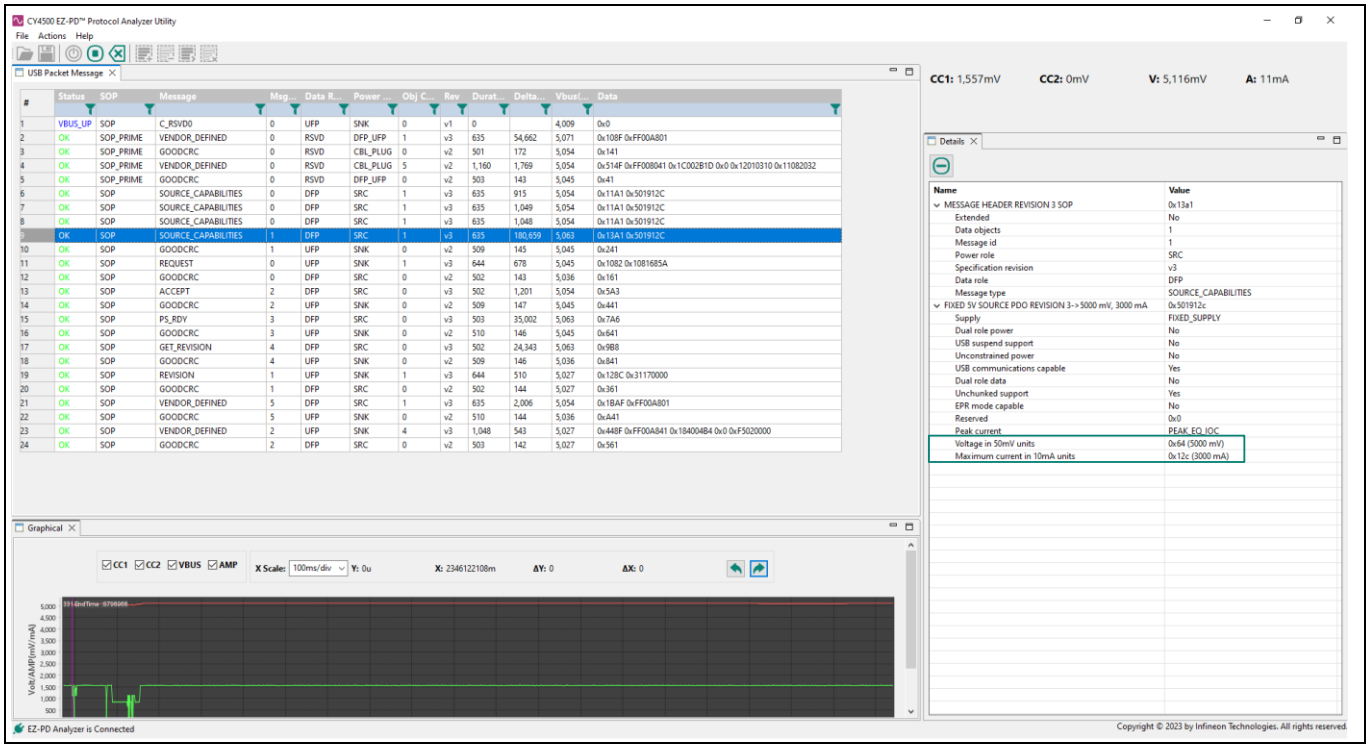


Figure 8 Source PDO advertised on DS1

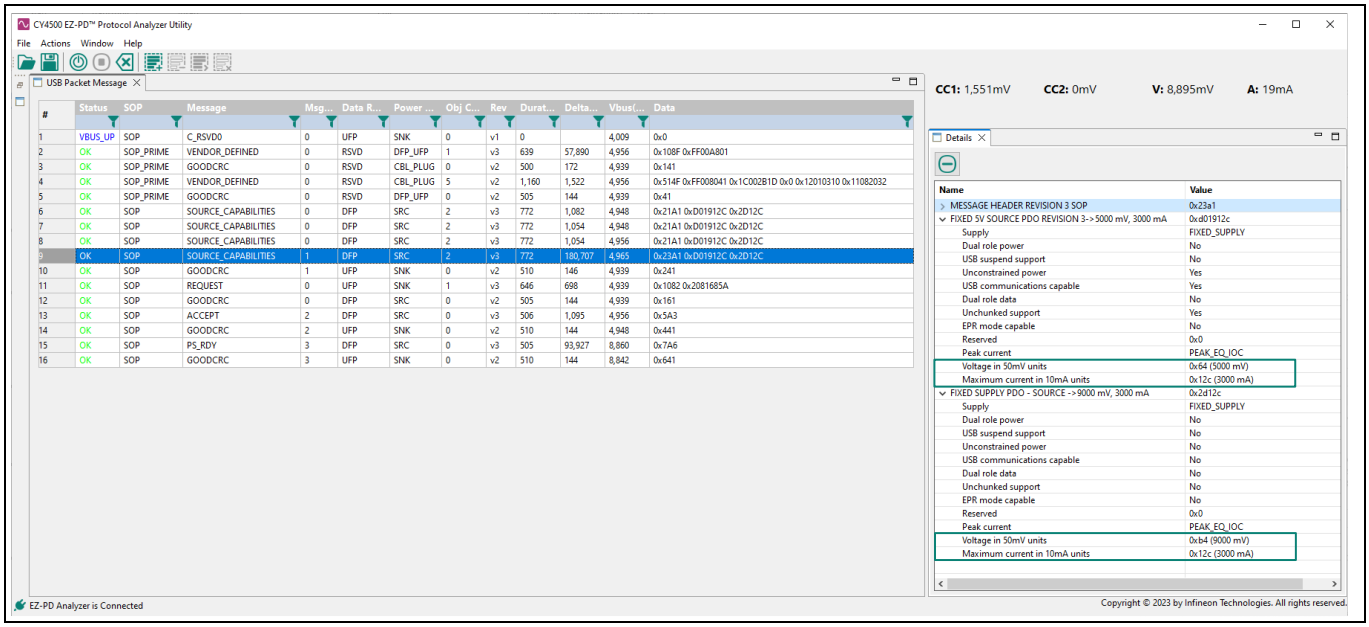


Figure 9 Source PDO's advertised by CCG7SC on DS3 port are up to 27 W

4 Griffin Creek Thunderbolt Dock

4.1 Griffin Creek dock hardware

Figure 10 and Figure 11 show the top and bottom views of the Griffin Creek Dock hardware.

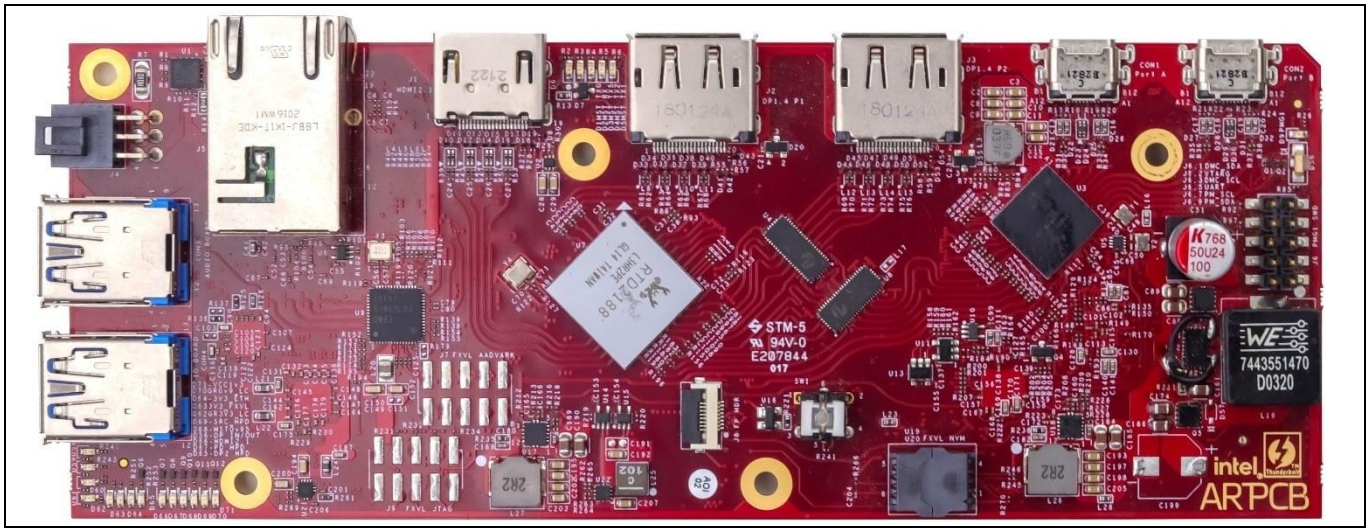


Figure 10 Griffin Creek Dock top view

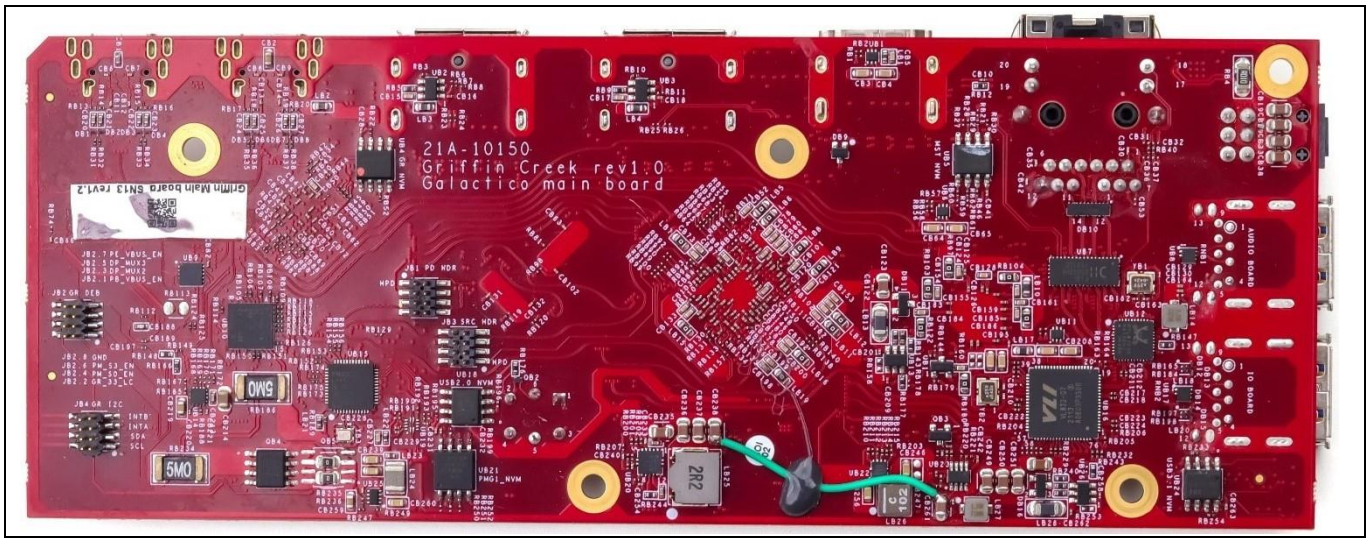


Figure 11 Griffin Creek Dock bottom view

4.2 Connectors and Jumper settings

Table 6 Griffin Creek Dock hardware connectors and jumper details

Reference	Connector type	Description
CON1	Type-C connector	Port-A: US port of the dock, which is controlled by PMG1-S3 and acts as UFP and is used to connect to the notebook.
CON2	Type-C connector	Port B: The DS1 port of the dock, which is controlled by PMG1-S3, typically acts as DFP, and is used to connect to Type-C devices.
J4		Supply a minimum of 150 W power to observe EPR PDO's. If a higher or lower adaptor is being used, it is required to modify the PMG1S3 TBT Griffin Creek Dock Solution code example and use the generated binaries for testing. Alternatively, customers can modify the **Adaptor power** parameter under **Smart Power** options using the EZ-PD™ Dock Configuration Utility.
J6		PMG1-S3 SWD header. Use Miniprog4 Programmer/Debugger to SWD program the prebuilt or newly generated hex file. See Griffin Creek Dock's PMG1-S3 SWD programming section to SWD program PMG1-S3 on Griffin Creek Dock hardware.

