Please note that Cypress is an Infineon Technologies Company.

The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content

The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers

Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CY8CKIT-064B0S2-4343W

PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Guide

Document Number: 002-29286 Rev. *E

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com
www.infineon.com

Copyrights

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, including its affiliates, and its directors, officers, employees, agents, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, and combinations thereof, WICED, ModusToolbox, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress or a subsidiary of Cypress in the United States or in other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents

Safety and Regulatory Compliance Information

Regulatory Compliance Information

Contains Transmitter Module FCC ID: VPYLB1DX and IC: 772C-LB1DX

This kit is intended to use for ENGINEERING DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY and is not considered by Cypress Semiconductor to be a finished end product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC or ICES-003 rules, which are designed to provide reasonable protection against radio frequency interference. Operation of the equipment may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

Due to change in the antenna pattern/type and gain used in CY8CKIT-064B0S2-4343W PSoC® 64 “Secure Boot” Wi-Fi BT Pioneer Kit, class II permissive changes are required to recertify this kit. The radiated emission tests must be performed again to obtain a new FCC ID for this host kit. Most conducted RF test results may still be reused. Customer also needs to take their product through other FCC/ISED testing such as unintentional radiators (FCC sub part 15B) and any other required regional product certifications including but not limited to EU directives. Refer FCC Regulatory Certification Guide by Murata on information on pre-certified and reference certified module concepts and information on what additional test are required for FCC certification. Customer should consult a Telecommunication Certification Body (TCB) lab for guidance on other requirements for the device certification.

For more details on Murata Type 1DX module refer https://wireless.murata.com/type-1dx.html.

PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Boards contain electrostatic discharge (ESD) sensitive devices. Electrostatic charges readily accumulate on the human body and any equipment, which can cause a discharge without detection. Permanent damage may occur on devices subjected to high-energy discharges. Proper ESD precautions are recommended to avoid performance degradation or loss of functionality. Store unused PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Boards in the protective shipping package.

End-of-Life/Product Recycling

The end-of-life cycle for this kit is five years from the date of manufacture mentioned on the back of the box. Contact your nearest recycler to discard the kit.

General Safety Instructions

ESD Protection

ESD can damage boards and associated components. Cypress recommends that you perform procedures only at an ESD workstation. If an ESD workstation is unavailable, use appropriate ESD protection by wearing an anti-static wrist strap attached to a grounded metal object.

Handling Boards

CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit is sensitive to ESD. Hold the board only by its edges. After removing the board from its box, place it on a grounded, static-free surface. Use a conductive foam pad, if available. Do not slide the board over any surface.

1. Introduction

Thank you for your interest in the CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit. The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit enables you to evaluate and develop your applications using the PSoC 64 Line of Secured MCUs (hereafter called PSoC 64) and CYW4343W WICED Wi-Fi/BT combo device.

PSoC 64 is Cypress’ latest, ultra-low-power PSoC specifically designed for wearables and IoT products. PSoC 64 chip 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, pre-configured with a root-of-trust and secure processing environment, up to 2 MB Flash and 1 MB SRAM, Secure Digital Host Controller (SDHC) supporting SD/SDIO/eMMC interfaces, 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 64 “Secure Boot” Wi-Fi BT Pioneer Board offers compatibility with Arduino™ shields. The board features a PSoC 64 chip, and a CYW4343W Wi-Fi/BT combo module. Cypress CYW4343W is a highly integrated single-chip solution that includes a 2.4 GHz WLAN IEEE 802.11 b/g/n MAC/ baseband/radio and Bluetooth 5.1 support. The WLAN section supports SDIO interface to the host MCU (PSoC 64), and the Bluetooth section supports high-speed 4-wire UART interface to the host MCU. In addition, the board features an onboard programmer/debugger (KitProg3), a 512-Mbit Quad SPI NOR flash, a 4-Mbit Quad SPI F-RAM, a micro-B connector for USB device interface, a 5-segment CapSense slider, two CapSense buttons, a microSD card holder, an RGB LED, two user LEDs, one potentiometer, and two push buttons. The board supports operating voltages of 1.8 V, 2.5 V and 3.3 V.

You can use ModusToolbox™ software to develop and debug your PSoC 64 projects. ModusToolbox software is a set of tools that enable you to integrate Cypress devices into your existing development methodology.

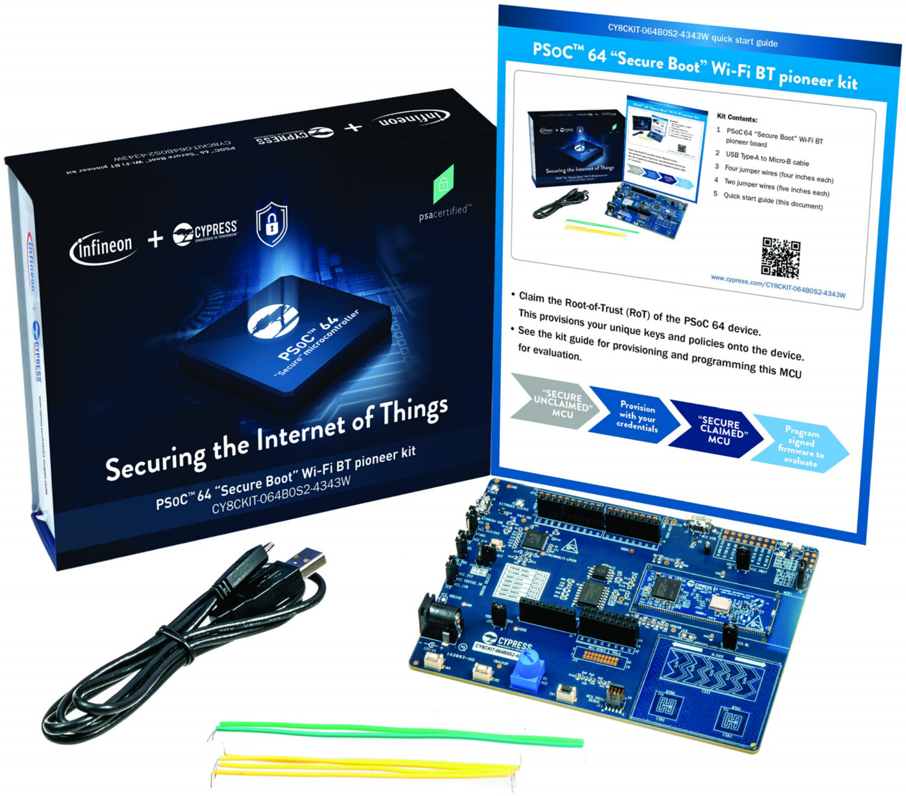
If you are new to PSoC 6 MCU and ModusToolbox software, refer to the application note AN228571 - Getting Started with PSoC 6 MCU on ModusToolbox to help you familiarize with the PSoC 6 MCU and help you create your own design using the ModusToolbox software.

1.1 Kit Contents

The CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit has the following contents, as shown in Figure 1-1.

- PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board
- USB Type-A to Micro-B cable
- Four jumper wires (4 inches each)
- Two jumper wires (5 inches each)
- Quick Start Guide

Figure 1-1. Kit Contents


infineon + CYPRESS
psacertified
PSoC™ 64 "Secure" microcontroller
Securing the Internet of Things
PSoC™ 64 "Secure Boot" Wi-Fi BT pioneer kit
CY8CKIT-064B0S2-4343W
CY8CKIT-064B0S2-4343W quick start guide
PSoC™ 64 "Secure Boot" Wi-Fi BT pioneer kit
Kit Contents:
1 PSoC 64 "Secure Boot" Wi-Fi BT pioneer board
2 USB Type-A to Micro-B cable
3 Four jumper wires (four inches each)
4 Two jumper wires (five inches each)
5 Quick start guide (this document)
www.cypress.com/CY8CKIT-064B0S2-4343W
• Claim the Root-of-Trust (RoT) of the PSoC 64 device. This provisions your unique keys and policies onto the device.
• See the kit guide for provisioning and programming this MCU for evaluation.
"SECURE UNCLAIMED" MCU
Provision with your credentials
"SECURE CLAIMED" MCU
Program signed firmware to evaluate

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

1.2 Getting Started

This guide will help you get acquainted with the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit:

- The Software Installation chapter on page 18 describes the installation of the kit software. This includes the ModusToolbox software which will be used to develop, program and debug applications on to the device.
- The Kit Operation chapter on page 20 describes the major features of the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit and functionalities such as programming, debugging, and the USB-UART and USB-I^{2}C bridges.
- The Running Code on PSoC 64 “Secure Boot” MCUs chapter on page 29 describes multiple PSoC 64 code examples that will help you understand how to create your own PSoC 64 projects.
- The Hardware chapter on page 40 provides a detailed hardware description, methods to use the onboard NOR flash, kit schematics, and the bill of materials (BOM).
- Application development using PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit is supported in various development ecosystems such as ModusToolbox software and Mbed OS. For the latest software support for this development kit including the different development ecosystems, refer to the kit webpage.
  - ModusToolbox software is a free development ecosystem that includes the ModusToolbox IDE. Using ModusToolbox IDE, you can enable and configure device resources, middleware libraries, and program and debug the device. You can download the software from the ModusToolbox home page. See the ModusToolbox User Guide for additional information.
  - Mbed OS: Visit Cypress’ Mbed OS page on instructions to develop applications on Cypress’ target board on the Mbed OS platform.

1.3 Code Examples

- There are wide range of code examples to evaluate the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board.These examples help you familiarize PSoC 64 “Secure” MCU and create your own design. These examples are available in various development ecosystems such as ModusToolbox IDE and Mbed OS. Visit Cypress’ code example page to access examples for the following development ecosystems:
  - ModusToolbox software based examples
  - Mbed OS based examples

1.4 Board Details

The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board has the following features:

- CY8CMOD-064B0S2-4343W board contains:
  - PSoC 64 device (CYB0644ABZI-S2D44)
  - Murata Type 1DX ultra-small 2.4-GHz WLAN and Bluetooth functionality module based on CYW4343W
- 512-Mbit external Quad SPI NOR Flash that provides a fast, expandable memory for data and code
- 4-Mbit Quad SPI ferroelectric random-access memory (F-RAM)
- KitProg3 onboard SWD programmer/debugger with USB-UART and USB-I2C bridge functionality
- CapSense touch sensing slider (5 elements) and two buttons. The slider and buttons are capable of using self-capacitance (CSD) or mutual-capacitance (CSX) sensing methods.
- A second Micro-B connector for USB device interface

- 1.8 V, 2.5 V and 3.3 V operation is supported
- Two user LEDs, an RGB LED, two user buttons, and a reset button
- A potentiometer
- One Mode selection button and one Status LED for KitProg3
- A microSD Card holder

Figure 1-2 shows the pinout of the Board.

Figure 1-2. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout

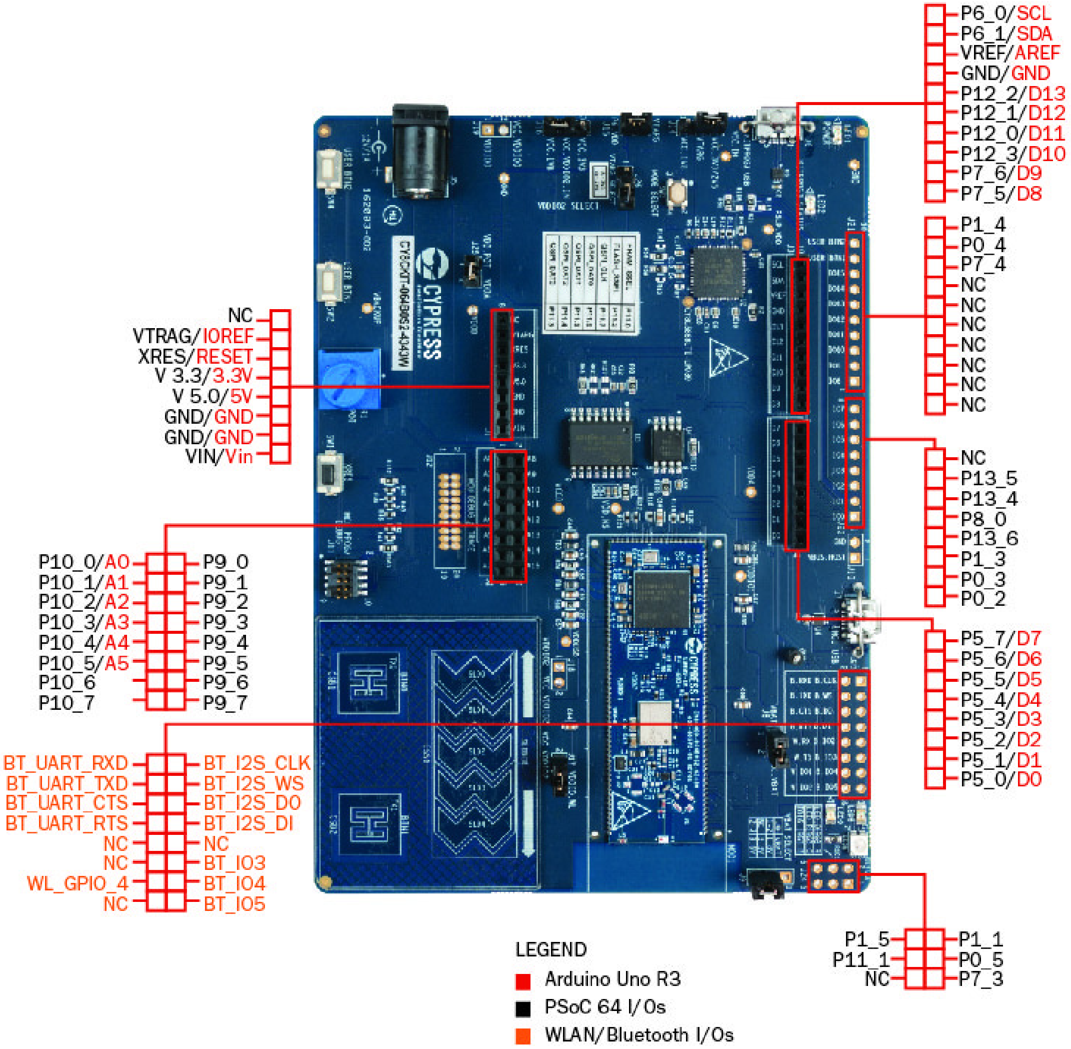


Table 1-1. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout

Pin	Primary On-board Function	Secondary On-board Function	Connection details
PSoC 64 Pins			
XRES	Hardware Reset	–	–
P0[2]	GPIO on non-Arduino header IO0 (J22.1)	–	–
P0[3]	GPIO on non-Arduino header IO1 (J22.2)	–	–
P0[4]	User button (SW2) with Hibernate wakeup capability	GPIO on non-Arduino header (J21.9)	–
P0[5]	RGB green LED (LED5)	GPIO on non-Arduino header (J24.3)	–
P1[0]	CapSense RX for buttons and CapSense TX for sliders	GPIO on non-Arduino header IO7 (J22.8)	Remove R33 to disconnect from CapSense. Populate R145 to connect to GPIO on non-Arduino header.
P1[1]	RGB red LED (LED5)	GPIO on non-Arduino header (J24.1)	–
P1[2]	USB Host Enable	–	–
P1[3]	GPIO on non-Arduino header IO2 (J22.3)	–	–
P1[4]	User button (SW4) with Hibernate wakeup capability	GPIO on non-Arduino header (J21.10)	–
P1[5]	Orange user LED (LED8)	GPIO on non-Arduino header (J24.2)	–
P5[0]	UART_RX	Arduino D0 (J4.1)	Remove R21 to disconnect from KitProg3. Populate R115 to connect to GPIO on non-Arduino header.
P5[1]	UART_TX	Arduino D1 (J4.2)	Remove R61 to disconnect from KitProg3. Populate R116 to connect to GPIO on non-Arduino header.
P5[2]	UART_RTS	Arduino D2 (J4.3)	Remove R19 to disconnect from KitProg3.
P5[3]	UART_CTS	Arduino D3 (J4.4)	Remove R18 to disconnect from KitProg3.
P5[4]	Arduino D4 (J4.5)	–	–
P5[5]	Arduino D5 (J4.6)	–	–
P5[6]	Arduino D6 (J4.7)	–	–
P5[7]	Arduino D7 (J4.8)	–	–
P6[0]	I2C SCL	Arduino (J3.10)	Remove R58 to disconnect from KitProg3.
P6[1]	I2C SDA	Arduino (J3.9)	Remove R59 to disconnect from KitProg3.
P6[2]	USB VBUS Detect	–	–
P6[3]	USB Interrupt	–	–

Table 1-1. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout *(continued)*

Pin	Primary On-board Function	Secondary On-board Function	Connection details
P6[4]	PSoC 64 device JTAG TDO/SWD SWO	–	–
P6[5]	PSoC 64 device JTAG TDI	–	–
P6[6]	PSoC 64 device JTAG TMS/SWD SWDIO	–	–
P6[7]	PSoC 64 device JTAG TCK/SWD SWCLK	–	–
P7[0]	ETM Clock	–	–
P7[1]	CapSense CINTA	–	–
P7[2]	CapSense CINTB	–	–
P7[3]	RGB blue LED (LED5)	GPIO on non-Arduino header (J24.5)	–
P7[4]	GPIO on non-Arduino header IO15 (J21.8)	CapSense Shield	Remove R155 to disconnect from IO15 (J21.8). Populate R38 to connect to CapSense Shield.
P7[5]	Arduino D8 (J3.1)	–	–
P7[6]	Arduino D9 (J3.2)	–	–
P7[7]	CapSense CMOD	–	–
P8[0]	GPIO on non-Arduino header IO4 (J22.5)	–	–
P8[1]	CapSense Button0 TX	GPIO on non-Arduino header IO8 (J21.1)	Remove R24 to disconnect from CapSense. Populate R144 to connect to GPIO on non-Arduino header.
P8[2]	CapSense Button1 TX	GPIO on non-Arduino header IO9 (J21.2)	Remove R25 to disconnect from CapSense. Populate R143 to connect to GPIO on non-Arduino header.
P8[3]	CapSense Slider0 RX	GPIO on non-Arduino header IO10 (J21.3)	Remove R28 to disconnect from CapSense. Populate R142 to connect to GPIO on non-Arduino header.
P8[4]	CapSense Slider1 RX	GPIO on non-Arduino header IO11 (J21.4)	Remove R29 to disconnect from CapSense. Populate R152 to connect to GPIO on non-Arduino header.
P8[5]	CapSense Slider2 RX	GPIO on non-Arduino header IO12 (J21.5)	Remove R30 to disconnect from CapSense. Populate R153 to connect to GPIO on non-Arduino header.
P8[6]	CapSense Slider3 RX	GPIO on non-Arduino header IO13 (J21.6)	Remove R31 to disconnect from CapSense. Populate R151 to connect to GPIO on non-Arduino header.

Table 1-1. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout *(continued)*

Pin	Primary On-board Function	Secondary On-board Function	Connection details
P8[7]	CapSense Slider4 RX	GPIO on non-Arduino header IO14 (J21.7)	Remove R32 to disconnect from CapSense. Populate R149 to connect to GPIO on non-Arduino header.
P9[0]	Extended Arduino A8 (J2.2)	ETM TRACEDATA[3]	Remove R125 to disconnect from J2 header. Populate R126 to connect to ETM Trace header.
P9[1]	Extended Arduino A9 (J2.4)	ETM TRACEDATA[2]	Remove R124 to disconnect from J2 header. Populate R127 to connect to ETM Trace header.
P9[2]	Extended Arduino A10 (J2.6)	ETM TRACEDATA[1]	Remove R123 to disconnect from J2 header. Populate R128 to connect to ETM Trace header.
P9[3]	Extended Arduino A11 (J2.8)	ETM TRACEDATA[0]	Remove R117 to disconnect from J2 header. Populate R129 to connect to ETM Trace header.
P9[4]	Extended Arduino A12 (J2.10)	–	–
P9[5]	Extended Arduino A13 (J2.12)	–	–
P9[6]	Extended Arduino A14 (J2.14)	–	–
P9[7]	Extended Arduino A15 (J2.16)	–	–
P10[0]	Arduino A0 (J2.1)	–	–
P10[1]	Arduino A1 (J2.3)	–	–
P10[2]	Arduino A2 (J2.5)	–	–
P10[3]	Arduino A3 (J2.7)	–	–
P10[4]	Arduino A4 (J2.9)	–	–
P10[5]	Arduino A5 (J2.11)	–	–
P10[6]	Potentiometer output	Extended Arduino A6 (J2.13)	Remove R51 to disconnect from potentiometer.
P10[7]	Extended Arduino A7 (J2.15)	–	–
P11[0]	QSPI F-RAM CS	–	–
P11[1]	Red user LED (LED9)	GPIO on non-Arduino header (J24.4)	–
P11[2]	QSPI Flash CS	–	–
P11[3:6]	QSPI Flash IO[3:0]	–	–
P11[7]	QSPI Flash CLK	–	–

Table 1-1. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout *(continued)*

Pin	Primary On-board Function	Secondary On-board Function	Connection details
P12[0]	Arduino header D11 (J3.4)	–	–
P12[1]	Arduino header D12 (J3.5)	–	–
P12[2]	Arduino header D13 (J3.6)	–	–
P12[3]	Arduino header D10 (J3.3)	–	–
P12[4]	microSD card CMD	–	Remove R168 to disconnect from microSD card connector.
P12[5]	microSD card CLK	–	Remove R166 to disconnect from microSD card connector.
P12[6]	ECO Crystal XIN	–	–
P12[7]	ECO Crystal XOUT	–	–
P13[0]	microSD card DAT0	microSD card MOSI	Remove R164 to disconnect microSD port (J20.7) from SDIO. Populate R169 to connect microSD (J20.3) to SPI.
P13[1]	microSD card DAT1	microSD card MISO	Remove R163 to disconnect microSD port (J20.8) from SDIO. Populate R165 to connect microSD (J20.7) to SPI.
P13[2]	microSD card DAT2	microSD card SPI CLK	Remove R162 to disconnect microSD port (J20.1) from SDIO. Populate R167 to connect microSD (J20.5) to SPI.
P13[3]	microSD card DAT3	microSD card SPI SSEL	–
P13[4]	GPIO on non-Arduino header IO5 (J22.6)	–	–
P13[5]	GPIO on non-Arduino header IO6 (J22.7)	–	–
P13[6]	GPIO on non-Arduino header IO3 (J22.4)	–	–
P13[7]	microSD card chip detect	GPIO on non-Arduino header IO16 (J24.6)	Remove R161 to disconnect from microSD card detect. Populate R160 to connect to IO16 (J24.6).
CYW4343W Pins			
BT_UART_TXD	UART interface with Host MCU (PSoC 64 device)	–	–
BT_UART_RXD	UART interface with Host MCU (PSoC 64 device)	–	–

Table 1-1. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board Pinout *(continued)*

Pin	Primary On-board Function	Secondary On-board Function	Connection details
BT_UART_CTS	UART interface with Host MCU (PSoC 64 device)	–	–
BT_UART_RTS	UART interface with Host MCU (PSoC 64 device)	–	–
BT_I2S_CLK	I2S serial clock	–	–
BT_I2S_WS	I2S serial word select	–	–
BT_I2S_DO	I2S serial data out	–	–
BT_I2S_DI	I2S serial data in	–	–
BT_IO_3	Bluetooth general-purpose IO	–	–
BT_IO_4	Bluetooth general-purpose IO	–	–
BT_IO_5	Bluetooth general-purpose IO	–	–
WL_GPIO_4	Programmable GPIO	–	–

1.5 Additional Learning Resources

Cypress provides a wealth of data at www.cypress.com/psoc64 to help you to select the right PSoC device for your design and to help you to quickly and effectively integrate the device into your design.

