RUTDevKit-PSoC62 User Manual

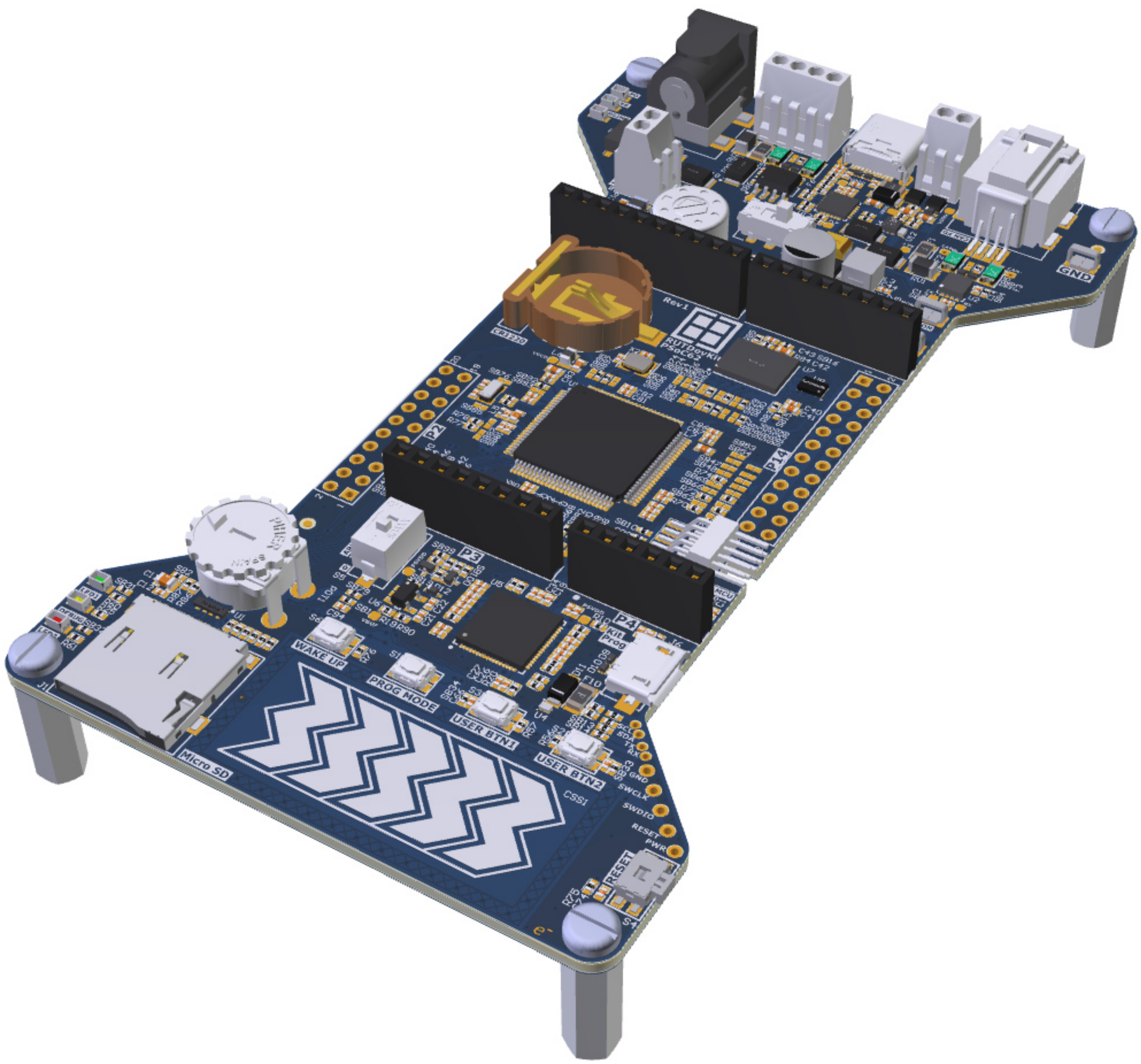

Table of Contents

Versions

Table 1

Version	Date	Rationale
0.1	September 20, 2021	First draft. Author: GDR
0.2	October 21, 2021	Pre-Release. Author: GDR
1.0	November 18, 2021	Release. Author: GDR
1.1	May 5. 2022	Known Issues added. Author: GDR
1.2	July 20. 2022	Power distribution diagram [Fig.4] extended. Author: GDR

Introduction

RUTDevKit-PSoC62 is a development platform used by firmware and hardware designers to develop their products. RUTDevKit-PSoC62 was designed by Rutronik to promote outstanding products selected only from their suppliers.

Features

- Infineon (Cypress) CY8C6245AZI-S3D72 Cortex®-M0+ and Cortex®-M4F 512KB Flash 256KB SRAM high-performance, ultra-low-power microcontroller.
- On-board debugger KitProg3 with I2C and UART USB bridge.
- On-board capacitive slider based on CapSense® CSD, CSX technologies.
- APMemory External QSPI 64Mbit PSRAM Memory APS6404L-3SQR-ZR.
- Infineon External QSPI 512Mbit Semper NOR Flash S25HL512TFABHI010.
- Infineon (Cypress) Stand-alone USB Power Delivery Sink controller CYPD3177.
- Nordic Semiconductor Li-ION Battery charger with nPM1100.
- Switching mode power supplies AP63357 from Diodes Inc. and BD83070GWL from ROHM.
- JAE USB Type-C connector interfaces with the microcontroller.
- Infineon CAN FD driver TLE9251VLE.
- MaxLinear RS485 driver SP3078EEN-L/TR.
- ADAM-TECH MicroSD card socket.
- Keystone Electronics Corp. CR1220 coin battery socket for RTC and low power applications.
- Keystone Electronics Corp. current monitoring shunt resistor.
- Panasonic mechanical SMD tactile buttons and OSRAM SMD LEDs.
- All CY8C6245AZI-S3D72 GPIOs are available at P2 and P14 headers.
- TOSHIBA Power MOSFETs SSM6J507NU and current limiters TCK22946G.
- NJR low power amplifier NJU77001F.
- 10-pin Amphenol ICC SWD header for J-Link.
- AVX multilayer ceramic transient voltage suppressors.
- Passive components from Samsung EM, Yageo, ASJ.
- Arduino compatible headers from ADAM-TECH.
- Arduino IoT, RF shields optimized board shape.
- The USB cable for debugging from ASSMAN.

