KIT_PSC3M8_EVK PSOC™ Control C3M8 Evaluation Kit guide

About this document

Scope and purpose

This document provides a comprehensive overview of the KIT_PSC3M8_EVK PSOC™ Control C3M8 Evaluation Kit, covering its operation, an out-of-the-box (OOB) example along with its functionality, and detailed hardware specifications of the board.

Intended audience

This document is intended for embedded developers who use the KIT_PSC3M8_EVK PSOC™ Control C3M8 Evaluation Kit.

Evaluation board

This board is to be used during the design-in process for evaluating and measuring characteristic curves, and for checking datasheet specifications.

***Note**: PCB and auxiliary circuits are NOT optimized for final customer design.*

Important notice

“Evaluation Boards and Reference Boards” shall mean products embedded on a printed circuit board (PCB) for demonstration and/or evaluation purposes, which include, without limitation, demonstration, reference and evaluation boards, kits and design (collectively referred to as “Reference Board”).

Environmental conditions have been considered in the design of the Evaluation Boards and Reference Boards provided by Infineon Technologies. The design of the Evaluation Boards and Reference Boards has been tested by Infineon Technologies only as described in this document. The design is not qualified in terms of safety requirements, manufacturing and operation over the entire operating temperature range or lifetime.

The Evaluation Boards and Reference Boards provided by Infineon Technologies are subject to functional testing only under typical load conditions. Evaluation Boards and Reference Boards are not subject to the same procedures as regular products regarding returned material analysis (RMA), process change notification (PCN) and product discontinuation (PD).

Evaluation Boards and Reference Boards are not commercialized products, and are solely intended for evaluation and testing purposes. In particular, they shall not be used for reliability testing or production. The Evaluation Boards and Reference Boards may therefore not comply with CE or similar standards (including but not limited to the EMC Directive 2004/EC/108 and the EMC Act) and may not fulfill other requirements of the country in which they are operated by the customer. The customer shall ensure that all Evaluation Boards and Reference Boards will be handled in a way which is compliant with the relevant requirements and standards of the country in which they are operated.

The Evaluation Boards and Reference Boards as well as the information provided in this document are addressed only to qualified and skilled technical staff, for laboratory usage, and shall be used and managed according to the terms and conditions set forth in this document and in other related documentation supplied with the respective Evaluation Board or Reference Board.

It is the responsibility of the customer’s technical departments to evaluate the suitability of the Evaluation Boards and Reference Boards for the intended application, and to evaluate the completeness and correctness of the information provided in this document with respect to such application.

The customer is obliged to ensure that the use of the Evaluation Boards and Reference Boards does not cause any harm to persons or third party property.

The Evaluation Boards and Reference Boards and any information in this document is provided "as is" and Infineon Technologies disclaims any warranties, express or implied, including but not limited to warranties of non-infringement of third party rights and implied warranties of fitness for any purpose, or for merchantability.

Infineon Technologies shall not be responsible for any damages resulting from the use of the Evaluation Boards and Reference Boards and/or from any information provided in this document. The customer is obliged to defend, indemnify and hold Infineon Technologies harmless from and against any claims or damages arising out of or resulting from any use thereof.

Infineon Technologies reserves the right to modify this document and/or any information provided herein at any time without further notice.

Safety precautions

***Note**:* *Please note the following warnings regarding the hazards associated with development systems.*

Table 1 **Safety precautions**

	Caution: *The evaluation or reference board contains parts and assemblies sensitive to electrostatic discharge (ESD). Electrostatic control precautions are required when installing, testing, servicing or repairing the assembly. Component damage may result if ESD control procedures are not followed. If you are not familiar with electrostatic control procedures, refer to the applicable ESD protection handbooks and guidelines.*
	Caution: *The evaluation or reference board is shipped with packing materials that need to be removed prior to installation. Failure to remove all packing materials that are unnecessary for system installation may result in overheating or abnormal operating conditions.*

Table of contents

Table of contents

1 Introduction

The KIT_PSC3M8_EVK PSOC™ Control C3M8 Evaluation Kit is built using Infineon's PSOC™ Control PSC3M8 series MCU to evaluate the features and capabilities of Infineon's PSOC™ Control C3 family devices. The evaluation kit integrates multiple hardware and software components to enable users to access the majority of MCU features on the board. This document provides guidance on getting started with Infineon's PSOC™ Control C3M8 Evaluation Kit and offers support for application development using Infineon's ModusToolbox™ software.

1.1 Features

PSOC™ Control C3M8 is a high-performance, low-power, 32-bit single-core Arm® Cortex®-M33-based microcontroller with a DSP, FPU, MPU, TrustZone (TZ) extension, advanced security features, 512 KB Read-While-Write (RWW) Flash with ECC support, and 128 KB SRAM.

- **Programmable Power Control Accelerator (PPCA)**:
    - Two independent Arm® Cortex®-M33 microcontroller cores with DSP, FPU, and 112 KB SRAM
    - Four 12-bit, 25 Msps SAR ADC modules with 24 channels
    - Nine differential comparator slope generators and fast analog comparators with <10 ns latency
    - Built-in 12-bit DAC and two coordinate rotation digital computer (CORDIC) blocks
    - 14 x 16-bit and 12 x 32-bit timer/counter pulse-width modulators (TCPWM)
    - 12 x 32-bit high-resolution PWMs (HRPWM) with <100 ps resolution
    - 14 x 16-bit PWMs with CPU clock resolution and complementary output
    - Configurable and programmable events processing unit (EPU) block
    - Two independent instances of six 3P3Z hardware filters and a custom logic block (CLB) implementing digital current observer and matrix converter
- **Communication interfaces**: Seven serial communication blocks (SCBs) with two PMBus interfaces and two CAN FD interfaces, one supporting up to 8 Mbps
- **Low-power operation**: Supports Sleep and Deep Sleep modes with three configurable options
- **Input/Output features**: Up to 60 GPIOs, along with 26 analog pins
- **Security**: PSA Level 3 (L3)-certified with configurable Flash partitioning and protection
- **Safety**: Class B and SIL 2-compliant safety test libraries
- **Packages**:
    - VQFN-48, E-LQFP-48, VQFN-64, E-LQFP-64, E-LQFP-80, E-LQFP-100
    - Operating temperature range: TA = -40°C to 115°C

1.2 Additional board features

In addition to the device features, the KIT_PSC3M8_EVK board includes the following evaluation-specific components:

- Onboard programmer/debugger using Infineon's PSOC™ 5LP (KitProg3) device
- Type-C USB connector for power and communication interface
- CAN FD transceiver with terminal connector
- Six user LEDs, three potentiometers, and two user buttons
- Arduino Uno-compatible header interface and expansion header for I/Os
- Onboard regulators, crystal oscillators, JTAG/SWD connector, and supporting discrete components

ModusToolbox™ is used to develop and debug PSC3M8 projects. It offers a comprehensive set of tools that enable seamless integration of these devices into development workflows, simplifying application creation and system design.

1.3 Target applications

- **Digital power applications**: Suitable for switching frequencies up to 2 MHz, used in power supplies for AI servers, telecom, EV chargers, and solar systems
- **High-switching-frequency wide-bandgap technologies**: Enables power conversion and motor control applications using silicon carbide (SiC) and gallium nitride (GaN)
- **Motor control applications**: Designed for appliances, light electric vehicles, robots, and drones

PSOC™ Control C3(P/M)(8/7)x devices are based on the Arm® Cortex®-M33 processor, running at up to 180 MHz with DSP and FPU capabilities. These devices also feature programmable power control accelerator (PPCA) peripherals with two Arm® Cortex®-M33 cores operating at up to 200 MHz. Key features include 12-bit, 25 Msps ADCs, comparators with differential slope generators, hardware filters, custom logic blocks, DACs, and advanced timers with high-resolution capabilities.

2 Kit details

This section provides detailed information about the PSOC™ Control C3M8 Evaluation Kit.

2.1 Kit contents

The kit includes the following components:

- PSOC™ Control C3M8 Evaluation Kit
- USB Type-A to USB Type-C cable
- Jumper wires

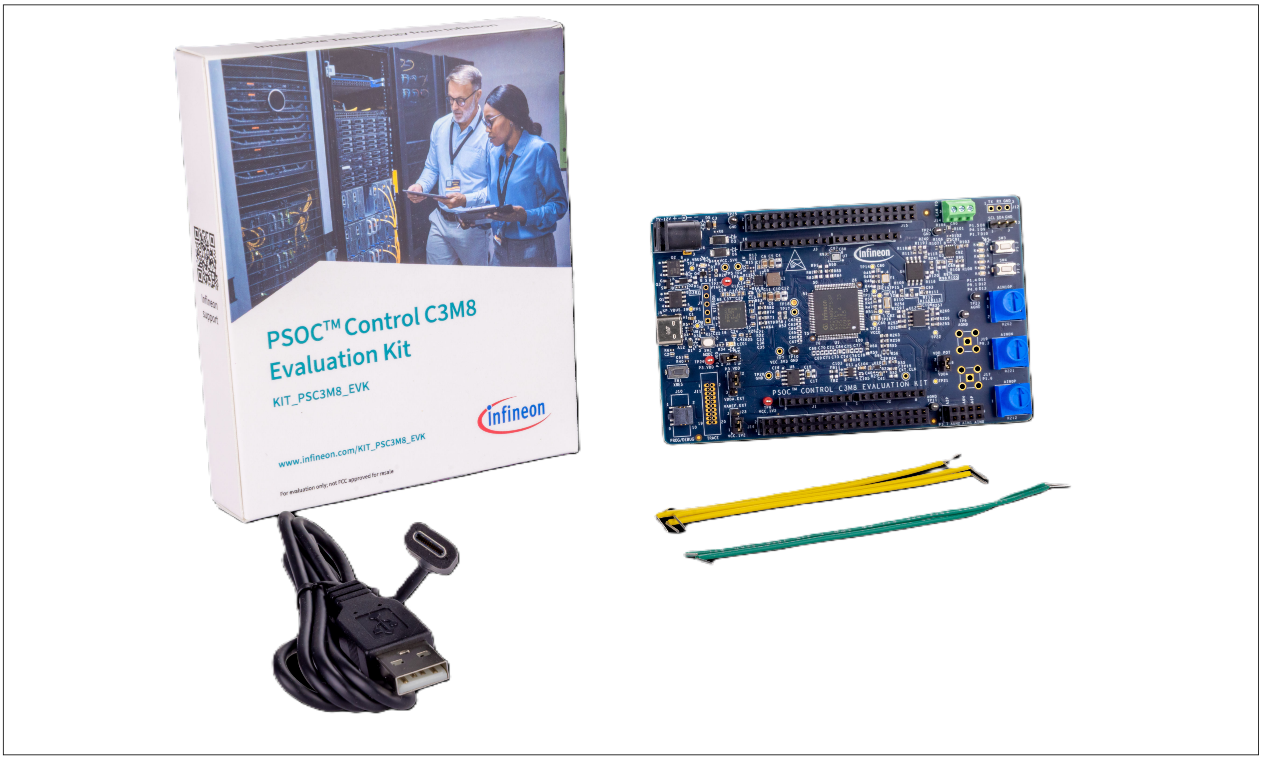

Figure 1 Kit contents

2.2 Board details

The PSOC™ Control C3M8 Evaluation Kit includes the following components and features:

- **PSOC™ Control C3M8 MCU**: PSC3M8GFS3PAHQ1
- **KitProg3 programmer/debugger**: CY8C5868LTI-LP039
- **USB Type-C connector**: For USB device interface
- **Power input options**: Power can be supplied via USB, DC jack, or Arduino header
- **CAN FD interface**: TLE9251VLE
- 2-channel DAC output header
- Onboard temperature sensor
- **LEDs**:
  - Six user LEDs
  - One power LED
  - One KitProg3 status LED
- **Potentiometers**: Three

- **Switches and buttons**:
  - Two user buttons
  - One reset switch
  - One KitProg3 mode switch
- **Arduino-compatible headers**: Compatible with Arduino Uno shields
- **Expansion header**: Provides I/O accessibility, along with I^2C and UART communication headers
- **Program/Debug interface**: JTAG/SWD interface for use with third-party emulators
- **External crystal oscillators**:
  - ECO (16 MHz)
  - WCO (32.768 kHz)
- **Current measurement header**: Includes jumper configuration to select an external voltage source for VDDA and VAREF

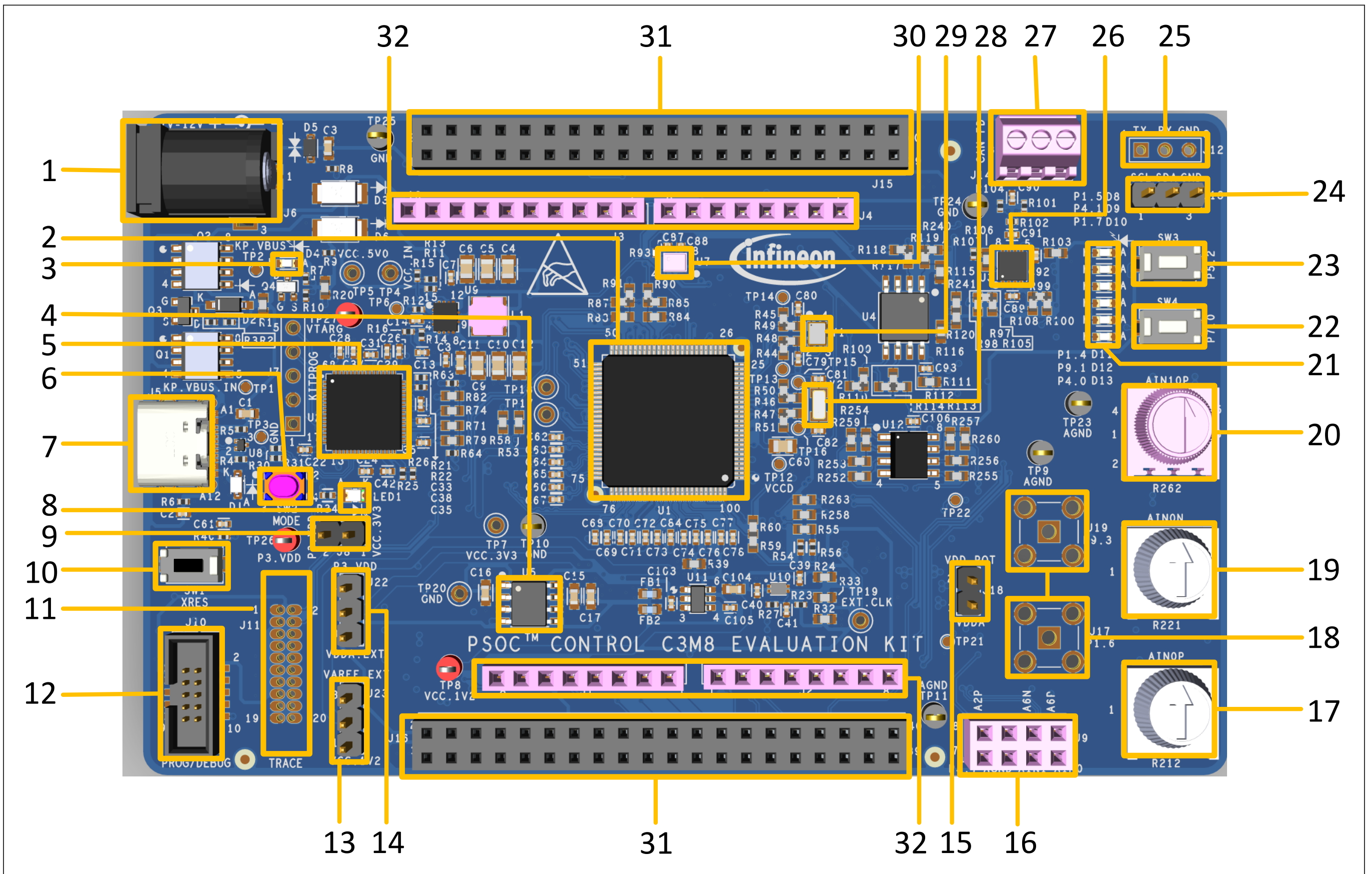

Figure 2 Rework to bypass the DAC buffer

1. **VIN external power supply connector (J6)**: External power supply through a DC jack. Supports a DC adapter with 7 V–12 V and 1 A as a power source for the kit
2. **PSOC™ C3M8 device (U1 - PSC3M8GFS3PAHQ1)**: The target MCU for evaluating the features of PSOC™ Control C3M8 on the evaluation kit
3. **Power LED (D4)**: Indicates the power ON/OFF status of the evaluation kit
4. **3.3 V LDO (U5 - TLS208D1EJV)**: A 3.3 V, 800 mA fixed output voltage regulator to power the MCU and other peripherals on the evaluation kit
5. **KitProg3 programmer and debugger (U2 - CY8C5868LTI-LP039)**: PSOC™ 5LP-based KitProg3 onboard programmer and debugger with USB-UART, USB-I^2C bridge, and SWD interface
6. **KitProg3 programming mode selection button (SW2)**: Mode selection switch for KitProg3

7. **USB Type-C connector (J5)**: USB Type-C connector for power sourcing and communication over USB with an external system
8. **KitProg3 Status LED (LED1)**: Indicates the status of KitProg3 and its mode of operation
9. **Current measurement jumper (J8)**: Used to measure MCU current consumption in active, deep sleep, and hibernate operating modes
10. **Reset switch (SW1)**: Manually resets the MCU on the evaluation kit
11. **ETM trace debug header (J11)**: Header for trace debugging of the MCU using external emulators
12. **SWD/JTAG Cortex® debug header (J10)**: SWD and JTAG programming and debugging interface header
13. **VAREF voltage selection header (J23)**: Header for selecting the VAREF voltage for the PSOC™ Control device, either from the onboard regulator or an external power source
14. **VDDA voltage selection header (J22)**: Header for selecting the VDDA voltage for the PSOC™ Control device, either from the onboard regulator or an external source
15. **Potentiometer power input header (J18)**: VDDA power supply header for the potentiometer
16. **DAC-ADC interface header (J9)**: Header for evaluating DAC output. Includes an option to loop back the DAC output to the ADC on the evaluation kit
17. **Potentiometer (R212 - 10K):** Potentiometer for evaluating analog functionality on the evaluation kit
18. **SMA connector interface (J17 and J19)**: SMA connector provision to evaluate HRPWM signals for probing purposes
19. **Potentiometer (R221 - 10K)**: Potentiometer for evaluating analog functionality on the evaluation kit
20. **Potentiometer (R262 - 500K)**: Potentiometer for evaluating analog functionality on the evaluation kit
21. **User LEDs (D8 - D13)**: Six active-low user application LEDs (3 green and 3 orange)
22. **User button (SW4)**: Provides input to the PSOC™ C3M8 MCU. Connects the pin to ground when pressed. The PSOC™ pin must be configured as a digital input with a resistive pull-up to detect button presses
23. **User button (SW3)**: Provides input to the PSOC™ C3M8 MCU. Connects the pin to ground when pressed. The PSOC™ pin must be configured as a digital input with a resistive pull-up to detect button presses
24. **I^{2}C external interface header (J13)**: External 3-pin header with I^{2}C interface from MCU for communication with external devices or sensors
25. **UART external interface header (J12)**: External 3-pin header with UART interface from MCU for communication with external devices or sensors
26. **CAN FD transceiver (U3)**: CAN FD interface for communication
27. **CAN FD terminal connector (J14)**: Terminal block for connecting external devices for CAN communication
28. **WCO crystal oscillator (Y2 - 32.768 kHz)**: Watch crystal oscillator on the evaluation kit
29. **ECO crystal oscillator (Y1 - 16 MHz)**: External crystal oscillator on the evaluation kit. Allows the MCU to operate with an internally generated clock source or ECO as an external clock source for the MCU PLL
30. **I^{2}C-based temperature sensor (U7)**: I^{2}C-based temperature sensor for evaluating the digital interface on the evaluation kit
31. **Expansion header (J15 and J16)**: Provides access to free/unused GPIOs of the MCU on the evaluation kit. Some I/Os are multiplexed with other functional units on the evaluation kit and may require resistor rework for connections
32. **Header compatible with Arduino Uno (J1 - J4)**: Compatible with Arduino Uno R3 shields. Allows evaluation of free/unused GPIOs from the MCU through the Arduino interface header

PIN	Primary function	Secondary function	Connection details
XRES	Hardware reset	-	Reset is connected to the MCU and headers, including the JTAG/SWD connector, ETM Trace connector, Arduino-compatible header, and I/O expansion header.

PIN	Primary function	Secondary function	Connection details
P0.0	WCO OUT(Y2)	-	Connected to a 32.768 kHz watch crystal
P0.1	WCO IN(Y2)	-	Connected to a 32.768 kHz watch crystal
P1.0	ECO IN(Y1)	-	Connected to a 16 MHz external crystal
P1.1	ECO OUT(Y1)	-	Connected to a 16 MHz external crystal
P1.2	TCLK_SWCLK	-	Connected between the onboard debugger, MCU 10-pin SWD/JTAG interface at J10, and 20-pin ETM header at J11.
P1.3	TMS_SWDIO	-	Connected between the onboard debugger, the MCU 10-pin SWD/JTAG interface at J10, and the 20-pin ETM header at J11.
P1.4	LED (D11)	Expansion header J15.5	By default, connected to LED (D11). Remove R184 and load R171 to use P1.4 as a GPIO on the expansion header.
P1.5	LED (D8)	Expansion header J15.7	By default, connected to LED (D8). Remove R168 and load R174 to use P1.5 as a GPIO on the expansion header.
P1.6	Connected to expansion header J15.9 and AIN7P	HRPWM (SMA)	By default, connected to expansion header J15.9 and ADC input AIN7P using an RC network. Remove R209 (10K) and populate R206 to probe the HRPWM signal on the SMA connector interface (SMA is not loaded by default on the evaluation kit).
P1.7	LED (D10)	Expansion header J15.11	By default, connected to LED (D10). Remove R178 and load R176 to use P1.7 as a GPIO on the expansion header.
P2.0	SPI MOSI	Expansion header J15.13	By default, connected to the SPI MOSI of NOR flash memory. Remove R109 and populate R110 to use P2.0 as a GPIO on J15.13.

PIN	Primary function	Secondary function	Connection details
P2.1	SPI CLOCK	Expansion headerJ15.15 and TDI on J10 and J11	By default, connected to the SPI Clock of NOR flash memory. When using an external debugger, remove R112 and populate R113 to use it as the TDI pin on J10 and J11 headers. Populate R114 to use P2.1 as a GPIO on J15.15.
P2.2	SPI MISO	Expansion header J15.17	By default, connected to the MISO of SPI NOR flash memory. Remove R117 and populate R118 to use P2.2 as a GPIO on J15.17.
P2.3	I^2C SCL	Expansion header J15.19	Connected to the I^2C SCL between the PSOC™ C3M8 MCU and the KitProg3 debugger. Alternatively, it is connected to expansion header J15.19 and J13 for accessibility of the I^2C interface from external devices.
P2.4	I^2C SDA	Expansion header J15.21	Connected to the I^2C SDA between the PSOC™ C3M8 MCU and the KitProg3 debugger. Alternatively, it is connected to expansion header J15.21 and J13 for accessibility of the I^2C interface from external devices.
P2.5	Expansion header J15_25	-	-
P3.0	SPI CS#	Expansion header J15.27	By default, connected to the CS# pin of SPI NOR flash memory. Remove R119 and populate R240 to use P3.0 as a GPIO on J15.27.
P3.1	Expansion header J15.29	SPI CS#2	By default, connected to J15.29. Alternatively, it can be connected to a pull-up resistor to use as a chip select pin for an external device interface. Populate R120 (0 Ω) and R116 (10K) for an external pull-up.
P3.2	Connected to header compatible with Arduino J4.3	-	-

PIN	Primary function	Secondary function	Connection details
P3.3	Debug UART TX	Expansion header J15.26	By default, connected to the Debug UART TX between the KitProg3 and the PSOC™ C3M8 MCU. Remove R85 and populate R90 to use P3.3 as a GPIO on J15.26.
P3.4	Debug UART RX	Expansion header J15.24	By default, connected to the Debug UART RX between the KitProg3 and the PSOC™ C3M8 MCU. Remove R87 and populate R91 to use P3.4 as a GPIO on J15.24.
P3.5	Connected to header compatible with Arduino J4.1 and expansion header J16.23	-	-
P3.6	Connected to header compatible with Arduino J4.2 and expansion header J16.21	-	-
P3.7	Connected to header J9.7 (TCPWM)	Expansion header J15.31	By default, connected to J9.7 to be used as a TCPWM input, which can be looped back to the ADC input AIN2P on J9.8. The GPIO is also available on expansion header J15.31 by populating R175.
P4.0	LED (D13)	Expansion header J15.4	By default, connected to LED (D13). Remove R197 and load R182 to use P4.0 as a GPIO on expansion header J15.4.
P4.1	LED (D9)	Expansion header J15.6	By default, connected to LED (D9). Remove R173 and load R185 to use P4.1 as a GPIO on expansion header J15.6.
P4.2	Connected to expansion header J15.8	-	-
P4.3	Connected to expansion header J15.10	-	-
P4.4	Connected to expansion header J15.12	-	-
P4.5	Connected to expansion header J15.14	-	-
P4.6	Connected to expansion header J15.16	-	-

PIN	Primary function	Secondary function	Connection details
P4.7	Connected to expansion header J15.18	-	-
P5.0	Connected to header compatible with Arduino J3.9 and Expansion header J16.19	-	-
P5.1	Connected to header compatible with Arduino J3.10 and Expansion header J16.17	-	-
P5.2	SW3, J16.36 and J16.38	-	By default, connected to SW3 and header pins for the push button interface. Remove R166 to use P5.2 as a GPIO on J16.36. Remove R169 to use the push button independently on J16.38.
P5.3	Connected to header compatible with Arduino J15.30	-	-
P6.0	Connected to expansion header J15.20	-	-
P6.1	Connected to expansion header J15.22	-	-
P6.2	Connected to header compatible with Arduino J3.3 and Expansion header J16.25	-	-
P6.3	Connected to header compatible with Arduino J3.5 and Expansion header J16.27	-	-
P6.4	Connected to header compatible with Arduino J3.4 and Expansion header J16.29	-	-
P6.5	Connected to header compatible with Arduino J3.6 and Expansion header J16.31	-	-
AIN0P-ADC0	Potentiometer (R212)	Expansion header J16.24	By default, connected to potentiometer R212 (10K). Remove R213 and populate R170 to use AIN0P as an analog pin on J16.24.

PIN	Primary function	Secondary function	Connection details
AIN0N-ADC0	Potentiometer (R221)	Expansion header J16.26	By default, connected to potentiometer R221 (10K). Remove R222 and populate R172 to use AIN0N as an analog pin on J16.26.
AIN1P-ADC0	Connected to header compatible with Arduino J2.8	-	-
AIN1N-ADC0	Connected to header compatible with Arduino J2.7	-	-
AIN2P-ADC0	DAC-ADC header J9.8	-	-
AIN2N-ADC0	Connected to expansion header J16.6	-	-
AIN0	DAC-ADC header J9.1	-	-
AIN1	DAC-ADC header J9.3	-	-
AIN3P-ADC0	Connected to header compatible with Arduino J2.6	-	-
AIN3N-ADC0	Connected to header compatible with Arduino J2.5	-	-
AIN4P-ADC1	Current sense+	-	-
AIN4N-ADC1	Current sense-	-	-
AIN5P-ADC1	Connected to header compatible with Arduino J2.3	-	-
AIN5N-ADC1	Connected to header compatible with Arduino J2.4	-	-
AIN6P-ADC2	DAC-ADC header J9.2	-	-
AIN6N-ADC2	DAC-ADC header J9.4	-	-
AIN7P-ADC2	Connected to P1.6 (HRPWM)	Expansion header J16.28	By default, connected to P1_6 GPIO using an RC network. Remove R210 and populate R207 to use AIN7P as an analog pin on J16.28.
AIN7N-ADC2	Connected to P9.3 (HRPWM)	Expansion header J16.30	By default, connected to P9_3 GPIO using an RC network. Remove R219 and populate R217 to use AIN7N as an analog pin on J16.30.
VAREF_EXT	ADC reference voltage	-	-

PIN	Primary function	Secondary function	Connection details
AIN8P-ADC3	Expansion header J16.32	-	-
AIN8N-ADC3	Expansion header J16.34	-	-
AIN9P-ADC3	Voltage sense+	Expansion header J16.16	By default, connected to the voltage sense + input (resistor divider). Remove R55 and populate R201 to use AIN9P as an analog pin on J16.16.
AIN9N-ADC3	Voltage sense-	Expansion header J16.18	By default, connected to the voltage sense - (minus) input. Remove R258 and populate R202 to use AIN9N as an analog pin on J16.18.
AIN10P-ADC3	Connected to potentiometer R262 and expansion header J16.20	-	-
AIN10N-ADC3	Connected to expansion header J16.22	-	-
AIN11P-ADC3	Connected to header compatible with Arduino J2.2	-	-
AIN11N-ADC3	Connected to header compatible with Arduino J2.1	-	-
P7.0	SW4	Expansion header J15.33	By default, connected to the push button (SW4). Remove R155 and load R244 to use P7.0 as a GPIO on expansion header J15.33.
P7.1	Connected to header compatible with Arduino J4.5	-	-
P7.2	Expansion header J15.35	-	-
P7.3	Expansion header J15.37 and Temperature sensor ADDR configuration pin	-	-
P7.4	CAN1_TX	Expansion header J16.3	By default, connected to the TXd of the CAN FD transceiver. Remove R97 and load R98 to use P7.4 as a GPIO on expansion header J16.3.
P7.5	CAN1_RX	Expansion header J16.5	By default, connected to the RXd of the CAN FD transceiver. Remove R99 and load R100 to use P7.5 as a GPIO on expansion header J16.5.