1.6 Technical Support

For assistance, go to www.cypress.com/support. Visit community.cypress.com to ask your questions in Cypress developer community.

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

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

1.7 Documentation Conventions

Table 1-2. Document Conventions for Guides

Convention	Usage
`Courier New`	Displays file locations, user entered text, and source code: `C:\...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Creator User Guide*.

Table 1-2. Document Conventions for Guides

Convention	Usage
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes cautions or unique functionality of the product.

1.8 Acronyms

Table 1-3. Acronyms Used in this Document

Acronym	Definition
ADC	Analog-to-Digital Converter
BLE	Bluetooth Low Energy
BOM	Bill of Materials
BT	Bluetooth
CINT	Integration Capacitor
CMOD	Modulator Capacitor
CPU	Central Processing Unit
CSD	CapSense Sigma Delta
CSX	CapSense Crosspoint
DC	Direct Current
Del-Sig	Delta-Sigma
DMA	Direct Memory Access
ECO	External Crystal Oscillator
ESD	Electrostatic Discharge
ETM	Embedded Trace Macrocell
GPIO	General-Purpose Input/Output
HID	Human Interface Device
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
IC	Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
LED	Light-emitting Diode
LPO	Low Power Oscillator
PC	Personal Computer
PDM	Pulse Density Modulation
PSoC	Programmable System-on-Chip
PWM	Pulse Width Modulation

Table 1-3. Acronyms Used in this Document *(continued)*

Acronym	Definition
QSPI	Quad Serial Peripheral Interface
SAR	Successive Approximation Register
SDHC	Secure Digital Host Controller
SDIO	Secure Digital Input Output
SMIF	Serial Memory Interface
SPI	Serial Peripheral Interface
SRAM	Serial Random Access Memory
SWD	Serial Wire Debug
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WCO	Watch Crystal Oscillator

2. Software Installation

This chapter describes the steps to install the software tools and packages on a PC for using the CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Pioneer Kit. This includes ModusToolbox software on which the applications will be built and used for programming. The detailed steps for installation on macOS and Linux can be found in the “Secure Boot” SDK user guide.

2.1 Before You Begin

To install Cypress software, you will require administrator privileges. However, they are not required to run the software once it has been installed. Before you install the kit software, close any other Cypress software that is currently running.

2.2 Install Software

Follow these steps to install the software:

1. Install the latest version of ModusToolbox software which should be version 2.3.1 or later.

> The installation of ModusToolbox 2.3.1 software or later on a Windows PC provides all the tools required to build, program and provision devices. Windows users may skip the rest of this chapter.

2. Install Python 3.7 or later on your Linux or macOS computer if it isn’t already installed.
3. Set up the appropriate environment variable:

Linux: Most distributions of Linux should already have python2 and python3 installed. To verify that python by default points to python3 run:

```
python --version
```

If python3 is not set as default run following commands. The number at the end of each command denotes a priority

```
update-alternatives --install /usr/bin/python python /usr/bin/python2.7 1
update-alternatives --install /usr/bin/python python /usr/bin/python3.7 2
```

macOS: By default, ‘python’ points to /usr/bin/python which is python2. To make 'python' and 'pip' resolve to python3 versions, execute the following:

```
echo 'alias python=python3' >> ~/.bash_profile
echo 'alias pip=pip3' >> ~/.bash_profile
source ~/.bash_profile
```

Make sure that you have the latest version of pip installed, use the following command

```
python -m pip install --upgrade pip
```

To verify that ‘python’ and ‘pip’ by default point to python3, run:

```
python --version
Python 3.7.4
pip --version
pip 19.0.3 from /Library/Frameworks/Python.framework/Versions/3.7/lib/
python3.7/site-packages/pip (python 3.7)
```

4. Install the “Secure Boot” SDK package by running the following from a terminal/command prompt.

   ```
   pip install -U cysecuretools
   ```

   During installation, there may be error messages when installing colorama, protobuf and json-schema. These can be safely ignored.

5. Install the libusb dependency for pyOCD. Please check README.md for the latest instructions on installing libusb. From the README.md file:

   How to install libusb depends on your OS:

   - macOS: use Homebrew: brew install libusb
   - Linux: should already be installed.

3. Kit Operation

This chapter introduces you to various features of the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board, including the theory of operation and the onboard KitProg3 programming and debugging functionality, USB-UART and USB-I2C bridges.

3.1 Theory of Operation

The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board is built around a PSoC 64 chip. Figure 3-1 shows the block diagram of the PSoC 64 device used on the board. For details of device features, see the device datasheet.

Figure 3-1. PSoC 64 Block Diagram

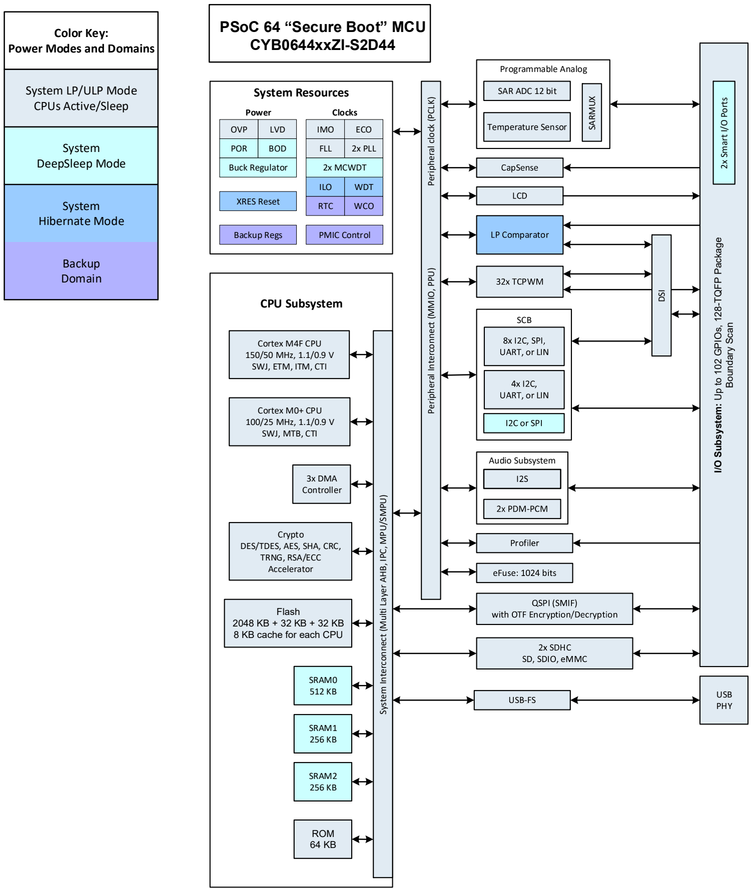


Figure 3-2 shows the block diagram of the CY8CKIT-064B0S2-4343W Carrier Module.

Figure 3-3 shows the block diagram of the CYW9-BASE-01 Pioneer Board. The CYW9-BASE-01 is the base board on which the CY8CMOD-064B0S2-4343W castellated carrier module is populated on.

Figure 3-2. Block Diagram of CY8CMOD-064B0S2-4343W (Carrier Module)

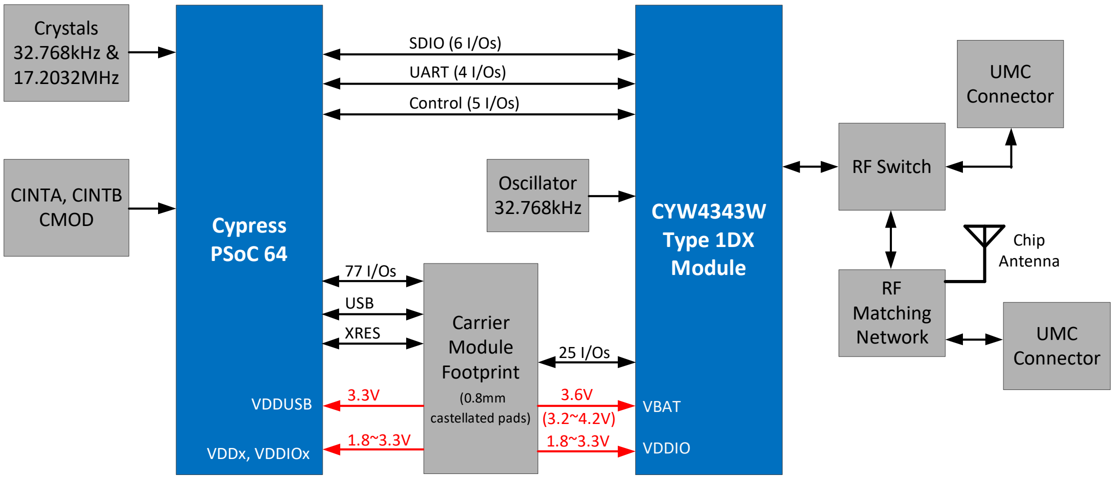
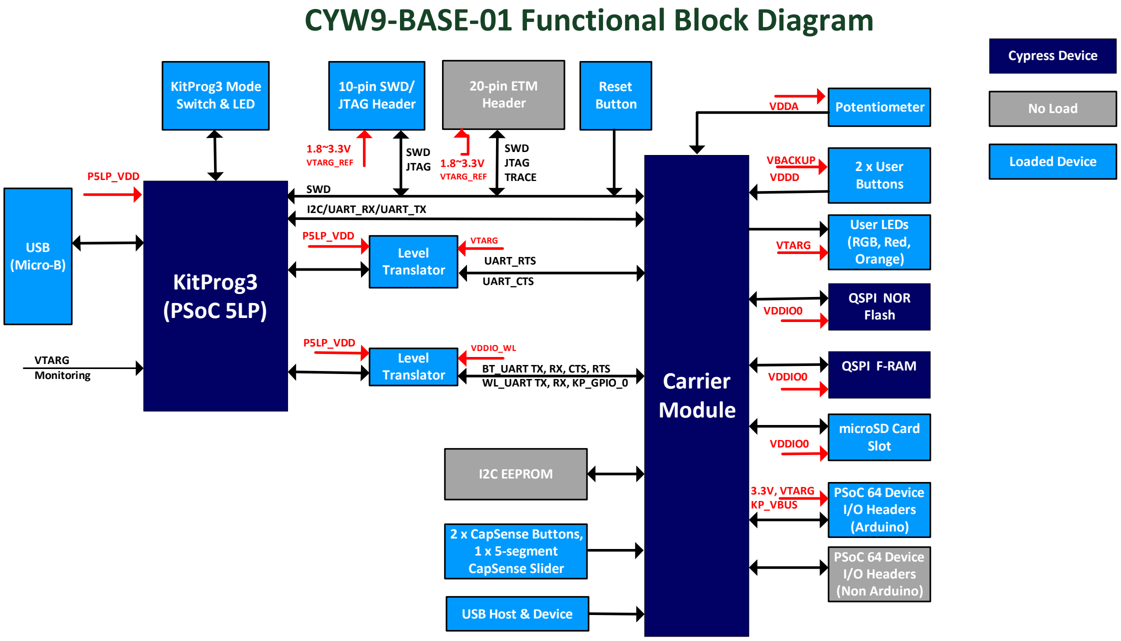


Figure 3-3. Block Diagram of Pioneer Board



The CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit comes with the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board. Figure 3-4 and Figure 3-5 show the markup of the Pioneer Board.

Figure 3-4. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board - Top View

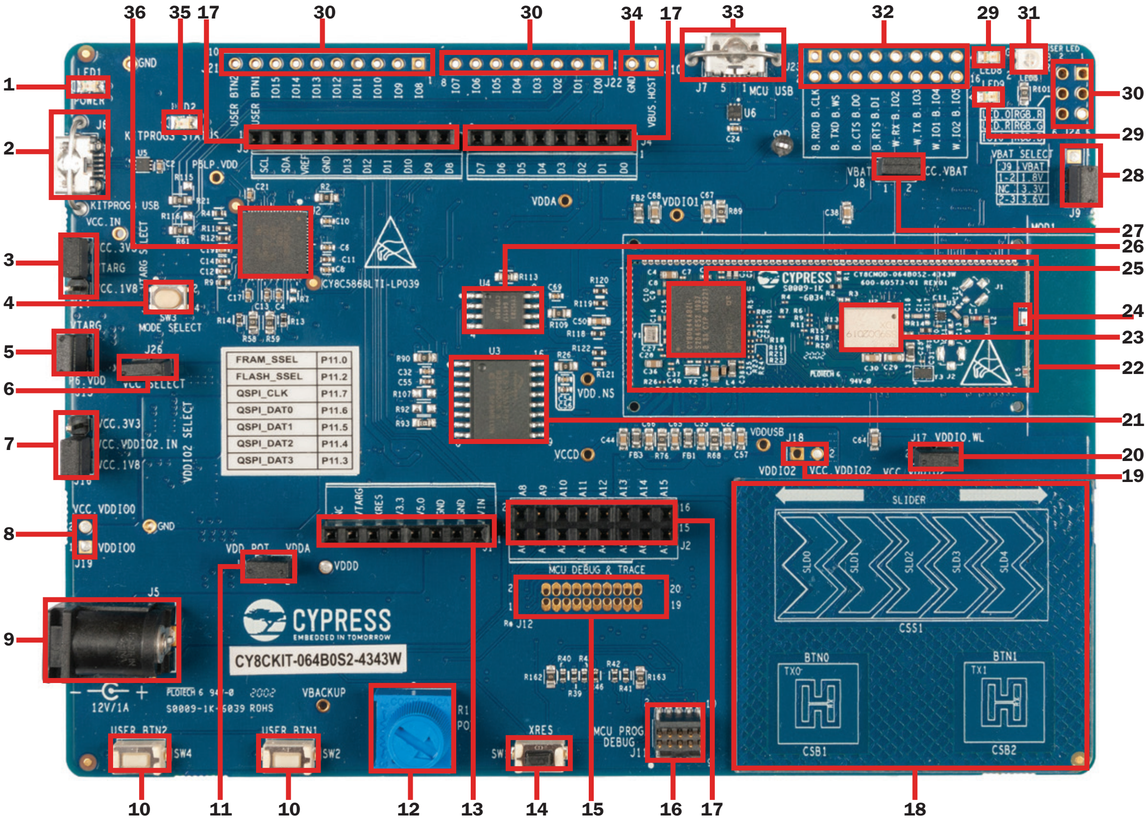
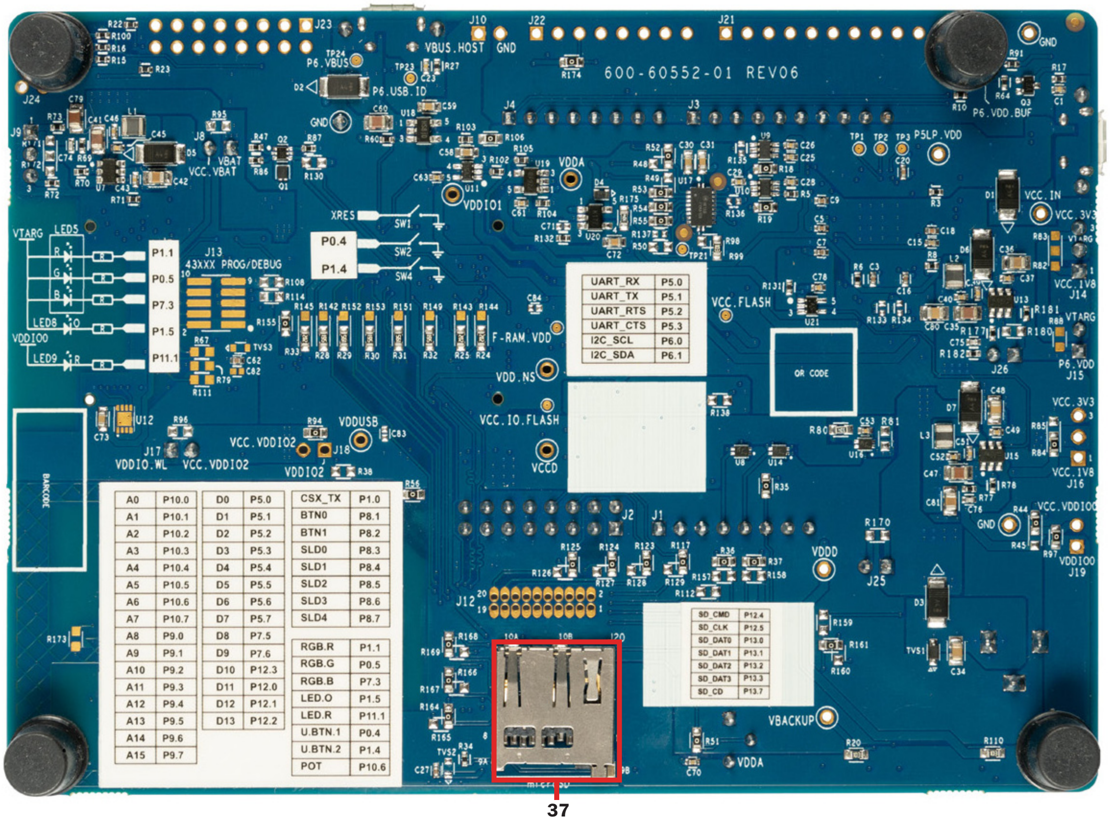


Figure 3-5. PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board - Bottom View



The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board has the following peripherals:

1. **Power LED (LED1):** This Yellow LED indicates the status of power supplied to board.
2. **KitProg3 USB connector (J6):** The USB cable provided along with the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board connects between this USB connector and the PC to use the KitProg3 onboard programmer and debugger and to provide power to the board.
3. **PSoC 64 VDD power selection jumper (J14):** This jumper is used to select the PSoC 64 device VDD supply voltage between 1.8 V and 3.3 V. During provisioning, this jumper must be in the 3.3 V position and jumper shunt from J26 must be removed. This provides 2.5 V to the PSoC 64 VDD which is required to blow eFuses.
4. **KitProg3 programming mode selection button (SW3):** This button can be used to switch between various modes of operation of KitProg3 (CMSIS-DAP BULK or DAPLink modes). For more details, see the KitProg3 User Guide.
5. **PSoC 64 VDD current measurement jumper (J15):** An ammeter can be connected to this jumper to measure the current consumed by the PSoC 64 VDD power domain.
6. **PSoC 64 VDD select jumper (J26):** This jumper is used to change VCC_3V3 to 2.5 V. When jumper shunt is present, VCC_3V3 is 3.3 V and when removed, it is 2.5 V. This is needed for provisioning. The kit is not intended to be used at 2.5 V during normal operation.
7. **PSoC 64 VDDIO2 and CYW4343W VDDIO power selection jumper (J16):** This jumper is used to select the PSoC 64 VDDIO2 and CYW4343W VDDIO supply voltage between 1.8 V and 3.3 V. The default value is 1.8 V.
8. **PSoC 64 VDDIO0 current measurement jumper (J19):** An ammeter can be connected to this jumper to measure the current consumed by the PSoC 64 VDDIO0 power domain. This is not loaded by default.