Overview

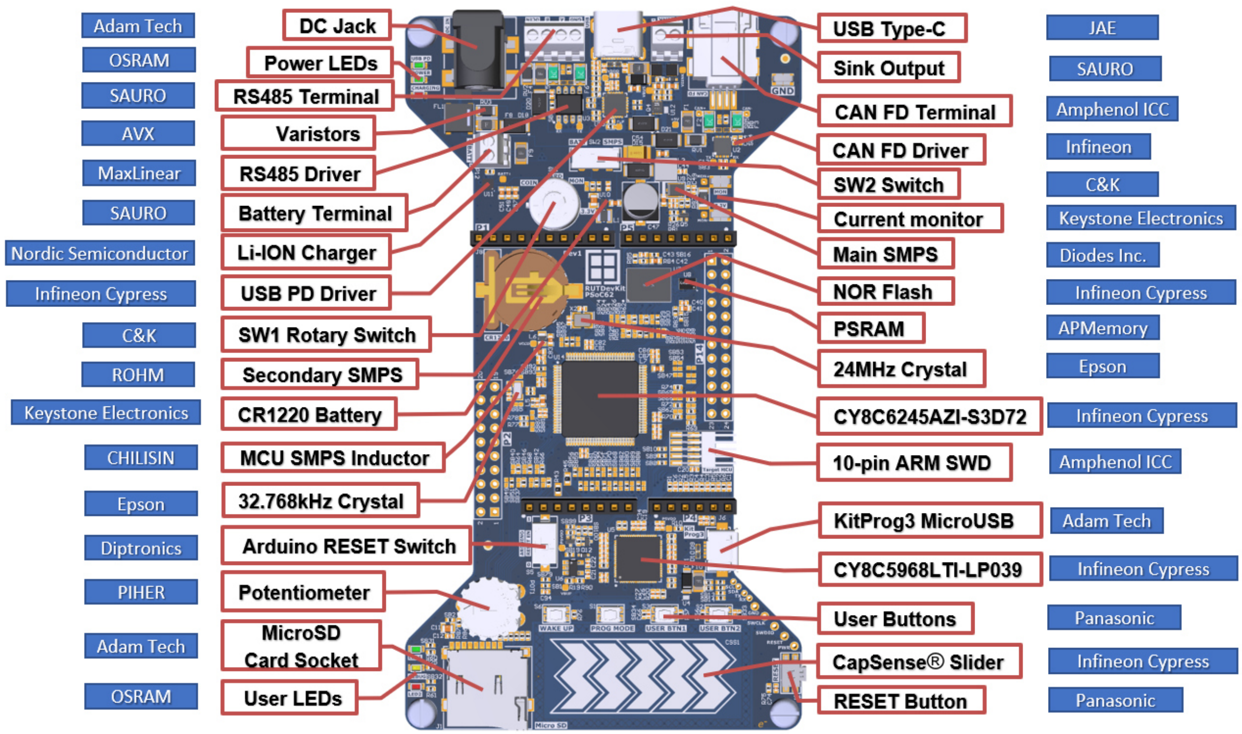

Fig. 1. RUTDevKit-PSoC62 Evaluation board's layout.

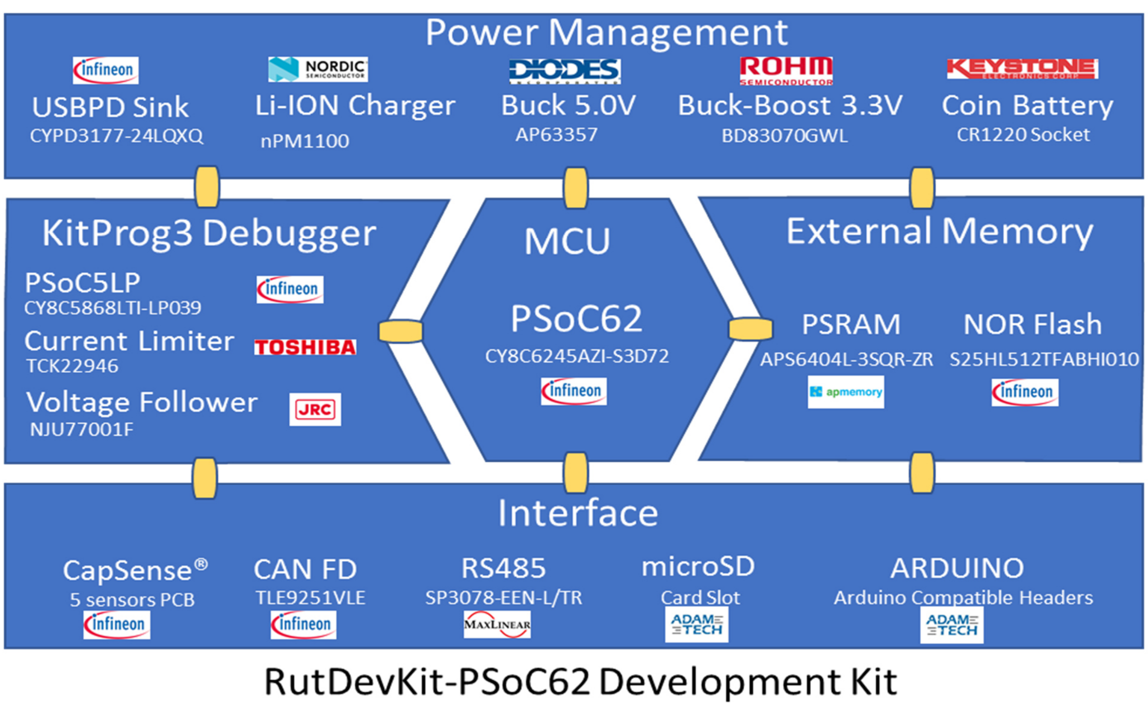

Fig. 2. Block diagram of the board.

Power Source Select

Six power sources are available in RUTDevKit-PSoC62:

1. KitProg3 USB port.
2. 5V SMPS powered from CAN FD, RS485, DC Jack, and USB Type C interfaces.
3. CR1220 coin battery socket.
4. Arduino connectors – configured using R43 and R45 0R 0402 resistors.
5. Li-ion Battery.
6. Current monitor TP17, only if R82 is removed.

Select the power source using SW1 the Coin Battery "**COIN**", Arduino headers "**ARD**", Current Monitor "**MON**", 3.3V SMPS "**3.3V**". With SW2 users can select the power source as **BATT** – Li-ION battery or 5V **SMPS**.

