Getting Started with EZ-PD™ CCG3PA

Author: Benjamin Avery

Associated Part Family: CYPD317x

Associated Software: EZ-PD™ Configuration Utility

Related Application Notes: AN210403, AN200210

AN218179 introduces the USB Type-C EZ-PD™ CCG3PA controller. It provides a brief overview of the CCG3PA architecture, its features and applications, and covers the evaluation kit in detail along with the development and debugging tools that can be used. It also references CCG3PA resources to help you ramp up quickly with your product designs.

Contents

1 Introduction

EZ-PD CCG3PA belongs to Cypress's family of USB Type-C controllers that complies with the latest USB Type-C and Power Delivery (PD) standards. In addition, with the built-in overvoltage protection (OVP) and overcurrent protection (OCP), it helps to reduce the need for additional components and the overall cost of a Type-C ecosystem. Typical applications using CCG3PA include mobile power adapters, PC power adapters, power banks, and car chargers.

1.1 EZ-PD CCG3PA Features

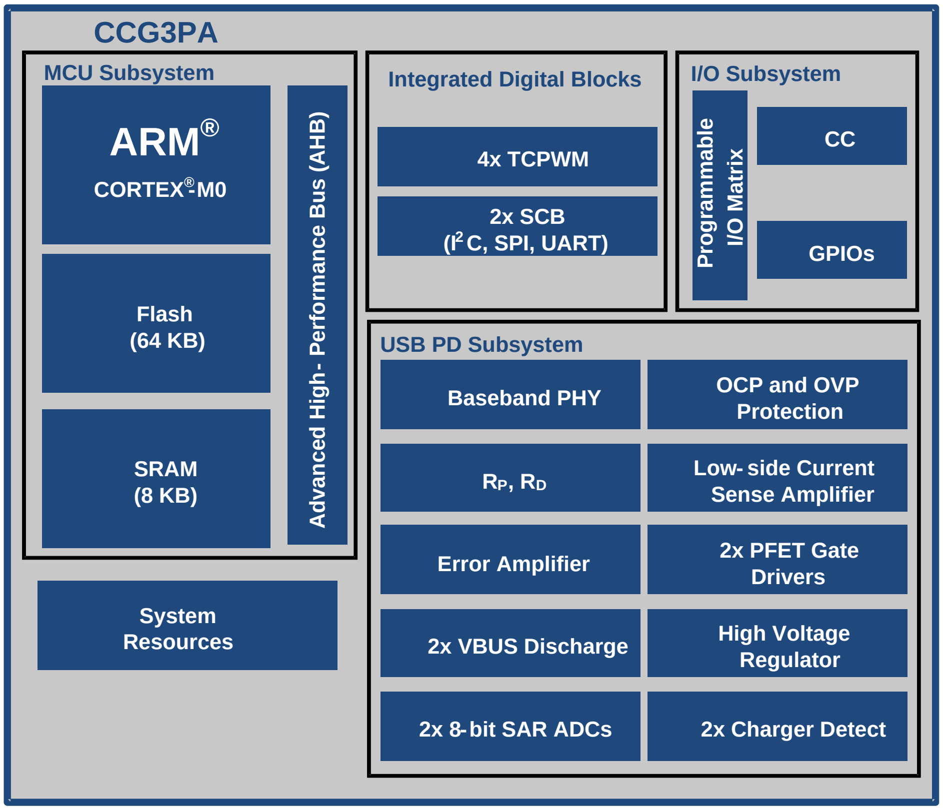
- Type-C Support and USB-PD Support
  - Supports USB PD 3.0 spec including Programmable Power Supply Mode
  - Configurable Resistors R_P and R_D
  - Supports one USB Type-C port and one Type-A port

- 2x Legacy/Proprietary Charging Blocks
  - Supports Quick Charge (QC) 4.0, Apple Charging 2.4A, Adaptive Fast Charging (AFC), Battery Charging (BC) 1.2
  - Integrates all required terminations on DP/DM lines
- Integrated Voltage (VBUS) Regulation and Current Sense Amplifier
  - Analog regulation of secondary-side feedback node (direct feedback or opto coupler)
  - Integrated shunt regulator function for VBUS control
  - Constant current or constant voltage mode
  - Supports low-side current sensing for constant current control
- System-Level Fault Protection
  - On-chip OVP, OCP, UVP (under voltage protection), and SCP (short circuit protection)
  - Supports OTP (one-time programming) through integrated ADC circuit
- 32-bit MCU Subsystem
  - ARM® Cortex®-M0 CPU
  - 64 KB Flash
  - 8 KB SRAM
- Clocks and Oscillators
  - Integrated oscillator eliminating the need for external clock
- Power
  - 3.0 V to 24.5 V operation (30-V tolerant)
- System-Level ESD Protection
  - On Configuration Channel (CC), VBUS, and DP/DM pins
  - ± 8-kV Contact Discharge and ± 15-kV Air Gap Discharge based on IEC61000-4-2 level 4C
- Packages
  - 24-pin QFN and 16-pin SOIC
  - Supports extended industrial temperature range (-40° C to +105° C)

1.2 CCG3PA Block Diagram

Figure 1 shows a block diagram of the CCG3PA architecture. For more details, refer to the CCG3PA datasheet.

Figure 1. CCG3PA Architecture Block Diagram



1.3 Prerequisites

This section lists the hardware and software required to get started with CCG3PA devices.

1.3.1 Hardware

- CY4532 EZ-PD CCG3PA Evaluation Kit (EVK)
- PC with Windows 7 or later and at least one Type-A USB port
- Type-C power sink that supports a 9-V or higher power profile (A 12 inch MacBook Retina from early 2015 is used in the example in Section 5)

  Note: The CCG3PA device works with all Type-C power sinks that support power profiles from 5 V to 20 V. This power profile requirement relates to the CY4532 CCG3PA EVK architecture.
- CY4500 EZ-PD Protocol Analyzer (optional; required only for firmware debugging)
- MiniProg3 device (optional; required only to program the CCG3PA device with a file in *.hex* format)

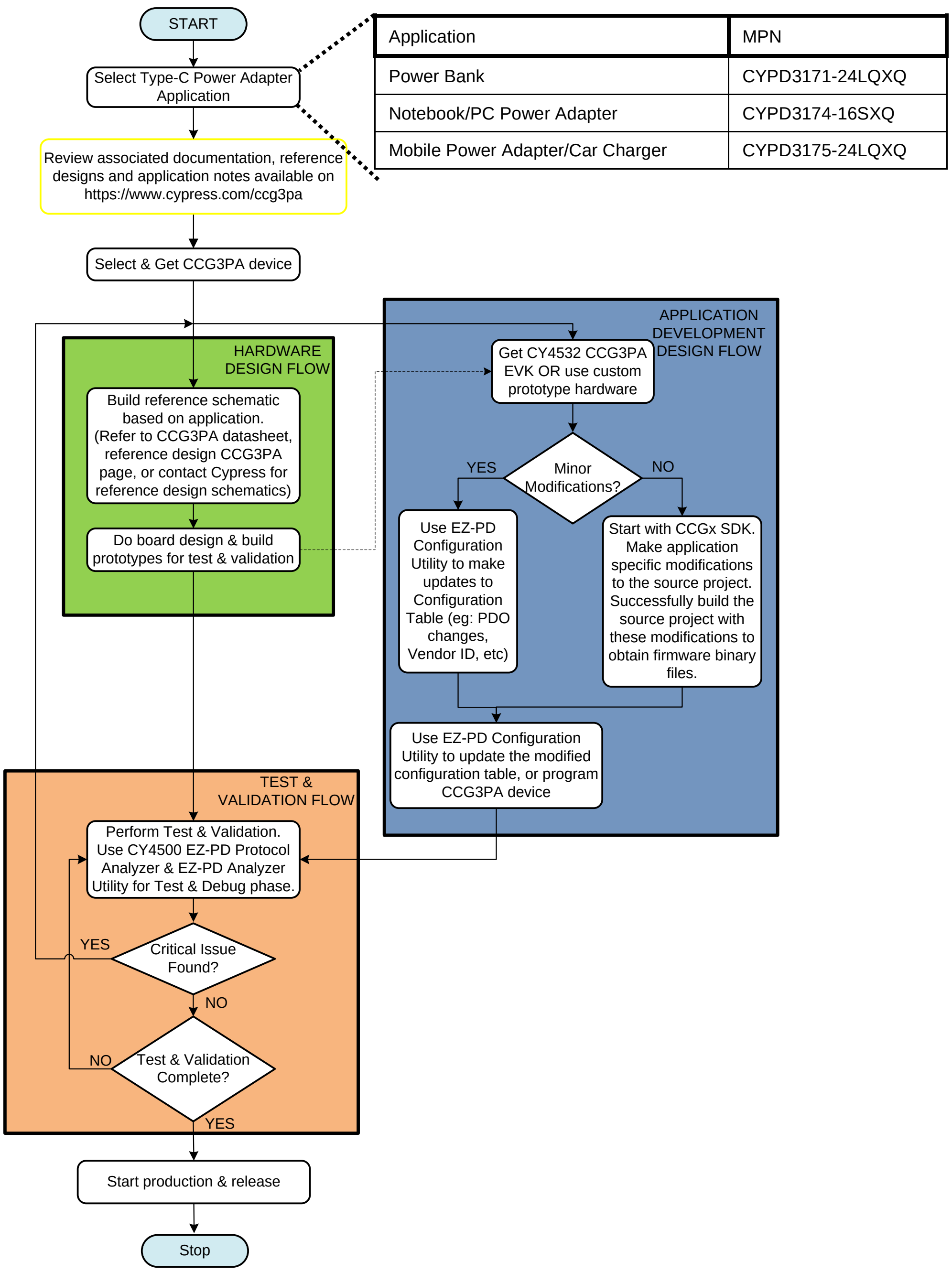
1.3.2 Software

- EZ-PD CCGx Software Development Kit (SDK) (Custom CCG3PA Firmware Package)
- EZ-PD Configuration Utility (version 1.00 or later)
- PSoC® Creator™ 4.1 or later with PSoC Programmer 3.26 or later (required only if modifying base firmware functionality)

1.4 CCG3PA Design Flow

This section describes a typical design flow that you would go through during the Type-C application design from conceptual stage to manufacturing using CCG3PA devices. It also covers how each of the hardware, software, and firmware resources described in this Application Note are used through the design flow. Figure 2. CCG3PA Design FlowFigure 2 shows a typical design flow using CCG3PA devices.