9. **External power supply VIN connector (J5):** This connector connects an external DC power supply input to the onboard regulators.
10. **PSoC 64 user buttons (SW2 and SW4):** These buttons can be used to provide an input to PSoC 64 device. Note that by default these buttons connect the PSoC 64 pin to ground when pressed, so you need to configure the PSoC 64 device pin as a digital input with resistive pull-up for detecting the button press. These buttons can also provide a wake-up source from low-power modes of the device.
11. **Potentiometer connection jumper (J25):** This jumper connects the PSoC 64 VDDA to the potentiometer. Remove this jumper when measuring the PSoC 64 power consumption.
12. **Potentiometer (R1):** This is a 10k Ohm potentiometer connected to PSoC 64 pin P10[6]. It can be used to simulate a sensor output to the PSoC 64 device.
13. **Arduino-compatible power header (J1):** This header powers the Arduino shields. It also has a provision to power the kit though the VIN input.
14. **PSoC 64 reset button (SW1):** This button is used to reset the PSoC 64 device. It connects the PSoC 64 reset (XRES) pin to ground when pressed.
15. **PSoC 64 debug and trace header (J12):** This header can be connected to an Embedded Trace Macrocell (ETM)-compatible programmer/debugger. This is not loaded by default.
16. **PSoC 64 program and debug header (J11):** This 10-pin header allows you to program and debug the PSoC 64 using an external programmer such as MiniProg4.
17. **Arduino Uno R3-compatible I/O headers (J2, J3, and J4):** These I/O headers bring out pins from PSoC 64 device to interface with the Arduino shields. Some of these pins are multiplexed with onboard peripherals and are not connected to PSoC 64 device by default. For detailed information on how to rework the kit to access these pins, see Table 1-1 on page 11.
18. **CapSense slider (SLIDER) and buttons (BTN0 and BTN1):** The CapSense touch-sensing slider and two buttons, all of which are capable of both self-capacitance (CSD) and mutual-capacitance (CSX) operation, allow you to evaluate Cypress' fourth-generation CapSense technology. The slider and buttons have a 1-mm acrylic overlay for smooth touch sensing.
19. **PSoC 64 VDDIO2 current measurement jumper (J18):** An ammeter can be connected to this jumper to measure the current consumed by the PSoC 64 VDDIO2 power domain. This jumper is not loaded by default on the board. Before populating the jumper for current measurements, ensure that R94 is removed.
20. **CYW4343W VDDIO_WL current measurement jumper (J17):** An ammeter can be connected to this jumper to measure the current consumed by the CYW4343W VDDIO power domain.
21. **Cypress serial NOR flash memory (S25FL512S, U3):** A S25FL512S NOR flash of 512-Mbit capacity is connected to the Quad SPI interface of the PSoC 64. The NOR device can be used for both data and code memory with execute-in-place (XIP) support and encryption.
22. **Cypress PSoC 64 with CYW4343W Carrier Module (CY8CMOD-064B0S2-4343W, MOD1):** This kit is designed to highlight the features of the PSoC 64 chip on the CY8CMOD-064B0S2-4343W. For details, see CY8CMOD-064B0S2-4343W (MOD1) on page 40.
23. **CYW4343W based Murata Type 1DX module:** The Type 1DX module is an ultra-small module that includes 2.4 GHz and WLAN and Bluetooth functionality. Based on Cypress CYW4343W, the module provides high-efficiency RF front end circuits. To ease Wi-Fi certification, the Type 1DX module complies with IEEE 802.11b/g/n and Bluetooth 5.1 plus EDR, Power Class 1 + BLE.
24. **Wi-Fi/BT antenna:** This is the onboard antenna connected to the Wi-Fi and Bluetooth module.
25. **PSoC 64:** This kit is designed to highlight the features of the PSoC 64 device. For details on PSoC 64 pin mapping, refer to Table 1-1 on page 11.

26. **Cypress serial Ferroelectric RAM (CY15B104QSN, U4):** The CY15B104QSN is a 4-Mbit non-volatile memory employing an advanced ferroelectric process. F-RAM is nonvolatile and performs reads and writes similar to a RAM. It provides reliable data retention for 151 years and is connected to the Quad SPI interface of the PSoC 64.
27. **CYW4343W VBAT current measurement jumper (J8):** An ammeter can be connected to this jumper to measure the current consumed by the CYW4343W VBAT power domain.
28. **CYW4343W VBAT power selection jumper (J9):** This jumper is used to select the CYW4343W VBAT supply voltage between 1.8 V, 3.3 V and 3.6 V. This board supports VBAT voltages of 3.3 V and 3.6 V. VBAT is 3.3 V when the jumper is not inserted and 3.6 V when the jumper is inserted.
29. **PSoC 64 device user LEDs (LED8 and LED9):** These two user LEDs can operate at the entire operating voltage range of the PSoC 64 device. The LED is active LOW, so the pins must be driven to ground to turn ON the LED.
30. **PSoC 64 device I/O headers (J21, J22, and J24):** These headers provide connectivity to PSoC 64 GPIOs that are not connected to the Arduino compatible headers. Some of these I/Os are also connected to on-board peripherals see Table 1-1 on page 11 for pin mapping. These are not loaded by default.
31. **RGB LED (LED5):** This onboard RGB LED can be controlled by the PSoC 64. The LEDs are active LOW, so the pins must be driven to ground to turn ON the LEDs.
32. **Wi-Fi/BT GPIO header (J23):** This header brings out a few I/Os of the CYW4343W for general purpose applications. This is not loaded by default.
33. **PSoC 64 USB device connector (J7):** The USB cable provided with the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit can be connected between this USB connector and the PC to use the PSoC 64 USB device applications.
34. **Optional USB Host power supply header (J10):** This header provides an option to supply external power to the PSoC 64 USB when used as a USB Host. This is not loaded by default.
35. **KitProg3 status LED (LED2):** This Yellow LED indicates the status of KitProg3. The KitProg3 mode is selected using Mode Select button SW3. For details on the KitProg3 status, see the KitProg3 User Guide.
36. **KitProg3 (PSoC 5LP) programmer and debugger (CY8C5868LTI-LP039, U2):** The PSoC 5LP device (CY8C5868LTI-LP039) serving as KitProg3 is a multi-functional system which includes a SWD programmer, debugger, USB-I2C bridge and USB-UART bridge. For more details, see the KitProg3 User Guide.
37. **microSD Card holder (J20):** Provide SDHC interface with microSD cards with the option to detect the presence of the card.

See Hardware Functional Description on page 40 for details on various hardware blocks.

3.2 KitProg3: On-Board Programmer/Debugger

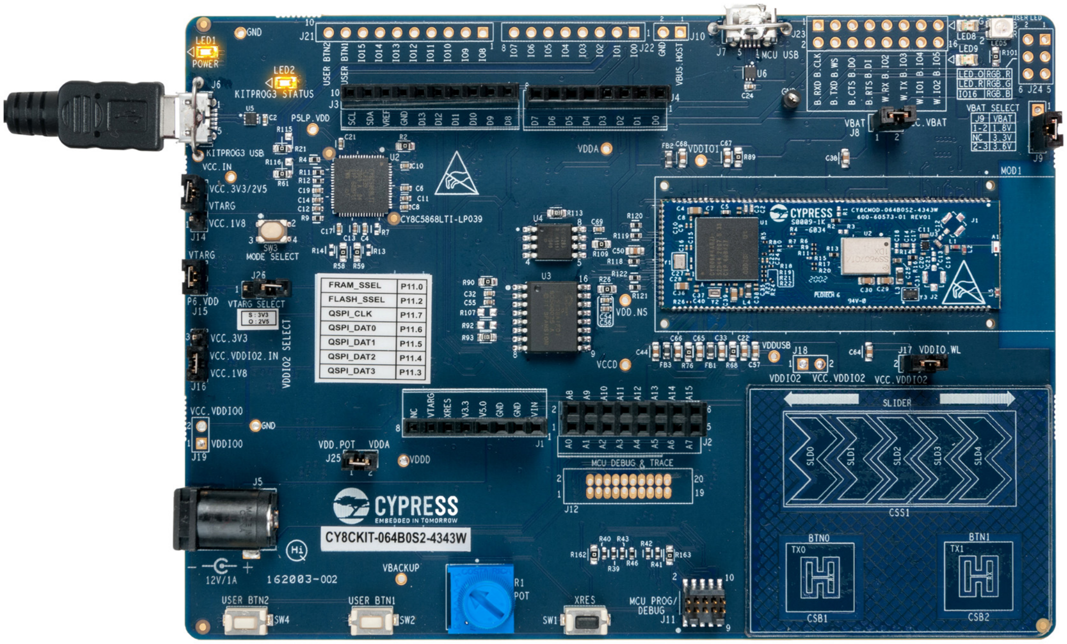
The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board can be programmed and debugged using the onboard KitProg3. KitProg3 is a programmer/debugger with USB-UART and USB-I2C functionality. A Cypress PSoC 5LP device is used to implement KitProg3 functionality. For more details on the KitProg3 functionality, see the KitProg3 User Guide.

3.2.1 Programming and Debugging using ModusToolbox Software

This section presents a quick overview on how to connect the kit and select the KitProg3 mode for programming and debugging. For detailed instructions, see **Help** > **ModusToolbox IDE Documentation** > **User Guide**.

Connect the board to the PC using the USB cable, as shown in Figure 3-6. It enumerates as a USB Composite Device if you are connecting it to your PC for the first time. KitProg3 operates in either CMSIS-DAP Bulk mode, or DAPLink mode (default). ModusToolbox software requires CMSIS-DAP Bulk mode to program or debug the device while the provisioning process requires DAPLink mode. DAPLink mode is also required for programming in Arm Mbed. In CMSIS-DAP Bulk mode, two UART interfaces are supported. The status LED (Yellow) is always ON in CMSIS-DAP Bulk mode, and ramping at 2 Hz rate in DAPLink mode. Press and release the Mode select button (SW3) to switch between these modes. If you do not see the desired LED status, see the KitProg3 User Guide for details on the KitProg3 status and troubleshooting instructions.

Figure 3-6. Connect USB Cable to USB Connector on the Board

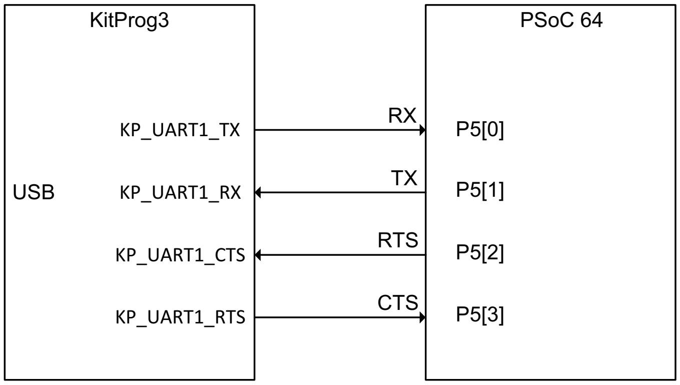


3.2.2 USB-UART Bridge

The KitProg3 on the PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board can act as a USB-UART bridge.

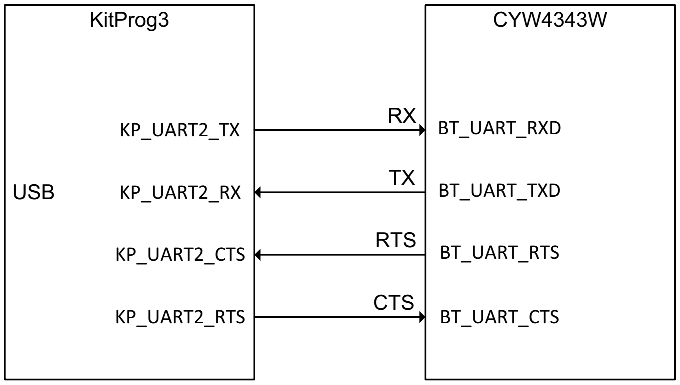
The primary UART and flow-control lines between the PSoC 64 device and the KitProg3 are hard-wired on the board, as Figure 3-7 shows.

Figure 3-7. UART Connection between KitProg3 and PSoC 64 Device



The secondary UART and flow-control lines between the CYW4343W and the KitProg3 are hard-wired on the board, as Figure 3-8 shows.

Figure 3-8. UART Connection between KitProg3 and CYW4343W

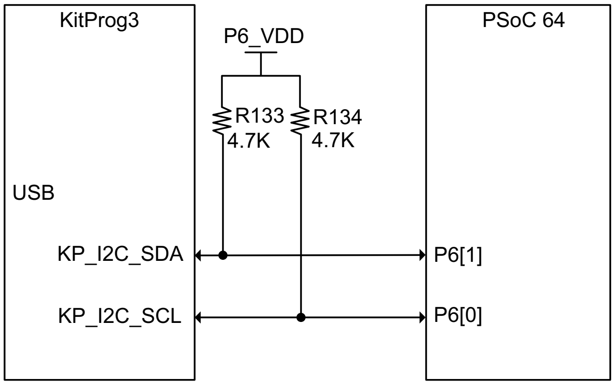


For more details on the KitProg3 USB-UART functionality, see the KitProg3 User Guide.

3.2.3 USB-I2C Bridge

The KitProg3 can function as a USB-I2C bridge and can communicate with the Bridge Control Panel (BCP) software which acts as an I2C master. The I2C lines on the PSoC 64 chip are hard-wired on the board to the I2C lines of the KitProg3, with onboard pull-up resistors as Figure 3-9 shows. The USB-I2C supports speeds of 50 kHz, 100 kHz, 400 kHz, and 1 MHz. For more details on the KitProg3 USB-I2C functionality, see the KitProg3 User Guide.

Figure 3-9. I2C Connection between KitProg3 and PSoC 64 Chip



4. Running Code on PSoC 64 “Secure Boot” MCUs

The CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit can run code examples available on ModusToolbox software. However, prior to running any code on the PSoC 64 Line of Secured MCUs, they must first be provisioned with keys and device security policies so only signed code can be executed.

This section will go through the process to provision, build, program and run the *“Secure Blinky LED FreeRTOS”* example. Before going through the detailed steps, an overview of the provisioning process will be presented.

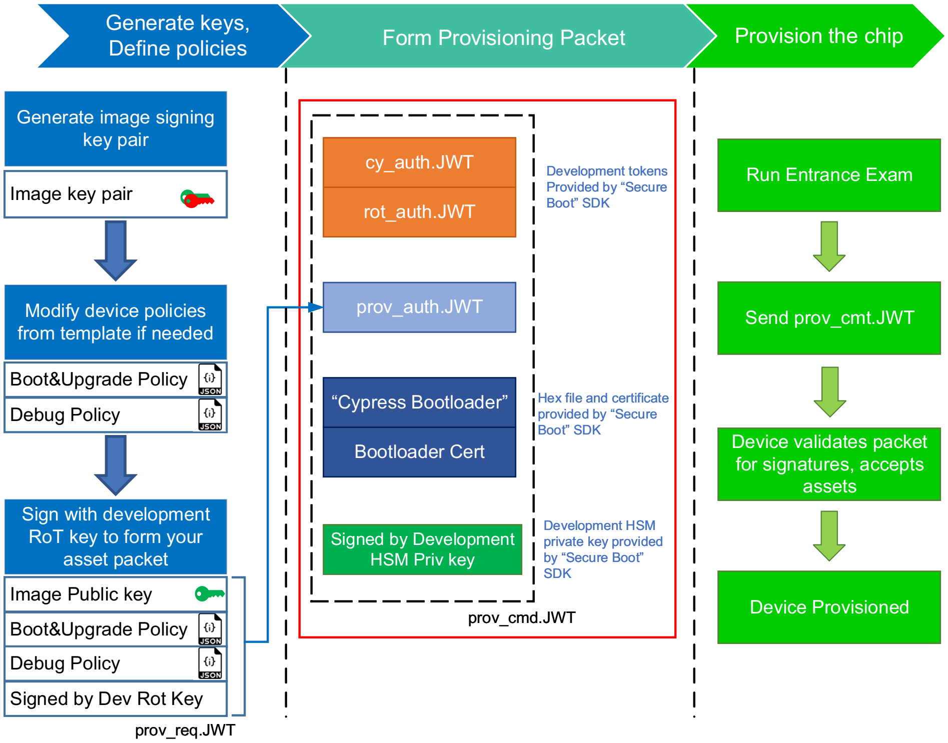
4.1 Provisioning Overview

Provisioning is a process by which secure assets like keys and security policies are injected into the device. This step typically occurs in a secure manufacturing environment that has a Hardware Security Module (HSM).

While this kit user guide covers the steps required to evaluate the kit, it is recommended to go through the “Secure Boot” SDK user guide for understanding the flow in more detail.

In the context of evaluating this kit, the provisioning flow can be visualized as follows:

Figure 4-1. Provisioning Flow


Generate keys, Define policies
Form Provisioning Packet
Provision the chip
Generate image signing key pair
Image key pair
Modify device policies from template if needed
Boot&Upgrade Policy
Debug Policy
Sign with development RoT key to form your asset packet
Image Public key
Boot&Upgrade Policy
Debug Policy
Signed by Dev Rot Key
prov_req.JWT
cy_auth.JWT
rot_auth.JWT
Development tokens Provided by “Secure Boot” SDK
prov_auth.JWT
“Cypress Bootloader”
Bootloader Cert
Hex file and certificate provided by “Secure Boot” SDK
Signed by Development HSM Priv key
Development HSM private key provided by “Secure Boot” SDK
prov_cmd.JWT
Run Entrance Exam
Send prov_cmt.JWT
Device validates packet for signatures, accepts assets
Device Provisioned

For evaluation purposes, the “Secure Boot” SDK provides the following assets to easily provision your device:

1. A development cy_auth JWT token; this authorizes a development HSM keypair which is used by your PC to provision the chip.
2. A development rot_auth JWT token; this authorizes a development RoT keypair which can be used to sign your assets, such as image keys and policies.

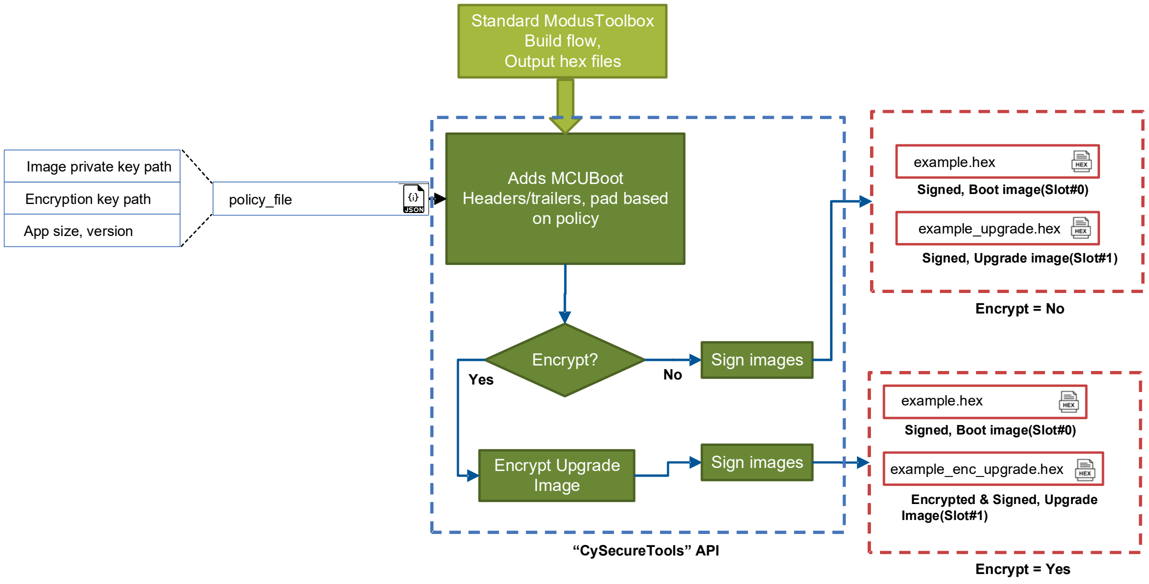
In addition, the SDK provides “CySecureTools” to do the following:

1. Generate image keys
2. Form provisioning packets
3. Scripts to run the entrance exam and provisioning process on your development PC

Once the chip has been provisioned with the Public Image key, it will only boot images signed by the associated Private key. Optionally, the image can be encrypted if the Boot and Upgrade policy specifies it.

The signing and encryption process is a post build script provided by the “Secure Boot” SDK. The build and encrypt/signing flow for a CY8CKIT-064B0S2-4343W target using the ModusToolbox software make process is shown below.

Figure 4-2. Build and Encrypt/Signing Flow

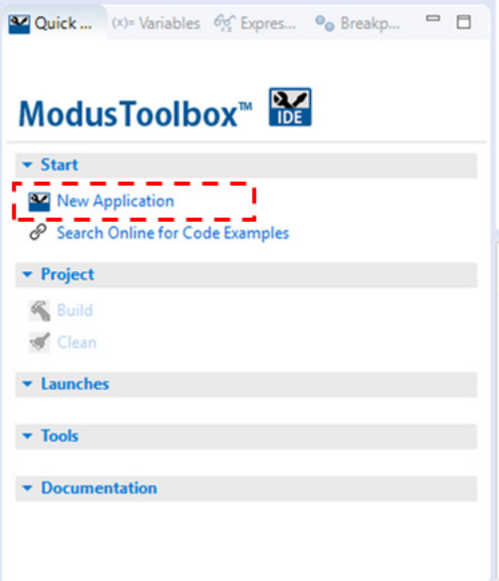


4.2 Create ModusToolbox Example Project

Now that an overview of the provisioning process has been provided, the steps to create an application, provision the device, and then build, program, run the application will be shown in detail.

