PSoC™ 6 CY8CKIT-062-BLE with AIROC™ Bluetooth® LE Pioneer Kit user guide

About this document

Scope and purpose

Thank you for your interest in the PSoC™ 6 CY8CKIT-062-BLE with AIROC™ Bluetooth® LE Pioneer Kit. The PSoC™ 6 Bluetooth® Low Energy (BLE) Pioneer Kit enables you to evaluate and develop your applications using the PSoC™ 6 MCU with Bluetooth® LE Connectivity (hereafter called “PSoC™ 6 MCU”).

Intended audience

This document is intended for PSoC™ 6 CY8CKIT-062-BLE with AIROC™ Bluetooth® LE Pioneer Kit users. This board is intended to be used under laboratory conditions.

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: *A drive that is incorrectly applied or installed can lead to component damage or reduction in product lifetime. Wiring or application errors such as undersizing the motor, supplying an incorrect or inadequate AC supply, or excessive ambient temperatures may result in system malfunction.*
	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

1 Introduction

PSoC™ 6 MCU is Infineon's, ultra-low-power PSoC™ specifically designed for wearables and IoT products. PSoC™ 6 MCU is a true programmable embedded system-on-chip, integrating a 150-MHz Arm® Cortex®-M4 as the primary application processor, a 100-MHz Arm® Cortex®-M0+ that supports low-power operations, up to 1 MB Flash and 288 KB SRAM, an integrated Bluetooth® LE 4.2 radio, CAPSENSE™ touch-sensing, and programmable analog and digital peripherals that allow higher flexibility, in-field tuning of the design, and faster time-to-market.

The PSoC™ 6 Bluetooth® LE Pioneer board offers compatibility with Arduino shields. The board features a PSoC™ 6 MCU, a 512-Mb NOR flash, onboard programmer/debugger (KitProg3), USB Type-C power delivery system (EZ-PD™ CCG3), 5-segment CAPSENSE™ slider, two CAPSENSE™ buttons, one CAPSENSE™ proximity sensing header, an RGB LED, two user LEDs, and one push button. The board supports operating voltages from 1.8 V to 3.3 V for PSoC™ 6 MCU.

The CY8CKIT-062-BLE package includes a CY8CKIT-028-EPD E-INK display shield that contains a 2.7-inch E-INK display, a motion sensor, a thermistor, and a PDM microphone. The kit package also contains a CY5677 CySmart BLE 4.2 USB Dongle that is factory-programmed to emulate a Bluetooth® LE GAP Central device, enabling you to emulate a Bluetooth® LE host on your computer.

You can use ModusToolbox™ to develop and debug your PSoC™ 6 MCU projects. ModusToolbox™ Creator is Infineon's standard integrated design environment (IDE). ModusToolbox™ also supports exporting your designs to other third-party firmware development tools.

If you are new to PSoC™ 6 MCU and ModusToolbox™ IDE, you can find introductions in the application note AN237038 - Getting Started with PSoC™ 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

1.1 Kit contents

The CY8CKIT-062-BLE package has the following contents, as shown in Figure 1.

- PSoC™ 6 Bluetooth® LE Pioneer Board
- CY8CKIT-028-EPD E-INK display shield
- USB Type-A to Type-C cable
- Four jumper wires (4 inches each)
- Two proximity sensor wires (5 inches each)
- Quick start guide

Kit contents

1. PSoC™ 6 Bluetooth® LE Pioneer Board
2. CY8CKIT-028-EPD E-INK Display Shield
3. USB Type-A to Type-C cable
4.
5. Four jumper wires (4 inches each)
6. Two proximity sensor wires (5 inches each)
7. Quick Start Guide (this document)

Figure 1 **Kit contents**

Inspect the contents of the kit; if you find any part missing, contact Infineon Support or visit the nearest Infineon sales office.

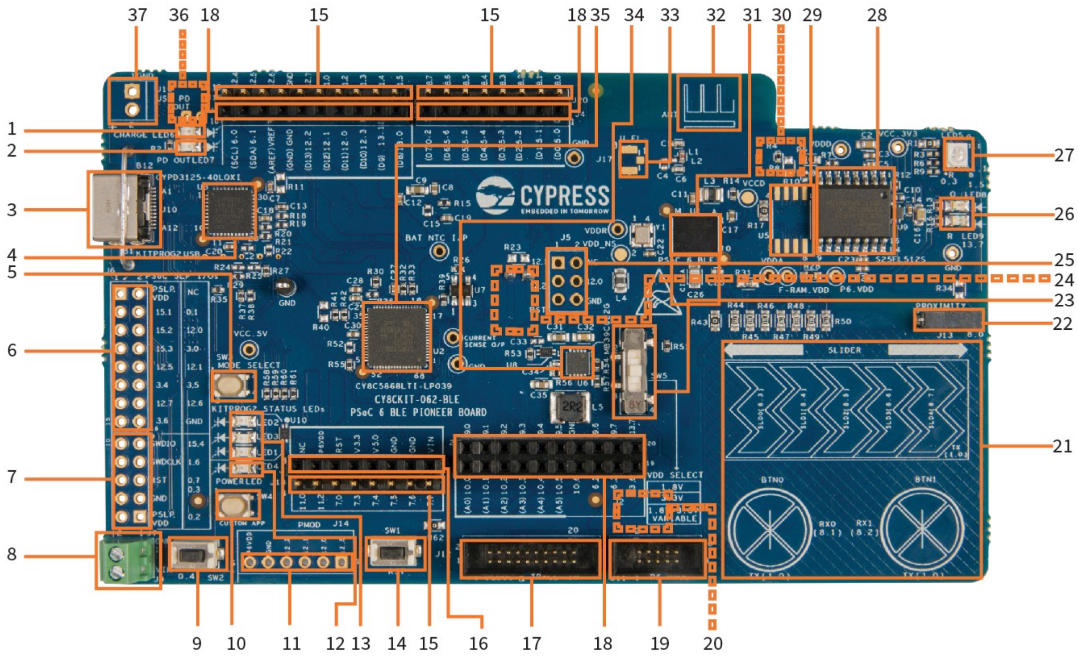
1.2 Board details

Figure 2 shows the Pioneer board that has the following features:

- PSoC™ 6 MCU with Bluetooth® LE connectivity
- Expansion headers that are compatible with Arduino Uno 3.3 V shields[1] and Digilent Pmod modules
- 512-Mbit external quad-SPI NOR Flash that provides a fast, expandable memory for data and code
- KitProg3 onboard programmer/debugger with mass storage programming, USB to UART/I^2C/SPI bridge functionality, and custom applications support
- EZ-PD™ CCG3 USB Type-C power delivery (PD) system with rechargeable lithium-ion polymer (Li-Po) battery support[2]
- CAPSENSE™ touch-sensing slider (5 elements), two buttons, all of which are capable of both self-capacitance (CSD) and mutual-capacitance (CSX) operation, and a CSD proximity sensor that lets you evaluate Infineon's' fourth-generation CAPSENSE™ technology
- 1.8 V to 3.3 V operation of PSoC™ 6 MCU is supported. An additional 330 mF super-capacitor is provided for backup domain supply (Vbackup)
- Two user LEDs, a RGB LED, a user button, and a reset button for PSoC™ 6 MCU. Two buttons and three LEDs for KitProg3

1 5 V shields are not supported
2 Battery and power-delivery capable USB Type-C to Type-C cable are not included in the kit package and should be purchased separately

PSoC™ 6 Bluetooth® LE Pioneer Board Details



1 Battery charging indicator (LED6)
2 USB PD output voltage availability indicator (LED7)
3 KitProg3 USB connector (J10)
4 EZ-PD™ CCG3 Type-C Port Controller with PD (CYPD3125-40LQXI, U3)
5 KitProg3 programming mode selection button (SW3)
6 KitProg3 I/O header (J6)*
7 KitProg3 programming/custom application header (J7)*
8 External power supply connector (J9)
9 PSoC™ 6 Bluetooth® LE user button (SW2)
10 KitProg3 application selection button (SW4)
11 I/O header compatible with Digilent Pmod (J14)*
12 Power LED (LED4)
13 KitProg3 status LEDs (LED1, LED2, and LED3)
14 PSoC™ 6 Bluetooth® LE reset button (SW1)
15 PSoC™ 6 Bluetooth® LE I/O header (J18, J19 and J20)
16 Power header compatible with Arduino Uno R3 (J1)
17 PSoC™ 6 Bluetooth® LE debug and trace header (J12)
18 I/O headers compatible with Arduino Uno R3 (J2, J3, J4)
19 PSoC™ 6 Bluetooth® LE program and debug header (J11)
20 KitProg3 programming target selection switch (SW6)**
21 CAPSENSE™ slider and buttons
22 CAPSENSE™ proximity header (J13)
23 PSoC™ 6 Bluetooth® LE VDD selection switch (SW5)
24 PSoC™ 6 Bluetooth® LE power monitoring jumper (J8)**
25 ICSP header compatible with Arduino Uno R3 (J5)*
26 PSoC™ 6 Bluetooth® LE user LEDs (LED8 and LED9)
27 RGB LED (LED5)
28 512-Mbit serial NOR flash memory (S25FL512S, U4)
29 Serial Ferroelectric RAM (U5)*
30 Vbackup and PMIC control selection switch (SW7)
31 PSoC™ 6 Bluetooth® LE (CY8C6347BZI-BLD53, U1)
32 Bluetooth® LE antenna
33 U.FL connector for external antenna (J17)*
34 Main voltage regulator (MB39C022G, U6)
35 KitProg3 (PSoC™ 5LP) programmer and debugger (CY8C5868LTI-LP039, U2)
36 Battery connector (J15)**
37 USB PD output voltage (9V/12V) connector (J16)*

*Footprints only, not populated on the board.
**Components at the bottom side of the board.

Figure 2 **Pioneer board**

Table 2 **Selection switches in the Pioneer board**

Switch	Location on the board	Purpose	Default position
SW5	Front	Selects the V_{DD} supply of PSoC™ 6 MCU between 1.8 V, 3.3 V, and variable 1.8 V to 3.3 V that is controlled by KitProg3.	3.3 V
SW6	Back	In the PSoC™ 6 MCU position: • KitProg3 can program the onboard PSoC™ 6 MCU • In addition, the PSoC™ 6 MCU can be programmed by an external programmer such as a MiniProg4 connected to J11 Avoid connecting any external devices to J11 in the PSoC™ 6 MCU position, as it can cause programming failure. In the "External Device" position: • KitProg3 can program any PSoC™ 4/5/6 devices connected to J11	PSoC™ 6 MCU
SW7	Back	Selects the Vbackup supply connection of PSoC™ 6 MCU between $V_{DDD}$and the super-capacitor. When V_{DDD} is selected, the regulator can be turned ON/OFF by the KitProg3. When the super-capacitor is selected, PSoC™ 6 MCU can turn the regulator ON/OFF.	V_{DDD}/KitProg3

Figure 3 shows the pinout of the Pioneer board.

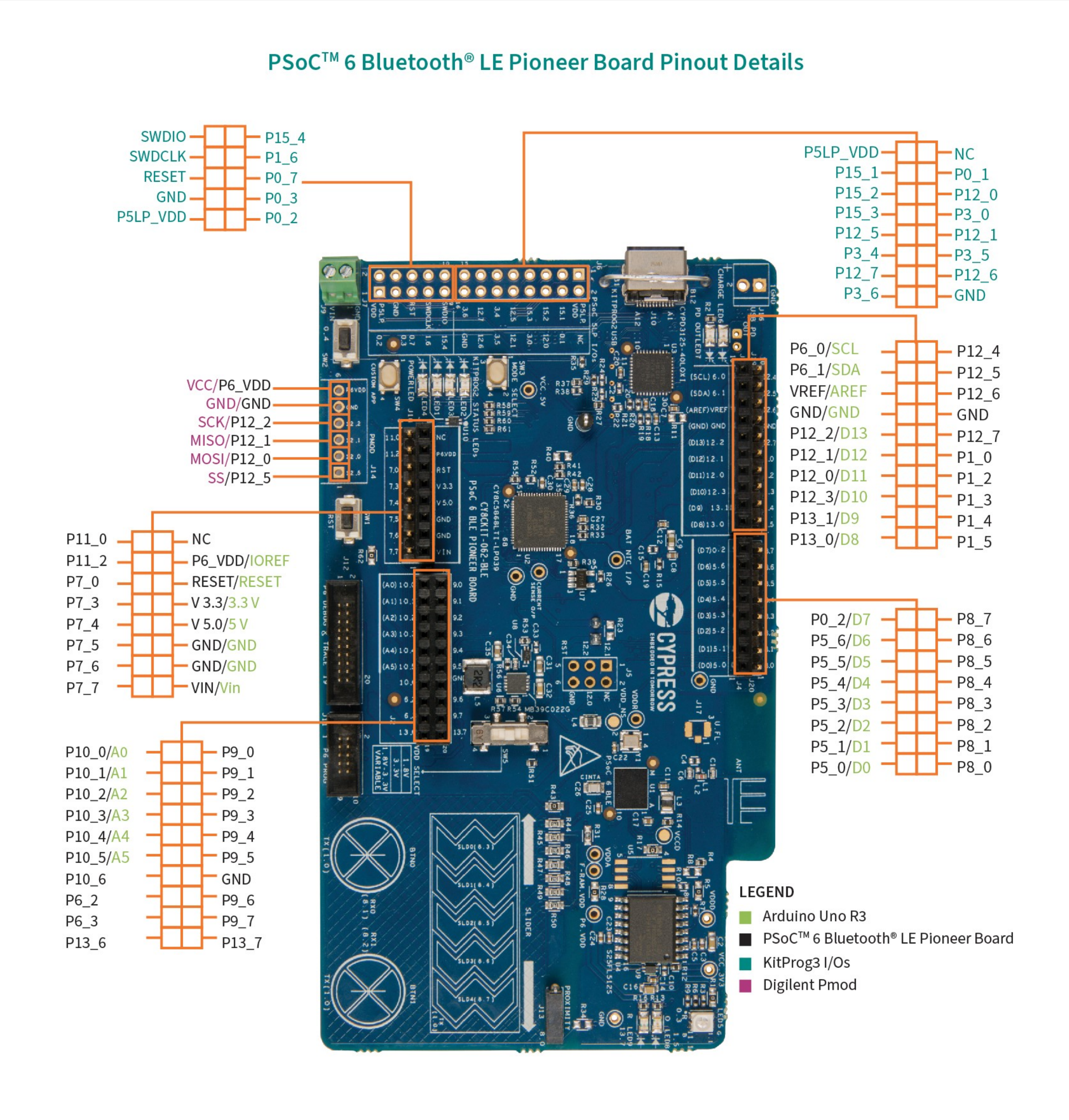


Figure 3 **Pioneer board pinout**

Table 3 **Poineer board pinout**

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
ANT	RFIO, Antenna	–	
XI	ECO IN	–	
XO	ECO OUT	–	

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
XRES	Reset	–	
P0.0	WCO IN	–	
P0.1	WCO OUT	–	
P0.2	Arduino header J4.8, D7	–	
P0.3	RGB Red LED	–	
P0.4	User button with Hibernate wakeup capability	PMIC wake in signal	Connected to ground as active low logic by default. Remove R150 and populate R149 and R148 to change the switch to active high logic. Refer to Appendix A.15 for additional information on rework, on Rev. 08 or earlier versions.
P0.5	PMIC control	–	
P1.0	CAPSENSE™ Tx	GPIO on non-Arduino header (J19.5)	Connected to CAPSENSE™ by default. Remove R43 to disconnect CAPSENSE™ or add R101 to connect to header.
P1.1	RGB Green LED	–	
P1.2	GPIO on non-Arduino header (J19.4)	–	
P1.3	GPIO on non-Arduino header (J19.3)	–	
P1.4	GPIO on non-Arduino header (J19.2)	–	
P1.5	Orange User LED	GPIO on non-Arduino header (J19.1)	Connected to primary and secondary function by default. Remove R13 to disconnect from LED.
P5.0	Arduino J4.1, D0 UART RX KitProg3 UART TX	–	Remove R120 to disconnect from KitProg3 UART TX
P5.1	Arduino J4.2, D1 UART TX KitProg3 UART RX	–	Remove R119 to disconnect from KitProg3 UART RX

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
P5.2	Arduino J4.3, D2 UART RTS KitProg3 UART CTS	–	Remove R77 and load R78 to disconnect from KitProg3 UART CTS (This will also disconnect RTS and SPI lines from KitProg3
P5.3	Arduino J4.4, D3 UART CTS KitProg3 UART RTS	–	Remove R77 and load R78 to disconnect from KitProg3 UART CTS (This will also disconnect RTS and SPI lines from KitProg3
P5.4	Arduino J4.5, D4	–	
P5.5	Arduino J4.6, D5	–	
P5.6	Arduino J4.7, D6	–	
P6.0	Arduino J3.10, SCL KitProg3 I^2C SCL	–	Remove R109 to disconnect from KitProg3 I^2C SCL
P6.1	Arduino J3.9, SDA KitProg3 I^2C SDA	–	Remove R114 to disconnect from KitProg3 I^2C SDA
P6.2	GPIO on non-Arduino header (J2.15)	–	
P6.3	GPIO on non-Arduino header (J2.17)	CAPSENSE™ Shield	Remove R144 to disconnect from GND and populate R138 to connect to CAPSENSE™ shield (hash pattern on board)
P6.4	TDO/SWO	–	
P6.5	TDI	–	
P6.6	TMS/SWDIO	–	Remove R147 to disconnect from KitProg3 SWDIO
P6.7	TCLK/SWCLK	–	Remove R146 to disconnect from KitProg3 SWCLK
P7.0	TRACECLK	GPIO on non-Arduino header (J18.6)	Populate R135 to connect to J18 header
P7.1	CINTA	–	
P7.2	CINTB	CSH	Remove C87 (0.47 nF) and populate 10 nF for CSH
P7.3	GPIO on non-Arduino header (J18.5)	CSH	Remove R88 to disconnect from header and populate C88 (10 nF) for CSH

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
P7.4	TRACEDATA[3]	GPIO on non-Arduino header (J18.4)	Populate R132 to connect to J18
P7.5	TRACEDATA[2]	GPIO on non-Arduino header (J18.3)	Populate R133 to connect to J18
P7.6	TRACEDATA[1]	GPIO on non-Arduino header (J18.2)	Populate R134 to connect to J18
P7.7	CMOD	GPIO on non-Arduino header (J18.1)	Populate R87 to connect to J18
P8.0	Proximity	GPIO on non-Arduino header (J20.1)	Replace R31 with Zero Ω and populate R34 to connect to header
P8.1	CAPSENSE™ Button0 Rx	GPIO on non-Arduino header (J20.2)	Remove R44 to disconnect CAPSENSE™ pad and populate R100 to connect to header
P8.2	CAPSENSE™ Button1 Rx	GPIO on non-Arduino header (J20.3)	Remove R50 to disconnect CAPSENSE™ pad and populate R103 to connect to header
P8.3	CAPSENSE™ Silder0 Rx	GPIO on non-Arduino header (J20.4)	Remove R45 to disconnect CAPSENSE™ pad and populate R99 to connect to header
P8.4	CAPSENSE™ Silder1 Rx	GPIO on non-Arduino header (J20.5)	Remove R46 to disconnect CAPSENSE™ pad and populate R97 to connect to header
P8.5	CAPSENSE™ Silder2 Rx	GPIO on non-Arduino header (J20.6)	Remove R47 to disconnect CAPSENSE™ pad and populate R105 to connect to header
P8.6	CAPSENSE™ Silder3 Rx	GPIO on non-Arduino header (J20.7)	Remove R48 to disconnect CAPSENSE™ pad and populate R98 to connect to header
P8.7	CAPSENSE™ Silder4 Rx	GPIO on non-Arduino header (J20.8)	Remove R49 to disconnect CAPSENSE™ pad and populate R104 to connect to header
P9.0	GPIO on non-Arduino header (J2.2)	–	

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
P9.1	GPIO on non-Arduino header (J2.4)	–	
P9.2	GPIO on non-Arduino header (J2.6)	–	
P9.3	TRACEDATA[0]	GPIO on non-Arduino header (J2.8)	Populate R131 to connect to header
P9.4	GPIO on non-Arduino header (J2.10)	–	
P9.5	GPIO on non-Arduino header (J2.12)	–	
P9.6	GPIO on non-Arduino header (J2.16)	–	
P9.7	GPIO on non-Arduino header (J2.18)	–	
P10.0	Arduino J2.1, A0	–	
P10.1	Arduino J2.3, A1	–	
P10.2	Arduino J2.5, A2	–	
P10.3	Arduino J2.7, A3	–	
P10.4	Arduino J2.9, A4 PDM_CLK	–	
P10.5	Arduino J2.11, A5 PDM_DAT	–	
P10.6	GPIO on non-Arduino header (J2.13)	–	
P11.0	FRAM CS	GPIO on non-Arduino header (J18.8)	Connected to primary and secondary function by default. Remove R17 to disconnect from J18 and load R10 (10K) as FRAM pull-up.
P11.1	RGB Blue LED	–	
P11.2	QSPI FLASH CS	GPIO on non-Arduino header (J18.7)	Populate R8 to connect to J18, remove R5 and R7 to disconnect from Flash and pull-up.
P11.3	QSPI FLASH/FRAM DATA3	–	
P11.4	QSPI FLASH/FRAM DATA2	–	
P11.5	QSPI FLASH/FRAM DATA1	–	

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
P11.6	QSPI FLASH/FRAM DATA0	–	
P11.7	QSPI FLASH/FRAM CLK	–	
P12.0	Arduino J3.4, D11 SPI MOSI	–	Remove R77 and load R78 to disconnect from KitProg3_SPI lines (This will also disconnect RTS and CTS lines from KitProg3)
P12.1	Arduino J3.5, D12 SPI MISO	–	Remove R77 and load R78 to disconnect from KitProg3_SPI lines (This will also disconnect RTS and CTS lines from KitProg3)
P12.2	Arduino J3.6, D13 SPI CLK	–	Remove R77 and load R78 to disconnect from KitProg3_SPI lines (This will also disconnect RTS and CTS lines from KitProg4)
P12.3	Arduino J3.3, D10 SPI SELECT	–	Remove R77 and load R78 to disconnect from KitProg3_SPI lines (This will also disconnect RTS and CTS lines from KitProg5)
P12.4	KitProg3 SPI SELECT	GPIO on non-Arduino header (J19.10)	Connected to primary and secondary function by default. Remove R81 to disconnect from J19 or remove R84 to disconnect KitProg3_SPI_SELECT
P12.5	PMOD SPI SELECT	GPIO on non-Arduino header (J19.9)	Connected to primary and secondary function by default. Remove R71 to disconnect from J19 or remove R82 to disconnect PMOD_SPI_SELECT
P12.6	GPIO on non-Arduino header (J19.7)	–	
P12.7	GPIO on non-Arduino header (J19.5)	–	
P13.0	Arduino J3.1, D8	–	
P13.1	Arduino J3.2, D9	–	

(table continues...)

