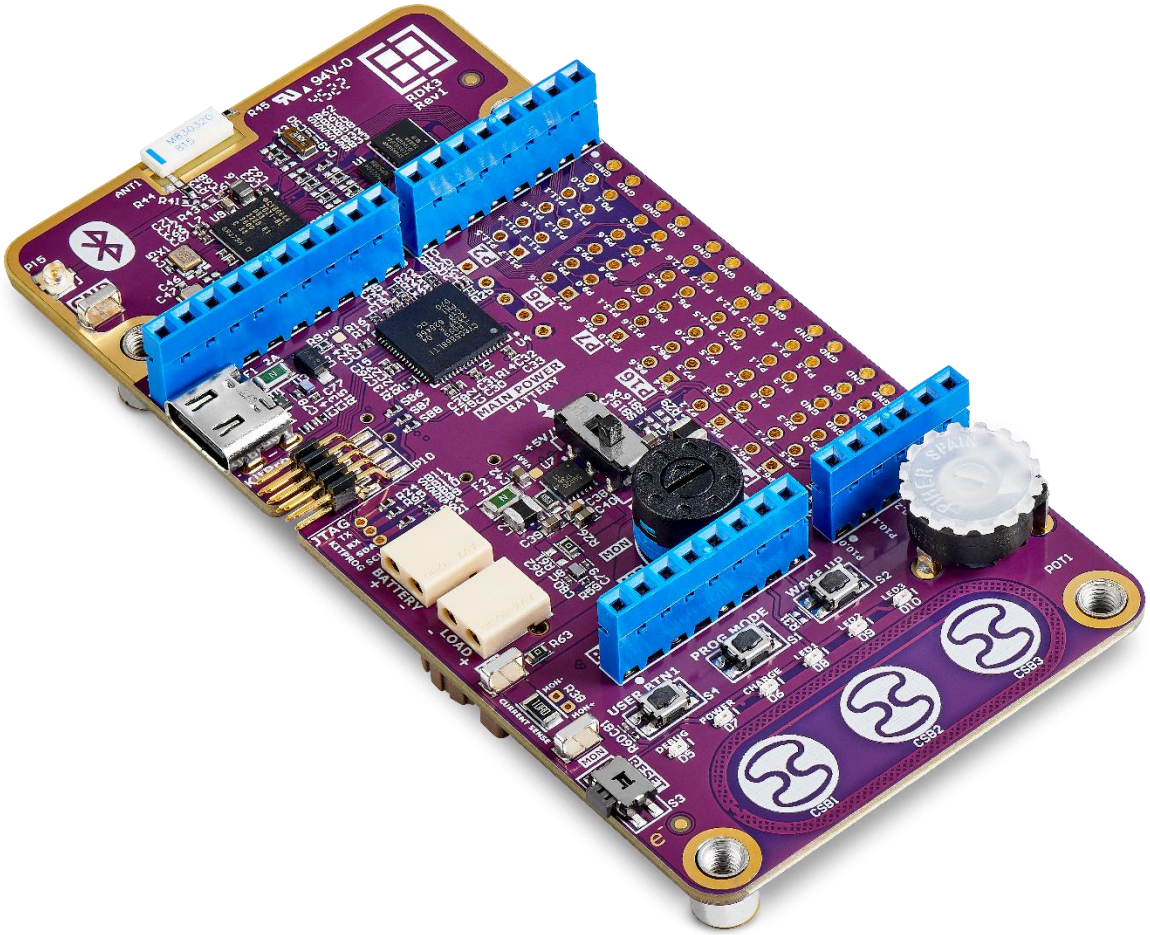
RDK3 User Manual



Versions

Version	Date	Rationale
0.1	November 03, 2022	First draft. Author: GDR
1.0	March 1, 2023	New structure, software and firmware description is added. Autor: KOA
1.1	May 25, 2023	New software workflow. Autor: KOA
1.2	October 20, 2023	Update of Provisioning chapter. Author: KOA, GDR.
1.3	June 04, 2024	Update of Getting Started chapter (changes in secure tools). Author: GDR, KOA.
1.4	October 15, 2024	Update of chapter Running the Project. Autor: KOA
1.5	May 19, 2025	PSoC 64 disclaimer. Author: KOA.

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.

Table of Contents

Overview

Features

The RDK3 is a Bluetooth™ LE 5.0 technology-based development board with enhanced security features that are inherited from PSoC64 family microcontrollers. Key features of RDK3:

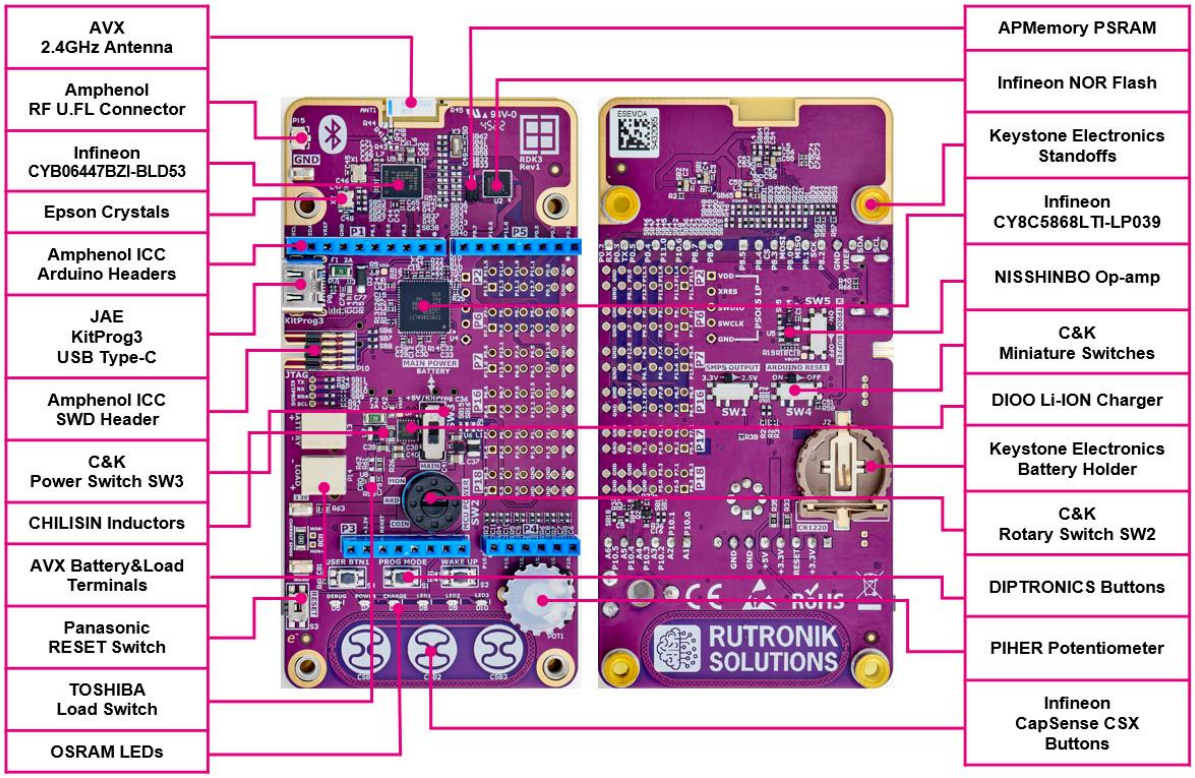
- CYB06447BZI-BLD53 – Infineon's High-Performance, Ultra-Low-Power secured MCU.

