Designing USB 3.1 Type-C Cables Using EZ-PD™ CCG2

Author: Gayathri Vasudevan

Associated Project: No

Associated Part Family: CYPD2103-20FNXIT, CYPD2103-14LHXIT

Software Version: N/A

Related Application Notes: AN95599

To get the latest version of this application note, or the associated project file, please visit http://www.cypress.com/go/AN95615.

AN95615 explains the design of passive[1] USB 3.1 Type-C cables using EZ-PD™ CCG2. This application note explains how manufacturers can easily design and manufacture passive electronically marked cable assembly (EMCA)[1] using CCG2.

Contents

Introduction

The USB Type-C Cable and Connector Specification defines a new sub-3-mm thin receptacle, a 2.4-mm reversible plug, a cable that can be reversed in both directions and enables the 100-W USB-Power Delivery specification. The USB Type-C Specification requires cables to be electronically marked to report their capabilities to Type-C ports on hosts and devices. The electronic marking is accomplished by embedding a controller chip into the plug at one or both ends of the cable. Key requirements for such a controller chip are low cost, small footprint, low power, a turnkey solution, and a flexible firmware upgrade process.

EZ-PD™ CCG2 (CCG2) is Cypress's low-cost USB Type-C Cable controller device targeted for such electronically marked cable assemblies (EMCA, described in section "EMCA Applications"). CCG2 is available in 1.63 mm × 2.03 mm, 20-ball WLCSP and 2.5 mm × 3.5 mm × 0.6 mm, 14-pin DFN packages, and requires only five external passive components. CCG2 is the second-generation product of the Cypress family of USB Power Delivery and Type-C controllers with an ARM® Cortex®-M0 CPU. CCG2 includes a hardware implementation of the USB Type-C transceiver and USB power delivery IP. In addition, it has six Timer/Counter/Pulse Width Modulators (TCPWMs), two serial communication blocks (SCBs), nine GPIO pins, 32 KB of flash, and 4 KB of SRAM.

This application note explains various aspects related to the design of a USB 3.1 EMCA Type-C cable using EZ-PD CCG2.

[1] Passive EMCA is an Electronically Marked Cable Assembly without a re-driver or electronics to condition the USB data signals.

Introduction to USB Type-C Cables

USB has emerged as the preferred standard for connectors for data transfer and charging for PCs and smartphones. Standard Type-A, Type-B and Micro-AB connectors shown in Figure 1 are the current USB-IF standards, but they have the following limitations:

- They use large connectors that prevent slim industrial designs (plug height: A = 4.5 mm; B = 10.4 mm).
- They require a fixed-plug orientation and a fixed-cable direction.
- They carry only USB signals and VBUS (= 5 V only).
- Power delivery implementation is complicated, expensive, and limited to 7.5 W.

The USB Type-C specification is the new USB-IF standard that solves these problems and provides the following advantages:

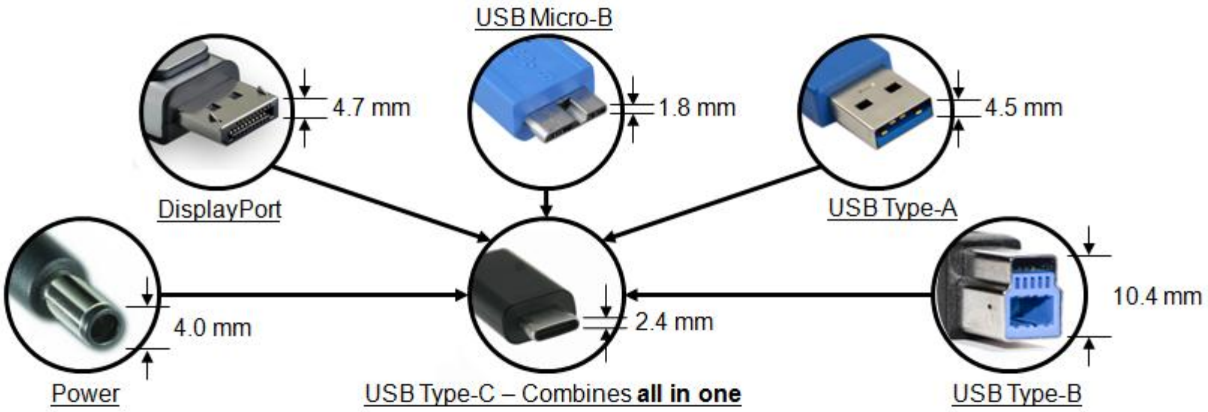
- Slim industrial design with a 2.4-mm plug height
- Reversible plug orientation and cable direction
- Transport of both USB signals and alternate mode signals, such as PCIe or DisplayPort signals, on the same connector
- Easy implementation of low-cost power delivery up to 100 W

A USB Type-C receptacle, plug, and cable provide a smaller, thinner, and more robust alternative to the existing USB 3.1 interconnects (Standard and Micro USB cables and connectors). These are targeted for use in very thin platforms, ranging from ultra-thin notebook PCs to smartphones in which existing Standard-A and Micro-AB receptacles are deemed too large, difficult to use, or inadequately robust.

While the USB Type-C interconnect (with Type-C connectors at both ends) no longer physically differentiates plugs on a cable by being an A-type or B-type (that is, the Type-C cable is reversible), the USB interface still maintains such a host-to-device logical relationship. Determination of this host-to-device relationship is accomplished through a Configuration Channel (CC) that is implemented within the cable. Using the CC, the USB Type-C interconnect defines a simplified 5-V VBUS-based power delivery and charging solution that supplements what is already defined in the USB 3.1 specification. For more details, see Type-C specification.

In addition, the CC is used for USB-PD (USB Power Delivery, discussed in the Introduction to PD section) communication to set up and manage advanced power delivery features and Alternate/Accessory Modes. The USB-PD messages are delivered across the dedicated Type-C CC using the Bi-Phase Marked Coding (BMC) method. For more details, see the USB PD specification.

Figure 1. USB Type-C: Connector of the Future



Type-C Receptacle/ Plug Interface

Figure 2 and Figure 3 show the USB Type-C Receptacle and Plug signals. Table 1 and Table 2 summarize the list of signals used on the USB Type-C receptacle and plug.

Table 1. USB Type-C Receptacle Signals

Signal Group	Signal	Description
USB 3.1	TX1p, TX1n RX1p, RX1n TX2p, TX2n RX2p,RX2n	The SuperSpeed USB serial data interface defines a differential transmit pair and a differential receive pair. On a USB Type-C receptacle, two sets of SuperSpeed USB signal pins are defined to enable the plug-flipping feature.
USB 2.0	Dp1, Dn1 Dp2, Dn2	The USB 2.0 serial data interface defines a differential pair. On a USB Type-C receptacle, two sets of USB 2.0 signal pins are defined to enable the plug-flipping feature.
Configuration Channel	CC1, CC2	The CC channel in the receptacle detects the signal orientation and channel configuration.
Auxiliary Signals	SBU1, SBU2	Sideband Use
Power	VBUS	USB cable bus power
	GND	USB cable return current path

Table 2. USB Type-C Plug Signals

Signal Group	Signal	Description
USB 3.1	TX1p, TX1n RX1p, RX1n TX2p, TX2n RX2p,RX2n	The SuperSpeed USB serial data interface defines a differential transmit pair and a differential receive pair. On a USB Type-C receptacle, two sets of SuperSpeed USB signal pins are defined to enable the plug-flipping feature.
USB 2.0	Dp, Dn	The USB 2.0 serial data interface defines a differential pair. On a USB Type-C receptacle, two set of USB 2.0 signal pins are defined to enable the plug-flipping feature.
Configuration Channel	CC	The CC in the plug used for connection detect and interface configuration
Auxiliary Signals	SBU1, SBU2	Sideband Use
Power	VBUS	USB cable bus power
	VCONN	Type-C cable plug power
	GND	USB cable return current path

As shown in Figure 2, the receptacle signal functionally delivers both USB 3.1 (TX and RX pairs) and USB 2.0 (D+ and D−) data buses, USB power (VBUS), ground (GND), Configuration Channel signals (CC1 and CC2), and two Sideband Use (SBU) signal pins. The two sets of USB data-bus signal locations in this layout facilitate functionally mapping of the USB signals independent of the plug orientation in the receptacle.