Table 3 (continued) Poineer board pinout

PSoC™ 6 pin	Primary onboard function	Secondary onboard function	Connection details
P13.6	GPIO on non-Arduino header (J2.19)	CAPSENSE™ shield	Remove R144 to disconnect from GND and populate R137 to connect to CAPSENSE™ shield (hash pattern on board)
P13.7	Red User LED	GPIO on non-Arduino header (J2.20)	Connected to primary and secondary function by default. Remove R16 to disconnect from LED.
V_{REF}	SAR BYPASS, J3.8, AREF	–	

Figure 4 shows the E-INK display shield that has the following features:

- A 2.7 inch monochrome E-INK display with a resolution of 264 × 176. The E-INK display can retain its contents even in the absence of power, which provides an ultra low-power, "always-on" display functionality
- A thermistor that allows temperature-compensation of the display as well as general purpose temperature measurement
- A 3-axis acceleration and 3-axis gyroscopic motion sensor
- A PDM microphone for voice input
- An I/O level translator that allows the board to operate at any voltage between 1.8 V and 1.8 V ~ 5 V, by providing a constant 3.3 V interface to the E-INK display
- A load switch that can be controlled by the board to toggle the E-INK display's power

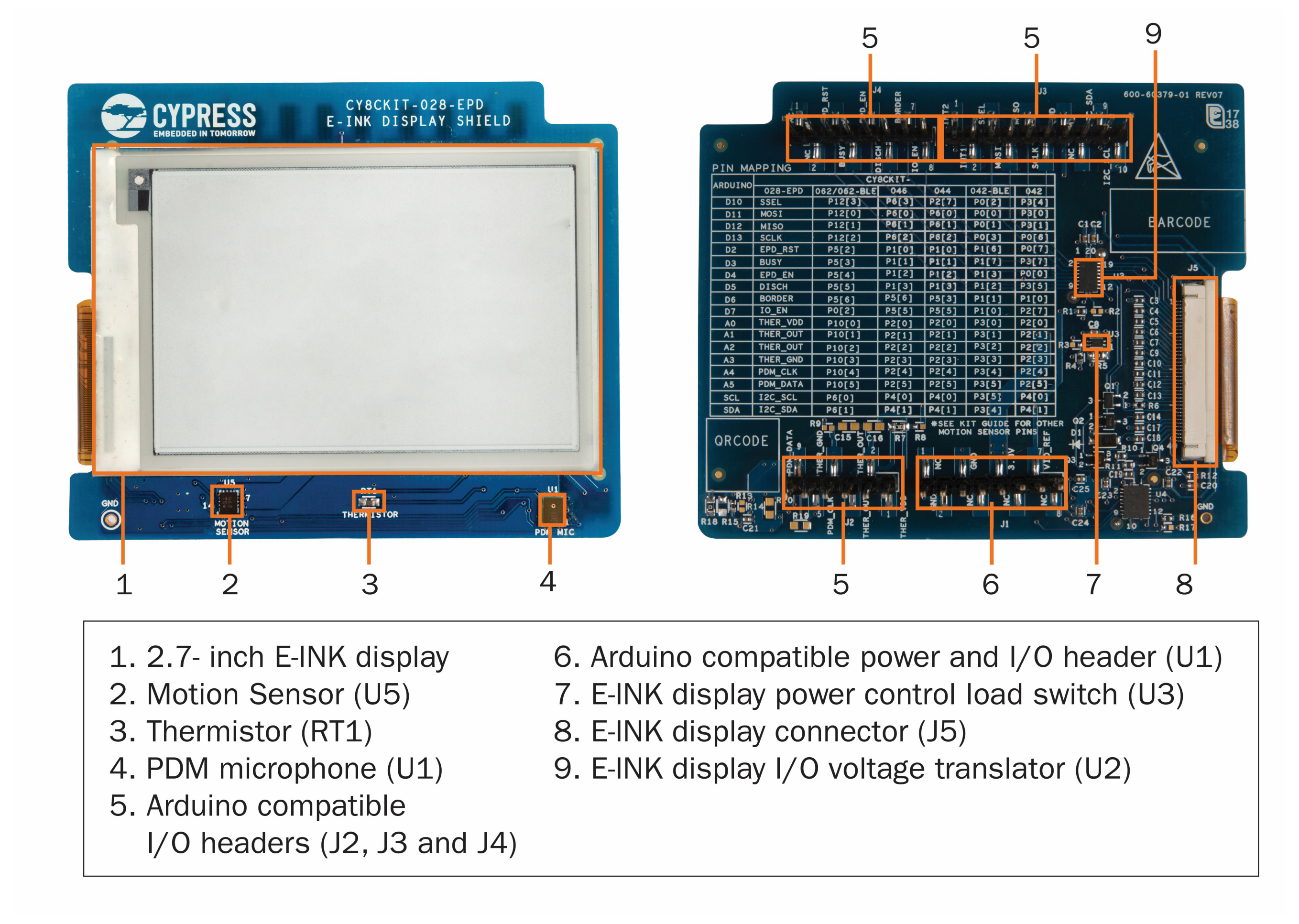


Figure 4 **E-INK display shield**

Figure 5 shows the pinout of the E-INK display shield.

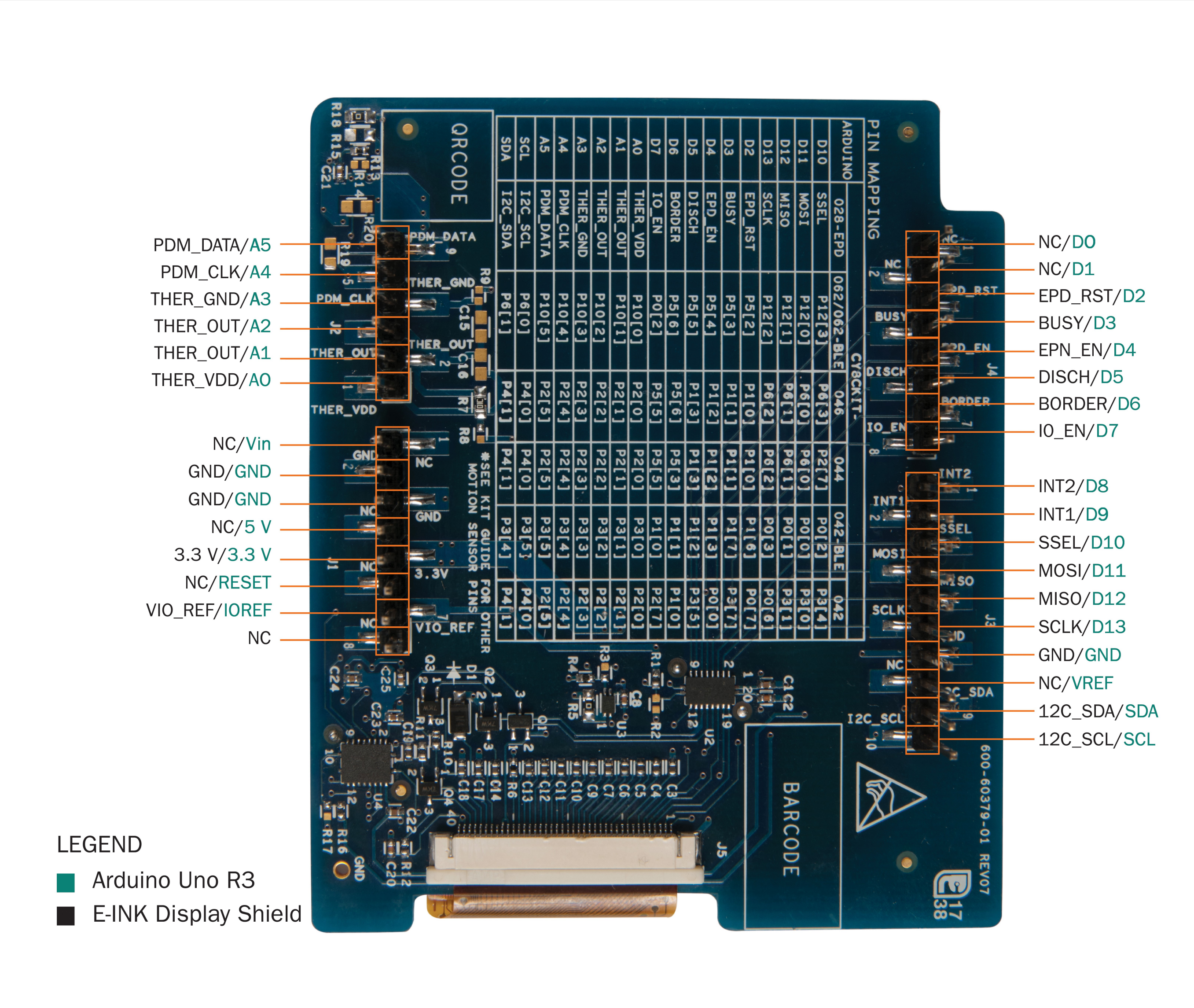


Figure 5 E-INK shield pinout

Table 4 E-INK shield pinout

Pin #	Arduino pin	Arduino function	E-INK shield function	Pioneer board connection
J3.1	D8	DIGITAL I/O	IMU INT1	P13[0]
J3.2	D9	PWM	IMU INT2	P13[1]
J3.3	D10	SS/PWM	SSEL	P12[3]
J3.4	D11	MOSI/PWM	MOSI	P12[0]
J3.5	D12	MISO	MISO	P12[1]
J3.6	D13	SCK	SCLK	P12[2]
J3.7	D14	GND	GND	GND
J3.8	D15	analog ref i/p	NC	VREF

(table continues...)

Table 4 (continued) E-INK shield pinout

Pin #	Arduino pin	Arduino function	E-INK shield function	Pioneer board connection
J3.9	SCL	SDA	IMU I^2C SDA	P6[1]
J3.10	SDA	SCL	IMU I^2C SCL	P6[0]
J4.1	D0	RX	NC	NC
J4.2	D1	TX	NC	NC
J4.3	D2	DIGITAL I/O	EPD reset	P5[2]
J4.4	D3	PWM, I/O	BUSY	P5[3]
J4.5	D4	DIGITAL I/O	EPD enable	P5[4]
J4.6	D5	PWM, I/O	DISCHARGE	P5[5]
J4.7	D6	PWM, I/O	BORDER	P5[6]
J4.8	D7	DIGITAL I/O	EPD I/O enable	P0[2]
J2.1	A0	ADC0	THERM VDD	P10[0]
J2.2	A1	ADC1	THERM OUT	P10[1]
J2.3	A2	ADC2	THERM OUT	P10[2]
J2.4	A3	ADC3	THERM GND	P10[3]
J2.5	A4	ADC4	PDM CLK	P10[4]
J2.6	A5	ADC5	PDM DATA	P10[5]

2 ModusToolbox™

ModusToolbox™ is a free software development ecosystem that includes the Eclipse IDE for ModusToolbox™, AIROC™ BTSTACK, Bluetooth® SDK, and PSoC™ 6 SDK to develop applications for Infineon IoT products. Eclipse IDE for ModusToolbox™ is a multi-platform, integrated development environment (IDE) used to create new applications, update application code, change middleware settings, and program or debug applications. Using ModusToolbox™, you can enable and configure device resources and middleware libraries, write C source code, and program and debug the device. The build system infrastructure includes the new project creation wizard that can run independently of the Eclipse IDE, the make infrastructure, and other tools. This means you can choose your compiler, RTOS, and ecosystem without compromising usability or access to our industry-leading CAPSENSE™ (Human-Machine Interface), AIROC™ Wi-Fi and Bluetooth®, security, and various other features.

Infineon's AIROC™ BTSTACK is a software implementation of the Bluetooth® core 5.4 host protocol stack. The stack is hosted as a library on Infineon's GitHub. The stack library includes both Bluetooth® BR/EDR and Bluetooth® Low Energy hosts and provides APIs for them. The application can choose to use Bluetooth® Low Energy or both BR/EDR + LE. The stack is available for different Arm® cores, such as CM4 and CM33, and can be used with three toolchains Arm®, GCC, and IAR.

For more details on ModusToolbox™ installation and usage, see the ModusToolbox™ user guide.

2.1 ModusToolbox™ code examples

ModusToolbox™ includes many code examples. Many of these code examples are compatible with this kit. Browse the collection of starter applications during application setup through **File** > **New** > **ModusToolbox™ Application** or browse the collection of code examples on Infineon's GitHub repository.

2.1.1 Create a new application

This part takes you step-by-step through creating a new ModusToolbox™ application. Before performing the steps in this section, decide whether you want to create and run the code example as-is or you would instead learn how to create an application from scratch. Depending on your choice, the steps you need to follow are as shown below:

Path	"Using CE directly" path(Evaluate existing code example (CE) directly)	"Working from Scratch" path(Use existing code example (CE) as reference only)
Actions	Follow the sections Select a new workspace, Create a new ModusToolbox™ application, Select PSoC™ 63 MCU-based target hardware, and Create the Bluetooth® LE Find Me code example (applicable only for the "Using CE directly" flow). Ignore section Select a starter application and create the application (Applicable only for "Working from Scratch" flow).	Follow the sections Select a new workspace, Create a new ModusToolbox™ application, Select PSoC™ 63 MCU-based target hardware and Select a starter application and create the application (Applicable only for "Working from Scratch" flow). Ignore section Create the Bluetooth® LE Find Me code example (applicable only for the "Using CE directly" flow).

Launch ModusToolbox™ and get started.

2.1.2 Select a new workspace

At launch, ModusToolbox™ presents a dialog to choose a directory for use as the workspace directory. The workspace directory is used to store workspace preferences and development artifacts such as device configuration and application source code.

You can choose an existing empty directory by clicking the **Browse** button, as Figure 6 shows. Alternatively, you can type in a directory name to be used as the workspace directory along with the complete path, and ModusToolbox™ will create the directory for you.

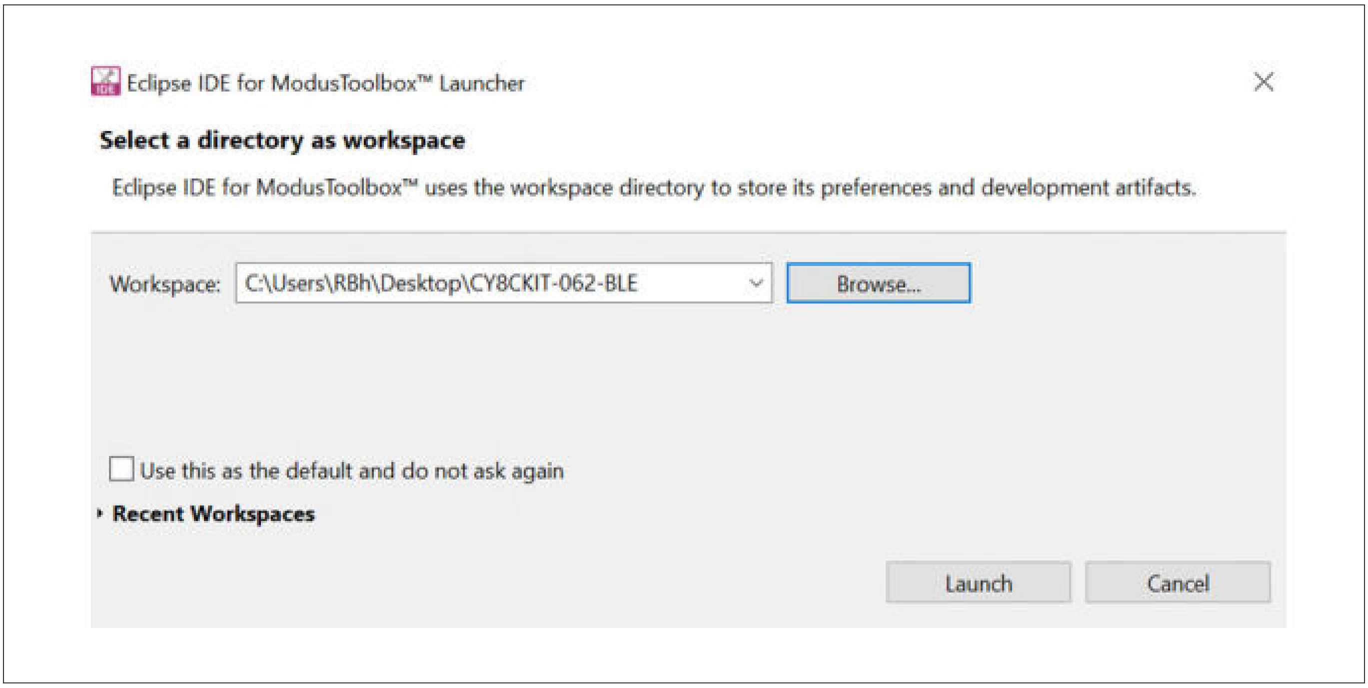


Figure 6 Select a directory as workspace

2.1.3 Create a new ModusToolbox™ application

Click **New Application** in the Start group of the Quick Panel. Alternatively, you can choose **File** > **New** > **ModusToolbox™ Application** (Figure 7).

The Eclipse IDE for ModusToolbox™ Application window appears.

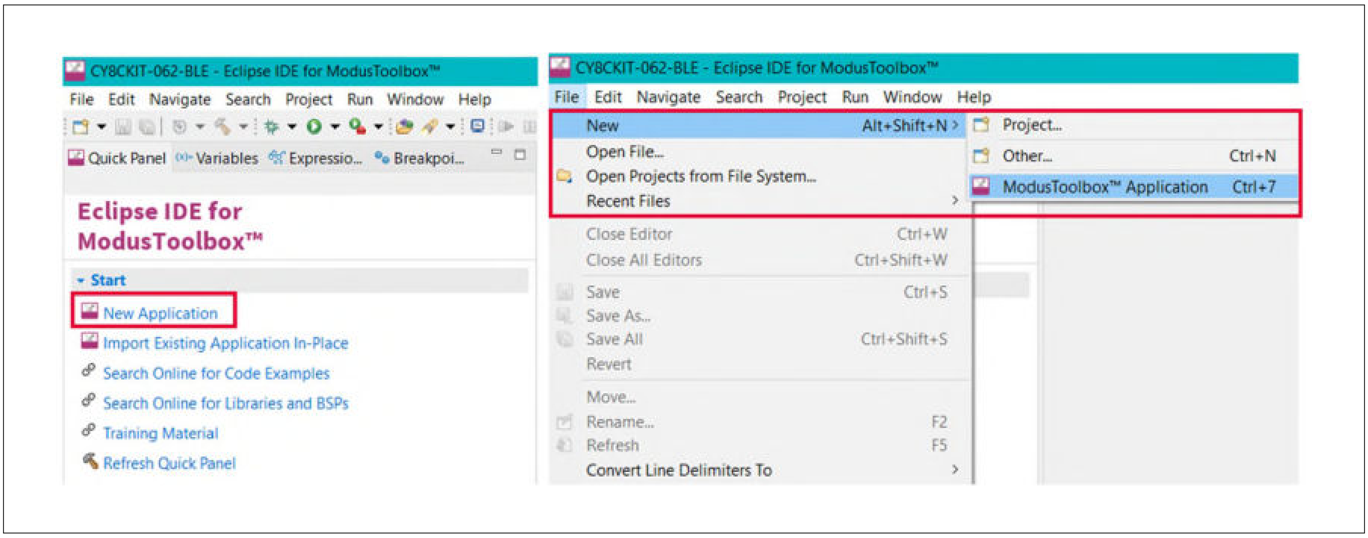


Figure 7 Create a new ModusToolbox™ application

2.1.4 Select PSoC™ 63 MCU-based target hardware

ModusToolbox™ presents the list of Infineon kits to start your application development. In this case, we want to develop an application on the CY8CKIT-062-BLE evaluation board that uses the PSoC™ 63-line device. Select **CY8CKIT-062-BLE** and click **Next** (Figure 8).

