1. Introduction

1.1 EZ-PD™ Device Family

Cypress 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 CCG1, CCG2, CCG3, CCG4, CCG3PA, CCG3PA2, PAG1S, CCG5. CCG5C, CCG6, ACG1F and CMG1 generations of EZ-PD controllers are available from Cypress. These controllers are collectively called as CCGx controllers in the rest of the document.

1.2 EZ-PD Configuration Utility

The Cypress 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 Cypress customers according to their use cases and requirements.

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

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

1.3 Software Pre-requisites

The following table 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 2.0 or later</td>
</tr>
</tbody>
</table>

Note 1: Microsoft .NET framework can be downloaded and installed from Microsoft.com. The link to the installable on Microsoft’s website is:

2. EZ-PD Configuration Utility Overview

EZ-PD Configuration Utility can be installed through a single click installer. After successful installation, the Application can be started from the icon provided on the Desktop for Windows 7, or from the Apps menu in case of Windows 8 or Windows 10.

Figure 1: Welcome Screen with Version Information
2.1 Usage flow

The utility usage flow typically consists of three steps:

1. Create Configuration
2. Select Parameters
3. Configure Device

In the first stage, the user selects 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, the corresponding family should be selected from the dropdown menu.

Once a device type has been selected, the user moves to the second "Select Parameters" stage. All configurable parameters of the active application and their default values are shown on the Configuration tab of the GUI window. The user can view, modify and save the parameters from the configuration tab.

Once all parameters are selected and saved, the user moves to the final "Configure Device" stage. In this stage, the saved configuration is programmed onto the internal flash memory on the EZ-PD CCG controller.

Each of the above stages is described in subsequent sections.

2.1.1 Create Configuration

The user can start a New Configuration from the File Menu of the utility. On clicking the “New” option from File menu, a Select Device Type dialog is displayed. This dialog displays a list of USB-PD applications supported by CCG type selected from dropdown menu.

![Select Device Type Dialog](image)

Figure 2: New Project Creation Dialog

Since the USB-PD applications supported vary across the product families, the CCG product family should be chosen first before selecting the type of USB-PD application. The utility filters the applications available for selection based on the product family selected.
The current version of the utility supports the following CCGx applications:

<table>
<thead>
<tr>
<th>Product Family</th>
<th>CCG 1</th>
<th>CCG2</th>
<th>CCG3</th>
<th>CCG4</th>
<th>CCG5/CCG5C</th>
<th>CCG6</th>
<th>CCG3PA</th>
<th>ACG1F</th>
<th>PAG1S/PAG1P</th>
<th>CCG3PA2</th>
<th>CMG1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Electronically Marked Type-C Cables (Passive or Active EMCA)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Type-C to Display (DisplayPort, HDMI or VGA) dongles</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Type-C Port Controllers for Notebooks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Type-C Power Adapters</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Billboard controller</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The leaf nodes on the tree are the selectable options representing devices that can be configured. The Select button will be active only when one of these leaf nodes has been selected.

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

### 2.1.2 Select Parameters

The Configuration tab appears only after the device type for configuration is selected. Various groups of parameters available for configuration of the selected device are arranged in a tree for selection on the left pane of the configuration tab. In the case of devices with multiple Type-C ports, each port should be configured independently by entering parameter values under the nodes of each ports.

The user can select the appropriate group to view the parameters and the corresponding values on the right pane. Values can be modified as needed by selecting from the drop-down lists or by entering the value in the text box. The utility performs validity checks on the values input through the text box.
A description of each parameter can be viewed in the form of a tool-tip by doing mouse hover on top of the parameter name. The description is also displayed in the Help window if you click on the parameter name.

Once all parameters have been updated, the user can save the configuration. All modifications are saved to a XML file for future use. User will be prompted for selecting the desired folder location and file name when the “Save” button is clicked.

The saved file can be loaded later using the “Open” item from the File menu.

Note that the XML file itself cannot be programmed to the device. A special file with “cyacd” extension is saved along with the XML file to the selected folder. This file should be used 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. This source file can be imported into a CCGx SDK firmware project work-space so that a firmware binary with the desired default values can be compiled.

2.1.3 Device Configuration

The user can program the configuration to the device flash using the “Configure Device” option available on the “Tools” menu.
This brings up the Configure Device dialog box where the target device to be programmed can be selected, and then the programming initiated.
3. Hardware Setup

This chapter describes the hardware setup required for EZ-PD device programming.

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 like Type-C Notebook 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 Cypress's USB-PD Host and Client boards, the Embedded Controller is emulated by a Cypress 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: USB to I2C based Programming Setup](image)

The CY4501 CCG1 Development Kit is an example of a USB-PD Host (Downstream Facing Port or DFP) implementation using the CCG1 controller. This board makes use of a Cypress USB-Serial controller to allow the CCG1 device to be controlled through the USB port.

