# CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit guide

## About this document

### Scope and purpose

This guide will help you get acquainted with the PSoC™ 4100S Max pioneer kit (CY8CKIT-041S-MAX). This document also provides hardware details of the board and its usage information.

### Intended audience

This kit is intended for all technical specialists who are familiar with PSoC™ 4 and CAPSENSE™. This kit is intended to be used under laboratory conditions.

# Table of contents

**About this document** ..... **1**
**Table of contents** ..... **2**
**Safety and regulatory compliance information** ..... **4**
General safety instructions ..... 4
ESD protection ..... 4
Handling boards ..... 4
**1 Introduction** ..... **5**
1.1 Kit contents ..... 5
1.2 Getting started ..... 6
1.3 Board details ..... 7
1.4 Additional learning resources ..... 11
1.5 Technical support ..... 11
1.6 Documentation conventions ..... 12
1.7 Abbreviations and definitions ..... 12
**2 Kit operation** ..... **14**
2.1 Theory of operation ..... 14
2.2 KitProg3: Onboard programmer/debugger ..... 18
2.3 Creating a project and program/debug using ModusToolbox™ ..... 18
2.3.1 Using the OOB example – [MSC CSX button project] ..... 23
2.3.2 USB-UART bridge ..... 24
2.3.3 USB-I2C bridge ..... 24
**3 Hardware** ..... **25**
3.1 Schematics ..... 25
3.2 Hardware functional description ..... 25
3.2.1 PSoC™ 4100S Max MCU ..... 25
3.2.2 PSoC™ 5LP-based KitProg3 program and debug ..... 28
3.2.2.1 Serial interface between KitProg3 and PSoC™ 4100S Max MCU ..... 29
3.2.3 PSoC™ 4100S Max MCU SWD 10-pin programming header ..... 30
3.2.4 Power supply system ..... 31
3.2.4.1 Voltage regulators ..... 32
3.2.4.2 Voltage selection ..... 32
3.2.4.3 Current measurement header ..... 33
3.2.5 CAPSENSE™ ..... 33
3.2.6 LEDs ..... 35
3.2.7 Push buttons ..... 36
3.2.8 Crystal oscillators ..... 36
3.2.9 Thermistor ..... 37
3.2.10 Expansion headers ..... 38
3.2.10.1 Arduino Uno compatible connectors ..... 38
3.2.10.2 Additional GPIO expansion connector ..... 40
3.2.10.3 FFC connector for expansion board ..... 41
3.2.11 CY8CKIT-041S-MAX capacitive sensing expansion board ..... 41
3.2.11.1 Touchpad and proximity interface ..... 41
3.2.11.2 Slider interface ..... 44
3.2.11.3 Hatch pattern configuration ..... 45

**Table of contents**

3.3 CY8CKIT-041S-MAX Kit rework ..... 46
3.4 Bill of materials ..... 53
3.5 Frequently asked questions ..... 53
**Revision history ..... 55**

# Safety and regulatory compliance information

The CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit is intended for development purposes only. This kit is not a finished product and when assembled may not be resold or otherwise marketed unless all required authorizations are obtained. Contact **support** for details.

| | |
|---|---|
| | PSoC™ 4100S Max pioneer 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 PSoC™ 4100S Max pioneer boards in the protective shipping package. |
| | End-of-Life/Product Recycling<br><br>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. It is recommended 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

CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer 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

The PSoC™ 4100S Max pioneer kit enables you to evaluate and develop your applications using the PSoC™ 4100S Max series MCU (hereafter called “PSoC™ 4100S Max MCU”).

The PSoC™ 4100S Max MCU is a true programmable embedded system-on chip, integrating custom digital and analog peripheral functions, memory, and an Arm® Cortex®-M0+ MCU on a single chip with flexible automatic routing, and the 5th generation multi sense converter (MSC) - a capacitive sensing system with best-in-class performance.

The CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit lets you evaluate all the PSoC™ 4100S Max MCU features. It comes with two boards, CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer board (hereafter called “PSoC™ 4100S Max pioneer board”) and CY8CKIT-041S-MAX capacitive sensing expansion board (hereafter called “Capacitive Sensing Expansion Board”). The PSoC™ 4100S Max pioneer board consists of a PSoC™ 4100S Max MCU, an onboard programmer/debugger (KitProg3), voltage regulators, user LEDs and buttons, CAPSENSE™ buttons, thermistor, a micro-B connector for USB device interface, crystal oscillators and other passives. The capacitive sensing expansion board consists of CAPSENSE™ Touchpad, slider and proximity sensors.

The PSoC™ 4100S Max pioneer kit introduces an evaluation platform for the 5th generation CAPSENSE™ - multi sense converter in PSoC™ 4100S Max devices. Refer the **AN85951 - PSoC™ 4 and PSoC™ 6 MCU CAPSENSE™ Design Guide** to know the detailed features of 5th generation CAPSENSE™ - multi sense converter.

You can use ModusToolbox™ to develop and debug your PSoC™ 4 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™ 4 and ModusToolbox™ IDE, refer to the application note **AN79953 - Getting Started with PSoC™ 4** to help you familiarize with the PSoC™ 4 and help you create your own design using the ModusToolbox™ IDE.

## 1.1 Kit contents

The CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit has the following contents, as shown in **Figure 1**.

- PSoC™ 4100S Max pioneer board
- Capacitive sensing expansion board
- USB Type-A to Micro-B cable
- Six jumper wires (5 inches each)
- 40-pin FFC (2 inches)
- Quick start guide

**Figure 1 CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit contents**

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

## 1.2 Getting started

This guide will help you get acquainted with the PSoC™ 4100S Max pioneer kit:

- Follow **Using the OOB example – [MSC CSX button project]** section to have a quick review of OOB project in this kit.
- The **Kit Operation** chapter describes the major features of the PSoC™ 4100S Max pioneer 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™ 4100S Max pioneer kit is supported in ModusToolbox™. For the latest software support for this development kit, refer to the **kit webpage**.
  - ModusToolbox™ software is a free development ecosystem that includes the ModusToolbox IDE. 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™ 4100S Max pioneer board. These examples help you familiarize yourself with the PSoC™ 4100S Max MCU and create your own design. These examples are available in the ModusToolbox™ IDE. You can also find code examples on the GitHub page dedicated to **ModusToolbox™ based examples**.
  - To access code examples through ModusToolbox™, see "Code Example" section in **AN79953 - Getting Started with PSoC® 4** under "ModusToolbox™ Resources".

## 1.3 Board details

The CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit comes with two boards, a PSoC™ 4100S Max pioneer board and a capacitive sensing expansion board.

The pioneer board has the following features (see **Figure 5**):

1. Power LED (**D2**)
2. KitProg3 USB connector (**J8**)
3. KitProg3 status LED (**D3**)
4. PSoC™ 5LP based KitProg3 programmer and debugger (CY8C5868LTI-LP039, **U2**)
5. KitProg3 programming mode selection button (**SW3**)
6. Voltage regulator (U3)
7. PSoC™ 4100S Max MCU power selection jumper (**J10**)
8. PSoC™ 4100S Max MCU current measurement header (**J11**)
9. PSoC™ 4100S Max MCU reset button (**SW1**)
10. Arduino UNO R3 compatible power header (**J1**)
11. PSoC™ 4100S Max MCU program and debug header (**J12**)
12. Arduino UNO R3 compatible I/O headers (**J2**, **J3** and **J4**)
13. PSoC™ 4100S Max MCU user button (**SW2**)
14. CAPSENSE™ buttons (**CSB1** and **CSB2**)
15. User LEDs (**D1, D4** and **D5**)
16. Flat flex cable (FFC) connector (40-pin) to connect to capacitive sensing expansion board (**J9**)
17. Thermistor (**TH1**)
18. PSoC™ 4100S Max MCU (CY8C4149AZI-S598, **U1**)
19. PSoC™ 4100S Max MCU extended I/O Header (**J6**) (footprint only)
20. External power supply connector (**J5**) (footprint only)

The capacitive sensing board has the following features (see **Figure 6**):

1. Flat flex cable (FFC) connector (**J2**)
2. Proximity (**CSP1**)
3. Touchpad (**CST1**)
4. Flat flex cable (FFC) connector (**J1**)
5. Slider (**CSS1**)

## Introduction

**Figure 2** shows the pinout of the PSoC™ 4100S Max pioneer board.

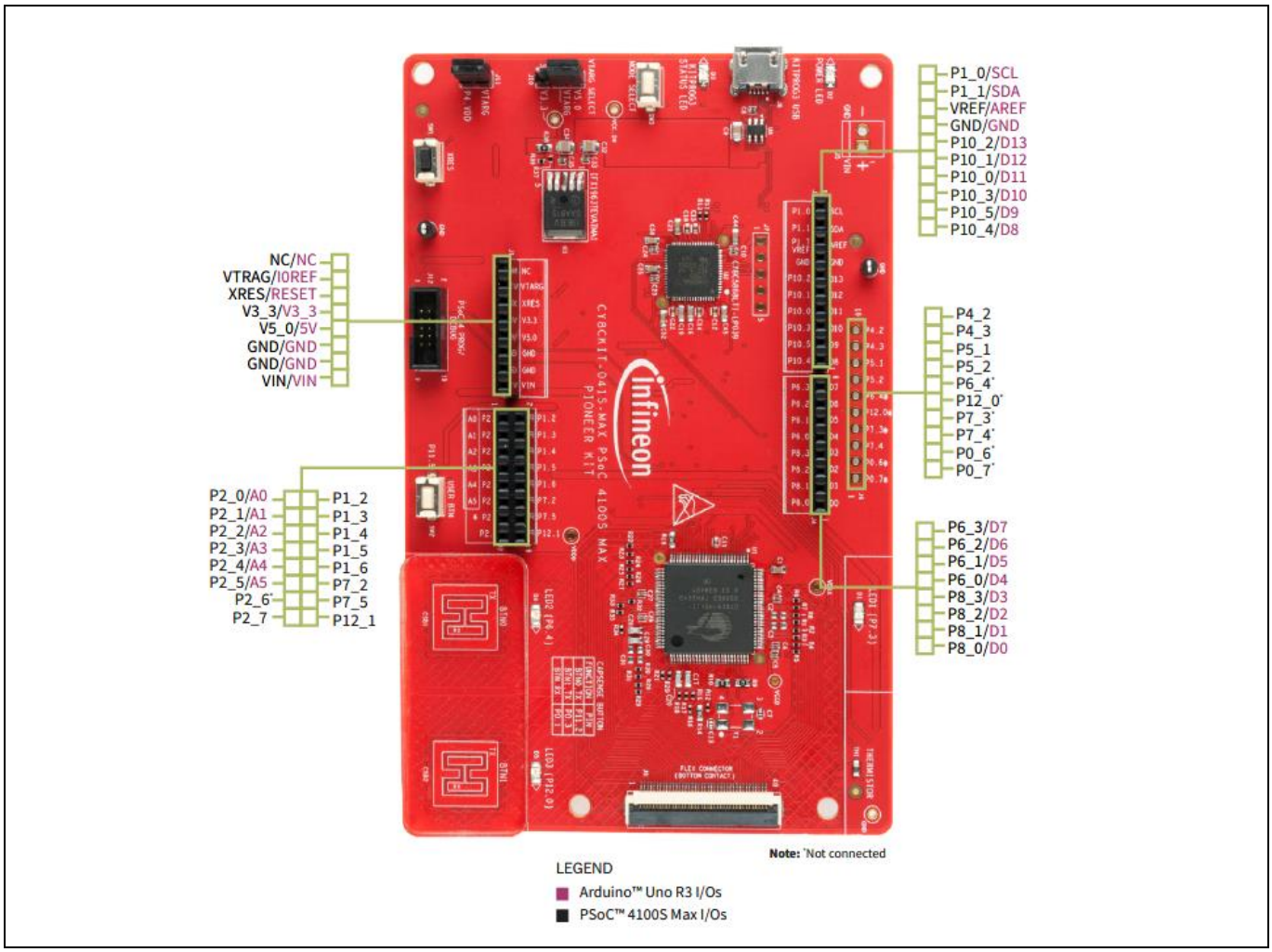



**Figure 2 CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer board pinout**

**Table 1 PSoC™ 4100S Max pioneer board pinout**

| Pin | Primary on-board function | Secondary on-board function | Connection details |
|---|---|---|---|
| XRES | Hardware reset | – | – |
| P0[0] | FFC connector – J9.28 | – | Touchpad side – CAPSENSE™ Touchpad (RX0)<br>Slider side – None |
| P0[1] | CAPSENSE™ button (RX) | – | – |
| P0[2] | CAPSENSE™ shield 1 | – | – |
| P0[3] | CAPSENSE™ button 1 (TX) | – | – |
| P0[4] | UART_TX | – | – |
| P0[5] | UART_RX | – | – |
| P0[6] | UART_RTS | GPIO on J6.2 header | Remove R10 and mount R62 to connect to Arduino header |

| Pin | Primary on-board function | Secondary on-board function | Connection details |
|---|---|---|---|
| P0[7] | UART_CTS | GPIO on J6.1 header | Remove R9 and mount R58 to connect to Arduino header |
| P1[0] | I2C_SCL (J3.10) | – | – |
| P1[1] | I2C_SDA (J3.9) | – | – |
| P1[2] | GPIO on J2.2 header | – | – |
| P1[3] | GPIO on J2.4 header | – | – |
| P1[4] | GPIO on J2.6 header | – | – |
| P1[5] | GPIO on J2.8 header | – | – |
| P1[6] | GPIO on J2.10 header | – | – |
| P2[0] | Arduino A0 (J2.1) | – | – |
| P2[1] | Arduino A1 (J2.3) | – | – |
| P2[2] | Arduino A2 (J2.5) | – | – |
| P2[3] | Arduino A3 (J2.7) | – | – |
| P2[4] | Arduino A4 (J2.9) | Thermistor VDD | Remove R66 and mount R64 to connect to thermistor |
| P2[5] | Arduino A5 (J2.11) | Thermistor Out | Remove R73 and mount R71 to connect to thermistor |
| P2[6] | Thermistor out | GPIO on J2.13 header | Remove R76 and mount R81 to connect to Arduino header |
| P2[7] | GPIO on J2.15 header | Thermistor GND | Remove R91 and mount R88 to connect to thermistor |
| P3[0] | FFC connector – J9.5 | – | Touchpad side – CAPSENSE™ Touchpad (TX8)<br>Slider side – CAPSENSE™ slider (TX) |
| P3[1] | FFC connector – J9.6 | – | Touchpad side – CAPSENSE™ Touchpad (TX9)<br>Slider side – None |
| P3[4] | FFC connector – J9.7 | – | Touchpad side – CAPSENSE™ Touchpad (TX10)<br>Slider side – None |
| P3[5] | FFC connector – J9.8 | – | Touchpad side – CAPSENSE™ Touchpad (TX11)<br>Slider side – CAPSENSE™ slider (RX7) |
| P3[6] | FFC connector – J9.9 | – | Touchpad side – CAPSENSE™ Touchpad (TX12)<br>Slider side – CAPSENSE™ slider (RX6) |
| P3[7] | FFC connector – J9.10 | – | Touchpad side – CAPSENSE™ Touchpad (TX13)<br>Slider side – CAPSENSE™ slider (RX5) |
| P4[4] | FFC connector – J9.18 | – | Touchpad side – CAPSENSE™ Touchpad (RX8)<br>Slider side – CAPSENSE™ slider (RX3) |
| P4[5] | FFC connector – J9.19 | – | Touchpad side – CAPSENSE™ Touchpad (RX7)<br>Slider side – CAPSENSE™ slider (RX2) |
| P4[6] | FFC connector – J9.20 | – | Touchpad side – CAPSENSE™ Touchpad (RX6)<br>Slider side – CAPSENSE™ slider (RX1) |

| Pin | Primary on-board function | Secondary on-board function | Connection details |
|---|---|---|---|
| P4[7] | FFC connector – J9.21 | – | Touchpad side – CAPSENSE™ Touchpad (RX5)<br>Slider side – CAPSENSE™ slider (RX0) |
| P5[0] | FFC connector – J9.36 | – | Touchpad side – CAPSENSE™ Touchpad (TX4)<br>Slider side – None |
| P5[3] | FFC connector – J9.37 | – | Touchpad side – CAPSENSE™ Touchpad (TX5)<br>Slider side – None |
| P5[4] | FFC connector – J9.38 | – | Touchpad side – CAPSENSE™ Touchpad (TX6)<br>Slider side – None |
| P5[5] | FFC connector – J9.39 | – | Touchpad side – CAPSENSE™ Touchpad (TX7)<br>Slider side - None |
| P5[6] | FFC connector – J9.24 | – | Touchpad side – CAPSENSE™ Touchpad (RX4)<br>Slider side – None |
| P5[7] | FFC connector – J9.25 | – | Touchpad side – CAPSENSE™ Touchpad (RX3)<br>Slider side – None |
| P6[0] | Arduino D4 (J4.5) | – | – |
| P6[1] | Arduino D5 (J4.6) | – | – |
| P6[2] | Arduino D6 (J4.7) | – | – |
| P6[3] | Arduino D7 (J4.8) | – | – |
| P6[4] | Orange User LED (D4) | GPIO on extended header J6.6 | Remove R109 and mount R112 to connect to extended header |
| P6[5] | FFC connector – J9.2 | – | Touchpad side – CAPSENSE™ proximity<br>Slider side – None |
| P7[2] | GPIO on J2.12 header | – | – |
| P7[3] | Red User LED (D1) | GPIO on extended header J6.4 | Remove R42 and mount R43 to connect to extended header |
| P7[4] | GPIO on extended header J6.3 | – | – |
| P7[5] | GPIO on J2.14 header | – | – |
| P7[6] | FFC connector – J9.26 | – | Touchpad side – CAPSENSE™ Touchpad (RX2)<br>Slider side – None |
| P7[7] | FFC connector – J9.27 | – | Touchpad side – CAPSENSE™ Touchpad (RX1)<br>Slider side – None |
| P8[0] | Arduino D0 (J4.1) | – | – |
| P8[1] | Arduino D1 (J4.2) | – | – |
| P8[2] | Arduino D2 (J4.3) | – | – |
| P8[3] | Arduino D3 (J4.4) | – | – |
| P9[0] | FFC connector – J9.32 | – | Touchpad side – CAPSENSE™ Touchpad (TX0)<br>Slider side – None |
| P9[1] | FFC connector – J9.33 | – | Touchpad side – CAPSENSE™ Touchpad (TX1)<br>Slider side – None |

| Pin | Primary on-board function | Secondary on-board function | Connection details |
|---|---|---|---|
| P9[2] | FFC connector – J9.34 | – | Touchpad side – CAPSENSE™ Touchpad (TX2)<br>Slider side – None |
| P9[3] | FFC connector – J9.35 | – | Touchpad side – CAPSENSE™ Touchpad (TX3)<br>Slider side – None |
| P10[0] | Arduino D11 (J3.4) | – | – |
| P10[1] | Arduino D12 (J3.5) | – | – |
| P10[2] | Arduino D13 (J3.6) | – | – |
| P10[3] | Arduino D10 (J3.3) | – | – |
| P10[4] | Arduino D8 (J3.1) | – | – |
| P10[5] | Arduino D9 (J3.2) | – | – |
| P11[0] | FFC connector – J9.11 | – | Touchpad side – CAPSENSE™ Touchpad (TX14)<br>Slider side – None |
| P11[1] | FFC connector – J9.12 | – | Touchpad side – CAPSENSE™ Touchpad (TX15)<br>Slider side – None |
| P11[2] | CAPSENSE™ button 0 (TX) | – | – |
| P11[3] | CAPSENSE™ shield 0 | – | – |
| P11[4] | FFC connector – J9.17 | – | Touchpad side – CAPSENSE™ Touchpad (RX9)<br>Slider side – CAPSENSE™ slider (RX4) |
| P11[5] | User button | – | – |
| P12[0] | Orange user LED (D5) | GPIO on extended header J6.5 | Remove R110 and mount R111 to connect to extended header |
| P12[1] | GPIO on J2.16 header | – | – |

## 1.4 Additional learning resources

Infineon provides a wealth of data at **www.cypress.com/psoc** 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

For assistance, visit **Cypress Support** or visit **community.cypress.com** to ask your questions in Cypress Developer Community.

You can also use the following support resources if you need quick assistance:

- **Self-help (Technical Documents)**
- **Local Sales Office Locations**

## 1.6 Documentation conventions

**Table 2 Document conventions for guides**

| Convention | Usage |
|---|---|
| Courier New | Displays file locations, 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 |
| BOM | Bill of Materials |
| BSP | Board Support Package |
| CLI | Command Line Interface |
| CMOD | Modulator Capacitor |
| CMSIS-DAP | Cortex Microcontroller System Interface Standard – Debug Access Port |
| CPU | Central Processing Unit |
| CSD | Self-Capacitance |
| CSX | Mutual-Capacitance |
| CTS | Clear to Send (UART) |
| DC | Direct current |
| ECO | External Crystal Oscillator |
| EMC | Electromagnetic Compatibility |
| ESD | Electrostatic Discharge |
| FFC | Flat Flex Cable |
| GND | Ground |
| GPIO | General Purpose Input/Output |
| HID | Human Interface Device |
| I2C | Inter-Integrated Circuit |
| IDE | Integrated Development Environment |
| LED | Light Emitting Diode |
| MCU | Micro Controller Unit |

## Introduction

| Abbreviation | Definition |
|---|---|
| MISO | Master In Slave Out (SPI) |
| MOSI | Master Out Slave In (SPI) |
| MSC | Multi Sense Converter |
| NTC | Negative Temperature Coefficient |
| OOB | Out-of Box |
| PSoC™ | Programmable System on Chip |
| PWM | Pulse Width Modulation |
| QSPI | Quad Serial Peripheral Interface |
| RTS | Request to Send (UART) |
| SAR | Successive Approximation Register |
| SCL | Serial Clock (I2C) |
| SDA | Serial Data (I2C) |
| SWD | Serial Wire Debug |
| UART | Universal Asynchronous Receiver-Transmitter |
| USB | Universal Serial Bus |
| VIN | Voltage-In |
| WCO | Watch Crystal Oscillator |
| XRES | External Reset |

# 2 Kit operation

This chapter introduces you to various feature of PSoC™ 4100S Max pioneer kit 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™ 4100S Max pioneer board is built around a PSoC™ 4100S Max MCU. **Figure 3** shows the block diagram of the PSoC™ 4100S Max MCU device used on the board. For details of device features, see the **device datasheet**.