Figure 2. CCG3PA Design Flow



Application	MPN
Power Bank	CYPD3171-24LQXQ
Notebook/PC Power Adapter	CYPD3174-16SXQ
Mobile Power Adapter/Car Charger	CYPD3175-24LQXQ

Once the CCG3PA based Type-C application is determined and reference designs have been reviewed, the hardware and the application development phase of the design flow can be started in parallel.

The hardware flow includes building reference schematics based on the end application and doing the board design to get a few prototypes ready for the next phase. These reference schematics can be based on the reference designs available on Cypress's CCG3PA webpage.

The application development flow can get started using the CY4532 EZ-PD CCG3PA Evaluation Kit (EVK) so that it can proceed in parallel with hardware development. The EZ-PD Configuration Utility can be used to make most updates to the configuration table of the CCG3PA device (for example, changing PDOs and Vendor IDs). For making application-specific modifications, you can use the custom CCG3PA FW Package in the EZ-PD CCGx Software Development Kit (SDK).

The EZ-PD CCGx SDK (referred to as CCGx SDK through the rest of the document) along with PSoC Creator (version 4.1 or later) can be used when necessary to build the source projects and create firmware binary files. The SDK is only required when source code modification is necessary. Most projects will not require using PSoC Creator.

Once either configuration changes (using the EZ-PD Configuration Utility) or firmware changes (using the CCGx SDK) are made, the EZ-PD Configuration Utility can be used to update the modified configuration table or to program the CCG3PA device. More information on which tool can be used for what purpose is covered in detail in Table 2.

Once the hardware and application development flows are completed, the existing system design is ready for the test and validation cycle. The CY4500 EZ-PD Protocol Analyzer can be used for testing, firmware debugging, and performance analysis. Mass production and manufacturing can start once test and validation is complete and the system design is final.

1.5 CCG3PA Resources

Table 1 lists the web resources available to help you design end applications using CCG3PA devices.

Table 1. CCG3PA Design Resources

Category	Available Resources
Datasheet	CCG3PA datasheet
Hardware	CY4532 CCG3PA EVK – Contains documentation and design files
Programming Specifications Document	Programming specifications – Provides guidelines on how to program the flash memory of CCG3PA devices (Contact Cypress)
Host PC Software	EZ-PD CCGx SDK
Host PC Software Debugging Tools	EZ-PD Configuration Utility 1.1 or later (GUI-based Windows application to help you configure CCGx controllers)
	PSoC Creator 4.1 or later (firmware development tool)
	PSoC Programmer 3.26 or later (firmware programming tool)
	CY4500 EZ-PD™ Protocol Analyzer – Includes EZ-PD Analyzer Utility and documentation
Videos	USB Type-C Essentials, USB Type-C 101 Video Training Series
Other Collateral	CCG3PA specific Knowledge Base Articles
Reference Designs	CCG3PA Datasheet and Reference Designs

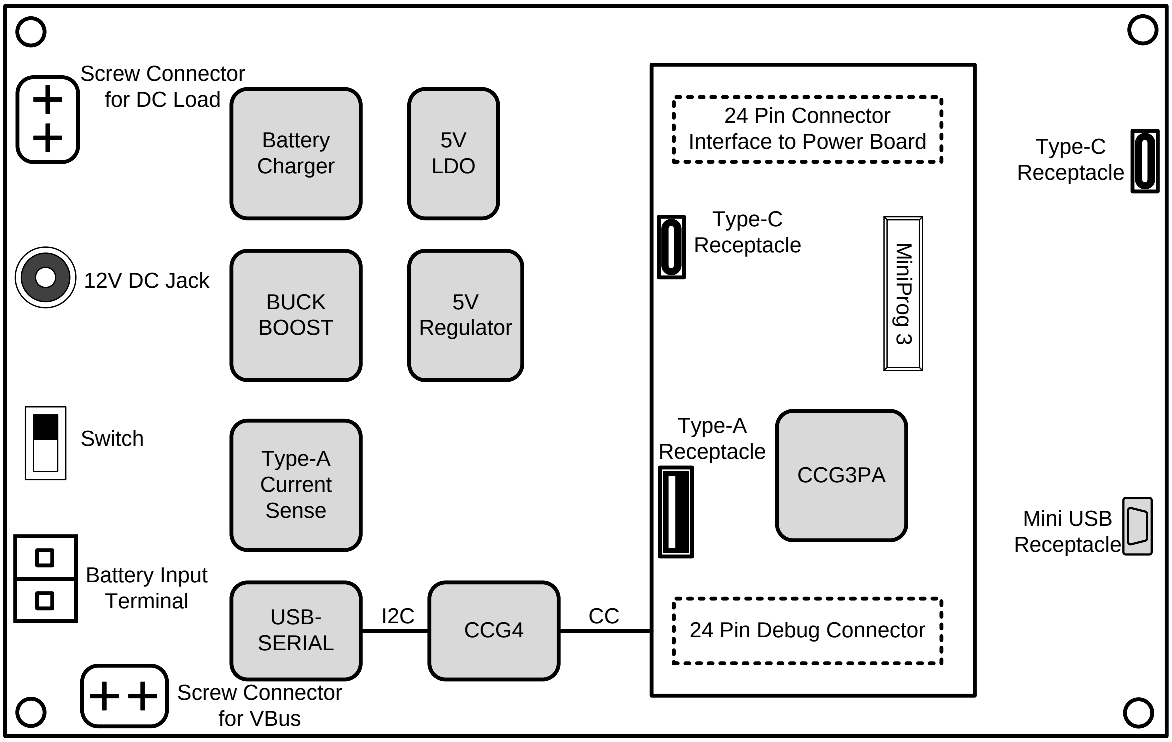
2 CCG3PA Hardware Details

This section discusses the hardware to be used for getting started with the CCG3PA device family. It focuses on the CY4532 EZ-PD CCG3PA EVK and the CY4500 EZ-PD Protocol Analyzer.

2.1 CY4532 EZ-PD CCG3PA EVK

The CY4532 EZ-PD CCG3PA EVK consists of a power board and a main board. The CCG3PA device is mounted on the main board, which is connected to the power board to enable the CCG3PA device's Type-C port functionality. Figure 3 shows a block diagram of the CY4532 EZ-PD CCG3PA EVK architecture.

Figure 3. CY4532 EZ-PD CCG3PA EVK Architecture Block Diagram



The CY4532 EVK supports mobile power adapters, PC power adapters, power banks, car chargers, and other applications that primarily charge other devices via a Type-C interface. The kit is intended as an evaluation vehicle for USB Host systems that house a Type-C connector. Refer to the CY4532 EZ-PD CCG3PA EVK Guide for in-depth information about EVK use cases.

The CY4532 Main Board consists of the following:

- CCG3PA silicon (CYPD3171-40LQXIT)
- Type-C port
- Type-A ports
- MiniProg3 SWD header

Debug header The CY4532 Power Board consists of the following:

- DC Input
- Battery Input Terminal
- Mini-USB interface (to download firmware into CCG3PA)
- Type-C port (to enable flashing of an external CCG3PA-based hardware)
- Regulators (to support battery charging and provide power for the Type-C interface)
- Type-A charging

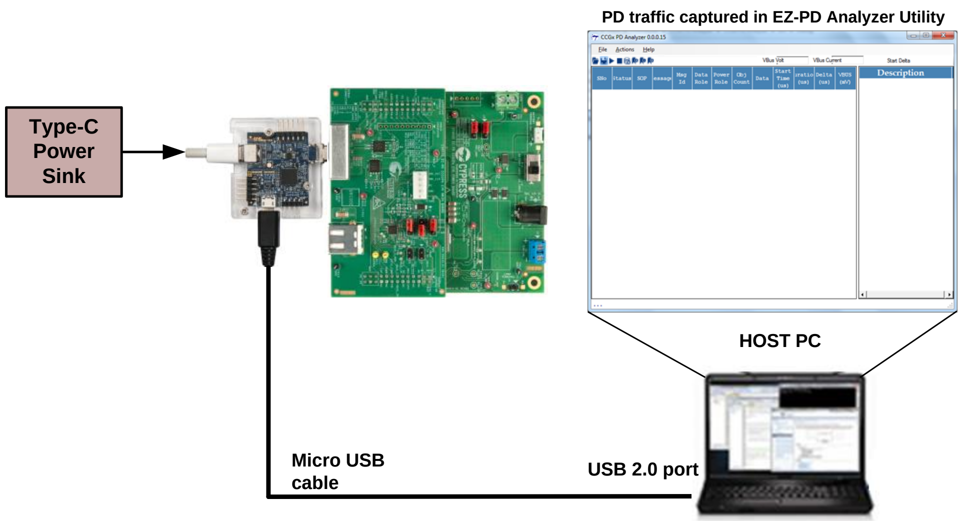
The Main Board is mounted on the Power Board using two sets of connectors.