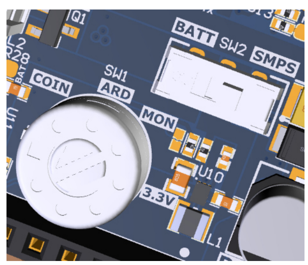

Fig. 3. Power source selectors.

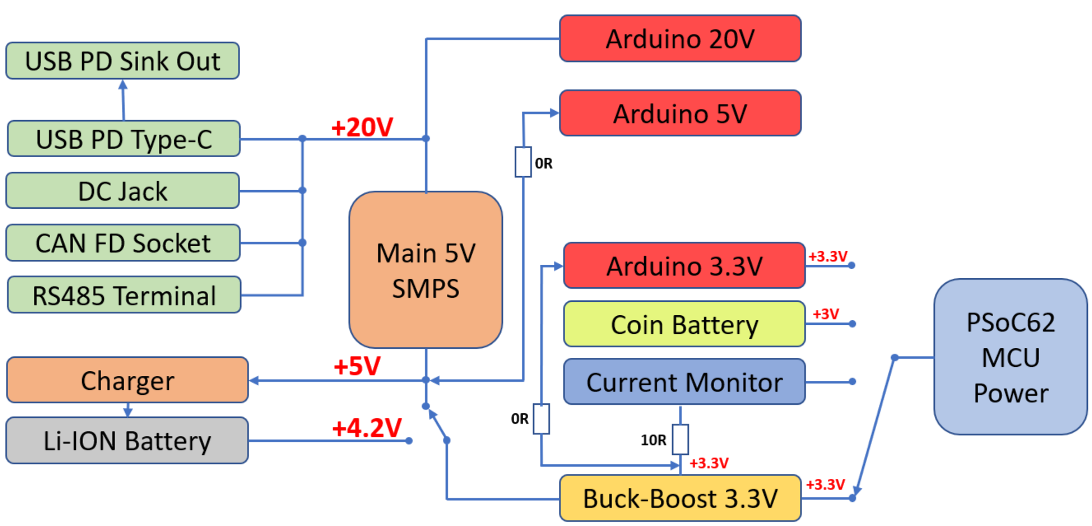

Fig. 4. RutDevKit-PSoC62 Power Distribution Diagram.

Programming Using External Connector

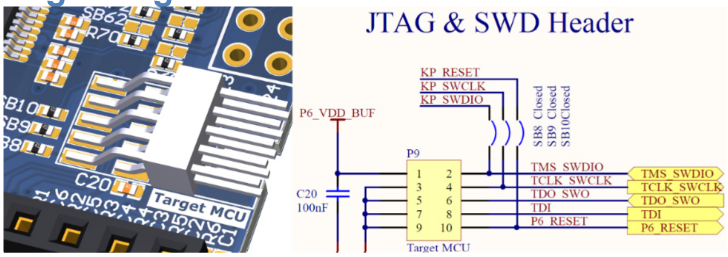

Fig. 5. 10-pin male 1.27mm pitch, SWD connector.

Users may use third-party programming devices to connect the CY8C6245AZI-S3D72 target via the P9 SWD connector. The onboard “KitProg3” debugger should not be powered while using an external JTAG connector.

CAN FD Socket

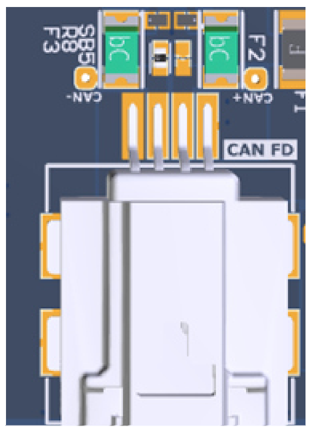

Fig. 6. CAN FD Connector.

The Amphenol ICC Minitek MicroSpace™ is used for CAN FD connection. Part No.: **10142344-104KLF**. One for CAN input cable, another for CAN output cable. To connect the wires certain receptacles are needed: 10142348-004LF and contacts 10141272-111LF have to be crimped and mounted into the receptacle.

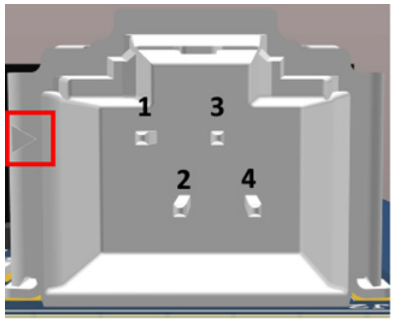

Fig. 7. CAN FD Pinout

Spare GPIOs

All GPIOs of CY8C6245AZI-S3D72 MCU are available at sockets P2 and P14. Some may need to be configured using solder bridges.

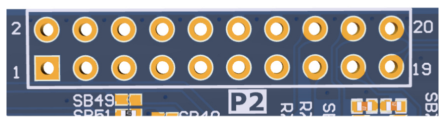

Fig. 8. P2 Socket for spare GPIOs

Table 2

Socket P2 Pinout			
Pin No.	**Name**	**Name**	**Pin No.**
1	P2.0	P2.4	2
3	P2.2	P2.3	4
5	P2.6	P2.1	6
7	P2.5	P11.6	8
9	P11.4	P11.2	10
11	P11.1	P11.3	12
13	P11.5	P11.7	14
15	P14.1	P14.0	16
17	P0.0	P0.1	18
19	P12.7	P12.6	20

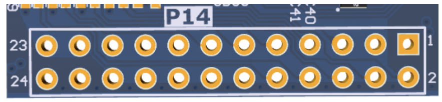

Fig. 9. P14 Socket for spare GPIOs

Table 3

Socket P14 Pinout			
Pin No.	**Name**	**Name**	**Pin No.**
1	P7.1	P9.2	2
3	P7.2	P9.0	4
5	P9.3	P7.7	6
7	P9.1	P7.3	8
9	P7.6	P7.0	10
11	P7.5	P5.0	12
13	P7.4	P6.2	14
15	P3.0	P6.6	16
17	P6.0	P5.1	18
19	P6.4	P6.3	20
21	P3.1	P6.5	22
23	P6.1	P6.7	24