The CY4531 EZ-PD CCG3 Development Kit and CY4541 EZ-PD CCG4 Development Kit are USB-PD Host implementations using the CCG3 and CCG4 controllers respectively.
Programming Type-C to Display Dongle Controllers

Devices like display dongles that have 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 CCG1 USB Type-C to DisplayPort Cable Solution shown in Figure 7 uses this scheme to program the CCG1 over the I2C interface.

The programming setup for dongle programming is as shown above. 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 CCG1 and CCG2 devices.

Figure 8: Programming of Type-C to Display Dongles

3.1.2 Programming over CC Channel

Devices like Type-C Cable (EMCA) controllers and USB Type-C Power Adapters do not have an Embedded Controller (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 or UFP) is not present; programming the CCGx based EMCA cable requires both a host (DFP) and client (UFP) to be connected.

Figure 9: Setup for Programming through the USB-PD CC Channel
The CY4501 CCG1 Development kit provides CCG1 based host and client boards that can be used in this setup.

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 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 leads to repeated role swaps that slow down device configuration. So, users are advised to connect only one of the DRP controllers to PC running the configuration utility in such situations.

Figure 10: Programming over CC using CY4501 boards
3.2 Programming Setup

3.2.1 Programming EMCA (Cables)

The setup shown in Figure 10 is used for EMCA programming. The CY4505 CCG1 EMCA board shown in the picture can be replaced by the target EMCA that needs to be programmed.

The CY4504 CCG1 Host board should be programmed with a Notebook DRP firmware image. The Firmware\Host folder in the EZ-PD Configuration Utility installation contains a copy of the host (notebook) firmware that can be used.

The CY4503 CCG1 Client board should be programmed with a Monitor (Client DRP) firmware image. The Firmware\Client folder in the EZ-PD Configuration Utility installation contains a copy of the client firmware that can be used.

Please refer to the CY4501 Kit User Guide for details on how to configure these boards and program them using the Miniprog programmer.

Once the Host and Client boards have been configured and programmed, they can both 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 chapters in this document provide instructions on reading the configuration from and programming the EMCA devices.

3.2.2 Batch Programming of EMCA Cables

There may be cases where users want to program a large number of EMCA cables with a standard set of configuration parameters. It would be cumbersome to keep connecting the cables to the host and client boards in such a case.

To facilitate easier batch programming of EMCA cables, a custom host firmware implementation is provided with the utility. The CY4504 Host board or CY4503 Client board can be programmed with the custom flashing firmware found under Firmware\FlashUtil folder. Separate files are provided for use on the host and client boards.

Once this flashing firmware has been loaded on the host board, there is no need for the client board to be connected for EMCA programming. The required setup is shown in Figure 11.
3.2.3 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 (Figure 12).

Figure 11: Setup for Batch Programming of EMCA

Figure 12: Programming Setup for Type-C Dongle with external Billboard Device
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 CY4504/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 13.

![Diagram of programming setup](image)

**Figure 13:** Programming Setup for Type-C Dongle with internal on-chip Billboard interface

### 3.2.4 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.

Please refer to the user guides for the respective kits for more detailed programming instructions and connection photographs.

### 3.2.5 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 14.
3.2.6 Programming Manufacturing Test Kit (MTK) Hardware

The MTK hardware consists of the CY4532 EZ-PD CCG3PA Power board (ensure 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 15.
4. GUI Usage

Specific usage details for using the Graphical User Interface is described in this chapter.

The File → New menu option is used to create a new device configuration. The CCG product family (CCG1/CCG2/CCG3/CCG4/CCG3PA/CCG5/CCG5C/CCG6/CCG3PA2/CMMG1/PAG1S/PAG1P/AGC1F or USB Serial) needs to be selected first. The utility displays the applications supported by the selected product family in a tree view. The target application can be selected from this tree view.

A description of the selected application is shown on the right side of the dialog. Click on the “Create Project” button to start configuring the device.

The File → Open menu option is used to re-open a previously saved configuration. The configuration is saved in the form of an XML file and includes all of the configuration data input by the user. The opened configuration can be modified, saved and programmed to the Type-C or Billboard controller.

The File → Read from Device menu option can be used to read the current configuration from the target device. All devices currently connected to the host PC are displayed in a tree view, and the desired target can be selected for reading from. The loaded configuration can be modified, saved and re-programmed 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
  - Power Protection parameters
  - Power Settings
  - Battery Configuration
  - Charging Configuration
  - Type-A Configuration
  - Thunderbolt Host Configuration

- User Parameters

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 refer Section 3.2 of the EZ-PD CCGx Host SDK User Guide and for Power based applications (power bank, power adapter) parameters refer Section 3.2 EZ-PD CCGx Power SDK User Guide.

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.
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. The UI does not allow modification of such parameters, and shows them in grayed-out fashion for informational purposes.

The Device IDs are VDOs that needs 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 the case of EMCA applications, the Cable VDO needs to be configured.
The AMA VDO needs to be configured for applications those support one or more Alternate Modes. No product specific VDOs need to be configured in the case of Notebook Port controllers.

A tool-tip describing the parameters being configured can be seen when a mouse hover is performed on top of the parameter. The same tool-tip information can also be seen in the help window on the lower side of the UI when the parameter is selected.

