# CY7110/CY7111/CY7112 EZ-PD™ PMG1 prototyping kit guide

## 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 development of USB PD sink applications using the EZ-PD™ PMG1 microcontrollers. You can refer to the relevant sections based on your requirement:

- Introduction – Provides basic information on the kits, the kit structure, and software information
- EZ-PD™ PMG1 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, download and debug applications

### Intended audience

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

## Table of contents

# 1 Introduction

EZ-PD™ PMG1 is a family of high-voltage power delivery microcontrollers. These microcontrollers include Arm® Cortex®-M0 CPU, USB Power Delivery (PD) controller and configurable analog and digital peripherals. EZ-PD™ PMG1 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 | PMG1 microcontroller MPN |
|---|---|---|
| CY7110 EZ-PD™ PMG1-S0 prototyping kit | PMG1-S0 | CYPM1011-24LQXI |
| CY7111 EZ-PD™ PMG1-S1 prototyping kit | PMG1-S1 | CYPM1111-40LQXI |
| CY7112 EZ-PD™ PMG1-S2 prototyping kit | PMG1-S2 | CYPM1211-40LQXI |

See the device datasheets [4] to understand and compare the various features supported by the EZ-PD™ PMG1 controllers.

The EZ-PD™ PMG1 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 bread board 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 switch
- I/O Headers

For a summary of the different MCUs available in the PMG1 product line, visit the **webpage**.

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

If you are new to PMG1 MCUs, see the application note [5], to learn about the PMG1 microcontroller family.

## 1.1 Kit contents

CY711x prototyping kits contain the following:

- CY711x EZ-PD™ PMG1-SX prototyping kit
- Quick start guide

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

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)
- USB PD 3.0-compliant Type-C cable compatible with the USB-C power adapter
- 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 kit can be downloaded from the respective kit webpages **[3]**. The documents include quick start guide, kit guide (this document) and release notes. The hardware design files include schematic, BoM, and layout files.

## 1.3 Board details

This section lists the key interfaces of CY711x boards.

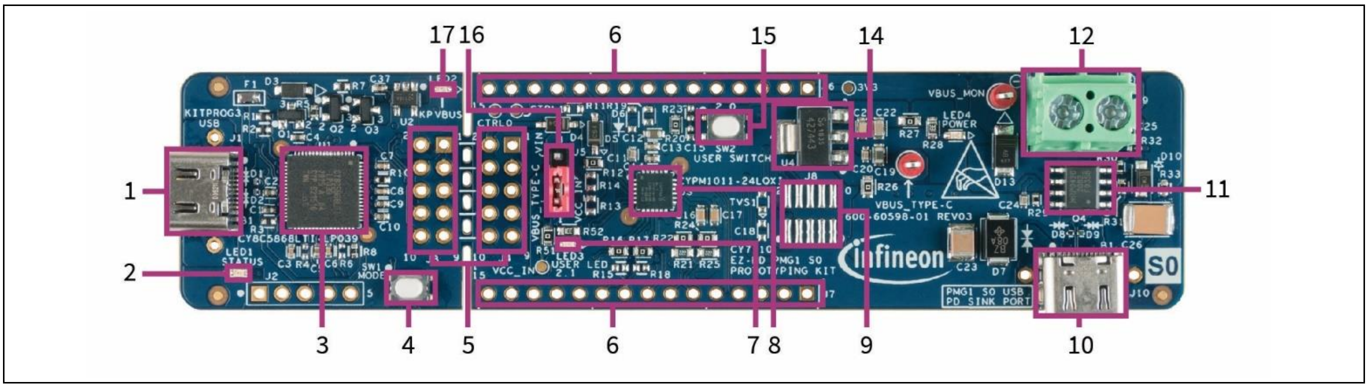


**Figure 1 CY7110 EZ-PD™ PMG1-S0 prototyping kit**

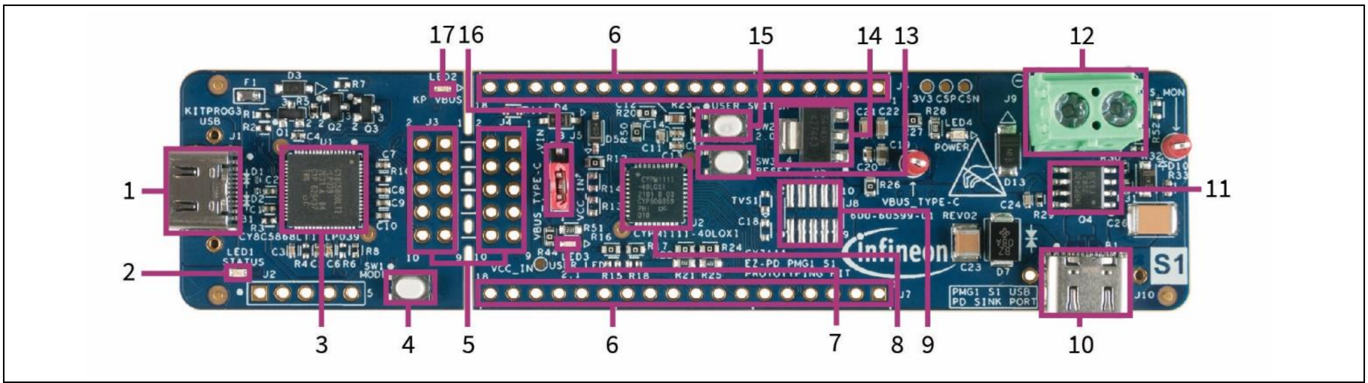


**Figure 2 CY7111 EZ-PD™ PMG1-S1 prototyping kit**

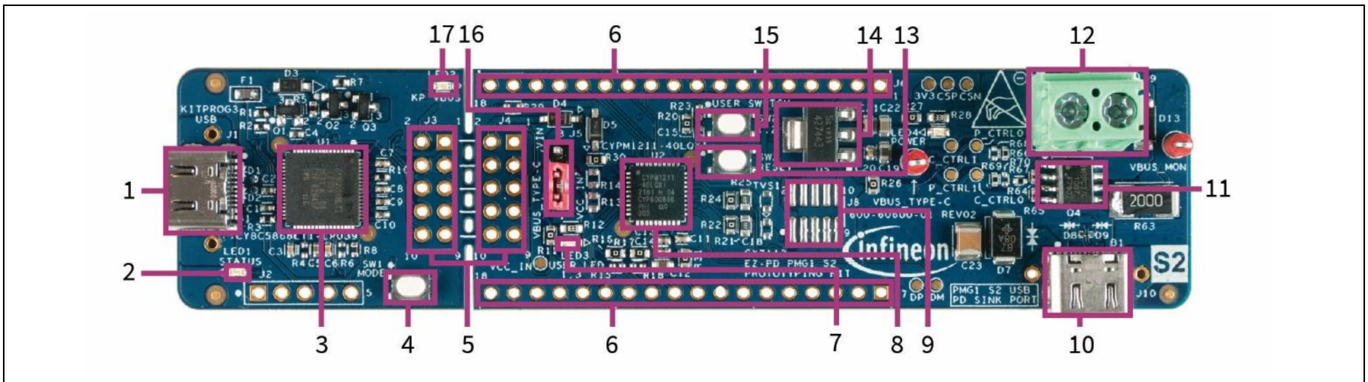


**Figure 3 CY7112 EZ-PD™ PMG1-S2 prototyping kit**

1. KitProg3 USB Type-C port (J1)
2. KitProg3 status LED (LED1)
3. PSoC™ 5LP controller (U1)
4. KitProg3 mode switch (SW1)

5. KitProg3 headers (J3, J4) [1]
6. I/O headers (J6, J7)[1]
7. User LED (LED3)
8. EZ-PD™ PMG1 microcontroller (CYPM1011-24LQXI in CY7110, CYPM1111-40LQXI in CY7111, and CYPM1211-40LQXI in CY7112)
9. 10-pin SWD/JTAG header[1]
10. PMG1 USB PD sink port (J10)
11. Load switch
12. DC_OUT terminal block (J9)
13. Reset switch (SW3)
14. 3.3-V on-board LDO
15. User switch (SW2)
16. Power selection jumper (J5)
17. KitProg3 VBUS LED (LED2)

### 1.3.1 Kit features

The PMG1 prototyping kit features MCUs from the PMG1 high-voltage USB-PD microcontroller family. The following are the features of the prototyping kit:

- The prototyping kit is powered through the PMG1 USB PD sink port and the MCU is powered directly from the port. The USB-PD 3.0 connector is capable of supporting PD contracts up to 20 V/5 A (100 W).
- The prototyping kit connects to a PC through the KitProg3 Type-C port (J1) for firmware programming and debugging. The kit has a KitProg3 on-board 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 in the board design for connecting programming or debugging modules such as MiniProg4.
- Four on-board LEDs—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 device is powered. |
| User LED (LED3) | Green | This is a firmware-driven LED. The kit must be powered though 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 on-board 3.3-V LDO. |

- There are two switches on the CY7110 kit and three switches on the CY7111/CY7112 kits. See **Table 3** for the details.

[1] Footprint only, not populated on the board.

**Table 3 Prototyping kit – switches**

| Switch | Description |
|---|---|
| KitProg3 mode switch (SW1) | Mode switch required to enter the KitProg3 into bootloader mode. See the KitProg3 user guide [6] for details. |
| User switch (SW2) | This button can be used to provide an input to the PMG1 MCU. This button pulls the pin to ground, by default. |
| Reset switch (SW3)[2] | This button is used to reset the PMG1 MCU. It connects the PMG1 MCU reset (XRES) pin to 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 in the DC_OUT terminal block when successful PD contract is established.
- 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 both sides of the prototyping kit. The headers are compatible with 2.54-mm standard breadboard to extend the kit functionality. **Table 4** lists the pins count of I/O headers in all 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 |

## 1.3.2 Power selection jumper

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

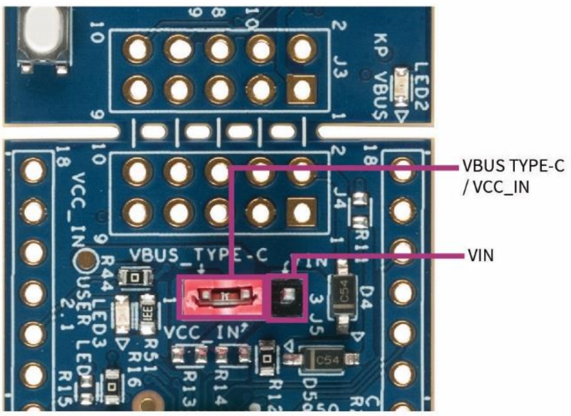


**Figure 4 Power selection jumper**

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

**Table 5 Power selection jumpers**

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

## 1.3.3 Prototyping kit pinout

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

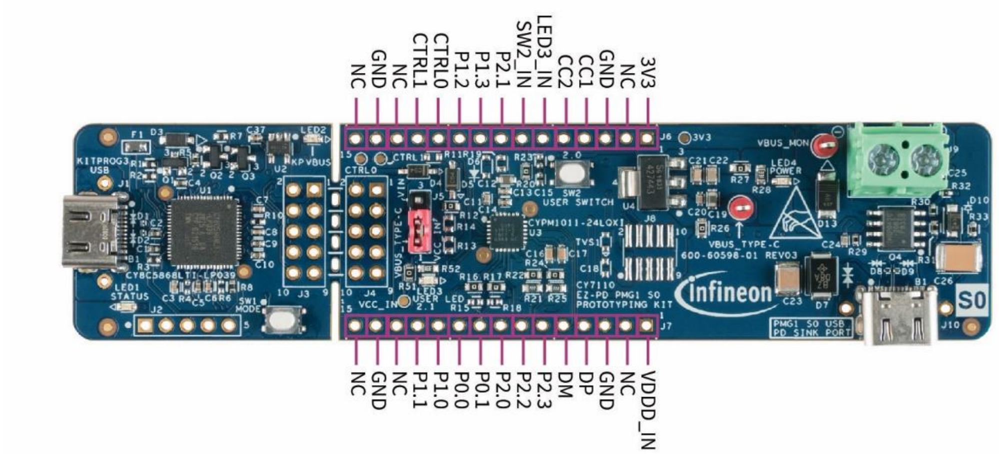


**Figure 5 CY7110 prototyping kit – I/O header**

**Table 6 CY7110 MCU I/O header pinout**

| CY7110 pin | CY7110 pin name | PMG1-S0 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| **J6** | | | | | |
| 1 | 3V3 | – | 3.3 V regulator output | – | Connected to the output of the on-board 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 switch output. SW2 is input to this PMG1 device. This pin can be used to read switch 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) | See **KitProg3 (PSoC™ 5LP)** for establishing UART connection between the MCU and KitProg3 |
| 10 | P1.2 | 5 | UART-Tx | P1.2 (GPIO) | |
| 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 | – | – | – | – |
| 15 | NC | – | – | – | – |
| **J7** | | | | | |
| 1 | VDDD_IN | 23 | Alternate VDDD input pin | – | Connected through the non-populated resistor/diode |
| 2 | NC | – | – | – | – |
| 3 | GND | – | – | – | – |
| 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) | |

| CY7110 pin | CY7110 pin name | PMG1-S0 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| 8 | P2.0 | 9 | User switch 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 | – | – | – | – |
| 15 | NC | – | – | – | – |