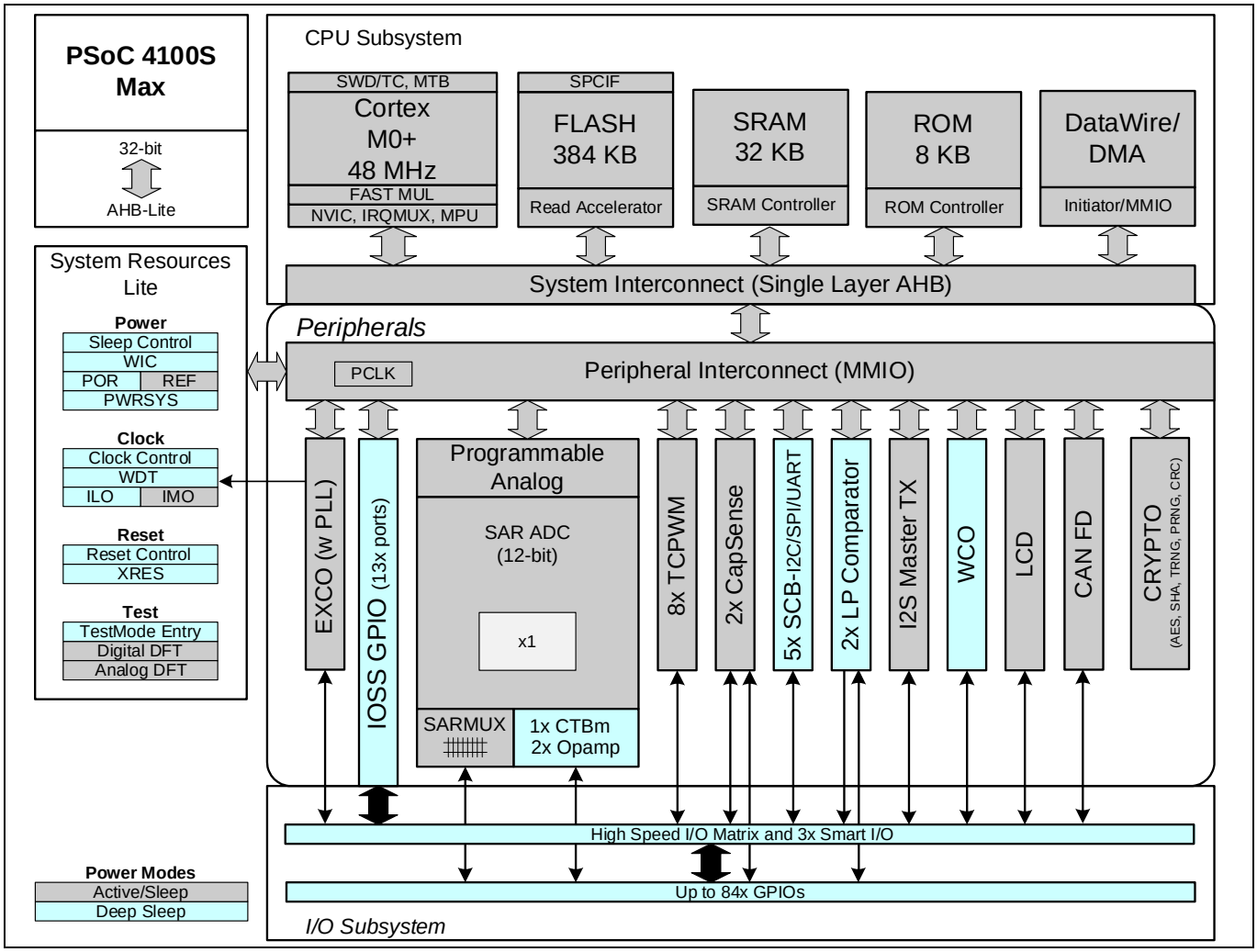



**Figure 3 PSoC™ 4100S Max MCU block diagram**

## Kit operation

**Figure 4** shows the block diagram of the PSoC™ 4100S Max pioneer board.

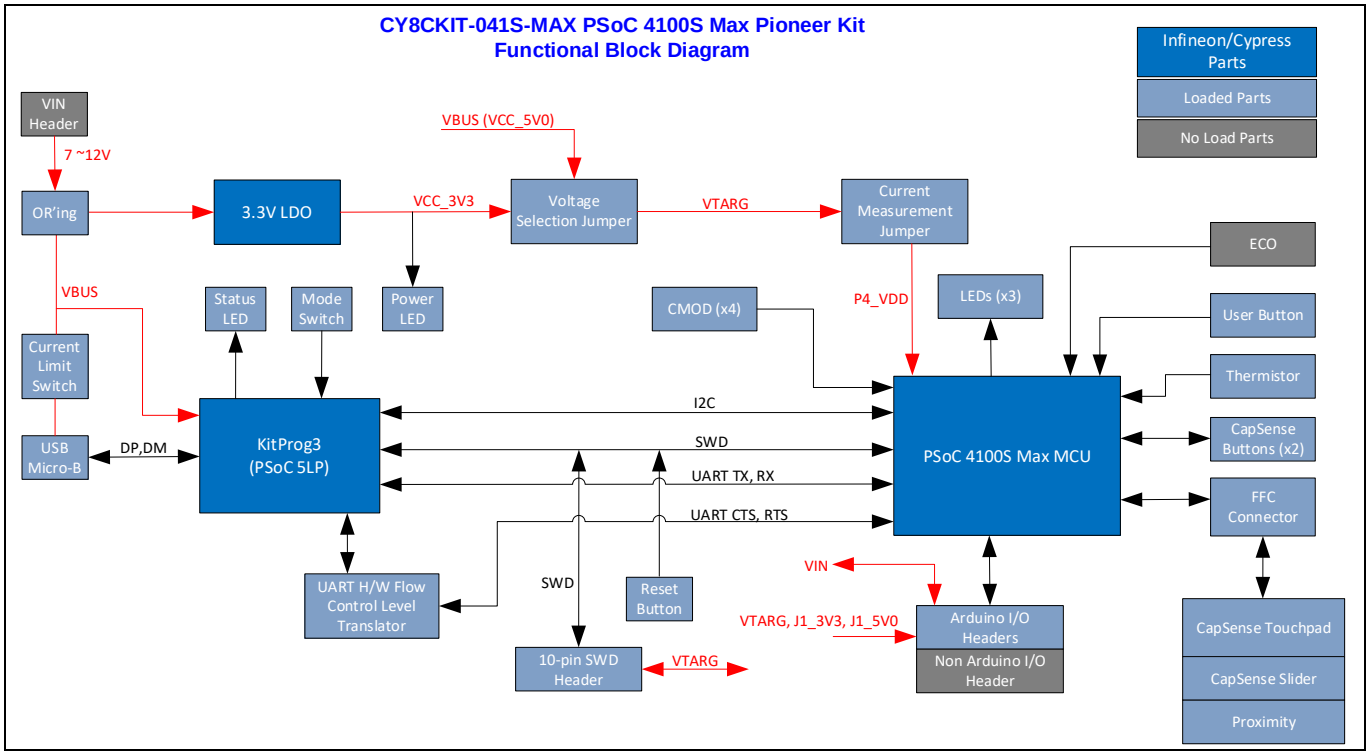



**Figure 4 Functional block diagram of CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer board**

The CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer kit comes with two boards, PSoC™ 4100S Max pioneer board and capacitive sensing expansion board. **Figure 5** shows the markup of the PSoC™ 4100S Max pioneer board.

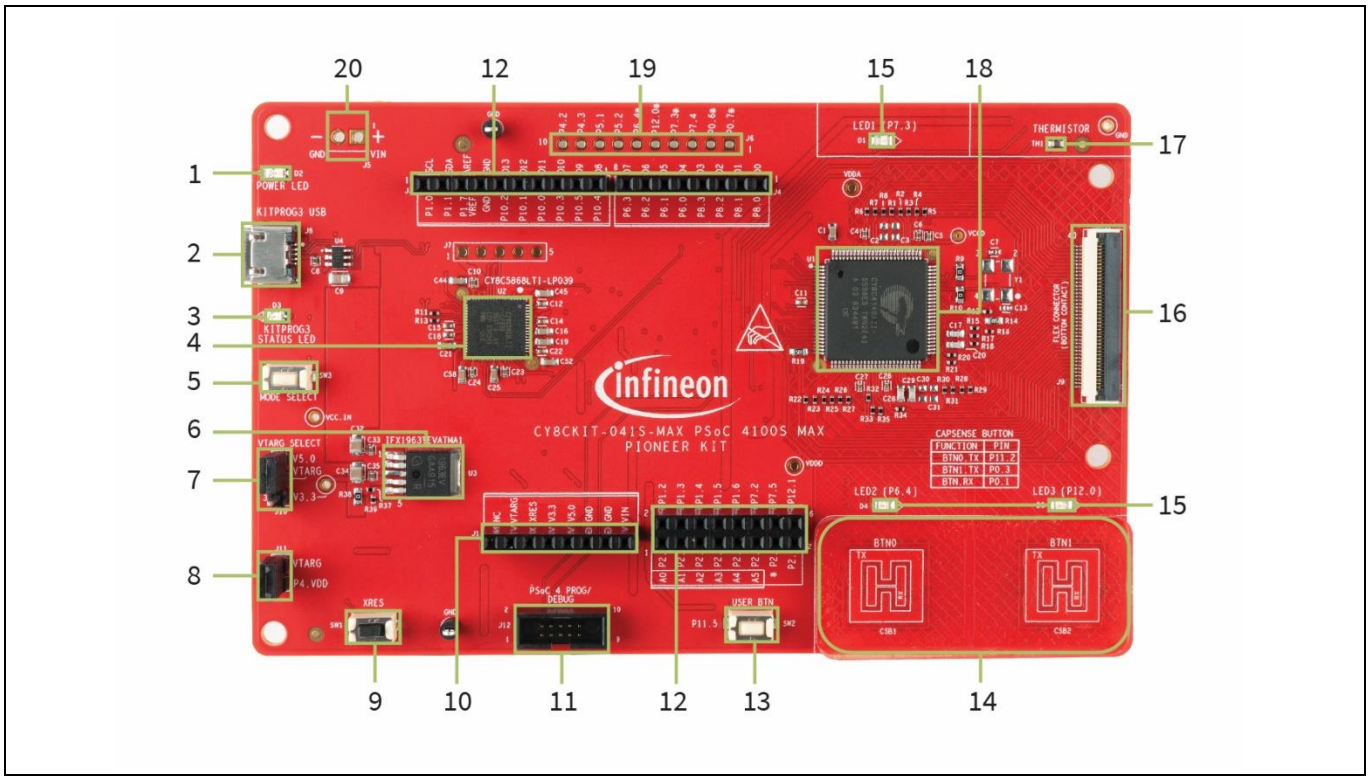



**Figure 5 CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer board details**

### Kit operation

The PSoC™ 4100S Max pioneer board has the following peripherals:

1. **Power LED (D2)**: This Yellow LED indicates the status of power supplied to board.
2. **KitProg3 USB connector (J8)**: The USB cable provided along with the 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. **KitProg3 status LED (D3)**: This yellow LED indicates the status of KitProg3. For details on the KitProg3 status, see the **KitProg3 User Guide**.
4. **KitProg3 (PSoC™ 5LP) programmer and debugger (CY8C5868LTI-LP039, U2)**: 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**.
5. **KitProg3 programming mode selection button (SW3)**: This button function is reserved for future use. Note that the button connects the PSoC™ 5LP pin to ground when pressed. For more details, see the **KitProg3 User Guide**.
6. **Voltage regulator (U3):** The kit has an onboard regulator IFX1963 that is configured to generate 3.3 V, this voltage is derived from the USB connector or from the external power supply connector.
7. **PSoC™ 4100S Max MCU VDD power selection jumper (J10)**: This jumper is used to select the PSoC™ 4100S Max MCU VDD supply voltage between 5.0 V and 3.3 V.
8. **PSoC™ 4100S Max MCU VDD current measurement header (J11)**: An ammeter can be connected to this header to measure the current consumed by the PSoC™ 4100S Max MCU.
9. **PSoC™ 4100S Max MCU reset button (SW1)**: This button is used to reset the PSoC™ 4100S Max MCU. It connects PSoC™ 4100S Max MCU reset (XRES) pin to ground when pressed.
10. **Arduino-compatible power header (J1)**: This header powers Arduino shields. It also has a provision to power the kit though the VIN input.
11. **PSoC™ 4100S Max MCU program and debug header (J12)**: This 10-pin header allows you to program and debug the PSoC™ 4100S Max MCU using an external programmer such as MiniProg4. It is not typically used since the kit has an onboard programmer/debugger.
12. **Arduino Uno R3-compatible I/O headers (J2, J3, and J4)**: These I/O headers bring out pins from the PSoC™ 4100S Max MCU to interface with Arduino shields. Some of these pins are multiplexed with onboard peripherals and are not connected to PSoC™ 4100S Max MCU by default. For detailed information on how to rework the kit to access these pins, see **3.3 CY8CKIT-041S-MAX Kit rework**.
13. **PSoC™ 4100S Max MCU user button (SW2)**: This button can be used to provide an input to P11[5] pin of PSoC™ 4100S Max MCU. Note that this button connects the PSoC™ 4100S Max MCU pin to ground when pressed, so you need to configure the PSoC™ 4100S Max 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.
14. **CAPSENSE™ buttons (CSB1 and CSB2)**: Two CAPSENSE™ buttons which can be configured as self-capacitance (CSD) or mutual-capacitance (CSX) buttons, allow you to evaluate Infineon' fifth-generation CAPSENSE™ technology. These buttons have a 3-mm acrylic overlay for smooth touch sensing.
15. **User LEDs (D1, D4 and D5)**: These three user LEDs can operate at the entire operating voltage range of the PSoC™ 4100S Max MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs. D4 and D5 LEDs are placed above CAPSENSE™ buttons and can be used to show the CAPSENSE™ button status.
16. **Flat Flex Cable (FFC) connector (J9):** This connector is used to connect the expansion board to the pioneer board. This connector brings out 29 I/Os of PSoC™ 4100S Max MCU. The cable should be oriented so that the metal contacts are down (i.e. against the top of the PCBs).
17. **Thermistor (TH1):** The sensor allows you to evaluate low-power analog blocks of the PSoC™ 4100S Max MCU.

18. **PSoC™ 4100S Max MCU (CY8C4149AZI-S598, U1)**: This kit is designed to highlight the features of the PSoC™ 4100S Max MCU. For details on PSoC™ 4100S Max MCU pin mapping, refer to **Table 1**.
19. **PSoC™ 4100S Max MCU extended I/O header (J6)**: This header provides connectivity to 8 PSoC™ 4100S Max 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.
20. **External power supply VIN connector (J5)**: This connector connects an external DC power supply input to the onboard regulators. This is not loaded by default.

See **3.2 Hardware functional description** for details on various hardware blocks.

**Figure 6** shows the markup of the capacitive sensing expansion board.

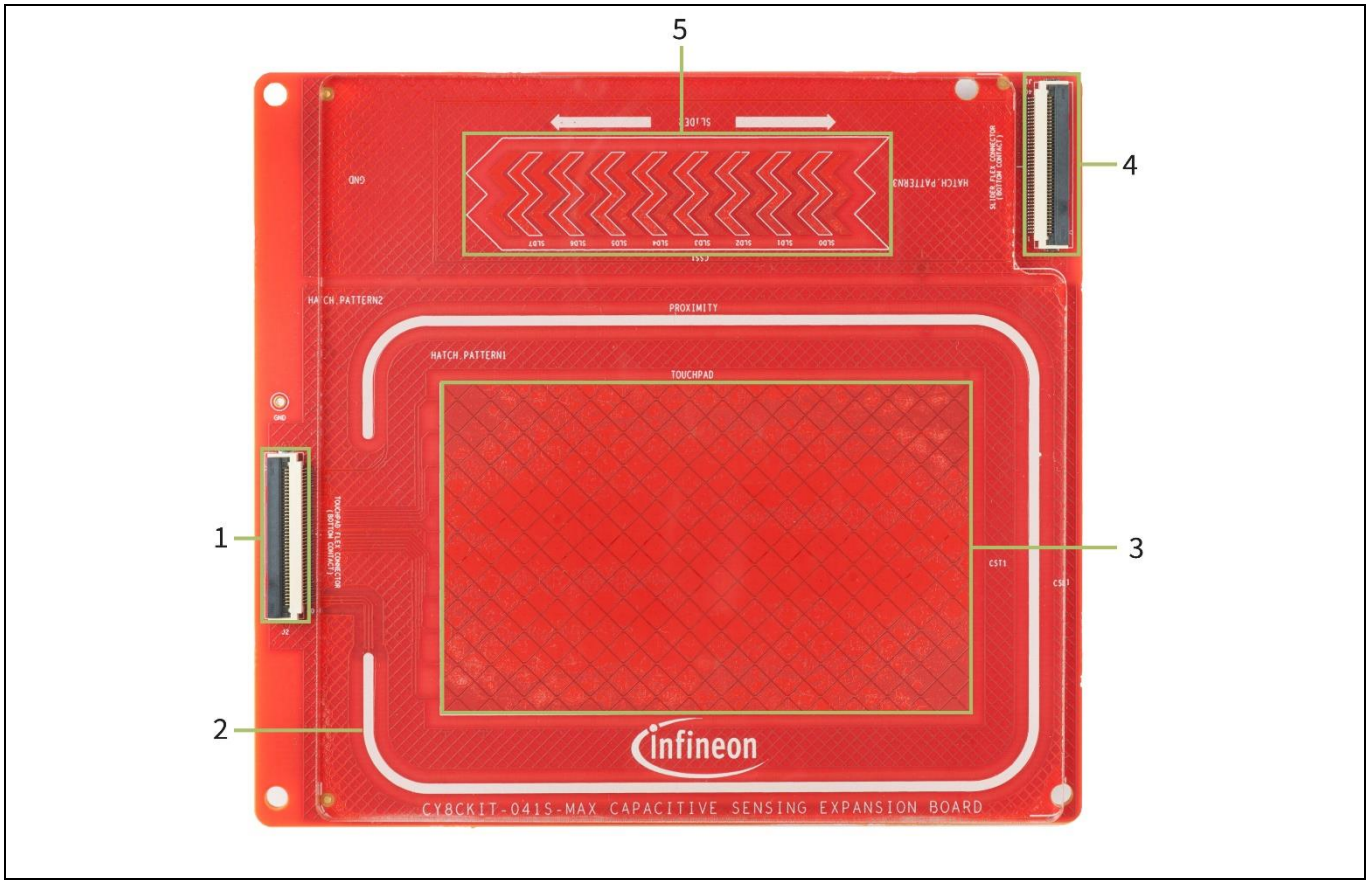



**Figure 6 CY8CKIT-041S-MAX capacitive sensing expansion board details**

The capacitive sensing expansion board has the following peripherals:

1. **Flat Flex Cable (FFC) connector (J2):** This connector is used to connect the expansion board to the pioneer board on the Touchpad side.
2. **Proximity (CSP1)**: Proximity loop with ~11cm diagonal length.
3. **Touchpad (CST1)**: 160-segment touchpad that has 10 rows and 16 columns, which can be configured as a self-capacitance (CSD) or a mutual-capacitance (CSX) based Touchpad.
4. **Flat flex cable (FFC) connector (J1):** This connector is used to connect the expansion board to the pioneer board on the slider side.
5. **Slider (CSS1)**: 8-segment slider, which can be configured as a self-capacitance (CSD) or mutual-capacitance (CSX) based slider.

## 2.2 KitProg3: Onboard programmer/debugger

The PSoC™ 4100S Max pioneer board can be programmed and debugged using the onboard KitProg3. KitProg3 includes programming and debugging along with USB-UART and USB-I2C capabilities. A PSoC™ 5LP device is used to implement the KitProg3 functionality. For more details on the KitProg3 functionality, see the **KitProg3 User Guide**.

## 2.3 Creating a project and program/debug using ModusToolbox™

This section presents a quick overview of creating a project program/debug 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 7**. It enumerates as a USB composite device if you are connecting it to your PC for the first time. KitProg3 operates in CMSIS-DAP Bulk mode, the status LED (yellow) is always ON in CMSIS-DAP Bulk mode. If you do not see the desired LED status, see the **KitProg3 User Guide** for details on the KitProg3 status and troubleshooting instructions.

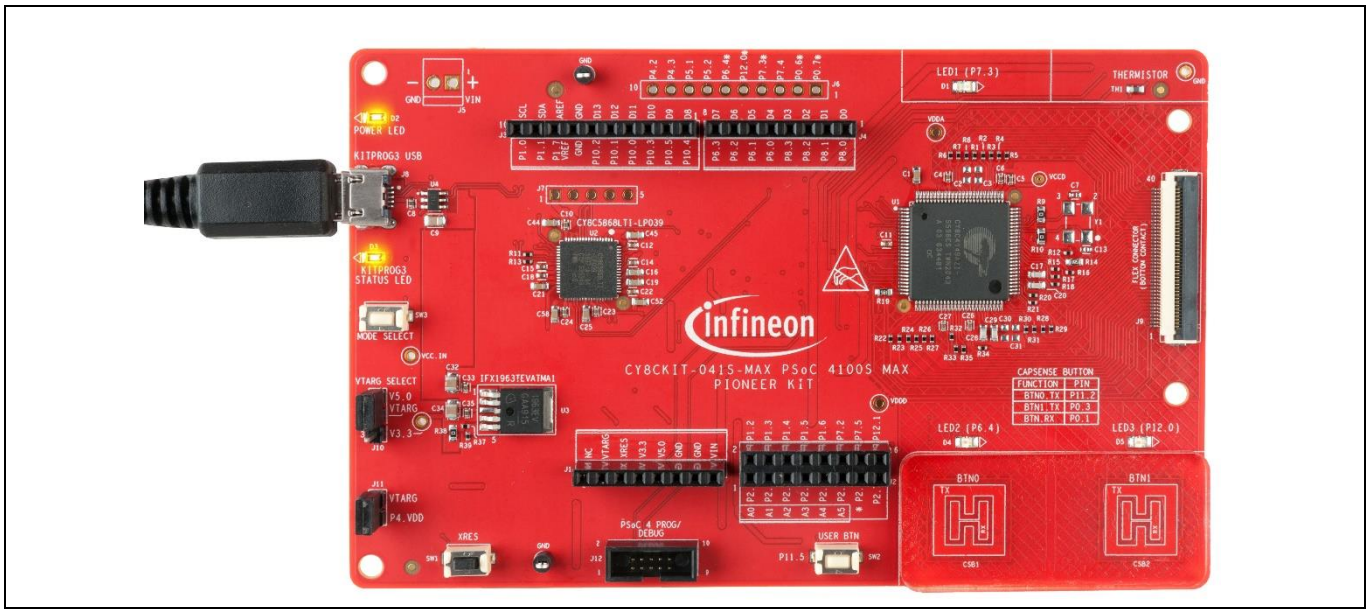



**Figure 7 Connect USB cable to USB connector on the board**

## Kit operation

2. In the ModusToolbox™ IDE, import the desired code example (application) into a new workspace.

   a. Click on New Application from Quick Panel.

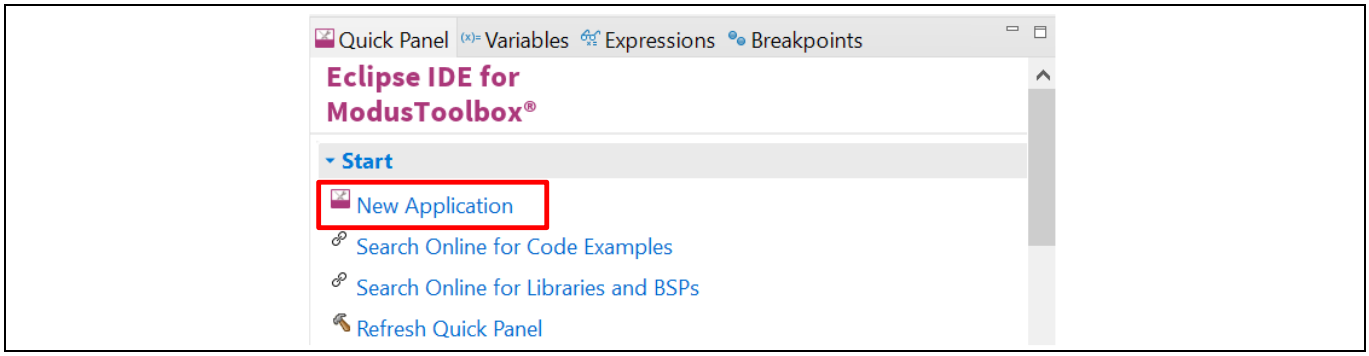



**Figure 8 Example figure**

   b. Select the CY8CKIT-041S-MAX in the Choose Board Support Package (BSP) window and click **Next**, as shown in **Figure 9**.

