EZ-PD™ Configuration Utility user manual

About this document

Scope and purpose

This document describes the usage of EZ-PD™ Configuration Utility, a Microsoft Windows application that guides through the process of configuring and programming CCGx devices.

Intended audience

This document is intended for embedded developers using EZ-PD™ CCGx power/host software development kits.
Table of contents

About this document.............................................................................................................. 1
Table of contents.................................................................................................................... 2
1 Introduction .......................................................................................................................... 5
  1.1 EZ-PD™ device family .................................................................................................. 5
  1.2 EZ-PD™ Configuration Utility ..................................................................................... 5
  1.3 Software pre-requisites ............................................................................................... 5
2 EZ-PD™ Configuration Utility overview .......................................................................... 6
  2.1 Usage flow .................................................................................................................. 7
    2.1.1 Creating configuration ......................................................................................... 7
    2.1.2 Selecting parameters ......................................................................................... 9
    2.1.3 Configuring device ........................................................................................... 10
3 Hardware setup .................................................................................................................... 11
  3.1 Programming methods ............................................................................................... 11
    3.1.1 Programming over I2C ....................................................................................... 11
    3.1.1.1 Programming Type-C to display dongle controllers ......................................... 12
    3.1.2 Programming over CC channel .......................................................................... 13
  3.2 Programming setup ..................................................................................................... 15
    3.2.1 Programming EMCA (cables) ............................................................................. 15
    3.2.2 Programming Type-C dongles and Billboard devices .......................................... 15
    3.2.3 Programming Notebook Type-C port controllers .............................................. 16
    3.2.4 Programming Type-C power adapters ............................................................... 16
    3.2.5 Programming manufacturing test kit hardware ................................................... 17
4 GUI usage ............................................................................................................................ 18
  4.1 USB PD device configuration ....................................................................................... 19
    4.1.1 Device parameters ............................................................................................. 20
    4.1.2 Discover identity ............................................................................................... 20
    4.1.3 SVID configuration ........................................................................................... 23
    4.1.4 PDO configuration ............................................................................................ 24
    4.1.5 Port information ............................................................................................... 26
    4.1.6 Billboard parameters ......................................................................................... 27
      4.1.6.1 Designs with External Billboard controller ................................................. 28
      4.1.6.2 Designs with On-Chip Billboard interface ................................................. 28
    4.1.7 SCEDB configuration .......................................................................................... 30
      4.1.7.1 Peak current configuration ........................................................................... 31
    4.1.8 SKEDB configuration ......................................................................................... 32
    4.1.9 Application configuration .................................................................................. 32
    4.1.10 Protection parameters ..................................................................................... 33
        4.1.10.1 VBUS over voltage protection ................................................................. 33
        4.1.10.2 VBUS over current protection ................................................................. 34
        4.1.10.3 VBUS under voltage protection ............................................................... 34
        4.1.10.4 Short circuit protection .......................................................................... 35
        4.1.10.5 Over temperature protection ................................................................. 35
        4.1.10.6 VCONN over current protection .............................................................. 36
    4.1.11 Power settings .................................................................................................... 36
    4.1.12 Battery configuration ......................................................................................... 37
    4.1.13 Charging configuration ....................................................................................... 37
        4.1.13.1 Source setting ......................................................................................... 37
# Table of contents

4.1.14  BCR configuration ........................................................................................................... 39
4.1.15  Type-A configuration ...................................................................................................... 40
4.1.15.1 Power setting ............................................................................................................... 40
4.1.15.2 Charging configuration ............................................................................................... 40
4.1.16  Auto configuration ......................................................................................................... 41
4.1.16.1 Policy information ...................................................................................................... 41
4.1.16.2 Input voltage throttling information .......................................................................... 41
4.1.16.3 Temperature throttling information .......................................................................... 41
4.1.16.4 VIN fault voltage ....................................................................................................... 42
4.1.17  Thunderbolt host configurations ................................................................................. 42
4.1.18  Base alternate modes configuration .............................................................................. 43
4.1.19  Custom alternate mode configuration .......................................................................... 44
4.1.20  Custom host configuration ........................................................................................... 44
4.1.21  ICL/TGL configuration ................................................................................................. 44
4.1.22  AMD configuration ....................................................................................................... 45
4.1.23  User parameters ........................................................................................................... 45
4.2  Billboard device configuration ............................................................................................ 46
4.2.1  Device parameters ......................................................................................................... 46
4.2.2  Flashing parameters ...................................................................................................... 47
4.2.3  Billboard parameters .................................................................................................... 47
4.2.4  Alternate modes ............................................................................................................ 48
4.2.5  GPIO configuration ........................................................................................................ 49
4.3  Saving the configuration ..................................................................................................... 50
4.4  Reading configuration from firmware file .......................................................................... 50
4.5  Saving configuration to firmware file ................................................................................ 51
4.6  Configuring the device ....................................................................................................... 52
4.6.1  Extensions for dual firmware devices ............................................................................ 53
4.6.1.1 Single firmware mode (legacy boot mode) ................................................................. 53
4.6.1.2 Dual firmware mode .................................................................................................. 54
4.6.2  Extensions for hybrid architecture applications ............................................................ 55
4.7  Reading configuration from device .................................................................................... 57
4.7.1  Extensions for dual firmware devices ............................................................................ 58
4.7.2  Extensions for hybrid architecture applications ............................................................ 59
4.8  Firmware update ................................................................................................................. 59
4.8.1  Extensions for dual firmware devices ............................................................................ 60
4.8.2  Extensions for hybrid architecture applications ............................................................ 61
4.9  Manufacturing test kit option ............................................................................................. 61
4.9.1  Download MTK firmware ............................................................................................. 63
4.9.2  Configure MTK tester ID ............................................................................................... 64
4.9.3  Create EMCA configuration .......................................................................................... 66
4.9.4  MTK operation .............................................................................................................. 66
4.10  Power adapter batch programming ................................................................................... 70
4.10.1  Power adapter batch programmer operation ................................................................. 72
4.11  Utility options .................................................................................................................... 74
4.11.1  Enable device verbose logs ......................................................................................... 75
4.11.2  Allow multiple PDOs of the same voltage ................................................................. 75
4.11.3  Re- enumeration timeout ............................................................................................ 75
4.11.4  UFP restart timeout .................................................................................................... 75
4.11.5  HPI slave address ........................................................................................................ 75
4.11.6  Default flashing VID and index ................................................................................... 75
## Table of contents

4.11.7  C File output template ................................................................. 76  
4.11.8  Sort PDOs....................................................................................... 76  
4.12    Firmware download for USB serial.................................................. 76  
4.13    Help............................................................................................... 77  

Revision history......................................................................................... 78  
Disclaimer................................................................................................... 80
1 Introduction

1.1 EZ-PD™ device family

EZ-PD™ controllers are fully programmable USB Type-C port controllers that are suited for all USB PD applications such as cables, dongles, notebooks, monitors, and power adapters.

Currently the CCG2, CCG3, CCG4, CCG3PA, CCG3PA-N, CCG3PA2, PAG1S, PAG2S, CCG5, CCG5C, CCG6, ACG1F, CCG6DF, CCG6SF, CCG7D, CCG7S, CMG1, CMG2 generations of EZ-PD™ controllers are available. These controllers are collectively called CCGx controllers in this document.

1.2 EZ-PD™ Configuration Utility

The EZ-PD™ CCGx controller is a highly configurable and programmable solution. The chip can be configured using parameters stored in the internal flash memory. These parameters are to be chosen and programmed by customers according to their use cases and requirements.

The EZ-PD™ Configuration Utility is a Microsoft Windows application that guides the CCGx user through the process of configuring and programming the chip. The utility works in tandem with the supplied hardware, which hosts the CCGx controllers along with a USB interface.

The Graphical User Interface (GUI) of the EZ-PD™ Configuration Utility allows you to intuitively select and configure the parameters for your application.

1.3 Software pre-requisites

Table 1 lists the software pre-requisites needed to run the configuration utility.

<table>
<thead>
<tr>
<th>#</th>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operating System</td>
<td>Microsoft Windows 7 or later</td>
</tr>
<tr>
<td>2</td>
<td>.NET Framework</td>
<td>.NET Framework 4.0 or later</td>
</tr>
<tr>
<td>3</td>
<td>Microsoft Visual C++ Redistributable</td>
<td>2015-2022</td>
</tr>
</tbody>
</table>

Note: Microsoft .NET framework can be downloaded and installed from Microsoft.com.
2 EZ-PD™ Configuration Utility overview

You can install the EZ-PD™ Configuration Utility through a single click installer. After successful installation, you can start the application from the desktop icon for Windows 10.

![Welcome screen with version information](image)

**Figure 1** Welcome screen with version information
2.1 Usage flow

The utility usage flow typically consists of three steps:

1. Creating configuration
2. Selecting parameters
3. Configuring device

In the first stage, you select the target application to be configured. The applications are categorized by device family. To select an application that is available in a specific device family, select the corresponding family from the drop-down menu.