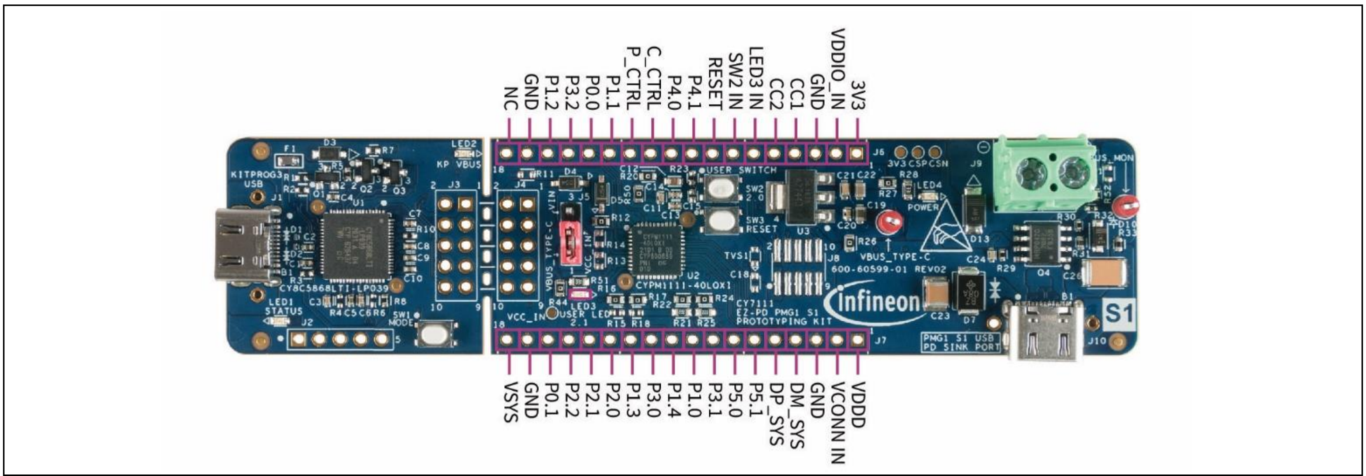


**Figure 6** CY7111 prototyping kit – I/O header

**Table 7** CY7111 MCU I/O header pinout

| CY7111 pin | CY7111 pin name | PMG1-S1 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 | 32 | VDDIO alternate input | | Connected through the non-populated resistor/diode |
| 3 | GND | – | – | – | – |
| 4 | CC1 | 9 | CC1 signal | – | – |
| 5 | CC2 | 7 | CC2 signal | – | – |
| 6 | LED3_IN | 14 | User LED external input. LED3 is driven by firmware, by | P2.1 (GPIO) | Connected to LED3 and P2.1 pin through a 0-ohm resistor |

| CY7111 pin | CY7111 pin name | PMG1-S1 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| | | | default. Use this pin to drive it externally. | | |
| 7 | SW2_IN | 13 | User switch output. SW2 is input to the PMG1 device. This pin can be used to read the switch 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) | Refer to **KitProg3 (PSoC™ 5LP)** section for establishing a UART connection between the MCU and KitProg3 |
| 10 | P4.0 | 29 | UART-Tx | P4.0 (GPIO) | |
| 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 | – | – | – | – |
| 18 | NC | – | – | – | – |
| **J7** | | | | | |
| 1 | VDDD | 31 | PMG1-S1 Internal LDO output | – | – |
| 2 | VCONN_IN | 8 | VCONN source external input. By default, V5V is left floating on the device. | – | Connected through the non-populated resistor/diode |
| 3 | GND | – | – | – | – |
| 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) | – |

| CY7111 pin | CY7111 pin name | PMG1-S1 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| 10 | P1.4 | 6 | SWD data | P1.4 (GPIO) | – |
| 11 | P3.0 | 18 | GPIO | – | – |
| 12 | P1.3 | 5 | GPIO | – | – |
| 13 | P2.0 | 13 | User switch 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 | – | – | – | – |
| 18 | VSYS | 19 | Alternate power input option-VSYS | – | Connected through the non-populated resistor/diode |

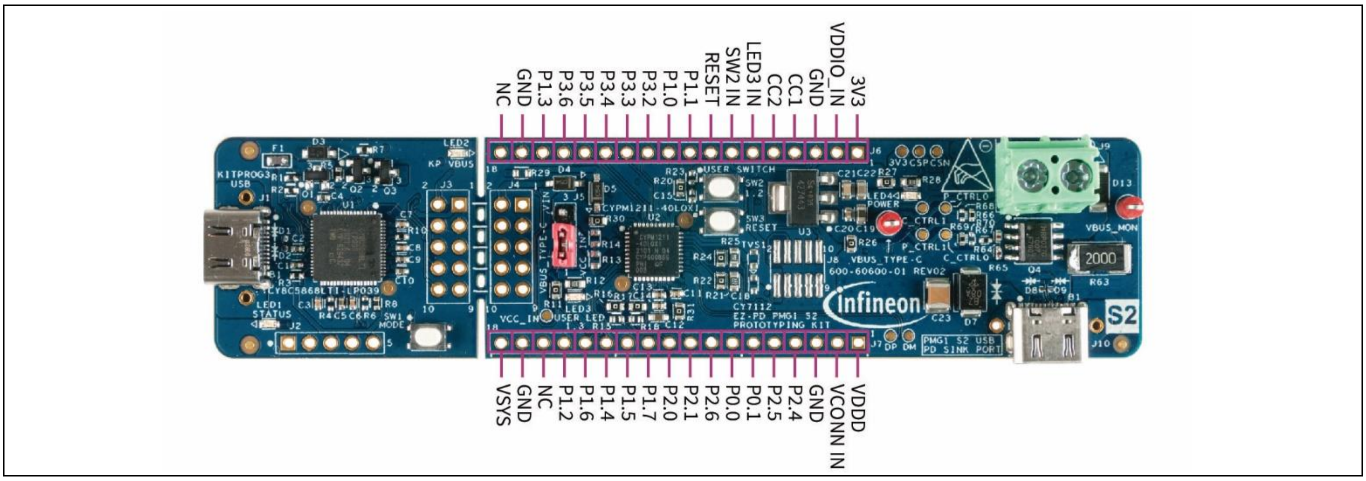


**Figure 7 CY7112 prototyping kit – I/O header**

**Table 8 CY7112 MCU I/O header pinout**

| CY7112 pin | CY7112 pin name | PMG1-S2 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 | 5 | CC1 signal | – | – |
| 5 | CC2 | 3 | CC2 signal | – | – |

| CY7112 pin | CY7112 pin name | PMG1-S2 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| 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 switch output. SW2 is input to PMG1 device. This pin can be used to read the switch 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) | Refer to **KitProg3 (PSoC™ 5LP)** 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 | – | – | – | – |
| 18 | NC | – | – | – | – |
| **J7** | | | | | |
| 1 | VDDD | 17 | PMG1-S1 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 | – | – | – | – |
| 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 | – | – |

| CY7112 pin | CY7112 pin name | PMG1-S2 pin | Primary function | Secondary function | Connection details |
|---|---|---|---|---|---|
| 12 | P1.5 | 13 | GPIO | – | – |
| 13 | P1.4 | 12 | GPIO | – | – |
| 14 | P1.6 | 11 | GPIO | – | – |
| 15 | P1.2 | 9 | User switch SW2 input | P1.2 (GPIO) | – |
| 16 | NC | – | – | – | – |
| 17 | GND | - | – | – | – |
| 18 | VSYS | 20 | Alternate power input option-VSYS | – | Connected through the non-populated resistor/diode |

### 1.3.4 KitProg3 header

The KitProg3 module is the on-board 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).

The signal lines between J3 and J4 are connected on the board except UART. The UART signals (Tx and Rx) between the KitProg3 and PMG1 MCU can be connected using jumper wires as explained in **KitProg3 (PSoC™ 5LP)**.

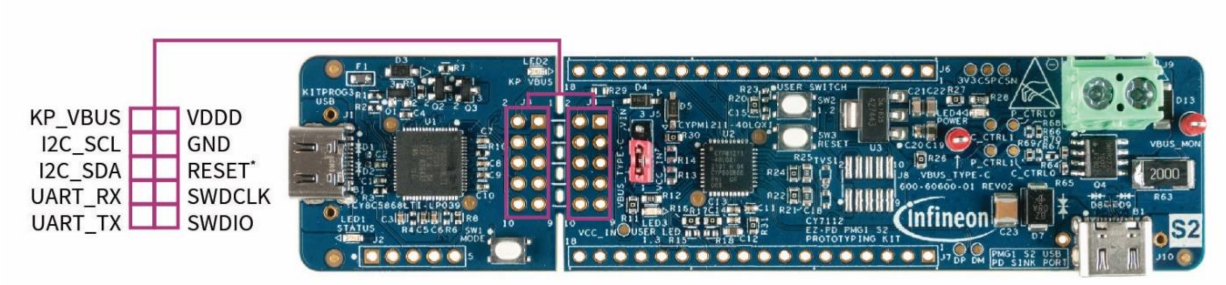


**Figure 8 KitProg3 header**

**Table 9 KitProg3 header pinout**

| Pin # | Pin name | Description |
|---|---|---|
| 1 | VDDD | Target (PMG1) Internal regulator output |
| 2 | KP_VBUS | KitProg3 VBUS/programming VBUS |
| 3 | GND | Ground |
| 4 | I2C_SCL | I2C SCL signal between KitProg3 and PMG1 |
| 5 | RESET[3] | Reset signal for programming PMG1 |
| 6 | I2C_SDA | I2C SDA signal |
| 7 | SWDCLK | SWD clock |

[3] Reset not available on CY7110.

| Pin # | Pin name | Description |
|---|---|---|
| 8 | UART_RX | UART Rx |
| 9 | SWDIO | SWD data |
| 10 | UART_TX | UART Tx |

## 1.4 ModusToolbox™ software

ModusToolbox™ software provides various software tools required for the development of PMG1 MCU-based applications.

ModusToolbox™ software includes a multi-platform Eclipse-based integrated development environment (IDE) that supports application configuration and development. You can create new projects, exercise example projects, build, program and debug using the Eclipse-based IDE.

# 2 EZ-PD™ PMG1 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 PMG1 MCU-based applications. CYPRESS™ programmer is a GUI-based tool to program 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. However, the privilege is not required to run the software once it is installed. See the installation guide **[9]** for the detailed procedure on installing ModusToolbox™ software in different platforms. 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 a PMG1 MCU-based application. An Eclipse-based IDE in ModusToolbox™ software supports application configuration and development. Note that the support for PMG1 is added in ModusToolbox™ software version 2.3 onwards.

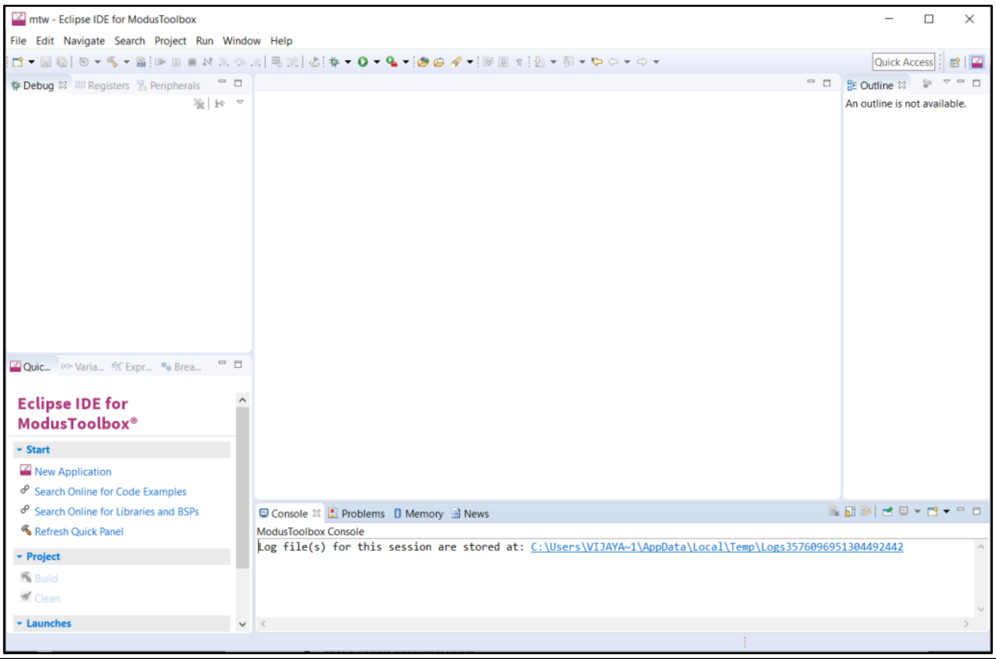


**Figure 9 Eclipse IDE for ModusToolbox™ software**

### 2.2.1 Device configurator tool

The Device configurator Tool, another tool available in ModusToolbox™ software, helps you to configure a hardware block or a middleware library. The configurator is available as a GUI and CLI tool. If you use the Eclipse IDE provided with ModusToolbox™ software, you can access the Device configurator tool from the IDE and open the appropriate application details.

