CY8CKIT-062S4 PSoC 62S4 pioneer kit guide

About this document

Scope and purpose

This guide will help you get acquainted with the CY8CKIT-062S4 PSoC™ 62S4 pioneer kit. Hardware details of the board and its usage information is provided.

Intended audience

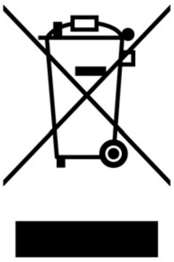
This document is intended for CY8CKIT-062S4 users.

Table of contents

Table of contents

Safety and regulatory compliance information

This kit is intended as a development platform for hardware or software in a laboratory environment. This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required authorizations are first obtained. Contact **support@cypress.com** for details.

	These boards contain electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, which can cause a discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused boards in the protective shipping package.
	End-of-Life/Product Recycling The end-of-life cycle for this kit is five years from the date of manufacture mentioned on the back of the box. Contact your nearest recycler to discard the kit.

General safety instructions

ESD protection

ESD can damage boards and associated components. Infineon recommends that you perform procedures only at an ESD workstation. If such a workstation is not available, use appropriate ESD protection by wearing an antistatic wrist strap attached to the chassis ground (any unpainted metal surface) on your board when handling parts.

Handling boards

These boards are sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad if available. Do not slide board over any surface.

1 Introduction

Thank you for your interest in the CY8CKIT-062S4 PSoC™ 62S4 pioneer kit. The PSoC™ 62S4 pioneer kit enables you to evaluate and develop your applications using the CY8C62x4 PSoC™ 62 Series MCU (hereafter called "PSoC™ 6 MCU").

PSoC™ 6 CY8C62x4 MCU is an ultra-low-power PSoC device specifically designed for battery-operated analog sensing applications. The CY8C62x4 device is a true programmable embedded system-on-chip, integrating a 150-MHz Arm® Cortex®-M4 CPU as the primary application processor, a 100-MHz Arm® Cortex®-M0+ CPU that supports low-power operations, up to 256 KB Flash and 128 KB SRAM, programmable analog sensing, CapSense™ touch-sensing, and programmable digital peripherals that allow higher flexibility, in-field tuning of the design, and faster time-to-market.

The pioneer board offers compatibility with Arduino™ shields. The board features a PSoC™ 6 MCU from the CY8C62x4 product line (CY8C6244LQI-S4D92) which is a 40-nm, ultra-low-power device that supports deep-sleep-capable SAR ADCs, opamps and other analog blocks. In addition, the board features an onboard programmer/debugger (KitProg3), a 512-Mbit Quad SPI NOR flash, a micro-B connector for USB device interface, a thermistor, an ambient light sensor, a 5-segment CapSense™ slider, two CapSense™ buttons, two user LEDs, and a push button. The board supports operating voltages from 1.8 V to 3.3 V for PSoC™ 6 MCU.

You can use ModusToolbox™ software to develop and debug your PSoC™ 6 MCU projects. **ModusToolbox™ software** is a set of tools that enable you to integrate Infineon devices into your existing development methodology.

If you are new to PSoC™ 6 MCU and Eclipse IDE for ModusToolbox™, see the application note **AN228571 - getting started with PSoC™ 6 MCU on ModusToolbox™** to help you familiarize with the PSoC™ 6 MCU and help you create your own design using ModusToolbox™.

1.1 Kit contents

The pioneer kit has the following contents, as shown in **Figure 1**.

- PSoC™ 62S4 pioneer board
- USB Type-A to Micro-B cable
- Six jumper wires (5 inches each)
- Quick start guide

Figure 1 CY8CKIT-062S4 PSoC™ 62S4 pioneer kit contents

Inspect the contents of the kit; if you find any part missing, contact your nearest Infineon sales office for help: **https://www.cypress.com/support**.

1.2 Getting started

This guide will help you get acquainted with the PSoC™ 62S4 pioneer kit:

- The **Kit Operation** chapter describes the major features of the kit and functionalities such as programming, debugging, and the USB-UART and USB-I2C bridges.
- The **Hardware** chapter provides a detailed hardware description, kit schematics, and the bill of materials (BOM).
- Application development using the PSoC™ 62S4 pioneer kit is supported in ModusToolbox™. For the latest software support for this development kit, see the **kit webpage**.
  ModusToolbox™ software is a free development ecosystem that includes the Eclipse IDE for ModusToolbox™. Using ModusToolbox™, you can enable and configure device resources, middleware libraries, and program and debug the device. You can download the software from the **ModusToolbox™ home page**. See the **ModusToolbox™ user guide** for additional information.
- There is a wide range of code examples to evaluate the PSoC™ 62S4 pioneer board. These examples help you familiarize PSoC™ 6 MCU and create your own design. These examples are available in the Eclipse IDE for ModusToolbox™ and Project Creator tool. You can also find code examples on the GitHub page dedicated to **ModusToolbox™-based examples**.

1.3 Board details

The PSoC™ 62S4 pioneer board that has the following features:

1. KitProg3 status LED (**D10**)
2. KitProg3 USB connector (**J10**)
3. PSoC™ 5LP programming header (**J7**) (footprint only)
4. 2.5 V selection jumper (**J14**)
5. PSoC™ 6 MCU power selection jumper (**J12**)
6. PSoC™ 5LP based KitProg3 programmer and debugger (CY8C5868LTI-LP039, **U7**)
7. PSoC™ 6 MCU current measurement jumper (**J13**)
8. KitProg3 programming mode selection button (**SW3**)
9. Power LED (**D14**)
10. PSoC™ 6 MCU reset button (**SW1**)
11. Arduino UNO R3 compatible power header (**J1**)
12. PSoC™ 6 MCU program and debug header (**J8**)
13. Arduino UNO R3 compatible I/O headers (**J2**, **J3**, **J4**)
14. User LEDs (**D12** and **D13**)
15. PSoC™ 6 MCU user button/hibernate wakeup (**SW2**)
16. CapSense™ buttons and slider (**CSB0**, **CSB1** and **CSS1**)
17. Infineon PSoC™ 6 MCU (CY8C6244LQI-S4D92, **U8**)
18. Thermistor (**TH1**)
19. Ambient Light Sensor (**Q8**)
20. Infineon QSPI NOR flash (S25FL512SAGMFIR10, **U9**)
21. PSoC™ 6 MCU USB connector (**J5**)
22. PSoC™ 6 MCU extended I/O header (**J11**) (footprint only)
23. External power supply connector (**J9**) (footprint only)

Figure 2 shows the pinout of the pioneer board.

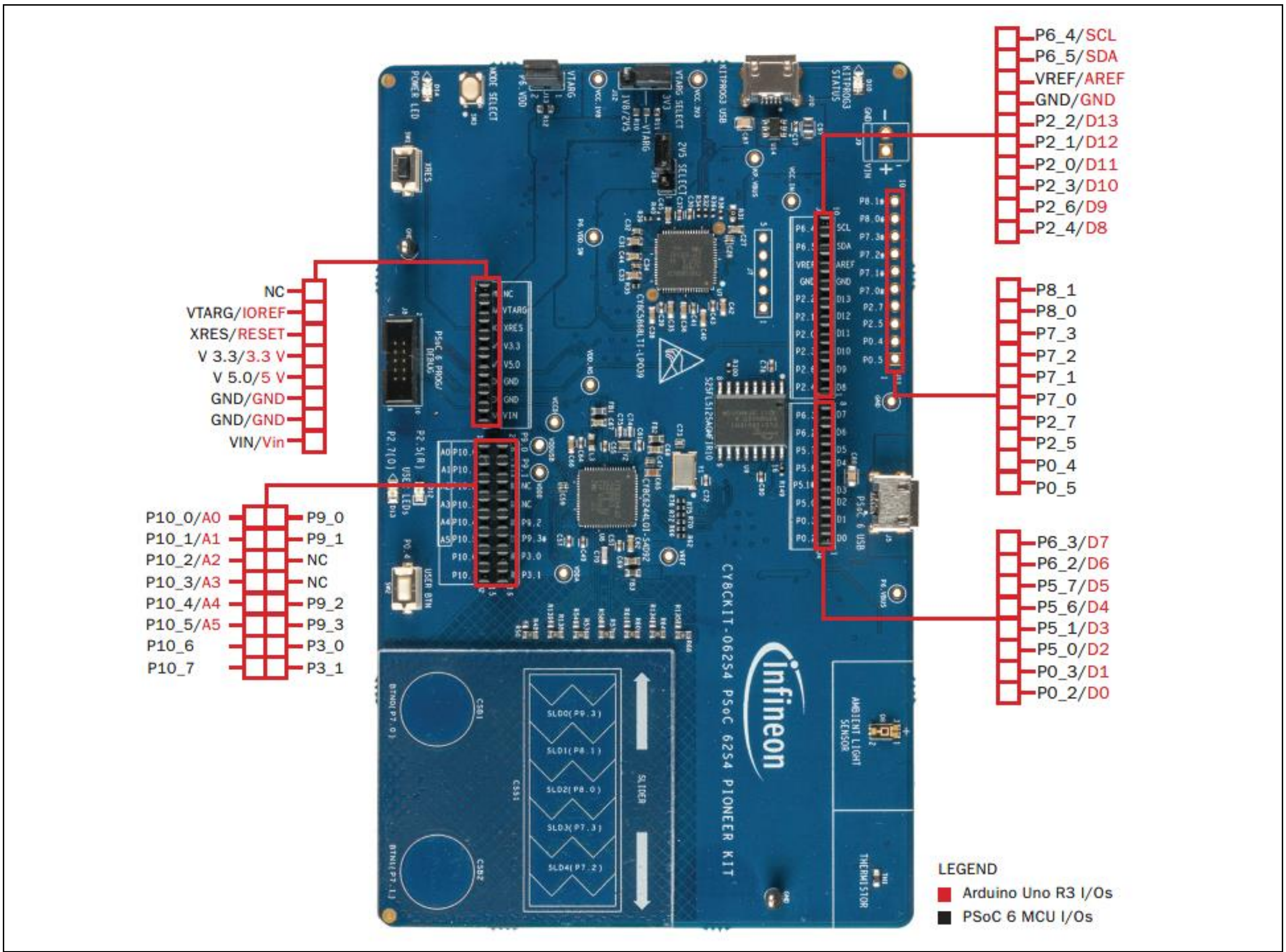


Figure 2 **CY8CKIT-062S4 PSoC™ 62S4 pioneer board pinout**

Table 1 **PSoC™ 62S4 pioneer board pinout**

Pin	Primary on-board function	Secondary on-board function	Connection details
XRES	Hardware reset	–	–
P10[0]	Arduino A0 (J2.1)	–	–
P10[1]	Arduino A1 (J2.3)	–	–
P10[2]	Arduino A2 (J2.5)	–	–
P10[3]	Arduino A3 (J2.7)	–	–
P10[4]	Thermistor supply	Arduino A4 (J2.9)	Remove R108 and mount R128 to connect to the Arduino header.
P10[5]	Thermistor output	Arduino A5 (J2.11)	Remove R110 and mount R127 to connect to the Arduino header.
P10[6]	Thermistor ground	GPIO on J2.13 Arduino header	Remove R114 and mount R130 to connect to the Arduino header.

Pin	Primary on-board function	Secondary on-board function	Connection details
P10[7]	ALS output	GPIO on J2.15 Arduino header	Remove R111 and mount R131 to connect to the Arduino header.
P3[0]	KitProg3 UART_TX	GPIO on J2.14 Arduino header	Remove R86 to disconnect from KitProg3.
P3[1]	KitProg3 UART_RX	GPIO on J2.16 Arduino header	Remove R85 to disconnect from KitProg3.
P9[0]	GPIO on J2.2 Arduino header	–	–
P9[1]	GPIO on J2.4 Arduino header	–	–
P9[2]	GPIO on J2.10 Arduino header	–	–
P9[3]	CapSense™ slider 0	GPIO on J2.12 Arduino header	Remove R68 and mount R135 to connect to the Arduino header.
P6[4]	I2C SCL	JTAG _TDO	Remove R74 and mount R76 for JTAG enable.
P6[5]	I2C SDA	JTAG _TDI	Remove R79 and mount R81 for JTAG enable.
P2[2]	Arduino D13 (J3.6)	–	–
P2[1]	Arduino D12(J3.5)	–	–
P2[0]	Arduino D11 (J3.4)	–	–
P2[3]	Arduino D10 (J3.3)	–	–
P2[6]	Arduino D9 (J3.2)	–	–
P2[4]	Arduino D8 (J3.1)	–	–
P6[3]	Arduino D7 (J4.8)	–	–
P6[2]	Arduino D6 (J4.7)	–	–
P5[7]	Arduino D5 (J4.6)	–	–
P5[6]	Arduino D4 (J4.5)	–	–
P5[1]	P6 USB VBUS detect	Arduino D3 (J4.4)	Remove R124 and mount R125 to connect to the Arduino header.
P5[0]	Arduino D2 (J4.3)	–	–
P0[3]	Arduino UART_TX/Arduino D1 (J4.2)	–	–
P0[2]	Arduino UART_RX/Arduino D0 (J4.1)	–	–
P0[5]	GPIO on extended header J11.1	–	–
P0[4]	User button/Hibernate wakeup	GPIO on extended header J11.2	–
P2[5]	Red user LED (D12)	GPIO on extended header J11.3	–

Pin	Primary on-board function	Secondary on-board function	Connection details
P2[7]	Orange user LED (D13)	GPIO on extended header J11.4	–
P7[0]	CapSense™ button 0	GPIO on extended header J11.5	Remove R48 and mount R50 to connect to the extended header.
P7[1]	CapSense™ button 1	GPIO on extended header J11.6	Remove R138 and mount R139 to connect to the extended header.
P7[2]	CapSense™ slider 4	GPIO on extended header J11.7	Remove R53 and mount R54 to connect to the extended header.
P7[3]	CapSense™ slider 3	GPIO on extended header J11.8	Remove R57 and mount R58 to connect to the extended header.
P8[0]	CapSense™ slider 2	GPIO on extended header J11.9	Remove R60 and mount R61 to connect to the extended header.
P8[1]	CapSense™ slider 1	GPIO on extended header J11.10	Remove R64 and mount R134 to connect to the extended header.

1.4 Additional learning resources

Infineon provides a wealth of data at **https://www.cypress.com/psoc6** to help you to select the right PSoC™ device for your design and to help you to quickly and effectively integrate the device into your design.

1.5 Technical support

Go to **www.cypress.com/support**.

1.6 Documentation conventions

Table 2 Document conventions for guides

Convention	Usage
`Courier New`	Displays user entered text, and source code: `C:\...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC™ Creator User Guide*.
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes Cautions or unique functionality of the product.

1.7 Abbreviations and definitions

Table 3 Abbreviations

Abbreviation	Definition
ADC	Analog-to-Digital Converter
ALS	Ambient Light Sensor
BOM	Bill of Materials
CMOD	Modulator Capacitor
CPU	Central Processing Unit
CSD	CapSense™ Sigma-Delta
DC	Direct Current
ECO	External Crystal Oscillator
ESD	Electrostatic Discharge
GPIO	General Purpose Input/Output
HID	Human Interface Device
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
LED	Light Emitting Diode
PSoC™	Programmable System on Chip
PWM	Pulse Width Modulation
QSPI	Quad Serial Peripheral Interface
SAR	Successive Approximation Register
SWD	Serial Wire Debug
WCO	Watch Crystal Oscillator

2 Kit operation

This chapter introduces you to various features of the PSoC™ 62S4 pioneer board, including the theory of operation and the onboard KitProg3 programming and debugging functionality, USB-UART and USB-I2C bridges.