Infineon has discontinued the PSOC™ 64 Secured MCU product line. As a result, the CYB06447BZI-BLD53 MCU used in the RDK3 is not recommended for new designs. The Infineon CY8C6347BZI-BLD53 MCU may be considered a suitable alternative.

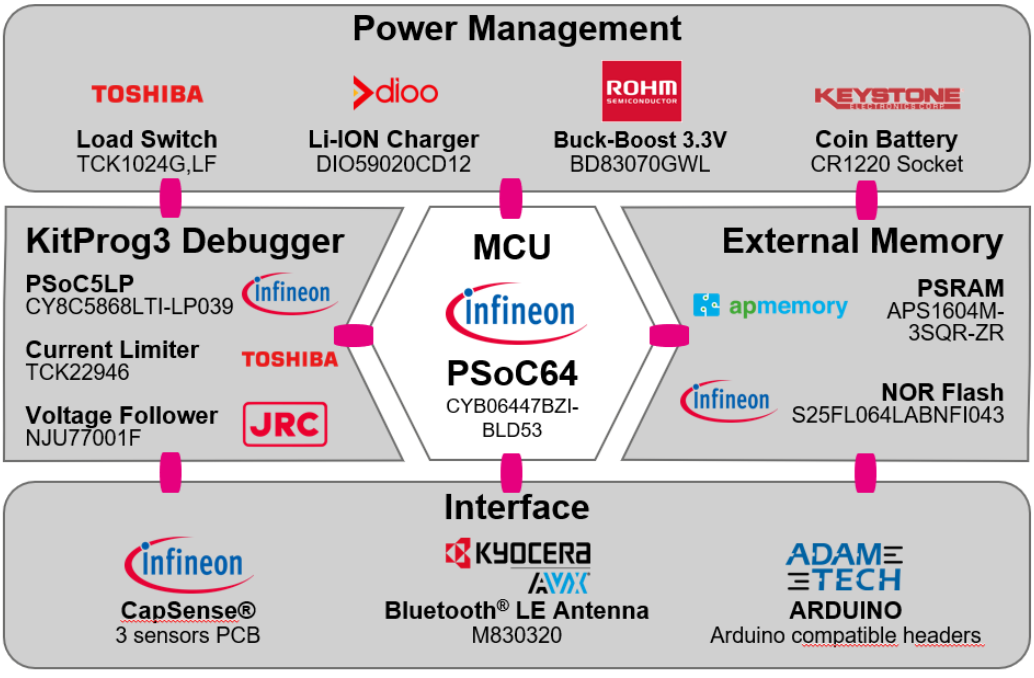
- All CYB06447BZI-BLD53 GPIOs are accessible via onboard headers.
- On-board debugger KitProg3 with I2C and UART USB bridge.
- 10-pin Amphenol ICC SWD header for J-Link.
- JAE USB Type-C connector for the KitProg3 debugger.
- On-board capacitive buttons based on CapSense® CSX technology.
- APS1604M-3SQR-ZR - APMemory External QSPI 16Mbit PSRAM Memory.
- S25FL064LABNFI043 - Infineon External QSPI 64Mbit NOR Flash.
- M830320 - On-board 2.45GHz Bluetooth antenna from AVX.
- U.FL connector for the external Bluetooth antenna from Amphenol RF.
- AVX multilayer ceramic transient voltage suppressors for the USB.
- AVX 9296 series POKE-HOME connectors.
- BD83070GWL - Switching mode power supply from ROHM.
- DIO59020CD12 - Li-ION Battery charger with USB-OTG Boost from DIOO.
- Keystone Electronics Corp. CR1220 coin battery socket for RTC and low-power applications.
- Current monitoring shunt resistor with Keystone Electronics Corp. P/N5019 test points.
- TOSHIBA Load Switches (with the current limiting capability) TCK1024G,LF.
- NISSHINBO low power amplifier NJU77001F.
- DIPTRONICS tactile buttons.
- Panasonic Right-angled tactile switch.
- C&K Rotary and Slider switch for power supply selections.
- PIHER Potentiometer for ADC peripheral evaluation.

- Passive components from Samsung EM, Yageo, and ASJ.
- CHILISIN Power Inductors.

Component Placement



Block Diagram



Delivery Set

The delivery set of RDK3 includes:

- RDK3 development board.
- On-board debugger KitProg3 with I2C and UART USB bridge.
- USB 2.0 Cable A Male to C Male to connect the board to PC.

Applicable Boards

The following Rutronik System Solution boards are compatible with RDK3 and can be connected to it to provide the additional functionality.

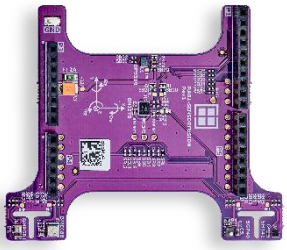
CO2

Evaluation of CO_2, relative humidity and temperature sensors provided by Infineon and Sensirion. More details see here.

Text to Speech

Testing Epson Text to Speech solution.
More details see here.

Sensor Fusion



Evaluation of sensors (pressure, indoor air quality, temperature, humidity, accelerometer and gyroscope) provided by Infineon, Sensirion and Bosch.
More details see here.

Hardware

Microcontroller

PSoC® 6 is an MCU platform built specially for Internet of Things applications. The PSoC 64 product line also provides out-of-box security functionality, including an isolated root-of-trust with true attestation and provisioning services.

Its 32-bit dual CPU includes:

- the main 150-MHz Arm® Cortex®-M4F CPU with single-cycle multiply, floating point, and memory protection unit (MPU);
- the secondary 100-MHz Cortex-M0+ CPU with single-cycle multiply and MPU [cannot be used by the user].