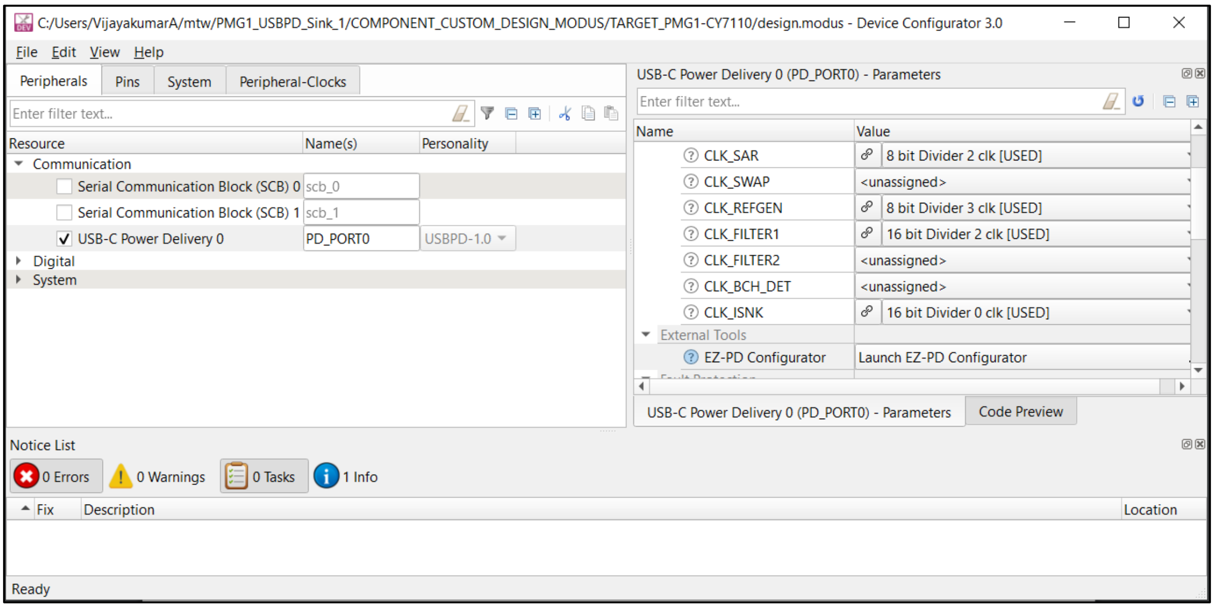


**Figure 10 Device configurator**

### 2.2.2 EZ-PD™ configurator tool

EZ-PD™ configurator is another tool in ModusToolbox™ software used to configure USB-PD features in an application which supports it. You can launch the EZ-PD™ configurator from the device configurator in the IDE.

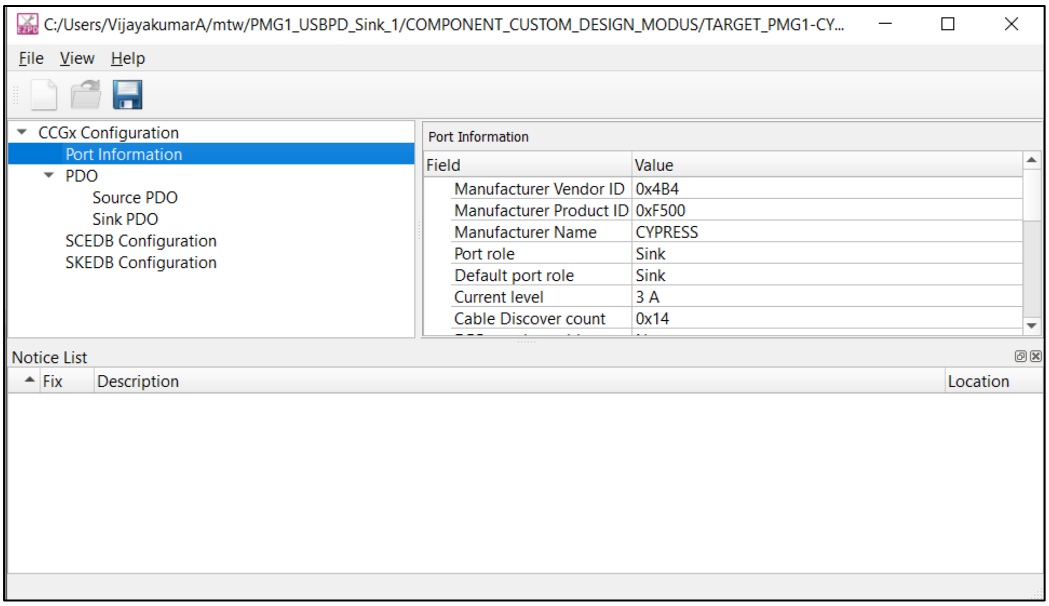


**Figure 11 EZ-PD™ configurator**

## 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.

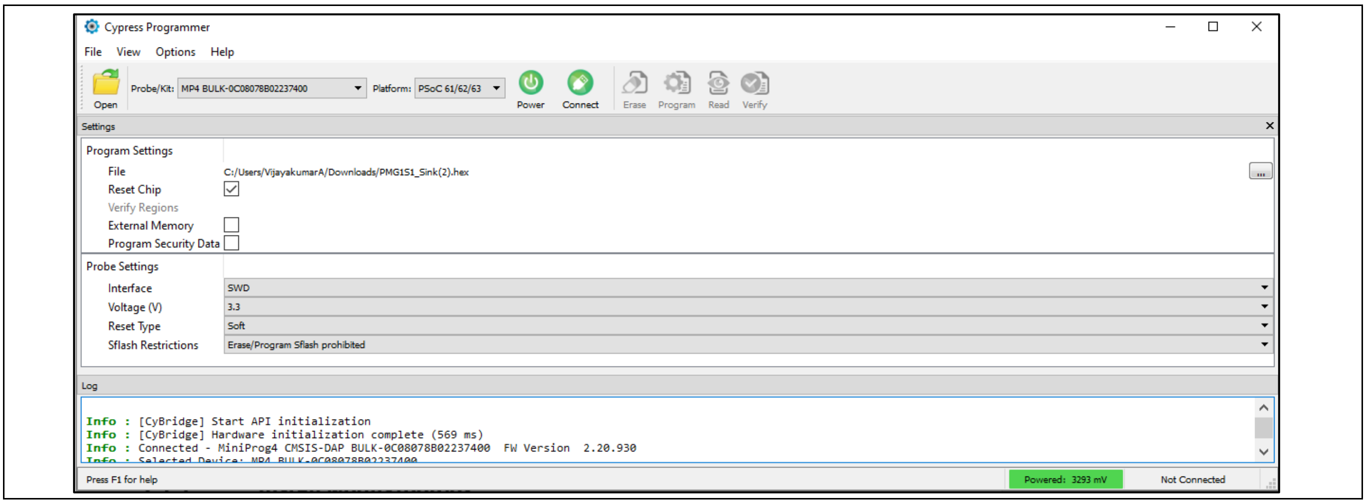


**Figure 12 CYPRESS™ programmer GUI**

See the CYPRESS™ programmer user guide **[10]** for the detailed procedure to download and install CYP.

# 3 EZ-PD™ PMG1 prototyping kit system design

The section describes the features supported by the CY711x PMG1 prototyping kits and the hardware design details.

## 3.1 Top-level hardware design

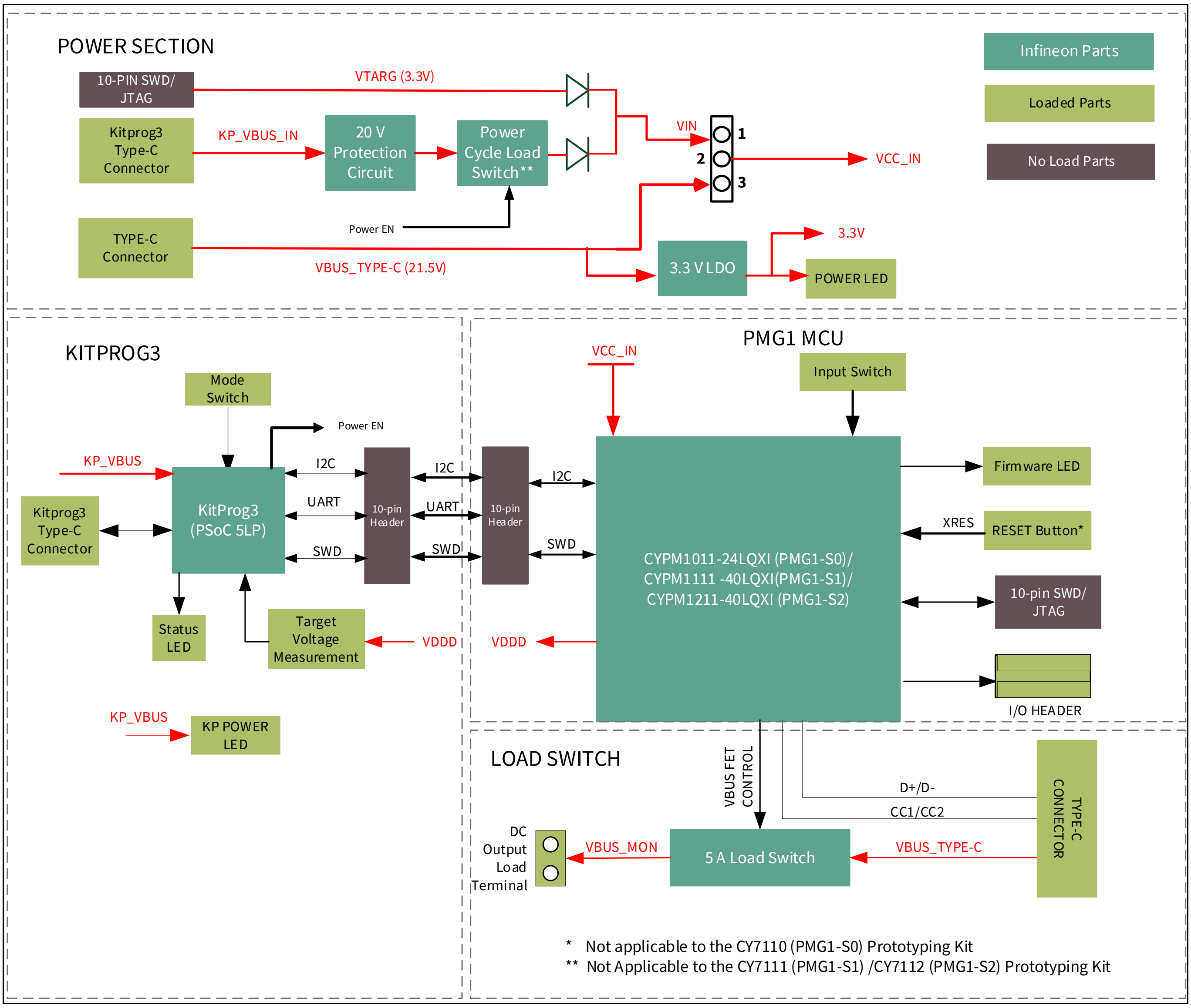


**Figure 13 CY711x EZ-PD™ PMG1-SX prototyping kit block diagram**

## 3.2 Power section

### 3.2.1 Power modes

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

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

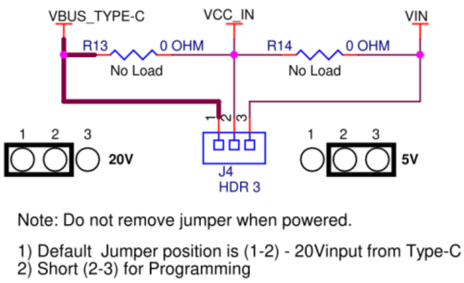


**Figure 14 Power selection jumper**

#### 3.2.1.1 Programming mode

For programming using the on-board 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 PMG1 MCU.

#### 3.2.1.2 Operational mode

For normal operation, the prototyping kit gets powered through 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.

CY711x prototyping kits support up to 100 W (20 V, 5 A) of the USB PD contract.

### 3.2.2 3.3 V LDO and power LED

The on-board 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 switch 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 on-board 3.3 V LDO can source a maximum current of 400 mA. Power LED glows green when the regulator output is available.

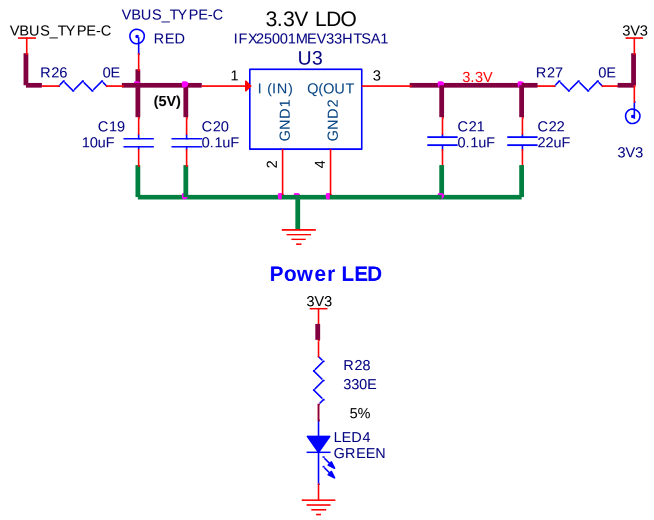


**Figure 15 On-board 3.3-V LDO and power LED**

### 3.2.3 Overvoltage protection circuit

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

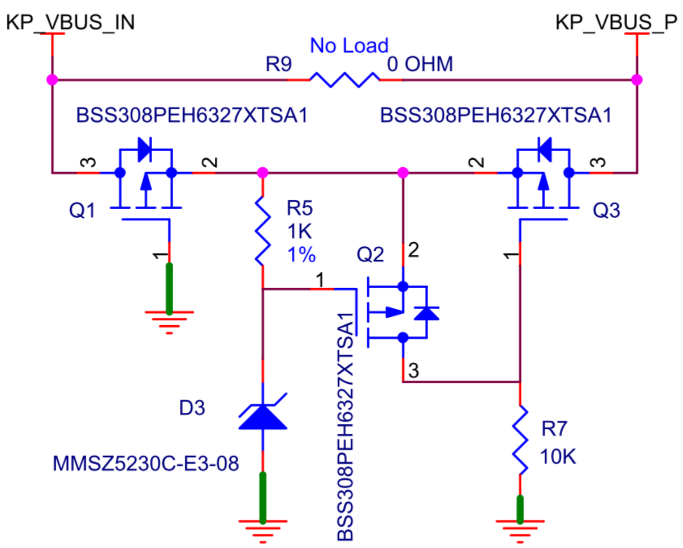


**Figure 16 Overvoltage protection circuit for KitProg3**

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 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 over voltage threshold for the circuit.