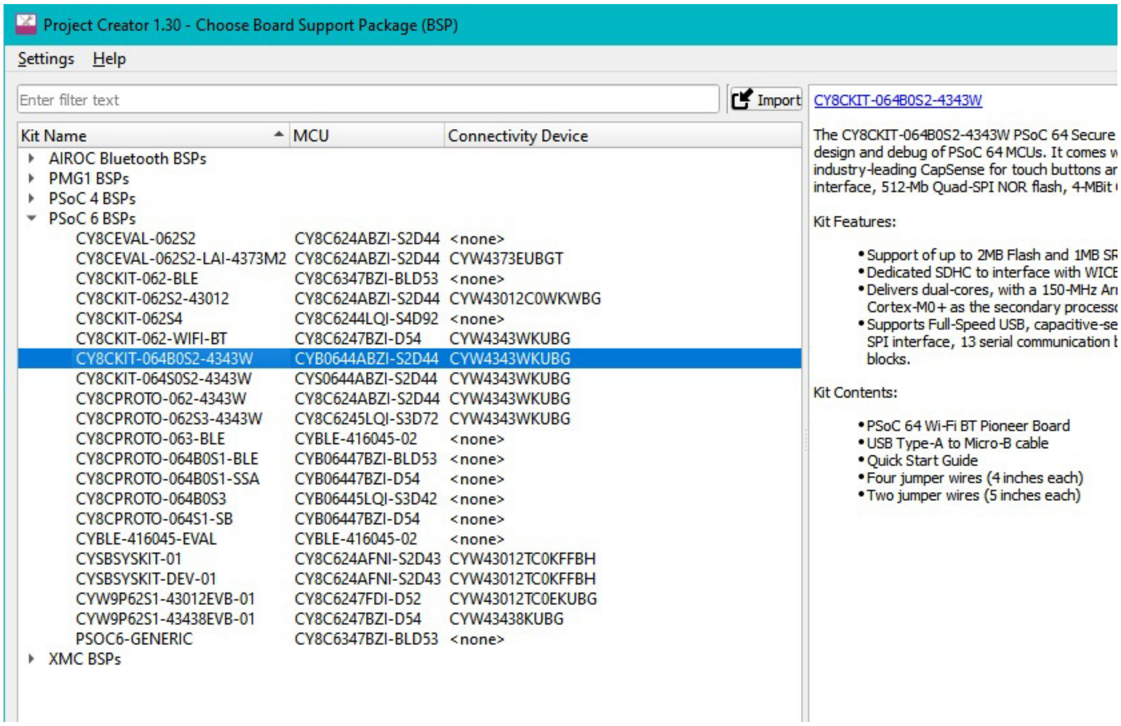
1. Open the Eclipse IDE for ModusToolbox and create a new application.

Figure 4-3. Create New Application



2. Select the CY8CKIT-064B0S2-4343W target and click **Next**.

Figure 4-4. CY8CKIT-064B0S2-4343W Target Selection



Kit Name	MCU	Connectivity Device
AIROC Bluetooth BSPs		
PMG1 BSPs		
PSoC 4 BSPs		
PSoC 6 BSPs		
CY8CEVAL-062S2	CY8C624ABZI-S2D44	<none>
CY8CEVAL-062S2-LAI-4373M2	CY8C624ABZI-S2D44	CYW4373EUBGT
CY8CKIT-062-BLE	CY8C6347BZI-BLD53	<none>
CY8CKIT-062S2-43012	CY8C624ABZI-S2D44	CYW43012C0WKWBG
CY8CKIT-062S4	CY8C6244LQI-S4D92	<none>
CY8CKIT-062-WIFI-BT	CY8C6247BZI-D54	CYW4343WKUBG
CY8CKIT-064B0S2-4343W	CYB0644ABZI-S2D44	CYW4343WKUBG
CY8CKIT-064S0S2-4343W	CYS0644ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062-4343W	CY8C624ABZI-S2D44	CYW4343WKUBG
CY8CPROTO-062S3-4343W	CY8C6245LQI-S3D72	CYW4343WKUBG
CY8CPROTO-063-BLE	CYBLE-416045-02	<none>
CY8CPROTO-064B0S1-BLE	CYB06447BZI-BLD53	<none>
CY8CPROTO-064B0S1-SSA	CYB06447BZI-D54	<none>
CY8CPROTO-064B0S3	CYB06445LQI-S3D42	<none>
CY8CPROTO-064S1-SB	CYB06447BZI-D54	<none>
CYBLE-416045-EVAL	CYBLE-416045-02	<none>
CYSBSYSKIT-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYSBSYSKIT-DEV-01	CY8C624AFNI-S2D43	CYW43012TC0KFFBH
CYW9P62S1-43012EVB-01	CY8C6247FDI-D52	CYW43012TC0EKUBG
CYW9P62S1-43438EVB-01	CY8C6247BZI-D54	CYW43438KUBG
PSOC6-GENERIC	CY8C6347BZI-BLD53	<none>
XMC BSPs		

The CY8CKIT-064B0S2-4343W PSoC 64 Secure
design and debug of PSoC 64 MCUs. It comes w
industry-leading CapSense for touch buttons ar
interface, 512-Mb Quad-SPI NOR flash, 4-MBit

Kit Features:

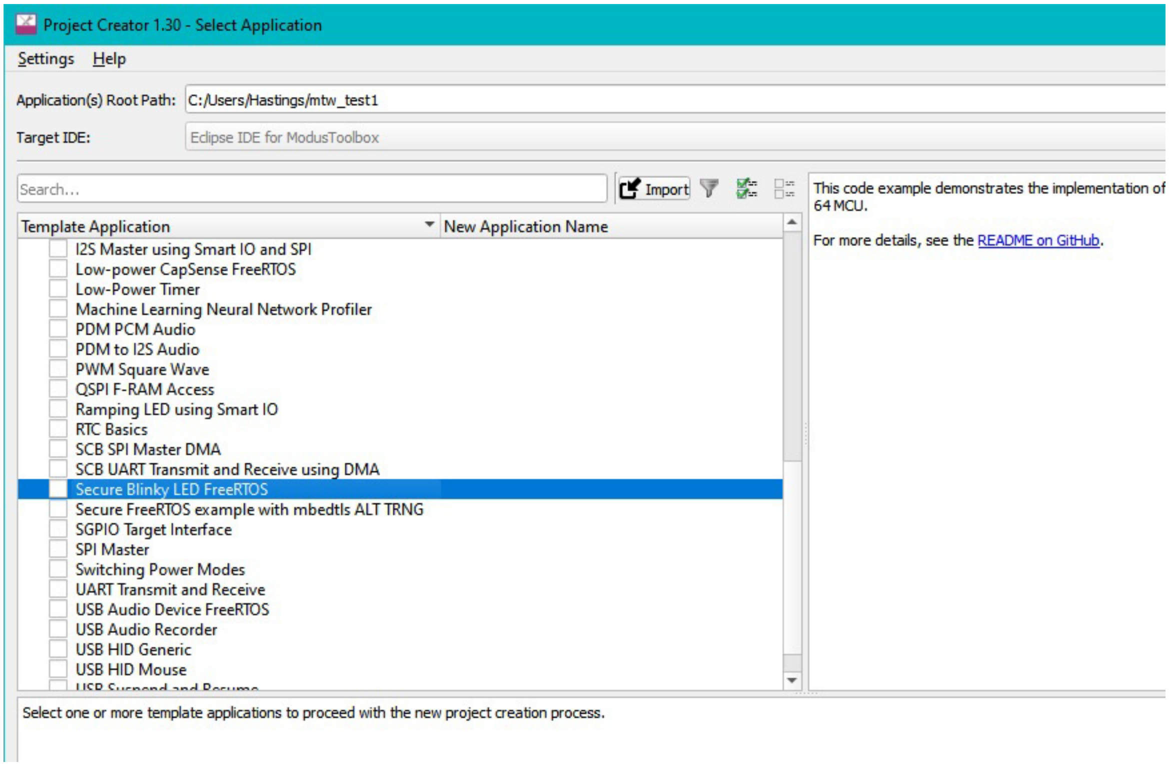
- Support of up to 2MB Flash and 1MB SR
- Dedicated SDHC to interface with WICE
- Delivers dual-cores, with a 150-MHz Ar Cortex-M0+ as the secondary process
- Supports Full-Speed USB, capacitive-se SPI interface, 13 serial communication b blocks.

Kit Contents:

- PSoC 64 Wi-Fi BT Pioneer Board
- USB Type-A to Micro-B cable
- Quick Start Guide
- Four jumper wires (4 inches each)
- Two jumper wires (5 inches each)

3. Select the *“Secure Blinky LED FreeRTOS”* example and click **Create** to create the application. Once creation is done, click **Close** to close the window and import the project into the Eclipse IDE.

Figure 4-5. “Secure Blinky LED FreeRTOS” Example Selection



Project Creator 1.30 - Select Application

Application(s) Root Path: C:/Users/Hastings/mtw_test1

Target IDE: Eclipse IDE for ModusToolbox

This code example demonstrates the implementation of
64 MCU.

For more details, see the README on GitHub.

Template Application	New Application Name
□ I2S Master using Smart IO and SPI	
□ Low-power CapSense FreeRTOS	
□ Low-Power Timer	
□ Machine Learning Neural Network Profiler	
□ PDM PCM Audio	
□ PDM to I2S Audio	
□ PWM Square Wave	
□ QSPI F-RAM Access	
□ Ramping LED using Smart IO	
□ RTC Basics	
□ SCB SPI Master DMA	
□ SCB UART Transmit and Receive using DMA	
□ Secure Blinky LED FreeRTOS	
□ Secure FreeRTOS example with mbedtls ALT TRNG	
□ SGPIO Target Interface	
□ SPI Master	
□ Switching Power Modes	
□ UART Transmit and Receive	
□ USB Audio Device FreeRTOS	
□ USB Audio Recorder	
□ USB HID Generic	
□ USB HID Mouse	
□ USB Suspend and Resume	

Select one or more template applications to proceed with the new project creation process.

4.3 Provision the Device

1. **Navigate to your ModusToolbox application directory folder in a command-line program:**

   For Windows users, use the command line program “modus-shell” instead of a standard Windows command line application. Modus shell provides access to all ModusToolbox tools including “CySecureTools” that is used to provision a device. You can access modus-shell by typing “modus-shell” in the Windows search box in the Windows menu.

   For example, in Windows if you created your ModusToolbox application in “`C:/ModusWorkplaces/mtw/Secure_Blinky_LED_FreeRTOS`" then navigate to the folder.

2. **Setup your project workspace for “CySecureTools”.**

What does this step do?

“CySecureTools” provides a default policy which can be used to quickly setup the chip with a set of development parameters like leaving the CM4 DAP (Debug Access Port) open to reprogram the chip. Based on the selected target, this step sets up all the necessary files in your workspace that are used for subsequent steps.

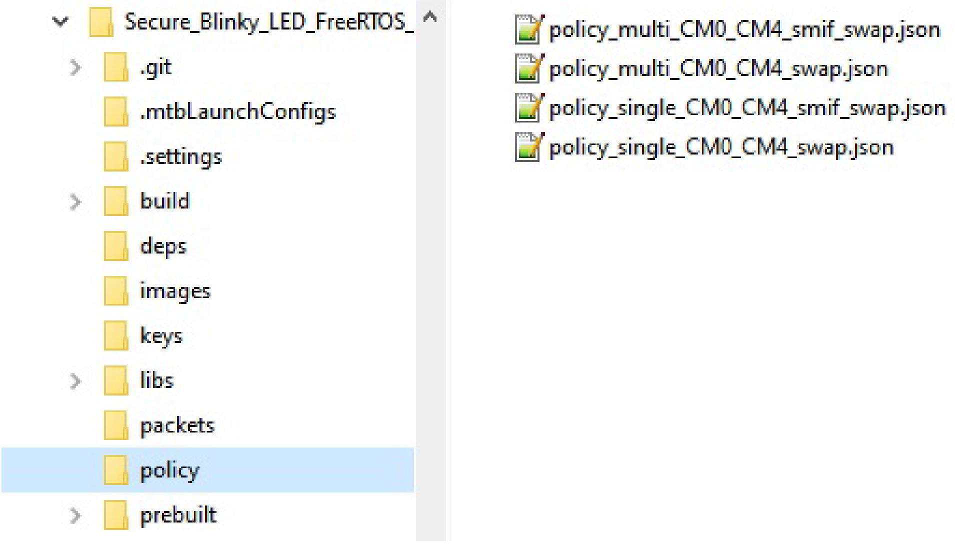
The `--target` option for “CySecureTools” functions can either be the family name “cyb06xxa” or the kit name “cy8ckit-064b0s2-4343w”.

Run the following command:

```
    cysecuretools --target cyb06xxa init
```

After running this step, the project workspace will have an added 'policy' directory with multiple template policies in it. This step also creates the keys, packets, and prebuilt directories. For more information on the policies, please refer to the “Secure Boot” SDK user guide.

Figure 4-6. Project Workspace having a new ‘policy’ directory


Secure_Blinky_LED_FreeRTOS_
.git
.mtbLaunchConfigs
.settings
build
deps
images
keys
libs
packets
policy
prebuilt
policy_multi_CM0_CM4_smif_swap.json
policy_multi_CM0_CM4_swap.json
policy_single_CM0_CM4_smif_swap.json
policy_single_CM0_CM4_swap.json

The four template policy files shown for this example are:

policy_multi_CM0_CM4_smif_swap.json - Multiple application images, serial memory interface enabled, SWAP update mode.

policy_multi_CM0_CM4_swap.json - Multiple application images, serial interface disabled, SWAP update mode.

policy_single_CM0_CM4_smif_swap.json - Single application image, serial interface enabled, SWAP update mode.

policy_single_CM0_CM4_swap.json - Single application image, serial interface enabled, SWAP update mode. (Default)

3. Create new keys.

What does this step do?

“CySecureTools” reads the provided policy and generates the keys defined.

Depending on the policy chosen, there can be multiple keys generated under the “keys” folder. By default only one key, the USERAPP_CM4_KEY, a P-256 Elliptic curve key-pair is generated.

“CySecureTools” generates keys in two formats, PEM and JSON. Both the PEM and JSON files represent the same key.

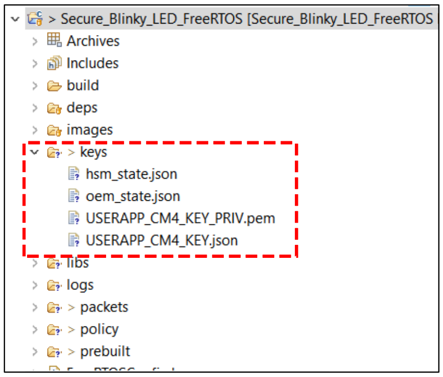
For a full description of all fields, please refer to the “Secure Boot” SDK user guide.

Run the following command:

```
cysecuretools --target cyb06xxa --policy ./policy/
policy_single_CM0_CM4_swap.json create-keys
```

After running this step, the ‘keys’ directory will have public/private key pairs and symmetric key pairs (if encrypt is enabled) generated according to the policy. These keys are used either to sign (or) encrypt images built by ModusToolbox software.

Figure 4-7. Project Workspace having a new ‘keys’ directory



```
> Secure_Blinky_LED_FreeRTOS [Secure_Blinky_LED_FreeRTOS
  Archives
  Includes
  build
  deps
  images
  > keys
      hsm_state.json
      oem_state.json
      USERAPP_CM4_KEY_PRIV.pem
      USERAPP_CM4_KEY.json
  libs
  logs
  > packets
  > policy
  > prebuilt
```

4. **Connect your CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit to PC using the provided USB cable through the KitProg3 USB connector.**

   ATTENTION: Remove jumper shunt from J26 to change VCC_3V3 voltage to 2.5 V and make sure jumper shunt on J14 is placed in VCC_3V3 position (between pin 2 and 3) before plugging in the kit to the PC. The 2.5 V supply is necessary for Step 5, where PSoC 64 eFuses are blown. KitProg3 must be in DAPLink mode for provisioning. The Status LED (LED2) will be ramping ON/OFF (~2Hz) in this mode. Press and release the Mode button (SW3) one or more times until the KitProg3 is in DAPLink mode.

Table 4-1. Voltage Configuration Details

PSoC 64 VTARG	J14 Position	J26
1.8 V	VCC_1V8	X
2.5 V	VCC_3V3	Not Loaded
3.3 V	VCC_3V3	Loaded

5. **Perform provisioning.**

   ATTENTION: PSoC 64 supply voltage of 2.5 V is required to perform provisioning. The 2.5 V requirement is because this step involves blowing eFuses to change the device lifecycle to “SECURE CLAIMED”. If the chip is not at 2.5 V, it may cause provisioning to fail and permanently lock the chip in a dead state.

What does this step do?

The “CySecureTools” provision-device API does the following steps:

- Reads the provided policy and forms the final provisioning packet, named prov_cmd.jwt
- Performs the entrance exam
- Provisions the device by sending the prov_cmd.jwt to the PSoC 64 “Secure Boot” MCU.

Before running this step, you can modify the default policy to match your end use-case. For most development use-cases, you don’t need to modify it.

The Entrance exam is a test routine that does the following things:

- Verify that the Device is in the correct lifecycle stage
- Verify that Boot Code has not been modified/tampered
- Verify that User flash is empty and no code is running before any provisioning takes place

Failing the entrance exam returns an error in the command line. If there is any firmware running on the device, existing firmware can be erased using the tools like Cypress Programmer.

Run the below command:

```
cysecuretools --target cyb06xxa --policy ./policy/
policy_single_CM0_CM4_swap.json provision-device
```

KitProg3 driver issues:

There can be sporadic issue with KitProg3 and drivers can prevent the kit from being recognized or cause other failures during the provisioning flow. Please see the Chapter 7 of the KitProg3 User Guide for information on how to resolve this.

Note: The provisioning procedure may take several seconds. Do not disconnect or reset the device during this procedure.

Note: The 2.5 V voltage option is provided only for the purpose of provisioning. For normal operation of the kit after provisioning, please load the jumper shunt on J26.

Note: The entrance exam can be run separately without provisioning to verify the lifecycle stage of a device by using the following command:

```
cysecuretools --target cyb06xxa --policy ./policy/
policy_single_CM0_CM4_swap.json entrance-exam
```

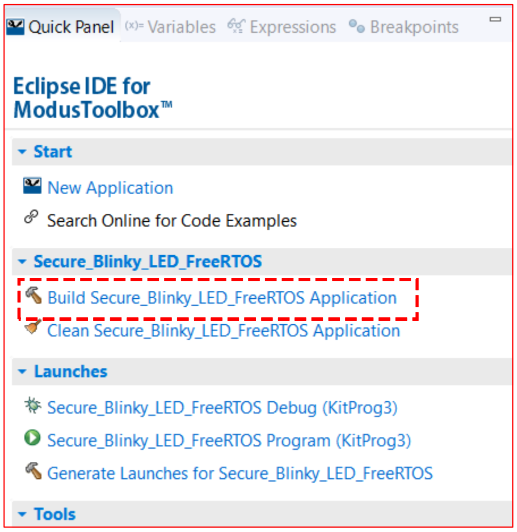
Note: In case you have a kit which has already been provisioned in the past with different credentials, you can re-provision the chip with new keys & policies by running the below command. Re-provisioning is possible if the policy provisioned into the chip allows this, by default all policies provided in “CySecureTools” allow re-provisioning. Note that the Entrance Exam is not run during a re-provisioning process.

```
cysecuretools --target cyb06xxa --policy ./policy/
policy_single_CM0_CM4_swap.json re-provision-device
```

4.4 Build and Program the Example Project

1. From the Eclipse IDE, click on the “Build Secure_Blinky_LED_FreeRTOS Application” link in the Quick Panel.

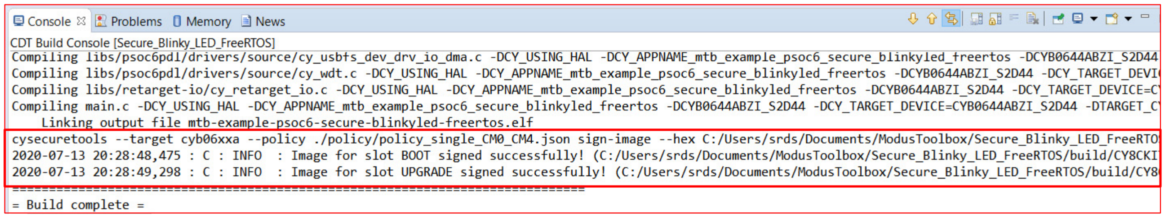
Figure 4-8. Build Secure_Blinky_LED_FreeRTOS Application Selection


Quick Panel
Variables
Expressions
Breakpoints
Eclipse IDE for ModusToolbox™
Start
New Application
Search Online for Code Examples
Secure_Blinky_LED_FreeRTOS
Build Secure_Blinky_LED_FreeRTOS Application
Clean Secure_Blinky_LED_FreeRTOS Application
Launches
Secure_Blinky_LED_FreeRTOS Debug (KitProg3)
Secure_Blinky_LED_FreeRTOS Program (KitProg3)
Generate Launches for Secure_Blinky_LED_FreeRTOS
Tools

Note: Ensure that you have clicked the project in the explorer otherwise the option will not be visible.