The memory includes:

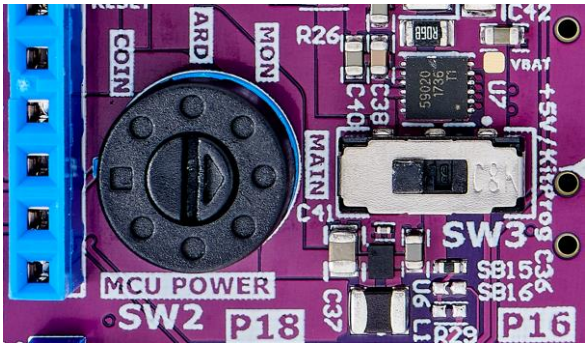
- 832 KB application flash, 32 KB auxiliary flash, and 32 KB supervisory flash; read-while-write support. Two 8 KB flash caches (one for each CPU).
- 176 KB SRAM with power and data retention control.
- One-time-programmable 1 Kb eFuse array.

CapSense is supported in PSoC 6 MCU through a CapSense sigma-delta (CSD) hardware block. It provides best-in-class signal-to-noise ratio, liquid tolerance, and proximity sensing. It enables dynamic usage of both self and mutual sensing.

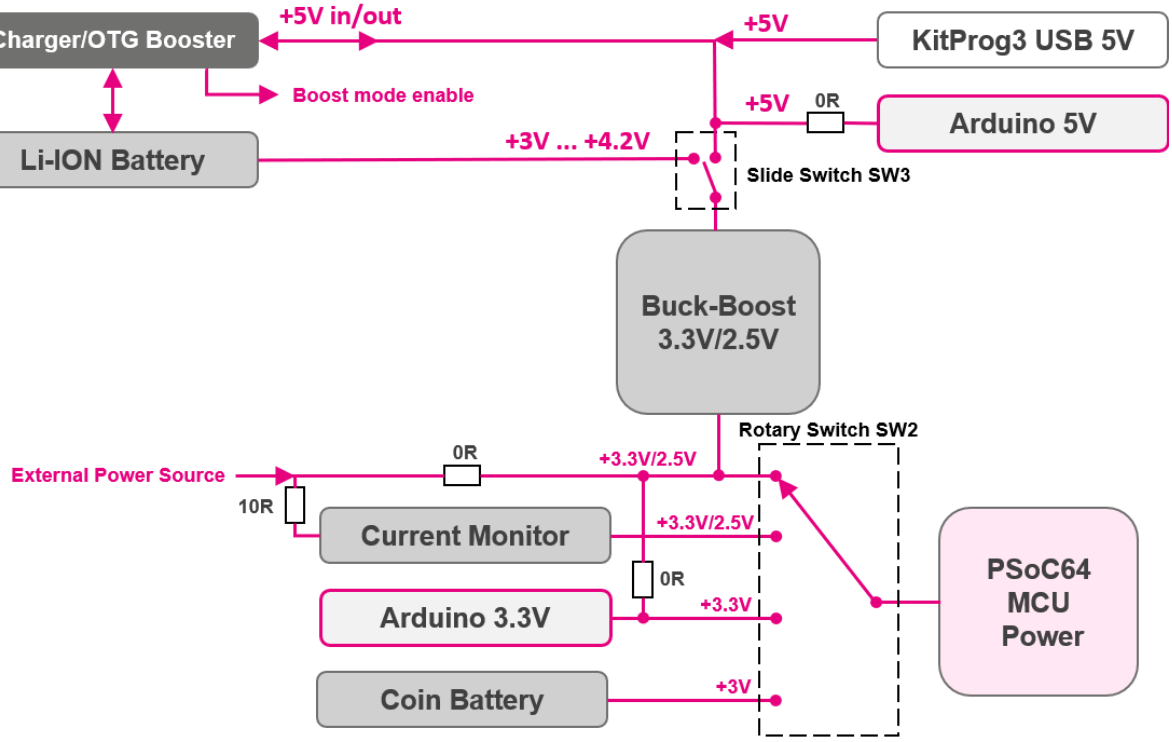
Power Sources

There are five ways to provide power for the MCU possible in RDK3:

1. KitProg3 USB Type-C port.
2. CR1220 coin battery socket.
3. Arduino connectors – configured using R35 and R33 0R 0603 resistors.
4. Li-ion Battery.
5. Current monitor TP6, only if R38 and R63 are removed.



Select one of the power sources using SW2 - the Coin Battery "**COIN**", Arduino headers +5V "**ARD**", Current Monitor Terminals "**MON**", 3.3V SMPS "**MAIN**". With SW3 users can select the power source as **BATT** – Li-ION battery or +5V power rail.



Power Distribution Diagram

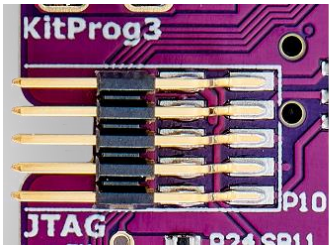
The Battery Charger

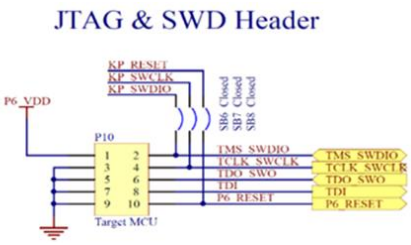
The RDK3 has an onboard battery charger DIO59020. The Single Cell Li-Ion, Li-Polymer switching charger DIO59020 is capable of 2A charging from a 5V power supply with high efficiency and can be controlled via I2C. The input current limit, battery charging current, battery charging termination current, and configuration of the pins are programmable. There are I2C access-enabled registers that provide information about the charging status and enable a charging process monitoring. The safety features such as overheat protection of the IC and the over-discharge of the battery detection are also included.

The DIO59020 battery charger also has a USB-OTG Boost Regulator. This feature may provide a 5V 1A power source that feeds the Arduino 5V power rail from the single Lithium-Ion cell. The Boost Mode of the DIO59020 needs to be activated by sending an I2C command or setting the OTG pin to high logic level.

Programming Using External Connector

Users may use third-party programming devices to connect the CYB06447BZI-BLD53 target via the P10 SWD connector. The onboard “KitProg3” debugger should not be powered while using an external JTAG connector.





Spare GPIOs

All GPIOs of CYB06447BZI-BLD53 MCU are available at sockets P2, P6, P7, P16, P17, P18. Some may need to be configured using solder bridges.

