

EZ-PD™ PMG1 MCU prototyping kits guide

CY7110/CY7111/CY7112/CY7113

About this document

Scope and purpose

This document is a common user guide for the EZ-PD™ Power Delivery microcontroller Gen1 (PMG1) prototyping kits. These kits enable the development of USB PD sink applications using the EZ-PD™ PMG1 microcontrollers. You can see the relevant sections based on your requirement:

- **Introduction** – Provides basic information on the kits, the kit structure, and software information
- **EZ-PD™ PMG1 MCU software tools** – Explains ModusToolbox™ software and CYPRESS™ programmer
- **EZ-PD™ PMG1 prototyping kit system design** – Explains the kit architecture and system design details
- **Kit operation** – Describes the procedure to operate the kit for USB PD sink application
- **Application development on CY711x kits using ModusToolbox™ software** – Explains the code examples, procedure to build, program, and debug applications

Intended audience

The document is intended for users of EZ-PD™ PMG1 MCU prototyping kits (CY7110, CY7111, CY7112, and CY7113).

Table of contents

About this document.....	1
Table of contents.....	2
1 Introduction	4
1.1 Kit contents	5
1.2 Downloading kit documents and hardware design files	5
1.3 Board details	6
1.3.1 Kit features	8
1.3.2 Power selection jumper	9
1.3.3 Prototyping kit pinout.....	10
1.3.4 KitProg3 header	19
2 EZ-PD™ PMG1 MCU software tools.....	21
2.1 Before you begin	21
2.2 ModusToolbox™ software.....	21
2.2.1 Device configurator.....	22
2.2.2 EZ-PD™ configurator	22
2.2.3 Device firmware update (DFU) host tool.....	23
2.2.4 CAPSENSE™ configurator.....	24
2.2.5 CAPSENSE™ tuner	24
2.3 CYPRESS™ programmer (CYP)	25
3 EZ-PD™ PMG1 prototyping kit system design.....	26
3.1 Top-level hardware design	26
3.2 Power section.....	27
3.2.1 Power modes.....	27
3.2.1.1 Programming mode.....	27
3.2.1.2 Operational mode.....	27
3.2.2 3.3 V LDO and power LED.....	28
3.2.3 Overvoltage protection circuit	28
3.2.4 Power-cycle load switch (CY7110)	29
3.2.5 Transient voltage protection diode.....	30
3.3 EZ-PD™ PMG1 MCU subsystem	31
3.3.1 EZ-PD™ PMG1 MCU device	31
3.3.2 User LED and button	31
3.3.3 User button (SW2)	31
3.3.4 10-pin SWD header.....	32
3.3.5 MCU I/O header	32
3.3.6 Reset button	32
3.3.7 CAPSENSE™ buttons and slider	33
3.4 KitProg3 (PSoC™ 5LP MCU)	35
3.4.1 Mode button	36
3.4.2 KitProg3 power and status LEDs	36
3.5 Target voltage measurement	36
3.5.1 Power enable.....	36
3.6 Load switch.....	37
3.6.1 USB PD sink port	37
3.6.2 5-A load switch	37
3.6.2.1 P-FET load switch.....	37
3.6.2.2 N-FET load switch	38

Table of contents

3.6.3	DC_OUT terminal block	39
4	Kit operation	40
4.1	USB PD sink operation	40
5	Application development on CY711x kits using ModusToolbox™ software	43
5.1	Kit code examples	43
5.2	Using the kit code example.....	44
5.3	Programming and debugging using the KitProg3 interface	46
5.3.1	Programming using ModusToolbox™ software	46
5.3.2	Debugging using ModusToolbox™ software	49
5.3.3	Programming using CYPRESS™ programmer (CYP).....	51
5.4	Programming and Debugging using the 10-pin SWD interface	54
5.4.1	Programming using MiniProg4 from ModusToolbox™ software	54
5.4.2	Debug using MiniProg4 from ModusToolbox™ software	55
5.4.3	Programming using MiniProg4 from CYPRESS™ programmer (CYP)	56
5.5	Programming and debugging using J-Link debug probes	58
5.5.1	Programming and debugging the CY7110 kit	58
5.5.2	Programming and debugging the CY7111/CY7112/CY7113 kits.....	59
5.6	Configuring the PDOs and PD parameters	61
	Related documents	64
	Technical support.....	65
	Revision history.....	66

Introduction

1 Introduction

EZ-PD™ PMG1 is a family of high-voltage power delivery microcontrollers. These microcontrollers include Arm® Cortex®-M0/M0+ CPU, USB power delivery (PD) controller, and configurable analog and digital peripherals. EZ-PD™ PMG1 MCU is targeted for any embedded systems that provide/consume power to/from a high-voltage USB Type-C port and need a microcontroller to control the actions and features of a product.

Table 1 provides details of the kits and the associated microcontrollers.

Table 1 EZ-PD™ PMG1 prototyping kits

Kit name	PMG1 microcontroller type	EZ-PD™ PMG1 microcontroller MPN
CY7110 EZ-PD™ PMG1-S0 MCU prototyping kit	PMG1-S0	CYPM1011-24LQXI
CY7111 EZ-PD™ PMG1-S1 MCU prototyping kit	PMG1-S1	CYPM1111-40LQXI
CY7112 EZ-PD™ PMG1-S2 MCU prototyping kit	PMG1-S2	CYPM1211-40LQXI
CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit	PMG1-S3	CYPM1311-48LDXI

Visit the [webpage](#) or look at the device datasheets [\[4\]](#) to understand and compare the various features supported by the EZ-PD™ PMG1 MCU controllers.

The EZ-PD™ PMG1 MCU prototyping kits support the following key features:

- Support for single-port USB PD 3.0 sink role (UFP)
- USB bus-powered operation
- Compatible with the example projects implemented in ModusToolbox™ software
- Onboard KitProg3 programming and debugging module
- 100 W (20 V/5 A) power handling capability
- Standard breadboard compatible design
- Snap-away design to separate KitProg3 and EZ-PD™ PMG1 microcontroller board
- Onboard 3.3-V regulator
- Optional 10-pin standard SWD interface for MiniProg4 or third-party programmer/debugger module
- Firmware-controlled LED and user button
- Two self-capacitance-based CAPSENSE™ buttons and one 5-segment slider (only in CY7113 kit)
- I/O headers

The EZ-PD™ PMG1 MCU-based projects are developed using the ModusToolbox™ software development environment. ModusToolbox™ software is a set of multi-platform development tools and a comprehensive suite of GitHub-hosted firmware libraries. If you are new to ModusToolbox™ software, see the documentation on the [ModusToolbox™ software environment home page](#).

If you are new to EZ-PD™ PMG1 MCUs, see the application note [\[5\]](#) to learn about the EZ-PD™ PMG1 microcontroller family.

Introduction

1.1 Kit contents

CY711x prototyping kits contain the following:

- CY711x EZ-PD™ PMG1-Sx MCU prototyping kit board
- Quick start guide

Note: CY711x terminology is used to refer to all EZ-PD™ PMG1 MCU kits in this document. Explicit kit MPN will be used to highlight the kit-specific features wherever required.

Note: Each kit's board has a revision number (600-60xxx-01 Revxx) and the same is marked on the silkscreen as shown in [Figure 1](#).

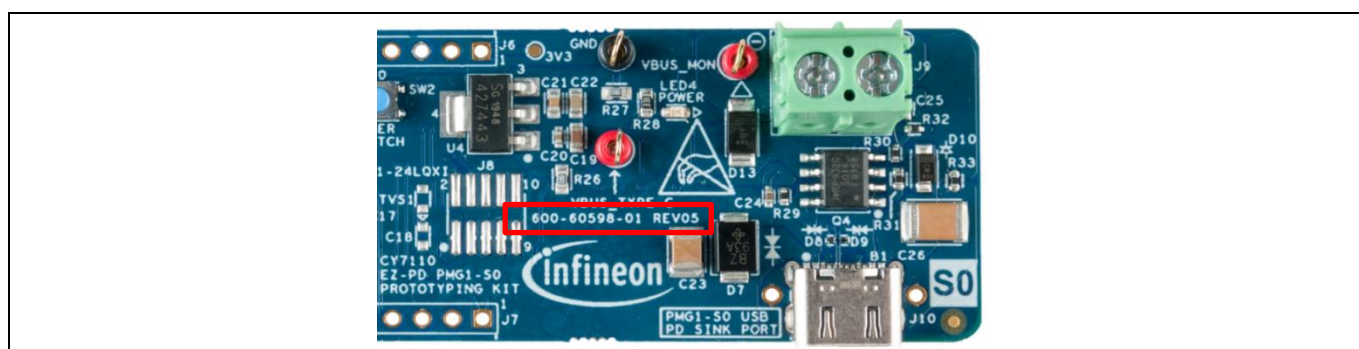


Figure 1 Identifying PMG1 MCU kit board revision number

Following items may be required for developing applications with USB PD sink capability using CY711x kits. Note that these are not included in the kit.

- USB PD 3.0-compliant USB Type-C power adapter (for example, Apple 30 W USB-C power adapter) for CY7110, CY7111, and CY7112 kits. USB PD 3.1-capable USB Type-C power adapter (for example, Apple 140 W USB-C power adapter) for CY7113 kit.
- USB PD 3.0-compliant Type-C cable compatible with the USB-C power adapter for CY7110, CY7111, and CY7112 kits. USB PD 3.1-capable Type-C cable for CY7113 kit.
- Through-hole connectors for I/O header and KitProg3 header (2.54-mm pitch)
- Jumper wires

1.2 Downloading kit documents and hardware design files

The documents and the hardware design files for each of the CY711x kits can be downloaded from the respective kit webpages [\[3\]](#). The documents include a quick start guide, kit guide (this document), and release notes. The hardware design files include a schematic, BOM, and layout files.

Introduction

1.3 Board details

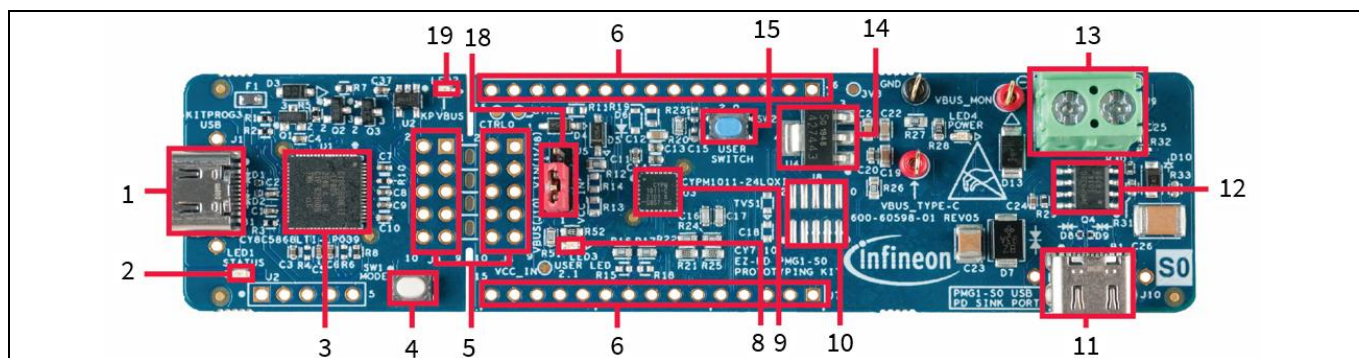


Figure 2 CY7110 EZ-PD™ PMG1-S0 MCU prototyping kit

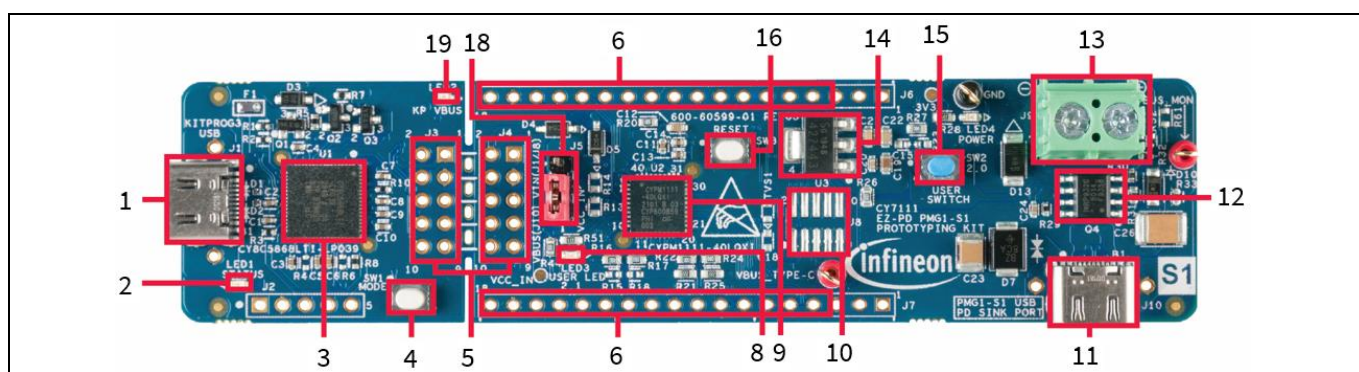


Figure 3 CY7111 EZ-PD™ PMG1-S1 MCU prototyping kit

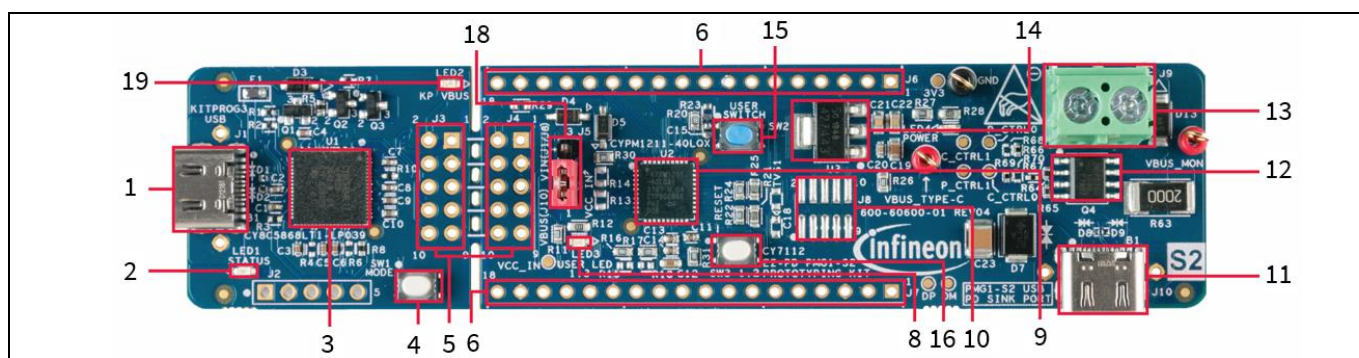


Figure 4 CY7112 EZ-PD™ PMG1-S2 MCU prototyping kit

Introduction

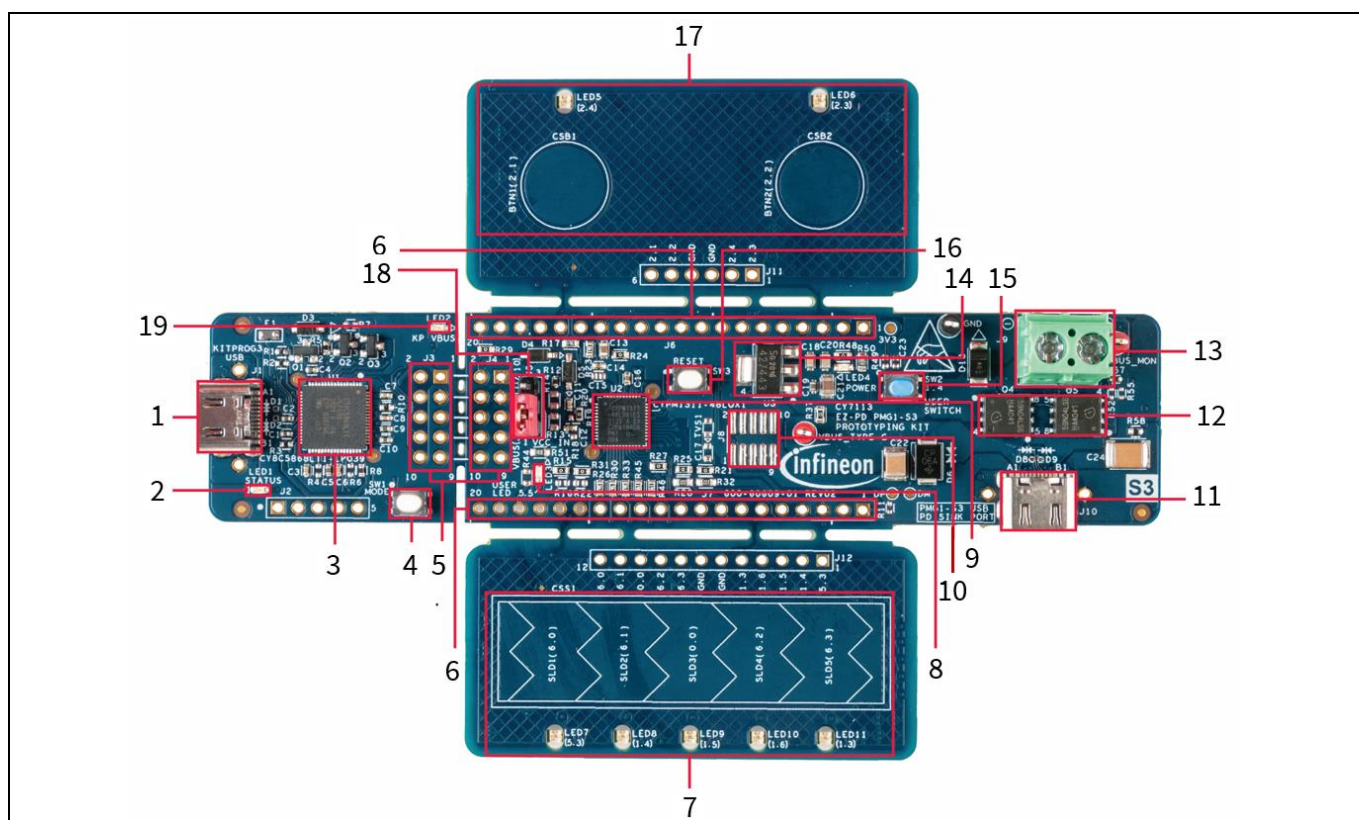


Figure 5 CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit

1. KitProg3 USB Type-C port (J1)
2. KitProg3 status LED (LED1)
3. PSoC™ 5LP MCU controller (U1)
4. KitProg3 mode button (SW1)
5. KitProg3 headers (J3, J4)¹
6. I/O headers (J6, J7)¹
7. CAPSENSE™ slider supported in CY7113
8. User LED (LED3)
9. EZ-PD™ PMG1 microcontroller (CYPM1011-24LQXI in CY7110, CYPM1111-40LQXI in CY7111, CYPM1211-40LQXI in CY7112, and CYPM1311-48LDXI in CY7113)
10. 10-pin SWD/JTAG header¹
11. PMG1 USB PD sink port (J10)
12. Load switch
13. DC_OUT terminal block (J9)
14. 3.3-V onboard LDO
15. User button (SW2)
16. Reset button (SW3)
17. 2 CAPSENSE™ buttons supported in CY7113
18. Power selection jumper (J5)
19. KitProg3 VBUS LED (LED2)

¹ Footprint only, not populated on the board.

Introduction

1.3.1 Kit features

The EZ-PD™ PMG1 MCU prototyping kits feature MCUs from the EZ-PD™ PMG1 high-voltage USB PD microcontroller family.

- The prototyping kits are powered through the EZ-PD™ PMG1 USB PD sink port and the MCU is powered directly from the port. The USB PD 3.1 connector is capable of supporting PD contracts up to 100 W (20 V/ 5 A) for the CY7110, CY7111, and CY7112 kits. It can support PD contract up to 140 W (28 V/ 5 A) for the CY7113 kit.
- The prototyping kits connect to a PC through the KitProg3 Type-C port (J1) for firmware programming and debugging. The kits have a KitProg3 onboard programmer and debugger interface with USB-to-I2C/UART bridge functionality
- The power selection jumper (J5) switches the mode of operation between programming and kit operation
- An option for adding a 10-pin SWD header is integrated into the board design for connecting programming or debugging modules such as MiniProg4
- Four onboard LEDs. The CY7113 kit has seven additional LEDs to indicate the status of CAPSENSE™ sensors (CAPSENSE™ buttons and slider) — see [Table 2](#).

Table 2 Prototyping kit LEDs

LED	Color	Description
KitProg3 status LED (LED1)	Amber	This LED indicates the programming mode and status. The LED functions only when the KitProg3 Type-C port is connected to a PC. See the KitProg3 user guide [6] for details.
KitProg3 power LED (LED2)	Amber	This LED turns ON when the kit is powered through KitProg3 Type-C port, and indicates that the PSoC™ 5LP MCU device is powered.
User LED (LED3)	Green	This is a firmware-driven LED. The kit must be powered through the USB PD Type-C port to use this LED.
Power LED (LED4)	Green	This LED indicates that the board is powered via the USB PD Type-C port. The LED is connected to the output of the onboard 3.3-V LDO.
CAPSENSE™ button LEDs (LED5, LED6)	Green	Feedback LEDs for CAPSENSE™ buttons BTN1 and BTN2. Firmware-driven LEDs to indicate the state of CAPSENSE™ buttons. LED5 – BTN1 LED6 – BTN2
CAPSENSE™ slider LEDs (LED7, LED8, LED9, LED10, LED11)	Green	Feedback LEDs for CAPSENSE™ slider. Firmware-driven LEDs to indicate the state of each sensor in a 5-sensor CAPSENSE™ slider. LED7 – SLD1 LED8 – SLD2 LED9 – SLD3 LED10 – SLD4 LED11 – SLD5

Introduction

- There are two push buttons on the CY7110 kit and three push buttons on the CY7111/CY7112/CY7113 kits. See [Table 3](#) for the details

Table 3 Prototyping kit – push buttons

Button	Description
KitProg3 mode (SW1)	The mode button is used to put the KitProg3 into bootloader mode. See the KitProg3 user guide [6] for details.
User switch (SW2)	This button can be used to provide input to the EZ-PD™ PMG1 MCU. This button pulls the pin to the ground, by default.
Reset switch (SW3) ²	This button is used to reset the EZ-PD™ PMG1 MCU. It connects the EZ-PD™ PMG1 MCU reset (XRES) pin to the ground.

- DC_OUT terminal block to connect the output of the load switch to an external circuit through wires. The Type-C contract voltage is available at the DC_OUT terminal block when a successful PD contract is established
- An onboard LDO regulates the Type-C voltage to 3.3 V and powers the user LED. This is available as a power output to power external modules if required
- All GPIOs and alternate power supply pins are routed to I/O headers placed on the sides of the prototyping kit. The headers are compatible with a 2.54-mm standard breadboard to extend the kit functionality. [Table 4](#) lists the pins count of I/O headers in all EZ-PD™ PMG1 kits

Table 4 I/O header pin count

Kit MPN	Number of I/O header pins
CY7110	15 + 15
CY7111	18 + 18
CY7112	18 + 18
CY7113	20 + 20

1.3.2 Power selection jumper

The power selection jumper (J5) is a three-pin selection header used to switch between operational mode and programming mode.

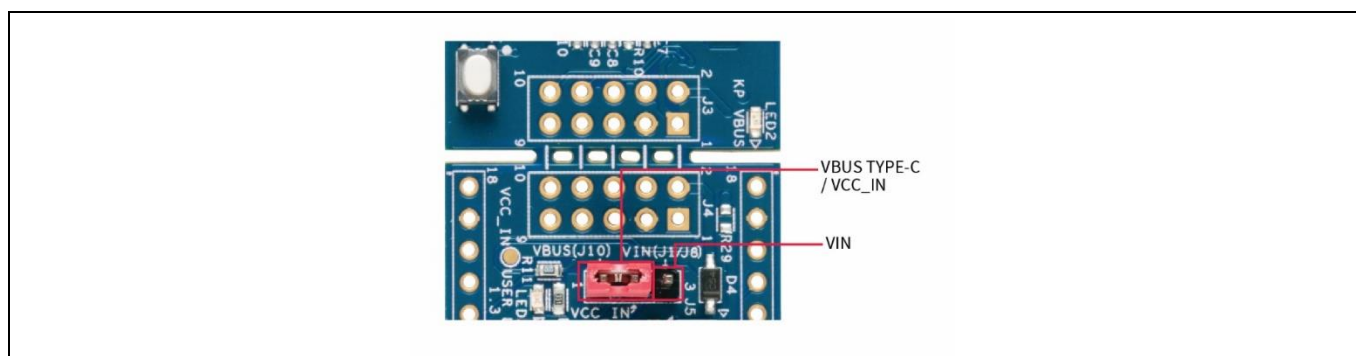


Figure 6 Power selection jumper

² Not applicable for the CY7110 prototyping kit because the EZ-PD™ PMG1-S0 MCU does not support XRES.

Introduction

Table 5 Power selection jumpers

Jumper shunt position	Mode	Description
1-2	Operational mode	The kit is powered from the EZ-PD™ PMG1 USB PD sink port. The jumper should be in this position for normal kit operation.
2-3	Programming mode	The kit is powered from the KitProg3 USB Type-C port. The jumper should be in this position to update the EZ-PD™ PMG1 MCU firmware.

1.3.3 Prototyping kit pinout

EZ-PD™ PMG1 MCU family kits have two I/O headers, J6 and J7, placed on either side of the prototyping kit.