2. Once build is complete, you will see in the ModusToolbox console that “CySecureTools” has signed the image with the keys you generated in the previous section. Note that this message is before the memory consumption table in the build console window.

Figure 4-9. ModusToolbox Console after Build is Complete



```
CDT Build Console [Secure_Blinky_LED_FreeRTOS]
Compiling libs/psoc6pdl/drivers/source/cy_usbfs_dev_drv_io_dma.c -DCY_USING_HAL -DCY_APPNAME_mtb_example_psoc6_secure_blinkyled_freertos -DCYB0644ABZI_S2D44
Compiling libs/psoc6pdl/drivers/source/cy_wdt.c -DCY_USING_HAL -DCY_APPNAME_mtb_example_psoc6_secure_blinkyled_freertos -DCYB0644ABZI_S2D44 -DCY_TARGET_DEVI
Compiling libs/retarget-io/cy_retarget_io.c -DCY_USING_HAL -DCY_APPNAME_mtb_example_psoc6_secure_blinkyled_freertos -DCYB0644ABZI_S2D44 -DCY_TARGET_DEVICE=C
Compiling main.c -DCY_USING_HAL -DCY_APPNAME_mtb_example_psoc6_secure_blinkyled_freertos -DCYB0644ABZI_S2D44 -DCY_TARGET_DEVICE=CYB0644ABZI_S2D44 -DTARGET_C
    Linking output file mtb-example-psoc6-secure-blinkyled-freertos.elf
cysecuretools --target cyb06xxa --policy ./policy/policy_single_CM0_CM4.json sign-image --hex C:/Users/srds/Documents/ModusToolbox/Secure_Blinky_LED_FreeRTO
2020-07-13 20:28:48,475 : C : INFO  : Image for slot BOOT signed successfully! (C:/Users/srds/Documents/ModusToolbox/Secure_Blinky_LED_FreeRTOS/build/CY8CKI
2020-07-13 20:28:49,298 : C : INFO  : Image for slot UPGRADE signed successfully! (C:/Users/srds/Documents/ModusToolbox/Secure_Blinky_LED_FreeRTOS/build/CY8
==============================================================================
= Build complete =
```

Note:

When using the CY8CKIT_064B0S2_4343W target, the ModusToolbox BSP has a post-build image signing command located in:

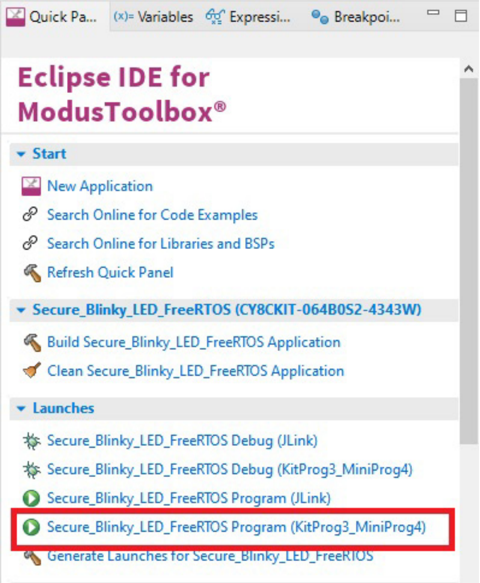
“<ModusToobox project>/libs/TARGET_CY8CKIT-064B0S2-4343W/CY8CKIT-064B0S2-4343W.mk”

This post-build script uses the policy file to find the private key path used to sign the application. If a different/invalid key is used then the “Secure Boot” process will fail.

3. The KitProg3 will still be in DAPLink mode from the provisioning step. It will need to be changed to CMSIS-DAP Bulk mode to program or debug the project. To switch to CMSIS-DAP Bulk mode, press and release the mode switch (SW3) on the kit until the status LED (LED2) is constantly on (not ramping). The kit will stay in this mode even after a power cycle, until the mode button is pressed again.

   To program the kit with the example application, click on the “Secure_Blinky_LED_FreeRTOS Program (KitProg3_MiniProg4)” in the Quick Panel. The status of the programming sequence will be displayed in the Console window.

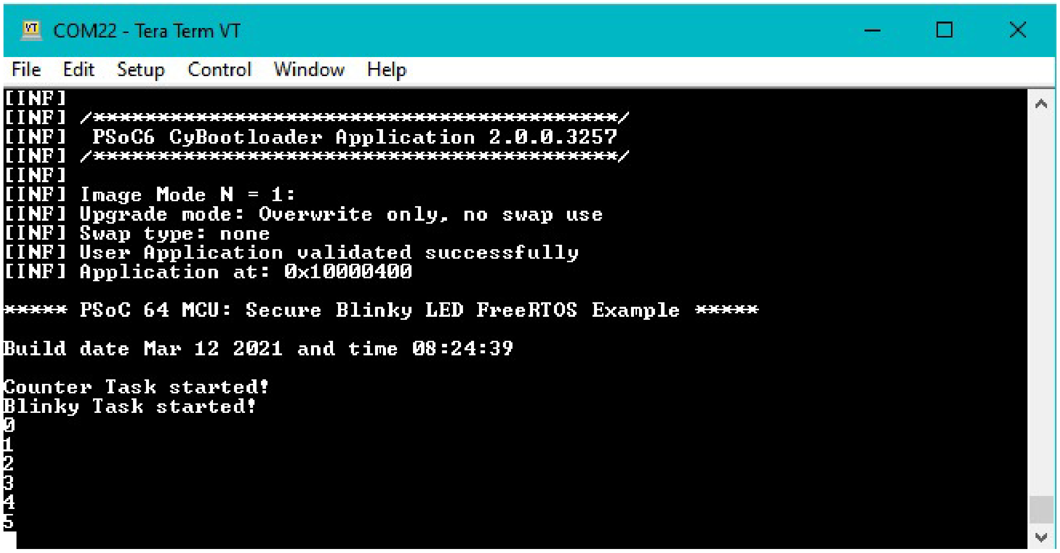
Figure 4-10. Secure_Blinky_LED_FreeRTOS Program (KitProg3_MiniProg4) Launch


Quick Pa...
(x)= Variables
Expressi...
Breakpoi...
Eclipse IDE for ModusToolbox®
Start
New Application
Search Online for Code Examples
Search Online for Libraries and BSPs
Refresh Quick Panel
Secure_Blinky_LED_FreeRTOS (CY8CKIT-064B0S2-4343W)
Build Secure_Blinky_LED_FreeRTOS Application
Clean Secure_Blinky_LED_FreeRTOS Application
Launches
Secure_Blinky_LED_FreeRTOS Debug (JLink)
Secure_Blinky_LED_FreeRTOS Debug (KitProg3_MiniProg4)
Secure_Blinky_LED_FreeRTOS Program (JLink)
Secure_Blinky_LED_FreeRTOS Program (KitProg3_MiniProg4)
Generate Launches for Secure_Blinky_LED_FreeRTOS

4. Observe LED8 (Orange) blinking every second. Open a serial terminal with a baud rate of 115200 to see the message as shown below.

   Note: if you open the terminal after programming the kit, just press/release the reset button to see the message on the terminal window.

Figure 4-11. UART Output



```
[INF]
[INF] /*********************************************/
[INF]  PSoC6 CyBootloader Application 2.0.0.3257
[INF] /*********************************************/
[INF]
[INF] Image Mode N = 1:
[INF] Upgrade mode: Overwrite only, no swap use
[INF] Swap type: none
[INF] User Application validated successfully
[INF] Application at: 0x10000400

***** PSoC 64 MCU: Secure Blinky LED FreeRTOS Example *****

Build date Mar 12 2021 and time 08:24:39

Counter Task started!
Blinky Task started!
0
1
2
3
4
5
```

4.5 Additional Code Examples

Additional code examples for PSoC 64 device can be found on the Cypress git repository

https://github.com/cypresssemiconductorco

Note that most PSoC 6 MCU code examples will run on PSoC 64 devices. If you wish to run other code examples on the existing kit, you can follow the same steps outlined in Section 4.2 to 4.4 with the alternate project imported into your workspace. There are two alternative flows for this,

1. If you want to re-use existing keys and policies provisioned into the kit, please copy the ‘keys’ and ‘policy’ folder from your existing project workspace into your new project workspace. In this flow, you will re-use the existing key and policy file to sign firmware and program.
2. If you want to create new keys and policies, please follow the re-provisioning note provided in Section 4.3, step 5.

5. Hardware

5.1 Schematics

Refer to the schematic files available on the kit webpage.

5.2 Hardware Functional Description

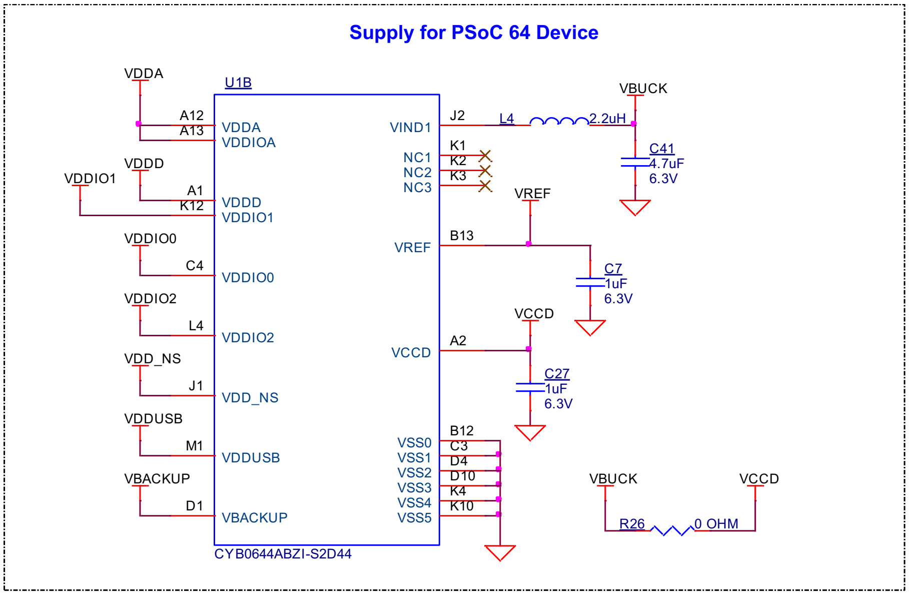
This section explains in detail the individual hardware blocks.

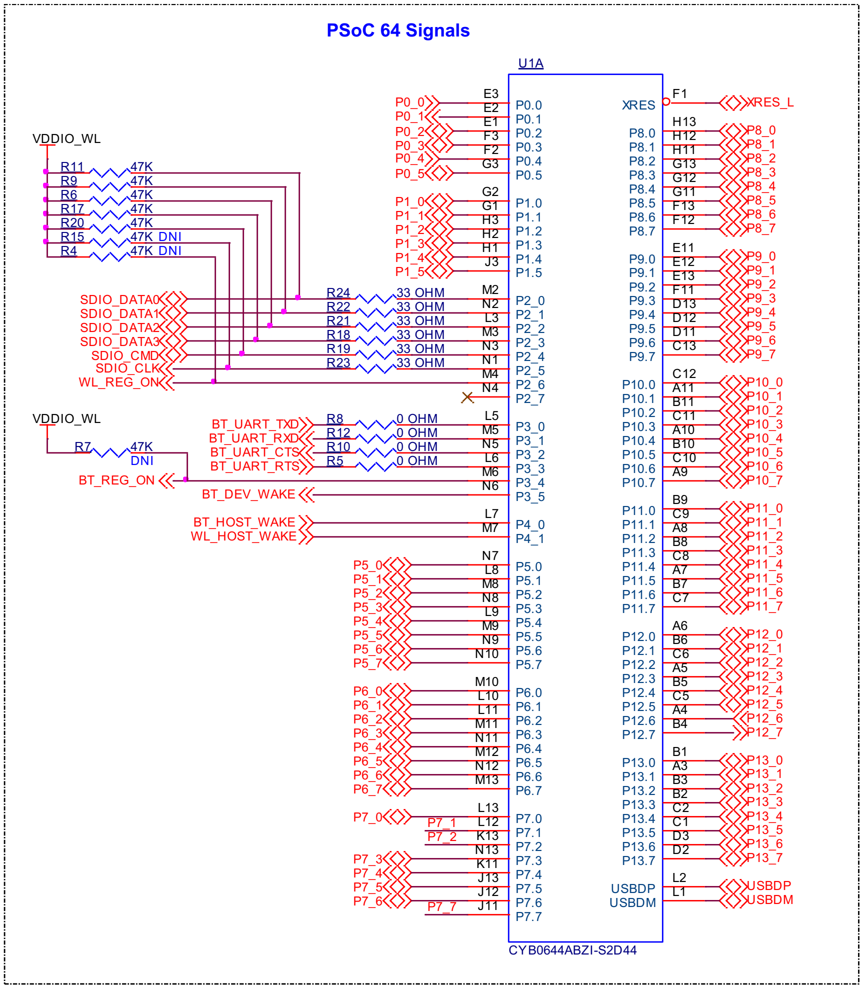
5.2.1 CY8CMOD-064B0S2-4343W (MOD1)

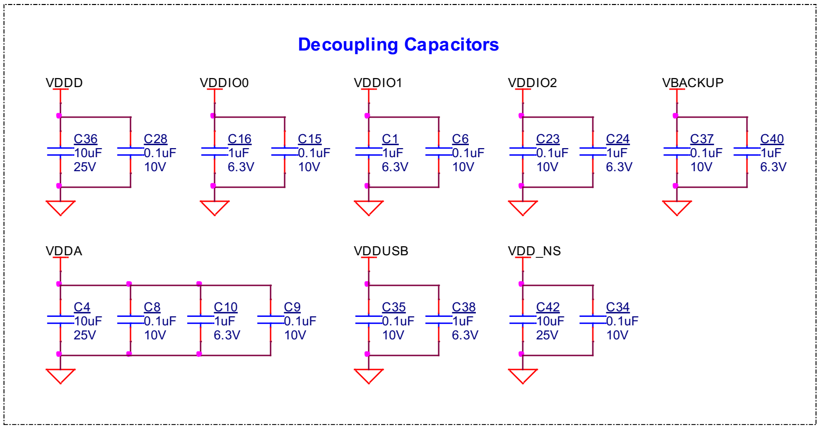
CY8CMOD-064B0S2-4343W is a castellated PCB module which consists mainly of PSoC 64 chip and CYW4343W devices. The module also houses a 2.45 GHz/5.5 GHz dual-band chip antenna, RF switch for antenna diversity, Low Power Oscillator (LPO) for CYW4343W, crystal oscillators for PSoC 64, modulation and integration capacitors to support CapSense and other passive components required for the proper working of PSoC 64 and CYW4343W. A pre-certified Type 1DX module with CYW4343W from Murata, LBEE5KL1DX, is used for ease of development. CYW4343W supports only the 2.45 GHz band, but the antenna used is a 2450AD14A5500 Dual Band 2.45 GHz/ 5.5 GHz Mini Chip Antenna from Johanson, to use the same antenna across different designs. The castellated PCB module has 137 castellated pads, which are used for different voltage rails and I/O signals of the PSoC 64 device and the CYW4343W.

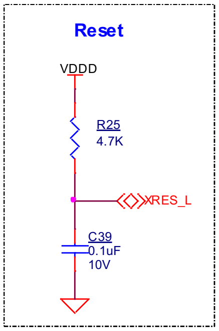
For more information, see the PSoC 64 webpage, Murata Type 1DX webpage and the datasheet.

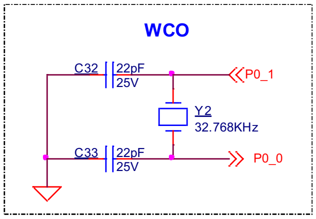
Figure 5-1. Schematics of CY8CMOD-064B0S2-4343W

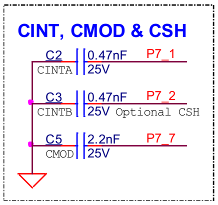


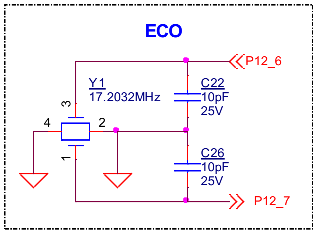

PSoC 64 Signals
U1A
CYB0644ABZI-S2D44
VDDIO_WL
R11 47K
R9 47K
R6 47K
R17 47K
R20 47K
R15 47K DNI
R4 47K DNI
SDIO_DATA0
SDIO_DATA1
SDIO_DATA2
SDIO_DATA3
SDIO_CMD
SDIO_CLK
WL_REG_ON
R24 33 OHM
R22 33 OHM
R21 33 OHM
R18 33 OHM
R19 33 OHM
R23 33 OHM
VDDIO_WL
R7 47K DNI
BT_REG_ON
BT_UART_TXD
BT_UART_RXD
BT_UART_CTS
BT_UART_RTS
R8 0 OHM
R12 0 OHM
R10 0 OHM
R5 0 OHM
BT_DEV_WAKE
BT_HOST_WAKE
WL_HOST_WAKE
P0_0 E3 P0.0
P0_1 E2 P0.1
P0_2 E1 P0.2
P0_3 F3 P0.3
P0_4 F2 P0.4
P0_5 G3 P0.5
P1_0 G2 P1.0
P1_1 G1 P1.1
P1_2 H3 P1.2
P1_3 H2 P1.3
P1_4 H1 P1.4
P1_5 J3 P1.5
M2 P2_0
N2 P2_1
L3 P2_2
M3 P2_3
N3 P2_4
N1 P2_5
M4 P2_6
N4 P2_7
L5 P3_0
M5 P3_1
N5 P3_2
L6 P3_3
M6 P3_4
N6 P3_5
L7 P4_0
M7 P4_1
P5_0 N7 P5.0
P5_1 L8 P5.1
P5_2 M8 P5.2
P5_3 N8 P5.3
P5_4 L9 P5.4
P5_5 M9 P5.5
P5_6 N9 P5.6
P5_7 N10 P5.7
P6_0 M10 P6.0
P6_1 L10 P6.1
P6_2 L11 P6.2
P6_3 M11 P6.3
P6_4 N11 P6.4
P6_5 M12 P6.5
P6_6 N12 P6.6
P6_7 M13 P6.7
P7_0 L13 P7.0
P7_1 L12 P7.1
P7_2 K13 P7.2
P7_3 N13 P7.3
P7_4 K11 P7.4
P7_5 J13 P7.5
P7_6 J12 P7.6
P7_7 J11 P7.7
XRES F1 XRES_L
P8.0 H13 P8_0
P8.1 H12 P8_1
P8.2 H11 P8_2
P8.3 G13 P8_3
P8.4 G12 P8_4
P8.5 G11 P8_5
P8.6 F13 P8_6
P8.7 F12 P8_7
P9.0 E11 P9_0
P9.1 E12 P9_1
P9.2 E13 P9_2
P9.3 F11 P9_3
P9.4 D13 P9_4
P9.5 D12 P9_5
P9.6 D11 P9_6
P9.7 C13 P9_7
P10.0 C12 P10_0
P10.1 A11 P10_1
P10.2 B11 P10_2
P10.3 C11 P10_3
P10.4 A10 P10_4
P10.5 B10 P10_5
P10.6 C10 P10_6
P10.7 A9 P10_7
P11.0 B9 P11_0
P11.1 C9 P11_1
P11.2 A8 P11_2
P11.3 B8 P11_3
P11.4 C8 P11_4
P11.5 A7 P11_5
P11.6 B7 P11_6
P11.7 C7 P11_7
P12.0 A6 P12_0
P12.1 B6 P12_1
P12.2 C6 P12_2
P12.3 A5 P12_3
P12.4 B5 P12_4
P12.5 C5 P12_5
P12.6 A4 P12_6
P12.7 B4 P12_7
P13.0 B1 P13_0
P13.1 A3 P13_1
P13.2 B3 P13_2
P13.3 B2 P13_3
P13.4 C2 P13_4
P13.5 C1 P13_5
P13.6 D3 P13_6
P13.7 D2 P13_7
USBDP L2 USBDP
USBDM L1 USBDM

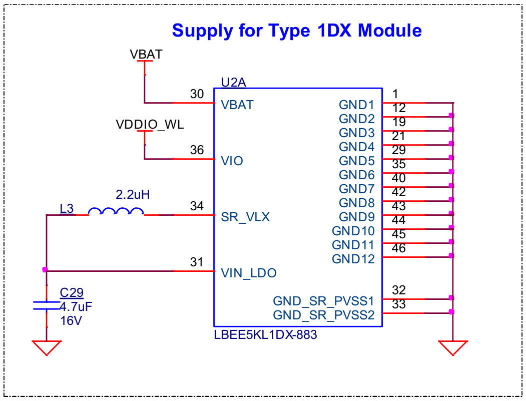

Decoupling Capacitors
VDDD
C36 10uF 25V
C28 0.1uF 10V
VDDIO0
C16 1uF 6.3V
C15 0.1uF 10V
VDDIO1
C1 1uF 6.3V
C6 0.1uF 10V
VDDIO2
C23 0.1uF 10V
C24 1uF 6.3V
VBACKUP
C37 0.1uF 10V
C40 1uF 6.3V
VDDA
C4 10uF 25V
C8 0.1uF 10V
C10 1uF 6.3V
C9 0.1uF 10V
VDDUSB
C35 0.1uF 10V
C38 1uF 6.3V
VDD_NS
C42 10uF 25V
C34 0.1uF 10V