### 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 the Cypress SVID (0x04B4) and Cypress flashing mode (0x01). This alternate mode is used by the EZ-PD controllers to implement configuration update and firmware upgrades through CC Vendor Defined Messages (VDMs). The value of the Flashing VID and Mode can be changed under the Device Parameters...
section. The values displayed under SVID0 will automatically be changed when flashing VID and Mode Index are changed.

The Add and Remove buttons on the top of the left panel are used to add new SVIDs and to remove existing SVIDs from the configuration. The Add button is only active when the SVID Configuration node is selected and the Remove button is only active when one of the SVIDs is selected. Figure 21 shows the buttons to add or remove SVIDs to the device configuration.

**Warning:** Devices that are programmed over the CC channel (EMCA and Power Adapters) need to support a special SVID and mode which is 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. This SVID should only be deleted if the user wants to disable the update capability in the design.

Figure 21: Adding or Removing SVIDs

Once an SVID node has been selected, the corresponding SVID value can be edited; and modes can be added / deleted on the right panel on the screen. The + and - buttons on the right panel can be used to add or delete modes from the active SVID. Figure 22 shows the locations to edit the SVID and mode values and the buttons to add or remove modes to a selected SVID.

Figure 22: 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.

Sink PDOs can describe three kinds of power requirements:

1. Fixed Voltage Supply
2. Variable Voltage Supply
3. Battery

The supply type needs to be selected first and then the rest of the parameters can be configured.

An Enable checkbox is associated with each PDO and is used to specify whether the PDO is enabled at system start-up. Solutions that support the Host Processor Interface (HPI) allow dynamic enable/disable of PDOs at runtime. In other solutions, all the PDOs present should be left enabled.

Figure 23: Configuring Source PDOs

Figure 23 and Figure 24 show the configuration screens used to set the properties of PDOs. All of the fields shown under Source PDOs are defined in the USB-PD specification. Please refer to the specification for more details.

Sink PDO configuration involves two additional properties:

1. The sink give back field specifies whether the Sink Give back feature is enabled for this PDO.
2. The minimum operating current field specifies the minimum current required for device operation in the case where give back is enabled. If give back is not enabled, this field specifies the maximum operating current that may be required.
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 properties defined by the USB-PD Specification, and the rest of Cypress device specific parameters. The tool-tips provided in the utility describe the intent of each of these parameters.

Figure 25: Configuring PD Port Information
4.1.6 Billboard Parameters

USB-PD devices like display dongles and monitors which 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 in the case of CCG3 designs or an external USB Billboard controller in the case of other designs.

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.

Designs with external Billboard controller

CCG1/CCG2 devices do not have an on-chip USB block, hence billboard functionality shall be 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 like the USB IDs used by the Billboard Controller, the string descriptors, the supported alternate mode descriptions etc. will be configured separately on the Billboard controller itself (see Section 4.2).

Figure 26: CCG1/CCG2 Billboard Control Parameter Configuration

```
1. 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.
2. The Billboard programmer support field specifies whether the vendor mode used for CCGx flash read and update should be enabled on the Billboard device.
3. The Billboard time-out field specifies the amount of time 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.
```
Designs with on-chip Billboard interface

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

The following sets of parameters need to be configured for CCG3 designs with Billboard:

a. CCGx Settings
b. Billboard Settings
c. Alternate Modes

4.1.6.1.1 CCGx Settings

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

a. Billboard Type – Internal/External/No Billboard
b. Billboard Enable – Whether Billboard functionality is always enabled or enabled only when error occurs on CCGX device
c. Billboard programmer Support - Whether flash update through Billboard device is enabled
d. Billboard timeout in seconds
e. Billboard power settings – Bus powered/Self powered
f. VConn power setting – Power that the design will draw from the VConn supply.
g. Billboard container ID and serial number settings

![CCG3 Billboard CCGx Parameters Configurations](image)

**Figure 27: CCG3 Billboard CCGx Parameters Configurations**

4.1.6.1.2 Billboard Settings

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

a. VID and PID used to identify the device
b. Manufacturer, Product, Serial Number, Configuration and Interface strings used to identify the device.

c. Specify an URL that can provide more information about the device.

![Figure 28 CCG3 Billboard Parameter Configuration](image)

### 4.1.6.1.3 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 SVID section will be listed under this node and user can edit its description in this section. To add or remove an alternate mode user need to add or remove modes under SVID section.

![Figure 29: CCG3 Billboard Alternate modes Parameter Configuration](image)
4.1.7 DP Mode Parameters

This screen is only used in Type-C port controller configuration of peripheral type PD devices. It configures the properties associated with DisplayPort alternate mode support in the system. Parameters displayed in this node shall vary depending on CCGx family and its application type.

![Figure 30: DisplayPort Mode Parameter Configuration]

1. The Modes supported property specifies the DisplayPort pin configurations supported.
2. The Mux control property specifies whether USB / DisplayPort mux is controlled by CCGx or Embedded Controller.
3. The Mode trigger property specifies whether DP mode entry is done by CCGx directly or through Embedded Controller trigger.
4. The Preferred DP Mode property only applies to DisplayPort Sink implementations (Monitor) and specifies whether the device should prefer 4-lane DisplayPort or Multi-function operation by default.