4.3 Features

Upstream Type-C port (port-A):

- SPR range PDO's up to 100 W, SPR PPS PDO 5 V – 21 V at 5 A, and EPR PDO up to 140 W
- Sink PDO: 5 V at 0 A
- Supports USB4 Gen3 speeds; supports DP and TBT alternate modes
- Backward compatible with USB3 and USB2
- VBUS OVP, OCP, UVP, and VCONN fault protections

Downstream Type-C port (port-B):

- 15 W source PDO
- Supports USB4 Gen3 speed, DP, and TBT alternate modes
- Backward compatible with USB3 and USB2
- VBUS OVP, OCP, UVP, and VCONN fault protections

Billboard:

- Supports USB Billboard functionality to indicate ALT mode entry failures

Firmware update:

- Supports signed and unsigned firmware updates, customers are advised to use the signed firmware update for production releases. Note that signed-to-unsigned firmware update rollback is not supported.
- Dedicated SPI flash to store dock composite image
- Firmware update to PMG1-S3, CCGx devices, Goshen Ridge controller, and Intel's Foxville Ethernet controller
- Firmware recovery from SPI flash in case of power loss during firmware update

4.4 Griffin Creek Dock's PMG1-S3 SWD programming

To SWD program the EZ-PD™ PMG1-S3 MCU:

1. Customers must download pre-built binaries for Griffin Creek Dock from the PMG1-S3 Dock webpage.
2. Additionally, customers can create a ModusToolbox™ based example using PMG1S3 TBT Griffin Creek Dock Solution template by referring to [1] and use the generated hex file.
3. Miniprog4 is to be connected to the J6 connector available on the hardware, which is shown in Figure 12.
4. As the Miniprog4 cannot be plugged directly into the J6 connector, one needs to jump wire between the Miniprog4 and J6 as given in Table 7.



Figure 12 SWD header to program PMG1-S3 using Miniprog4 Programmer/Debugger

Table 7 J6 to Miniprog4 jumper wire

Griffin Creek	Miniprog4
J6.2	VTARG
J6.4	GND
J6.6	XRES
J6.8	SWCLK
J6.10	SWDIO

Customers should contact Intel for the Griffin Creek hardware, Goshen Ridge controller firmware, Foxville Ethernet controller firmware, and steps to be followed for pre-programming them.

5 CCG7SC

The CCG7SC is used to control the downstream (DS3) Type-C port on the EZ-PD™ PMG1-S3 USB-C Dock. The CCG7SC is configured for buck-only mode to advertise source PDO's up to 27 W. For the EZ-PD™ PMG1-S3 USB-C Dock solution, pre-built binaries are made available from the Infineon PMG1-S3 Dock webpage.

Using the EZ-PD™ Configuration Utility, the configuration parameters of pre-built binaries can be modified; see EZ-PD™ Configuration Utility section to learn about the tool.

5.1 I2C slave address

The CCG7SC implements an I2C slave interface. The CCG7SC's bootloader and application image use 0x08 as a 7-bit I2C slave address.

5.2 Data Mux control

The EZ-PD™ PMG1-S3 USB-C Dock uses FUSB340 as a data switch to control the USB3 SuperSpeed lanes from the Type-C connector on the downstream DS3 port. Details about GPIO's used from CCG7SC are provided in Table 8.

Table 8 CCG7SC's GPIO control for DS3 Data Mux

CCG7SC's GPIO	Type-C connect event on DS3 port	Type-C disconnect event on DS3 port
P3.0	• Drive P3.0 low for Type-C unflipped orientation. • Drive P3.0 high for Type-C flipped orientation.	P3.0 is not controlled by the Type-C disconnect.
P3.1	Drive P3.1 to high.	Drive P3.1 to low.

Customers are advised to contact Infineon Support for any customization on CCG7SC.

6 Customizing the EZ-USB™ HX3 Hub

Customers can customize the HX3 hub parameters for their designs using the Blaster Plus Configuration Utility. The utility is available under the Getting started section of the EZ-USB™ HX3 Hub controller web page.

7 EZ-PD™ Dock Configuration Utility

The EZ-PD™ Dock Configuration Utility enables customers to customize EZ-PD™ PMG1-S3 PD Port-0, Port-1, PMG1-S3 USB device, and Dock topology parameters.

7.1 Prerequisite

Download the EZ-PD™ Dock Configuration Utility setup and install it on the host PC. In the Windows Start menu, search for EZ-PD™ Dock Configuration Utility and launch it, which appears as shown in Figure 13.

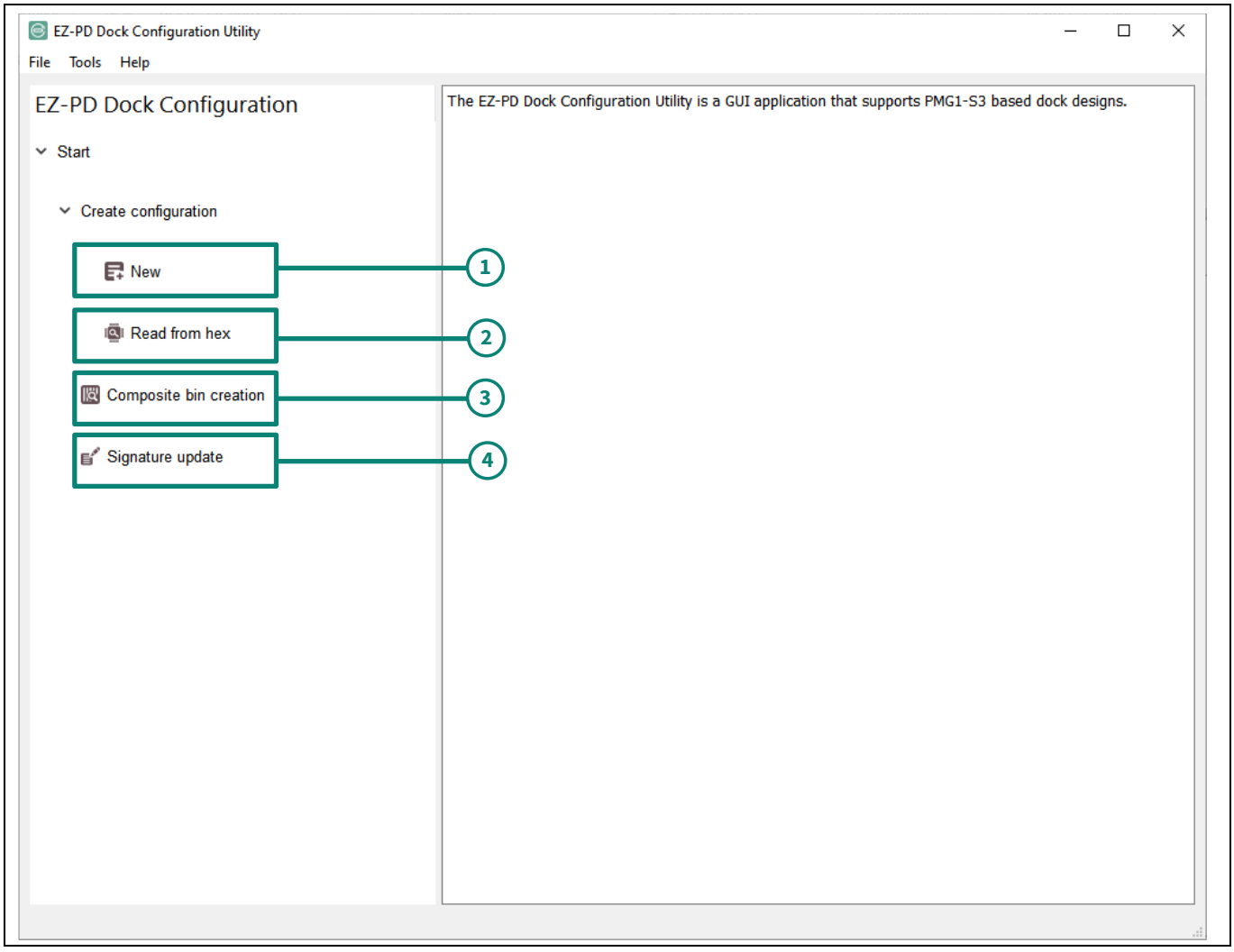


Figure 13 EZ-PD™ Dock Configuration Utility

The tool can be used for:

1. Creating a new dock configuration.
   - The generated configuration '*config.c*' file can be replaced when the ModusToolbox™ application is created for PMG1S3 USB-C Dock Solution or PMG1S3 TBT Griffin Creek Dock Solution.
2. Loading the configuration by reading from a hex file.
   - Modify and save the configuration to a hex file. The modified hex file can be SWD programmed using mtb-programmer.

  - Modify and save the configuration to cyacd2 files. The modified cyacd2 files can be used to create a composite bin image, which can be programmed using the EZ-PD™ FwUpdateUtility. This avoids SWD programming of PMG1-S3 and CCG7SC present on the dock.
3. Composite bin creation, the tool supports dock components such as PMG1-S3, CCGx devices, Intel's Goshen Ridge controller, and Foxville controller. For any third-party device support, contact Infineon Support.
4. Signature update to composite bin when signed update support is enabled.

7.2 Modifying PMG1-S3 VID and PID

Follow the steps provided in the Prerequisite section.

Launch the EZ-PD™ Dock Configuration Utility. Using the **Read from hex** option, load the EZ-PD™ PMG1-S3 USB-C Dock's hex file and follow these steps:

1. Go to the **Port-0** tab.
2. Click on **Port information**.
3. Modify the **Manufacturer vendor ID (0x)** and the **Manufacturer product ID (0x)**.
4. Click on the save icon, which opens the **Update configuration** window.
5. Provide the configuration file path.
6. Choose the **Both images** check box to update the configuration to FW1 and FW2.
7. Provide the hex file path to which the new configuration is to be updated.
8. Provide the FW1 cyacd2 file path to which the new configuration is to be updated.
9. Provide the FW2 cyacd2 file path to which the new configuration is to be updated.
10. Click **Update**.

SWD program the newly saved hex file using mtb-programmer and power cycle the dock. Verify in device manager to see if the PMG1-S3 on the USB-C dock can be enumerated with the new VID and PID.