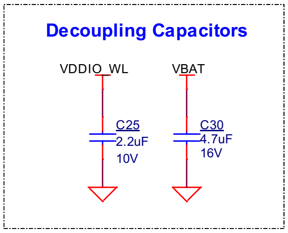

Reset
VDDD
R25
4.7K
XRES_L
C39
0.1uF
10V


WCO
C32
22pF
25V
P0_1
Y2
32.768KHz
C33
22pF
25V
P0_0

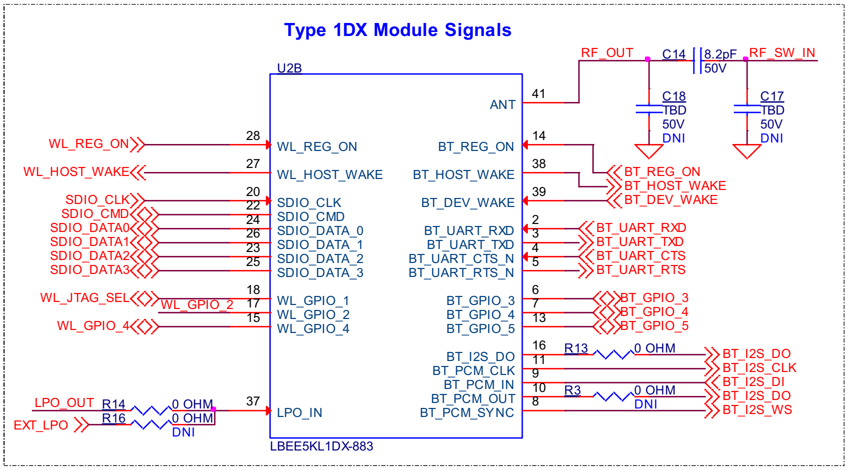

CINT, CMOD & CSH
C2
CINTA
0.47nF
25V
P7_1
C3
CINTB
0.47nF
25V
Optional CSH
P7_2
C5
CMOD
2.2nF
25V
P7_7


ECO
P12_6
Y1
17.2032MHz
3
4
2
1
C22
10pF
25V
C26
10pF
25V
P12_7

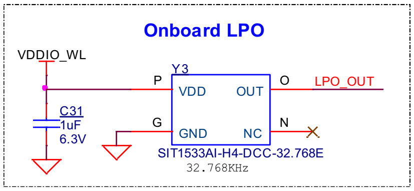

Supply for Type 1DX Module
VBAT
U2A
30
VBAT
VDDIO_WL
36
VIO
L3
2.2uH
34
SR_VLX
31
VIN_LDO
C29
4.7uF
16V
GND1
GND2
GND3
GND4
GND5
GND6
GND7
GND8
GND9
GND10
GND11
GND12
1
12
19
21
29
35
40
42
43
44
45
46
GND_SR_PVSS1
GND_SR_PVSS2
32
33
LBEE5KL1DX-883



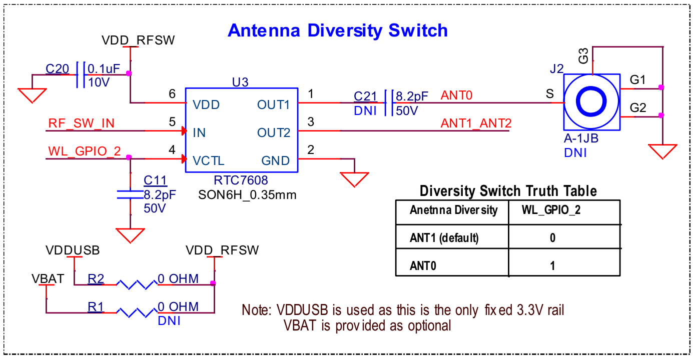
Decoupling Capacitors



Type 1DX Module Signals



Onboard LPO

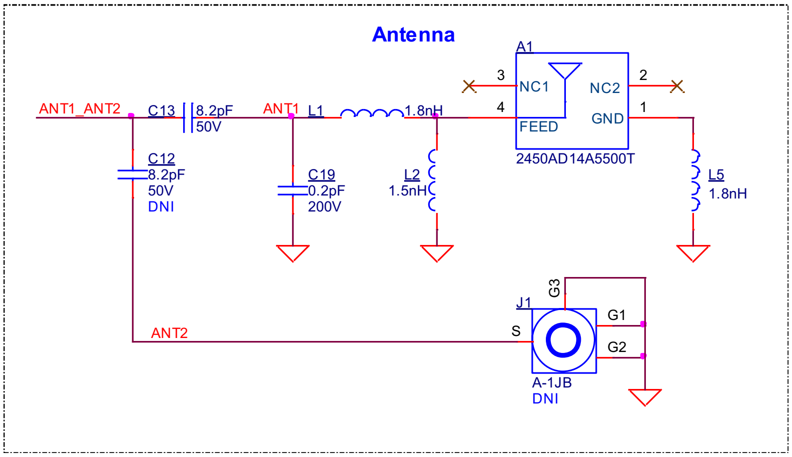


Antenna Diversity Switch

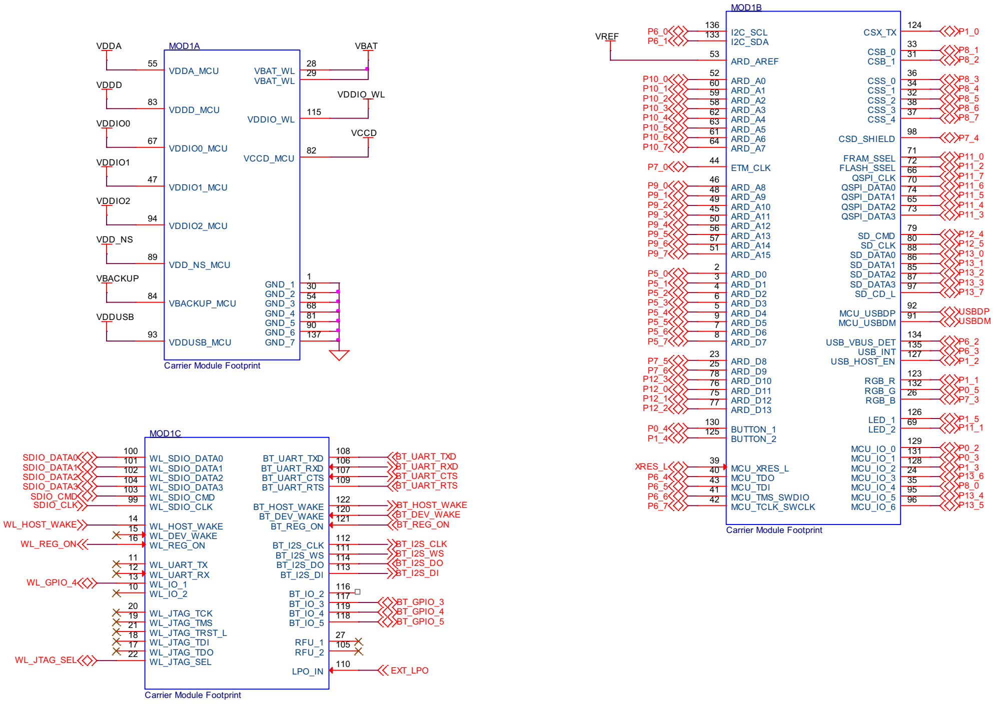
Diversity Switch Truth Table

Anetnna Diversity	WL_GPIO_2
ANT1 (default)	0
ANT0	1

Note: VDDUSB is used as this is the only fixed 3.3V rail
VBAT is provided as optional



Carrier Module Footprint

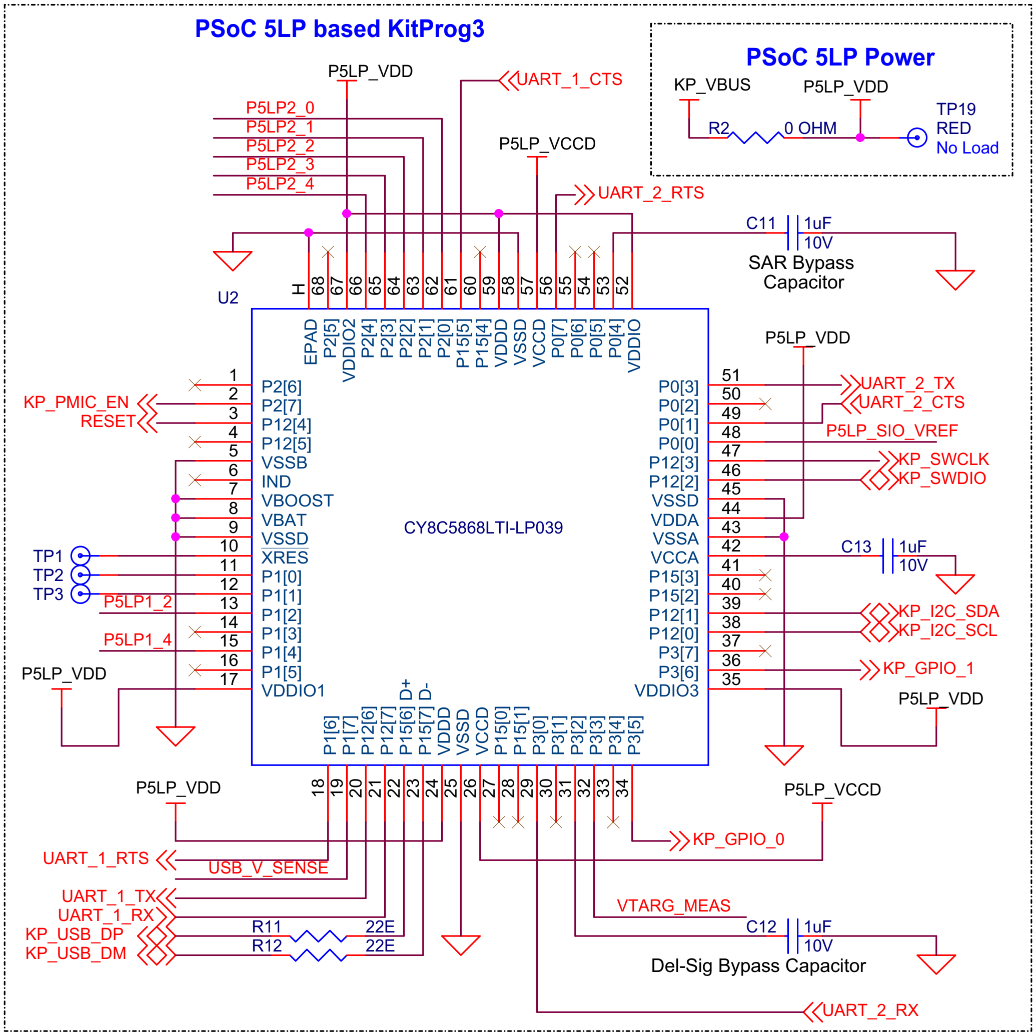


5.2.2 PSoC 5LP-based KitProg3 (U2)

An onboard PSoC 5LP (CY8C5868LTI-LP039) device is used as KitProg3 to program and debug the PSoC 64 device. The PSoC 5LP device connects to the USB port of a PC through a USB connector and to the SWD and other communication interfaces of the PSoC 64.

The PSoC 5LP device is a true system-level solution providing MCU, memory, analog, and digital peripheral functions in a single chip. For more information, visit the PSoC 5LP web page. Also, see the CY8C58LPxx Family datasheet.

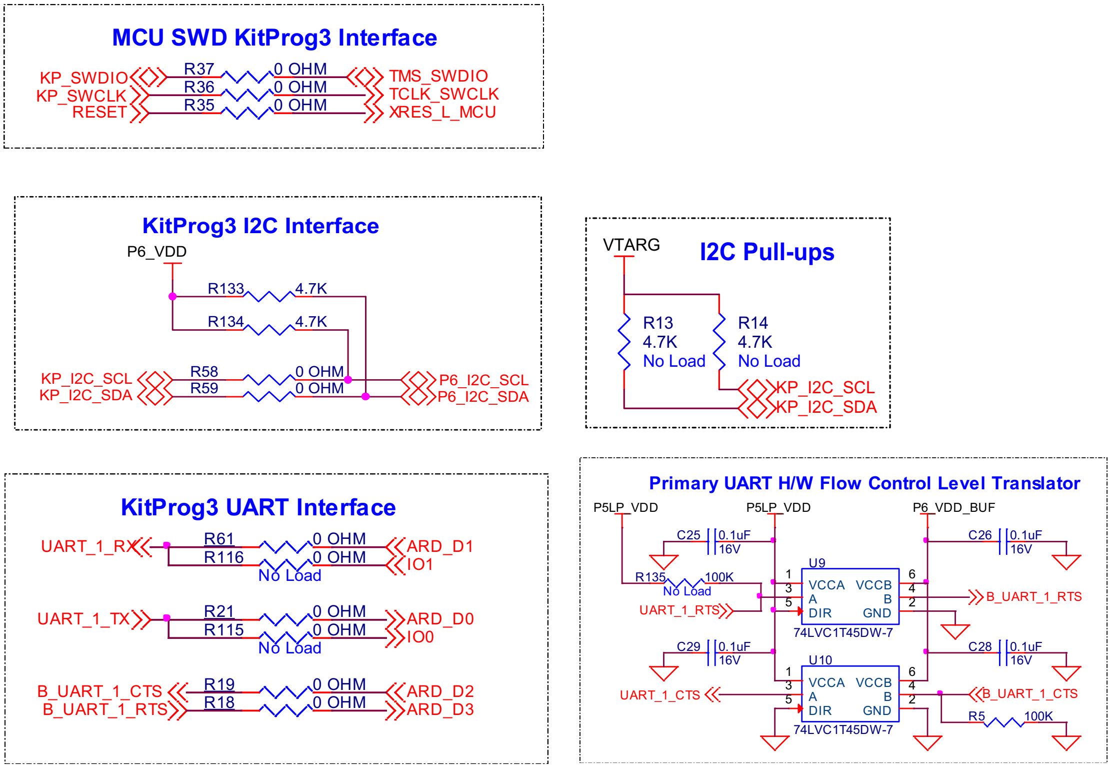
Figure 5-2. Schematics of PSoC 5LP based KitProg3



5.2.3 Serial Interconnection between PSoC 5LP and PSoC 64 Device

In addition to the use as an onboard programmer, the PSoC 5LP device functions as an interface for the USB-UART and USB-I2C bridges, as shown in Figure 5-3. The USB-Serial pins of the PSoC 5LP device are hard-wired to the I2C/UART pins of the PSoC 64 device. These pins are also available on the Arduino-compatible I/O headers.

Figure 5-3. Schematics of Programming and Serial Interface Connections

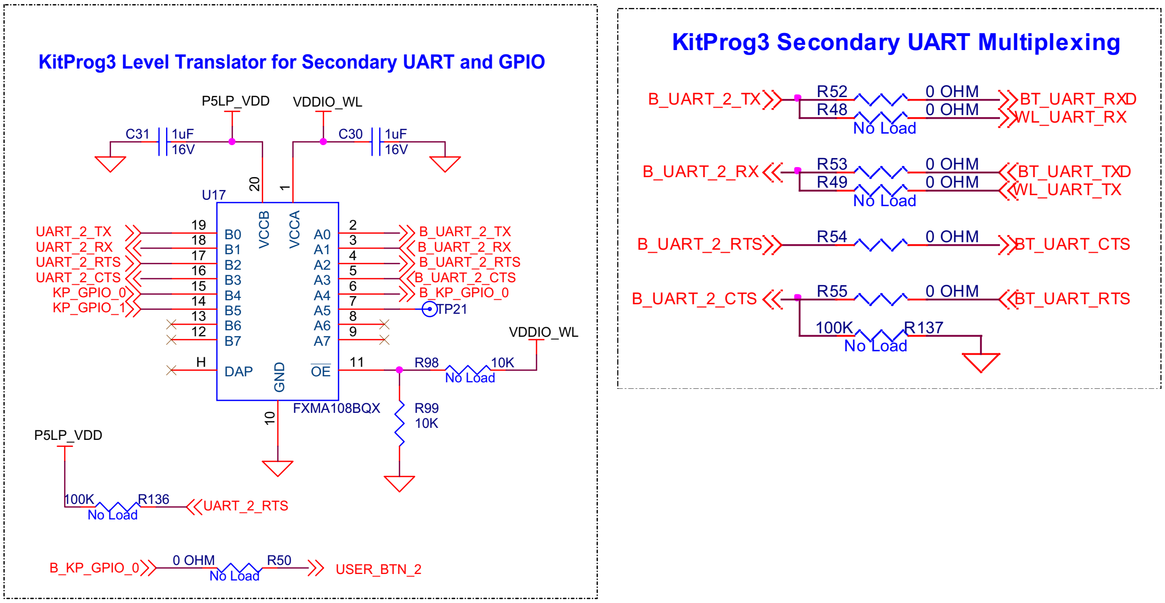


5.2.4 Serial Interconnection Between PSoC 5LP and CYW4343W

The PSoC 5LP device also has a secondary UART that is connected to the BT_UART of CYW4343W (Murata Type 1DX).

Note: BT_UART is also connected to PSoC 64 device on the carrier module and this is the communication interface between the PSoC 64 and the Bluetooth section of the CYW4343W.

Figure 5-4. Serial Interconnection Between PSoC 5LP and CYW4343W

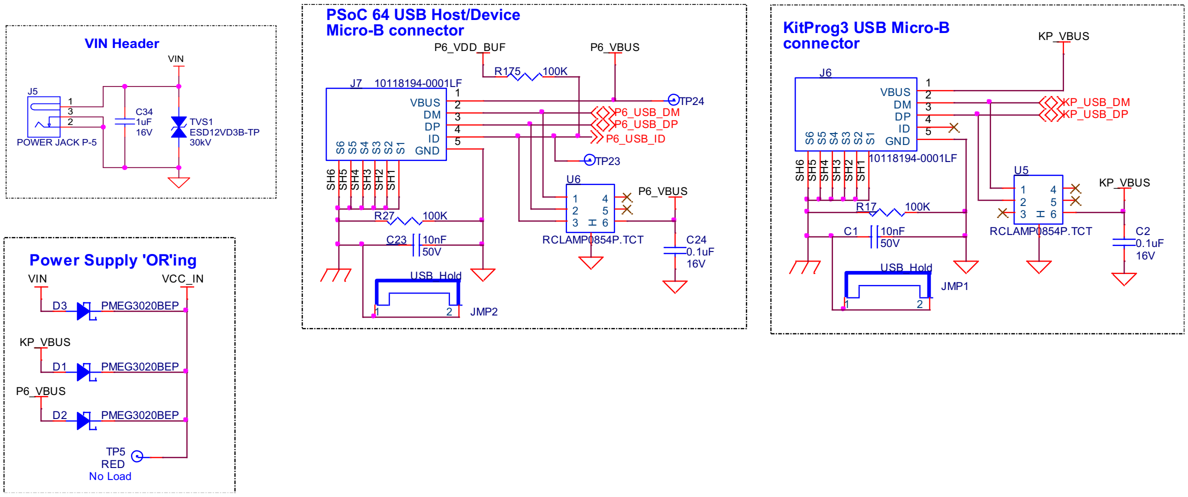


5.2.5 Power Supply System

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

- 5 V from the onboard USB Micro-B connectors (**J6** and **J7**)
- 7 V–12 V from external power supply at VIN through barrel jack (**J5**) or from Arduino header pin J1.1

Figure 5-5. Schematics of Power Supply Input and OR'ing



5.2.5.1 Voltage regulators

The power supply system is designed for the voltage configurations listed in Table 5-1. Some configurations achievable on this kit are outside the operating range for the device. However, it is not possible to achieve all applicable configurations by changing jumper positions but rather requires re-work of respective 0-ohm resistors.

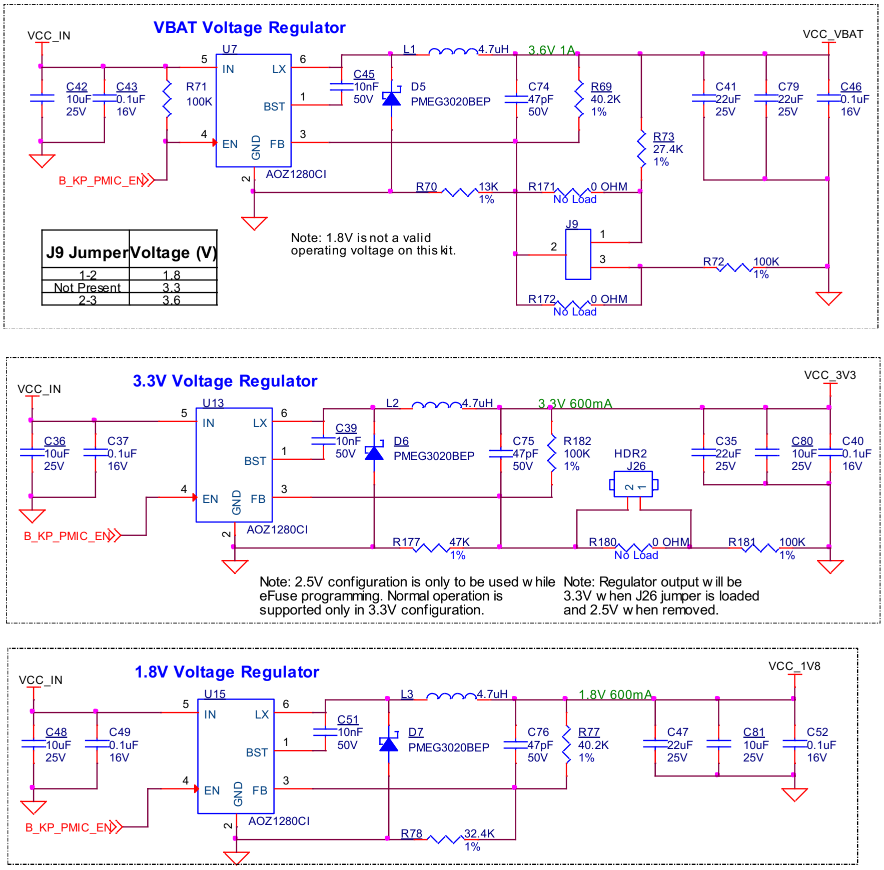
VDDIO_WL and VDDIO2_MCU must be at the same voltage since they power the SDIO interface between PSoC 64 device and CYW4343W. Hence both are supplied by the VCC_VDDIO2_IN domain.