2.1 Theory of operation

The PSoC™ 62S4 pioneer board is built around the CY8C62x4 product line, based on the PSoC™ 6 MCU platform. **Figure 3** shows the block diagram of the PSoC™ 6 MCU device used on the board. For details of device features, see the device datasheet.

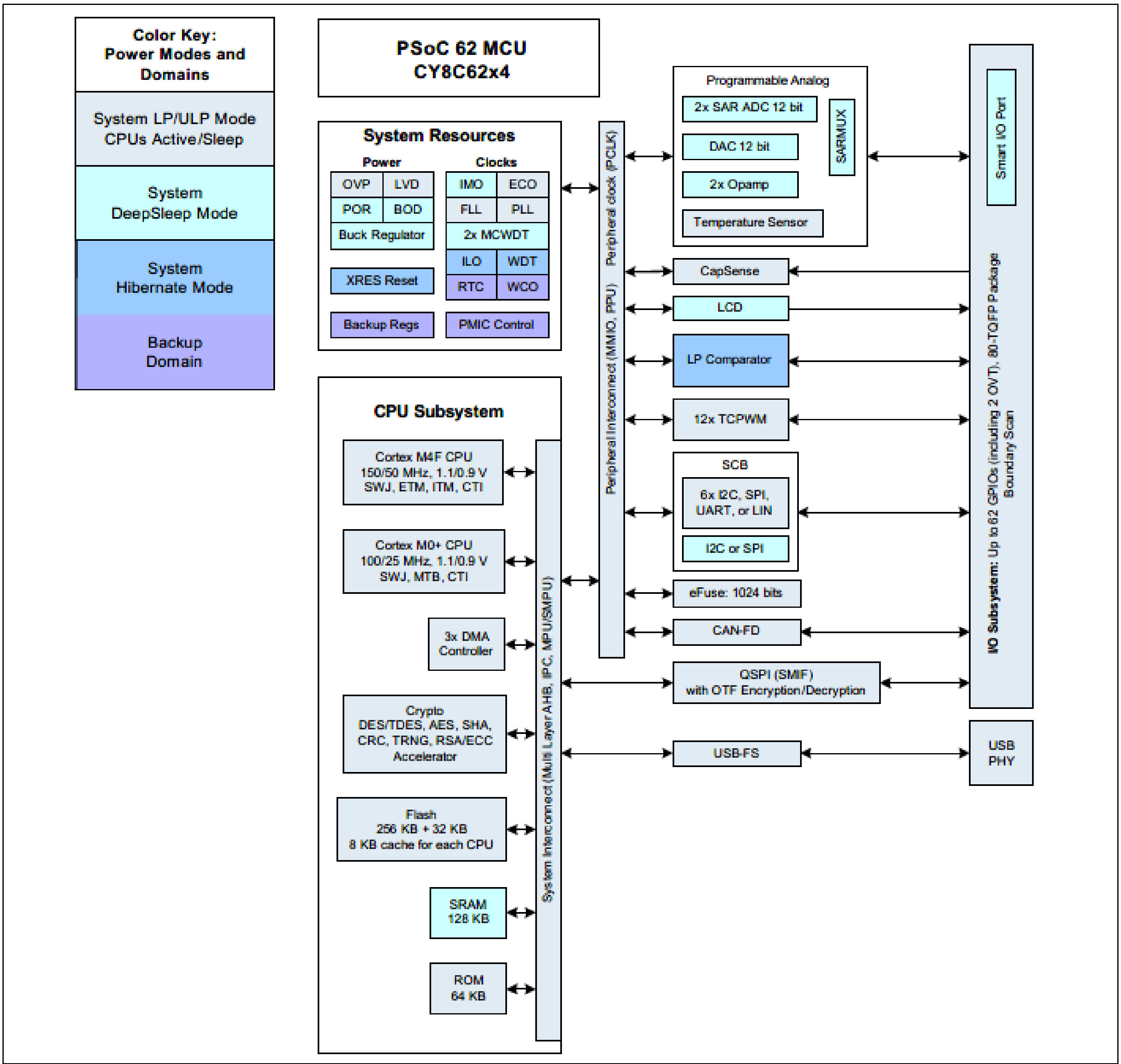


Figure 3 PSoC™ 62S4 MCU block diagram

Kit operation

Figure 4 shows the functional block diagram of the pioneer board.

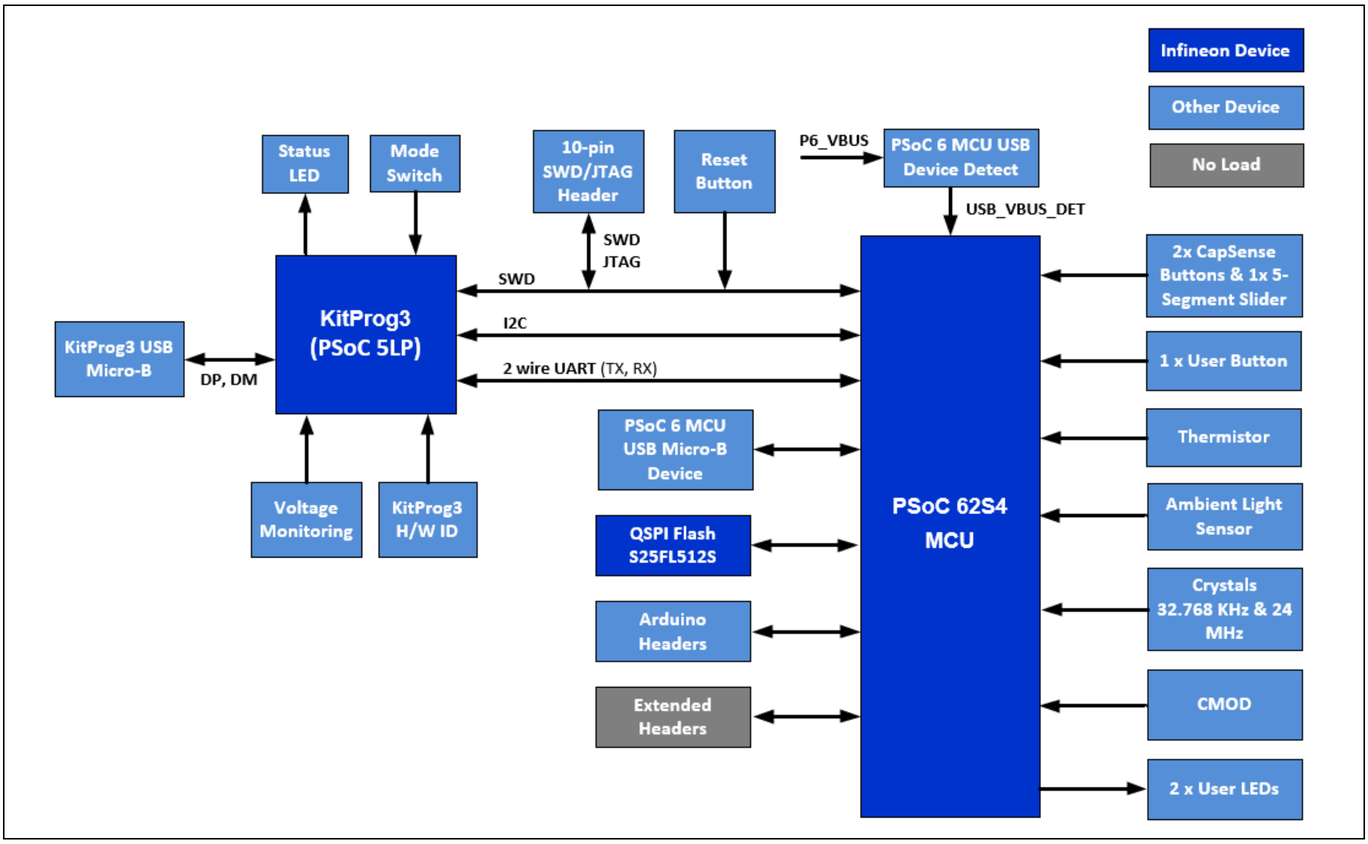


Figure 4 **Functional block diagram of CY8CKIT-062S4 PSoC™ 62S4 pioneer kit**

The CY8CKIT-062S4 PSoC™ 62S4 pioneer kit comes with the PSoC™ 62S4 pioneer board. **Figure 5** shows the markup of the PSoC™ 62S4 pioneer board.

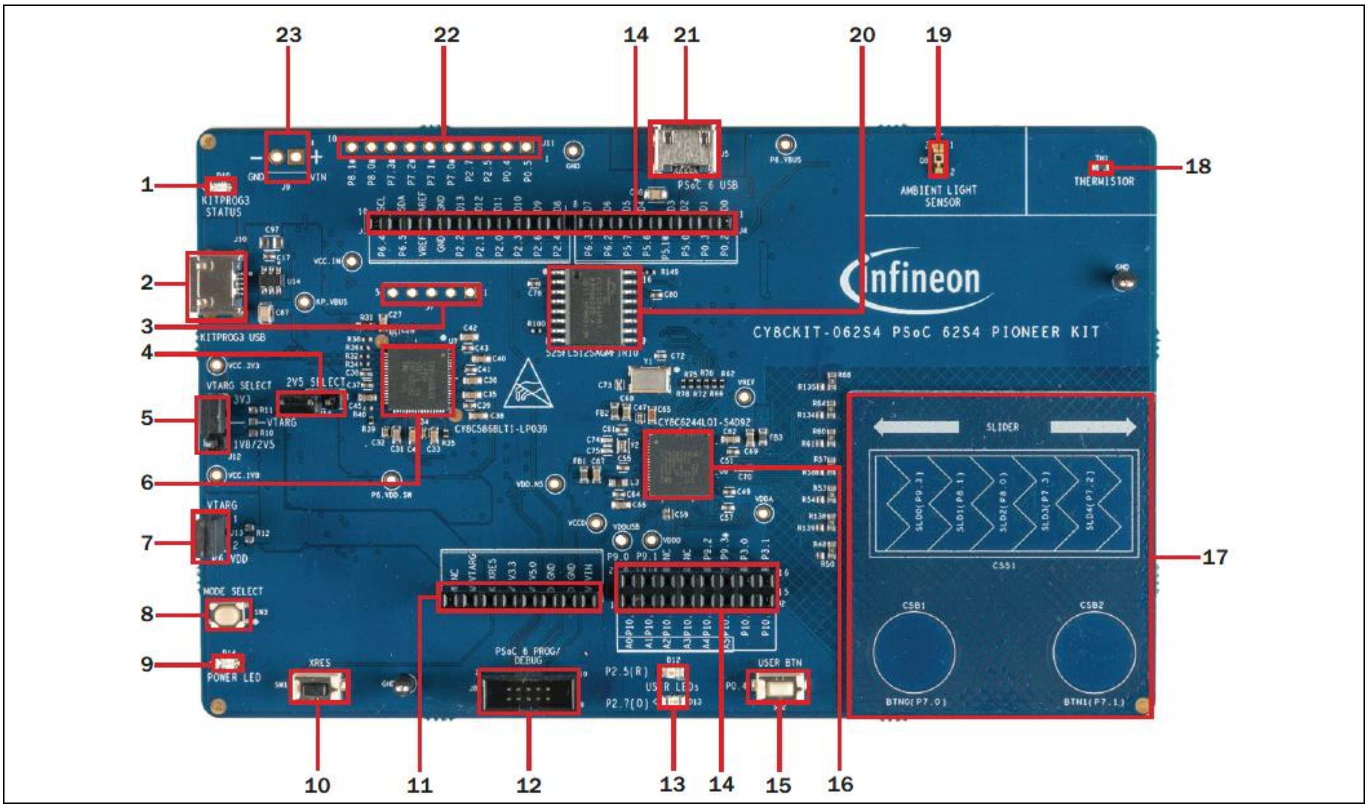


Figure 5 CY8CKIT-062S4 PSoC™ 62S4 pioneer board details

The PSoC™ 62S4 pioneer board has the following peripherals:

1. **KitProg3 status LED (D10)**: This Yellow LED indicates the status of KitProg3. For details on the KitProg3 status, see the **KitProg3 user guide**.
2. **KitProg3 USB connector (J10)**: The USB cable provided along with the PSoC™ 62S4 pioneer board connects between this USB connector and the PC to use the KitProg3 onboard programmer and debugger and to provide power to the board.
3. **PSoC™ 5LP programming header (J7)**: The header is used to program the onboard PSoC™ 5LP device. This is not loaded by default.
4. **2.5 V selection jumper (J14):** The jumper is used to select 2.5 V power output for the e-Fuse programming through the VCC_1V8 rail. Shorting the jumper provides 2.5 V power and keeping the jumper open provides 1.8 V.
5. **PSoC™ 6 MCU VDD power selection jumper (J12)**: This jumper is used to select the PSoC™ 6 MCU VDD supply voltage between 1.8 V/2.5 V and 3.3 V.
6. **KitProg3 (PSoC™ 5LP) programmer and debugger (CY8C5868LTI-LP039, U7)**: The PSoC™ 5LP device (CY8C5868LTI-LP039) serving as KitProg3 is a multi-functional system which includes a SWD programmer, debugger, USB-I2C bridge and USB-UART bridge. For more details, see the **KitProg3 user guide**.
7. **PSoC™ 6 MCU VDD current measurement jumper (J13)**: An ammeter can be connected to this jumper to measure the current consumed by the PSoC™ 6 MCU.
8. **KitProg3 programming mode selection button (SW3)**: This button can be used to switch between various modes of operation of KitProg3 (CMSIS-DAP BULK or CMSIS-DAP HID). For more details, see the **KitProg3 user guide**.

9. **Power LED (D14)**: This yellow LED indicates the status of power supplied to board.
10. **PSoC™ 6 MCU reset button (SW1)**: This button is used to reset PSoC™ 6 MCU. It connects the PSoC™ 6 MCU reset (XRES) pin to ground when pressed.
11. **Arduino-compatible power header (J1)**: This header powers Arduino shields. It also has a provision to power the kit though the VIN input.
12. **PSoC™ 6 MCU program and debug header (J8)**: This 10-pin header allows you to program and debug the PSoC™ 6 MCU using an external programmer such as **MiniProg4**.
13. **User LEDs (D12 and D13)**: These two user LEDs can operate at the entire operating voltage range of PSoC™ 6 MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LED.
14. **Arduino Uno R3-compatible I/O headers (J2, J3, and J4)**: These I/O headers bring out pins from PSoC™ 6 MCU to interface with Arduino shields. Some of these pins are multiplexed with onboard peripherals and are not connected to PSoC™ 6 MCU by default. For a detailed information on how to rework the kit to access these pins, see **Table 1** or **PSoC™ 62S4 pioneer board rework**.
15. **PSoC™ 6 MCU user button (SW2)**: This button can be used to provide an input to P0[4] pin of PSoC™ 6 MCU. Note that this button connects the PSoC™ 6 MCU pin to ground when pressed, so you need to configure the PSoC™ 6 MCU pin as a 'digital input with resistive pull-up' for detecting the button press. This button also provides a wake-up source from low-power modes of the device.
16. **PSoC™ 6 MCU (CY8C6244LQI-S4D92)**: This kit is designed to highlight the features of the PSoC™ 6 MCU. For details on PSoC™ 6 MCU pin mapping, refer to **Table 1**.
17. **CapSense™ slider (CSS1) and buttons (CSB1 and CSB2)**: The CapSense™ touch-sensing slider and two buttons, all of which are capable of self-capacitance (CSD) operation, allow you to evaluate the fourth-generation CapSense™ technology. The slider and buttons have a 1-mm acrylic overlay for smooth touch sensing.
18. **Thermistor (TH1):** An analog temperature sensor allows you to evaluate low-power analog blocks of the PSoC™ 6 MCU.
19. **Ambient light sensor (Q8)**: An analog light sensor allows you to evaluate low-power analog blocks of the PSoC™ 6 MCU.
20. **Serial NOR flash memory (S25FL512S, U9)**: The S25FL512S NOR flash of 512-Mbit capacity is connected to the Quad SPI interface of the PSoC™ 6 MCU. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption.
21. **PSoC™ 6 MCU USB device connector (J5)**: The USB cable provided with the PSoC™ 62S4 pioneer kit can also be connected between this USB connector and the PC to use the PSoC™ 6 MCU USB device applications.
22. **PSoC™ 6 MCU extended I/O header (J11)**: This header provides connectivity to PSoC™ 6 MCU GPIOs that are not connected to the Arduino-compatible headers. Some of these I/Os are also connected to on-board peripherals. See **Table 1** for pin mapping. This is not loaded by default.
23. **External power supply VIN connector (J9)**: This connector connects an external DC power supply input to the onboard regulators. This is not loaded by default.