PIN	Primary function	Secondary function	Connection details
P7.6	CAN STB (Standby) mode pin	Expansion header J16.7	By default, connected to the STB of the CAN FD transceiver. Remove R106 and load R181 to use P7.6 as a GPIO on expansion header J16.7.
P7.7	Connected to expansion header J16.9	-	-
P8.0	Connected to header compatible with Arduino J4.6	-	While using DFU function, this pin will be configured as UART Rx
P8.1	Connected to header compatible with Arduino J4.7	-	While using DFU function, this pin will be configured as UART Tx
P8.2	Connected to expansion header J4.8	-	DFU Enable - This pin will be configured to high state (P3_VDD) to use DFU via UART
P8.3	Connected to expansion header J15.32	-	-
P8.4	Connected to expansion header J16.11	-	-
P8.5	Connected to header compatible with Arduino J3.2 and Expansion header J16.33	-	-
P9.0	Connected to header compatible with Arduino J3.1	-	-
P9.1	LED (D12)	Expansion header J15.36	By default, connected to LED (D12). Remove R190 and load R179 to use P9.1 as a GPIO on expansion header J15.36.
P9.2	Connected to header compatible with Arduino J4.4	-	-
P9.3	Connected to expansion header J15.38 and AIN7N	HRPWM (SMA)	By default, connected to expansion header J15.38 and ADC input AIN7N using an RC network. Remove R218 (10K) and populate R215 to probe the HRPWM signal on the SMA connector interface (SMA is not loaded by default on the evaluation kit).

3 Kit operation

3.1 Board support package (BSP) selection

The PSOC™ Control C3M8 Evaluation Board includes a KIT_PSC3M8_EVK board support package (BSP) within the PSOC™ Control BSPs. To build and run the code examples on the kit, follow the instructions provided in the Create an Application section of the ModusToolbox™ user guide.

3.2 KitProg3: Onboard programmer and debugger

The PSOC™ Control C3M8 Evaluation Kit can be programmed and debugged using the onboard KitProg3. KitProg3 functions as an onboard programmer and debugger with USB-UART and USB-I^{2}C bridge capabilities. KitProg3 supports CMSIS-DAP but does not support mass storage. The PSOC™ 5LP device is used to implement the KitProg3 functionality.

3.3 Programming and debugging using the ModusToolbox™

1. Connect the board to the PC using the provided USB cable through the KitProg3 USB connector. If you are connecting it to the PC for the first time, the board enumerates as a USB composite device
2. On this kit, KitProg3 operates in CMSIS-DAP Bulk mode. The status LED (LED1) remains 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
3. In the Eclipse IDE for ModusToolbox™, import the desired code example (application) into a new workspace:
   a. Click on **New Application** from the **Quick Panel**

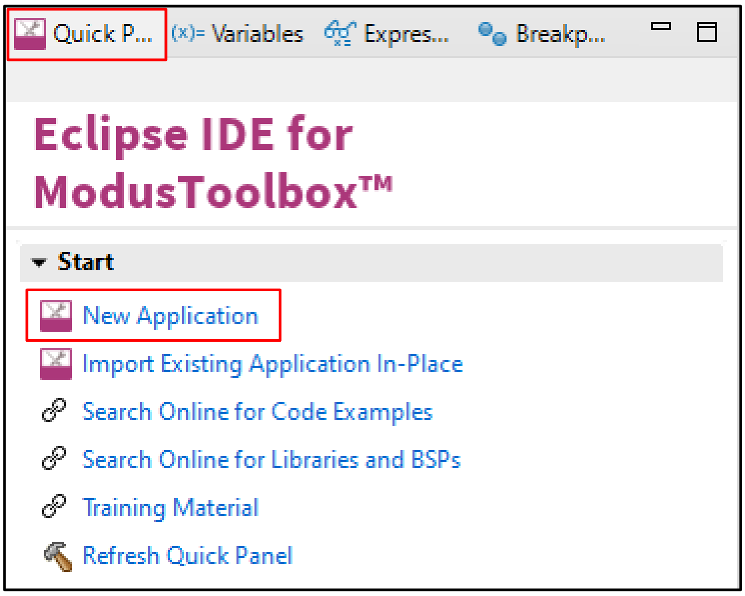

Figure 3 Create new application

   b. Select the BSP in the **Choose Board Support Package** window

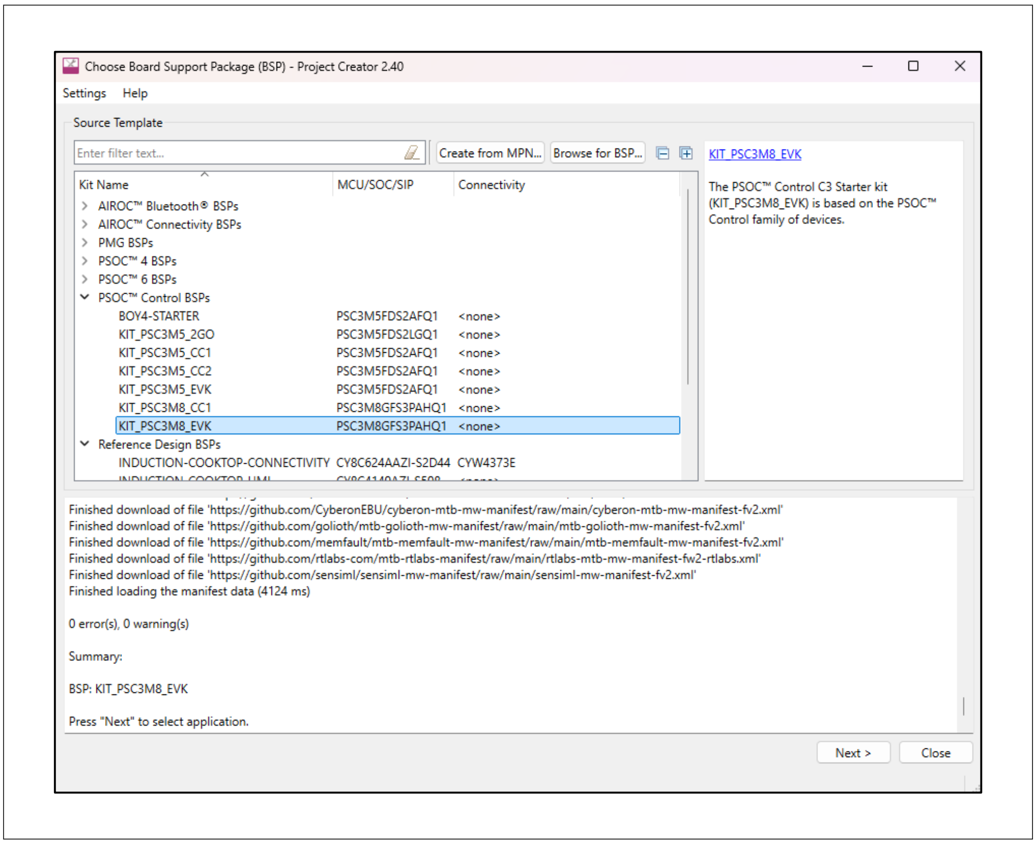

Figure 4 **Board support package in the new application window**

c. Click **Next**

d. Select the application in the **Select Application** window and click **Create**

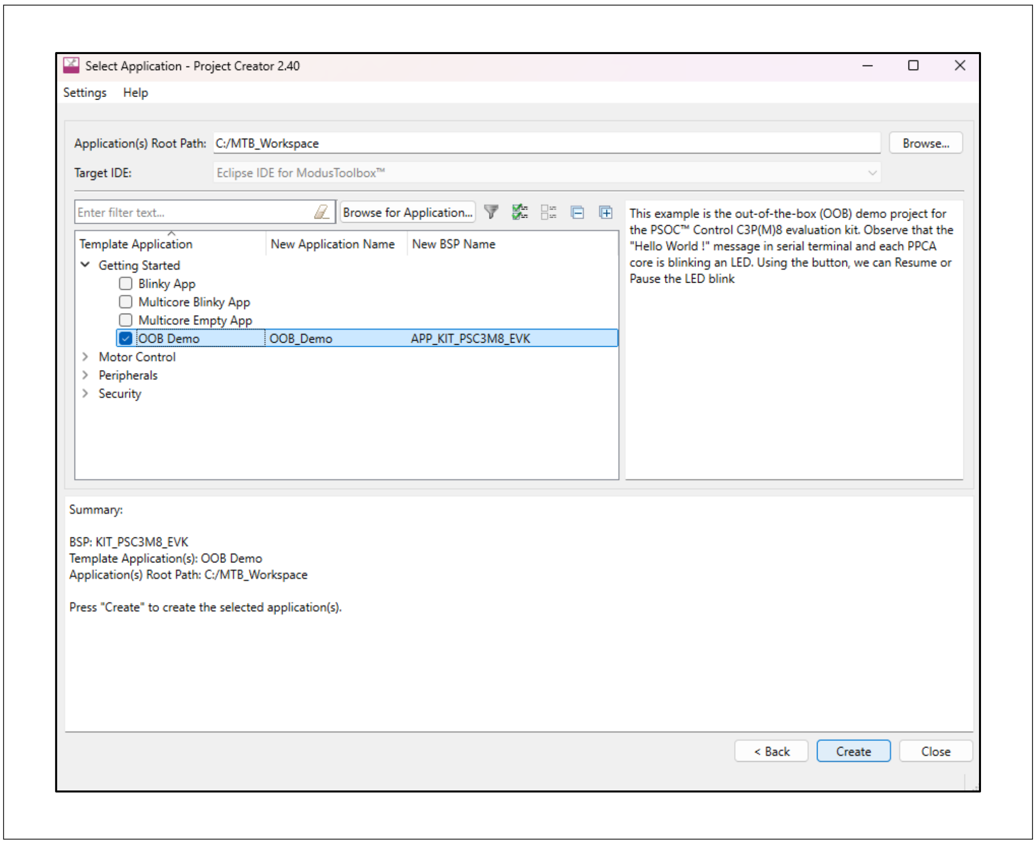

Figure 5 **Select application window**

4. To build and program an application:
   a. In the **Project Explorer**, select a project
   b. In the **Quick Panel**, scroll to the **Launches** section and click **Program Application** configuration

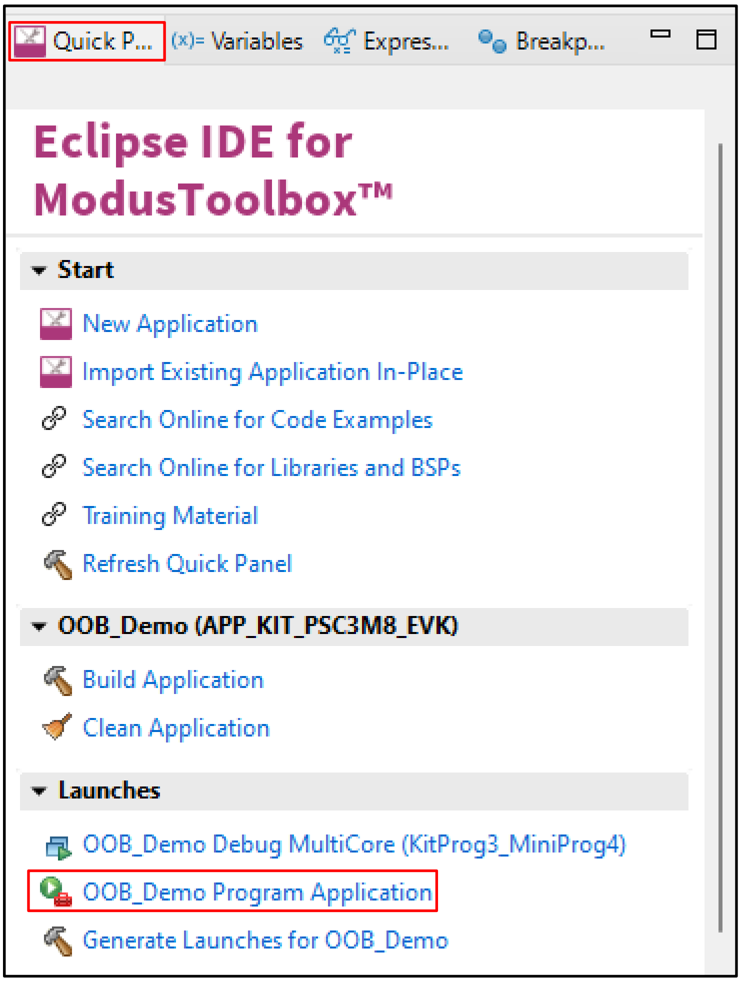

Figure 6 Programming in the ModusToolbox™ IDE

5. ModusToolbox™ software has an integrated debugger. To debug an application:

a. In the **Project Explorer**, select the project

b. In the **Quick Panel**, scroll to the **Launches** section and click **Debug (KitProg3_MiniProg4)** configuration

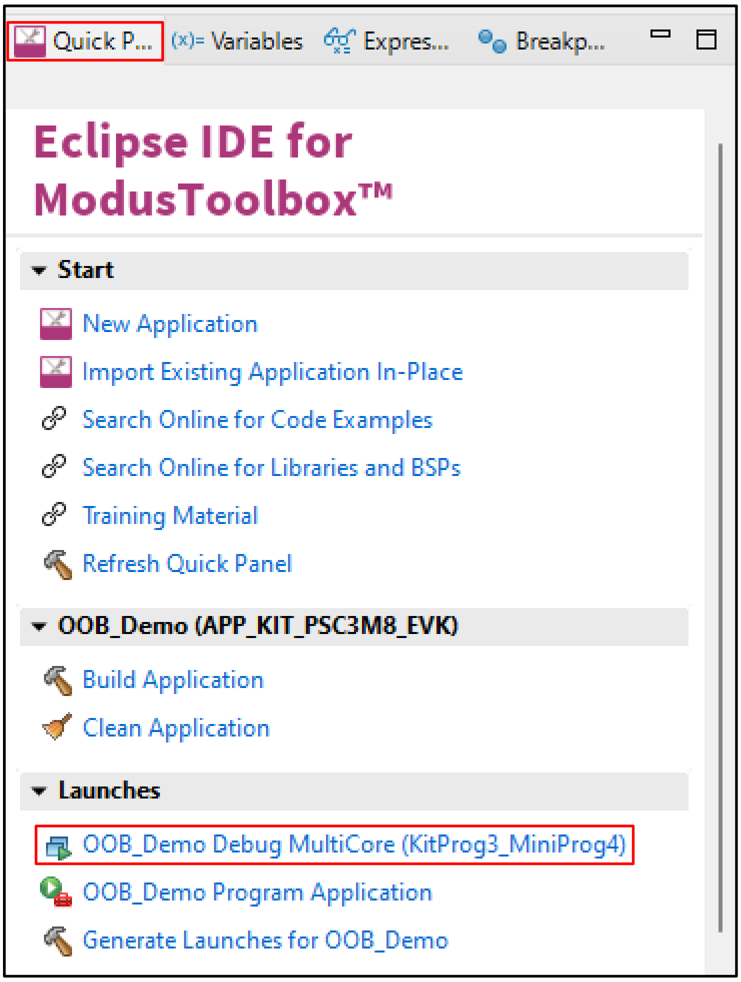

Figure 7 **Debugging in ModusToolbox™ IDE**

4 Hardware

The following section outlines the hardware design for the peripheral interfaces connected to the PSC3M8 device on the evaluation board. The schematic is designed using Allegro Capture CIS 2023.

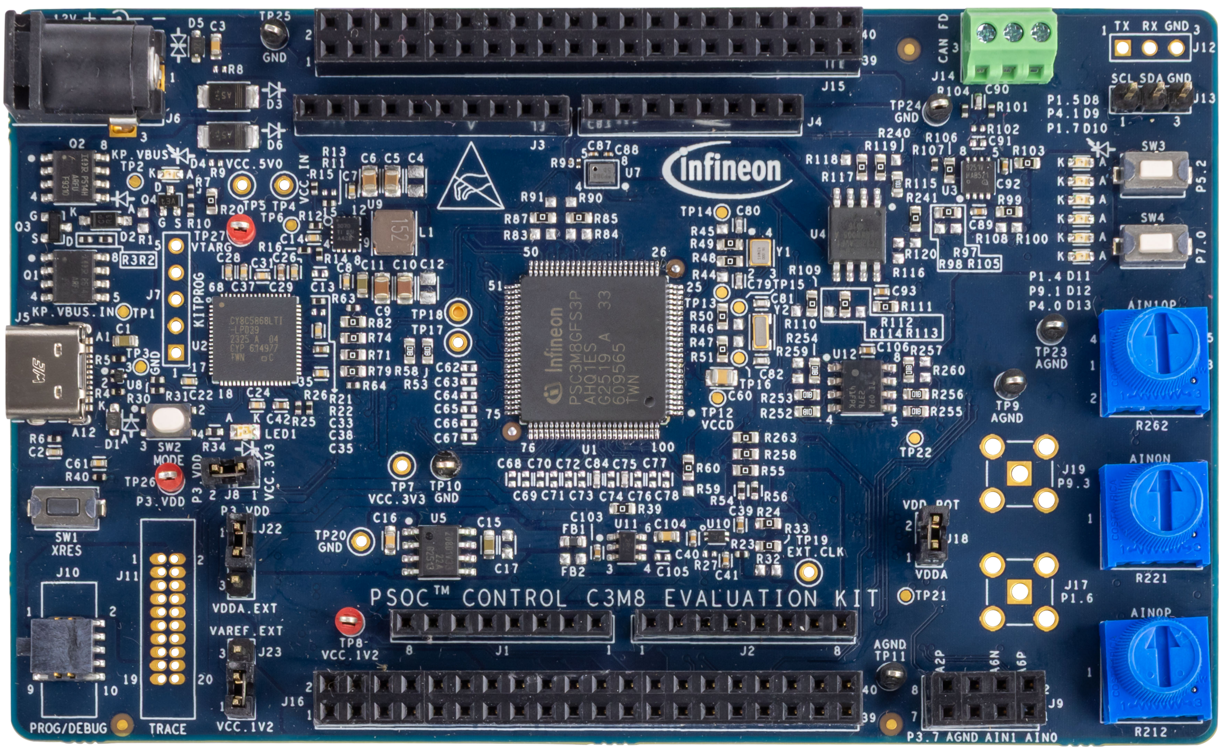

Figure 8 **PSOC™ Control C3M8 Evaluation Kit top view**

4.1 Schematics

The schematic files are available on the PSOC™ Control C3M8 Evaluation Kit webpage.

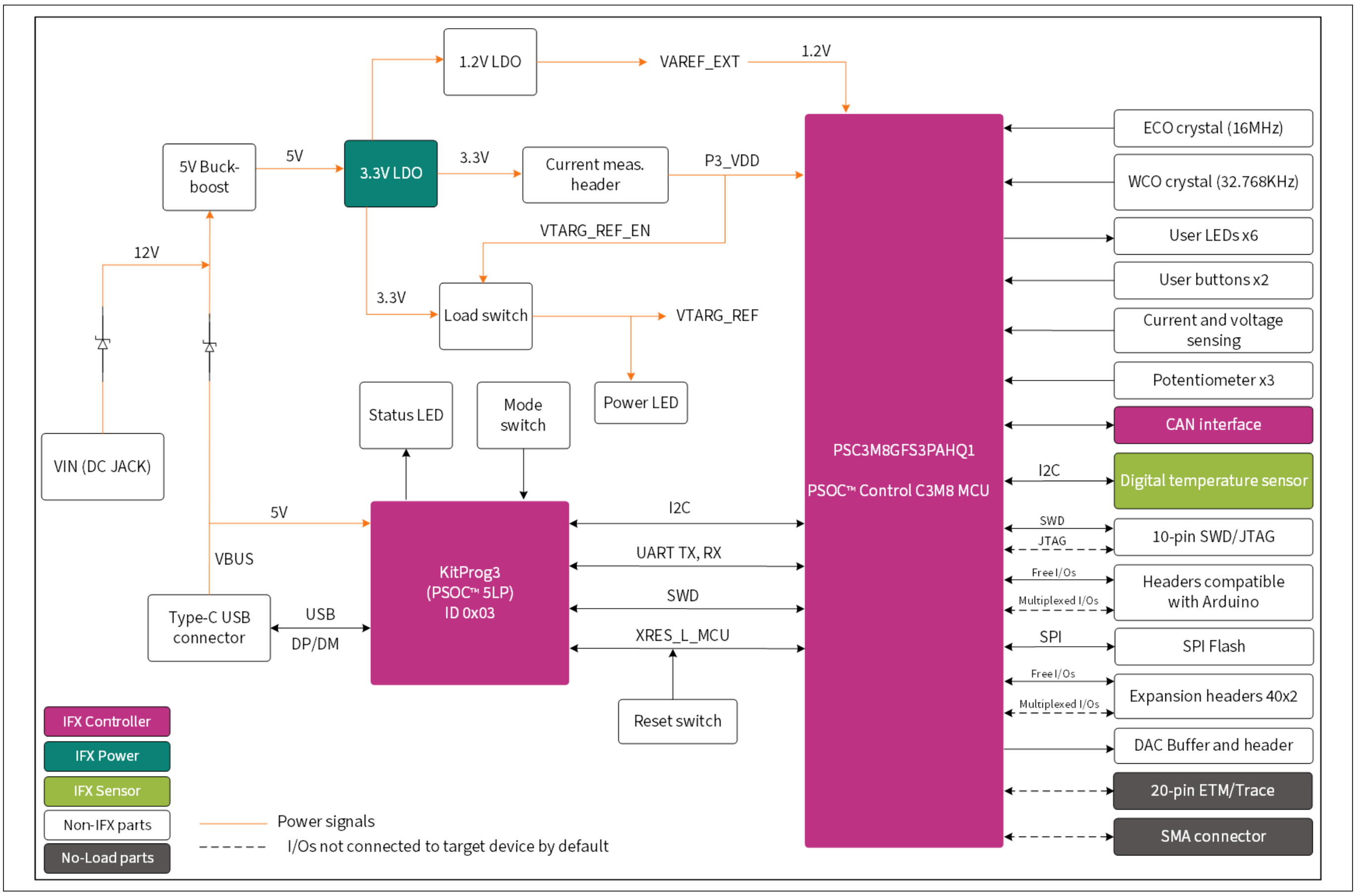

Figure 9 **Block diagram**

4.2 Hardware functional description

4.2.1 PSC3M8 (U1)

PSOC™ Control C3(P/M)(8/7)x devices are built on the Arm® Cortex®-M33 processor, operating at up to 180 MHz with DSP and FPU capabilities. These devices include programmable power control accelerator (PPCA) peripherals with two Arm® Cortex®-M33 cores running at up to 200 MHz. Key features include:

- 12-bit, 25 Msps ADCs
- Comparators with differential slope generators
- Hardware filters
- Custom logic blocks
- DACs
- Advanced timers with high-resolution capabilities

The USB Type-C cable provided with the kit allows users to power the evaluation kit and establish communication between the evaluation kit and the test system over USB.

To evaluate the full functionality of the MCU, all I/Os are either routed to functional interfaces or connected to headers for external evaluation. However, some I/Os, such as those used for dedicated functions (for example, WCO, ECO, and current sensing circuits with analog signals), are not available on the headers.

Not all GPIOs are connected to headers by default. GPIOs connected to functional units are multiplexed using zero-ohm resistors and can be routed to the expansion header or the header compatible with the Arduino interface. Users must perform the necessary rework on the evaluation kit to enable GPIOs that are not connected to headers by default.

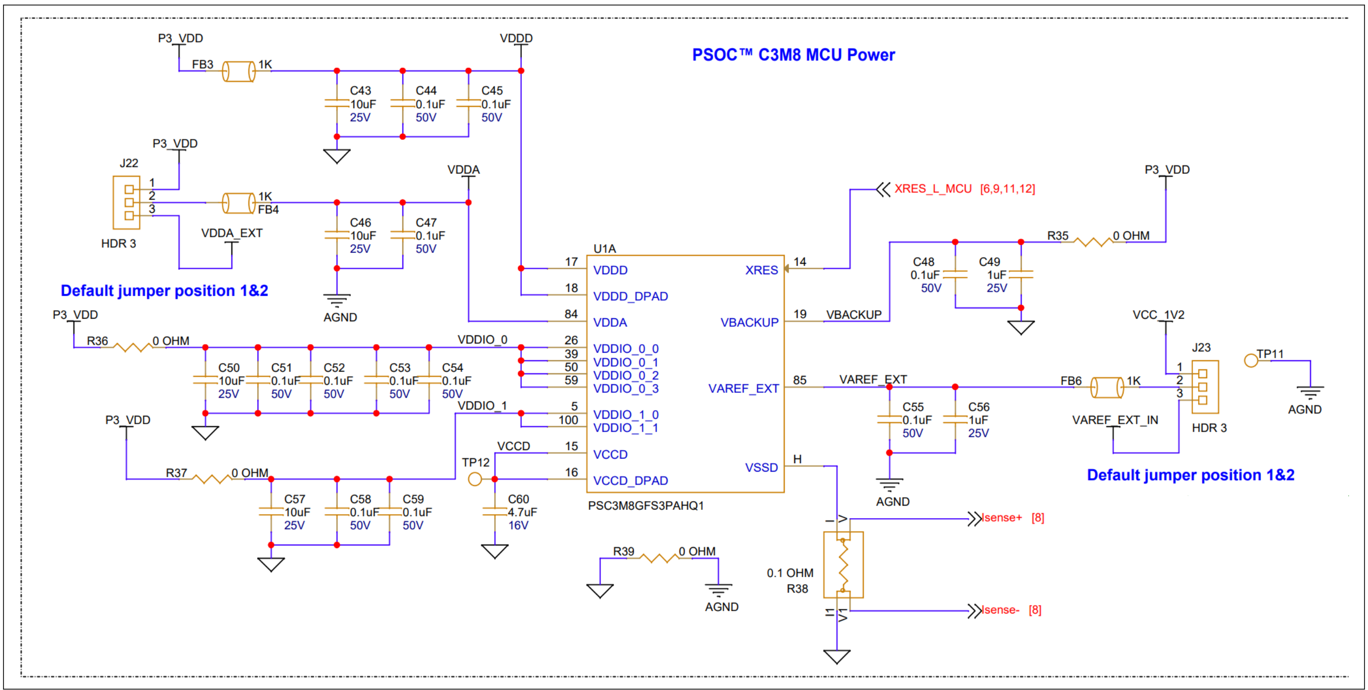

Figure 10 Power supply for PSOC™ C3M8 device

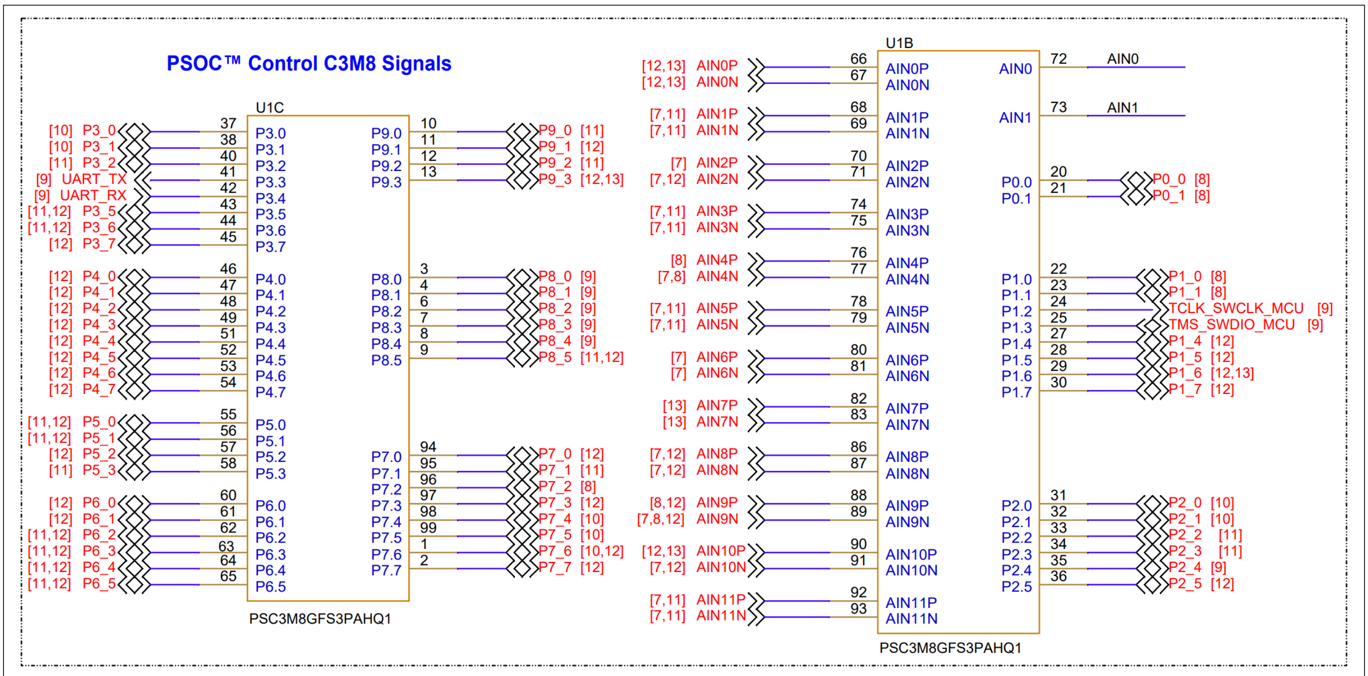

Figure 11 PSOC™ C3M8 signals

4.2.2 PSOC™ 5LP-based KitProg3 (U2)

An onboard PSOC™ 5LP (CY8C5868LTI-LP039) functions as KitProg3 to program and debug the PSC3M8GFS3PAHQ1. The PSOC™ 5LP device connects to the USB port of a PC through a USB connector and interfaces with the SWD and other communication interfaces of the PSC3M8GFS3PAHQ1.