Figure 2. USB Type-C Receptacle Interface (Front View)

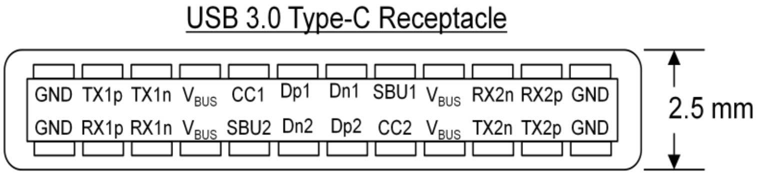
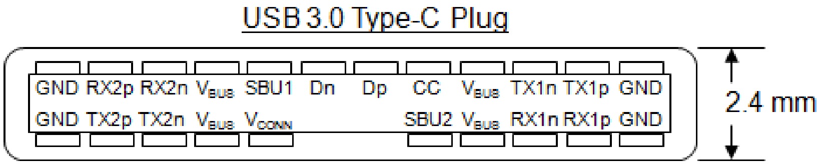

USB 3.0 Type-C Receptacle
GND TX1p TX1n VBUS CC1 Dp1 Dn1 SBU1 VBUS RX2n RX2p GND
GND RX1p RX1n VBUS SBU2 Dn2 Dp2 CC2 VBUS TX2n TX2p GND
2.5 mm

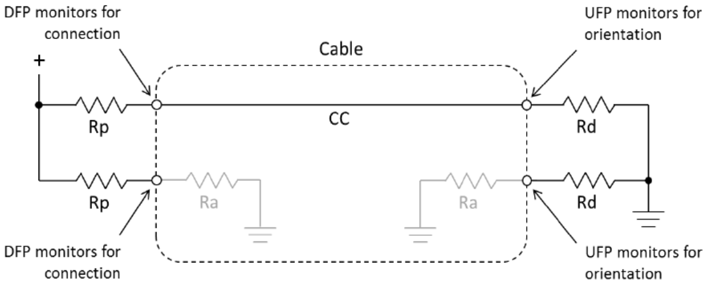
Figure 3 depicts the USB Type-C plug signals. Only one CC pin is connected through the cable to establish signal orientation; the other CC pin is repurposed as VCONN for powering the electronics in the USB Type-C plug.

Figure 3. USB Type-C Plug Interface (Front View)


USB 3.0 Type-C Plug
GND RX2p RX2n VBUS SBU1 Dn Dp CC VBUS TX1n TX1p GND
GND TX2p TX2n VBUS VCONN SBU2 VBUS RX1n RX1p GND
2.4 mm

A Type-C downstream-facing port (DFP) exposes R_p terminations on its CC pins (CC1 and CC2) while a Type-C upstream-facing port (UFP) exposes R_d terminations on its CC pins, as shown in Figure 4. DFPs, specifically associated with the flow of data in a USB connection, are typically the ports on a host, such as a PC or a hub, to which devices are connected. In its initial state, the DFP sources VBUS and VCONN. On the other hand, UFPs are the ports on a device or a hub that connects to a host. In its initial state, the UFP sinks VBUS.

Figure 4. Type-C Connection/ Orientation Detection[1]

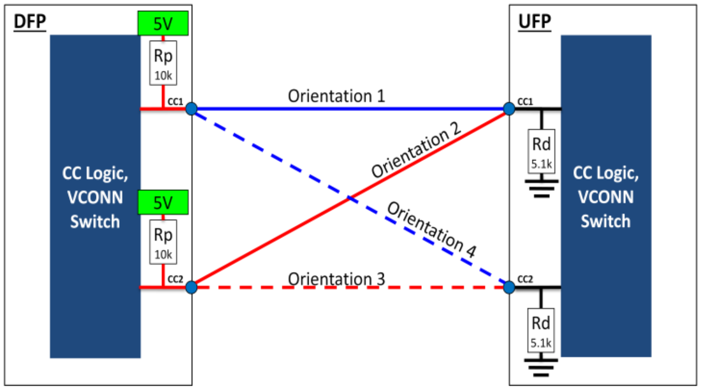

DFP monitors for connection
UFP monitors for orientation
Cable
+
Rp
CC
Rd
Rp
Ra
Ra
Rd
DFP monitors for connection
UFP monitors for orientation

These cables expose R_a terminations on the VCONN pin, as shown in Figure 4. R_p and R_d terminations on CC pins detect the connection event and identify the cable orientation. The DFP monitors both CC pins for a voltage lower than its unterminated voltage to detect the connection event.

By being able to detect which of the CC pins (CC1 or CC2) at the Type-C receptacle is terminated by R_d at the UFP, the DFP can determine one among the four possible cable orientations, as shown in Figure 5 and Table 3. The DFP can use this to control the functional switch (MUX) for routing the SuperSpeed USB signal pairs appropriately irrespective of the cable orientation.

Similarly, the UFP can control the functional switch to route its SuperSpeed USB signal pairs appropriately, as shown in Figure 6. After a connection and orientation is established, the DFP will repurpose CC1 or CC2 to provide cable power over the VCONN pin of the plug. See Type-C Specification for more details on the Type-C connection and orientation detection mechanism.

Figure 5. CC Pins Determine Plug Orientation



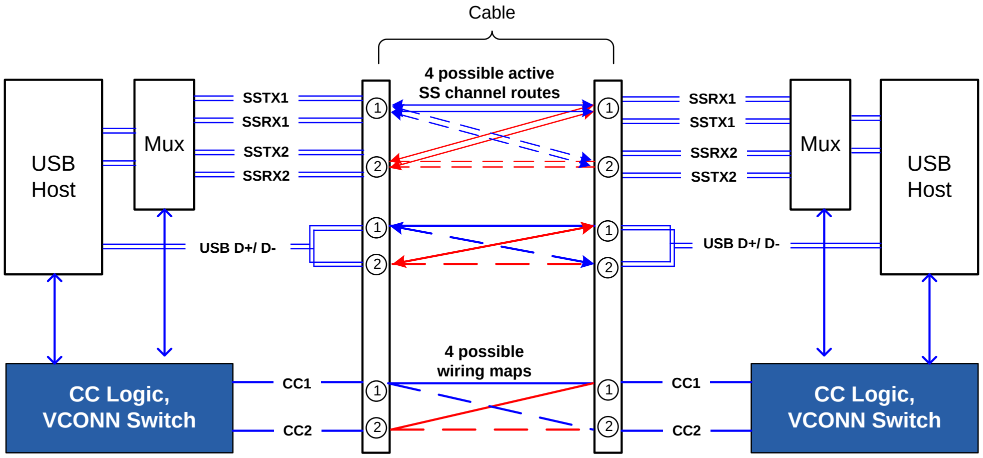
[1] Image source: USB Type-C Specification

Table 3. Cable Orientations

Wiring Map Orientation	Plug #1 Configuration	Plug #2 Configuration	Cable Configuration	Diagram Line Style
Orientation 1	Upside Up (1)*	Upside Up (1)	Un-flipped, Straight Through	Blue, Solid ━━━
Orientation 2	Upside Up (1)*	Upside Down (2)	Un-flipped, Twisted Through	Red, Solid ━━━
Orientation 3	Upside Down (2)*	Upside Down (2)	Flipped, Straight Through	Red, Dotted - - - -
Orientation 4	Upside Down (2)*	Upside Up (1)	Flipped, Twisted Through	Blue, Dotted - - - -

* For (1) and (2), see Figure 6.

Figure 6. Cable Flip/Twist



For USB 2.0, only one set of USB 2.0 signals (D+ and D-) is implemented in a USB Type-C cable. SBU signals are used in the Alternate Mode supported by the Type-C specification, which enables multi-purposing of Type-C signals for alternate uses such as DisplayPort. For example, in a DisplayPort application, USB 3.0 lines are used for video transmission and the SBU lines for audio transmission. See Appendix A, which shows a possible application of alternate mode in an Accessory cable. See Type-C Specification for more on Alternate modes.

Possible Type-C Cable Assemblies

The following USB Type-C cables are defined by Type-C specification:

- USB Full-Featured Type-C cable, a USB Type-C to Type-C cable that supports USB 2.0 and USB 3.1 data operation. This cable also includes SBU wires.
- USB 2.0 Type-C cable with a USB 2.0 Type-C plug at both ends for USB 2.0 applications
- Captive cable with either a USB Full-Featured Type-C plug or USB 2.0 Type-C plug at one end

Table 4 lists various Type-C standard cable assemblies in the perspective of electronic marking of cables.

Table 4. USB Type-C Standard Cable Assemblies

Plug 1	Plug 2	USB Version	Current Rating	USB PD (BMC)	Electronic Marking
Type-C	Type-C	USB 2.0	3 A	Supported	Optional
			5 A		Required
Type-C	Type-C	USB 3.1	3 A	Supported	Required
			5 A		Required

As listed in Table 4, all cables that are either full-featured or are rated at more than 3 A current need to be electronically marked cables. Electronic marking is optional for USB 2.0 cables rated at 3 A, while it is needed for USB 3.1 cables rated at 3 A.