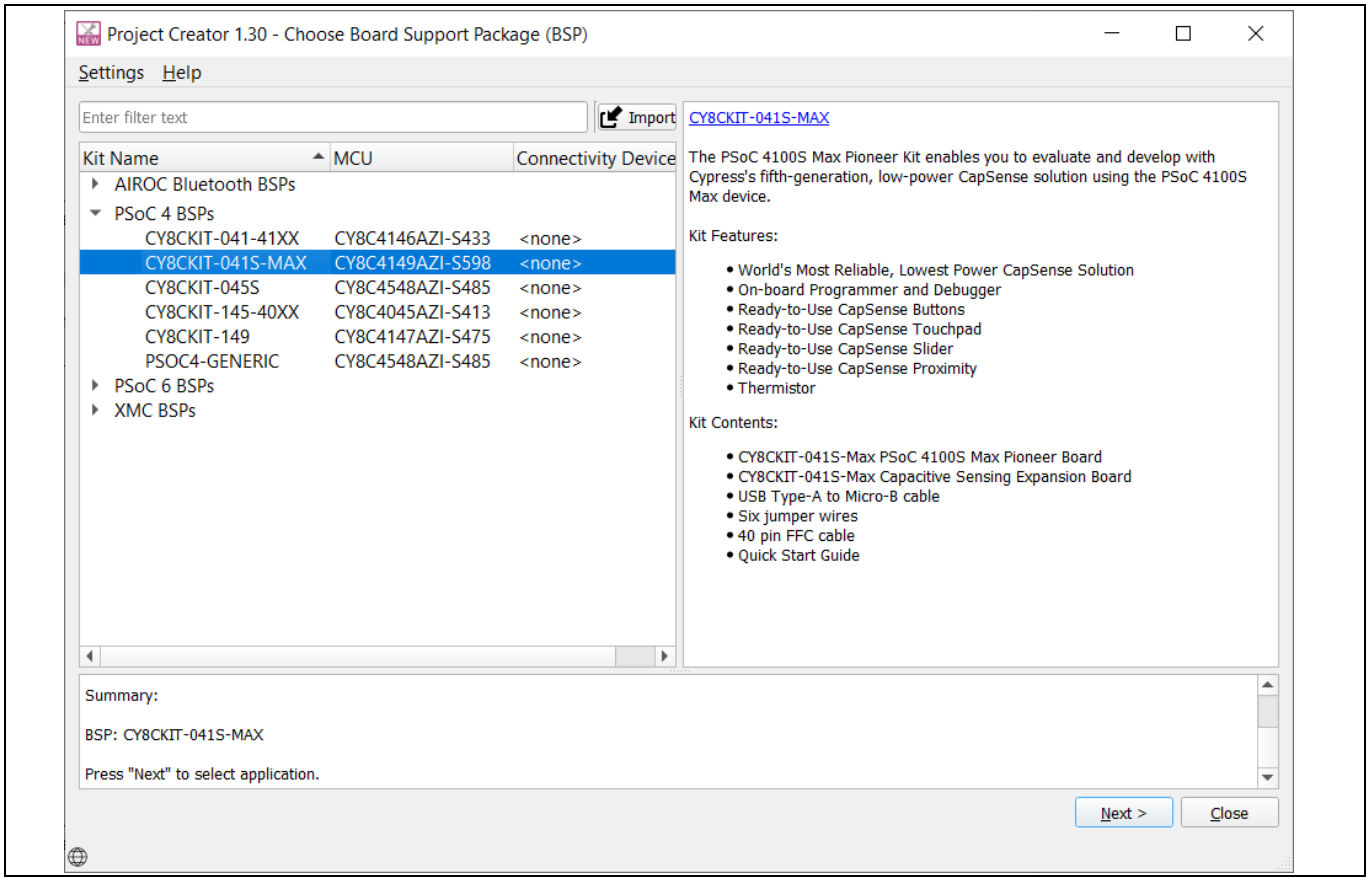



**Figure 9 Board Support Package (BSP) window**

c. Select the application and click **Create**, as shown in **Figure 10**.

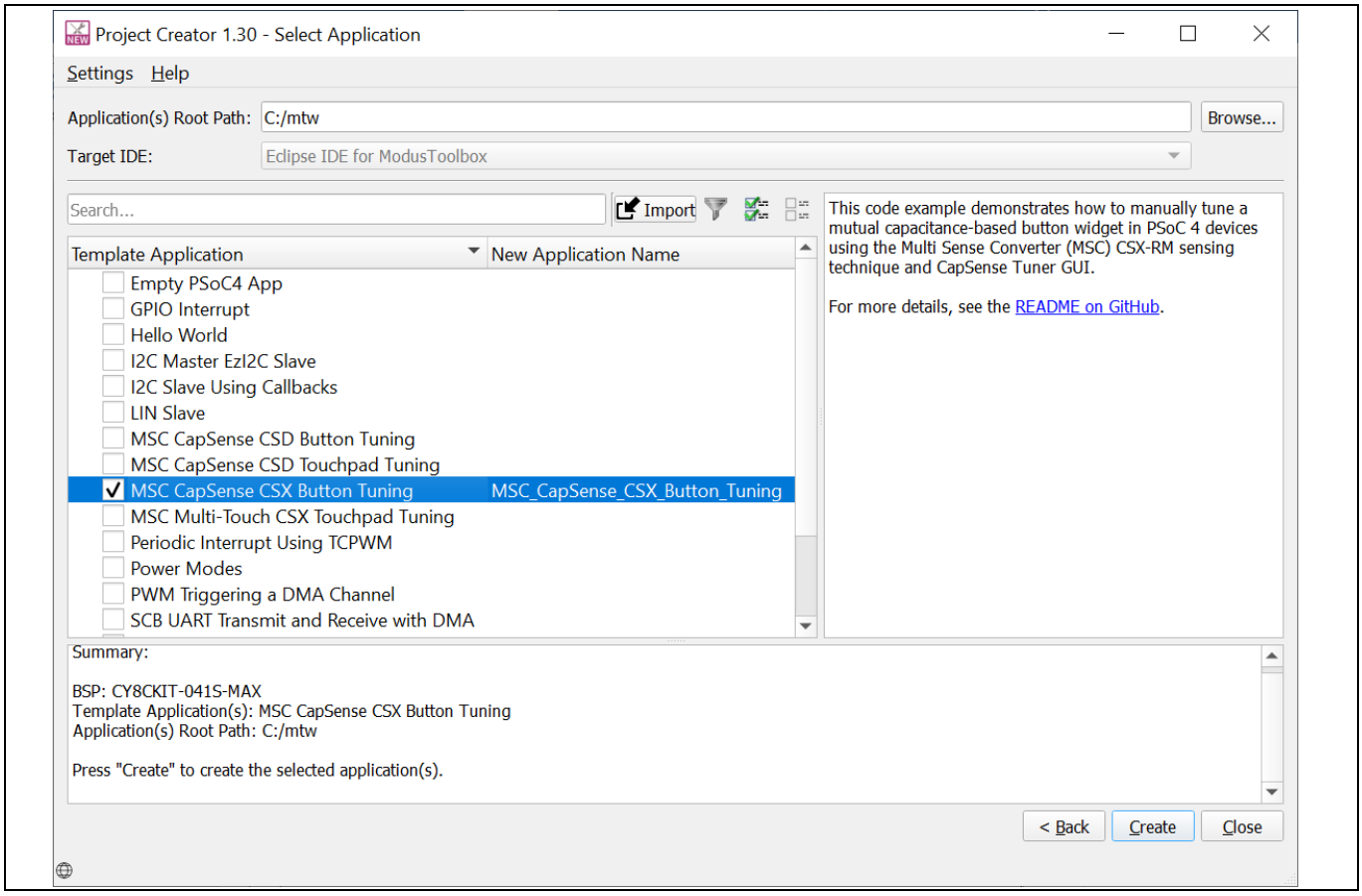



**Figure 10 CY8CKIT-041S-MAX Application selection**

## Kit operation

3. To build and program a PSoC™ 4100S Max 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 11**.

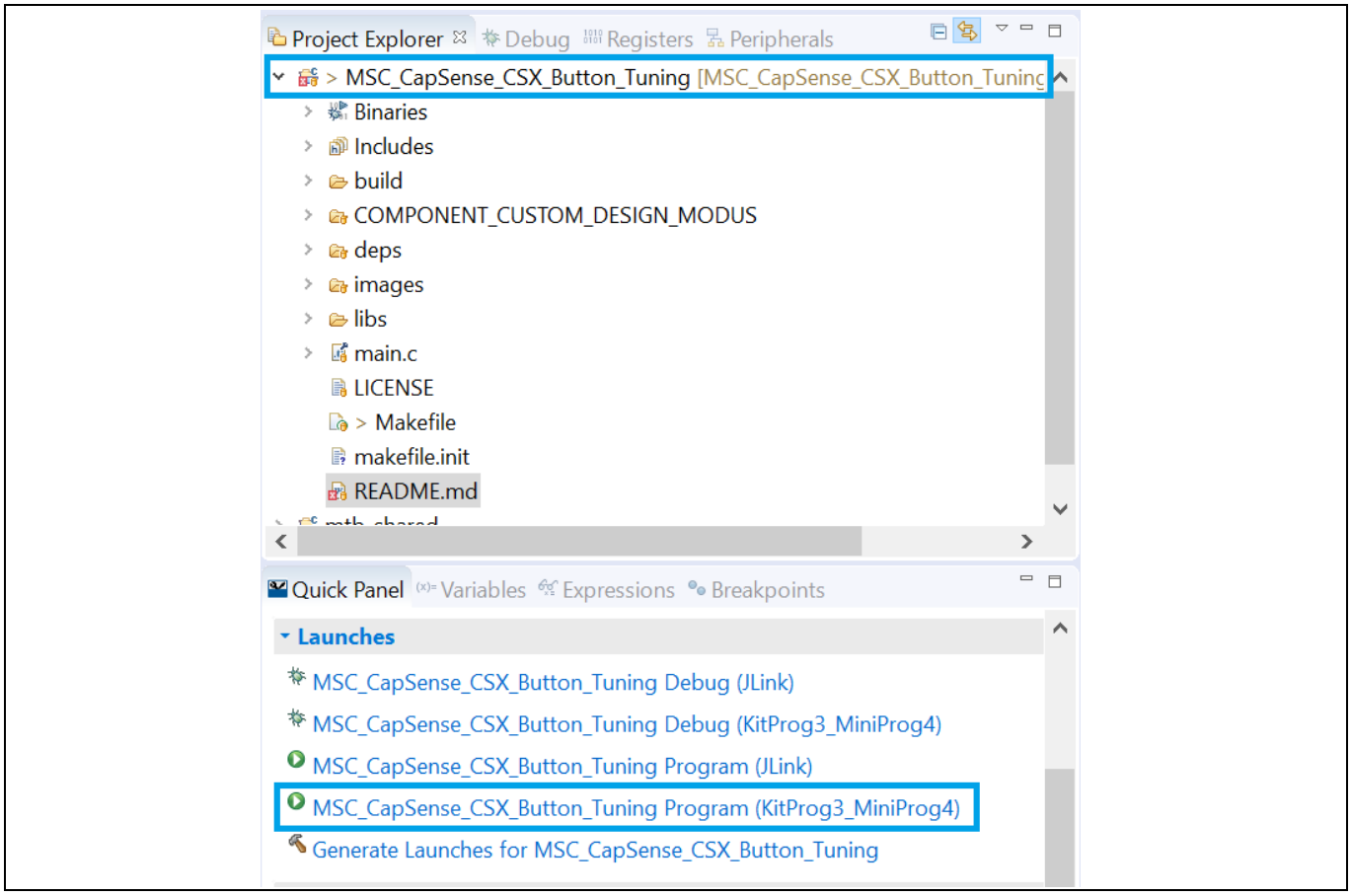



**Figure 11 Build and program the code example**

## Kit operation

4. ModusToolbox™ has an integrated debugger. To debug a PSoC™ 4100S Max 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 12**. For a detailed explanation on how to debug using ModusToolbox™, see **KBA224621**.

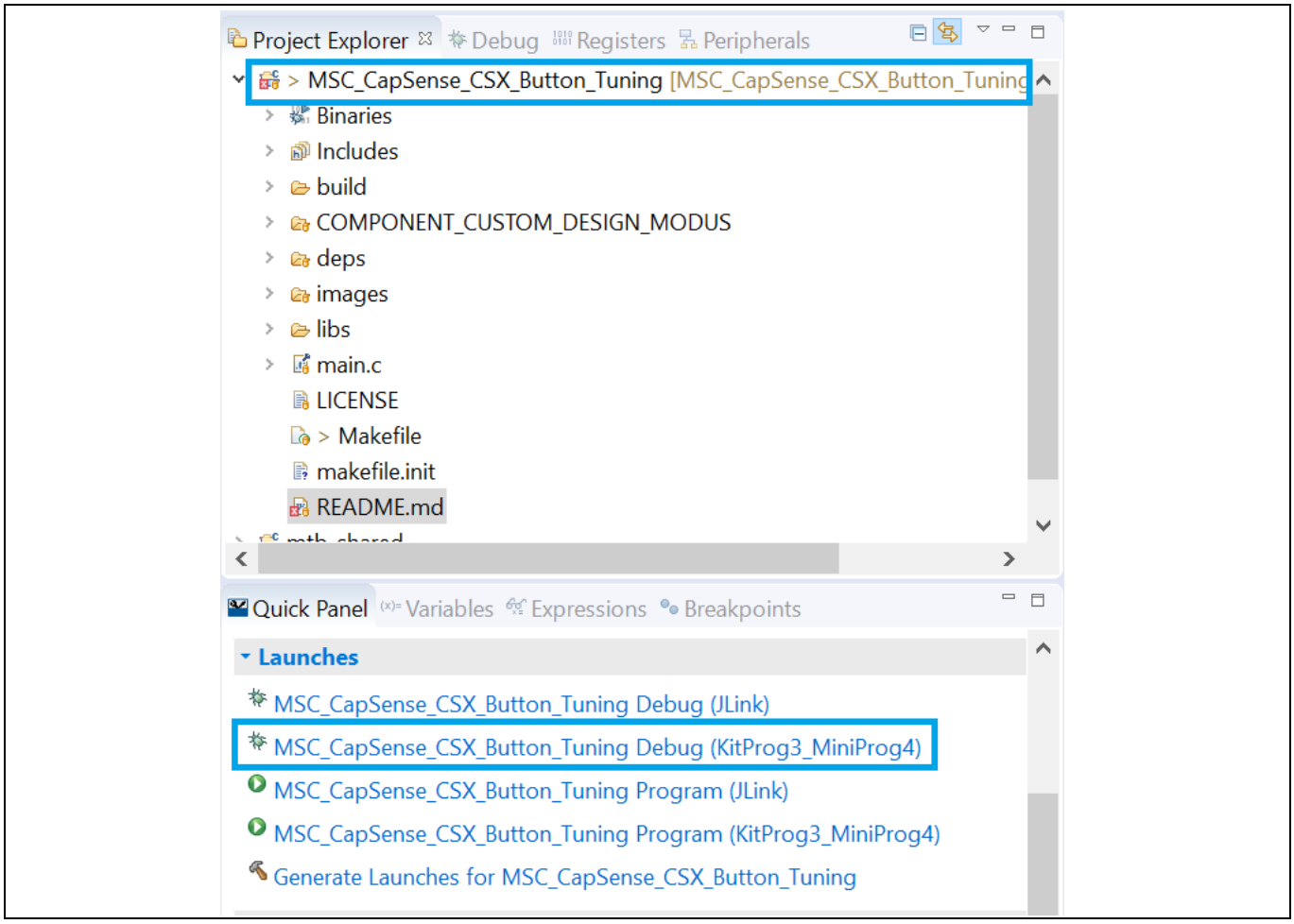



**Figure 12 Debug the code example**

### 2.3.1 Using the OOB example – [MSC CSX button project]

The PSoC™ 4100S Max pioneer board is by default programmed with the code example: **PSoC™ 4: MSC CAPSENSE™ CSX Button Tuning**. The steps below describe on how to use the example. For a detailed description of the project refer to the example's readme file in the GitHub repository or from the application's directory once the example has been created using ModusToolbox™.

Note: At any point of time, if you overwrite the OOB example, you can restore it by programming the PSoC™ 4: MSC CAPSENSE™ CSX Button Tuning code example. Refer **2.3 Creating a project and program/debug using ModusToolbox™** for programming the board.

1. Connect the board to your PC using the provided USB cable through the KitProg3 USB connector.
2. Touch a CAPSENSE™ button and observe the corresponding LED turn ON as **Figure 13** shows.

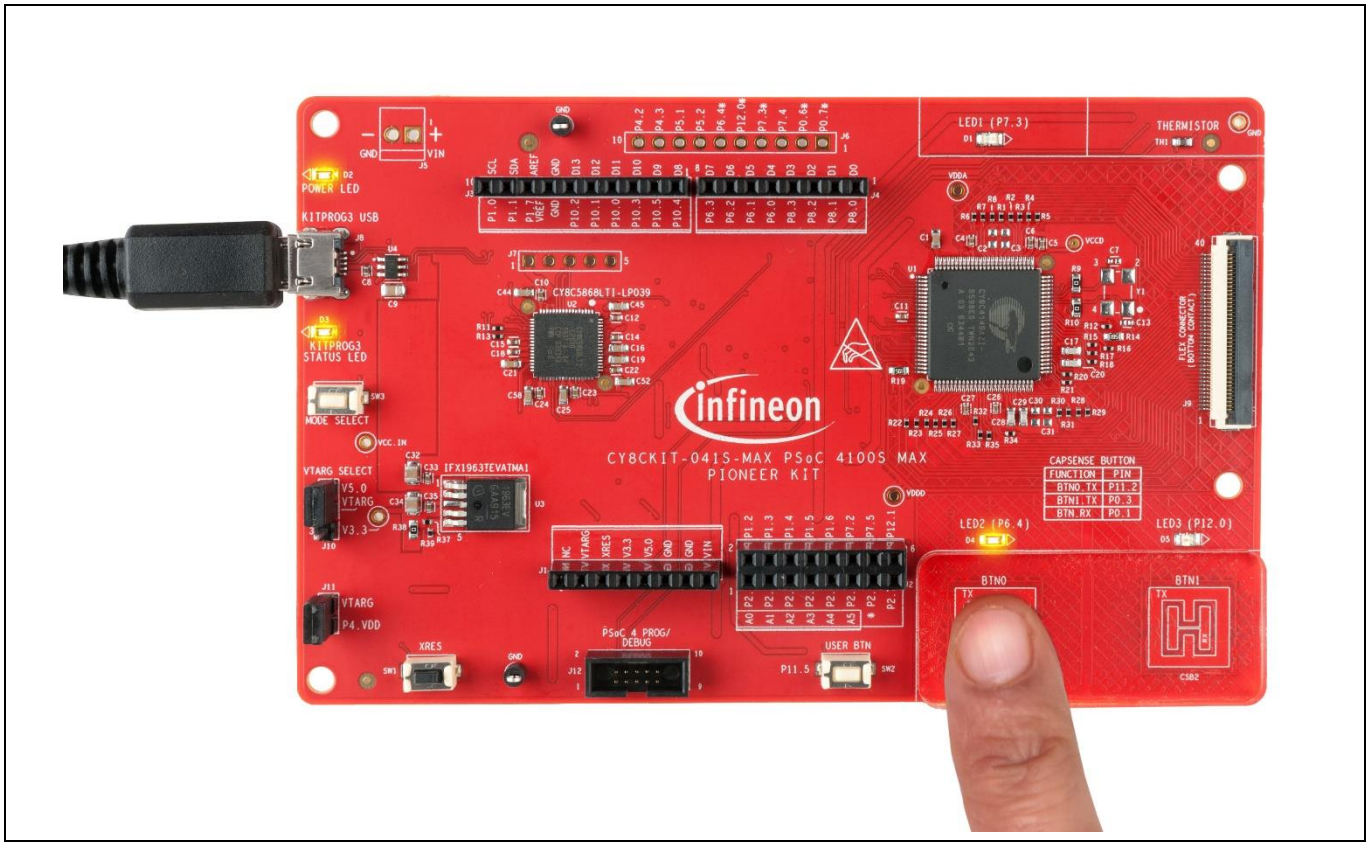



**Figure 13 MSC CAPSENSE™ CSX button tuning project**

*Note: There are more code examples available in ModusToolbox™ IDE (see **Figure 10**) or in GitHub page dedicated to **ModusToolbox™ based examples**, to evaluate the capacitive sensing expansion board: **PSoC™ 4: MSC multi-touch mutual-capacitance touchpad tuning**, **PSoC™ 4: MSC Self-Capacitance Touchpad Tuning**, **PSoC™ 4: MSC Self-Capacitance Slider Tuning** etc.*

### 2.3.2 USB-UART bridge

The KitProg3 on the PSoC™ 4100S Max pioneer board can act as a USB-UART bridge. The primary UART and flow-control lines between the PSoC™ 4100S Max MCU and the KitProg3 are hard-wired on the board, as **Figure 14** shows.

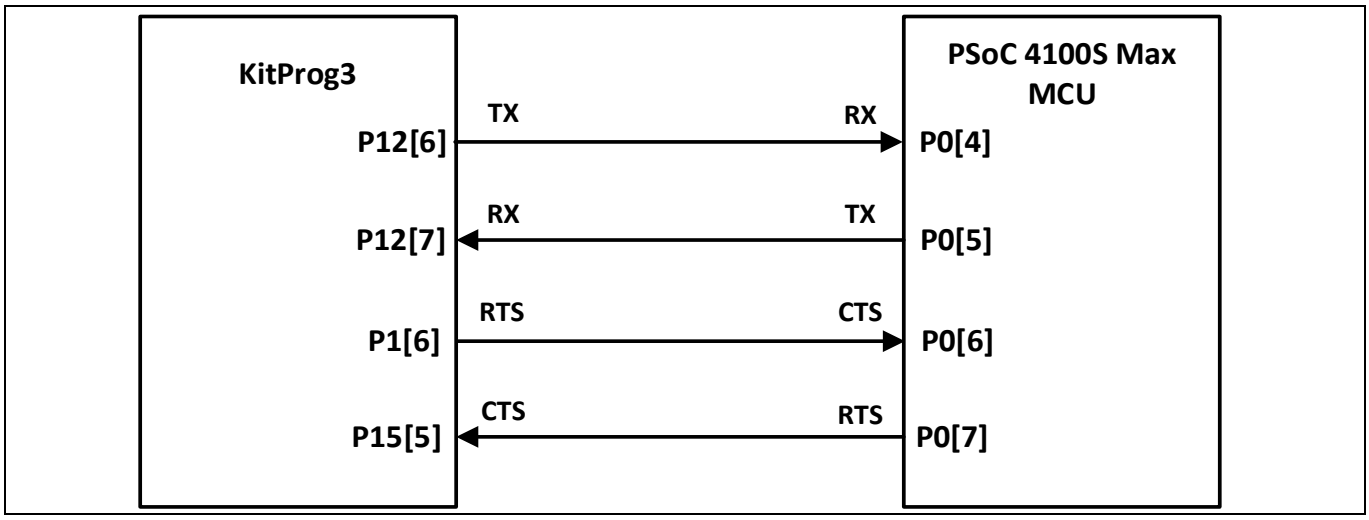



**Figure 14 UART connection between KitProg3 and PSoC™ 4100S Max MCU**

### 2.3.3 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™ 4100S Max MCU are hard-wired on the board to the I2C lines of the KitProg3, with onboard pull-up resistors as **Figure 15** 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**.