The CY4532 CCG3PA silicon contains overvoltage protection (OVP) and overcurrent protection (OCP) circuitry for VBUS and supports EZ-PD CCG3PA device programming over the Type-C interface.

2.2 CY4500 EZ-PD Protocol Analyzer

Cypress's CY4500 EZ-PD Protocol Analyzer supports protocol analysis of the USB PD and USB Type-C specifications. It performs nonintrusive probing, and captures accurate protocol messages on the CC line. This analyzer consists of Cypress's programmable MCU (PSoC 5LP), which monitors the data on the CC line and sends the data to the host application over a USB interface. The Type-C plug and Type-C receptacle on this analyzer provide a pass-through for the Power Delivery (PD) packets transmitted between each Type-C PD connection. The processor MCU monitors these PD packets without disturbing the system and transfers them over the USB interface to a PC running the host application. Figure 4 shows the connections between the CY4500 EZ-PD Protocol Analyzer and the CY4532 CCG3PA EVK.

Figure 4. Connections Between a CY4500 EZ-PD Protocol Analyzer and CY4532 CCG3PA EVK



Downloading and installing the latest analyzer software setup (*CY4500Setup.exe*) from the Cypress website http://www.cypress.com/CY4500 installs the EZ-PD Analyzer Utility for running the analyzer, required drivers, and all relevant documentation. An overview of this utility is provided in the EZ-PD Analyzer Utility section.

3 CCG3PA Firmware Details and Build Environment

This section provides an overview of the firmware architecture, organization, and operation, and introduces PSoC Creator and the CCGx SDK, which are used as the build environment for custom firmware development. This section is followed by an overview of the software tools that can be used for development and debugging. Before covering the in-depth details regarding the firmware build environment and software tools, a brief summary of which tool can be used for what purpose is provided in Table 2 below.

Note that many configuration settings can be changed without the need for modifying the base firmware using the SDK. For a discussion of how the SDK fits into the CCG3PA ecosystem, see the CCG3PA Design Flow section.

Table 2. Firmware, Software Tools, and Their Purpose

Tool	Purpose	Output and Its Use	Described in
EZ-PD Configuration Utility	Used for modifying parameters stored in the configuration table area of the CCGx device flash. Also used for programming the CCGx device flash.	Configuration parameters can be read, modified, and saved for firmware development and debugging. Modified parameters can be updated in device flash intuitively without firmware development.	EZ-PD Configuration Utility

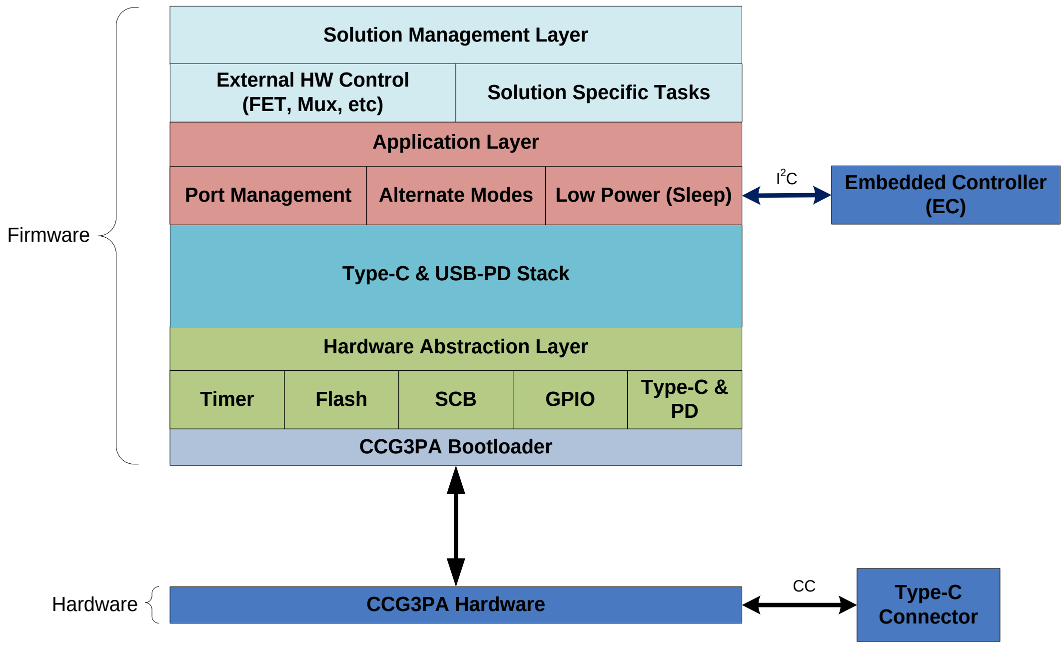
Tool	Purpose	Output and Its Use	Described in
CCGx SDK (with Custom CCG3PA Add-on)	The build environment for custom CCGx firmware development. It has application code with Type-C and USB-PD-compliant firmware, which can be used for further development and customization.	Compiling the application source project generates firmware binaries, which can be used to program/re-program respective CCGx devices.	CCG3PA SDK

3.1 CCG3PA Firmware Architecture Overview

Figure 5 shows a block diagram of the CCG3PA firmware architecture. The CCG3PA firmware architecture allows users to implement a variety of USB-PD applications using the CCG3PA device. It contains the following components:

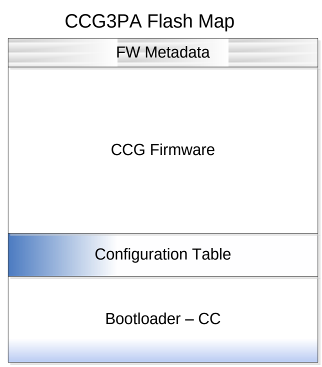
- **Hardware Abstraction Layer (HAL):** Low-level drivers for various hardware blocks of the CCG3PA device. This includes drivers for the Type-C and USB-PD block, Serial Communication Blocks (SCBs), GPIOs, flash module, and timer module.
- **USB Type-C and USB-PD Protocol Stack:** The complete USB-PD protocol stack that includes the Type-C and USB-PD port managers, USB-PD protocol layer, the USB-PD policy engine, and the device policy manager. The device policy manager is designed to allow all policy decisions to be made at the application level, either on an external Embedded Controller (EC) or in the CCG3PA firmware itself.
- **Application Layer:** This is the layer responsible for managing the functions of the PD port, handling alternate modes and power management, and manages the Host Processor Interface (HPI). It is further sub-categorized into the following components:
    - **Port Management:** This module handles all PD port management functions including the algorithm for optimal contract negotiations, source and sink power control, source voltage selection, port role assignment, and swap request handling.
    - **Alternate Modes:** This module implements the alternate mode handling for CCG as a downstream facing port (DFP) and upstream facing port (UFP). The module allows users to implement their own alternate mode support in both DFP and UFP modes.
    - **Low Power:** This module attempts to keep the CCG device in the low-power standby mode as often as possible to minimize power consumption.
- **Solution Management Layer:** This layer consists of the following components:
    - **External Hardware Control:** This is a hardware-design-dependent module that controls external hardware blocks such as FETs, regulators, and Type-C switches.
    - **Solution-specific tasks:** This is an application layer module where any custom tasks required by the user solution can be implemented.

Figure 5. CCG3PA Firmware Architecture Diagram



3.2 Flash Memory Organization

Figure 6. CCG3 Firmware Organization



The CCG3 device has 64 KB of flash that is divided into four sections as shown in Figure 6. The CC bootloader is used to upgrade the CCG3PA application firmware. It is allocated in a fixed area. The bootloader memory area can only be written using the SWD interface. The configuration table holds the PD configuration for the CCG3PA application, and is located at the beginning of the firmware binary. Many configuration parameters can be updated by modifying this table using the EZ-PD Configuration Utility. An overview of this utility is provided in the EZ-PD Configuration Utility section. The CCG firmware area is used for the CCG3PA firmware application. The metadata area holds the metadata about the firmware binary. The firmware metadata follows the definition provided by the PSoC Creator Bootloader Component, and includes the firmware checksum, firmware size, and the start address.

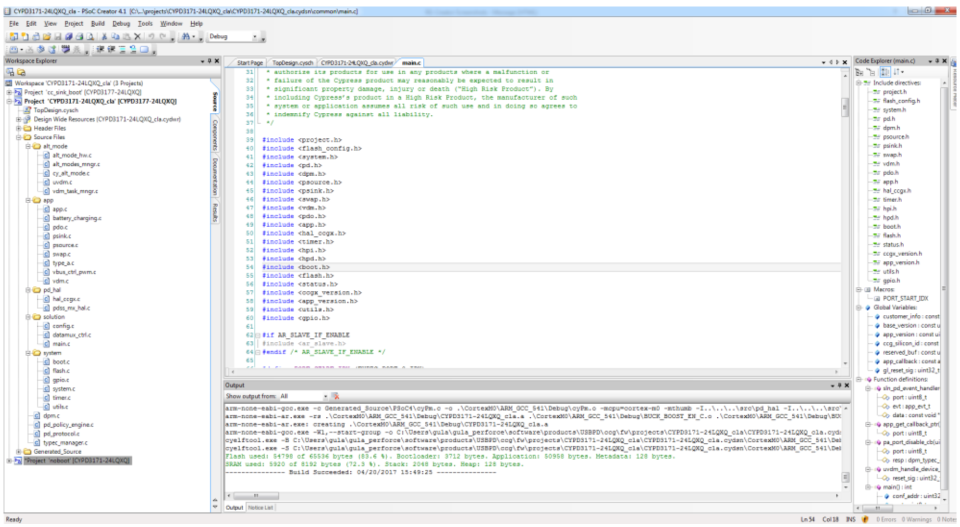
3.3 Firmware Build Environment

In cases where just modifying the configuration parameters is not sufficient customization for the intended application, the tool used for CCG3PA firmware development is PSoC Creator, which is a free, Windows-based Integrated Design Environment (IDE). PSoC Creator 4.1 or later and the CCGx SDK are required to edit, compile, download, and debug the firmware for the CCG3PA Notebook application, as shown in Figure 7. The PSoC Creator compiler tool chain is ARM GCC (build 493, provided along with the PSoC Creator build). Refer to the Help file in PSoC® Creator™ for more details on the PSoC Creator build environment.