After selecting the device type, you move to the Selecting parameters stage. All configurable parameters of the active application and their default values are shown on the Configuration tab of the GUI. You can view, modify, and save the parameters from the Configuration tab.

After selecting and saving the parameters, you move to the final Configuring device stage. In this stage, the saved configuration is programmed onto the internal flash memory on the EZ-PD™ CCG controller.

2.1.1 Creating configuration

To start a new configuration, follow the menu path File > New. In the Select Device Type dialog, select the CCGx device from the Device type drop-down list. The USB PD applications supported by the selected CCG device are listed.
Since the USB PD applications supported vary across the product families, select the CCG product family first before selecting the type of USB PD application. The utility filters the applications available for selection based on the product family selected.

Table 2 lists the CCGx applications supported by the current version of the utility.

<table>
<thead>
<tr>
<th>Product family</th>
<th>Electronically marked Type-C cables (passive or active EMCA)</th>
<th>Type-C to display (DisplayPort, HDMI or VGA) dongles</th>
<th>Type-C port controllers for Notebooks</th>
<th>Type-C power applications (power adapter, power bank, car charger, and so on)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCG2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CCG3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CCG4</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>CCG5/CCG5C</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>CCG6/CCG6DF/CCG6SF</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>CCG3PA</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>ACG1F</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>–</td>
</tr>
<tr>
<td>PAG1S</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>CCG3PA2</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>CMG1</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>CCG3PA-N</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>CCG7D¹</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
<tr>
<td>CCG7S</td>
<td>–</td>
<td>–</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CMG2</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PAG2S</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>✓</td>
</tr>
</tbody>
</table>

The leaf nodes on the tree represent devices that can be configured. Select the device, from the left pane, and click **Create Project** to configure the device as shown in Figure 2.

Since the USB-Serial devices are used to implement the “Billboard controller”, the application type ‘Billboard controller’ is available when the USB Serial device family is selected.

Note: The CCG7S and PAG2S product families support only ModusToolbox™ software based solutions and all remaining product family support PSoC™ Creator based solutions.

¹ The CCG7D product family supports both PSoC™ Creator and ModusToolbox™ software based solutions.
2.1.2 Selecting parameters

The Configuration tab appears only after the device type for the configuration is selected. The left pane of the Configuration tab displays parameter groups available for the configuration of the selected device, in a tree structure. For devices with multiple Type-C ports, configure each port independently by entering parameter values under the nodes of each ports.

You can select a group, from the left pane, to view the parameters and the corresponding values on the right pane. To modify the parameters, you can either select the values from the drop-down list or enter the values in the textbox. The utility performs validity checks on the values entered in the textbox.

To view the description of a parameter, hover the mouse over the parameter name. A tool-tip will display the description. You can also click a parameter name and view the description in the Help tab at the bottom of the dialog.

After updating the parameters, click Save As and save the configuration in the specified folder. The modifications will be saved as an XML file.

To load the saved configuration, follow the menu path File > Open.

Note that the XML file cannot be programmed to the device. A file with .cyacd/.cyacd2 extension is saved in the same folder along with the XML file. Use this file to program the configuration to the device.

This version of the EZ-PD™ Configuration Utility also generates a C source file containing the configuration data in the form of an array. You can import this source file to a CCGx SDK firmware project workspace, so that a firmware binary with the desired default values can be compiled.
2.1.3 Configuring device

To program the configuration to the device flash, follow the menu path **Tools > Configure Device.** In the Configure Device dialog, select the target device to be programmed and initiate programming.

![Device configuration dialog](image)

**Figure 4** Device configuration dialog
Hardware setup

3  Hardware setup

3.1  Programming methods

Different USB PD applications have different requirements in terms of configuration parameters as well as the means of updating the configuration.

3.1.1  Programming over I2C

Devices such as Type-C Notebooks or Monitor adapters will have an Embedded Controller (EC) that can talk to the CCGx controller through a control interface such as I2C. In this case, the CCGx controller will be configured with a program which can receive flash read/write commands through the I2C interface.

On the USB PD Host and Client boards, EC is emulated by a USB-Serial bridge device. The USB-Serial bridge device receives a set of USB-based vendor commands, which are translated into I2C transfers performed on the CCGx controller. Figure 5 shows a block diagram of the setup where the CCGx device can be programmed using USB – I2C bridge.

![Figure 5](image)

The CY4531 EZ-PD™ CCG3 evaluation kit and CY4541 EZ-PD™ CCG4 evaluation kit are examples of a USB PD Host (Downstream Facing Port (DFP)) implementations using the CCG3 and CCG4 controllers, respectively. These boards use a USB-serial controller to allow the CCG Host device to control through the USB port.
3.1.1.1  Programming Type-C to display dongle controllers

Devices such as display dongles with no standard USB functionality will have a USB Billboard device connected to the CCGx controller. This Billboard device will be connected to the CCGx controller through I2C, and can be used to program the CCGx controller. The CCG2 USB Type-C to DisplayPort Cable Solution shown in Figure 7 uses this scheme to program the CCG2 over the I2C interface.

Figure 7 shows the programming setup for dongle programming. The CCGx host board is used to make the USB connection between a Type-A USB port on the host computer and the USB Type-C port on the display dongle to be programmed. The actual device programming is done through the CY7C65211 based Billboard controller on the display dongle.

Figure 8 shows the programming setup used for programming the Type-C to Display dongles based on CCG2 devices.
3.1.2 Programming over CC channel

Devices such as Type-C Cable (EMCA) controllers and USB Type-C power adapters do not have an EC which can be used to read/update the configuration, and the USB PD CC channel is the only data interface for the CCGx controller. In such cases, the configuration read, configuration update, and firmware update are performed using Vendor Defined Messages (VDMs) that are sent through the CC channel in the USB Type-C interface. The target CCGx controller is configured to receive flash read/write commands in the form of USB PD unstructured VDM messages.

As in the earlier case, the USB-Serial Bridge is used to receive USB vendor commands and translate them into a set of I2C based commands. These I2C commands are addressed to a CCGx based flashing controller, which translates them into the required USB PD messages.

Standard DFP implementations can only initiate USB PD commands when a USB PD power contract is in place. Since the DFP cannot setup a power contract while a power consumer (Upstream Facing Port (UFP)) is not present; programming the CCGx based EMCA cable requires both a host (DFP) and client (UFP) to be connected.
Hardware setup

The CY4532 EZ-PD™ CCG3PA EVK consists of a Power board and a Main board. The Main board is pre-mounted on the Power board during manufacturing. The CCG3PA device is located on the Main board.

Power adapter designs are programmed by connecting them to the CCGx host board using a Type-C cable or using the Type-C plug on the power adapter itself.

Note: If two Dual Role Power (DRP) controllers are connected by a Type-C cable, and both DRP controllers are connected to PC through serial bridge, then device tree will have duplicate devices and will lead to repeated role swaps, slowing the device configuration. So, it is recommended to connect only one of the DRP controllers to the PC running the configuration utility in such situations.

![Programming over CC using CY4532 boards](image)

Figure 10 Programming over CC using CY4532 boards

Note: The CC firmware update feature is not intended for field firmware update. See KBA230192.
3.2 Programming setup

3.2.1 Programming EMCA (cables)

The setup shown in Figure 10 is used for EMCA programming. The USB Type-C cable shown in Figure 10 can be replaced by the target EMCA that needs to be programmed.

The CY4532 Power board should be programmed with a Notebook DRP firmware image. See CY4532 kit user guide for details on how to configure these boards and program them using the EZ-PD™ Configuration Utility.

Once the Host and Client boards have been configured and programmed, both can be connected to USB hosts to power them up. The host board needs to be connected to the computer running the EZ-PD™ Configuration Utility, while the client board only needs to be powered through some source.

Later sections in this document provide instructions on how to read configuration and program EMCA devices.

3.2.2 Programming Type-C dongles and Billboard devices

Programming a Type-C to Display (DisplayPort or HDMI) dongle requires access to the USB and USB PD CC channels on the CCGx controller. The CC channel will be connected to the CCGx based host board as in the earlier case. The USB data pins are connected from the Billboard Controller to the PC host (see Figure 11).

The CCGx host is only used to power the CCGx target device through the VConn supply. The actual programming is done through USB Vendor Commands addressed to the Billboard Device.

In the case of the CY4541 CCGx Host board, the USB data lines are also connected through the host board itself to the Type-C connector.

In case of CCG3 dongle devices, the internal USB-FS block present on CCG3 chip is used for providing Billboard interface. As there is no external Billboard device, USB-C pins are directly connected to CCG3 dongle as shown in Figure 12.
3.2.3 Programming Notebook Type-C port controllers

The CCGx Notebook port controller can be programmed directly by connecting it to the host computer using the Mini-USB port on the host board.

See the user guides for the respective kits for more detailed programming instructions and connections.