Three buck regulators **U15**, **U13** and **U7** are used to achieve 1.8 V, 3.3 V and 3.6 V outputs respectively. Figure 5-6 shows the schematics of the voltage regulator circuits.

Table 5-1. Operating voltage ranges of domains

Voltage Domain	Carrier Module (MOD1) Power Pins powered by the domain	Operating Voltage		Voltage Configuration applicable in kit	Voltage Selection Header
		Min (V)	Max (V)		
VCC_VBAT	VBAT_WL	3.2	4.2	3.6V, 3.3V	J9
VCC_VDDIO2_IN	VDDIO2_MCU, VDDIO_WL	1.71	3.63	1.8V, 3.3V	J16
VTARG	VDDD_MCU, VDDIO1_MCU, VDDA_MCU, VDD_NS_MCU, VBACKUP_MCU	1.7	3.6	1.8, 2.5, 3.3V	J14, J26
VCC_VDDIO0	VDDIO0_MCU	1.7	3.6	1.8, 2.5, 3.3V	None (uses 0 Ohms)

Figure 5-6. Voltage Regulators



Note: 1.8V is not a valid operating voltage on this kit.

Note: 2.5V configuration is only to be used while eFuse programming. Normal operation is supported only in 3.3V configuration.

Note: Regulator output will be 3.3V when J26 jumper is loaded and 2.5V when removed.

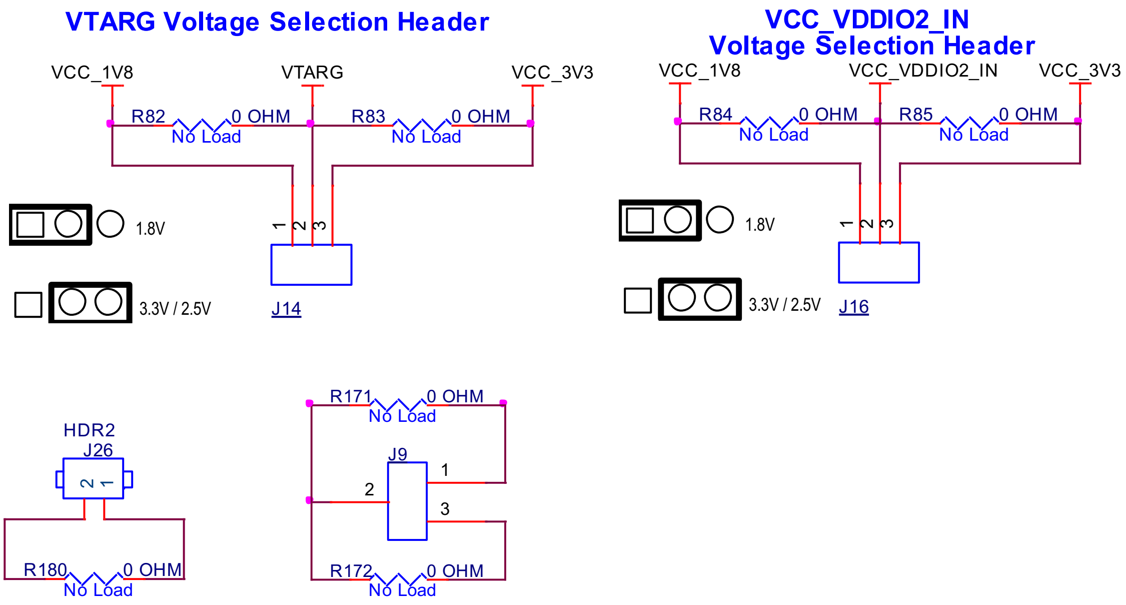
5.2.5.2 Voltage Selection

VCC_VBAT has a dedicated regulator that changes voltage by varying the feedback voltage through the resistor network at **J9**.

VTARG and VCC_VDDIO2_IN have dedicated 3-pin voltage selection headers **J14** and **J16** respectively that select between VCC_3V3 or VCC_1V8 voltages. Figure 5-7 shows the schematics of the power selection circuits.

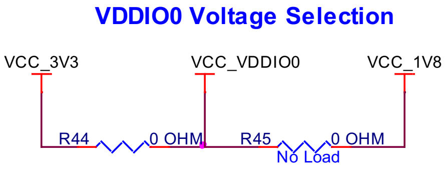
Note: Jumper shunt from **J26** should only be removed during provisioning. This kit is not designed to fully operate at 2.5 V. In order to select 2.5 V during provisioning, set **J14** to VCC_3V3 and remove **J26**. VCC_VDDIO2_IN will also be 2.5 V if **J16** is set to VCC_3V3.

Figure 5-7. Voltage Selection Headers



VCC_VDDIO0 voltage can be selected between VCC_3V3 and VCC_1V8 using zero-ohm resistors. It is connected to VCC_3V3 by default as microSD card (powered by VCC_VDDIO0) works only at 3.3V. Figure 5-8 shows the schematics of the voltage selection circuits.

Figure 5-8. Voltage Selection



5.2.5.3 *Current Measurement Headers*

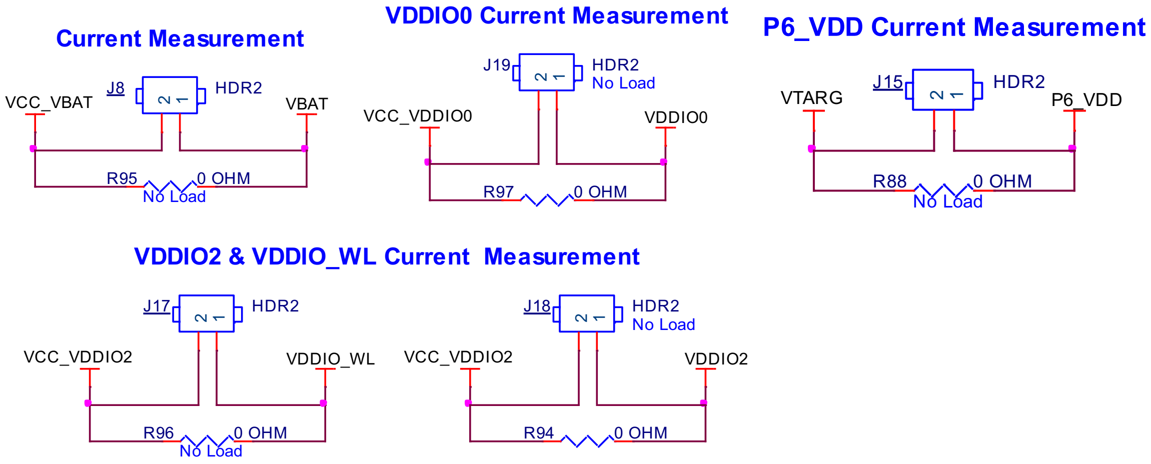
The current of the following domains have dedicated 2-pin headers to facilitate easy current measurement using an ammeter across the pins.

Note: If a header is not loaded by default, it is bypassed using a 0-ohm resistor parallel to it. Please make sure to remove the corresponding 0-ohm resistor (as per Figure 5-9) before measuring current across the header.

Table 5-2. Current Measurement Headers

Domain Name	Header Reference Designator	Loaded by default
VBAT	J8	Y
P6_VDD	J15	Y
VDDIO_WL	J17	Y
VDDIO2	J18	N
VDDIO0	J19	N

Figure 5-9. Current Measurement Headers



Note: When measuring P6_VDD current, make sure that the J25 jumper shunt is removed. This will disconnect the potentiometer from VDDA and removes the leakage caused by it.

5.2.6 I/O Headers

5.2.6.1 Arduino-compatible Headers (J1, J2, J3, J4)

The board has four Arduino-compatible headers: **J1**, **J2**, **J3**, and **J4**. You can connect 3.3 V Arduino-compatible shields to develop applications based on the shield’s hardware.

Note: 5-V shields are not supported and connecting a 5-V shield may permanently damage the board.

Note: All Arduino header pins are not connected to the same voltage reference. ARD_D[10:13] are powered by VDDIO0 whereas rest are powered by domains connected to VTARG. Hence Arduino shields particularly that use ARD_D[10:13] must not be used when VTARG is 1.8 V.

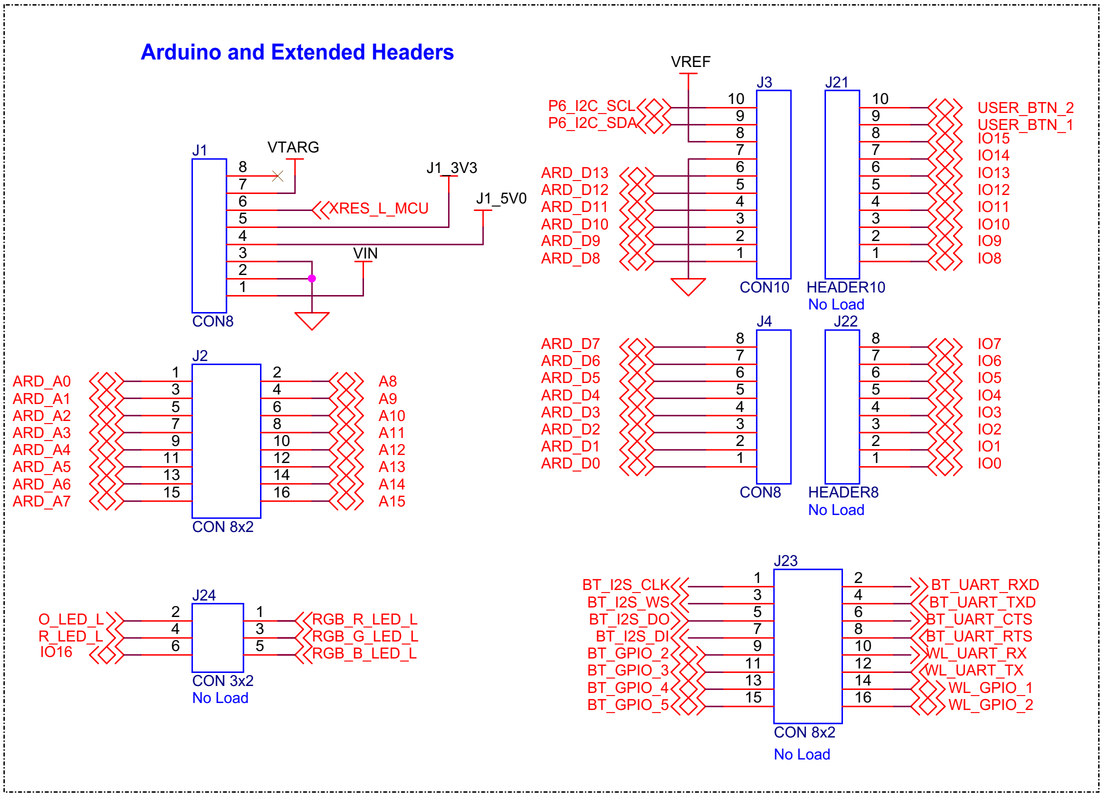
5.2.6.2 PSoC 64 I/O Headers (J21, J22, and J24)

These headers provide connectivity to PSoC 64 GPIOs that are not connected to the Arduino-compatible headers. The majority of these pins are multiplexed with onboard peripherals and are not connected to the PSoC 64 by default. They can be connected to the PSoC 64 device using 0-ohm resistors.

5.2.6.3 WL/BT I/O Headers (J23)

These headers provide connectivity to a few of the CYW4343W GPIOs that are available at the castellated pads. All these I/Os work at the VDDIO_WL voltage (1.8 V by default).

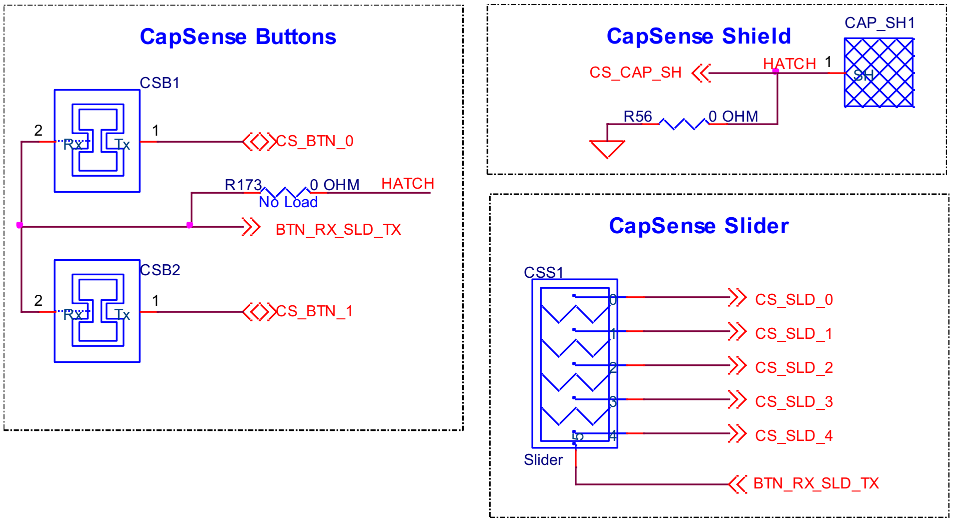
Figure 5-10. I/O Headers



5.2.7 CapSense Circuit

A CapSense slider and two buttons, all supporting both self-capacitance (CSD) and mutual-capacitance (CSX) sensing are connected to the PSoC 64 as Figure 5-11 shows. Three external capacitors - CMOD for CSD, CINTA and CINTB for CSX - are present on the CY8CMOD-064B0S2-4343W. Note that CINTB can be reused as CSH. For details on using CapSense including design guidelines, see the Getting Started with CapSense Design Guide.

Figure 5-11. 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.

5.2.8 LEDs

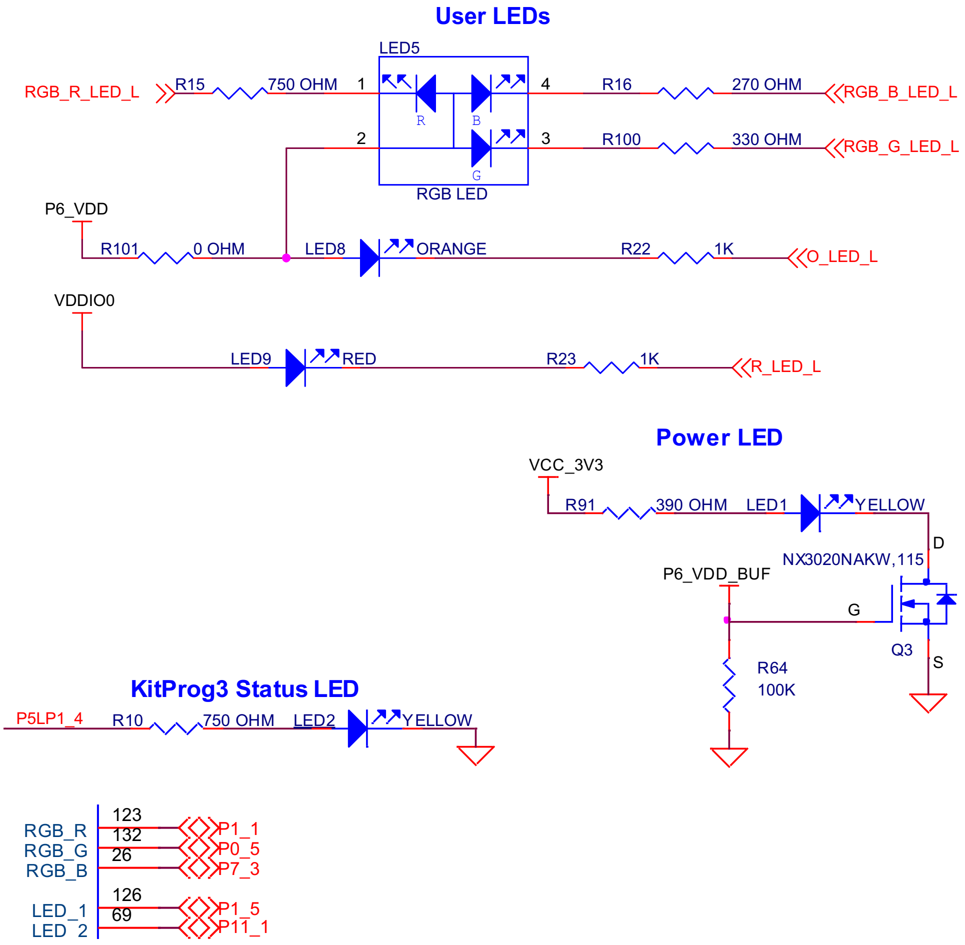
LED2 (Yellow) indicates the status of KitProg3 (See the KitProg3 User Guide for details). **LED1** (Yellow) indicates the status of the power supplied to the board.

The board also has two user-controllable LEDs (**LED8** and **LED9**) and an RGB LED (**LED5**) connected to PSoC 64 pins for user applications.

Table 5-3. User LEDs

User LED	PSoC Pin
RGB.R	P1[1]
RGB.G	P0[5]
RGB.B	P7[3]
LED.O	P1[5]
LED.R	P11[1]

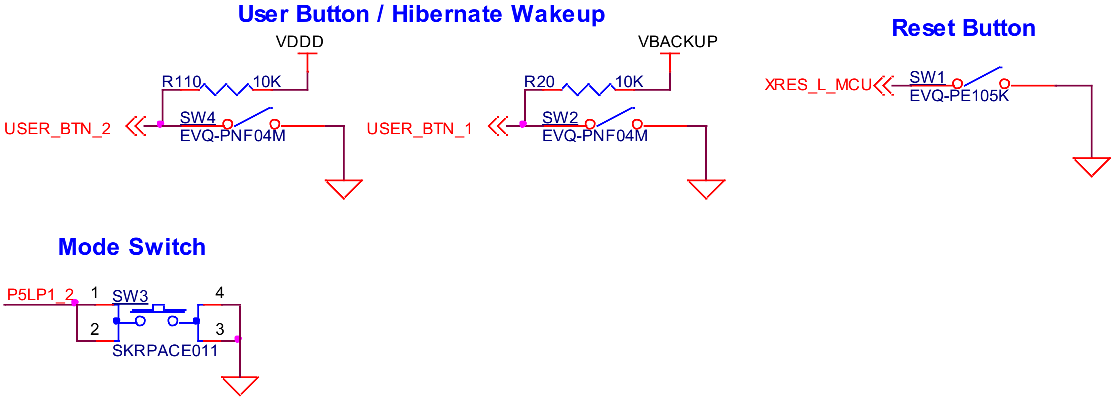
Figure 5-12. LEDs



5.2.9 Push Buttons

The board has a reset button, two user-controllable buttons and a KitProg3 Mode selection button. The reset button (**SW1**) is connected to the XRES pin of the PSoC 64 device and is used to reset the device. Two user buttons (**SW2** and **SW4**) are connected to pin P0[4] and P1[4] of the PSoC 64 respectively. In addition, the Mode selection button (**SW3**) is connected to the PSoC 5LP device for programming mode selection (Refer to the KitProg3 User Guide for details). All buttons are active LOW configuration and short to GND when pressed. The CY8CMOD-064B0S2-4343W has a pull-up on the PSoC 64 XRES line.

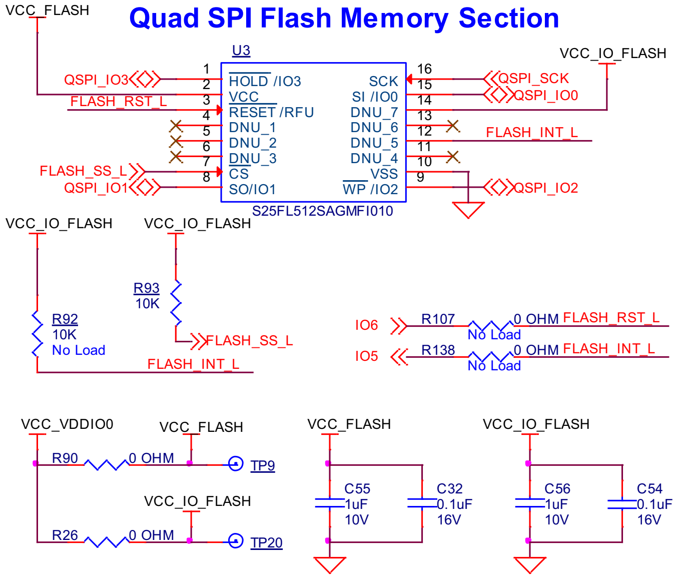
Figure 5-13. Schematics of Push Buttons



5.2.10 Cypress Quad SPI NOR Flash

The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board has a Cypress NOR flash memory (S25FL512SAGMFI010) of 512-Mbit capacity. The NOR flash is connected to the Quad SPI interface of the PSoC 64 device. The NOR flash device can be used for both data and code with execute-in-place (XIP) support and encryption. Note that the policy used during provisioning must support the SMIF interface to be able to store code in the external flash. The policies that enable the SMIF interface are provided in the templates.