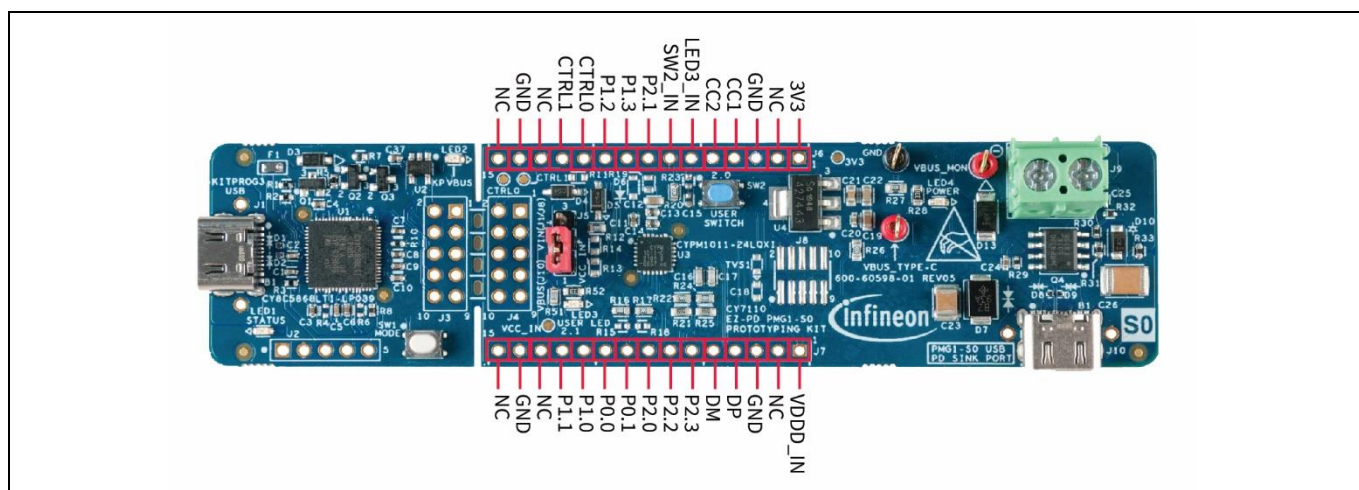


Figure 7 CY7110 prototyping kit - I/O header

Introduction

Table 6 CY7110 MCU I/O header pinout

CY7110 pin	CY7110 pin name	EZ-PD™ PMG1-S0 MCU pin	Primary function	Secondary function	Connection details
J6					
1	3V3	–	3.3-V regulator output	–	Connected to the output of the onboard LDO
2	NC	–	–	–	Not connected
3	GND	–	–	–	Ground
4	CC1	15	CC1 signal	–	–
5	CC2	14	CC2 signal	–	–
6	LED3_IN	10	User LED external input. LED3 is driven by firmware, by default. Use this pin to drive it externally.	P2.1 (GPIO)	Connected to LED3 and P2.1 pin through a 0-ohm resistor
7	SW2_IN	9	User button output. SW2 is input to this EZ-PD™ PMG1 MCU device. This pin can be used to read button status externally.	P2.0 (GPIO)	Connected to SW2 and P2.0 pin through a 0-ohm resistor.
8	P2.1	10	User LED control input	P2.1 (GPIO)	–
9	P1.3	6	UART-Rx	P1.3 (GPIO)	Connected to KitProg3 by default from board revision 5 onwards.
10	P1.2	5	UART-Tx	P1.2 (GPIO)	For earlier revision boards, see KitProg3 (PSoC™ 5LP MCU) for establishing UART connection between the MCU and KitProg3
11	CTRL0	3	VBUS-FET CONTROL-0	–	Default PFET control signal
12	CTRL1	4	VBUS-FET CONTROL-1	–	Optional PFET control signal
13	NC	–	–	–	–
14	GND	–	–	–	Ground
15	NC	–	–	–	–
J7					
1	VDDD_IN	23	Alternate VDDD input pin	–	Connected through the non-populated resistor/diode
2	NC	–	–	–	–

CY7110 pin	CY7110 pin name	EZ-PD™ PMG1-S0 MCU pin	Primary function	Secondary function	Connection details
3	GND	–	–	–	Ground
4	DP	17	USB DP signal	P3.0 (GPIO)	–
5	DM	16	USB DM signal	P3.1 (GPIO)	–
6	P2.3	13	I2C SCL signal	P2.3 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
7	P2.2	12	I2C SDA signal	P2.2 (GPIO)	
8	P2.0	9	User button SW2 input	P2.0 (GPIO)	–
9	P0.1	8	SWD clock	P0.1 (GPIO)	–
10	P0.0	7	SWD data	P0.0 (GPIO)	–
11	P1.0	1	GPIO	–	–
12	P1.1	2	GPIO	–	–
13	NC	–	–	–	–
14	GND	–	–	–	Ground
15	NC	–	–	–	–

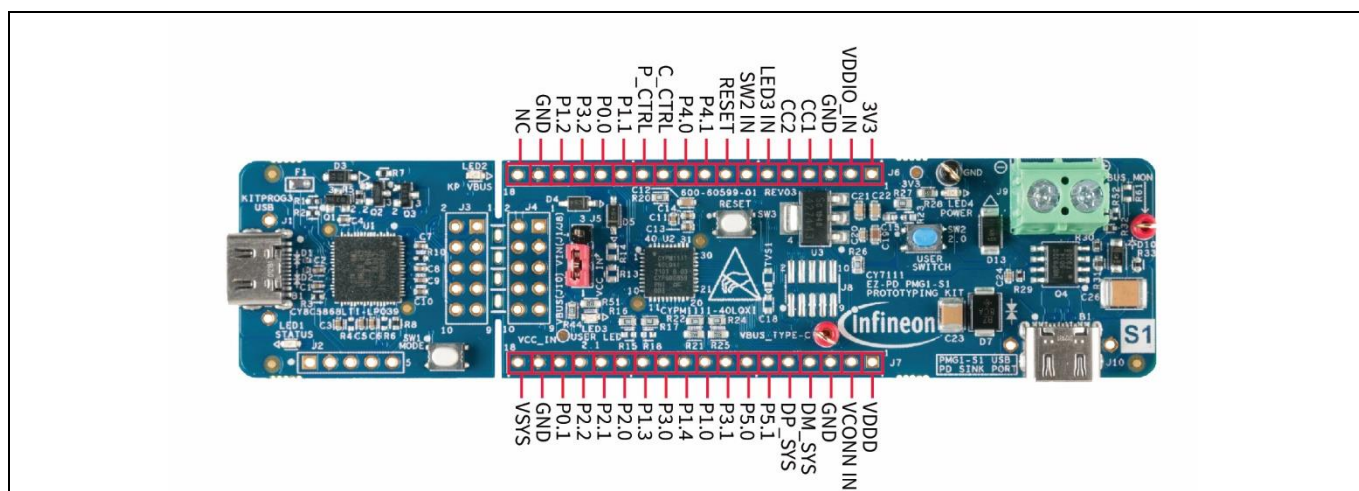


Table 7 CY7111 MCU I/O header pinout

002-32601 Rev. *C
2021-12-02

Introduction

CY7111 pin	CY7111 pin name	EZ-PD™ PMG1-S1 MCU pin	Primary function	Secondary function	Connection details
3	GND	–	–	–	Ground
4	CC1	9	CC1 signal	–	–
5	CC2	7	CC2 signal	–	–
6	LED3_IN	14	User LED external input. LED3 is driven by firmware, by default. Use this pin to drive it externally.	P2.1 (GPIO)	Connected to LED3 and P2.1 pin through a 0-ohm resistor
7	SW2_IN	13	User button output. SW2 is input to the EZ-PD™ PMG1 MCU device. This pin can be used to read the button status externally.	P2.0 (GPIO)	Connected to SW2 and P2.0 pin through a 0-ohm resistor
8	RESET	10	Reset pin	–	
9	P4.1	30	UART-Rx	P4.1 (GPIO)	Connected to KitProg3 by default from board revision 3 onwards.
10	P4.0	29	UART-Tx	P4.0 (GPIO)	For earlier revision boards, refer to KitProg3 (PSoC™ 5LP MCU) section for establishing a UART connection between the MCU and KitProg3
11	C_CTRL	12	Consumer PFET load switch control signal	–	Connected to the load switch
12	P_CTRL	11	Provider PFET load switch control signal	–	Connected to the load switch through the non-populated resistor
13	P1.1	3	GPIO	–	–
14	P0.0	38	GPIO	–	–
15	P3.2	21	GPIO	–	–
16	P1.2	4	GPIO	–	–
17	GND	–	–	–	Ground
18	NC	–	–	–	–

J7

1	VDDD	31	EZ-PD™ PMG1-S1 Internal LDO output	–	–
2	VCONN_IN	8	VCONN source external input. By	–	Connected through the non-populated resistor/diode

EZ-PD™ PMG1 MCU prototyping kits guide

CY7110/CY7111/CY7112/CY7113



Introduction

CY7111 pin	CY7111 pin name	EZ-PD™ PMG1-S1 MCU pin	Primary function	Secondary function	Connection details
			default, V5V is left floating on the device.		
3	GND	–	–	–	Ground
4	DM_SYS	24	Connect to USB 2.0 DM from the host side	–	–
5	DP_SYS	23	Connect to USB 2.0 DP from the host side	–	–
6	P5.1	17	I2C SCL signal	P5.1 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
7	P5.0	16	I2C SDA signal	P5.0 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
8	P3.1	20	GPIO	–	–
9	P1.0	2	SWD clock	P1.0 (GPIO)	–
10	P1.4	6	SWD data	P1.4 (GPIO)	–
11	P3.0	18	GPIO	–	–
12	P1.3	5	GPIO	–	–
13	P2.0	13	User button SW2 input	P2.0 (GPIO)	–
14	P2.1	14	User LED control input	P2.1 (GPIO)	–
15	P2.2	15	GPIO	–	–
16	P0.1	39	GPIO	–	–
17	GND	–	–	–	Ground
18	VSYS	19	Alternate power input option-VSYS	–	Connected through the non-populated resistor/diode

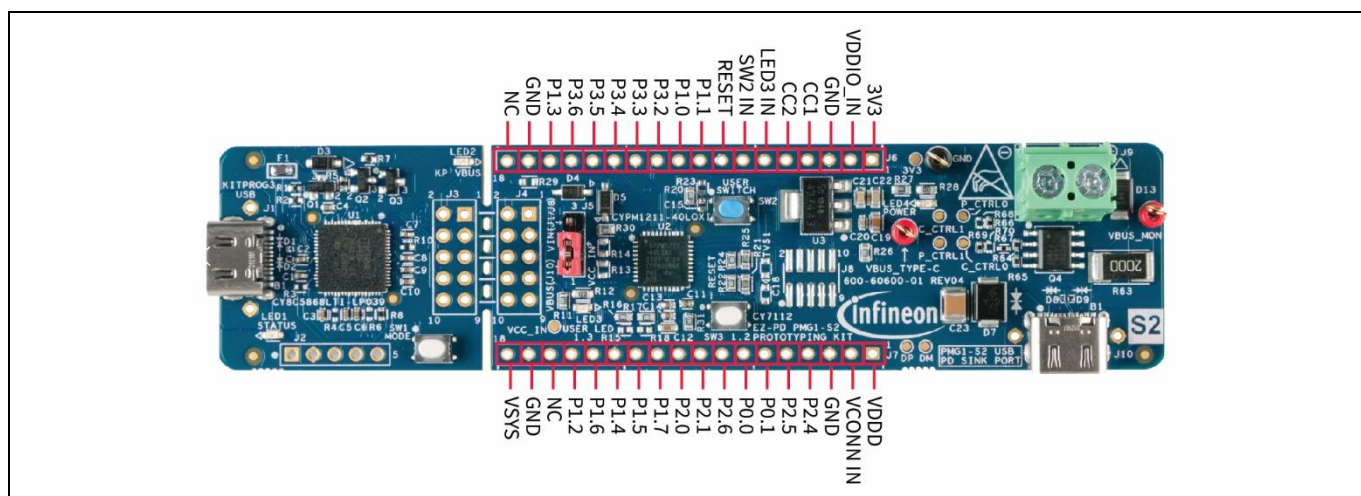


Figure 9 CY7112 prototyping kit - I/O header

Introduction

Table 8 CY7112 MCU I/O header pinout

CY7112 pin	CY7112 pin name	EZ-PD™ PMG1-S2 MCU pin	Primary function	Secondary function	Connection details
J6					
1	3V3	3.3-V regulator output	–	Connected to the output of the onboard LDO	3.3-V regulator output
2	VDDIO_IN	18	VDDIO alternate input	–	Connected through the non-populated resistor/diode
3	GND	–	–	–	Ground
4	CC1	5	CC1 signal	–	–
5	CC2	3	CC2 signal	–	–
6	LED3_IN	10	User LED external input. LED3 is driven by firmware, by default. Use this pin to drive it externally.	P1.3 (GPIO)	Connected to LED3, and P1.3 pin through a 0-ohm resistor.
7	SW2_IN	9	User button output. SW2 is input to EZ-PD™ PMG1 MCU device. This pin can be used to read the button status externally.	P1.2 (GPIO)	Connected to SW2, and P1.2 pin through a 0-ohm resistor
8	RESET	26	Reset pin	–	–
9	P1.1	8	UART-Rx	P1.1 (GPIO)	Connected to KitProg3 by default from board revision 3 onwards. For earlier revision boards, refer to KitProg3 (PSoC™ 5LP MCU) section for establishing a UART connection between the MCU and KitProg3
10	P1.0	7	UART-Tx	P1.0 (GPIO)	
11	P3.2	34	GPIO	–	–
12	P3.3	35	GPIO	–	–
13	P3.4	36	GPIO	–	–
14	P3.5	37	GPIO	–	–
15	P3.6	38	GPIO	–	–
16	P1.3	10	User LED control input	P1.3 (GPIO)	–
17	GND	–	–	–	Ground
18	NC	–	–	–	–

Introduction

CY7112 pin	CY7112 pin name	EZ-PD™ PMG1-S2 MCU pin	Primary function	Secondary function	Connection details
J7					
1	VDDD	17	EZ-PD™ PMG1-S1 MCU internal LDO output	–	–
2	VCONN_IN	4	VCONN source external input. By default, V5V is left floating on the device.	–	Connected through the non-populated resistor/diode
3	GND	–	–	–	Ground
4	P2.4	23	GPIO	–	–
5	P2.5	24	GPIO	–	–
6	P0.1	28	I2C SCL signal	P0.1 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
7	P0.0	27	I2C SDA signal	P0.0 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
8	P2.6	25	GPIO	–	–
9	P2.1	16	SWD clock	P2.1 (GPIO)	–
10	P2.0	15	SWD data	P2.0 (GPIO)	–
11	P1.7	14	GPIO	–	–
12	P1.5	13	GPIO	–	–
13	P1.4	12	GPIO	–	–
14	P1.6	11	GPIO	–	–
15	P1.2	9	User button SW2 input	P1.2 (GPIO)	–
16	NC	–	–	–	–
17	GND	–	–	–	Ground
18	VSYS	20	Alternate power input option-VSYS	–	Connected through the non-populated resistor/diode

Introduction

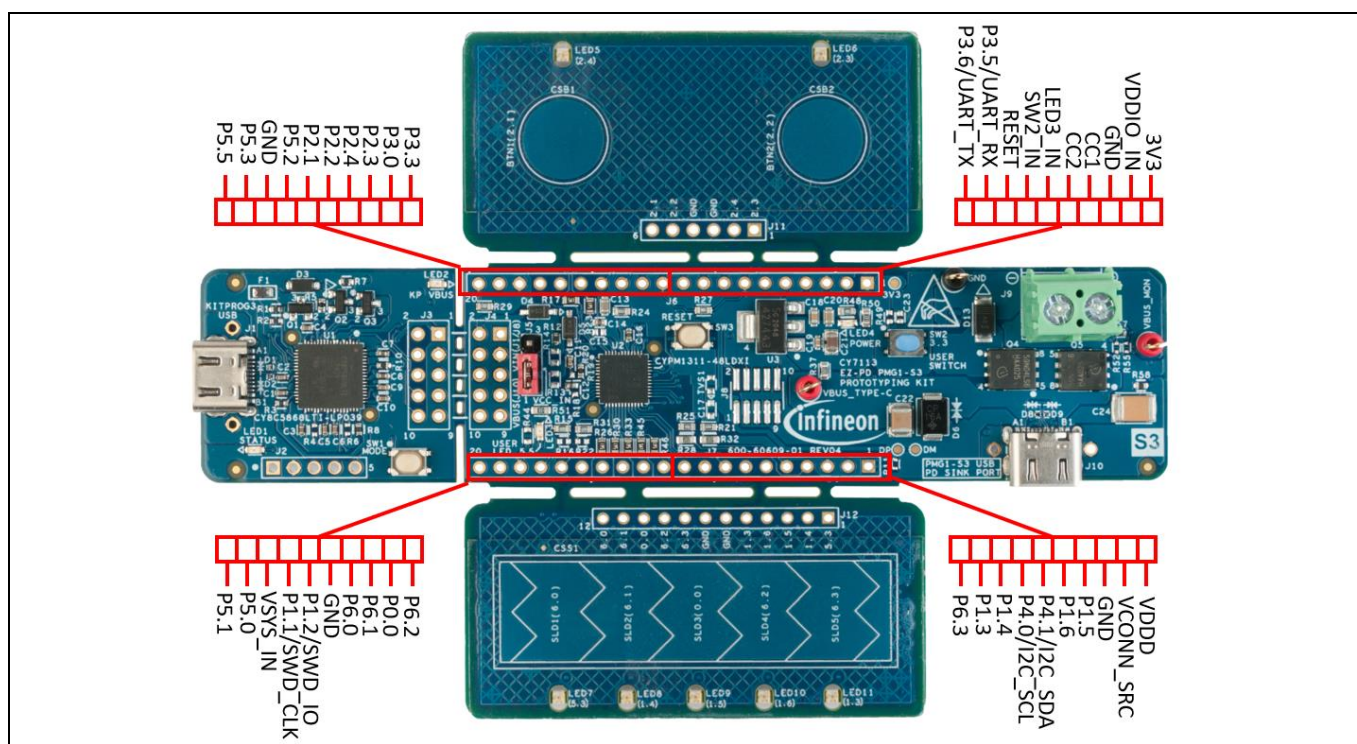


Figure 10 CY7113 prototyping kit - I/O header

Table 9 CY7113 MCU I/O header pinout

CY7113 pin	CY7113 pin name	EZ-PD™ PMG1-S3 MCU pin	Primary function	Secondary function	Connection details
J6					
1	3V3	3.3-V regulator output	–	Connected to the output of the on-board LDO	3.3-V regulator output
2	VDDIO_IN	18	VDDIO alternate input	–	Connected through the non-populated resistor/diode
3	GND	–	–	–	–
4	CC1	28	CC1 Signal	–	–
5	CC2	30	CC2 Signal	–	–
6	LED3_IN	11	User LED external input. By default, LED3 is driven by firmware. Use this pin to drive it externally.	P5.5 (GPIO)	Connected to LED3, and P5.5 pin through a 0-ohm resistor.
7	SW2_IN	2	User button output. SW2 is input to EZ-PD™ PMG1 MCU device. This pin can be used to read the button status externally.	P3.3 (GPIO)	Connected to SW2, and P3.3 pin through a 0-ohm resistor
8	RESET	33	Reset pin	–	–

Introduction

CY7113 pin	CY7113 pin name	EZ-PD™ PMG1-S3 MCU pin	Primary function	Secondary function	Connection details
9	P3.5	3	UART-Rx	P3.5 (GPIO)	Connected to KitProg3 to implement USB-to-UART bridge functionality
10	P3.6	4	UART-Tx	P3.6 (GPIO)	
11	P3.3	2	User button SW2 input	P3.3 (GPIO)	-
12	P3.0	1	GPIO	-	-
13	P2.3	46	CAPSENSE™ button BTN2 status LED	P2.3 (GPIO)	Indicate status of BTN2 CAPSENSE™ button
14	P2.4	45	CAPSENSE™ button BTN1 status LED	P2.4 (GPIO)	Indicate status of BTN1 CAPSENSE™ button
15	P2.2 ³	47	CAPSENSE™ button BTN2 sensor signal	P2.2 (GPIO)	-
16	P2.1 ³	48	CAPSENSE™ button BTN1 sensor signal	P2.1 (GPIO)	-
17	P5.2	9	GPIO	Driving shield GPIO	-
18	GND	-	-	-	-
19	P5.3	10	CAPSENSE™ slider sensor SLD1 status LED	P5.3 (GPIO)	Indicates status of CAPSENSE™ slider sensor SLD1
20	P5.5	11	User LED control input	P5.5 (GPIO)	-
J7					
1	VDDD	43	EZ-PD™ PMG1-S3 MCU internal LDO output	-	VDDD, VDDA, and VDDIO shorted in the design
2	VCONN_SRC	29	VCONN source external input	-	Connected through the non-populated resistor/diode
3	GND	-	Ground	-	-
4	P1.5	21	CAPSENSE™ slider sensor SLD3 status LED	P1.5 (GPIO)	Indicates status of CAPSENSE™ slider sensor SLD3
5	P1.6	22	CAPSENSE™ slider sensor SLD4 status LED	P1.6 (GPIO)	Indicates status of CAPSENSE™ slider sensor SLD4
6	P4.1	35	I2C-SDA	P4.1 (GPIO)	Connected to KitProg3 for USB-to-I2C bridge
7	P4.0	34	I2C-SCL	P4.0 (GPIO)	
8	P1.4	20	CAPSENSE™ slider (CSS) sensor SLD2 status LED	P1.4 (GPIO)	Indicates status of CAPSENSE™ slider (CSS) sensor SLD2

³ These I/Os are connected to CAPSENSE™ buttons/sliders by default and need hardware modification to route the I/Os to J6/J7 headers. Refer to the kit schematic file for more details.

Introduction

CY7113 pin	CY7113 pin name	EZ-PD™ PMG1-S3 MCU pin	Primary function	Secondary function	Connection details
9	P1.3	14	CAPSENSE™ slider (CSS) sensor SLD5 status LED	P1.3 (GPIO)	Indicates status of CAPSENSE™ slider (CSS) sensor SLD5
10	P6.3 ⁴	19	CAPSENSE™ slider (CSS) SLD5 sensor signal	P6.3 (GPIO)	-
11	P6.2 ⁴	18	CAPSENSE™ slider (CSS) SLD4 sensor signal	P6.2 (GPIO)	-
12	P0.0 ⁴	17	CAPSENSE™ slider (CSS) SLD3 sensor signal	P0.0 (GPIO)	-
13	P6.1 ⁴	16	CAPSENSE™ slider (CSS) SLD2 sensor signal	P6.1 (GPIO)	-
14	P6.0 ⁴	15	CAPSENSE™ slider (CSS) SLD1 sensor signal	P6.0 (GPIO)	-
15	GND	Ground	-	-	-
16	P1.2	13	SWD data	P1.2 (GPIO)	-
17	P1.1	12	SWD clock	P1.1 (GPIO)	-
18	VSYS_IN	36	Alternate power input option-VSYS	-	Connected through the non-populated resistor/diode
19	P5.0	7	GPIO	CTANK	CTANK capacitor connected through no-load resistor
20	P5.1 ⁴	8	CMOD	P5.1 (GPIO)	-

1.3.4 KitProg3 header

The KitProg3 module is the onboard programming and debug module on the prototyping kit. It also functions as the USB-to-I2C and USB-to-UART bridge.

All the signals from KitProg3 including power, reset, UART, and I2C are routed to the 10-pin header located on the KitProg3 section of the board (J3). Similarly, the signals from the PMG1 MCU are routed to the 10-pin header on the PMG1 section of the board (J4).

Signal lines between J3 and J4 are connected on the board by default. However, in the following older revisions of the EZ-PD™ PMG1 kit MCU boards, the UART signals (Tx and Rx) must be explicitly connected using jumper wires:

- CY7110 board revision 3 or lower
- CY7111 board revision 2 or lower
- CY7112 board revision 2 or lower
- CY7113 board revision 3 or lower

⁴ These I/Os are connected to CAPSENSE™ buttons/sliders by default and need hardware modification to route the I/Os to J6/J7 headers. Refer to the kit schematic file for more details.

Introduction

See [KitProg3 \(PSoc™ 5LP MCU\)](#) for more details.

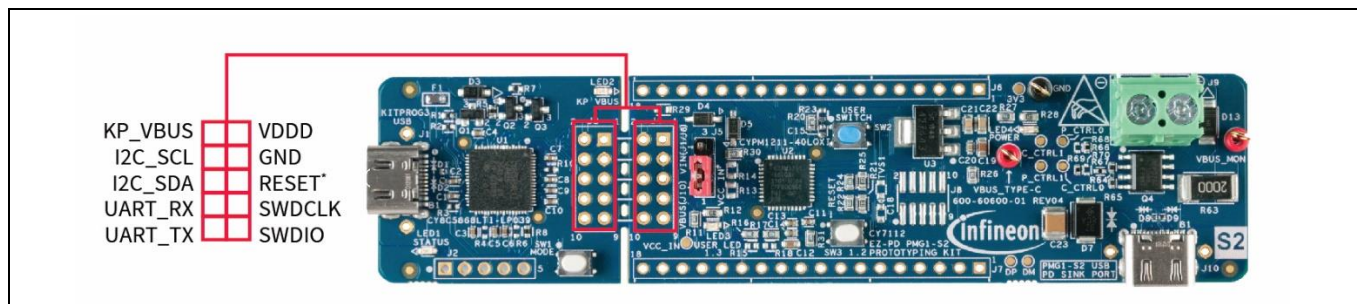


Figure 11 KitProg3 header

Table 10 KitProg3 header pinout

Pin #	Pin name	Description
1	VDDD	Target (EZ-PD™ PMG1 MCU) internal regulator output
2	KP_VBUS	KitProg3 VBUS/programming VBUS
3	GND	Ground
4	I2C_SCL	I2C SCL signal between KitProg3 and EZ-PD™ PMG1 MCU
5	RESET ⁵	Reset signal for programming EZ-PD™ PMG1 MCU
6	I2C_SDA	I2C SDA signal
7	SWDCLK	SWD clock
8	UART_RX	UART Rx
9	SWDIO	SWD data
10	UART_TX	UART Tx

⁵ Reset not available on CY7110.

2 EZ-PD™ PMG1 MCU software tools

This section describes the details of ModusToolbox™ software and CYPRESS™ programmer, and the installation procedure. ModusToolbox™ software provides a complete development platform for developing EZ-PD™ PMG1 MCU-based applications. CYPRESS™ programmer is a GUI-based tool to program EZ-PD™ PMG1 MCU devices. This is a programming option in addition to the programming and debugging options available in ModusToolbox™ software.

2.1 Before you begin

The installation of ModusToolbox™ software and other Infineon software require administrator privilege; this is not required to run the software once it is installed. See the installation guide [9]. Before you install the software, close any other Infineon software or application that are currently running.

2.2 ModusToolbox™ software

ModusToolbox™ software includes various tools required for developing, configuring, building, programming, and debugging an EZ-PD™ PMG1 MCU-based application. An Eclipse-based IDE provided along with the ModusToolbox™ software supports application configuration and development. Other IDEs such as Visual Studio Code, IAR Embedded Workbench, and uVision are also supported. Note that the support for EZ-PD™ PMG1 MCU is added in ModusToolbox™ software version 2.3 onwards.

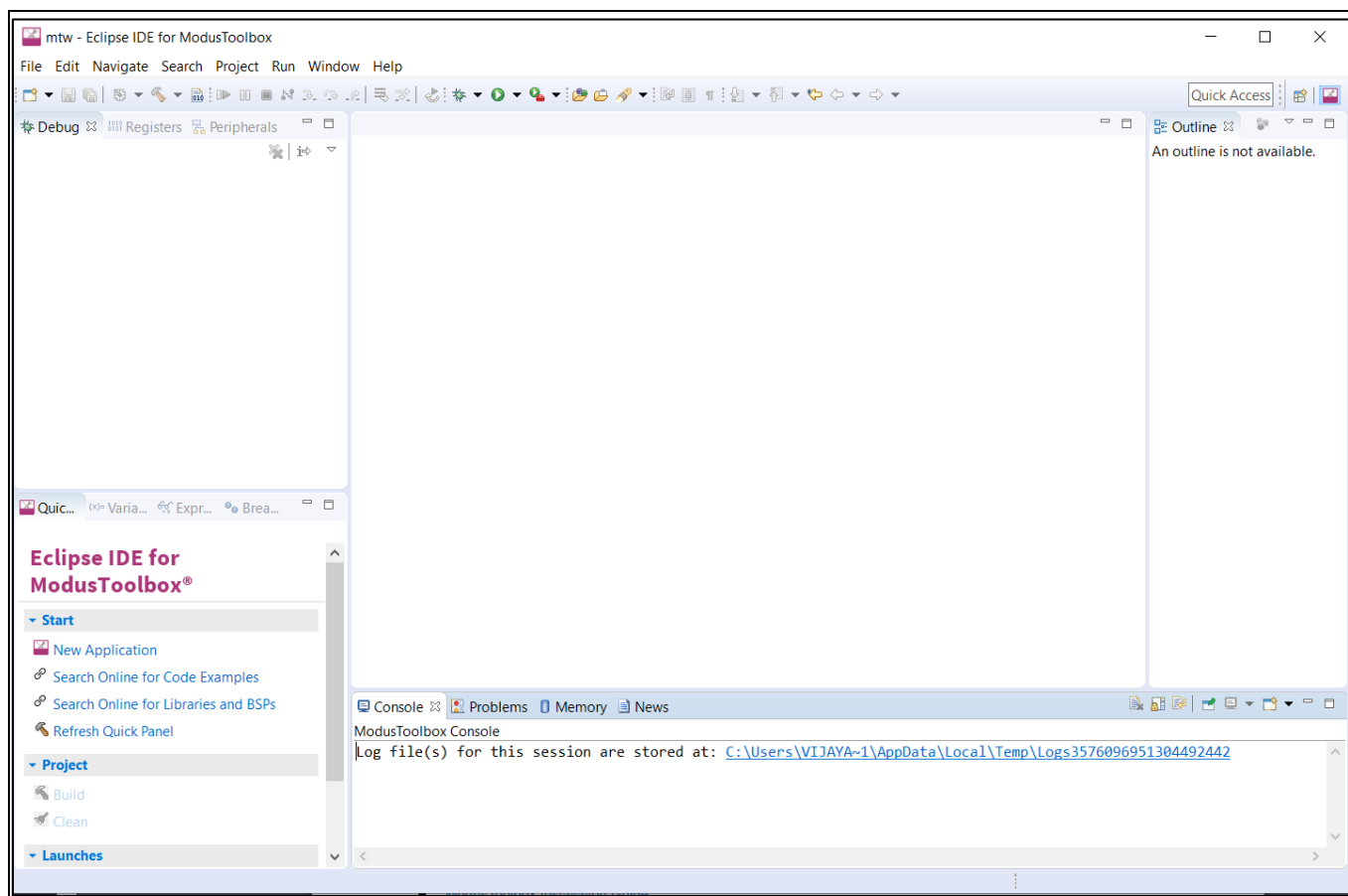


Figure 12 Eclipse IDE for ModusToolbox™ software

2.2.1 Device configurator

The device configurator tool in ModusToolbox™ software helps you to configure hardware blocks or middleware libraries. The configurator is available as a GUI and Command Line Interface (CLI) tool. If you use the Eclipse IDE provided with ModusToolbox™ software, you can access the device configurator tool from the IDE's Quick Panel.