For additional information, visit the PSOC™ 5LP webpage and refer to the CY8C58LPxx family datasheet.

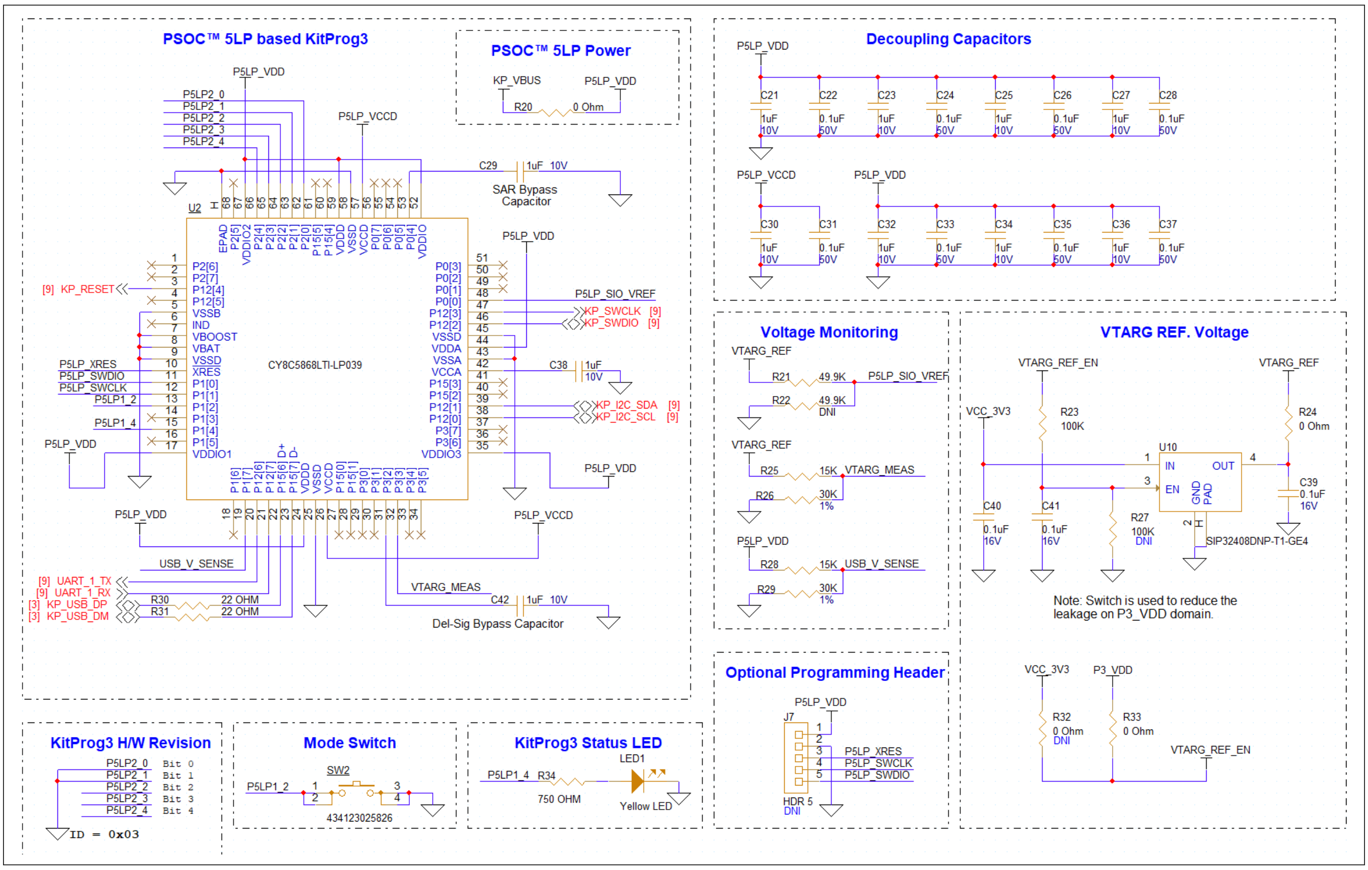

Figure 12 PSOC™ 5LP based KitProg3 (onboard programmer and debugger)

4.2.3 Serial interconnection between PSOC™ 5LP and PSC3M8

The PSOC™ 5LP device functions as an onboard programmer and also serves as an interface for USB-UART and USB-I²C bridges. The USB-serial pins of the PSOC™ 5LP device are hard-wired to the I²C and UART pins of the PSC3M8 device. Additionally, the UART and I²C pins are accessible on the I/O headers J12 and J13 for interfacing with external devices.

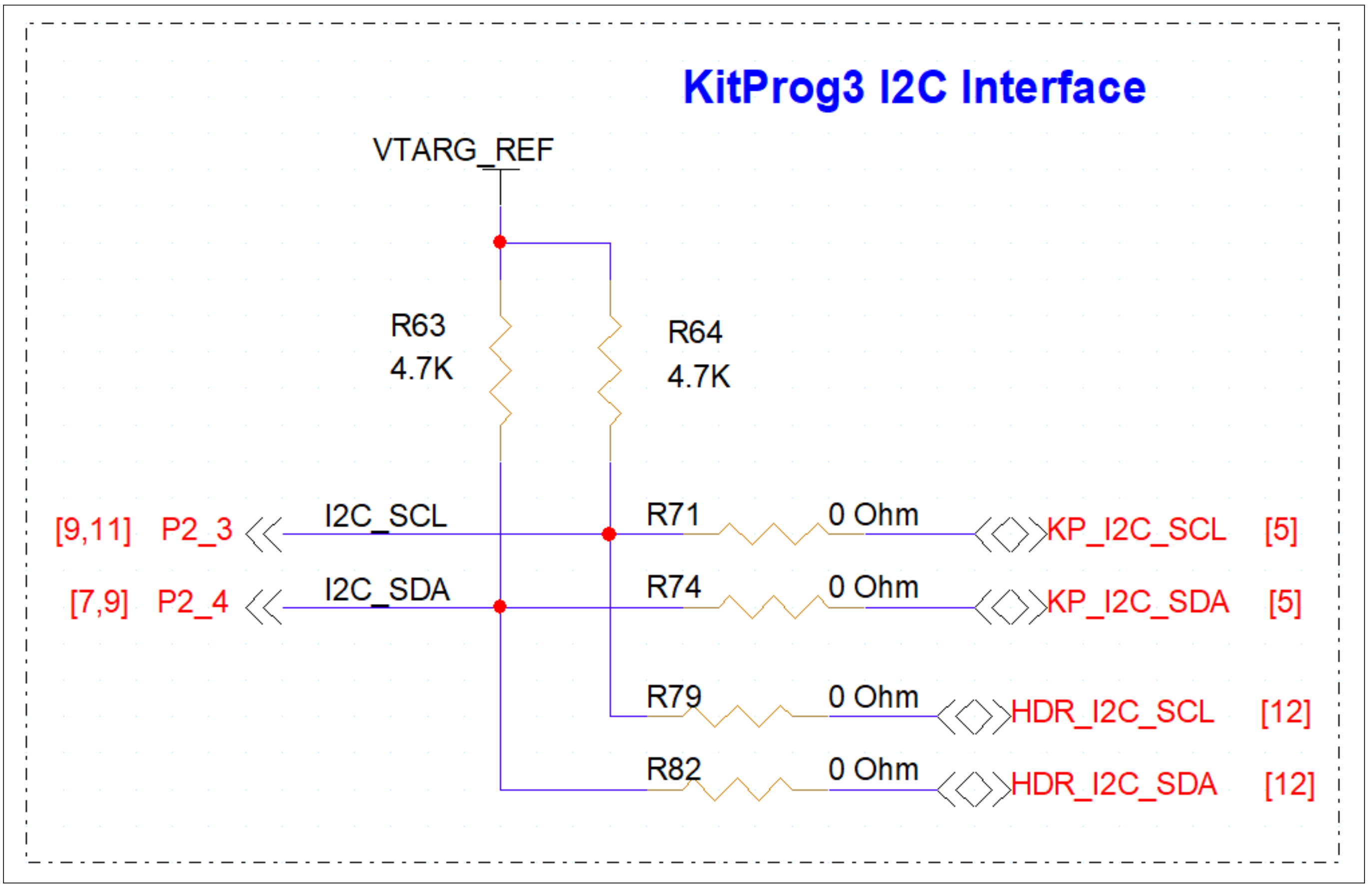

Figure 13 I^2C interface between PSOC™ C3M8 device and the onboard debugger

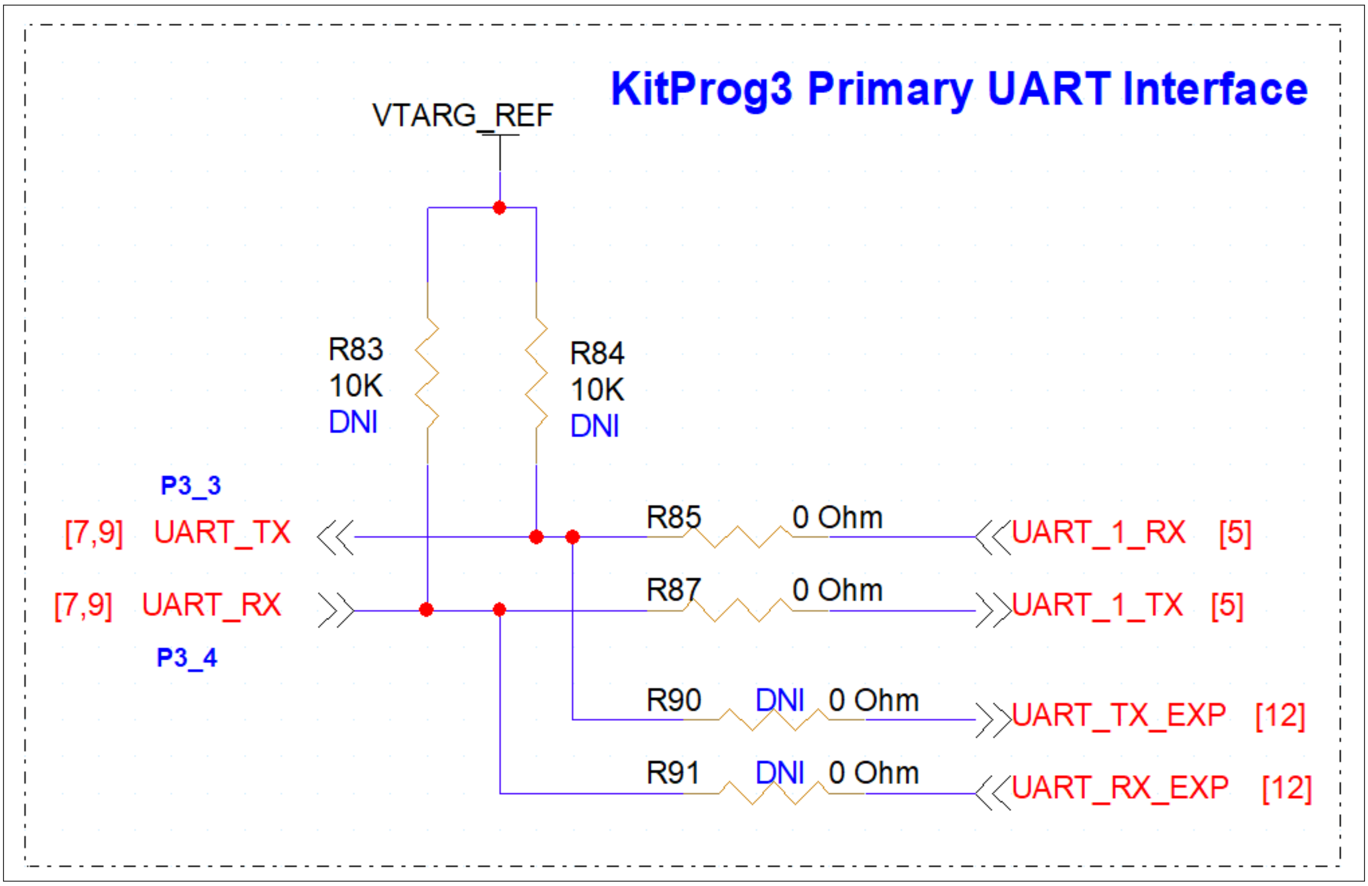

Figure 14 UART interface between PSOC™ C3M8 device and the onboard debugger

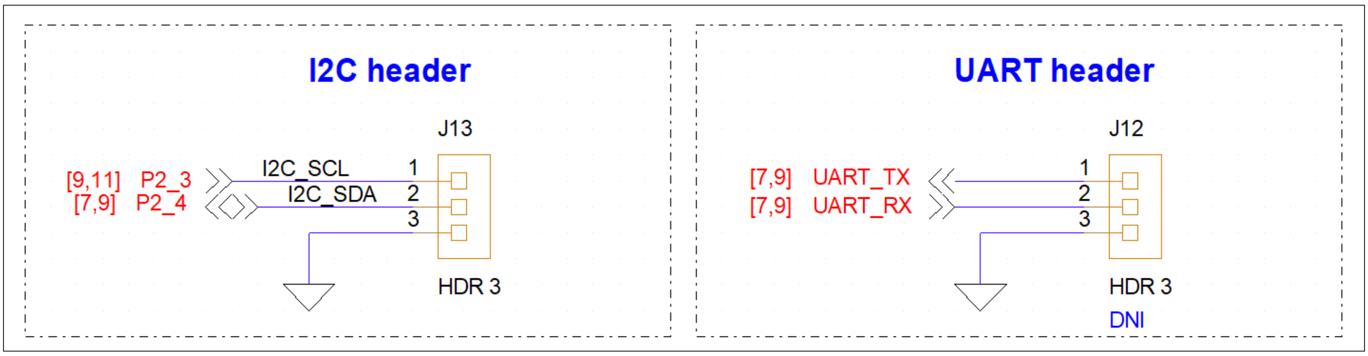

Figure 15 I²C and UART headers to interface PSOC™ C3M8 device with external devices

4.2.4 Programming/debugging headers

The PSOC™ Control C3M8 Evaluation Kit is equipped with dedicated headers for programming the PSC3M8 device.

A 10-pin SWD/JTAG header (J10) provides the capability to program the target device using either the SWD or JTAG interface. Additionally, a 20-pin ETM standard header (J11) is included for trace debugging. The trace output from the PSC3M8 device can be utilized for debugging purposes.

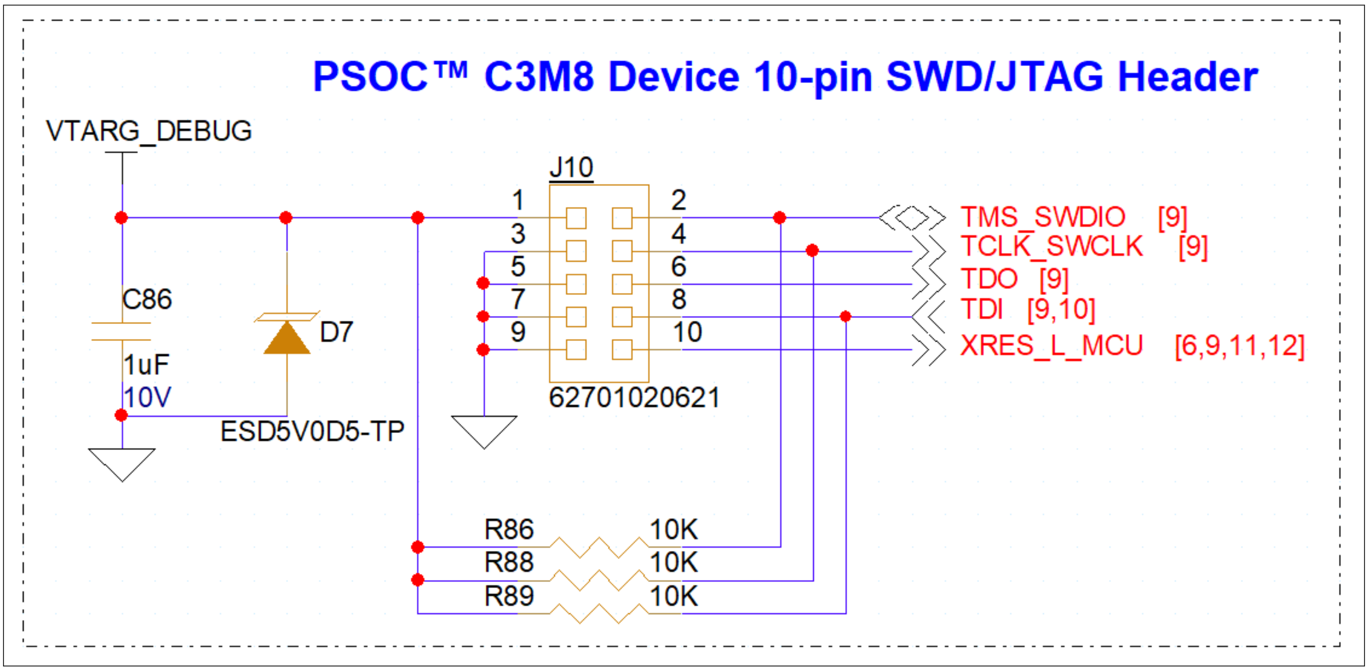

Figure 16 PSOC™ C3M8 device programming using SWD or JTAG

This 20-pin header is also connected to the SWD and JTAG signals, enabling a unified programming header concept for both programming and debugging of the PSC3M8 device.

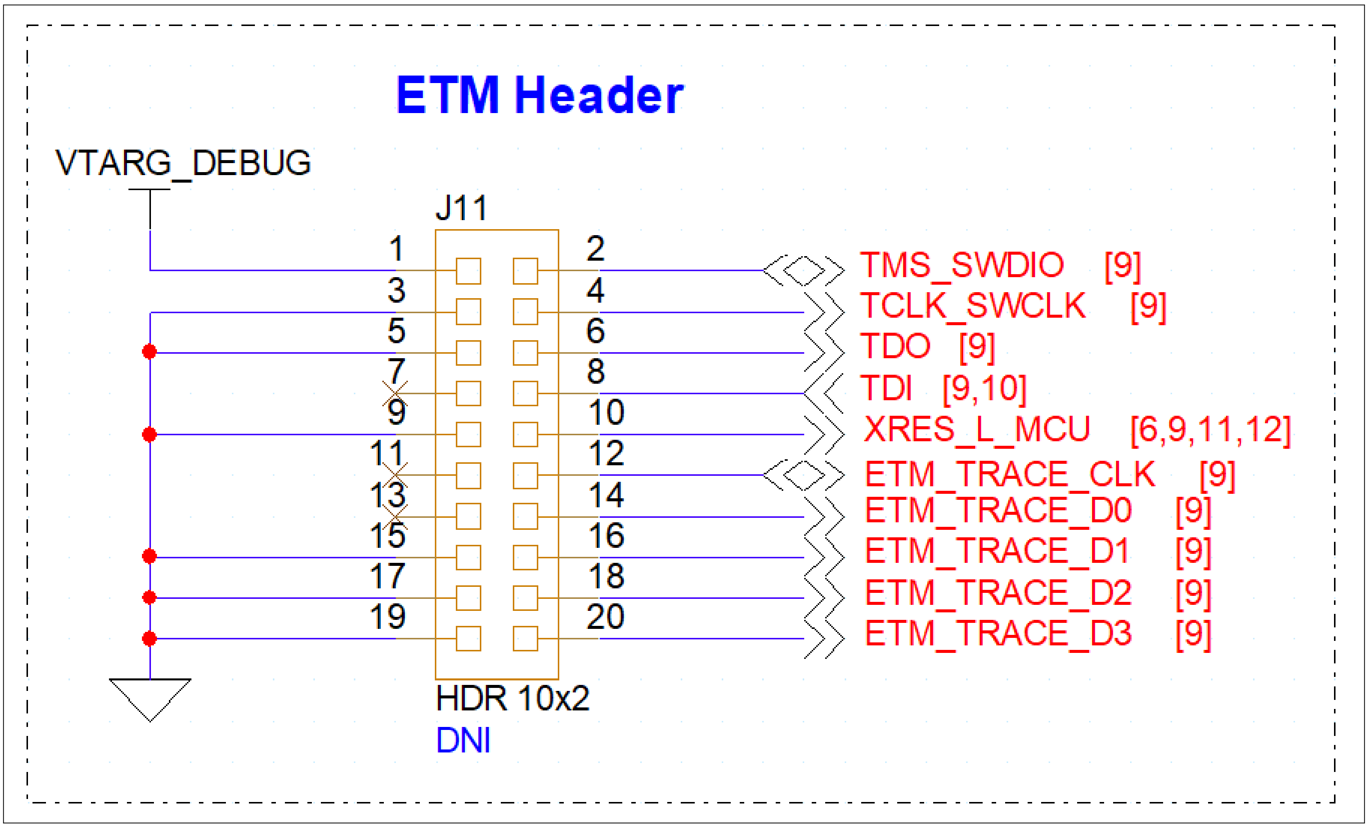

Figure 17 PSOC™ C3M8 device trace debugging header

4.2.5 Power supply system

The power supply system on this board is versatile and supports input from the following sources:

- 5 V from the onboard USB Type-C connector (J5)
- 7 V–12 V from an external power supply through the VIN barrel jack (J6) or the J1 header, which is compatible with Arduino

4.2.6 Voltage regulators

The power supply system is designed to support the following voltage configurations. The evaluation kit includes two switching regulators:

- **5 V buck-boost regulator (U9)**: Powers the CAN transceiver and the Arduino headers
- **3.3 V LDO regulator (U5)**: Provides a 3.3 V output to power all peripherals, including the MCU
- 1.2 V high-precision voltage reference regulator (U11)

On the PSC3M8GFS3PAHQ1, the analog block uses a 1.2 V VAREF voltage as a reference. This voltage can either be generated internally by the MCU or provided externally through the onboard 1.2 V VAREF reference voltage generator. The evaluation kit includes a jumper (J23) that allows users to configure the source for VAREF based on their requirements.