See **Hardware Functional Description** for details on various hardware blocks.

2.2 KitProg3: on-board programmer/debugger

The PSoC™ 62S4 pioneer board can be programmed and debugged using the onboard KitProg3. KitProg3 is an onboard programmer/debugger with USB-UART and USB-I2C functionality. A PSoC™ 5LP device is used to implement KitProg3 functionality. For more details on the KitProg3 functionality, see the **KitProg3 user guide**.

2.2.1 Programming and debugging using ModusToolbox™

This section presents a quick overview of programming and debugging using ModusToolbox™. For detailed instructions, see **Help > ModusToolbox IDE Documentation > User Guide**.

1. Connect the board to the PC using the USB cable, as shown in **Figure 6**. It enumerates as a USB composite device if you are connecting it to your PC for the first time. KitProg3 can operate either in CMSIS-DAP Bulk mode (default) or CMSIS-DAP HID mode. Programming is faster with the Bulk mode. The status LED (Yellow) is always ON in Bulk mode, ramping at 1-Hz rate in HID mode. see the **KitProg3 user guide** for details on the KitProg3 status and troubleshooting instructions.

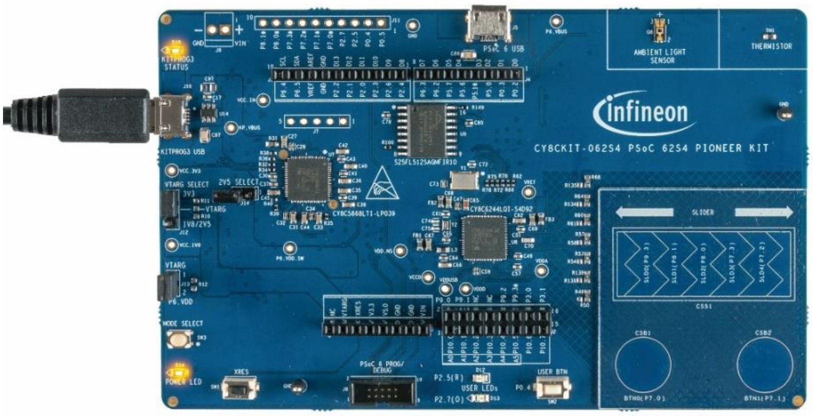


Figure 6 Connect USB cable to USB connector on the board

2. In the Eclipse IDE for ModusToolbox™, create the desired code example (application) in a new workspace.

a) Click on **New Application** from Quick Panel.

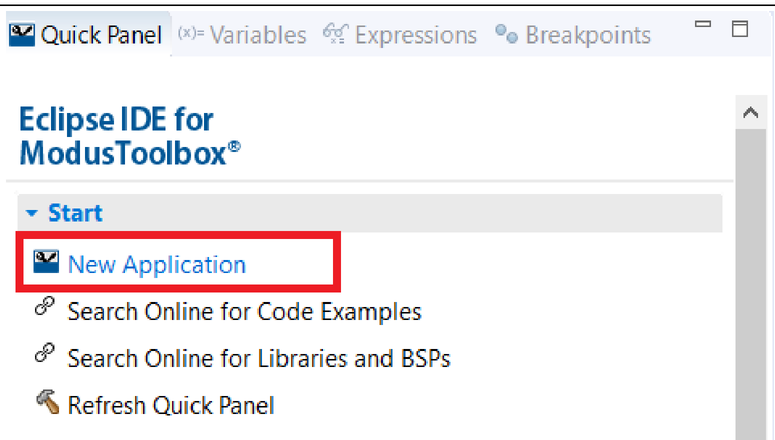


Figure 7 Starting a new application

b) Select the CY8CKIT-062S4 in the “Choose Board Support Package (BSP)” window and click **Next**, as shown in **Figure 8**.

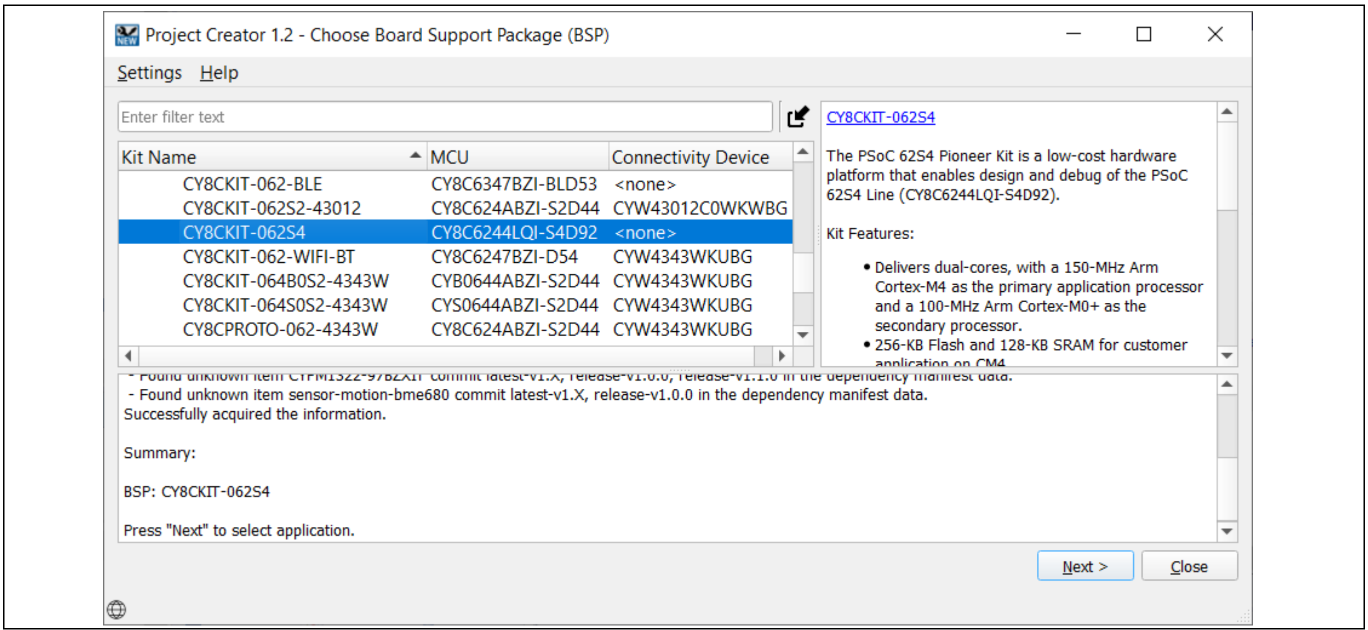


Figure 8 **Board Support Package (BSP) window**

c) Select the application and click **Create**, as shown in **Figure 9**.

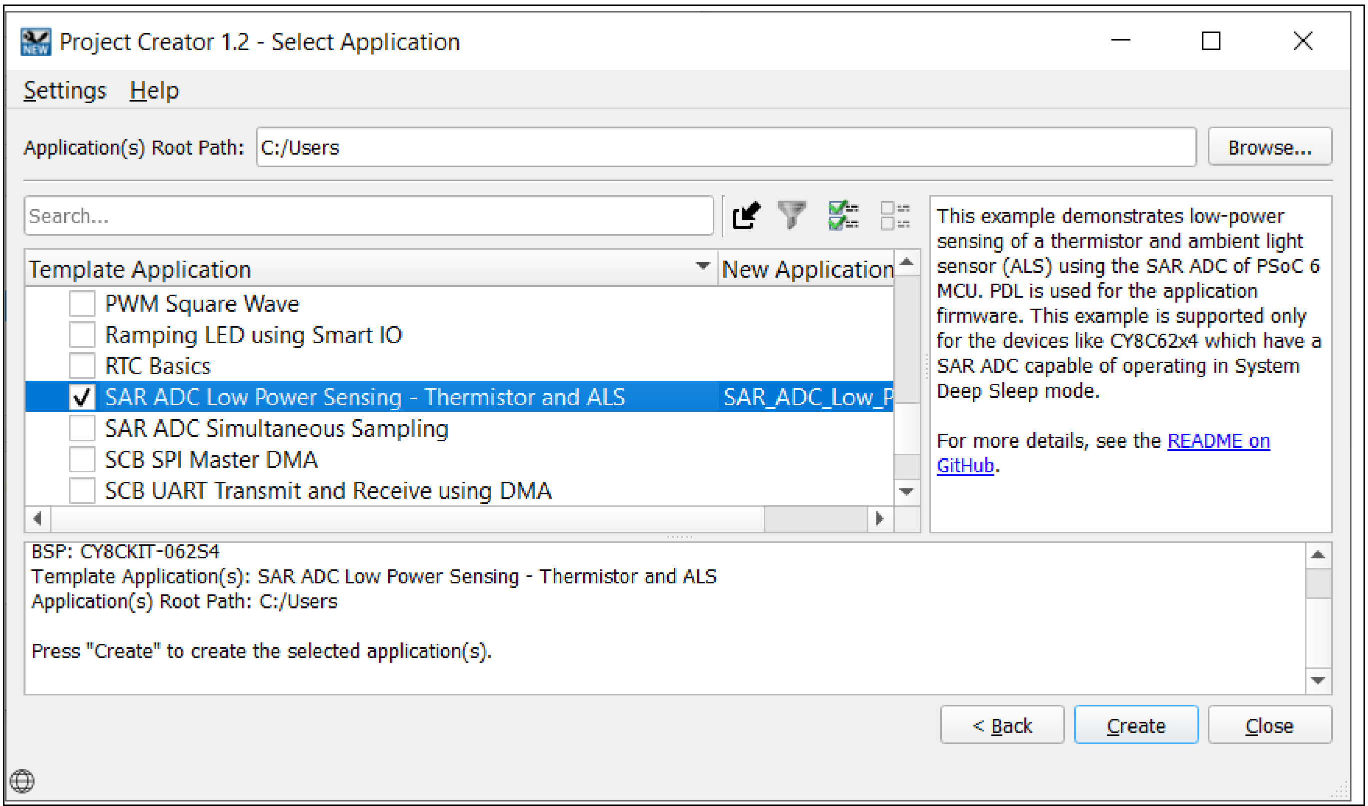


Figure 9 **CY8CKIT-062S4 application selection**

3. To build and program a PSoC™ 6 MCU application, in the Project Explorer, select <App_Name> project. In the Quick Panel, scroll to the **Launches** section and click the **<App_Name> Program (KitProg3_MiniProg4)** configuration as shown in **Figure 10**.

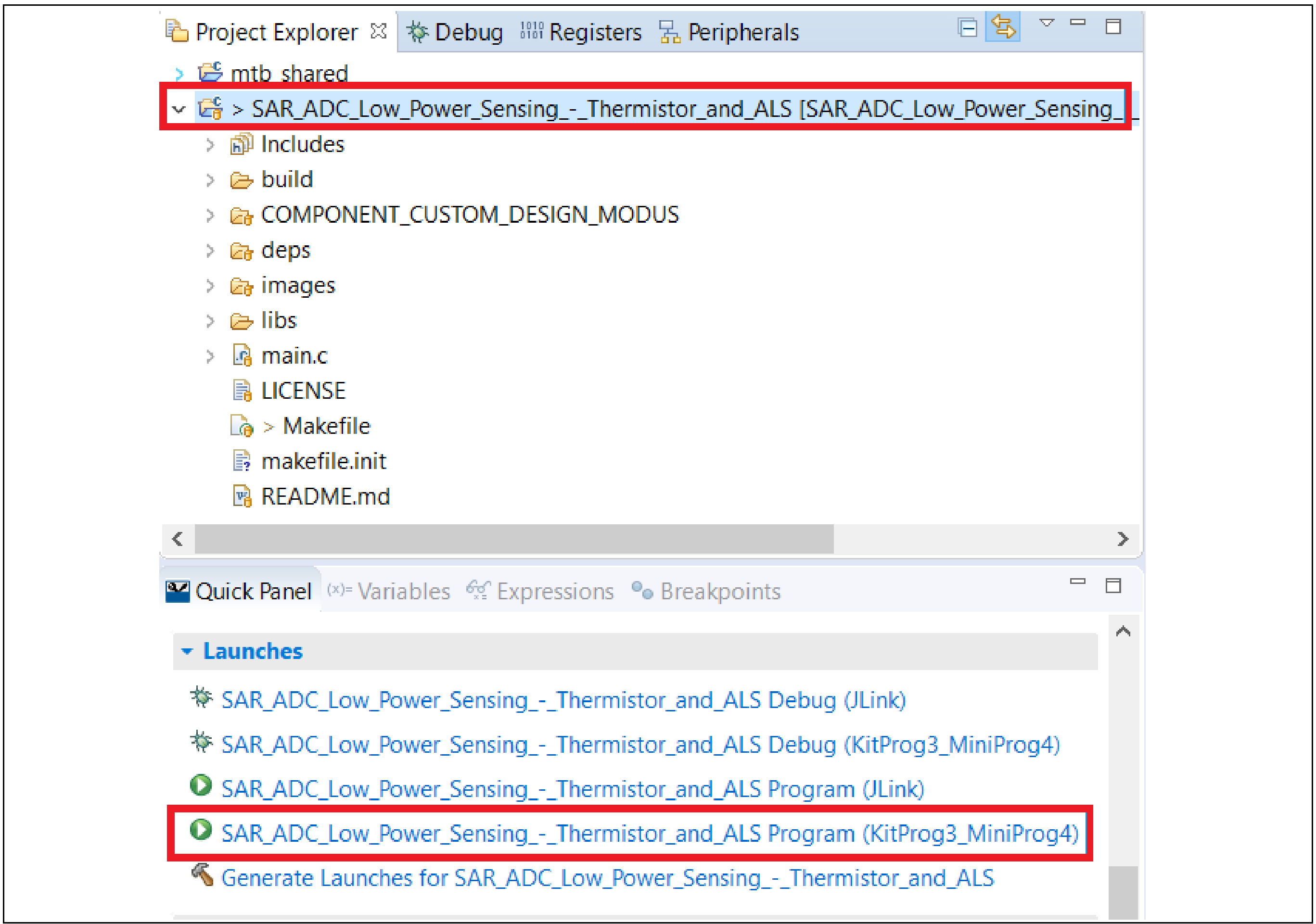


Figure 10 Build and program the code example

4. ModusToolbox™ has an integrated debugger. To debug a PSoC™ 6 MCU application, in the Project Explorer, select <App_Name> project. In the Quick Panel, scroll to the **Launches** section and click the <App_Name> Debug (KitProg3_MiniProg4) configuration as shown in **Figure 11**. For a detailed explanation on how to debug using ModusToolbox™, see the **ModusToolbox™ user guide**.