Visit the PSoC Creator product page to download and install the latest version of PSoC Creator (4.1 or later). This web page also contains links to video training and additional documentation. Within the PSoC Creator tool, additional help is available via the following documents:

- **Quick Start Guide:** Choose **Help** > **Documentation** > **Quick Start Guide**. This guide gives you the basics for developing PSoC Creator projects.
- **System Reference Guide:** Choose **Help** > **System Reference Guides**. This guide lists and describes the system functions provided by PSoC Creator.

Figure 7. PSoC Creator IDE



3.3.1 CCG3PA SDK Add-on

The CCGx Software Development Kit (SDK) is a PSoC Creator project that allows you to harness the capabilities of Cypress's CCG families of Type-C Controllers. It provides a firmware stack compliant with Type-C and USB-PD specification along with the necessary drivers and software interfaces required to implement applications using CCG controllers. The CCGx SDK also includes reference projects implementing standard Type-C applications and documentation that guides users in customizing existing applications, or creating new ones. CCG3PA application development requires an additional add-on download from the CCG3PA website. For more information on the CCGx SDK, refer to the CCGx SDK User Guide.

In addition to the CCGx SDK, a separate add-on is required to enable firmware modifications for CCG3PA devices. In general, the CCGx SDK consists of the following basic components:

- Source Code
  - USB PD-compliant stack in a pre-compiled library form, supporting all standard power profiles and PPS
  - Firmware sources for other blocks
  - Reference application
  - Control of integrated UFP (R_D), and DFP (R_P) termination resistors
- Firmware Binaries
  - Application-specific *.hex* and *.cyacd* files
- CCG3PA Application Projects[1]

[1] More Details for these applications can be found in the Resources list in Section 1.4.

- Power Adapter Application with Direct Feedback Control
- Power Adapter Application with Opto Coupler
- Car Charger Application
- Power Bank Application

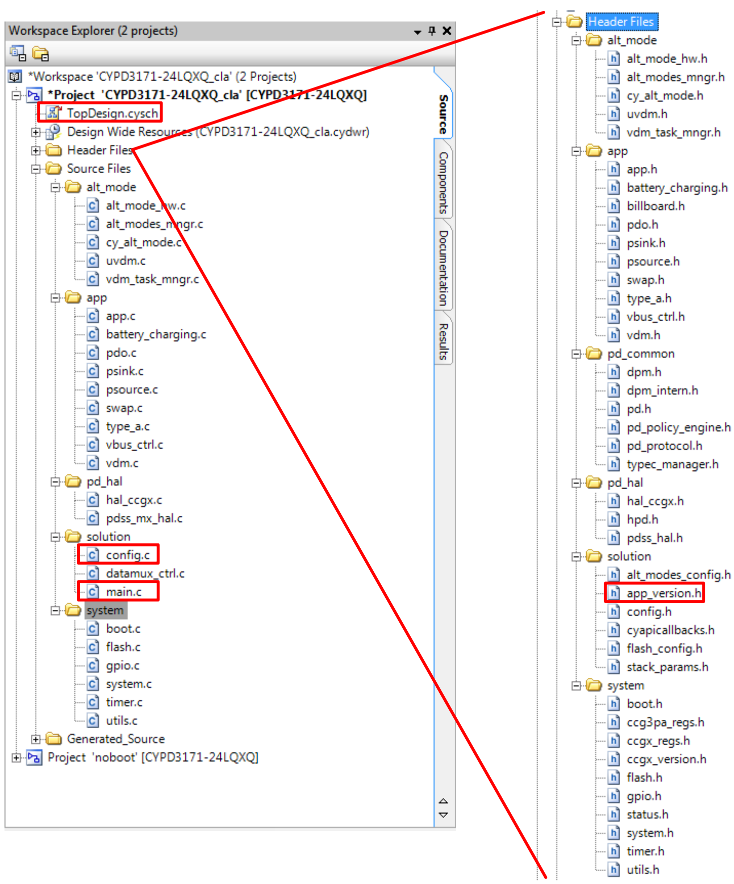
- Supporting Documentation
  - CCGx SDK User Guide
  - CCG3PA SDK Add-on User Guide
  - Firmware API Reference Guide
  - Release Notes

3.3.2 PSoC Creator Project Structure Overview

The CCGx SDK CCG3PA add-on includes a reference firmware project for a notebook power adapter application with the CCG3PA device. Figure 8 shows the PSoC Creator workspace file structure for a CCG3PA device-based power adapter application. PSoC Creator generates bootloadable *.cyacd* files and a Cypress format *.hex* file every time a project is successfully compiled and built.

In every PSoC Creator project, a chip-level design schematic diagram is included. This schematic is located in the *TopDesign.cysch* file. The firmware version is controlled by an ‘application type’ string and ‘application major’ and ‘application minor’ numbers. This information is stored in the *app_version.h* file of the PSoC Creator project. The application string and the version numbers can be modified by customers based on their requirements.

Figure 8. PSoC Creator Workspace Structure for CCG3PA Notebook Application



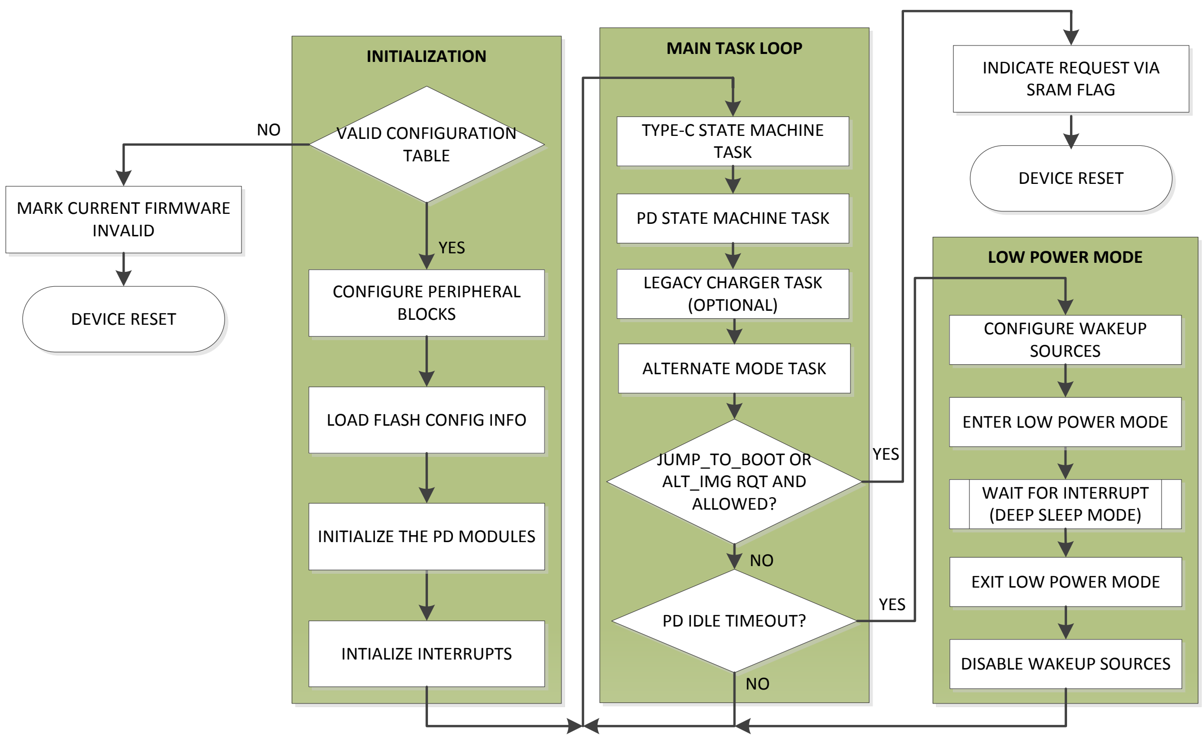
3.4 Firmware Configurable Features

The notebook application firmware in the CCGx SDK can be modified for customer projects using compile-time configurable features. These parameters control optional features of the device for different configurations. Details on these parameters are available in the EZ-PD User Guide.

3.5 Firmware Operation

The code flow for the application is implemented in the *Source Files\Solution\main.c* file. Figure 9 shows the flow diagram of the firmware operation. Based on this, it is clear that the firmware implementation is a simple round-robin loop that services each of the tasks that the application has to perform. All PD management and Vendor-Defined Message (VDM) handling are encapsulated in the task handlers in the CCG3PA firmware stack.

Figure 9. CCG3PA Power Adapter Firmware Flow Diagram



See CCGx SDK User Guide for more details on the CCG3PA SDK Package firmware architecture, firmware APIs, and getting started with the CCG3PA SDK Package. You can use this guide also for customizing the CCG3PA firmware application, that is, modifying and updating the PSoC Creator project for a different hardware design, building and debugging the project, and programming the modified firmware in the CCG3PA device.