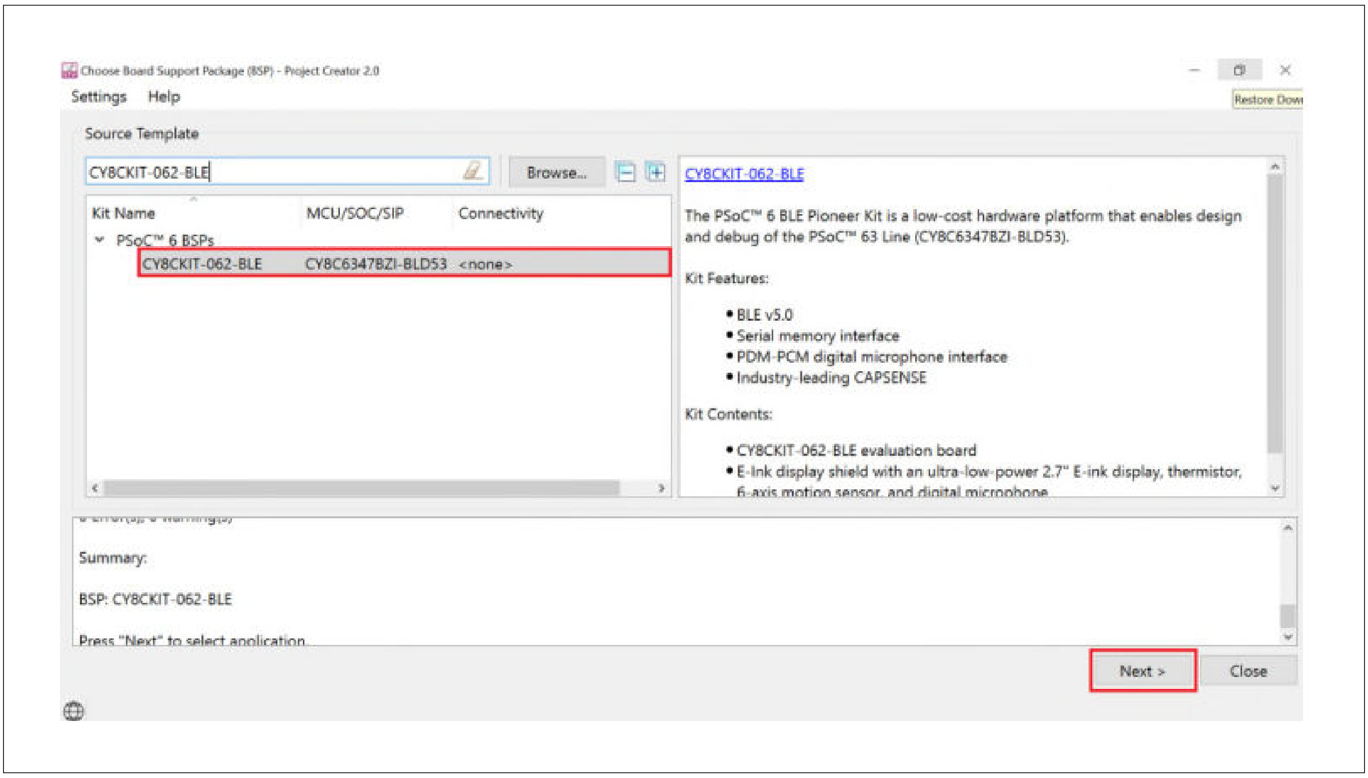


Figure 8 **Choose target hardware**

2.1.5 Create the Bluetooth® LE Find Me code example (applicable only for the "Using CE directly" flow)

Here, you **Create** an existing code example into Eclipse IDE for ModusToolbox™. Use this feature to create the Bluetooth® LE Find Me code example for the *Using CE directly flow*. Figure 9 shows the **Select Application** dialog of the project creator tool. Select the Bluetooth® LE Find Me application, and optionally, in the 'New application Name' field, change the name of the application. Click on **Create** and wait for the application to get downloaded and created in the workspace. Click on **Close** to complete the application creation process.

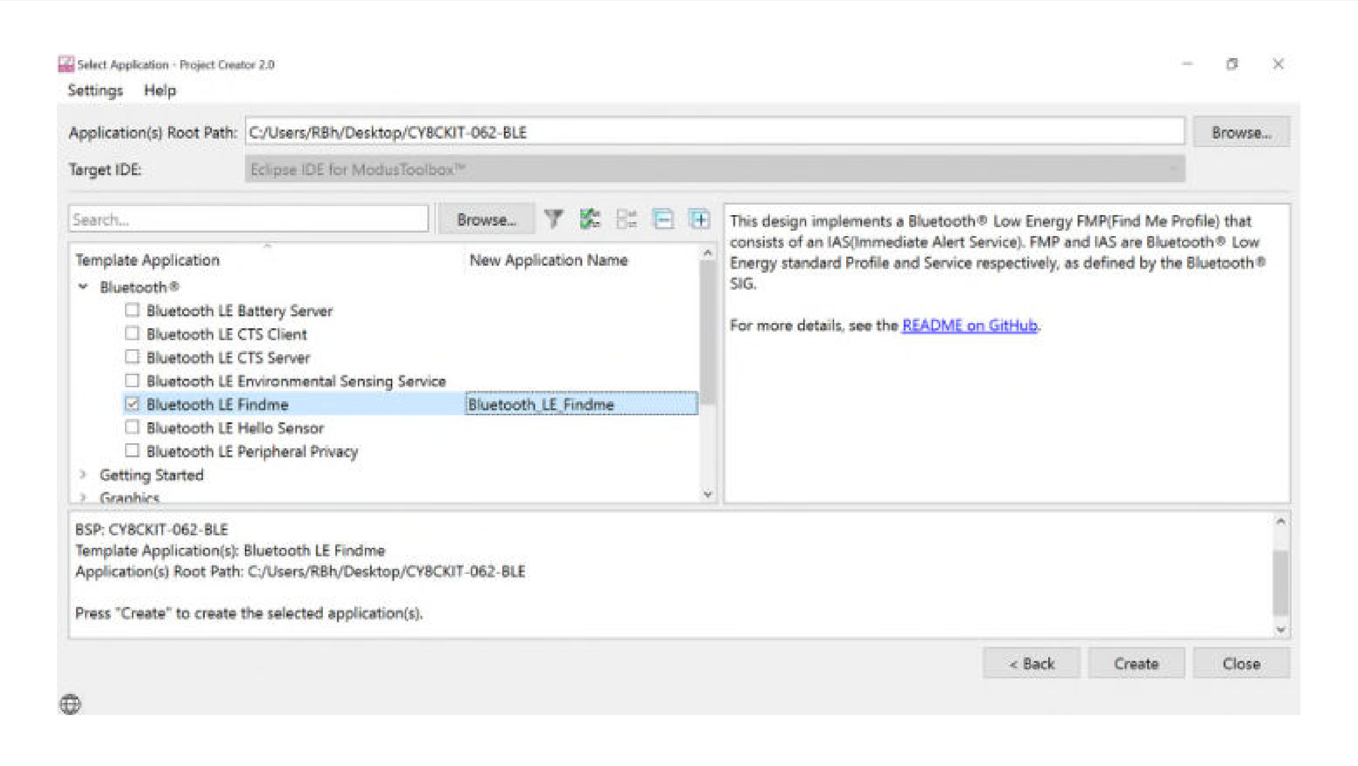


Figure 9 **Create Bluetooth® LE Find Me code example**

You have successfully created a new ModusToolbox™ application for CY8CKIT-062-BLE.

2.1.6 Select a starter application and create the application (Applicable only for "Working from Scratch" flow)

Here, you use an existing template application as the starting point for the *Working from Scratch* development flow. In the **Select Application** dialog shown in Figure 10, select **Empty_ App**. In the **Name** field, type in a name for the application and click **Next**; the application summary dialog appears. Click on **Create** and wait for the application to get downloaded and created in the workspace. Click on **Close** to complete the application creation process.

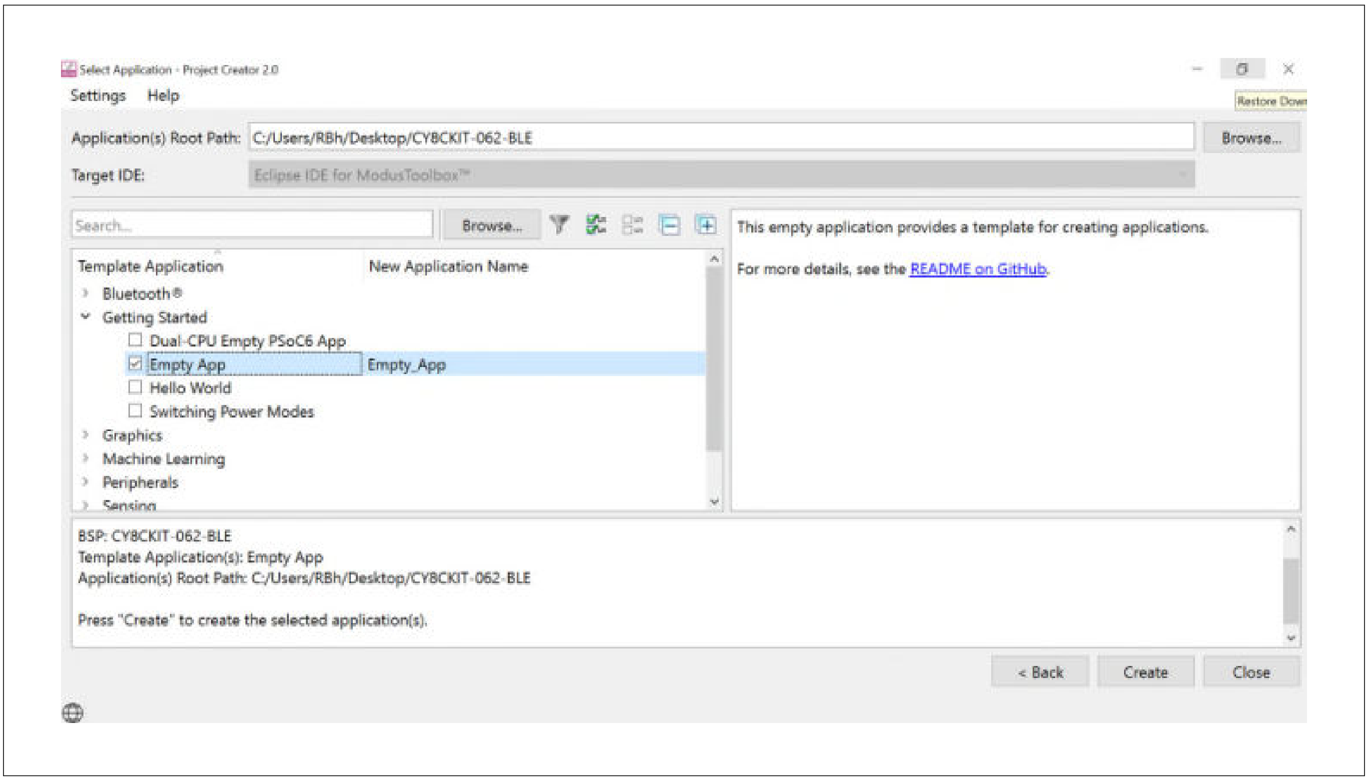


Figure 10 **Starter application window**

You have successfully created a new ModusToolbox™ application for CY8CKIT-062-BLE.

2.2 Related code examples

The kit code examples are accesses from the GitHub repository. This code example is available in ModusToolbox™. For the most recent version of the code example, check the GitHub repository. You can find the CE230297 code example in mtb-example-btstack-freertos-findme. For more details on the functionality, design, and implementation of the code example, see the `Readme` file present in the same folder as that of the code example. To import the most recent version of the GitHub code examples into ModusToolbox™, see the ModusToolbox™ user guide document (**Help** > **ModusToolbox™** > **Documentation** > **User guide**).

2.3 ModusToolbox™ help

Launch ModusToolbox™ and navigate to the following items:

- **Quick start guide**: Choose **Help** > **Eclipse IDE for ModusToolbox™ Documentation** > **Quick start guide**. This guide gives you the basics of using ModusToolbox™
- **ModusToolbox™ General Documentation**: Choose **Help** > **ModusToolbox™ General Documentation** > **ModusToolbox™ Documentation Index**. This page provides links to various ModusToolbox™ documents
- **ModusToolbox™ User guide**: Choose **Help** > **Eclipse IDE for ModusToolbox™ Documentation** > **User guide**. This is a comprehensive guide for creating, building, and programming ModusToolbox™ applications

2.4 Getting started

This guide will help you to get acquainted with the PSoC™ 6 Bluetooth® LE Pioneer Kit:

- The Kit operation chapter describes the major features of the PSoC™ 6 Bluetooth® LE Pioneer Kit and functionalities such as programming, debugging, and the USB-UART and USB-I^2C bridges

- The ModusToolbox™ code examples chapter describes multiple PSoC™ 6 MCU code examples that will help you understand how to create your own PSoC™ 6 projects
- The Appendix A provides a detailed hardware description, methods to use the onboard NOR Flash and onboard EZ-PD™ CCG3 Type-C power delivery system, kit schematics, and the bill of materials (BOM)

2.5 Additional learning resources

Infineon provides a wealth of data available on the 32-bit PSoC™ 6 Arm® Cortex®-M4/M0+ webpage 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.

2.6 IoT resources and technical support

To select the right IoT device for the design, Infineon provides a wide range of product documentation at www.infineon.com. Also, a professional community at Infineon Developer Community supplies developers with the latest software and tools to solve common evaluation and integration problems while interacting directly with both Infineon engineers and experienced peers.

For assistance, visit Infineon Support or contact customer support at +49 89 234 65555.

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

- Self-help (Technical documents)
- Local sales office locations

3 Kit operation

This chapter introduces you to various features of the PSoC™ 6 Bluetooth® LE Pioneer Kit, including the theory of operation and the onboard programming and debugging functionality, KitProg3 USB-UART, USB-I^{2}C, USB-SPI bridges, and USB Type-C power delivery.

3.1 Theory of operation

The PSoC™ 6 Bluetooth® LE Pioneer Kit is built around PSoC™ 6 MCU. Figure 11 shows the block diagram of the PSoC™ 6 MCU device used in the PSoC™ 6 Bluetooth® LE Pioneer Kit. For details of device features, see the device datasheet.

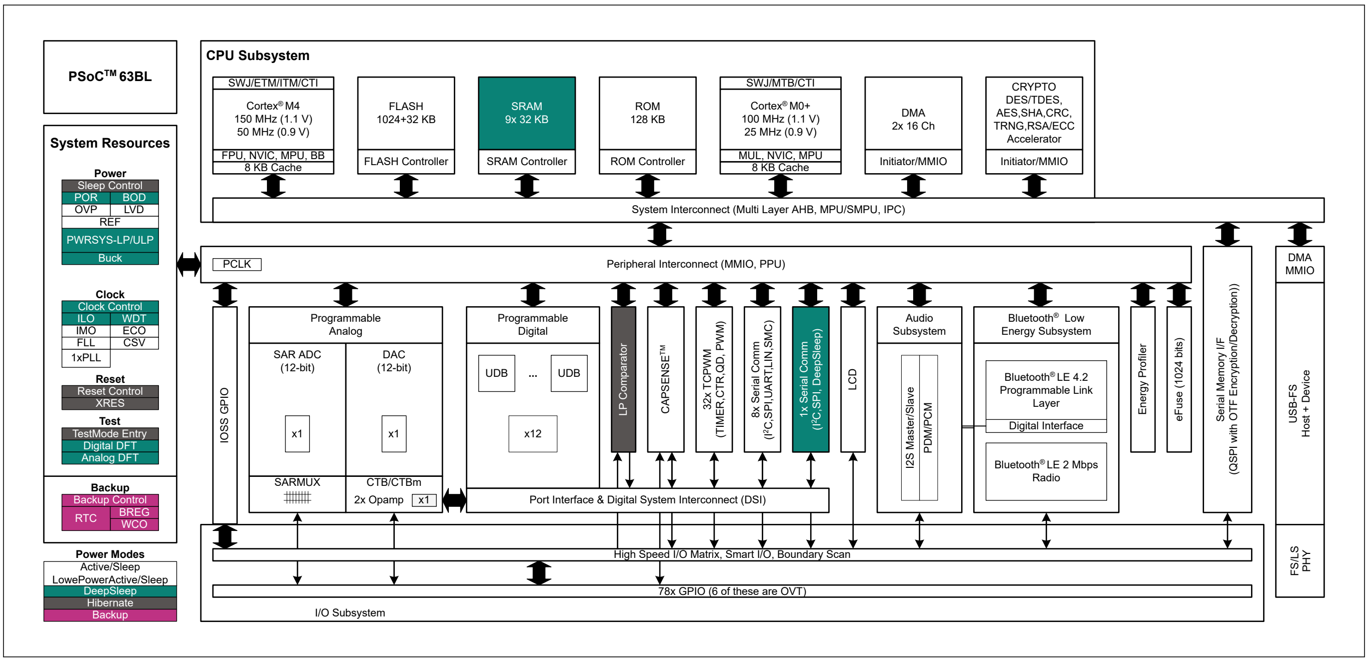


Figure 11 **PSoC™ 6 MCU block diagram**

Figure 12 shows the block diagram for the Pioneer board.

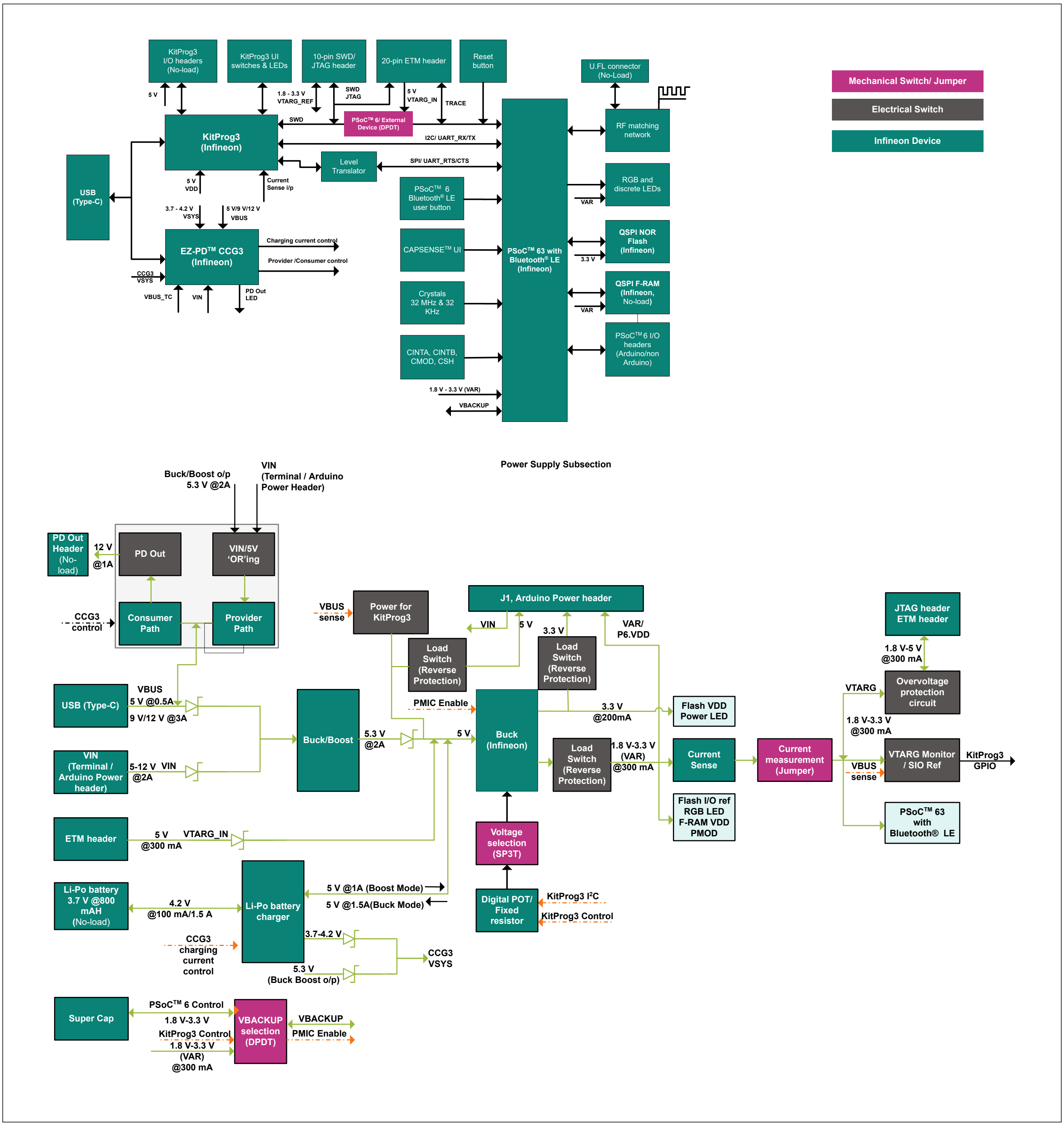


Figure 12 Block diagram of Pioneer board

The CY8CKIT-062-BLE Pioneer Kit comes with the PSoC™ 6 Bluetooth® LE Pioneer Board with the CY8CKIT-028-EPD E-INK display shield connected, as shown in Figure 13.