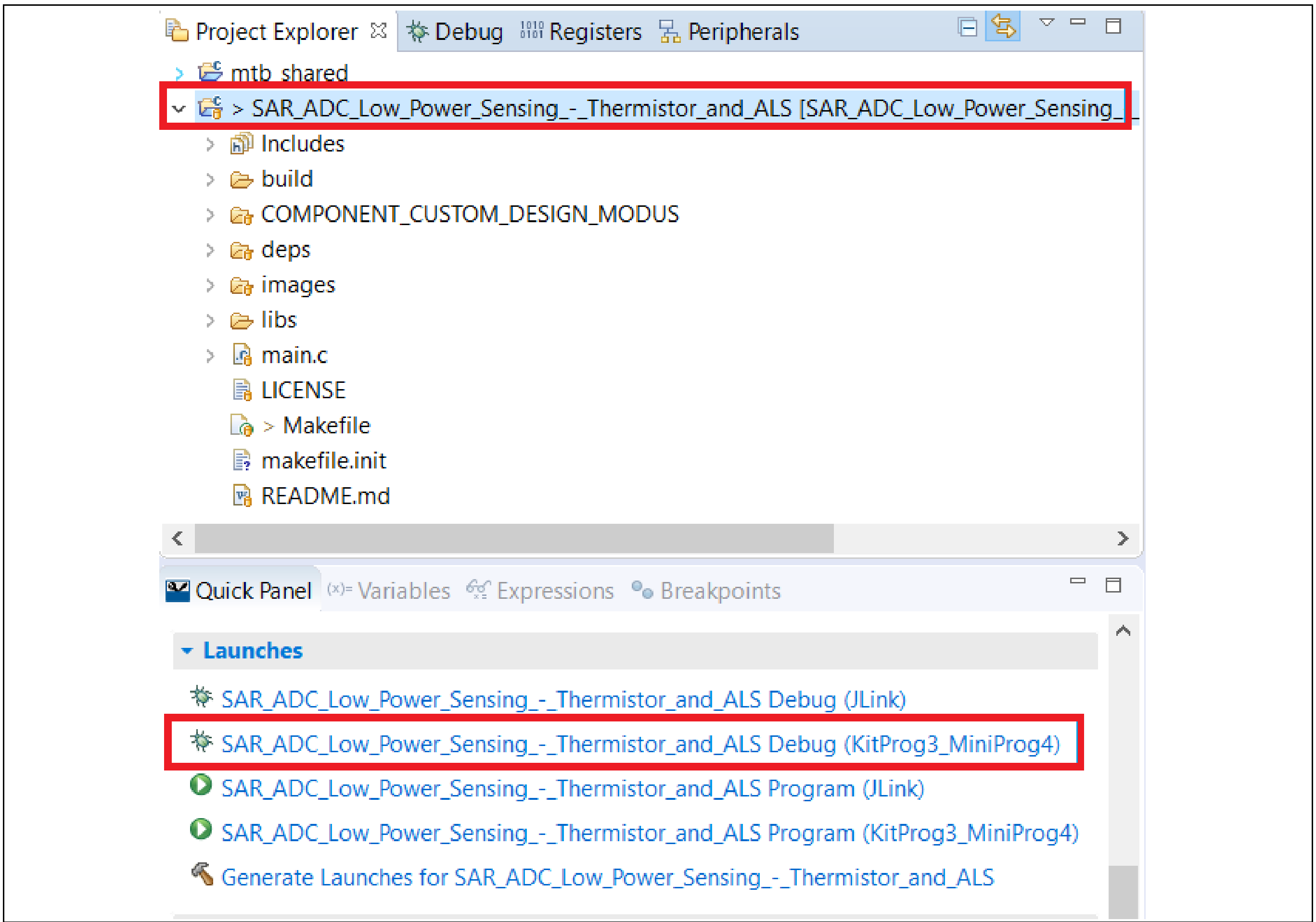


Figure 11 Debug the code example

2.2.1.1 Using the OOB example – PSoC™ 6 MCU: SAR ADC low-power sensing – thermistor and ambient light sensor

The PSoC™ 62S4 pioneer board is by default programmed with the code example: **PSoC™ 6 MCU: SAR ADC low-power sensing – thermistor and ambient light sensor** which showcases the low-power sensing application of the kit by putting the SAR ADC in Deep Sleep mode. The steps below describe how to use the example. For a detailed description of the project, see the example's readme file in the GitHub repository.

Note: *At any point of time, if you overwrite the OOB example, you can restore it by programming the PSoC™ 6 MCU: "SAR ADC low-power sensing – thermistor and ambient light sensor" project onto the kit. See* ***Programming and debugging using ModusToolbox™*** *for programming the board.*

1. Connect the board to your PC using the provided USB cable through the KitProg3 USB connector.
2. Open a terminal program and select the KitProg3 COM port. Set the serial port parameters to 8N1 and 115200 baud.
3. Press the reset button (**SW1**) on the board and confirm that terminal application displays the code example title and other text as **Figure 12** shows.

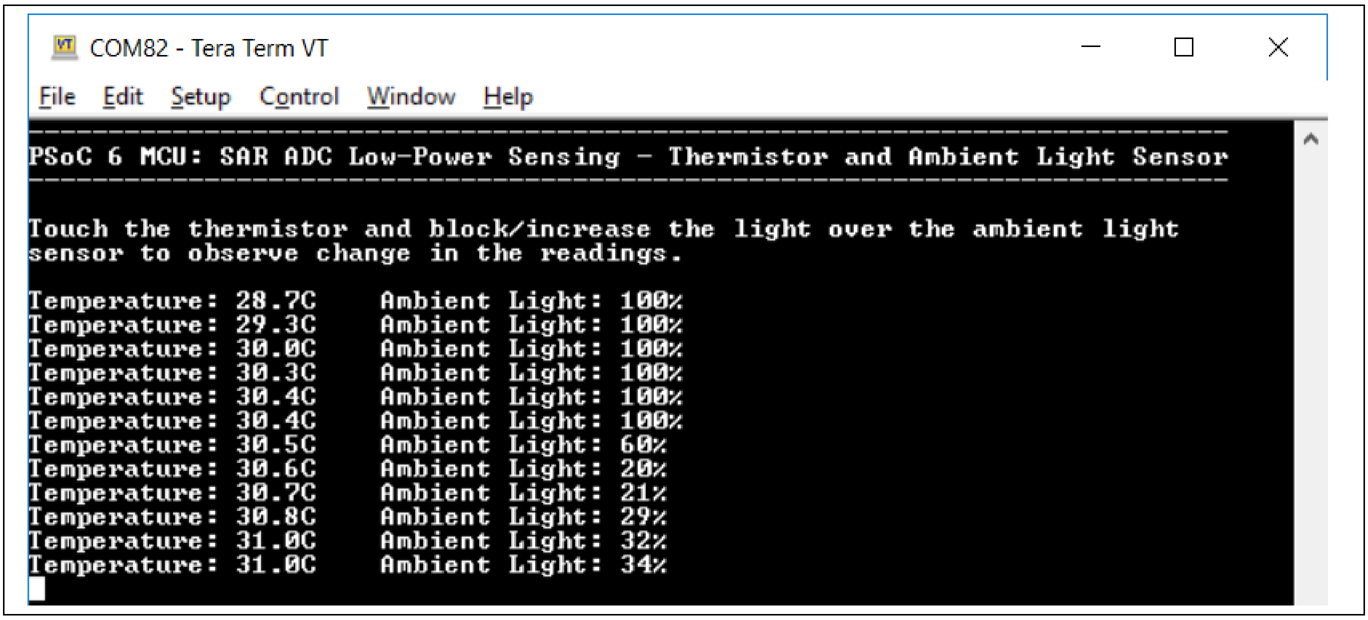


Figure 12 Tera Term window output of OOB code example

4. Touch the thermistor on the board and observe the change in the temperature readings on the terminal.
5. Block or increase the light over the ambient light sensor and observe the change in the percentage readings on the terminal.
6. Confirm that the user LED turns ON when the light over the ambient light sensor is blocked.

2.2.1.2 USB-UART bridge

The KitProg3 on the PSoC™ 62S4 pioneer board can act as a USB-UART bridge. The primary UART and flow-control lines between the PSoC™ 6 MCU and the KitProg3 are hard-wired on the board, as **Figure 13** shows.

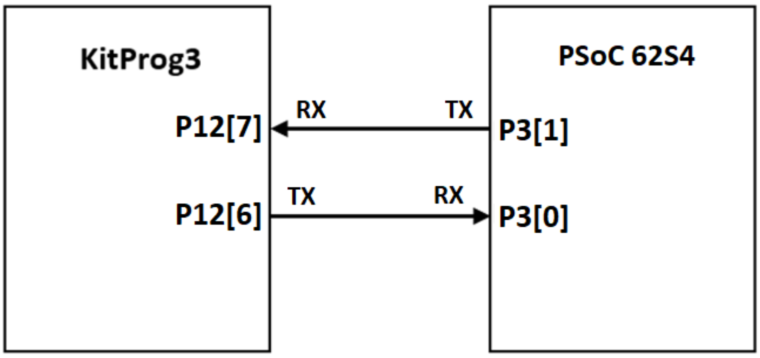


Figure 13 UART connection between KitProg3 and CY8C62x4 MCU

2.2.1.3 USB-I2C bridge

The KitProg3 can function as a USB-I2C bridge and can communicate with the Bridge Control Panel (BCP) software which acts as an I2C master. The I2C lines on the PSoC™ 6 MCU are hard-wired on the board to the I2C lines of the KitProg3, with onboard pull-up resistors as **Figure 14** shows. The USB-I2C supports I2C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg3 USB-I2C functionality, see the **KitProg3 user guide**.

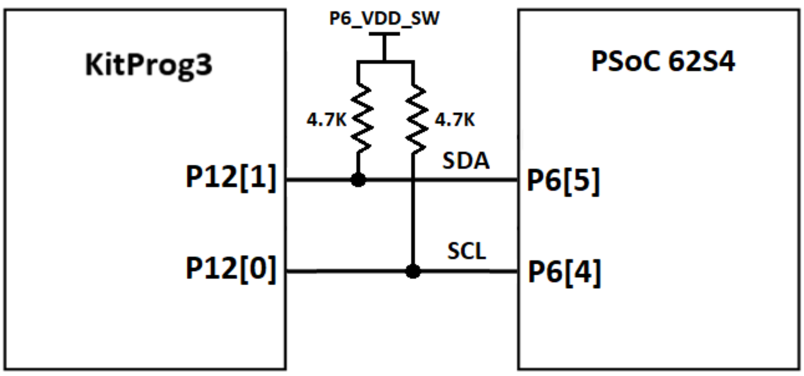


Figure 14 I2C connection between KitProg3 and the CY8C62x4 device

3 Hardware

3.1 Schematics

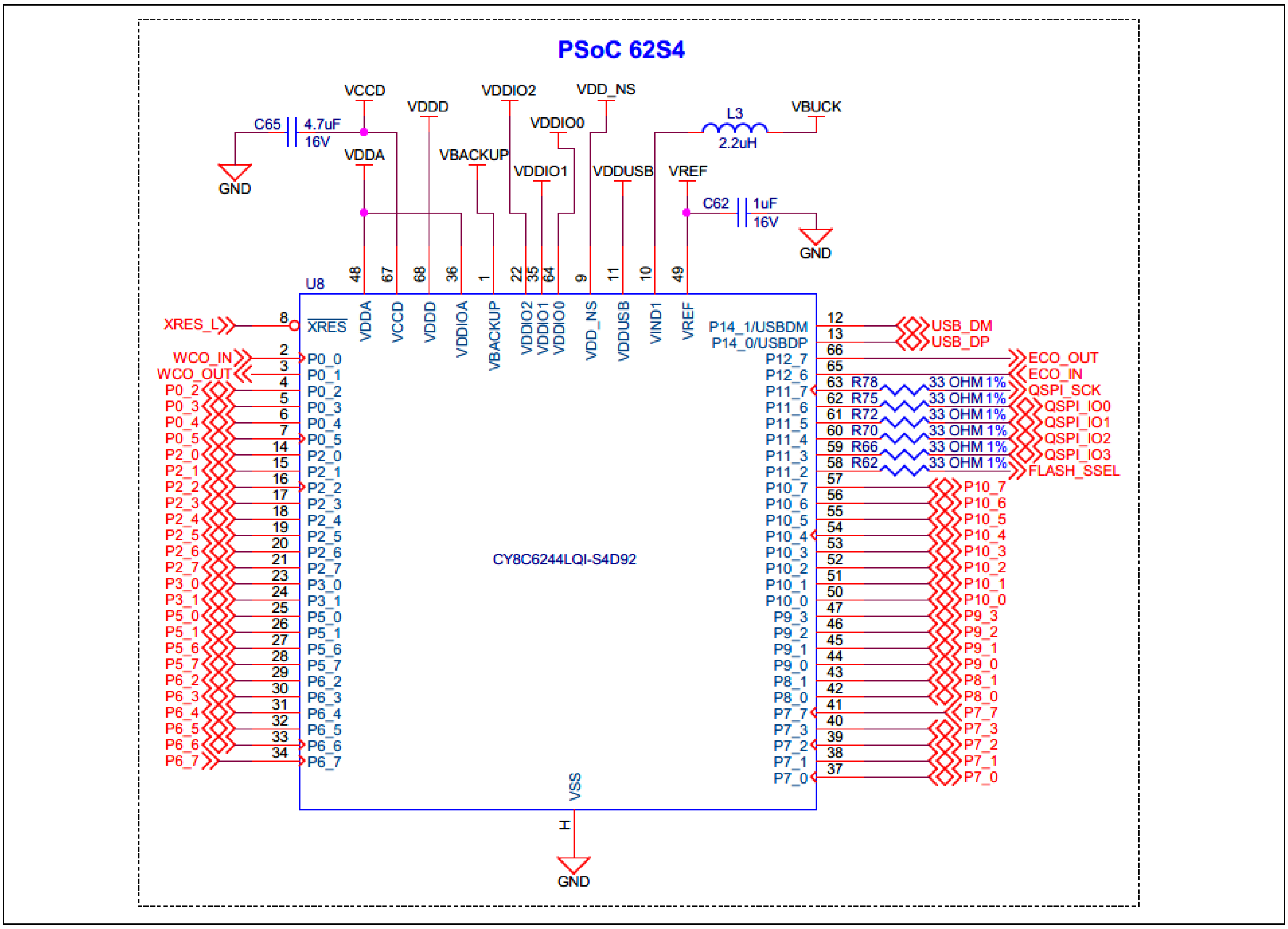
See the schematic files available on the **kit webpage**.

3.2 Hardware functional description

This section explains in detail the individual hardware blocks.