Socket P2 Pinout				Socket P6 Pinout			
Pin No.	**Name**	**Name**	**Pin No.**	**Pin No.**	**Name**	**Name**	**Pin No.**
1	P11.7	P11.5	2	1	P7.7	P7.6	2
3	P11.3	P11.4	4	3	P9.0	P9.1	4
5	P11.2	P11.6	6	5	P9.4	P9.5	6
7	P13.7	P11.1	8	7	P9.2	P9.6	8
9	P0.1	P0.0	10	9	P9.7	P9.3	10
11	GND	GND	12	11	GND	GND	12

Socket P7 Pinout				Socket P16 Pinout			
Pin No.	**Name**	**Name**	**Pin No.**	**Pin No.**	**Name**	**Name**	**Pin No.**
1	P13.0	P5.6	2	1	P6.4	P6.6	2
3	P13.6	P13.1	4	3	P7.3	P6.7	4
5	P6.0	P7.4	6	5	P7.5	P7.0	6
7	P6.1	P6.5	8	7	P12.2	P12.5	8
9	P12.7	P12.6	10	9	P12.3	P12.4	10
11	GND	GND	12	11	GND	GND	12

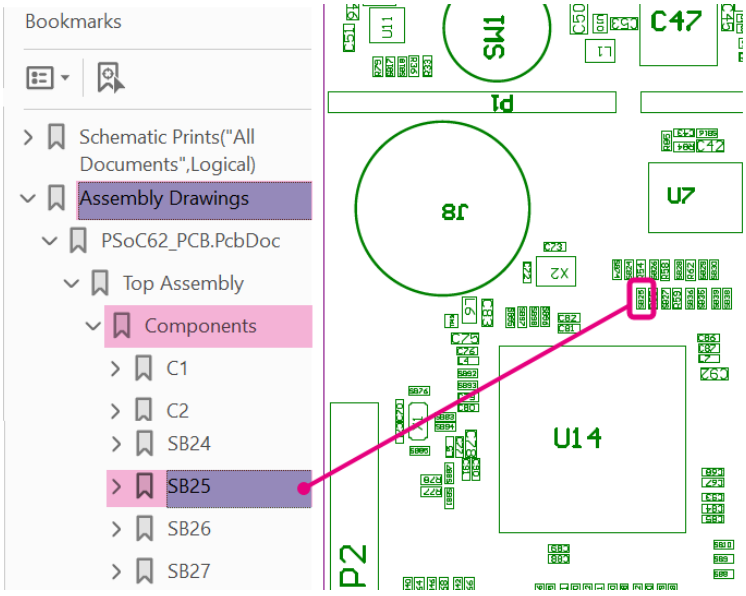
Socket P17 Pinout				Socket P18 Pinout			
Pin No.	**Name**	**Name**	**Pin No.**	**Pin No.**	**Name**	**Name**	**Pin No.**
1	P5.4	P7.2	2	1	P6.3	P6.2	2
3	P5.2	P5.1	4	3	P5.5	P7.1	4
5	P1.0	P1.2	6	5	P5.0	P5.3	6
7	P1.3	P1.1	8	7	GND	GND	8
9	P1.5	P1.4	10	9	GND	GND	10
11	GND	GND	12	11	GND	GND	12

Solder Bridges

Name	Circuit	Default
SB1	+3.3V Supply for APS1604M-3SQR-ZR	Closed
SB2	+3.3V Supply for S25FL064LABNFI043	Closed
SB3	P6_VDD_BUF Supply for Potentiometer	Closed
SB4	Potentiometer output with ADC5 (P10.4)	Closed
SB5	P6_VDD NJU77001F (U5) Input+	Closed
SB6	KitProg3 SWDIO with MCU SWDIO	Closed
SB7	KitProg3 SWCLK with MCU SWCLK	Closed
SB8	KitProg3 RESET with MCU RESET	Closed
SB9	KitProg3 I2C SCL with MCU I2C SCL	Closed
SB10	KitProg3 I2C SDA with MCU I2C SDA	Closed
SB11	KitProg3 UART TX with MCU UART RX	Closed
SB12	KitProg3 UART RX with MCU UART TX	Closed
SB13	MCU I2C SDA with Charger (U7) I2C SDA	Closed
SB14	MCU I2C SCL with Charger (U7) I2C SCL	Closed
SB15	SMPS Power Input with SMPS EN pin	Closed
SB16	MCU P0.5 with SMPS EN pin	Opened
SB17	Battery Voltage Divider Control Input	Closed
SB18	Battery Voltage Divider Output with P10.5	Closed
SB19	P6.0 with P7 header GPIO 5	Opened
SB20	KitProg3 UART TX with MCU P6.0	Closed
SB21	P6.1 with P7 header GPIO 7	Opened
SB22	KitProg3 UART RX with MCU P6.1	Closed
SB23	P6.4 with P16 header GPIO 1	Opened
SB24	JTAG TDO with MCU P6.4	Closed
SB25	P6.5 with P7 header GPIO 8	Opened
SB26	JTAG TDI with MCU P6.5	Closed
SB27	P6.6 with P16 header GPIO 2	Opened
SB28	JTAG TMS/SWDIO with MCU P6.6	Closed
SB29	P6.7 with P16 header GPIO 4	Opened
SB30	JTAG TCLK/SWCLK with MCU P6.7	Closed
SB31	P7.0 with P18 header GPIO 4	Opened
SB32	P11.1 with P2 header GPIO 8	Opened
SB33	Flash QSPI SSEL with MCU P11.1	Closed
SB34	P11.2 with P2 header GPIO 5	Opened
SB35	PSRAM QSPI SSEL with MCU P11.2	Closed
SB36	P7.3 with P16 header GPIO 3	Opened
SB37	P11.3 with P2 header GPIO 3	Opened
SB38	P11.4 with P2 header GPIO 4	Opened
SB39	P7.4 with P7 header GPIO 6	Opened
SB40	P11.5 with P2 header GPIO 2	Opened
SB41	P11.6 with P2 header GPIO 6	Opened
SB42	P7.5 with P16 header GPIO 5	Opened