Solder Bridges

Table 4

Solder Bridge	Circuit	Default
SB1	Potentiometer output signal with P10.4	Closed
SB2	Power for the microSD card interface.	Closed
SB3	+5V Power for the CAN FD driver.	Closed
SB4	+3.3V Power for the CAN FD driver.	Closed
SB5	120 Ohm termination for CAN FD connector.	Opened
SB6	Power for the RS485 Driver.	Closed
SB7	120 Ohm termination for RS485 terminal.	Opened
SB8	SWDIO signal with KitProg3.	Closed
SB10	RESET signal with KitProg3.	Closed
SB11	KitProg3 I2C SCL with PSoC6 MCU I2C SCL.	Closed
SB14	KitProg3 UART RX with PSoC6 MCU UART TX.	Closed
SB15	Power supply for the PSRAM IC.	Closed
SB16	Power supply for the NOR Flash IC.	Closed
SB17	PSoC6 MCU I2C SDA with charger SDA.	Closed
SB18	PSoC6 MCU I2C SCL with charger SCL.	Closed

SB19	Target voltage follower input isolation.	Closed
SB20	3.3V SMPS Enable with VIN.	Closed
SB21	3.3V SMPS Enable with external signal (PMIC).	Opened
SB22	PSoC6 MCU I2C SDA with USB PD IC SDA.	Closed
SB23	PSoC6 MCU I2C SDL with USB PD IC SDL.	Closed
SB24	PSoC6 P11.7 with P2 header GPIO 14.	Opened
SB25	PSoC6 P11.6 with P2 header GPIO 8.	Opened
SB26	PSoC6 P11.5 with P2 header GPIO 13.	Opened
SB27	PSoC6 P11.4 with P2 header GPIO 9.	Opened
SB28	PSoC6 P11.3 with P2 header GPIO 12.	Opened
SB29	PSoC6 P11.1 with P2 header GPIO 11.	Opened
SB30	Flash SSEL Signal.	Closed
SB31	LED1 with PSoC6.	Closed
SB32	LED2 with PSoC6.	Closed
SB33	USER_BTN2 with PSoC6.	Closed
SB34	USER_BTN1 with PSoC6.	Closed
SB35	PSoC6 P11.2 with P2 header GPIO 10.	Opened
SB36	PSRAM SSEL Signal	Closed
SB37	PSoC6 P2.0 with P2 header GPIO 1.	Opened
SB38	Arduino IO 4 with PSoC6.	Closed
SB39	PSRAM RESET signal.	Opened
SB40	PSoC6 P2.1 with P2 header GPIO 6.	Opened
SB41	PSoC6 P2.2 with P2 header GPIO 3.	Opened
SB42	PSoC6 P2.3 with P2 header GPIO 4.	Opened
SB43	PSoC6 P2.4 with P2 header GPIO 2.	Opened
SB44	PSoC6 P9.0 with P14 header GPIO 4.	Opened
SB46	PSoC6 P2.6 with P2 header GPIO 5.	Opened
SB47	PSoC6 P9.1 with P14 header GPIO 7.	Opened
SB49	PSoC6 P2.6 with P2 header GPIO 5.	Opened
SB50	PSoC6 P9.2 with P14 header GPIO 2.	Opened
SB51	microSD Card Detect signal.	Closed
SB52	5V SMPS Power Good signal.	Opened
SB53	PSoC6 P9.3 with P14 header GPIO 5.	Opened
SB54	USB PD IC Interrupt signal.	Opened
SB55	PSoC6 P3.0 with P14 header GPIO 15.	Opened
SB56	PSoC6 UART RX signal.	Closed
SB57	PSoC6 P3.1 with P14 header GPIO 21.	Opened
SB58	PSoC6 UART TX signal.	Closed
SB59	PSoC6 P7.0 with P14 header GPIO 10.	Opened
SB60	PSoC6 P5.0 with P14 header GPIO 12.	Opened
SB61	CAN FD RX signal.	Closed

SB62	PSoC6 P7.3 with P14 header GPIO.	Closed
SB63	PSoC6 P5.1 with P14 header GPIO.	Opened
SB64	CAN FD TX signal.	Closed
SB65	PSoC6 P7.4 with P14 header GPIO 8.	Opened
SB66	PSoC6 P7.5 with P14 header GPIO 11.	Opened
SB67	PSoC6 P7.6 with P14 header GPIO 9.	Opened
SB68	PSoC6 P6.2 with P14 header GPIO 14.	Opened
SB69	PSoC6 P7.7 with P14 header GPIO 6.	Opened
SB70	CAN FD Driver Stand-By control input.	Closed
SB71	PSoC6 P6.0 with P14 header GPIO 17.	Opened
SB72	RS485 RX signal.	Closed
SB73	PSoC6 P6.1 with P14 header GPIO 23.	Opened
SB74	PSRAM Interrupt signal.	Opened
SB75	RS485 TX signal.	Closed
SB76	PSoC6 P0.0 with P2 header GPIO 17.	Opened
SB77	PSoC6 P6.3 with P14 header GPIO 20.	Opened
SB78	RS485 DE signal.	Closed
SB79	Potentiometer VDD terminal.	Closed
SB80	PSoC6 P6.4 with P14 header GPIO 19.	Opened
SB81	PSoC6 P14.0 with P2 header GPIO 16.	Opened
SB82	JTAG TDO signal.	Closed
SB83	WCO Input signal.	Closed
SB84	PSoC6 P6.5 with P14 header GPIO 22.	Opened
SB85	PSoC6 P0.1 with P2 header GPIO 18.	Opened
SB86	JTAG TDI signal.	Closed
SB87	PSoC6 P14.0 with P2 header GPIO 16.	Opened
SB88	PSoC6 P6.6 with P14 header GPIO 16.	Opened
SB89	JTAG TMS, SWDIO signal.	Closed
SB90	PSoC6 P6.7 with P14 header GPIO 24.	Opened
SB91	JTAG TCLK, SWCLK signal.	Closed
SB92	PSoC6 Backup Voltage Input.	Closed
SB93	Coin Battery with PSoC6 Backup Voltage Input.	Opened
SB94	WCO Output signal.	Closed
SB95	PSoC6 P12.7 with P2 header GPIO 19.	Opened
SB96	PSoC6 P12.6 with P2 header GPIO 20.	Opened
SB97	ECO Input Signal.	Closed
SB98	ECO Output Signal.	Closed
SB99	Voltage Divider Enable signal.	Closed
SB100	Voltage Divider ADC signal.	Closed