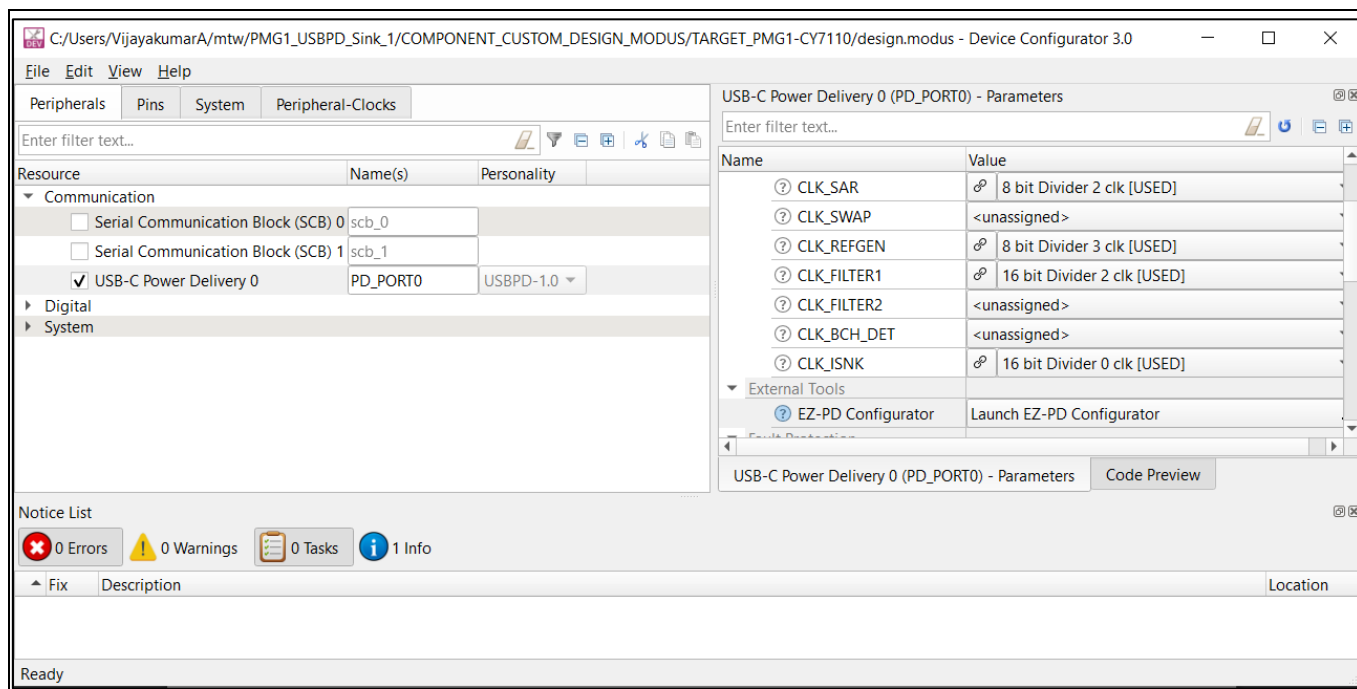


Figure 13 Device configurator

2.2.2 EZ-PD™ configurator

The EZ-PD™ configurator is another tool in ModusToolbox™ software used to configure USB PD features in an application that supports it. You can launch the EZ-PD™ configurator from Eclipse IDE's Quick Panel, or stand-alone.

EZ-PD™ PMG1 MCU prototyping kits guide

CY7110/CY7111/CY7112/CY7113

EZ-PD™ PMG1 MCU software tools

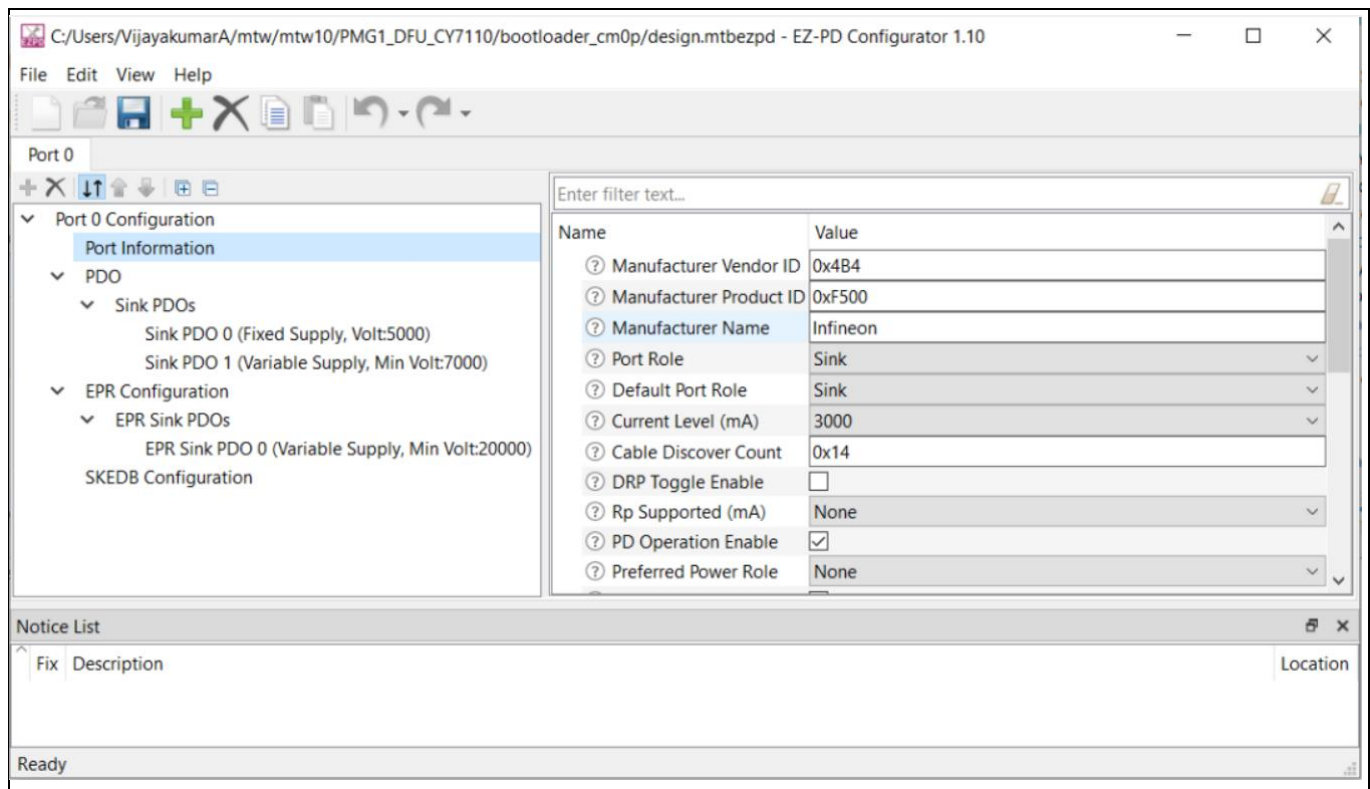


Figure 14 EZ-PD™ configurator

2.2.3 Device firmware update (DFU) host tool

The device firmware update (DFU) host tool is a stand-alone program included with ModusToolbox™ software. This tool is used to communicate with an EZ-PD™ PMG1 MCU that has already been programmed with an application that includes the DFU capability. The tool communicates with the EZ-PD™ PMG1 MCU over an I2C or UART interface to update the firmware. See the [Device firmware update host guide](#) for details.

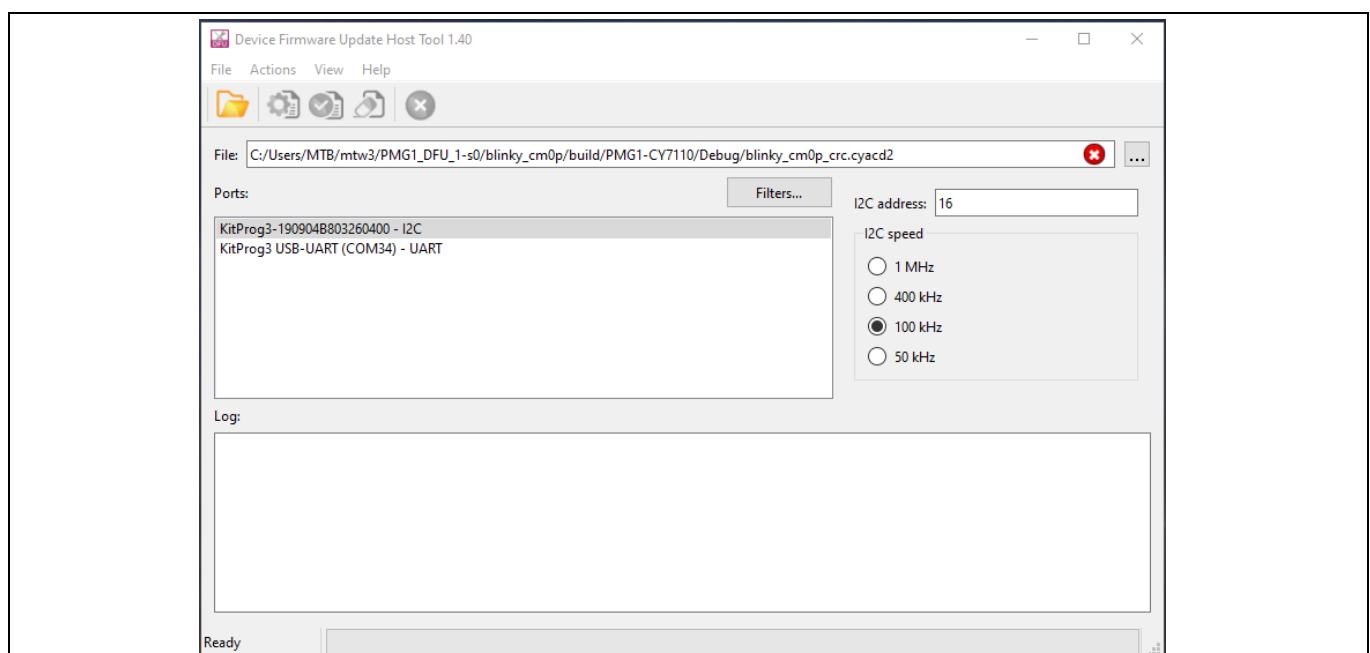


Figure 15 Device firmware update (DFU) host tool

2.2.4 CAPSENSE™ configurator

The CAPSENSE™ configurator is included with ModusToolbox™ software, to create and configure CAPSENSE™ widgets, and generate code to control the application firmware. The CAPSENSE™ configurator is available for CAPSENSE™ example projects using the CY7113 prototyping kit. See the [CAPSENSE™ configurator guide](#) for details.

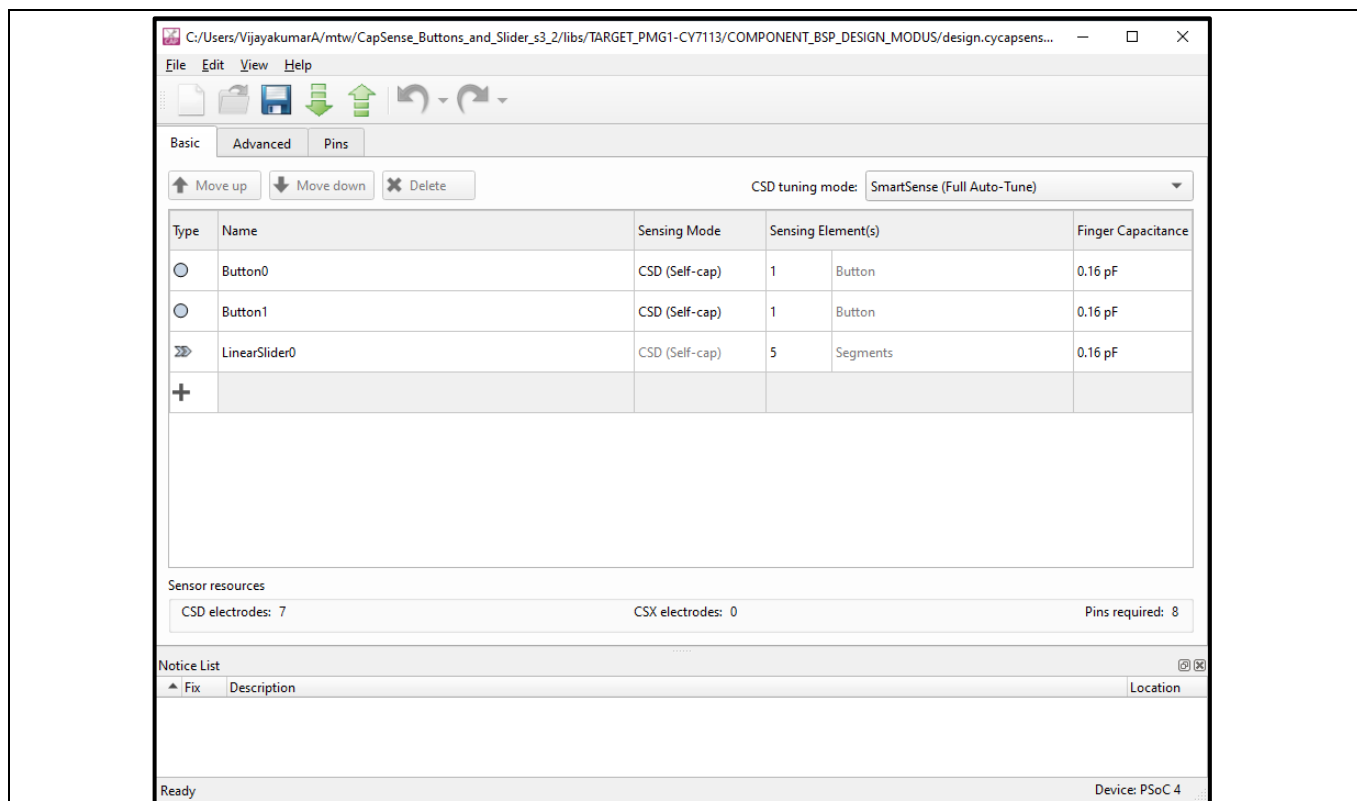


Figure 16 CAPSENSE™ configurator

2.2.5 CAPSENSE™ tuner

A CAPSENSE™ tuner is a stand-alone tool included with the ModusToolbox™ software. The tool is used for tuning, testing, and debugging for the easy and smooth design of CAPSENSE™ interfaces such as buttons and sliders. See the [CAPSENSE™ tuner guide](#) for details.

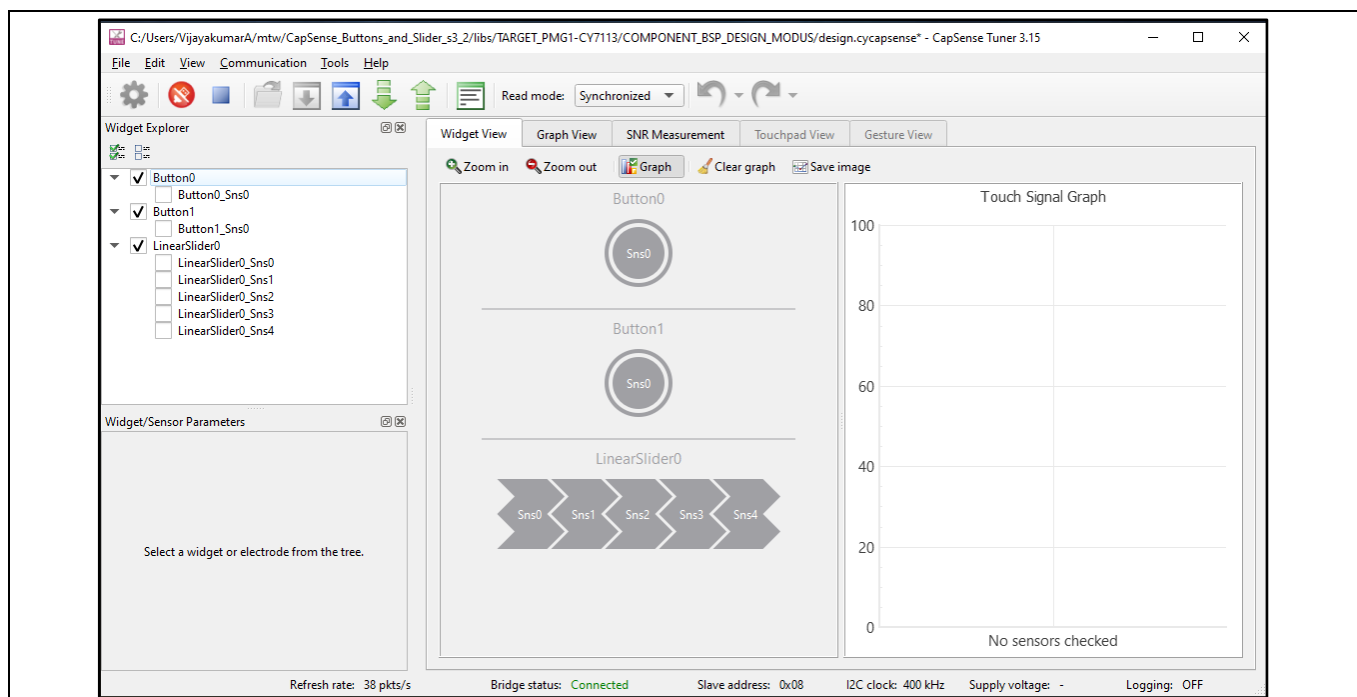


Figure 17 CAPSENSE™ tuner

2.3 CYPRESS™ programmer (CYP)

CYPRESS™ programmer (CYP) is a stand-alone, cross-platform, flash programmer tool. CYP provides a GUI to program, erase, verify, and read the flash of the target MCU. CYP supports the HEX, SREC, ELF, and BIN programming file formats. Note that ModusToolbox™ can also program PMG1 devices so the user of CYP is optional.

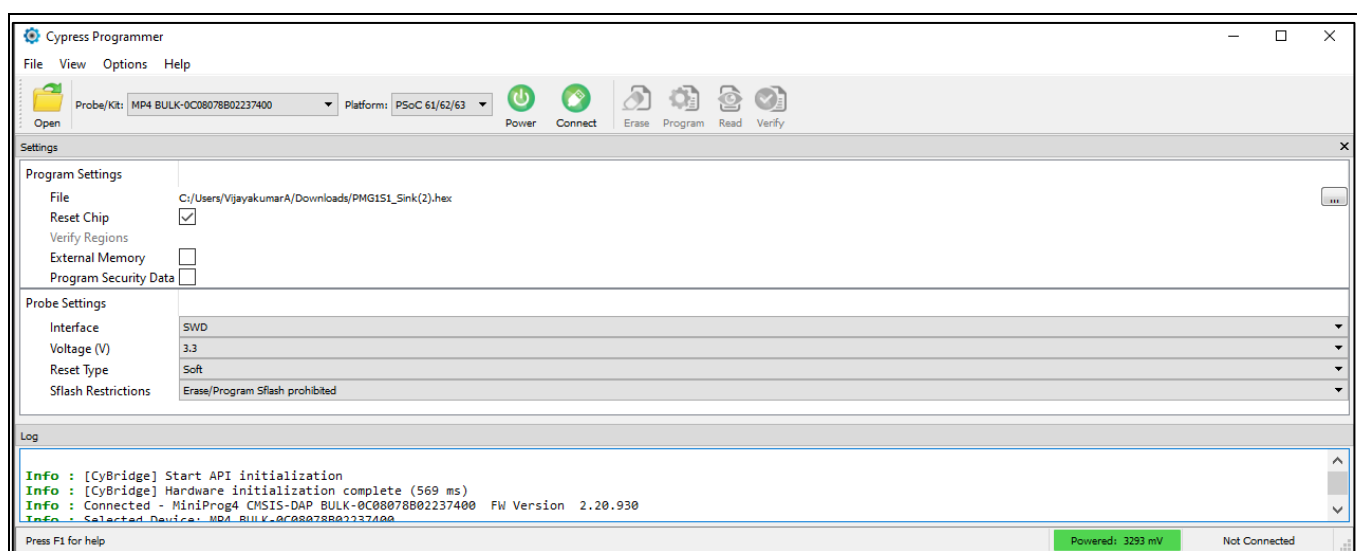


Figure 18 CYPRESS™ programmer

See the CYPRESS™ programmer user guide [\[10\]](#) for the detailed procedure to download and install CYP.

3 EZ-PD™ PMG1 prototyping kit system design

3.1 Top-level hardware design

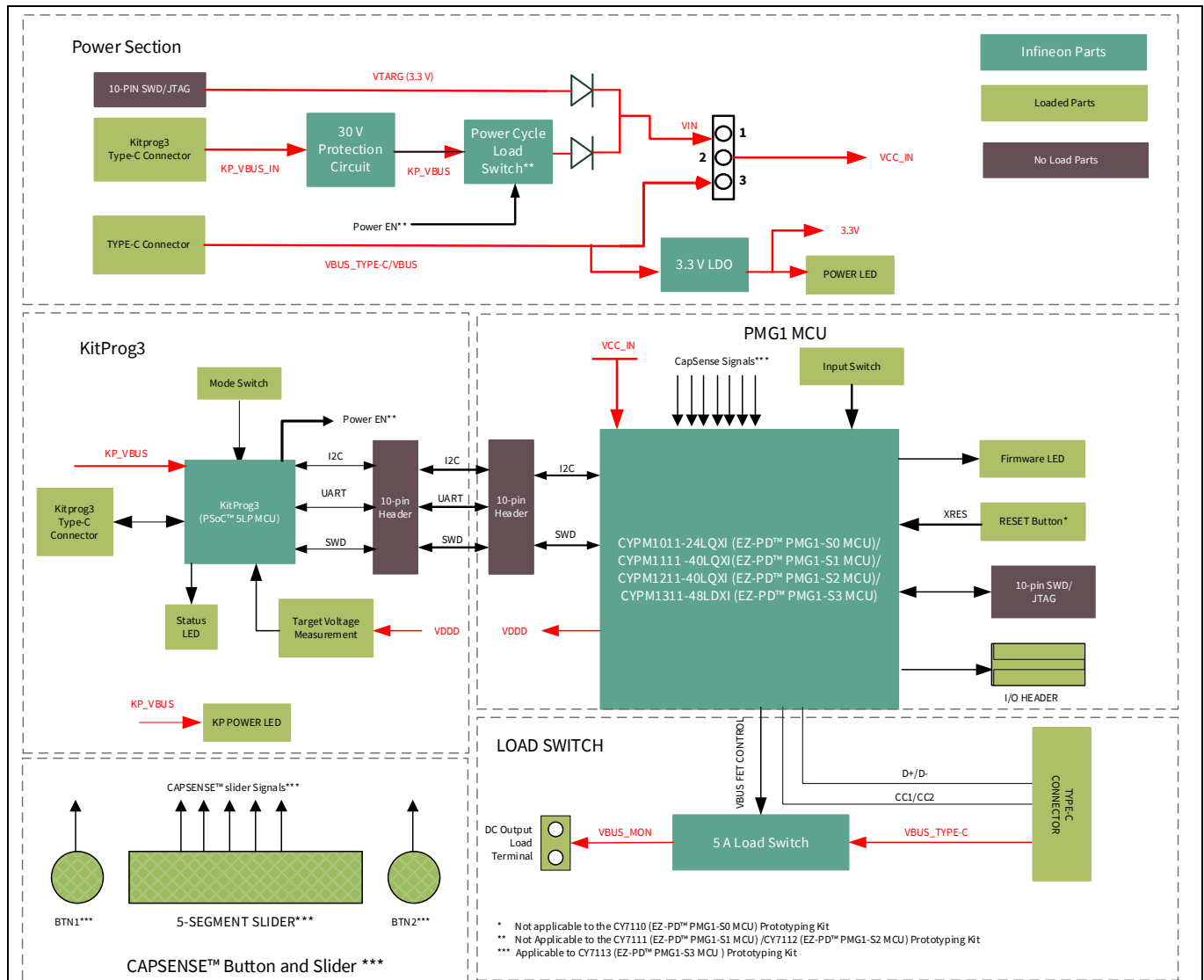


Figure 19 CY711x EZ-PD™ PMG1-Sx MCU prototyping kit block diagram

3.2 Power section

3.2.1 Power modes

EZ-PD™ PMG1 MCU kits support two types of powering modes: programming mode and operational mode.

The power selection jumper position decides the power input connected to the MCU and selects between the programming and operation power supply. Short pins 1–2 for operational mode (VBUS Type-C) and short 2–3 for programming mode (VIN).

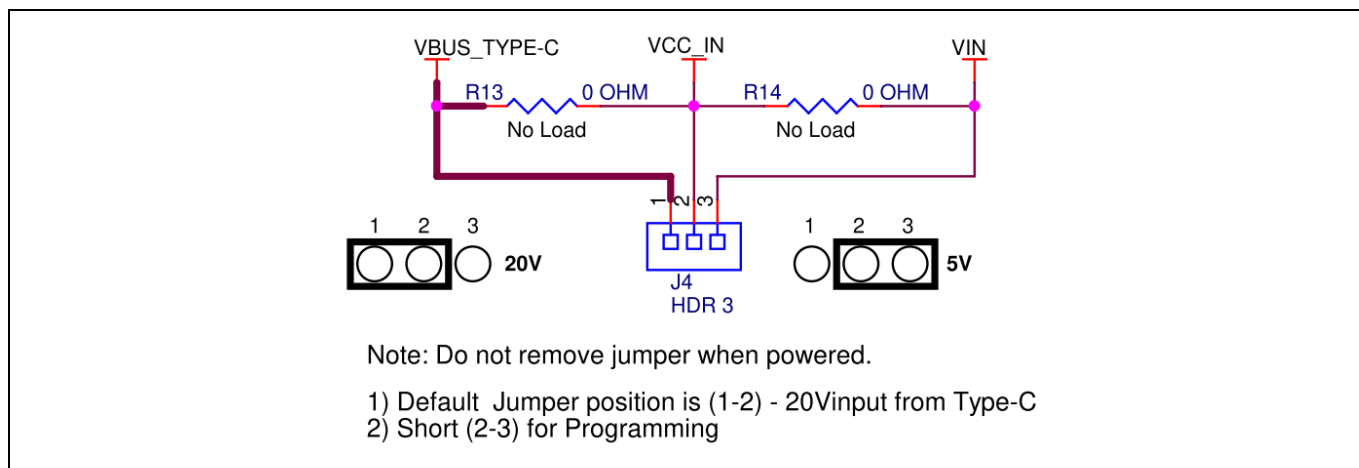


Figure 20 Power selection jumper

3.2.1.1 Programming mode

For programming using the onboard KitProg3, the kit gets powered through the KitProg3 USB Type-C port (J1). When the programming mode setting is selected, power from the programming connector gets connected to the VBUS pin of the EZ-PD™ PMG1 MCU.

3.2.1.2 Operational mode

For normal operation, the prototyping kit gets powered through the EZ-PD™ PMG1 USB PD sink port. This voltage is directly connected to the VBUS pin via the power selection jumper, and the internal LDO down converts VBUS voltage to 3.3 V for chip operation.

The CY7110, CY7111, and CY7112 kits support USB PD contracts up to 100 W (20 V, 5 A). The CY7113 kit supports USB PD contracts up to 140 W (28 V, 5 A).

3.2.2 3.3-V LDO and power LED

The onboard 3.3-V LDO generates 3.3-V from the PMG1 USB PD sink port VBUS and is available as the power input for the user button and user LED. This 3.3-V regulated output is available on an I/O header and can be used as a power rail for external components and modules. The onboard 3.3-V LDO can source a maximum current of 400 mA. The power LED glows green when the regulator output is available.