3.2.1 PSoC™ 6 CY8C62x4 device (CY8C6244LQI-S4D92)

PSoC™ 6 CY8C62x4 device is a high-performance, ultra-low-power and secure MCU platform, purpose-built for IoT applications. Based on the PSoC™ 6 MCU platform, this product line is a combination of a dual CPU microcontroller with low-power flash technology, digital programmable logic, high-performance analog peripherals, standard communication and timing peripherals. For more information, see the **PSoC™ 62S4 pioneer kit web page** and the CY8C62x4 MCU family datasheet.



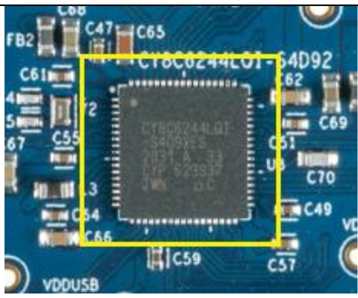


Figure 15 Schematic of CY8C62x4 device (Target)

3.2.2 PSoC™ 5LP-based KitProg3

An onboard PSoC™ 5LP (CY8C5868LTI-LP039) device is used as KitProg3 to program and debug the PSoC™ 6 MCU. The PSoC™ 5LP device connects to the USB port of a PC through a USB connector and to the SWD and other communication interfaces of the PSoC™ 6 MCU. The PSoC™ 5LP device is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. For more information, visit the **PSoC™ 5LP web page**. Also, see the **CY8C58LPxx Family datasheet**.

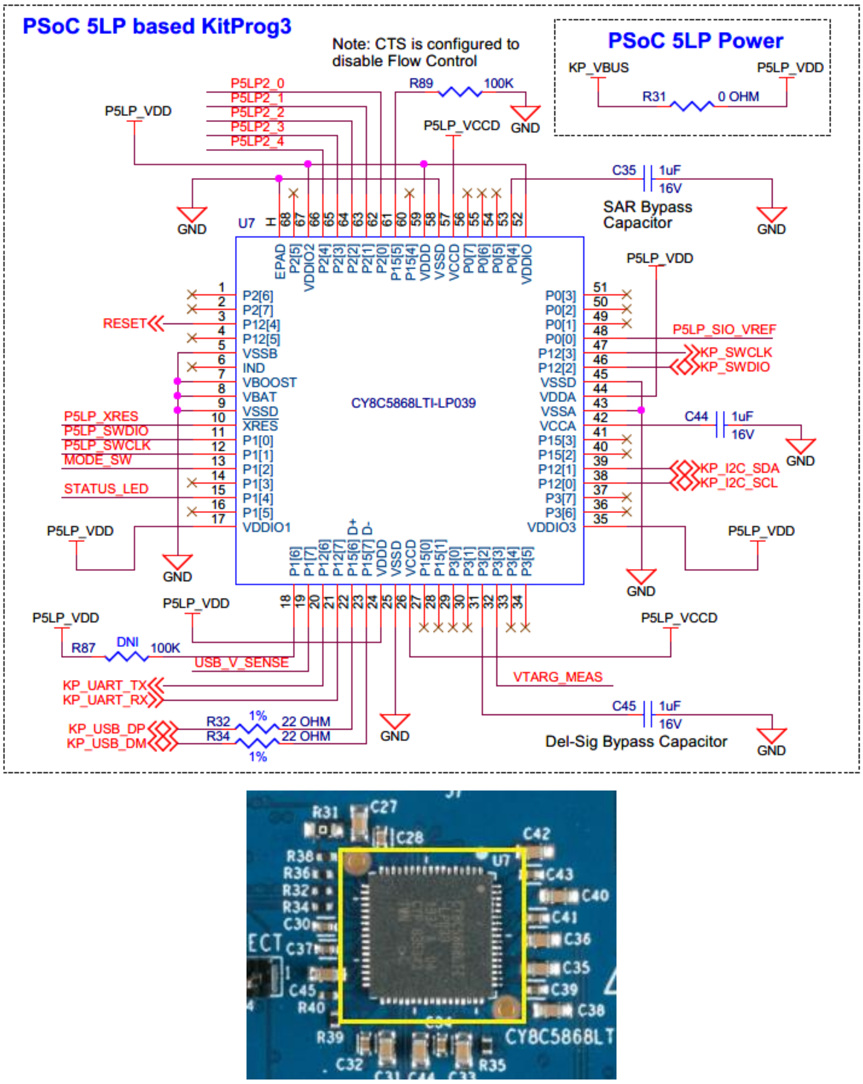


Figure 16 Schematic of PSoC™ 5LP (KitProg3)

3.2.3 Serial interconnection between PSoC™ 5LP and PSoC™ 6 MCU

In addition to the use as an onboard programmer, the PSoC™ 5LP device functions as an interface for the USB-UART and USB-I2C bridges, as shown in **Figure 17**. The USB-Serial pins of the PSoC™ 5LP device are hard-wired to the I2C/UART pins of the PSoC™ 6 MCU. The I2C pins are also available on the Arduino-compatible I/O header; therefore, the PSoC™ 5LP device can be used to control Arduino shields with an I2C interface.

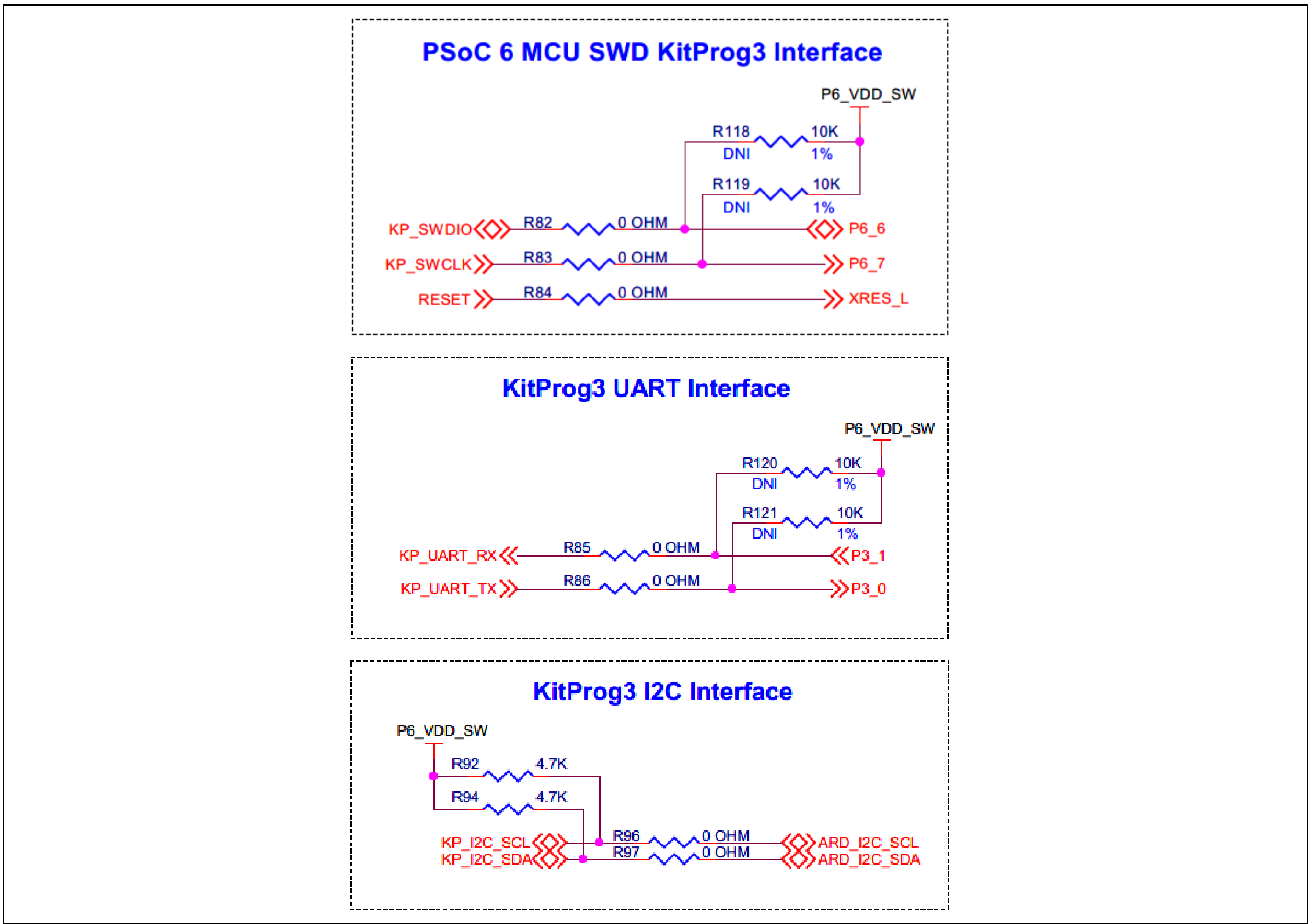


Figure 17 Schematic of programming and serial interface connections

3.2.4 Power supply system

The power supply system on this board is versatile, allowing the input supply to come from the following sources:

- 5 V from the onboard KitProg3 USB connector (**J10**)
- 5 V from the onboard PSoC™ 6 USB connector (**J5**)
- 7 V to 12 V power from Arduino header **J1.1** or from external power supply through VIN header **J9**

The power supply system is designed to support 1.8 V/2.5 V or 3.3 V operation of the PSoC™ 6 MCU. On the other hand, 5 V is provided for the KitProg3 operation.

The supply rails VIN (7 V to 12 V from header **J9** or **J1.1**), KP_VBUS (5 V KitProg3 USB connector **J10**) and P6_VBUS (5 V PSoC™ 6 USB connector **J5**) are combined into VCC_IN through ORing diodes as **Figure 18** shows. The VIN header (**J9**) is not populated on the board by default.

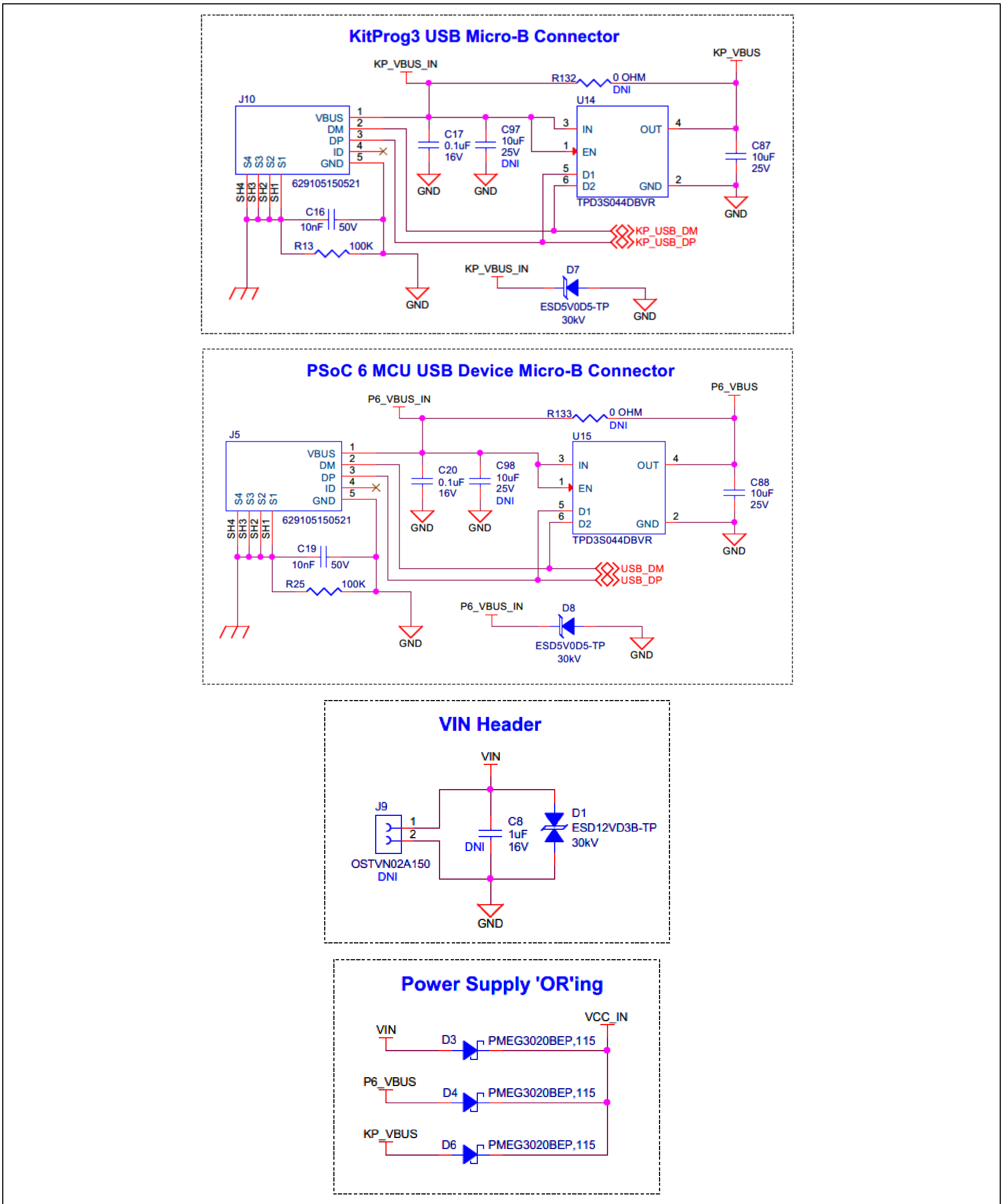


Figure 18 Power supply sources

It is important to understand that 5 V will not be generated to the Arduino header and KitProg3 section when the kit is powered from the VIN header (**J9**) or PSoC™ 6 USB connector (**J5**) as there is no separate 5 V regulator. 5 V will be present only when the kit is powered through the KitProg3 USB connector (**J10**).

3.2.4.1 Voltage regulators & reverse voltage protection

Two onboard buck regulators (**U1**, **U2**) are used to generate 3.3 V and 1.8 V output from VCC_IN for the PSoC™ 6 MCU. The regulator outputs (VOUT_3V3 and VOUT_1V8) are fed to the reverse voltage protection circuits before supplying to the PSoC™ 6 MCU and other peripherals as shown in the **Figure 19**. The 1.8 V regulator output (VOUT_1V8) can also be configured to supply 2.5 V to the PSoC™ 6 MCU for e-Fuse programming by shorting the jumper **J14** or populating **R8**. Note that **J14** is kept open by default.