3.6 Programming Firmware in CCG3PA Devices

There are two methods to program CCG3PA devices:

- Using the EZ-PD Configuration Utility (utilizing the bootloader)
- Using a MiniProg3 device and PSoC Creator or PSoC Programmer software (utilizing the SWD interface)

The EZ-PD Configuration Utility requires a *.cyacd* file as the input for the firmware binary file to be programmed. Refer to Chapter 4 of the CY4532 EZ-PD CCG3PA Evaluation Kit User Guide for more details on how to program CCG3PA devices using the EZ-PD Configuration Utility. The PSoC Creator/PSoC Programmer uses a *.hex* file for programming all devices. Refer to the knowledge base article KBA96477 for more details on how to program CCG3PA devices using PSoC Creator/PSoC Programmer along with a MiniProg3 device.

For detailed information on bootloader options, how to program the bootloader if necessary, how to configure the EVK for power bank applications and more, refer to the CCG3PA datasheet.

4 Software Tools for CCG3PA Application Firmware Development and Debugging

This section provides an overview of the EZ-PD Configuration Utility and the EZ-PD Analyzer Utility. In general, the tools can be categorized as follows:

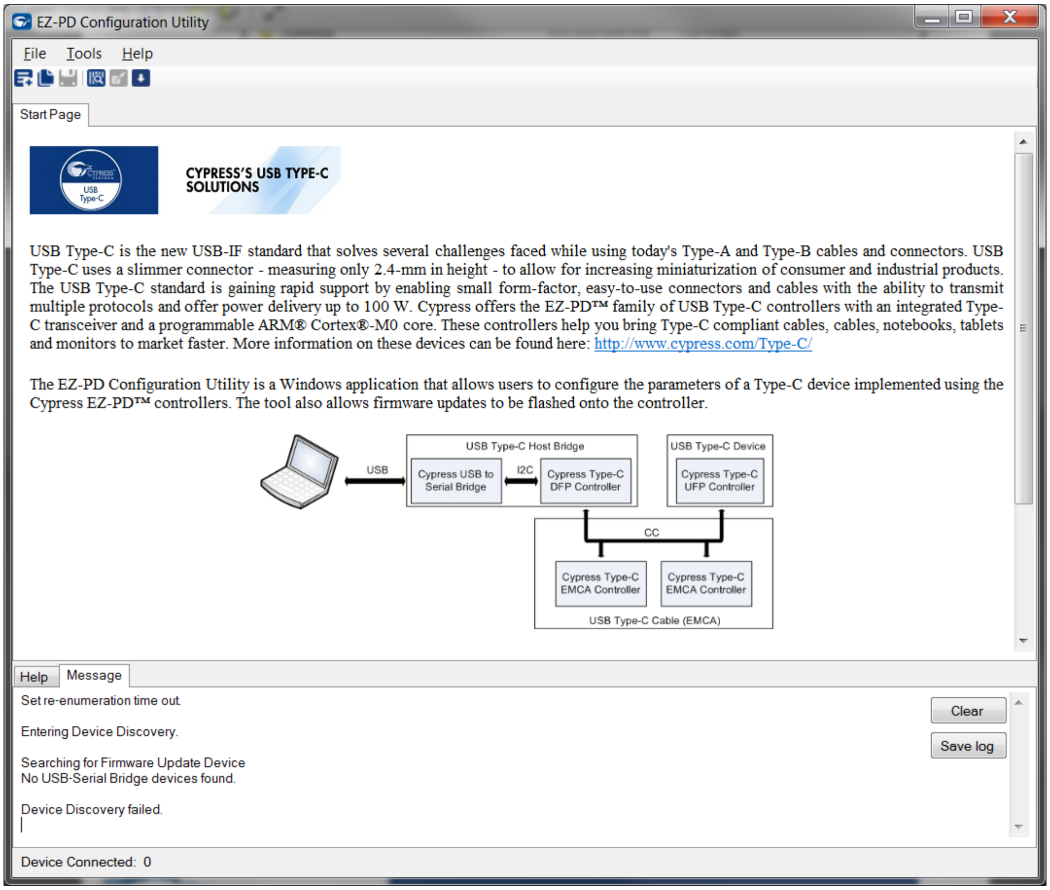
- **Firmware development and programming tools**
  - EZ-PD Configuration Utility: The EZ-PD Configuration Utility is used to read, modify, and update the configuration parameters of a CCGx device over the CC line. It is also used to update the application firmware of the CCGx device.
  - PSoC Creator (CCG3PA devices are supported from version 4.1 or later): As described in the Firmware Build Environment section, PSoC Creator is used to modify, debug, and program the firmware into the CCG3PA device. This option is required only if the functionality of the CCG3PA firmware is being modified from that of the standard firmware, or if the firmware is being modified for a different hardware design.
- **Debugging tool for CCGx applications**
  - EZ-PD Analyzer Utility: The EZ-PD Analyzer Utility along with a CY4500 EZ-PD Protocol Analyzer is used to capture PD messages between the CCG3PA device and an attached Type-C device with the firmware application that is running. An overview of its functionality is provided in the CY4500 EZ-PD Protocol Analyzer section.

4.1 EZ-PD Configuration Utility

The EZ-PD Configuration Utility is a Windows application that configures the parameters stored in the configuration table areas of the internal flash memory of the CCGx device. These parameters can be set based on your application or system requirements. The utility allows you to intuitively select and configure the parameters for your application and thus saves time on firmware development. This utility also allows programming of the firmware applications.

You can download and install this tool from the Cypress website. After installation, launch the utility from the following location in the Windows Start Menu: **All Programs** > **Cypress** > **EZ-PD Configuration Utility** > **EZ-PD Configuration Utility**. Figure 10 shows the utility running on a Windows machine.

Figure 10. EZ-PD Configuration Utility Start Page

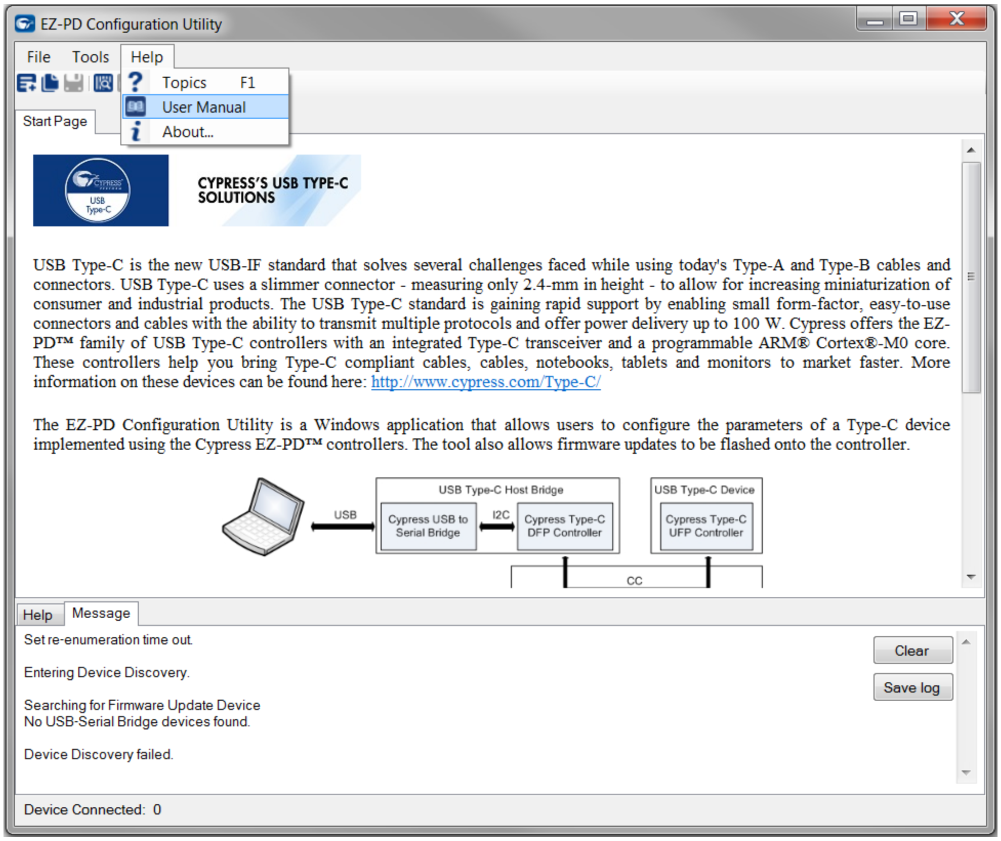


The EZ-PD Configuration Utility can be used to update the application firmware and configure a CCGx device. The utility shows the target application for the CCG3PA device as a power adapter. The workflow for configuring any CCGx device is completed in three stages:

1. **Create configuration:** Create a new configuration from the **File** menu of the utility, or read an existing configuration to be modified.
2. **Select parameters:** Select the parameters available for the target application such as a notebook.
3. **Device configuration:** Program the device flash using the **Configure Device** option.

See the EZ-PD Configuration Utility User Manual for more details on firmware update and configuration of the device, found in the application's Help menu.