SB43	P11.7 with P2 header GPIO 1	Opened
SB44	P12.2 with P16 header GPIO 7	Opened
SB45	Load Switch Control EN with P12.2	Closed
SB46	P12.3 with P16 header GPIO 9	Opened
SB47	Battery Voltage Divider Input with P12.3	Closed
SB48	P12.4 with P16 header GPIO 10	Opened
SB49	Charger DISABLE pin with P12.4	Closed
SB50	P12.5 with P16 header GPIO 8	Opened
SB51	Charger BOOST EN with P12.5	Closed
SB52	USER LED1 with P13.1	Closed
SB53	P12.6 with P7 header GPIO 10	Opened
SB54	USER LED2 with P13.6	Closed
SB55	X2 pin 3 with MCU P12.6	Closed
SB56	X2 pin 1 with MCU P12.7	Closed
SB57	P12.7 with P7 header GPIO 9	Opened
SB58	P0.0 with P2 header GPIO 10	Opened
SB59	USER LED3 with P13.7	Closed
SB60	X3 pin 2 with P0.0	Closed
SB61	X3 pin 1 with P0.1	Closed
SB62	P0.1 with P2 header GPIO 9	Opened
SB63	P6_VDD with MCU VBACKUP pin	Closed
SB64	VCOIN with MCU VBACKUP pin	Opened
SB65	P6_VDD with MCU VREF pin	Closed
SB66	USER BUTTON with MCU P13.0 pin	Closed

The locations of the solder bridges can be found in 3D model and assembly drawings of RDK3.



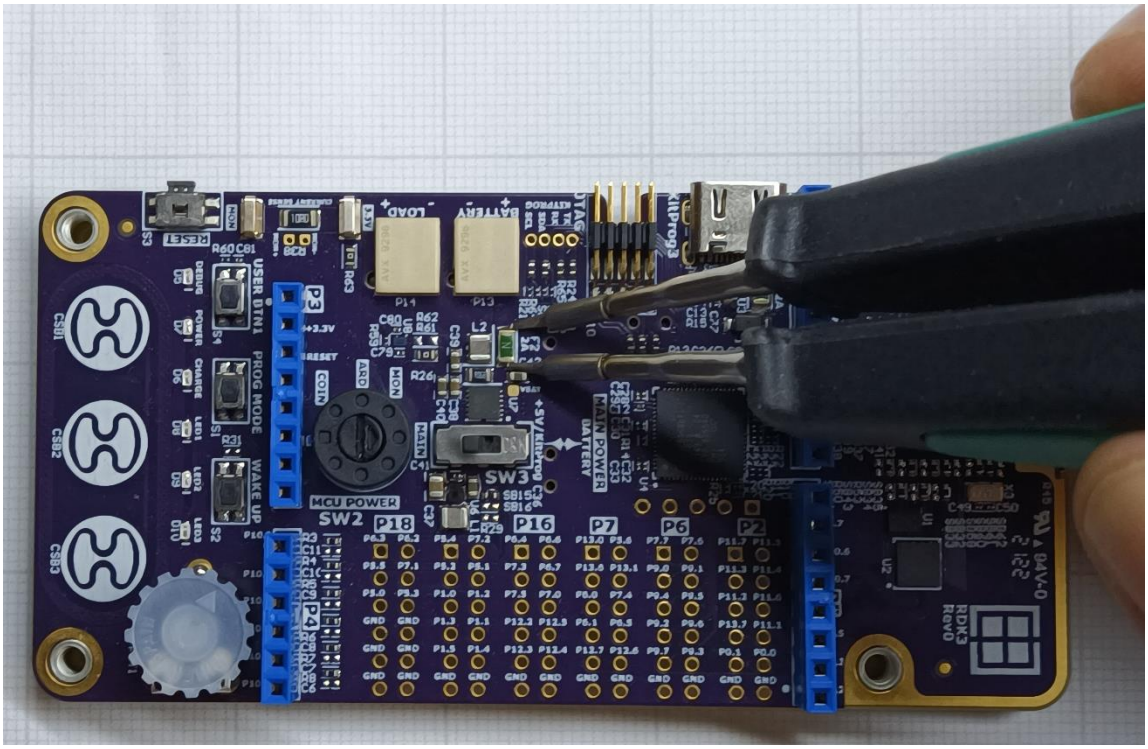
How to find a component on the layout

Fuses

The RDK3 board has two 2A fast-acting fuses F1 and F2 in a 1206 package; part No: CC12H2A-TR „Eaton“.

Changing the Fuses or Solder Bridges

The SMD „Chipping Tool“ is recommended to use for SMD solder bridges or fuses soldering on the RDK3 development board.



Soldering the RDK3’s fuse

Insertion and Extraction of Wire from AVX 9296 Connectors

The RDK3 board has two AVX 9296 2-pin connectors for the Li-ion battery and load connection (P13 and P14). The 20/22/24/26AWG wires are recommended to be striped from 3.5mm to 4.5mm before insertion. Once inserted it can be extracted without any tools. Gently rotate the wire while pulling until the extraction is complete. Please refer to the application note 201-01-167 provided by the AVX for more detailed information.

Bluetooth® SIG Qualification

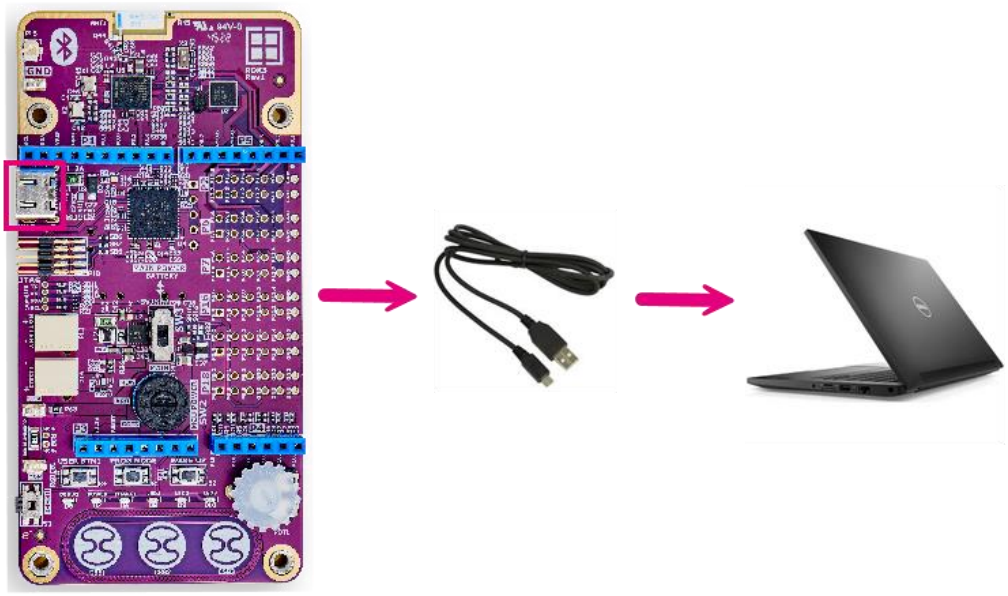
The RDK3 board has been qualified with Bluetooth® SIG. The Declaration ID of the RDK3 is D061890 and it is referenced to the qualified design QDID: 99158.