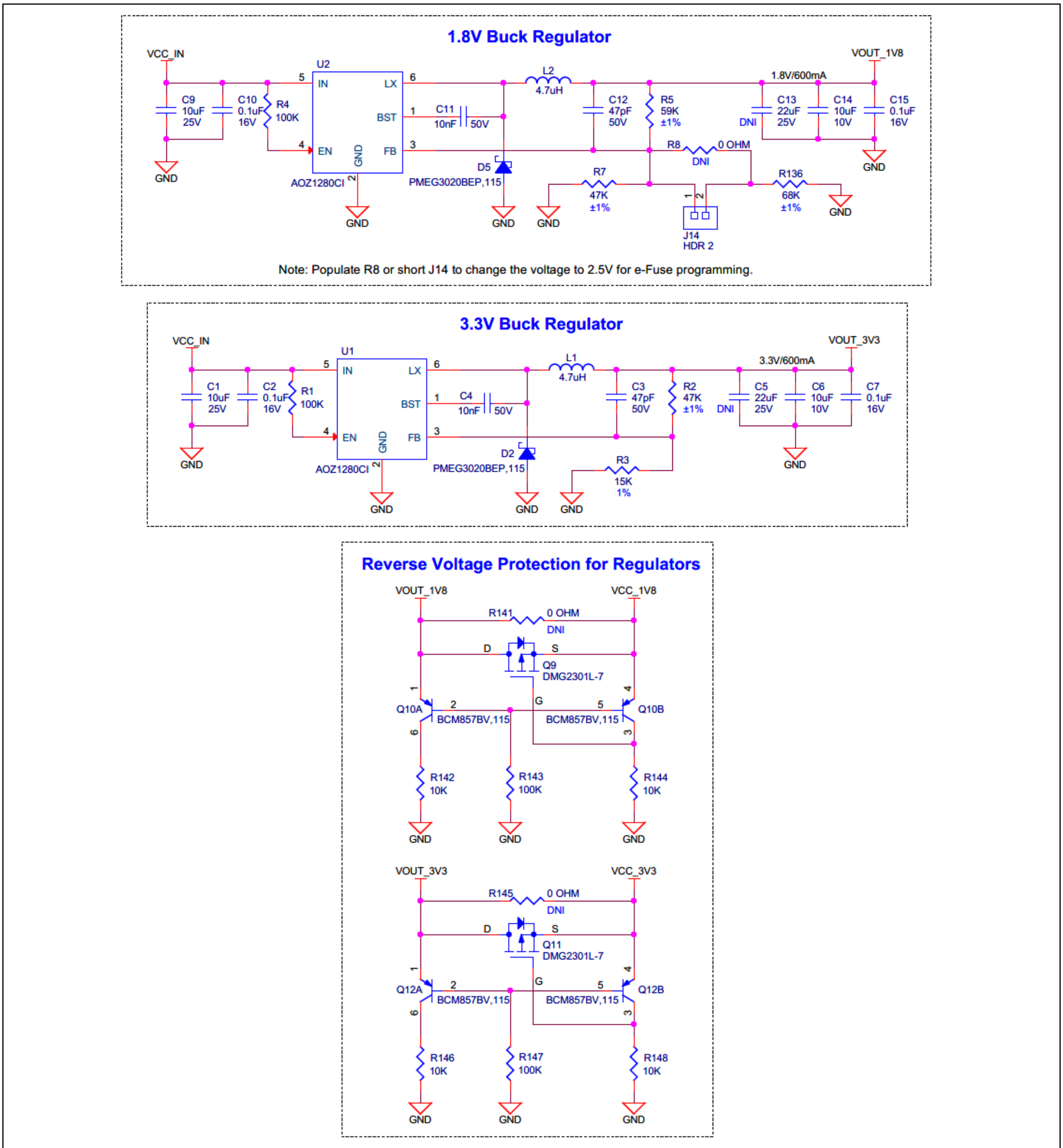


Figure 19 Voltage regulators & reverse voltage protection

3.2.4.2 Power switch

A power switch (**U4**) is provided to supply power for the onboard peripherals such as thermistor, ambient light sensor, QSPI flash I/O domain and user LEDs. The circuit is used to reduce the current leakage on the P6_VDD power domain.

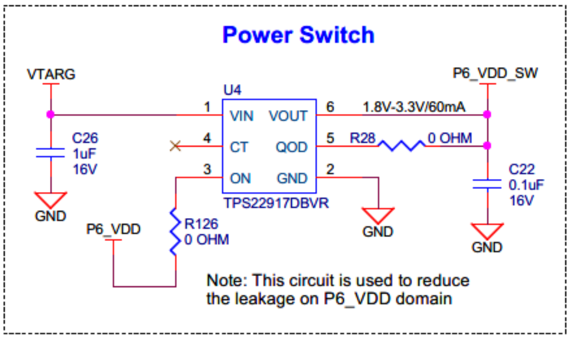


Figure 20 **Power switch circuit**

3.2.4.3 PSoC™ 6 MCU power selection & current measurement headers

VTARG has a dedicated 3-pin voltage selection header **J12** that selects between 1.8 V/2.5 V and 3.3 V operating voltages for the PSoC™ 6 MCU. The default jumper position for **J12** is 2–3 which selects 3.3 V.

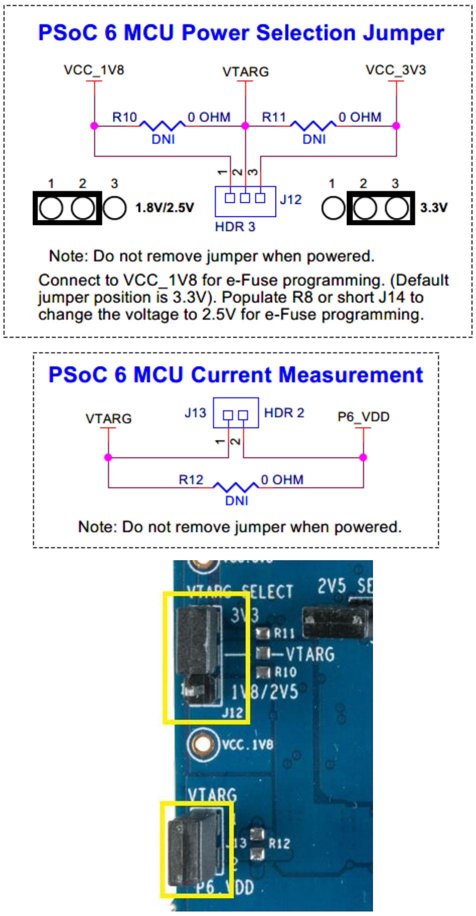


Figure 21 **Voltage selection & current measurement headers**

The P6_VDD domain has a dedicated 2-pin header **J13** to facilitate easy current measurement across the pins VTARG and P6_VDD using an ammeter.

3.2.5 I/O headers

3.2.5.1 Arduino-compatible headers (J1, J2, J3, J4)

The board has four Arduino-compatible headers: **J1**, **J2**, **J3** and **J4**. You can connect 3.3 V Arduino-compatible shields to develop applications based on the shield's hardware. See **Board Details** for details on PSoC™ 6 MCU pin mapping to these headers.

3.2.5.2 Extended header (J11)

There is an extended header **J11** which provides connectivity to PSoC™ 6 MCU GPIOs that are not connected to the Arduino-compatible headers. The majority of these pins are multiplexed with CapSense™ pins and are not connected to the PSoC™ 6 MCU by default. For detailed information on how to rework the kit to access these pins, see **PSoC™ 62S4 pioneer board rework**. Note that **J11** is not loaded on board by default.

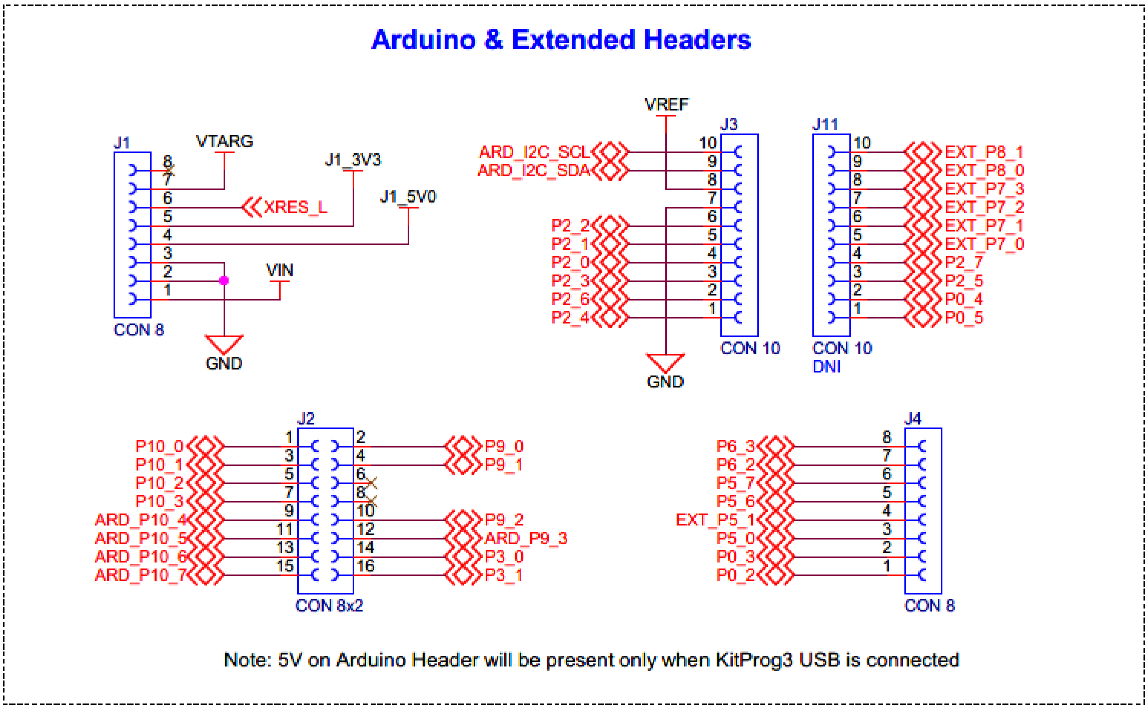


Figure 22 Arduino-compatible headers

3.2.6 CapSense™ circuit

One CapSense™ 5-segment slider (**CSS1**) and two CapSense™ buttons (**CSB1**, **CSB2**) are connected to the PSoC™ 6 MCU as **Figure 23** shows. The CapSense™ slider and buttons support only self-cap (CSD) sensing for this kit. An external capacitor – CMOD (**C70**) for CSD is present on the PSoC™ 62S4 Pioneer board. Note that the CapSense™ shield is connected to ground. Moreover, the CapSense™ GPIOs can also be used as external GPIOs which are routed to extended header **J11**. For more detailed information on how to rework the kit to access these pins, see **PSoC™ 62S4 pioneer board rework**. For details on using CapSense™ including design guidelines, see the ***Getting started with CapSense™***.

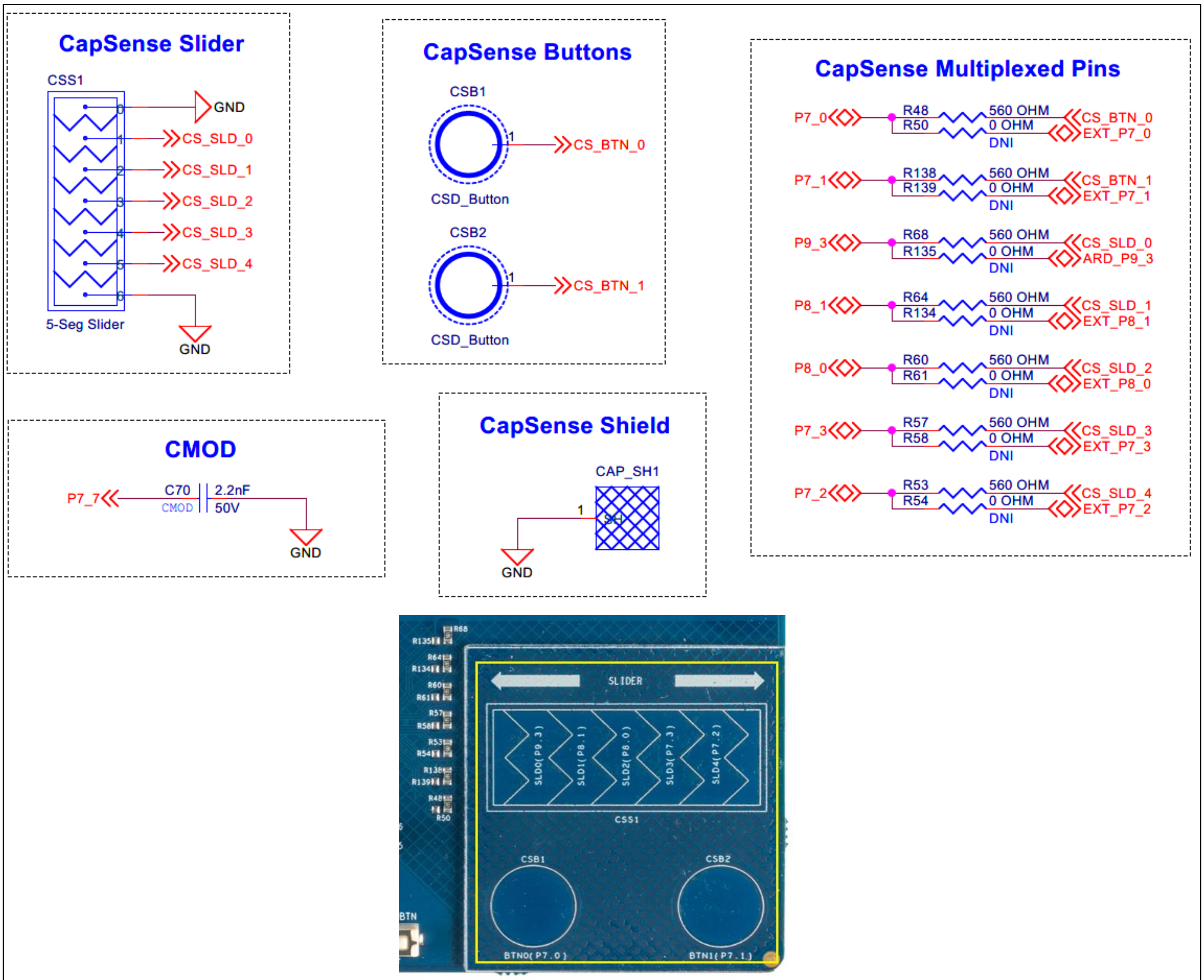


Figure 23 CapSense™ circuit

Table 4 CapSense™ pin assignment

CapSense™ element	PSoC™ 6 MCU GPIO	CapSense™ element	PSoC™ 6 MCU GPIO
CS_BTN_0	P7[0]	CS_SLD_2	P8[0]
CS_BTN_1	P7[1]	CS_SLD_3	P7[3]
CS_SLD_0	P9[3]	CS_SLD_4	P7[2]
CS_SLD_1	P8[1]	CMOD	P7[7]

3.2.7 LEDs

The PSoC™ 62S4 Pioneer kit contains four LEDs:

- The yellow power LED (**D14**): Indicates that the board is powered from the onboard KitProg3 USB Micro-B connector, PSoC™ 6 USB Micro-B connector or VIN header.
- The yellow Status LED (**D10**): Indicates the KitProg3 status (the LED is connected to **P1[4]** of the KitProg3 PSoC™ 5LP device). See **Table 5** for a summary of the status LED states.

Table 5 KitProg3 mode switching

KitProg3 programming modes *	Status LED (D10)
CMSIS-DAP/Bulk mode (default)	ON
CMSIS-DAP/HID mode	RAMPING at 1Hz

*Note: * Switching between the programming modes can be done through **Firmware-loader** tool. See Section 2.2.3, "Mode Switching" of the **KitProg3 user guide** for details.*

- Two User LEDs (**D12**, **D13**): One red LED (**D12**) and an orange LED (**D13**) are connected to **P2[5]** and **P2[7]** GPIOs of the target PSoC™ 6 MCU device respectively. The LEDs are active low so the pins must be driven LOW to turn ON the LEDs.