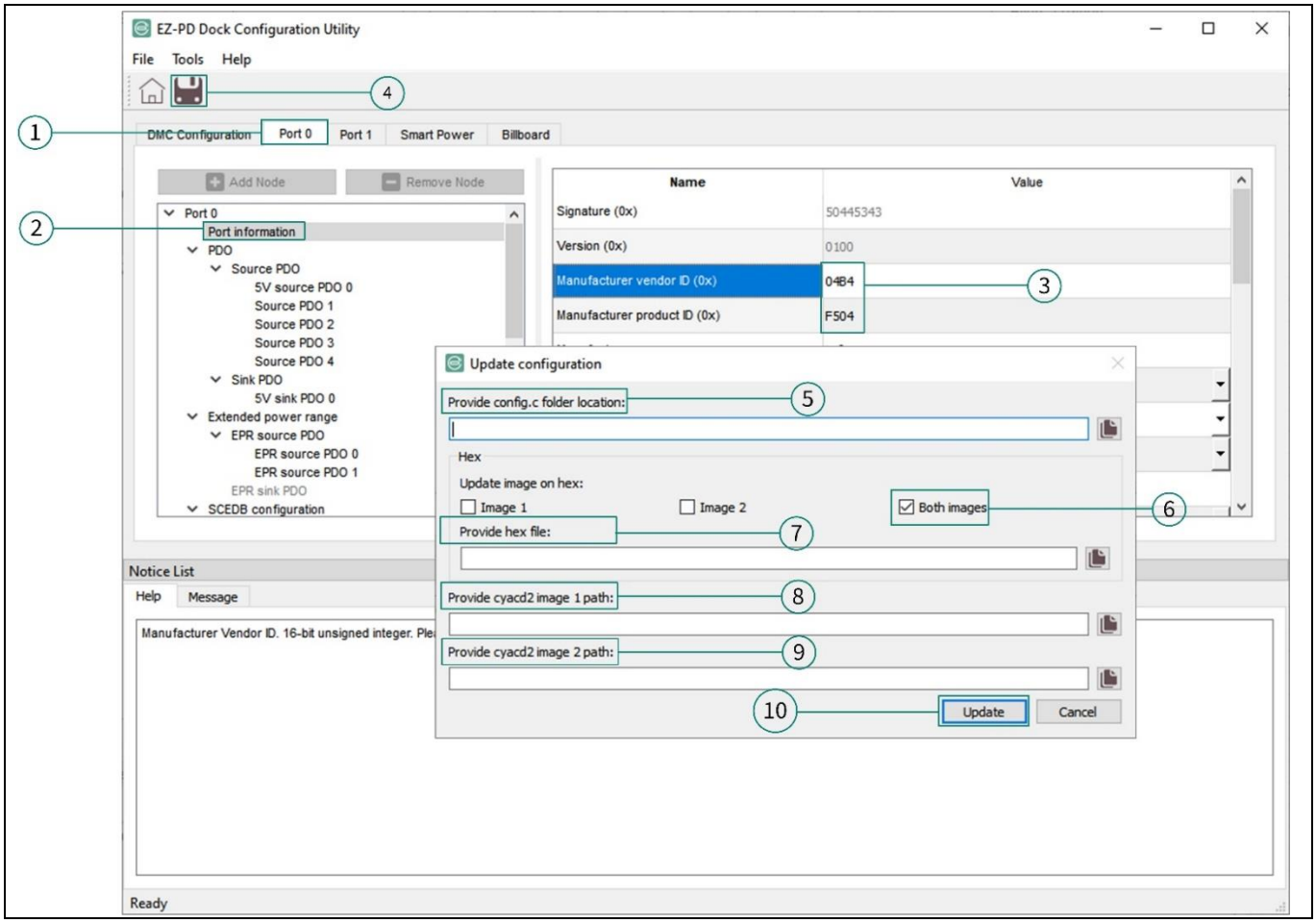


Figure 14 Modify VID and PID example

7.3 Enable/Disable Smart Power

1. Launch the EZ-PD™ Dock Configuration Utility.
2. Using the read from hex option, load the EZ-PD™ PMG1-S3 USB-C Dock's hex file.
3. Go to the **Smart Power** tab.
4. Set **Smart power support** to **Yes** or **No** to enable or disable the smart power feature. When smart power support is enabled, choose other parameters accordingly.
5. Click on the save icon, which opens the **Update configuration** window.
6. Provide the configuration file path.
7. Choose the **Both images** check box to update the configuration to FW1 and FW2.
8. Provide the hex, FW1, and FW2 cyacd2 file paths to which the new configuration is to be updated.
9. SWD program the newly saved hex file using mtb-programmer and power cycle the dock.
10. Customers can see that there is no change in PDO's being advertised based on power consumption on DS ports.

7.4 Adding a new devices component or removal of a component for firmware update support

1. Open the EZ-PD™ Dock Configuration Utility.
2. Using the read from hex option, load the PMG1-S3 USB-C Dock's hex file.
3. Select the **Devices** from the **DMC Configuration** tab, under **CDTT parameters**.
4. Choose **Add Node** to add a new device.
5. The tool opens a window to enter **Device Type**, **Image Mode**, and **Row Size**, which are described in [1].
6. See Table 9 for device types supported by the EZ-PD™ Dock Configuration Tool.
7. Contact Infineon Support about adding a new component to the dock.

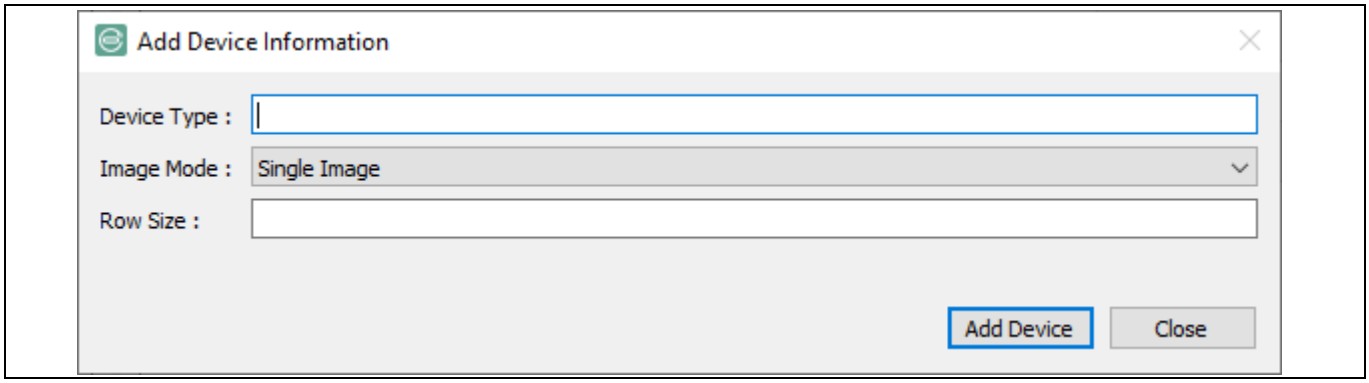


Figure 15 Add Device Information

8. When a component is not present in the dock designed by customers, select the component and click on **Remove Node**.
   a) For example, remove the CCG7SC node for the PMG1-S3 USB-C Dock solution.
   b) Save the configuration to the EZ-PD™ PMG1-S3 USB-C Dock solution pre-built binary.
   c) SWD program the hex to the EZ-PD™ PMG1-S3 USB-C Dock.
   d) When the Dock status query command is executed using *EZPD.FwUpdateUtility*, the utility output appears as shown in Figure 16.

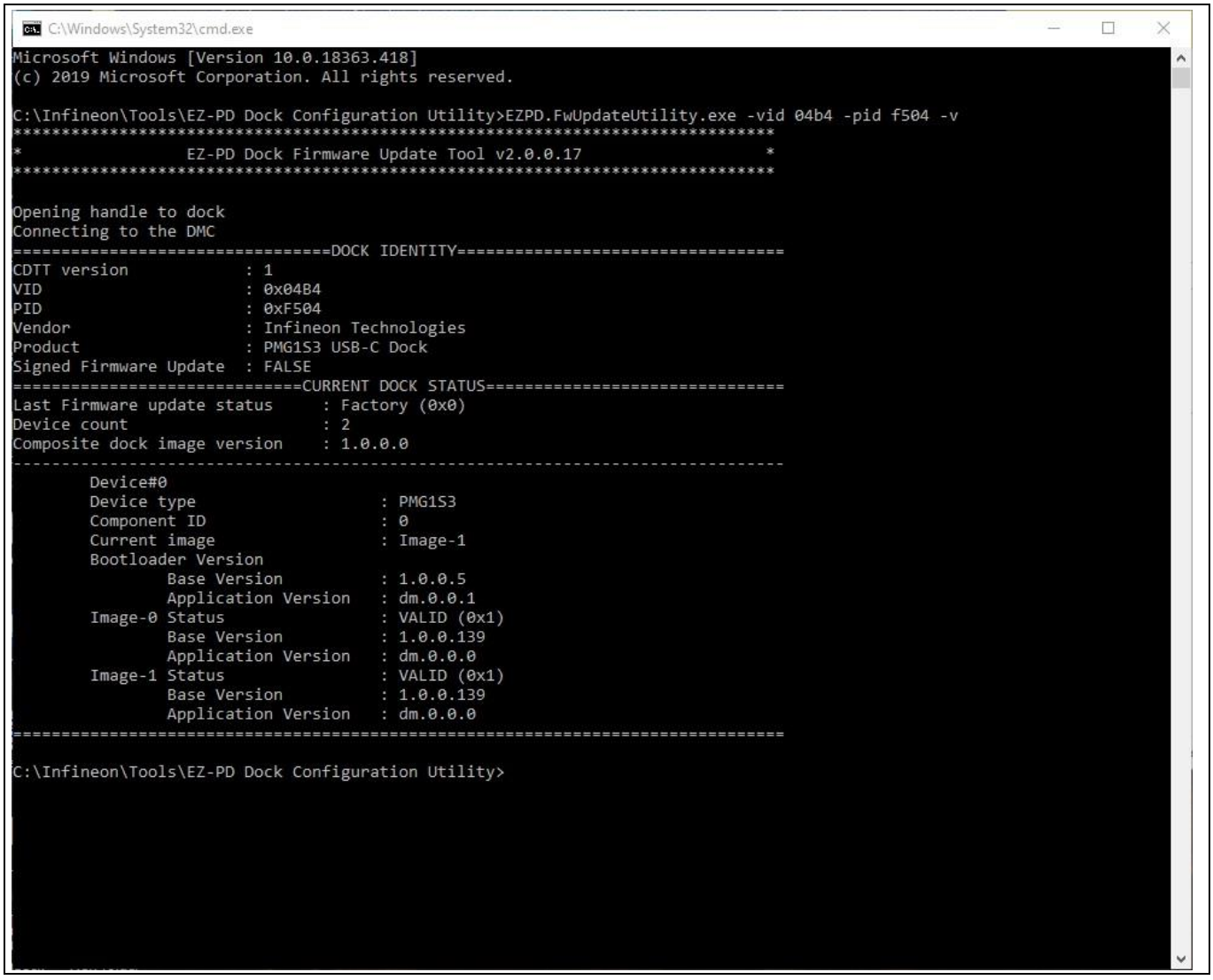


Figure 16 Dock status query after removing CCG7SC from USB-C Dock solution binaries

9. It is mandatory to have PMG1-S3 and SPI flash as components in the dock. The SDK supports firmware updates for up to eight devices; therefore, customers can add six more components.

Table 9 Device type values for adding a new component to CDTT

Device type value	Description
3	CCG4
4	CCG5
5	HX3
29	Goshen Ridge controller
30	Foxville
240	PMG1-S3
241	CCG7SC
255	SPI flash

7.5 Enabling signed firmware update

Customers can generate an RSA-2048 asymmetric key pair, which is outside the scope of this document. The public key file will be in DER format.