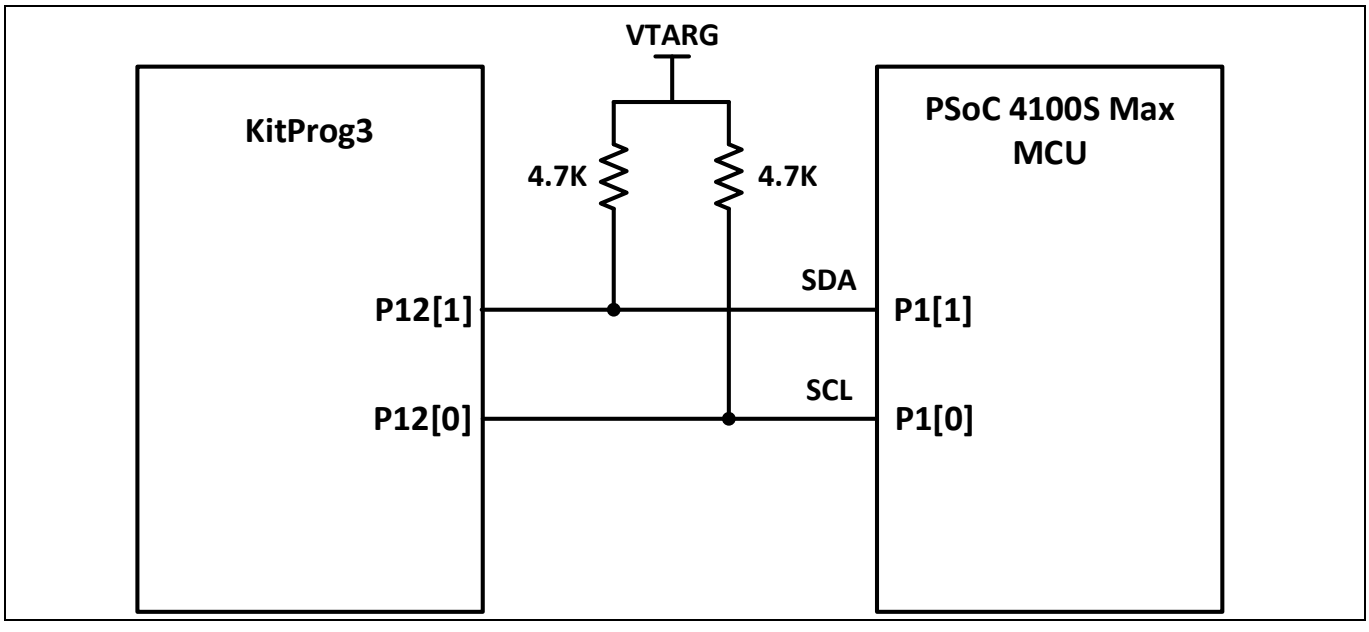



**Figure 15 I2C Connection between KitProg3 and PSoC™ 4100S Max MCU**

# 3 Hardware

## 3.1 Schematics

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

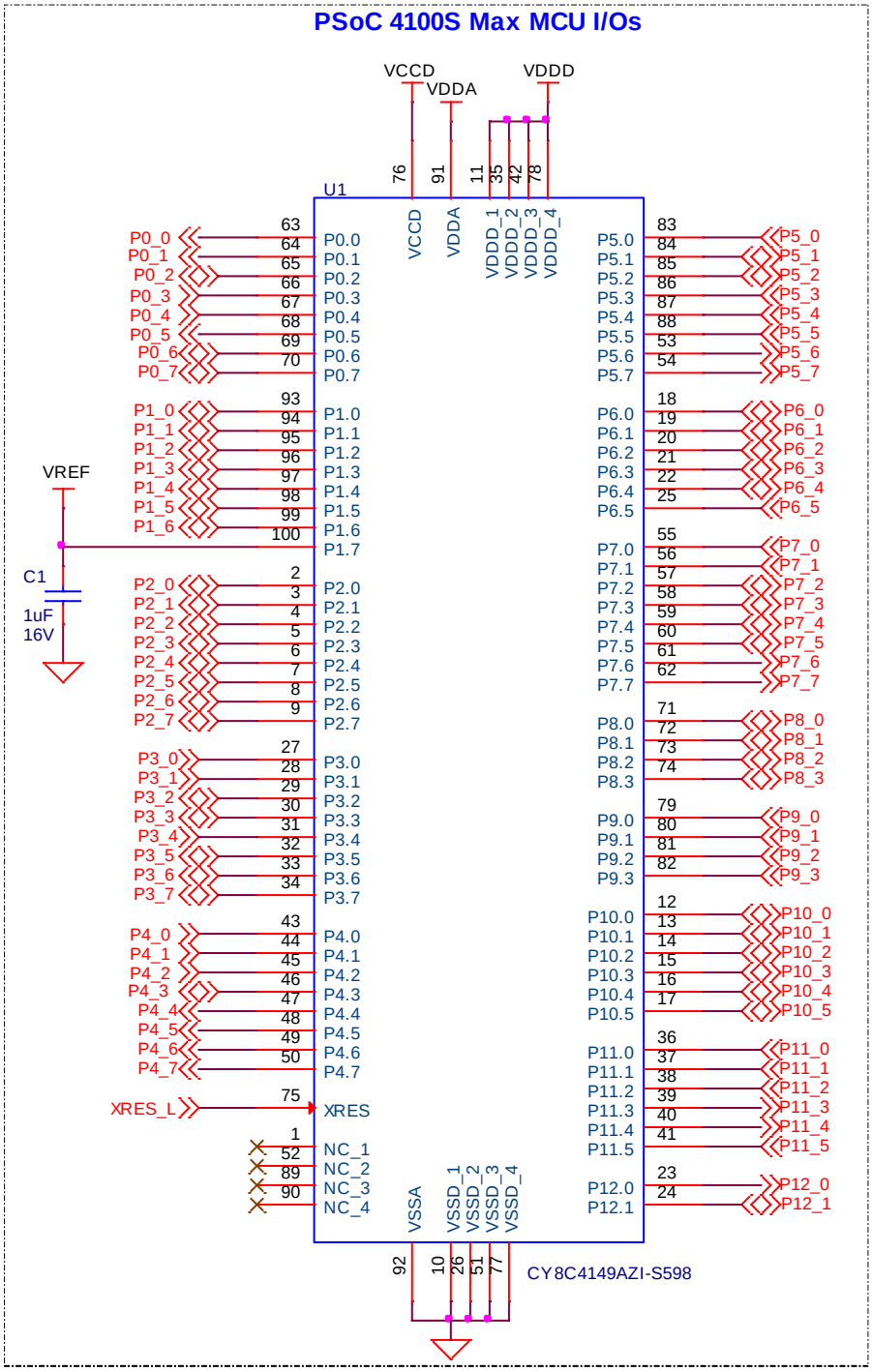
## 3.2 Hardware functional description

This section explains in detail the individual hardware blocks. The kit comes with two boards: CY8CKIT-041S-MAX PSoC™ 4100S Max pioneer board and CY8CKIT-041S-MAX capacitive sensing expansion board. The pioneer board consists of the PSoC™ 4100S Max MCU device, KitProg3 programmer/debugger and bridge, voltage regulators, user LEDs and buttons, CAPSENSE™ buttons, thermistor, crystal oscillators, and other passives required for basic operation of the board. GPIOs of the PSoC™ 4100S Max MCU device are fanned out to expansion headers that include headers in Arduino Uno formfactor and an FFC connector for CAPSENSE™ expansion. The expansion board consists of CAPSENSE™ Touchpad, slider, and proximity sensors.

### 3.2.1 PSoC™ 4100S Max MCU

This kit features a PSoC™ 4100S Max MCU. PSoC™ 4 is a scalable and reconfigurable platform architecture for a family of programmable embedded system controllers with an Arm® Cortex®-M0+ CPU. It combines programmable and reconfigurable analog and digital blocks. PSoC™ 4100S Max MCU is a member of the PSoC™ 4 platform architecture. It is a combination of a microcontroller with standard communication and timing peripherals, a multi-sense capacitive, and inductive sensing system with best-in-class performance, programmable general-purpose continuous-time analog blocks, and programmable connectivity.

For more information, see the **PSoC™ 4100S Max MCU family datasheet**.


PSoC 4100S Max MCU I/Os
VCCD
VDDA
VDDD
U1
VREF
C1
1uF
16V
XRES_L
CY8C4149AZI-S598

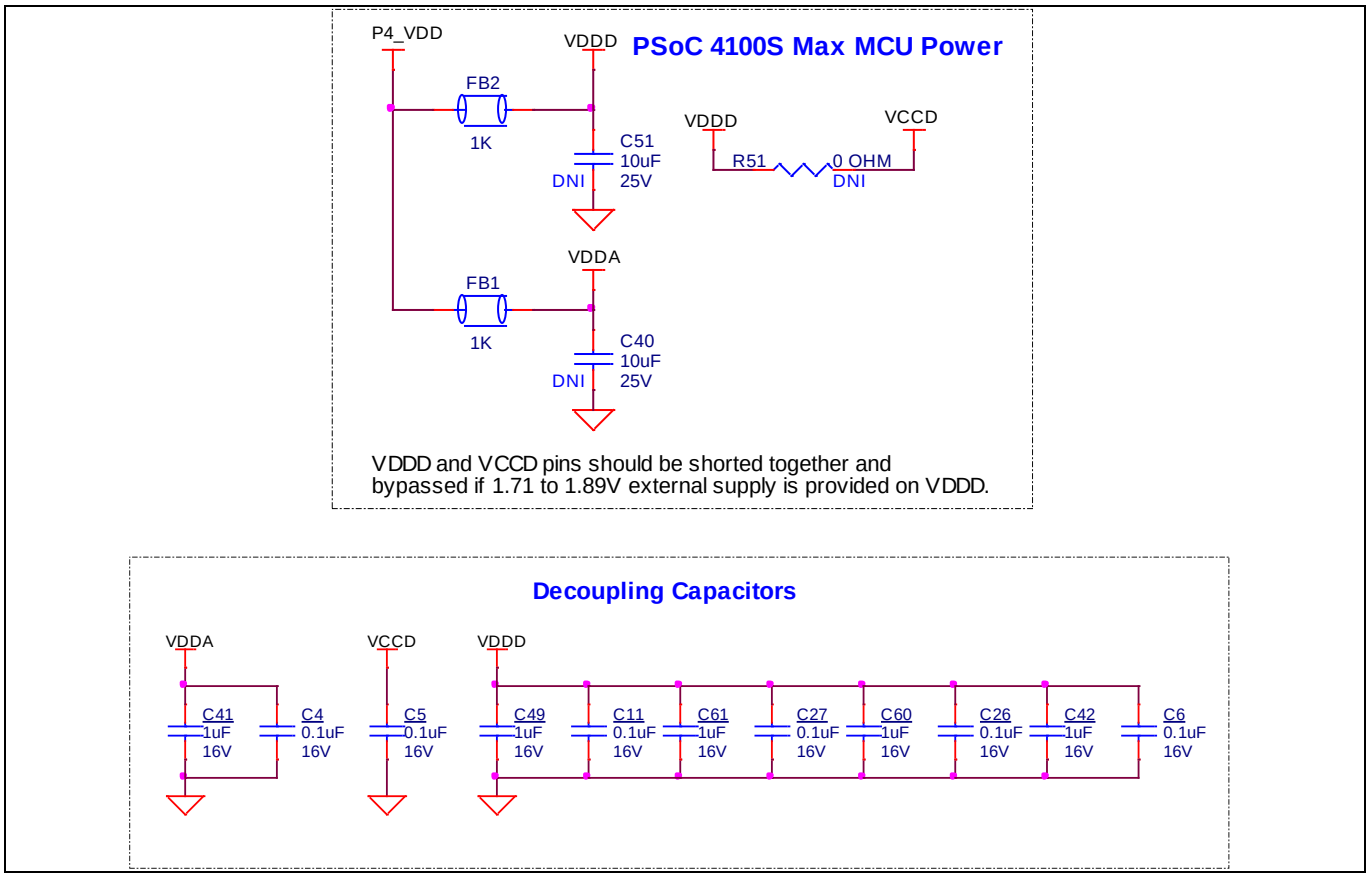
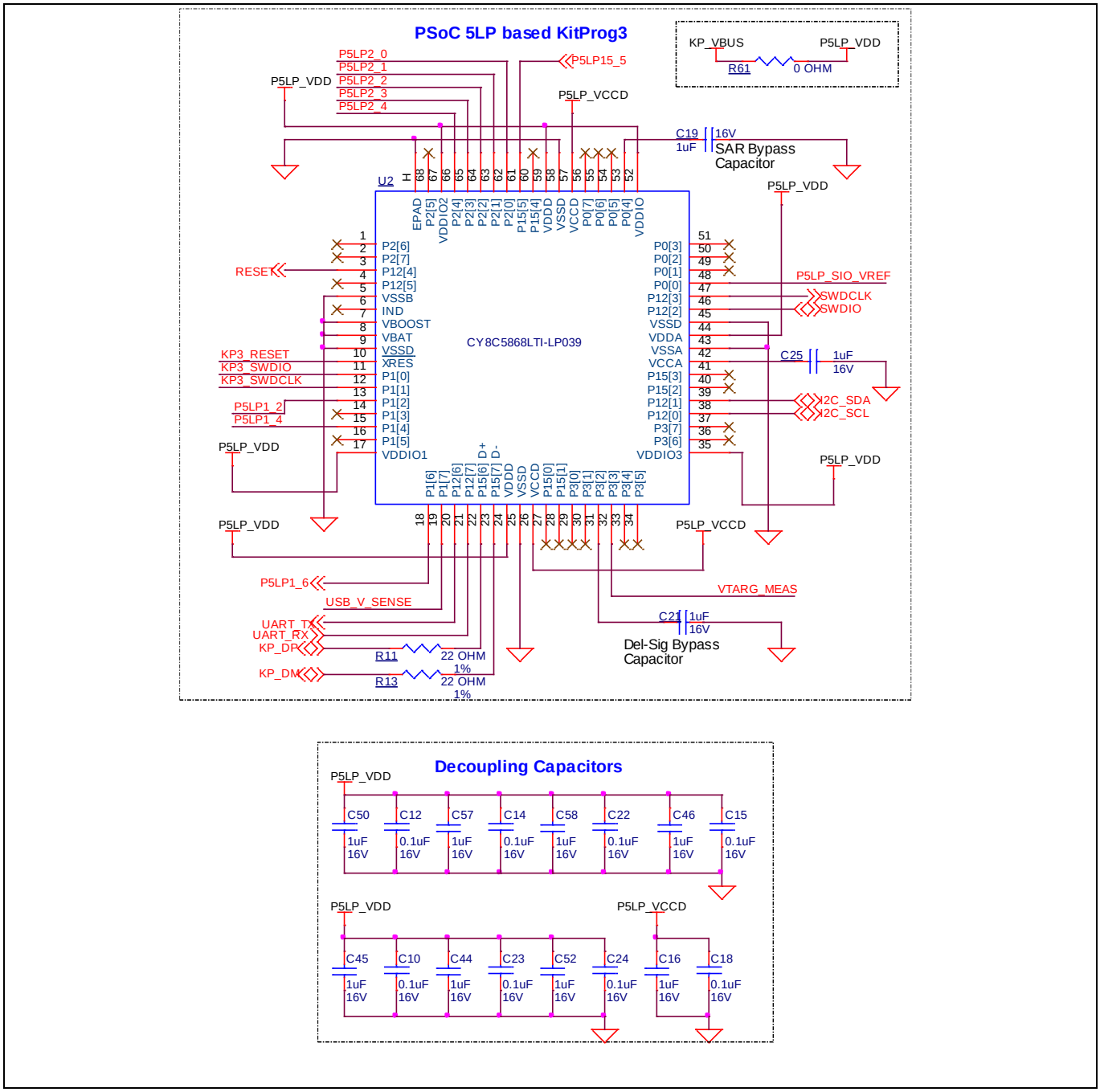



**Figure 16 Schematics of PSoC™ 4100S Max MCU device**

## 3.2.2 PSoC™ 5LP-based KitProg3 program and debug

An onboard PSoC™ 5LP (CY8C5868LTI-LP039) device is used as a KitProg3 to program and debug PSoC™ 4100S Max 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™ 4100S Max 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**.



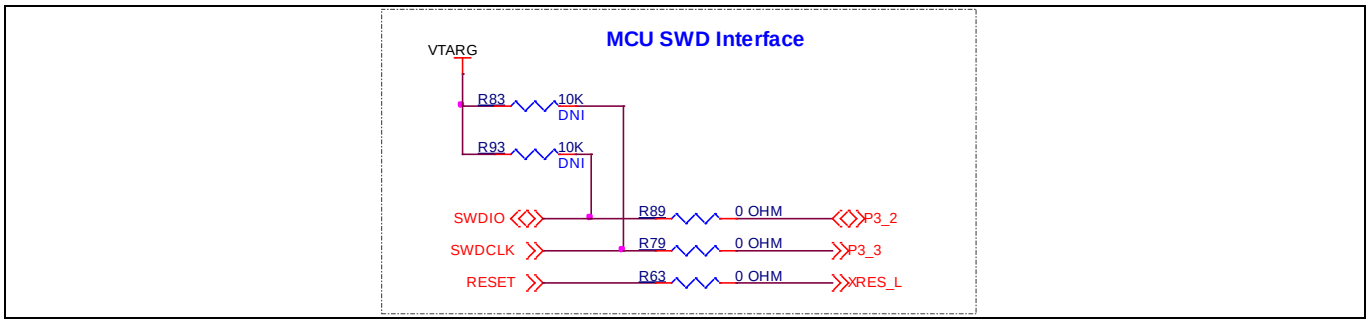



**Figure 17 Schematics of PSoC™ 5LP-based KitProg3**

### 3.2.2.1 Serial interface between KitProg3 and PSoC™ 4100S Max MCU

In addition to the use as an onboard programmer/debugger, the PSoC™ 5LP device functions as an interface for the USB-UART and USB-I2C bridges, as shown in **Figure 18**. The USB-Serial pins of the PSoC™ 5LP device are hard-wired to the I2C/UART pins of PSoC™ 4100S Max MCU. I2C pins are also available on the Arduino-compatible I/O headers.

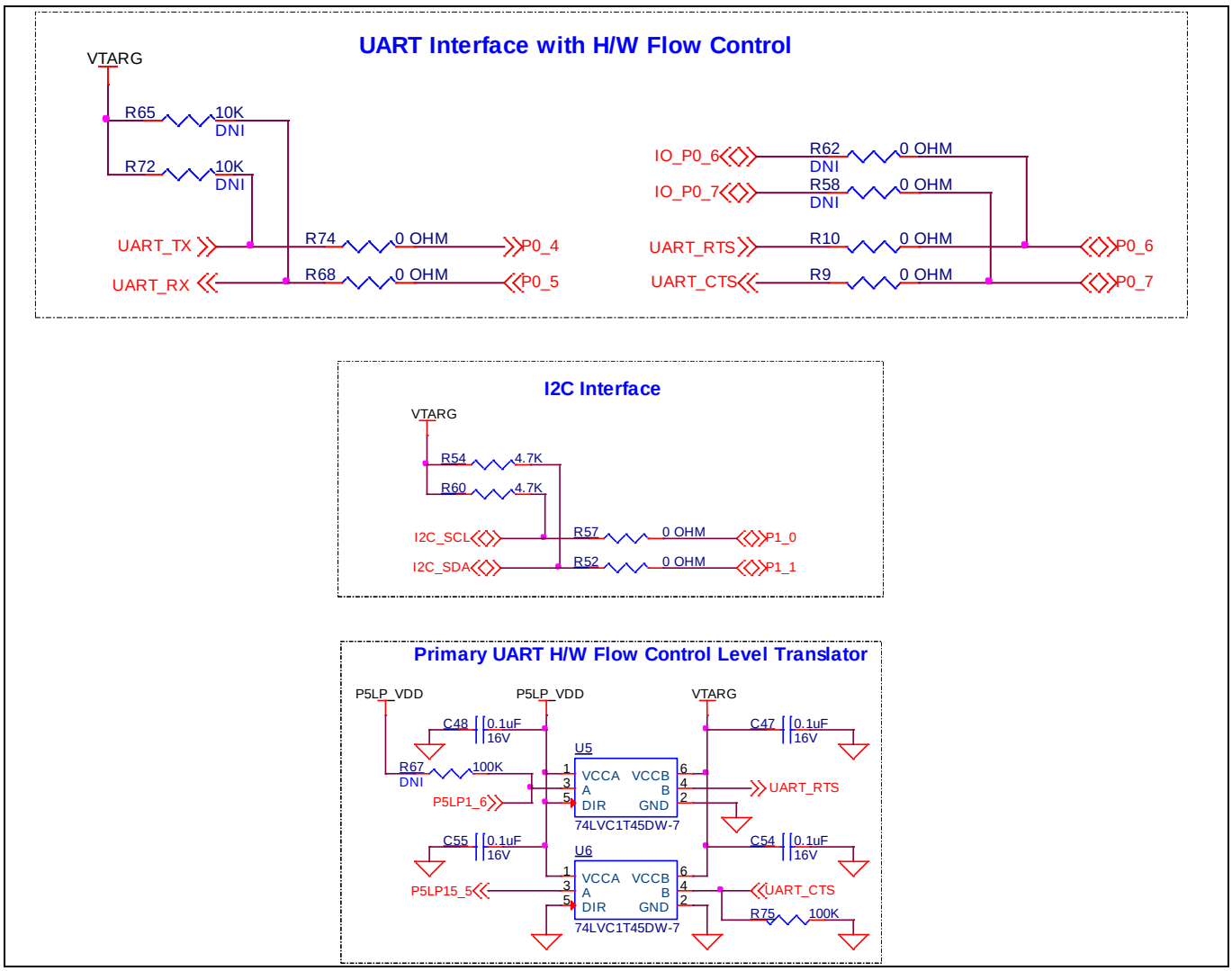



**Figure 18 Schematics of serial interface between KitProg3 and PSoC™ 4100S Max MCU**

### 3.2.3 PSoC™ 4100S Max MCU SWD 10-pin programming header

This 10-pin header allows you to program and debug the PSoC™ 4100S Max MCU using an external programmer such as **MiniProg4**.