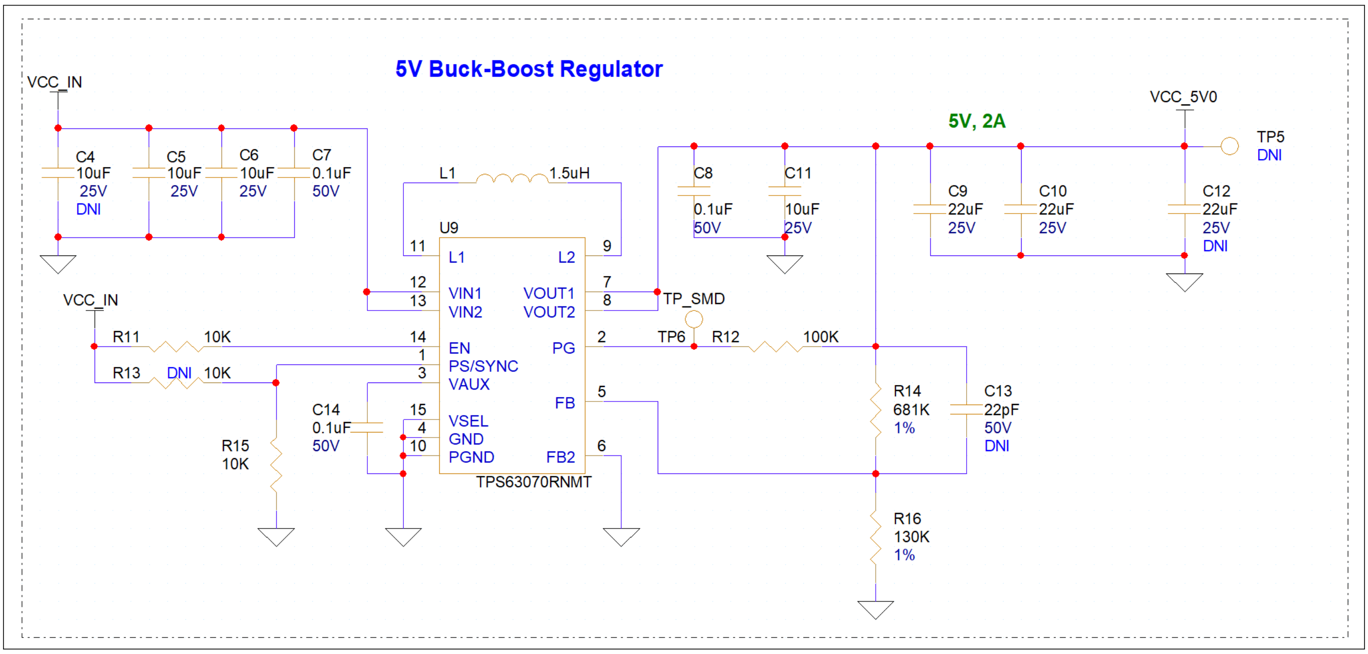

Figure 18 **5 V buck-boost regulator**

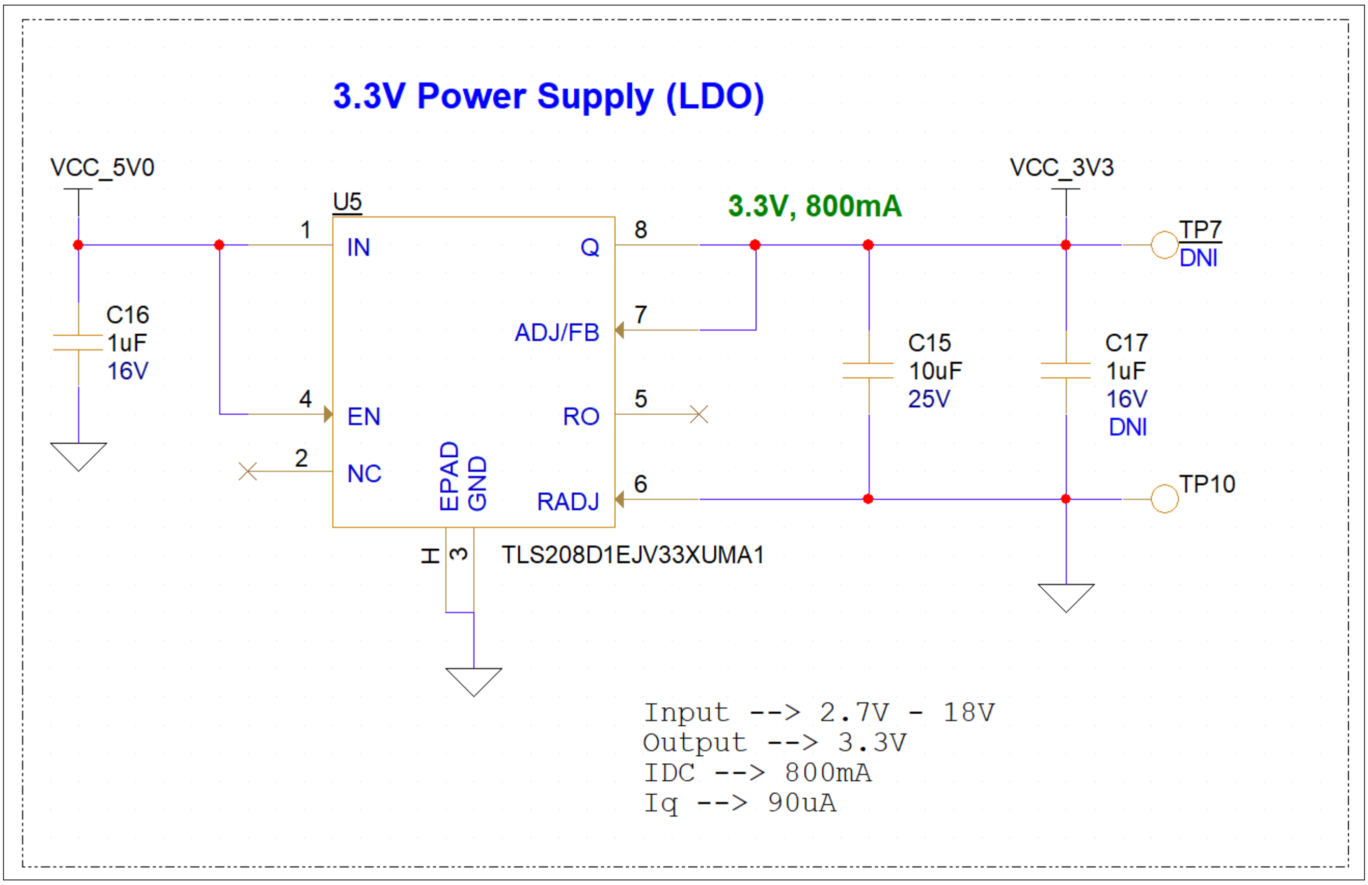

Figure 19 3.3 V fixed output LDO

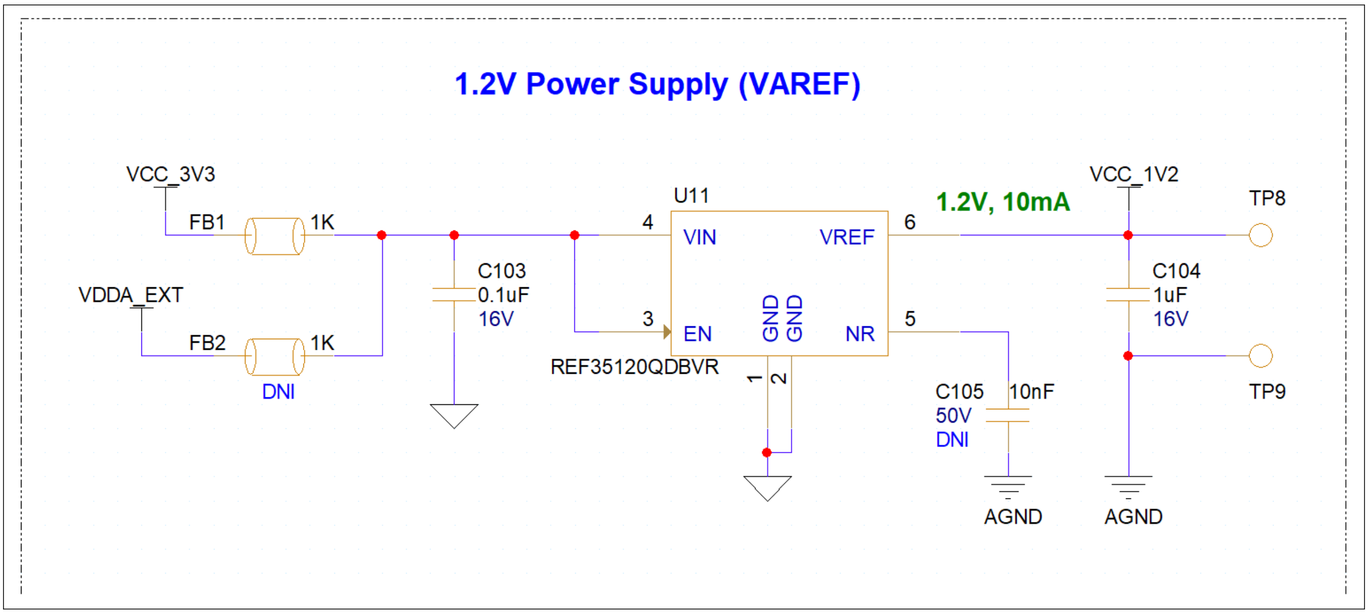

Figure 20 1.2 V precision voltage reference generator

4.2.7 Headers

4.2.7.1 Current measurement headers

The following domains have dedicated 2-pin headers to enable easy current measurement using an ammeter across the pins. The 2-pin header (J8) is specifically used to measure the current consumption of the MCU.

J8 facilitates the measurement of active current, deep sleep current, and hibernate current consumption from the MCU. This jumper isolates other peripherals from the measurement path, ensuring that only the current consumption of the MCU is measured on the evaluation kit.

4.2.7.2 VDDA voltage selection header

VDDA is, by default, connected to the P3_VDD domain through a 3-pin jumper (J22). Users can configure this jumper to supply VDDA from external sources, such as a bench power supply.

Note: *VDDA must always be connected to a 3.3 V supply, either from the onboard regulator or an external supply source. Leaving VDDA unconnected may lead to unpredictable behavior on the evaluation kit.*

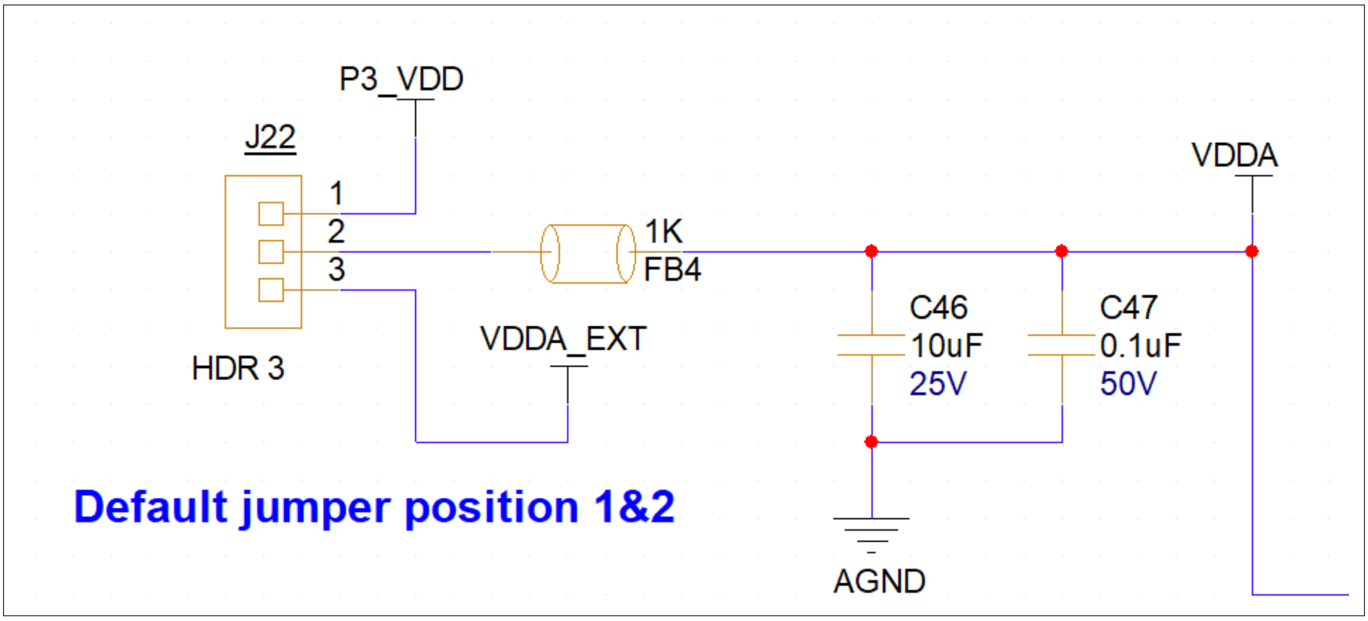

Figure 21 **VDDA voltage selection**

4.2.7.3 VREF voltage selection header

Similar to VDDA, the evaluation kit provides a voltage selection option for the VAREF supply.

Nominally, the ADC reference voltage of 1.2 V is generated within the EVK to serve as the analog reference voltage. In addition to the internal VAREF, the evaluation kit includes a provision to configure the VAREF voltage using the jumper (J23).

- Placing the jumper on pins 1 and 2 of J23 connects the onboard 1.2 V supply to the MCU
- Placing the jumper on pins 2 and 3 of J23 allows the user to supply an external 1.2 V source to the board

***Note**:* *VAREF is the fixed reference voltage for the ADC block in the MCU. When supplying VAREF from an external source, ensure that the voltage levels meet the required specifications. Any violations in the VAREF supply voltage may lead to unpredictable behavior on the evaluation kit and could cause permanent damage to the MCU.*

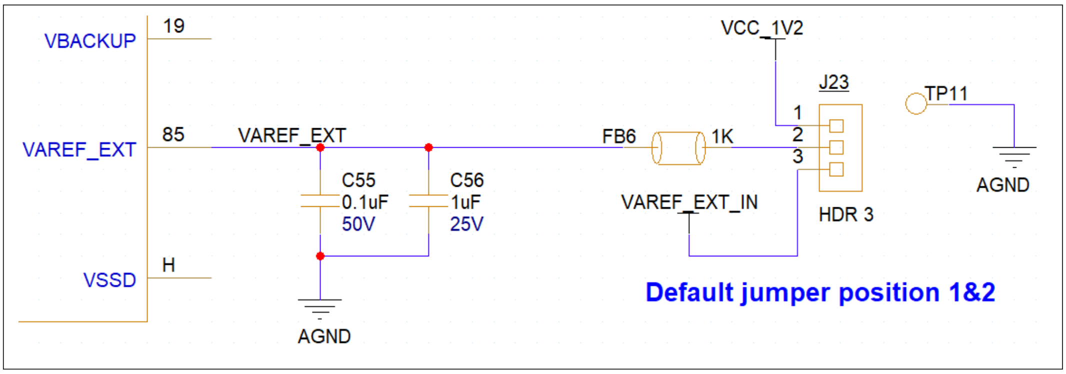

Figure 22 VAREF voltage selection

4.2.7.4 Potentiometer power header

When the MCU is operating in a low-power mode application, remove jumper J18 to avoid leakage through the potentiometer resistance. The VDDA supply is connected to the potentiometer via this jumper, providing an option for isolation during low-power measurements on the evaluation kit.

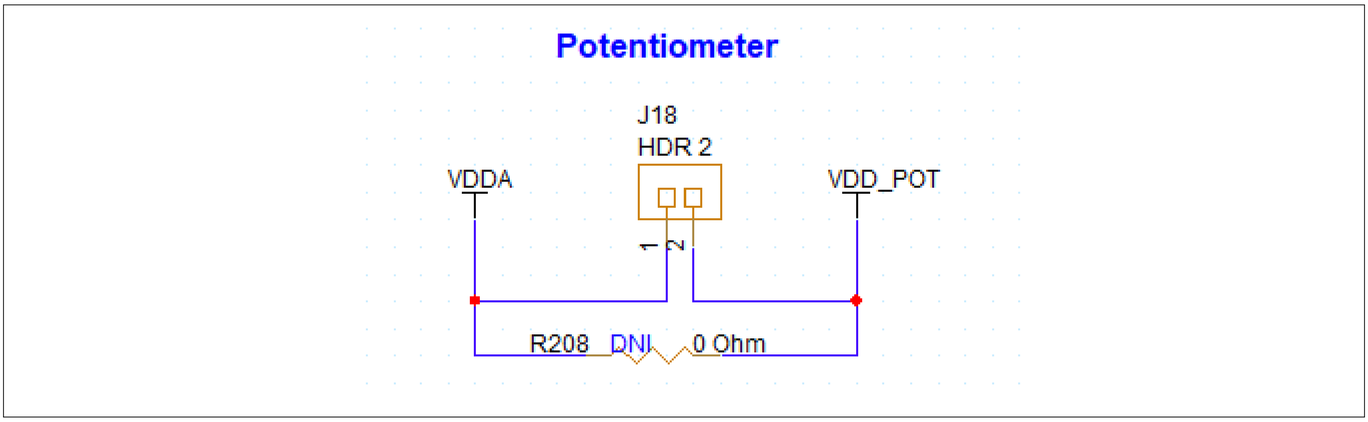

Figure 23 Power selection for potentiometer

4.2.7.5 DFU mode selection

PSC3M8GFS3PAHQ1 on the evaluation kit provides functionality to update the device firmware over UART serial communication interface.

The PSOC™ Control C3M8 MCU offers firmware upgrade capabilities through its Device Firmware Update (DFU) which can be accessed utilizing the BootROM DFU serial interface, which provides a direct pathway for upgrading the firmware.

The Device Firmware Upgrade (DFU) protocol, is a proprietary Infineon-owned serial interface feature built into the immutable BootROM, supports UART. For factory programming and upgrading images with device reset, the BootROM-based method is highly recommended. This approach leverages the immutable BootROM, which has built-in support for Infineon's custom serial DFU protocol, enabling firmware upgrades. The DFU serial interface is versatile, accommodating UART ports.

The selection of the DFU serial interface port is determined by the state of one GPIO pin i.e P8.2 for DFU enablement, which is scanned during power cycles, resets, and not during run time. As a part of Boy4 - DFU

(SCB3-UART) capability implementation, it shall require three Pins, namely UART Tx, Rx, and DFU detection (enable/disable) pins. When DFU is enabled through OEM policy and the P8.2 GPIO pin meet specific conditions (GPIO pin pulled to high (Vdd).), the BootROM launches DFU mode. To Establish the communication between host (DFU Host Tool) and PSOC™ C3 MCU target according to UART interface.

Table 2 **DFU serial interface configuration**

Pin	UART
P8.0	Rx
P8.1	Tx
P8.2	DFU Enable = HIGH(3.3V)

4.2.7.6 I²C header

P2.3 and P2.4 are used as the I²C interface between the KitProg3 debugger and the PSOC™ C3M8 MCU. The same I²C interface is also connected to header J13, allowing external devices to connect via the I²C interface. The SCL, SDA, and GND signals are available on the header.

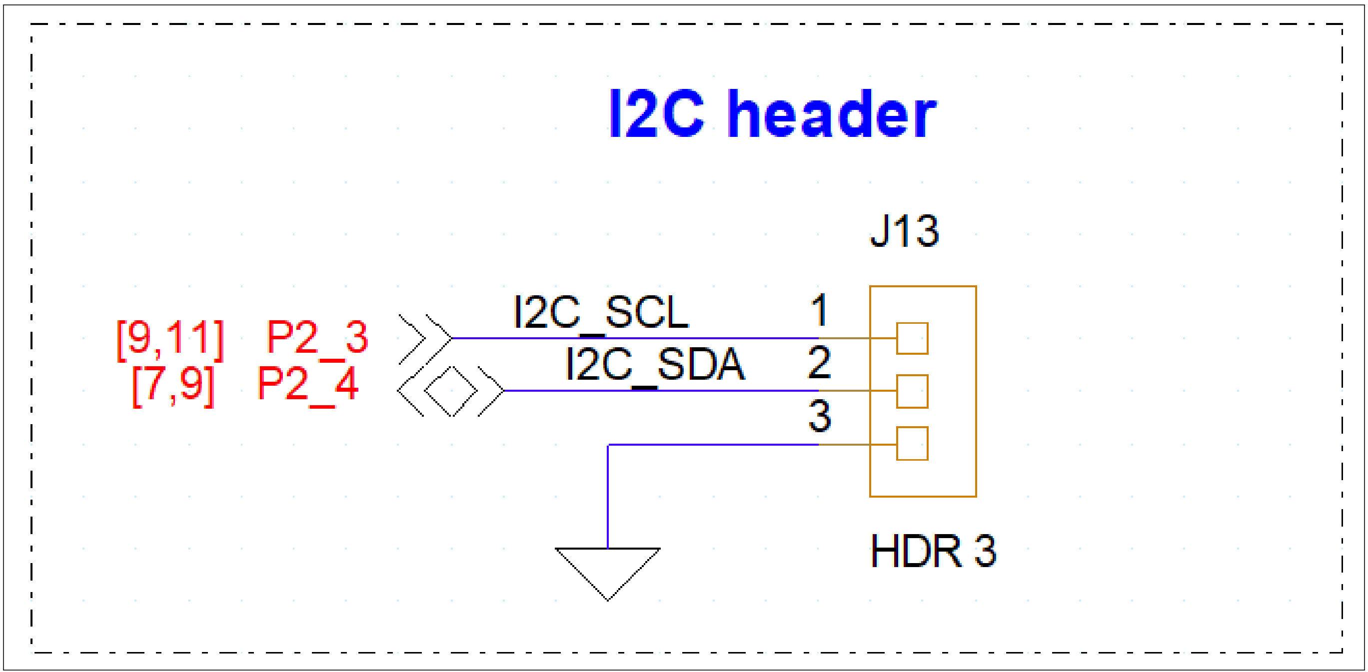

Figure 24 **I²C header for external device interface**

4.2.7.7 UART header

By default, the UART interface header is not populated on the evaluation kit. A three-pin header can be used to enable the user to connect to the UART interface through the P3.3 and P3.4 pins of the MCU. Since a dedicated UART connection exists between the KitProg3 debugger and the PSOC™ C3M8 MCU, the user must perform the necessary rework to enable UART communication via header J12.

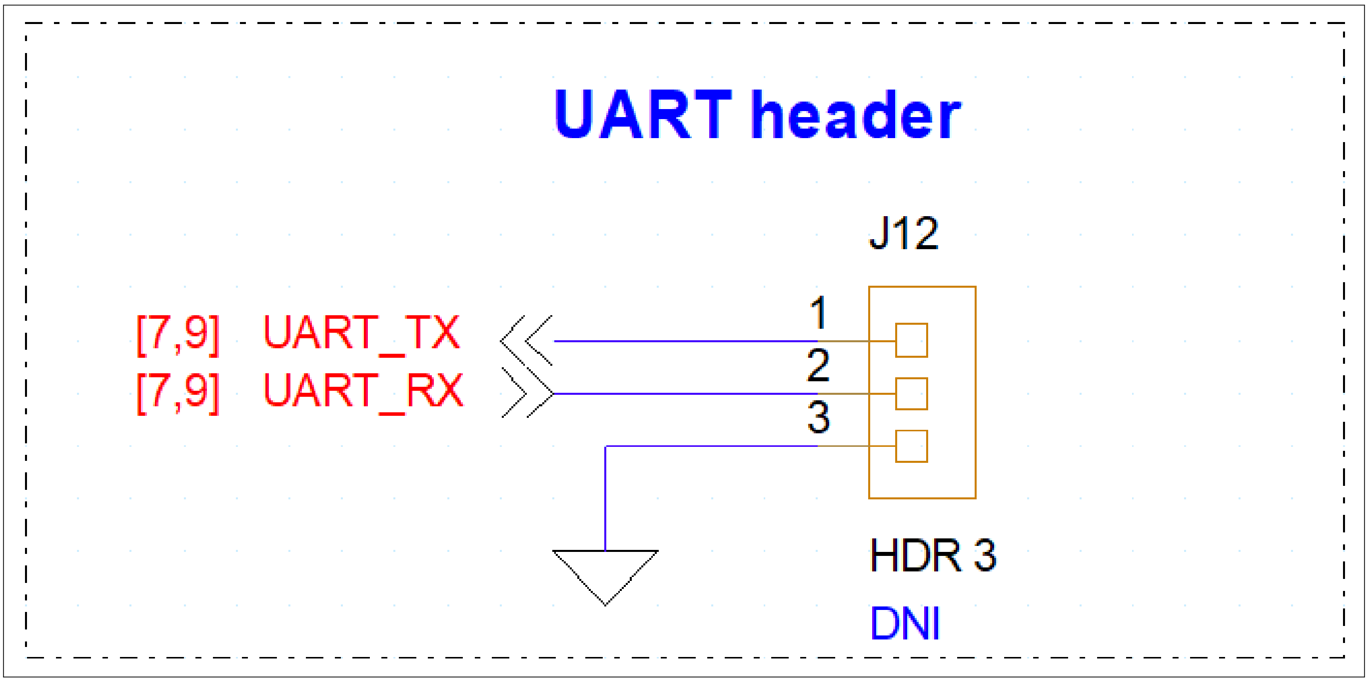

Figure 25 **UART header to interface PSOC™ C3M8 device with external device**

4.2.7.8 Arduino Uno R3 compatible header

The board features four headers (J1, J2, J3, and J4) that are compatible with the Arduino Uno R3 standard. These headers allow users to connect Arduino Uno R3-compatible shields to develop applications utilizing the shield's hardware functionalities.

Shields with I/O references that fall within the limitations of the PSOC™ C3M8 device are supported on KIT_PSC3M8_EVK.

J4 header also supports DFU (Device Firmware Update) using UART interface.

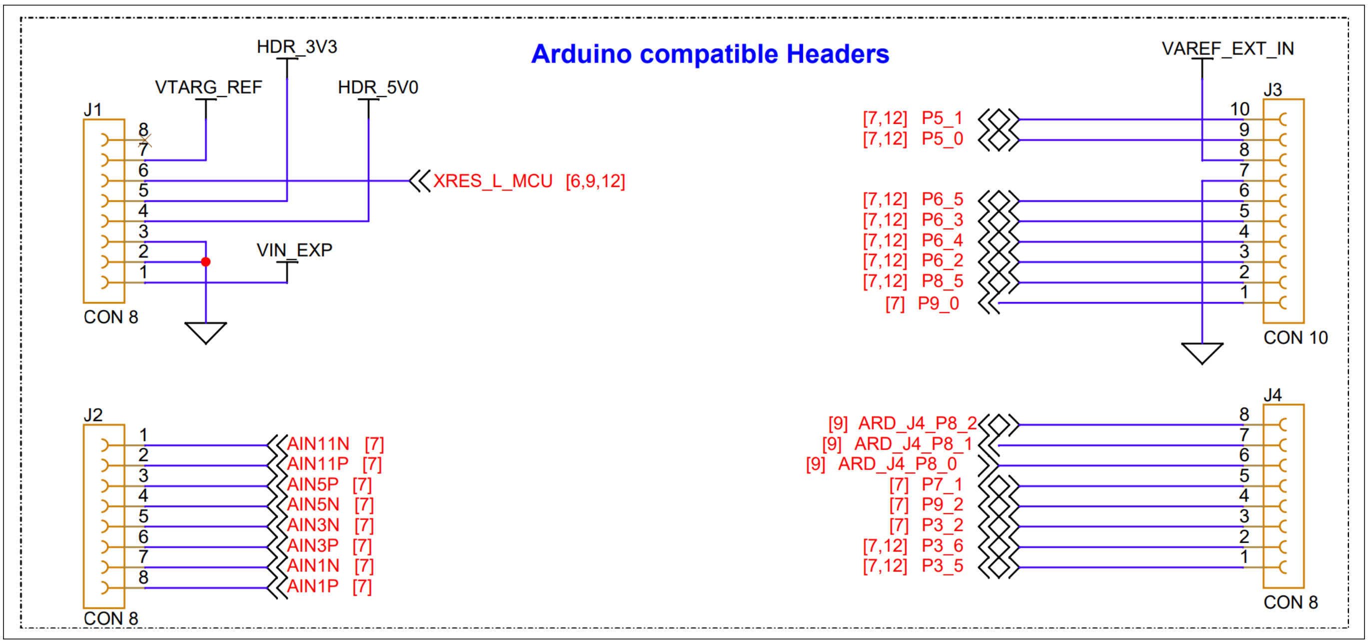

Figure 26 **Headers compatible with Arduino Uno**

4.2.7.9 I/O expansion header

KIT_PSC3M8_EVK includes two GPIO headers, J15 and J16, each featuring a 40-pin connector. Most of the signals from the PSOC™ C3M8 device are routed to these connectors.

All GPIOs and analog signals that are free or unused are connected to these headers. Pins from the PSOC™ C3M8 device that are not assigned to any specific function on the evaluation kit are, by default, connected to these headers. Additionally, other multiplexed signals are routed to these headers through zero-ohm resistors, which are not populated by default.

To validate a specific MCU signal on the expansion header, the user must perform the necessary rework on the evaluation kit by populating the corresponding resistor.

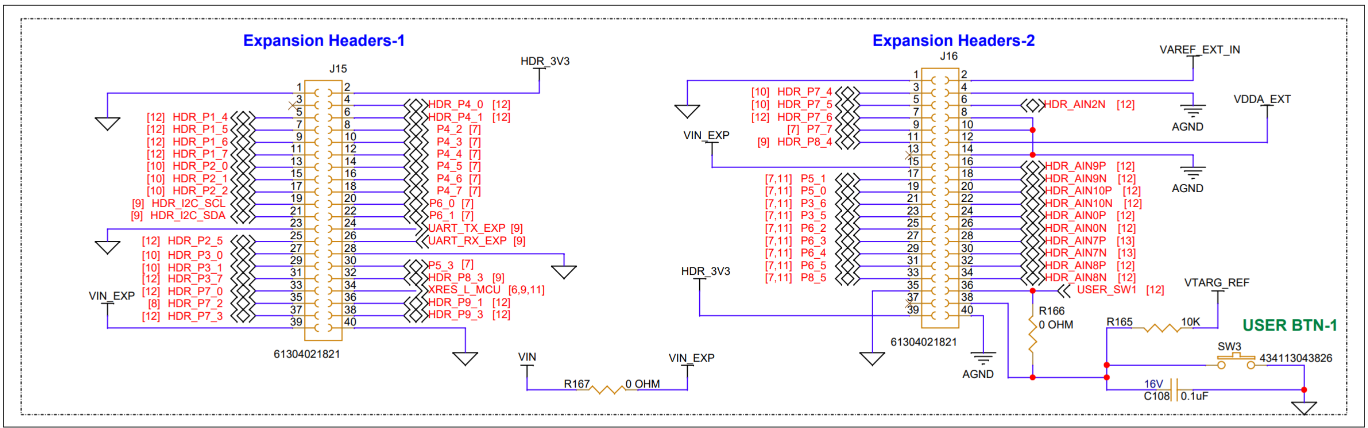

Figure 27 **Expansion headers to access PSOC™ C3M8 signals**

4.2.7.10 DAC-ADC interface header

The evaluation kit is populated with an 8-pin receptacle header, J9, which provides access to DAC and ADC pins for evaluation purposes.

DAC CH0 and DAC CH1 from the PSOC™ C3M8 device are connected to pins J9.1 and J9.3, respectively. The DAC signals from the PSOC™ C3M8 device are routed through an amplifier circuit with a gain of two. The DAC output from the PSOC™ C3M8 device is configurable in two modes: either 1.2 V or VDDA/2. To align the output with the device voltage, a 2x gain is applied using an operational amplifier. The amplified DAC output is then routed to header J9, providing user access.

Additionally, header J9 includes ADC signals routed from the PSOC™ C3M8 device, which can be looped back for interaction between the ADC and DAC signals on the device.