1. Launch the EZ-PD™ Dock Configuration Utility. Using the read from hex option, load the EZ-PD™ PMG1-S3 USB-C Dock's hex file.
2. Select the **DMC Configuration** tab.
3. Select **CDTT parameters** and **Dock information**.
4. Change the firmware update mode from **Unsigned** to **Signed**.
5. Select the **Public key**, modify the **Public key length enable** value to **Yes**.
6. Click on the **File path** to provide the public key path, which is in DER format.
7. Note that the tool displays files that are in text format. Therefore, customers are required to type the filename with the **der** extension.
8. As shown in Figure 17, when **.der* is typed, the file explorer shows the files with the der extension. Select the required public key file and click **Open**.



Figure 17 Selecting new public key with der extension

8 Dock firmware update

The Dock firmware update feature allows ODMs or OEMs to update Dock's components in production. EZ-PD™ PMG1-S3 Dock SDK supports firmware updates to PMG1-S3, Infineon's CCGx controllers, and writing the composite binary into SPI flash connected to PMG1-S3.

In addition, it can also update firmware for third-party controllers. For example, PMG1-S3 can update firmware for Intel's Goshen Ridge controller and Foxville Ethernet controller over I2C for Griffin Creek Dock.

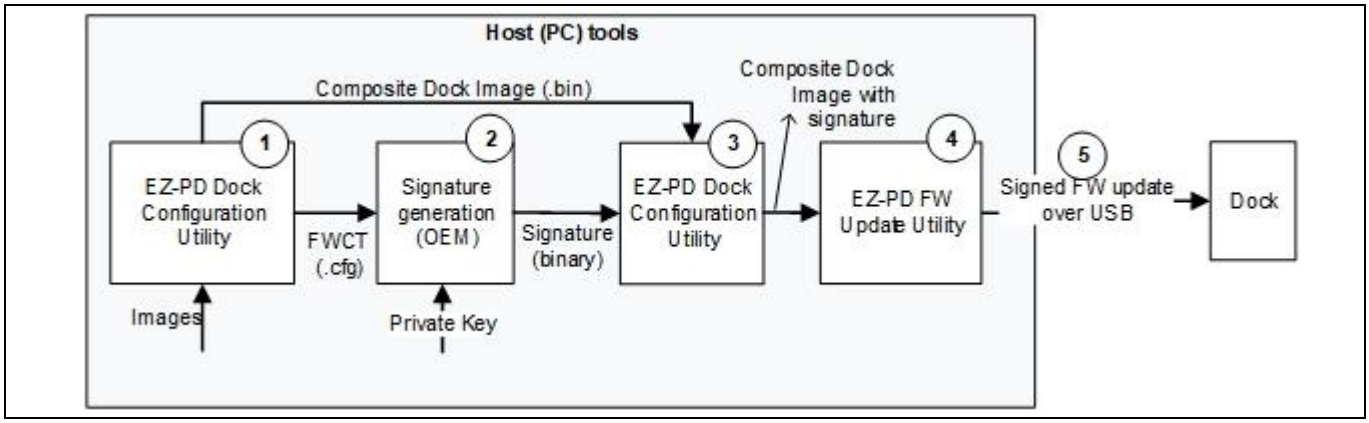


Figure 18 Steps from composite binary creation to Dock phase-1 update

The firmware update is performed in two phases:

1. Phase-1 update:
   a) Downloading the composite binary using Infineon's *EZPD.FwUpdateUtility* from the host PC to dock over the EZ-PD™ PMG1-S3's USB 2.0 vendor interface, as shown in Figure 18. On the host PC, the *EZPD.FWUpdateUtility* uses the WinUSB Driver to send data over USB.
   b) If the firmware is operating in signed mode, it verifies the integrity of the image being downloaded using the SHA-256 and RSA-2048 Crypto algorithms.
   c) Updating the composite binary to SPI flash connected to PMG1-S3.
2. Phase-2 update:
   a) PMG1-S3 queries the connected components to determine if a firmware update is needed.
   b) The composite image on SPI flash is verified for integrity.
   c) Update firmware for PMG1-S3 and components that are connected to the PMG1-S3 on the dock hardware.
   d) For example, in the case of the USB-C dock, firmware updates are supported for the CCG7SC. Whereas for the Griffin Creek Dock, firmware updates are supported for the Goshen Ridge and Foxville controllers.

Follow these steps to perform the signed firmware update:

1. Update the public key in the PMG1-S3's cyacd2 images as described in the Enabling signed firmware update section.
2. Create a composite binary with individual firmwares of each updatable component by PMG1-S3.
3. Sign the Firmware Configuration Table (FWCT) data with a private key using third-party tools, for example, OpenSSL. The key pair generation, storing, and signing of the FWCT file using a private key will be owned by the ODM or OEM.
4. Update the signature to the composite binary in the case of signed firmware update support.

5. Perform the Phase-1 update to the dock using the *EZPD.FwUpdateUtility.exe*, which will be installed as part of the EZ-PD™ Dock Configuration Utility.

The customer can use *EZPD.FwUpdateUtility.exe* to query the overall firmware update completion status and the firmware update completion status for individual components.

8.1 Prerequisite

Install the EZ-PD™ DOCK Configuration Utility and launch it from the Windows menu after installation is complete.

8.2 Dock status

To query the dock status, which displays the firmware of each device that can be updated by PMG1-S3, it is displayed in the command line.

Usage command: *EZPD.FwUpdateUtility.exe -v -vid <PMG1-S3 Device VID in hex> -pid <PMG1-S3 Device PID in hex>*

EZ-PD™ PMG1-S3 USB-C Dock uses 0x04B4 and 0xF504 as VID and PID. The following is an example command:

EZPD.FwUpdateUtility.exe -v -vid 04B4 -pid F504

```
C:\Infineon\Tools\EZ-PD Dock Configuration Utility>EZPD.FwUpdateUtility.exe -v -vid 04b4 -pid f504
*************************************************************************
*                EZ-PD Dock Firmware Update Tool v2.0.0.17                *
*************************************************************************

Opening handle to dock
Connecting to the DMC
================================DOCK IDENTITY================================
CDTT version            : 1
VID                     : 0x04B4
PID                     : 0xF504
Vendor                  : Infineon Technologies
Product                 : PMG1S3 USB-C Dock
Signed Firmware Update  : FALSE
============================CURRENT DOCK STATUS==============================
Last Firmware update status     : Factory (0x0)
Device count                    : 3
Composite dock image version    : 1.0.0.0
-------------------------------------------------------------------------------
        Device#0
        Device type                     : PMG1S3
        Component ID                    : 0
        Current image                   : Image-1
        Bootloader Version
                Base Version            : 1.0.0.5
                Application Version     : dm.0.0.1
        Image-0 Status                  : VALID (0x1)
                Base Version            : 1.0.0.135
                Application Version     : dm.0.0.0
        Image-1 Status                  : VALID (0x1)
                Base Version            : 1.0.0.135
                Application Version     : dm.0.0.0
-------------------------------------------------------------------------------
        Device#2
        Device type                     : CCG7SC
        Component ID                    : 2
        Current image                   : Image-0
        Bootloader Version
                Base Version            : 4.0.1.41
                Application Version     : pa.0.0.0
        Image-0 Status                  : VALID (0x1)
                Base Version            : 4.0.1.20
                Application Version     : pa.0.0.0
===============================================================================
```

Figure 19 Dock status query sample log for EZ-PD™ PMG1-S3 USB-C Dock solution with pre-built binary programmed for PMG1-S3 and CCG7SC

8.3 Dock Composite bin creation

This example uses PMG1-S3's cyacd2 files, which are updated with the public key as instructed in the Enabling signed firmware update section. Select the **Composite bin creation** option from the EZ-PD™ Dock Configuration Utility after launching it.

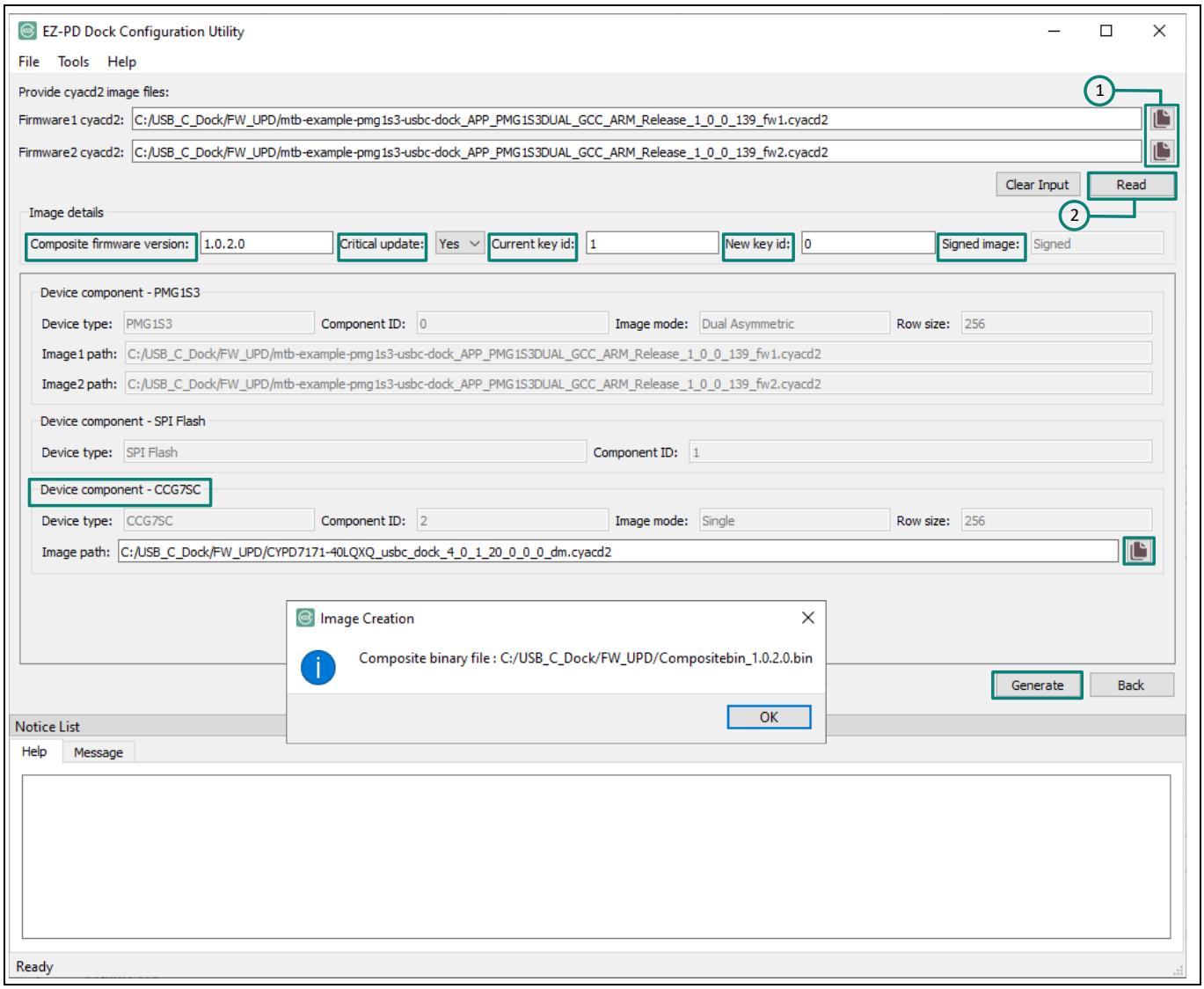


Figure 20 Composite bin creation

1. Load the PMG1-S3's FW1 and FW2 cyacd2 files.
2. Click on the **Read** button to read the configuration from the firmware by the tool.
3. EZ-PD™ Dock Configuration Utility displays the **Signed image** value as **Signed** or **Unsigned** based on whether PMG1-S3 supports signed or unsigned firmware update. The tool also lists the device components for which PMG1-S3 can support firmware updates. Customers are required to provide an **Image path** for each component.
4. Customers need to enter the **Composite firmware version** in [***<Major_version_1>.<Major_version_0>.<Minor_version_1>.<Minor_version_0>***] format, the maximum value that can be provided is 99 for each field.

   a) Make sure to enter this value greater than the existing composite dock image version, which can be queried using *EZPD.FWUpdateUtility* as described in the Dock status section.

b) When customers use a composite bin with a lower version, use *-force* as input to *EZPD.FWUpdateUtility* while invoking from the command line.

c) When the Composite firmware version value is specified as 1.0.2.0, the Utility generates the file with the name *Compositebin_1.0.2.0.bin*, as shown in Figure 20.