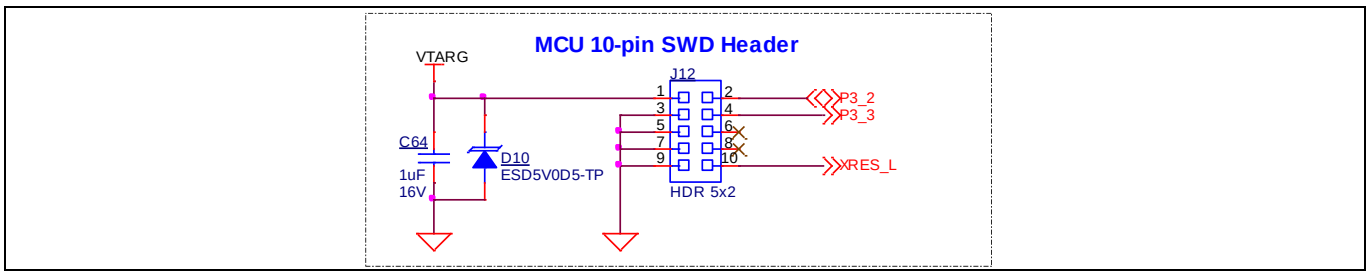



**Figure 19 Schematics of programming header**

## 3.2.4 Power supply system

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

- 5 V from the KitProg3 USB Micro-B connector (J8)
- 7 V to 12 V from VIN header or Arduino header (J5)
- 5 V from the PSoC™ 4100S Max MCU SWD 10-pin programming header (J12)

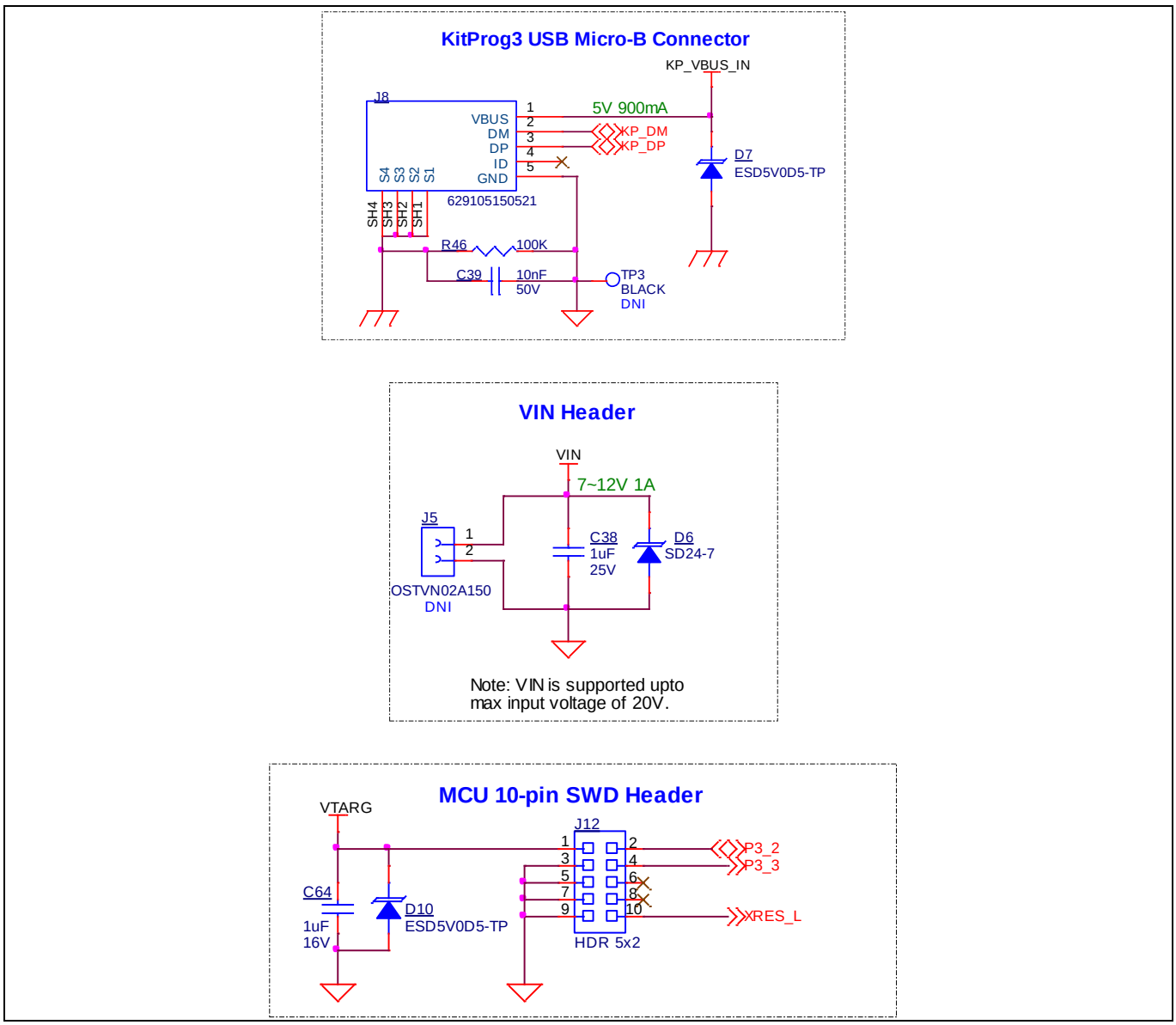



**Figure 20 Schematics of power supply input system**

### 3.2.4.1 Voltage regulators

The different output voltages required are 5 V or 3.3 V for PSoC™ 4100S Max MCU, user LEDs, thermistor, and expansion header and 5 V for KitProg3 (PSoC™ 5LP), power LED, and regulator. The onboard voltage regulator is designed to generate 3.3 V, required for onboard peripherals, when powered through the KitProg3 USB Micro-B connector or VIN; while 5 V is driven directly from the KitProg3 USB Micro-B connector. Suggested output current for the 3.3 V regulator is 1.5 A while powered from VIN. Current sourced from USB depends on the generation of the USB port to which the board is plugged into; as a rule of thumb USB 2.0 can source 500 mA, while USB 3.0 can source 900 mA.

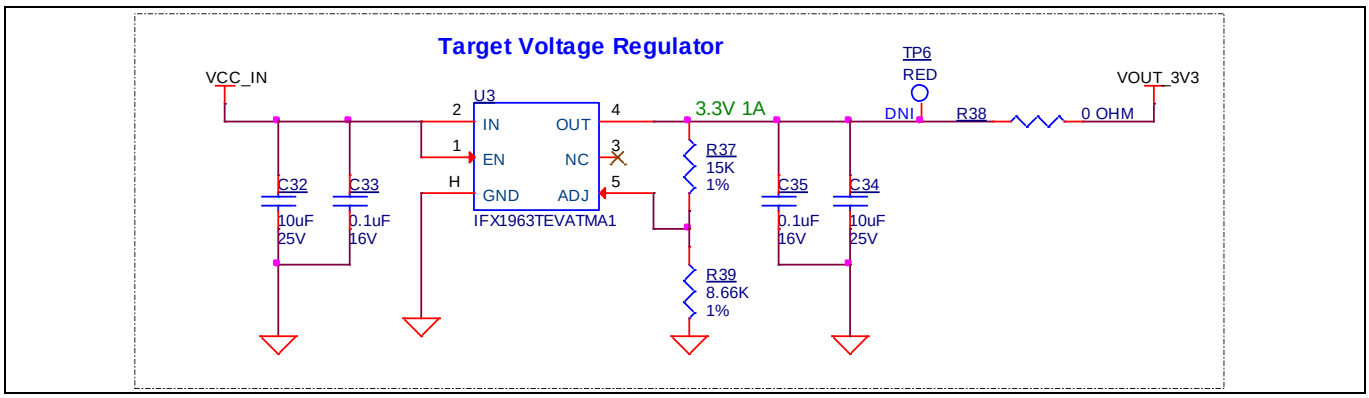



**Figure 21 Schematics of voltage regulator**

### 3.2.4.2 Voltage selection

A voltage selection jumper is used to select the power for the target device with two voltage levels, 5 V direct or 3.3 V from the regulator. By default, the power for target device is set to 5 V.

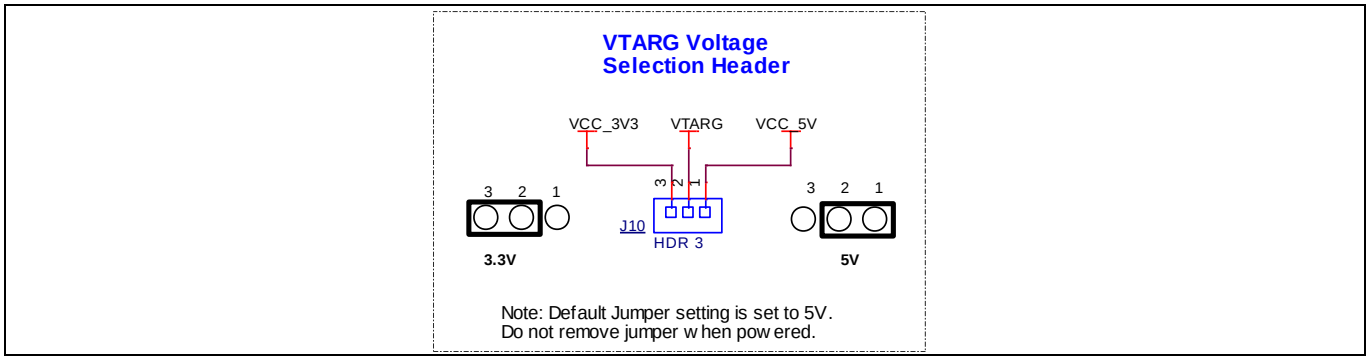



**Figure 22 Schematics of voltage selection jumper**

### 3.2.4.3 Current measurement header

The current for PSoC™ 4100S Max MCU has a dedicated two-pin header to facilitate easy current measurement using an ammeter across the pins. Ensure that either the jumper or ammeter is present before the kit is powered.

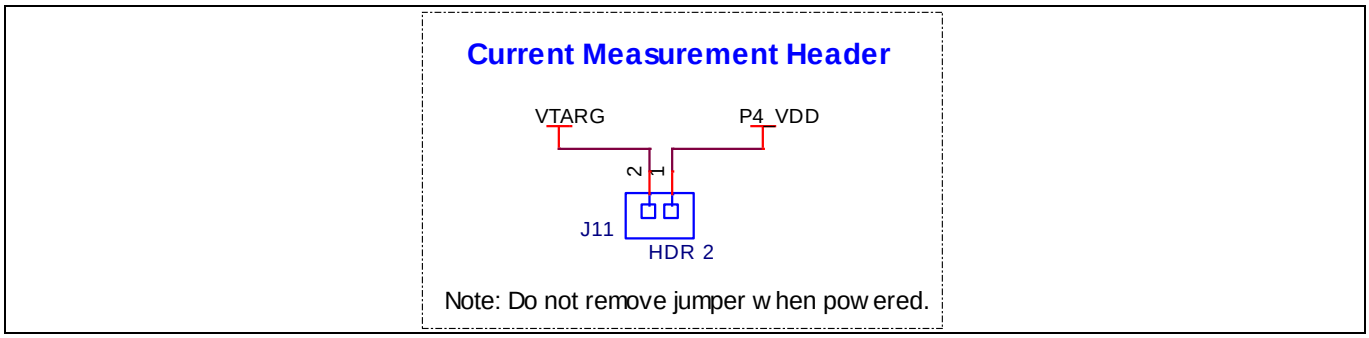
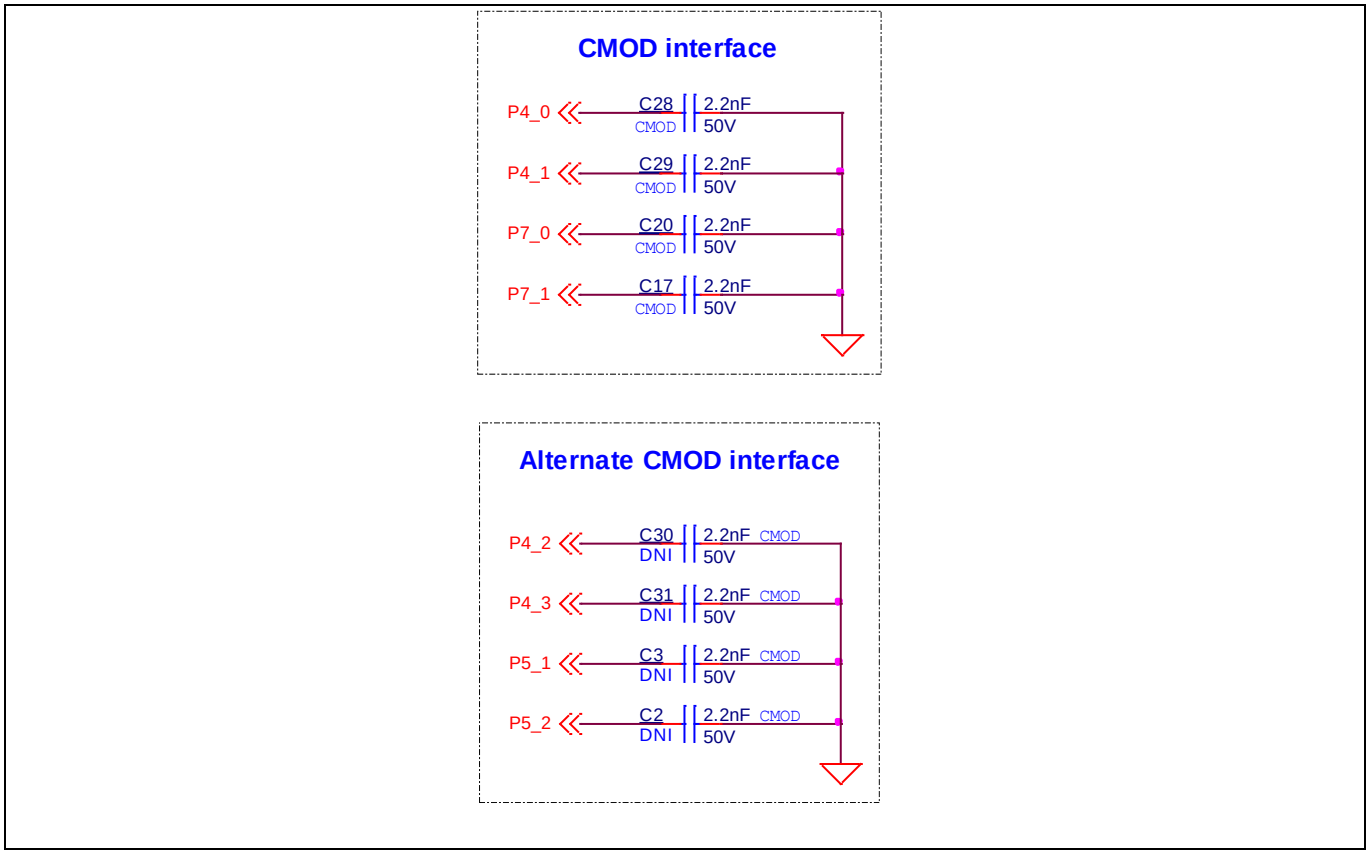



**Figure 23 Schematics of current measurement jumper**

### 3.2.5 CAPSENSE™

Two CAPSENSE™ buttons, supporting both self-capacitance (CSD) and mutual-capacitance (CSX) sensing are connected to PSoC™ 4100S Max MCU. Four external modulation capacitors (CMOD) are present on the pioneer board to enable CAPSENSE™ functionality. The board supports driven shield that can drive the hatch pattern surrounding the sensor region; by default these hatch patterns are connected to ground.

For details on using CAPSENSE™ including design guidelines, see the **PSoC™ 4 and PSoC™ 6 MCU CAPSENSE™ Design Guide**.



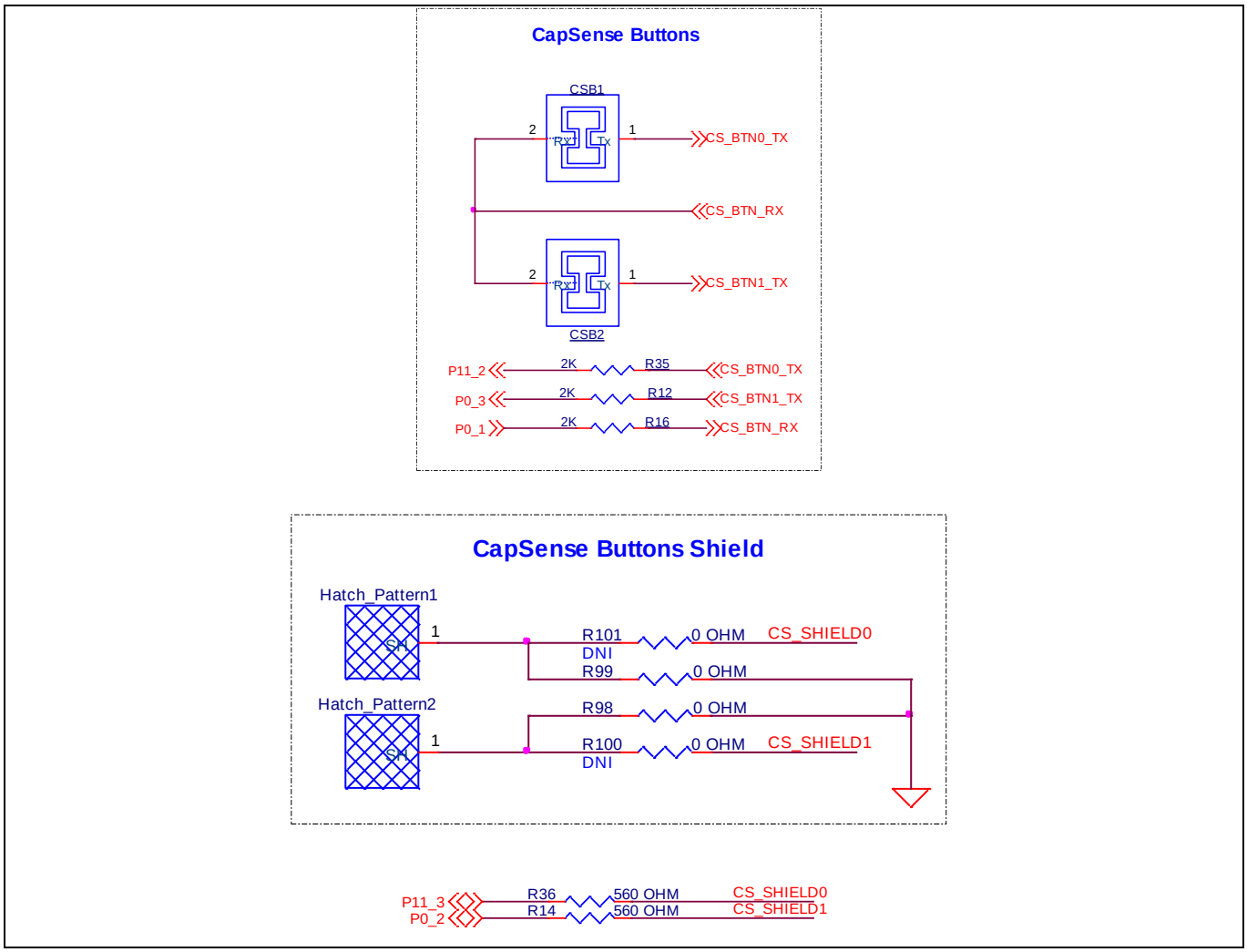



**Figure 24 Schematics of CAPSENSE™ sections**

Additional CAPSENSE™-compatible pins are routed to the FFC connector for expansion boards. See section **3.2.10.3 FFC connector for expansion board** for more details on this connector and pin mapping.

## 3.2.6 LEDs

D3 LED (yellow) indicates the status of KitProg3 (see the **KitProg3 User Guide** for details). D2 LED (yellow) indicates the status of the power supplied to the board. This power LED will not light up when powered from an external programmer. The board also has three user-controllable LEDs (LED1, LED2, and LED3) connected to PSoC™ 4100S Max MCU pins for user applications. LED2 and LED3 (orange) are placed near CAPSENSE™ buttons and are typically used in the CAPSENSE™ code example to demonstrate CAPSENSE™ buttons status. These two LEDs are connected to pins P6[4] and P12[0] of PSoC™ 4100S Max MCU. LED1 (red) is connected to P7[3]. All three user LEDs are active LOW.

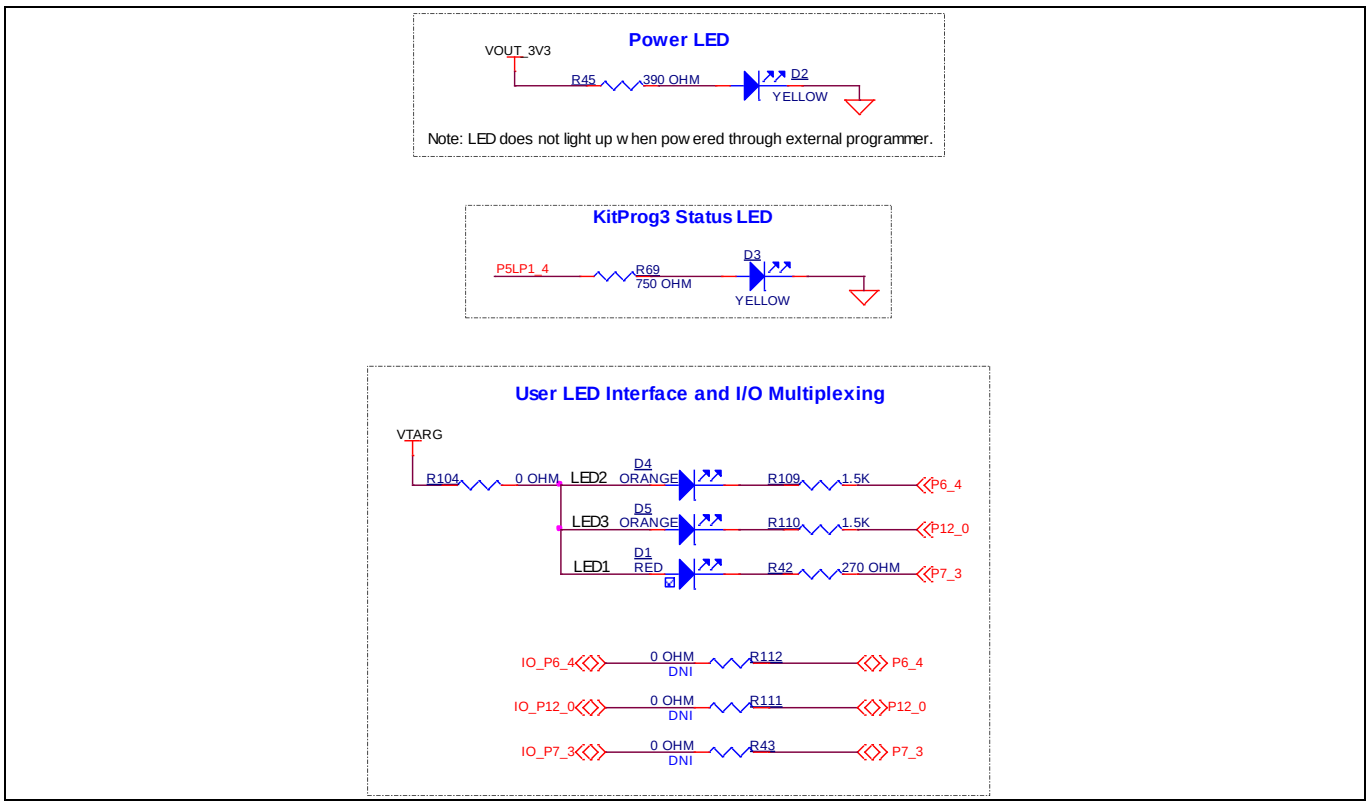



**Figure 25 Schematics of LEDs**

## 3.2.7 Push buttons

The board has a reset button, one user-controllable button, and a KitProg3 mode selection button. The reset button (SW1) is connected to the XRES pin of the PSoC™ 4100S Max MCU and is used to reset the device. A user button (SW2) is connected to pin P11[5] of the PSoC™ 4100S Max MCU. The Mode selection button (SW3) is connected to the PSoC™ 5LP device for programming mode selection (see the **KitProg3 User Guide** for details). All buttons are active LOW and short to GND when pressed.