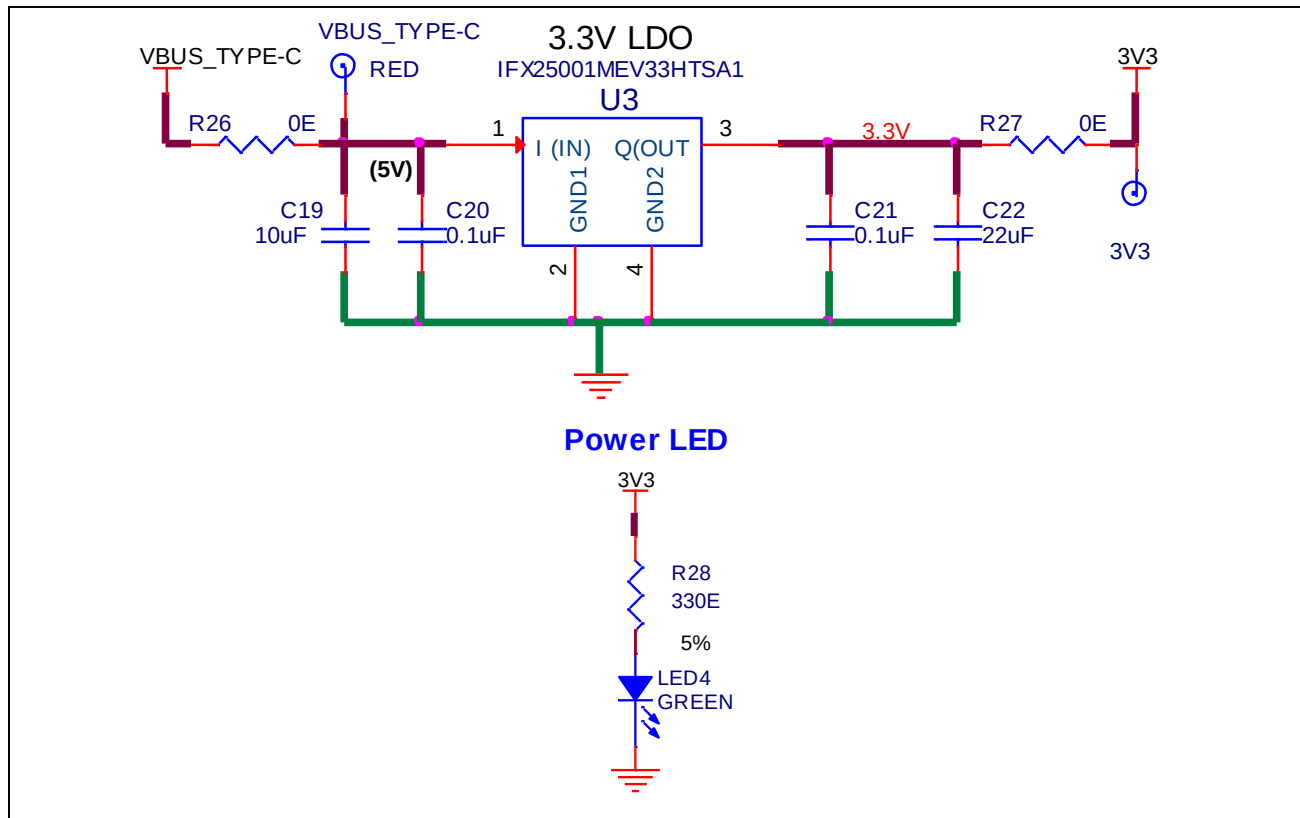


Figure 21 Onboard 3.3-V LDO and power LED

3.2.3 Overvoltage protection circuit

A MOSFET and Zener diode-based protection circuit is implemented to protect the KitProg3 (PSoC™ 5LP MCU) from damage due to overvoltage and reverse voltage fault on the programming interface. The maximum rating for the PSoC™ 5LP MCU device is 6 V; the circuit will shut down the supply if the voltage exceeds 6 V.

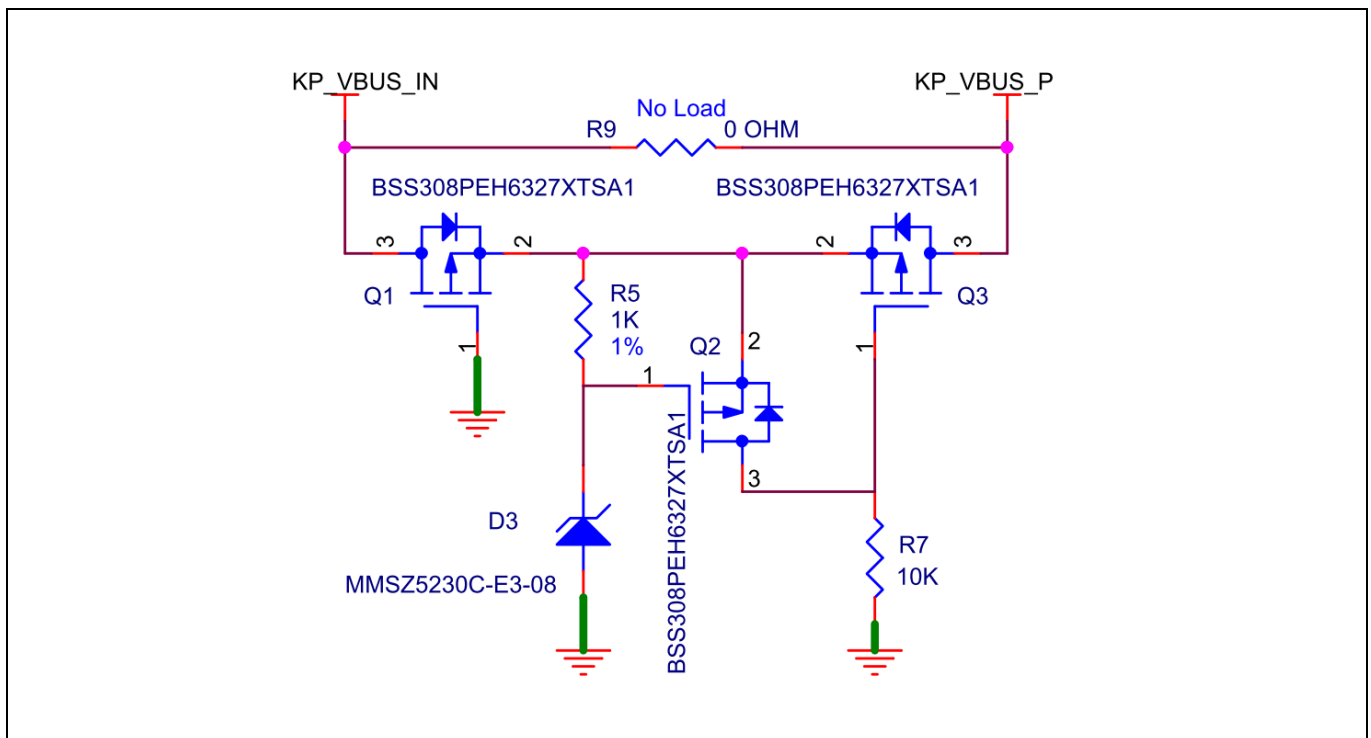


Figure 22 Overvoltage protection circuit for KitProg3

Q2 MOSFET protects the rest of the circuit from reverse voltage by acting as an open switch for reverse potential. When KP_VBUS_IN is reverse voltage, Q1 will be OFF, its body diode is reverse-biased, and Q1 acts as an open switch. For normal potential, Q1 is a closed path.

Q3 is ON during normal operation through R5, and Q2 is OFF. When an overvoltage fault occurs, Q2 turns ON and pulls up the gate of Q3 to Q2's drain voltage, and turns OFF Q3 to prevent overvoltage from reaching the rest of the circuit. The diode D3 controls the overvoltage threshold for the circuit.

3.2.4 Power-cycle load switch (CY7110)

The approach to acquire the EZ-PD™ PMG1-S0 MCU during programming/debugging is different from other EZ-PD™ PMG1 MCU family devices because the EZ-PD™ PMG1-S0 MCU does not have a reset pin. Therefore, KitProg3 uses a power cycling approach to acquire the target in programming/debug mode. It is implemented using a load switch (U2) which is controlled by the KitProg3 to toggle the power to the EZ-PD™ PMG1-S0 MCU.

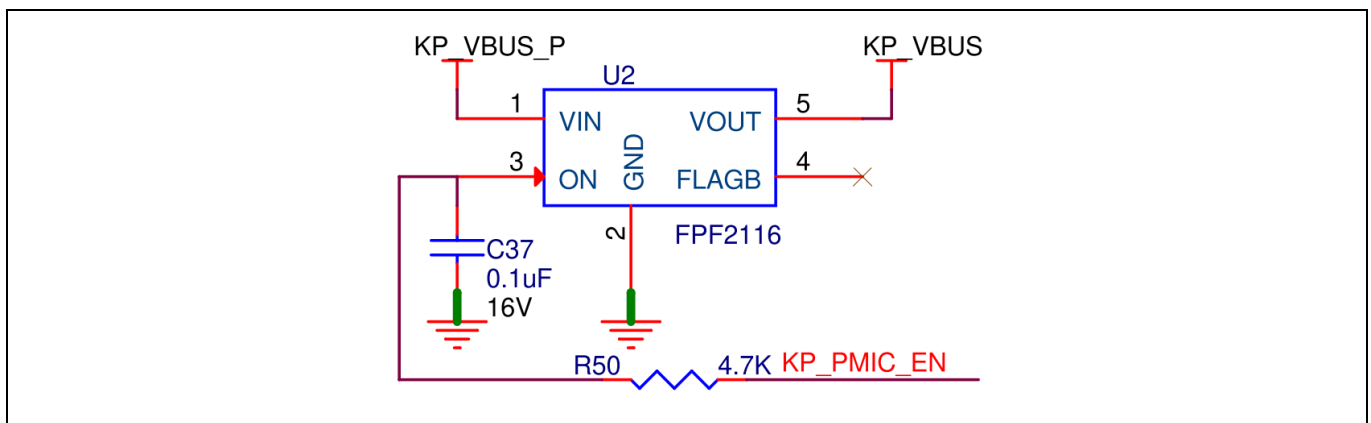


Figure 23 Power cycle load switch

3.2.5 Transient voltage protection diode

The reverse diodes on VTARG (D5) and KitProg3 VBUS (D4) lines protect the power rail from transient voltage from VBUS (up to 20 V) during power selection jumper switching.

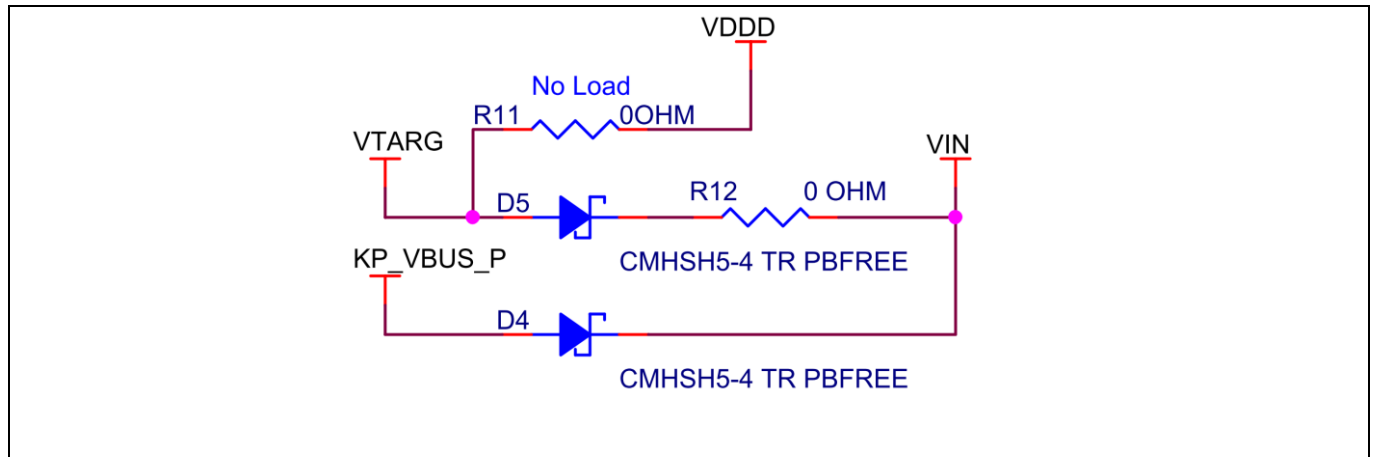


Figure 24 Transient voltage protection diode

3.3 EZ-PD™ PMG1 MCU subsystem

3.3.1 EZ-PD™ PMG1 MCU device

Visit the [webpage](#) or look at the device datasheets [4] to understand and compare the various features supported by the EZ-PD™ PMG1 MCU controllers.

3.3.2 User LED and button

The user LED (green) is an EZ-PD™ PMG1 MCU-driven LED (LED3); you can configure the function through firmware.

Table 11 lists the GPIO assigned for LEDs in each kit. The GPIOs need to be driven LOW to turn ON the user LED.

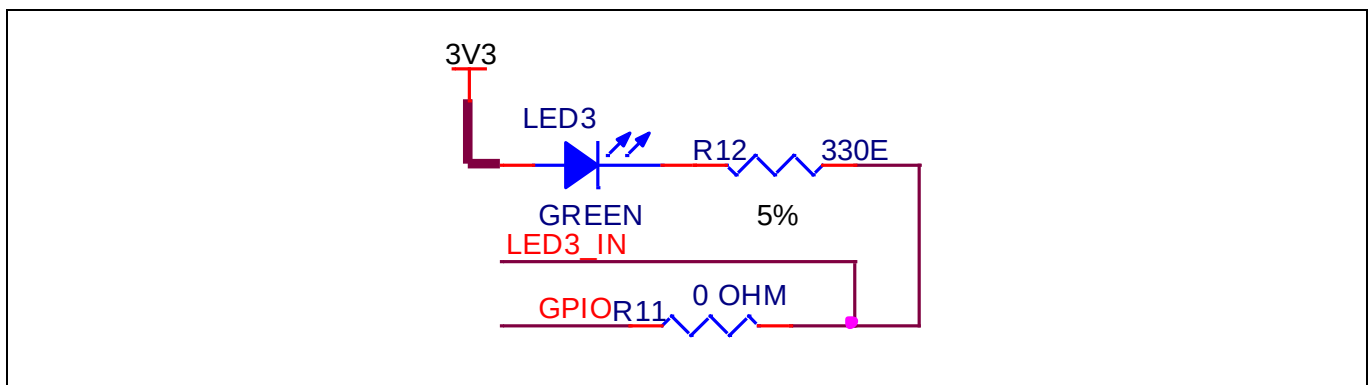


Figure 25 User LED

3.3.3 User button (SW2)

The user button is a tactile push button (SW2) on the kit; the functionality of the button is configurable through firmware. When the button is pressed, the GPIO connected to the button is pulled down to the ground.

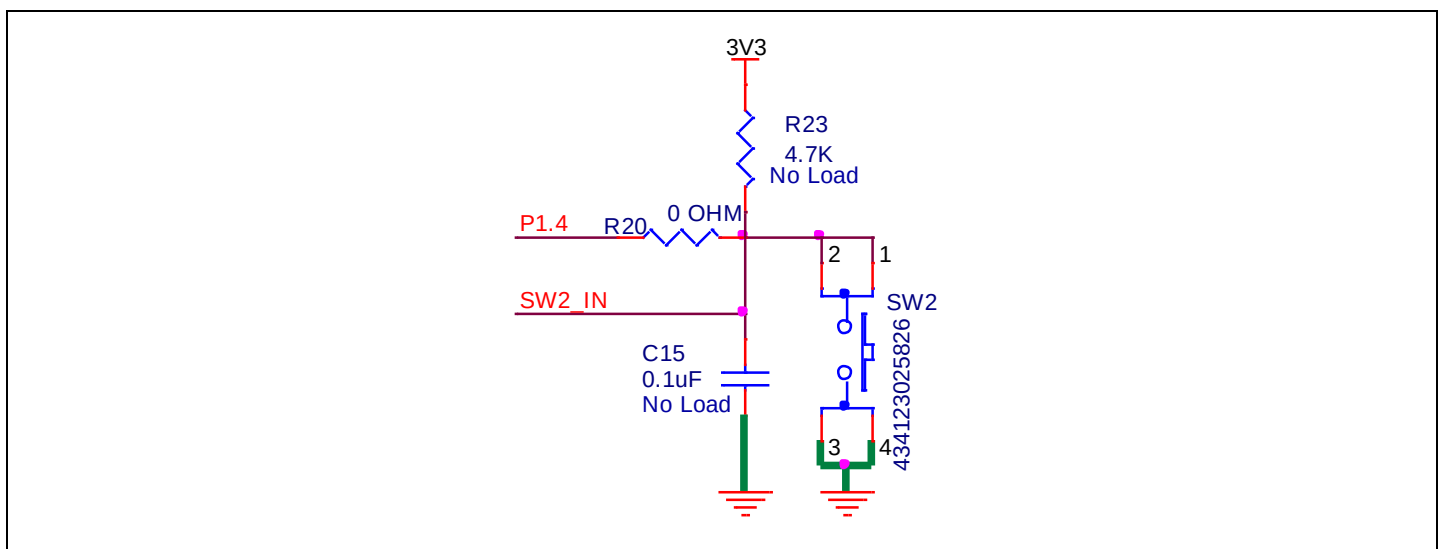


Figure 26 User button

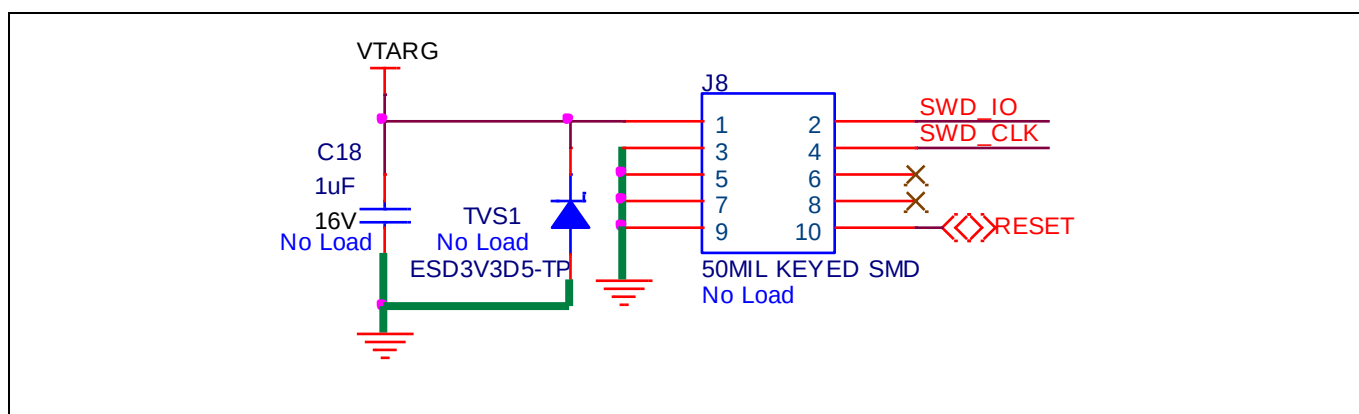
Table 11 GPIO details for user LEDs and user buttons

Kit	User LED GPIO	User button GPIO
CY7110	P2.1	P2.0
CY7111	P2.1	P2.0
CY7112	P1.3	P1.2
CY7113	P5.5	P3.3

3.3.4 10-pin SWD header

The 10-pin SWD/JTAG connector is a programming and debug interface connector. The header is pin-compatible with all standard 10-pin SWD/JTAG interfaces, and supports Infineon's MiniProg4 program and debug kit.

In CY7110, the reset signal is left unconnected because the PMG1-S0 device does not have a reset pin.

**Figure 27** 10-pin SWD header

3.3.5 MCU I/O header

PMG1 kits have two I/O headers; all GPIOs, PD-specific function signals, regulator output, and other power signals are routed to these headers. See the pin diagram of the I/O header in each kit explained in the [Prototyping kit pinout](#).

3.3.6 Reset button

CY7111 EZ-PD™ PMG1-S1 MCU, CY7112 EZ-PD™ PMG1-S2 MCU, and CY7113 EZ-PD™ PMG1-S3 MCU kits have reset buttons to manually reset the device. When the reset button is pressed, the XRES pin on the device will pull down to the ground to reset the device. The EZ-PD™ PMG1-S0 MCU device does not have a reset pin; therefore, the CY7110 device does not support a manual reset.

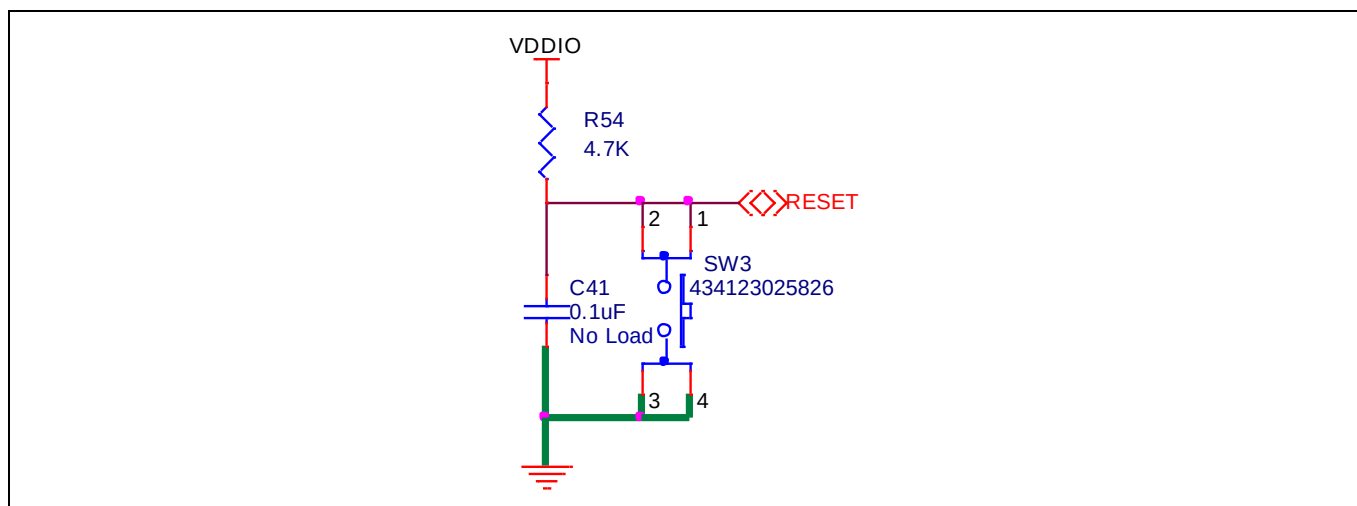


Figure 28 Reset circuit

3.3.7 CAPSENSE™ buttons and slider

The EZ-PD™ PMG1-S3 MCU supports Infineon's capacitive touch sensing, called CAPSENSE™. The CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit has two touch buttons and one 5-sensor slider implemented using the CAPSENSE™ touch sensing feature. CAPSENSE™ buttons BTN1 and BTN2 are designed to exercise a self-capacitance-based touch button implementation using. The CAPSENSE™ slider is implemented using five sensors and also uses self-capacitance sensing.

See the [Getting started with CAPSENSE™ design guide](#) for a detailed introduction to CAPSENSE™ and See [PSoC™ 4 and PSoC™ 6 MCU CAPSENSE™ design guide](#) for CAPSENSE™ design guidelines.

Each CAPSENSE™ sensor has an associated LED to indicate the status of the sensor. For the CAPSENSE™ button, the LED corresponding to the button turns ON when the button detects a touch. In the CAPSENSE™ slider, the LED corresponding to the sensor on which a finger is placed during a sliding action will turn ON. The table below lists the CAPSENSE™ button, LEDs, and corresponding GPIO.

Table 12 CAPSENSE™ buttons and slider

CAPSENSE™ widget	CAPSENSE™ sensor	Sensor GPIO	LED	LED GPIO
CAPSENSE™ button	BTN1	P2.1	LED5	P2.4
	BTN2	P2.2	LED6	P2.3
CAPSENSE™ slider	SLD1	P6.0	LED7	P5.3
	SLD2	P6.1	LED8	P1.4
	SLD3	P0.0	LED9	P1.5
	SLD4	P6.2	LED10	P1.6
	SLD5	P6.3	LED11	P1.3

CAPSENSE™ also requires an external capacitor CMOD for self-capacitance sensing. CAPSENSE™ sensor pads are connected to the EZ-PD™ PMG1-S3 MCU through 560 Ω series resistors placed close to the MCU GPIOs that are configured as CAPSENSE™ sensor pins. The CAPSENSE™ sensors on the kit are covered with a transparent overlay (insulator material) which serves as the touch surface for the sensor below it. The sensor pads are surrounded by grounded-shield electrodes and provide optional provisions for active driven shields which

EZ-PD™ PMG1 prototyping kit system design

provide liquid tolerance and proximity sensing support. The driven shield feature can be enabled on the kit by populating series resistors and removing the ground connections. The CTANK capacitor also needs to be connected for enabling the drive-in shield feature.

Table 13 **Component and associated feature**

Component	Associated feature	GPIO	LED GPIO
CMOD	Self-capacitance sensing	P5.1	2.2 nF capacitor connected to CMOD pin
CTANK	Driven shield, liquid tolerance, and proximity sensing	P5.0	10 nF capacitor connected to CTANK pin through “no load” component. Populate R19 to enable CTANK
SHIELD1, SHIELD2	Driven shield, liquid tolerance, and proximity sensing	P5.2	Shields are connected to the ground through resistors (R63, R80). Remove ground connection and populate series resistor (R66, R77) to enable driven shield

3.4 KitProg3 (PSoC™ 5LP MCU)

An onboard PSoC™ 5LP MCU (CY8C5868LTI-LP039)-based KitProg3 module is used to program and debug the EZ-PD™ PMG1 microcontroller.