Various USB Type-C to USB legacy cable and adapter assemblies are also defined by the Type-C specification such as the following:

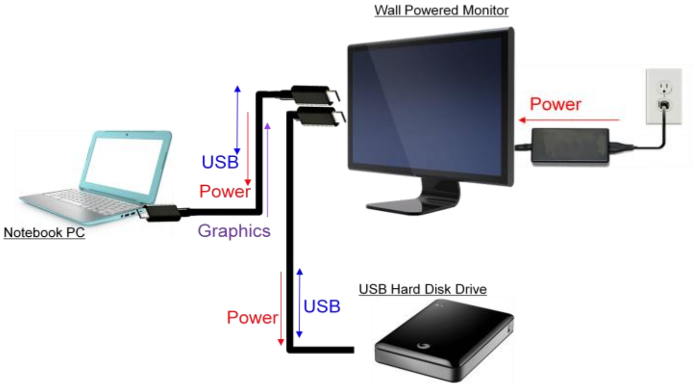
- USB 3.1/USB 2.0 Type-C (Type-C plug) to Legacy Host cable (Standard-A plug)
- USB 3.1/USB 2.0 Type-C (Type-C plug) to Legacy Device cable (Standard-B plug)
- USB 3.1/USB 2.0 Type-C (Type-C plug) to Legacy Micro/ Mini Device cable (Micro/ Mini-B plug)
- USB 3.1 Type-C (Type-C plug) to Legacy Standard-A adapter (Standard-A receptacle)
- USB 2.0 Type-C (Type-C plug) to Legacy Micro-B adapter (Micro-B receptacle)

See Type-C specification for more details.

USB Power Delivery

USB Power Delivery (PD) is a new USB standard that increases the power delivery over VBUS from 7.5 W to 100 W (with voltage/ current as high as 20V/ 5A). With USB Power Delivery**,** the power direction is no longer fixed: Both hosts and devices can act as either a Provider (a Type-C port that sources power over VBUS) or a Consumer (a Type-C port that sinks power from VBUS) of power. For example, a monitor can be powered from the wall charger and, in turn, can power a notebook and a hard disk drive, as shown in Figure 7.

Figure 7. USB-PD: One Power Adapter for the Desk



The USB PD specification provides more flexible power delivery with data over a single cable without the need for a device driver. It has the potential to minimize waste as it becomes a standard for charging devices that are not satisfied by the Battery charging specification, version 1.2. This standardization was driven as a means to increase charger reuse and, thus, reduce electronic waste.

This specification, in addition to providing mechanisms to negotiate power, can be used for standard- and vendor-defined messaging needed for custom cable applications. It also enables the discovery of cable capabilities such as supported speeds and current levels.

USB PD Specifications, Revision 1.0, covered a power negotiation method over VBUS line of the USB bus (using BFSK modulation over VBUS). USB PD Specifications, Revision 2.0, recommends the method of using power delivery protocol messages over the CC.

The USB Power Delivery specification is guided by the following principles:

1. Works seamlessly with legacy USB devices
2. Compatible with existing compliant USB cables
3. Minimizes potential damage from non-compliant cables
4. Optimized for low-cost implementations

See the USB PD specification for more details.

SOP* Communication in USB PD

A Power Delivery communication starts with sequences of special symbols (called K-code marker) to delineate the start of a packet. K-codes are special symbols provided by the 4b5b line-encoding scheme used in PD communication to delineate packet boundaries.

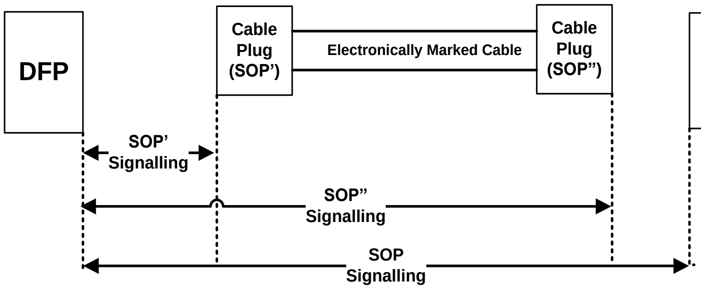
In addition to encoding data, K-codes are used for special control functions, such as a hard reset and cable reset. The special K-code sequence signifying a start of sequence is called “Start Of Packet” (SOP). Three sequences are defined: SOP, SOP’, and SOP’’. SOP* is used to refer all the three SOP sequences. Figure 8 defines and differentiates SOP* packets:

- **SOP Packet:** Any Power Delivery packet that starts with an SOP sequence. The communication between Port Partners (DFP and UFP) uses SOP packets. These packets are not recognized by either Cable Plug.
- **SOP’ Packet:** Any Power Delivery packet that starts with an SOP’ sequence used to communicate with a Cable Plug. SOP’ packets are recognized by the electronics in the Cable Plug attached to the DFP (cable plug marked SOP’ in Figure 8) and are not recognized by the other Cable Plug or the port partner in UFP.
- **SOP’’ Packet:** Any Power Delivery packet that starts with an SOP’’ sequence used to communicate with a Cable Plug when SOP’ packets are being used to communicate with the other Cable Plug. SOP’’ packets are recognized only by the electronics in the Cable Plug attached to the UFP (cable plug marked SOP” in Figure 8) and are not recognized by the other Cable Plug attached to the DFP or the port partner in UFP.

Notes: The term “Cable Plug” in the SOP’/SOP’’ communication case is used to represent a logical entity in the cable that is capable of PD communication. These entities may or may not be physically located in the plug.

Response to SOP” packets by the cable plug attached to UFP is optional, but the response to SOP’ packets by the cable plug attached to DFP is mandatory in an EMCA.

Figure 8. SOP* Communication



SOP* communication takes place over a single wire (CC). This means that SOP* communication periods must be coordinated to prevent important communication from being blocked. Communications between the Port Partners (SOP packets) take precedence, implying that communications with the Cable Plug (SOP’/ SOP” packets) can be interrupted. See USB PD specification for more details.

CCG2 Overview

EZ-PD CCG2 is a USB Type-C cable controller that complies with the latest USB Type-C and PD standards. Salient features of CCG2 are:

- Uses industry-standard 32-bit, 48-MHz ARM®Cortex®-M0 processor and 32 KB flash
- Integrates a single Type-C Transceiver, termination resistors[2] (R_P, R_D, and R_A shown in Figure 4) and system-level ESD (8-kV contact, 15-kV air)
- Available in 20-ball, 1.63 mm × 2.03 mm WLCSP and 14-pin, 2.5 mm × 3.5 mm × 0.6 mm DFN packages
- Provides a complete one-chip hardware and firmware solution for a USB Type-C EMCA
- Capable of operating from three power sources: VCONN1, VCONN2, and VDDD
- Two independent VCONN rails with integrated isolation between the two

[2] See Type-C Specification for more details on termination resistors.

CCG2 provides a complete USB Type-C and USB Power Delivery port control solution for passive cables, active cables, and powered accessories. Table 5 shows the available CCG2 parts for various applications.

Table 5. EZ-PD^{TM} CCG2 Product options

Features	CYPD2103	CYPD2104	CYPD2105
Application	Passive Cable	Accessory[3]	Active Cable[4]
Package	20-ball WLCSP, 14-pin DFN	20-ball WLCSP	20-ball WLCSP

[3] An upstream-facing port (UFP) with the form factor of a cable or dongle.

[4] Electronically Marked Cable Assembly with a re-driver to condition USB data signals.

See CCG2 datasheet for more details.

EMCA Applications

All USB full-featured Type-C cables are electronically marked. Such cables incorporate electronics that require VCONN as the cable power. This provides a method to determine the characteristics of the cable, such as its current-carrying capability, performance, and vendor identification (USB Type-C Cable ID function).

Characteristics of the cable include:

- Product type, such as Passive cable, Active cable, and Alternate mode adapter
- Cable hardware and firmware version assigned by the vendor
- The other end of Type-C cable: indicating if the cable is a Type-C to Type-C or a Type-C to legacy cable or a Type-C to receptacle, and so on
- Cable latency: indicates the latency of cables
- Current-handling capability (1.5 A, 3 A or 5 A)
- SOP” controller present or not: indicating if the cable is enabled for SOP” response by the electronics at the UFP end of the cable
- USB SuperSpeed signaling support: indicating the speed supported by the USB data signals in the cable (USB 2.0 or USB 3.1, Gen 1 and Gen 2)

See the USB PD specification.

An EMCA can be any one of the following types:

- Passive EMCA: An EMCA that does not modify the USB data signals
- Active EMCA: An EMCA with additional electronics to condition the USB data signals, such as drivers, to allow for implementing longer cables
- Accessory: A UFP with the form factor of a cable or dongle

EMCA cables can be implemented with or without the VCONN wire extending through the cable. EMCA cables with the VCONN wire extending through the cable (Refer section Passive EMCA with One CCG2 per Cable for more details) need isolation elements. These isolation elements prevent VCONN from traversing end-to-end through the cable. In EMCA cables in which the VCONN wire does not extend through the cable (Refer section Passive EMCA with One CCG2 per Cable Plug (Two Chips per Cable) - SOP’ Response Only for more details), an SOP’ element is required at each end of the cable. In this case, no isolation elements are needed.