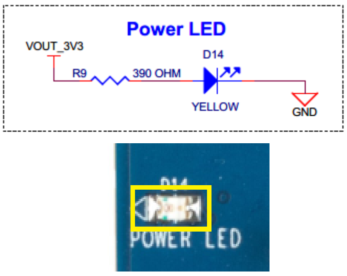


Figure 24 Power LED

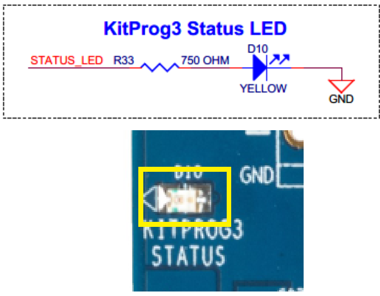


Figure 25 KitProg3 status LED

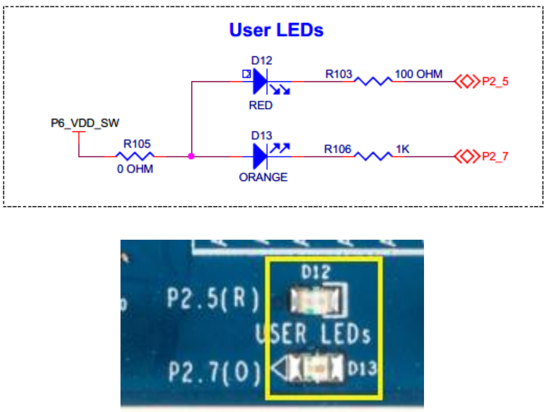


Figure 26 User LEDs

3.2.8 Push buttons

3.2.8.1 User button/hibernate wakeup

The target PSoC™ 62S4 Pioneer board contains one user button (**SW2**) connected to the **P0[4]** pin on the PSoC™ 6 MCU device. This button can be used for general user inputs or to control different states in an application such as hibernate wakeup.

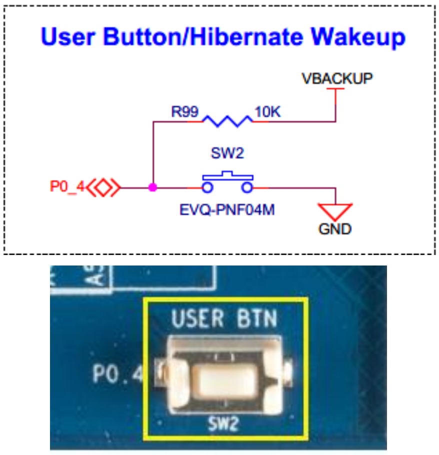


Figure 27 User button

3.2.8.2 Reset switch

When this **SW1** button is pressed, the XRES_L line of the PSoC™ 6 MCU is pulled to ground, which, in turn, resets the target device.

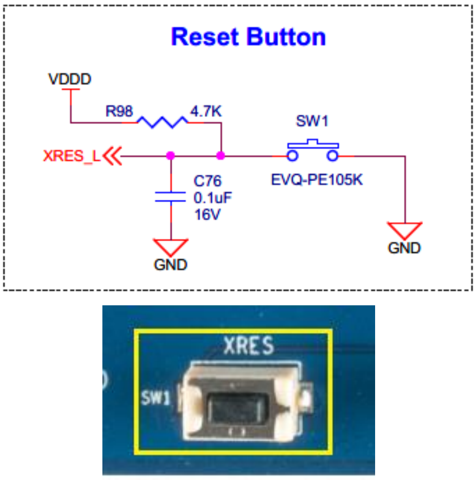


Figure 28 Reset switch

3.2.8.3 Mode switch

The PSoC™ 62S4 pioneer board contains a push button (**SW3**) connected to **P1[2]** of PSoC™ 5LP. This button is used to change the kit programming mode. See the **KitProg3 user guide** for more details.

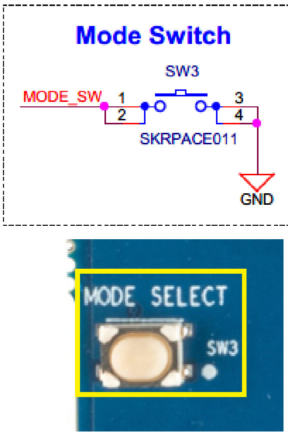


Figure 29 Mode switch

3.2.9 Quad SPI NOR flash (S25FL512SAGMFIR10)

The PSoC™ 62S4 pioneer board has a 512Mb Quad-SPI NOR flash memory (**U9**). The NOR flash is connected to the Quad SPI interface of the PSoC™ 6 MCU device. The NOR flash device supports four-bit (Quad I/O) serial commands.

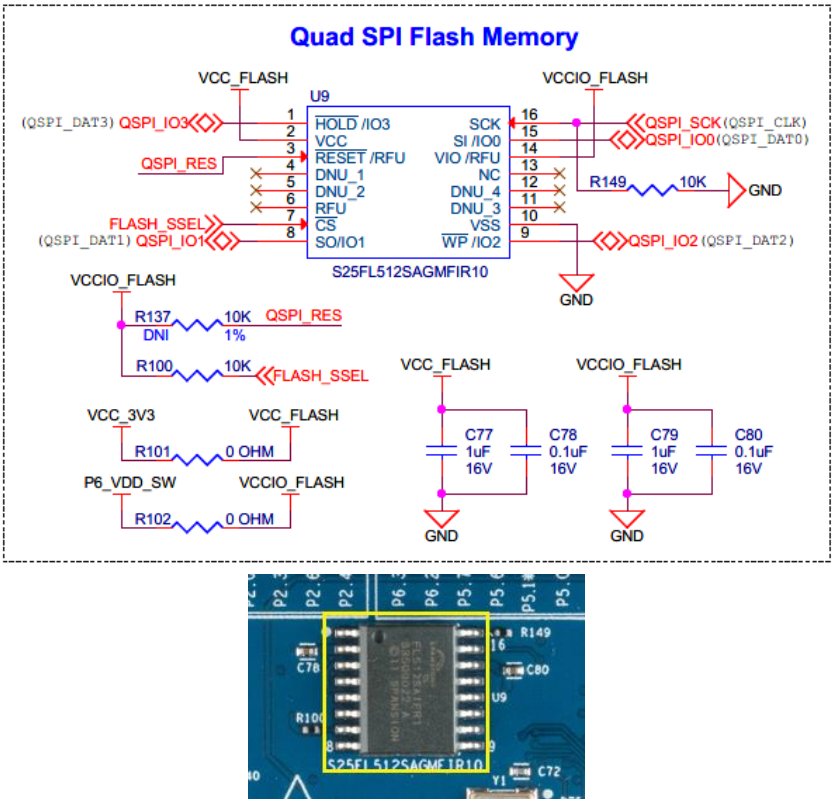


Figure 30 QSPI NOR flash memory

3.2.10 Thermistor

The kit contains one NTC thermistor for temperature sensing. It is capable of sensing in both single-ended and differential modes. The differential mode is supported by default. **P10[4]**, **P10[5]** and **P10[6]** are the GPIOs which are used for the temperature sensing in differential mode.

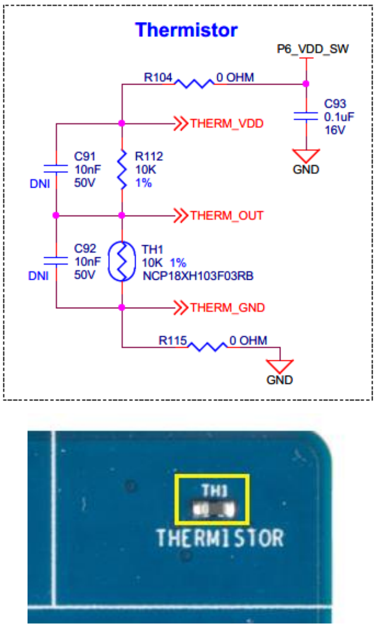


Figure 31 Thermistor

3.2.11 Ambient light sensor

The kit contains one ambient light senor (ALS) which uses GPIO **P10[7]** of the PSoC™ 6 MCU for ambient light sensing.

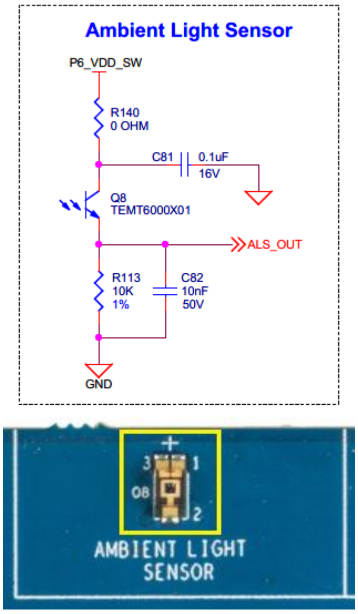


Figure 32 Ambient light sensor

3.2.12 Crystal oscillators

The PSoC™ 62S4 pioneer board includes a 24-MHz ECO (**Y1**) and a 32.768-kHz WCO (**Y2**) for the PSoC™ 6 MCU device.

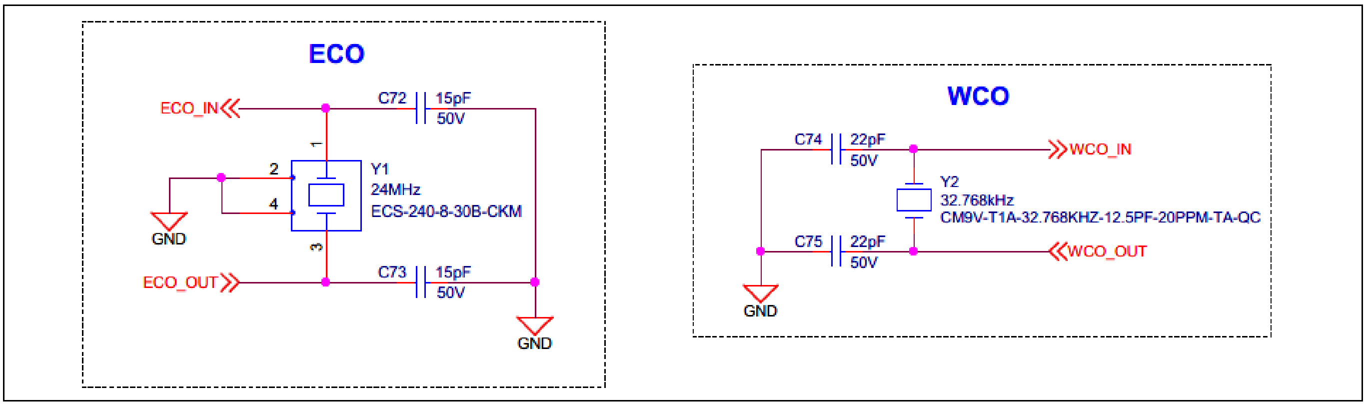


Figure 33 **ECO and WCO**

3.2.13 PSoC™ 6 MCU 10-pin SWD/JTAG header

PSoC™ 6 MCU can be programmed alternatively through a 10-pin SWD/JTAG header (**J8**) using a MiniProg4 programmer or any third-party programmer. Note that the JTAG functionality is not supported by default. For details on how to enable the JTAG functionality, see the **PSoC™ 62S4 pioneer board rework**.

Also, there is a protection circuit provided to save the PSoC™ 6 MCU from overvoltage (>3.45 V) through the **J8** header.

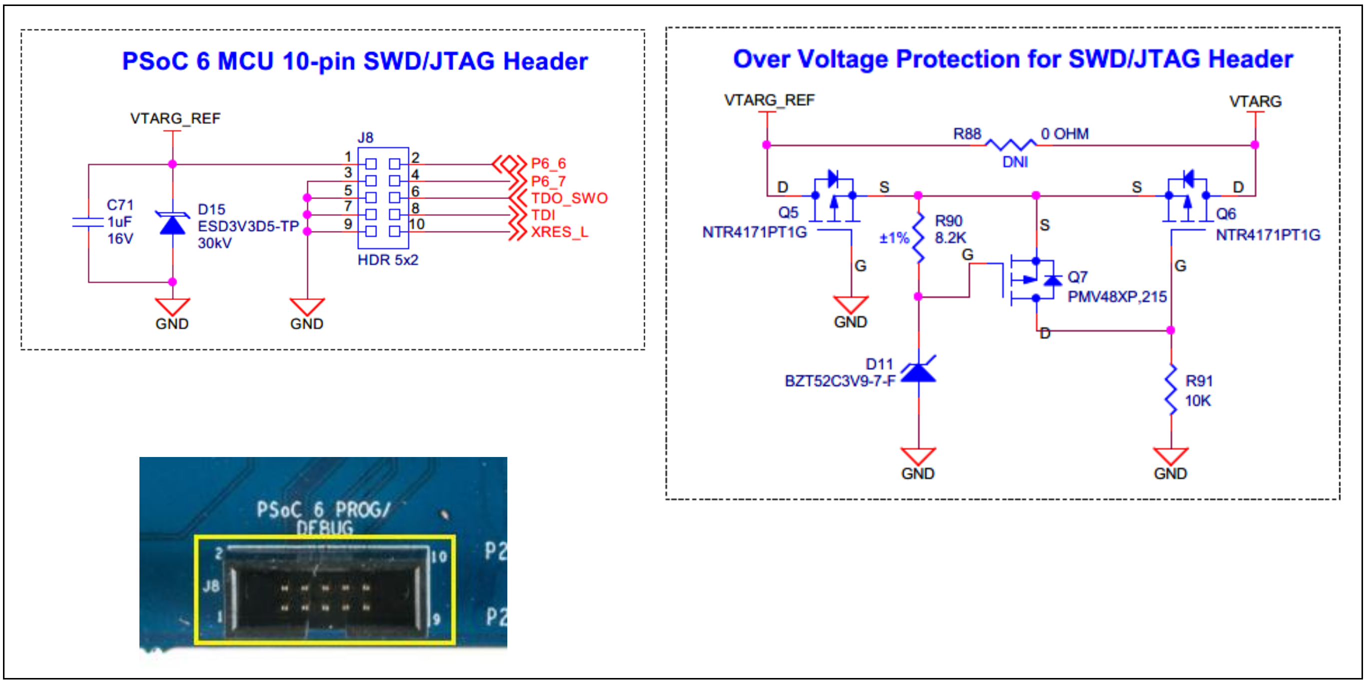


Figure 34 **PSoC™ 6 MCU 10-pin SWD/JTAG header with overvoltage protection**

3.2.14 PSoC™ 6 MCU USB device detect

The board contains a micro-B USB connector (**J5**) for the PSoC™ 6 MCU. Note that the host functionality is not supported; it only acts as a device. Whenever a device is connected at **J5**, the device detect circuit pulls the **USB_VBUS_DET** pin HIGH, which is then read by the PSoC™ 6 MCU.