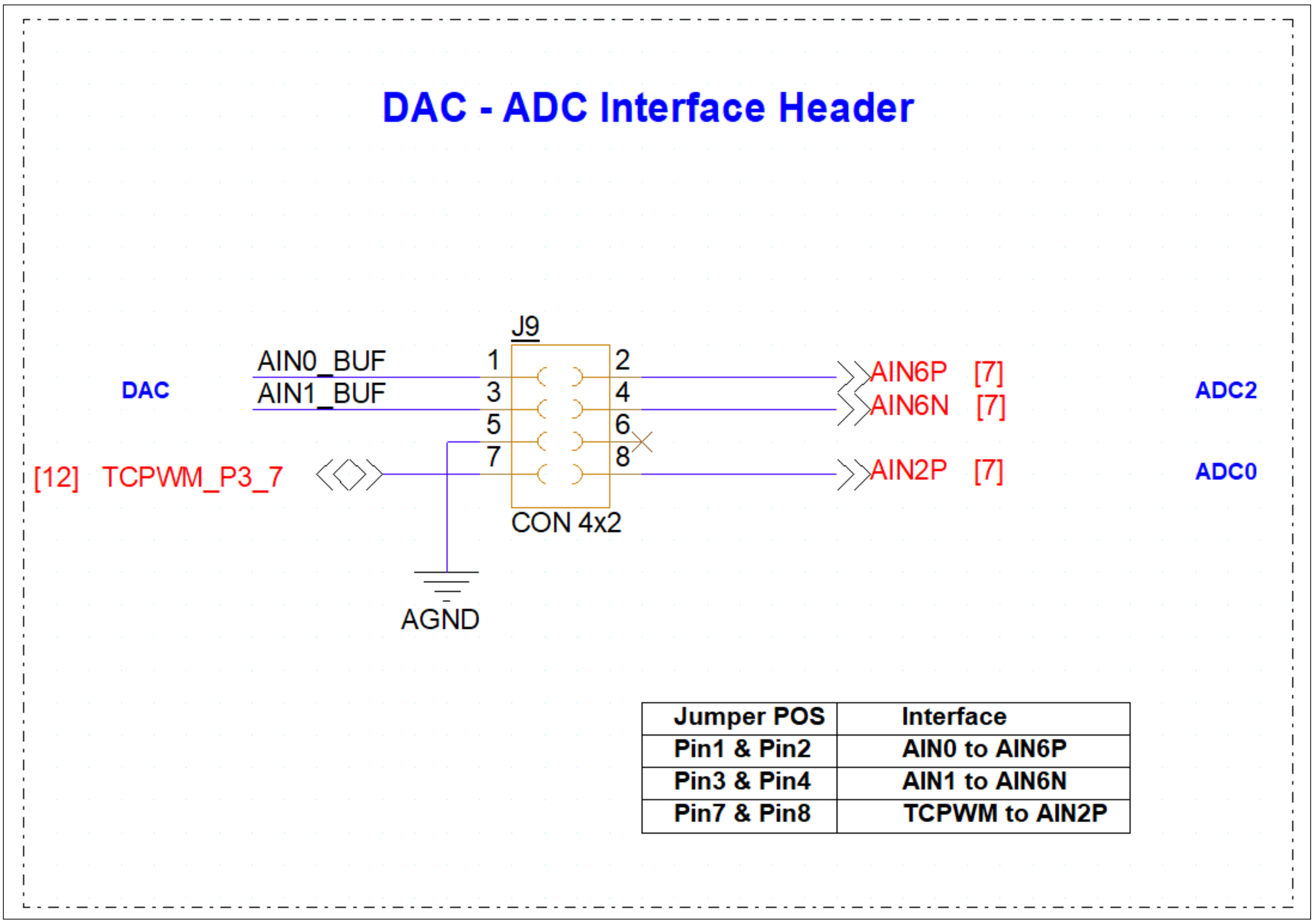

Figure 28 DAC and ADC interface header

4.2.8 CAN FD transceiver

The evaluation kit includes a CAN FD transceiver with its output routed to a terminal connector. The TLE9251V is a high-speed CAN transceiver that operates as an interface between the CAN controller and the physical bus medium. A high-speed CAN network is a two-wire, differential network capable of supporting data transmission rates of up to 5 Mbps.

Single-ended CAN signals, TxD and RxD, from the PSOC™ C3M8 device are connected to Infineon's CAN FD transceiver. The transceiver's output is routed to a terminal block, providing CANL, CANH, and GND signals for connecting external devices that support the CAN interface. The voltage level for the digital input (TxD) and digital output (RxD) is determined by the power supply level at the VIO pin.

The signal levels on the logic pins (STB, TxD, and RxD) are compatible with MCUs that have a 3.3 V I/O supply.

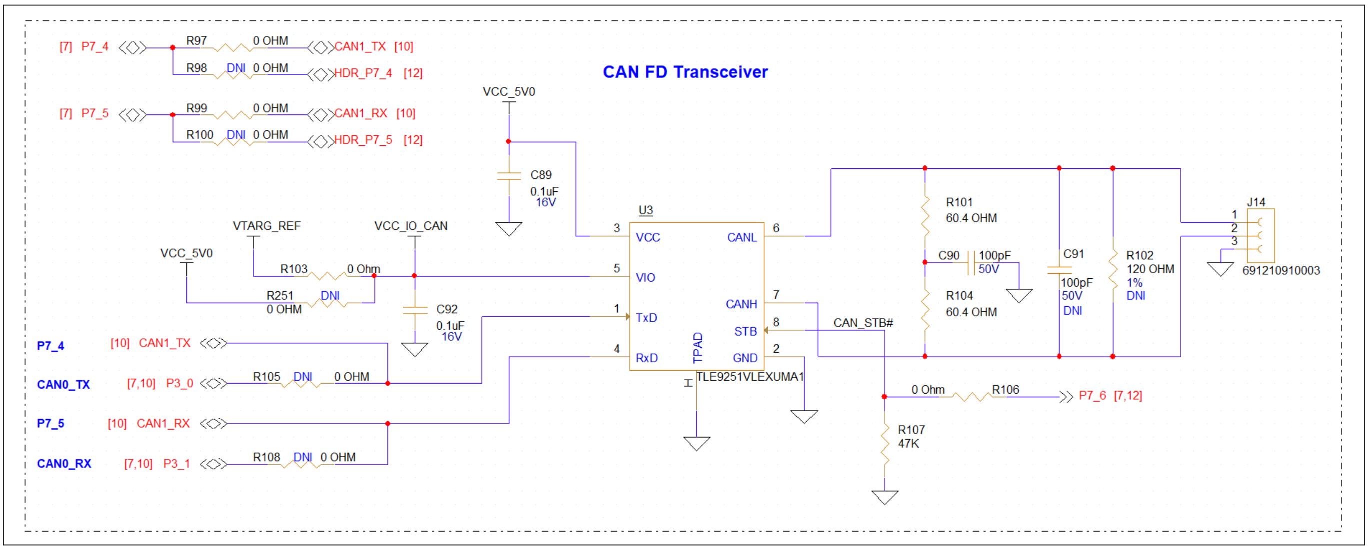

Figure 29 **CAN FD transceiver with terminal connector**

4.2.9 SPI NOR flash

A SPI NOR flash memory is populated on the evaluation kit to evaluate the high-speed SPI interface of the PSOC™ C3M8 device. The SCB5 block is connected by default to the SPI flash interface, enabling memory operations at a frequency of 50 MHz.

Alternatively, the signals used for the SPI interface are routed to the expansion header. To use these SPI signals as GPIOs on the expansion header, the user must perform the necessary rework on the evaluation kit.

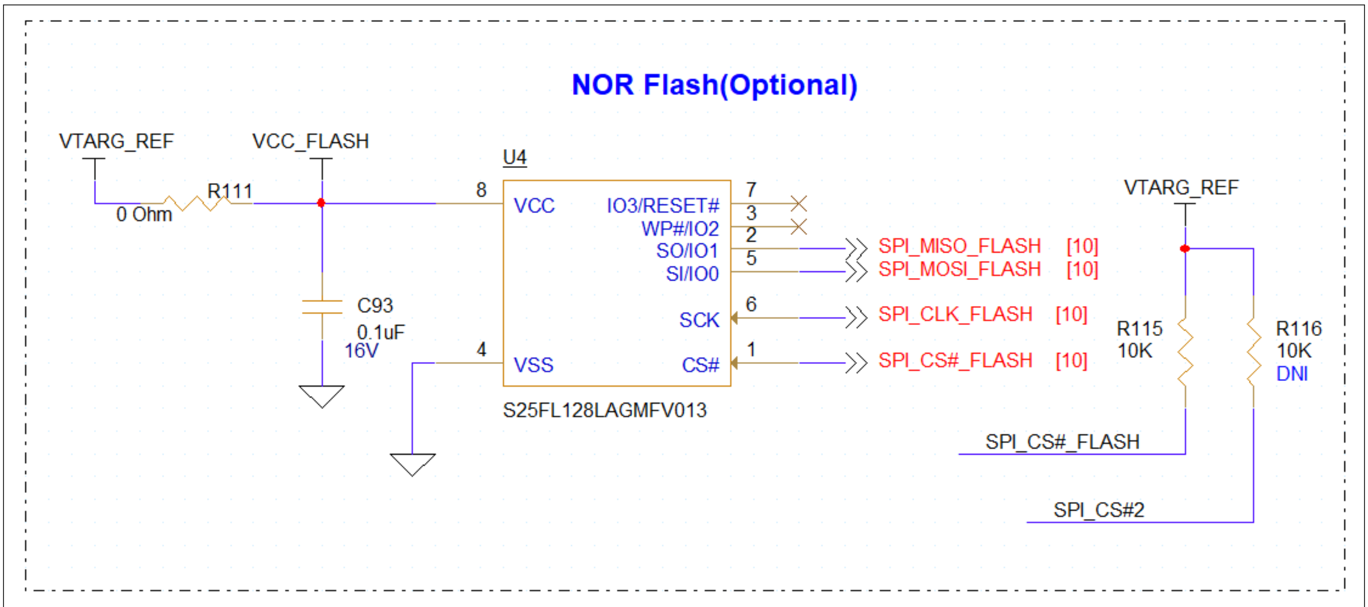

Figure 30 **SPI NOR flash memory interface**

4.2.10 Current sensing

The evaluation kit is populated with a shunt resistor (R38) of 100 mΩ, connected in series with the GND path of the PSOC™ C3M8 device.

The EPAD pin of the C3M8 device is connected through the 100 mΩ resistor, and two ADC pins, AIN4P and AIN4N, are differentially routed from the MCU to sense the current through the shunt. The device current is measured across the resistor as an IR drop, which is read by the differential input of the PSOC™ C3M8 device.

Note: *To perform current measurements using the shunt resistor, the MCU must run an application that consumes a minimum current of 40 mA or more to effectively evaluate the ADC output.*

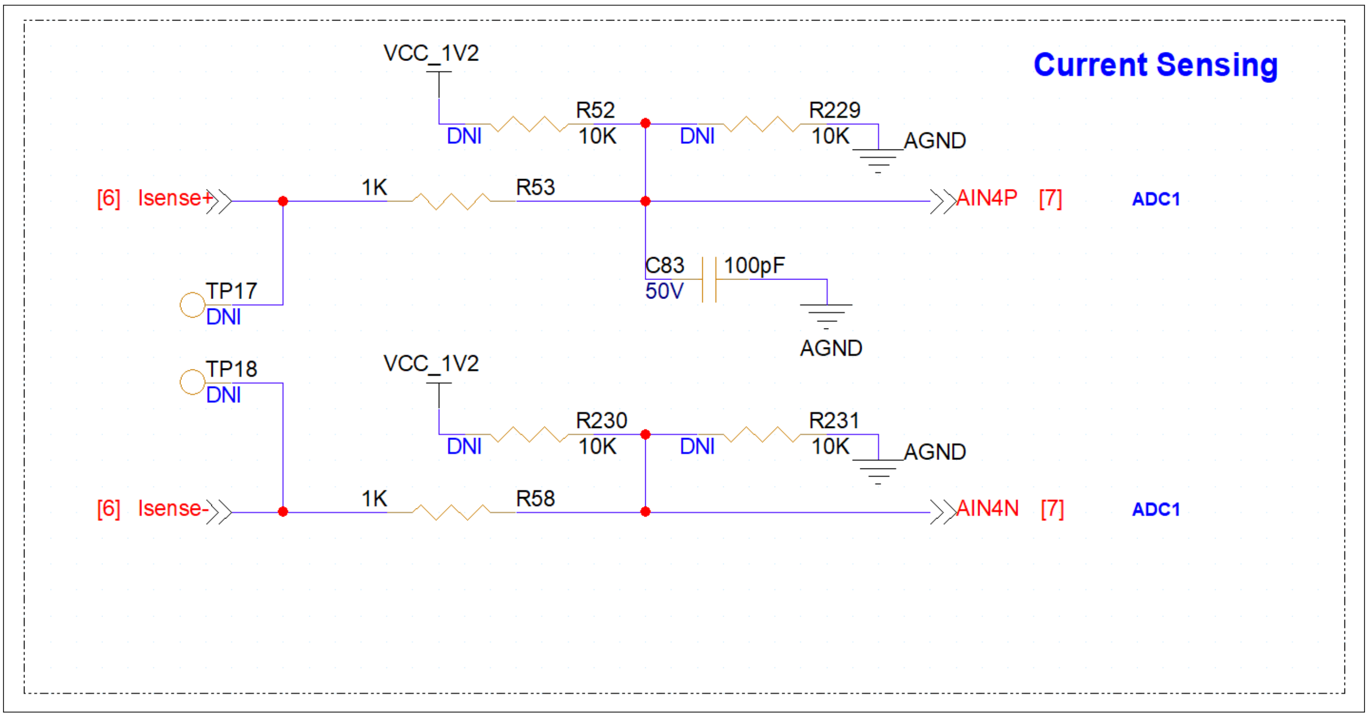

Figure 31 **PSOC™ C3M8 current sensing using internal gain amplifier**

4.2.11 Voltage sensing

A voltage divider, consisting of resistors R54 and R56, outputs a signal that is connected to AIN9P, with GND connected to AIN9N, to sense a fixed output voltage through the ADC.

The fixed voltage is measured by the ADC based on the voltage divider ratio and the VTARG_REF voltage on the evaluation kit. By default, the readings are approximately 0.5 V, based on the existing voltage input and the divider network.

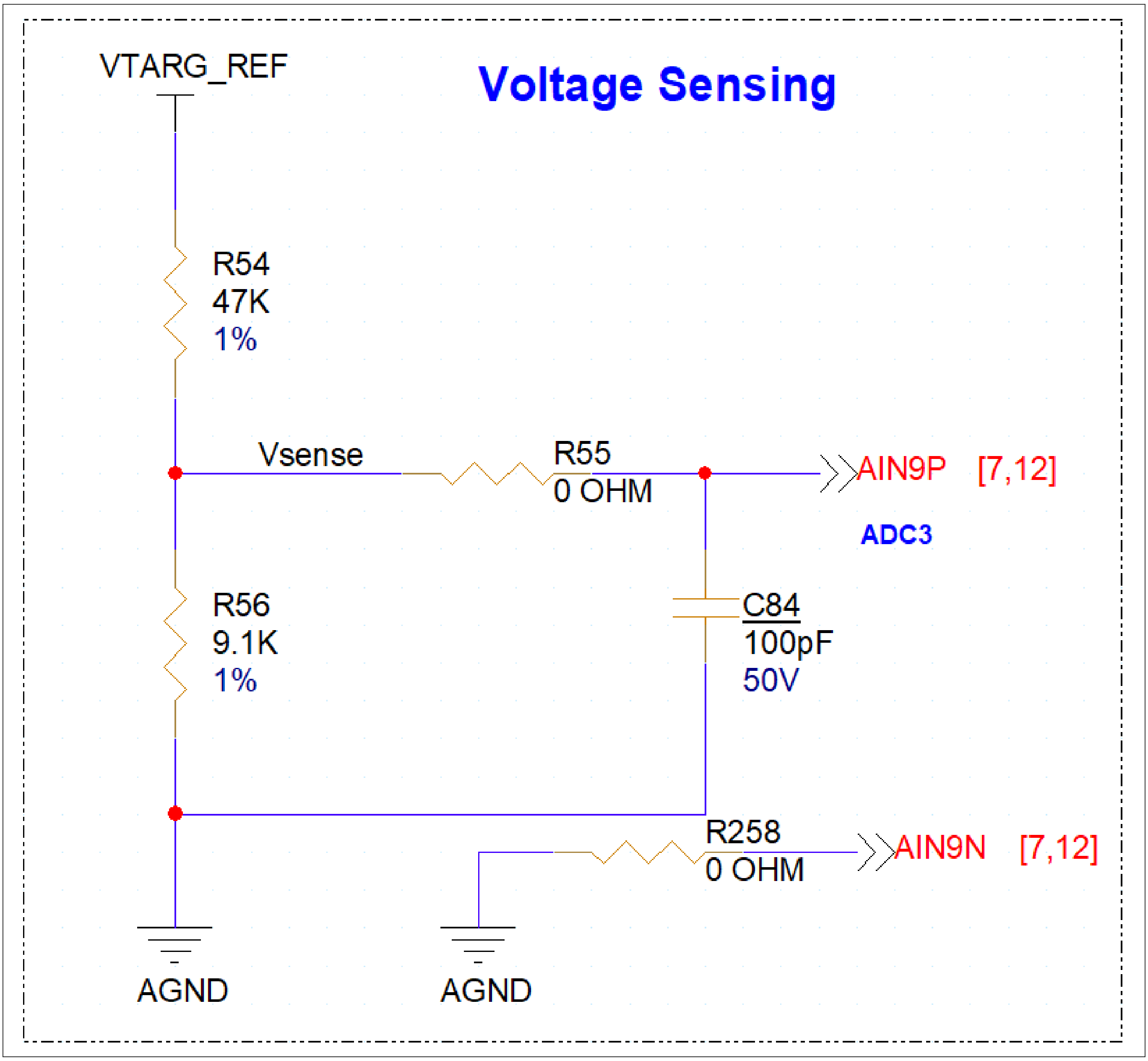

Figure 32 Voltage sensing

4.2.12 Temperature sensing

The evaluation kit is populated with an I²C-based temperature sensor to evaluate the digital capabilities of the PSOC™ C3M8 device.

The I²C interface used between KP3 and the MCU (P2.3 and P2.4) is also utilized to connect the temperature sensor on the evaluation kit. The PSOC™ C3M8 device reads digital data over the I²C interface from the temperature sensor to sense the temperature.

The I²C address of the temperature sensor is configurable between two values. By default, the SDO pin of the temperature sensor is left floating, resulting in an I²C address of 0x77. A provision is provided on the evaluation kit to load a pull-down resistor (R93), which configures the I²C address to 0x76.

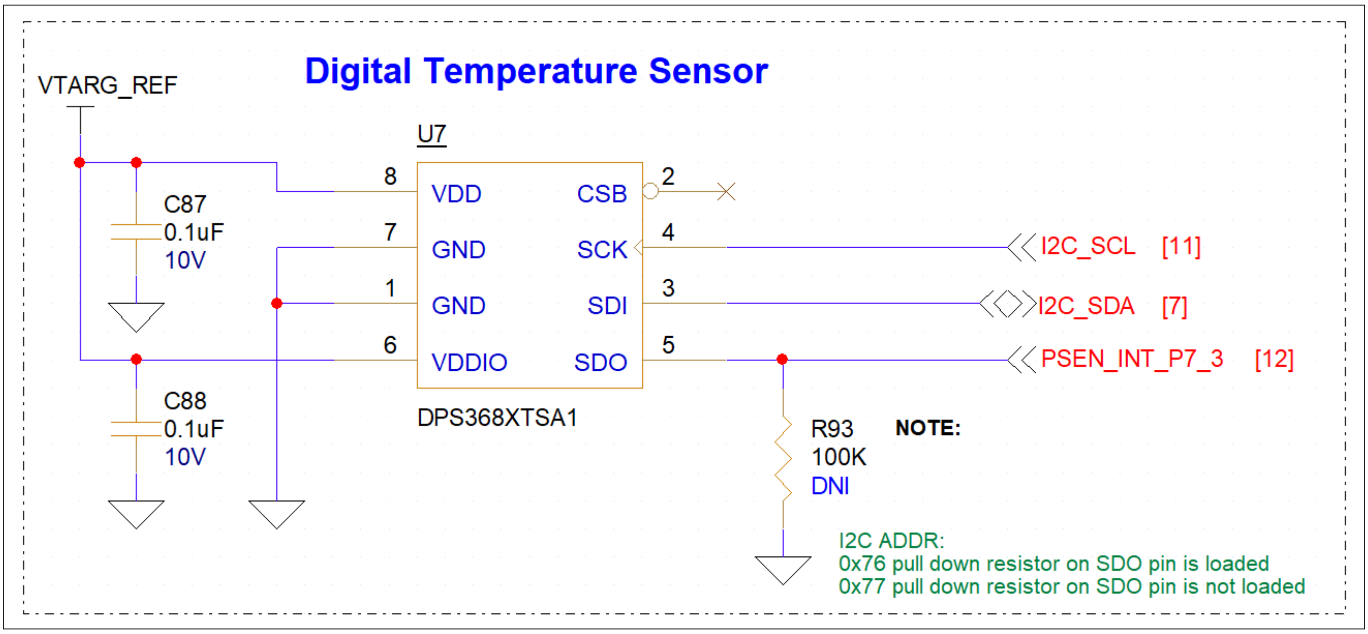

Figure 33 **Digital temperature sensor**

4.2.13 LEDs

The evaluation kit is equipped with multiple LEDs on the board, designed for power indication, status monitoring, and user application purposes.

4.2.13.1 Power indication LED

Power LED: Indicates the power status of the evaluation kit. The Power LED illuminates when the evaluation kit is powered, either through the USB connection or the VIN supply source.

Jumper requirement: Jumper J8 must be populated on the evaluation kit to power the MCU, which ensures that the 3.3V from the regulator is powering up the PSC3M8 device.

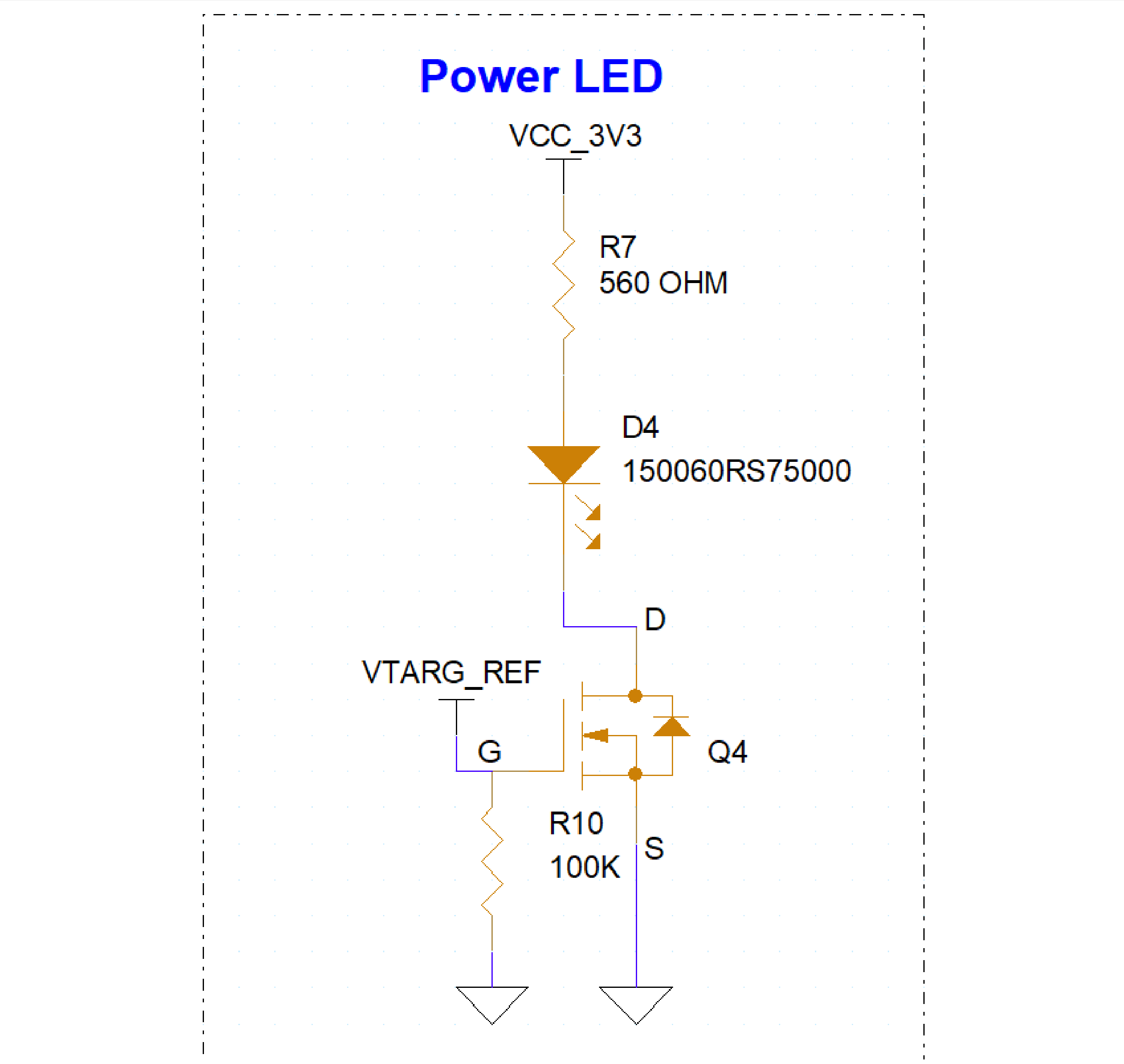

Figure 34 **Power LED**

4.2.13.2 KitProg3 status LED

Status LED: The KitProg3 debugger is equipped with a status LED that indicates the status of the on-board debugger and its mode of operation. By default, LED1 (amber) illuminates on the evaluation kit as soon as the kit is powered ON using USB power.

Mode indication: The status of LED1 changes based on the mode of operation of the KitProg3 debugger, such as Bulk or HID mode. The mode is indicated by the blinking pattern of LED1, which changes when the switch (SW2) is pressed.

External power limitation: Powering the evaluation kit with an external supply through the DC jack will not power the KitProg3 debugger.

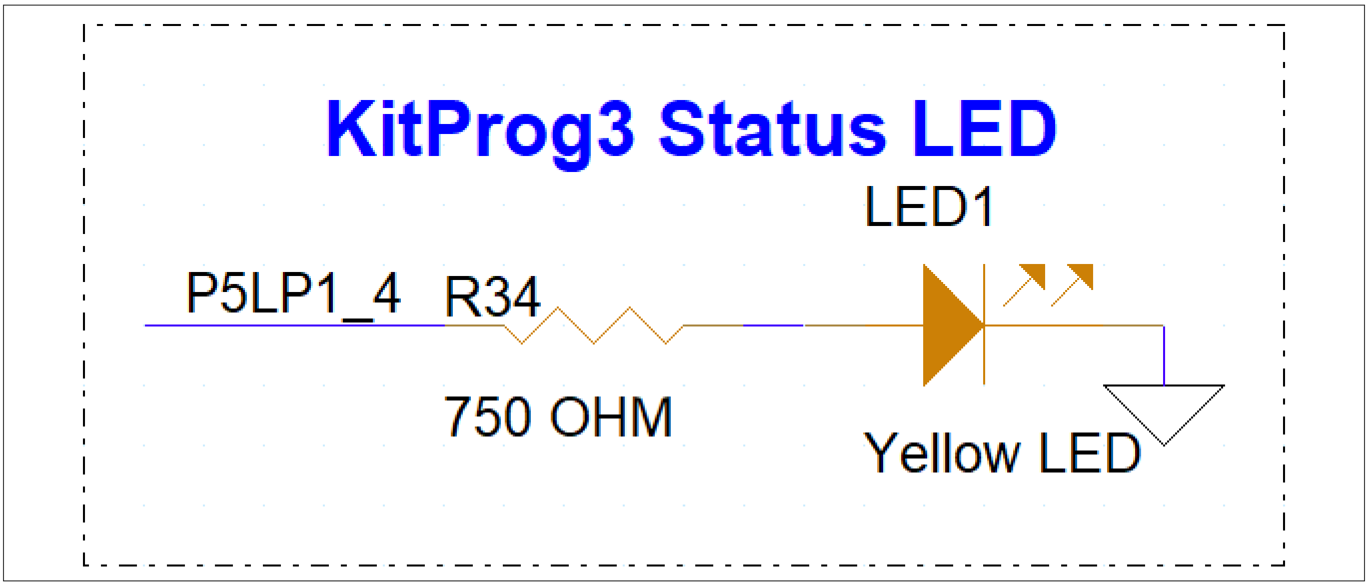

Figure 35 **KitProg3 status LED**

4.2.13.3 User LEDs

The evaluation kit is populated with six user LEDs, consisting of three green LEDs and three amber LEDs. These LEDs can be used for indication purposes based on the specific application use case. All user LEDs are active-low, meaning that driving the respective MCU pin low will turn ON the corresponding LED.