Some of the key application-level requirements for EMCA applications are as follows:

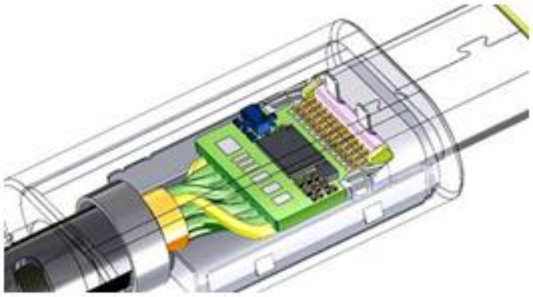
- Support USB-PD protocol as defined in the PD 2.0 specification
- Support SOP’ and SOP” as defined in the PD 2.0 specification
- Support an integrated R_a resistor on the VCONN
- The ability to power the chip from VCONN

- Integrated isolation between the two VCONN pins (VCONN1 and VCONN2)
- Ability to disconnect the R_a resistor to conserve power
- Integrated system-level ESD protection on CC and VCONN pins
- Integrated bootloader to support a firmware update over CC
- Cable authentication using a secure external EEPROM

CCG2 in Passive EMCA Applications

EZ-PD CCG2 is targeted at electronically marked cable assemblies. This section describes two representative applications of CCG2 in passive EMCAs. (Additional applications are described in the Appendix). In each application, CCG2 and associated circuits are assembled into one or both ends of a cable called the “plug” (see Figure 9). Inside each plug housing or mold, the chips are assembled on PCBs called “paddle boards.” Each EMCA includes at least one plug with a CCG2, which responds to identification commands from the USB Host over the CC.

Figure 9. USB Type-C Plug Housing



The two examples in this section show the application of CCG2 in passive EMCAs. Before discussing CCG2 in passive EMCA applications, the CCG2 power subsystem is discussed to promote a better understanding of the VCONN signal handling.

CCG2 Power Subsystem

CCG2 can operate from one of the three power sources: VCONN1, VCONN2, or VDDD.

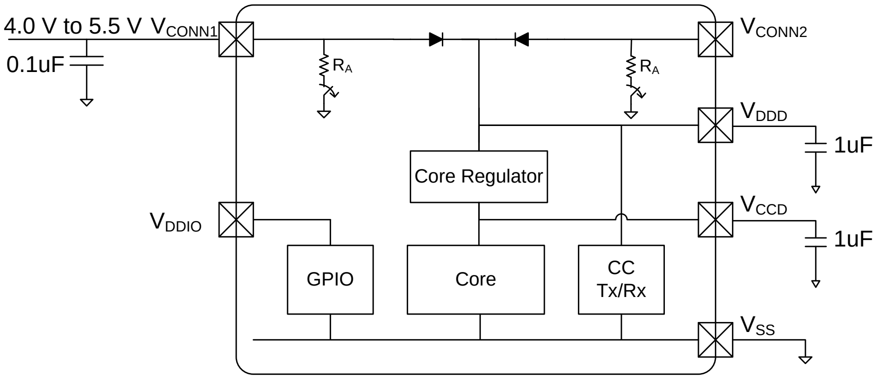
VCONN1 and VCONN2 pins can be used as connections to the VCONN pins in a USB Type-C cable system. Each of these inputs supports operation from 4.0 V to 5.5 V. An internal isolation between VCONN1 and VCONN2 pins is provided, allowing them to be at different levels simultaneously, as shown in Figure 10. These internal diodes act as the isolation elements needed to implement an EMCA solution with the VCONN wire extending through the cable.

CCG2 can also operate from 2.7 V to 5.5 V when operated from the VDDD supply pin. In this mode of operation, VCONN1 and VCONN2 must not be connected in the system. In applications in which the VCONN pins are used as supply sources, the VDDD pin can be used as an output voltage.

The internal GPIO buffers of CCG2 are powered from VDDIO rail. Typically, this rail will be shorted to VDDD in cable applications.

For more details on Power subsystem, see the CCG2 datasheet.

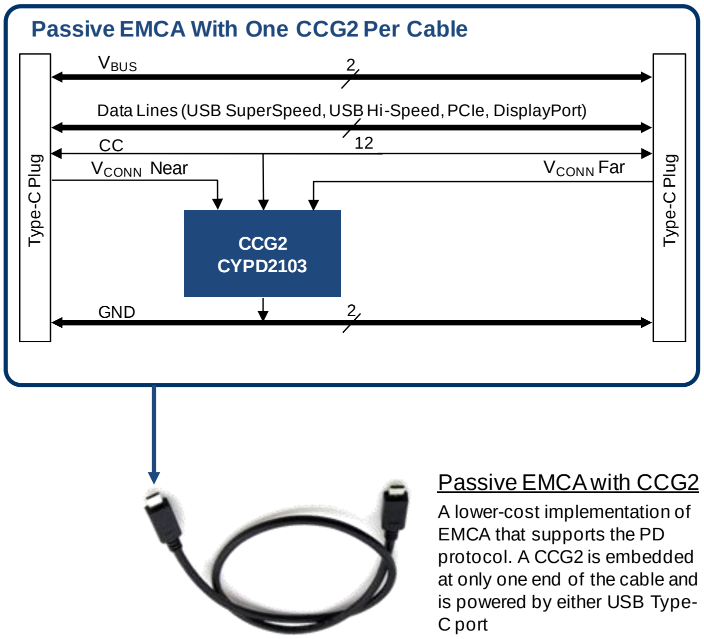
Figure 10. CCG2 Power and Bypass Scheme



Passive EMCA with One CCG2 per Cable

This EMCA solution contains a CCG2 chip in only one of its plugs. This solution requires the single VCONN wire to run through the cable, so that the chip can be powered irrespective of which plug is connected to the host (DFP).

Figure 11. Passive EMCA Solution with One CCG2 per Cable


Passive EMCA With One CCG2 Per Cable
V_{BUS}
2
Data Lines (USB SuperSpeed, USB Hi-Speed, PCIe, DisplayPort)
12
CC
V_{CONN} Near
V_{CONN} Far
Type-C Plug
CCG2
CYPD2103
GND
2
Type-C Plug

Passive EMCA with CCG2

A lower-cost implementation of EMCA that supports the PD protocol. A CCG2 is embedded at only one end of the cable and is powered by either USB Type-C port

In this case, the CCG2 will respond only to SOP’ packets. After the cable is enumerated, the host may shut down the VCONN supply. One of the key and unique requirements for this application is the ability to power the chip from two separate VCONN pins.

Pros:

This solution needs only one CCG2 chip at one of the cable ends.

Cons:

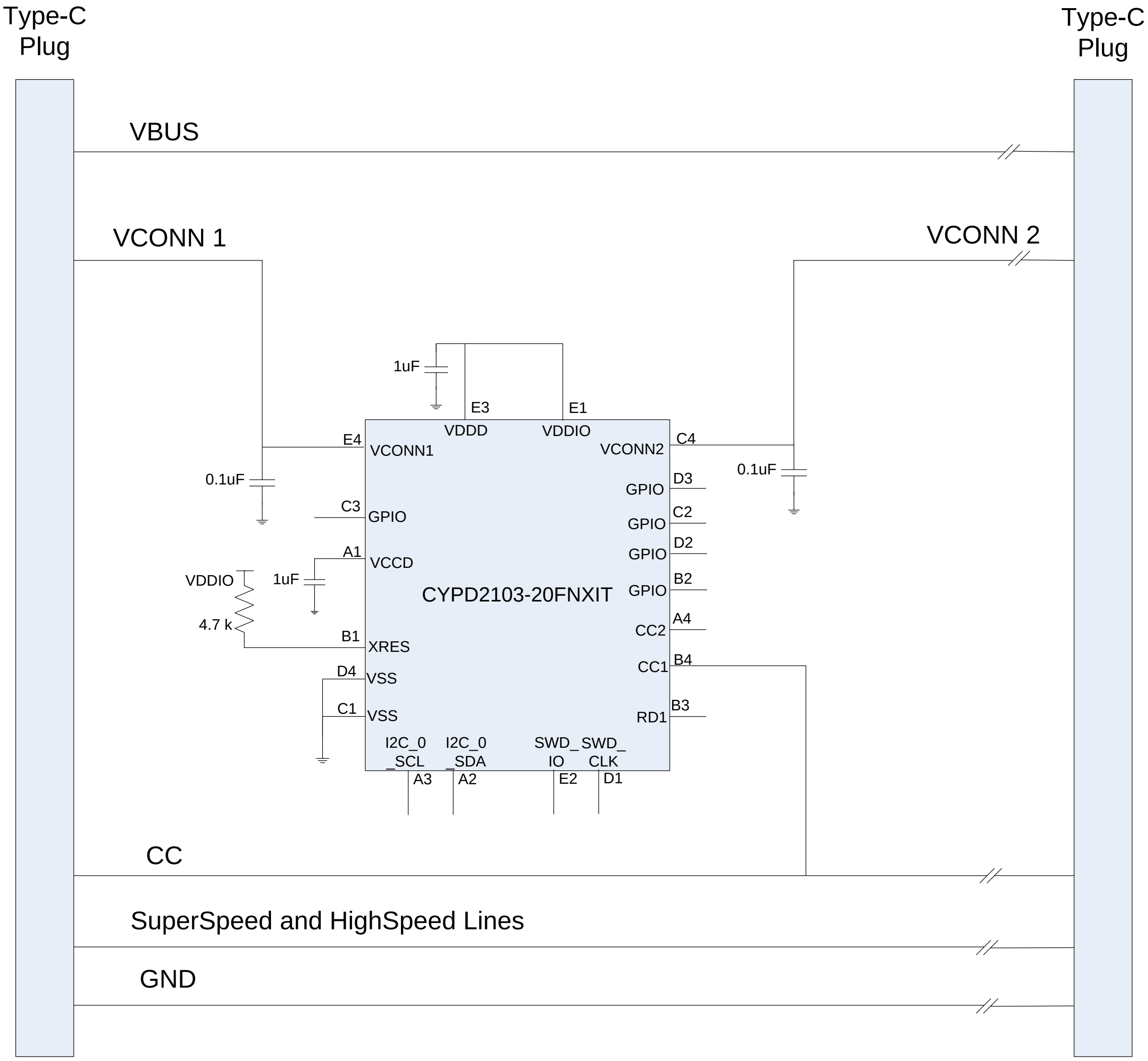
This solution increases the wire cost (needed for the VCONN signal to run through the cable).