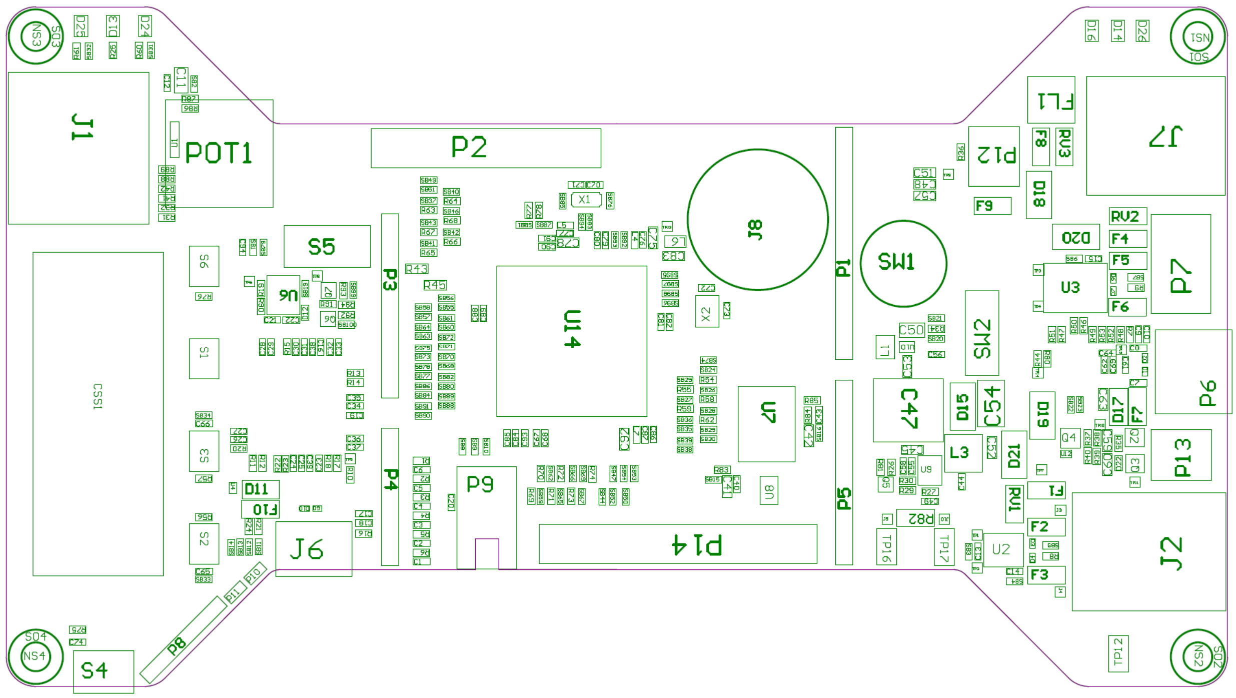

Fig. 10. Locations of the Solder Bridges (please check the schematics and layout document to see in detail).

Fuses

Resettable and non-resettable fuses are used for this project. In case the fuses are not fit for the user's final application the user must change the fuses by unsoldering and soldering new ones that meet the requirements. RutDevKit-PSoC62 Rev1 fuses list:

1. **F1, F4, F8, F9, F10** "High I²t Chip" 2A, 63V 1206 SMD. Part No.: CC12H2A-TR.
2. **F2, F3, F5, F6** "Resettable PTC" 50mA 60V 1206 SMD. Part No.: PTS120660V005.
3. **F7** "High I²t Chip" 5A, 32V 1206 SMD. Part No.: CC12H5A-TR.

RutDevKit-PSoC62 Firmware Examples

RutDevKit-PSoC62 BSP

This project is needed as a board support package while creating a new project with the RutDevKit-PSoC62 development kit.

Hello World

This example is an introduction to the basic components of the board: LEDs, Buttons, and KitProg3 UART for debugging.

Arduino ADC DMA

This example demonstrates how to use the PDL library to measure all the ADC channels on the Arduino ADC header.

Arduino ADC HAL

This example demonstrates how to use the HAL library to measure all the ADC channels on the Arduino ADC header.

CAN FD Test

This example demonstrates how to use the PDL library to send and receive CAN FD packages while in external loop mode.

CapSense CSD Slider

This example demonstrates how to use the onboard slider configured in CSD mode.

I2C Scanner

This application is used to find all the devices connected to the I2C.

QSPI Semper NOR Flash XIP

This example demonstrates how to use the serial flash library and XIP (execute-in-place) feature.

QSPI APMemory PSRAM Dynamic Allocation

This example demonstrates how to configure and use the PSRAM APS6404L-3SQR-ZR with standard dynamic memory allocation functions such as malloc() etc.

QSPI APMemory PSRAM XIP

This example demonstrates how to configure and use PSRAM APS6404L-3SQR-ZR in XIP mode.

RS485 Modbus

This example demonstrates the RS485 interface capabilities using the Modbus protocol.

RTC Hibernate

This example demonstrates one of the low power modes: hibernation. RTC alarm is used as a wake-up source.

USB Power Delivery Control

This example demonstrates how to access and control the CYPD3177 power delivery, sink controller.

microSD Card FAT

This example demonstrates how to access microSD cards using a FAT file system.

USB Type-C CDC Test

This example is used for testing the USB port and demonstrates the CDC device software features.

4D Systems GEN4-FT812-43-T Display Demo

This example is for introduction to the 4D Systems GEN4-FT812-43T 4.3“ display

Fig. 11. GEN4-FT812-43T demo.

with resistive touch. The display is driven using an SPI interface therefore the adapter for Arduino headers is needed. The demo application has static display objects and rotating an image using EVE capabilities. The image rotation may be started or stopped using a button on the screen.

4D Systems SK-GEN4-43DCT-CLB Display Demo

This example is for introduction to the 4D Systems „gen4-uLCD-43DCT-CLB“ 4.3“ display module with capacitive touch. The display is controlled using the UART interface.

Fig. 12. SK-GEN4-43DCT-CLB demo.

The SGP30 sensor attached to the I2C is needed to have this demo working. The demo application has a gauge and a scope that represents CO2 equivalent concentration values in ppm.

The CO^2 Alarm Application

This application demonstrates the Epson's S1V3G340 Text-to-Speech Engine IC and Sensirion's SGP30 Air Quality sensor capabilities. The firmware example connects with the

Fig. 13. The hardware.

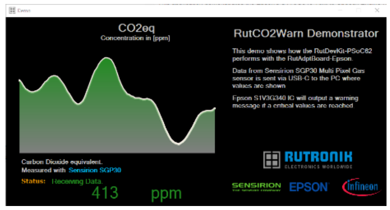

Fig. 14. The monitoring application.