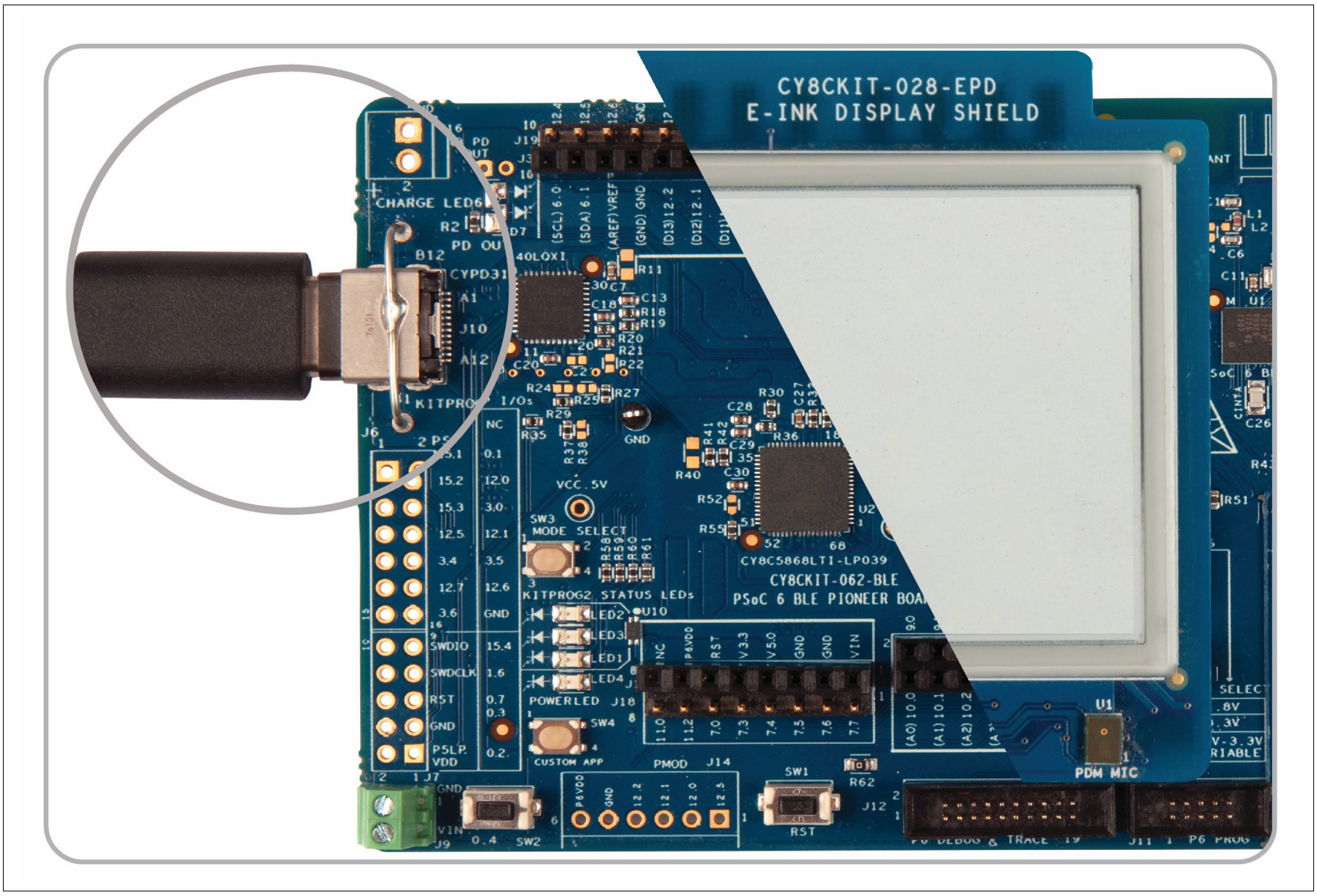
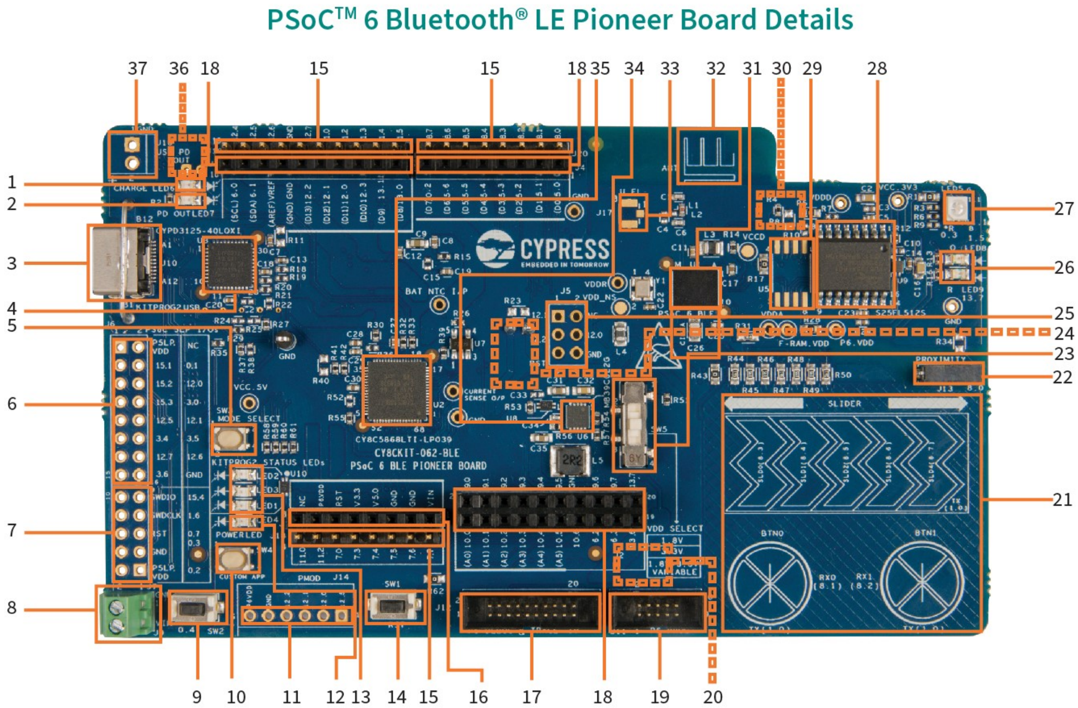


Figure 13 **PSoC™ 6 Bluetooth® LE Pioneer board and E-INK display shield**

Figure 14 shows the markup of the Pioneer board.



1 Battery charging indicator (LED6)
2 USB PD output voltage availability indicator (LED7)
3 KitProg3 USB connector (J10)
4 EZ-PD™ CCG3 Type-C Port Controller with PD (CYPD3125-40LQXI, U3)
5 KitProg3 programming mode selection button (SW3)
6 KitProg3 I/O header (J6)*
7 KitProg3 programming/custom application header (J7)*
8 External power supply connector (J9)
9 PSoC™ 6 Bluetooth® LE user button (SW2)
10 KitProg3 application selection button (SW4)
11 I/O header compatible with Digilent Pmod (J14)*
12 Power LED (LED4)
13 KitProg3 status LEDs (LED1, LED2, and LED3)
14 PSoC™ 6 Bluetooth® LE reset button (SW1)
15 PSoC™ 6 Bluetooth® LE I/O header (J18, J19 and J20)
16 Power header compatible with Arduino Uno R3 (J1)
17 PSoC™ 6 Bluetooth® LE debug and trace header (J12)
18 I/O headers compatible with Arduino Uno R3 (J2, J3, J4)
19 PSoC™ 6 Bluetooth® LE program and debug header (J11)
20 KitProg3 programming target selection switch (SW6)**
21 CAPSENSE™ slider and buttons
22 CAPSENSE™ proximity header (J13)
23 PSoC™ 6 Bluetooth® LE VDD selection switch (SW5)
24 PSoC™ 6 Bluetooth® LE power monitoring jumper (J8)**
25 ICSP header compatible with Arduino Uno R3 (J5)*
26 PSoC™ 6 Bluetooth® LE user LEDs (LED8 and LED9)
27 RGB LED (LED5)
28 512-Mbit serial NOR flash memory (S25FL512S, U4)
29 Serial Ferroelectric RAM (U5)*
30 Vbackup and PMIC control selection switch (SW7)
31 PSoC™ 6 Bluetooth® LE (CY8C6347BZI-BLD53, U1)
32 Bluetooth® LE antenna
33 U.FL connector for external antenna (J17)*
34 Main voltage regulator (MB39C022G, U6)
35 KitProg3 (PSoC™ 5LP) programmer and debugger (CY8C5868LTI-LP039, U2)
36 Battery connector (J15)**
37 USB PD output voltage (9V/12V) connector (J16)*

*Footprints only, not populated on the board.
**Components at the bottom side of the board.

Figure 14 **PSoC™ 6 Bluetooth® LE Pioneer board - Top view**

The PSoC™ 6 Bluetooth® LE Pioneer board has the following peripherals:

1. **Battery charging indicator (LED7):** This LED turns ON when the onboard battery charger is charging a lithium-iron polymer battery connected to J15. Note that the battery connector and battery are not included in the kit and should be purchased separately if you have to test the battery charging functionality

2. **USB PD out indicator (LED6):** This LED turns ON when the USB Type-C power delivery output is available for use
3. **KitProg3 USB connector (J10):** The USB cable provided along with the PSoC™ 6 Bluetooth® LE Pioneer Kit connects between this USB connector and the PC to use the KitProg3 onboard programmer and debugger and to provide power to the Pioneer board. J10 is also used for the USB Type-C power delivery system. See EZ-PD™ CCG3 Type-C power delivery for more details
4. **Infineon EZ-PD™ CCG3 Type-C Port Controller with PD (CYPD3125-40LQXIT, U3):** The Pioneer Board includes a Infineon EZ-PD™ CCG3 USB Type-C Port controller with Power Delivery system. This EZ-PD™ CCG3 device is pre-programmed and can deliver power from a Type-C port to onboard header J16, while simultaneously charging a lithium-ion polymer battery connected to J15. In addition, the power delivery system can deliver power to a Type-C power sink or consumer such as a cell phone with the power derived from the VIN supply. See EZ-PD™ CCG3 Type-C power delivery for more details
5. **KitProg3 programming mode selection button (SW3):** This button can be used to switch between various modes of operation of KitProg3 (Proprietary SWD Programming or Mass Storage programming/ CMSIS-DAP mode). This button can also be used to provide input to PSoC™ 5LP in custom application mode. For more details, see the KitProg3 User guide
6. **KitProg3 I/O header (J6):** This header brings out several GPIOs of the onboard KitProg3 PSoC™ 5LP device. This includes the USB-I^{2}C, USB-UART, and USB-SPI bridge lines. The additional PSoC™ 5LP pins are direct connections to the internal programmable analog logic of the PSoC™ 5LP. You can also use these pins for custom applications. For more details on the KitProg3, see the KitProg3 User guide
7. **KitProg3 programming/custom application header (J7):** This header brings out more GPIOs of the PSoC™ 5LP, which can be used for custom applications. It also contains a 5-pin SWD programming header for the PSoC™ 5LP
8. **External Power Supply VIN connector (J9):** This connector connects an external DC power supply input to the onboard regulators and the USB Type-C power delivery system. The voltage input from the external supply should be between 5 V and 12 V. Moreover, when used as an input to the USB Type-C power delivery system, the external power supply should have enough current capacity to support the load connected via the Type-C port. See EZ-PD™ CCG3 Type-C power delivery for more details
9. **PSoC™ 6 MCU user button (SW2):** This button can be used to provide an input to PSoC™ 6 MCU. Note that by default the button connects the PSoC™ 6 MCU pin to ground when pressed, so you need to configure the PSoC™ 6 MCU pin as a digital input with resistive pull-up for detecting the button press. This button also provides a wake-up source from low-power modes of the device. In addition, this button can be used to activate the regulator control output from PSoC 6 MCU
10. **KitProg3 application selection button (SW4):** This button can be used to switch between KitProg3 programming mode and custom application mode. For more details, see the KitProg3 User guide
11. **Digilent Pmod compatible I/O header (J14):** This header can be used to connect Digilent Pmod 1 x 6 pin modules
12. **Power LED (LED4):** This is the amber LED that indicates the status of power supplied to PSoC™ 6 MCU
13. **KitProg3 status LEDs (LED1, LED2, and LED3):** Red, amber and green LEDs (LED1, LED2, and LED3 respectively) indicate the status of KitProg3. For details on the KitProg3 status, see the KitProg3 User guide
14. **PSoC™ 6 MCU reset button (SW1):** This button is used to reset PSoC™ 6 MCU. This button connects the PSoC™ 6 MCU reset (XRES) pin to ground
15. **PSoC™ 6 MCU I/O headers (J18, J19, and J20):** These headers provide connectivity to PSoC™ 6 MCU GPIOs that are not connected to the Arduino compatible headers. Majority of these pins are multiplexed with onboard peripherals and are not connected to PSoC™ 6 MCU by default. For the detailed information on how to rework the kit to access these pins, see
16. **Arduino compatible power header (J1):** The Arduino-compatible power headers power Arduino shields. This header also has a provision to power the kit though the VIN input
17. **PSoC™ 6 MCU debug and trace header (J12):** This header can be connected to an Embedded Trace Macrocell (ETM) compatible programmer/debugger

18. **Arduino Uno R3 compatible I/O headers (J2, J3, and J4):** The Arduino-compatible I/O headers bring out pins from PSoC™ 6 MCU to interface with the Arduino shields. Few of these pins are multiplexed with onboard peripherals and are not connected to PSoC™ 6 MCU by default. For a detailed information on how to rework the kit to access these pins, see
19. **PSoC™ 6 MCU program and debug header (J11):** This 10-pin header allows you to program and debug the PSoC™ 6 MCU using an external programmer such as MiniProg4. In addition, an external PSoC™ 4/5/6 device can be connected to this header and programmed using KitProg3. To program the external device, SW6 should be used to select the "External" option
20. **KitProg3 programming target selection switch (SW6, on the bottom side of the board):** This switch selects the programming target of the onboard KitProg3 between the onboard PSoC™ 6 MCU and an external PSoC™ 4/5/6 device connected to J11
21. **CAPSENSE™ slider (SLIDER) and buttons (BTN0 and BTN1):** CAPSENSE™ touch-sensing slider and two buttons, all of which are capable of both self-capacitance (CSD) and mutual-capacitance (CSX) operation, let you evaluate Infineon' fourth-generation CAPSENSE™ technology. The slider and the buttons have a 1-mm acrylic overlay for smooth touch sensing
22. **CAPSENSE™ proximity header (J13):** A wire can be connected to this header to evaluate the proximity sensing feature of CAPSENSE™
23. **System Power VDD selection switch (SW5):** This switch is used to select the PSoC™ 6 MCU's VDD supply voltage between constant 1.8 V, constant 3.3 V, and variable 1.8 to 3.3 V. In the variable 1.8 to 3.3 V mode, the PSoC™ programmer software can control the voltage via the KitProg3
24. **PSoC™ 6 MCU current measurement jumper (J8, on the bottom side of the board):** An ammeter can be connected to this jumper to measure the current consumed by the PSoC™ 6 MCU
25. **Arduino compatible ICSP header (J5):** This header provides an SPI interface for Arduino ICSP compatible shields
26. **PSoC™ 6 MCU user LEDs (LED8 and LED9):** These two user LEDs can operate at the entire operating voltage range of PSoC™ 6 MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs
27. **RGB LED (LED5):** This onboard RGB LED can be controlled by the PSoC™ 6 MCU. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs
28. **Infineon 512-Mbit serial NOR flash memory (S25FL512SAGMFI011, U4):** This kit features a Infineon NOR flash (S25FL512SAGMFI011) of 512 Mb capacity. The NOR Flash is connected to the serial memory interface (SMIF) of the PSoC™ 6 MCU. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption
29. **Infineon 4-Mbit serial Ferroelectric RAM (FM25V10, U5):** Footprint to connect a FM25V10 or any other pin compatible FRAM
30. **Vbackup and PMIC control selection switch (SW7, on the bottom side of the board):** This switches the Vbackup supply connection to PSoC™ 6 MCU between V_{DDD} and the super-capacitor. When V_{DDD} is selected, the regulator ON/OFF is controlled by the KitProg3. When super-capacitor is selected, the regulator ON/OFF is controlled by PSoC™ 6 MCU
31. **Infineon PSoC™ 6 MCU (CY8C6347BZI-BLD53, U1):** This kit is designed to highlight the features of the PSoC™ 6 MCU. For details on PSoC™ 6 MCU pin mapping, see Table 12
32. **BLE antenna:** This is the onboard wiggle antenna for BLE
33. **U.FL connector (J17):** This connector can be used for conductive measurements and also to connect external antenna
34. **Infineon main voltage regulator (MB39C022GPN-G-ERE1, U6):** This is the main regulator that powers PSoC™ 6 MCU. This regulator has two output channels. One channel provides fixed LDO-based 3.3 V output from 5 V input and the other channel is a buck DC to DC converter that is configured to provide variable voltage from 1.8 V to 3.3 V
35. **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 programmer,

debugger, USB-I²C bridge, USB-UART bridge, and a USB-SPI bridge. KitProg3 also supports custom applications. For more details, see the KitProg3 User guide

36. **Battery connector (J15, on the bottom side of the board):** This connector can be used to connect a lithium-ion polymer battery. Note that a battery is not included in the kit package and should be purchased separately if you want to demonstrate battery charging

37. **USB PD output (J16):** This header provides a voltage output when the USB Type-C power delivery system receives power from an external host connected to J10. See EZ-PD™ CCG3 Type-C power delivery for more details

See Appendix A.2 for details on various hardware blocks.

For some devices in the PSoC™ 6 MCU family, simultaneous GPIO switching with unrestricted drive strengths and frequency can induce noise in on-chip subsystems affecting CAPSENSE™ and ADC results. For more details, see the Errata section of the corresponding device datasheet.

3.1.1 CY8CKIT-028-EPD E-INK display shield

The E-INK display shield has the following peripherals:

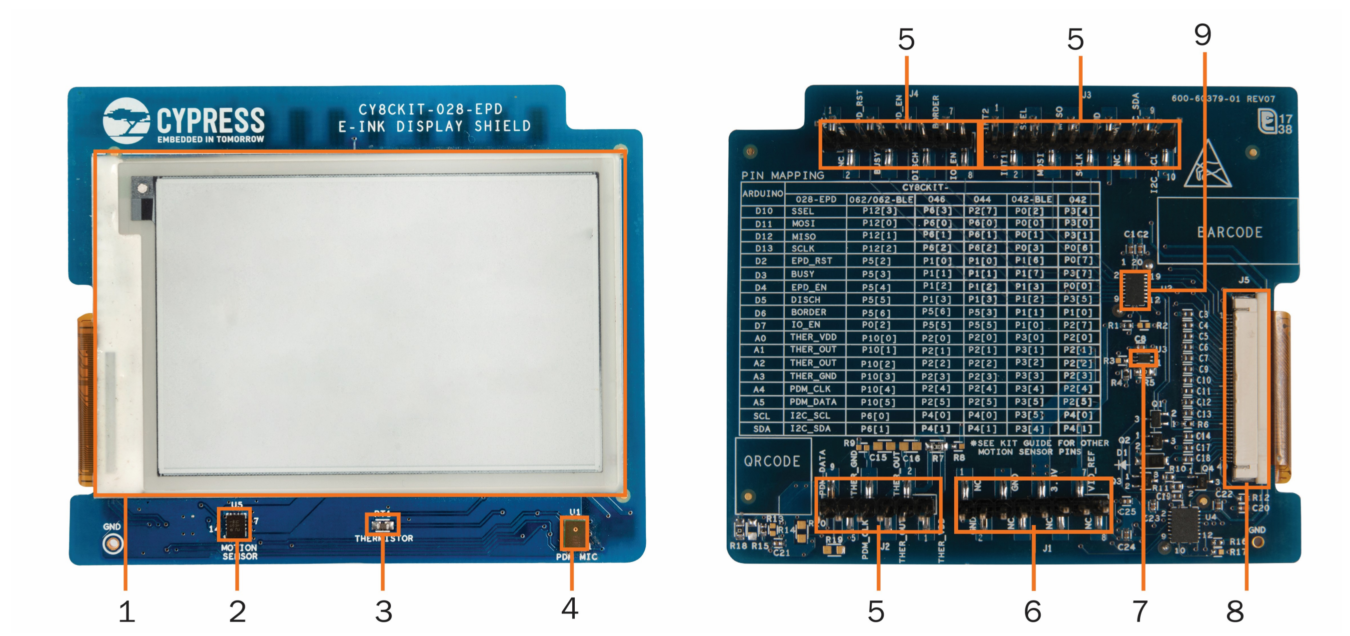


Figure 15 E-INK display shield

1. **2.7 inch E-INK display:** This is a monochrome E-INK display with a resolution of 264 x 176. The E-INK display can retain its contents even in the absence of power, which provides a low-power, "always-on" display functionality
2. **Motion sensor (U5):** This is a 3-axis acceleration and 3-axis gyroscopic motion sensor that can be used to count steps to emulate a pedometer or similar application
3. **Thermistor(RT1):** This thermistor can be used for temperature compensation of the display or as a general purpose ambient temperature sensor
4. **PDM microphone (U1):** This microphone converts voice inputs to pulse-density modulated (PDM) digital signals
5. **Arduino compatible I/O header (J2, J3 and J4):** This header interfaces with the PSoC™ 6 MCU GPIOs through header J2 on the board
6. **Arduino compatible power & I/O header (J1):** This header receives power from header J1 on the board
7. **E-INK display power control load switch (U3):** This load switch can be controlled by the board to toggle the E-INK display's power

8. **E-INK display connector (J5):** This connector is used to connect the E-INK display to the circuits on the E-INK display shield
9. **E-INK display I/O voltage translator (U2):** This I/O level translator allows the board to operate at any voltage between 1.8 and 3.3 V by providing a constant 3.3 V interface to the E-INK display

3.2 KitProg3

The PSoC™ 6 Bluetooth® LE Pioneer Kit can be programmed and debugged using the onboard KitProg3. The KitProg3 is a multi-functional system, which includes a programmer, debugger, USB-I^{2}C bridge, USB-UART bridge, and a USB-SPI bridge. KitProg3 also supports mass storage programming and CMSIS-DAP, and custom applications. Infineon's PSoC™ 5LP device is used to implement KitProg3 functionality. The KitProg3 is integrated in most PSoC™ development kits. For more details on the KitProg3 functionality, see the KitProg3 User guide.

3.2.1 Programming and debugging using ModusToolbox™

You can build and program the application in ModusToolbox™. In the project explorer, select the **<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 16.

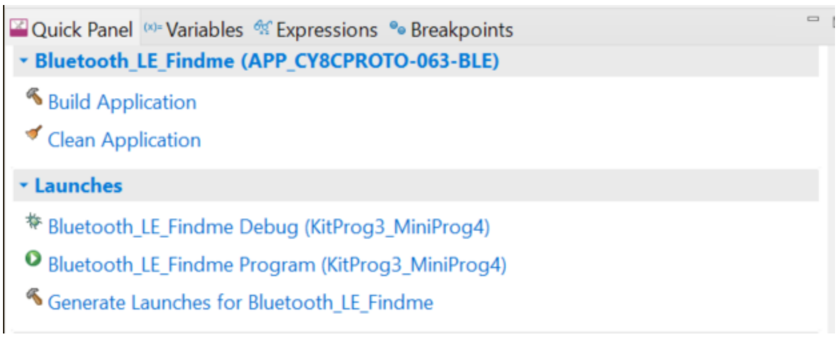


Figure 16 **Programming the CY8CKIT-062-BLE device using ModusToolbox™**

3.2.2 Programming using PSoC™ Programmer

PSoC™ Programmer can be used to program existing `.hex` files into the PSoC™ 6 Bluetooth® LE Pioneer Kit. For a detailed explanation on how to program using PSoC™ Programmer, see the **Programming Using PSoC™ Programmer** section in the KitProg3 User guide.

The KitProg3 firmware normally does not require any update. If necessary you can use the PSoC™ Programmer software to update the KitProg3 firmware. For a detailed explanation on how to update the KitProg3 firmware, see the **Updating the KitProg3 Firmware** section in the KitProg3 User guide.