3.2.4 Programming Type-C power adapters

Type-C Power Adapter solutions built using the CCGx controllers need to be programmed through the CC channel, as they do not provide an I2C interface for device programming.

The setup for programming power adapter designs is similar to that used for EMCA programming. The CCGx client board will be replaced by the power adapter to be programmed as shown in Figure 13.

Figure 12 Programming setup for Type-C dongle with Internal On-chip Billboard interface

Figure 13 Setup for programming Type-C power adapters
3.2.5 Programming manufacturing test kit hardware

The Manufacturing Test Kit (MTK) hardware consists of the CY4532 EZ-PD™ CCG3PA power board (ensure that the CY4532 EZ-PD™ CCG3PA EVK Main board is detached from the Power board). The MTK hardware will be programmed over I2C. The setup used for programming the MTK hardware is shown in Figure 14.

![Figure 14: Programming setup for MTK hardware](image-url)
GUI usage

4 GUI usage

To create a new device configuration, follow the menu path **File > New**. Then, select the CCG product family (CCG2/CCG3/CCG4/CCG5/CCG5C/CCG6/CCG3PA/CCG1/PAG1S/PAG1P/AGC1F/CCG6DF/CCG6SF/CCG3PA_N/CCG7D/CCG7S/PAG2S or USB Serial) from the **Device type** drop-down list. The applications supported by the selected product family is displayed. Select the target application from the list.

A description of the selected application is displayed on the right pane. Click **Create Project** to start configuring the device.

![New configuration dialog box](image.png)

The configuration is saved as an XML file. To open a saved configuration, follow the menu path **File > Open**. You can modify, save, and program this configuration to the Type-C or Billboard controller.

To read the current configuration from the target device, follow the menu path **File > Read from Device**. All devices currently connected to the host PC are displayed in a tree view. You can select a desired target to read from. You can modify, save, and re-program the loaded configuration to the controller.
4.1 USB PD device configuration

Device configuration parameters are classified into device parameters and port parameters. For multi-port devices there will be multiple copies of port parameter entry options. Port parameters are further classified and grouped into a hierarchical Tree structure. The following sets of parameters will be displayed:

- Device parameters
- Port 0
  - Discover identity
  - SVID configuration
  - PDO configuration
  - Port information
  - Billboard parameters
  - DP mode parameters
  - SCEDB configuration
  - SKEDB configuration
  - Application configuration
  - Power protection parameters
  - Power settings
  - Battery configuration
  - Charging configuration
  - BCR configuration
  - Type-A configuration
  - Auto configuration
  - Thunderbolt host configuration
  - Base alternate modes configuration
  - Custom alternate mode configuration
  - Custom host configuration
  - ICL/TGL configuration
- User parameters

Note: For ModusToolbox™ software based solutions, a few of the configuration nodes may be modified. Refer to the following sub-sections for more details.

Some of these sections only apply to specific device types and are visible only when working with such devices. Each of the above parameter sets are described in subsequent sections.

For description of the Notebook application parameters, see Section 3.3 of the EZ-PD™ CCGx Host SDK user guide from EZ-PD™ Host SDK and for Power based applications (power bank, power adapter) parameters, see Section 3.2 EZ-PD™ CCGx Power SDK user guide from EZ-PD™ CCGx Power SDK.
GUI usage

4.1.1 Device parameters

When creating a new configuration, the first step is to select the EZ-PD™ part that will be used for the design. In addition to the part selection, the Device Parameters node also contains EZ-PD™ controller specific parameters that determine how the Type-C controller works.

![CCGx device parameters configuration](image)

**Note:** The Flashing VID and Flashing mode index parameters are only required by EMCA and Power Adapter solutions where the device programming is done through the CC interface.

*For ModusToolbox™ software based solutions, Manufacturer Info is available under Port Information configurations.*

4.1.2 Discover identity

The Discover Identity nodes contain the parameters that are reported by the EZ-PD™ controller in the form of the USB PD Discover Identity response. The parameters include the responses included in the Id Header, Cert Stat, Product, and any Product Specific VDOs.

Some of these fields like Product Type are fixed for a specific application. These values are grayed out and cannot be modified.

![Device ID configuration](image)
GUI usage

The Device IDs are VDOs that need to be configured for all USB PD devices. These contain fields that describe the device’s identity.

The product-specific VDOs vary with the type of USB PD application. In case of EMCA applications, the Cable VDO needs to be configured.

![Figure 18 Cable VDO configuration](image1)

The AMA VDO needs to be configured for applications that support one or more Alternate Modes.

![Figure 19 AMA VDO configuration](image2)

The UFP VDOs need to be configured for ports capable of operating as a UFP; including traditional USB peripherals, USB hub’s upstream Port, and DRD capable host Ports.
The DFP VDO needs to be configured for ports capable of operating as a DFP; including those implemented by Hosts, Hubs, and Power Bricks.
GUI usage

Figure 22 DFP VDO configuration

To view the description of a parameter, hover the mouse over the parameter name. A tool-tip will display the description. You can also click a parameter name and view the description in the Help tab at the bottom of the dialog.

Figure 23 Tool-tip Display for parameters

4.1.3 SVID configuration

The SVID Configuration node is used to specify the various USB PD alternate modes supported by the Type-C controller.

For EMCA and Power Adapter devices, this node is pre-populated with SVID (0x04B4) and flashing mode (0x01). This alternate mode is used by the EZ-PD™ controllers to implement configuration update and firmware upgrades through CC VDMs. The value of the Flashing VID and Mode can be changed in the Device Parameters section. The values displayed under SVID0 will automatically be changed when flashing VID and Mode Index are changed.

Use Add and Remove on the top of the left pane to add new SVIDs and to remove existing SVIDs from the configuration. Add is active only when you select a SVID Configuration node. Remove is active only when you select one of the SVIDs. See Figure 24.

Attention: Devices that are programmed over the CC channel (EMCA and Power Adapters) need to support a special SVID and mode, which are used for sending the Unstructured VDMs used for the programming operation. Deleting the flashing SVID (SVID0) in these cases will cause the CCGx device to stop supporting the configuration and firmware update feature. Delete this SVID only if you want to disable the update capability in the design.
GUI usage

Figure 24  Adding or removing SVIDs

After selecting an SVID note, you can edit the corresponding SVID value from the right pane. You can use Add and Remove to add and delete modes from the active SVID. Figure 25 shows the SVID and mode values to be edited, and the options to add or remove modes to a selected SVID.

Figure 25  Editing modes for an SVID

4.1.4  PDO configuration

Power Data Objects (PDOs) are data sets that describe the power capabilities of a USB PD device. PDOs can be Source PDOs or Sink PDOs, depending on whether they describe the power sourcing or sinking capabilities of the USB PD device.

The PDO configuration section is used to define one or more PDOs describing the capabilities of the current device. This section is not applicable to EMCA devices, as EMCA does not require any PDOs.

Type-C Dongles are not expected to have any external power source and hence will support only Sink PDOs. Type-C Power Adapters will only function as power sources and will support only Source PDOs. Notebook Port Controllers will support both Source and Sink PDOs.

Figure 26 and Figure 27 show the configuration screens used to set the properties of PDOs. All fields listed under Source and Sink PDOs are defined in the USB PD specification. For more details, see the specification.
GUI usage

First, select the supply type, and then configure the remaining parameters.

An **Enable** checkbox is associated with each PDO, which specifies whether the PDO is enabled at system start-up. Solutions that support the Host Processor Interface (HPI) allow to dynamically enable/disable of PDOs at runtime. In other solutions, all available PDOs should be enabled.

**Figure 26** Configuring sink PDOs

Sink PDO configuration includes two additional parameters:

- The **Sink gives back** parameter specifies whether the feature is enabled for this PDO.
- The **Min/Max operating current** parameter specifies the minimum current required for device operation if give back is enabled. If give back is disabled, this field specifies the maximum operating current that may be required.

**Figure 27** Configuring source PDOs
4.1.5 Port information

The PD Port Information node contains a set of properties that control how the CCGx controller manages the USB PD Port. Most of these properties are mapped to the properties defined by the USB PD Specification, and the rest of the device-specific parameters. View the tooltips for the description of each parameter.

![Configuring PD Port Information](image)

Figure 28 Configuring PD Port information
Note: For ModusToolbox™ software based solutions, the configuration parameters: Sink USB suspend, Sink USB communication, DR_Swap response, PR_Swap response, and VCONN_Swap response are available under the Application configuration node.

4.1.6 Billboard parameters

USB PD devices such as display dongles and monitors that do not have a standard USB function need to present a USB Billboard interface to report the device capabilities and status. The Billboard interface can make use of the on-chip (internal) USB-FS block for CCG3 designs or an external USB Billboard controller for other designs.
GUI usage

The Billboard parameters section is only applicable to USB PD devices that have either an internal USB block or an external Billboard controller associated with them.