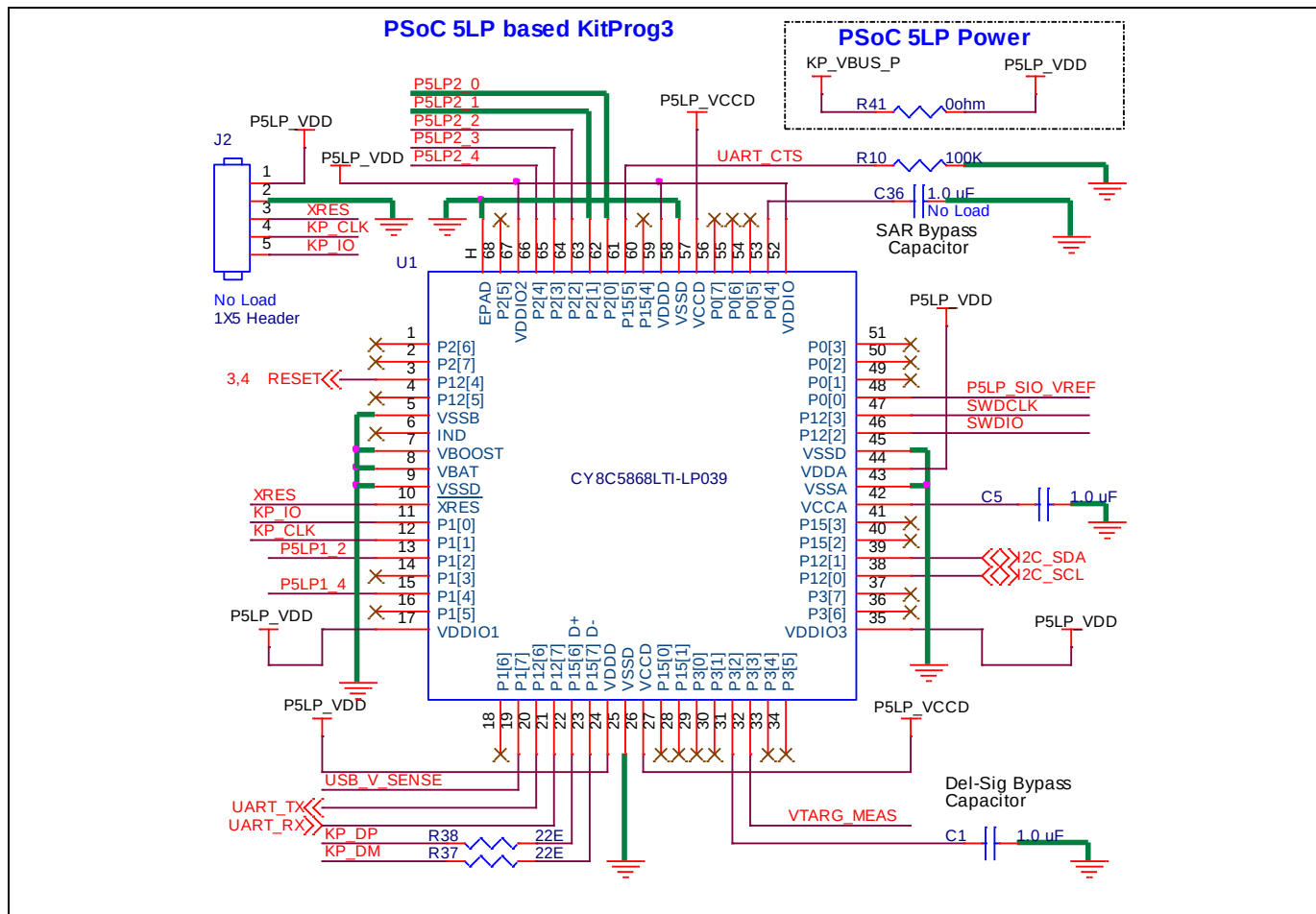


Figure 29 PSoC™ 5LP MCU device

The PSoC™ 5LP MCU device interfaces with a PC through a Type-C USB connector (J1), and functions as a bridge between the PC and EZ-PD™ PMG1 MCU devices over SWD, I2C, and UART interfaces. The KitProg3 module gets power through the J1 port and receives and transmits data between the host PC through D+/D- signals. The programming/debugging module can access the EZ-PD™ PMG1 MCU device in programming or debugging mode via the SWD header. In addition to being an onboard programmer, the KitProg3 functions as an interface for the USB-I2C and USB-UART bridges.

The USB-Serial pins of PSoC™ 5LP MCU are hard-wired to the I2C pins of the EZ-PD™ PMG1 MCU, and these pins are also available on the KitProg3 headers (J3 and J4).

The USB-UART bridge functionality is enabled by default by hard-wired connections of the UART lines between KitProg3 and EZ-PD™ PMG1 MCU. However, the following older revisions of the PMG1 kit boards require manual external wire connection:

- CY7110 board revision 3 or lower
- CY7111 board revision 2 or lower
- CY7112 board revision 2 or lower
- CY7113 board revision 3 or lower

EZ-PD™ PMG1 prototyping kit system design

The UART lines from the EZ-PD™ PMG1 MCU are routed to the MCU I/O header (J6) and UART lines from KitProg3 are connected to the KitProg3 header (J3). Do the following to establish the UART connection on older revisions of the PMG1 kit boards as listed above:

1. Connect J6.10 (EZ-PD™ PMG1 MCU UART Tx) to J3.8 (KitProg3 UART Rx).
2. Connect J6.9 (EZ-PD™ PMG1 MCU UART Rx) to J3.10 (KitProg3 UART Tx).

3.4.1 Mode button

The KitProg3 mode button on the EZ-PD™ PMG1 MCU family kit enables the KitProg3 module to enter bootloader mode. The bootloader mode is required to update the KitProg3 firmware on PSoC™ 5LP MCU when the existing firmware is corrupted or a newer version is available.

3.4.2 KitProg3 power and status LEDs

The KitProg3 power LED (amber) turns ON when the KitProg3 module is supplying power to the target MCU. The LED will always be ON in a fault-free condition when the kit is powered through the programming connector.

The status LED (amber) indicates the KitProg3 programming mode and the programming status. See the KitProg3 user guide [\[6\]](#) for details.

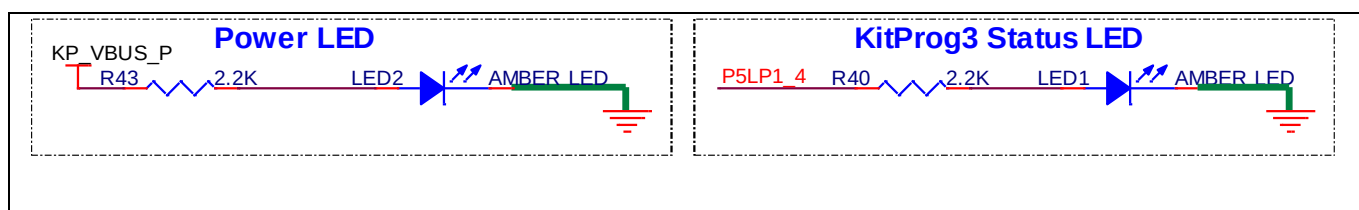


Figure 30 KitProg3 power and status LED

3.5 Target voltage measurement

The KitProg3 module is required to monitor the voltage at which the target MCU is working. The measured target voltage is used as the reference to configure the logic level for communication between the KitProg3 and the EZ-PD™ PMG1 MCU device.

3.5.1 Power enable

Power enable (Power EN) is a GPIO control signal for driving the power-cycle programming load switch in the CY7110 EZ-PD™ PMG1-S0 MCU prototyping kit.

3.6 Load switch

3.6.1 USB PD sink port

For the power aspects of the USB PD sink Type-C connector on the kit, see the [Power section](#). The CC1/CC2 and D+/D- lines are routed from the Type-C connector to the EZ-PD™ PMG1 MCU. The USB PD contract negotiation, charger detection, and data communication occur through these signal lines. These signals are susceptible to ESD events; therefore, additional ESD diodes are added to the traces.

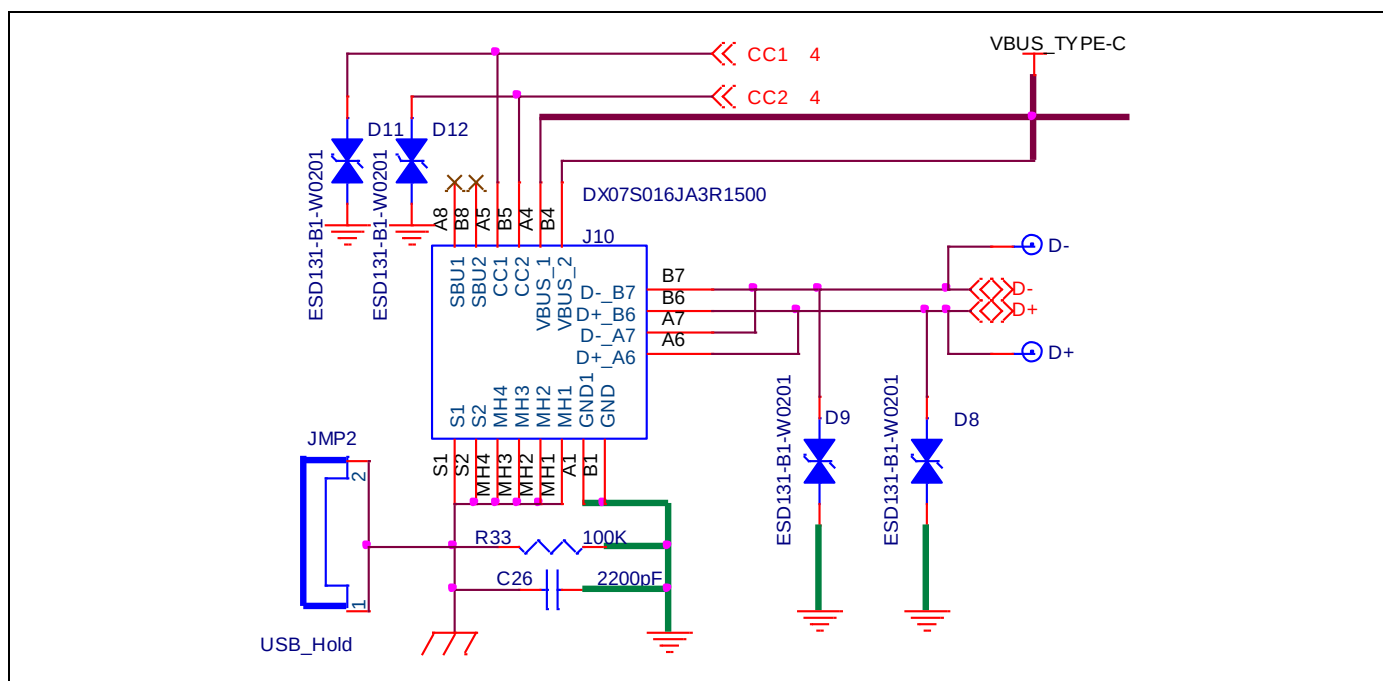


Figure 31 USB PD sink Type-C port

The USB PD voltage negotiated with the USB PD source can be measured from the VBUS_TYPE-C test point (red) located on the PMG1 kit board at any time.

In the CY7111 EZ-PD™ PMG1-S1 MCU kit, D+ bottom, D+ top, D- bottom, and D- top signals are routed to the device in separate traces, but in CY7110, and CY7112 kits, the top and bottom D+/D- signals are shorted near the Type-C connector. In PMG1-S1, there is an internal USB 2.0 analog mux present which routes the system D+ and D- lines to the Type-C top or bottom port based on the CC (Type-C plug) orientation. See the EZ-PD™ PMG1-S1 MCU datasheet [\[4\]](#) for details.

3.6.2 5-A load switch

CY7110, CY7111 and CY7112 kits support up to 100 W (20 V/5 A) USB PD contracts through the USB PD sink port. The load switch supports maximum contract voltage (20 V) and maximum contract current (5 A).

CY7113 kit can support up to 140 W (28 V/5 A) USB PD contracts and the load switch on the CY113 kit can support this power.

3.6.2.1 P-FET load switch

CY7110 and CY7111 kits have dual-P channel MOSFET as the load switch because the EZ-PD™ PMG1-S0 MCU and EZ-PD™ PMG1-S1 MCU devices have integrated PFET gate drivers to drive external PFETs. IRF9358 is a dual P-

EZ-PD™ PMG1 prototyping kit system design

channel MOSFET from Infineon with a maximum voltage and current rating of 30 V (VDSS) and 9.2 A (ID), suitable for the USB PD power rating supported in the kits.

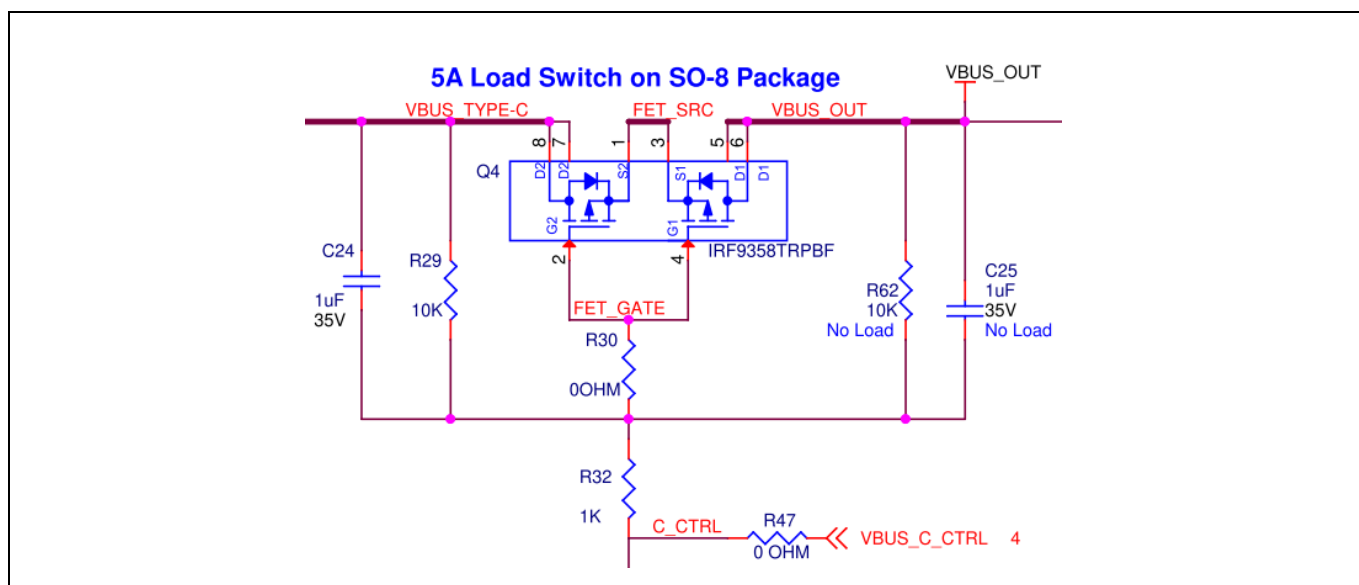


Figure 32 P-MOSFET load switch

3.6.2.2 N-FET load switch

In the EZ-PD™ PMG1-S2 MCU device, the gate drivers can be configured to support both P- and N-type external power FETs. The gate drivers are configured by default for NFET devices. CY7112 EZ-PD™ PMG1-S2 MCU kit has an IRF7907 dual N-channel MOSFET from Infineon as the load switch, with maximum voltage and current rating of 30 V (VDSS) and 9.1 A (ID).

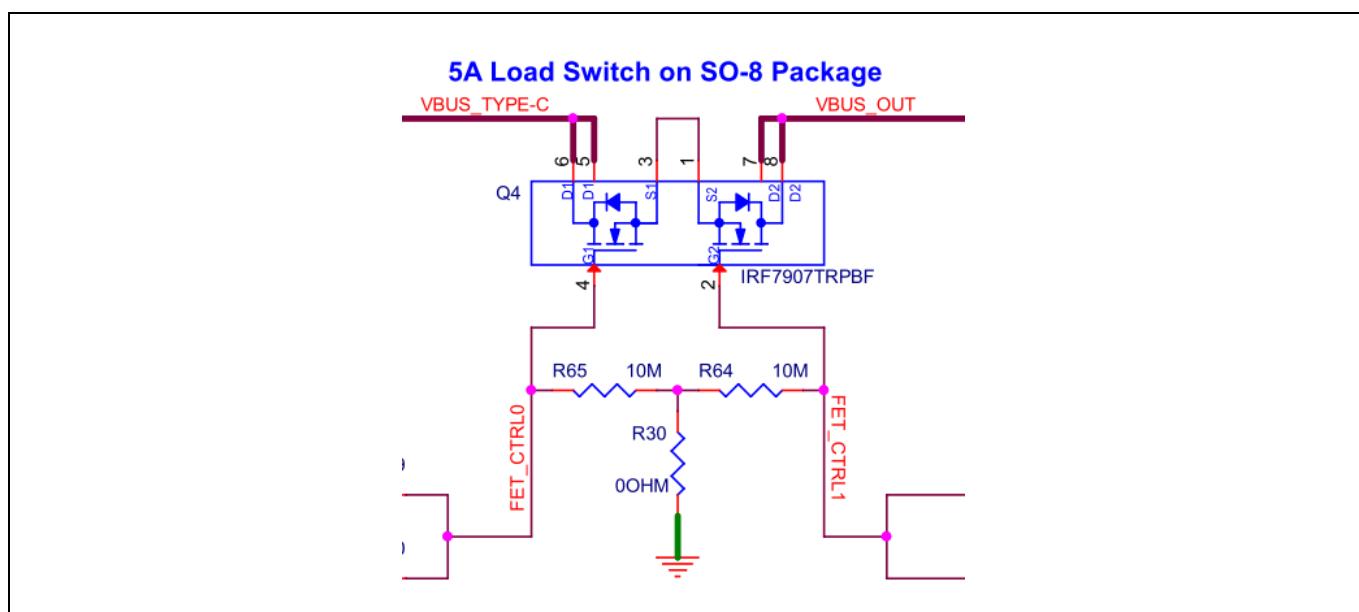


Figure 33 N-MOSFET load switch

CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit is designed to support up to 140 W (28 V, 5 A) operation. In the CY7113 EZ-PD™ PMG1-S3 MCU kit, the load switch circuit is designed using two BSC059N04LS N-Channel MOSFETs from Infineon, with maximum voltage and current rating of 40 V (VDS) and 73 A (ID).

3.6.3 DC_OUT terminal block

DC_OUT terminal block (J9) is a screw connector connected to the output of the load switch. The USB PD contract established on the EZ-PD™ PMG1 MCU USB PD sink port is available at the screw terminal when the load switch is ON; you can then connect it to an external load or circuit.

Kit operation

4 Kit operation

This section explains the operation, programming, and debugging modes of the kit. It also describes, in detail, the procedure for programming and debugging the application firmware on the kit.

4.1 USB PD sink operation

The prototyping kit is designed to work as a USB PD sink. The firmware associated with the USB PD sink functionality is available as a code example in ModusToolbox™ software. Firmware from that code example is pre-loaded in the factory on these kits. CY7110, CY7111 and CY7112 kits support up to a maximum contract of 100 W (20 V, 5 A) on the EZ-PD™ PMG1 MCU USB PD sink port. CY7113 kit supports up to a maximum contract of 140 W (28V, 5 A).

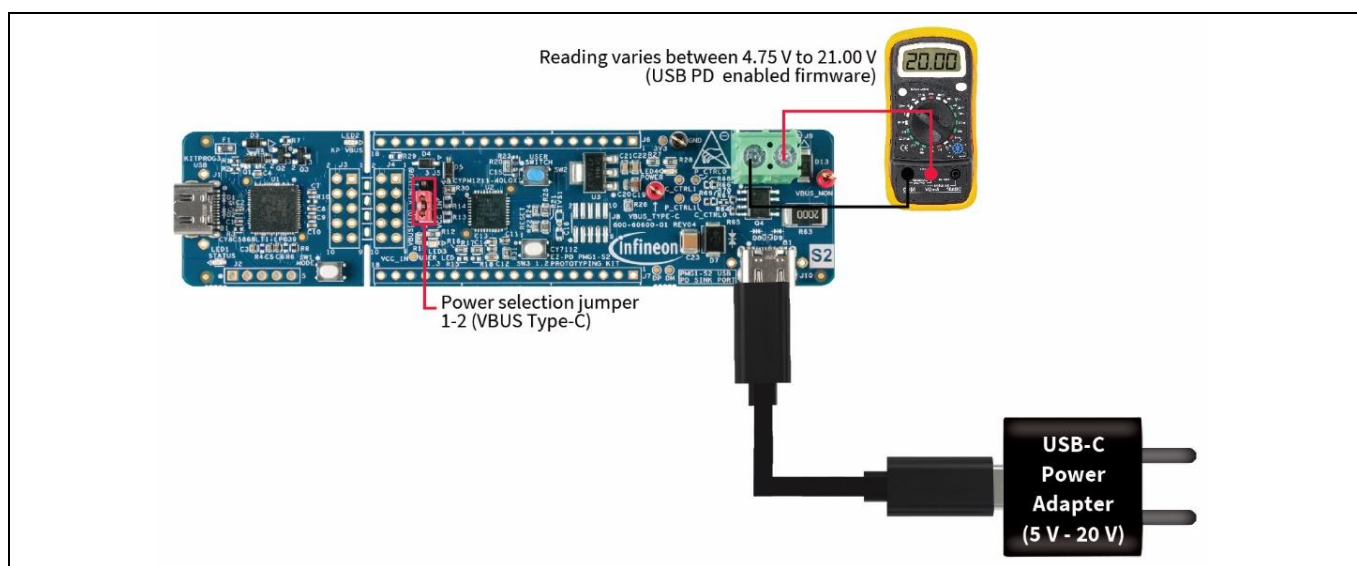


Figure 34 Prototyping kit in operation mode (USB PD sink-enabled firmware)

Do the following to run the USB PD sink firmware which is loaded in the factory on the CY711x EZ-PD™ PMG1 MCU prototyping kits:

1. Put the kit in operational mode by placing a jumper shunt on the power selection jumper (J5) at positions 1-2 to select the USB-C power adapter as the power source for the CY711x board.
2. Connect the USB PD sink port (J10) of the CY711x board to the USB-C power adapter using the USB Type-C cable.
3. Confirm that the power LED (LED4) glows green and the user LED (LED3) blinks green.
4. Measure the DC_OUT voltage by connecting a multimeter to the terminal block (J9). Confirm that the DC_OUT voltage value is within the 4.75 V–21.00 V range. The actual value is determined by the maximum voltage which the USB-C power adapter can supply.
5. If desired, disconnect power, remove the multimeter, connect an external load to the terminal block (J9), and reconnect power to verify powering an external load.

Attention: *The maximum current that can be consumed by an external load is limited to 5 A by the USB PD specification. If the load exceeds 5 A, the USB-C power adapter may trigger overcurrent protection (OCP) and cut off the power supply to the kit. This behavior is dependent on the implementation of the USB-C power adapter.*

Kit operation

See [Table 14](#) for troubleshooting information.

Note: The default USB PD sink firmware can be replaced by new firmware as part of the application development process. See [Application development on CY711x kits using ModusToolbox™ software](#) for more details.

The CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit firmware support CAPSENSE™ sensing in addition to USB PD sink operation. The procedure for exercising CAPSENSE™ operation is as follows:

1. The initial setup is the same to run USB PD sink firmware. Follow Steps 1 to 5.
2. The CAPSENSE™ buttons are exercised by touching each button and observing the corresponding feedback LED behavior. When BTN1 is touched, LED5 will turn ON and for BTN2, LED6 will turn ON. The LEDs turn OFF when corresponding sensors are not touched.
3. For testing the CAPSENSE™ slider, place your finger on SLD1 and swipe from left to right. The feedback LEDs will glow in a sequence (LED7 → LED8 → LED9 → LED10 → LED11). Then swipe from right to left, and LEDs will glow in reverse sequence (LED11 → LED10 → LED9 → LED8 → LED7).

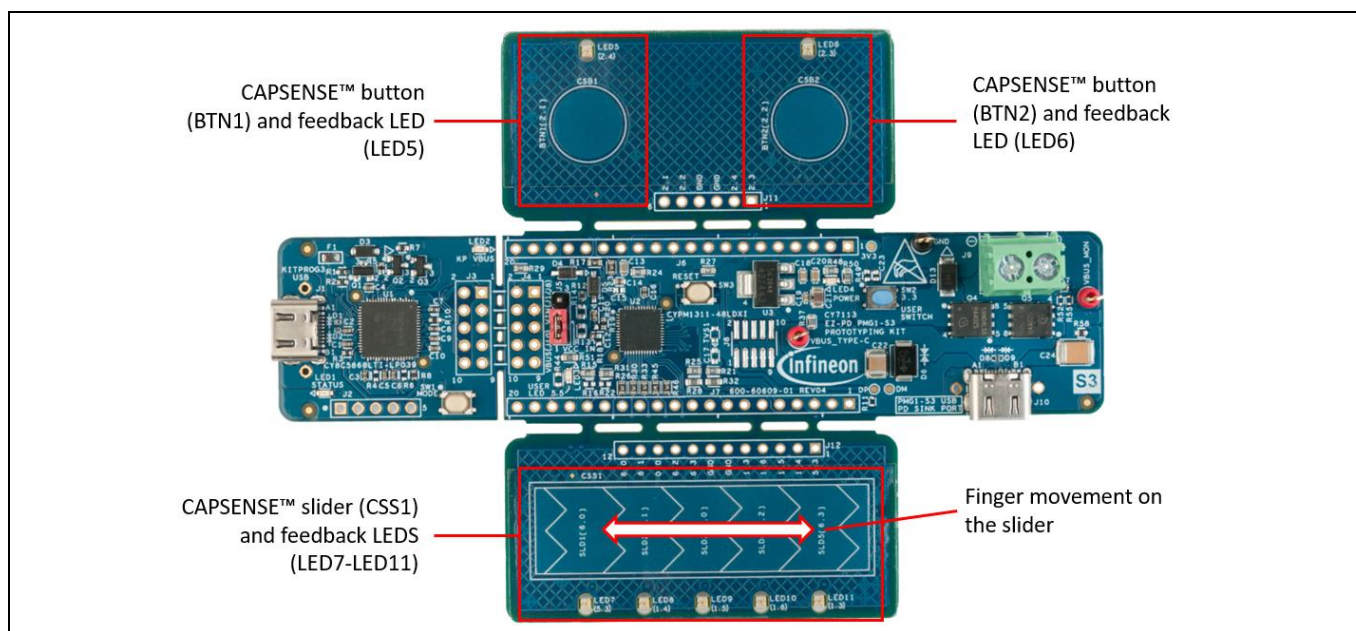


Figure 35 CAPSENSE™ buttons and slider

Kit operation

Table 14 Troubleshooting the kit

#	Issue	Possible cause	Fix
1	User LED (LED3) does not blink green	The kit is not configured for operational mode.	Ensure that the jumper shunt on the power selection jumper (J5) is placed at positions 1–2.
		The kit may not be running the default USB PD sink firmware.	Check whether the application code associated with the firmware downloaded on the kit implements blinking LED logic.
2	No voltage is observed at the terminal block (J9)	The firmware downloaded on the kit may not implement USB PD functionality.	Check whether the application code associated with the firmware downloaded on the kit implements the USB PD logic.
		The USB PD contract may not have been successfully established with the connected USB-C power adapter.	Check whether the USB-C power adapter connected to the USB PD sink port (J10) is compliant with the USB PD 3.0 specification.

5 Application development on CY711x kits using ModusToolbox™ software

5.1 Kit code examples

The following kit code examples are available in ModusToolbox™ software to exercise the kit functionality and as a reference for new application development.

Table 15 Example projects in ModusToolbox™ software

Example project	Description	Kits supporting the example project
Hello World	This code example demonstrates simple UART communication by printing a “Hello World” message on a terminal and blinking the User LED (LED3) using the EZ-PD™ PMG1 MCU.	CY7110, CY7111, CY7112, CY7113
USBPD Sink	This code example demonstrates USB-C attach detection and USB PD contract negotiation using the EZ-PD™ PMG1 MCU. This is the default firmware loaded on CY7110, CY7111 and CY7112 kits.	CY7110, CY7111, CY7112
Basic Device Firmware Upgrade	This code example demonstrates a device firmware upgrade (DFU) on an EZ-PD™ PMG1 MCU.	CY7110, CY7111, CY7112, CY7113
USB HID Mouse	This code example demonstrates how to configure the USB block in an EZ-PD™ PMG1 MCU as a human interface device (HID). The device enumerates as a 3-button mouse.	CY7112, CY7113
USBPD Sink CAPSENSE™	This code example demonstrates CAPSENSE™ functionality using a five-segment CAPSENSE™ slider and two CAPSENSE™ buttons along with USB-C attach detection and USB power delivery contract negotiation using EZ-PD™ PMG1-S3 MCU. This is the default firmware loaded on the CY7113 kit.	CY7113

A detailed explanation of the example projects is available in the ModusToolbox™ software.

5.2 Using the kit code example

1. Open Eclipse IDE for ModusToolbox™ software.
2. Navigate to **quick panel** and click **New application** under **Start**.