3.2.3 Mass Storage Programmer

The KitProg3 in the PSoC™ 6 Bluetooth® LE Pioneer Kit supports programming through a USB mass storage interface. This interface allows you to program PSoC™ 6 MCU by copying `.hex` files into an emulated USB mass storage device. The user can press the mode button to switch to mass storage mode. At that time, the Amber LED will show a breathing effect. Press the mode button again to switch to the normal programming mode. For more details on KitProg3 Mass Storage Programmer, see the KitProg3 User guide.

3.2.4 USB-UART bridge

The KitProg3 on the PSoC™ 6 Bluetooth® LE Pioneer Kit can act as a USB-UART bridge. The UART and flow-control lines between the PSoC™ 6 MCU and the KitProg3 are hard-wired on the board, as Figure 17 shows. For more details on the KitProg3 USB-UART functionality, see the KitProg3 User guide.

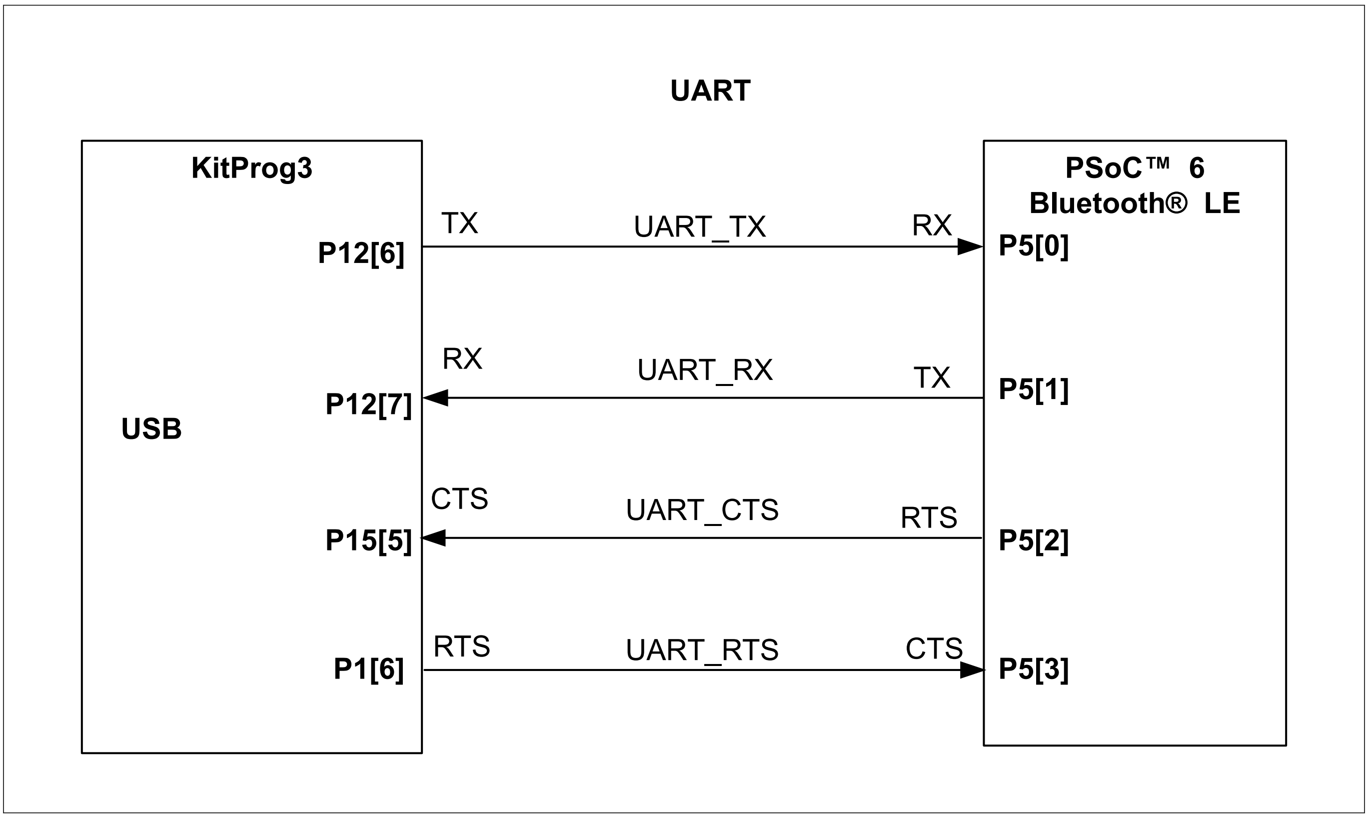


Figure 17 UART connection between KitProg3 and PSoC™ 6

3.2.5 USB-I^{2}C bridge

The KitProg3 can function as a USB-I^{2}C bridge and communicate with the Bridge Control Panel (BCP) software. The I^{2}C lines on the PSoC™ 6 MCU are hard-wired on the board to the I^{2}C lines of the KitProg3, with onboard pull-up resistors as Figure 18 shows. The USB-I^{2}C supports I^{2}C speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg3 USB-I^{2}C functionality, see the KitProg3 User guide.

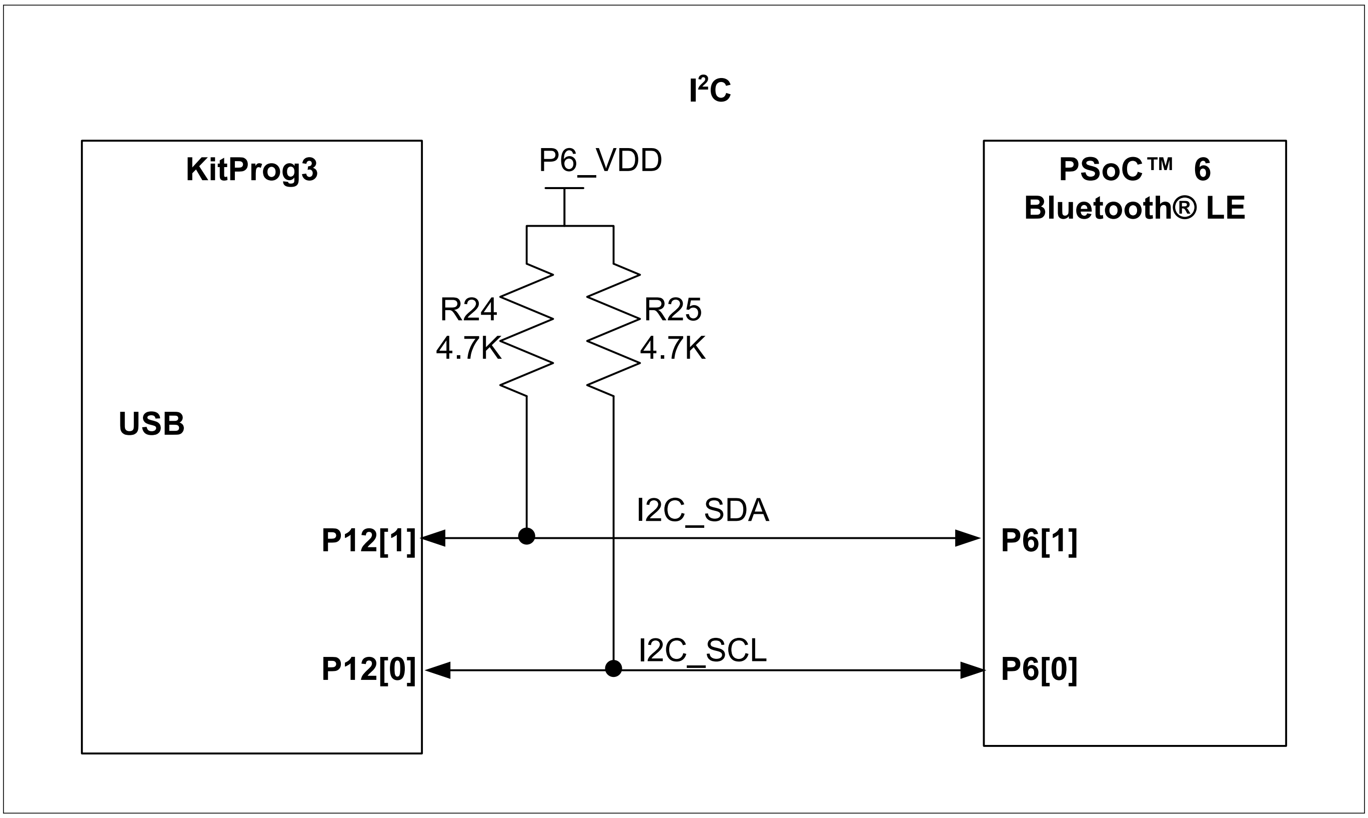


Figure 18 I²C connection between KitProg3 and PSoC™ 6

3.2.6 USB-SPI bridge

The KitProg3 can function as a USB-SPI bridge. The SPI lines between the PSoC™ 6 MCU and the KitProg3 are hard-wired on the board, as Figure 19 shows. For more details on the KitProg3 USB-SPI functionality, see the KitProg3 User guide.

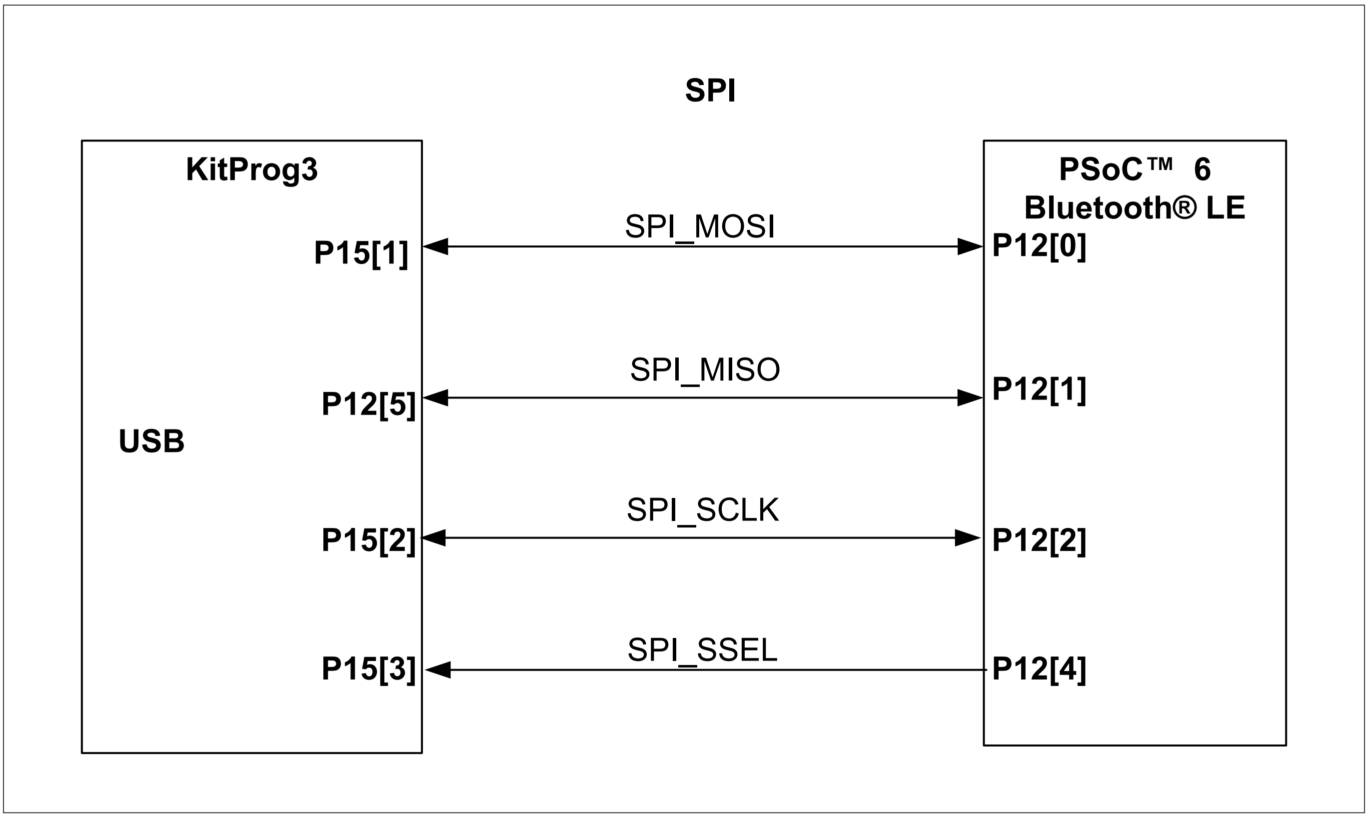


Figure 19 **SPI connection between KitProg3 and PSoC™ 6**

3.3 EZ-PD™ CCG3 Type-C power delivery

The Pioneer board includes an Infineon EZ-PD™ CCG3 power delivery system. This EZ-PD™ CCG3 is pre-programmed and can deliver power from a Type-C port to onboard header **J16** (known as the consumer path), while simultaneously charging a 3.7 V, lithium-ion polymer battery connected to **J15**. In addition, the power delivery system can deliver power to a Type-C peripheral such as a cell phone with the power derived from the VIN **(J9)** supply (known as the provider path). Note that to use the EZ-PD™ CCG3 Type-C power delivery system, a power delivery capable USB Type-C to Type-C cable should be connected to **J10**. This cable is not included in the kit, and should be purchased separately.

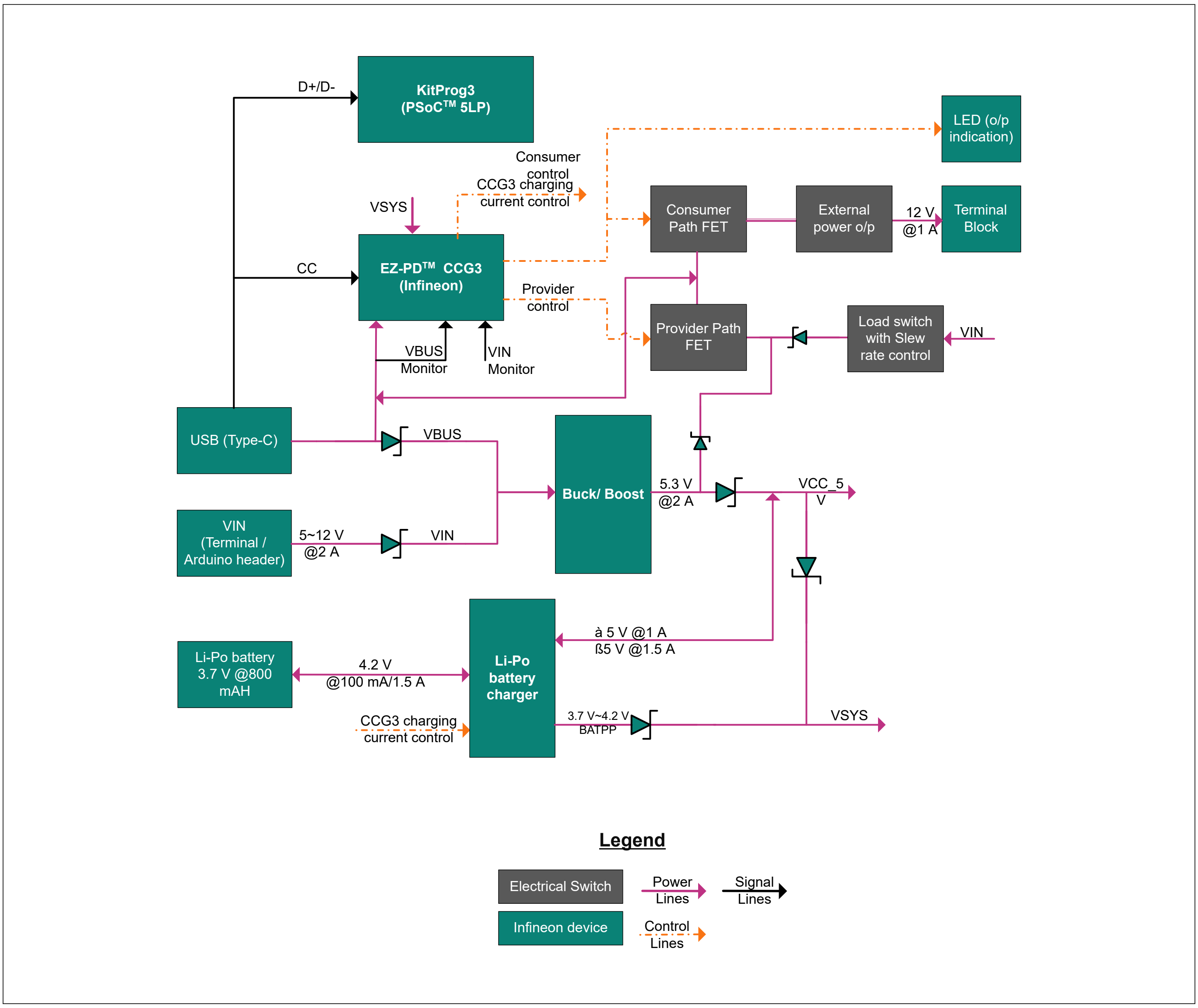


Figure 20 **Type-C block diagram**

The power delivery system works as follows:

1. If the power delivery system detects a non Type-C power adapter (Legacy USB), CCG3 will charge the battery at 100 mA. CCG3 will also disable the consumer and provider paths
2. On detection of a Type-C power adapter, CCG3 will request 5 V at 3 A, 9 V at 3 A, or 12 V at 3 A depending on the host capability. Once the power level is successfully negotiated, the Consumer path is enabled by turning on load switch **U12**. This load switch is hardware limited to supply up to 1 A through header **J16** to an external device. CCG3 will use the remaining current to charge the battery connected to **J15** at a higher charging rate up to 1.5 A and PD output voltage availability indicator **(LED6)** will be turned ON
3. CCG3 will also advertise that it can provide 5 V, 9 V, or 12 V if a DC power supply capable of providing either of these voltages is connected at VIN **(J9)**. The current is limited in this case to 1 A. Note that the external supply must be capable of providing this current. If a connected, Type-C device requests power, the provider path is enabled by turning on load switch **U22**. Table 5 details the power delivery scenarios for onboard CCG3

Table 5 Type-C table power delivery scenarios

USB Host/ consumer capability	VIN	Consumer capability	Provider capability	External USB PD out (J16 header)	Battery charging current
Non Type-C Power adapter (Legacy USB)	<5V	N/A	0	0	100 mA
	>5V	N/A	0	0	0
Type-C, PD power adapter (12 V capable)	<12V	12 V@3 A	0	12 V@1 A[1)]	1.5 A max
	>12V	N/A	0	0	0
Type-C, capable of providing max 9 V[2)]	<9V	9 V@3 A	0	9 V@1 A	1.5 A max
	>9V	N/A	0	0	0
Type-C only, capable of providing max 5 V[2)]	<5	5 V@3 A	0	5 V@1 A	1.5 A max
	>5	N/A	0	0	0
Type-C, requesting 12 V[2)]	≠12V	0	5 V@1 A	0	0
	12V	0	12 V@1 A	0	0
Type-C, requesting 9 V[2)]	≠9V	0	5 V@1 A	0	0
	9V	0	9 V@1 A	0	0
Type-C, requesting 5 V[2)]	≠5V	0	5 V@1 A	0	0
	5V	0	5 V@1 A	0	0
Type-C, requesting another voltage[2)]	5V < VIN <12V	0	5 V@1 A	0	0

1) Due to the voltage drop in series components, the voltage at J16 is ~9 V when 12 V PD power adapter is used. Populate R79 resistor to bypass this drop.

2) The table is valid only if Type-C cable is connected first and then VIN is applied. If VIN is applied first, consumer capability will be N/A.

For more information on USB Type-C power delivery with CCG3 device, see the EZ-PD™ CCG3 web page.

4 PSoC™ 6 Bluetooth® LE Pioneer Board Rework

4.1 Bypassing protection circuit on PSoC™ 6 MCU program and debug header (J11)

The 10-pin header allows you to program and debug PSoC™ 6 MCU using an external programmer such as MiniProg4. This header has a protection circuit that cuts-off any voltage greater that 3.4 V on *VTARG_REF* pin. This is to ensure that PSoC™ 6 MCU and other 3.3 V devices do not get damaged due to overvoltage.

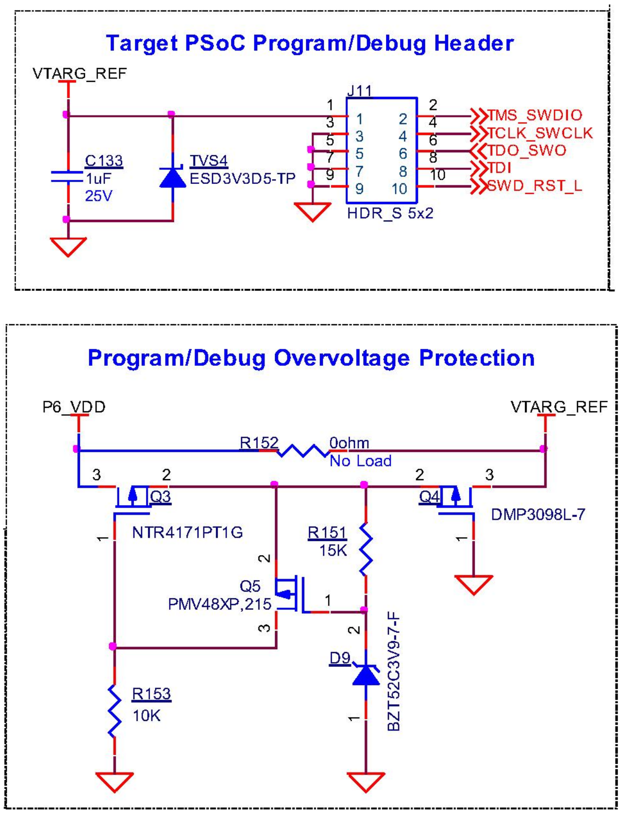


Figure 21 **Schematics of Bypassing protection circuit**

In case the external programmer provides slightly higher voltage, say 3.42 V, and you need to still use the programmer, you can bypass this protection circuit by populating the bypass zero-Ω resistor R152.

Do note that this change will compromise the protection circuit when an external supply is used and will permanently damage any 3.3 V device if the external voltage exceeds absolute maximum limit of devices. For example, 3.6 V for PSoC™ 6 MCU device, see the respective device datasheet for absolute maximum voltage limits.