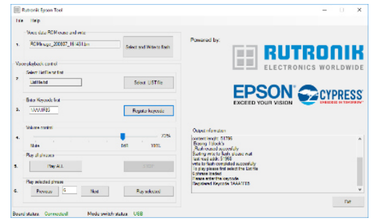

Fig. 15. Text-to-Speech programming.

Windows OS application via USB Type-C as a "Generic HID" device and represents the carbon dioxide equivalent values in ppm that are read from the SGP30 sensor. The Text-to-Speech IC is used for audible voice announcements. The LED1 on the RutDevKit-PSoC62 board will be blinking if the USB HID connection is established and active. A new RutAdaptBoard-TextToSpeech board has to be programmed with audio files using "Rutronik Epson Tool" that are already prepared and provided with the RutDevKit-PSoC62 RutCO2Alarm project.

The OBD-II Test Application

This application is intended to be used as a reference firmware example for the developers who need to quick-start with PSoC62 and CANFD OBD-II protocol.

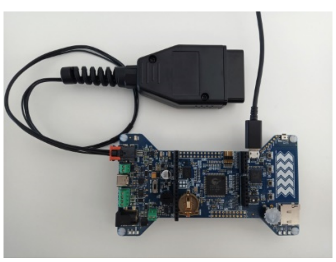

Fig. 16.OBD-II demo set-up.

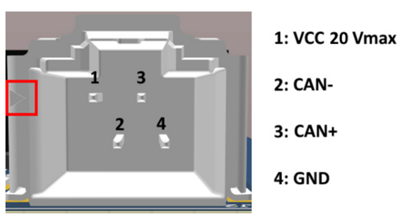

Fig. 17.CAN FD pinout.

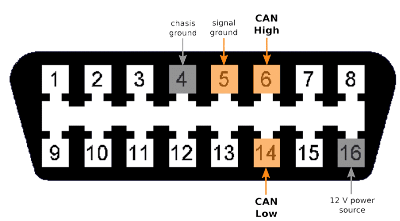

Fig. 18.OBD2 Socket.

The firmware example uses KitProg3 Debug UART for debugging output. Some of the most common OBD-II PIDs are presented once per second: Vehicle speed (0x0D), Engine speed (0x0C), Control module voltage (0x66), Intake air temperature (0x0F), Intake manifold absolute pressure (0x0B), Mass airflow sensor air flowrate (0x10).

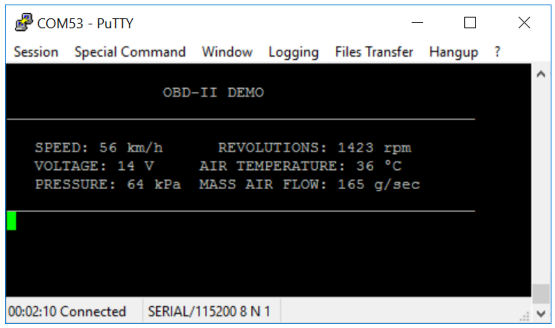

Fig. 19. Running the demo.

The OBD-II Turbocharger Monitor Application

This example is a practical implementation using 4D Systems "GEN4-FT812-43T" display for representing the vehicle's parameters gathered over the OBD cable using the OBD-II protocol. The OBD cable and SPI adapter are needed for this application to run. The demo application has a gauge that represents turbocharger boost and other values (battery voltage, engine speed, MAP, air intake temperature, MAF) are represented together with icons that are sensitive for touching and can be enabled or disabled when touched.

Fig. 20. Demo attached to the vehicle.

RutDevKit-PSoC62 (Factory) Production Firmware

This firmware example is preprogrammed in a factory and used to test most of the peripherals on board. The firmware checks external PSRAM, Flash, and micro SD Card memory. The debug information is available on the KitProg3 UART port. The RS485 and USB peripherals might be tested as they are echoing every symbol sent through the terminal. The CAN FD is tested once in loop-back mode. The CapSense is initialized and operational after the tests are complete and can be tested manually as it controls the LED1 brightness. The CapSense Tuner is also available via KitProg3 I2C. All the Arduino ADC inputs are shown as well. The date and time shown in the terminal window will notify if the RTC peripheral is functional. After the initial test is complete the LED1 will shine green and the brightness will depend on the CapSense slider position. If the test fails due to some peripheral faults, the LED2 stays on indefinitely.

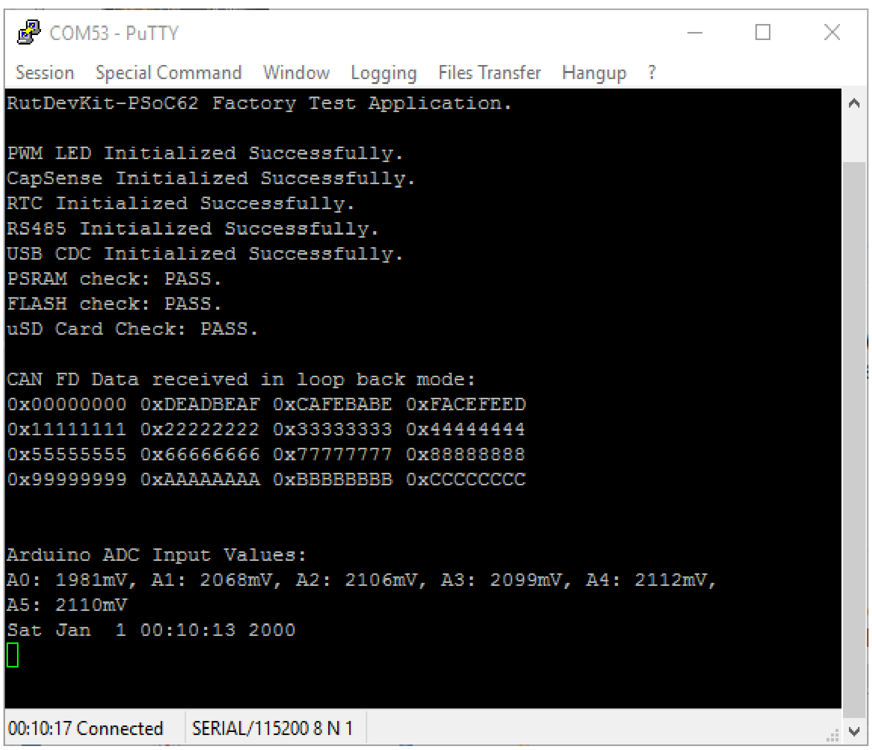

Fig. 21. Testing the RutDevKit-PSoC62

Telit NB-IoT, Cat-M, GSM ME310G1-WW Modem Application.