5. Set the **Critical update** to **Yes** or **No**. When critical update is set to **Yes** after the firmware update is completed on all devices, the composite image will be copied from the primary area of the SPI flash into the factory area of the SPI flash.
6. The following steps are applicable for signed firmware updates only.

   a) The **Current key id** should match the key id used in the PMG1-S3 Dock SDK. Look for the KEY_ID_EMBEDDED value in the PMG1S3 USB-C Dock Solution and the PMG1S3 TBT Griffin Creek Dock Solution. As the value is set to 1 in SDK, customers should use the same value for **Current key id**. If customers use a different value, the Phase-1 firmware update will fail.

   b) The **New key id** can be left empty. This field is to be used when customers need to switch from one key pair to another.

      i. To change the **New key id**, customers are required to create ModusToolbox™ based application using the PMG1-S3 USB-C Dock solution or PMG1S3 TBT Griffin Creek Dock Solution.

      ii. Modify the KEY_ID_EMBEDDED to a new value and generate the hex file.

      iii. Update the new public key to the generated cyacd2 files as described in Enabling signed firmware update section.

      iv. Use the public key updated cyacd2 files for composite bin creation. Provide the **Current key id** value as 1 and the **New key id** with the value assigned to KEY_ID_EMBEDDED while compiling the project from ModusToolbox™.

      v. Generate the signature with an older public key and a new public key. Follow the steps mentioned in the Generating signature section to generate the signature.

      vi. As part of the signature update, signature one file is the signature being generated with an older public key, whereas signature two file is the signature being generated with a new public key.
7. Provide the **Image path** for the device component CCG7SC. As CCG7SC has a single application, the firmware tool provides one entry for the image path. When the component supports dual images, the tool provides image paths for two images.
8. When the user clicks on **Generate**, the tool generates two files: a composite bin with the file naming convention *Compositebin_1.0.2.0.bin* and a configuration file *Compositebin_1.0.2.0.cfg*. The file with the *cfg* extension will be used for generating the signature using a private key.

8.4 Generating and updating the signature

8.4.1 Generating signature

Customers should have their own infrastructure to sign the configuration file *Compositebin_1.0.2.0.cfg* using a private key. The following instructions show how to sign using OpenSSL:

1. *Compositebin_1.0.2.0.cfg*: is the file generated by the EZ-PD™ Dock Configuration Utility from the Dock Composite bin creation section.
2. Signing command: *openssl.exe dgst -sha256 -sign private.pem -out <Output Signature file.sig> Compositebin_1.0.2.0.cfg.*
3. Sample usage command: *openssl.exe dgst -sha256 -sign private.pem -out Compositebin_1.0.2.0.cfg.sig Compositebin_1.0.2.0.cfg.*

   a) *Compositebin_1.0.2.0.cfg.sig* is the output of OpenSSL.

b) Whereas *Compositebin_1.0.2.0.cfg* is the input to OpenSSL.

8.4.2 Updating signature using Dock Configuration tool

1. Launch the EZ-PD™ Dock Configuration Utility and click the **Signature update** option, which is shown in Figure 13.
2. Upload the composite binary file, which is generated from the Dock Composite bin creation section.
3. Upload Signature File 1 and Signature File 2, which are the same for this use case. Select the **Dual Signature** check box and click on **Update Signature**.
4. Now the *Compositebin_1.0.2.0.bin* will be updated with the signatures.

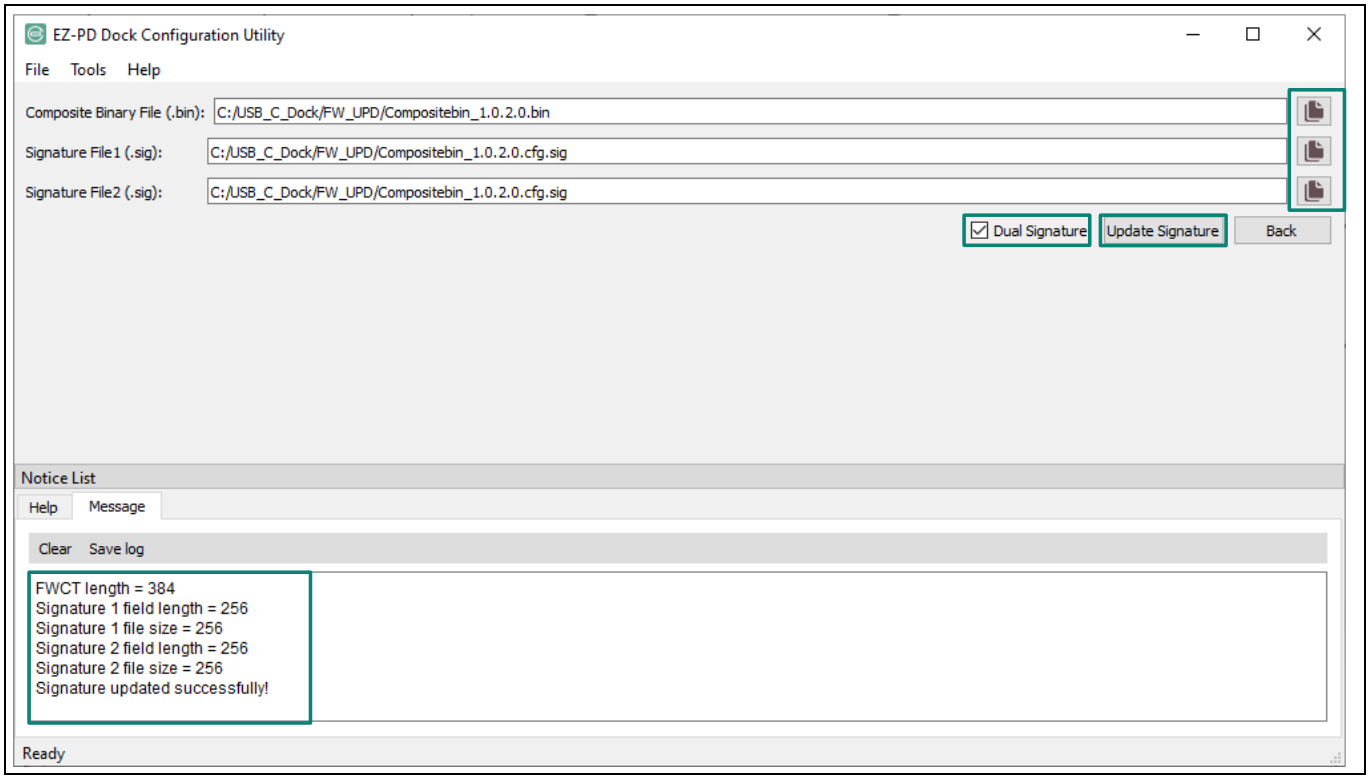


Figure 21 Updating signature to Composite binary

*Note: It is mandatory to select the **Dual Signature** option.*

8.5 Firmware update

Customers should use the *EZPD.FwUpdateUtility.exe* to perform firmware updates on the Dock using the following command.

EZPD.FwUpdateUtility.exe -vid <PMG1-S3-VID> -pid <PMG1-S3-PID> -i <Compositebin file name>

For the USB-C Dock, use the following command from the command line:

EZPD.FwUpdateUtility.exe -vid 04B4 -pid F504 -i Compositebin_1.0.2.0.bin

Customers must provide the signature updated composite binary in order to perform signed updates. For unsigned updates, make sure to use cyacd2 files without signing support while creating composite binary.

Once the firmware update is complete, the *EZPD.FwUpdateUtility* displays the status as shown in Figure 22.

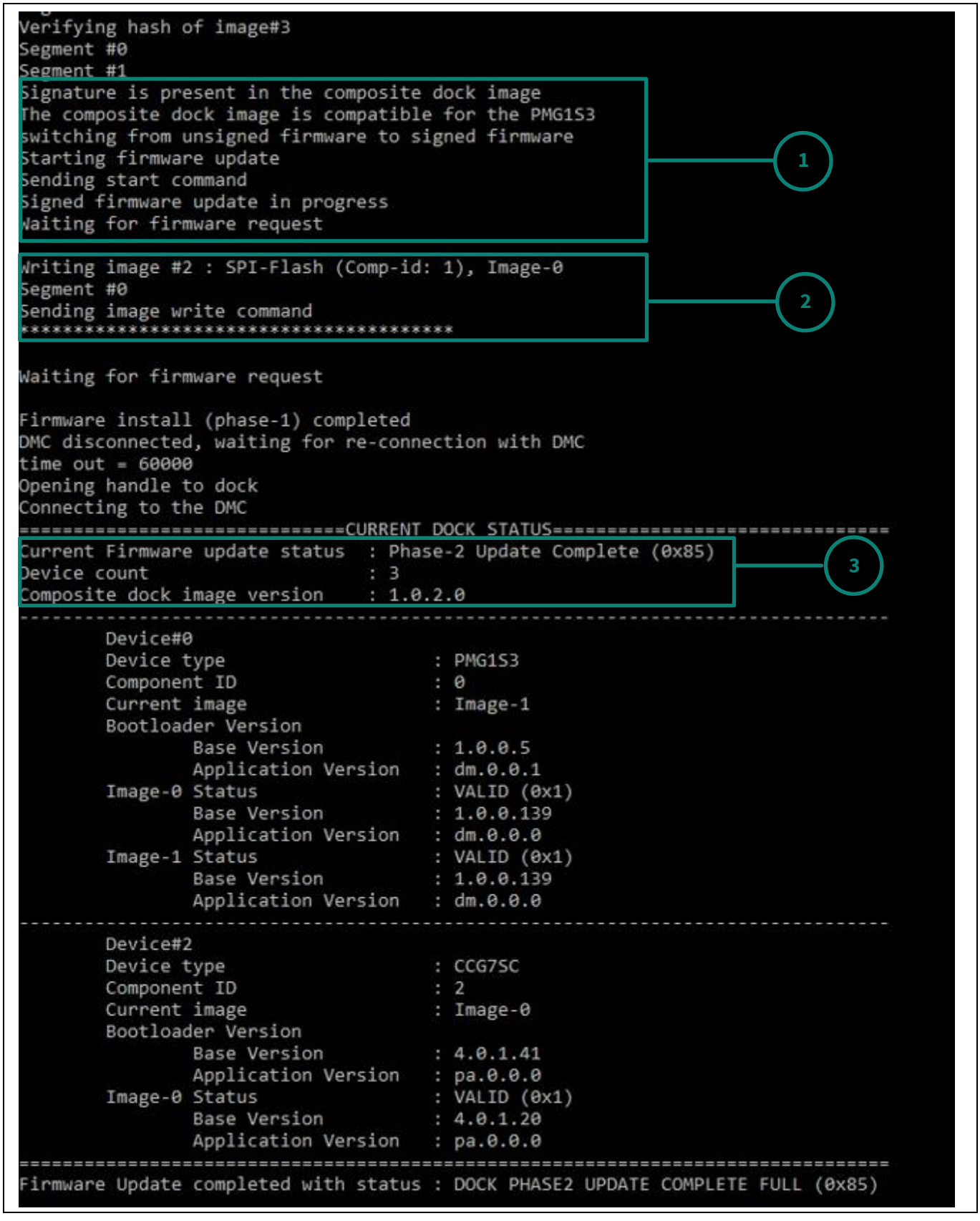


Figure 22 Firmware update complete status

8.6 Switching from unsigned firmware update to signed firmware update

By default, the pre-built binaries and EZ-PD™ PMG1-S3 Dock SDK support unsigned firmware update.