4.1.8 SCEDB Configuration

Source Capabilities Extended Data Block (SCEDB) contains the additional information about a Port’s Source Capabilities. SCEDB configurations are shown below in Figure 31.
Peak Current Configuration

The Peak Current field contains the combinations of Peak Current that the Source supports. Peak Current provides a means for Source 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 % overload, duration and duty cycle defined as PeakCurrent1, PeakCurrent2 and PeakCurrent3 that the Source supports. A Source May offer no Peak Current capability.

4.1.9 Protection Parameters

USB-PD designs like Notebook port controllers, power adapters and dock/monitors need to implement protection circuitry that ensures that the device does not get harmed due to over voltage, over current, over temperature, under voltage or short circuit conditions.

CCGx based designs can implement VBUS Over Voltage (OVP), VBUS Over Current (OCP), VBUS Under Voltage (UV), VCONN Over Current (VCONN-OCP), Over Temperature (OTP), and Short Circuit (SCP) protection schemes.
Utility allows to enable/disable these protection schemes and configure the parameters associated with them 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 corresponding configuration parameters. Please note that these protection schemes are not implemented in all firmware solutions.

**VBUS Over Voltage Protection**

This node contains the parameters that configure VBUS Over Voltage Protection (OVP).

![Figure 33: VBUS Over Voltage Protection Parameter Configuration](image)

**VBUS Over Current Protection**

This node contains the parameters that configure VBUS Over Current Protection (OCP). The configuration parameters ‘Off time and ‘Sample period’ are displayed only when value of ‘config table major version’ parameter is set to ‘1’ in ‘Device Parameter’ configuration node.

![Figure 34: VBUS Over Current Protection Parameter Configuration](image)
VBUS Under Voltage Protection

This node contains the parameters that configure VBUS Under Voltage Protection (UVP).

![Parameter Configuration](image)

Figure 35: VBUS Under Voltage Protection Parameter Configuration

Short Circuit Protection

This node contains the parameters that configure the Short Circuit Protection (SCP).

![Parameter Configuration](image)

Figure 36: 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.

Over Temperature Protection

This node contains the parameters that configure the Over Temperature Protection (OTP).

![Parameter Configuration](image)

Figure 37: Over Temperature Protection Parameter Configuration
VCONN Over Current Protection

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

![Figure 38: VCONN Over Current Protection Parameter Configuration](image)

4.1.10 Power Settings

Power settings node provides VBUS control configurations,

![Figure 39: Power Settings Parameter Configuration](image)
4.1.11 Battery Configuration

Battery configuration provides following battery and charging parameters configurations.

- Maximum battery voltage beyond which it is not safe to charge.
- Battery cutoff voltage: The battery voltage at which the device is expected to stop discharging and be on low voltage consumption.
- Battery discharge enable voltage: The battery voltage at which discharge can be re-enabled.
- Maximum charging current allowed for the battery in mA.

![Figure 40 Battery Configuration]

4.1.12 Charging Configuration

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

Source Setting

Legacy charging source setting configuration parameters for BC 1.2 and Apple charging mode are shown in the Figure 41 and Figure 42 respectively.
4.1.12.1.1  AFC Source Caps

This node contains list of Adaptive Fast Charging Source capability objects. This node is visible only when AFC charge has been enabled in source setting configuration node. AFC Source capability objects can be added or removed using buttons provided on upper right corner in the right pane of utility.
Sink Setting

This node is valid only for CCG3PA Power Bank application. It contains legacy charging sink setting configuration parameters.

4.1.13 Type-A Configuration

In addition to the Type-C port, device can optionally support a Type-A port. This node provides the configuration information for the Type-A port. This is only for CCG3PA applications.
Power Setting
This node provides VBUS control additional configurations for type-A port. These configuration parameters are same as Type-C power setting parameters.

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

4.1.14 Thunderbolt Host Configurations
This node contains Thunderbolt host configuration parameters. The Thunderbolt Host configurations are only applicable for CCG5, CCG5C and CCG6 device applications.

4.1.15 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 Cypress 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 in two ways:

a. Using a set of eight 4-byte hexadecimal values. This method allows any parameter value to be entered.

b. Using an ASCII string of 32-bytes or lesser. Only printable characters can be input in this case.

Note: Please note that any non-printable characters in the hexadecimal input will be lost when switching to the ASCII string view. Please 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 sets of parameters 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.

### 4.2.2 Flashing Parameters

The Billboard device implementation can have two modes of operation:

- **Billboard mode**: In this mode, the Billboard device provides the 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 users to update 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 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.