The Telit Cloud demo using the ME310G1-WW modem is available from Rutronik Kaunas. Please contact technical support for further information.

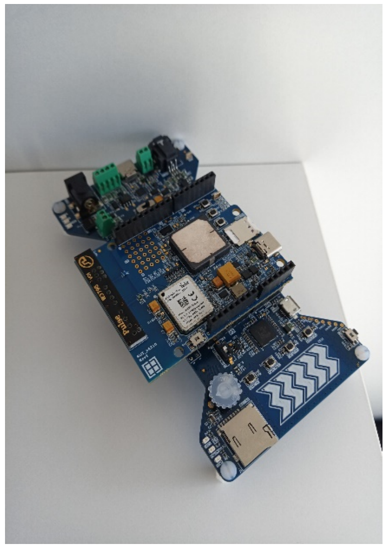

Fig. 22. The ME310G1-WW Arduino shield for RutDevKit-PSoC62

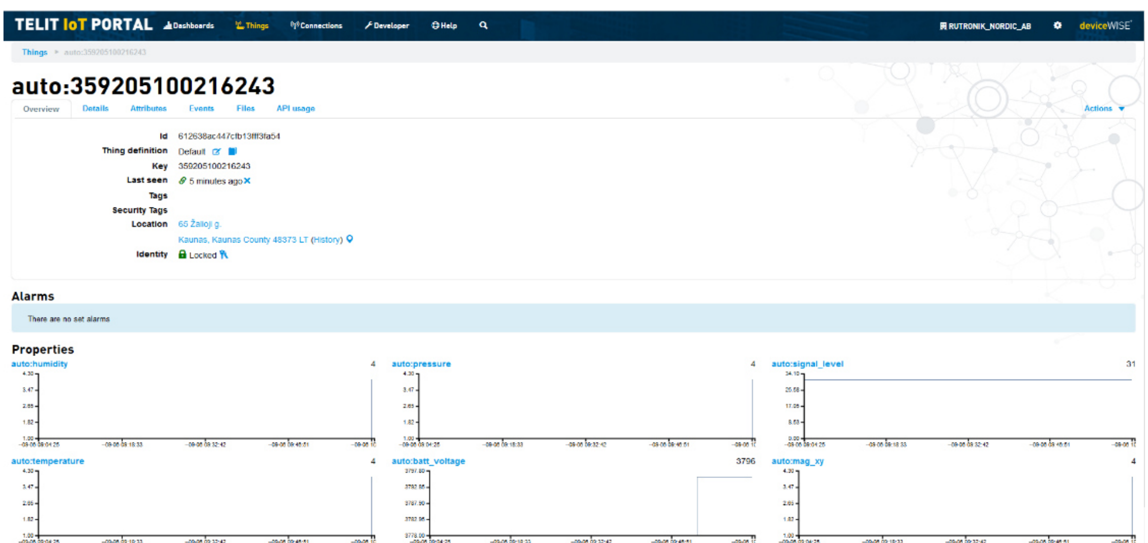

Fig. 23. The module uploading data to the Telit IoT Portal.

Electromagnetic Compatibility

RUTDevKit-PSoC62 was tested for electromagnetic disturbances and electromagnetic immunity and meet the requirements as in normative documents listed below:

Electromagnetic disturbances:
Radiated disturbance to 1 GHz
IEC 61000-4-20

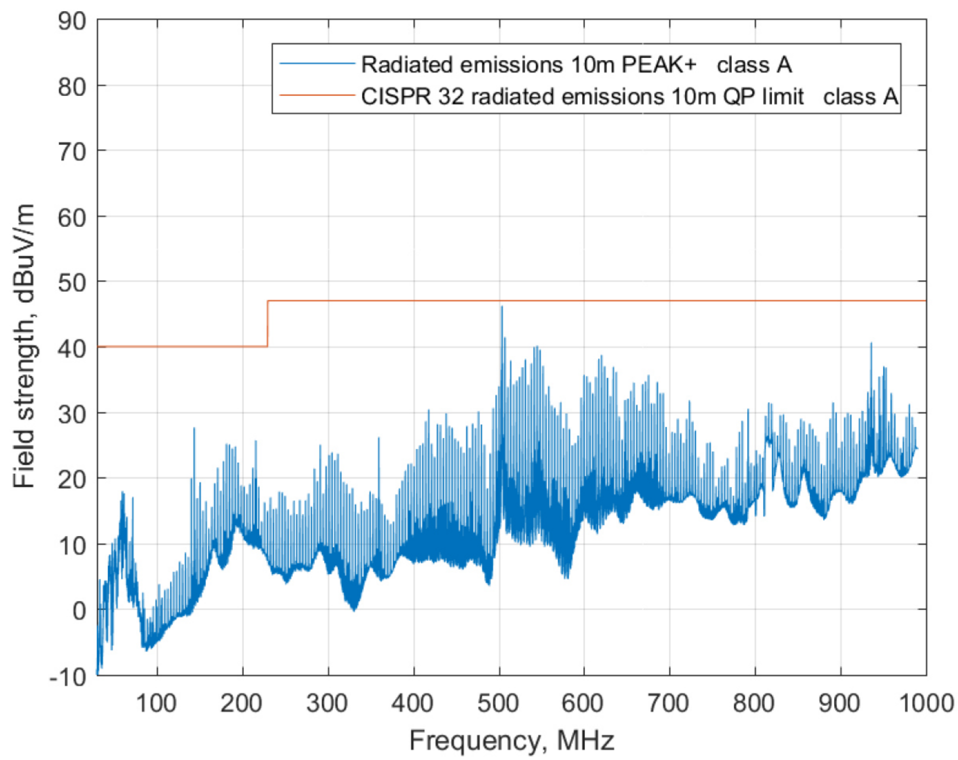

Fig. 24. Radiated disturbances while running PSRAM Test.

Legal Disclaimer

The evaluation board is for testing purposes only and, because it has limited functions and limited resilience, is not suitable for permanent use under real conditions. If the evaluation board is nevertheless used under real conditions, this is done at one's responsibility; any liability of Rutronik is insofar excluded.

Changing the fuses or solder bridges

Some of the components might be hard to access, therefore the SMD „Chipping Tool" is recommended to use for SMD solder bridges or fuses soldering on the RutDevKit-PSoC62 development board.