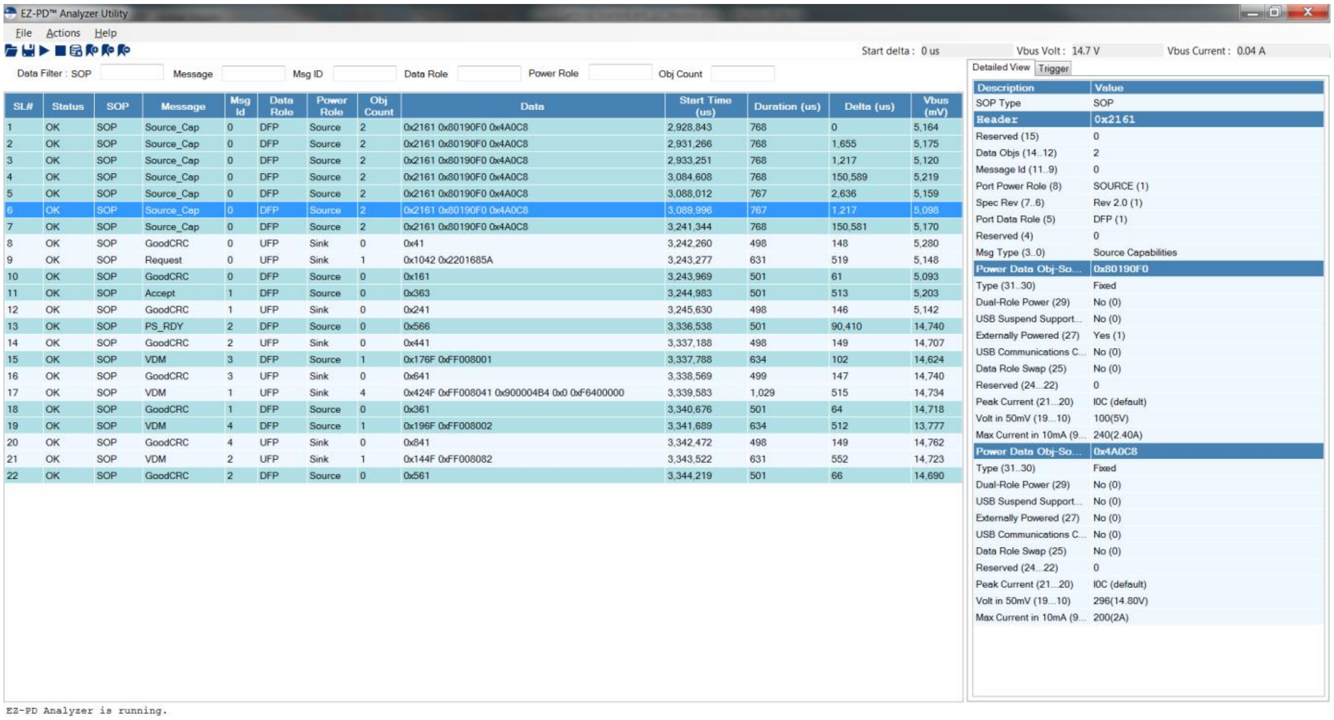
Figure 11. Opening the EZ-PD Configuration Utility User Manual



4.2 EZ-PD Analyzer Utility

The EZ-PD Analyzer Utility is a software application that is run along with a CY4500 EZ-PD Protocol Analyzer. When this utility is run, it shows PD messages over CC (for example, PR_SWAP, DR_SWAP, and PDOs) as shown in Figure 12 while the CCG3PA device is establishing a PD contract with the connected Type-C device.

Figure 12. EZ-PD Analyzer Utility Showing PD Packets Captured on CC Line



Whenever a Type-C device is connected to the Type-C port of the CY4532 CCG3PA EVK, a PD contract is established between the CCG3PA device and the attached Type-C device. The flow of CC messages as well as source and sink Power Data Objects (PDOs—used to expose source/sink port's power capabilities) between the power port partners can be monitored on the PC by using the EZ-PD Protocol Analyzer and running the EZ-PD Analyzer Utility. This information helps in debugging CCG3PA firmware if there are any issues (errors, delays, no contract, and so on) related to establishing the power contract between the CCG3PA device and the attached Type-C device.

5 CCG3PA Configuration Parameter Modification Example

Many CCG configuration parameters are stored in the configuration tables in the device's internal flash memory. You may need to make changes to these parameters based on system requirements and the end application. This section provides a simple example of a Type-C ecosystem to describe how to change the configuration parameters using the EZ-PD Configuration Utility and how to verify the changes using the CY4500 EZ-PD™ Protocol Analyzer.

Consider the example of a Type-C device connected to the CY4532 CCG3PA EVK. Assuming that this Type-C device supports a custom PDO, the power contract between it and the CCG3PA device of the EVK will be successfully established utilizing the PDO only if the CCG3PA device supports the same custom PDO. The EZ-PD Configuration Utility allows you to modify parameters such as the sink or source PDOs without making any changes to the base firmware so that the CY4532 CCG3PA EVK can successfully establish a power contract with the device.

From a top-level perspective, the following are the steps to modify the CCG3PA device's configuration parameters. The following sections explain each step in detail.

1. Test the CY4532 CCG3PA EVK in the default configuration with a Type-C device to identify the established power contract.
2. Update the configuration parameters (add a sink PDO supported by the device such as 14.8 V, 2 A) by using the EZ-PD Configuration Utility.
3. Transmit the updated configuration parameters into the CCG3PA device by using the EZ-PD Configuration Utility and verify.

4. Retest the CY4532 CCG3PA EVK setup with the Type-C device to verify the updated PDOs by using the CY4500 EZ-PD Protocol Analyzer.

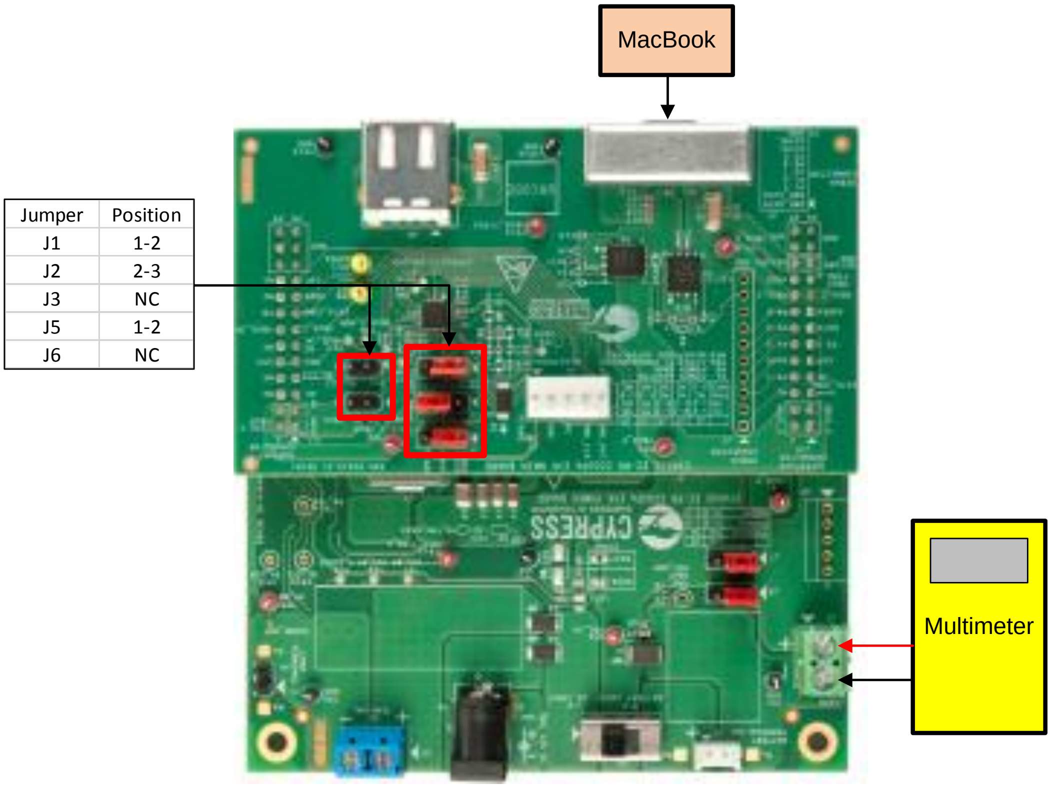
The CCG3PA kit is also able to run in power bank mode. Power banks take power from power adapters and disperse it to multiple devices. One CY4532 CCG3PA EVK can be used as a power adapter while a second CY4532 CCG3PA EVK can be used as the power supply. The voltage output of the power bank can then be confirmed with a multimeter.

5.1 Test CY4532 CCG3PA EVK Setup with the Default Configuration and Type-C Power Sink (Apple MacBook)

This section describes the steps to observe the behavior when the CY4532 CCG3PA EVK with the default configuration parameters is connected to a Type-C device. This example will use an Apple MacBook, but any Type-C power consumer should work. When the MacBook is connected to the Type-C port of the CY4532 CCG3PA EVK, the CCG3PA device of the CY4532 CCG3PA EVK will be able to establish a power contract with the MacBook. Since the CCG3PA device is configured for a limited set of PDOs, which may or may not be supported by the MacBook, a power contract negotiation is required. In this case, the power adapter will provide only 5.0 V because that is the only PDO that is supported by the MacBook and the CCG3PA in its default configuration. The voltage measured manually at the output header J4 can also be verified by using the CY4500 EZ-PD Protocol Analyzer. Ensure that the latest version of the CY4500 EZ-PD Protocol Analyzer is installed.