![Flashing Parameter Configuration](image)

**Figure 49: Flashing Parameter Configuration**

The Flashing Interface VID and PID values cannot be changed as these are used to bind the Billboard device to the Cypress USB driver used to do the programming. The Manufacturer and 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:

a. VID and PID used to identify the device
b. Manufacturer, Product, Serial Number, Configuration and Interface strings used to identify the device.
c. Container ID value to be reported in the BOS descriptor.
d. Amount of power required by the USB-PD device from the VConn supply.
e. Configuration for the I2C interface between the Billboard controller and the USB-PD controller.
f. Specify an URL that can provide more information about the device.
g. The GPIOs used to avoid simultaneous access of CCGx HPI by billboard controller and EC.
h. Configuration for SCB1 block.

Figure 50: Billboard Parameter Configuration

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 Cypress Billboard implementation varies from 1 to 8. These alternate mode definitions are added/deleted using the + and - buttons on the left panel of the screen. Once an alternate mode has been added to the project, the following fields need to be configured:
4.2.5 GPIO Configuration

Unused GPIOs can be configured as input or output GPIO. For output GPIOs, the initial state & drive mode can be configured. The initial state of output GPIO can be set as 0 or 1 using “GPIO-x initial state” field. The drive mode of output GPIOs can be set as “Open-drive, drive-low”, “Open-drive, drive-high”, “Strong drive”, “resistive pull-up”, “resistive pull-down”, “resistive pull-up/down”.

4.3 Saving the Configuration

The above options complete the configuration required for the USB-PD or Billboard solutions. Once the configuration is complete, the configuration can be saved using the File → Save As button.
Every configuration made on the GUI will be saved to an XML file for further action by the user. The saved XML file can be re-opened and edited for further changes as required.

The configuration is also saved in the form of a Cypress defined binary file called the cyacd file. The cyacd file will be created with the same file name 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 work-space so that a firmware binary with the desired default values can be compiled.

4.4 Reading Configuration from Firmware file

The configuration data can be read from firmware file stored in `.hex` or `.cyacd` file format using **File → Read from Firmware File** menu option.

In CCG1 and CCG2 applications `.cyacd` format firmware file doesn’t contain configuration data hence utility doesn’t support reading configuration from `.cyacd` firmware file of CCG1/2 applications.
In dual firmware applications `.hex` format firmware file contains two firmware images and corresponding configurations. EZ-PD Configuration Utility provides option for selecting firmware image Id from which configuration to be read (Figure 53).

**Figure 53: Reading Configuration from Firmware File**

### 4.5 Saving Configuration to Firmware File

The configuration created or opened in EZ-PD Configuration Utility can be saved to firmware file using **File → Save to Firmware File** menu option. The configuration can be saved to either `.cyacd` or `.hex` firmware file formats.

In CCG1 and CCG2 `.cyacd` format firmware file doesn’t contain configuration data hence saving configuration to `.cyacd` firmware file is not supported for CCG1/2 applications.

In dual firmware applications `.hex` format firmware file contains two firmware images and corresponding configurations. While saving configurations to dual firmware `.hex` format firmware file EZ-PD configuration utility updates configuration of both the firmware images.
4.6 Configuring the Device

Once all relevant values are entered and verified by the user, the configuration can be programmed to the device using the Tools → Configure Device menu option.

On the left half of the dialog, the devices are shown in a tree structure that represents the way the devices are connected. On selecting (clicking on) any of the devices, various properties of the device are displayed on the right half of the dialog.

Once the file and a supported device have been selected, the Program button can be used to initiate programming of the device.

The Refresh button can be used to start a fresh scan of EZ-PD devices connected to the PC host. The status bar on 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.
4.6.1 Extensions for dual firmware devices

The flash architecture is different for different CCG device families. CCG1/2 devices have only 32 KB of flash memory. CCG3PA devices have 64KB flash memory. CCG1/2, CCG3PA and PAG1S devices make use of a single firmware binary with flash access (firmware or configuration update) supported through a dedicated boot-loader application. CCG3/4/5 devices have 128 KB of flash memory, allowing the corresponding solutions to implement a robust dual-firmware solution.

In the case of CCG1, CCG2, CCG3PA and PAG1S controllers, configuration or firmware update can only be performed by switching the device to a special boot-loader mode. The USB-PD port functionality is not supported in boot-loader mode; which means that these operations will disrupt the USB-PD port operation.

In the case of 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. The user can reset the dual firmware device at an appropriate time to switch to the newly updated firmware or configuration.
Single Firmware Mode (Legacy Boot Mode)

Since CCG1, CCG2, CCG3PA and PAG1S devices only support device programming through the boot-loader, only one programming option is available with these devices.

On start of programming, the utility will request the device to jump to bootloader mode where all PD capabilities are disabled. Once the configuration update is completed, the device will be reset again into the firmware application mode.

Dual Firmware Mode

There are two ways user can configure the device if device has dual firmware:

i. 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.

After clicking the program button, the running firmware will update the configuration table of alternate firmware without interrupting activities of 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.

In case the alternate firmware version is older than the running firmware, the utility will pop-up a warning that indicates that the firmware will roll back to an older version at the end of the operation. The user has an option to continue or abort the operation at this stage.
ii. 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 the user wants 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 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 device supports dual firmware functionality, running firmware can be switched to alternate firmware by clicking 'Change App Priority' button. This operation will be successfully performed only when alternate firmware is valid.

