



Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



Cypress EZ-PD™ CCGx Host SDK Release Notes

Version 3.4, July 11, 2020

Thank you for your interest in the EZ-PD™ CCGx Host Software Development Kit (SDK). This SDK supports the CCG5, CCG5C, CCG6, CCG6DF and CCG6SF families of Type-C controllers and is targeted at creating USB-PD port controller applications for desktop and notebook computers.

Introduction

The CCGx Host SDK provides a set of firmware resources that allows users to build customized applications using the Type-C port controllers from Cypress.

This is based on a CCGx firmware stack and provides programming hooks and interfaces for customers to implement their own policy and system management schemes.

The key application-level requirements for the firmware stack are as follows:

- USB Type-C Revision 2.0 and USB-PD Revision 3.0 specification compliant PD stack for CCG5, CCG5C, CCG6 and CCG6xF (CCG6DF and CCG6SF).
- USB Type-C Connector System Software Interface Revision 1.1 support for CCG5, CCG5C, CCG6 and CCG6xF.
- Drivers for the various hardware blocks on the CCGx controllers.
- Allow manufacturing-level customization of device parameters such as power profiles, default port behavior, Over Voltage Protection, and so on, without changing the firmware.
- Map firmware to system hardware design changes without any impact on the core firmware modules.
- Facilitate source-level debugging so that new customers can get familiarized with the stack.

Note: Support for CCG3 and CCG4 devices is in maintenance mode, and the corresponding firmware projects have been replicated from past SDK versions without any new feature additions.