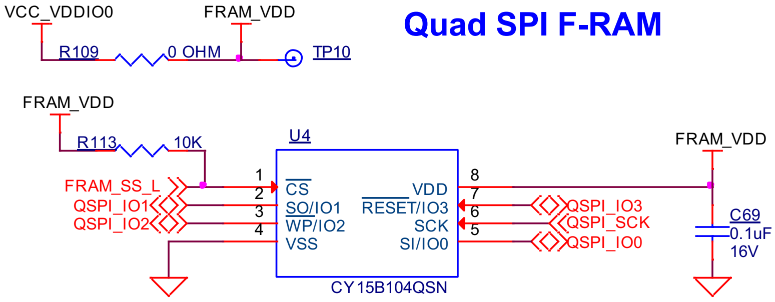
Figure 5-14. Schematics of QSPI Flash



5.2.11 Cypress Quad SPI F-RAM

The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board contains a CY15B104QSN Excelon™ F-RAM device, which can be accessed through Quad SPI interface. The F-RAM is 4-Mbit (512K × 8) and is capable of Quad SPI speed up to 108 MHz but the PSoC 64 is limited to 80 MHz.

Figure 5-15. Schematics of Quad SPI F-RAM

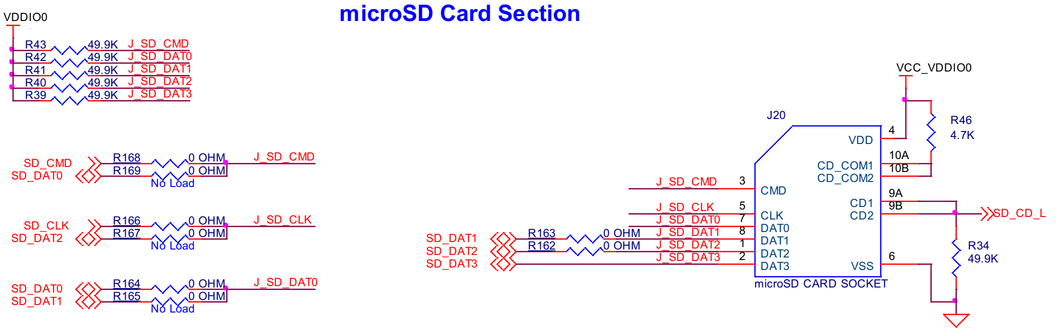


5.2.12 microSD card section

The PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Board contains a bottom-mounted microSD card holder with card detect pin that is connected to the PSoC 64 device. It is powered by VDD_VDDIO0 (connected to VCC_3V3 by default). The PSoC 64 is capable of UHS-I but is limited to High-Speed mode (50 MHz clock) in this kit.

By default, the PSoC 64 device is connected using an SDHC interface but optionally can be connected using SPI by re-working a few zero-ohm resistors.

Figure 5-16. Schematics of microSD Card Section



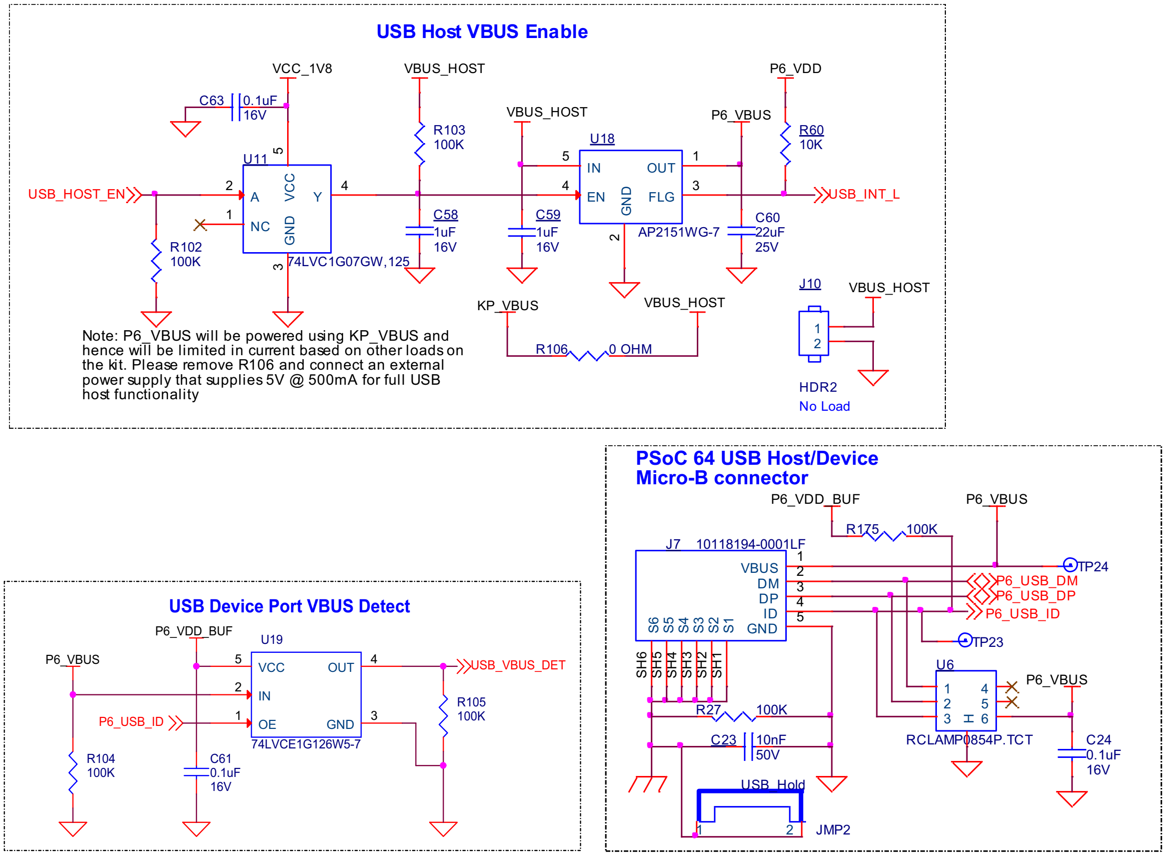
Note: Load R165, R167 and R169 and remove R162, R163, R164, R166 and R168 to change from SDHC to SPI interface for microSD card

5.2.13 PSoC 64 USB Section

The board contains a micro-B USB connector for the PSoC 64 device. It is capable of both device and host functionality. Although the PSoC 64 does not support USB-OTG, the hardware is compliant with it. By default, the PSoC 64 device will work as a USB device; when an OTG cable (all such cables have ID pin connected to GND) is connected, it will work as a USB Host.

As a USB Host, the board must provide power to a USB device that is connected to it. This power is provided by VBUS_HOST which is controlled by the PSoC 64 using a load switch. By default, VBUS_HOST is powered using KP_VBUS and optionally can be powered using external sources through **J10**.

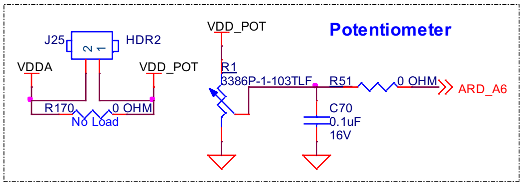
Figure 5-17. PSoC 64 USB



5.2.14 Potentiometer Section

The board contains a 10K potentiometer connected to pin A6 (P10[6]) of Arduino-header (J2). The fixed ends are connected to VDDA (VDD_POT through J25) and GND and hence may contribute to leakage current on the P6_VDD power supply. Remove jumper J25 to disconnect power from the potentiometer when measuring P6_VDD current.

Figure 5-18. Schematics of Potentiometer



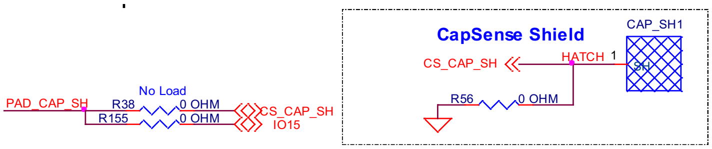
5.3 PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Rework

5.3.1 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 a shield pin. This pattern can be connected to P7[4] by populating R38 and removing R56. Pin P7[4] needs to be configured as a shield pin in the firmware. CINTB (C15 on MOD1) connected to P7[2] must be configured as CSH in firmware when using the CapSense Shield.

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

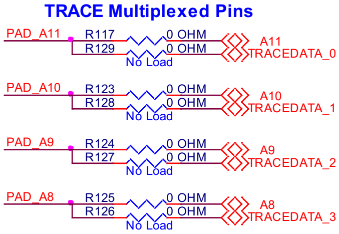
Figure 5-19. Schematics of CapSense Shield



5.3.2 ETM Trace Header

The 20-pin ETM trace header J12 is not loaded by default and the lines to the header are used as I/Os on header J2. To connect the PSoC 64 to trace header, populate the resistors R126–R129 and remove resistors R117, R123–R125.

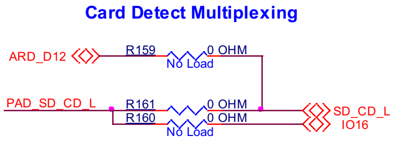
Figure 5-20. ETM Trace Header



5.3.3 microSD Card Detect Multiplexing

On the PSoC 64 device, the default card detect pin for the SHDC block is P12[1]. However, on this kit, P13[7] is connected to the card detect pin on the microSD card slot. Therefore, the firmware must be modified to use P13[7] as the card detect pin. In order to instead use the default PSoC 64 card detect pin, remove R161 and load R159. In this case, P13[7] can optionally be used as a GPIO by loading R160 which connects it to an I/O header.

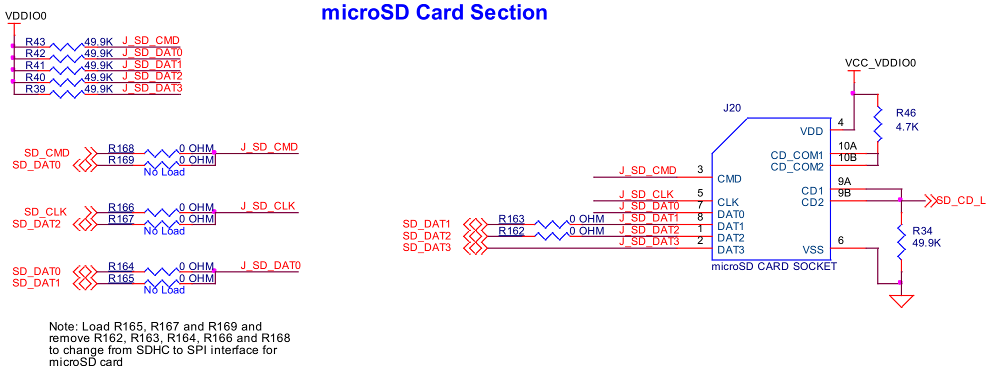
Figure 5-21. microSD Card Detect Multiplexing



5.3.4 microSD Card SPI Multiplexing

The microSD card is connected by a 6-pin SDHC interface by default i.e., CLK, CMD and DAT[0:3]. There is an optional provision to connect it over a 4-pin SPI interface i.e., CLK, MOSI, MISO and SSEL. To do this, load R165, R167, and R169 and remove R162, R163, R164, R166, and R168.

Figure 5-22. microSD Card SPI Multiplexing

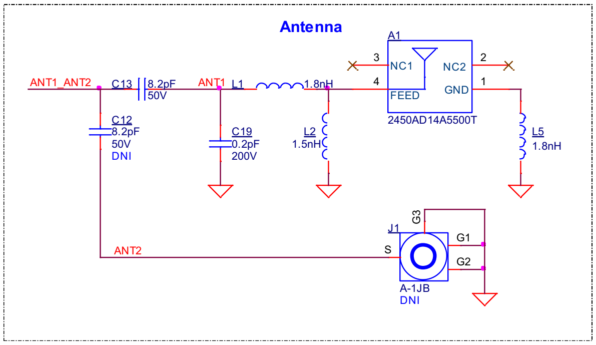


Note: Load R165, R167 and R169 and remove R162, R163, R164, R166 and R168 to change from SDHC to SPI interface for microSD card

5.3.5 U.FL (UMCC) Connector for External Antenna

The RF output of CYW4343W is connected to the chip antenna by default. To disconnect the chip antenna and connect an external antenna, remove C13 and populate C12 and J1 on CY8CMOD-064B0S2-4343W.

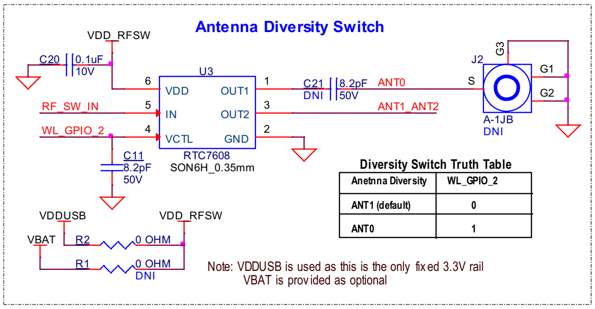
Figure 5-23. U.FL (UMCC) Connector for External Antenna



5.3.6 U.FL (UMCC) Connector for Antenna Diversity

To evaluate antenna diversity, an external antenna can be connected to the output of antenna diversity RF switch by populating C21 and J2 on CY8CMOD-064B0S2-4343W.

Figure 5-24. U.FL (UMCC) Connector for Antenna Diversity



Diversity Switch Truth Table

Anetnna Diversity	WL_GPIO_2
ANT1 (default)	0
ANT0	1

Note: VDDUSB is used as this is the only fixed 3.3V rail
VBAT is provided as optional

5.4 Bill of Materials

Refer to the BOM files in the kit webpage.

5.5 Frequently Asked Questions

1. How does CY8CKIT-064B0S2-4343W handle a voltage connection when multiple power sources are plugged in?

   There are three different options to power the baseboard; KitProg3 Micro-B USB connector (**J6**), PSoC 64 device's Micro-B USB connector (**J7**), and External DC supply via VIN connector (**J5**). The voltage from each of the sources in passed through ORing diodes that supply VCC_IN.
2. What are the input voltage tolerances? Is there any overvoltage protection on this kit?

   Input voltage levels are as follows:

Table 5-4. Input voltage levels

Supply	Typical I/P Voltage	Absolute max
USB Micro-B connector **(J6, J7)**	4.5 V to 5.5 V	5.5 V
VIN connector **(J5)**	7 V to 12 V	18 V

   The Kit can't be powered through the programming header **J11** and has reverse voltage protection on this header
3. Why is the voltage of the kit restricted to 3.3 V? Can't it drive external 5-V interfaces?

   PSoC 64 chip is not meant to be operated at voltages greater than 3.6 V. Powering PSoC 64 chip to more than 4 V will damage the chip. It is recommended to power PSoC 64 at 3.3 V.
4. I am unable to program the target device.
   a. Check **J15** to ensure that jumper shunt is placed.
   b. Make sure that no external devices are connected to the external programming header J11.
   c. Update your KitProg3 version to the latest one using the steps mentioned in the KitProg3 User Guide.
5. 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 CapSense. Note that additional tuning may be required when the overlay is changed.

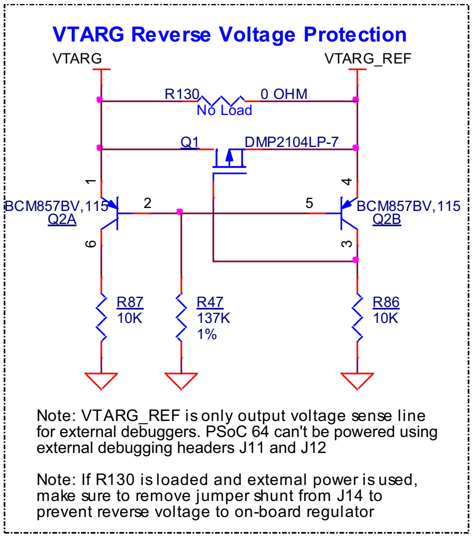
6. Can I power the kit using external program/debug headers J11 and J12?

   No, this is not possible by default in this board. The target MCU is powered by on-board regulators only and hence one of the 3 main sources (**J5**, **J6** and **J7**) must be present.

   There is a protection circuit that prevents reverse voltage from VTARG_REF to VTARG. Hence the board can't be powered through **J11** and **J12**. However this can be by-passed by loading R130.

   Note: This modification is not recommended as the target device will have no protection and will be permanently damaged if 5 V is supplied.

Figure 5-25. VTARG Reverse Voltage Protection



Note: VTARG_REF is only output voltage sense line for external debuggers. PSoC 64 can't be powered using external debugging headers J11 and J12

Note: If R130 is loaded and external power is used, make sure to remove jumper shunt from J14 to prevent reverse voltage to on-board regulator

Revision History

Document Revision History

Document Title: CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Guide Document Number: 002-29286			
Revision	**ECN Number**	**Issue Date**	**Description of Change**
**	6793005	03/24/2020	New kit guide.
*A	6950947	08/24/2020	Updated Document Title to read as “CY8CKIT-064B0S2-4343W PSoC 64 Secure Boot Wi-Fi BT Pioneer Kit Guide”. Replaced “PSoC 64 Wi-Fi BT Secure Boot Pioneer Kit” with “PSoC 64 Secure Boot Wi-Fi BT Pioneer Kit” in all instances across the document. Updated Safety and Regulatory Compliance Information chapter on page 5: Replaced “Safety and Regulatory Compliance Information” with “Safety Information” in chapter heading. Updated description. Removed “Regulatory Compliance Information”. Updated Introduction chapter on page 7: Updated hyperlinks. Updated description. Updated “Kit Contents” on page 8: Updated Figure 1-1. Updated “Getting Started” on page 9: Updated description. Updated “Board Details” on page 9: Updated description. Updated Figure 1-2. Updated Table 1-1. Updated “Additional Learning Resources” on page 15: Updated hyperlinks. Updated description.

Document Revision History *(continued)*

Document Title: CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Guide Document Number: 002-29286			
Revision	**ECN Number**	**Issue Date**	**Description of Change**
*A (cont.)	6950947	08/24/2020	Updated Software Installation chapter on page 18: Updated “Install Software” on page 18: Updated description. Added “Updating tools for ModusToolbox 2.1 or older”. Updated Kit Operation chapter on page 20: Updated “Theory of Operation” on page 20: Updated description. Updated Figure 3-3. Updated Figure 3-4. Updated Figure 3-5. Updated “KitProg3: On-Board Programmer/Debugger” on page 26: Updated description. Updated “Programming and Debugging using ModusToolbox Software” on page 26: Updated description. Updated Figure 3-6. Updated “USB-I2C Bridge” on page 28: Updated Figure 3-9. Updated Running Code on PSoC 64 “Secure Boot” MCUs chapter on page 29: Updated “Provisioning Overview” on page 29: Updated description. Updated Figure 4-2. Updated “Create ModusToolbox Example Project” on page 31: Updated description. Updated Figure 4-4. Updated Figure 4-5. Removed figure “Finish Selection”. Updated “Provision the Device” on page 33: Updated description. Updated Figure 4-6. Updated Figure 4-7. Removed “Project Workspace having a new ‘packet’ directory”. Updated “Build and Program the Example Project” on page 37: Updated description. Updated Figure 4-8. Updated Figure 4-9. Updated “Additional Code Examples” on page 39: Updated description.

Document Revision History *(continued)*

Document Title: CY8CKIT-064B0S2-4343W PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Guide			
Document Number: 002-29286			
Revision	ECN Number	Issue Date	Description of Change
*A (cont.)	6950947	08/24/2020	Updated Hardware chapter on page 40: Updated “Hardware Functional Description” on page 40: Updated “CY8CMOD-064B0S2-4343W (MOD1)” on page 40: Updated description. Updated “Serial Interconnection between PSoC 5LP and PSoC 64 Device” on page 46: Updated Figure 5-3. Updated “Serial Interconnection Between PSoC 5LP and CYW4343W” on page 47: Updated description. Updated “Power Supply System” on page 47: Updated Figure 5-5. Updated “Voltage regulators” on page 48: Updated Table 5-1. Updated “Voltage Selection” on page 49: Updated description. Updated Figure 5-7. Updated “I/O Headers” on page 51: Updated “PSoC 64 I/O Headers (J21, J22, and J24)” on page 51: Updated description. Updated “CapSense Circuit” on page 52: Updated Figure 5-11. Updated “LEDs” on page 53: Added Table 5-3. Updated Figure 5-12. Updated “Cypress Quad SPI F-RAM” on page 55: Updated description. Updated “microSD card section” on page 55: Updated Figure 5-16. Updated “Potentiometer Section” on page 56: Updated description. Updated “PSoC 64 “Secure Boot” Wi-Fi BT Pioneer Kit Rework” on page 57: Updated “microSD Card SPI Multiplexing” on page 58: Updated Figure 5-22. Updated “Frequently Asked Questions” on page 59: Updated description.