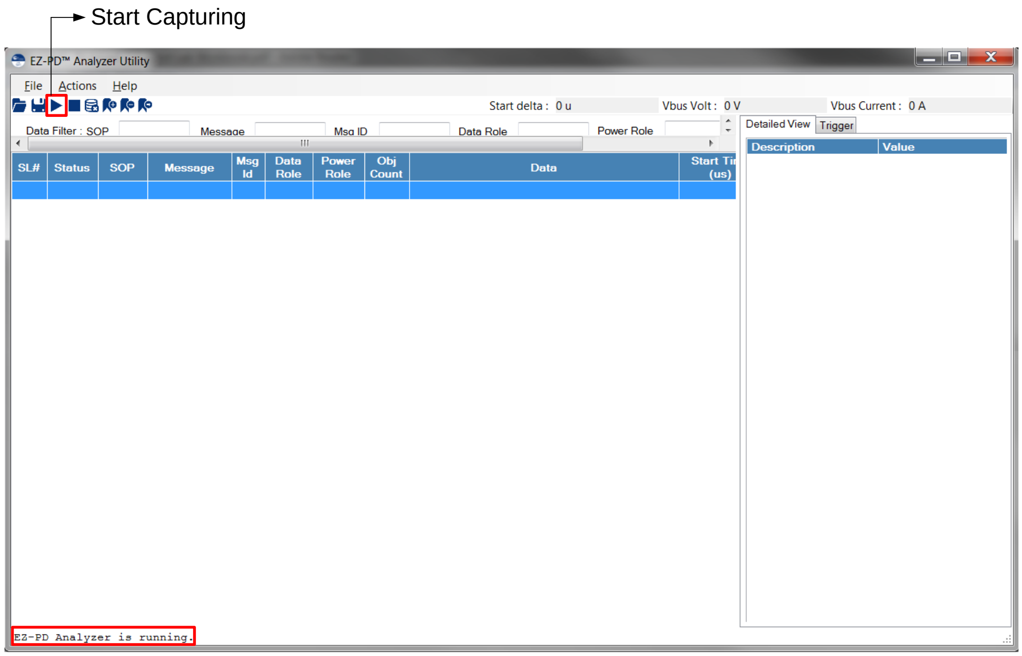
1. Ensure that the CCG3PA device of the CY4532 CCG3PA EVK is programmed with the latest default binaries provided on the CCGx SDK webpage. As shown in Figure 13, ensure that the jumpers are set to the proper Power Adapter position (the positions are also listed on the silkscreen on the board).
2. Connect a MacBook to the CY4532 CCG3PA EVK, as shown in Figure 13. Connect a multimeter to power output header J5 of the CY4532 CCG3PA EVK. Connect the 12-V power adapter provided with the CY4532 kit to the DC power jack (J3) of the power board and move switch SW1 to the ON position.
3. Verify that a 5.0 V contract has been established by measuring the voltage on the power output header J4 of the CCGPA baseboard using a multimeter.

Figure 13. CY4532 CCG3PA Measuring VBUS on the MacBook



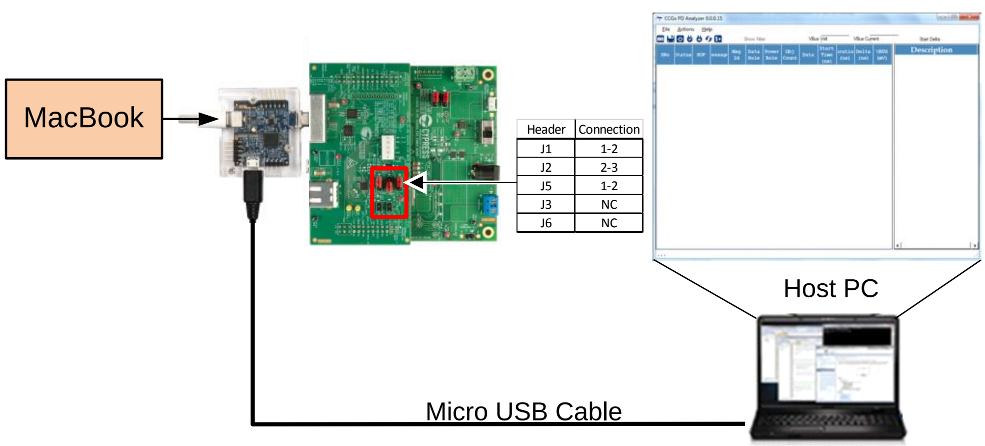
4. This output voltage can also be verified using the CY4500 EZ-PD Protocol Analyzer and capturing a CC trace.
5. Disconnect the MacBook from the EVK and connect CY4500 EZ-PD Protocol Analyzer to the PC (USB Host) using a Micro-USB cable.
6. Connect the Type-C plug of the CY4500 EZ-PD Protocol Analyzer to the Type-C port of the CCG3PA EVK.
7. Launch the EZ-PD Analyzer Utility from **Windows** > **All Programs** > **Cypress** > **EZ-PD Analyzer Utility** > **EZ-PD Analyzer Utility** and click on the **Start Capturing** icon shown in Figure 14 to start capturing the CC traffic.

Figure 14. Start Capturing Traffic in EZ-PD Analyzer Utility



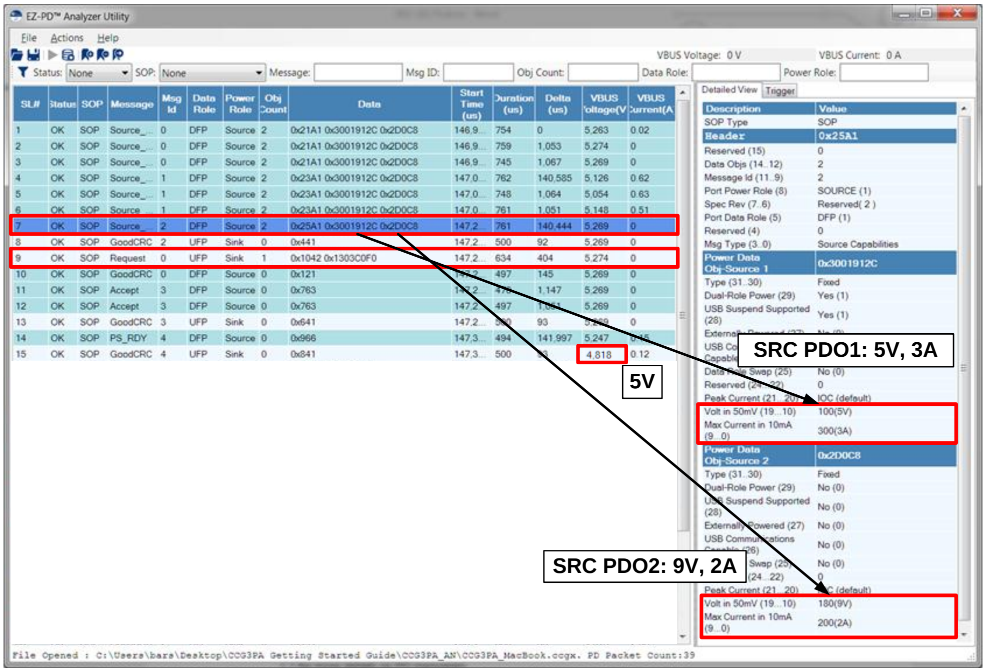
8. Connect the MacBook to the Type-C receptacle of the CY4500 EZ-PD Protocol Analyzer (see Figure 15) using a Type-C cable. Observe that the PD traffic is being captured on the EZ-PD Analyzer Utility. The utility running on the host PC should look similar to that in Figure 16.

Figure 15**.** Connecting CY4500 EZ-PD Protocol Analyzer to CY4532 CCG3PA EVK



9. A successful PD contract can be seen from the PD message sequence. From Figure 16, it is clear that the Type-C device used in this example is a UFP (Sink/Power Consumer) and the Type-C port of the CY4532 CCG3PA EVK is a DFP (Source/Power Provider). Looking at the 'VBUS' column in Figure 16, it can be seen that the VBUS voltage is still set to 5 V at the end of the power contract negotiation.

Figure 16. EZ-PD Analyzer Utility Showing PD Packets Captured on CC Line



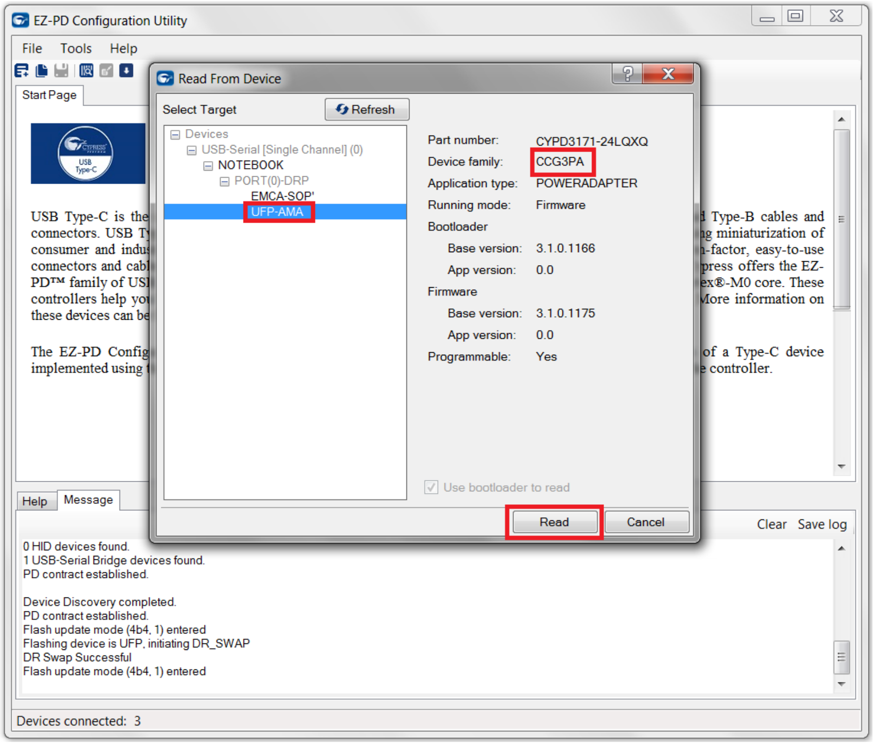
10. The CC trace capture shown in Figure 16 shows that a power contract for 5 V is established, and the CY4532 CCG3PA EVK is providing 4.8 V to the device. During the power negotiation phase, the power provider (that is, the CY4532 CCG3PA) sends the **Source_Capabilities** message to the attached power consumer device (that is, the MacBook). The **Source_Capabilities** message in Figure 16 shows the custom PDOs (**Power Data Obj-Source 1** and **Power Data Obj-Source 2**) supported by the Type-C power adapter.
11. The **Request** message after the **Source_Capabilities** message is always sent by the power consumer device to request power. Because the Type-C device can establish a power contract for 5 V but not the 9 V that is also offered, the MacBook requests the PDO object at position 1, which corresponds to the 5-V PDO. Refer to USB Power Delivery (PD) Spec Revision 2.0, Version 1.1 for more details on PD messages.
12. Looking at the 'VBUS' column in Figure 16 and as mentioned in step 9, it can be seen that the VBUS voltage is set to 4.8 V at the end of the power contract negotiation. This confirms that at the initial stage of the setup, the power contract of 4.8 V is established between the CY4532 CCG3PA EVK and the Type-C device.