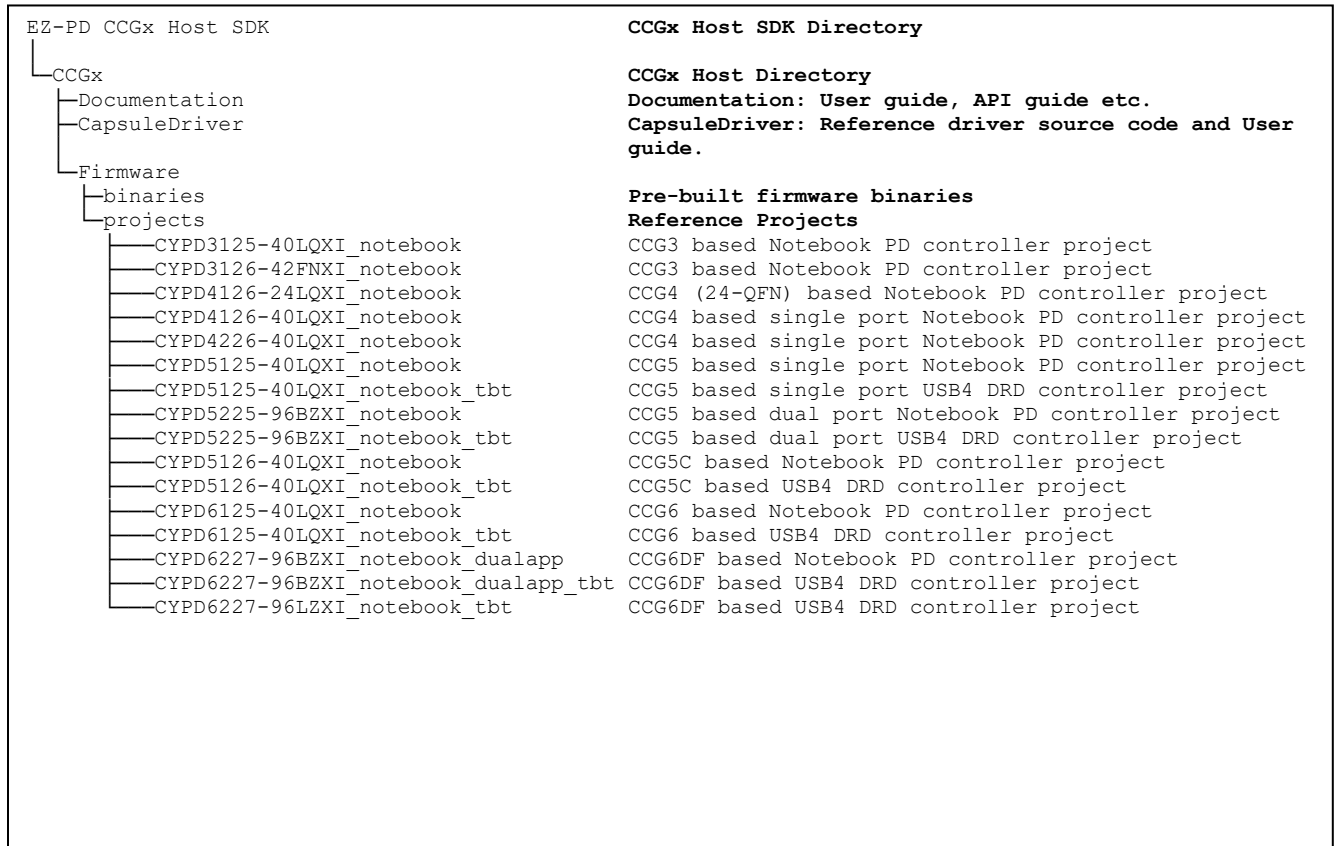
Firmware Organization

The CCGx Firmware Stack is released in the form of an SDK, which contains the firmware stack as well as reference firmware projects that use the stack to implement USB-PD applications.

The firmware stack and reference projects are available in the **CCGx** sub-directory. The current stack version is 3.4.0 Build 2559 for CCG5, CCG5C and CCG6. The current stack version is 3.4.0 Build 425 for CCG6xF. All stack documentation for CCGx device family is available under **CCGx > Documentation**.



Figure 1: EZ-PD™ CCGx Host SDK Directory Structure



CCGx contains the following folders:

- **Documentation:** The docs folder contains the EZ-PD™ CCGx Host SDK documentation, which includes release notes, user guide, and API reference guide.
- **CapsuleDriver:** The capsule driver folder contains the capsule reference driver source code and the user guide.
- **Firmware:** The Firmware folder contains the firmware stack sources, reference projects, and pre-built firmware binaries targeted for the Kits and reference designs from Cypress.
 - **binaries:** The binaries folder contains the pre-built firmware binaries
 - **projects:** The projects folder contains the sources and PSoC Creator workspaces for the port controller designs.
 - **CYPD5125-40LQXI_notebook:** Single port Notebook PD port controller application using CCG5.
 - **CYPD5125-40LQXI_notebook.cywrk:** PSoC Creator workspace settings for the application.
 - **CYPD5125-40LQXI_notebook.cydsn:** PSoC Creator projects container directory.
 - **Bootloader:** Bootloader binaries used in the application.
 - **common:** Application specific source files.
 - **src:** General CCGx firmware sources and headers.
 - **lib:** USB-PD stack, HPI and Intel platform support libraries.
 - **CYPD5125-40LQXI_notebook.cypri:** PSoC Creator build settings for the application.
 - **backup_fw.cydsn:** Container for the reduced feature secondary (back-up) application which is combined with the main application.
 - **CYPD5126-40LQXI_notebook:** Single port Notebook PD port controller application using CCG5C.
 - **CYPD6125-40LQXI_notebook:** Single port Notebook PD port controller application using CCG6.
 - **CYPD5225-96BZXI_notebook:** Dual port Notebook PD port controller application using CCG5.
 - **CYPD5125-40LQXI_notebook_tbt:** Single port USB4 DRD port controller application using CCG5.
 - **CYPD5126-40LQXI_notebook_tbt:** Single port USB4 DRD port controller application using CCG5C.
 - **CYPD6125-40LQXI_notebook_tbt:** Single port USB4 DRD port controller application using CCG6.
 - **CYPD5225-96BZXI_notebook_tbt:** Dual port USB4 DRD port controller application using CCG5.
 - **CYPD6227-96BZXI_notebook_dualapp:** Notebook PD port controller application using CCG6SF (single port) or CCG6DF (dual port).
 - **CYPD6227-96BZXI_notebook_dualapp_tbt:** USB4 DRD port controller application using CCG6SF (single port) or CCG6DF (dual port).
 - **CYPD6227-96BZXI_notebook_tbt:** USB4 DRD port controller application using CCG6SF (single port) or CCG6DF (dual port).
 - **CYPD3125-40LQXI_notebook:** Single port Notebook PD port controller using CCG3
 - **CYPD3126-42FNXI_notebook:** Single port Notebook PD port controller using CCG3
 - **CYPD4126-24LQXI_notebook:** Single port Notebook PD port controller using CCG4
 - **CYPD4126-40LQXI_notebook:** Single port Notebook PD port controller using CCG4
 - **CYPD4226-40LQXI_notebook:** Dual port Notebook PD port controller using CCG4

The src folder inside each reference application has the following sub-folders:

- **app:** The app folder contains the top-level application layer functionality that implements the required USB-PD controller functions. This includes functionality such as PDO evaluation and contract negotiation, fault detection and handling, BC 1.2 charging support etc.
 - **app/alt_mode:** The alternate mode specific implementation can be found in this directory.
 - **app/intel_tbt:** This folder contains the headers for the Intel Platform support including I2C slave interface to the Platform SoC and I2C master interface to control retimers.
- **hpiiss:** The hpiiss folder contains the API interface definition for the Host Processor Interface implemented by the CCGx firmware.
- **pd_common:** The pd_common folder contains the headers for the core Type-C and USB-PD stack for the CCGx device. This includes the HAL, the Type-C port manager, the USB-PD protocol layer, the USB-PD policy engine, and the Device Policy Manager.
- **pd_hal:** The pd_hal folder contains the low-level driver header and source files for USB-PD hardware block.
- **scb:** The scb folder contains the API interface definition for the dedicated I2C slave driver using the Serial Controller Blocks (SCB) on the CCGx device. Since I2C slave mode is the most commonly-used interface for CCGx, a specially optimized driver is provided for the same.
- **system:** The system folder contains header and source files relating to the CCGx device hardware and registers, bootloader and flash access functions, low-level drivers for the GPIO blocks on the CCGx device, and a soft timer implementation that is used by the firmware stack.
- **ucsi:** The ucsi folder contains implementation of the USB Type-C Connector System Software Interface (UCSI v1.1). The UCSI protocol is implemented on top of the HPI I2C slave interface using separate registers and commands.

Tool Requirements

1. This version of SDK requires PSoC Creator 4.3.0.34 or higher for compilation.
Download the latest PSoC Creator version from:
<http://www.cypress.com/products/psoc-creator-integrated-design-environment-ide>
You can download older versions of PSoC Creator from:
<http://www.cypress.com/documentation/software-and-drivers/psoc-creator-software-archive>
2. This version of SDK requires ARM MDK Compiler version 5.26 or higher.
3. This version of SDK requires Python 3.x or higher.
4. Use the Tools → Find New Devices option of PSoC Creator 4.3 to obtain support for all new devices including the CCG6SF and CCG6DF devices referred to in this SDK.
5. This version of SDK requires EZ-PD™ Configuration Utility version 1.3.1 or higher.
Download the latest version of the EZ-PD™ Configuration Utility installer from:
<http://www.cypress.com/ezpdutility>

Changes from CCGx Host SDK 3.3.1

1. All projects in this version of the SDK build only with ARM MDK compiler.
2. Added support and provided code examples for CCG6xF product family.
 - a. CCG6xF supports Hybrid Architecture in addition to the Legacy Architecture supported on devices such as CCG5, CCG5C and CCG6.
3. Added support and provided code examples for PD controller projects for Intel's Tiger Lake and Ice Lake Platforms.
 - a. Supports USB4 mode of operation in DFP and UFP roles
 - b. Supports Intel's Burnside Bridge Retimer
4. Added VCONN OCP retry logic.
5. Added HPI command to exit modes and restart the VDM process.
6. Added support for Custom Alt Modes.
7. Added three different algorithms (max power, max current, max voltage) to pick up source PDO.



8. Added default HPI event mask for covering events in DeadBattery scenario.

Please refer to the release notes for each reference code example for a complete description of features supported in the primary and backup applications.

Limitations of CCGx Host SDK 3.4.0

1. Vendor Defined Message (VDM) and extended message requests with wrong arguments to the HPI results in Transaction Failed error code instead of Invalid Arguments.
2. The device does not enter low power mode when Fast Role Swap (FRS) receive is enabled and acting as a sink.
3. Failure to enter USB4 mode of operation maybe seen when passive cables which do not support Thunderbolt mode are used. Please contact Cypress Customer support for resolution.
4. The Backup/Secondary Base binary is not expected to be fully compliant to USB PD 2.0 Specification and does not handle BIST messages correctly.

Please refer to the release notes for each reference code example for a list of limitations specific to the code example.

Technical Support

For further assistance, go to <http://www.cypress.com/go/support>.

Additional Information

For more information about the EZ-PD Configuration Utility, visit the web page:

<http://www.cypress.com/documentation/software-and-drivers/ez-pd-configuration-utility>

For more information about the Cypress Type-C controller family, visit the web page:

<http://www.cypress.com/products/usb-type-c-and-power-delivery>



Cypress Semiconductor
198 Champion Ct.
San Jose, CA 95134-1709 USA
Tel: 408.943.2600
Fax: 408.943.4730
Application Support Hotline: 425.787.4814
www.cypress.com

© Cypress Semiconductor Corporation, 2018-2020. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC[®] and CapSense[®] are registered trademarks and Programmable System-on-Chip[™], EZ-PD[™], PSoC Designer[™], and PSoC Creator[™] are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.