Software and Firmware

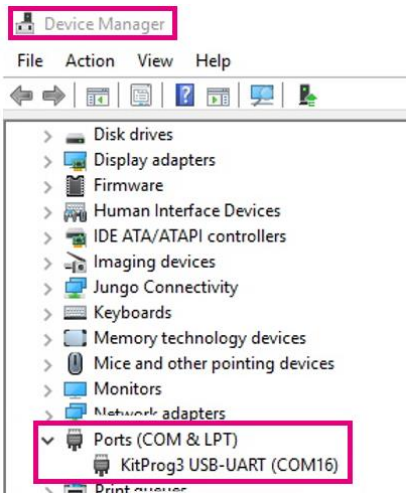
Getting Started

1. Register or/and login at Infineon website (myInfineon tab). License generation takes up to several days.
2. Download and install the latest version of ModusToolbox™ software.
3. For ModusToolbox™ v3.2, please install the Python for Windows OS.

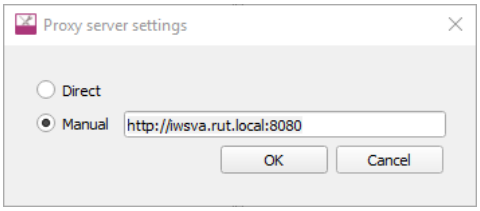
   If you installed the software in a non-standard way, make sure the Python installation is added to Path environment variable.
4. Use command `pip install cysecuretools` in Command Prompt to install the cysecuretools package.
5. Unzip the secure policy configurator tool to your ModusToolbox tools_3.2 directory (for example, C:\Users\userX\ModusToolbox\tools_3.2).
6. *[Optional]* Download and install your preferred terminal emulator, for example: PuTTY, Tera Term, etc.
7. Connect your board (micro USB socket with a marking “KitProg3”) and a PC via USB Type-C cable.



8. Check if RDK3 is ready. Its “POWER” and “DEBUG” LEDs should shine constantly. The “CHARGE” LED will be blinking if no battery is connected.
9. The “KitProg3” port must be seen in MS Windows Device Manager window.



10. *[For Rutronik laptops only]* Run **File** – **New** – **ModusToolbox Application** – **Settings** – **Proxy server settings** and enter the proxy address: http://iwsva.rut.local:8080



If you still have no Internet connection, please set the environment variables as shown here: "ModusToolbox Project Creator and Library Manager report no Internet connection when accessing manifest files ".

Provisioning of RDK3

The RDK3 is equipped with a PSoC™ 64 "Secure" MCU CYB06447BZI-BLD53. The PSoC™64 device must be provisioned with keys and policies before being programmed. If the unsigned or not properly signed image will be written to the RDK3 PSoC™64 – the microcontroller will not startup.

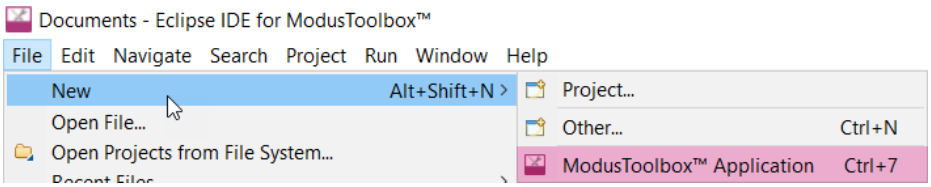
You may also refer to:

"Secure Policy" Configurator guide

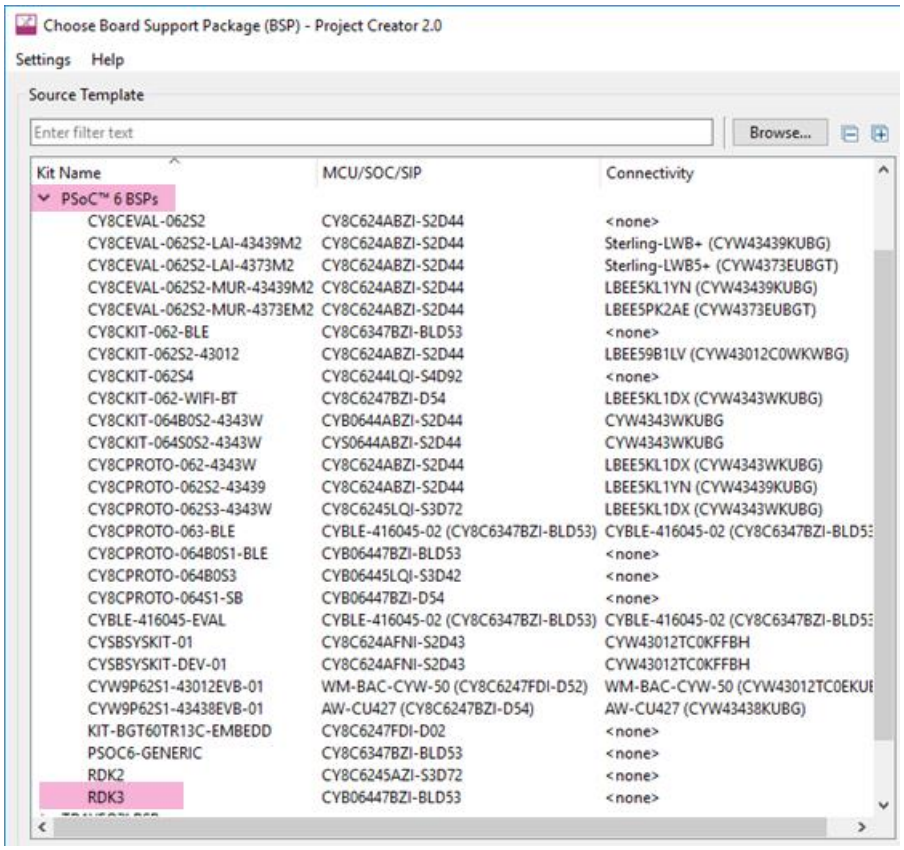
PSoC™ 64 - Secured MCU

PSoC™ 64 Provisioning Specification

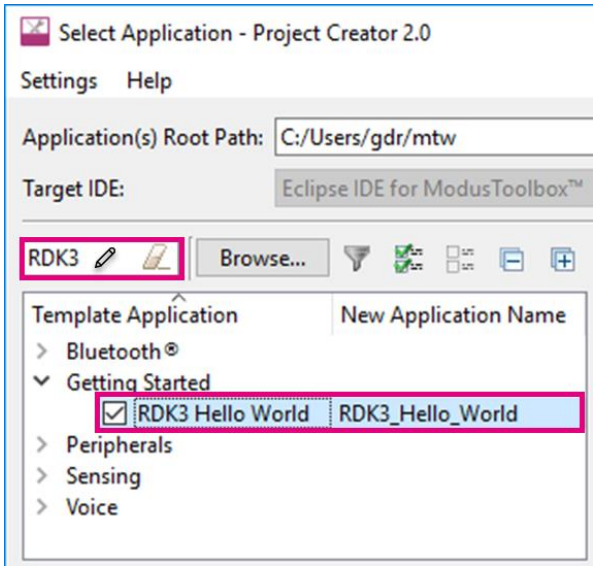
1. Select **File** – **New** – **ModusToolbox Application**.