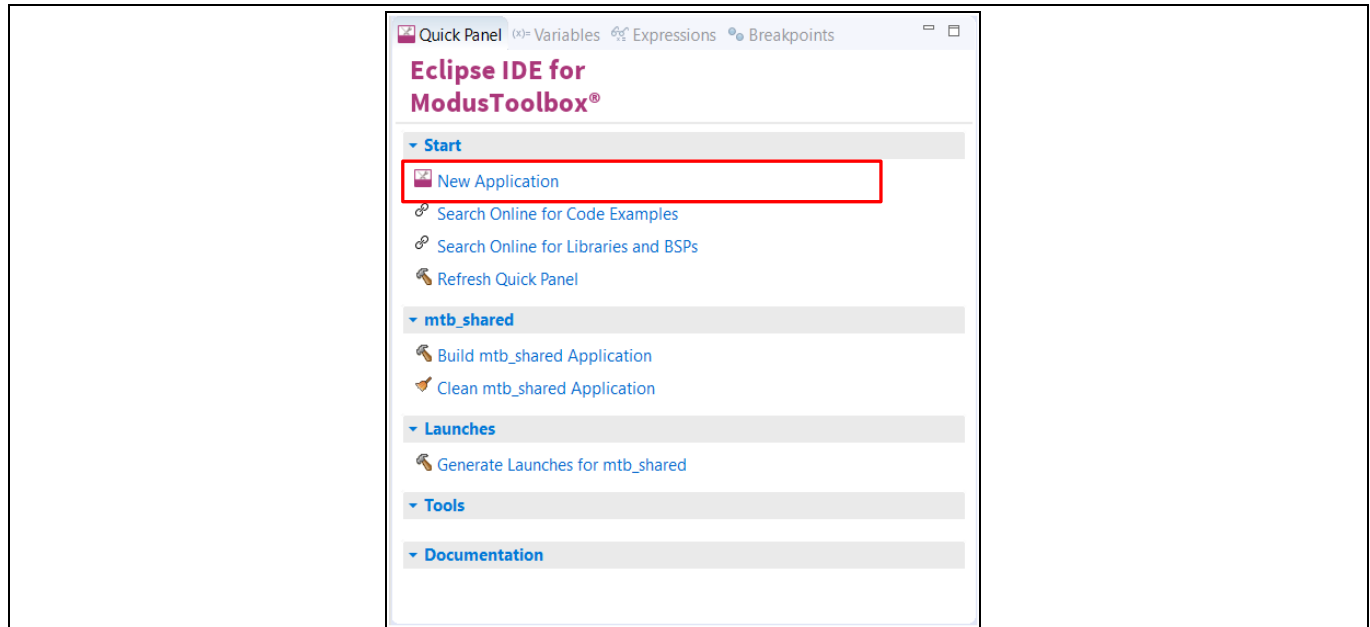


Figure 36 Creating a new application

3. In the **Project Creator** window, choose the board support package (BSP) from the list. Click **Next**.

Note: The PMG1 prototyping kits are listed under **PMG1 BSPs**.

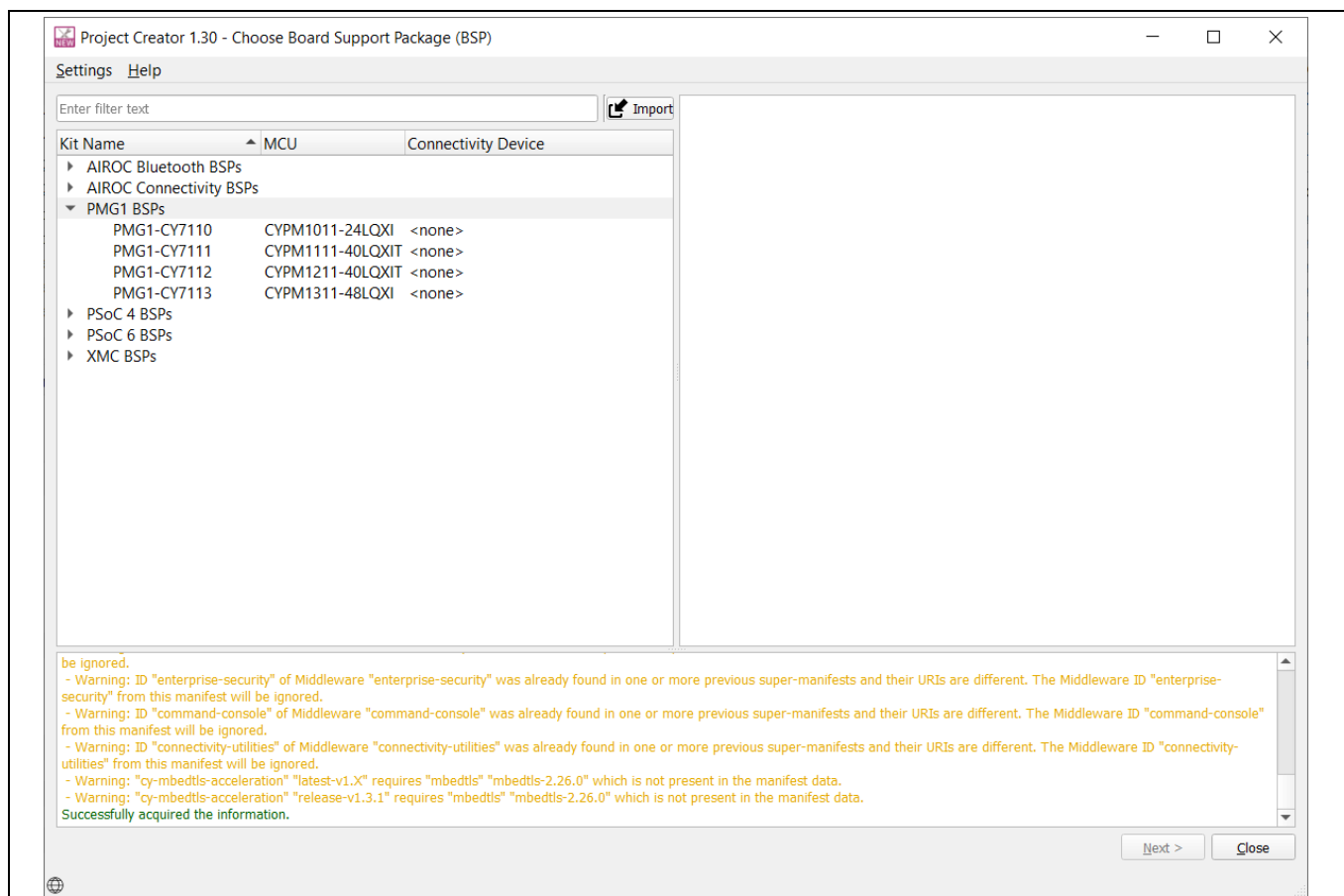


Figure 37 Selecting the BSP

4. Select the desired example project. Enter the **New application name** if you want to change it from the default. Click **Create**.

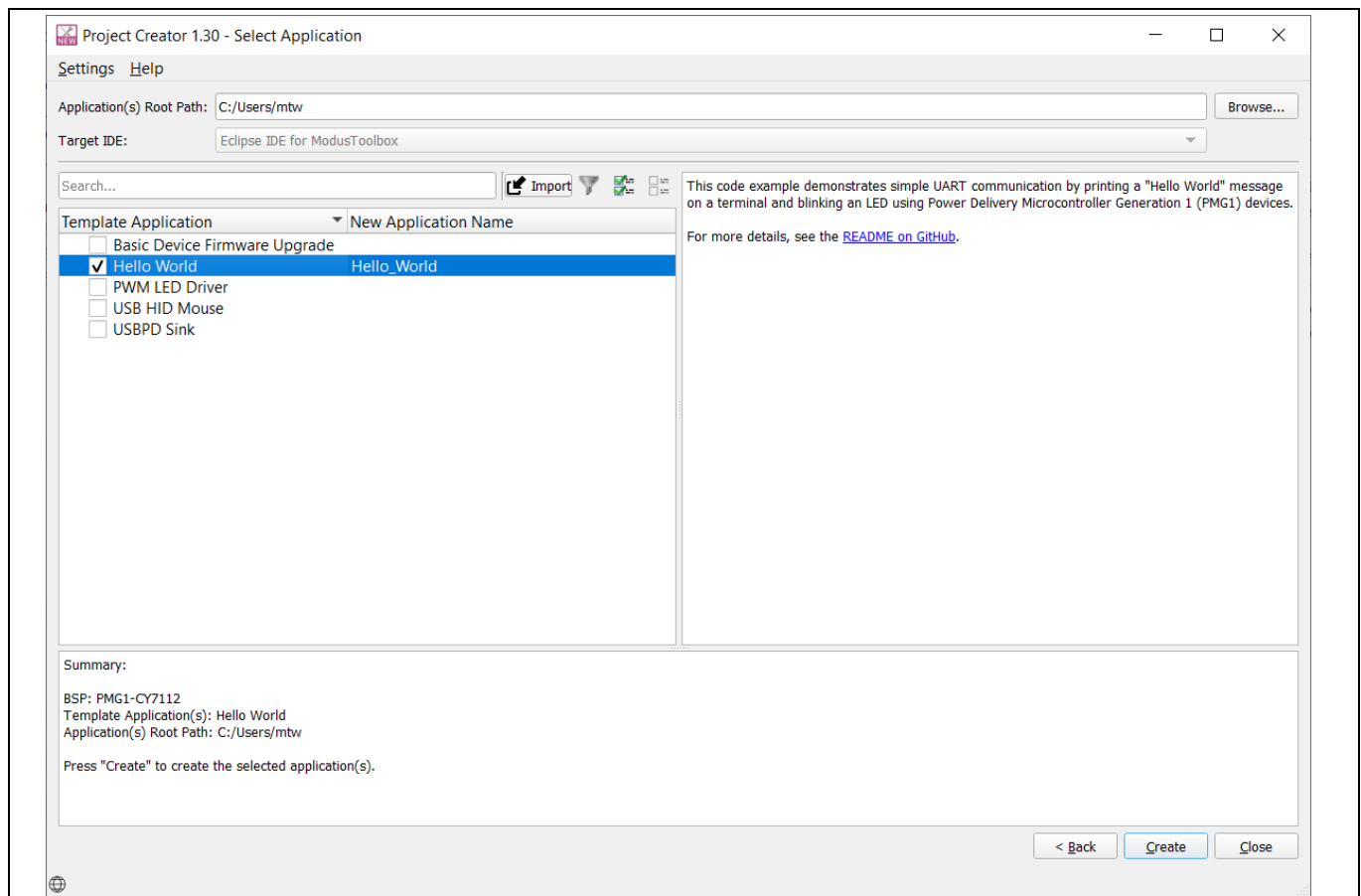


Figure 38 **Selecting the application**

ModusToolbox™ software downloads the resources required from GitHub for creating the project and automatically closes the project creator window when the project creation is completed.

Note: *Ensure that the PC is connected to the internet to access the GitHub repository.*

See the application note [5] to learn more about the procedure to create new or modify existing applications, the procedure to configure applications, and the procedure to build applications.

5.3 **Programming and debugging using the KitProg3 interface**

5.3.1 **Programming using ModusToolbox™ software**

KitProg3 is a PSoC™ 5LP MCU-based onboard programming and debugging solution integrated with the EZ-PD™ PMG1 MCU prototyping kits. You can program and debug the kit without using any separate programming and debug module by setting the power selection jumper to short pins 2 and 3 and then connecting the kit to the PC using the programming connector as shown in [Figure 39](#).

KitProg3 supports two modes for acquiring the device for programming/debugging:

- **Power cycle mode:** In the CY7110 EZ-PD™ PMG1-S0 MCU prototyping kit, KitProg3 acquires the device by toggling the target (EZ-PD™ PMG1-S0 MCU) power in a sequence.
- **Reset mode:** In all other kits, KitProg3 acquires the device by controlling the reset signal to the target MCU (EZ-PD™ PMG1 MCU)

The KitProg3 programmer chooses the acquisition mode depending on the target MCU and prototyping kit.

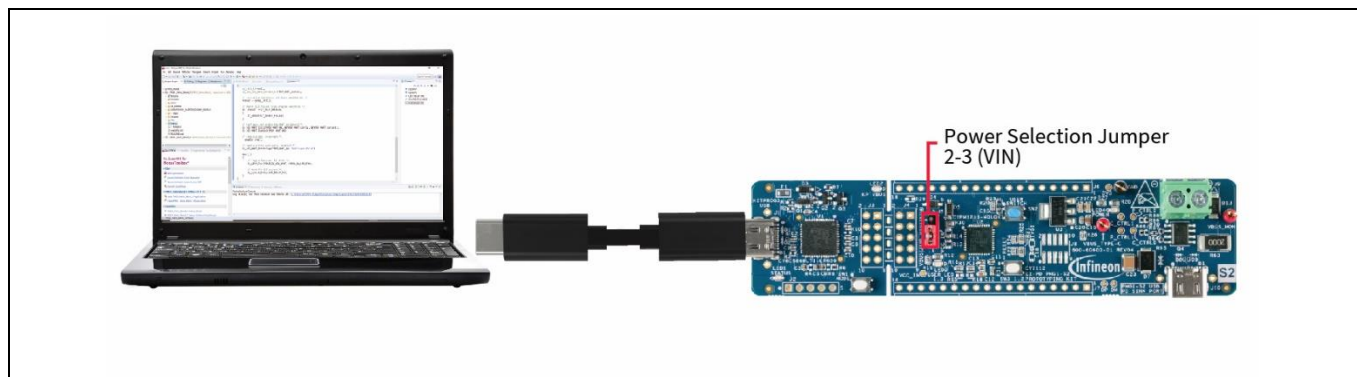


Figure 39 Programming through the KitProg3 interface

Do the following to program the prototyping kit using the KitProg3 interface from the ModusToolbox™ software:

1. Open the Eclipse IDE for ModusToolbox™ software on the host PC. For details, see the Eclipse IDE for ModusToolbox™ software user guide [\[8\]](#).
2. Connect the power section jumper (J5) to position 2-3 (VIN).
3. Connect the kit to the host PC through KitProg3 USB Type-C port (J1).
4. Ensure that LED1 and LED2 glow amber.
LED2 (KitProg3 Power LED) indicates that the KitProg3 module and target MCU are powered. LED1 (status LED) indicates the programming mode and status and is ON when KitProg3 is powered.

5. In the project explorer, select the correct project as the current active project as shown in **Figure 40**.

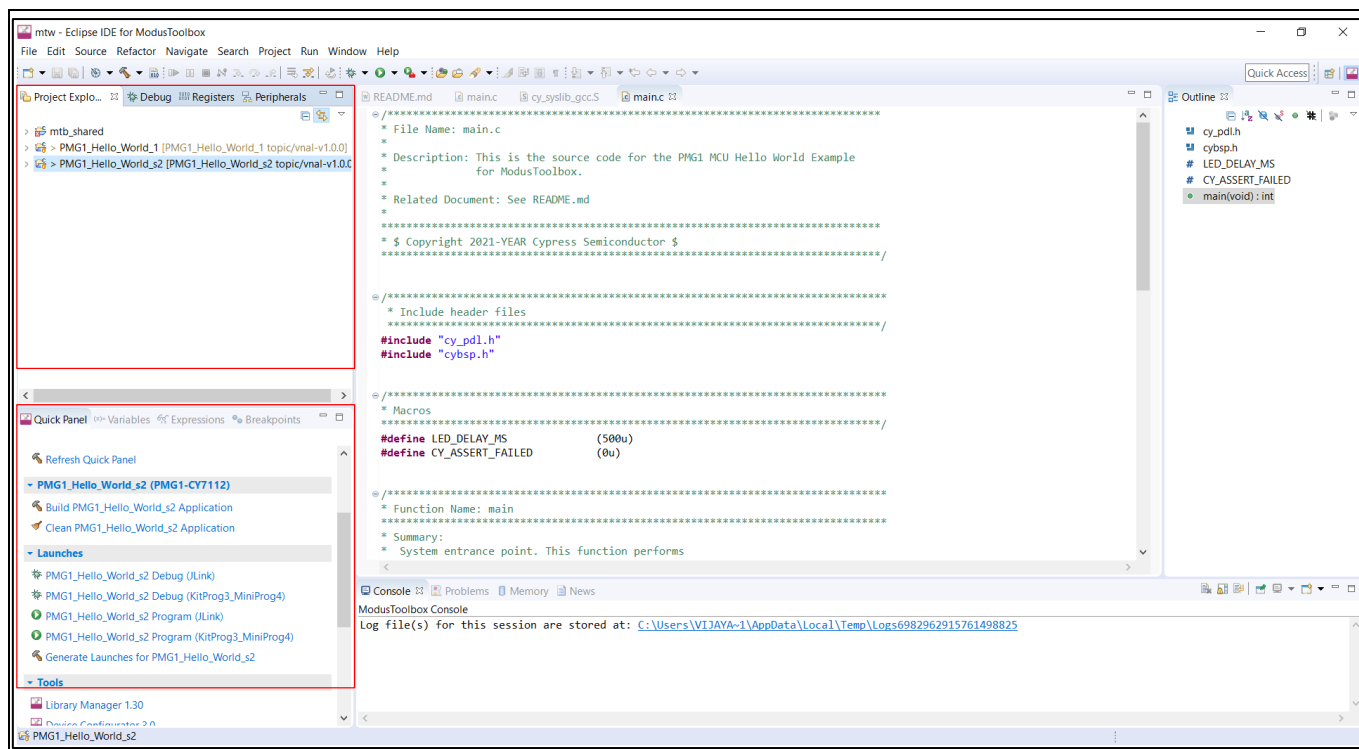


Figure 40 Eclipse IDE for ModusToolbox™ software

6. On the **Quick Panel** tab, click **<Application name> Program (KitProg3_MiniProg4)** from the **Launches** section.

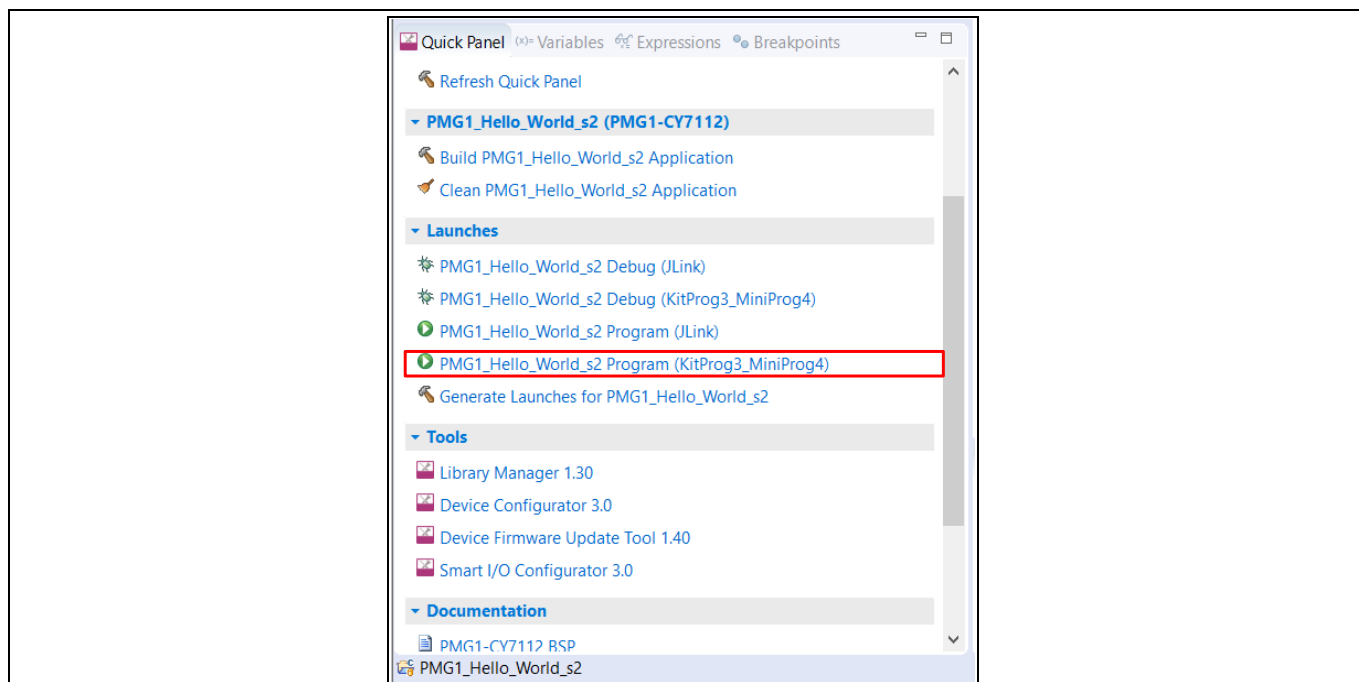


Figure 41 Quick Panel and programming options

Application development on CY711x kits using ModusToolbox™ software

The **Console** tab displays the progress of the build/program. You can check the **Problems** tab for any errors and warnings. The success message is displayed on the Console tab if the programming is successfully completed.

5.3.2 Debugging using ModusToolbox™ software

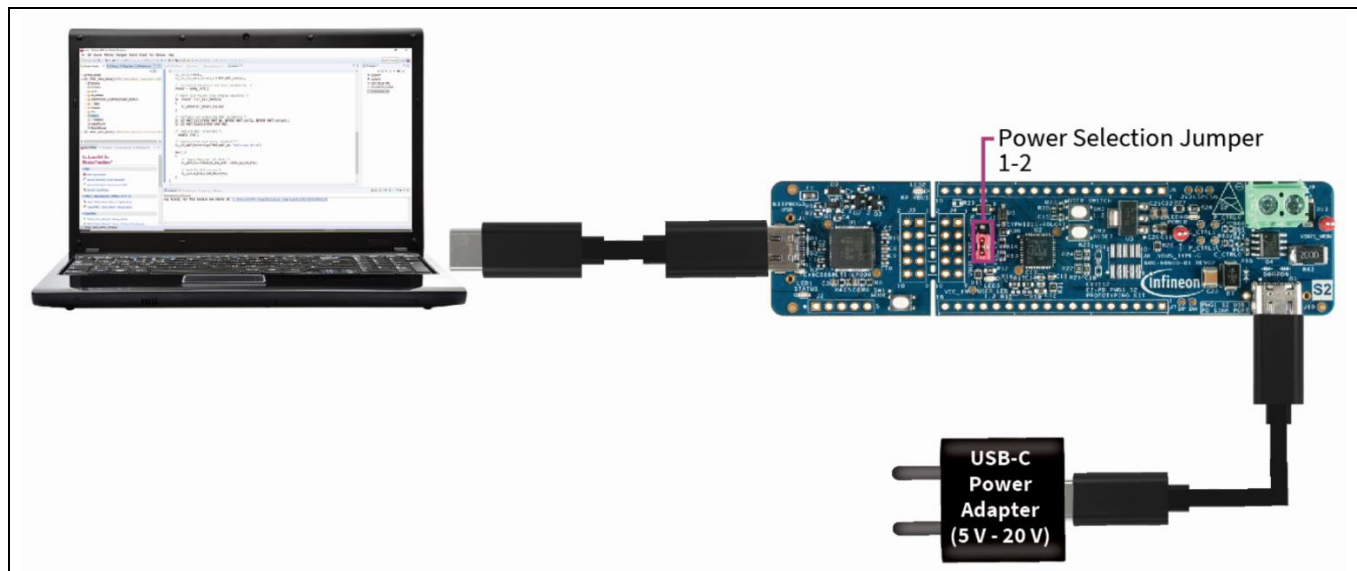


Figure 42 Debugging through KitProg3

Do the following to debug an application on the CY7111, CY7112, and CY7113 prototyping kits over the KitProg3 interface using the Eclipse IDE for ModusToolbox™ software.

1. Open the Eclipse IDE for ModusToolbox™ software on the host PC and select the PMG1 project. For details, see the Eclipse IDE for ModusToolbox™ software user guide [8].
2. Place the jumper shunt on pins 1–2 of the power section jumper (J5) to configure the kit in operational mode.
3. Connect the USB PD sink port to the USB PD source to activate the onboard LDO, Load switch, and user LED. Ensure that LED4, which is the power LED, glows green.
4. Connect the kit to the host PC through programming KitProg3 USB Type-C port. Ensure that LED1 and LED2 glow amber. LED2, which is KitProg3 Power LED, indicates that the KitProg3 module is powered. LED1, which is the status LED, indicates the programming/debug mode and status, and is ON when KitProg3 is powered.
5. On the **Quick Panel**, click **<Application name> Debug (KitProg3_MiniProg4)** from the **Launches** section.

The IDE will switch to debugging mode and will halt at the first line of the main function. This indicates that the application is ready for debugging.

Do the following to debug an application on the CY7110 prototyping kits over the KitProg3 interface using the Eclipse IDE for ModusToolbox™ software.

1. Open the Eclipse IDE for ModusToolbox™ software and select the **PMG1** project on the host PC. For details, see the Eclipse IDE for ModusToolbox™ software user guide [8].
2. Place the power selection jumper (J5) shunt on pins 2–3 of the power selection jumper (J5) to enable ModusToolbox™ software to acquire the target (EZ-PD™ PMG1-S0 MCU) in power cycle mode.
3. Connect the USB PD sink port to the USB PD source to activate onboard LDO, Load switch, and user LED. Ensure that LED4 glows green.

Application development on CY711x kits using ModusToolbox™ software

4. Connect the kit to the host PC through the KitProg3 USB Type-C Port. Ensure that LED1 and LED2 glow amber. LED2, which is KitProg3 power LED, indicates that the KitProg3 module is powered. LED1, which is the status LED, indicates the programming/debug mode and status, and is ON when KitProg3 is powered.
5. On the **quick panel**, click **<Application name> Program (KitProg3_MiniProg4)** from the **Launches** section to download the firmware into the kit.
6. After program download completes, change the jumper to short 1–2 of the power section jumper (J5) to configure the kit in the operational mode for debugging.
7. Select **Run > Debug configurations**, as shown in [Figure 43](#).
8. Under GDB OpenOCD Debugging, select **<Application name> Attach (KitProg3_MiniProg4)** and click on **Debug** as shown in [Figure 44](#).

This will start a debugging session attaching to a running target without programming or reset. For more information on debug configurations, see the MTB user guide [\[7\]](#).