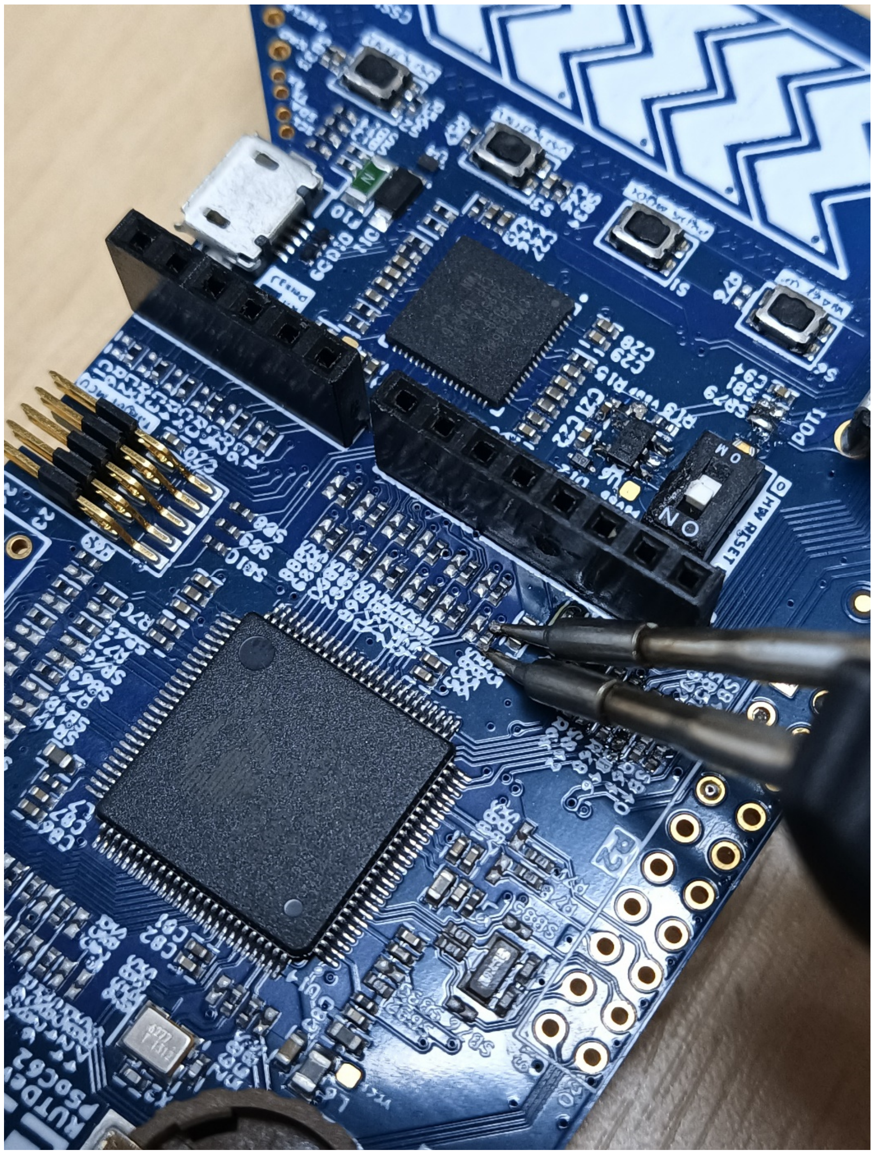

Fig. 25. Soldering the RutDevKit-PSoC62's solder bridge.

Known Issues

Power dropout warning while debugging with KitProg3 and working with I2C.

The power dropout warning appears in the ModusToolbox console window while debugging and working with the I2C interface.

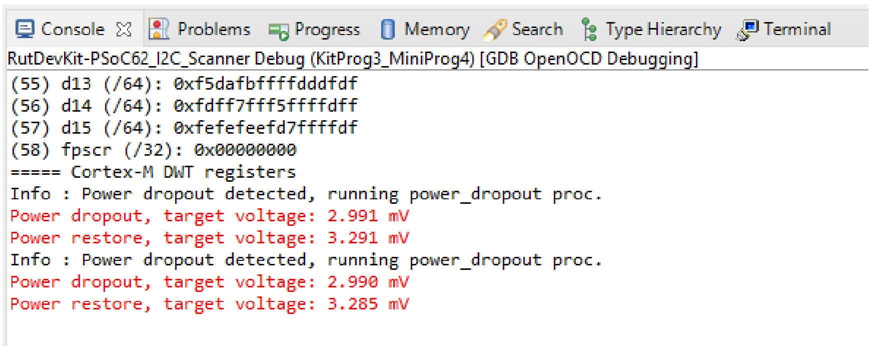

Fig. 26. Power dropout warning in a Console window.

The cause:

The 4.7K pull-up resistors R21 and R24 draw too much current from the voltage-follower/buffer U6 and the KitProg3 target voltage divider gets a false low voltage signal. Actual MCU voltage is not affected and remains stable all the time.

How to fix:

Though this false alarm cannot be disabled from ModusToolbox IDE it can be fixed by changing the pull-up resistors R21 and R24 to much higher values like 18K or could be unsoldered completely if the onboard USB Power Delivery controller CYPD3177 is not used with I2C communications.

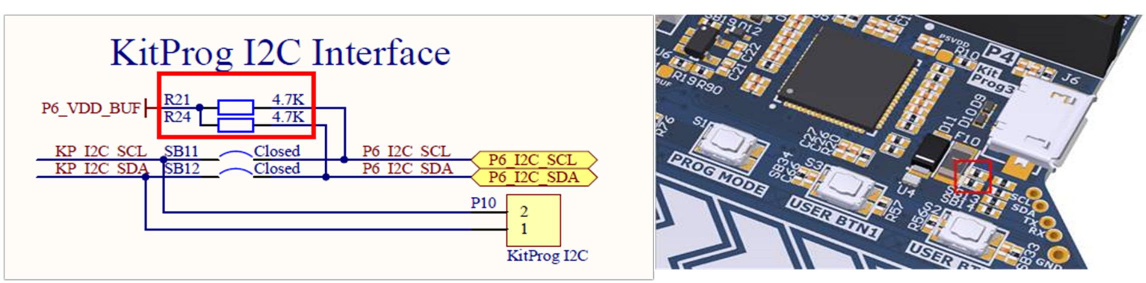

Fig. 27. I2C Pull-up resistors.

Mechanical Layout