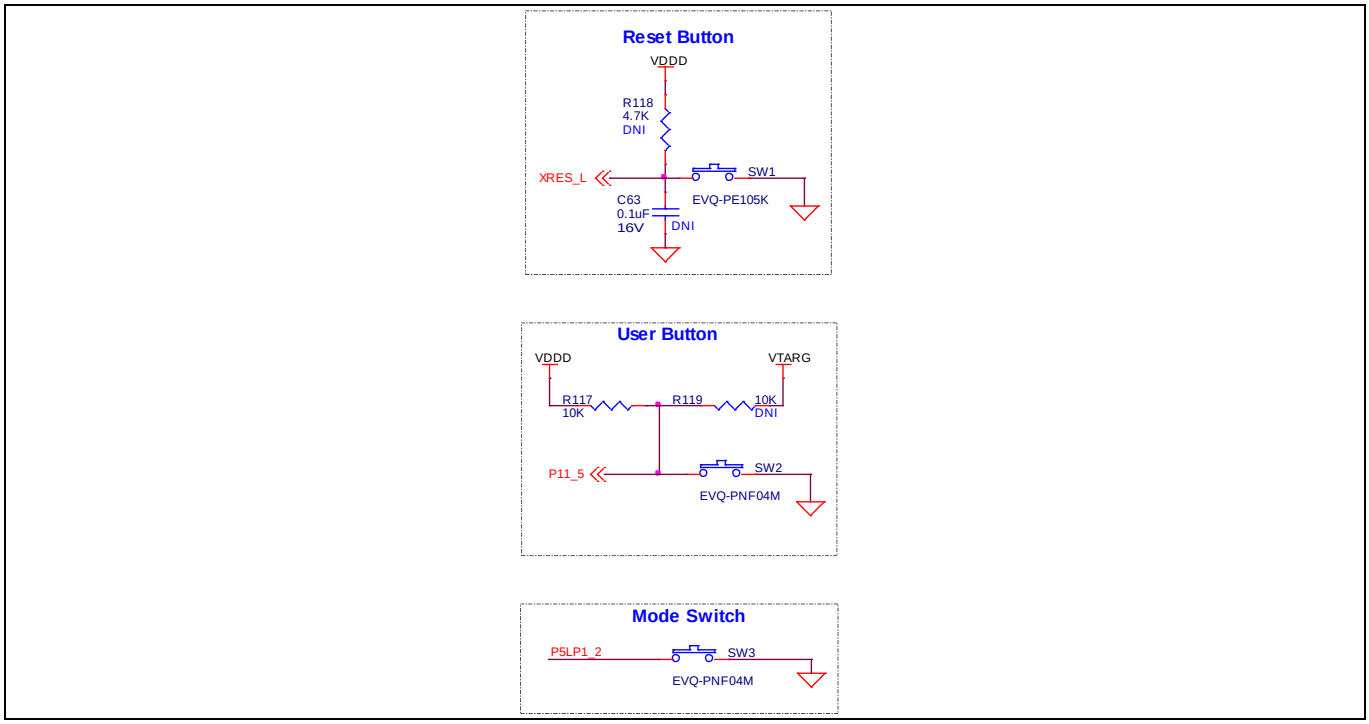



**Figure 26 Schematics of push buttons**

## 3.2.8 Crystal oscillators

A 24-MHz crystal oscillator footprint is provided onboard and connected to P0[6] and P0[7] of the PSoC™ 4100S Max MCU. These pins are, by default, connected to UART RTS and CTS. Remove these connections before using crystal oscillator.

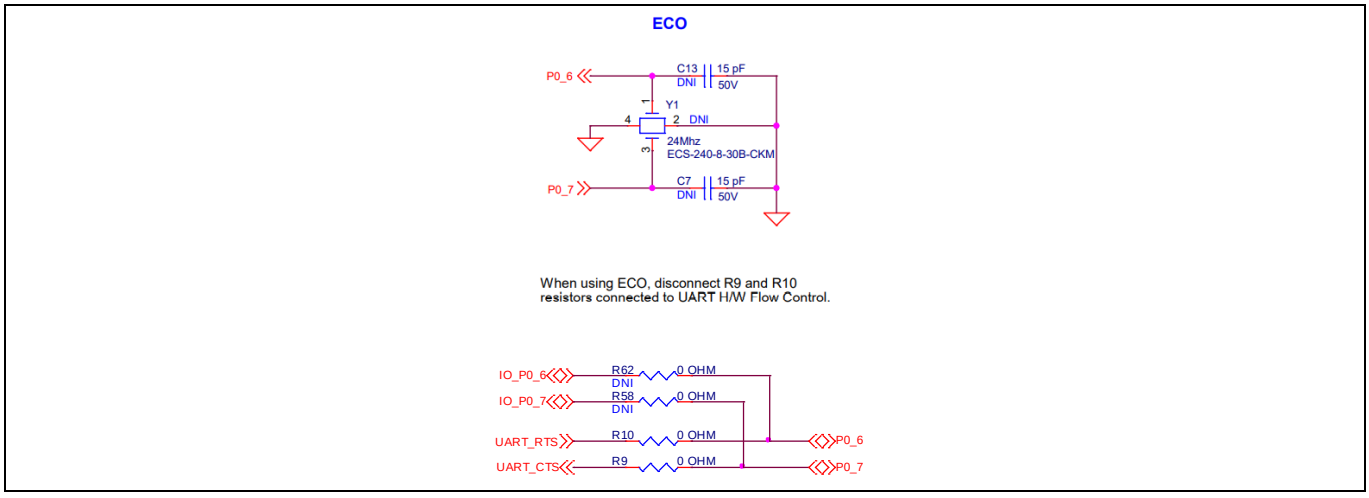



**Figure 27 Schematics of crystal oscillators**

## 3.2.9 Thermistor

A 10K negative temperature coefficient (NTC) thermistor is present onboard for ambient temperature measurement. By default, the thermistor is connected in single-ended mode to P2[6], and can be changed to ratio-metric differential mode by modifying the connections using zero-ohm resistors. The optional pins provided for ratio-metric mode are also connected to Arduino header (J2). Remove these resistors while connecting the thermistor in ratio-metric mode.

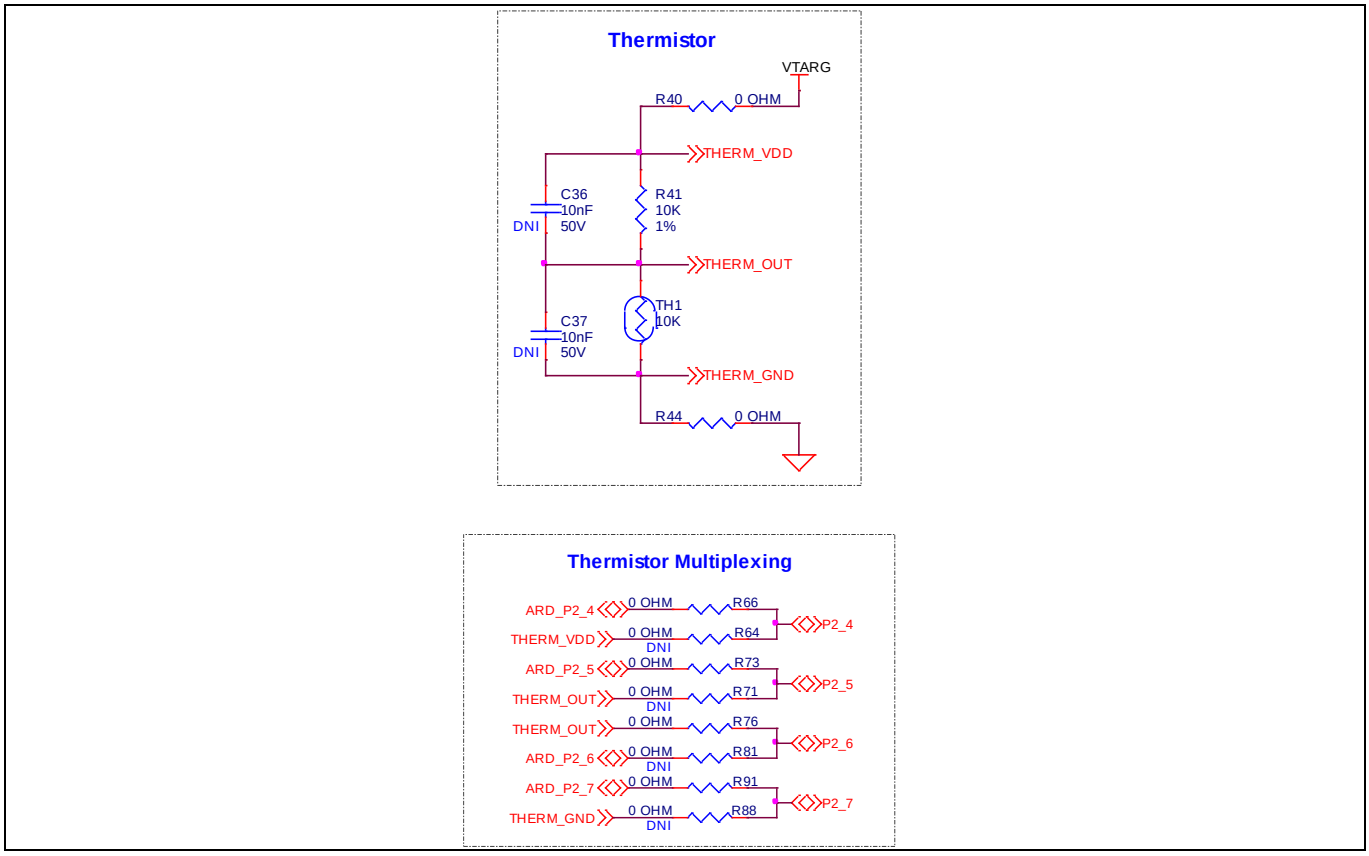



**Figure 28 Schematics of thermistor**

## 3.2.10 Expansion headers

I/Os of the PSoC™ 4100S Max MCU are connected to expansion headers on the pioneer board. I/Os are connected to three primary categories of connectors.

### 3.2.10.1 Arduino Uno compatible connectors

The board has four Arduino Uno compatible connectors: J1, J2, J3, and J4. You can connect Arduino Uno compatible shields to develop applications based on the shield's hardware.

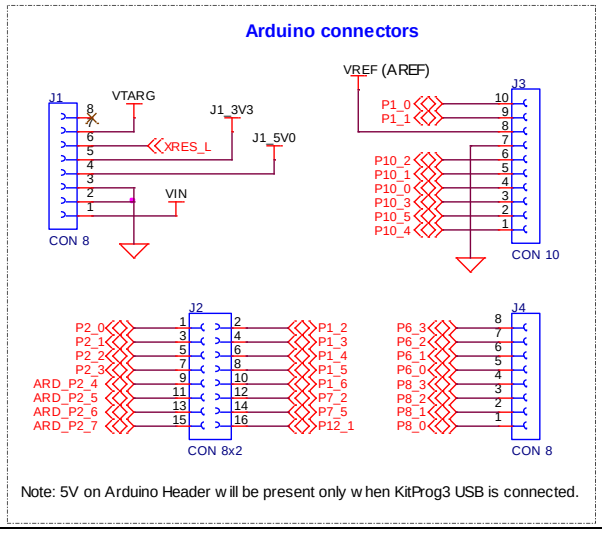



**Figure 29 Schematics of Arduino Uno compatible connectors**

#### 3.2.10.1.1 Arduino Uno compatible power connector

The J1 connector powers Arduino shields. The header also has a provision to power the pioneer board though the VIN input on the connector.

**Table 4 Pinout Arduino Uno compatible power connector**

| Pin # | Arduino pin | Pioneer board function |
|---|---|---|
| J1.1 | VIN | 7~12 V i/o |
| J1.2 | GND | Ground |
| J1.3 | GND | Ground |
| J1.4 | 5 V | 5 V o/p |
| J1.5 | 3.3 V | 3.3 V o/p |
| J1.6 | RESET | PSoC™ 4100S Max MCU reset |
| J1.7 | IOREF | IOREF o/p |
| J1.8 | N/C | N/C |

### 3.2.10.1.2 Arduino Uno compatible I/O connectors

The pioneer board is compatible with and operates flawlessly with third-party Arduino shields. The connectors J2, J3, and J4 bring out Arduino Uno-compatible I/Os of the PSoC™ 4100S Max MCU.

**Table 5 Pinout Arduino Uno compatible I/O connectors**

| Pin # | Arduino pin | Pioneer board function | PSoC™ 4100S Max MCU pin |
|---|---|---|---|
| J2.1 | A0 | ADC0 | P2[0] |
| J2.2 | N/C | N/C | P1[2] |
| J2.3 | A1 | ADC1 | P2[1] |
| J2.4 | N/C | N/C | P1[3] |
| J2.5 | A2 | ADC2 | P2[2] |
| J2.6 | N/C | N/C | P1[4] |
| J2.7 | A3 | ADC3 | P2[3] |
| J2.8 | N/C | N/C | P1[5] |
| J2.9 | A4 | ADC4 | P2[4] |
| J2.10 | N/C | N/C | P1[6] |
| J2.11 | A5 | ADC5 | P2[5] |
| J2.12 | N/C | N/C | P7[2] |
| J2.13 | N/C | N/C | P2[6] |
| J2.14 | N/C | N/C | P7[5] |
| J2.15 | N/C | N/C | P2[7] |
| J2.16 | N/C | N/C | P12[1] |
| | | | |
| J3.1 | D8 | DIGITAL I/O | P10[4] |
| J3.2 | D9 | PWM / DIGITAL I/O | P10[5] |
| J3.3 | D10 | SS / PWM / DIGITAL I/O | P10[3] |
| J3.4 | D11 | MOSI / PWM / DIGITAL I/O | P10[0] |
| J3.5 | D12 | MISO / DIGITAL I/O | P10[1] |
| J3.6 | D13 | SCK / DIGITAL I/O | P10[2] |
| J3.7 | GND | Ground | GND |
| J3.8 | AREF | Analog ref | P1[7] |
| J3.9 | SDA | SDA | P1[1] |
| J3.10 | SCL | SCL | P1[0] |
| | | | |
| J4.1 | D0 | RX | P8[0] |
| J4.2 | D1 | TX | P8[1] |
| J4.3 | D2 | DIGITAL I/O | P8[2] |
| J4.4 | D3 | PWM / DIGITAL I/O | P8[3] |
| J4.5 | D4 | DIGITAL I/O | P6[0] |
| J4.6 | D5 | PWM / DIGITAL I/O | P6[1] |

| Pin # | Arduino pin | Pioneer board function | PSoC™ 4100S Max MCU pin |
|---|---|---|---|
| J4.7 | D6 | PWM / DIGITAL I/O | P6[2] |
| J4.8 | D7 | DIGITAL I/O | P6[3] |

### 3.2.10.2 Additional GPIO expansion connector

Additional unused I/Os and a few shared I/Os of the PSoC™ 4100S Max MCU are brought out to this connector. This connector is not populated by default. Five shared I/Os are not connected to these connector pads, by default. Zero-ohm resistor configuration needs to be changed to connect those I/Os to this expansion connector.

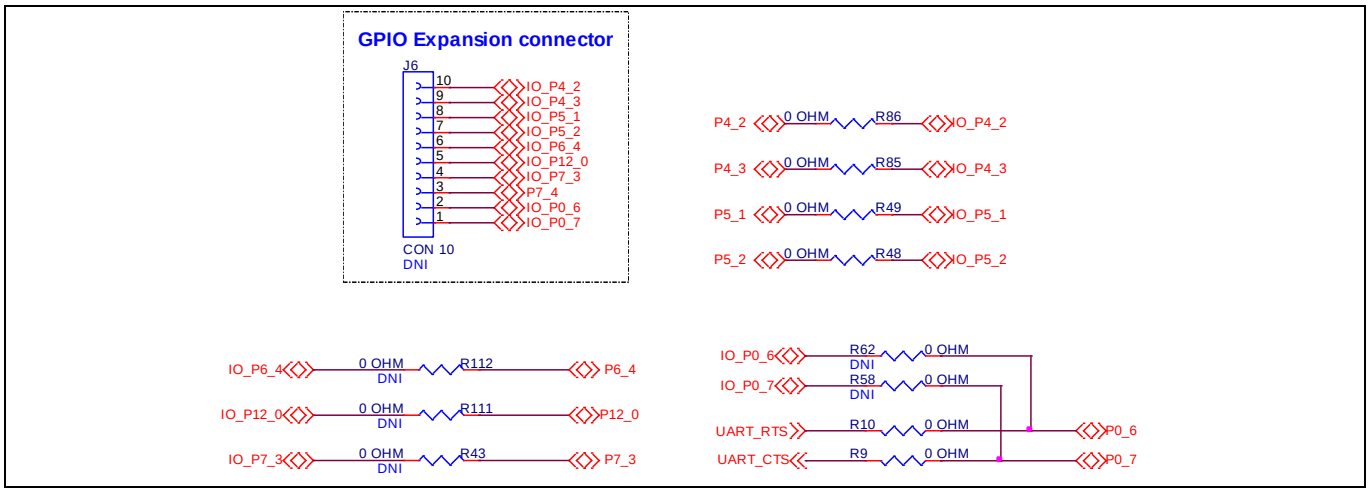



**Figure 30 Schematics of additional GPIO expansion connector**

**Table 6 Pinout additional GPIO expansion connector**

| Pin # | PSoC™ 4100S Max MCU pin | Primary on-board function | Secondary on-board function | Secondary on-board function |
|---|---|---|---|---|
| J6.1 | P0[7] | UART_CTS | GPIO on J6 header | Remove R10 and mount R62 to connect to J6 header |
| J6.2 | P0[6] | UART_RTS | GPIO on J6 header | Remove R9 and mount R58 to connect to J6 header |
| J6.3 | P7[4] | GPIO on J6 header | – | – |
| J6.4 | P7[3] | Red User LED (D1) | GPIO on J6 header | Remove R42 and mount R43 to connect to extended header |
| J6.5 | P12[0] | Orange user LED (D5) | GPIO on J6 header | Remove R110 and mount R111 to connect to extended header |
| J6.6 | P6[4] | Orange User LED (D4) | GPIO on J6 header | Remove R109 and mount R112 to connect to J6 header |
| J6.7 | P5[2] | GPIO on J6 header | Alternate CMOD | Remove R48 and mount 2.2 nF capacitor on C2 |
| J6.8 | P5[1] | GPIO on J6 header | Alternate CMOD | Remove R49 and mount 2.2 nF capacitor on C3 |

| Pin # | PSoC™ 4100S Max MCU pin | Primary on-board function | Secondary on-board function | Secondary on-board function |
|---|---|---|---|---|
| J6.9 | P4[3] | GPIO on J6 header | Alternate CMOD | Remove R85 and mount 2.2 nF capacitor on C31 |
| J6.10 | P4[2] | GPIO on J6 header | Alternate CMOD | Remove R86 and mount 2.2 nF capacitor on C30 |

### 3.2.10.3 FFC connector for expansion board

A 40-pin bottom-contact flat flex cable (FFC) connector is provided, on the pioneer board, to bring out the CAPSENSE™ pins of PSoC™ 4100S Max MCU. This connector is used to connect the expansion board to the pioneer board. This connector brings out 29 I/Os of PSoC™ 4100S Max MCU. Refer **3.5 Frequently asked questions** section for information on precautions that needs to be taken while connecting and disconnecting the flex cable.

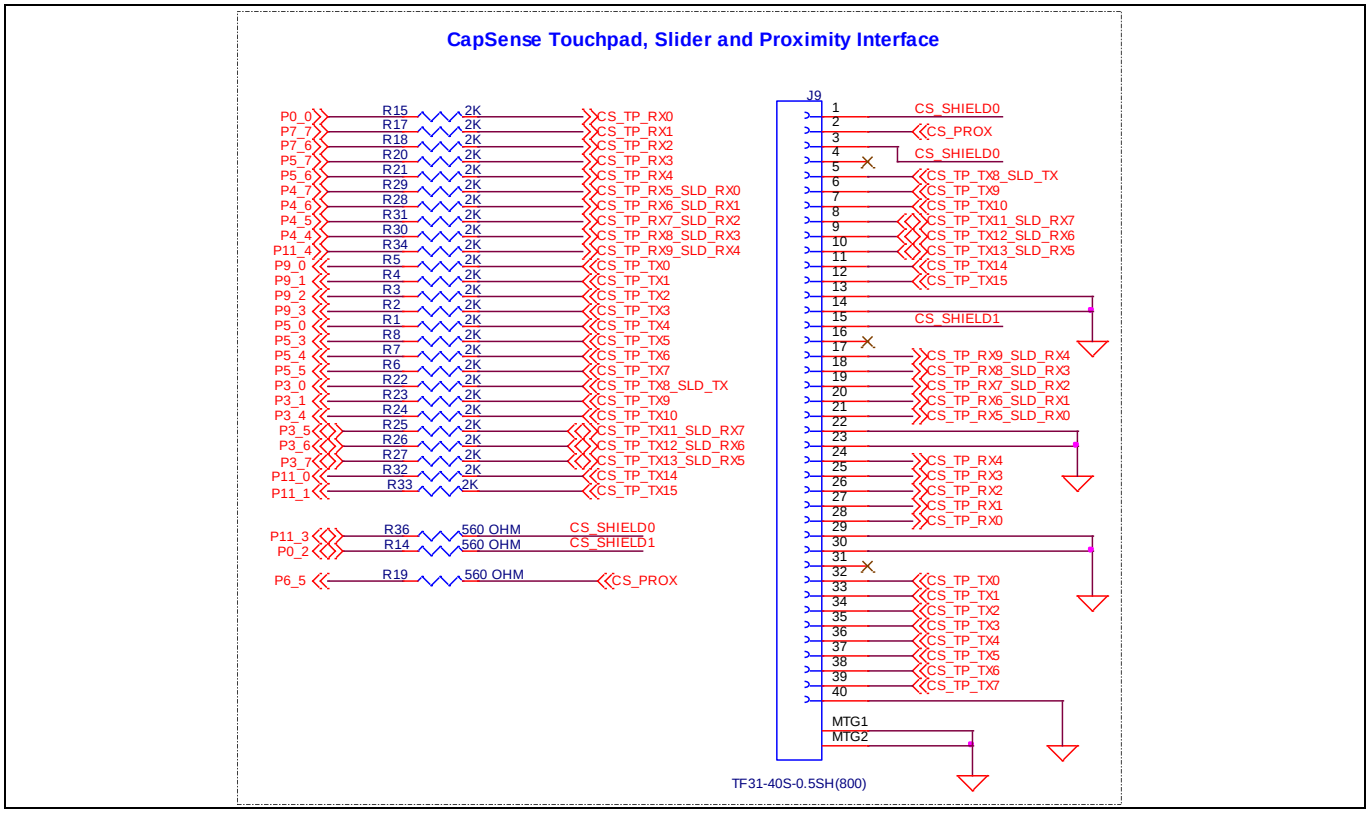



**Figure 31 Schematics of FFC connector for expansion board**

## 3.2.11 CY8CKIT-041S-MAX capacitive sensing expansion board

The expansion board connects to the pioneer board using the 40-pin FFC connector. The expansion board has two FFC connectors that connect the I/Os of PSoC™ 4100S Max MCU to either the Touchpad and proximity section or to the Slider section.

### 3.2.11.1 Touchpad and proximity interface

The Expansion board consists of a 160-segment Touchpad that has 10 rows and 16 columns. The board also has a proximity loop with ~11 cm diagonal length that can demonstrate a proximity range of about 18 cm distance.