4.1.6.1 Designs with External Billboard controller

CCG2 devices do not have an on-chip USB block, hence billboard functionality are provided through an external Billboard controller.

These Billboard properties for such designs control the behavior of the Billboard device and specify whether it is possible to perform configuration updates through the Billboard device. Parameters such as the USB IDs used by the Billboard Controller, the string descriptors, the supported alternate mode descriptions, and so on will be configured separately on the Billboard controller itself (see Billboard device configuration).

![CCG2 Billboard control parameter configuration](image)

- The **Billboard enable** field specifies whether the Billboard Device USB enumeration is always enabled or enabled only in case of any error on the CCGx device side.
- The **Billboard programmer support** field specifies whether the vendor mode used for CCGx flash read and update should be enabled on the Billboard device.
- The **Billboard time-out in seconds** field specifies the duration for which the Billboard device will remain enabled before being put into sleep to save power. This parameter only applies when the CCGx controller is working without any errors. If there is an error on the CCGx side, the Billboard device remains enabled continuously.

4.1.6.2 Designs with On-Chip Billboard interface

CCG3 devices have an internal USB-FS block that can be used for Billboard implementation. Since the Billboard device is implemented on CCG3 itself, all Billboard configuration is embedded in the CCG3 device’s configuration table.

The following parameters need to be configured for CCG3 designs with Billboard.
GUI usage

CCGx settings

This node specifies the parameters that control the behavior of the Billboard interface. These parameters include:

- **Billboard type**: Specifies the Billboard type – Internal, External, or No Billboard
- **Billboard enable**: Indicates whether the Billboard functionality is always enabled or enabled only when an error occurs on CCGX device
- **Billboard programmer support**: Indicates whether flash update through Billboard device is enabled
- **Billboard timeout in seconds**: Indicates delay in seconds before the Billboard USB device is turned OFF to save power
- **Billboard power setting**: Indicates whether the Billboard is Bus powered or Self powered
- **Billboard VConn power setting**: Specifies the power that the design will draw from the VConn supply
- **Billboard container ID settings**
- **Billboard serial number settings**

![Figure 31: CCG3 Billboard CCGx parameters configurations](image)

Billboard settings

This node specifies parameters associated with Billboard USB device. These parameters include:

- VID and PID used to identify the device.
- Manufacturer, Product, Serial Number, Configuration, and Interface strings used to identify the device.
- URL that can provide more information about the device.
GUI usage

Alternate modes

Alternate modes configuration parameters specify the alternate modes supported by USB PD controller. These parameters include Standard or Vendor ID (SVID), mode index, and description corresponding to each supported alternate mode.

All alternate modes defined under the SVID section are listed under this node and you can edit its description in this section. To add or remove an alternate mode, you need to add or remove modes under the SVID section.

4.1.7 SCEDB configuration

Source Capabilities Extended Data Block (SCEDB) contains additional information about a Port’s Source Capabilities. Figure 34 shows the SCEDB configurations.
4.1.7.1 Peak current configuration

The Peak Current field contains the combination of Peak Currents that the Source supports. Peak Current provides a means for the Source to report its ability to provide current in excess of the negotiated amount for short periods. The Peak Current descriptor defines up to three combinations of percent overload, duration, and duty cycle defined as PeakCurrent1, PeakCurrent2, and PeakCurrent3 that the Source supports. In some cases, a Source might not offer Peak Current capability.
GUI usage

4.1.8 SKEDB configuration

Sink Capabilities Extended Data Block (SKEDB) contains additional information about a Port’s Sink Capabilities. Figure 36 shows the SKEDB configurations.

![Figure 36 SKEDB configuration](image.jpg)

4.1.9 Application configuration

This node is applicable only for ModusToolbox™ software based solutions, and contains the following application configuration parameters.
GUI usage

4.1.10 Protection parameters

USB PD designs such as Notebook port controllers, power adapters, and dock/monitors need to implement protection circuitry that ensures that the device is not harmed due to over voltage, over current, over temperature, under voltage, or short circuits.

CCGx based designs can implement VBUS Over Voltage (OVP), VBUS Over Current (OCP), VBUS Under Voltage (UVP), VCONN Over Current (VCONN-OCP), Over Temperature (OTP), and Short Circuit (SCP) protection schemes.

The EZ-PD™ Configuration Utility allows to enable/disable these protection schemes, and configure the parameters associated with the schemes to match the target system behavior. The Protection parameters node in the configuration includes sub nodes for all protection mechanisms implemented by the corresponding firmware solution. Each protection scheme node contains the corresponding configuration parameters.

Note that these protection schemes are not implemented in all firmware solutions.

4.1.10.1 VBUS over voltage protection

This node contains the parameters that configure VBUS OVP.
GUI usage

4.1.10.2 VBUS over current protection

This node contains the parameters that configure VBUS OCP. The configuration parameters Off time and Sample period are displayed only when value of the Config table major version parameter is set to 1 in the Device Parameter configuration node.

Figure 39 VBUS OCP parameter configuration

4.1.10.3 VBUS under voltage protection

This node contains the parameters that configure VBUS UVP.

Figure 40 VBUS UVP parameter configuration
GUI usage

4.1.10.4 Short circuit protection

This node contains the parameters that configure SCP.

![Short circuit protection parameter configuration](image)

*Figure 41  Short circuit protection parameter configuration*

*Note:* Enabling SCP may cause false triggering of SCP on attach/detach when using Type-C to micro-B adapter.

4.1.10.5 Over temperature protection

This node contains the parameters that configure OTP.

![OTP parameter configuration](image)

*Figure 42  OTP parameter configuration*
GUI usage

4.1.10.6 VCONN over current protection

This node contains the parameters that configure the VCONN over current protection.

![Figure 43 VCONN over current protection parameter configuration](image)

4.1.11 Power settings

Power settings node provides VBUS control configurations.

![Figure 44 Power settings parameter configuration](image)
GUI usage

4.1.12 Battery configuration

Battery configuration provides the following battery and charging parameter configurations:

- **Battery maximum voltage (mV):** Maximum battery voltage beyond which it is unsafe to charge.
- **Battery cutoff voltage (mV):** Battery voltage at which the device is expected to stop discharging and be on low voltage consumption.
- **Battery discharge enable voltage (mV):** Battery voltage at which discharge can be re-enabled.
- **Maximum battery charging current (mA):** Maximum charging current allowed for the battery.

![Battery configuration](image)

Figure 45 Battery configuration

4.1.13 Charging configuration

Charging configuration contains configurations of legacy charging support as source and sink.

4.1.13.1 Source setting

Figure 46 and Figure 47 show the legacy charging source setting configuration parameters for BC 1.2 and Apple charging modes, respectively.
GUI usage

Figure 46  Legacy charging source setting parameter configuration for BC 1.2 charging mode

Figure 47  Legacy charging source setting parameter configuration for Apple charging mode

AFC source caps

This node contains a list of Adaptive Fast Charging Source capability objects. This node is visible only when AFC charge has been enabled in the source setting configuration node. You can add or remove AFC Source capability objects using the Add and Remove buttons provided on the upper right corner.
GUI usage

Figure 48  AFC source capability objects

Sink settings
This node is valid only for CCG3PA Power Bank application, and contains legacy charging sink setting configuration parameters.

Figure 49  Legacy charging sink setting parameter configuration

4.1.14  BCR configuration
This node is applicable only for CCG3PA BCR Power Sink application, and contains BCR configuration parameters.
Figure 50 BCR configuration parameters

4.1.15 Type-A configuration

In addition to the Type-C port, the device can optionally support a Type-A port. This node provides the configuration information for the Type-A port. This is applicable only for CCG3PA applications.

Figure 51 Enable Type-A configuration

4.1.15.1 Power setting

This node provides VBUS control additional configurations for Type-A port. These configuration parameters are the same as the Type-C power setting parameters.

4.1.15.2 Charging configuration

This node contains configurations of legacy charging support as Source for Type-A port. These configuration parameters are the same as the Type-C power charging configuration Source setting parameters.
GUI usage

4.1.16 Auto configuration
This node provides configurations for Automotive Power applications.

Note: For ModusToolbox™ software based solutions, auto configuration node is not part of port configurations and is displayed at the same level as device parameters.

4.1.16.1 Policy information
This node contains policy manager information.

Figure 52 Policy information

4.1.16.2 Input voltage throttling information
This node contains input voltage throttling information.

Figure 53 Input voltage throttling information

4.1.16.3 Temperature throttling information
This node contains temperature throttling information. It contains sub-nodes for sensor information.
GUI usage

4.1.16.4 VIN fault voltage

This node contains VIN fault voltage information.