Follow the steps provided in sections Dock Composite bin creation, Generating and updating the signature, and Firmware update to change from an unsigned firmware update to a signed firmware update.

9 PMG1-S3 Dock SDK customization

The following sections cover the most common customization examples performed by customers for the PMG1-S3 USB-C Dock solution.

9.1 Prerequisite

1. Refer to the EZ-PD™ PMG1-S3 Dock SDK user guide.
2. Create a project in Eclipse IDE for ModusToolbox™ software using the PMG1S3 USB-C Dock Solution Template Application.

9.2 Customizing the US port buck boost controller

The PMG1-S3 USB-C Dock Solution supports Monolithic Power Systems and Richtek's buck boost controller on the US port. The following sections cover changes to be made to the PMG1-S3 USB-C Dock solution to support a new buck boost controller.

Before proceeding to the next section, follow the steps provided in the Prerequisite section.

9.2.1 Makefile changes

Create a new Makefile variable corresponding to the new buck boost controller. For example, the CY_APP_NEWBB name is chosen, as shown in Figure 23 and Figure 24. Set this value to 0 for FW1 and to 1 for FW2.

```
ifeq ($(APPNAME_EXT), fw1)

DEFINES+=SYS_DEEPSLEEP_ENABLE=0

DMC_DEFINES+=CY_APP_USB_ENABLE=0 CY_APP_SMART_POWER_ENABLE=0 \
             CY_APP_DMC_PHASE1_UPDATE_ENABLE=0 CCGX_UPDATE=0 HX3_BOOT_WAIT_ENABLE=0 \
             CY_HPI_MASTER_ENABLE=0 CY_APP_USB_HID_INTF_ENABLE=0 ETAG_SUPPORT_ENABLE=1

PD_DEFINES+=CY_PD_EPR_ENABLE=0 CY_PD_EPR_AVS_ENABLE=0 CY_PD_PPS_SRC_ENABLE=0 \
            EXTENDED_ALERT_EVENTS_SUPPORT=0 CY_APP_PD_ENABLE=0 PMG1_V5V_CHANGE_DETECT=0 \
            CY_APP_BUCKBOOST_MP4247_ENABLE=0 CY_APP_BUCKBOOST_RT6190_ENABLE=0 \
            CY_APP_NEWBB=0

FAULT_DEFINES+=VBUS_RCP_ENABLE=0 VBUS_SCP_ENABLE=0 VCONN_OCP_ENABLE=0 VBUS_OCP_ENABLE=0 \
               VBUS_OVP_ENABLE=0 VBUS_UVP_ENABLE=0 CY_APP_DEFER_SNK_VBUS_UVP_HANDLING=0
```

Figure 23 Adding new buck boost controller defines to FW1

```
154 else
155
156 DEFINES+=SYS_DEEPSLEEP_ENABLE=1 CY_APP_GET_REVISION_ENABLE=1
157
158 DMC_DEFINES+=CY_APP_USB_ENABLE=1 CY_APP_SMART_POWER_ENABLE=1 \
159              CY_APP_DMC_PHASE1_UPDATE_ENABLE=1 CCGX_UPDATE=1 HX3_ENABLE=1 \
160              CY_HPI_MASTER_ENABLE=1 CY_APP_USB_HID_INTF_ENABLE=1 ETAG_SUPPORT_ENABLE=1
161
162 PD_DEFINES+=CY_PD_EPR_ENABLE=1 CY_PD_EPR_AVS_ENABLE=1 CY_PD_PPS_SRC_ENABLE=1 \
163             EXTENDED_ALERT_EVENTS_SUPPORT=1 CY_APP_PD_ENABLE=1 PMG1_V5V_CHANGE_DETECT=1 \
164             CY_APP_BUCKBOOST_MP4247_ENABLE=0 CY_APP_BUCKBOOST_RT6190_ENABLE=0 \
165             CY_APP_NEWBB=1
166
167 FAULT_DEFINES+=VBUS_RCP_ENABLE=0 VBUS_SCP_ENABLE=1 VCONN_OCP_ENABLE=1 VBUS_OCP_ENABLE=1     \
168                VBUS_OVP_ENABLE=1 VBUS_UVP_ENABLE=1 CY_APP_DEFER_SNK_VBUS_UVP_HANDLING=1
169
```

Figure 24 Adding new buck boost controller defines to FW2

9.2.2 Creating new buck boost support header and source files

Copy and paste the *cy_app_mp4247.c* and *cy_app_mp4247.h* files to the *mtb_shared\pmg-app-common\..* folder and rename them. For example, as shown in Figure 25, these files are renamed to *cy_app_new_buck_boost.c* and *cy_app_new_buck_boost.h.*

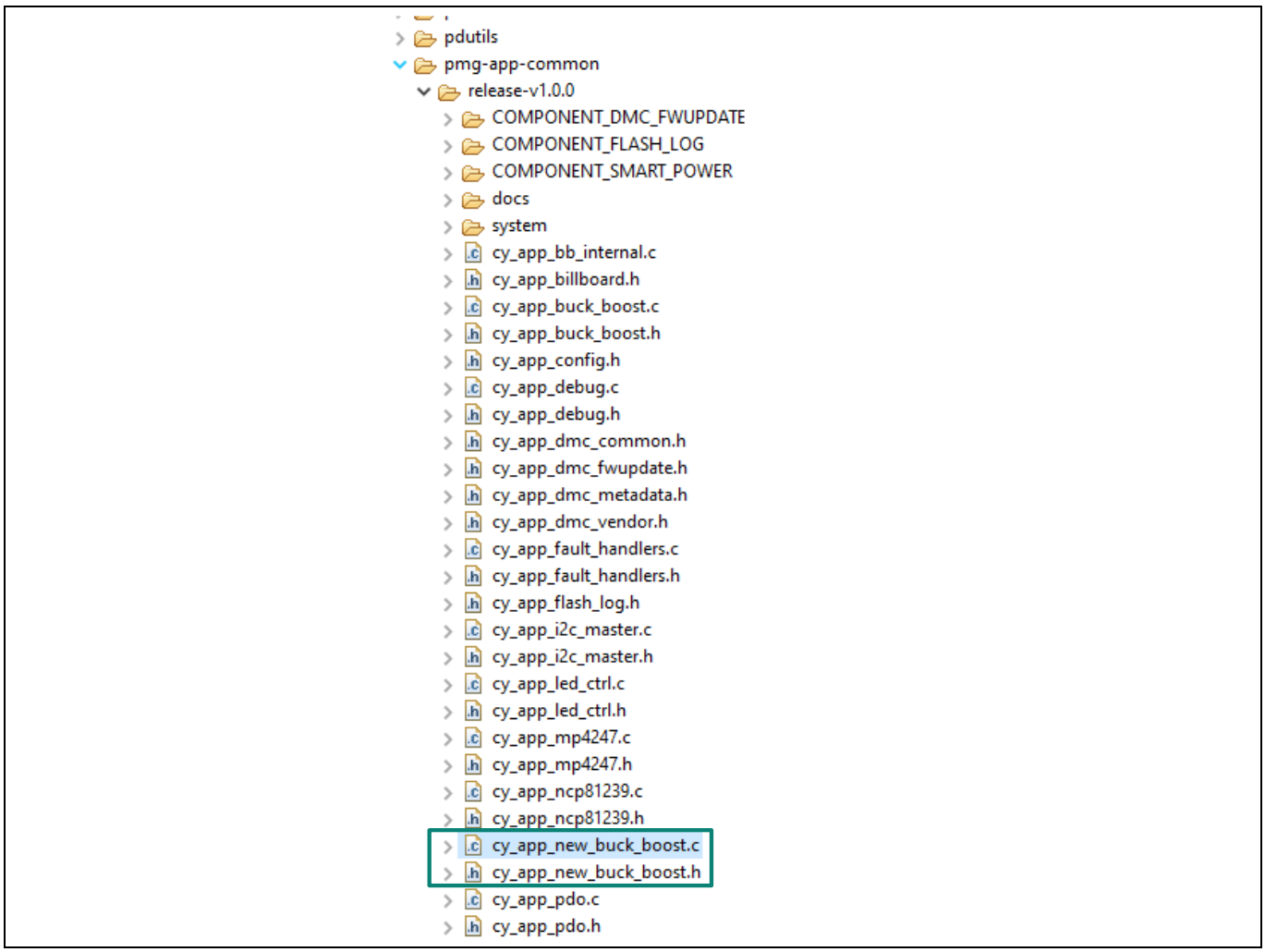


Figure 25 Creating new buck boost controller source and header files

9.2.3 Modifications to new controller header file cy_app_new_buck_boost.h

1. Change the *cy_stc_app_mp4247_context_t* name as per the new buck boost controller name and add the additional required data structure members to the context at run time. In the following example, it is named as *cy_stc_app_newbb_context_t.*
2. The APIs given in Table 10 need to be renamed as per the new buck boost being used. In addition, remove the other APIs that are not applicable to the new buck boost controller.

Table 10 APIs to be added for new buck boost

Existing API	New API	Description
Cy_App_MP4247_Init	Cy_App_NewBB_Init	Refer to the PmgAppCommon API guide for the existing API description.
Cy_App_MP4247_SetVolt	Cy_App_NewBB_SetVolt	Refer to the PmgAppCommon API guide for the existing API description.
Cy_App_MP4247_Enable	Cy_App_NewBB_Enable	Refer to the PmgAppCommon API guide for the existing API description.
Cy_App_MP4247_Disable	Cy_App_NewBB_Disable	Refer to the PmgAppCommon API guide for the existing API description.

3. Rename the context structure type from *cy_stc_app_mp4247_context_t* to *cy_stc_app_newbb_context_t*.

9.2.4 Modification to cy_app_buck_boost.h

Add *cy_app_new_buck_boost.h* as part of *cy_app_buck_boost.h*, as shown below.