4.2 PSoC™ 6 MCU User button (SW2)

By default, this button connects the PSoC™ 6 MCU pin to ground when pressed, and you need to configure the PSoC™ 6 MCU pin as a digital input with resistive pull-up for detecting the button press. In case you need to sense active HIGH on PSoC™ 6 MCU pin, resistor R150 should be removed and R149 should be populated. This will connect the button connecting the PSoC™ 6 MCU pin to V_{DDD} when pressed. Additionally, there are footprints provided for pull-up and pull-down resistors that can be populated in case external pull-up is required.

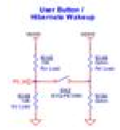


Figure 22 Schematics of PSoC™ 6 MCU User button (SW2)

4.3 CAPSENSE™ Shield

The hatched pattern around the CAPSENSE™ buttons and slider are connected to ground. In case liquid tolerance is required, this pattern needs to be connected to shield pin. This pattern can be connected to either of the two ports P6.3 or P13.6 populated by R138 or R137, respectively. In both cases, resistor R144 connecting the hatched pattern to ground needs to be removed. These pins need to be configured as shield pin in PSoC™ creator.

Connecting the hatched pattern to shield instead of ground will also reduce parasitic capacitance of the sensors.

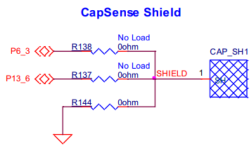


Figure 23 Schematics of CAPSENSE™ Shield

4.4 CSH

The shield tank capacitor (CSH) is not populated by default. This capacitor is optional, and can be used for an improved shield electrode driver when CSD sensing is used. You can remove R88 to disconnect port 7.3 from header and populate C88 (10 nF) for CSH. See the bill of materials (BOM) for recommended part number.

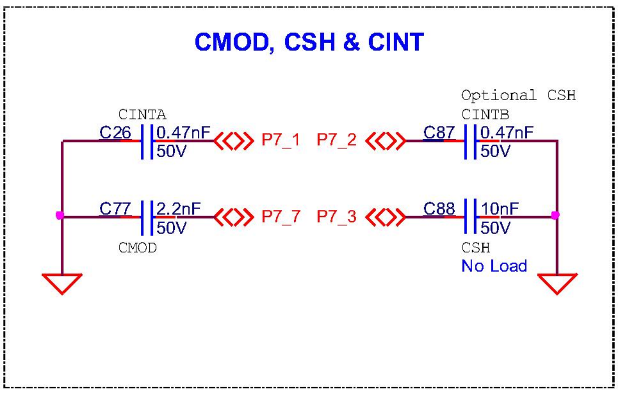


Figure 24 Schematics of CMOD, CSH and CINT

4.5 U.FL

This connector can be used for conductive measurements and can also be used to connect external antenna. This is not loaded by default. Remove L1, C1, populate L2, C4 and populate the U.FL connector (J17) to connect ANT pin from PSoC™ 6 MCU to connector. See the BOM for recommended part numbers.

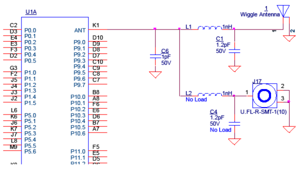


Figure 25 **Schematics of U.FL connector (J17)**

4.6 LiPo Battery Charger

Battery connector (J15) for lithium-ion polymer battery charger is not loaded by default, this need to populate to evaluate battery charging and battery powering option. See the BOM for recommended part numbers. Recommended lithium-ion polymer rate is 3.7 V @850 mAH or higher. SparkFun Electronics PRT-13854 or equivalent. batteries can be used.

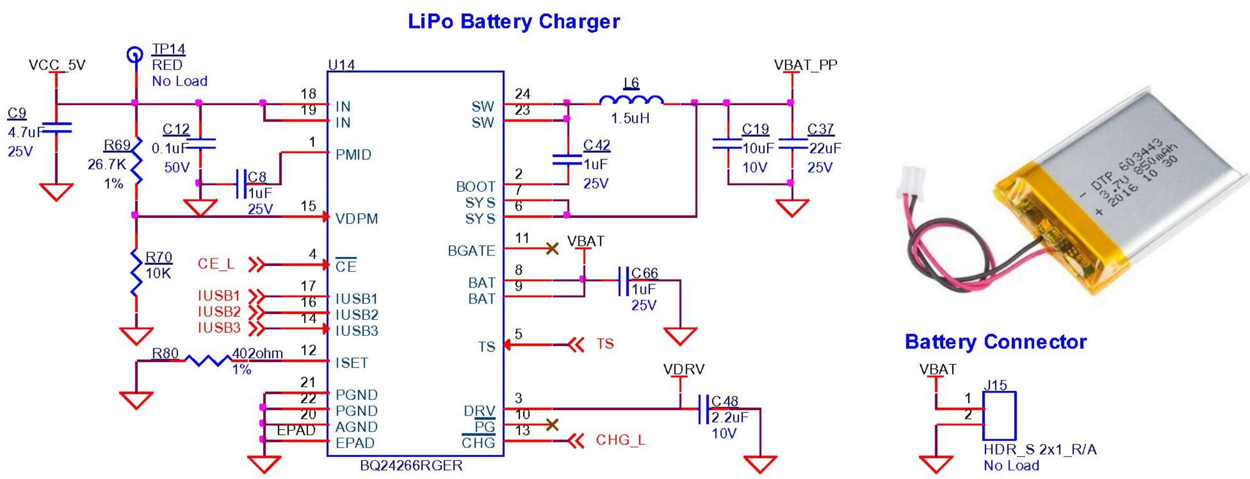


Figure 26 **Schematics of LiPo Battery Charger**

4.7 Multiplexed GPIOs

Some of the PSoC™ 6 MCU pins are multiplexed with onboard peripherals and are not connected to connectors or other secondary components by default. See the PSoC™ 6 pin mapping table for details on modification required to access these pins.

4.8 Bill of materials

Refer to the BOM files accessible via www.infineon.com/cy8ckit-062-ble.

A Appendix

A.1 Schematics

Refer to the schematics files available in the kit installation directory, accessible via www.infineon.com/cy8ckit-062-ble.

A.2 Hardware functional description

This section explains in detail the individual hardware blocks of the PSoC™ 6 Bluetooth® LE Pioneer board.

A.3 PSoC™ 6 MCU (U1)

PSoC™ 6 MCU is Infineon's latest, ultra-low-power PSoC™ specifically designed for wearables and IoT products. PSoC™ 6 MCU is a true programmable embedded system-on-chip, integrating a 150-MHz Arm® Cortex®-M4 as the primary application processor, a 100-MHz Arm® Cortex®-M0+ that supports low-power operations, up to 1 MB Flash and 288 KB SRAM, an integrated Bluetooth® LE 4.2 radio, CAPSENSE™ touch-sensing, and custom analog and digital peripheral functions. The programmable analog and digital peripheral functions allow higher flexibility, in-field tuning of the design, and faster time-to-market.

For more information, see the PSoC™ 6 MCU web page and the datasheet.

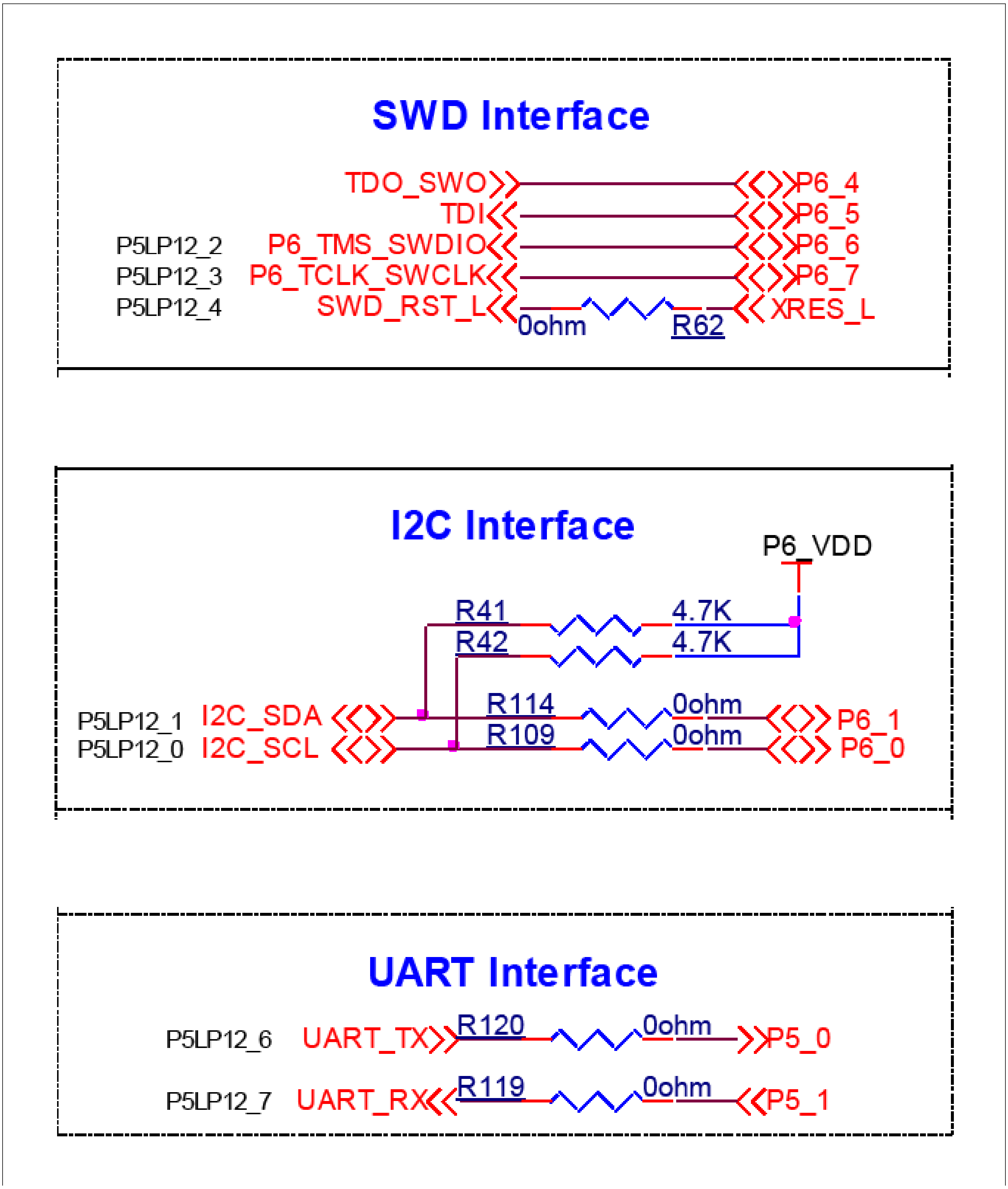
A.4 PSoC™ 5LP (U2)

An onboard PSoC™ 5LP (CY8C5868LTI-LP039) is used as KitProg3 to program and debug PSoC™ 6 MCU. The PSoC™ 5LP connects to the USB port of a PC through a USB connector and to the SWD and other communication interfaces of PSoC™ 6 MCU. The PSoC™ 5LP is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. The CY8C58LPxx family offers a modern method of signal acquisition, signal processing, and control with high accuracy, high bandwidth, and high flexibility. Analog capability spans the range from thermocouples (near DC voltages) to ultrasonic signals.

For more information, visit the PSoC™ 5LP web page. Also, see the CY8C58LPxx family datasheet.

A.5 Serial Interconnection between PSoC™ 5LP and PSoC™ 6 MCU

In addition of use as an onboard programmer, the PSoC™ 5LP functions as an interface for the USB-UART, USB-I^2C, and USB-SPI bridges, as shown in Figure 27. The USB-Serial pins of the PSoC™ 5LP are hard-wired to the I^2C/UART/SPI pins of the PSoC™ 6 MCU. These pins are also available on the Arduino-compatible I/O headers; therefore, the PSoC™ 5LP can be used to control Arduino shields with an I^2C/UART/SPI interface.


SWD Interface
TDO_SWO
TDI
P5LP12_2
P6_TMS_SWDIO
P5LP12_3
P6_TCLK_SWCLK
P5LP12_4
SWD_RST_L
0ohm
R62
P6_4
P6_5
P6_6
P6_7
XRES_L
I2C Interface
P6_VDD
R41
4.7K
R42
4.7K
P5LP12_1
I2C_SDA
P5LP12_0
I2C_SCL
R114
0ohm
R109
0ohm
P6_1
P6_0
UART Interface
P5LP12_6
UART_TX
R120
0ohm
P5_0
P5LP12_7
UART_RX
R119
0ohm
P5_1

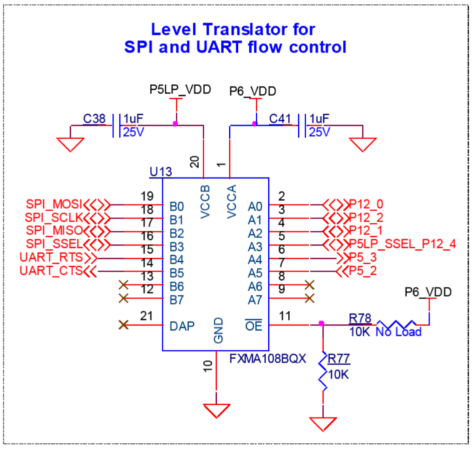


Figure 27 **Schematics of programming and serial interface connections**

A.6 EZ-PD™ CCG3 power delivery system

Infineon's EZ-PD™ CCG3 provides a complete solution ideal for power adapters, power banks, Type-C dongles, monitors, docks and notebooks. See EZ-PD™ CCG3 Type-C power delivery for more details of the power delivery system implementation in the Pioneer board.

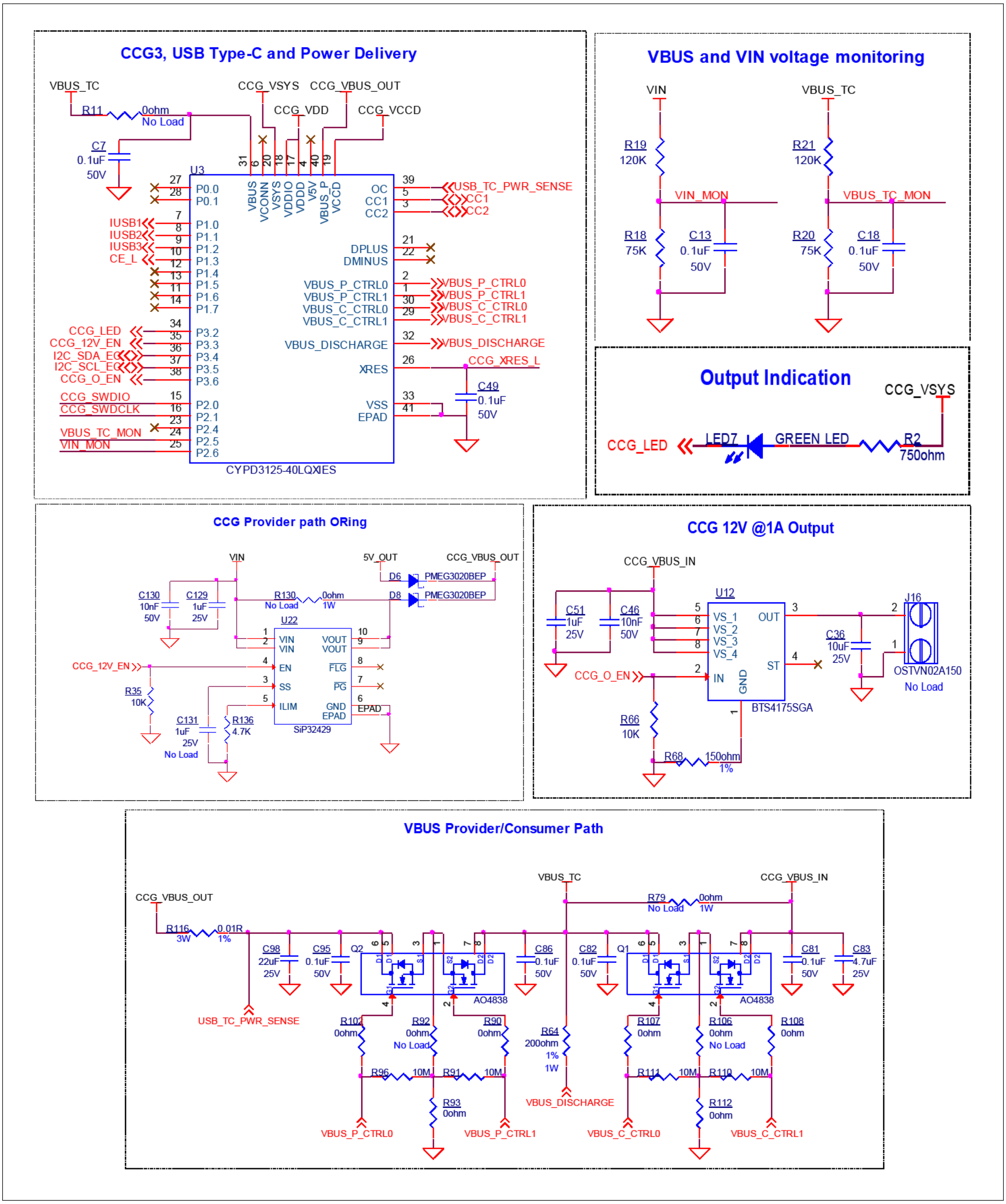


Figure 28 Schematics of EZ-PD™ CCG3 power delivery system

A.7 Power Supply System

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

- 5 V, 9 V, or 12 V from the onboard USB Type-C connector
- 5 V to 12 V power from an Arduino shield or from external power supply through VIN header **J9** or **J1**
- 3.7 V from a rechargeable Li-Po battery connected to **J15**
- 5 V from an external programmer/debugger connected to **J11** and **J12**

The power supply system is designed to support 1.8 V to 3.3 V operation of the PSoC™ 6 MCU. In addition, an intermediate voltage of 5 V is required for the operation of the power delivery circuitry and KitProg3. Therefore, three regulators are used to achieve 1.8 to 3.3 V and 5 V outputs - a buck boost regulator **(U21)** that generates a fixed 5 V from an input of 5 to 12 V, and a Main regulator **(U6)** that generates a variable 1.8 V to 3.3 V and a fixed 3.3 V from the output of **U21**. Figure 29 shows the schematics of the voltage regulator and power selection circuits. In addition to this, the battery charger **U14** also functions as a boost regulator. **U14** boosts the battery voltage to provide a 5 V to the Main regulator **U6**. This feature is enabled only when the VIN and the USB supply are unavailable.

The voltage selection is made through switch **SW5**. In addition, an onboard 330 mF super-capacitor **(C52)** can be used to power the backup domain power (Vbackup) of PSoC™ 6 MCU. Switch **SW7** selects the Vbackup supply connection of PSoC™ 6 MCU between V_{DDD} and the super-capacitor. When V_{DDD} is selected, the variable regulator ON/OFF terminal is controlled by KitProg3. When the super-capacitor is selected, the regulator ON/OFF terminal is controlled by PSoC™ 6 MCU. To ensure proper operation of PSoC™ 6 MCU, the super-capacitor, when selected, must be charged internally by PSoC™ 6 MCU before turning OFF the regulator. For more details of the PSoC™ 6 MCU backup system and power supply, see the PSoC™ 6 Bluetooth® LE Technical Reference Manual.

Table 6 details the different powering scenarios for Pioneer board.

Table 6 Power supply scenarios

Power inputs					Board condition		
USB	**VIN**	**ETM header (VTARG_IN)**	**Battery connected**	**JTAG/SWD header (VTARG_REF)**	**Main regulator powered by**	**PSoC™ powered by**	**Battery charging**
Non Type-C power adapter (Legacy USB), 5 V	<5	N/A	Yes	N/A	Type-C	Main regulator	100 mA
	>5	N/A	N/A	N/A	VIN	Main regulator	No
Type-C, PD power adapter	< PD power adapter	N/A	Yes	N/A	Type-C	Main regulator	1.5 A
	> PD power adapter, <12 V	N/A	N/A	N/A	VIN	Main regulator	No
0 V	5 V–12 V	N/A	N/A	N/A	VIN	Main regulator	No

(table continues...)