4.1.17 Thunderbolt host configurations

This node contains Thunderbolt host configuration parameters. The Thunderbolt host configurations are only applicable for CCG5, CCG5C, CCG6, CCG6DF, and CCG6SF device applications.
GUI usage

4.1.18 Base alternate modes configuration

This node contains information about alternate modes and simultaneous entry with other alternate modes. These alternate mode definitions are added/deleted using the Add a node and Remove buttons on the left pane. Once an alternate mode has been added to the project, the following fields need to be configured:

- SVID# 0: SVID corresponding to this alternate mode. This is mandatory field.
- Optionally, add SVIDs which could be processed/entered simultaneously with the main SVID (SVID# 0) using Add button on the right panel.
- Check Supported in DFP checkbox if the alternate mode is supported in DFP data role.
- Check Supported in UFP checkbox if the alternate mode is supported in UFP data role.
GUI usage

4.1.19 Custom alternate mode configuration

This node contains custom alternate mode configuration parameters. Custom alternate mode could be run simultaneously with any alternate mode from the base alternate mode configurations.

![Custom alternate mode configuration](image)

Figure 58 Custom alternate mode configuration

4.1.20 Custom host configuration

This node contains custom host configuration parameters.

![Custom host configuration](image)

Figure 59 Custom host configuration

4.1.21 ICL/TGL configuration

This node contains Ice Lake/Tiger Lake (ICL/TGL) configuration settings. ICL/TGL Platform selection configuration parameter value should be the same for both ports, however it can be edited only in Port 0.

![ICL/TGL configuration](image)
GUI usage

**Figure 60**  
ICL/TGL configuration

### 4.1.22 AMD configuration

This node contains AMD configuration parameters.

**Figure 61**  
AMD configuration

### 4.1.23 User parameters

The CCGx device configuration table also provides a 32-byte region where user-defined parameters can be saved. This data is not used by the CCGx firmware provided by  

or the configuration utility itself. These values can be programmed through the configuration utility, and can be read out by a controller talking to the CCGx device.

The 32-byte wide user parameters can be input using either of the following:
GUI usage

- A set of eight 4-byte hexadecimal values. This method allows any parameter value to be entered.
- An ASCII string of 32-bytes or lesser. Only printable characters can be input in this case.

![User parameters in CCGx configuration – hexadecimal input](image1)

**Figure 62** User parameters in CCGx configuration – hexadecimal input

![User parameters in CCGx configuration - ASCII input](image2)

**Figure 63** User parameters in CCGx configuration - ASCII input

*Note:* Any non-printable characters in the hexadecimal input will be lost when switching to the ASCII string view. Use only one form of input for each CCGx configuration project.

### 4.2 Billboard device configuration

The Billboard Controller is a standard USB device and not a USB PD controller. Therefore, the configuration parameters applicable to Billboard Devices are different from those for the USB PD Devices. The following need to be configured for Billboard controllers:

- Device parameters
- Flashing parameters
- Billboard parameters
- Alternate modes

#### 4.2.1 Device parameters

The device parameters section for Billboard Devices is similar to that for USB PD devices, and is used to select the part number used for Billboard implementation.
GUI usage

4.2.2 Flashing parameters

The Billboard device implementation can have following modes of operation:

- Billboard mode: In this mode, the Billboard device provides the Binary Object Store (BOS) descriptors that allows the USB host to identify the capabilities of the USB PD controller associated with the Billboard device. In addition to the Billboard interface, this mode also supports a Human Interface Device (HID) function, which can be used to control the device.

- Flashing mode: The Billboard and HID interfaces do not allow you to update the configuration or firmware on the USB PD controller or the Billboard interface itself. Such functionality is provided through a special flashing or programming mode. The switch from the billboard to the flashing mode is achieved by sending a specific HID report command.

The Flashing Parameters Node is used to specify the properties of the flashing interface of the Billboard device.

![Figure 64 Flashing parameter configuration](image)

The **Flashing interface VID** and **Flashing interface PID** values cannot be changed as these are used to bind the Billboard device to the Cypress USB driver used for programming. The **Flashing interface manufacturer** and **Flashing interface product** strings can be changed as required.

4.2.3 Billboard parameters

The Billboard Parameters node specifies parameters associated with the Billboard mode operation. These parameters include:

- **VID** and **PID** used to identify the device
- **Manufacturer**, **Product**, **Serial number**, **Configuration**, and **Billboard interface** strings used to identify the device.
- **Container ID UUID** value to be reported in the BOS descriptor.
- **Billboard VConn power** setting specifies amount of power required by the USB PD device from the VConn supply.
- **CCG I2C slave address** is used to configure the I2C interface between the Billboard controller and the USB PD controller.
- **Additional info URL** provides more information about the device.
- **HOLD EC GPIO** and **HOLD BB GPIO** are used to avoid simultaneous access of CCGx HPI by billboard controller and EC.
- **SCB1 configuration** is used to configure SCB1 block.
GUI usage

4.2.4 Alternate modes

The main functionality of the Billboard device is to report the various alternate modes supported by the corresponding USB PD controller.

The number of alternate modes that can be reported through the Infineon Billboard implementation varies from 1 to 8. You can add and delete these alternate mode definitions using the Add and Delete buttons on the left pane. Once an alternate mode has been added to the project, the following fields need to be configured:

- **SVID**: The Standard or Vendor ID (SVID) corresponding to this mode.
- **Mode index**: The index corresponding to this mode.
- **Description**: A descriptive string associated with this mode.
- **Mode value**: Vendor defined mode value.
- **Preferred mode**: Preferred alternate mode. Select the mode from drop-down list as shown in Figure 67 (only one can be preferred mode).
4.2.5 GPIO configuration

Unused GPIOs can be configured as input or output GPIOs. For output GPIOs, the initial state and drive mode can be configured. The initial state of an output GPIO can be set as None, Low, or High using the GPIO-x initial state field. The drive mode of output GPIOs can be set as one of the following values.

- Mode 0 (Analog mode, High Z)
- Mode 1 (High Z)
- Mode 2 (Strong pull down, Resistive pull up)
- Mode 3 (Resistive pull down, Strong pull up)
- Mode 4 (Strong pull down, Open drain)
- Mode 5 (Open drain, Strong pull up)
- Mode 6 (Strong pull down, Strong pull up)
- Mode 7 (Resistive pull down, Resistive pull up)
4.3 Saving the configuration

The above options complete the configuration required for the USB PD or Billboard solutions. Once the configuration is complete, follow the menu path File > Save As.

Every configuration will be saved as an XML file. You can open and edit the saved file, as required.

The configuration is also saved in the form of a Infineon defined binary file formats called the cyacd/cyacd2. The cyacd2 file will be created for ModusToolbox™ software based solutions. The cyacd/cyacd2 file will be created with the same filename under the same path as the XML file.

This version of the EZ-PD™ Configuration Utility also generates a C source file containing the configuration data in the form of an array. This source file can be imported into a CCGx SDK firmware project workspace, so that a firmware binary with the desired default values can be compiled.

4.4 Reading configuration from firmware file

To read the configuration data from the firmware file stored in .hex, .cyacd or .cyacd2 format, follow the menu path File > Read from Firmware File.

In CCG2 applications, the .cyacd format firmware file does not contain configuration data, hence the utility does not support reading the configuration from the .cyacd firmware file of CCG2 applications.

In dual firmware applications, the .hex format firmware file contains two firmware images and corresponding configurations. EZ-PD™ Configuration Utility provides an option for selecting the firmware image Id from which configuration has to be read. See Figure 69.
4.5 Saving configuration to firmware file

To save the configuration created or opened in EZ-PD™ Configuration Utility to a firmware file, follow the menu path **File > Save to Firmware File**. The configuration can be saved to .cyacd, .cyacd2 or .hex firmware files.

In CCG2 applications, the .cyacd format firmware file does not contain configuration data, hence .cyacd firmware file format is not supported.

In dual firmware applications, the .hex format firmware file contains two firmware images and corresponding configurations. While saving configurations to dual firmware .hex format, EZ-PD™ Configuration Utility update the configuration of both firmware images.
4.6 Configuring the device

After entering and verifying the relevant values, follow the menu path **Tools > Configure Device** to program the configuration to the device.

On the left pane of the Configure Device dialog, the devices are displayed in tree structure, representing the way they are connected. Select a device to view its properties on the right pane,

Click **Program** to initiate programming of the device.

*Note:* The status bar at the bottom of the dialog displays the state of the device interface. The program option is only accepted when the device interface is in the ready state.

Click **Refresh** to start a fresh scan of EZ-PD™ devices connected to the PC host.
4.6.1 Extensions for dual firmware devices

The flash architecture is different for different CCG device families. CCG2, CCG3PA, PAG1S, PAG2S, CCG7D and CCG7S devices make use of a single firmware binary with flash access (firmware or configuration update) supported through a dedicated boot-loader application. The USB PD port functionality is not supported in boot-loader mode; which means that these operations will disrupt the USB PD port operation.

CCG3/4/5/6 devices support the dual-firmware solutions. In dual firmware devices, the internal device flash contains two copies of the firmware which can mutually update each other. The USB PD port operation can continue unaffected while the flash update is in progress. You can reset the dual firmware device at an appropriate time to switch to the newly updated firmware or configuration.