Code listing 1 Include the new buck boost header file

```
001  #if CY_APP_NEWBB
002  #include "cy_app_new_buck_boost.h"
003  #endif /* CY_APP_NEWBB */
```

9.2.5 Modifications to new controller source files cy_app_new_buck_boost.c and main.c

1. Replace *cy_app_mp4247.h* with *cy_app_new_buck_boost.h* in *cy_app_new_buck_boost.c*.
2. Create two global variables to hold context structures for the new buck boost controller in the existing file *cy_app_buck_boost.c.*

Code listing 2 Adding context for new buck boost for upstream and downstream (DS1) port

```
001  #if CY_APP_BUCKBOOST_MP4247_ENABLE
002  static cy_stc_app_mp4247_context_t *mp4247Port1CtxPtr = NULL;
003  static cy_stc_app_mp4247_context_t *mp4247Port2CtxPtr = NULL;
004  #elif CY_APP_BUCKBOOST_RT6190_ENABLE
005  static cy_stc_app_rt6190_context_t *rt6190Port1CtxPtr = NULL;
006  static cy_stc_app_rt6190_context_t *rt6190Port2CtxPtr = NULL;
007  #elif CY_APP_BUCKBOOST_NCP81239_ENABLE
008  static cy_stc_app_ncp81239_context_t *ncp81239Port1CtxPtr =
     NULL;
009  static cy_stc_app_ncp81239_context_t *ncp81239Port2CtxPtr =
     NULL;
010  #elif CY_APP_NEWBB
011  static cy_stc_app_newbb_context_t *newBBP1CtxPtr = NULL;
```

```
012         static cy_stc_app_newbb_context_t *newBBP2CtxPtr = NULL;
013         #endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

3. Implement the functions *Cy_App_NewBB_Init*, *Cy_App_NewBB_Enable, Cy_App_NewBB_Disable*, and *Cy_App_NewBB_SetVolt,* which are newly added in Table 10.
4. Modify *Cy_App_BuckBoost_InitPort1*, *Cy_App_BuckBoost_SetVoltPort1*, and *Cy_App_BuckBoost_SetOutputPort1*, as shown below.

Code listing 3 Initialization of new buck boost context pointer for upstream port in Cy_App_BuckBoost_InitPort1

```
001         #elif CY_APP_NEWBB
002             newBBP1CtxPtr = (cy_stc_app_newbb_context_t *)contextPtr;
003
004             return Cy_App_NewBB_Init(newBBP1CtxPtr);
005         #else
006             return false;
007         #endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

Code listing 4 Calling New buck boost set voltage function in Cy_App_BuckBoost_SetVoltPort1

```
001         #elif CY_APP_NEWBB
002             return (NULL == newBBP1CtxPtr)? false:
    Cy_App_NewBB_SetVolt(newBBP1CtxPtr, vol_in_mv);
003         #else
004             return false;
005         #endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

Code listing 5 Calling New buck boost Enable and Disable function in *Cy_App_BuckBoost_SetOutputPort1*

```
001         #elif CY_APP_NEWBB
002             if(NULL != newBBP1CtxPtr)
003             {
004                 if(output)
005                 {
006                     status = Cy_App_NewBB_Enable(newBBP1CtxPtr);
007                 }
008                 else
009                 {
010                     status = Cy_App_NewBB_Disable(newBBP1CtxPtr);
011                 }
012             }
013         #endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

5. Modify the functions *Cy_App_BuckBoost_InitPort2*, *Cy_App_BuckBoost_SetVoltPort2,* and *Cy_App_BuckBoost_SetOutputPort2* for Port-2 and choose *newBBP2CtxPtr* as context.
6. The main source file (*main.c*) requires changes to create a context for the new buck boost controller. Look for CY_APP_BUCKBOOST_MP4247_ENABLE in *main.c* and add context for new buck boost by referring to the existing context for MP4247 or RT6190 controllers.

Code listing 6 Creating the context for the new buck boost controller support

```
#elif CY_APP_NEWBB
cy_stc_app_newbb_context_t gl_newbbContextPort1 =
{
    .i2cAddr = MP4247BB_REG_I2C_ADDR_P1,
    .scbBase = I2C_PWR_MUX_HW,
    .i2cContext = &gl_i2c_pwr_mux_context,
    .enableGpioPort = REG_EN_PORT,
    .enableGpioPin = REG_EN_PIN,
    .fbRatio = MP4247BB_REG_FB_RATIO
};
#endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

7. The newly created context is to be used as an argument to *Cy_App_BuckBoost_InitPort1*.

Code listing 7 New buck boost Initialization for upstream port

```
/* Enable Buck-Boost converter. */
#if CY_APP_BUCKBOOST_MP4247_ENABLE
    Cy_App_BuckBoost_InitPort1(&gl_mp4247ContextPort1);
#elif CY_APP_BUCKBOOST_RT6190_ENABLE
    Cy_App_BuckBoost_InitPort1(&gl_rt6190ContextPort1);
#elif CY_APP_NEWBB
    Cy_App_BuckBoost_InitPort1(&gl_newbbContextPort1);
#endif /* CY_APP_BUCKBOOST_MP4247_ENABLE */
```

Note: The highlighted text indicates a newly added code listing to the existing code.

9.3 Customizing the Data Mux for USB-C Dock

This section covers modifications required to the US and DS data mux changes.

The EZ-PD™ PMG1-S3 USB-C Dock hardware uses ANX7443 and PI3USB31532 as upstream and downstream (for the DS1 port) Type-C port mux controllers, respectively.

If customers use Analogix data mux controllers, contact them for configuration information and hardware application examples. The source code for the US data mux is available in library form as part of the EZ-PD™ PMG1-S3 Dock SDK.

The DS1 Type-C port mux controller's source code is made available in source form for the PMG1S3 USB-C Dock Solution example. Refer to the source files "*src\mux\pericom_pi3usb31532.h*" and "*src\mux\pericom_pi3usb31532.c*" for the implementation of DS1 mux control.

The *Cy_PdAltMode_HW_CustomMuxSet* calls the US and DS1 mux controlling functions, which are shown in the following code snippet:

Code listing 8 US and DS1 mux control

```
if(!ptrAltModeContext->pdStackContext->port)
    {
        retval = anx7443_mux_cfg(ANX7443_I2C_SLAVE_ADDR_TOP,
                                 mux_cfg, ptrAltModeContext-
  >pdStackContext->dpmConfig.polarity,
                                 app_mux_i2c_write);
    }
    else
```