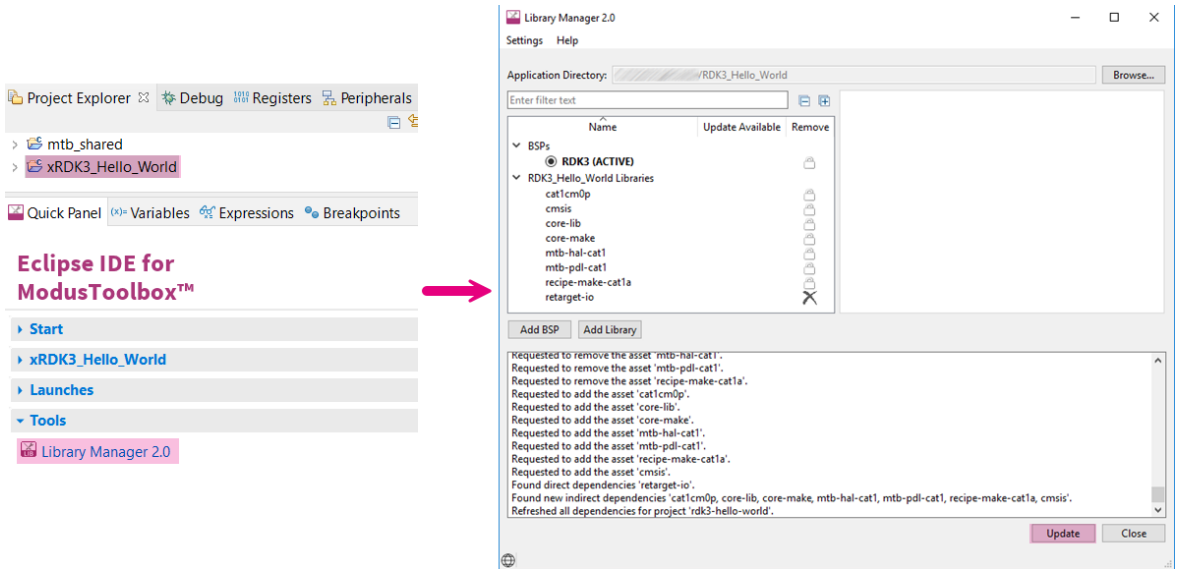
2. Select the **RDK3** BSP. It is in **PSoC™ 6 BSPs** list, press **Next** after that.



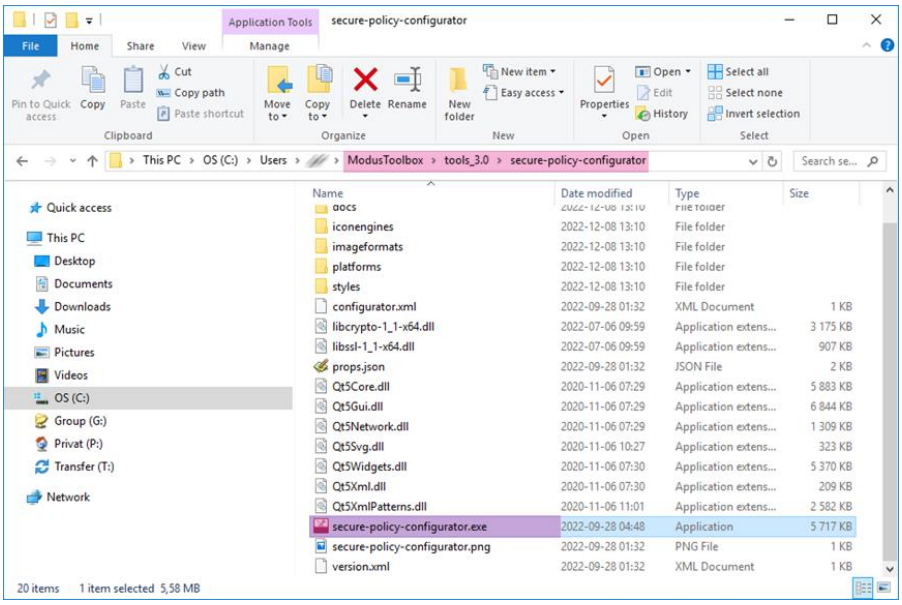
3. Type *RDK3* in the search field. Select the example from the list. Press **Create**. Wait until the process is finished.



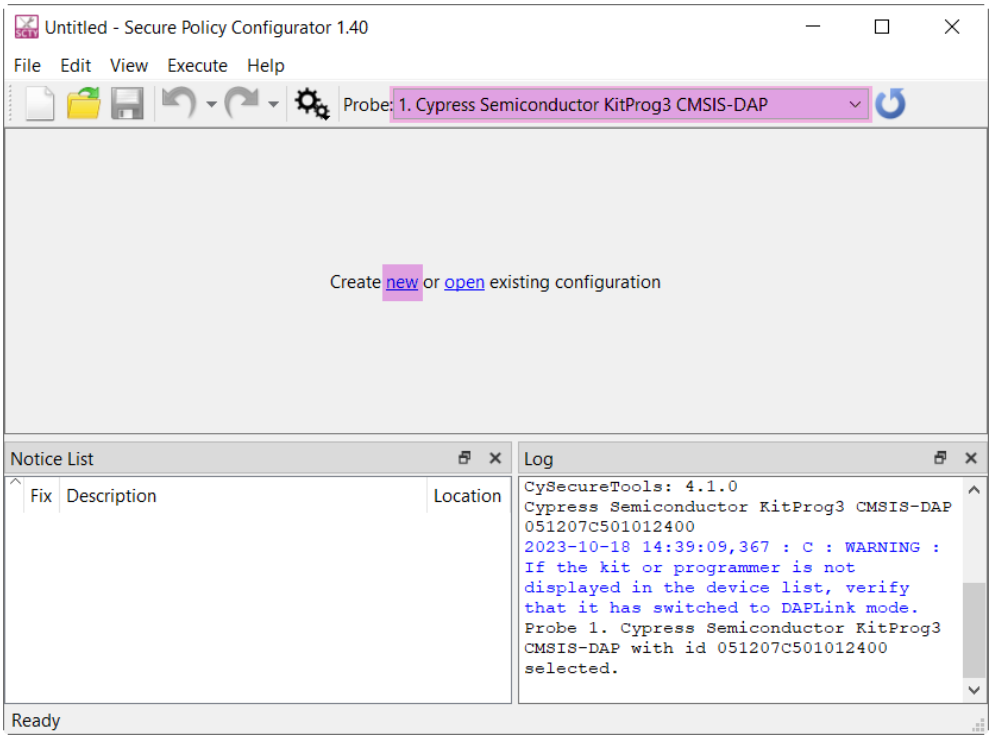
4. Update the libraries using **Library Manager** (optional).