5.2 Modify Configuration Parameters Using EZ-PD Configuration Utility

This section guides you through modifying one of the parameters using the EZ-PD Configuration Utility. By adding a higher, compatible power profile to the configuration of the CCG3PA firmware, the device will be able to accept a higher, faster charging voltage value. Ensure that the latest version of the EZ-PD Configuration Utility is installed on your computer.

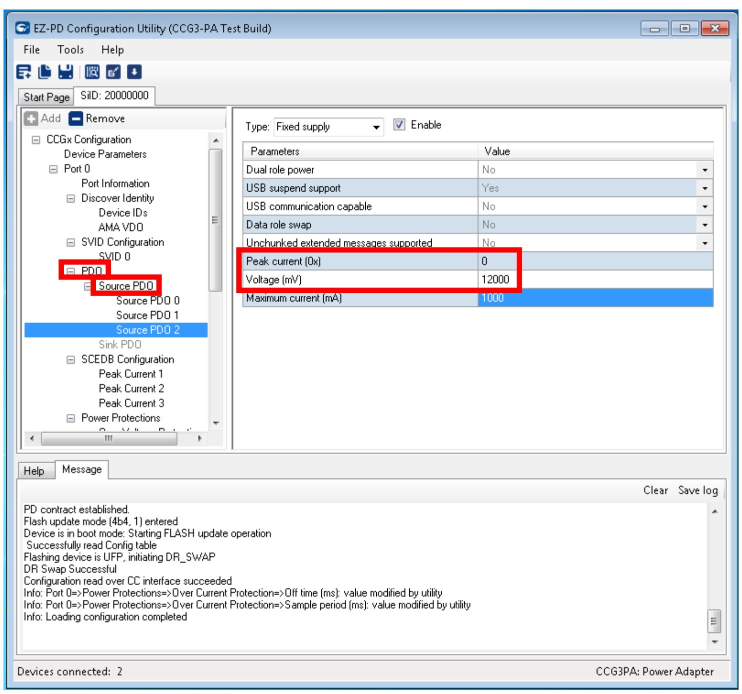
1. Make sure that your device's jumpers are set to Power Adapter mode using the jumper location description shown on the silk screen on the CY4532 Main Board. Connect a mini USB cable between your PC's USB port and J11 on the CY4532 Power Board. Also, connect the Type-C ports of the CY4532 CCG3PA EVK's main board and power board using a Type-C cable. Connect the 12-V power adapter to the DC power jack (J3) of the power board and move switch SW1 to the ON position.
2. Open the EZ-PD Configuration Tool. The GUI should report one device connected in the status bar of the GUI.
3. Go to **File** > **Read from device**. Ensure that the 'UFP-AMA' device is selected and then click **Read**. The EZ-PD GUI will be populated with the firmware settings read from the CCG3PA device.

Figure 17. Using the EZ-PD Tool to Read CCG3PA Device Configuration



4. Select the power profile PDOs under **CCGx Configuration** > **Port 0** > **PDO** > **Source PDO**. Click on the **Add** button in the upper left and configure the new source PDO to be fixed supply, Peak current 0, Voltage (mV) 12000, and maximum current (mA) 1000. See Figure 18 for an example.

Figure 18. Configuring a Third PDO to Offer MacBook's Ideal Power Profile



5. Save your project somewhere that is easy to remember.

5.3 Configure CCG3PA Device with Updated Configuration Parameters Using the EZ-PD Configuration Utility

The next step in modifying the CCG3PA device's behavior is to replace the firmware that was read from the device with the modified firmware from Section 5.2. Firmware updates of this scale are the most common and can be done in the EZ-PD Configuration Utility. Modifications that will be running new, custom code will require PSoC Creator with the CCG SDK 3.0.2 add-on and special CCG3PA supplement (available from your local FAE).

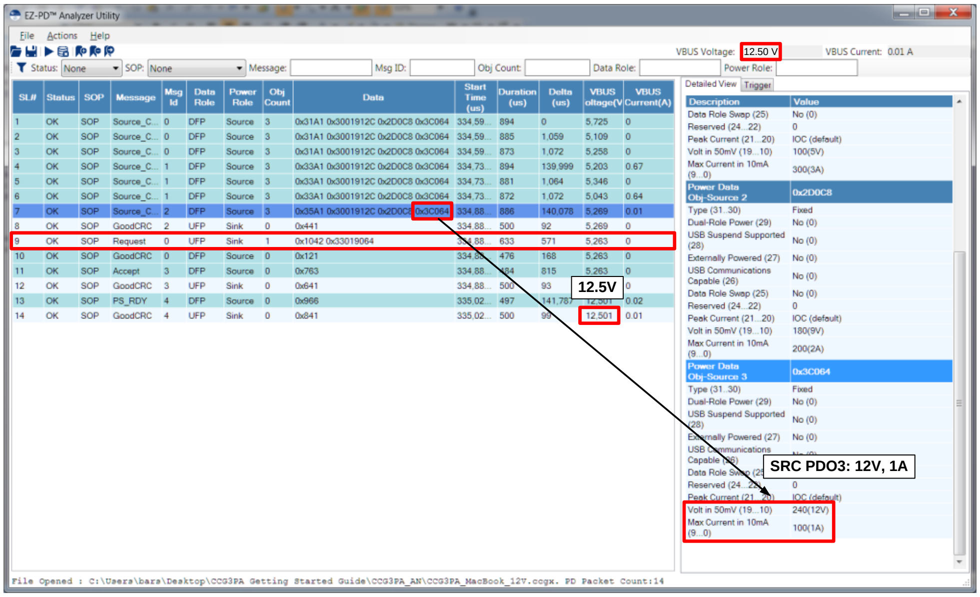
1. If you don't have the modified project open from Section 5.2, open the project in the EZ-PD Configuration Utility.
2. Ensure that the connection between the CY4532 CCG3PA EVK and the PC running EZ-PD Configuration Utility is the same as described in step 1 of Section 5.2.
3. Click **Tools** > **Configure device**. The CCG3PA device should show up as "UFP-AMA" under **Select Target** in a similar fashion as when the configuration was read from the device. Point to the configuration file saved at the end of Step 5 of Section 5.2. Click **Program;** the EZ-PD Configuration Utility will load the modified parameters onto the device and a message "**Flash updated successfully**" will pop up. Recycle power to the CY4532 kit by toggling switch SW1 on the power board.

5.4 Re-test CY4532 CCG3PA EVK Setup with Modified Configuration and Type-C Device

This section is very similar to the steps taken in Section 5.1. Some of the details will be removed to avoid redundancy, but the exercise should be performed almost identically while observing different results. The different voltages advertised should lead to a new voltage being output on the J4 terminal.

1. Ensure that the CCG3PA device of the CY4532 CCG3PA EVK has the configuration that was modified in Section 5.3. Confirm that the jumpers are still correct by consulting the silk screen on the CCG3PA EVK.
2. Connect the EVK, EZ-PD Protocol Analyzer, and multimeter similar to how they were configured in Section 5.1. Make sure that the MacBook is **not** connected via the EZ-PD Protocol Analyzer to the Type-C port yet.
3. If it is not already open, launch the EZ-PD Analyzer Utility. Click on the **Start Capturing** icon shown in Figure 14 to start capturing the CC traffic.
4. Connect the MacBook to the Type-C receptacle of the CY4500 EZ-PD Protocol Analyzer (see Figure 15). Observe that the PD traffic is being captured on the EZ-PD Analyzer Utility.
5. A successful PD contract can be seen from the PD message sequence. Just as before, the Type-C device used in this example is a UFP (Sink/Power Consumer) and the Type-C port of the CY4532 CCG3PA EVK is a DFP (Source/Power Provider). Looking at the 'VBUS' column in Figure 19, it can be seen that the VBUS voltage is set to 12 V at the end of the power contract negotiation as a result of the added PDO being accepted. Thus, a successful power contract is established at 12-V between the CY4532 CCG3PA EVK and the Type-C device.

Figure 19. Establishing a Connection with the New 12-V PDO



Document History

Document Title: AN218179 – Getting Started with EZ-PD™ CCG3PA

Document Number: 002-18179

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	5721256	BARS	5/17/2017	New application note.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709