Notes:

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 messages indicating the progress are displayed in the message box. A progress bar also indicates the progress of the operation.

![Configure Device Completion](image)

Figure 57: Configure Device Completion

### 4.7 Reading Configuration from Device

The Type-C configuration stored on the EZ-PD controller connected to the computer can be read by the utility and displayed in the UI screens. This operation is selected using the File → Read from Device menu option.

The Read dialog is similar to the Configure Device dialog and displays a tree of devices connected. One of these devices can be selected to read data from.

The configuration read from the device will be displayed using the same screens as used for completing the configuration. These parameter values can now be edited, saved and programmed back to the controller.
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.
The default behavior of the utility is to read and return the configuration associated with the alternate firmware. The “Use bootloader to read” option shown in Figure 59 can be used 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 boot-loader mode for doing the flash reads.

### 4.8 Firmware Update

The firmware running on the EZ-PD or Billboard controller can be updated using the EZ-PD Configuration Utility. This operation is selected using the Tools → Firmware Update menu option.

As in the case of device configuration and configuration load, the firmware update can be performed on any of the supported devices. The desired controller can be selected using the Target Device option and the firmware binary file should be chosen.

Firmware for the CCGx controllers are provided in .cyacd 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 is also shown during the firmware update process.
**Warning:** Please do not disconnect the devices from the computer while the firmware update is in progress.

![Firmware Update Dialog Box](image)

**Figure 60: Firmware Update Dialog Box**

### 4.8.1 Extensions for dual firmware devices

The user can specify one or both firmware binaries through the firmware update dialog, and the appropriate binary will be used for the update operation.
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 no FW2 binary is provided by the user, an error will be reported.

It is possible to update both FW1 and FW2 binaries at one shot by using “Use bootloader for flashing” option shown in Figure 61. In this case, CCG will disable all PD ports and revert to bootloader mode so that both firmware banks can be updated.
4.9 EMCA Batch Programming

The EZ-PD™ Configuration Utility supports a special Batch Programming mode for EMCA devices. This batch programming mode is designed to program the same configuration settings and/or firmware binaries into several USB Type-C cables without going through separate UI operations.

The required settings can be selected at the start of the session, and then the cables to be programmed can be plugged in one-by-one. The utility automatically detects the cable plug-in, programs the CCGx controller and then prompts the user to plug in a new cable.

The batch programmer requires the cables to be connected to a CCG1 Host or Client board (CY4504 or CY4503) running a specially designed flashing firmware. This firmware can be found at the path Firmware\FlashUtil within the utility installation. Separate files are provided for use on the CY4503 and CY4504 kits respectively. The CCG1 host programmed with this firmware will show up on the UI screen as "DFP-FLASHUTIL".

The EMCA Batch Programmer Dialog is used for the EMCA batch programming operation.

![EMCA Batch Programmer Dialog](image-url)

Figure 62: EMCA Batch Programmer Dialog
To start the programming, select the FLASHUTIL device to be used for programming, the configuration and firmware CYACD files, and select the EMCA controllers to be programmed (near-end SOP and/or far-end SOP). The "Update bcdDevice" option is used to specify that the bcdDevice field in the Product VDO should be incremented for each device programmed.

Once the Start button is programmed, the programming operation is started.

Unlike in the other operations like Configure Device, the Batch Programmer dialog stays active while the programming is in progress. The dialog needs to be explicitly dismissed by the user.

The dialog displays the number of devices programmed and the current status of the operation on the status bar at the bottom. When the status message says Please plug in new EMCA cable, the user is expected to remove the current EMCA cable and plug-in a new cable to be programmed. The operation continues until the user presses the Stop button, or a fatal error (FLASHUTIL controller error) is encountered.

4.9.1 Extension for CCG3 based EMCA Programming

If the EMCA controllers to be programmed use CCG3 devices, additional parameters need to be specified as part of the batch programming flow. The following operations can be performed:

1. Firmware update of one copy of the firmware
2. Firmware update of both copies of firmware
3. Configuration update of valid copies of firmware.
4. Firmware and configuration update on one or both copies of firmware.

All of these operations are performed through the CCG boot-loader and will involve a device reset.
4.10 Dongle Batch Programming

A special batch programming mode is supported for Type-C to Display dongle applications as well. The Dongle Batch Programmer dialog allows users to program multiple Type-C to Display dongles without repeated actions on the UI.

The files to be programmed include:

1. Configuration file for the CCGx AMA controller (generated through the utility).
2. Firmware binary for the CCGx AMA controller.
3. Configuration file for the USB-Serial Billboard controller (generated through the utility).
4. Firmware binary for the USB-Serial Billboard controller.

At least one of the above files needs to be selected for programming.

The programming is done through the USB-Serial Billboard controller. The Billboard controller is expected to support a HID USB device interface through which the programming request is initiated. The USB device IDs (Vendor and Product ID) associated with the HID device to be used for programming should be specified.
Once the programming is started, the utility will perform the following operations:

1. Wait for a Billboard device with the specified device IDs to be connected.
2. Switch the Billboard device to programming mode.
3. Program the CCGx dongle controller if required.
4. Program the Billboard controller if required.
5. Request the user to plug in the next dongle device.