### 3.2.4 Power cycle load switch (CY7110)

The approach to acquire PMG1-S0 during programming/debugging is different from other PMG1 family devices because PMG1-S0 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 PMG1-S0 MCU.

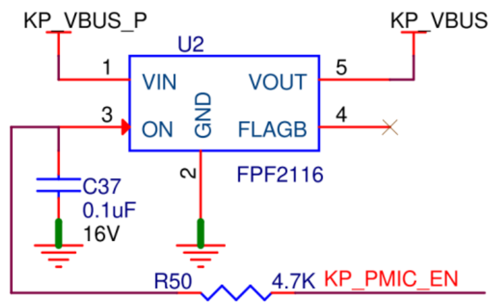


**Figure 17 Power cycle load switch**

### 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.

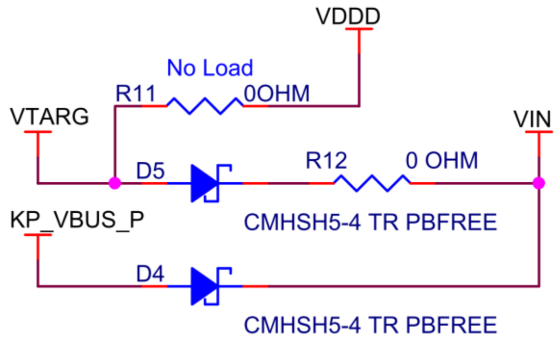


**Figure 18 Transient voltage protection diode**

## 3.3 PMG1 subsystem

### 3.3.1 PMG1 device

#### 3.3.1.1 PMG1-S0 MCU (CYPM1011-24LQXI)

PMG1-S0 includes 64-KB flash, 8-KB SRAM, 12 GPIOs, a complete Type-C USB-PD transceiver, a pull-down termination resistor $R_D$ to support sink on the Type-C port, and system-level ESD protection. It is available in a 24-pin QFN package.

See the PMG1-S0 datasheet [4] for detailed functional and electrical details of the device.

#### 3.3.1.2 PMG1-S1 MCU (CYPM1111-40LQXI)

PMG1-S1 includes 128-KB flash, 12-KB SRAM, 17 GPIOs, a complete Type-C USB PD transceiver with all termination resistors $R_P$, $R_D$ and dead battery $R_D$. It is available in a 40-pin QFN package.

See the PMG1-S1 datasheet [4] for detailed functional and electrical details of the device.

#### 3.3.1.3 PMG1-S2 MCU (CYPM1211-40LQXI)

PMG1-S2 has 128-KB flash, 8-KB SRAM, 20 GPIOs, full-speed USB device controller, a Crypto engine for authentication, a 20-V-tolerant regulator, and a pair of FETs to switch a 5-V (VCONN) supply. PMG1-S2 also integrates two pairs of gate drivers to control external VBUS FETs and system-level ESD protection. PMG1-S2 is available in a 40-QFN package.

See the PMG1-S2 datasheet [4] for detailed functional and electrical details of the device.

### 3.3.2 User LED and switch

The user LED (green) is a PMG1 MCU-driven LED (LED3); you can configure the function through firmware. **Table 10** lists the GPIO assigned for LEDs in each kit. The GPIOs need to be driven LOW to turn ON the user LED.

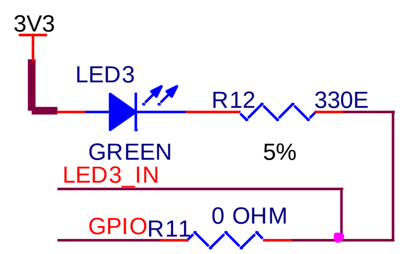


**Figure 19** **User LED**

### 3.3.3 User switch (SW2)

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

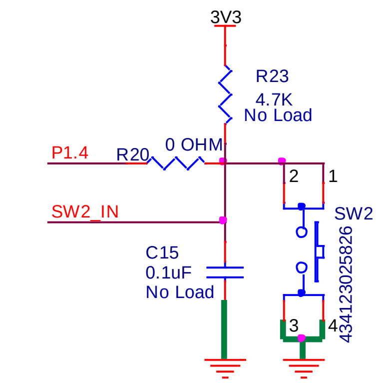


**Figure 20 User switch**

**Table 10 GPIO details for user LEDs and user switches**

| Kit | User LED GPIO | User switch GPIO |
|---|---|---|
| CY7110 | P2.1 | P2.0 |
| CY7111 | P2.1 | P2.0 |
| CY7112 | P1.3 | P1.2 |

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

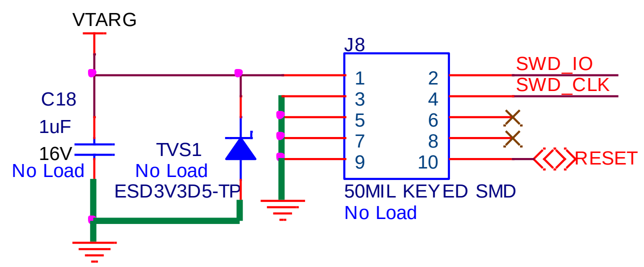


**Figure 21 10-pin SWD header**

### 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 **Prototyping kit pinout**.

### 3.3.6 Reset button

CY7111 PMG1-S1 and CY7112 PMG1-S2 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 ground and reset the device. The PMG1-S0 device does not have the reset pin; therefore, the CY7110 device does not support manual reset.

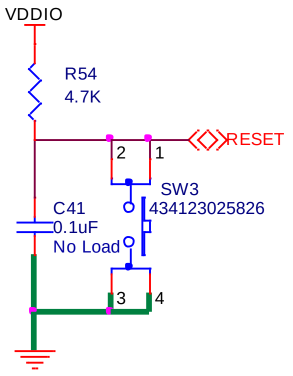


**Figure 22 Reset circuit**

## 3.4 KitProg3 (PSoC™ 5LP)

An on-board PSoC™ 5LP (CY8C5868LTI-LP039)-based KitProg3 module is used to program and debug the PMG1 microcontroller.

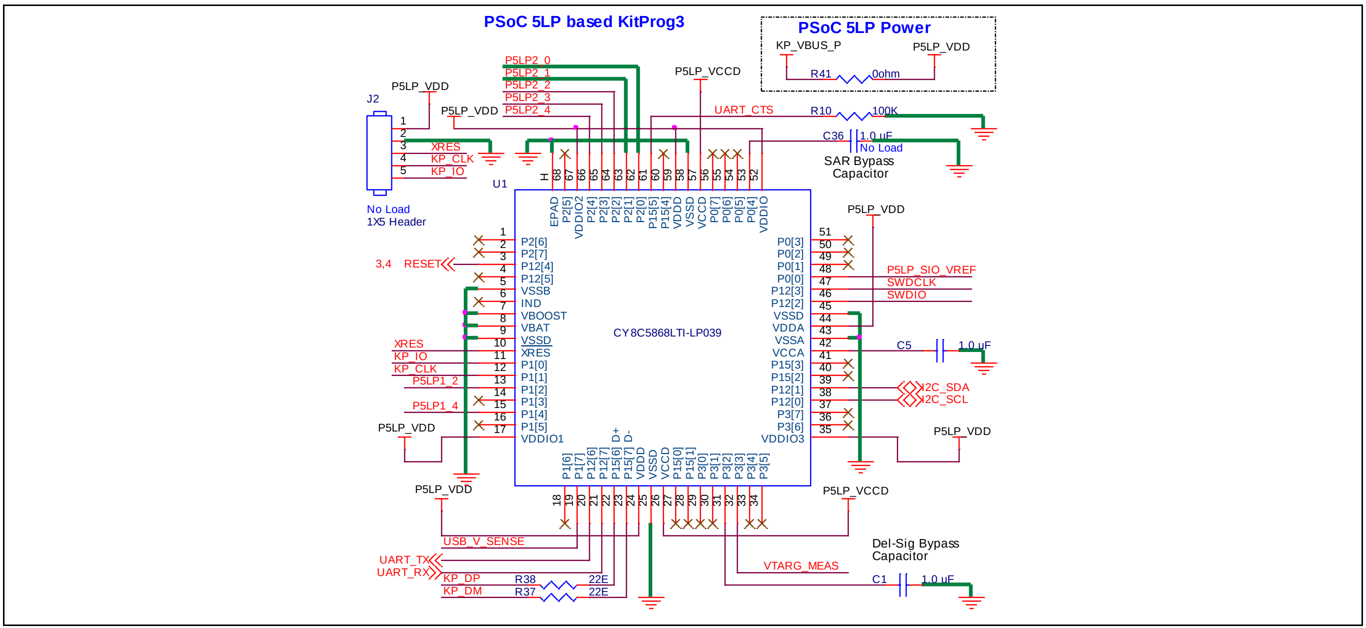


**Figure 23 PSoC™ 5LP device**

The PSoC™ 5LP device interfaces with a PC through a Type-C USB connector (J1), and functions as a bridge between the PC and PMG1 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 PMG1 device in programming or debugging mode via the SWD header. In addition to being an on-board programmer, the KitProg3 functions as an interface for the USB-I2C and USB-UART bridges.

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

The USB-UART bridge functionality can be enabled by connecting the UART lines of KitProg3 to the PMG1 MCU using external wires. The UART lines from the PMG1 MCU are routed to the MCU I/O header (J6) and UART lines from KitProg3 are connected to the KitProg3 header (J3). Connect J6.10 (PMG1 UART Tx) to J3.8 (KitProg3 UART Rx) and J6.9 (PMG1 UART Rx) to J3.10 (KitProg3 UART Tx) to establish the UART connection.

### 3.4.1 Mode switch

The KitProg3 mode switch in the PMG1 family kit enables the KitProg3 module to enter the bootloader mode. The bootloader mode is required to update the KitProg3 firmware on PSoC™ 5LP, 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.

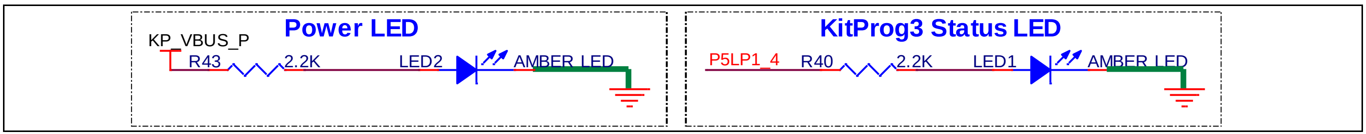


**Figure 24** KitProg3 power and status LED

## 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 PMG1 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 PMG1-S0 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 **Power section**. The CC1/CC2 and D+/D- lines are routed from the Type-C connector to the PMG1 MCU. The USB-PD contract negotiation, charger detection, and data communication occurs through these signal lines. These signals are susceptible to ESD events; therefore, additional ESD diodes are added in the traces.

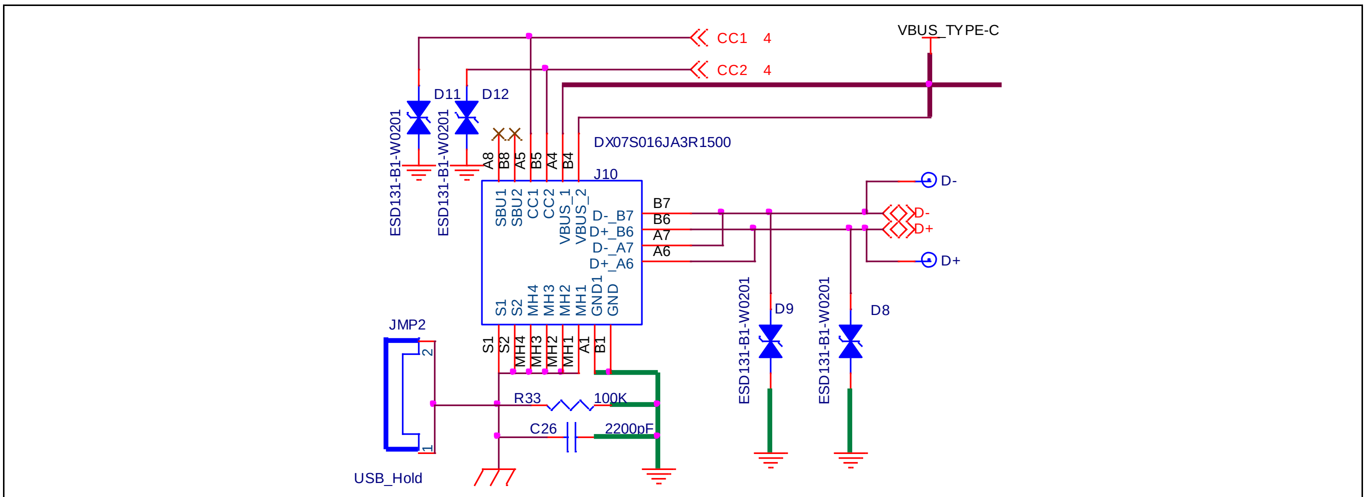


**Figure 25** USB-PD sink Type-C port

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 PMG1-S1 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 DPLUS and DMINUS lines to the Type-C top or bottom port based on the CC (Type-C plug) orientation. See the PMG1-S1 datasheet [4] more details.