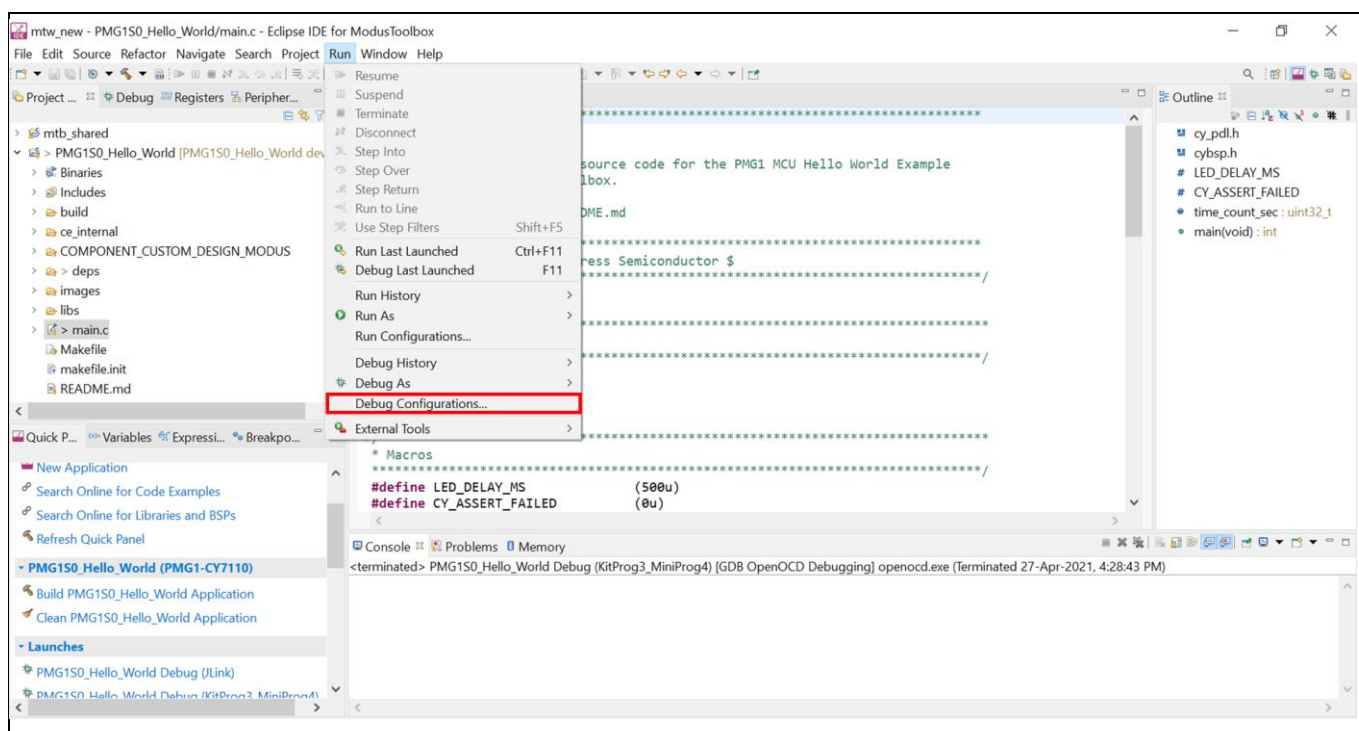


Figure 43 Selecting debug configurations from the Run menu

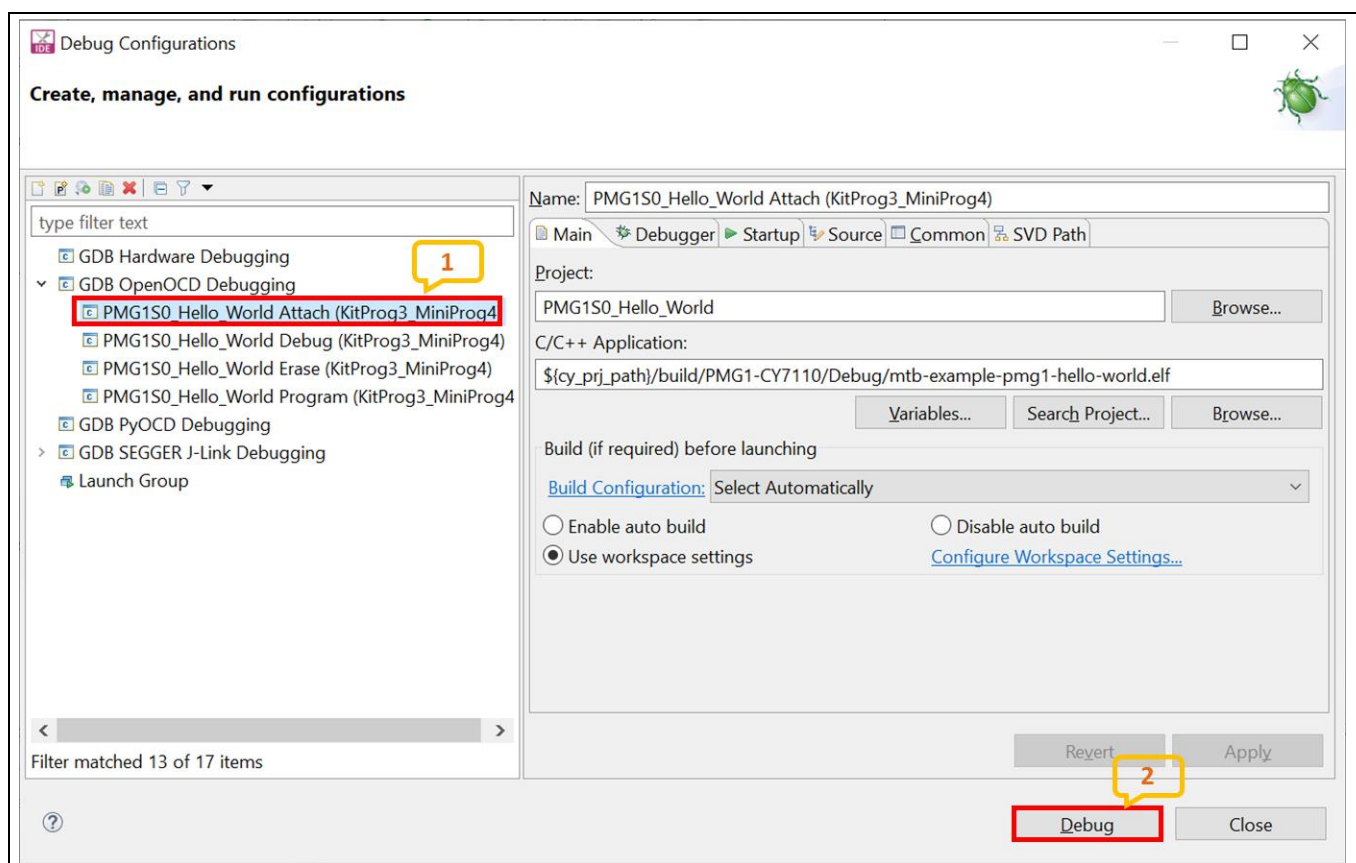


Figure 44 Selecting debug attach and starting a debug session

9. ModusToolbox™ software acquires the kit in debug mode and the debug tools appear on the toolbar.

5.3.3 Programming using CYPRESS™ programmer (CYP)

CYP v4.0 or higher supports programming of the EZ-PD™ PMG1 MCU prototyping kits and EZ-PD™ PMG1 MCU devices. Do the following to program the prototyping kit using the KitProg3 interface from the CYP.

1. Connect the power section jumper (J5) to position 2–3 (VIN).
2. Connect the prototyping kit to the host PC through the KitProg3 connector. See [Figure 39](#) for the setup.
3. Launch CYP. See the user guide [\[10\]](#) for the detailed procedure to download and install CYP.
4. From the **Probe/kits** drop-down list, select the connected prototyping kit. Based on the selection, the **Platform** will display PMG1 and the **Log** panel will display the selected device/kit/probe name.

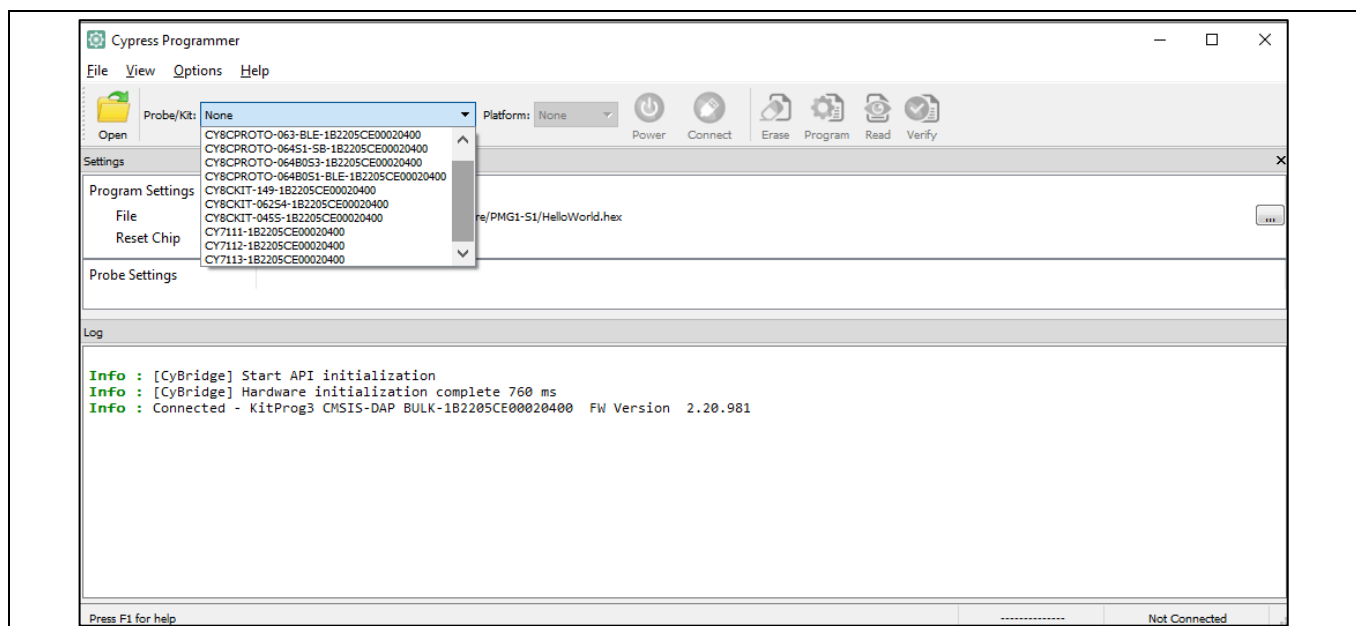


Figure 45 Selecting the probe/kits

- Click **Open** to load the programming file. In the **Open programming file** dialog, browse to the location of the HEX, SREC, ELF, or BIN file to be loaded. Select the file and click **Open**.

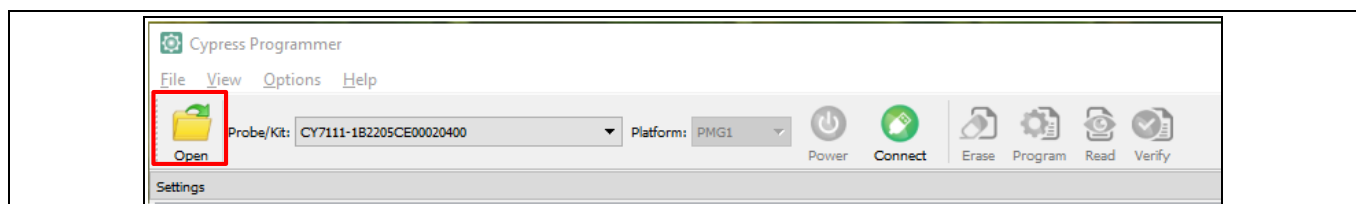


Figure 46 Loading the programming file

- Click **Connect**. CYP connects to the target board and the **Erase**, **Program**, **Read**, and **Verify** options appear.

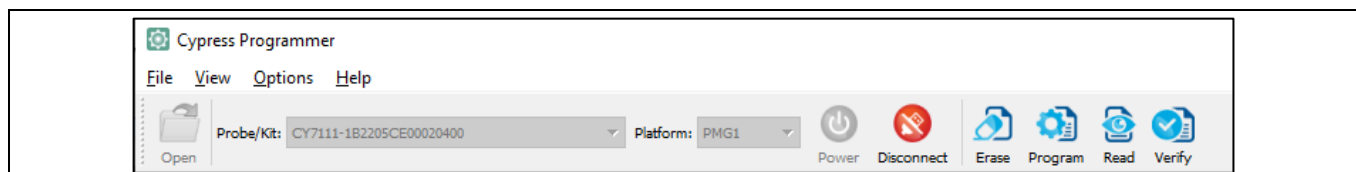


Figure 47 Displaying the firmware options

- Click **Program** to initiate the firmware update. The message “Device programmed successfully” appears on the bottom left once the programming is completed.

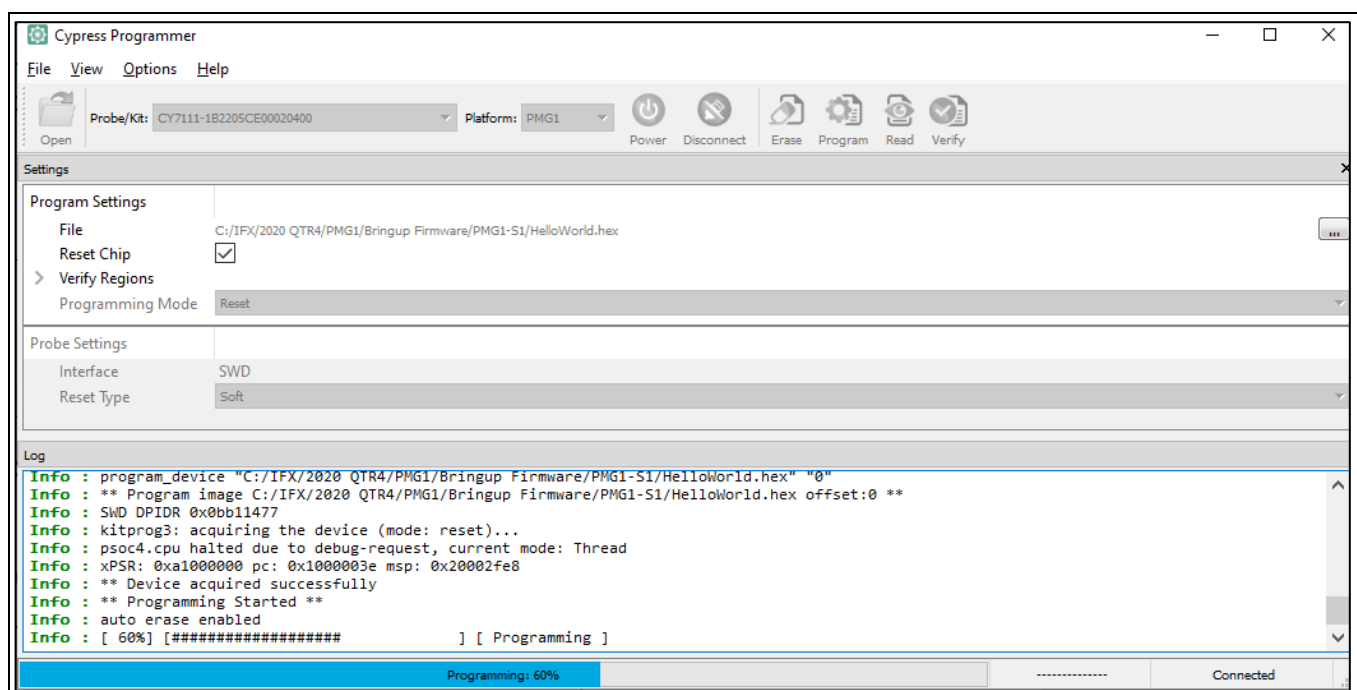


Figure 48 Programming through CYPRESS™ programmer

5.4 Programming and Debugging using the 10-pin SWD interface

5.4.1 Programming using MiniProg4 from ModusToolbox™ software

The 10-pin SWD header (J8) is provided on the prototyping kits to support 10-pin SWD programmer and debugger probes. The header is not populated on the board by default, and you need to solder the standard Mini SWD 0.05-inch-pitch connector. FTSH-105-01-L-DV-K-P-TR from Samtech is the recommended connector. See [MCU I/O header](#) for pinout connection details.

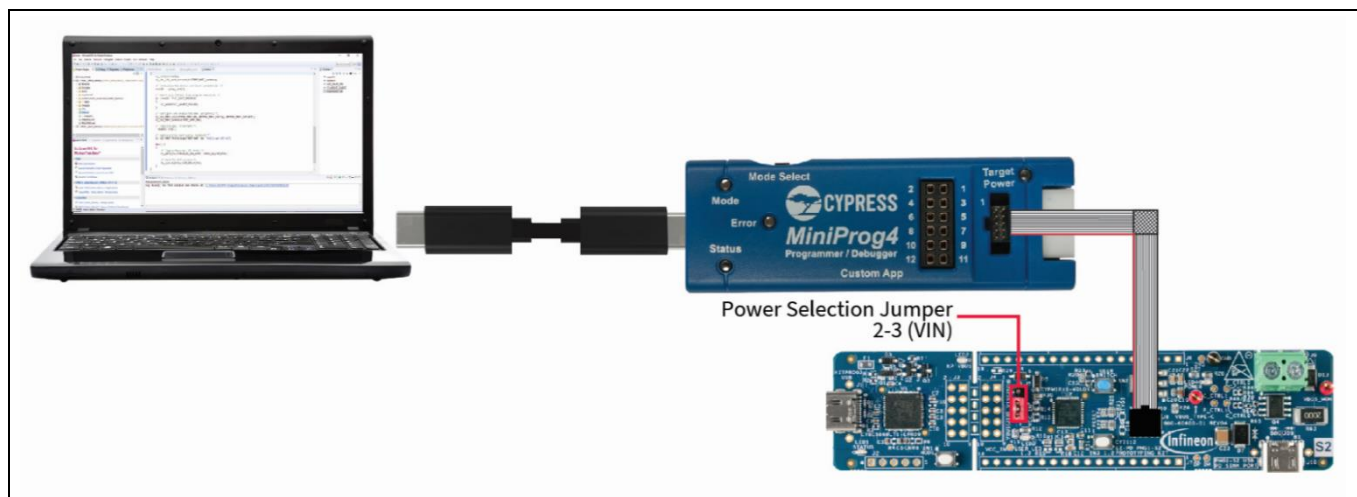


Figure 49 Programming through the 10-pin SWD interface

Figure 49 shows the connection diagram for programming the kit through a 10-pin SWD connector. Connect the MiniProg4 programmer to the 10-pin SWD connector on the kit using a 10-pin ribbon cable.

The MiniProg4 does not supply power to the target board in its default configuration. The following changes should be implemented on the **run configuration/debug configuration** to use MiniProg4 as a programmer.

6. In ModusToolbox™, right-click **Project Explorer**, and open **Run Configurations (Run As > Run Configurations...)**.
7. Select **<Application name> Program (KitProg3_MiniProg4)** under **GDB OpenOCD Debugging**.
8. On the right-side pane, navigate to the **Debugger** tab.
9. In the **Config** options, set **PSOC4_USE_ACQUIRE**.
 - To enable power cycle programming for the CY7110, set the value to **2**.
 - To enable reset programming for the CY7111, CY7112, and CY7113, set the value to **1**.
10. In **Config options**, add an **ENABLE_POWER_SUPPLY** parameter, and set the value to **5000**.

The **ENABLE_POWER_SUPPLY** parameter enables power on the VTARG pin of the MiniProg4 and sets the voltage to the value of the parameter (in mV). For more details, see [Figure 50](#).

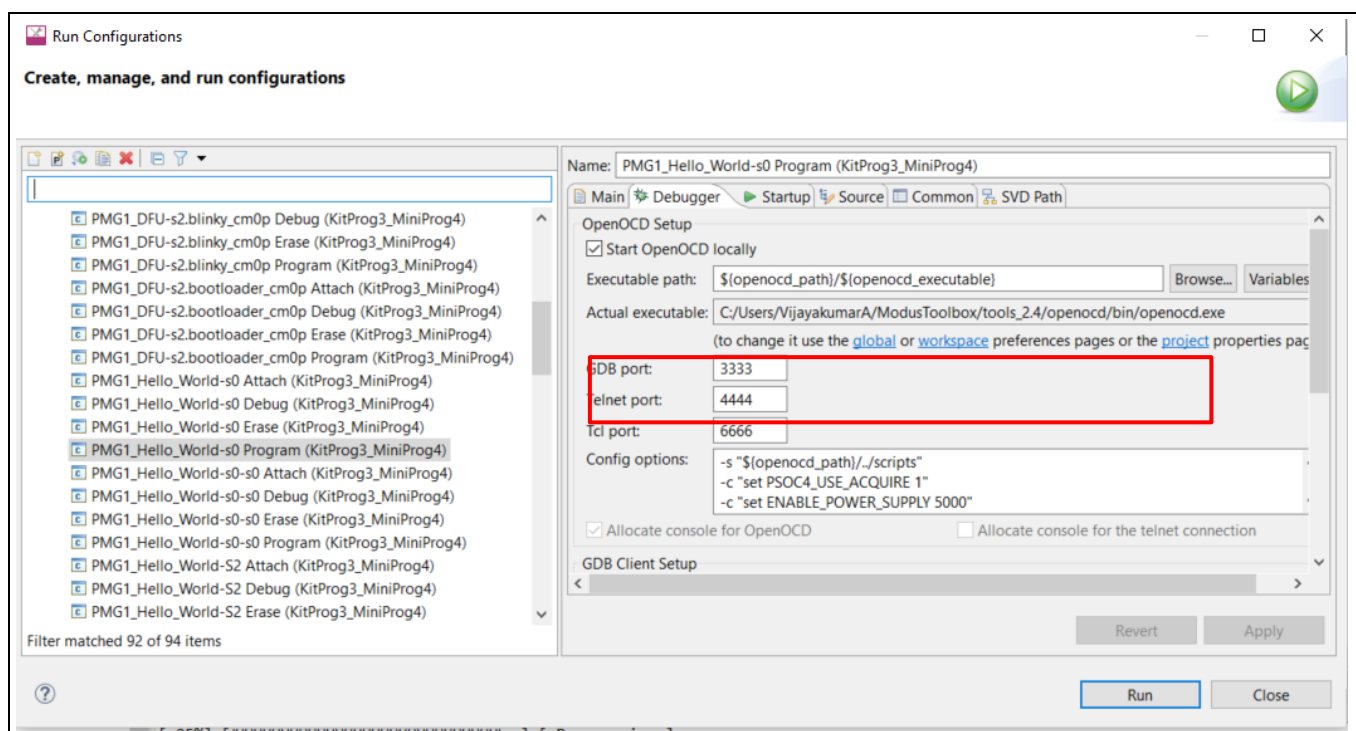


Figure 50 Configuring ModusToolbox to program PMG1 kits

Once the configuration has been updated as described above, the procedure to initiate programming through the Eclipse IDE for ModusToolbox™ software is the same as that of the KitProg3 interface. Follow the steps listed in [Programming using ModusToolbox™ software](#).

5.4.2 Debug using MiniProg4 from ModusToolbox™ software

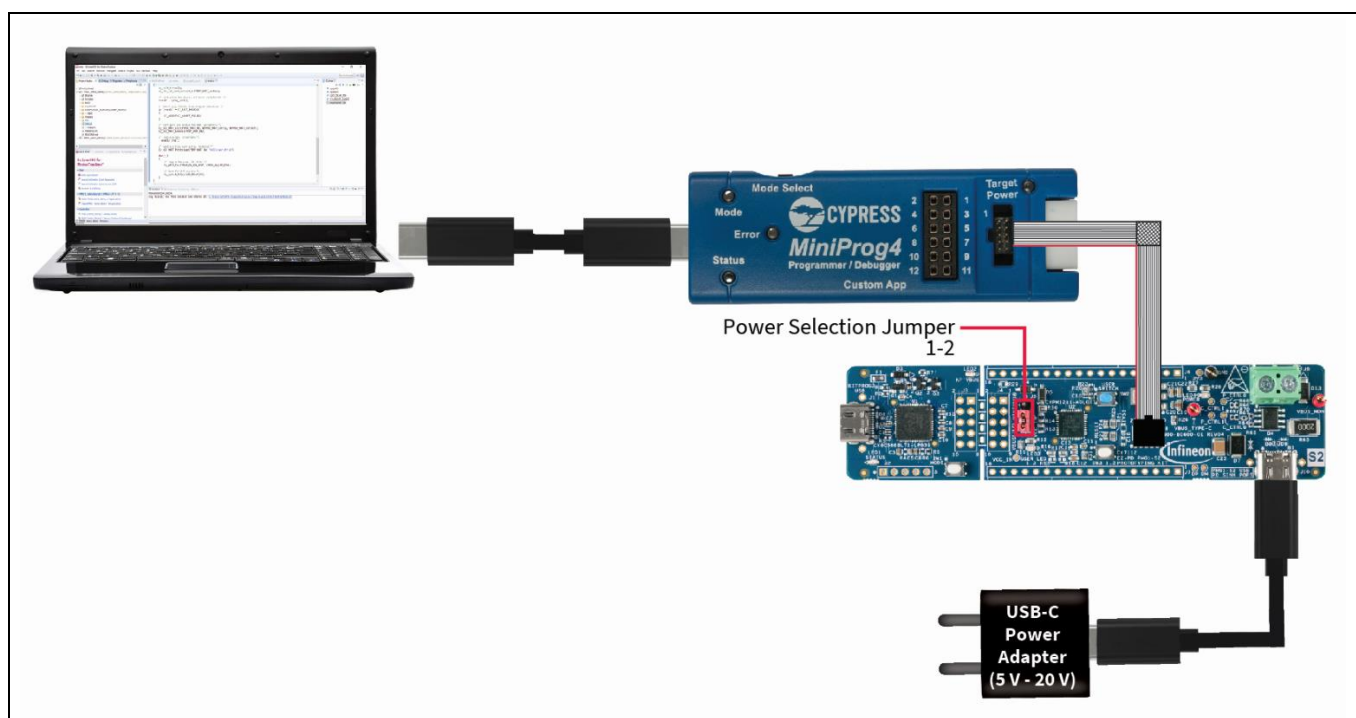


Figure 51 Debugging through the 10-pin SWD header

Application development on CY711x kits using ModusToolbox™ software

Figure 51 shows the connection diagram for debugging the kit through a 10-pin SWD connector. Connect the MiniProg4 programmer to the 10-pin SWD connector on the kit using a 10-pin ribbon cable.

The MiniProg4 does not supply power to the target board in its default configuration. The following changes should be implemented on the **debug configurations** to use MiniProg4 as a debugger.

1. In ModusToolbox™, right-click **Project Explorer** and open **Debug Configurations (Debug As > Debug Configurations...)**.
2. Select **<Application name> Debug (KitProg3_MiniProg4)** under **GDB OpenOCD Debugging**.
3. On the right-side pane, navigate to the **Debugger** tab.
4. In the **Config** options, set **PSOC4_USE_ACQUIRE**.
 - To enable power cycle programming for the CY7110, set the value to **2**.
 - To enable reset programming for the CY7111, CY7112, and CY7113, set the value to **1**.
5. In **Config options**, add an **ENABLE_POWER_SUPPLY** parameter, and set the value to **5000**.
The ENABLE_POWER_SUPPLY parameter enables power on the VTARG pin of the MiniProg4 and sets the voltage to the value of the parameter (in mV). For more details, see **Figure 52**.
6. Click Apply to save the changes and close the debug configuration window.

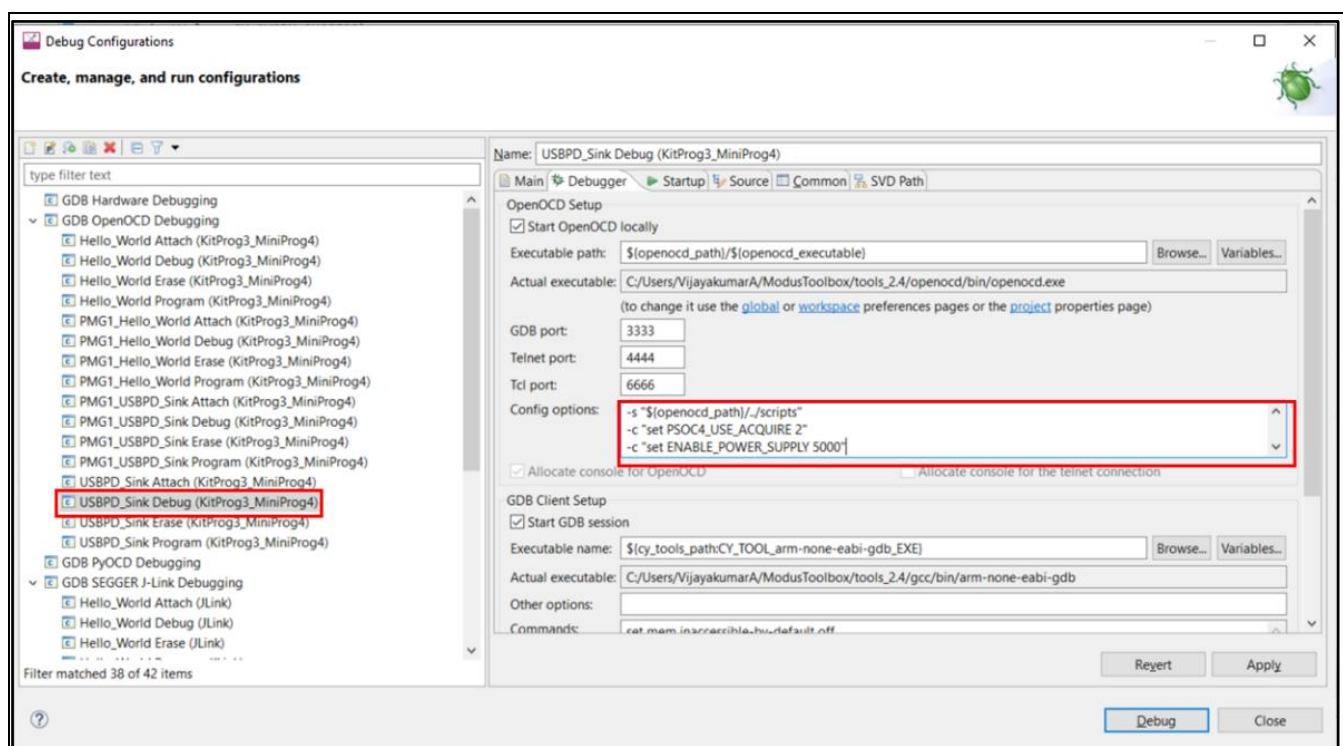


Figure 52 Enabling power supply on the VTARG pin

Once the configuration has been updated as described above, the procedure to initiate debugging through the Eclipse IDE for ModusToolbox™ software is the same as that of the KitProg3 interface. Follow the steps listed in **5.3.2**.

5.4.3 Programming using MiniProg4 from CYPRESS™ programmer (CYP)

See **Figure 49** for the connection details.