Figure 11 shows the block diagram for the Passive EMCA solution with one CCG2 per cable, while Figure 12 shows the schematic. As shown in Figure 12, VCONN from one end of the cable is to be connected to VCONN1 of the CCG2; VCONN from the other end of the cable is to be connected to VCONN2 of CCG2.

For the single-chip solution, GPIO (ball D3 of the WLCSP package/pin 13 of the DFN package) must be left floating to disable the SOP” response. When this GPIO is left floating, the CYPD2103 is configured to always respond to SOP’ packets only. The response to SOP’/ SOP” packets is not dynamically determined based on the connection to UFP or DFP.

This will be used in the case of a one-chip solution (cable with one CCG2 chip for the whole cable) in which the chip may get powered either from VCONN1 or VCONN2, based on the cable orientation. Irrespective of the end of the cable, CCG2 will always respond to SOP’ packets only irrespective of whether it is powered from VCONN1 or VCONN2.

Figure 12. CCG2 Single-Chip EMCA Schematic



Notes:

Figure 12 is for representative purposes only and is based on a WLCSP package. For DFN reference, see Appendix.

Passive EMCA with One CCG2 per Cable Plug (Two Chips per Cable) - SOP’ Response Only

This EMCA solution contains two CCG2 chips – one in each plug. In this solution, the VCONN signal does not run through the cable, but stops at the CCG2 device in each plug. In this case, only one CCG2 is powered at a time, depending on which is nearer to the DFP that supplies VCONN. The powered CCG2 responds to SOP’ packets only. After the cable is enumerated, the host may shut down the VCONN supply.

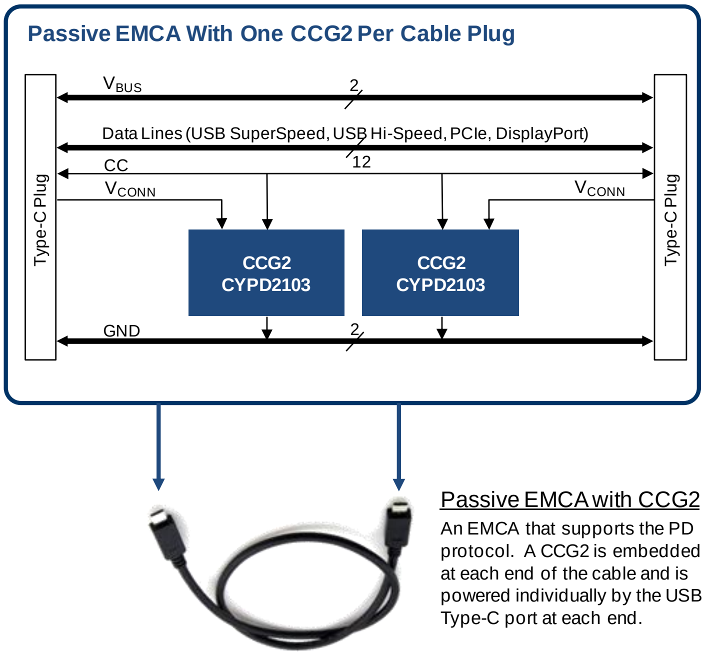
Pros:

This solution saves wire cost (needed for VCONN signal if it runs across the device).

Cons:

This solution needs two CCG2 chips, one at each cable end.

Figure 13. Passive EMCA Solution with One CCG2 per Cable Plug


Passive EMCA With One CCG2 Per Cable Plug
V_BUS
2
Data Lines (USB SuperSpeed, USB Hi-Speed, PCIe, DisplayPort)
12
CC
V_CONN
V_CONN
Type-C Plug
Type-C Plug
CCG2
CYPD2103
CCG2
CYPD2103
GND
2

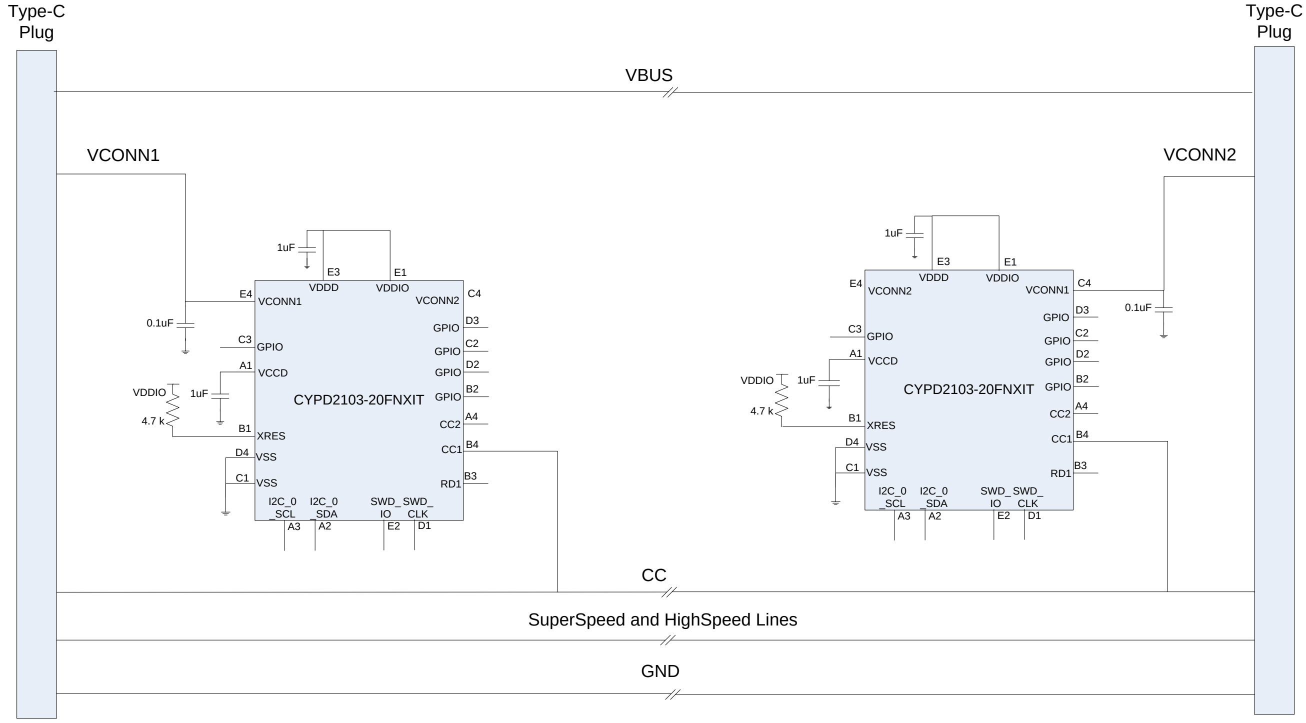
Passive EMCA with CCG2

An EMCA that supports the PD protocol. A CCG2 is embedded at each end of the cable and is powered individually by the USB Type-C port at each end.

Figure 13 depicts the block diagram and Figure 14 shows the schematic for the Passive EMCA solution with two CCG2 chips per cable, with one getting powered. As shown in Figure 14, VCONN from either end of the cable is connected to VCONN1 of CCG2 at the respective end. GPIO (ball D3 of the WLCSP package/ pin 13 of the DFN package) can be left floating or pulled LOW.

Whichever way the cable gets connected, one CCG2 always gets powered through VCONN1, and it will respond to SOP’ packets only.