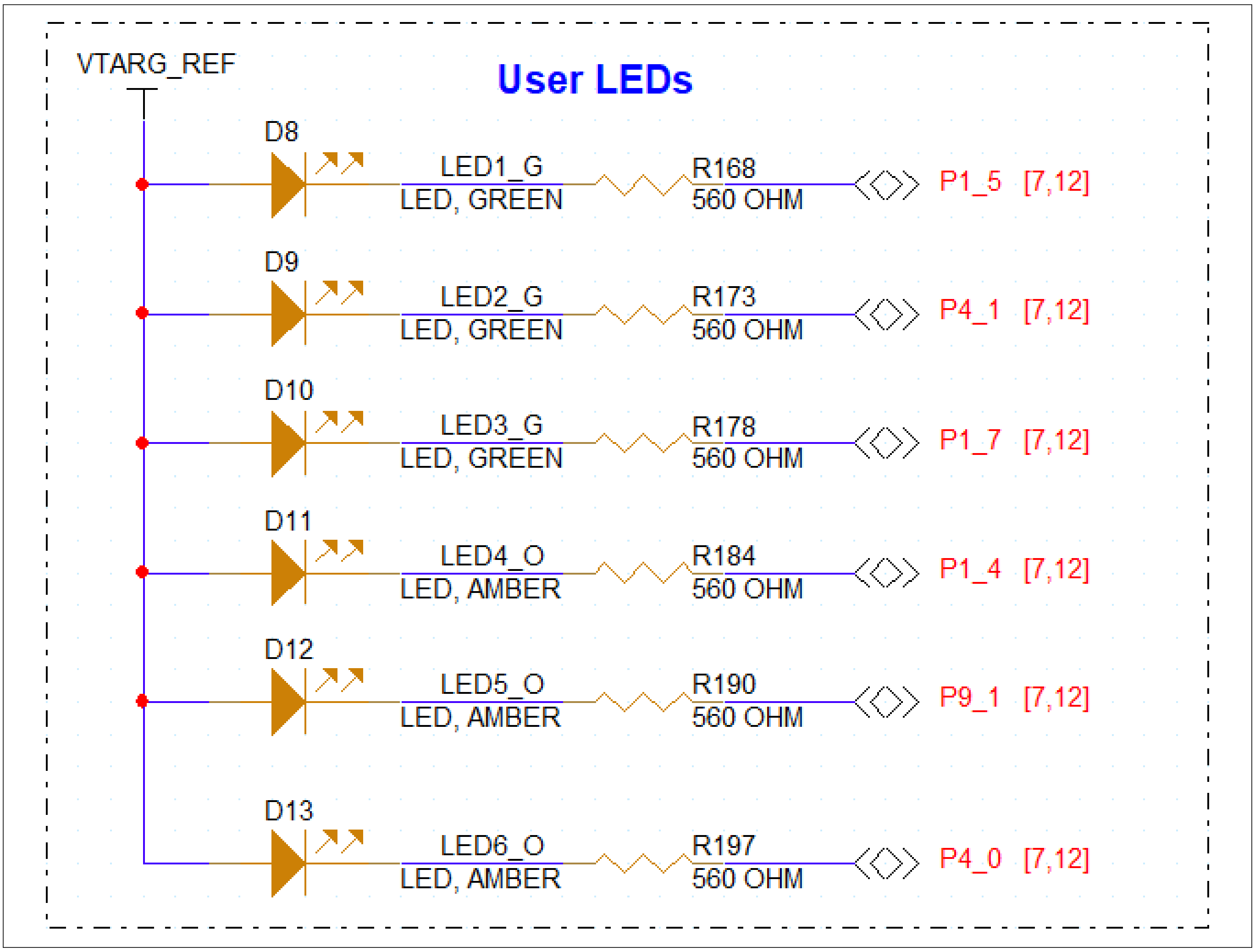

Figure 36 **User LEDs**

4.2.14 Switches

The board is equipped with the following switches:

- Reset switch (SW1)
- KitProg3 mode selection switch (SW2)
- User switches (SW3 and SW4)

4.2.14.1 Reset switch

The reset switch is connected to the XRES pin of the PSOC™ C3M8 device and is used to reset the device.

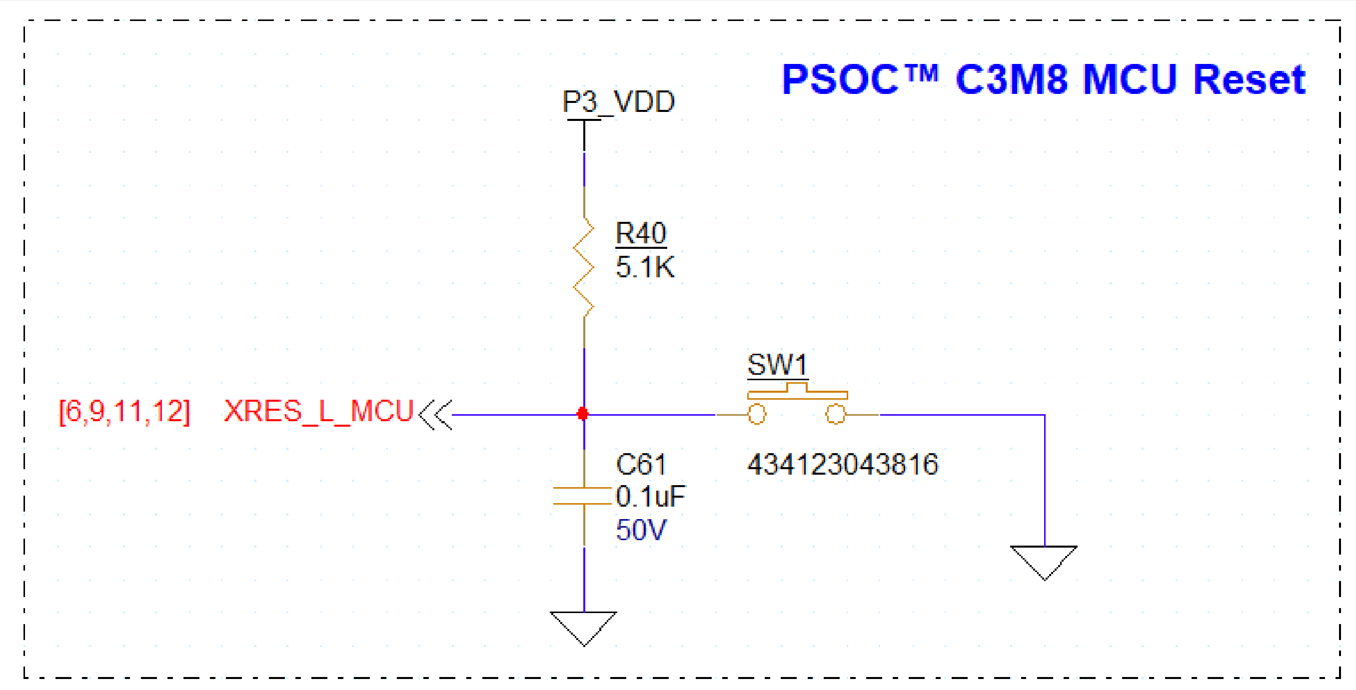

Figure 37 **Reset switch**

4.2.14.2 KitProg3 mode selection switch

A KitProg3 mode selection switch (SW2) is connected to the PSOC™ 5LP device for programming mode selection. This button's function is reserved for future use. For more details, see the KitProg3 user guide.

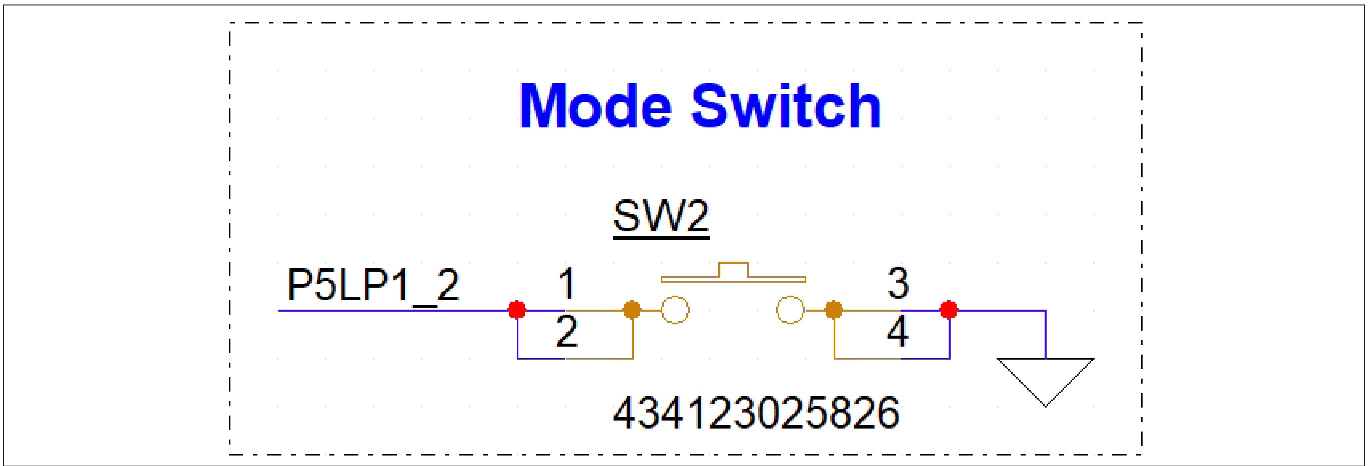

Figure 38 **Mode switch**

4.2.14.3 User switches

A user-controllable push button (SW3) is connected to the P5.2 port of the PSOC™ C3M8 device. This switch can be used to connect to any I/O pin through a jumper wire from J16.36 or J16.38.

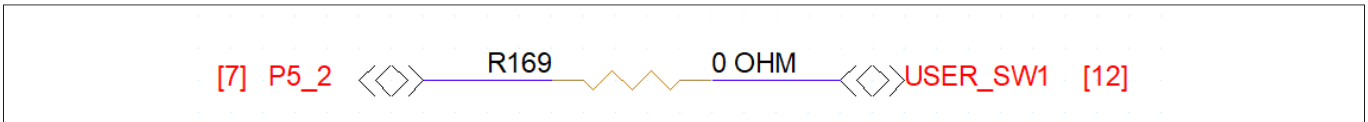

Figure 39 **User button 1**

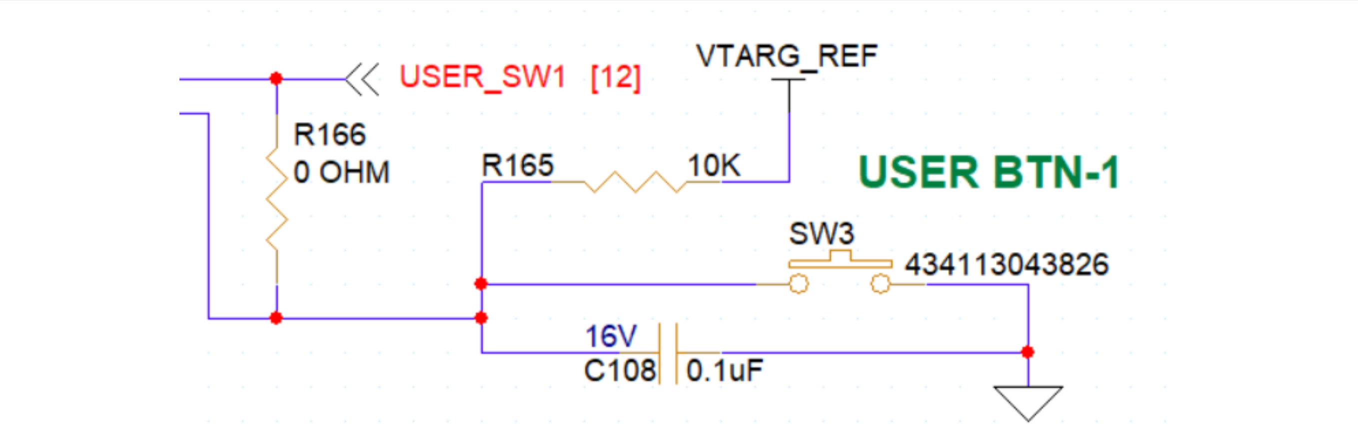

Figure 40 **Connection on header J16.36 and J16.38**

A user-controllable push button (SW4) is connected to the P7.0 pin of the MCU. This switch can be used in low-power applications to wake the MCU from Hibernate mode.

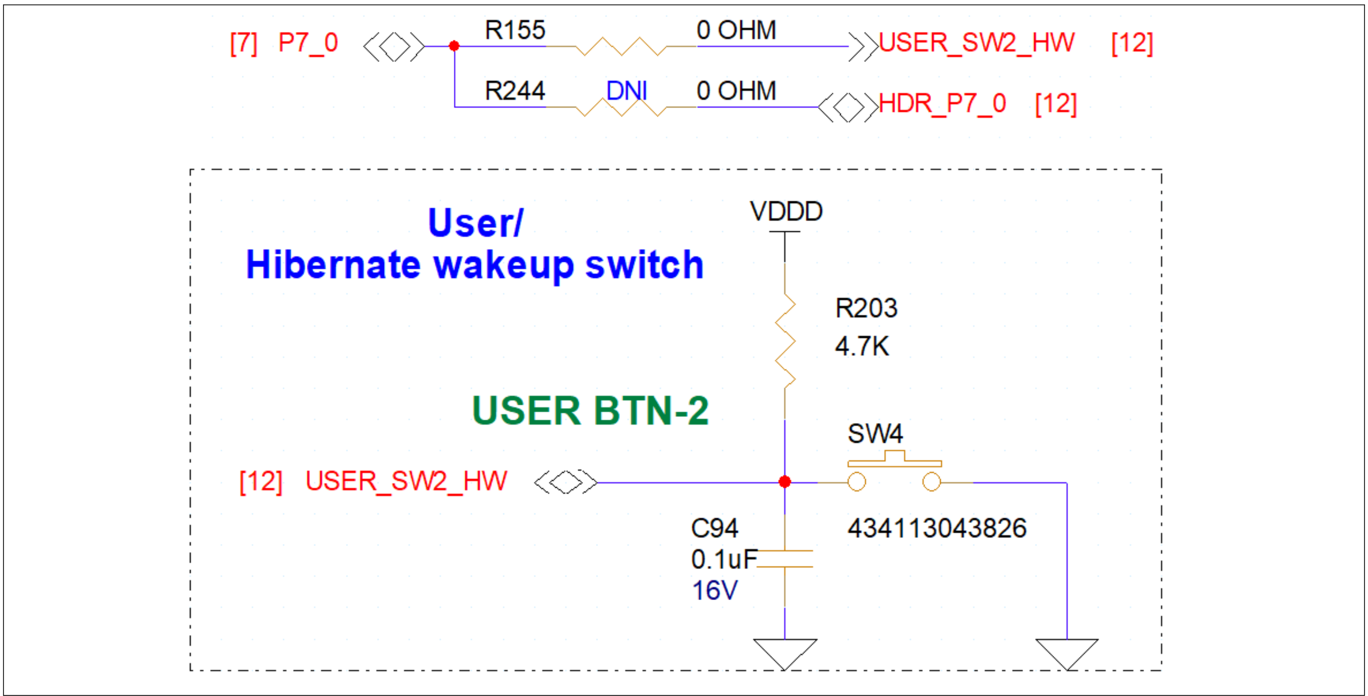

Figure 41 **User button 2**

4.2.15 Potentiometers

The evaluation kit is populated with three potentiometers. Potentiometer R212 (10 kΩ) is connected to the AIN0P pin of the MCU, and potentiometer R221 (10 kΩ) is connected to the AIN0N pin of the MCU. Additionally, a high-impedance potentiometer, R262 (500 kΩ), is connected to the AIN10P pin of the PSOC™ C3M8 device.

The fixed ends of the potentiometers are connected to VDDA (via VDD_POT through jumper J18) and GND, which may contribute to leakage current on the P3_VDD supply. To prevent this, remove jumper J18 to disconnect power from the potentiometers when measuring the P3_VDD current.

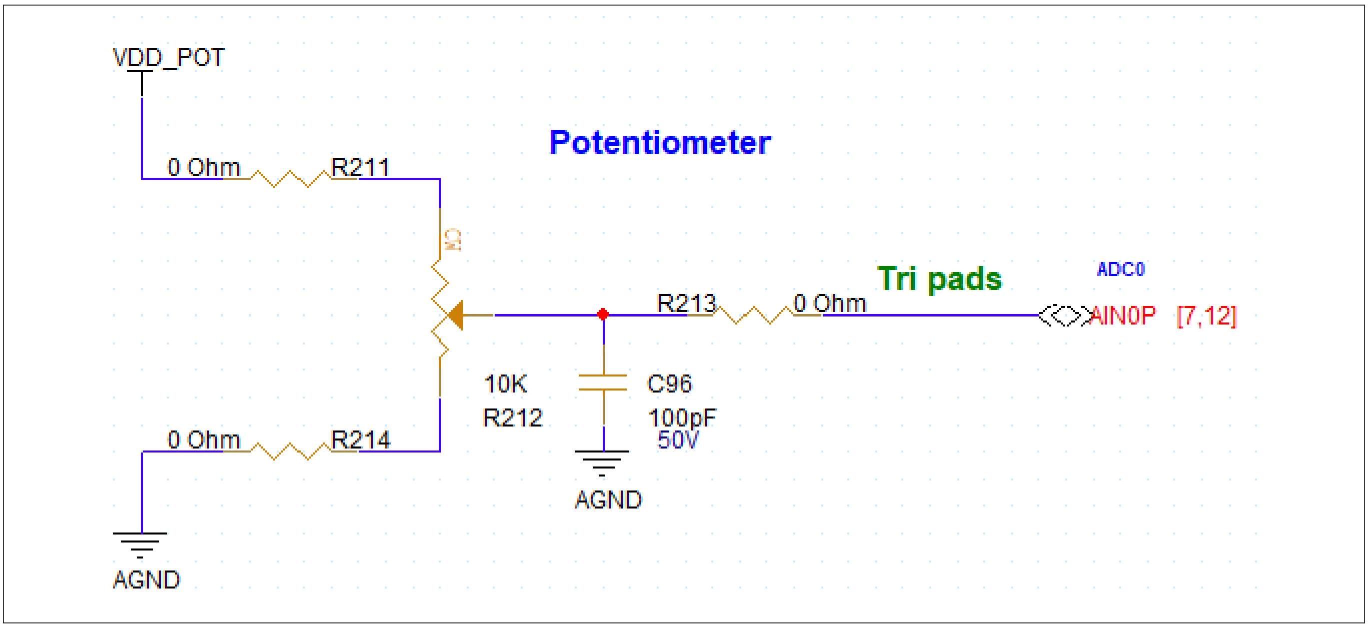

Figure 42 **Potentiometer 1 - 10K**

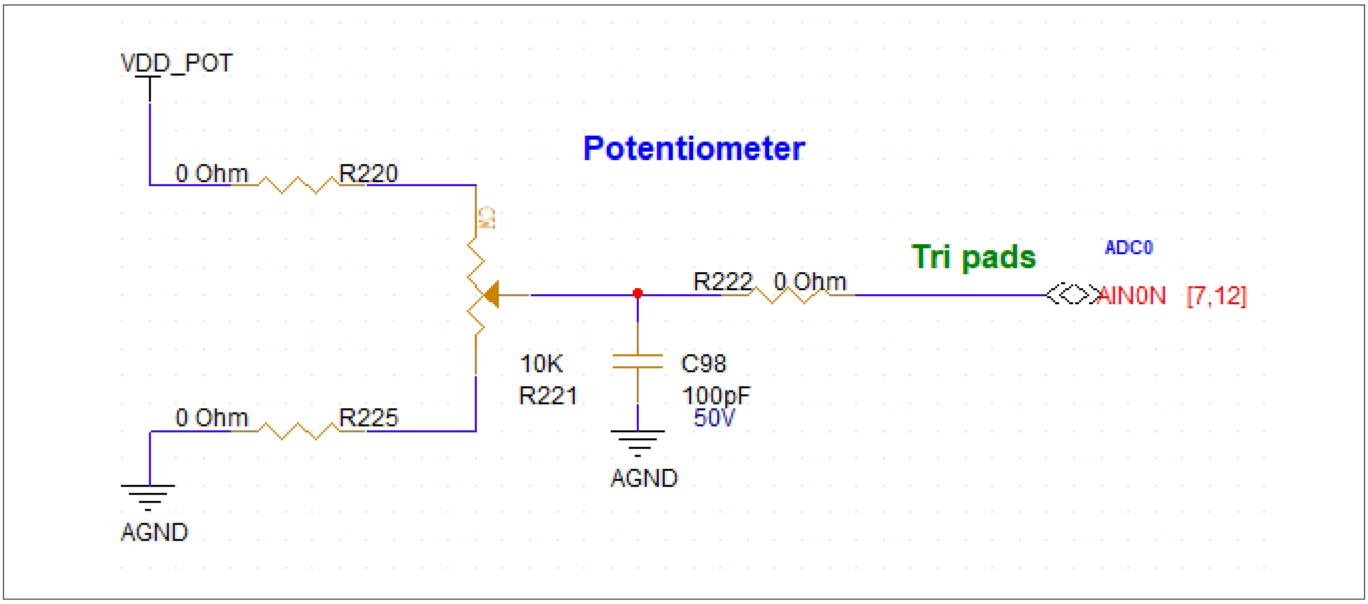

Figure 43 **Potentiometer 2 - 10K**

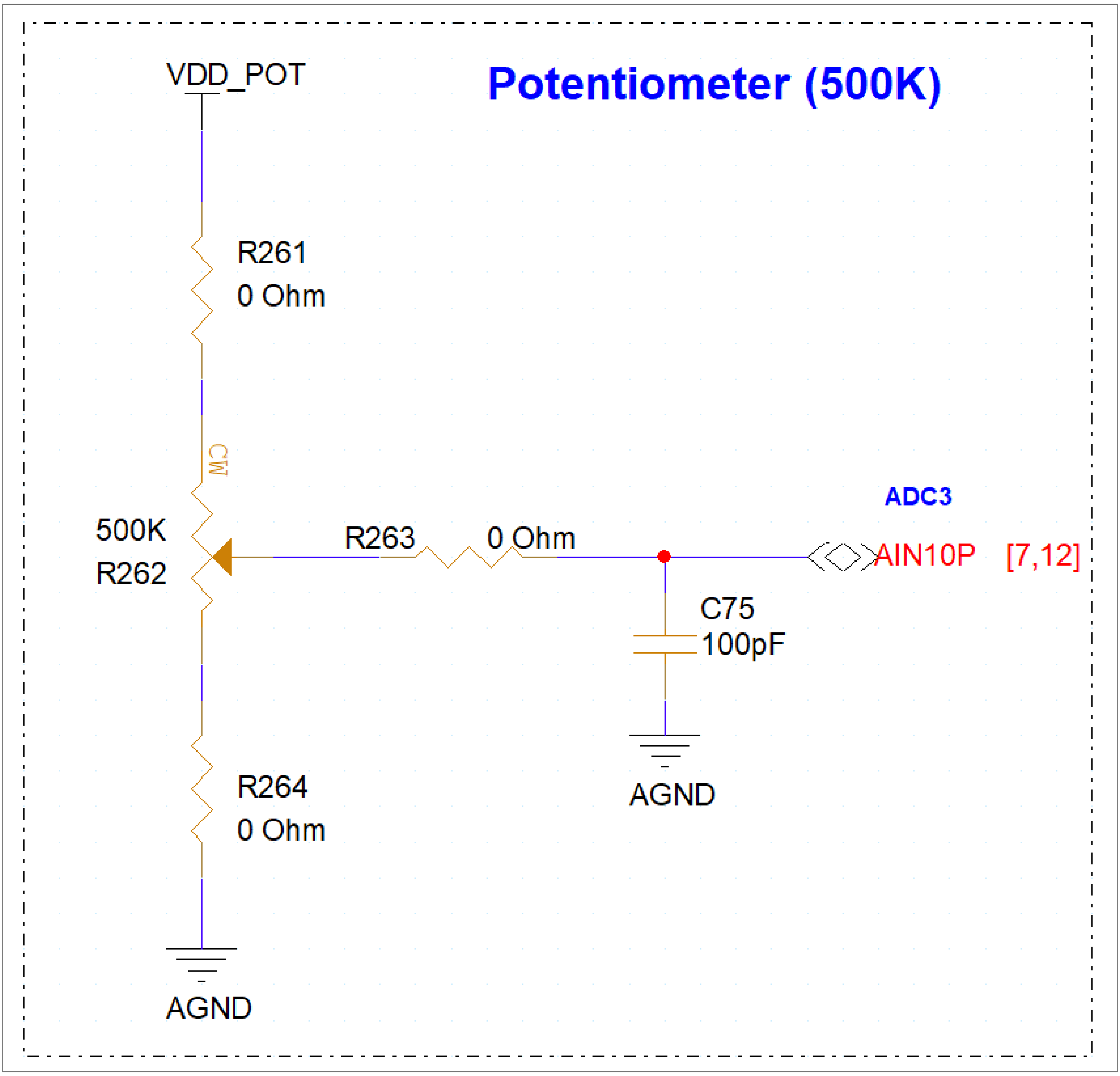

Figure 44 **Potentiometer 3 - 500K**

4.2.16 SMA connector

The evaluation kit is designed to provide two HRPWM signals through an SMA connector. By default, the signals from the PSOC™ C3M8 device are not directly connected to the SMA connector.

P1.6 and P9.3 serve as the sources of the HRPWM signals and are routed to the analog pins AIN7P and AIN7N, respectively, to loop back the digital (HRPWM/TCPWM) signals to the ADC block. These pins are connected to the ADC through an RC network and a series 0 Ω resistor for isolation.

To monitor the HRPWM signal on the SMA connector, the user must perform rework on the evaluation kit by removing the series resistor connected to the ADC pin and populating the 0 Ω resistor on the SMA connector path.

- To route P1.6 to the SMA connector, remove R209 and populate R206
- To route P9.3 to the SMA connector, remove R218 and populate R215

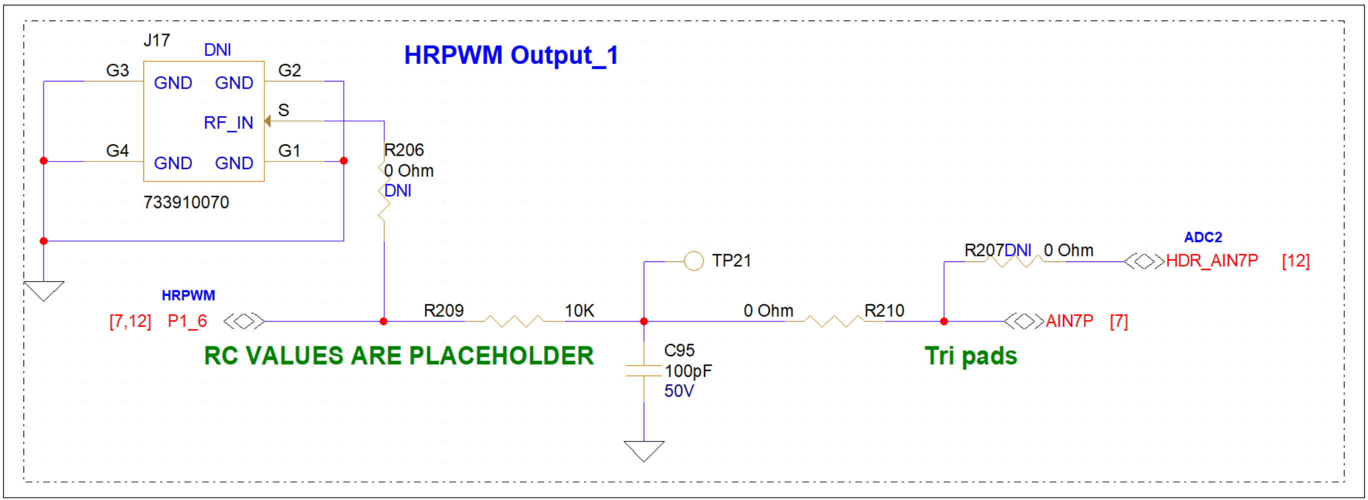

Figure 45 HRPWM output at P1.6

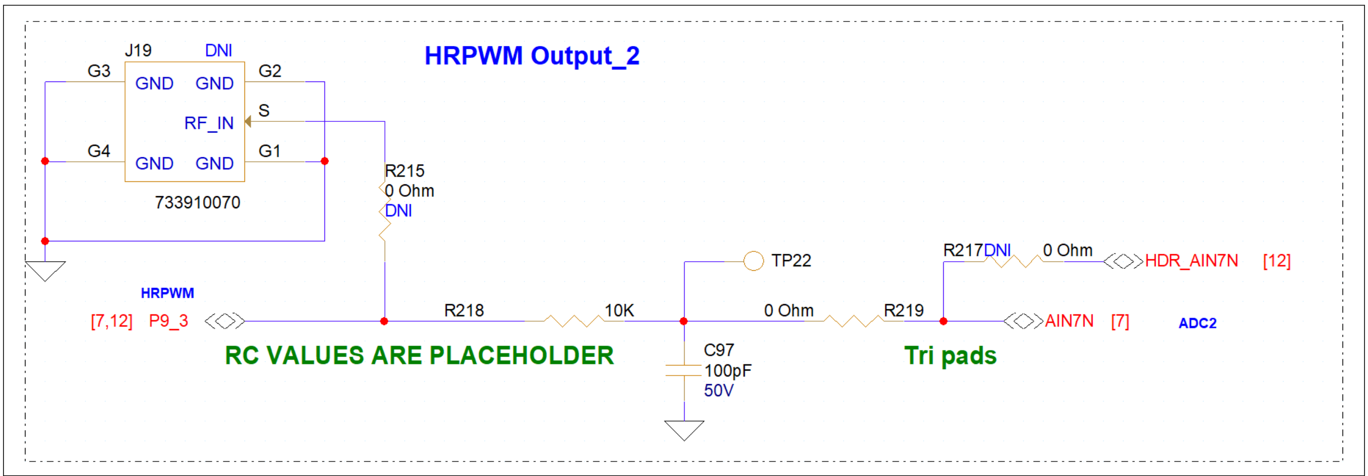

Figure 46 HRPWM output at P9.3

4.2.17 Crystals and external clock support

KIT_PSC3M8_EVK is equipped with two crystals to ensure efficient operation of the PSOC™ C3M8 device. A 16 MHz crystal is connected to pins P1.0 and P1.1, functioning as an external crystal oscillator (ECO).