```
008             {
009                 retval = pericom_mux_cfg(USB31532_SLAVE_ADDR,
010                                          mux_cfg,ptrAltModeContext->pdStackContext-
   >dpmConfig.polarity,
011                                          app_mux_i2c_write;
012             }
```

Customers should invoke the newly implemented mux controlling functions from *Cy_PdAltMode_HW_CustomMuxSet* for either the upstream or downstream port (Type-C DS1 Port).

Refer to the PdAltMode API reference guide for details about the *Cy_PdAltMode_HW_CustomMuxSet* function and its arguments.

9.4 USB and DP hub controllers modifications for custom designs

This section covers the existing implementation for USB and DP hub controllers that are used in EZ-PD™ PMG1-S3 USB-C Dock hardware.

The EZ-USB™ HX3 Hub and Synaptics DP hub controllers are used in the EZ-PD™ PMG1-S3 USB-C Dock hardware. The following table lists how the GPIO's of each controller are controlled.

Table 11 EZ-USB™ hub and Synaptics DP controller GPIO control

Controller	GPIO signal	US port Type-C connect event	US port Type-C disconnect event
EZ-USB™ HX3 Hub	HX3 RESET (Active Low)	Drive high	Drive low
	VBUS US (Active High)	Drive high 700 ms after the Type-C connect event	Drive low
Synaptics DP controller	VMM RESET (Active Low)	Drive high	Drive low

Refer to the "*sln_pd_event_handler*" function in *solution.c* for reference.

Customers need to contact the respective USB and Display hub vendors for configuration and hardware application examples.

9.5 Enabling the custom ALT mode support

1. Get the new customer VID <*vvvv*> for which the alternate mode support is required.
2. Create a new PMG1S3 USB-C Dock Solution using the ModusToolbox™ IDE. Refer to [1] for steps on how to create the PMG1S3 USB-C Dock Solution.
3. Make the following changes to the Makefile:
   a) Set CUSTOM_ALT_MODE_UFP_SUPP = 1
   b) Set CUSTOM_ALT_MODE_DFP_SUPP = 1
4. Modify CUSTOM_ALT_MODE_VID from 0x04B4 to the new value in hex <*vvvv*>. The customers should not use *0x04B4*, as it is associated with Infineon.
5. Build the project and generate the hex file.
6. Launch EZ-PD™ Dock Configuration Utility and load the configuration from the newly generated hex file using the **Read from hex** option.
7. Navigate to the **Port-0** tab and

a) Add a new SVID for which alt mode support is required to **Alternate mode 0** under **Base alternate modes**. Select “Supported in UFP” for the newly added SVID.

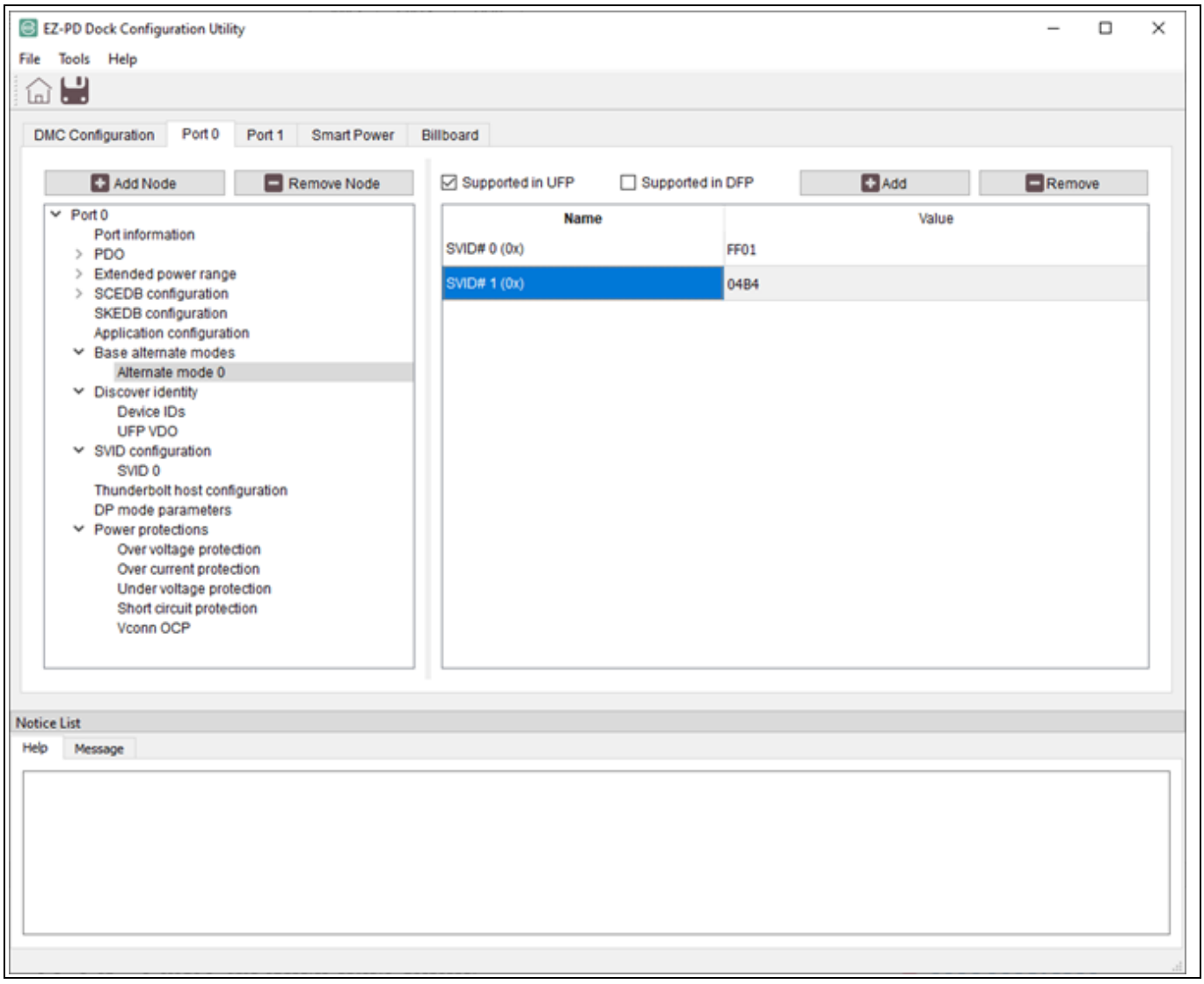


Figure 26 Adding new SVID for which alt mode support is required

b) Click on **Add Node** to add **Alternate mode 1** and enter the new SVID as **SVID# 0** and the existing SVID from **Alternate mode 0** as **SVID# 1**, which is shown in Figure 27.

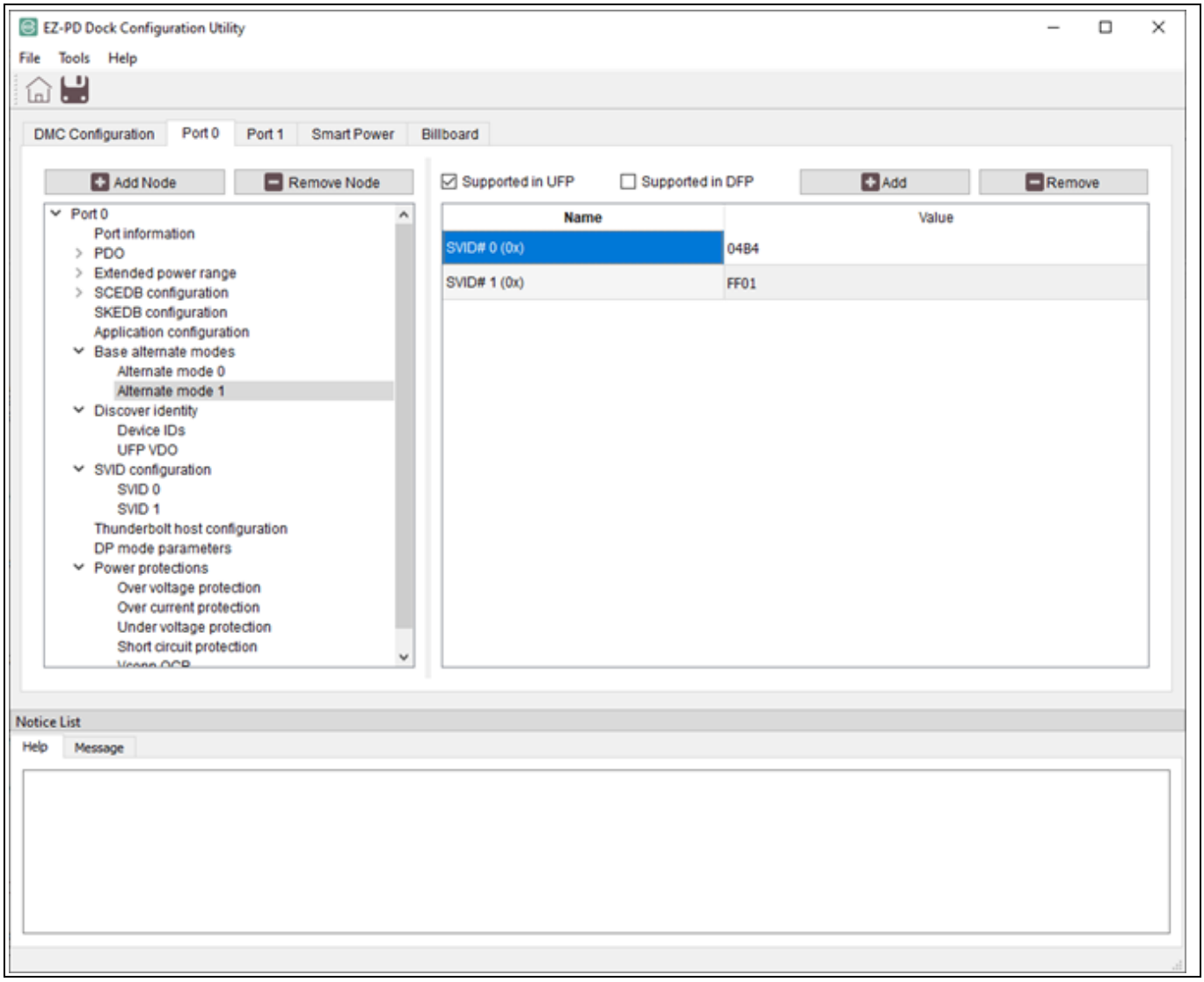


Figure 27 Adding support for custom alt mode SVID and DP ALT mode simultaneously

c) Navigate to the **SVID configuration** and select **Add Node** to add a new SVID and its mode value.

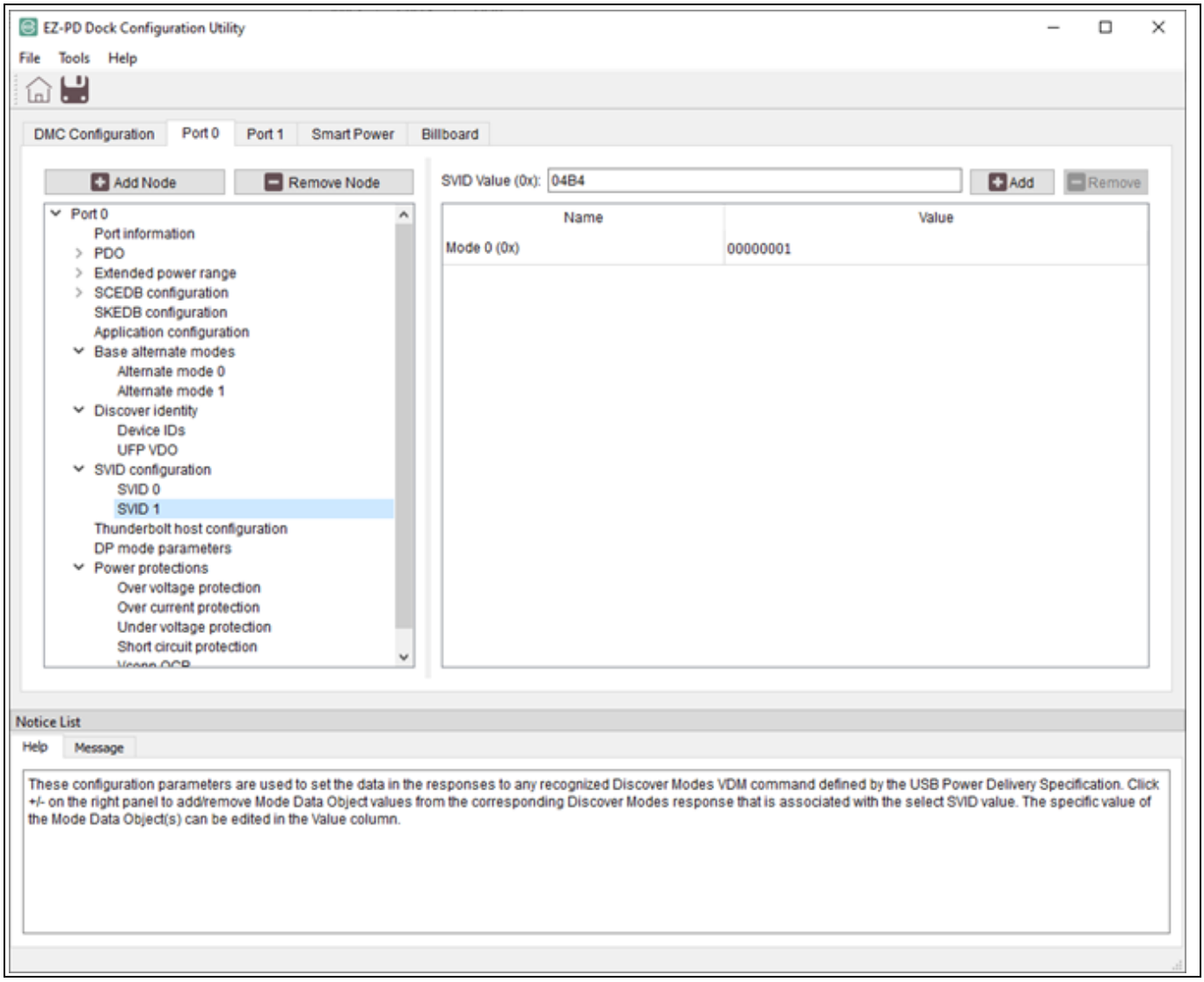


Figure 28 Adding Discover mode response for custom alt mode

d) Repeat steps a) and b) for Port-1 if custom alt mode support is required on the DS1 port. Select **Supported in DFP** for the custom alt mode SVID while adding to Port-1.

e) Save the configuration to the hex file, and SWD program the hex file to PMG1-S3.

Note: EZ-PD™ Dock Configuration Utility accepts SVID values in capitals only for hex values from A to F.

When the upstream port is connected to another port partner that supports the custom VID value being used, both ports can enter customer alt mode.

References

[1] EZ-PD™ PMG1-S3 Dock SDK user guide

[2] EZ-PD™ Dock Configuration Utility user guide

[3] EZ-PD™ Analyzer user guide

This user guide should be read with the following Industry Standard Specifications:

- USB Power Delivery specification, revision 3.1, v1.8 (USB-IF)
- USB Type C Cable and Connector specification, revision 2.2 (USB-IF)
- VESA DisplayPort Alt mode on USB Type-C Standard, version 1.0 (VESA)
- Universal Serial Bus Device Class Definition for Billboard Devices, revision 1.2.2 (USB-IF)
- Device Class Definition for Human Interface Devices (HID), 1.11 (USB-IF)

Glossary

DFP
Downstream Facing Port

DP
Display Port

EPR
Extended Power Range

HDMI
High Definition Media Interface

IDE
Integrated Development Environment

MCU
Microcontroller unit

OCP
Overcurrent protection

OVP
Overvoltage protection

PDO
Power Delivery Object

PPS
Programmable Power Supply

RSA
Rivest–Shamir–Adleman, public key cryptography algorithm

Glossary

SPI
Serial Peripheral Interface

SSD
Solid State Drive

SPR
Standard Power Range

SWD
Serial Wire Debug

TBT
Thunderbolt. High speed peripheral connect technology from Intel.

UFP
Upstream Facing Port

USB
Universal Serial Bus

Revision history

Document revision	Date	Description of changes
**	2024-03-13	Initial release.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.
The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

Edition 2024-03-13
Published by

Infineon Technologies AG
81726 Munich, Germany

© 2024 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?
Email:
erratum@infineon.com

Document reference
002-39544 Rev. **

Important notice
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.