### 3.6.2 5-A load switch

PMG1 family kits support 100 W (20 V/5 A) USB-PD contract through the USB-PD sink port. The load switch supports maximum contract voltage (20 V) and maximum contract current (5 A).

#### 3.6.2.1 P-FET load switch

CY7110 and CY7111 kits have dual-P channel MOSFET as the load switch because the PMG1-S0 and PMG1-S1 devices have integrated PFET gate driver to drive external PFETs. IRF9358 is a dual-P channel MOSFET from Infineon with 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.

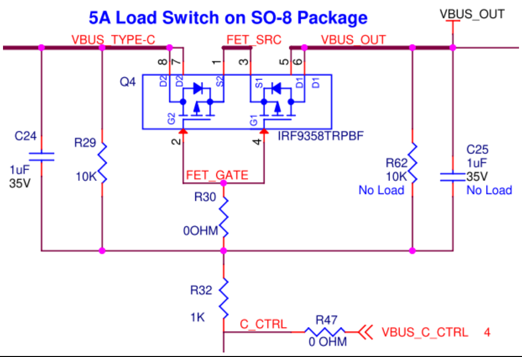


**Figure 26 P-MOSFET load switch**

### 3.6.2.2 N-FET load switch

In the EZ-PD™ PMG1-S2 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 PMG1-S2 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).

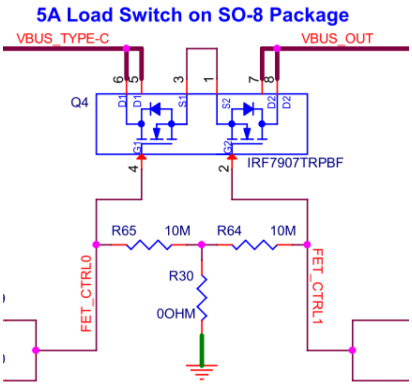


**Figure 27 N-MOSFET load switch**

## 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 PMG1 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.

# 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 and the firmware associated with the USB PD sink code example supported in ModusToolbox™ software is downloaded in the factory on these kits. These kits support up to a maximum contract of 20 V, 5 A (100 W) on the PMG1 USB PD sink port.

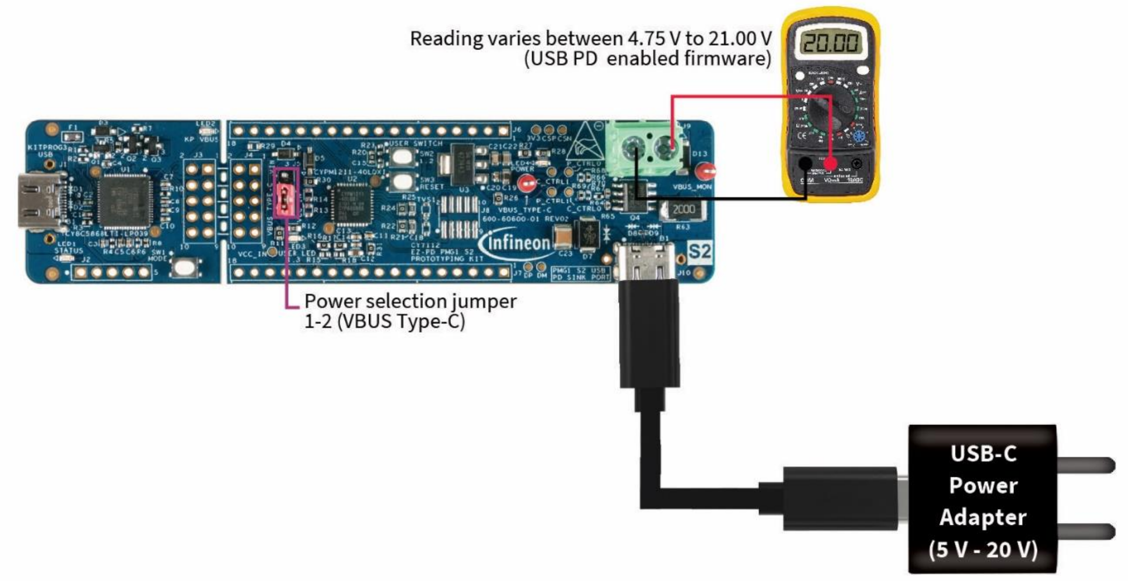


**Figure 28 Prototyping kit in operation mode (USB PD sink-enabled firmware)**

The following is the procedure to run the USB PD sink firmware which is loaded in the factory on the CY711X PMG1 prototyping kits:

1. Put the kit in operational mode by placing a jumper shunt on the power selection jumper (J5) at position 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. Remove the multimeter and connect an external load to the terminal block (J9).

***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.***

6. If you face any issue while operating the USB PD sink firmware, see the troubleshooting listed in **Table 11**.

*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*** *for more details.*

**Table 11 Troubleshooting the Kit**

| # | Issue | Possible Cause | Troubleshooting Procedure |
|---|---|---|---|
| 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 position 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 on USB PD sink port (J10) is compliant to USB PD 3.0 specification. |

# 5 Application development on CY711X kits using ModusToolbox™ software

This section explains the code examples available for PMG1 MCUs in ModusToolbox™ software, configurator tools available to configure the applications, the procedure to download the firmware onto the CY711x boards, and the procedure to debug the applications.

## 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:

- **Hello World**: The "Hello World" application is available for all the PMG1 prototyping kits. This code example demonstrates simple UART communication by printing a "Hello World" message on a terminal and blinking the User LED (LED3) using the PMG1 MCU.
- **USB-PD Sink**: This code example demonstrates USB-C attach detection and USB PD contract negotiation using the PMG1 MCU.

  The 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 on **New Application** option under **Start.**

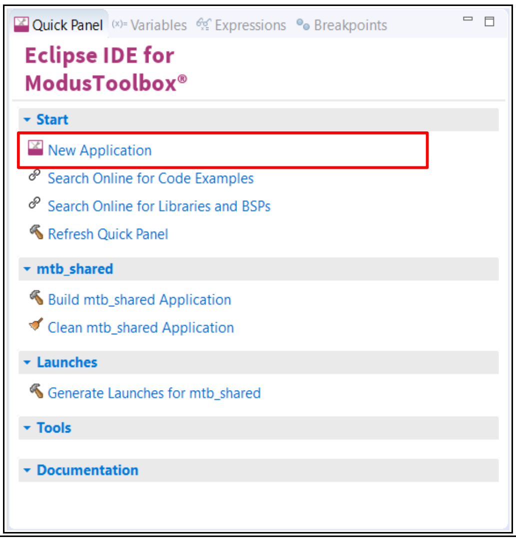


**Figure 29 Creating a new application**

3. In the Project Creator window, choose the Board Support Package (BSP) from the list. Note that the PMG1 prototyping kits are listed under PMG1 BSPs. Click **Next**.

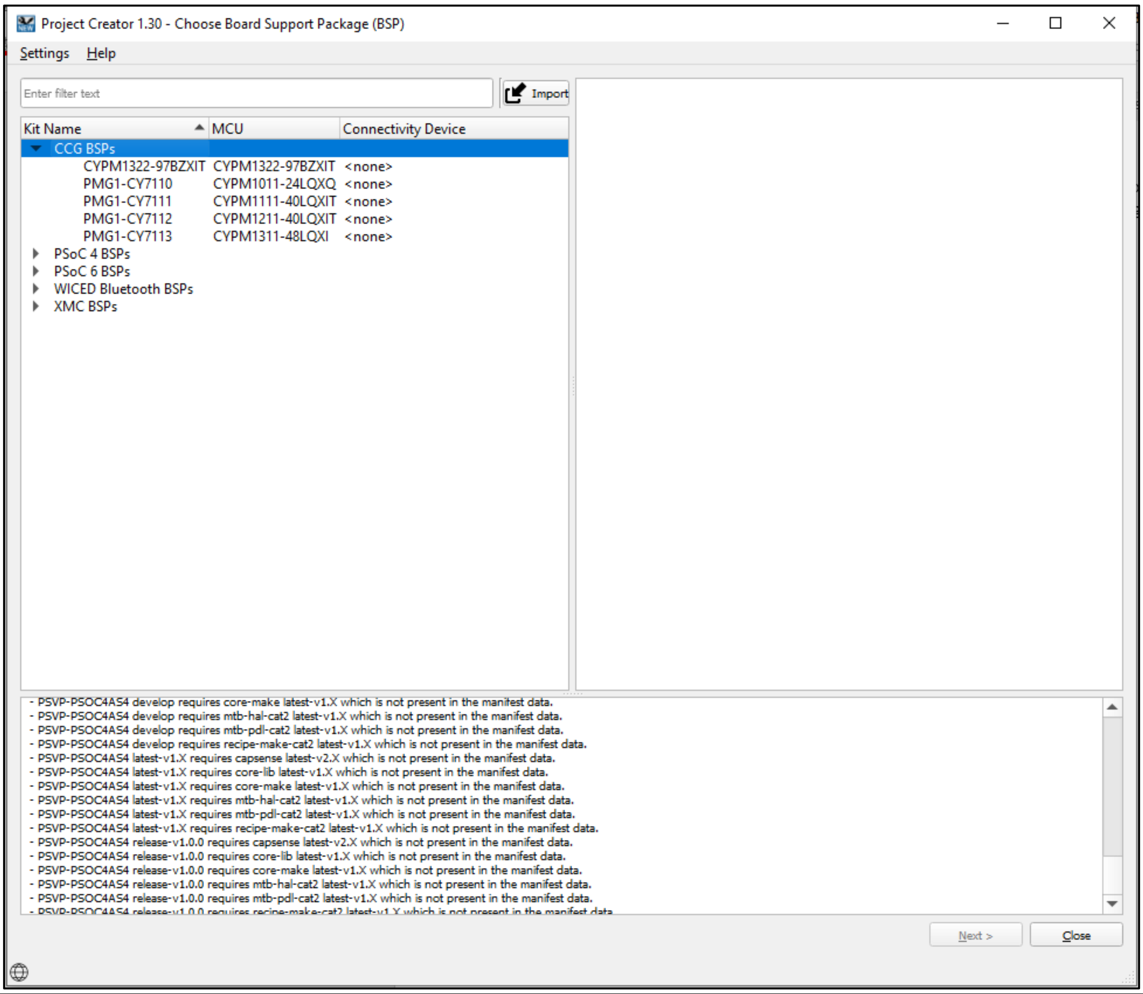


**Figure 30** **Selecting the BSP**

4. Select the desired example project. Enter the **New Application Name**. Click **Create**.

*Note: The prototyping kit does not support DRP application.*

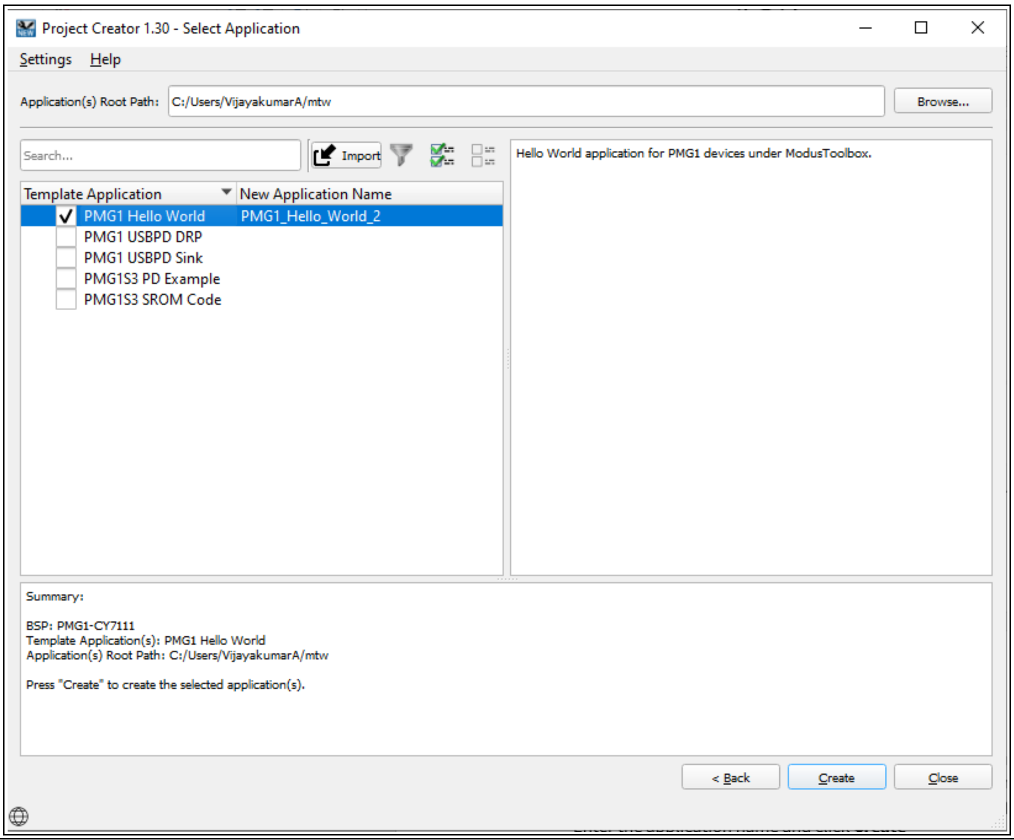


**Figure 31 Selecting the application**

5. 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 the kit

This section explains the procedure to program the PMG1 MCU on the prototyping kit using ModusToolbox™ software or CYPRESS™ programmer (CYP).