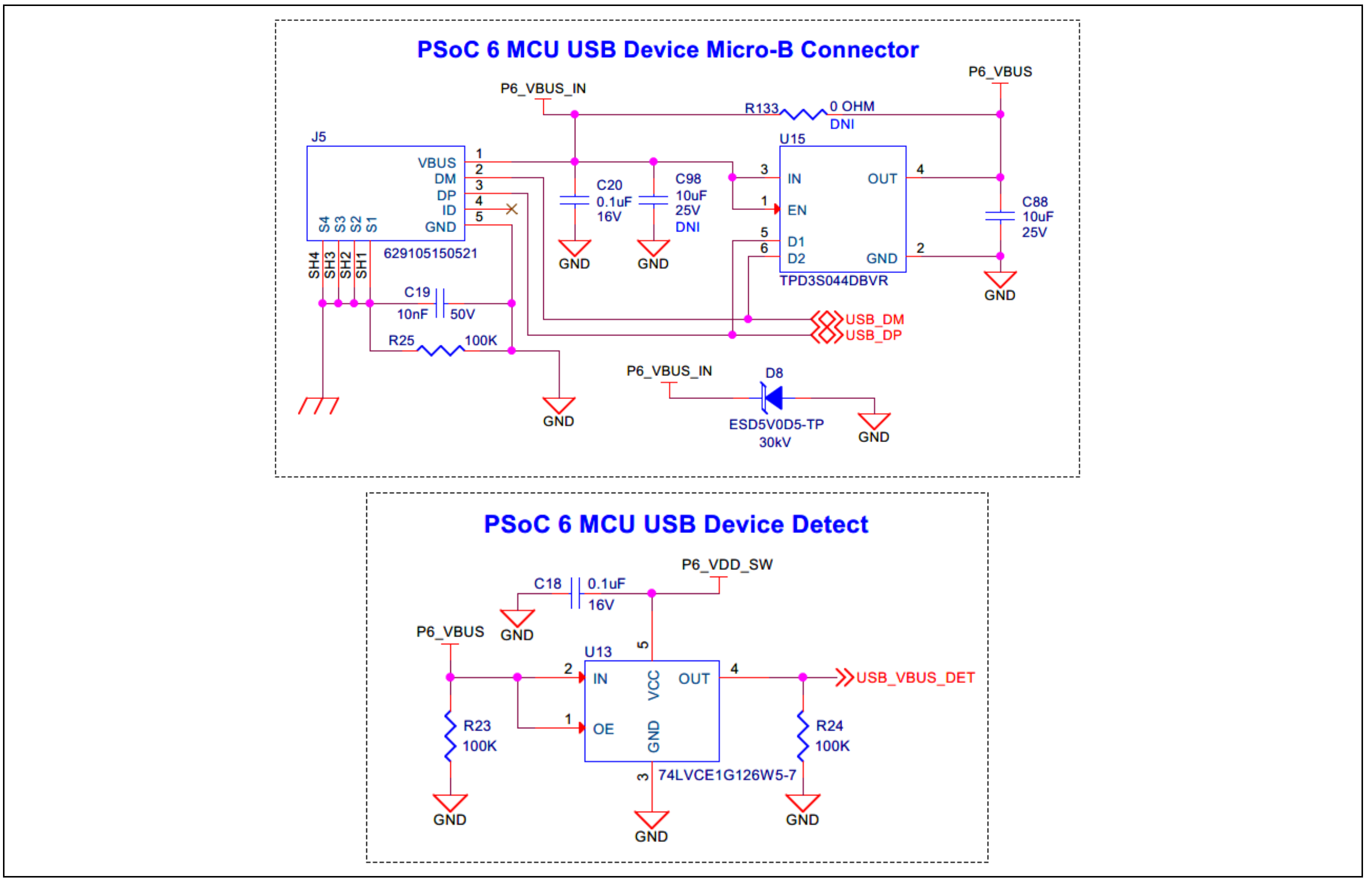


Figure 35 **PSoC™ 6 MCU USB device detect**

3.3 PSoC™ 62S4 pioneer board rework

3.3.1 Rework on CapSense™ and additional multiplexed pins

CapSense™ I/Os and the extended header pins are multiplexed using the resistor Y-networks as shown in **Figure 36**. These give the provision to use the CapSense™ I/Os as extended GPIOs by connecting them to the extended header **J11**. The resistor pairs used for CapSense™ multiplexing are: **R48**, **R50**; **R138**, **R139**; **R68**, **R135**; **R64**, **R134**; **R60**, **R61**; **R57**, **R58**; **R53**, **R54**. To use the alternative GPIOs, you need to remove the mounted resistor and populate the other resistor of the mentioned pairs. For example, remove **R48** and populate **R50** to use the **P7[0]** as extended GPIO at the header **J11.5**.

Similarly, the additional PSoC™ 6 MCU GPIOs user for thermistor, ALS and PSoC™ 6 USB detect are also multiplexed using the resistor Y-networks: **R108**, **R128**; **R110**, **R127**; **R114**, **R130**; **R111**, **R131**; **R124**, **R125.** These pins can also be used as extended GPIOs at their respective headers **J2** and **J4** by changing the resistor positions.

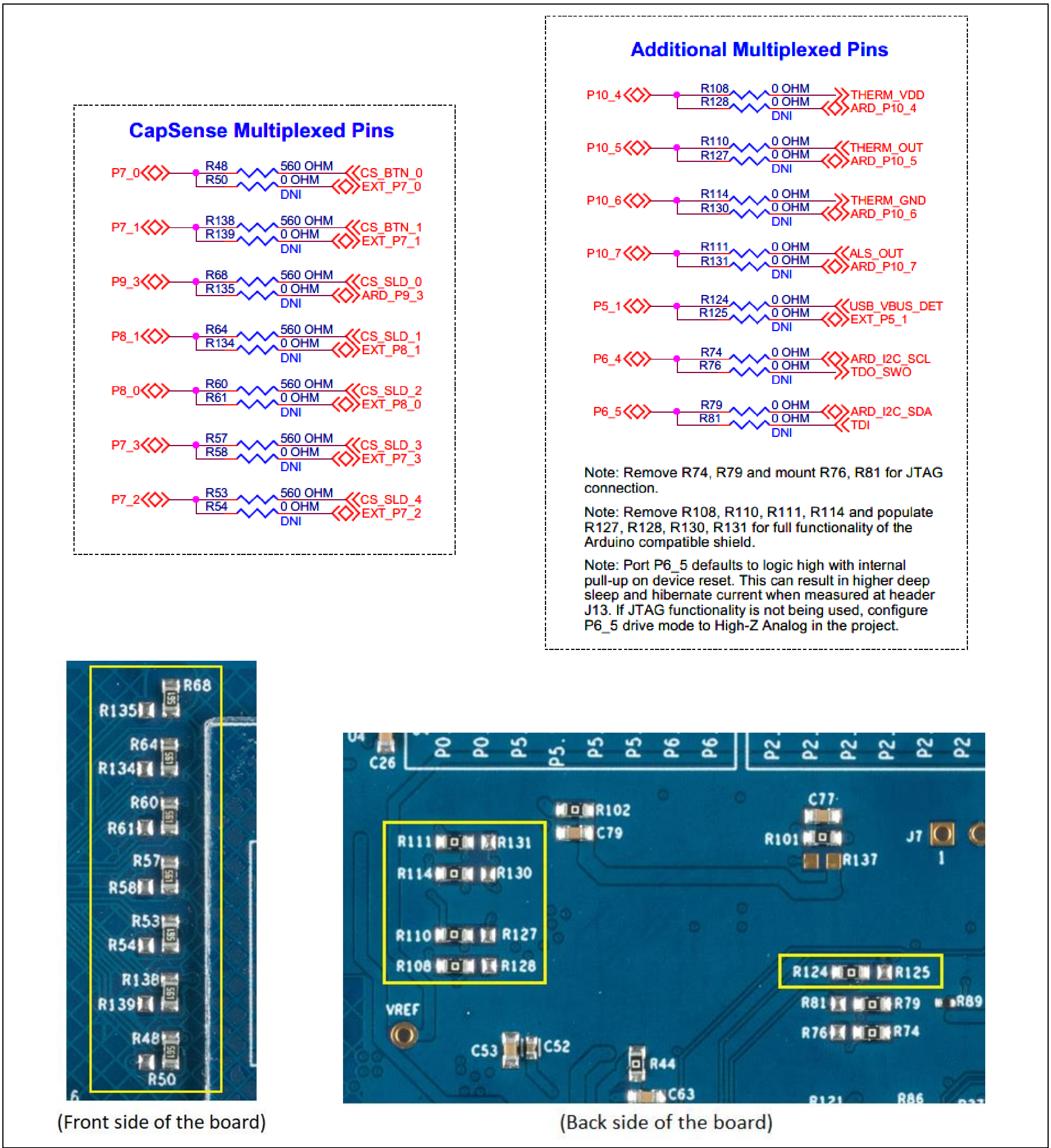
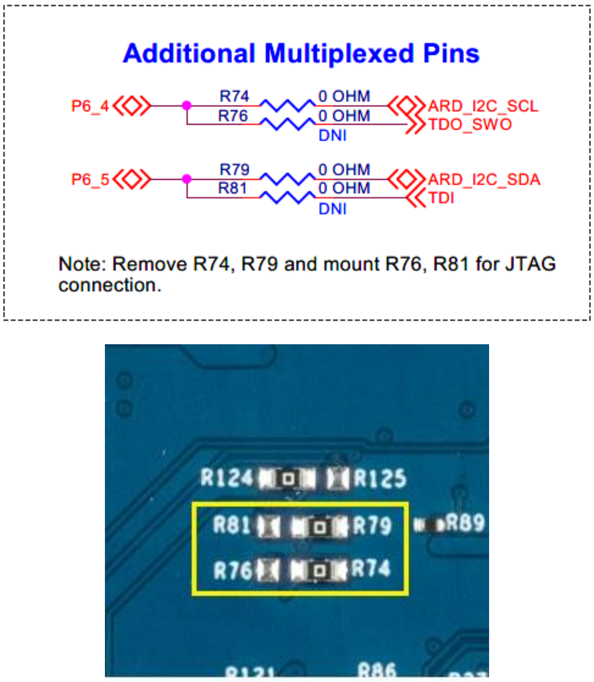


Figure 36 CapSense™ and additional multiplexed pins

3.3.2 Rework for JTAG connection

The JTAG pins are multiplexed with the I2C communication pins of the PSoC™ 6 MCU using the resistor Y-networks: **R74**, **R76** and **R79**, **R81** as shown in the **Figure 37**. In order to enable JTAG support for the **J8** header you need to remove resistors **R74**, **R79** and mount resistors **R76**, **R81**.



(Back side of the board)

Figure 37 JTAG pins

3.3.3 Bypassing reverse voltage protection circuits for voltage regulators and Arduino header

There are two pairs of reverse voltage protection circuits to protect the 1.8 V and 3.3 V Buck regulators and PSoC™ 5LP in case of any overvoltage occurring from VTARG, J1_3V3 or J1_5V0 rails. These protection circuits cut off any voltage greater than the rated max voltages by ~35mV.

Note that the protection circuits have a voltage drop of maximum ~140 mV. To avoid the voltage drop, bypass the protection circuits by populating the bypass zero-ohm resistors **R141**, **R145**, **R14** and **R15** respectively.

Note that this change will compromise the protection circuits when an overvoltage occurs at the respective supply connectors and may permanently damage the devices if the voltage exceeds the absolute maximum limit of any of the devices.

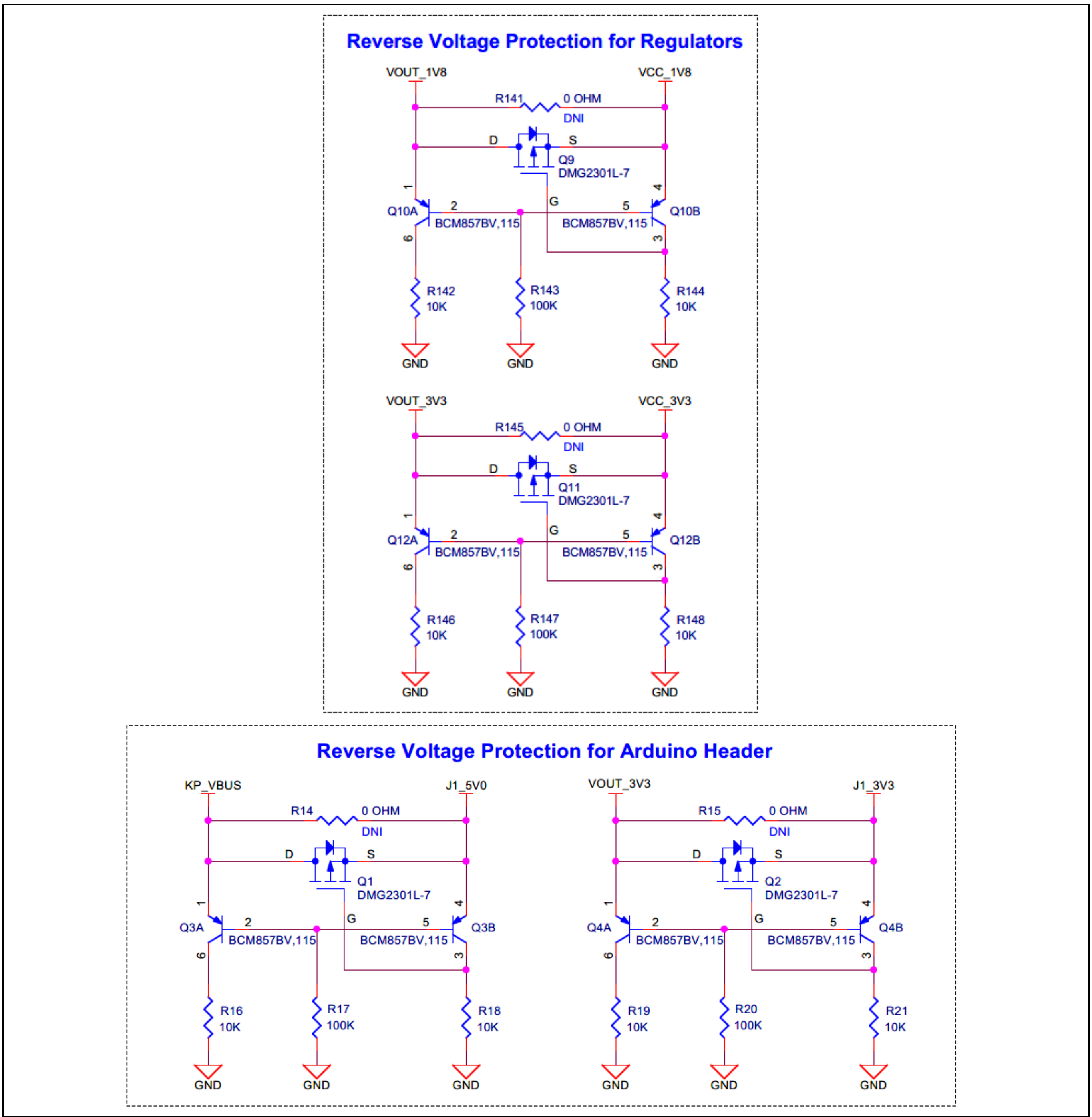


Figure 38 **Reverse voltage protection circuits for voltage regulators and Arduino header**

3.4 Bill of materials

Refer to the BOM file available on the **kit webpage**.

Revision history

Major changes since the last revision

Version	Date	Description
**	2021-03-12	Initial release.
*A	2021-05-19	Updated Introduction: Updated description. Updated Getting started: Updated description. Updated Kit operation: Updated Theory of operation: Updated description. Updated KitProg3: on-board programmer/debugger: Updated Programming and debugging using ModusToolbox™: Updated description. Updated USB-I2C bridge: Updated description. Updated Hardware: Updated Hardware functional description: Updated PSoC™ 6 CY8C62x4 device (CY8C6244LQI-S4D92): Updated description. Updated Quad SPI NOR flash (S25FL512SAGMFIR10): Updated description. Updated to new template.

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

Edition <2021-05>
Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Go to www.cypress.com/support

Document reference
002-32537 Rev. *A