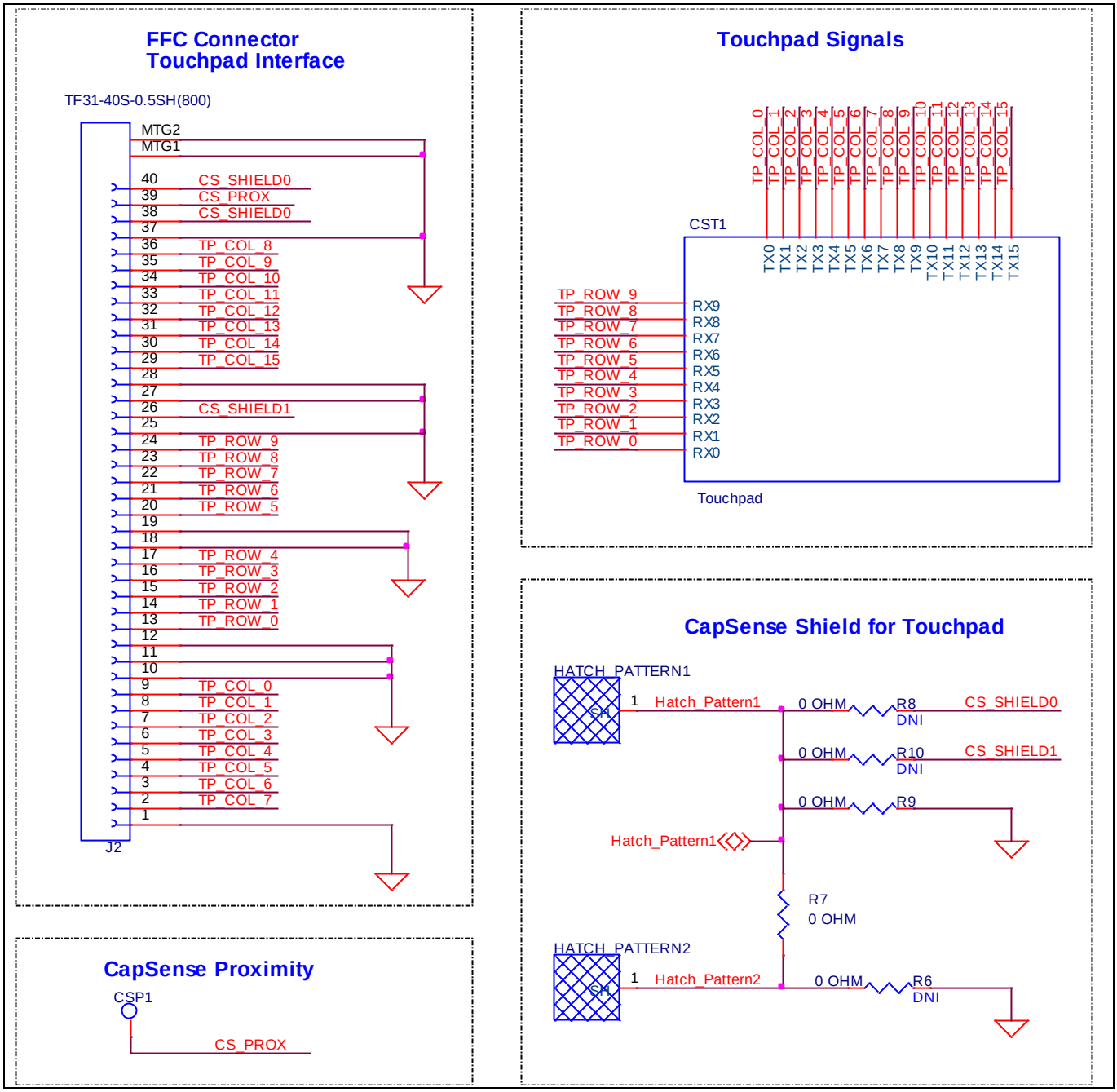



**Figure 32 Schematics of Touchpad and proximity section**

**Table 7 Touchpad and proximity interface**

| FFC pin # | Function | PSoC™ 4100S Max MCU pin |
|---|---|---|
| J2.1, J2.10, J2.11, J2.12, J2.18, J2.19, J2.25, J2.27, J2.28, J2.37 | GND | GND |
| J2.2 | TP.COL.7 | P5[5] |
| J2.3 | TP.COL.6 | P5[4] |
| J2.4 | TP.COL.5 | P5[3] |
| J2.5 | TP.COL.4 | P5[0] |
| J2.6 | TP.COL.3 | P9[3] |
| J2.7 | TP.COL.2 | P9[2] |
| J2.8 | TP.COL.1 | P9[1] |
| J2.9 | TP.COL.0 | P9[0] |
| J2.13 | TP.ROW.0 | P0[0] |
| J2.14 | TP.ROW.1 | P7[7] |
| J2.15 | TP.ROW.2 | P7[6] |
| J2.16 | TP.ROW.3 | P5[7] |
| J2.17 | TP.ROW.4 | P5[6] |
| J2.20 | TP.ROW.5 | P4[7] |
| J2.21 | TP.ROW.6 | P4[6] |
| J2.22 | TP.ROW.7 | P4[5] |
| J2.23 | TP.ROW.8 | P4[4] |
| J2.24 | TP.ROW.9 | P11[4] |
| J2.26 | SHIELD1 | P0[2] |
| J2.29 | TP.COL.15 | P11[1] |
| J2.30 | TP.COL.14 | P11[0] |
| J2.31 | TP.COL.13 | P3[7] |
| J2.32 | TP.COL.12 | P3[6] |
| J2.33 | TP.COL.11 | P3[5] |
| J2.34 | TP.COL.10 | P3[4] |
| J2.35 | TP.COL.9 | P3[1] |
| J2.36 | TP.COL.8 | P3[0] |
| J2.38, J2.40 | SHIELD0 | P11[3] |
| J2.39 | PROX | P6[5] |

### 3.2.11.2 Slider interface

The Expansion board consists of an 8-segment slider that supports both CSD and CSX sensing.

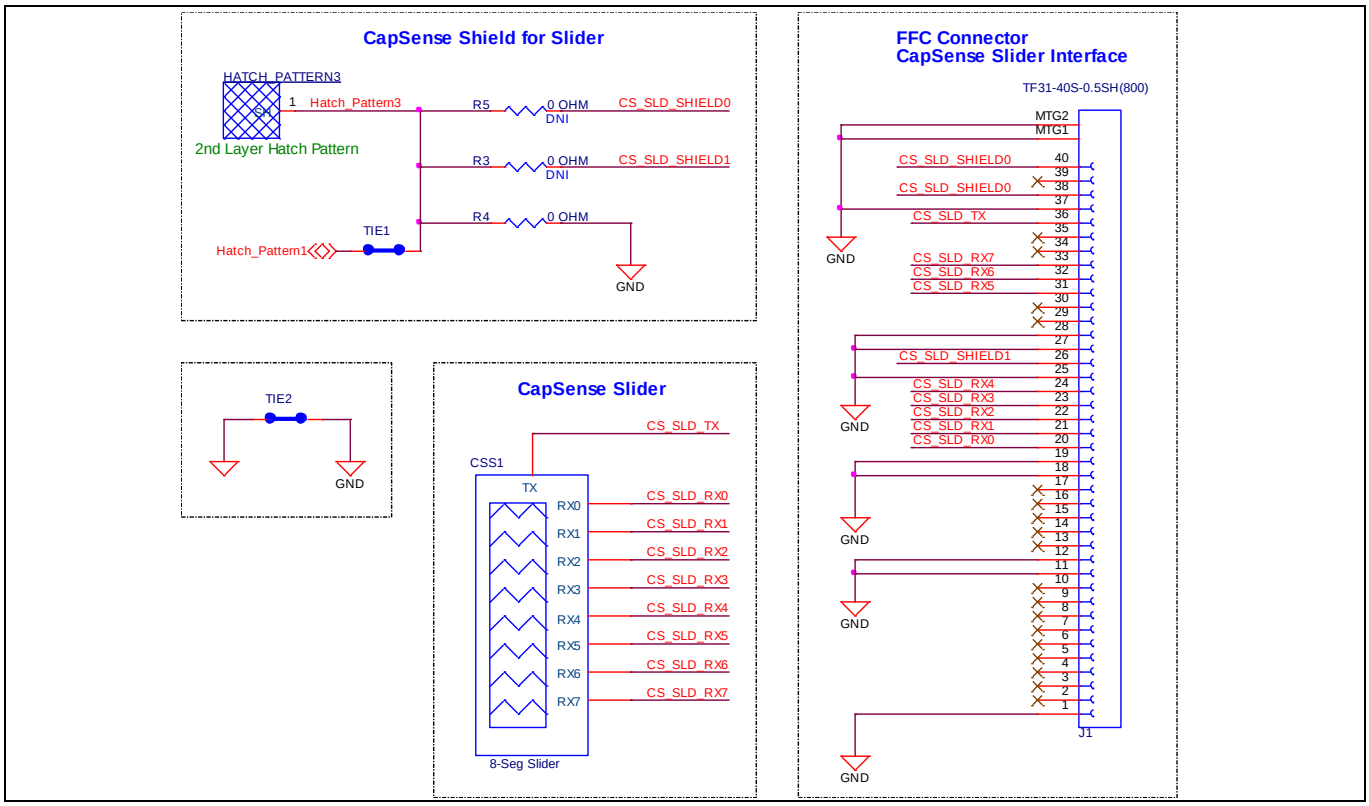



**Figure 33 Schematics of slider interface section**

**Table 8 Slider interface**

| FFC pin # | Function | PSoC™ 4100S Max MCU pin |
|---|---|---|
| J1.1, J1.11, J1.12, J1.18, J1.19, J1.25, J1.27, J1.28, J1.37 | GND | GND |
| J1.2–J1.10, J1.13–J1.17, J1.29, J1.30, J1.34, J1.35, J1.39 | No connect (NC) | NC |
| J1.20 | SLD.RX0 | P4[7] |
| J1.21 | SLD.RX1 | P4[6] |
| J1.22 | SLD.RX2 | P4[5] |
| J1.23 | SLD.RX3 | P4[4] |
| J1.24 | SLD.RX4 | P11[4] |
| J1.26 | SHIELD1 | P0[2] |
| J1.31 | SLD.RX5 | P3[7] |
| J1.32 | SLD.RX6 | P3[6] |
| J1.33 | SLD.RX7 | P3[5] |
| J1.36 | SLD.TX | P3[0] |
| J1.38, J1.40 | SHIELD0 | P11[3] |

### 3.2.11.3 Hatch pattern configuration

The board supports driven shield that can drive the hatch pattern surrounding the touchpad proximity region and slider region. By default, hatch patterns are connected to ground. A shield signal driven hatch pattern provides performance improvements for CSD (self-capacitance) sensing by reducing the paracitic capacitance and provides liquid tolerance.

Hatch_Pattern1 – Hatch pattern which surrounds the touchpad in top layer and below the touchpad in bottom layer. Driving this with shield signal provides performance improvements for CSD touchpad and provides liquid tolerance.

Hatch_Pattern2 - Hatch pattern which surrounds the proximity in top layer. Driving Hatch_Pattern1 and Hatch_Pattern2 together with shield signal provides better detection range for the proximity loop. And grounding Hatch_Pattern2 gives an option to improve EMC.

Hatch_Pattern3 - Hatch pattern which surrounds the slider in top layer and below the slider in bottom layer. Driving this with shield signal provides performance improvements for CSD slider and provides liquid tolerance.

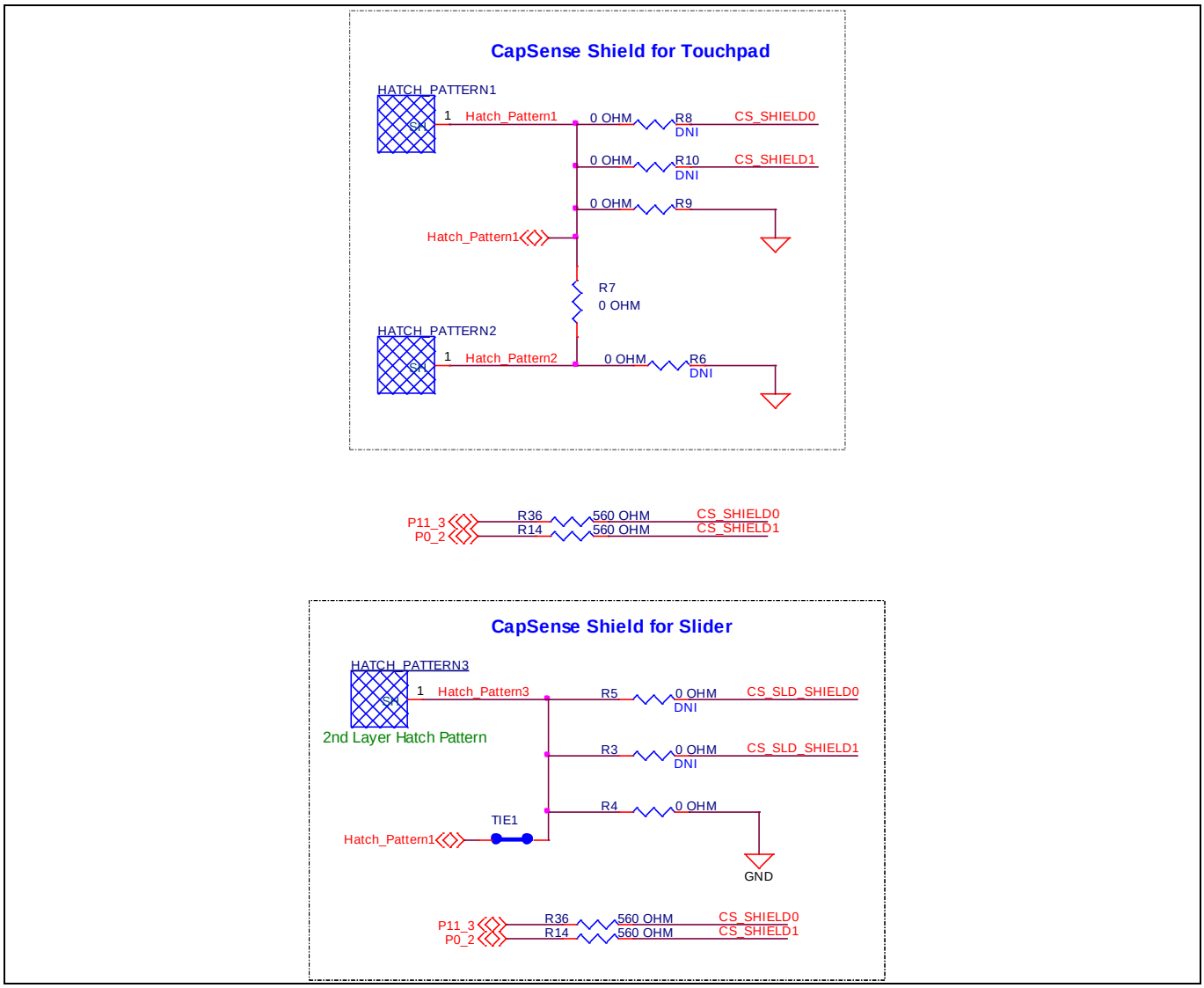



**Figure 34 Schematics of hatch pattern configuration**

## 3.3 CY8CKIT-041S-MAX Kit rework

This section explains modifications that can be made to the board to evaluate different use cases that are multiplexed and not available out of box.

1. Rework to add on board ECO for the PSoC™ 4100S Max MCU.
   - By default, the ECO is not populated on the pioneer board. Additionally, the I/Os for the ECO are used for UART hardware flow control of the USB-UART bridge.
   - While using an ECO, remove R10, and R9 and load Y1, C13, and C7

**Table 9 Rework (to add on board ECO)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| Y1 | CRYSTAL, 24.0000MHZ, 8PF, SMD | ECS Inc | ECS-240-8-30B-CKM |
| C7, C13 | CAP, CER, 15pF, 5%, NP0, 50V, 0402 | Samsung | CL05C150JB5NNNC |

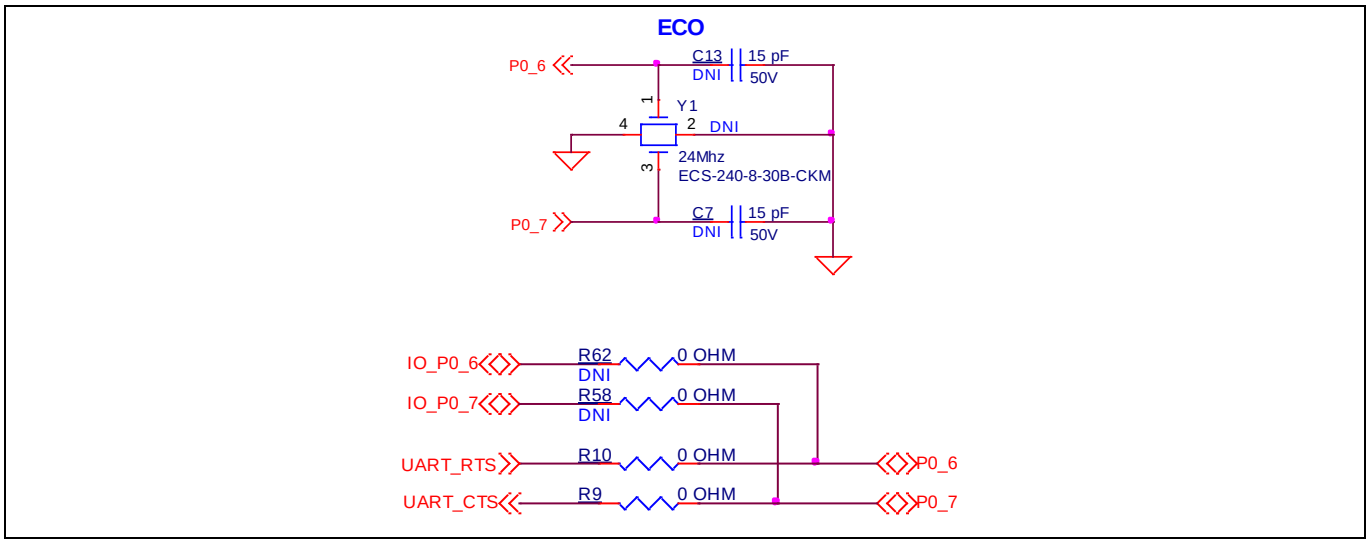



**Figure 35 Schematics of ECO rework**

2. Rework to change the hatch pattern configuration for the CAPSENSE™ buttons on the pioneer board.
   - By default, hatch patterns are connected to ground.
   - Remove R98 and R99 and load R100 and R101, to connect the hatch patterns to a driven shield signal. These hatch patterns are placed around CAPSENSE™ buttons (CSB1 and CSB2) and driving with a shield signal improves performance for CSD (self-capacitance) buttons.

**Table 10 Rework (to change hatch pattern configuration)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| R98, R99, R100, R101 | RES, 0 OHM, JUMPER, 1A, 0603 | Yageo | RC0603JR-070RL |

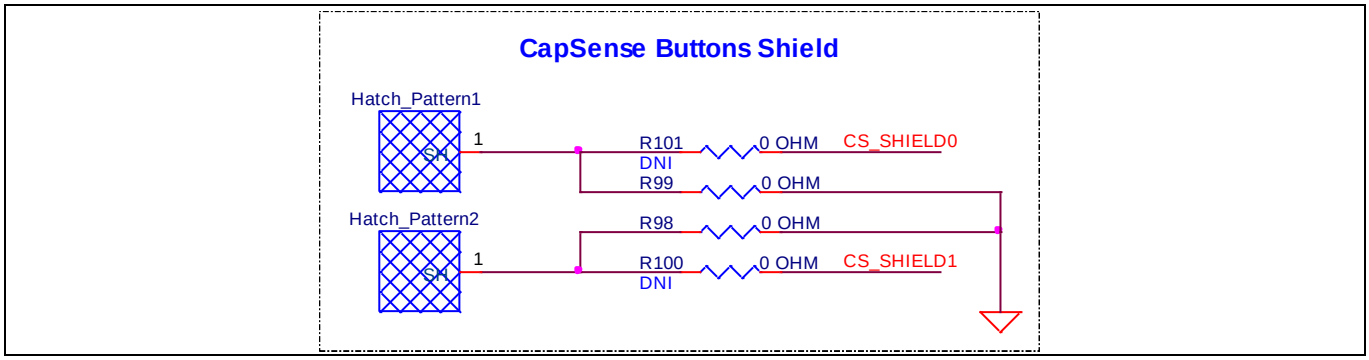



**Figure 36 Schematics of pioneer board hatch pattern rework**

## Hardware

3. Rework to change the hatch pattern configuration for the Touchpad on the expansion board.
   - By default, hatch patterns are connected to ground.
   - Remove R9 and load R8 to connect the hatch pattern to a shield signal generated by CAPSENSE™ MSC block channel 0 (MSC0). Or load R10 to connect the hatch pattern to a shield signal generated by CAPSENSE™ MSC block channel 1 (MSC1). Both MSC channels (if both are enabled) are capable of driving the shield for electrodes scanned from the other channel.
   - A shield driven hatch pattern provides performance improvements for the CSD (self-capacitance) Touchpad.
   - There is also an option to ground Hatch_Pattern2 to improve EMC, by removing R7 and loading R6.

**Table 11 Rework (to change hatch pattern configuration)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| R3, R4, R5, R6, R7, R8, R9, R10 | RES, 0 OHM, JUMPER, 1A, 0603 | Yageo | RC0603JR-070RL |

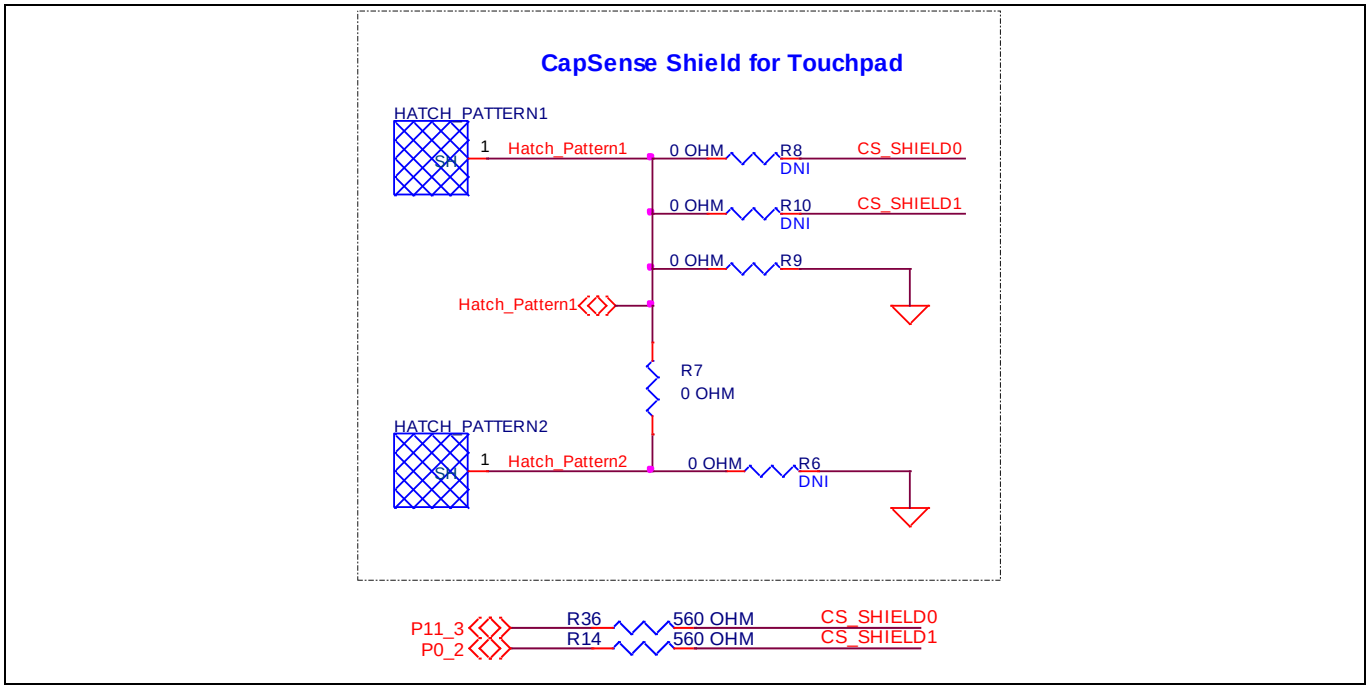



**Figure 37 Schematics of expansion board hatch pattern rework for Touchpad**

**Table 12 Touchpad interface**

| Touchpad hatch pattern configurations | Reference |
|---|---|
| Hatch_Pattern1 ↔ SHIELD 0 | R8 |
| Hatch_Pattern1 ↔ SHIELD 1 | R10 |
| Hatch_Pattern1 ↔ GND | R9 (default) |
| Hatch_Pattern1 ↔ Hatch_Pattern2 | R7 (default) |
| Hatch_Pattern2 ↔ GND | R6 |

4. Rework to change the hatch pattern configuration for the slider on the expansion board.
   - By default, hatch patterns are connected to ground.
   - Remove R4 and load R5 to connect the hatch pattern to a shield signal generated by CAPSENSE™ MSC block channel 0 (MSC0).
   - Remove R4 and load R3, to connect the hatch pattern to a shield signal generated by CAPSENSE™ MSC block channel 1 (MSC1).
   - A shield driven hatch pattern provides performance improvements for the CSD (self-capacitance) Slider.