4.6.1.1 Single firmware mode (legacy boot mode)

Since CCG2, CCG3PA, CCG7D, CCG7S, PAG1S and PAG2S devices only support device programming through the boot-loader, only one programming option is available for these devices.

When programming starts, the utility will request the device to jump to bootloader mode where all PD capabilities are disabled. Once the configuration update is complete, the device will be reset into the firmware application mode.
GUI usage

4.6.1.2 Dual firmware mode

If the device has dual firmware, you can configure the device in either of the following methods:

- Using firmware to flash

  The default behavior of the utility is to configure the alternate firmware without jumping to bootloader. This option will be available only if both firmware copies in the flash are valid (running Firmware is FW1/FW2 and alternate firmware is FW2/FW1). The new configuration will be applied to the alternate firmware image, and the active firmware binary is switched at the end of the configuration update. This will cause the current alternate firmware to become the new running firmware.

  When you click **Program**, the running firmware will update the configuration table of the alternate firmware without interrupting the activities of the CCGx controller. The device will be left in the current state (so as to not disturb the PD port operation) and will start running the alternate firmware when it next goes through a reset.

  If the alternate firmware version is older than the running firmware, the utility will pop-up a warning indicating that the firmware will roll back to an older version at the end of the operation. You have an option to continue or abort the operation at this stage.

![Configure device dialog box for dual firmware device](image)

**Figure 72** Configure device dialog box for dual firmware device
GUI usage

- Using bootloader to flash
  This option is available only when at least one valid FW is present (Running firmware is FW1/FW2). This option is used when you want to update the configuration of the active firmware binary without wanting to switch to the other firmware copy.
  The flash update is done by stopping the PD port activity, switching to the boot-loader mode, and then updating the configuration table. The active firmware version will remain the same at the end of the flash update and device reset.

Change App Priority

If the device supports dual firmware functionality, running firmware can be switched to alternate firmware by clicking Change App Priority. This operation will be successfully performed only when alternate firmware is valid.

4.6.2 Extensions for hybrid architecture applications

The Hybrid flash architecture scheme is currently supported on CCG6DF and CCG6SF devices. There are two major parts to this solution:

1. Primary: This is the main firmware application which supports the necessary Type-C, USB PD, Alternate Mode, and HPI features. The primary does not have any firmware update capabilities and transfers control to the secondary base when any update is required.

2. Secondary
   a) Secondary Base: This is a bootloader application. The secondary base is kept write protected in the CCG device flash and cannot be modified. Since all firmware updates are performed through the secondary base, it also supports a minimal set of Type-C and USB PD features so that the notebook can be charged while the firmware update is being performed.
   b) Secondary Extension: This is an optional binary which augments the Secondary Base application. Since the Secondary Base is write protected, it cannot be updated to add any new functionality that may be desired while the firmware updates are being performed. In such cases, a Secondary Extension application which acts as a set of wrappers around the Secondary Base can be brought into the device flash and executed.

In hybrid architecture solutions, all flash read/write operations are performed through secondary base firmware. The utility provides the option to select the firmware, Secondary Extension or Primary, for which the configuration should be updated.
GUI usage

Figure 73  Configure device dialog box for hybrid architecture application

Note:

1. The tree-view lists all devices that are detected using the USB and CC connections. Since a USB PD Client is connected through USB to one USB-Serial Device and through CC to the USB PD Host, the same device could be listed twice.

2. Properties of the EMCA and UFP controllers can only be retrieved if the target device supports the CC based flash read/write operations. If the device does not support these operations, the corresponding properties will not be displayed and will be BLANK.

The configuration is expected to take about 30 seconds and the messages indicating the progress are displayed in the message box. A progress bar also indicates the progress of the operation.
GUI usage

Figure 74  Configure device completion

4.7  Reading configuration from device

To read the Type-C configuration stored on the EZ-PD™ controller connected to the computer and display it, follow the menu path File > Read from Device.

The Read dialog is similar to the Configure Device dialog and displays the connected devices in a tree structure. You can select a device to read data from.

The configuration read from the device will be displayed using the same screens used for completing the configuration. You can edit, save, program the parameter values back to the controller.
GUI usage

4.7.1 Extensions for dual firmware devices

As in the case of device configuration, the dual bank firmware implementation has implications on the read functionality as well.

Figure 75  Read device dialog

Figure 76  Read device dialog box for dual firmware devices
GUI usage

The default behavior of the utility is to read and return the configuration associated with the alternate firmware. You can use the **Use bootloader to read** option, shown in Figure 76, to trigger a read of the configuration table associated with the running firmware. However, this operation will require the CCG device to disable the USB PD connections and revert to bootloader mode for doing the flash reads.

### 4.7.2 Extensions for hybrid architecture applications

In case of hybrid architecture applications, configuration read operation is performed through secondary base firmware. The utility provides the option to select the firmware, **Secondary Extension** or **Primary**, from which the configuration data should be read.

![Figure 77](image)

**Figure 77** Read device dialog box for hybrid architecture applications

### 4.8 Firmware update

To update the firmware running on the EZ-PD™ or Billboard controller follow the menu path **Tools > Firmware Update**.

As in the case of device configuration and configuration load, the firmware update can be performed on any of the supported devices. Select the controller using the Target Device option and select the firmware binary file.

Firmware for the CCGx controllers are provided in `.cyacd/.cyacd2` format, and firmware for the Billboard controllers are provided in `.img` format. Sample firmware binaries for each application and standard part numbers are provided under the **Firmware** folder in the utility installation.

The firmware update over CC will take about two minutes to complete. Progress messages will be displayed in the Messages window and a progress bar also shows the progress of the firmware update process.
GUI usage

Attention: Do not disconnect the devices from the computer while the firmware update is in progress.

![Firmware update dialog](image)

Figure 78  Firmware update dialog

4.8.1  Extensions for dual firmware devices

You can specify one or both firmware binaries through the firmware update dialog, and the appropriate binary will be used for the update operation.

![Firmware update dialog for dual firmware devices](image)

Figure 79  Firmware update dialog for dual firmware devices
GUI usage

If the device is currently running FW1, the utility will select the FW2 binary from the two files passed in, and update FW2 with the new version. If you do not provide the FW2 binary, an error will be reported.

It is possible to update both FW1 and FW2 binaries at once using **Use bootloader for flashing** as shown in **Figure 79**. In this case, CCG will disable all PD ports and revert to bootloader mode so that both firmware banks can be updated.

### 4.8.2 Extensions for hybrid architecture applications

In case of hybrid architecture applications, firmware update operation is performed through secondary base firmware. To update firmware, browse and select the primary, secondary extension, or both firmware files. Then, click **Program**. The utility will update all selected firmware images.

![Figure 80] Firmware update dialog for hybrid architecture application

#### 4.9 Manufacturing test kit option

The EZ-PD™ Configuration Utility supports MTK Option, a special mode to configure and test Infineon EMCA devices (CMG1, CMG2 and CCG2).

The MTK consists of the MTK UI, MTK firmware, and MTK hardware.

The MTK UI will be invoked from the EZ-PD™ Configuration Utility. The MTK Firmware will be available as part of the EZ-PD™ Configuration Utility installation package.

The MTK UI supports configuring and testing of up to 10 Infineon EMCA devices in parallel. This option is designed to be used in factories during the manufacturing cycle. **Figure 81** shows the setup for configuring/updating and testing 10 cables.
GUI usage

The MTK hardware consists of the CY4532 EZ-PD™ CCG3PA EVK Power board (ensure the CY4532 EZ-PD™ CCG3PA EVK Main board is detached from the power board). For more details, see the CY4532 product page.

The following operations are required to configure and test the Infineon EMCA devices with the MTK:

- Download MTK firmware
- Configure MTK Tester ID
- Create EMCA configuration
- MTK operation

Figure 81  Infineon EMCA device programming setup
GUI usage

4.9.1 Download MTK firmware

See Programming manufacturing test kit hardware for the hardware setup to download the firmware. The MTK firmware is available in the Firmware\MTK folder in the EZ-PD™ Configuration Utility installation.

Download both the firmware images CYPD4126-40LQXI_mtk_1.cyacd and CYPD4126-40LQXI_mtk_2.cyacd via the EZ-PD™ Configuration Utility.

To update the firmware:

1. Select Firmware Update and load appropriate firmware path for the Firmware path 1 and Firmware path 2.
2. Select Use bootloader to flash, as shown in Figure 82, and program the CY4532 power board.

Figure 82 MTK firmware update

Once the firmware images are updated successfully, the Application Type in Firmware update window will change to MTK as shown in Figure 83.
4.9.2 Configure MTK tester ID

To uniquely identify the MTK hardware, you can provide a tester ID by configuring the user parameter of the MTK hardware configuration. To configure the tester ID, first read the configuration information from the MTK hardware. Follow the menu path **File > Read from Device** (see **Figure 75**). From the User Parameters section, update **Parameter 1** with the 2-byte tester ID as shown in **Figure 84**.
Save the configuration and program the saved configuration as shown in the Figure 85. If the programming is successful, the programmed tester ID will be displayed in the Tester ID column of the MTK display window (see Figure 88).
GUI usage

4.9.3 Create EMCA configuration

Create a new configuration project with CMG1, CMG2 or CCG2 as the Device Type (see Creating configuration for more details).