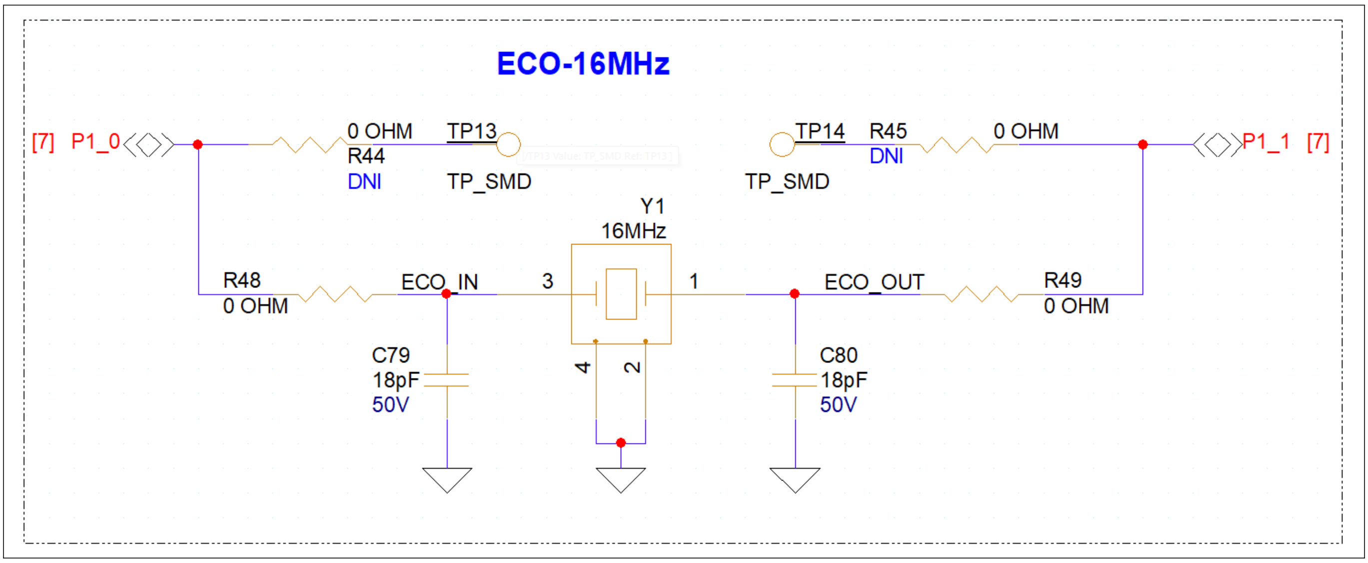

Figure 47 ECO 16 MHz

A 32.768 kHz crystal is connected to pins P0.0 and P0.1, functioning as a watch crystal oscillator (WCO).

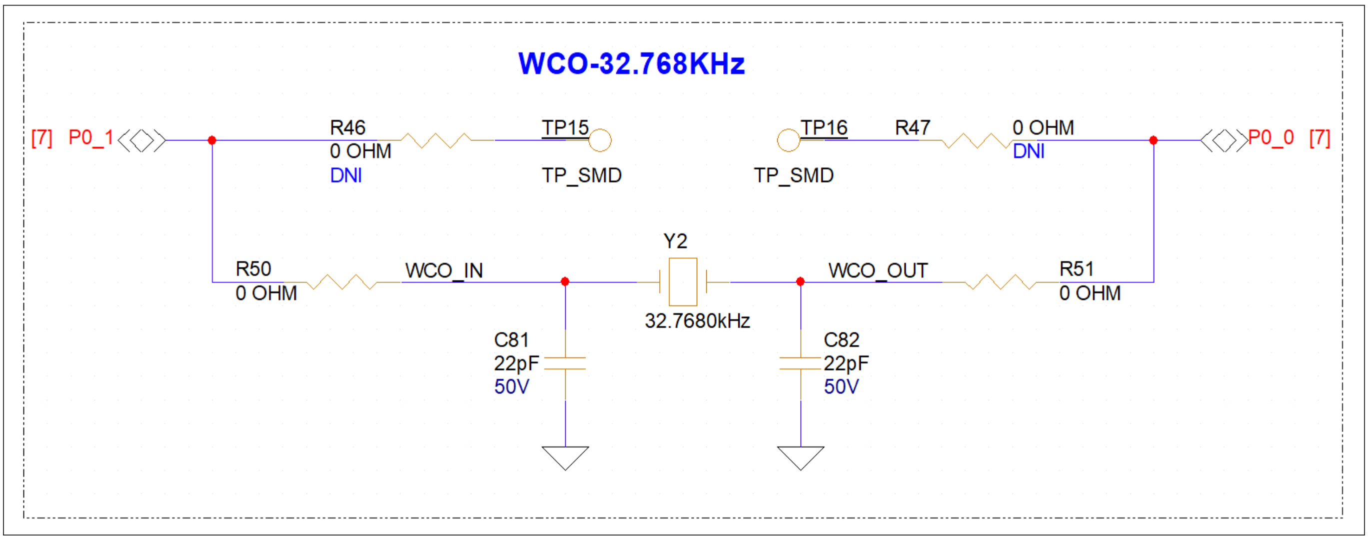

Figure 48 WCO 32.768 kHz

4.2.18 Reverse and over voltage protection

The circuit shown in Figure 49 and Figure 50 is designed to protect the device from reverse supply voltage. The evaluation kit includes a reverse voltage protection circuit to prevent damage caused by a reverse supply to the board. This circuit safeguards the PSOC™ C3M8 device by blocking any accidental reverse voltage from being supplied to the board.

The board utilizes a FET-based diode circuit for reverse voltage protection. This circuit protects the power supply pins associated with all headers, including the Arduino and expansion headers. However, programming headers are restricted to powering the PSOC™ Control C3 Evaluation Kit. To further protect the target device from accidental power fluctuations originating from external sources, the reverse voltage protection circuit is integrated into the design of the evaluation kit.

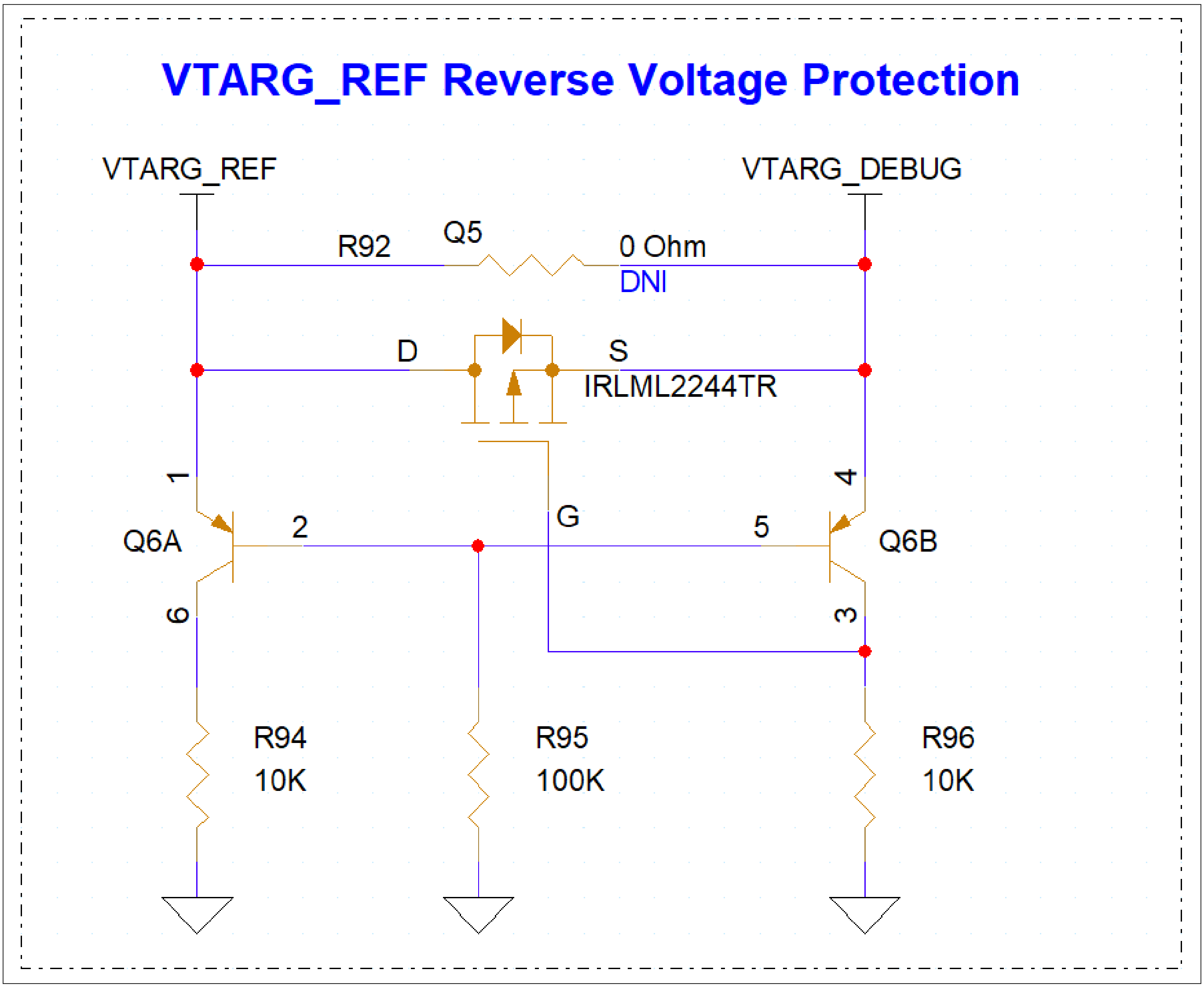

Figure 49 Reverse voltage protection from programming headers

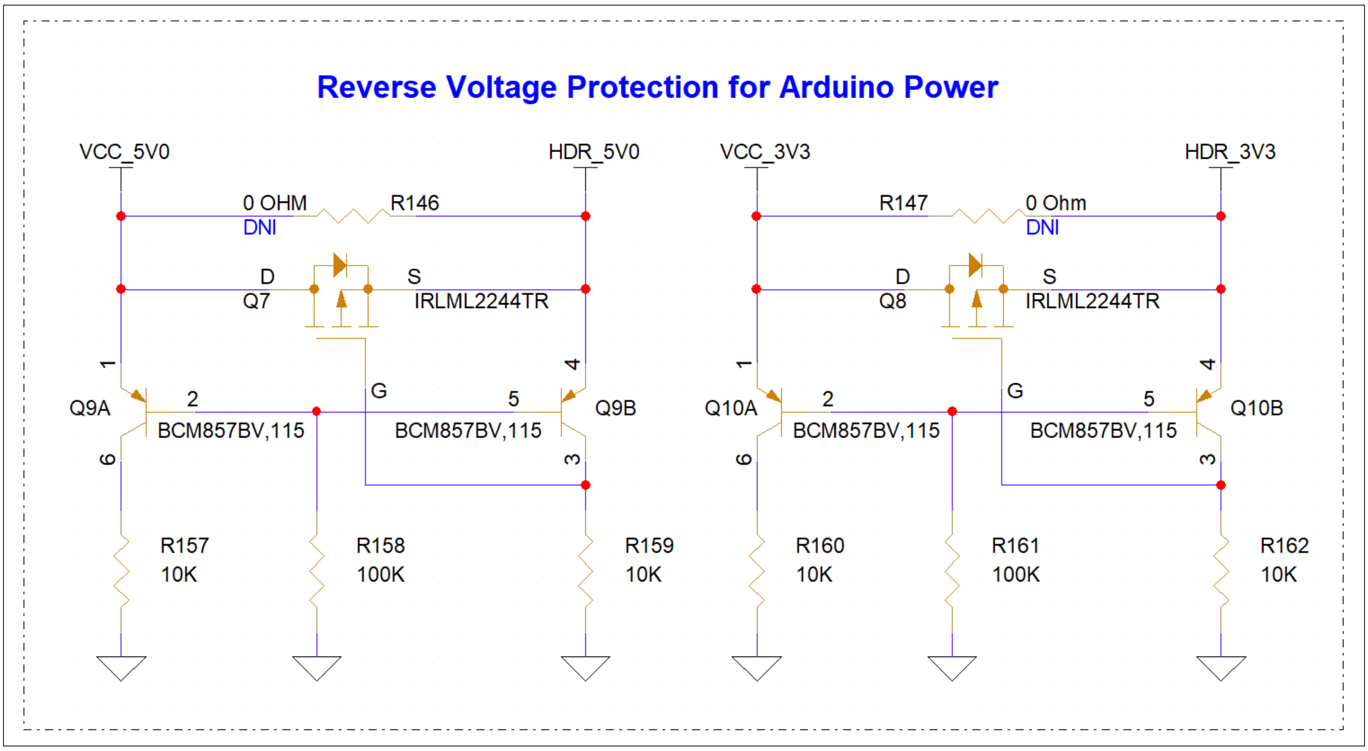

Figure 50 **Reverse voltage protection from headers compatible with Arduino**

ESD diodes are incorporated into the design to protect the USB supply connector from electrostatic discharge.

5 Kit rework

5.1 I/Os rework on expansion header

Some GPIOs from the PSOC™ C3M8 device are directly connected to the expansion header for user evaluation purposes.

GPIOs that are assigned to peripheral interfaces are also routed to the expansion header. However, users must perform the necessary rework on the evaluation kit to enable these multiplexed GPIO signals on the expansion header.

Table 3 provides details of the series resistor rework required to connect the GPIO signals to the expansion header.

Table 3 Resistor rework details for connecting GPIO signals to the expansion header

No.	PSC3M8 Port/Pin	Signal connected by default	Alternate function	Rework details
1	P1.4	User LED (D11)	J15.5 GPIO	To use P1.4 as a GPIO on the expansion header, remove R184 and load R171.
2	P1.5	User LED (D8)	J15.7 GPIO	To use P1.5 as a GPIO on the expansion header, remove R168 and load R174.
3	P1.6	J15.9 and AIN7P	HRPWM SMA	To probe the HRPWM signal on the SMA connector interface, remove R209 (10 kΩ) and populate R206.
4	P1.7	User LED (D10)	J15.11 GPIO	To use P1.7 as a GPIO on the expansion header, remove R178 and load R176.
5	P2.0	SPI MOSI	J15.13 GPIO	To use P2.0 as a GPIO on J15.13, remove R109 and populate R110.

(table continues...)

Table 3 (continued) Resistor rework details for connecting GPIO signals to the expansion header

No.	PSC3M8 Port/Pin	Signal connected by default	Alternate function	Rework details
6	P2.1	SPI CLOCK	J15.15	When using an external debugger, remove R112 and populate R113 to use the TDI pin on J10 and J11 headers. Populate R114 to use P2.1 as a GPIO on J15.15.
7	P2.2	SPI MISO	J15.17	To use P2.2 as a GPIO on J15.17, remove R117 and populate R118.
8	P3.0	SPI CS#	J15.27	To use P3.0 as a GPIO on J15.27, remove R119 and populate R240.
9	P3.1	J15.29	SPI CS#2	To use CS#2 with a pull-up configuration, populate R120 (0 Ω) and R116 (10 kΩ).
10	P3.3	UART TX	J15.26	To use P3.3 as a GPIO on J15.26, remove R85 and populate R90.
11	P3.4	UART RX	J15.24	To use P3.4 as a GPIO on J15.24, remove R87 and populate R91.
12	P3.7	J9.7 (TCPWM)	J15.31	The GPIO is also available on the expansion header J15.31 by populating R175.
13	P4.0	LED (D13)	J15.4	To use P4.0 as a GPIO on the expansion header J15.4, remove R197 and load R182.

(table continues...)

Table 3 (continued) Resistor rework details for connecting GPIO signals to the expansion header

No.	PSC3M8 Port/Pin	Signal connected by default	Alternate function	Rework details
14	P4.1	LED (D9)	J15.6	To use P4.1 as a GPIO on the expansion header J15.6, remove R173 and load R185.
15	P5.2	SW3	J16.36 and J16.38	To use the push button independently on J16.38, remove R169.
16	AIN0P	Potentiometer (R212)	J16.24	To use AIN0P as an analog pin on J16.24, remove R213 and populate R170.
17	AIN0N	Potentiometer (R221)	J16.26	To use AIN0N as an analog pin on J16.26, remove R222 and populate R172.
18	AIN7P	P1.6 (HRPWM)	J16.28	To use AIN7P as an analog pin on J16.28, remove R210 and populate R207.
19	AIN7N	P9.3 (HRPWM)	J16.30	To use AIN7N as an analog pin on J16.30, remove R219 and populate R217.
20	AIN9P	Voltage sense+	J16.16	To use AIN9P as an analog pin on J16.16, remove R55 and populate R201.
21	AIN9N	Voltage sense-	J16.18	To use AIN9N as an analog pin on J16.18, remove R258 and populate R202.
22	P7.0	SW4	J15.33	To use P7.0 as a GPIO on the expansion header J15.33, remove R155 and load R244.

(table continues...)

Table 3 **(continued) Resistor rework details for connecting GPIO signals to the expansion header**

No.	PSC3M8 Port/Pin	Signal connected by default	Alternate function	Rework details
23	P7.4	CAN1_TX	J16.3	To use P7.4 as a GPIO on the expansion header J16.3, remove R97 and load R98.
24	P7.5	CAN1_RX	J16.5	To use P7.5 as a GPIO on the expansion header J16.5, remove R99 and load R100.
25	P7.6	CAN STB	J16.7	To use P7.6 as a GPIO on the expansion header J16.7, remove R106 and load R181.
26	P9.1	LED (D12)	J15.36	To use P9.1 as a GPIO on the expansion header J15.36, remove R190 and load R179.
27	P9.3	J15.38 and AIN7N	HRPWM (SMA)	To probe the HRPWM signal on the SMA connector interface, remove R218 (10K) and populate R215.

5.2 SWD/JTAG header signals rework (J10)

The evaluation kit is provisioned with SWD, JTAG, and Trace debugging options for programming and debugging the PSOC™ C3M8 device. By default, the SWD interface is connected to the J10 header. Resistor rework is required to enable the JTAG interface on the evaluation kit.

- **TDO**: Remove R71 and populate R80
- **TDI**: Remove R112 and populate R113

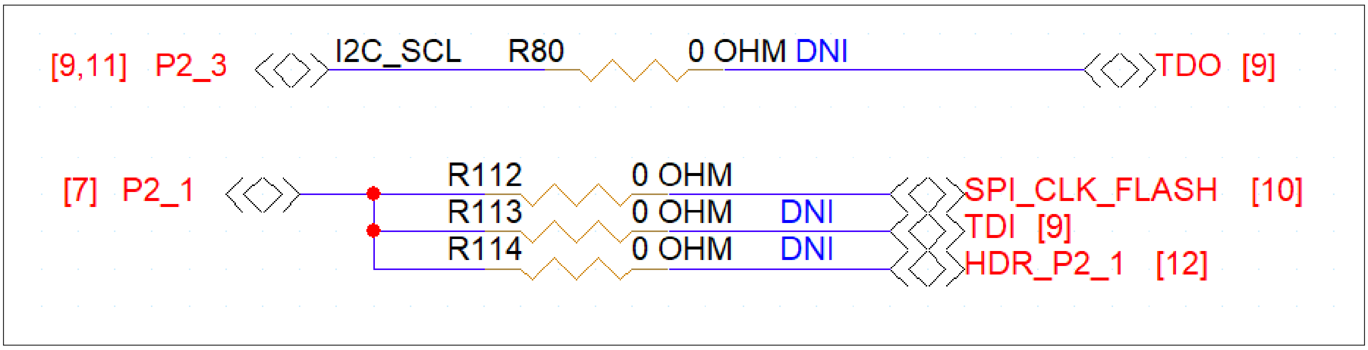

Figure 51 **JTAG interface rework**

5.3 ETM trace and debug header (J11)

The trace debugging header is not populated on the evaluation kit. To enable trace debugging, the user must populate a suitable connector. Additionally, the following series resistor rework is required to route the trace signals to the ETM header:

- **Trace CLK**: Populate R78
- **Trace Data0**: Populate R73
- **Trace Data1**: Populate R68
- **Trace Data2**: Populate R62
- **Trace Data3**: Populate R75

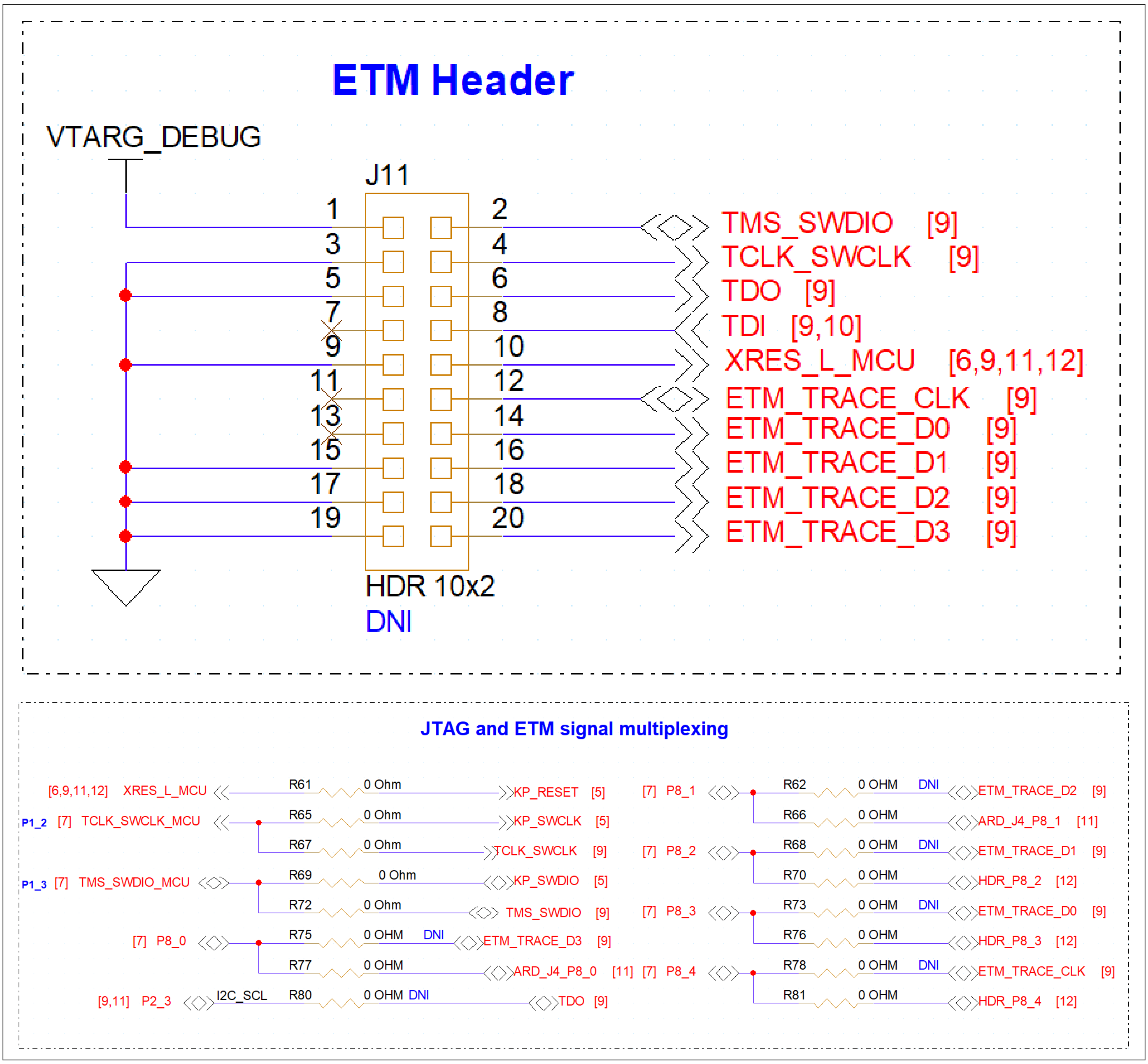

Figure 52 **ETM header and resistor rework**

5.4 Anti-aliasing filter for ADC input

All ADC signals are routed through an RC network. By default, the signals are routed with a 0 Ω series resistor, and no capacitor is populated on the evaluation kit. To enable the ADC anti-aliasing network, the user must perform the specific rework as shown in Figure 53.

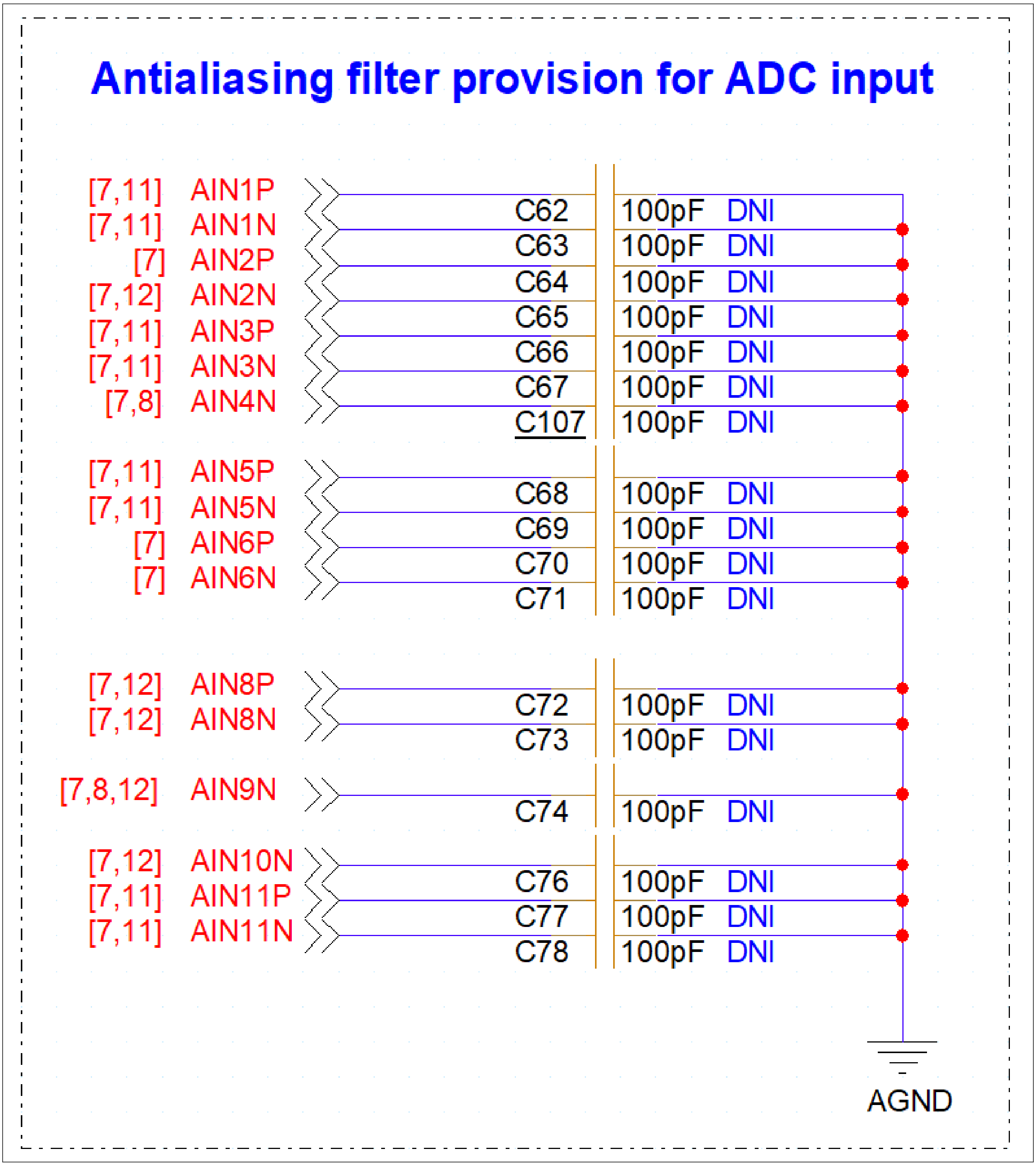

Figure 53 Anti-aliasing filter provision for ADC input

5.5 SMA connector (J17, J19)

Two HRPWM signals from the PSOC™ C3M8 device are routed to the ADC pins of the device through an RC network for loopback purposes. These HRPWM signals are also connected to an SMA connector, which requires resistor rework to enable probing of the high-resolution PWM signals on the SMA connector.