The utility will wait for a maximum of one minute for each dongle to be plugged in. If a dongle is not detected within this period, the batch programming will be aborted. The user can restart the operation at a later stage.
4.11 Manufacturing Test Kit Option

The EZ-PD™ Configuration Utility supports a special configuring and test mode for CMG1 EMCA devices.

The Manufacturing Test Kit consists of the MTK User Interface (UI), MTK firmware and the 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 install package.

The MTK UI supports configuring and testing of up-to 10 CMG1 cables in parallel. This option is designed to be used in customers factories during the manufacturing cycle. The setup for configuring and testing 10 cables is shown in Figure 65.

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 refer the following link:

The following operations are required to configure and test the CMG1 EMCA devices with the MTK.

1. Download MTK firmware
2. Configure MTK Tester ID
3. Create CMG1 configuration
4. MTK operation
4.11.1 Download MTK firmware

The hardware setup for downloading the firmware is described in section 3.2.6. 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 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. Check the “Use bootloader to flash” option as shown in Figure 66 and program the CY4532 power board.

Figure 66: MTK Firmware Update

Once the firmware images are updated successfully, the Application Type in Firmware update window will be changed to MTK as shown in Figure 67
4.11.2 Configure MTK Tester ID

To uniquely identify the MTK hardware, the user can provide a tester ID by configuring the user parameter of the MTK hardware configuration. To configure the tester ID, first we must read the configuration information from the MTK hardware. From the File menu select the “Read from Device option” (Refer Figure 58). Update the “Parameter 1” in the “User Parameters” section of the configuration with 2-byte tester ID as shown in Figure 68.
Save the configuration and program the saved configuration as shown in the Figure 69. If the programming is successful, the programmed tester ID will be displayed in the tester ID column of the MTK display window (Refer Figure 72).
4.11.3 Create CMG1 Configuration

Create a new configuration project with the “Device Type” as CMG1. (Refer Create Configuration section for more details).

The user can select the appropriate group to view the parameters and the corresponding values on the right pane. Values can be modified as needed by selecting from the drop-down lists or by entering the value in the text box. The utility performs validity checks on the values input through the text box. The CMG1 Configuration options are shown in Figure 70.
4.11.4 MTK Operation

The MTK UI will be launched from EZ-PD™ Configuration Utility by selecting the “Manufacturing Test Kit Tab” in Tools menu as shown in Figure 71

Note: Selecting "yes" for Lock configuration option will treat the CMG1 device as one-time configurable part.
The Manufacturing Test Kit GUI is displayed as shown in the Figure 72. The MTK UI has Display window, Message window and MTK control and settings options.
Figure 72: MTK GUI Layout
The MTK control options are described in Table 1

**Table 1: MTK Control Options**

<table>
<thead>
<tr>
<th>MTK Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>The MTK “Start” option will be enabled only when correct configuration cyacd is loaded. Clicking on “Start” will initiate the detecting and testing of the CMG1 devices.</td>
</tr>
<tr>
<td>Stop</td>
<td>The MTK “Stop” option will complete the current test in progress and stop testing and detecting CMG1 devices.</td>
</tr>
<tr>
<td>Refresh</td>
<td>The “Refresh” option will detect the MTK hardware connected to the PC.</td>
</tr>
</tbody>
</table>

The MTK Settings are described in Table 2

**Table 2: MTK Settings**

<table>
<thead>
<tr>
<th>MTK Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure cable</td>
<td>Selecting the “Configure cable” checkbox will configure the CMG1 device based on the configuration cyacd file selected.</td>
</tr>
<tr>
<td>Test cable</td>
<td>Discover Identity PD command will be sent and the response will be verified with the loaded configuration cyacd file to declare Pass/Fail.</td>
</tr>
<tr>
<td>Stop test on failure</td>
<td>When a failure is detected by one of the MTK’s, the current testing on the rest of detected CMG1 devices will complete and the MTK will stop detecting and testing CMG1 devices. If this option is not checked, then only the MTK that has detected a failure will stop detecting CMG1 device. The remaining MTKs will continue to detect and test CMG1 devices.</td>
</tr>
</tbody>
</table>

The various fields of the display window are described in Table 3

**Table 3: Fields in display window**

<table>
<thead>
<tr>
<th>Field Name</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 CMG1 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 (CMG1 device not attached)</td>
</tr>
<tr>
<td></td>
<td>Cable attached – Yellow LED (CMG1 device attached)</td>
</tr>
<tr>
<td></td>
<td>Configuring – Yellow LED (CMG1 device configuration in progress)</td>
</tr>
<tr>
<td>LED Status</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Yellow</td>
<td>Testing (CMG1 device testing in progress)</td>
</tr>
<tr>
<td>Green</td>
<td>Passed (Test passed)</td>
</tr>
<tr>
<td>Red</td>
<td>Failed (Test failed)</td>
</tr>
</tbody>
</table>

The logs/reports are maintained for each MTK, “Open report” option will open the report. “Save log” option is available in the Message window for consolidated log report.