You can select a group to view the parameters and the corresponding values on the right pane. To modify the parameters, you can either select the values from the drop-down list or enter the values in the textbox. The utility performs validity checks on the values entered in the textbox.

The CMG1 Configuration options are shown in Figure 86.

![CMG1 configuration parameters](image)

**Figure 86** CMG1 configuration parameters

*Note:* Lock configuration option is available for CMG1 and CMG2 devices. Selecting Yes for the Lock configuration option will treat the CMG1 or CMG2 device as a one-time configurable part.

4.9.4 MTK operation

To launch the MTK UI from the EZ-PD™ Configuration Utility, follow the menu path Tools > Manufacturing Test Kit -EMCA.
GUI usage

The Manufacturing Test Kit GUI is displayed as shown in the Figure 88 and Figure 89. The MTK UI has a Display window, Message window, and MTK control and settings options.

Manufacturing Test Kit EMCA provides configuration update for CMG1, CMG2 and CCG2 and firmware update for CCG2.

Figure 87 Manufacturing test kit - EMCA option

Figure 88 MTK GUI layout for CMG1/CMG2
The MTK control options are listed in Table 3.

**Table 3  MTK control options**

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>The MTK <strong>Start</strong> option will be enabled only when the correct configuration cyacd is loaded. Clicking <strong>Start</strong> will initiate the detecting and testing of the Infineon EMCA devices.</td>
</tr>
<tr>
<td>Stop</td>
<td>The MTK <strong>Stop</strong> option will complete the current test in progress and stop testing and detecting Infineon EMCA devices.</td>
</tr>
<tr>
<td>Refresh</td>
<td>The <strong>Refresh</strong> option will detect the MTK hardware connected to the PC.</td>
</tr>
</tbody>
</table>

The MTK Settings are listed in Table 4.

**Table 4  MTK settings**

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure cable</td>
<td>Configures the EMCA devices based on the selected configuration cyacd file.</td>
</tr>
<tr>
<td>Test cable</td>
<td>Sends the Discover Identity PD command and verifies the response with the loaded configuration cyacd file to declare Pass/Fail. This control is applicable for CMG1 and CMG2 devices.</td>
</tr>
<tr>
<td>Stop test on failure</td>
<td>When a failure is detected by one of the MTKs, the current testing on the rest of detected EMCA devices will complete and the MTK will stop detecting and testing EMCA devices.</td>
</tr>
</tbody>
</table>
GUI usage

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If this option is not checked, then only the MTK that has detected a failure will stop detecting the EMCA device. The remaining MTKs will continue to detect and test EMCA devices.</td>
</tr>
<tr>
<td>Program SOP'</td>
<td>Programs the SOP' CCG2 device based on configuration and firmware files selected.</td>
</tr>
<tr>
<td>Program SOP’’</td>
<td>Programs the SOP’’ CCG2 device based on configuration and firmware files selected.</td>
</tr>
</tbody>
</table>

The various fields of the display window are described in Table 5.

Table 5  Fields in display window

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sl No</td>
<td>Serial number of the connected MTK hardware</td>
</tr>
<tr>
<td>Tester ID</td>
<td>Unique tester ID of the MTK hardware</td>
</tr>
<tr>
<td>Cable ID</td>
<td>Unique cable ID of the connected EMCA device</td>
</tr>
<tr>
<td>Current Status</td>
<td>The Current Status can be one of the following status:</td>
</tr>
<tr>
<td></td>
<td>• Cable detached – Yellow LED (EMCA device not attached)</td>
</tr>
<tr>
<td></td>
<td>• Cable attached – Yellow LED (EMCA device attached)</td>
</tr>
<tr>
<td></td>
<td>• Configuring – Yellow LED (EMCA device configuration in progress)</td>
</tr>
<tr>
<td></td>
<td>• Updating Firmware – Yellow LED (CCG2 device firmware update in progress)</td>
</tr>
<tr>
<td></td>
<td>• Testing – Yellow LED (CMG1/CMG2 device testing in progress)</td>
</tr>
<tr>
<td></td>
<td>• Passed – Green LED (Test passed)</td>
</tr>
<tr>
<td></td>
<td>• Failed – Red LED (Test failed)</td>
</tr>
</tbody>
</table>

The logs/reports are maintained for each MTK. Select Open report to open the report. Select Save log to save a consolidated log.

The MTK UI can detect a maximum of 10 MTK hardware. Each MTK hardware should be programmed and configured with a unique Tester ID.
4.10 Power adapter batch programming

The EZ-PD™ Configuration Utility supports Power Adapter Batch Programmer option, a special mode to update firmware of Infineon power adapters devices in CC interface.

The Power Adapter Batch Programmer consists of the Power Adapter Batch Programmer UI, MTK firmware, and MTK hardware.

The Power Adapter UI will be invoked from the EZ-PD™ Configuration Utility. The MTK Firmware will be available as part of the EZ-PD™ Configuration Utility installation package.

The Power Adapter Batch Programmer supports firmware update of up to 6 Infineon power adapter devices in parallel. This option is designed to be used in factories during the manufacturing cycle. Figure 91 shows the setup for updating firmware for 6 power adapters in CC interface.

The MTK hardware consists of the CY4532 EZ-PD™ CCG3PA EVK Power board (ensure the CY4532 EZ-PD™ CCG3PA EVK Main board is detached from the power board). For more details, see the CY4532 product page.
Do the following to update the firmware of the Infineon power adapter devices with the MTK:

- Download MTK firmware
- Configure MTK tester ID
- Power Adapter Batch Programmer operation

For downloading the MTK Firmware and Configuring MTK Tester ID please refer sections 4.9.1 and 4.9.2 respectively.
GUI usage

4.10.1 Power adapter batch programmer operation

To launch the Power Adapter Batch Programmer from the EZ-PD™ Configuration Utility, follow the menu path 
Tools > Power Adapter Batch Programmer.

![Figure 92](image1.png)

**Figure 92** Power Adapter Batch Programmer option

![Figure 93](image2.png)

**Figure 93** Power adapter batch programmer layout
GUI usage

The MTK control options are listed in Table 6.

### Table 6 MTK control options

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>The MTK <strong>Start</strong> option will be enabled only when the correct firmware <code>cyacd/cyacd2</code> is loaded. Clicking <strong>Start</strong> will initiate firmware update of the Infineon power adapter devices.</td>
</tr>
<tr>
<td>Stop</td>
<td>The MTK <strong>Stop</strong> option will complete the current firmware update in progress and stop detecting Infineon power adapter devices.</td>
</tr>
<tr>
<td>Refresh</td>
<td>The <strong>Refresh</strong> option will detect the MTK hardware connected to the PC.</td>
</tr>
</tbody>
</table>

Table 7 shows the device parameters for power adapter batch programmer.

### Table 7 Fields containing device parameters

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part Number</td>
<td>The part number enumerates automatically when the firmware <code>cyacd/cyacd2</code> file is selected.</td>
</tr>
<tr>
<td>Device Family</td>
<td>The device family enumerates automatically when the firmware <code>cyacd/cyacd2</code> file is selected.</td>
</tr>
</tbody>
</table>

Table 8 shows the MTK power adapter unit fields.

### Table 8 Fields in unit layout

<table>
<thead>
<tr>
<th>MTK control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tester ID</td>
<td>Unique tester ID of the MTK hardware</td>
</tr>
<tr>
<td>Device ID</td>
<td>Unique device ID of the connected power adapter device</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Optional serial number that can be scanned from the device to add it to the log.</td>
</tr>
<tr>
<td>Status</td>
<td>The Current Status can be one of the following statuses:</td>
</tr>
<tr>
<td></td>
<td>- Detected – White status button (PA device detected)</td>
</tr>
<tr>
<td></td>
<td>- Detached – White status button (PA device not attached)</td>
</tr>
<tr>
<td></td>
<td>- Attached – White status button (PA device attached)</td>
</tr>
<tr>
<td></td>
<td>- In-Progress – Cyan status button (PA device firmware update in progress)</td>
</tr>
<tr>
<td></td>
<td>- Passed – Green status button (Firmware update passed)</td>
</tr>
<tr>
<td></td>
<td>- Failed – Red status button (Firmware update failed)</td>
</tr>
</tbody>
</table>

The power adapter batch programmer UI detects a maximum of six MTK hardware. Each power adapter’s firmware is updated parallely from the firmware `cyacd/cyacd2` file provided.

Press the **Stop** button as shown in Figure 94 to terminate the firmware update process. Click **View Report** to get an individual device's log.