Table 6 (continued) Power supply scenarios

Power inputs					Board condition		
USB	**VIN**	**ETM header (VTARG_IN)**	**Battery connected**	**JTAG/SWD header (VTARG_REF)**	**Main regulator powered by**	**PSoC™ powered by**	**Battery charging**
0 V	0 V	5 V	N/A	N/A	ETM (VTARG_IN)	Main regulator	No
0 V	0 V	0 V	3.2 V–4.2 V	N/A	Battery	Main regulator	No
0 V	0 V	0 V	0 V	1.8 V–3.3 V	N/A	JTAG/SWD (VTARG_REF)	No

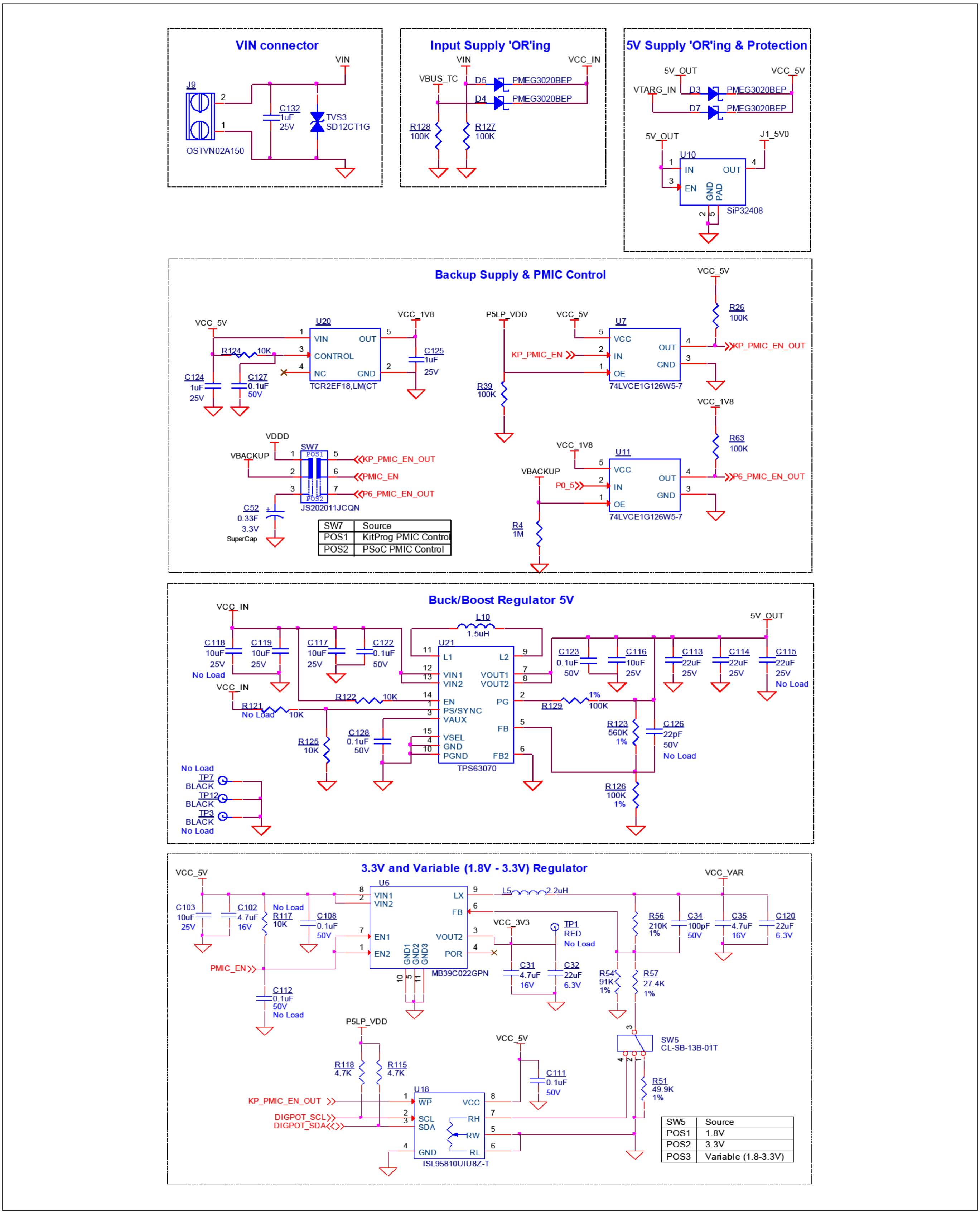


Figure 29 Schematics of power supply system

A.8 Expansion connectors

A.8.1 Arduino-compatible headers (J1, J2, J3, J4, and J5)

The board has five Arduino-compatible headers: **J1**, **J2**, **J3**, **J4**, and **J5** (J5 is not populated by default). You can connect 3.3 V Arduino-compatible shields to develop applications based on the shield's hardware. Note that 5 V shields are not supported and connecting a 5 V shield may permanently damage the board. See PSoC™ 6 Bluetooth® LE Pioneer Board Rework for details on PSoC™ 6 MCU pin mapping to these headers.

A.8.2 PSoC™ 6 MCU I/O headers (J18, J19, and J20)

These headers provide connectivity to PSoC™ 6 MCU GPIOs that are not connected to the Arduino compatible headers. Majority of these pins are multiplexed with onboard peripherals and are not connected to PSoC™ 6 MCU by default. For the detailed information on how to rework the kit to access these pins, see PSoC™ 6 Bluetooth® LE Pioneer Board Rework.

A.8.3 PSoC™ 5LP GPIO header (J6)

J6 is a 8x2 header provided on the board to bring out several pins of the PSoC™ 5LP to support advanced features such as a low-speed oscilloscope and a low-speed digital logic analyzer. This header also contains the USB-UART, USB-I^{2}C, and USB-SPI bridge pins that can be used when these pins are not accessible on the Arduino headers because a shield is connected. The additional PSoC™ 5LP pins are connected directly to the internal programmable analog logic of PSoC™ 5LP. This header also has GPIOs for custom application usage. J6 is not populated by default. Note that the SPI, RTS, and CTS lines on these headers are directly from PSoC™ 5LP (before level translator).

A.8.4 KitProg3 custom application header (J7)

A 5x2 header is provided on the board to bring out more GPIOs of PSoC™ 5LP for custom application usage. This header also brings out the PSoC™ 5LP programming pins and can be programmed using MiniProg4 and 5-pin programming connector. J7 is not populated by default.

A.9 CAPSENSE™ circuit

A CAPSENSE™ slider, two buttons, all of which support both self-capacitance (CSD) and mutual-capacitance (CSX) sensing, and a CSD proximity sensor (header) are connected to PSoC™ 6 MCU as Figure 30 shows. Four external capacitors - CMOD and CSH for CSD, CINTA and CINTB for CSX are present on the Pioneer board. Note that CSH is not loaded by default. For details on using CAPSENSE™ including design guidelines, see the Getting started with CAPSENSE™ design guide.

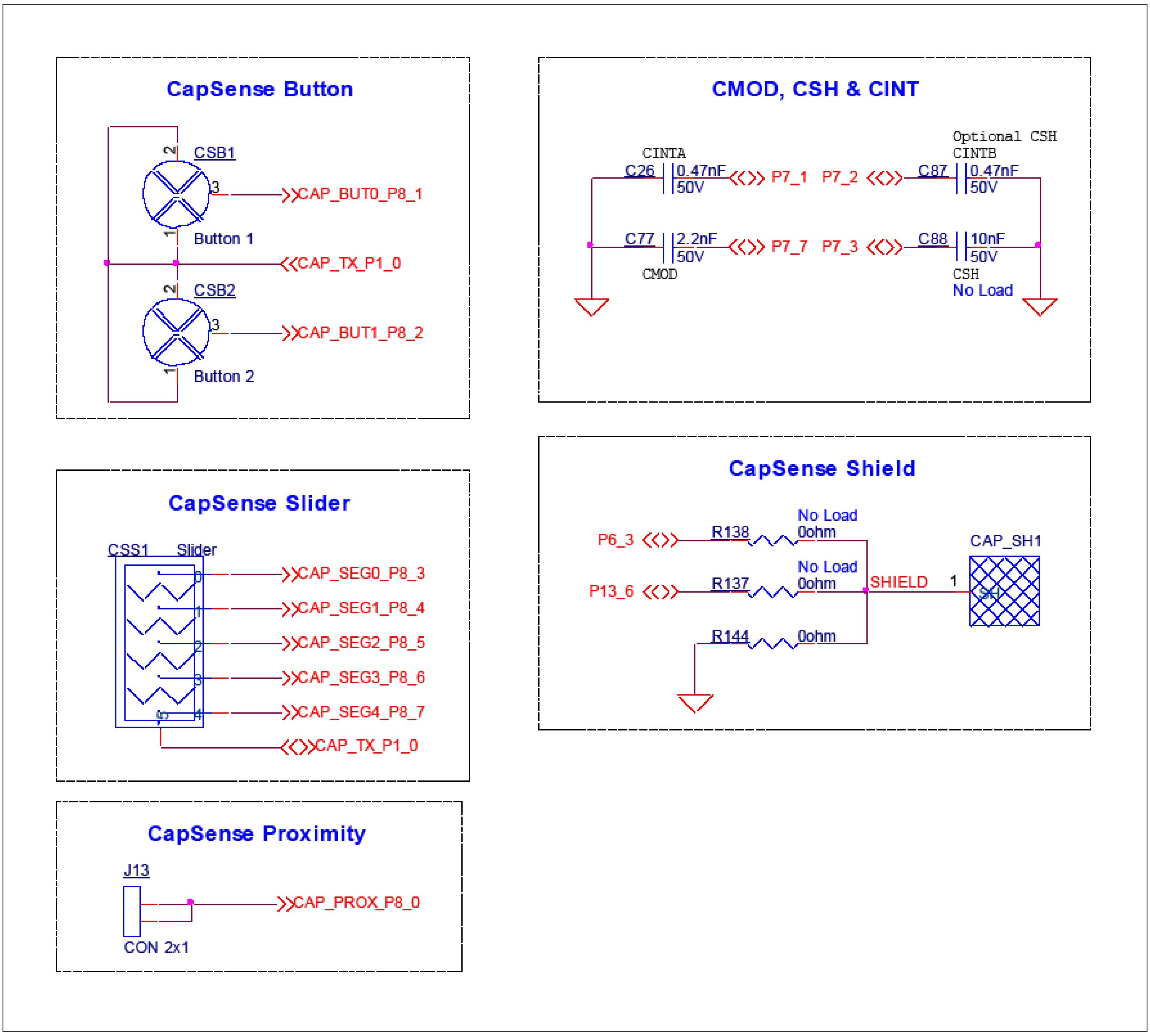


Figure 30 **Schematics of CAPSENSE™ circuit**

Simultaneous GPIO switching with unrestricted drive strengths and frequency can affect CAPSENSE™ and ADC performance. For more details, see the Errata section of the corresponding device datasheet.

A.10 LEDs

LED1, **LED2**, and **LED3** (Red, Amber, and Green respectively) indicate the status of the KitProg3 (See the KitProg3 User guide for details). **LED4** (amber LED) indicates the status of power supplied to PSoC™ 6 MCU. **LED7** (Green) indicates the status of power delivery output on **J16**. **LED6** (Red) indicates the battery charger status.

The Pioneer board also has two user controllable LEDs **(LED8** and **LED9)** and an RGB LED **(LED5)** connected to PSoC™ 6 MCU pins for user applications.

A.11 Push buttons

The PSoC™ 6 Bluetooth® LE Pioneer Kit has a reset button and three user buttons. The reset button **(SW1)** is connected to the XRES pin of the PSoC™ 6 MCU, and is used to reset the device. One user button **(SW2)** is connected to pin P0[4] of the PSoC™ 6 MCU. The remaining two buttons - **SW3** and **SW4** are connected to the PSoC™ 5LP device for programming mode and custom app selection respectively (Refer to the KitProg3 User guide for details). All the buttons connect to ground on activation (active low) by default. User button **(SW2)** can be changed to active high mode by changing the zero resistors shown below, to evaluate the PMIC control feature of PSoC™ 6 MCU. Refer to the Appendix A.15 section for additional information on rework, on Rev. 08 or earlier versions.

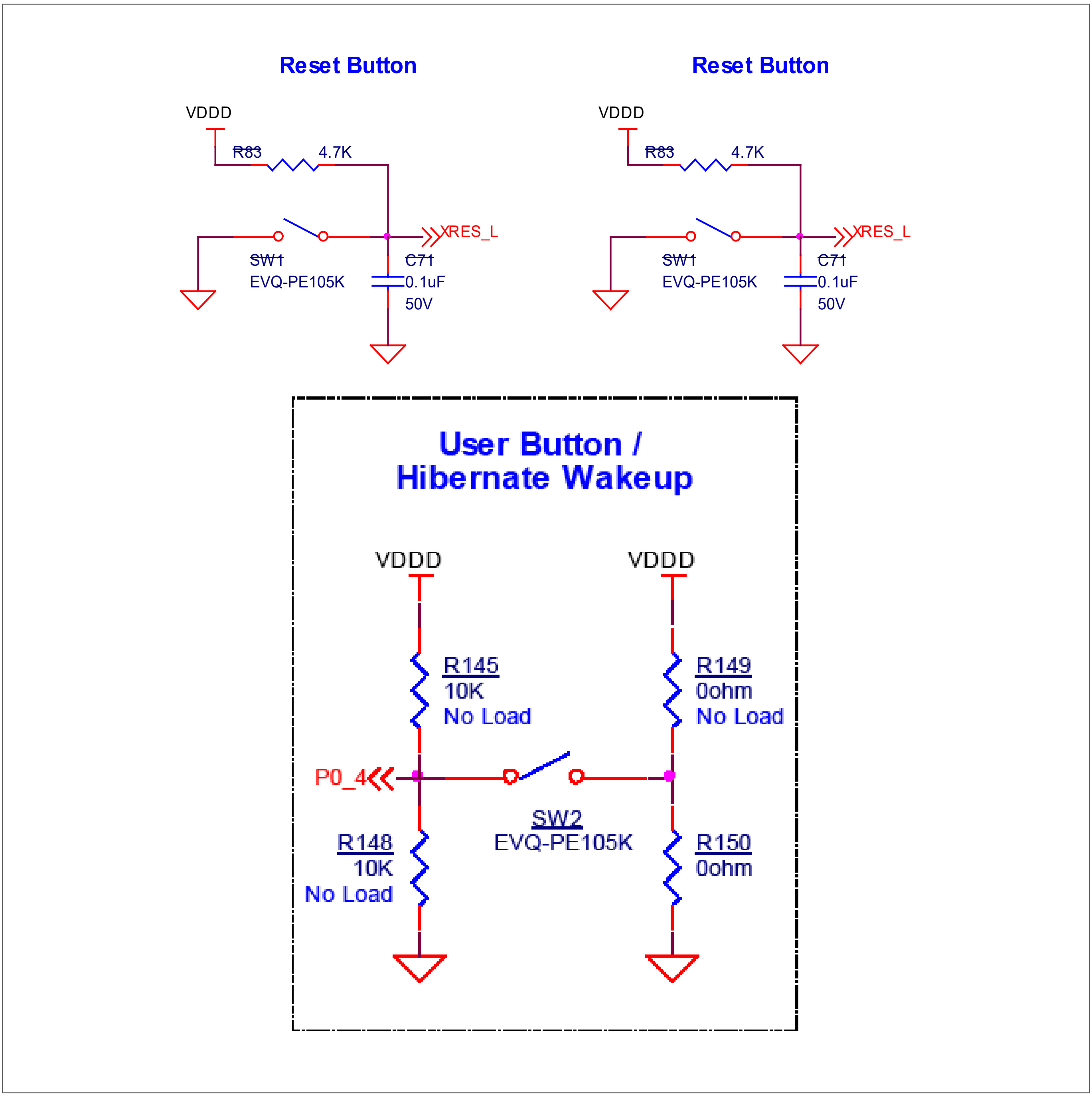


Figure 31 **Schematics of Push buttons**

A.12 Infineon NOR Flash

The Pioneer board has an Infineon NOR flash memory (S25FL512SAGMFI011) of 512 Mb capacity. The NOR Flash is connected to the serial memory interface (SMIF) of PSoC™ 6 MCU. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption.

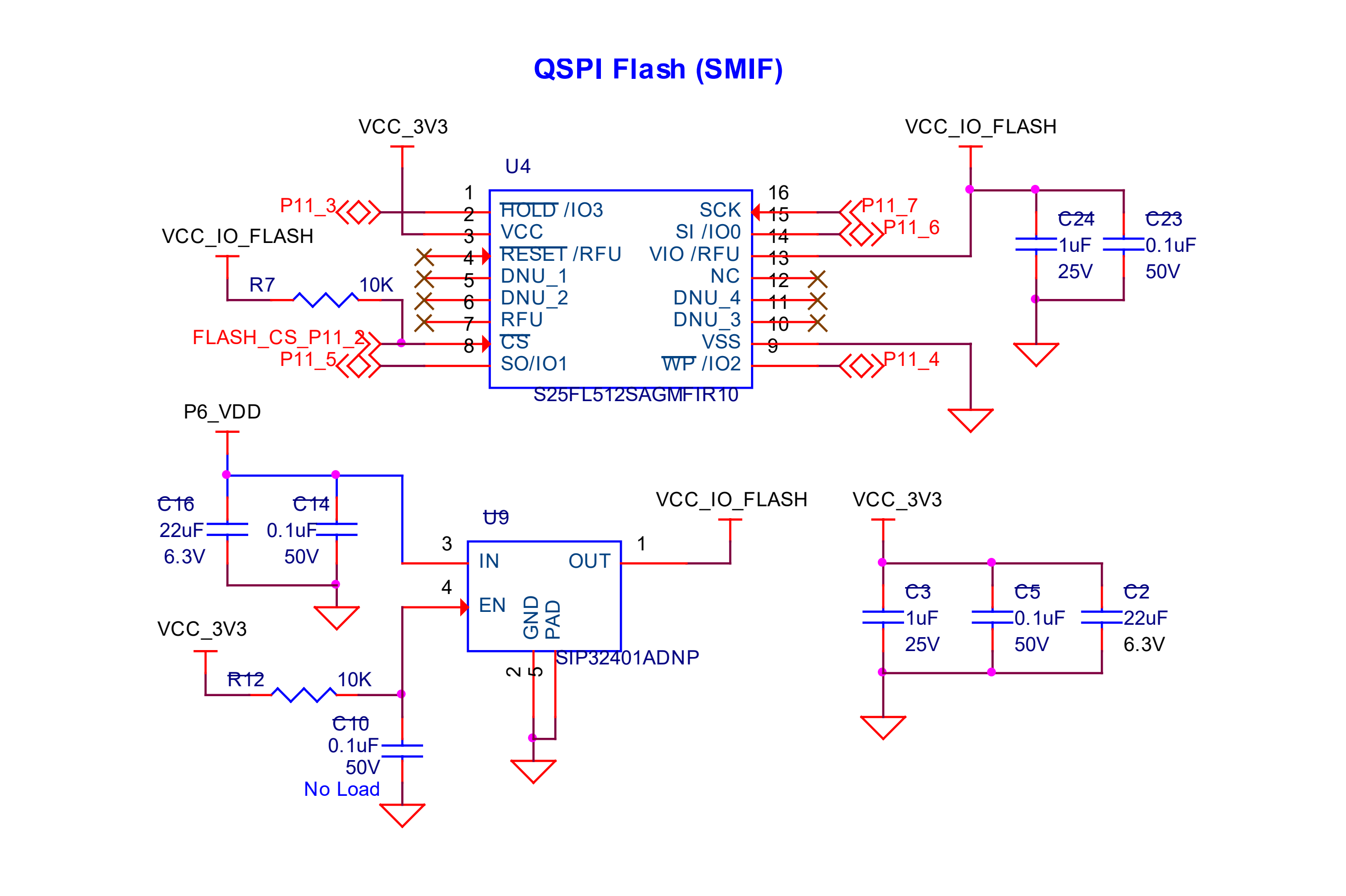


Figure 32 **Schematics of NOR Flash**

A.13 Infineon Ferroelectric RAM (F-RAM)

The Pioneer board contains provision for the FM24V10-GTR F-RAM device (see Figure 33), which can be accessed through SPI interface. The F-RAM is 1 Mbit (128 KB × 8) with SPI speed up to 40 MHz.

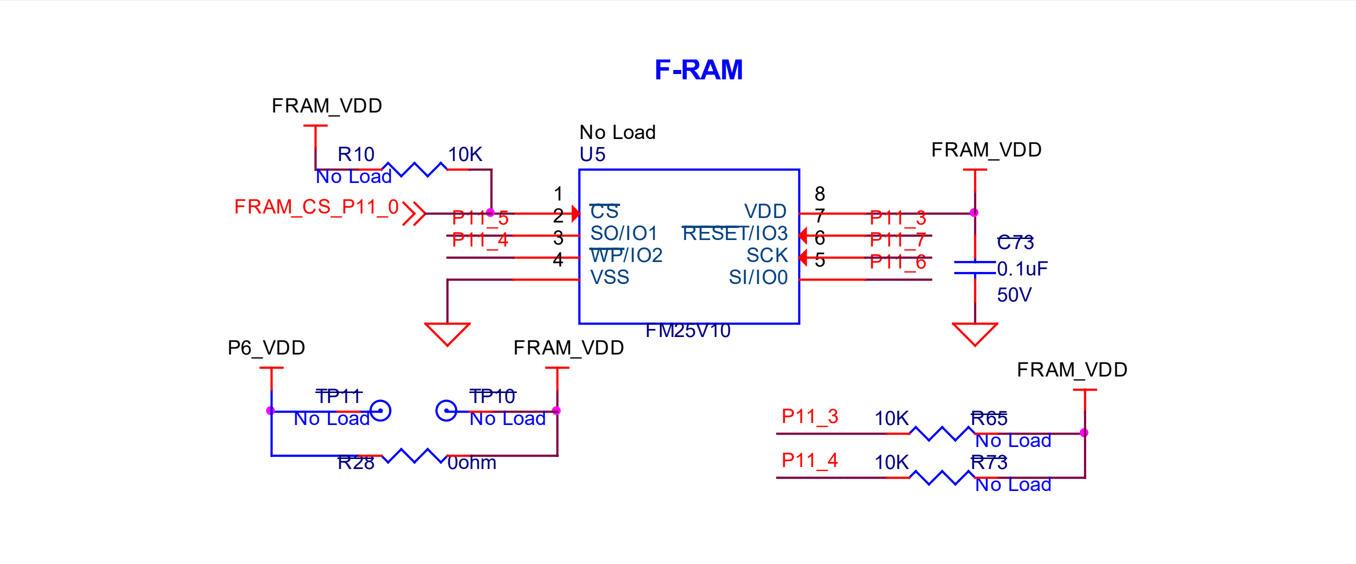


Figure 33 **Schematics of F-RAM**

A.14 Crystal oscillators

The Pioneer board includes a 32-MHz ECO and a 32-KHz WCO for PSoC™ 6 Bluetooth® LE device.

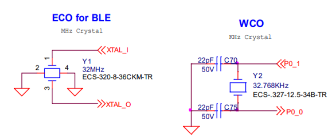


Figure 34 **Schematics of ECO and WCO**

A.15 Frequently Asked Questions

1. I don't have a Type-C connector on my PC. Can I still connect and use this kit?

Yes. To evaluate PSoC™ 6 MCU features, any PC with USB2.0 connectivity is sufficient. Type-C power adapter is required only to evaluate the CCG3 section of the Kit.

2. How does CY8CKIT-062-BLE handle voltage connection when multiple power sources are plugged in?

There are five different options to power the baseboard; Type-C USB connector **(J10)**, External DC supply via VIN connector **(J9/J1)**, Debug and trace header (J12, VTARG_IN), Program and debug header **(J11)**, and LiPo battery header **(J15)**. Both Type-C and VIN take priority over other supply options. These inputs are ORed using diode and the higher voltage among the two take precedence. Output of ORing diode is given to a buck-boost regulator **(U16)** that generate a constant 5.2 V. This output is ORed with ETM supply **(J12)** which is typically 5 V. For most of the practical uses, output from the 5.2 V regulator takes priority and the same is given as an input to Infineon buck regulator **(U6)**. LiPo battery voltage is used when all the above sources are absent. Output of Infineon buck regulator **(U6)** is ORed with supply voltage from the program and debug header **(J11)**, and higher voltage takes precedence. See Table 6 for more details on voltage input and output scenarios.