A maximum of 10 MTK hardware’s can be detected by the MTK UI. Each MTK hardware should be programmed and configured with a unique Tester ID.

The CMG1 devices will be configured from the configuration cyacd file that is provided.

Figure 73: MTK operation
4.12 Utility Options

User can change the options by selecting the Tools → Options menu. Following options are available (Figure 74).

- **Verbose Logs**
  The logs printed to the messages tab of the EZ-PD configuration utility can be functional logs required to identify status, or verbose logs that can be used for debugging.

- **Allow Multiple PDOs of Same Voltage**
  This setting can be used for creating configuration with multiple PDOs of same voltage.

- **Re-enumeration 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 is happening 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-enumeration 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.

- **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 takes long time to restart which leads to failure of power contract between UFP and flashing controller. The “UFP Restart Timeout” parameter specifies amount of time that the utility shall wait to get UFP device ready after restart.
4.12.5 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 which are configured with the default I2C slave addresses of 0x08, 0x12 and 0x44. If the target design uses I2C slave addresses outside of this list, you can use the “HPI Slave Address” text box to provide additional addresses. The user can add any number of I2C addresses, separated by spaces.

4.12.6 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 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, the corresponding values can be specified using the “Default Flashing VID” and “Default Flashing Mode Index” settings.

4.12.7 C File Output Template

The ‘.c’ configuration file generated while saving configuration can be one of the following two formats.

1. Default format: The default format ‘.c’ configuration file contains configuration table section along with following two function definitions.

   ```c
   const pd_config_t * get_pd_config(void)
   {
     return (pd_config_t *)&gl_config_table;
   }
   
   and
   
   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]);
   }
   ```

2. CCGx Auto application format: The auto application format ‘.c’ configuration file doesn’t contain above function definitions. It contains only configuration table section. This setting can be used for selecting template for generating output ‘.c’ configuration file. The ‘CCGx Auto’ template should be used only for auto applications. For all remaining CCGx applications select ‘Default’ template.

4.13 Firmware Download for USB Serial

A separate firmware update tool “CyUSBSerial_FwUpdate_Tool.exe” is available in C:\Program Files (x86)\Cypress\EZ-PD Configuration Utility\USBSerialFwDownloadTool. 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 the steps below to download the improved USB serial firmware:
1. Power the CY4532 kit and connect to the Windows PC.

2. Check in the Device manager if the CY4532 kit is enumerated and bound to Cypress driver.

3. Click on the “CyUSBSerial_FwUpdate_Tool.exe” tool.

4. After Successful firmware download, the following message will be displayed in the command window as shown in Figure 75.

![Figure 75: USB Serial Firmware Download Success](image)
4.14 Help

Three types of Help Documentation are integrated with the EZ-PD Configuration Utility:

1. This document describes the various UI screens and their usage in the form of a Windows Help document. This is accessible through the Help Topics menu.

2. A Context Sensitive help button (?) is part of every UI window. Clicking this button brings up the appropriate help documentation from the above Help document.

3. Small snippets of help texts (tool-tips) are displayed in the Help tab on the bottom of the GUI on mouse click over GUI components.
## 5. Revision History

### 5.1 Document Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Issue Date</th>
<th>Origin of Change</th>
<th>Description of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>0**</td>
<td>23/Nov/2017</td>
<td>SUKU</td>
<td>Initial release</td>
</tr>
<tr>
<td>*A</td>
<td>02/Apr/2018</td>
<td>NAMJ</td>
<td>Added Section 3.2.6 for Programming MTK Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Added Section 4.11 for CMG1 Manufacturing Test Kit operation.</td>
</tr>
<tr>
<td>*B</td>
<td>16/Jul/2018</td>
<td>NAMJ</td>
<td>Added Section 4.13 for USB Serial Firmware Download.</td>
</tr>
<tr>
<td>*C</td>
<td>29/Nov/2018</td>
<td>NAMJ</td>
<td>1. Updated section 4.1 to add Thunderbolt Host Configuration and reference to CCGx Host/Power SDK User Guides.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Added section 4.1.14 to capture Thunderbolt Host Configuration node</td>
</tr>
<tr>
<td>*D</td>
<td>02/Aug/2019</td>
<td>NAMJ</td>
<td>1. Updated Sections 1.1, 2.1.1, 4, 4.1.2, 4.6.1 for adding support for PAG1S/CCG5C/CCG6/ACG1F family.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Figures updated: 1, 2, 3, 17, 18, 19, 25, 34, 39, 41, 50, 70</td>
</tr>
<tr>
<td>*E</td>
<td>26/Sep/2019</td>
<td>NAMJ</td>
<td>Added section 4.14 Blackbox Utility for CCG3PA</td>
</tr>
<tr>
<td>*F</td>
<td>22/Oct/2019</td>
<td>NAMJ</td>
<td>Removed Section 4.14</td>
</tr>
</tbody>
</table>