Use **Save log** to save the consolidated log in a text file.
GUI usage

Figure 94  Power adapter batch programmer operation

4.11 Utility options

To change the options, follow the menu path **Tools > Options**. Figure 95 shows the available options.

Figure 95  Options menu
GUI usage

4.11.1 Enable device verbose logs
Select this option to view detailed logs. The logs displayed in the Messages tab can be functional logs required to identify status, or verbose logs that can be used for debugging.

4.11.2 Allow multiple PDOs of the same voltage
Select this option to create a configuration with multiple PDOs of the same voltage.

4.11.3 Re-enumetation timeout
Programming an USB-Serial based Billboard device involves sending a command, which causes the device to disconnect and reconnect in manufacturing mode. When this mode switch happens for the first time on a particular USB host port, there could be delays in binding the device to the Cypress USB driver. The Re-enumetation timeout parameter specifies the maximum amount of time that the utility will wait for detecting the billboard device in USB Serial Manufacturing mode. If this time is shorter than the delay in driver binding, the flashing operation in billboard device will fail.

4.11.4 UFP restart timeout
Programming UFP device over CC interface involves sending reset device command, which causes UFP device and flashing controller to restart. Some UFP devices take longer to restart, which leads to failure of power contract between UFP and flashing controller. The UFP restart timeout parameter specifies the amount of time that the utility will wait to get UFP device ready after restart.

4.11.5 HPI slave address
The CCGx devices are connected as I2C slave devices to the USB-Serial Bridge used as a bridge for programming. The utility looks for and detects CCGx devices that are configured with the default I2C slave addresses of 0x08, 0x12, and 0x44. If the target design uses I2C slave addresses that is not on this list, you can enter the additional addresses in the HPI slave address textbox. You can add any number of I2C addresses, separated by spaces.

4.11.6 Default flashing VID and index
CCGx devices in EMCA and power adapter applications are programmed through the CC interface. The flash programming commands in these applications are sent in the form of Unstructured Vendor Data Messages (UVDM) commands. These UVDMs will only be supported by CCGx when it has entered a dedicated flashing alternate mode. CCGx firmware uses the Cypress/Infineon VID (0x04b4) and alternate mode index 1 as the flashing alternate mode by default. If the target design uses a different VID or mode index, specify the corresponding values using the Default flashing VID and Default flashing mode index settings.
### C File output template

This setting can be used for selecting a template for generating the output .c configuration file. The .c configuration file generated while saving configuration can be in either of the following formats.

- **Default format:** In the default format, the .c configuration file contains the configuration table section along with the following function definitions.
  ```c
  const pd_config_t * get_pd_config(void)
  {
      return (pd_config_t *)&gl_config_table;
  }
  
  const pd_port_config_t * get_pd_port_config(uint8_t port)
  {
      return (pd_port_config_t *)&(((pd_config_t *)gl_config_table)>
          port_conf[port]);
  }
  ```

- **CCGx Auto application format:** In the auto application format, the .c configuration file does not contain the above function definitions. It contains only the configuration table section.

Use the CCGx Auto template only for auto applications. For all remaining CCGx applications, select the Default template.

### Sort PDOs

Select this option to sort Source and Sink PDOs.

### Firmware download for USB serial

A separate firmware update tool CyUSBSerial_FwUpdate_Tool.exe is available in the USBSerialFwDownloadTool folder in the utility installation.

This tool is used to update the firmware for the USB Serial. The improved USB serial firmware is intended for the CY4532 Kit and will reduce the firmware download time over the CC line.

Follow these steps to download the improved USB serial firmware:

1. Power the CY4532 kit and connect to the Windows PC.
2. In the Device Manager, check if the CY4532 kit is enumerated and bound to the Cypress driver.
3. Double-click the CyUSBSerial_FwUpdate_Tool.exe tool.

After the firmware is successfully downloaded, the message, shown in Figure 96, will be displayed in the command window.
4.13 Help

The EZ-PD™ Configuration Utility provides the following help options:

- Windows Help document, which describes the various UI screens and their usage. You can access this document from the Help Topics menu.
- A Context Sensitive help button (?), which is part of every UI window. Clicking this button will display documentation for that particular UI.
- Small snippets of help texts are displayed in the Help tab at the bottom of the dialog, and tooltips are displayed when you hover the mouse over the GUI components.
## Revision history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>2017-11-23</td>
<td>Initial release</td>
</tr>
</tbody>
</table>
| *A      | 2018-04-02 | Added Section 3.2.6 for Programming MTK Hardware  
Added Section 4.11 for CMG1 Manufacturing Test Kit operation.                                      |
| *B      | 2018-07-16 | Added Section 4.13 for USB Serial Firmware Download.                                                                                          |
| *C      | 2018-11-29 | Updated section 4.1 to add Thunderbolt Host Configuration and reference to CCGx Host/Power SDK User Guides.  
Added section 4.1.14 to capture Thunderbolt Host Configuration node                                                              |
| *D      | 2019-08-05 | Updated Sections 1.1, 2.1.1, 4, 4.1.2, 4.6.1 for adding support for PAG1S/CCG5C/CCG6/ACG1F family.  
Figures updated: 2-1, 2-2, 2-3, 4-2, 4-3, 4-4, 4-10, 4-19, 4-24, 4-26, 4-35, 4-57 |
| *E      | 2019-09-26 | Added section 4.14 Blackbox Utility for CCG3PA                                                                                               |
| *F      | 2020-03-26 | Updated Sections 1.1, 2.1.1, 4, 4.1.15 for adding support for CCG6DF and CCG6SF family.  
Added sections 4.1.9, 4.1.16, 4.1.17, 4.1.18 and 4.1.19 to capture SKEDB Configurations, Base Alternate Mode Configuration, Custom Alternate Mode Configuration, Custom Host Configuration and ICL/TGL Configurations respectively.  
Figure updated: 4-36  
Update Disclaimer on Page 2 |
| *G      | 2020-06-19 | Added Sections  
Extensions for hybrid architecture applications and Extensions for hybrid architecture applications support  
Added Figure 4-55, Figure 4-59, and Figure 4-62  
Updated Figure 3-1  
Added note in Section Programming over CC channel: 'The CC Firmware Update feature is not intended for field firmware update' |
| *H      | 2020-09-22 | Updated “Introduction” chapter.  
Updated “EZ-PD™ Device Family”:  
Updated description.  
Updated “EZ-PD™ Configuration Utility Overview” chapter:  
Updated “Usage Flow”:  
Updated “Create Configuration”:  
Updated table.  
Updated “GUI Usage” chapter:  
Updated description.  
Updated “USB PD Device Configuration”:  
Updated description.  
Updated “Power Settings”:  
Updated figure “Power Settings Parameter Configuration” |
## Revision history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Added “Auto Configuration”.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated “Thunderbolt Host Configurations”:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated description.</td>
</tr>
<tr>
<td>*I</td>
<td>2020-12-03</td>
<td>Moved to Infineon Template</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Figure 15, Figure 19, Figure 20, Figure 22, and Figure 52</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated USB PD device configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added BCR configuration</td>
</tr>
<tr>
<td>*J</td>
<td>2021-03-11</td>
<td>Updated Figure 28, Figure 34, Figure 46, Figure 50, and Figure 95.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Section 4.6.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Sort PDOs.</td>
</tr>
<tr>
<td>*K</td>
<td>2021-05-24</td>
<td>Updated Figure 1, Figure 15, Figure 18, Figure 20, Figure 47, Figure 48, and Figure 50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added section 4.1.22</td>
</tr>
<tr>
<td>*L</td>
<td>2022-02-18</td>
<td>Updated EZ-PD™ device family</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated sections Device parameters, Port information, and Auto configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added section Application configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Figure 69 and Figure 70</td>
</tr>
<tr>
<td>*M</td>
<td>2022-09-28</td>
<td>Added Power adapter batch programming section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Table 1 for .NET Framework</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added footnote 1 for CCG7D in Table 2 and a note in Creating configuration section</td>
</tr>
<tr>
<td>*N</td>
<td>2023-03-31</td>
<td>Updated EZ-PD™ device family</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Table 1 for Microsoft Visual C++ Redistributable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Figure 1, Figure 2, Figure 3, Figure 4, Figure 6, Figure 7, Figure 10, Figure 13, Figure 15, Figure 81, Figure 82, Figure 83, Figure 84, Figure 85, Figure 86, Figure 87, Figure 88, Figure 90, Figure 93, Figure 95, Figure 96</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Figures Figure 16 to Figure 80.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Figure 89, Figure 92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added Table 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Table 2 for adding CMG2 and PAG2S</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated sections 3.1.2, 3.2.1, 3.2.2, 4, 4.1.6.1, 4.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added section 4.10.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated section 4.9 for addition of CMG2 configuration and firmware update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Table 4, Table 5, Table 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removed EMCA batch programming and dongle batch programming sections</td>
</tr>
</tbody>
</table>
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.

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.