**Table 13 Rework (to change hatch pattern configuration)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| R3, R4, R5, R6, R7, R8, R9, R10 | RES, 0 OHM, JUMPER, 1A, 0603 | Yageo | RC0603JR-070RL |

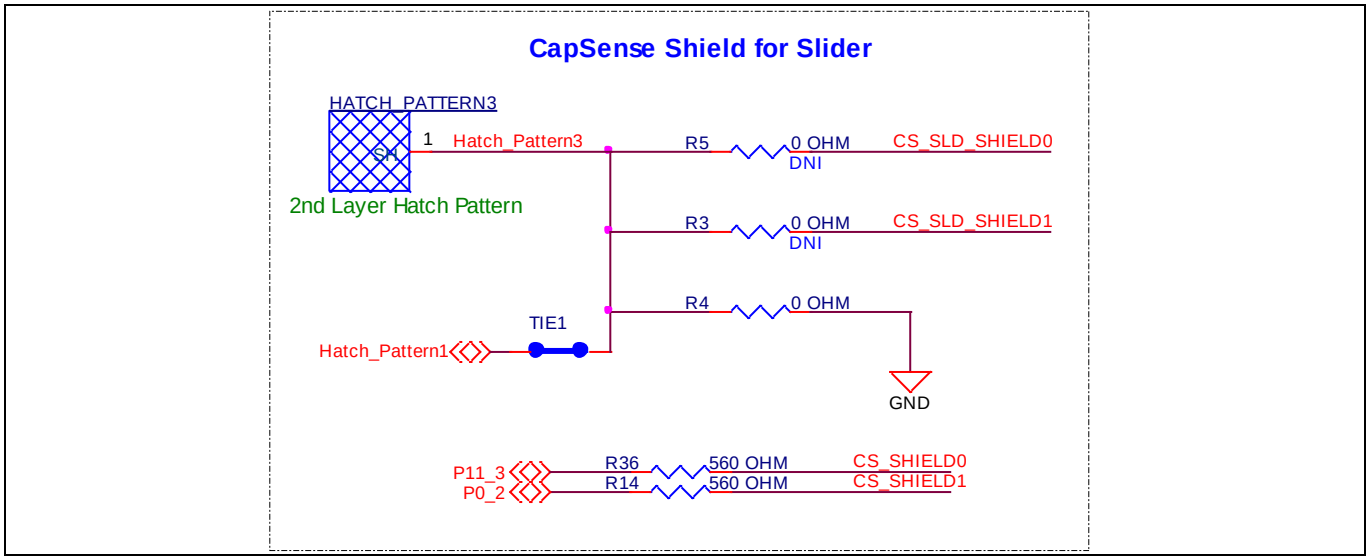



**Figure 38 Schematics of expansion board hatch pattern rework for slider**

**Table 14 Slider interface**

| Slider hatch pattern configurations | Reference |
|---|---|
| Hatch_Pattern3 ↔ SHIELD 0 | R5 |
| Hatch_Pattern3 ↔ SHIELD 1 | R3 |
| Hatch_Pattern3 ↔ GND | R4 (default) |

5. Rework to change the thermistor from single-ended mode to ratio-metric differential mode.
   - By default, the thermistor is connected in single-ended mode to P2[6] and this can be changed to ratio-metric differential mode by modifying zero-ohm resistor connections. The optional pins provided for ratio-metric mode are also connected to Arduino header (J2). Remove these resistors while connecting the thermistor in ratio-metric mode.
   - Remove R66, R73, R91 and load same on R64, R71, and R88.

**Table 15 Rework (to change hatch pattern configuration)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| R64, R71, R88 | RES, 0 OHM, JUMPER, 1A, 0603 | Yageo | RC0603JR-070RL |

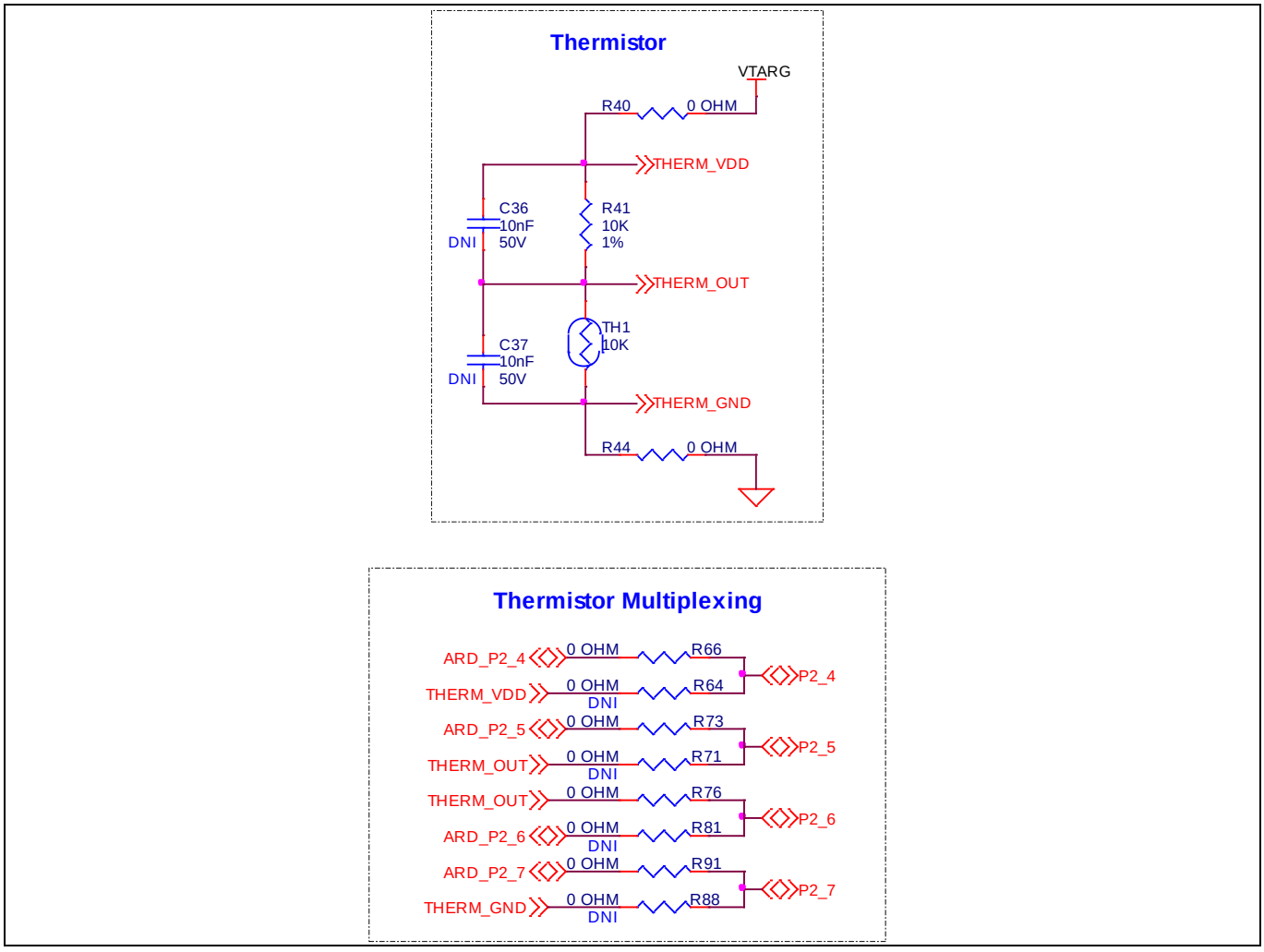



**Figure 39 Schematics of thermistor rework**

6. Rework to bring out I/Os connected to the LEDs of expansion headers.
   - PSoC™ 4100S Max MCU I/Os P6[4], P12[0], and P7[3] are, by default, connected to onboard LEDs. These can be brought out to the 10-pin GPIO expansion connector (no load by default), by changing the series resistor configuration.
   - Remove R42, R109 and R110 and load zero-ohm resistor on R43, R111 and R112.

**Table 16 Rework (to bring out I/Os)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| J6 | CONN, HDR, FEMALE, SINGLE, 10POS, 2.54 mm, GOLD, STR, TH | Sullins | PPPC101LFBN-RC |
| R43, R111, R112 | RES, 0 OHM, JUMPER, 1A, 0603 | Yageo | RC0603JR-070RL |

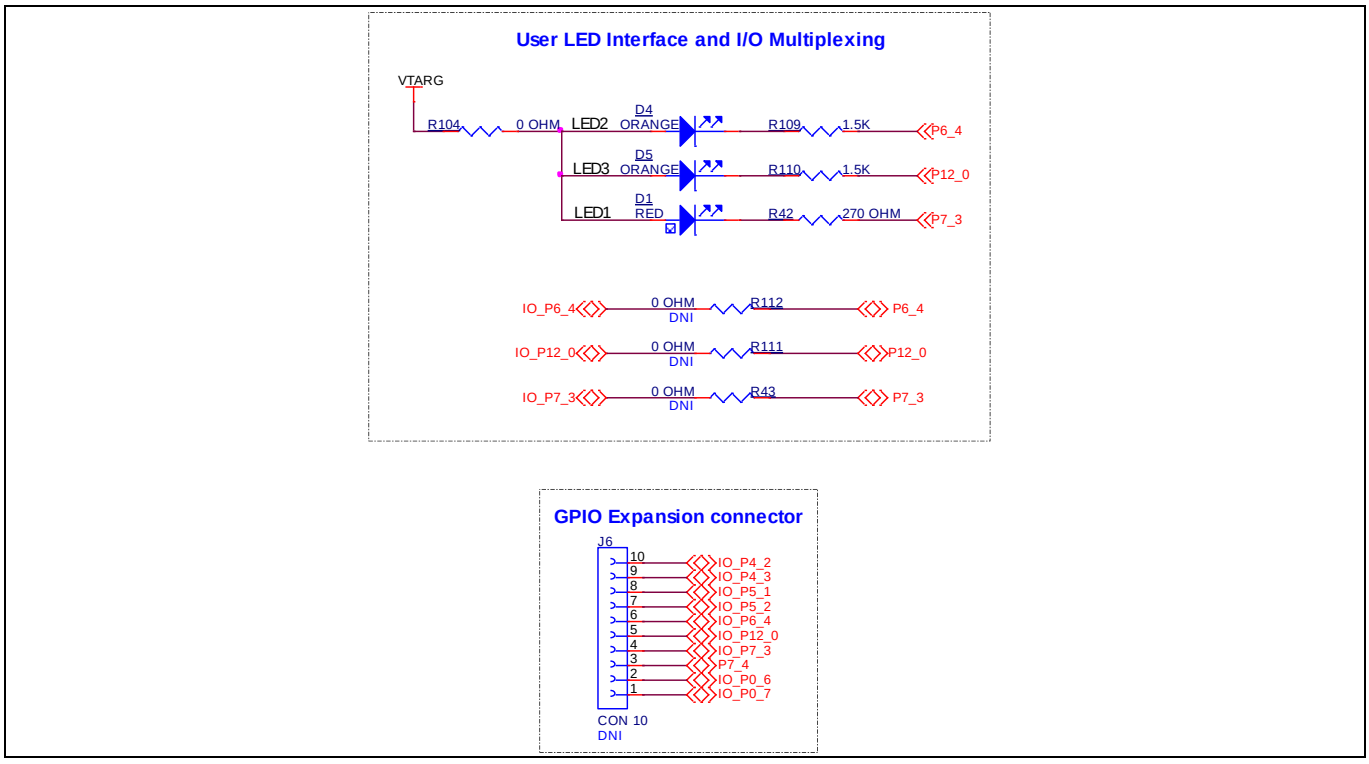



**Figure 40 Schematics of LED/GPIO expansion connector rework**

7. Rework to change CMOD connection to alternate pins.
   - By default, CMOD capacitors of CAPSENSE™ MSC block channel 0 (MSC0) are connected to P4.0 and P4.1.
   - Similarly, CMOD capacitors of CAPSENSE™ MSC block channel 1 (MSC1) are connected to P7.0, and P7.1.
   - The CMOD capacitors of MSC0 can alternatively be connected to P4.2 and P4.3.
   - The CMOD capacitors of MSC1 can alternatively be connected to P5.1 and P5.2.
   - Remove C17, C20 C28, and C29 and load the same capacitors on C2, C3, C30, and C31.

**Table 17 Rework (to change CMOD connection)**

| Reference | Description | Manufacturer | Mfr part # |
|---|---|---|---|
| C2, C3, C17, C20, C28, C29, C30, C31, | CAP, CER, 2.2 nF, 5%, NP0, 50 V, 0603 | Murata | GRM1885C1H222JA01D |

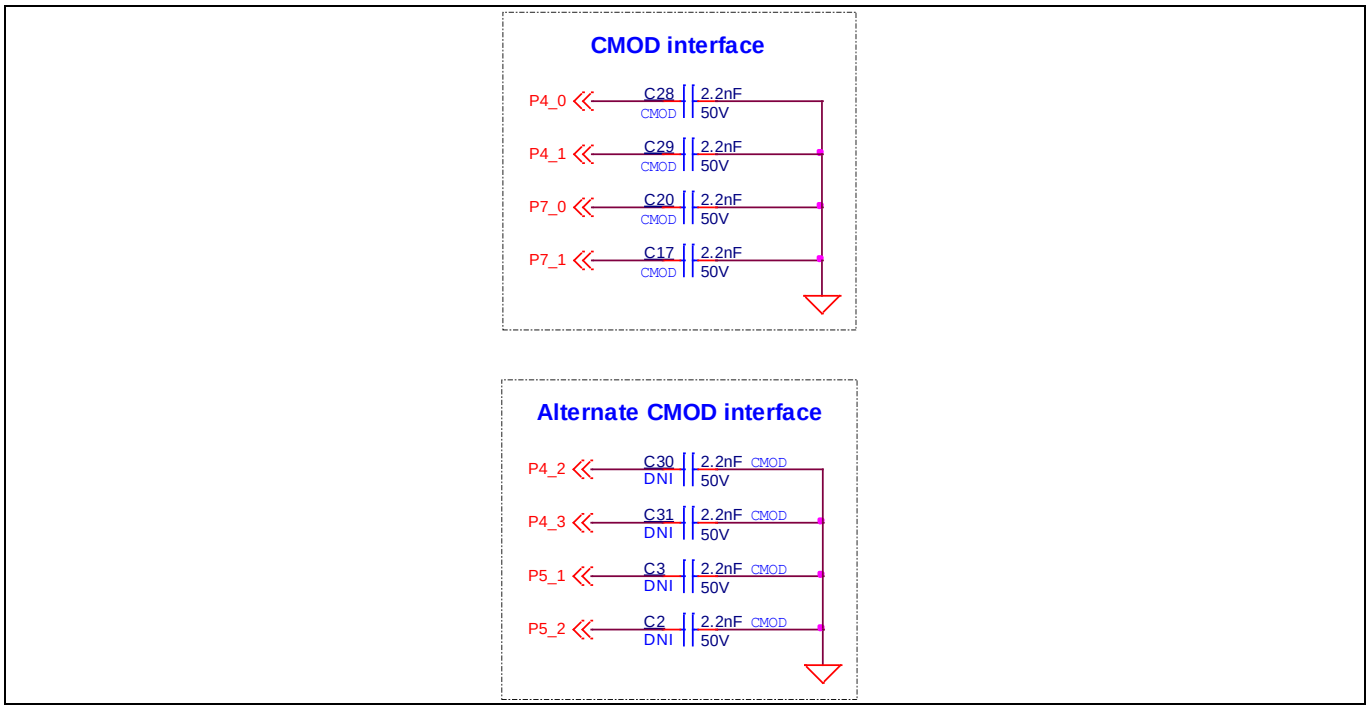


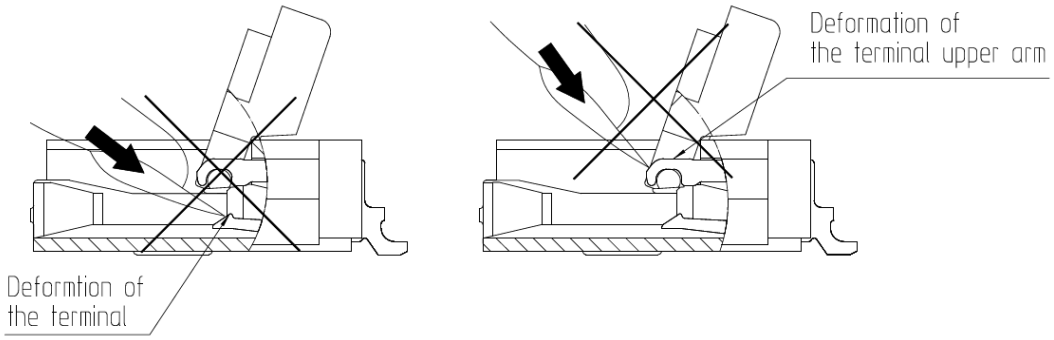

**Figure 41 Schematics of CMOD rework**

## 3.4 Bill of materials

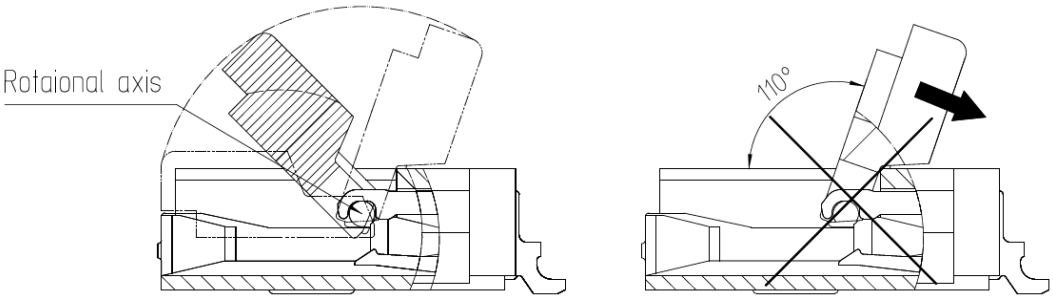
See the BoM files available in the **kit webpage**.

## 3.5 Frequently asked questions

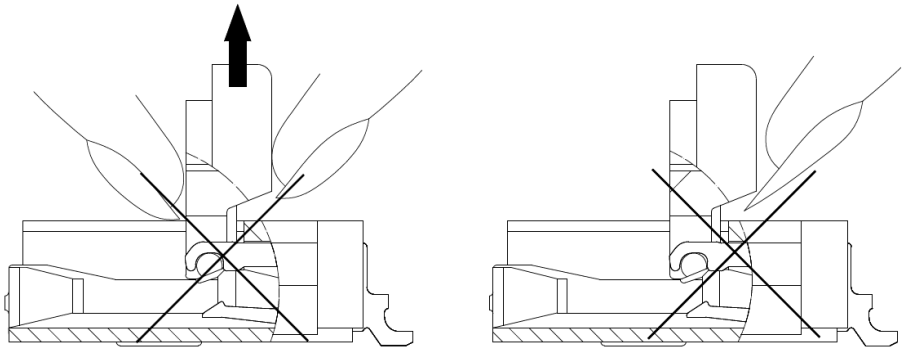
1. What additional overlays can be used with CAPSENSE™?
   - Any kind of non-conductive overlays (up to 5 mm total thickness) like wood, acrylic, and glass can be used with CAPSENSE™. Note that additional tuning may be required when the overlay is changed.
2. By mistake, I powered my Arduino board while powering the pioneer kit. Did anything get damaged?
   - No, the 5 V and 3.3 V pins on the J1 Arduino power connector are not input pins and have a protection circuit to prevent the voltage from entering the board. VIN is an input pin and this is routed to the regulator that is capable of taking an absolute maximum of 20 V.
3. What precautions needs to be taken while connecting and disconnecting the flex cable?
   - Do not apply excessive force on the actuator when releasing it from its initial position. Using a nail or finger as shown below, may result in terminal deformation.



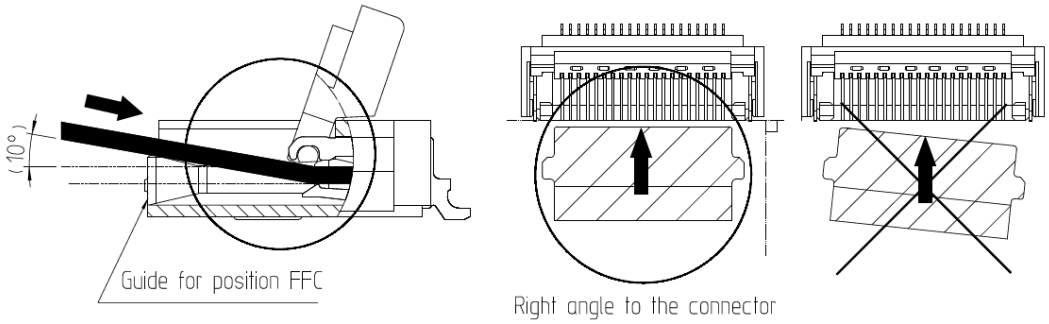

   - The actuator rotates around the axis as shown below. Make sure to rotate the whole actuator and not just the center portion.
   - The actuator opens up to 110°. Do not force the actuator beyond this point. Doing so may result in actuator damage or breakage.



- Operate the actuator towards the center. Operating one side only may result in actuator damage or incomplete mating.
- Do not pull or pick at the actuator. Doing so may damage it. (Do not carry out any operation other than the actuator rotation operation described above)



- This connector utilizes bottom contacts. Insert the FFC with the exposed conductors face down.
- Insert the FFC straight into the connector while at a 10° angle to the board surface (see illustration below). Make sure that it has been fully inserted.




- Do not insert the FFC diagonally. If the FFC is inserted in skewed orientation as illustrated above, the FFC may get caught by the terminals resulting in terminal deformation.
- After FFC insertion, lock the actuator.

4. What third-party IDEs does this kit support?
   - Multiple third-party IDEs are supported; IAR and µVision are some examples. For more details on all supported devices and procedures to export to these IDEs, see ModusToolbox™ User Guide (Help --> ModusToolbox™ General Documentation --> ModusToolbox™ User Guide [Section - 7. Exporting to IDEs]).

# Revision history

### Major changes since the last revision

| Date | Version | Description |
|---|---|---|
| 2021-03-22 | ** | Initial release. |
| 2021-05-18 | *A | Internal release. |
| 2021-06-29 | *B | Initial public release. |
| 2021-10-25 | *C | Updated **Introduction**:<br>Updated description.<br>Updated **Getting started**:<br>Updated hyperlinks.<br>Updated **Kit operation**:<br>Updated **Theory of operation**:<br>Updated hyperlinks.<br>Updated **Creating a project and program/debug using ModusToolbox™**:<br>Updated **Using the OOB example – [MSC CSX button project]**:<br>Updated hyperlinks.<br>Updated **Hardware**:<br>Updated **Schematics**:<br>Updated hyperlinks.<br>Updated **Hardware functional description**:<br>Updated **PSoC™ 4100S Max MCU**:<br>Updated hyperlinks. |
| 2021-10-27 | *D | Updated **Kit operation**:<br>Updated **Creating a project and program/debug using ModusToolbox™**:<br>Updated **Using the OOB example – [MSC CSX button project]**:<br>Updated hyperlinks. |

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

**Edition <2021-10>**
**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

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

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

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

**Document reference**
**002-32436 Rev. *D**

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

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

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

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

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

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

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