Figure 14. CCG2 Two-Chip EMCA Schematic (One Chip Powered)



Notes:

Figure 14 is for representative purposes only and is based on WLCSP package. For DFN reference, see Appendix.

Design Guidelines

Hardware Guidelines

Figure 12 and Figure 14 show the hardware schematics for the one-chip and two-chip solutions of EMCA using CCG2. As shown in Figure 12, CCG2 can be a one-chip USB Type-C cable solution with an integrated Type-C transceiver, termination resistors and system-level ESD-protection circuits. This solution requires only four external decoupling capacitors* and a resistor to operate.

It is recommended to have provisions for probes or jumpers on SWD lines to help facilitate debug and programming at the paddle-card level. CCG2 also allows the firmware of the assembled cable to be upgraded over the CC line through the built-in bootloader.

See AN95599 – Hardware Design Guidelines for EZ-PD^{TM} CCG2 for more details on hardware design.

Notes:

* In addition, four capacitors are needed on VBUS pins of Type-C connector (10-nF bypass capacitor per VBUS pin on the Type-C connector), according to the Type-C Specification. See Type-C Specification for more details.

Firmware Upgrade Guidelines

EZ-PD CCG2 is available in three pre-programmed versions to suit various design needs, as listed in Table 5. A vendor-specific cable application will need the vendor command implementation that determines the product capabilities and functionality. Contact Cypress for details about the firmware and a custom application.

The CCG2 firmware has the following basic capabilities:

1. The firmware detects the cable plug location – whether it is attached to the host port (downstream port) or the device port (upstream port).

2. The firmware responds to all structured VDMs (Vendor Defined Messages) with SOP' (if the cable plug-end is near to the host-end) and SOP'' (if the plug-end is near to the device-end).
3. The firmware responds to all Cypress-defined unstructured VDMs with SOP (for accessory mode application), SOP' (if the cable plug-end is near to the host-end) and SOP'' (if the plug-end is near to the device-end). These messages are used for firmware upgrades.
4. The firmware responds to a set of unstructured VDMs as reference. The VDM handlers must indicate the functionality by toggling a set of GPIOs.
5. The firmware includes a bootloader that is capable of upgrading the PD firmware on the device. The bootloader receives a firmware image over Cypress-defined unstructured VDMs.
6. The device stays in Deep-Sleep mode when the CC lines are in the Idle state and wakes up to respond to PD messages. It returns to the Deep-Sleep mode as soon as the PD bus is detected as idle.
7. The bootloader supports two copies of firmware (for CYPD2103 and CYPD2105). It verifies the validity of the firmware image based on the checksum before passing control to the firmware.
8. The SOP/ SOP" response is determined based on a GPIO (ball D3 of the WLCSP package/ pin 13 of the DFN package) status provided as an input to the CCG2 device along with VCONN1/ VCONN2 signals, as shown in Table 6.

Table 6. SOP Response from the CCG2 Controller

VCONN_1	VCONN_2	GPIO***	SOP'	SOP"
x**	x	Floating	✓	✗
4V – 5.5V	x	Pulled LOW	✓	✗
< 4 V	x	Pulled LOW	✗	✓

** x in Table 6 indicates Don't care

*** GPIO is ball D3 of the WLCSP package/pin 13 of the DFN package

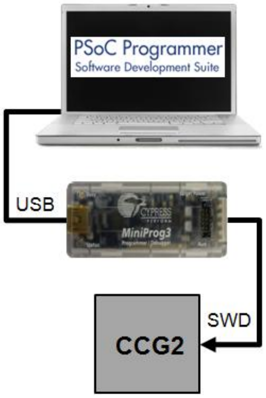
CCG2 can be upgraded to keep pace with USB-IF specification changes. Contact Cypress for firmware upgrades. The on-chip 32-KB flash can be programmed using the SWD interface or over the Type-C Configuration Channel (CC signal).

Upgrade CCG2 over SWD Using MiniProg3

A firmware upgrade using the SWD interface can be done using a PC running PSoC® Programmer™ software and a MiniProg3 programmer. This method of programming can be used on a paddle card equipped with the SWD pins or header and is typically used during product development.

For more details on the steps, see the knowledge base article Programming EZ-PD™ CCG2 Using PSoC® Programmer and MiniProg3.

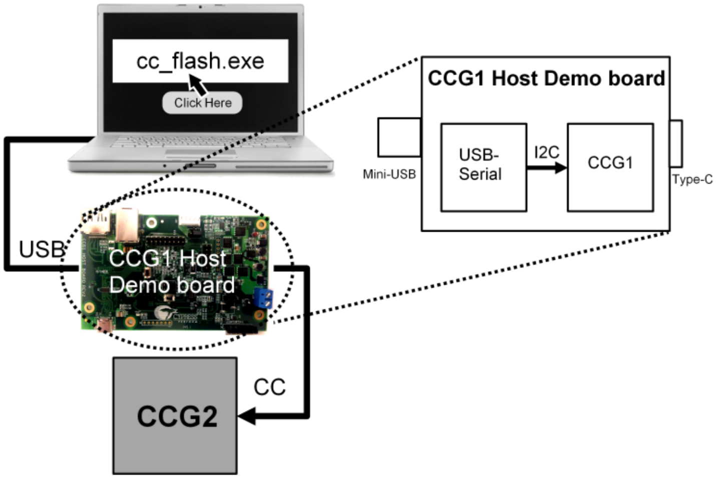
Figure 15. Upgrade over SWD (Using MiniProg3)



Upgrading CCG2 Firmware Over CC

A PC running a firmware upgrade application, *cc_flash.exe*, available at this webpage, can be used to program CCG2 in the USB Type-C cable directly over the CC line with the help of a CCG1 host demo board. This method can be deployed by cable manufacturers to provide upgrades to the end user or to program the firmware after the cable assembly is manufactured. CCG2 is factory-programmed with a bootloader to be first-time programmable in the field through the Type-C interface.

Figure 16. Upgrade Using CC



The USB-Serial Bridge in the CCG1 host demo board acts as an USB-I^2C Master bridge. The USB-Serial bridge communicates with the CCG1 Type-C Host Controller's I^2C slave interface. CCG1 is a USB Type-C port controller with PD that complies with the latest USB Type-C and PD standards, which provides a complete solution to add USB Type-C and PD support to power adapters, notebooks, tablets, monitors, and EMCA with up to 100 W of power delivery. For more details on CCG1, see the CCG1 webpage.

The Cypress-provided configuration utility, *cc_flash.exe*, sends I^2C commands to CCG1 Type-C Host Controller via a USB-Serial Bridge.

The CCG1 Type-C host controller is configured to communicate with the EMCA over the Type-C Interface. The CCG1 firmware detects the presence of an EMCA and applies VCONN. It tunnels I^2C commands received via the USB-Serial Bridge to the EMCA in the form of structured and unstructured VDMs.

For more details on the steps, see the knowledge base article Upgrading the Firmware of EZ-PD™ CCG2 Devices Over CC.

Summary

This application note discussed various aspects related to the design of passive USB 3.1 Type-C cable using CCG2. Customers can bring cost-optimized Type-C-compliant EMCAs to market faster with CCG2, an ARM Cortex-M0-based cable controller with 32 KB flash, requiring minimal external passive components or analog circuits.

Appendix A: Other CCG2 Applications

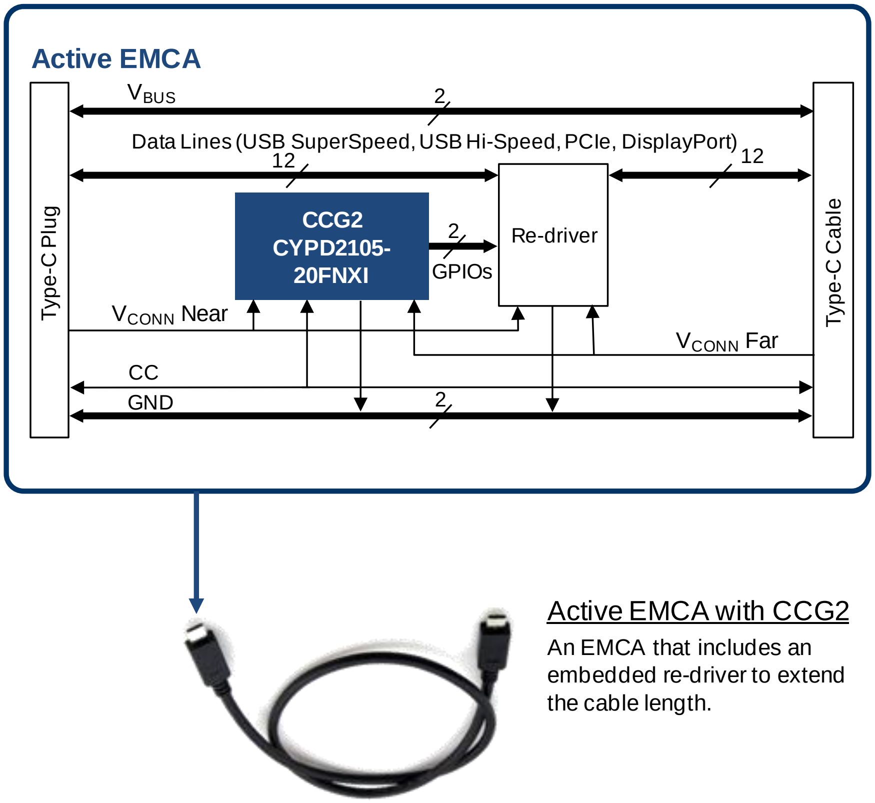
This section discusses possible CCG2 applications besides passive EMCA cables. Key application requirements are the same as those listed in the EMCA applications section.