1. Connect the power section jumper (J5) to position 2–3 (VIN).
2. Connect the prototyping kit to the host PC through the KitProg3 connector.

Application development on CY711x kits using ModusToolbox™ software

3. Launch CYP. See the user guide [\[10\]](#) for the detailed procedure to download and install CYP.
4. From the **Probe/Kits** drop-down list, select **MP4 BULK**. And select **PMG1** from the platform drop-down list.
5. Click **Open** to load the programming file. In the Open Programming File dialog, browse to the location of the HEX, SREC, ELF, or BIN file to be loaded. Select the file and click **Open**.
6. Under **Program Settings**, select **Programming Mode** according to the connected kit
 - a) Power cycle for CY7110
 - b) Reset for CY7111, CY7112, and CY7113

See [Figure 53](#) for more details.

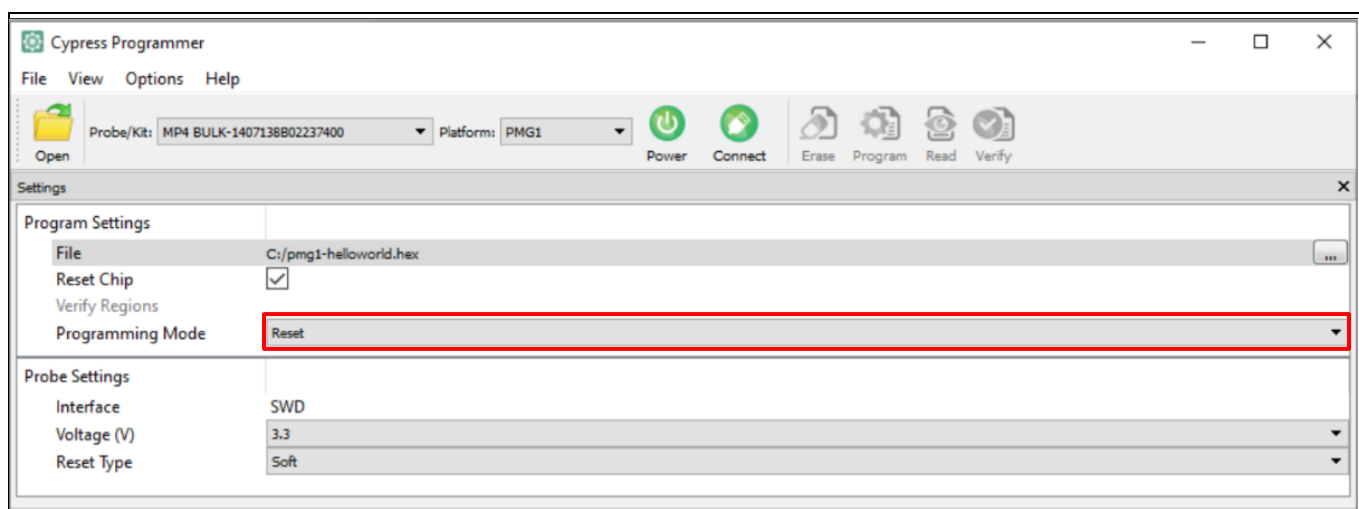


Figure 53 Configuring CYP to program EZ-PD™ PMG1 MCU kits

7. Click **Connect**. The CYP connects to the target board and the **Erase**, **Program**, **Read**, and **Verify** options appear.
8. Click **Program** to initiate firmware update. The message **Device Programmed successfully** appears on the bottom left once the programming is completed.

5.5 Programming and debugging using J-Link debug probes

The EZ-PD™ PMG1 family of MCUs is supported by Segger J-Link software. Download and install the latest version (version V7.54 or above) of J-Link software from the [J-Link/J-Trace download page](#).

CY711x EZ-PD PMG1-Sx MCU prototyping kits support programming and debug using Segger J-Link in-circuit debugger through the 10-pin SWD connector (J8).

The J-Link debugger probes have a 19-pin (2.54-mm) connector to interface with target boards. For interfacing J-Link Debugger probes to EZ-PD™ PMG1 MCU kits through the onboard 10-pin SWD connector, use a **9-pin Cortex®-M Adapter** from Segger.



Figure 54 9-Pin Cortex®-M adapter

5.5.1 Programming and debugging the CY7110 kit

The EZ-PD™ PMG1-S0 MCU is acquired using power cycle for programming/debug operations. J-Link software v7.54 and above supports power cycle mode acquisition for EZ-PD™ PMG1-S0 MCUs. To support the power cycle mode of acquisition, make the following changes in the adapter hardware and J-Link settings before using the J-Link debugger for EZ-PD™ PMG1-S0 MCU programming/debug.

Adapter modification:

Connect pin 1 to pin 19 of the 20-pin connector on the 9-pin Cortex®-M adapter. This modification allows the J-Link probe to supply power to EZ-PD™ PMG1-S0-MCU for power cycle mode acquisition.

J-Link settings change:

1. Connect J-Link debug probe to the PC (through the USB interface).
2. Launch the J-Link commander tool. The J-Link commander tool displays the firmware version, hardware version, and S/N (serial number) of the connect J-Link debugger probe.
3. Enter the `power on perm` command and press **Enter**. The command enables the power on the 5-V pin (pin #19) of the J-Link probe. The change is non-volatile. For disabling the power on the 5-V pin, enter the `power off perm` command and press **Enter**.

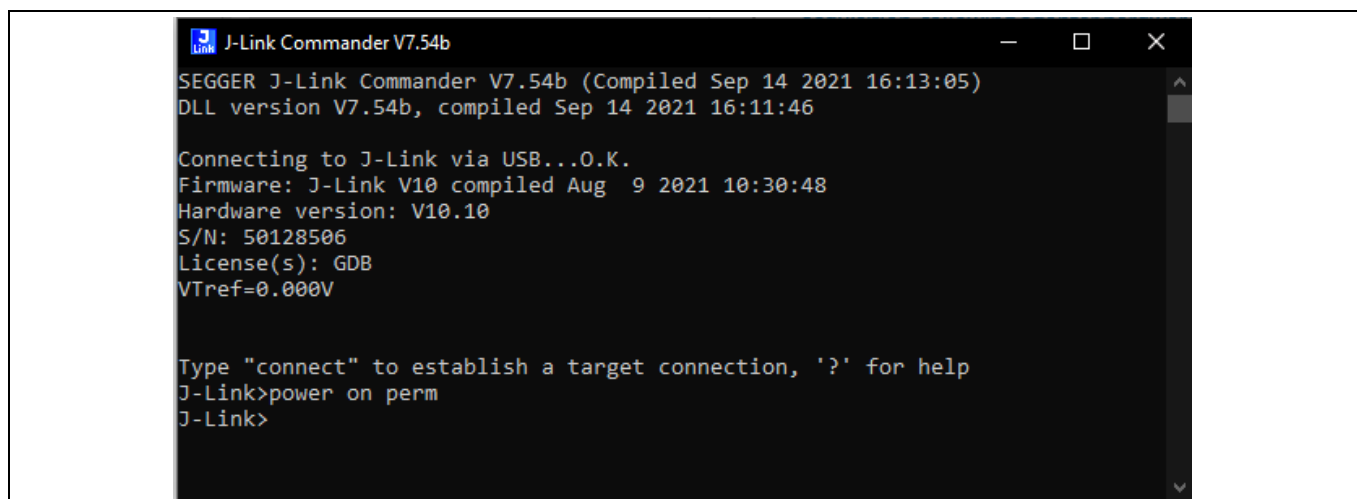


Figure 55 J-Link commander: Settings change for CY7110 programming/debug

Do the following to program or debug the CY7110 kit using the J-Link probe from ModusToolbox™ software.

1. Open the Eclipse IDE for ModusToolbox™ software and select the **PMG1** project on the host PC. For details, see the Eclipse IDE for ModusToolbox™ software user guide [8].
2. Place the jumper shunt on pins 2–3 of the power section jumper (J5).
3. Connect the USB PD sink port to the USB PD source to activate onboard LDO, Load switch, and user LED. Ensure that LED4 (power LED) glows green.
4. Connect J-Link debugger to prototyping kit through 10-pin SWD connector, and use a 9-pin Cortex®-M adapter.
5. In case you want to program the MCU, click <Application name> Program (J-Link) from the Launches section.
6. In case you want to debug the application, click <Application name> Debug (J-Link) from the Launches section. The IDE will switch to debugging mode and will halt at the first line of the main () function. This indicates that the application is ready for debugging.

Note: CY7110 kit supports only 5 V USB PD contract in debug mode using J-Link.

5.5.2 Programming and debugging the CY7111/CY7112/CY7113 kits

The J-Link debugger measures the target MCU's voltage through VTARG pin to detect and establish the connection. The reverse voltage protection diode (D5) between the VTART pin (of J8) and VIN (J5.3) prevents the measurement of the target MCU voltage on the VTARG pin. Do the following changes in hardware before using J-Link.

1. Solder on a standard Mini SWD 0.05-inch-pitch connector. FTSH-105-01-L-DV-K-P-TR from Samtech is the recommended connector.
2. Remove the 0-ohm series resistor between VTARG (J8.1) and VIN (J5.3):
 - R12 for CY7111 and CY7113
 - R30 for CY7112
3. Populate the 0-ohm series resistor between VTARG (J8.1) and VDDD (MCU):
 - R11 for CY7111 and CY7113
 - R29 for CY7112

Application development on CY711x kits using ModusToolbox™ software

Do the following to program or debug the CY7111/CY7112/CY7113 kit using J-Link probe from ModusToolbox™ software:

1. Open the Eclipse IDE for ModusToolbox™ software on the host PC and select the **PMG1** project. For details, see the Eclipse IDE for ModusToolbox™ software user guide [8].
2. Place the jumper shunt on pins 1–2 of the power section jumper (**J5**) to configure the kit in operational mode.
3. Connect the USB PD sink port to the USB PD source to activate the onboard LDO, load switch, and user LED. Ensure that LED4 (power LED) glows green.
4. Connect J-Link debugger to prototyping kit through 10-pin SWD connector, and use a 9-pin Cortex®-M adapter.
5. In case you want to program the MCU, click **<Application name> Program (J-Link)** from the **Launches** section.
6. In case you want to debug the application, click **<Application name> Debug (J-Link)** from the **Launches** section. The IDE will switch to debugging mode and will halt at the first line of the main () function. This indicates that the application is ready for debugging.

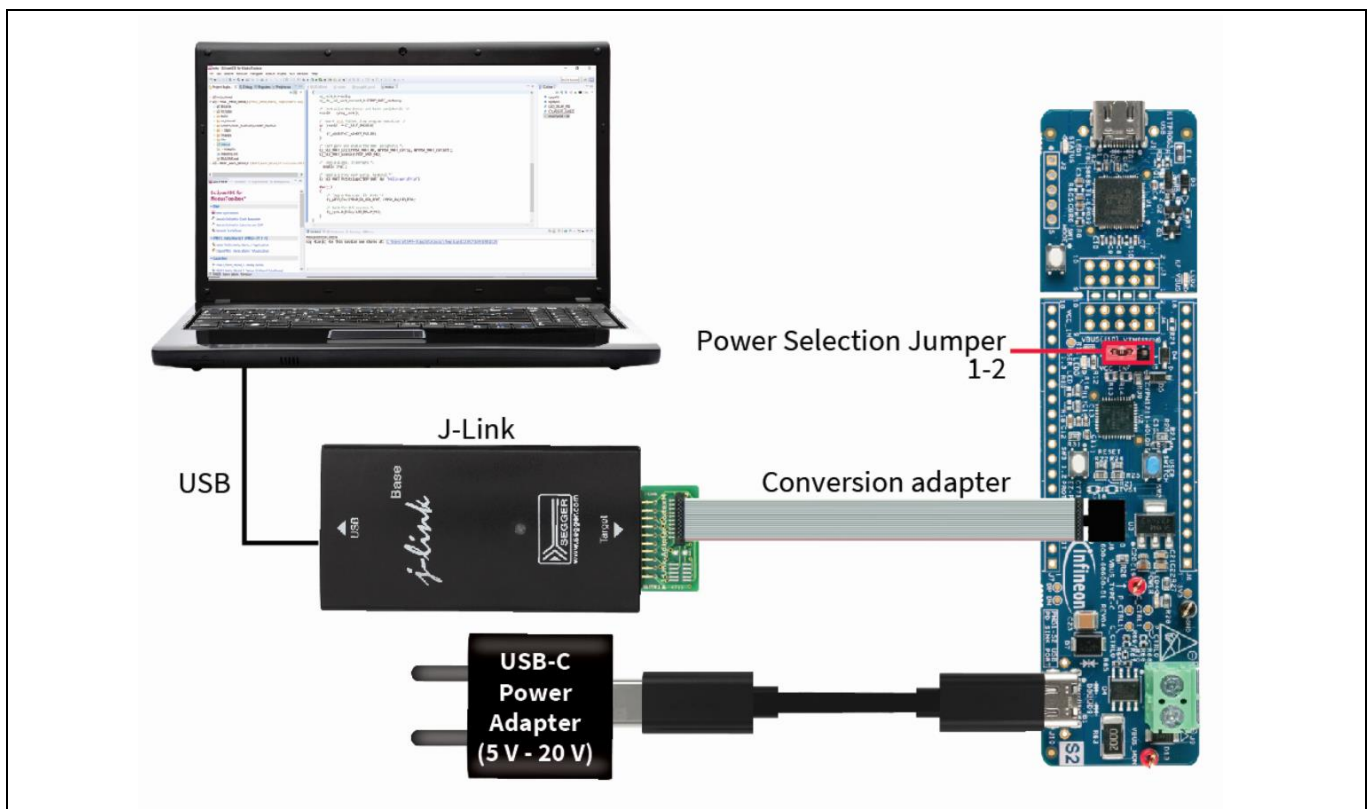


Figure 56 Program/debug using J-Link debugger probes

5.6 Configuring the PDOs and PD parameters

The EZ-PD™ configurator tool, integrated with the Eclipse IDE for ModusToolbox™ software, provides a tool for selecting and configuring parameters of the PD stack. Do the following to configure PDOs and PD parameters using the EZ-PD™ configurator Tool.

1. Open the Eclipse IDE for ModusToolbox™ software on the host PC and select the **PMG1** project. For details, see the Eclipse IDE for ModusToolbox™ software user guide [\[8\]](#).
2. Open EZ-PD™ configurator 1.0 from the **Tools** menu in the quick panel.

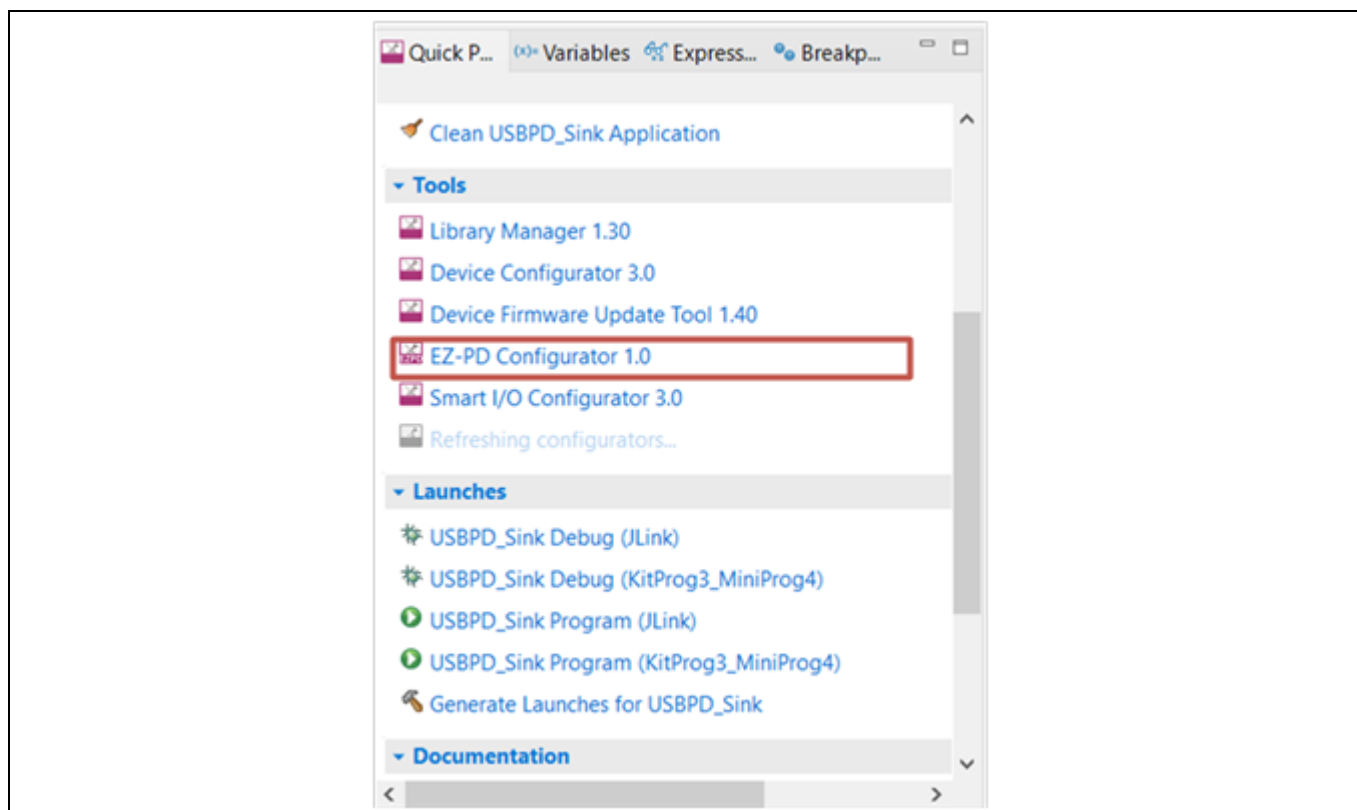


Figure 57 Launching the EZ-PD™ configurator tool

Application development on CY711x kits using ModusToolbox™ software

3. On the EZ-PD™ configurator tool window, click on the category name to expand.

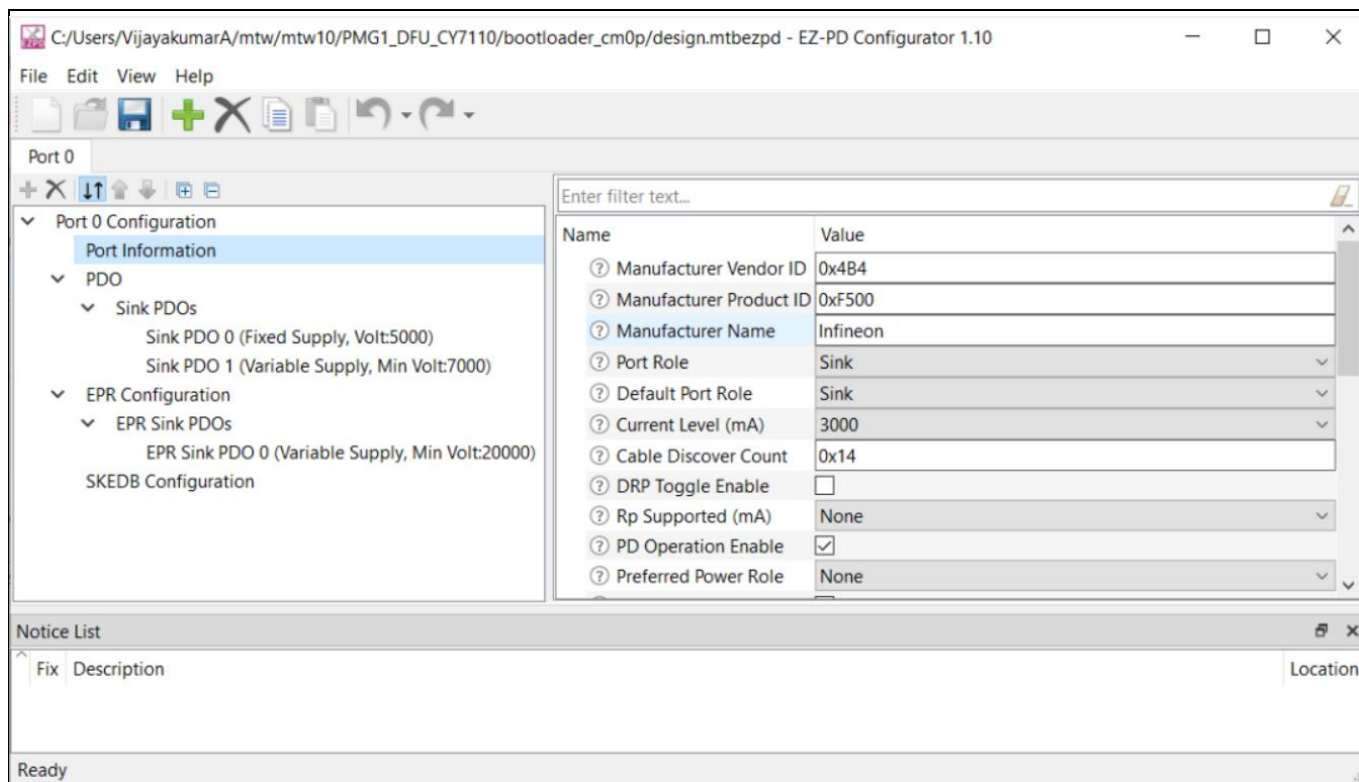


Figure 58 EZ-PD™ configurator and list of categories

You can configure the port Information, source PDO, sink PDO, SCEDB configuration, and SKEDB configuration parameter categories in the tool. For more details, see the ModusToolbox™ software EZ-PD™ configurator guide.

Note: To open ModusToolbox™ software EZ-PD™ configurator guide, press **F1** or click **Help > View Help**.

4. Select **Sink PDO** in the parameter category to configure parameters related to the sink PDO.

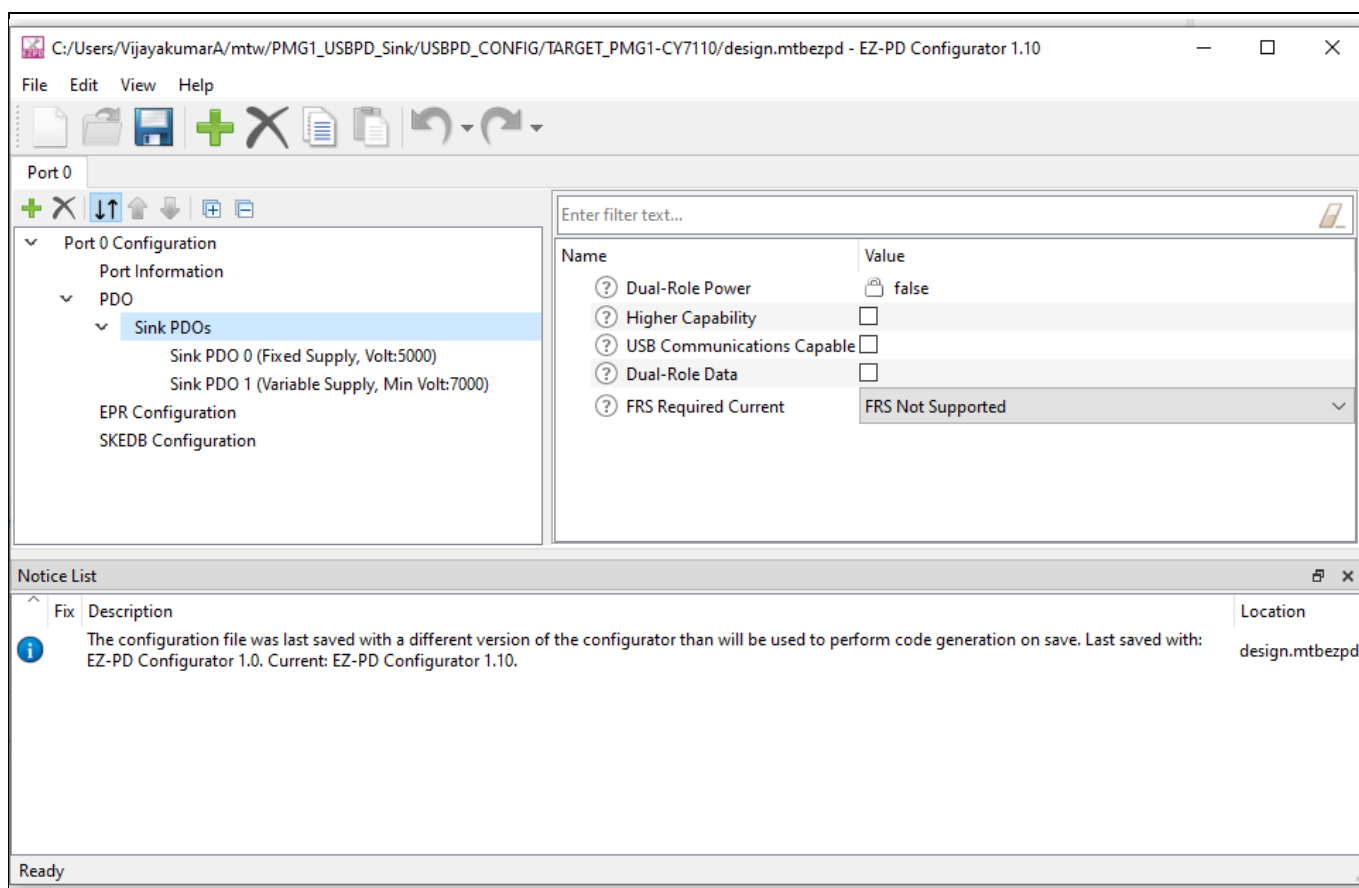


Figure 59 Sink PDO parameters

- **Sink PDO count:** Option to select number of sink PDOs. Maximum is 7
- **Sink PDO <PDO number> enable:** To enable to disable a specific PDO
- **Sink PD <PDO number>:** Configure the PDO value
- **Sink PDO min/max current:** Configure the min/max PDO current

Note: See section 6.4.1.3 of the [USB power delivery specification](#) to understand how to configure PDO value and Min/Max current.

5. Click **Save** to save the configuration and close the EZ-PD™ configurator tool. Recompile the project in Eclipse IDE of ModusToolbox™ software.

Related documents

- [1] Overview: [Infineon USB PD controller roadmap](#)
- [2] Product webpage:
 - [EZ-PD™ PMG1 MCU webpage](#)
- [3] Kit webpages
 - [CY7110](#)
 - [CY7111](#)
 - [CY7112](#)
 - [CY7113](#)
- [4] Datasheets
 - [EZ-PD™ PMG1-S0 MCU datasheet](#)
 - [EZ-PD™ PMG1-S1 MCU datasheet](#)
 - [EZ-PD™ PMG1-S2 MCU datasheet](#)
 - [EZ-PD™ PMG1-S3 MCU datasheet](#)
- [5] [AN232553](#) - Getting started with EZ-PD™ PMG1 MCU on ModusToolbox™ software
- [6] [KitProg3 user guide](#)
- [7] [ModusToolbox™ software user guide](#)
- [8] [Eclipse IDE for ModusToolbox™ software user guide](#)
- [9] [ModusToolbox™ software installation guide](#)
- [10] [CYPRESS™ programmer user guide](#)
- [11] [ModusToolbox™ software device configurator guide](#)

Technical support

If you have any questions, create a support request on the Infineon [Technical Support](#) page.

Revision history

Revision history

Document version	Date of release	Description of changes
**	2021-06-18	Initial release.
*A	2021-09-27	Updated Document Title to read as “EZ-PD™ PMG1 MCU prototyping kits guide, CY7110/CY7111/CY7112”. Added following details about the new revision of kit boards: - PMG1 MCU to KitProg3 UART signals are connected by default - Replaced kit board images with newer revision board images - Added a new section describing programming and debugging using J-link debug probes
*B	2021-11-19	Added contents for CY7113 EZ-PD™ PMG1-S3 MCU prototyping kit
*C	2021-12-02	Updates for ModusToolBox™ version 2.4 Restructured sections 5.1 , 5.2 , and 5.3 for better readability

Trademarks

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

Edition 2021-12-02

Published by

**Infineon Technologies AG
81726 München, Germany**

**© 2021 Infineon Technologies AG.
All Rights Reserved.**

Do you have a question about this document?

Go to: www.cypress.com/support

Document reference

002-32601 Rev. *C

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.