The prototyping kit can be programmed through any of the following interfaces:

- KitProg3 USB Type-C port
- 10-pin SWD header
- I/O header pins

The kit supports programming and debugging via the KitProg3 interface by default, and SWD connector/IO headers are not populated by default on the kit.

### 5.3.1 Programming through the KitProg3 interface

#### 5.3.1.1 Using ModusToolbox™ software

KitProg3 is a PSoC™ 5LP-based on-board programming and debugging solution integrated with the PMG1 prototyping kits. You can program and debug the kit without using any separate programming and debug module. Connect the kit to the laptop or PC using the programming connecter as shown in **Figure 32**.

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

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

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

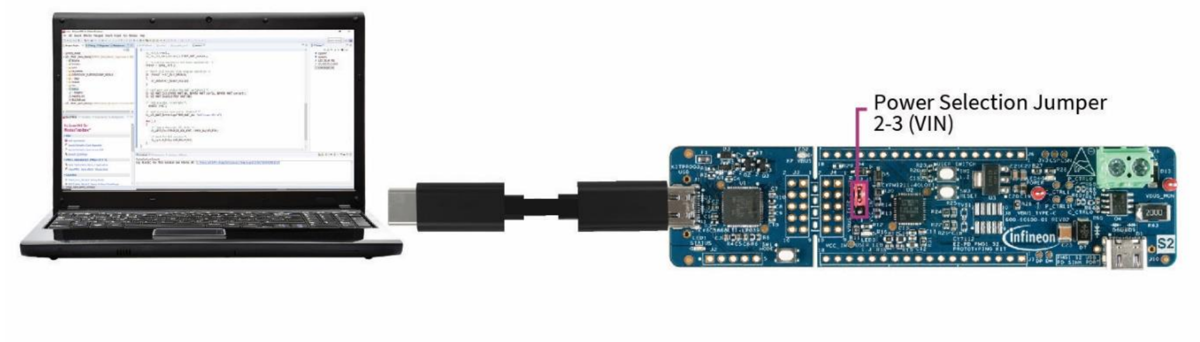


**Figure 32 Programming through the KitProg3 interface**

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

1. Open the Eclipse IDE for ModusToolbox™ software and select the PMG1 project 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. The Project Explorer on the IDE highlights the current active project as shown in **Figure 33**. If there are multiple projects, make sure to select the correct project.

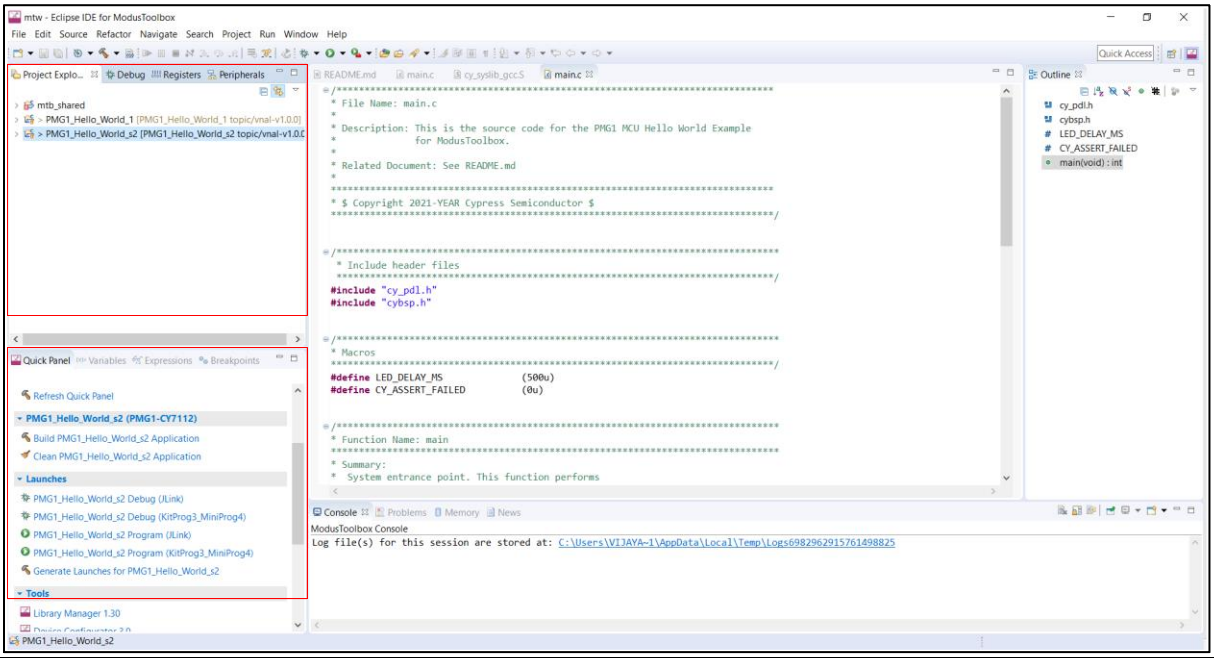

Figure 33 Eclipse IDE for ModusToolbox™ software

6. Go to the **Quick Panel** tab. Click **<Application name> Program (KitProg3_MiniProg4)** from the **Launches** section.

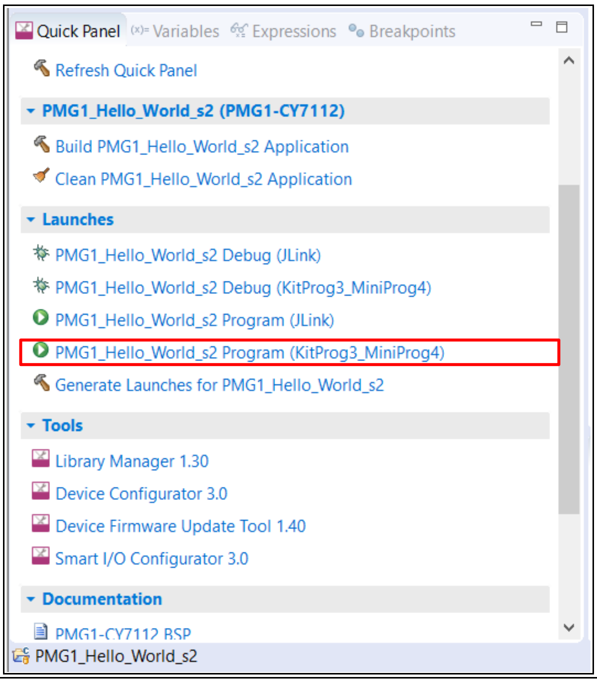


Figure 34 Quick panel and programming options

The **Console** tab displays the progress of 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.

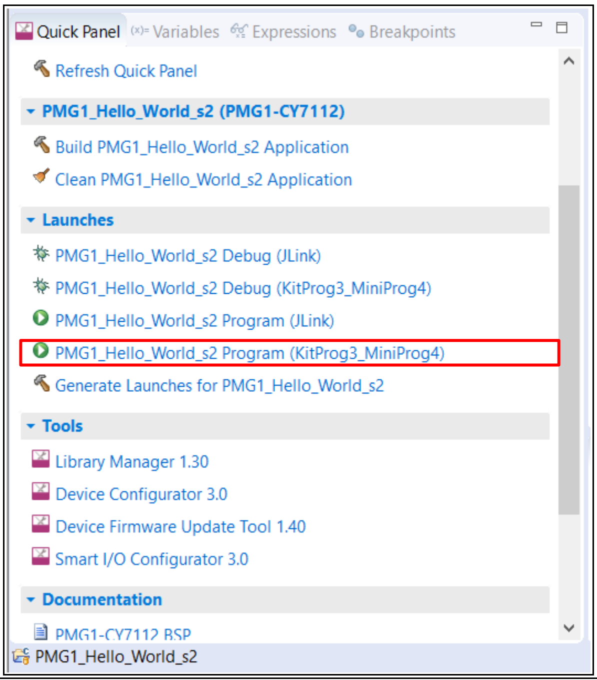


**Figure 35 Output console for build and program**

### 5.3.1.2 Using CYPRESS™ programmer (CYP)

CYP v4.0 or higher supports programming of the PMG1 prototyping kits and PMG1 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 32** 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, **Platform** will display PMG1 and the **Log** panel will display the selected device/kit/probe name.

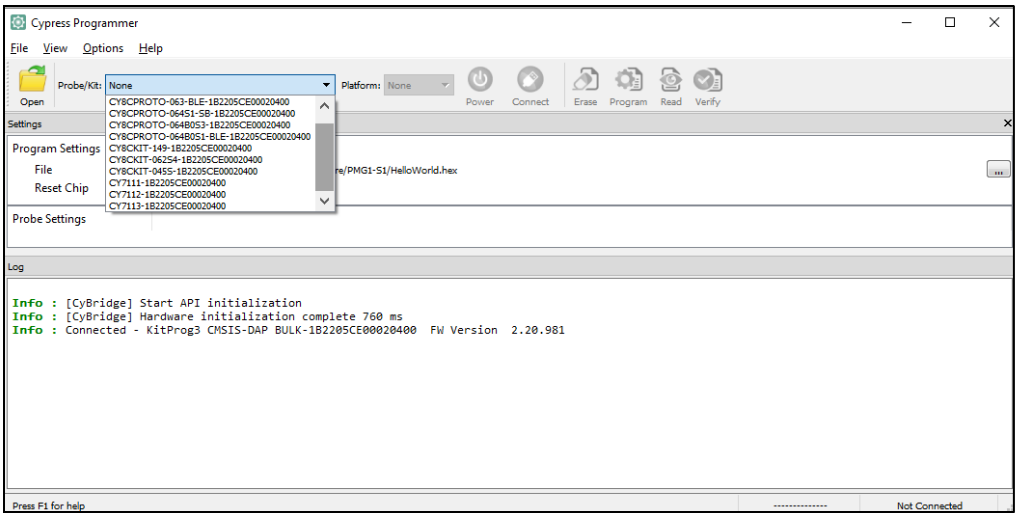


**Figure 36 Selecting the probe/kits**

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**.

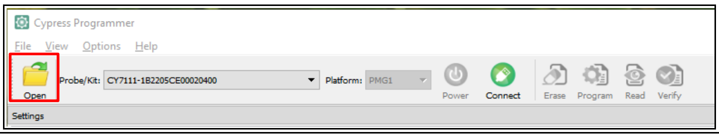


**Figure 37 Loading the programming file**

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

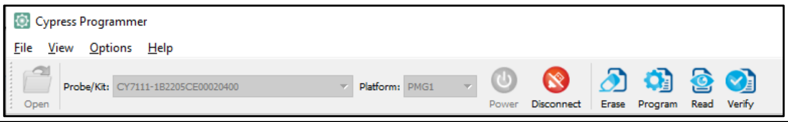


**Figure 38 Displaying the firmware options**

7. Click **Program** to initiate firmware update. The message **Device Programmed successfully** appears on the bottom left once the programming is completed.

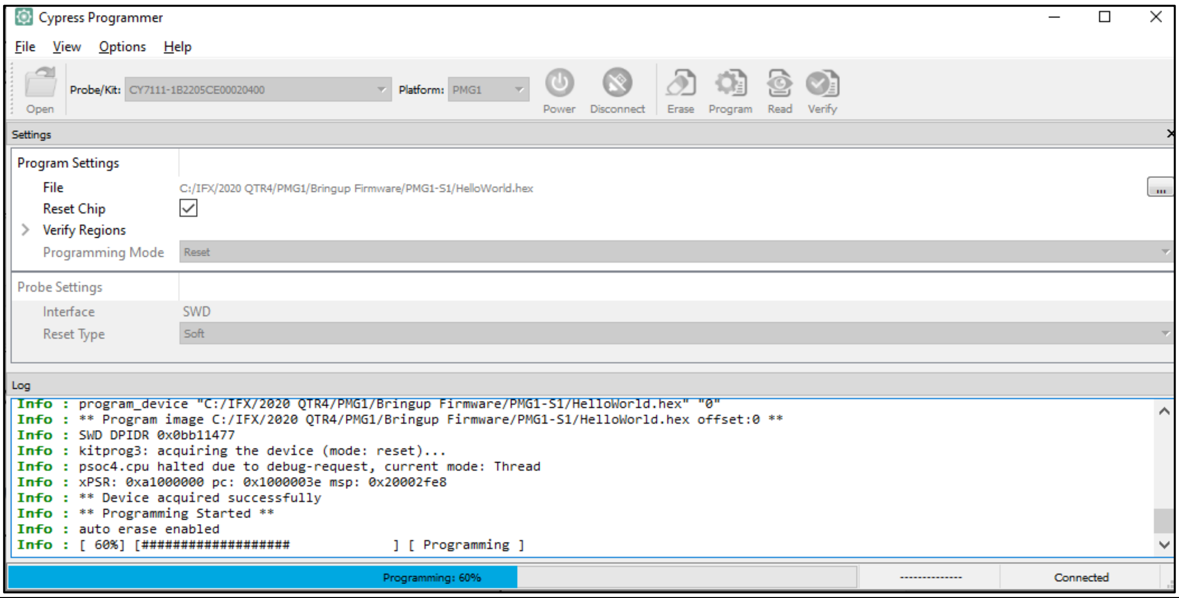


**Figure 39 Programming through CYPRESS™ programmer**

### 5.3.2 Programming though the 10-pin SWD interface

#### 5.3.2.1 Using MiniProg4 from ModusToolbox™ software

The 10-pin SWD header (J8) is 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.