Active EMCA Solution with One CCG2 per Cable – SOP’ Response Only

The main function of an active EMCA is to provide signal conditioning by adding a re-driver on the data path. Active cables that require configuration/signal conditioning (referred to as managed active cables) use USB Power Delivery Structured vendor defined messages (VDMs –SOP’ packets) to discover and configure the cable. Some managed active cables only have a single USB PD controller in the cable that responds to USB PD Structured VDMs (SOP’ packets only).

A Type-C active EMCA solution requires a CCG2 for electronic marking in addition to the signal-conditioning device (Re-driver in Figure 17). This solution draws power from the VCONN input to the cable. Such a cable advertises itself as an active cable by placing the R_a resistor on the VCONN line and an R_d resistor on the CC line. Unlike a passive EMCA, the host providing the power on VCONN does not shut down power.

Figure 17. Active EMCA Solution with One CCG2 per Cable



Active EMCA with CCG2

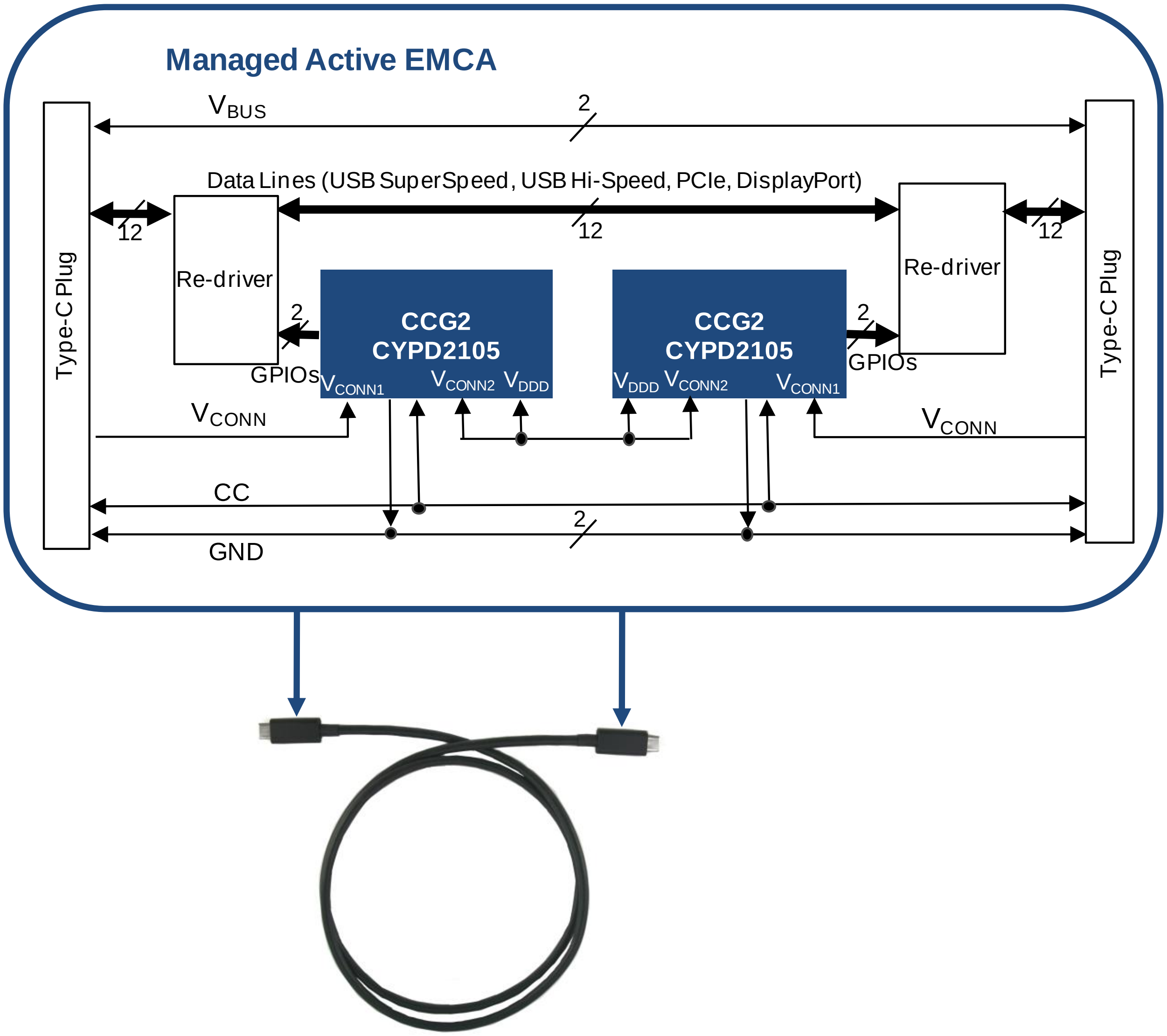
An EMCA that includes an embedded re-driver to extend the cable length.

Contact Cypress for the firmware solution for this application.

Managed Active EMCA Solution with One CCG2 per Cable Plug - SOP’ and SOP” Responses

When a managed active cable requires independent management or signal conditioning at each end of the cable, separate USB PD controllers responding to USB PD structured VDMs (SOP’ and SOP” packets) must be located in each plug.

Figure 18. Managed Active Cable with One CCG2 per Cable Plug



This active cable solution contains two CCG2 chips – one on each plug. VCONN is wired across the cable (but not straight through to the other end of the plug). In this solution, the wiring is done such that the cable is reversible. The host providing power on VCONN does not shut down power.

The VCONN signal of the cable plug at each end is connected to the VCONN1 input of the corresponding CCG2; another single conductor through the cable connects VCONN2 and VDDD pins of the CCG2 chip at both ends. When CCG2 is powered from any of the VCONN pins, VDDD acts as output voltage. In this case, both CCG2 chips will get powered: one through VCONN1 and the other through VCONN2. The internal firmware uses the GPIO (ball D3 of the WLCSP package/pin 13 of the DFN package) as a strap option to determine whether to respond to SOP”. SOP” response is enabled by strapping this GPIO to ground on the paddle card.

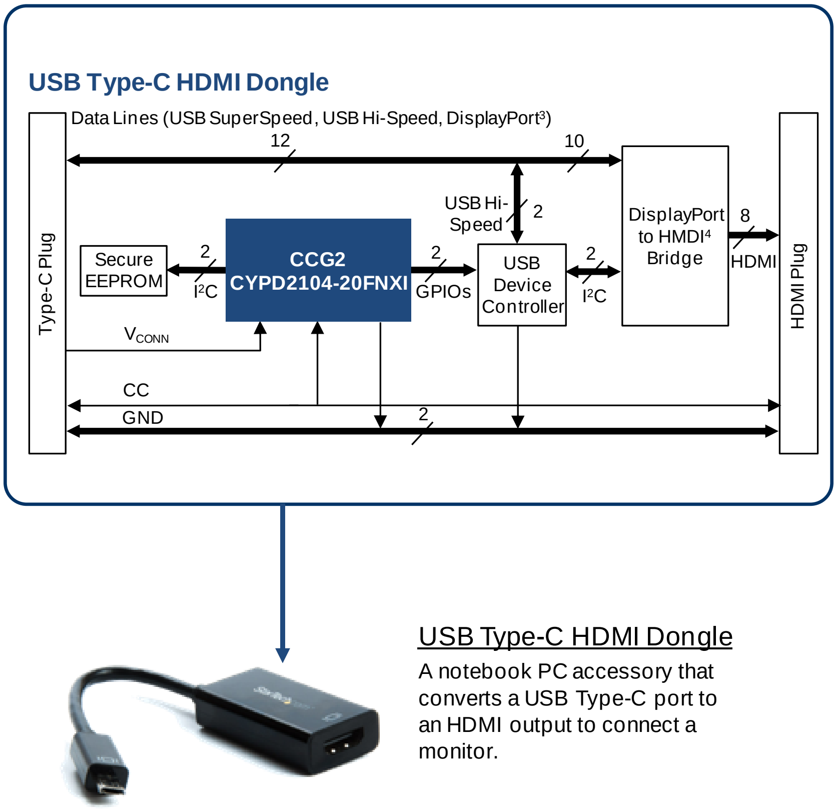
Regardless of which end of the cable is connected to the DFP, CCG2 chips of both ends are powered. CCG2 has detectors on both VCONN1 and VCONN2 that indicate to the firmware inside CCG2 as to which input is supplying power. The CCG2 with its VCONN1 powered is considered for SOP' response. CCG2 responds to SOP", if it is enabled for SOP" response by pulling GPIO (ball D3 of the WLCSP package/pin 13 of the DFN package) LOW and it is powered through VCONN2.