3. How can I access Smart I/O and other GPIOs connected to onboard peripherals?

Some of the Smart I/O (Port 8 and Port 9.3) and GPIO connected to onboard peripherals are multiplexed with PSoC™ 6 MCU I/O headers **(J18**, **J19**, and **J20)**. By default, some of these I/Os are connected to onboard peripherals using series resistors. These resistors can be changed to route these I/Os to headers. See Table 3 for details on list of resistors that needs to be changed.

4. Why does the Red LED of RGB LED **(LED5)** light up when switch **SW7** is set to SuperCap position?

This behavior is observed if SuperCap is charged below 1.5 V. The I/Os referring to this domain will leak current, in this case P0[3]. V_{BACKUP} feature needs to be enabled in silicon before switching **SW7** to SuperCap position. See the TRM/datasheet on options to enable SuperCap charging.

5. What can I use the U.FL connector for and what is the typical mating cycle for these connectors?

U.FL can be used for conductive measurements and to connect external antenna. U.FL connectors are not designed for reconnection. They are rated only for approximately 30 mating cycles.

6. What are the three selection switches on baseboard for?

Table 2 gives details on all three selection switches. Additionally, each code example documentation explains the selection switch setting required for each code example.

7. What is the Jumper on board for?

The jumper **J8** can be used to measure current of PSoC™ 6 MCU device without the need to desolder any component from the board. An ammeter can be connected across this jumper to measure the current consumed by the PSoC™ 6 MCU device. Remove the Jumper on **J8**, connect an ammeter (+ve terminal of ammeter to Pin 2), and power the kit though USB connector **J10**.

***Note**:* *Remove resistor (R4) which causes a leakage of 4 µA on V_{BACKUP} domain that is connected to VTARG.*

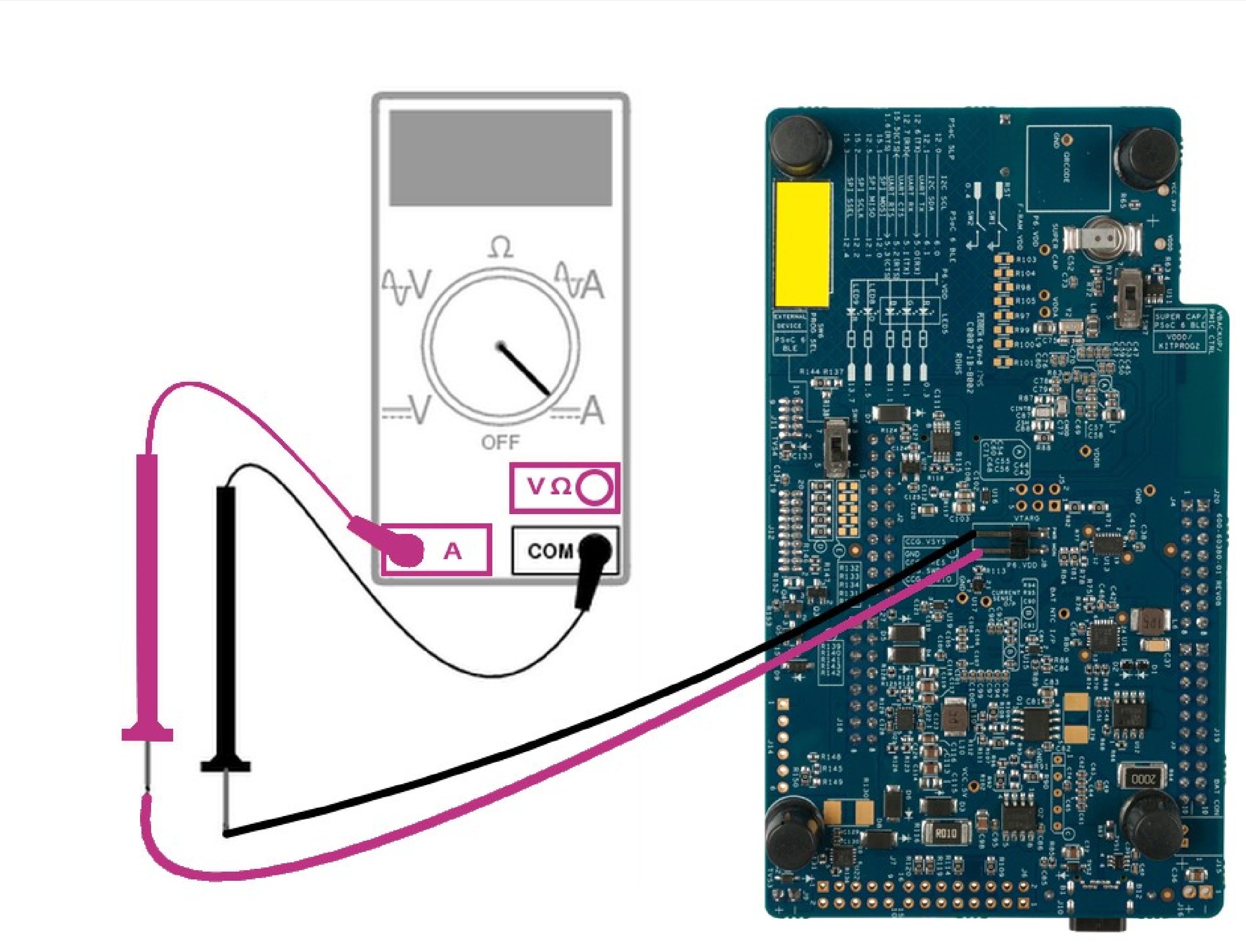


Figure 35 **Jumper J8 on board**

8. What are the input voltage tolerances? Are there any overvoltage protection on this kit?

Input voltage level are as follows:

Table 7 **Input voltage levels**

Supply	**Typical i/p voltage**	**Absolute max (overvoltage protection)**
USB Type-C connector (**J10**)	4.5 V to 12 V	15 V
VIN connector (**J9/J1**)	5 V to 12 V	15 V
Debug and trace header (**J12**)	5 V	5.5 V
Program and Debug header (**J11**)	1.8 V to 3.3 V	3.6 V
Li-Po battery connected (**J15**)	3.2 V to 4.2 V	5 V

9. Why is the voltage of the kit restricted to 3.3 V? Can't it drive external 5 V interfaces?

PSoC™ 6 is not meant to be operated at voltages more than 3.6 V. Powering PSoC™ 6 to more than 4 V will permanently damage the chip. You cannot drive I/O system with > 3.6 V supply voltages.

10. By mistake, I powered my Arduino board while powering PSoC™ 6 MCU. Is my PSoC™ 6 chip alive?

Yes. The 3.3 V and 5 V on Arduino power header are not input pins and have 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 15 V. P6.V_{DD} pin is not protected and care should be taken not to supply voltage to this pin.

11. What type of battery can I use for this kit?

Recommended lithium-ion polymer rating is 3.7 V @850 mAH or higher. SparkFun Electronics PRT-13854 or equivalent batteries can be used. The LiPo battery charger can charge at 100 mA or 1.5 mA based on whether the USB connection is a legacy device or PD capable.

12. By mistake, I connected the battery with opposite polarity. Did I fry the system?

There are relevant protection circuits to protect the system from permanent damage. Prolonged connection may lead to damage.

13. Can I charge any kind of Type-C device using this kit?

Kit is programmed to advertise the VIN voltage with 1 A current rating. 9 V and 12 V devices are the recommended options. VIN needs to be 9 V and 12 V respectively for this to work.

14. How can I evaluate the USB Type-C provider and consumer features to get started with?

You can use any kind of Type-C laptop, mobile phone, or PD adapters based on the feature that you are trying to evaluate. To use as consumer, note that devices like laptop may be able to provide only 5 V out and may not support 9 V/12 V without a docking station. To use as provider, any 5 V/9 V/12 V device that has a current requirement of less that 1 A may be used. Additionally, Infineon has its own USB Type-C evaluation Kit which can be used to evaluate the provider and consumer features and many more. Visit Infineon Type-C and Power-delivery controllers for details on these kits.

15. Why is the screen of EPD permanently ON?

Electronic paper and e-paper are display devices that mimic the appearance of ordinary ink on paper. Unlike conventional backlit flat panel displays that emit light, electronic paper displays reflect light like paper. This may make it more comfortable to read and provide a wider viewing angle than most light-emitting displays. The contrast ratio in electronic displays available as of 2008 approaches newspaper, and newly developed displays are slightly better. An ideal e-paper display can be read in direct sunlight without the image fading. Many electronic paper technologies hold static text and images indefinitely without electricity. To know the details, see https://en.wikipedia.org/wiki/Electronic_paper.

16. Why does the screen of EPD glitch from black to white during every transition?

The screen of EPD shield refreshes at every transition. It first clears all the cells of the display by spreading it with white pixels and then posts the images.

17. When I touch the resistors near the CAPSENSE™ slider/button, the E-INK display shield and LED are triggered. Why does this happen?

Those are the series limiting resistors which interface CAPSENSE™ lines with ICs I/O lines. As you are touching CAPSENSE™ lines, it triggers E-INK display shield which responds to CAPSENSE™.

18. The EPD shield shows white and black lines after I switched OFF the kit. Is the EPD shield damaged?

EPD retains the image when you may have powered OFF the kit. In case the kit is powered OFF during transition between images, lines might appear.

19. I am unable to program the target device?

Check **SW6** to ensure it is in **PSoC™ 6 MCU** position.

Check **SW7** to ensure it is in **V_{DDD}/KitProg3** position.

Make sure that no external devices are connected to **J11**.

Update your KitProg3 Version in programmer to 1.04 or later using the steps mentioned in KitProg3 User guide.

Ensure that device used in PSoC™ Creator is CY8C6347BZI-BLD53

20. Does the kit get powered when I power the kit from another Infineon Kit through the J1 header?

Yes, VIN pin on J1 header is the supply input/output pin and can take up to 12 V.

21. What additional overlays can be used with the CAPSENSE™?

Any kind of overlays (up to 5 mm thickness) like wood, acrylic, and glass can be used with this CAPSENSE™. Note that additional tuning may be required when the overlay is changed.

22. What is PMOD?

PMOD interface or Peripheral Module interface is an open standard defined by Digilent Inc in the Digilent Pmod Interface Specification for peripherals used with FPGAs or microcontrollers. Several types of modules are available from simple push buttons to more complex modules with network interfaces, analog to digital converters or LCD displays. PMODs are available from multiple vendors such as Diligent, Maxim Integrated, Analog Devices and a variety of hobby sites. This Kit supports only 1 x 6 pin PMOD modules.

23. With what type of shield from Infineon can I use this baseboard?

Any Arduino Uno shield which supports 3.3 V operation is compatible with this Pioneer board. Following Infineon shields are pin compatible with this pioneer board:

CY3280-MBR3

CY8CKIT-022

CY8CKIT-024

CY8CKIT-026

CY8CKIT-040

CY8CKIT-046

CY8CKIT-048

24. Can I use this Kit as a programmer to program external PSoC™ devices?

Yes, the onboard KitProg3 can program any PSoC™ 4/5/6 devices connected to **J11** header. Switch **SW6** should be switched to “External Device” position to program devices connected via **J11** header.

25. Which third-party debuggers 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 PSoC™ Creator ‘Help’ menu.

26. Can I power PSoC™ 6 MCU using only external programmer at 1.8 V through the J11 header?

Yes, but there may be chance of failure as there is a voltage drop across the Overvoltage protection circuit. **R152** can be populated to bypass the protection circuit but is not recommended due to the increased risk of damaging PSoC™ 6 MCU.

27. Can I power and program the PSoC™ 6 MCU using only MiniProg4 at 3.3 V?

The MiniProg4 Rev *B has an error in target voltage which exceeds the tolerance of the overvoltage protection circuit which can cause failure. However, this will work in MiniProg4 Rev *C which has lesser error in target voltage. **R152** can be populated to bypass the protection circuit but is not recommended due to the increased risk of damaging PSoC™ 6 MCU.

28. Why the on-board RGB LED **(LED5)** does not work when I select on a supply voltage of 1.8 V using **SW5**?

The on board RGB LED requires a supply voltage higher than 2.7 V to function correctly. Ensure that **SW5** is set to the 3.3 V or 1.8 V–3.3 V VARIABLE with PSoC™ Programmer selecting a voltage of 2.7 V or higher in the latter case. Using this kit with voltage lower than 2.7 V will affect the RGB LED operation. Alternatively, you can use discrete LEDs (**LED8** and **LED9**) if the application permits.

29. How to use SW2 for PMIC wake up?

SW2 is connected to the PMIC_Wakeup_In pin (P0.4) of PSoC™ 6 MCU. A logic high input at the PMIC_Wakeup_In pin can wake up the system and enable the PMIC. See the Backup chapter in PSoC™ 6 MCU architecture technical reference manual for more details of this feature.

SW2 should be externally pulled down to ground to use PMIC control. Moreover, when the switch is pressed, the active high logic should push P0.4 to the V_{BACKUP} supply. However, the kit is configured by default to use active low logic. In addition, in Rev08 and earlier versions of the kit, the active high logic for SW2 pushes P0.4 to V_{DDD}. Therefore, the following rework on the kit are required to use the PMIC control feature:

- Remove R150 and connect the common pad between R150 and R149 to the positive terminal of super-capacitor (C52)
- Load external pull-down resistor R148

See Figure 36 for details.

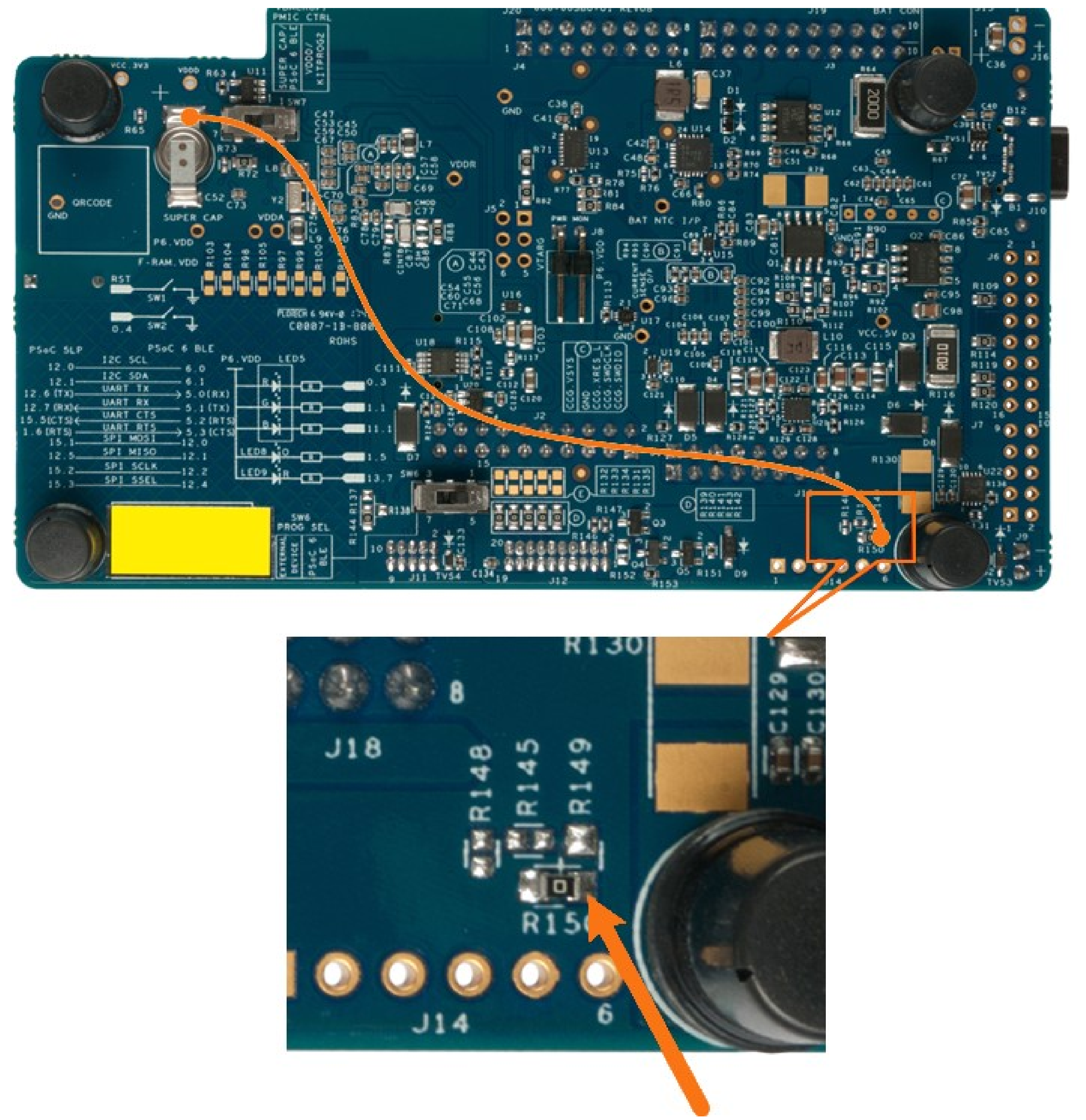


Figure 36 **Rework required to use PMIC Control**

Acronyms

Acronym	Definition
ADC	Analog-to-Digital Converter
BLE	Bluetooth® Low Energy
BOM	Bill of Materials
CINT	Integration Capacitor
CMOD	Modulator Capacitor
CPU	Central Processing Unit
CSD	CAPSENSE™ Sigma Delta
CTANK	Shield Tank Capacitor
DC	Direct Current
Del-Sig	Delta-Sigma
ECO	External Crystal Oscillator
EPD	Electronic Paper Display
ESD	Electrostatic Discharge
F-RAM	Ferroelectric Random Access Memory
FPC	Flexible Printed Circuit
GPIO	General-Purpose Input/Output
HID	Human Interface Device
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
ICSP	In-Circuit Serial Programming
IDAC	Current Digital-to-Analog Converter
IDE	Integrated Development Environment
LED	Light-emitting Diode
PC	Personal Computer
PD	Power Delivery
PDM	Pulse Density Modulation
PTC	Positive Temperature Coefficient
PSoC™	Programmable System-on-Chip
PWM	Pulse Width Modulation
RGB	Red Green Blue
SAR	Successive Approximation Register
SPI	Serial Peripheral Interface
SRAM	Serial Random Access Memory
SWD	Serial Wire Debug

Acronym	Definition
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WCO	Watch Crystal Oscillator

Revision history

Document version	Date of release	Description of changes
*B	2017-09-25	Initial version for public release
*C	2018-02-02	Updated Introduction Updated Kit contents Updated Figure 1 Updated Board details Updated Figure 4 Updated Figure 5 Updated Table 3 Updated Technical Support Replaced “Ext. 2” with “Ext. 3” Updated Software installation Updated Install software Updated description Updated Kit operation Updated Theory of operation Updated Figure 13 Updated CY8CKIT-028-EPD E-INK display shield Updated Figure 15 Updated KitProg2 Updated Programming and Debugging using PSoC Creator Updated Figure 3-6 Updated EZPD CCG3 Type-C power delivery Updated Table 5 Updated Code examples Updated Using the Kit code examples Updated Figure 4-1 Updated Code examples Updated Table 4-1 Updated Appendix Updated Hardware functional description Updated Push buttons Updated Figure 31 Updated Cypress NOR Flash Added Figure 32 Added Cypress Ferroelectric RAM (F-RAM) Added Crystal oscillators Updated Frequently Asked Questions Updated description Added Figure 35

Document version	Date of release	Description of changes
*D	2018-04-27	Updated Safety and Regulatory Compliance Information Replaced Safety Information with Safety and Regulatory Compliance Information in chapter heading Updated description Updated General Safety Instructions Updated Handling Boards Updated description Added Regulatory Compliance Information Updated Introduction Updated Kit contents Updated Figure 1 Updated Board details Updated Table 3 Updated Appendix Updated Hardware functional description Updated Push buttons Updated description Updated Frequently Asked Questions Updated description Added Figure 36 Updated to new template
*E	2018-05-02	Updated Appendix Updated PSoC 6 BLE Pioneer Board Rework Replaced 'reworks' with 'rework' in heading Updated Frequently Asked Questions Replaced 're-works' with 'rework' in all instances
*F	2018-07-02	Updated details (Document Properties) in **File > File Info**
*G	2018-07-17	Updated Code examples Updated Using the Kit code examples Updated Figure 4-1 Updated Code examples Updated Table 4-1
*H	2019-04-09	Updated Kit operation Updated Theory of operation Updated description Updated Appendix Updated Hardware functional description Updated CAPSENSE™ circuit Updated description

Document version	Date of release	Description of changes
*I	2019-08-26	Updated Document Title to read as CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit Guide Updated Appendix Updated Frequently Asked Questions Updated description Updated to new template
*J	2020-01-21	Updated Appendix Updated Hardware functional description Updated Crystal oscillators Updated Figure 34
*K	2020-02-05	Updated Kit operation Updated Theory of operation Updated description
*L	2024-09-30	Template update Updated Safety precautions Replaced PSoC™ creator software with ModusToolbox™ based procedure Updated PSoC to PSoC™ Updated BLE to Bluetooth® LE Removed the document conventions section Updated the title from CY8CKIT-062-BLE PSoC 6 Bluetooth LE Pioneer Kit Guide to PSoC™ 6 CY8CKIT-062-BLE with AIROC™ Bluetooth® LE Pioneer Kit Updated all diagrams to Infineon branding Updated CapSense to CAPSENSE™ Updated KitProg2 to KitProg3 Updated MiniProg3 to MiniProg4 Added About this document section Updated the term Cypress to Infineon

Trademarks

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc., and any use of such marks by Infineon is under license.

Trademarks

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

Edition 2024-09-30

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-eyp1725276793462