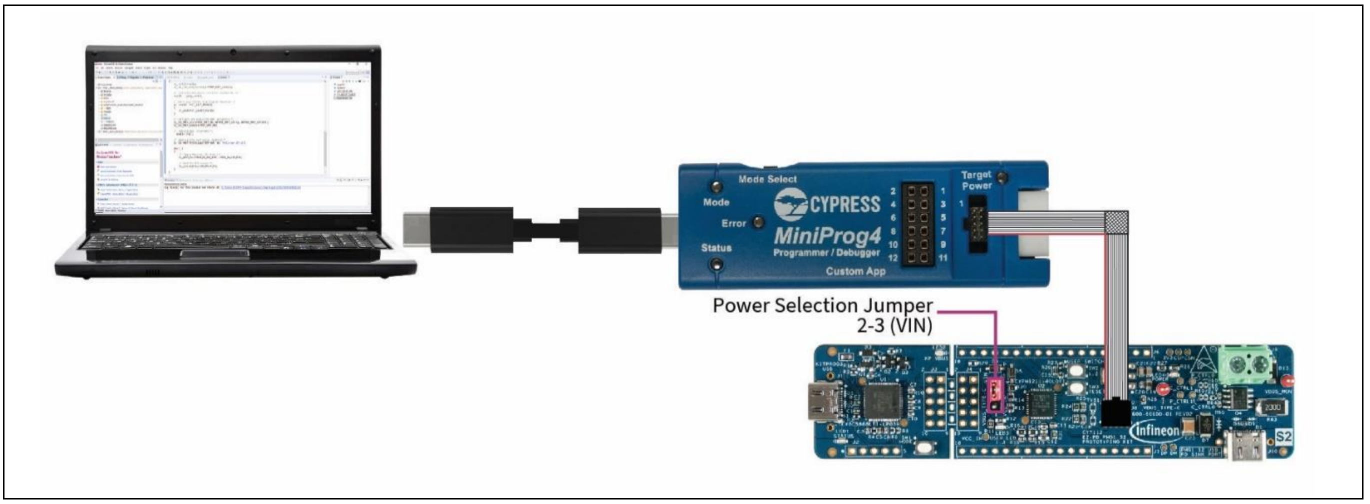


**Figure 40 Programming through the 10-pin SWD interface**

**Figure 40** shows the connection diagram for the programming 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 procedure to use Eclipse IDE for ModusToolbox™ software is the same as that of the KitProg3 interface. Follow the steps listed in **Using ModusToolbox**.

#### 5.3.2.2 Using MiniProg4 from CYP

The procedure to use the CYP is same as explained in **Using CYPRESS™ programmer**. See **Figure 40** for the connection details.

### 5.4 Debug mode

Eclipse IDE for ModusToolbox™ software can be used to debug applications. The prototyping kits can be acquired in debug mode though either of the following three interfaces similar to programming mode:

- KitProg3 USB Type-C port
- 10-pin SWD header
- I/O Header pins

The kit needs to be configured for operational mode before starting the debug.

### 5.4.1 Debugging through the KitProg3 interface

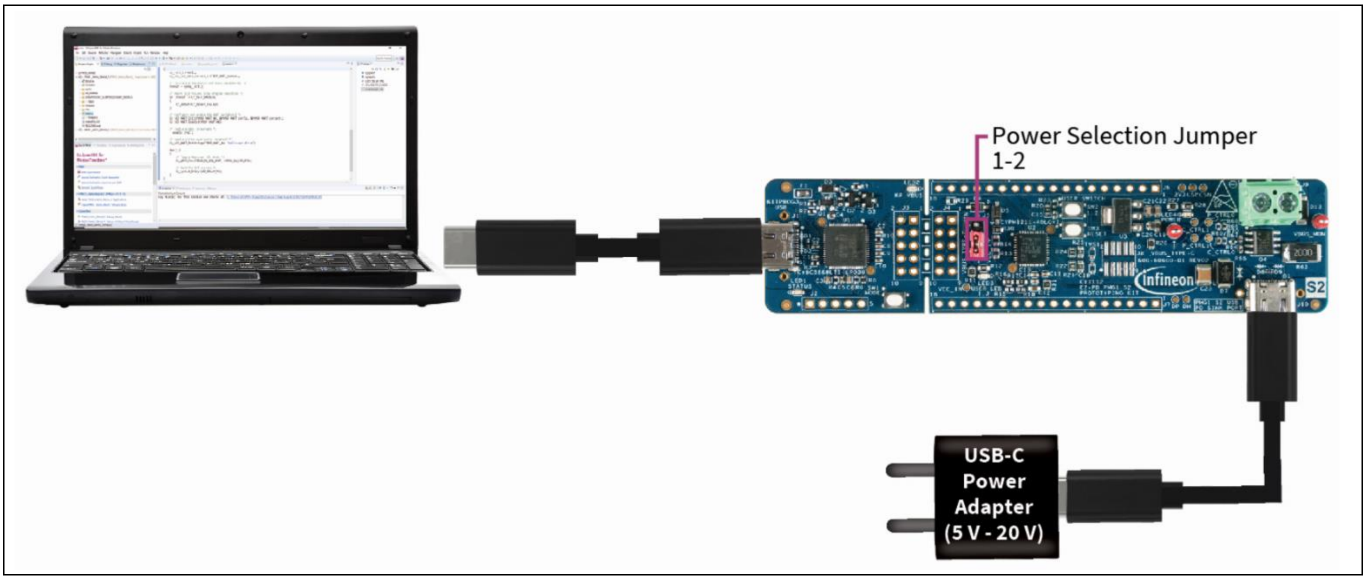


**Figure 41 Debugging though KitProg3**

Do the following to debug an application on the CY7111 and CY7112 prototyping kits over KitProg3 interface from 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 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 on-board LDO, Load switch, and user LED. Ensure that LED4, which is 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. Go to the **Quick Panel** tab. Click **<Application name> Debug (KitProg3_MiniProg4)** from the **Launches** section.
6. 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 KitProg3 interface from Eclipse IDE for ModusToolbox™ software.

1. Open the Eclipse IDE for ModusToolbox™ software and select the PMG1 project on the host PC or laptop. For details, see the Eclipse IDE for ModusToolbox™ software user guide [8].
2. Power selection jumper (J5) shunt needs to be placed on pins 2–3 of the power selection jumper (J5) to enable ModusToolbox™ software to acquire the target (PMG1-S0) in power cycle mode.
3. Connect the USB-PD sink port to the USB PD source to activate on-board LDO, Load switch, and user LED. Ensure that LED4, which is 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. Go to the **Quick Panel** tab. Click **<Application name> Program (KitProg3_MiniProg4)** from the **Launches** section to download the firmware into the kit as shown in **Figure 42**.
6. After program download completed, change the jumper to short 1–2 of the power section jumper (J5) to configure the kit in operational mode for debugging.
7. Go to **Run** menu, click on **Debug Configurations**, as shown in 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 refer MTB user guide **[7]**.