Figure 18 depicts the block diagram for the managed active cable with two CCG2 chips per cable, with both of them getting powered.

Accessory Solution

An accessory solution will require one CCG2 to implement USB Type-C and USB-PD protocols. A common form of an accessory is a converter dongle shown in this example. An example is a USB Type-C Thunderbolt adapter. Such an adapter will have a USB Type-C plug on one end and a set of PCIe and DisplayPort ports on the other. An accessory will also require an alternate-mode implementation. An accessory can be implemented as a standard UFP device capable of operating from VBUS and generating its own VCONN or a powered accessory operating from VCONN. Figure 19 shows one application of an accessory.

Figure 19. Accessory Solution



USB Type-C HDMI Dongle

A notebook PC accessory that converts a USB Type-C port to an HDMI output to connect a monitor.

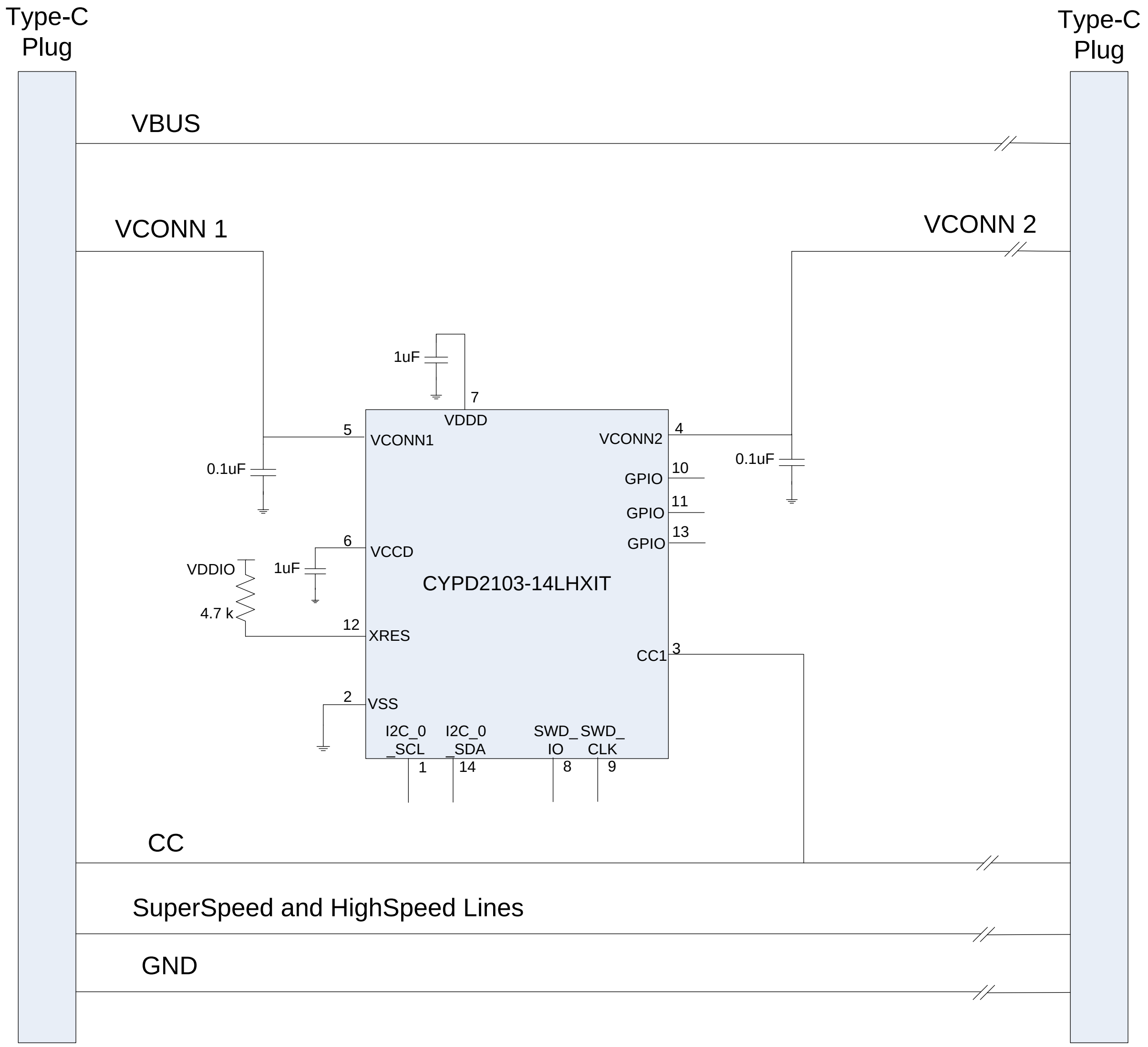
The key application-level requirements needed for this application are:

- Support integrated R_d resistors on CC pins per Type-C specification
- Ability to power from VBUS
- Support detection of current limit

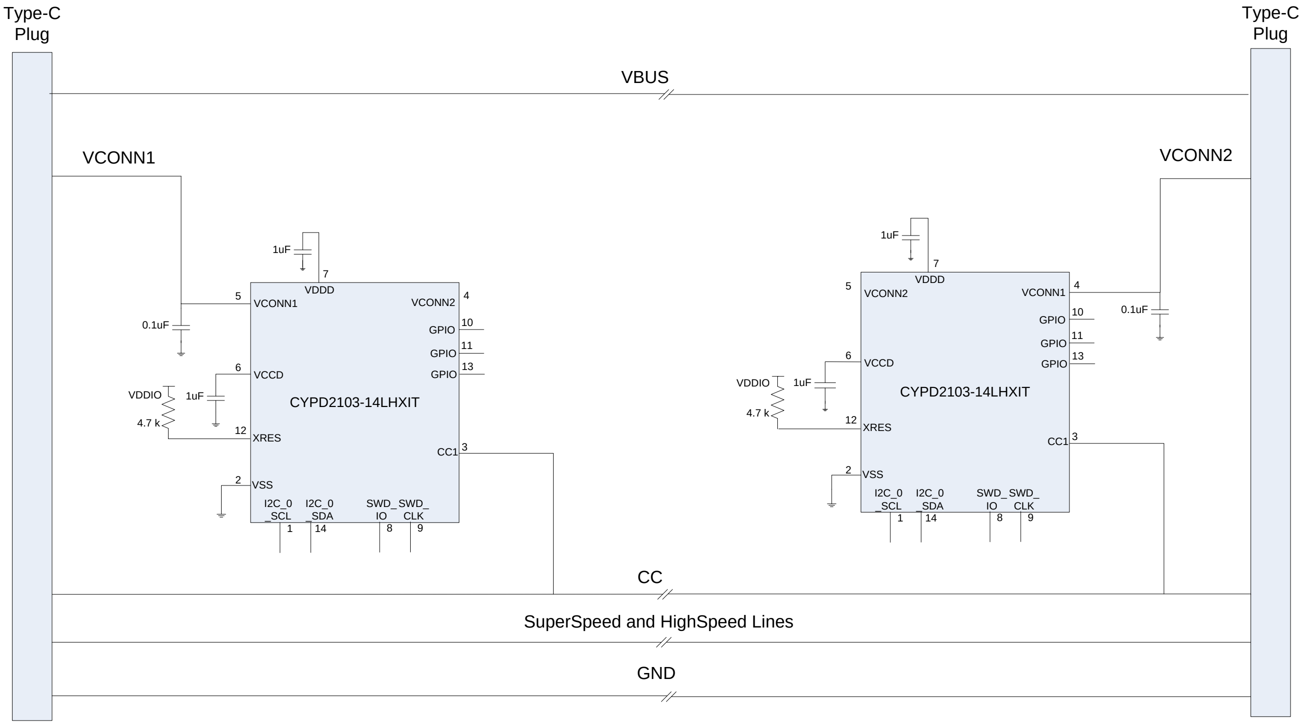
Contact Cypress for the firmware solution for this application.

Appendix B: Reference Schematics

CYPD2103-14LHXIT Single-Chip EMCA Schematic



CYPD2103-14LHXIT Dual-Chip EMCA Schematic



Document History

Document Title: AN95615 - Designing USB 3.1 Type-C Cables Using EZ-PD™ CCG2

Document Number: 001-95615

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	4710707	GAYA	04/03/2015	New Application Note
*A	4723949	GAYA	04/21/2015	Updated part numbers Updated links to Type-C Specifications

Worldwide Sales and Design Support

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

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

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

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

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

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

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