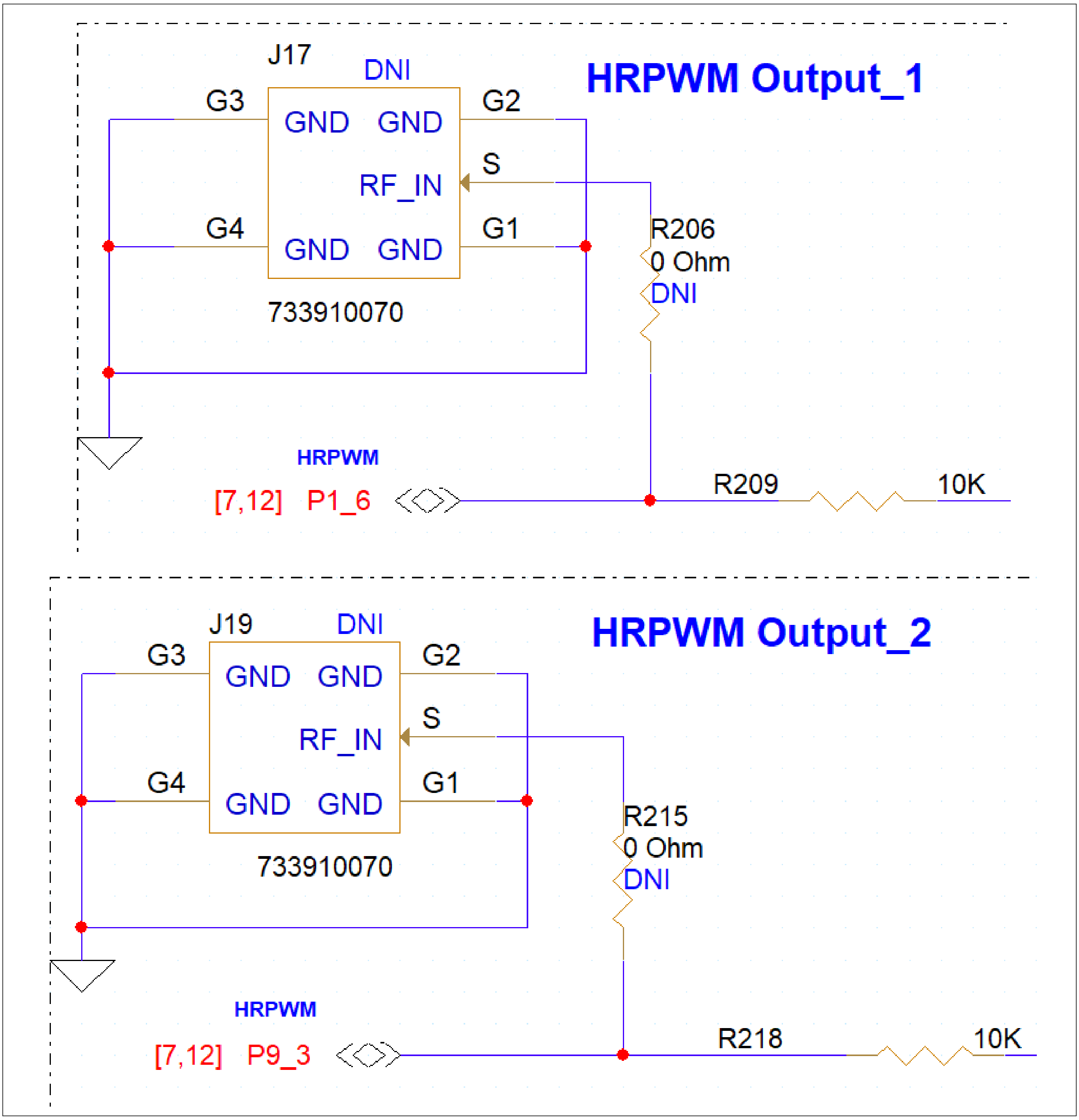

Figure 54 **SMA connector rework to probe HRPWM signals**

5.6 Power supply ORing rework

The evaluation kit is designed to support multiple power source options, including USB, VIN via the DC jack, or VIN from the Arduino J1 connector. A diode network (ORing circuit) is incorporated to prevent power conflicts when both USB and VIN are supplying power to the evaluation kit.

However, a stray voltage may occur at the connector end when only one power source is connected to the evaluation kit. To mitigate this, a pull-down resistor is provided on the evaluation kit as a discharge path. This resistor prevents stray voltage from appearing on the input side of the power connector.

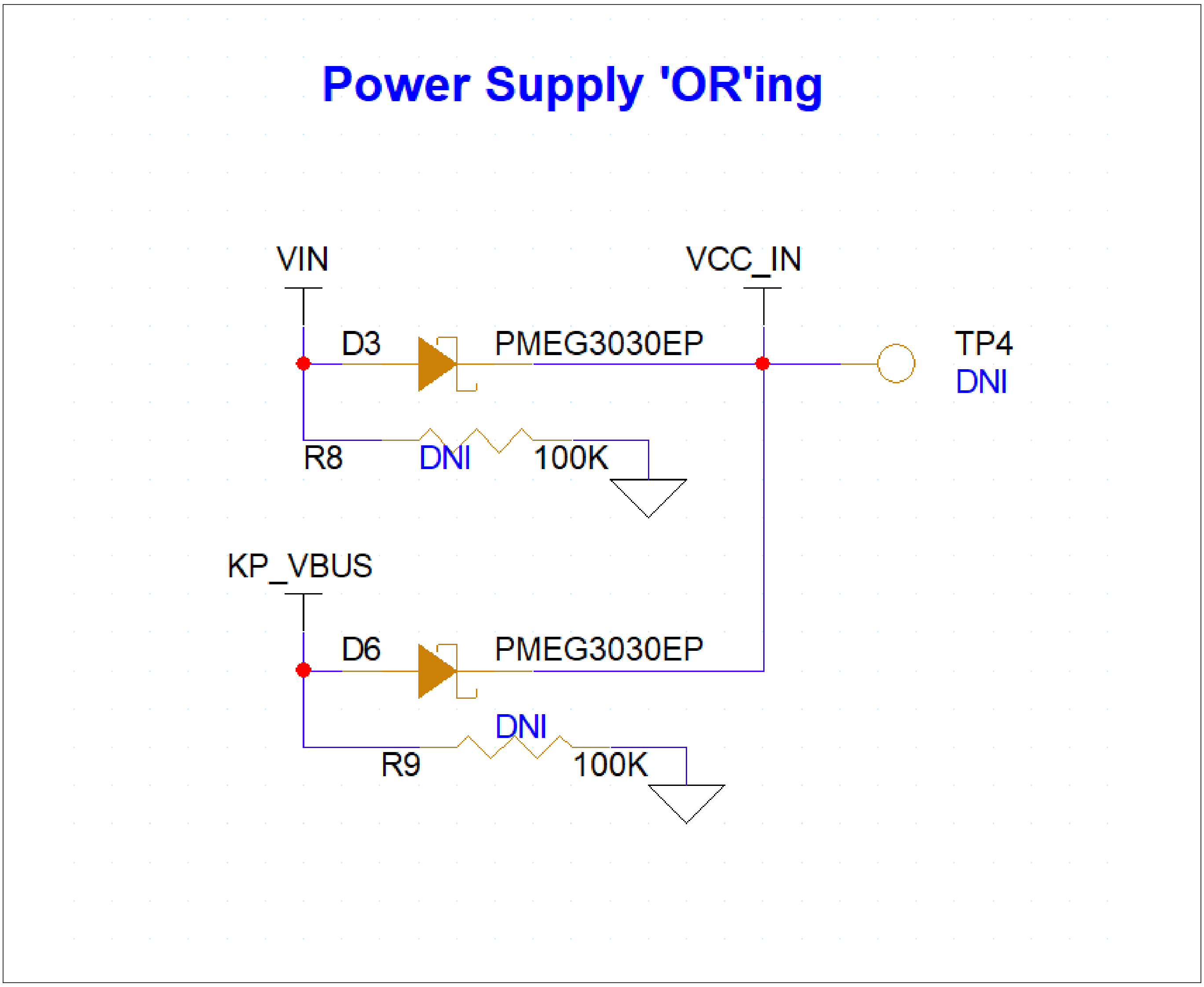

Figure 55 **Power supply ORing rework**

5.7 DAC buffer rework

An amplifier with a 2x gain is designed on the evaluation kit to boost the signal from the DAC output of the PSOC™ C3M8 device. However, a rework provision is included on the evaluation kit to bypass the amplifier network and connect the DAC output directly to header J9. Populate R259 and R260 to bypass amplifier from AIN0 and AIN1 respectively. While populating above resistors, U12 amplifier device must be disconnected either by removing R252 for AIN0 and R255 for AIN1 or remove U12 device for isolating the buffer circuit from DAC output.

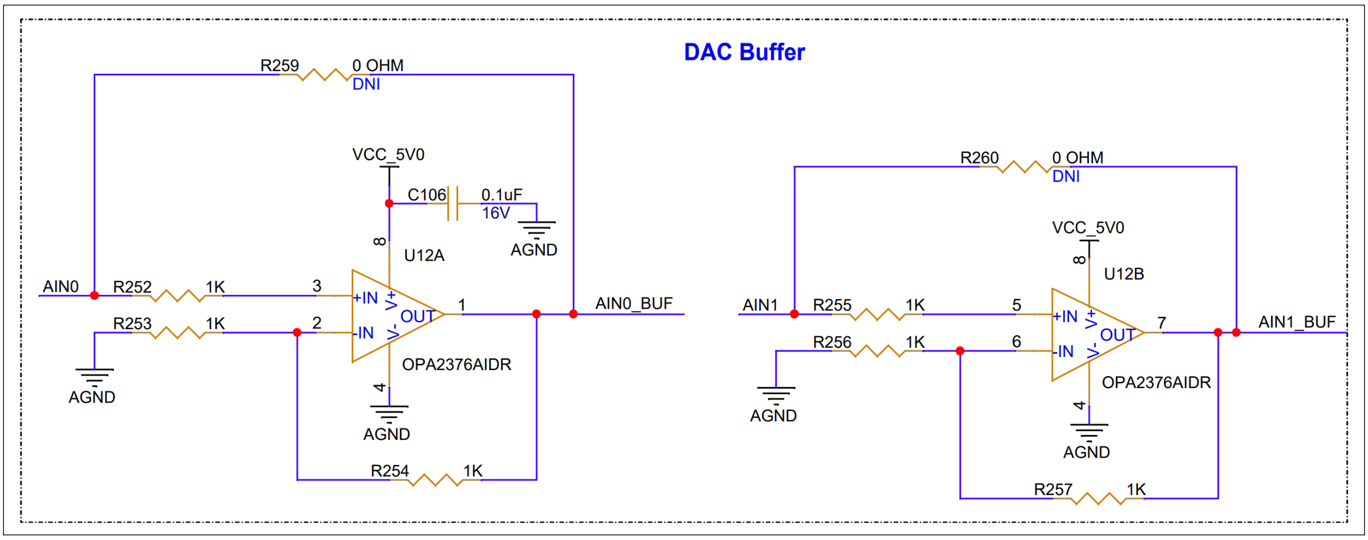

Figure 56 Rework to bypass the DAC buffer

5.8 External clock support

An external clock option is available on the evaluation kit to source the reference clock to the PSOC™ C3M8 device externally. By default, the signal is connected to the expansion header for use as a GPIO. To source the external clock to the PSOC™ C3M8 device, the user must perform the following rework on the evaluation kit as shown in Figure 57.

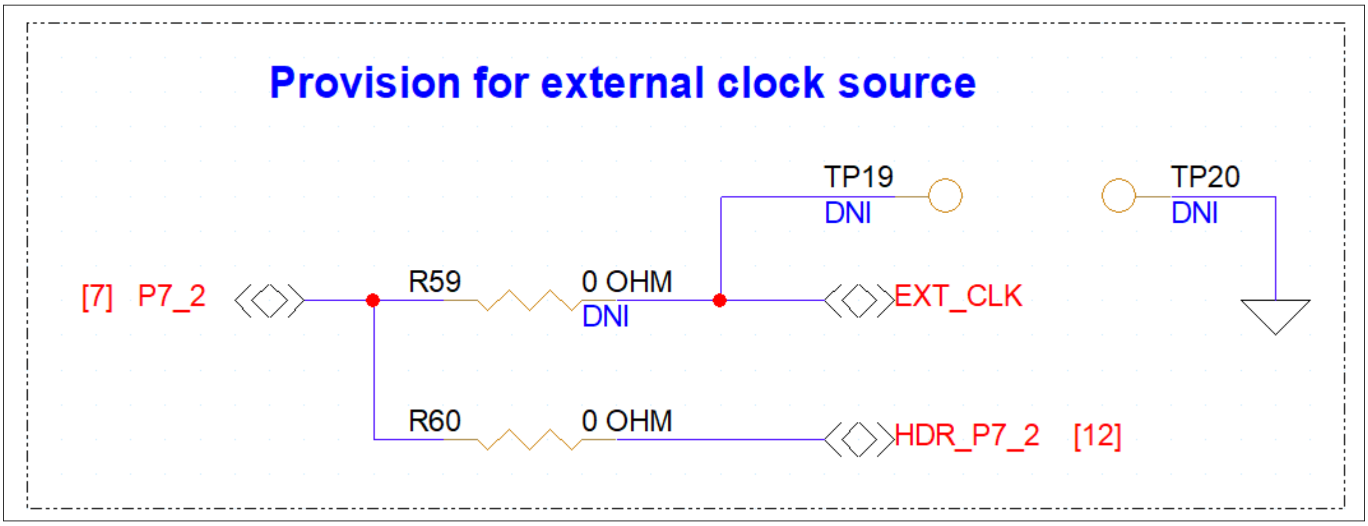

Figure 57 Provision for external clock source

5.9 CAN FD rework

P7.4 and P7.5 from the PSOC™ C3M8 device are connected to the CAN FD transceiver, with the transceiver output routed to a terminal connector for external communication. The evaluation kit provides a provision to connect an alternate port from the PSOC™ C3M8 device by disconnecting the existing CAN FD signals.

The rework shown in Figure 58 enables the connection of P3.0 and P3.1 as CAN FD signals from the PSOC™ C3M8 device.

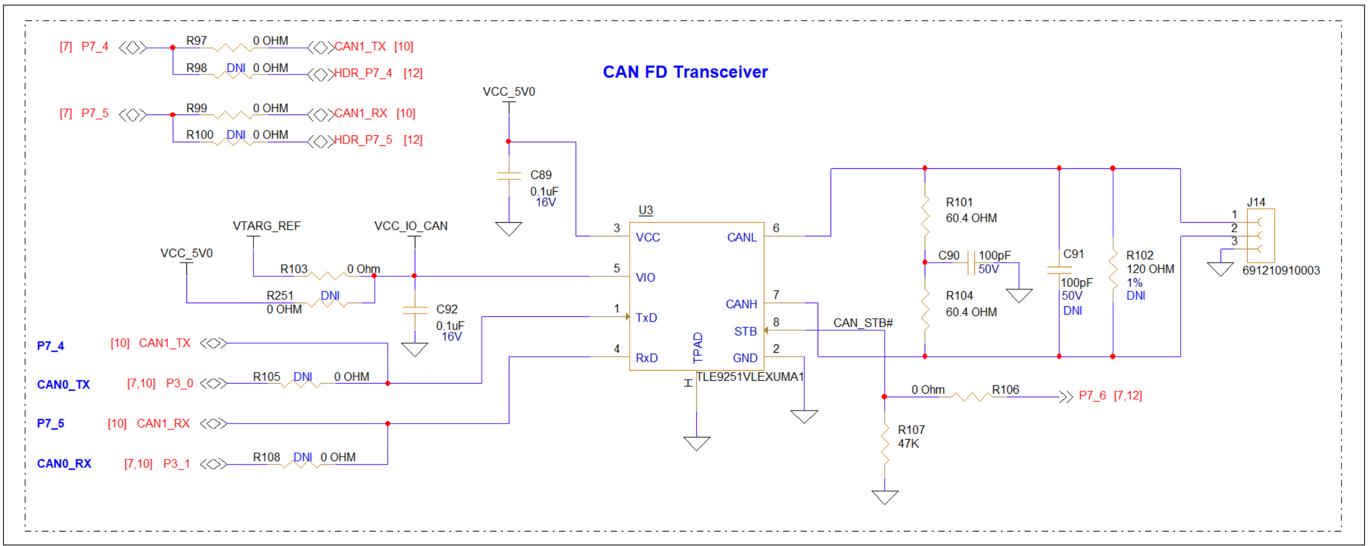

Figure 58 CAN FD rework for alternate port/pin use case

5.10 Offset network for current sensing

The current-sensing resistor, R38, is connected in the series path between the MCU and GND. The shunt circuit is designed to support an offset voltage ranging from 0 to 1 V, which can be configured by performing the necessary rework on the evaluation kit.

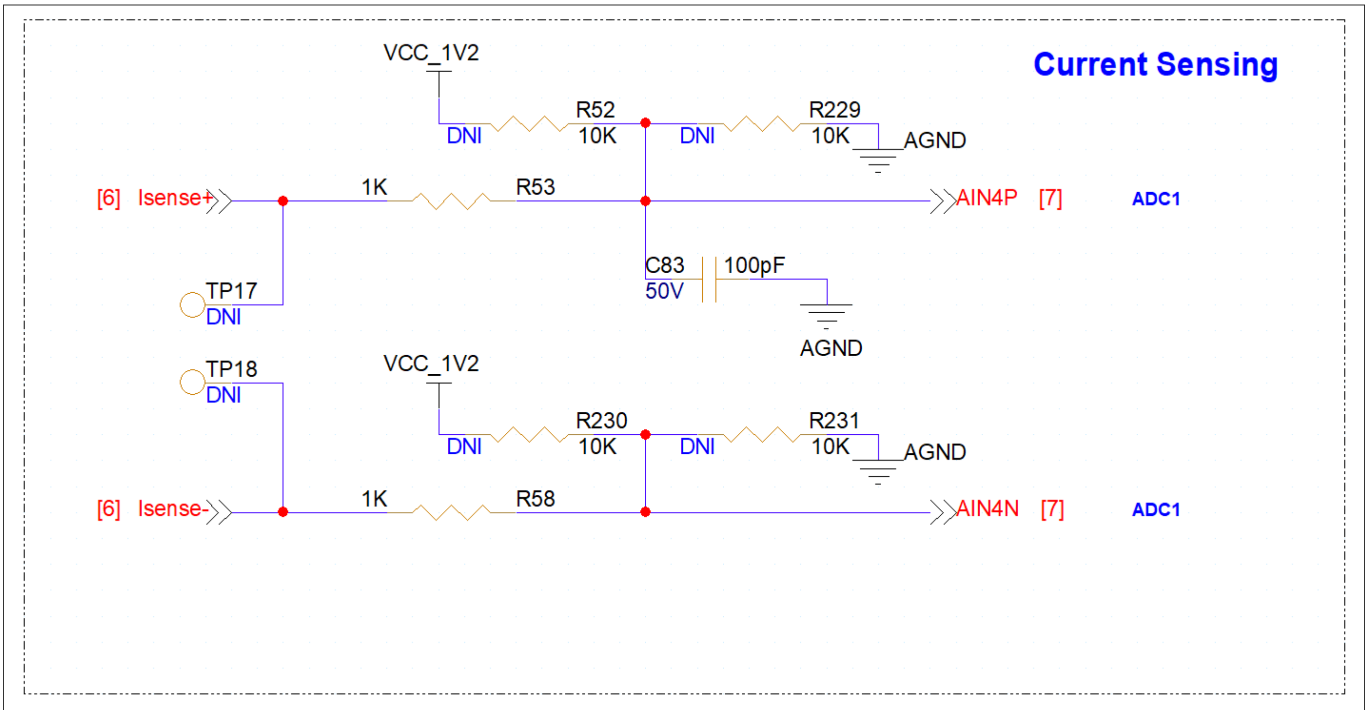

Figure 59 Offset network for current-sensing circuit

5.11 UART communication on external header

The UART interface is connected between the PSOC™ C3M8 device and the KitProg3 debugger. The same UART signals are also routed to an external 3-pin header, allowing the PSOC™ C3M8 UART communication lines to connect with external devices.

The rework details shown in Figure 60 outline the steps required to enable UART communication between the PSOC™ C3M8 device and external devices using the J12 header.

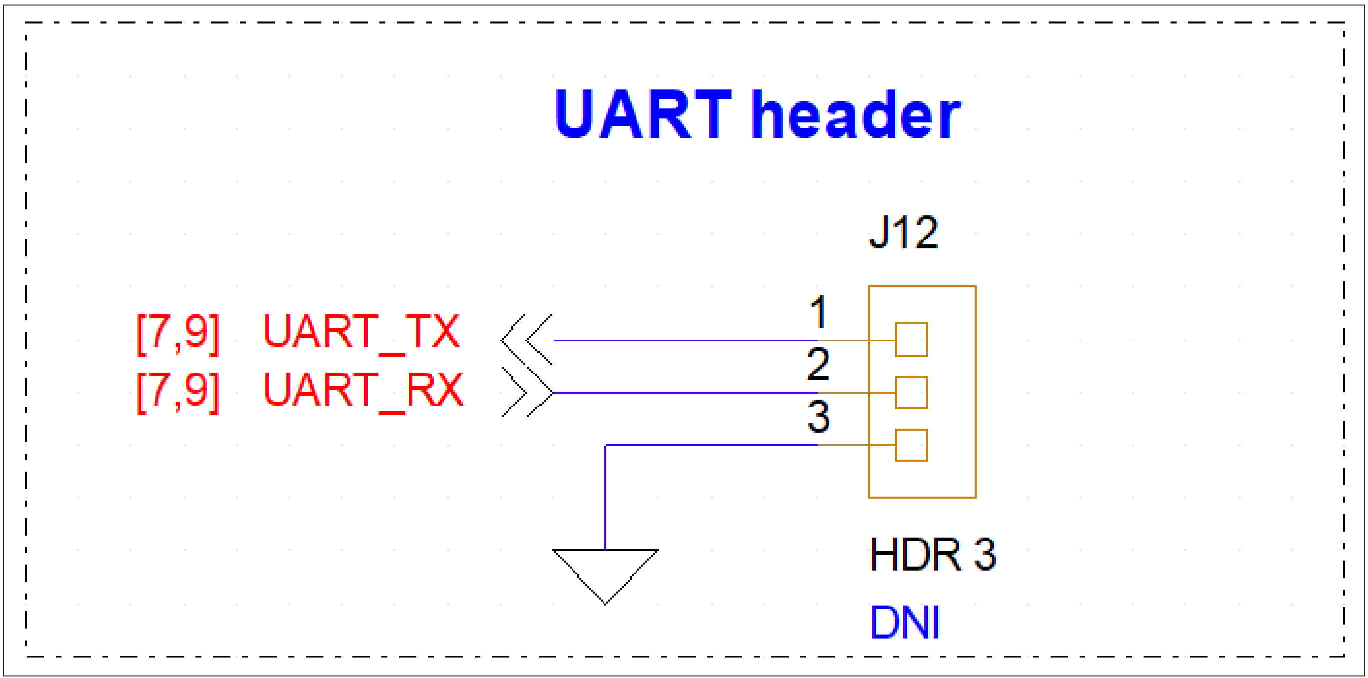

Figure 60 UART header rework for communication with external devices

5.12 I²C address of the temperature sensor

The temperature sensor on the evaluation kit has a configurable I²C address based on the status of the SDO pin.

- The SDO pin of the DPS368XTSA1 sensor device is connected to a GPIO to enable it as an interrupt pin
- A provision exists on the evaluation kit for a pull-down resistor (100 kΩ) to be populated on the SDO pin
- When the pull-down resistor is populated, the I²C address is configured to 0x76
- By default, without the pull-down resistor, the I²C address remains configured as 0x77

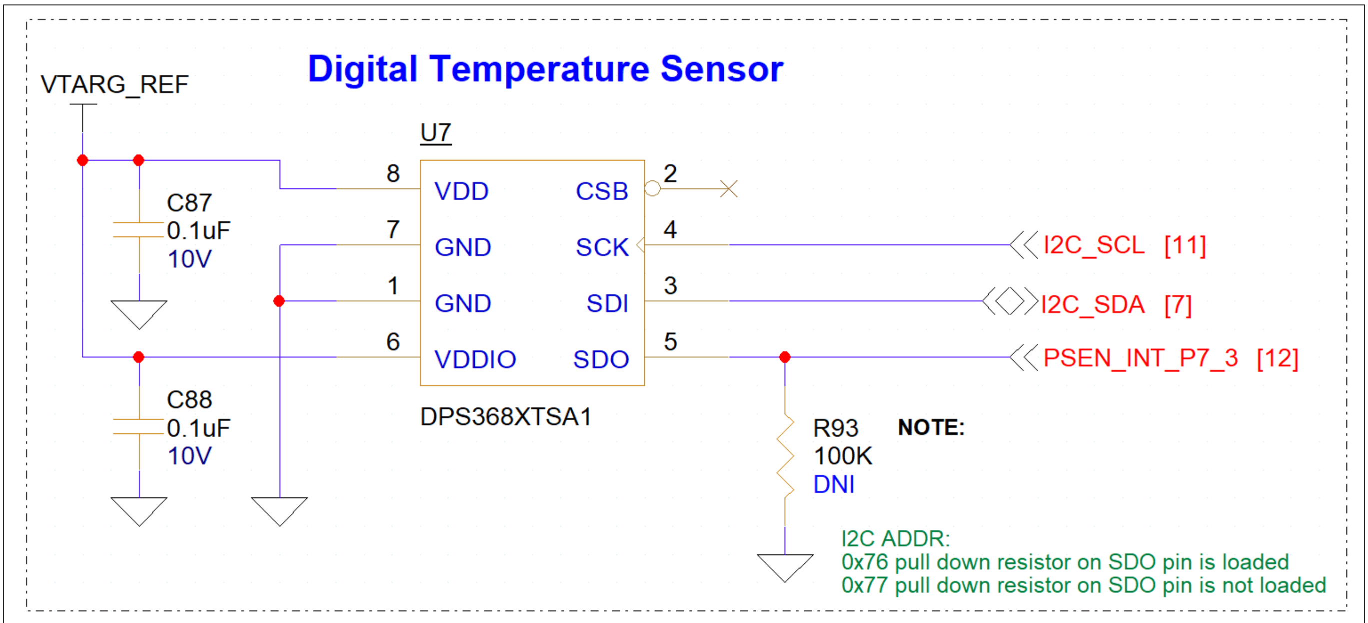

Figure 61 I²C address configuration on digital temperature sensor

Revision history

Document revision	Date of release	Description of changes
**	2025-07-30	Initial release
*A	2026-03-27	Added Figure 1 Updated Figure 2 Updated Board details Updated Programming and debugging using the ModusToolbox™ Added Figure 8 Added Figure 9 Updated VREF voltage selection header Updated DFU mode selection updated Power indication LED Updated User switches Updated DAC buffer rework
*B	2026-06-26	Publish to web

Trademarks

PSOC™, formerly known as PSoC™, is a trademark of Infineon Technologies. Any references to PSoC™ in this document or others shall be deemed to refer to PSOC™.

Trademarks

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

Edition 2026-06-26

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-yeg1747194201722