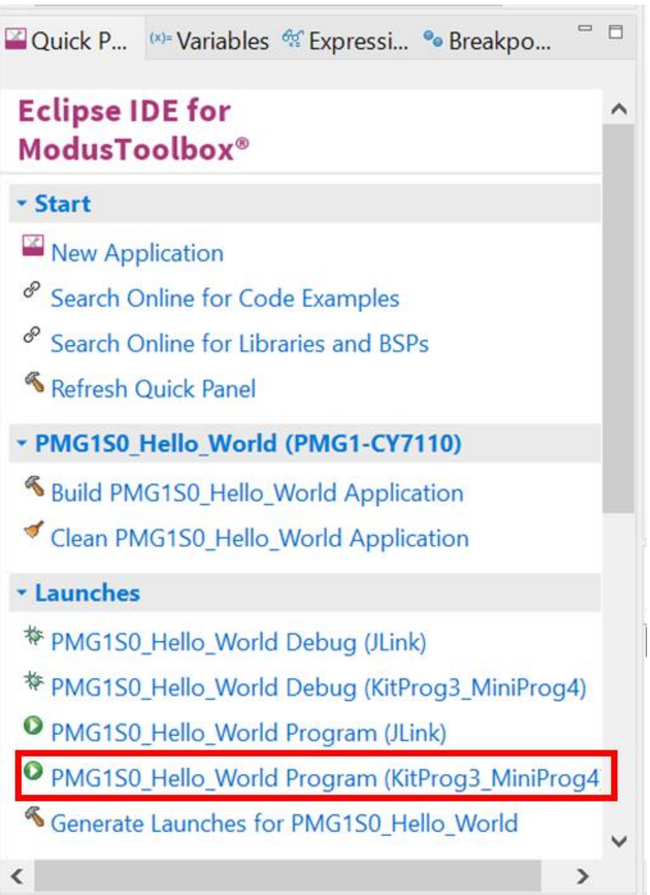


**Figure 42 Quick panel and debug options**

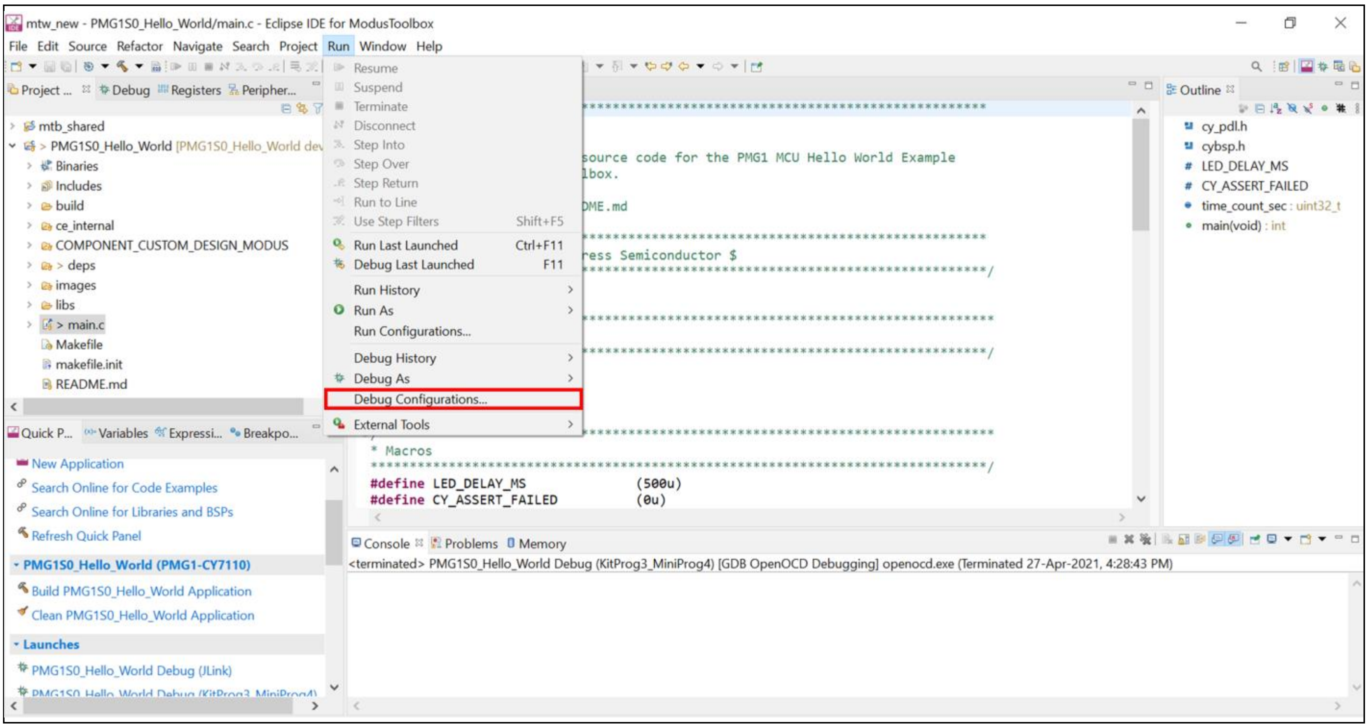


**Figure 43** Selecting Debug configurations from the Run menu

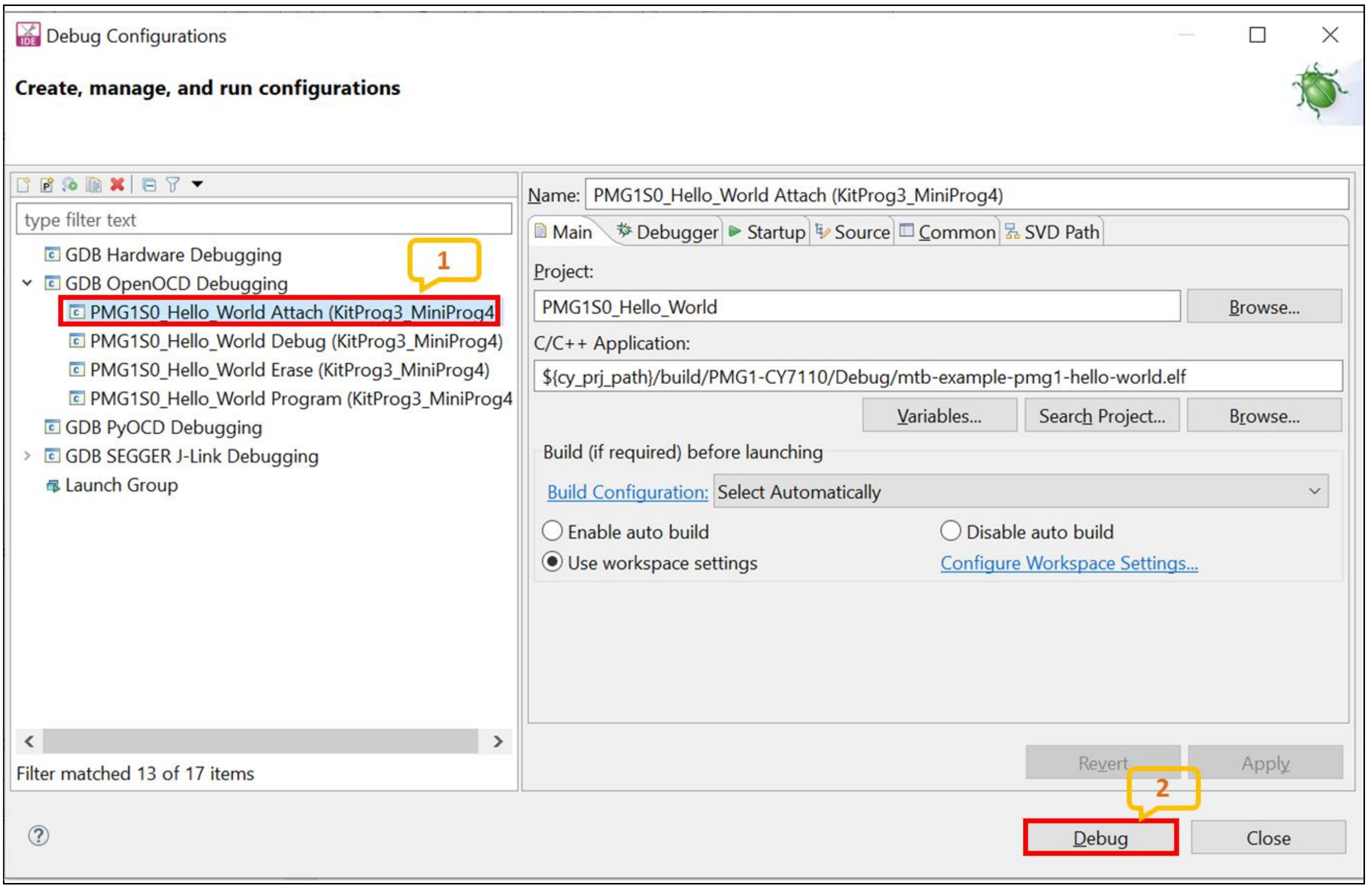


**Figure 44** Selecting Debug Attach and starting a Debug session

On successful completion of compilation, build and debug launch, ModusToolbox™ software acquires the kit in debug mode and the debug tools appear on the toolbar.

### 5.4.2 Debug through the 10-pin SWD header

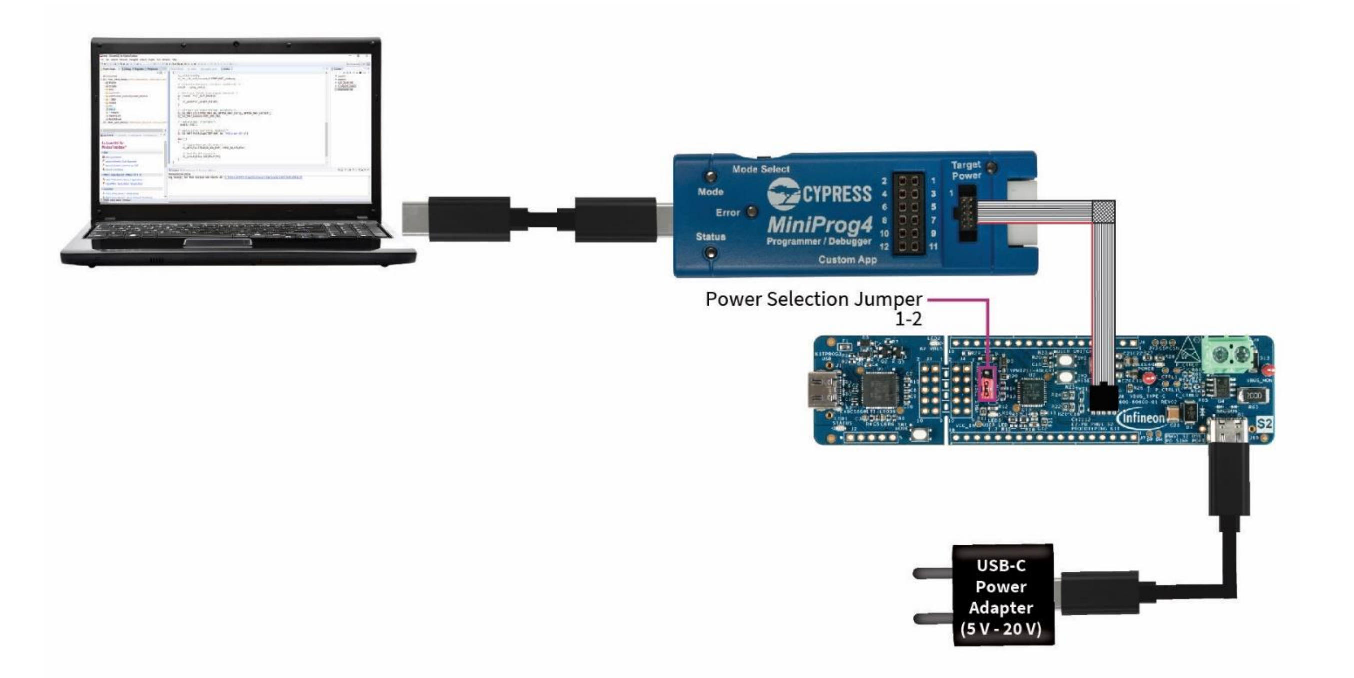


**Figure 45 Debugging through the 10-pin SWD header**

**Figure 45** shows the setup for debugging prototyping kits through 10-pin SWD connector. The procedure to use Eclipse IDE for ModusToolbox™ software is the same as that of the KitProg3 interface. Follow steps 5 to 7 listed in **Debugging through the KitProg3** interface.

## 5.5 Configuring PDOs and PD parameters

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

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. Open EZ-PD™ configurator 1.0 from the **Tools** menu in Quick Panel section.

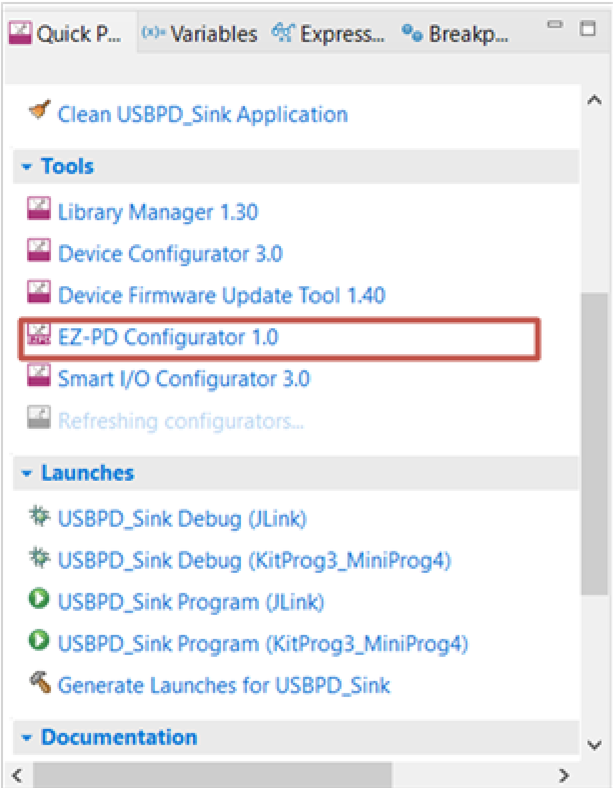


**Figure 46 Launching the EZ-PD™ configurator tool**

3. The EZ-PD™ configurator tool launches as separate window. The category of parameters available for configuration will be listed on the left-side panel. Click on the category name to expand.

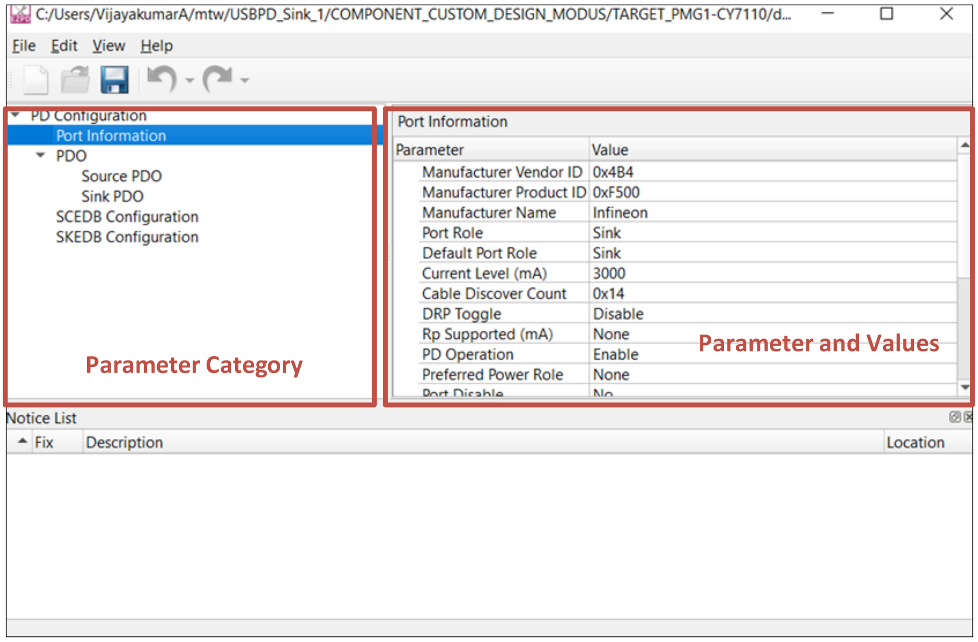


**Figure 47 EZ-PD™ configurator and list of categories**

4. Port Information, Source PDO, Sink PDO, SCEDB Configuration and SKEDB Configuration are the various parameter categories available on the tool. See the ModusToolbox™ software EZ-PD™ configurator guide for more details.

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

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

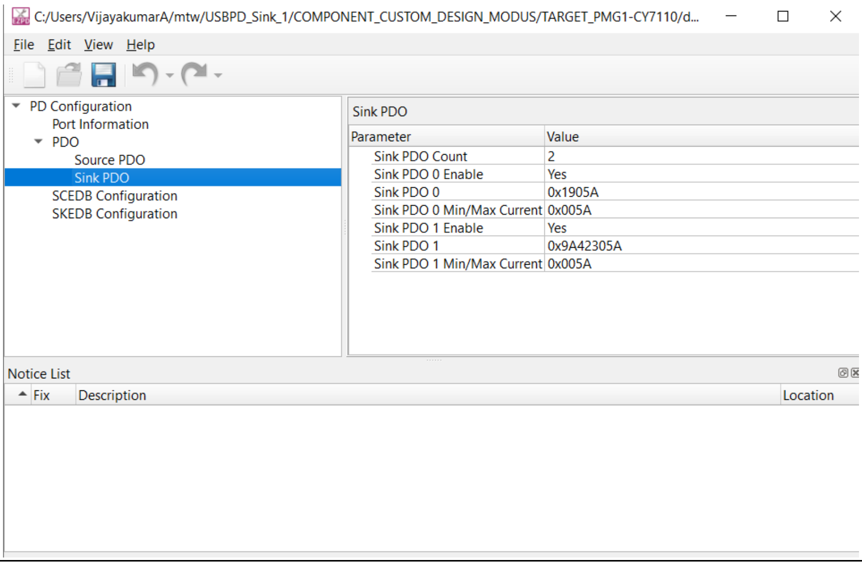


**Figure 48 Sink PDO parameters**

The following is the list of sink PDO parameters that can be configured:

- **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 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.*

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

## Related documents

[1] Overview: **Infineon USB PD controller roadmap**

[2] Product webpage:

- **EZ-PD™ PMG1 webpage**

[3] Kit webpages

- **CY7110**
- **CY7111**
- **CY7112**

[4] Datasheets

- **EZ-PD™ PMG1-S0 datasheet**
- **EZ-PD™ PMG1-S1 datasheet**
- **EZ-PD™ PMG1-S2 datasheet**

[5] **AN232553 - Getting started with EZ-PD™ PMG1 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

| Document version | Date of release | Description of changes |
|---|---|---|
| ** | 2021-06-18 | Initial release. |

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

**Edition 2021-06-18**

**Published by**

**Infineon Technologies AG**

**81726 München, Germany**



**Do you have a question about this document?**

**Go to: [www.cypress.com/support](www.cypress.com/support)**

**Document reference**

**002-32601 Rev.****