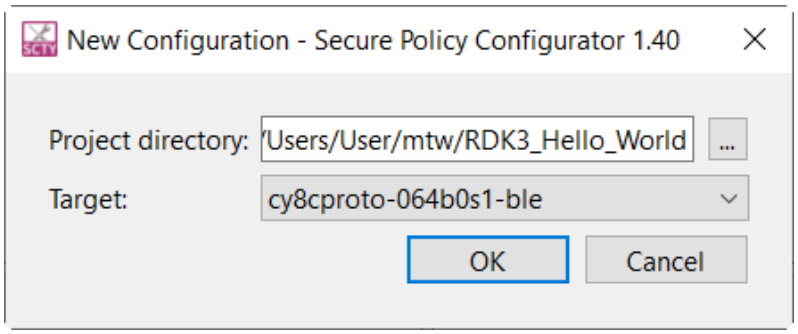
5. Run **Secure Policy Configurator**. It can be found in ModusToolbox installation folder, for example here:
…\ModusToolbox\tools_3.1\secure-policy-configurator



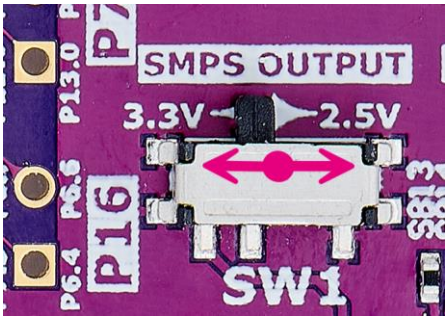
6. Select the probe **Cypress Semiconductor KitProg3 CMSIS-DAP**. Press the PROG MODE button if the CMSIS-DAP is not present in a list.



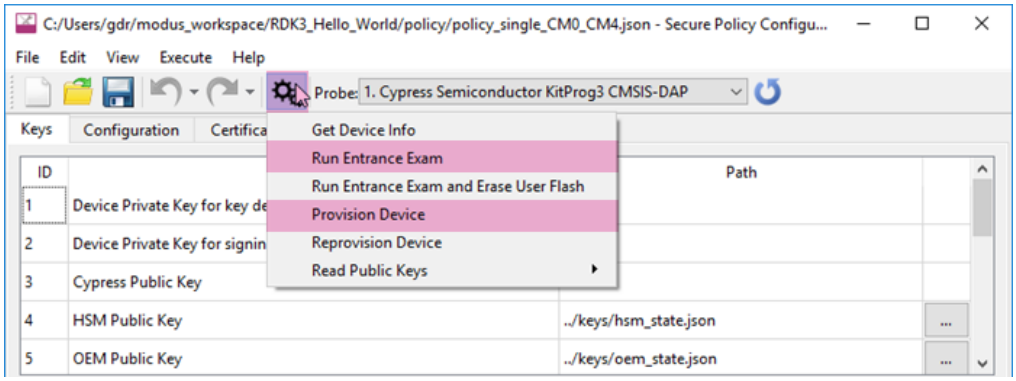
7. Create new configuration, select the *RDK3_Hello_World* **Project directory** and the specified **Target: cy8cproto-064b0s1-ble**.



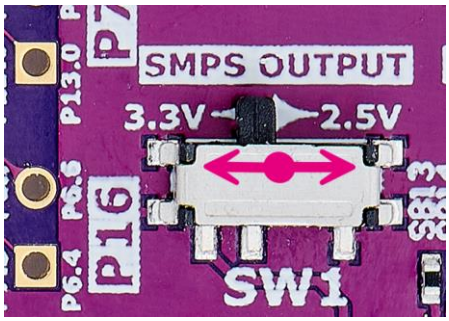
8. Set the SW1 “SMPS OUTPUT” to 2.5V position.



9. Press **Run Entrance Exam** and if you have a PASS, then **Provision Device**.

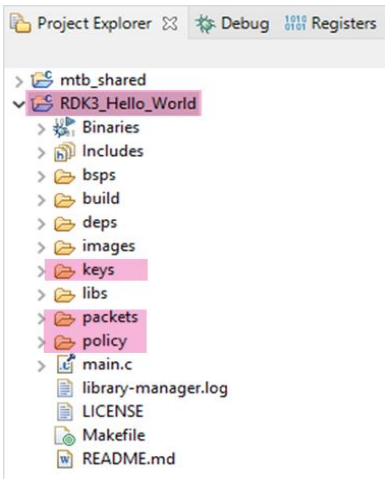


10. Set the SW1 “SMPS OUTPUT” back to the 3.3V position.



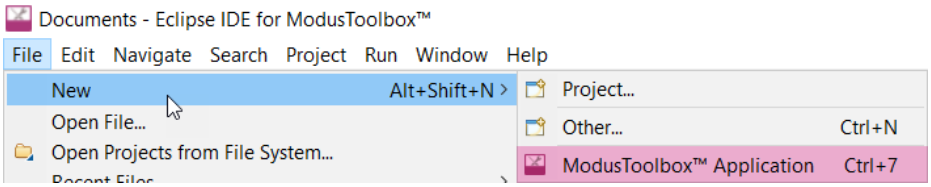
The **keys**, **packets**, and **policy** configurations are created in your project directory. Please store them safely for use in other projects with your RDK3. Also, you might need the „packets“ files for the re-provisioning procedure.

Please note that you may do the provisioning for all the new RDK3s you have using the same keys and policies that you have created with the first one.

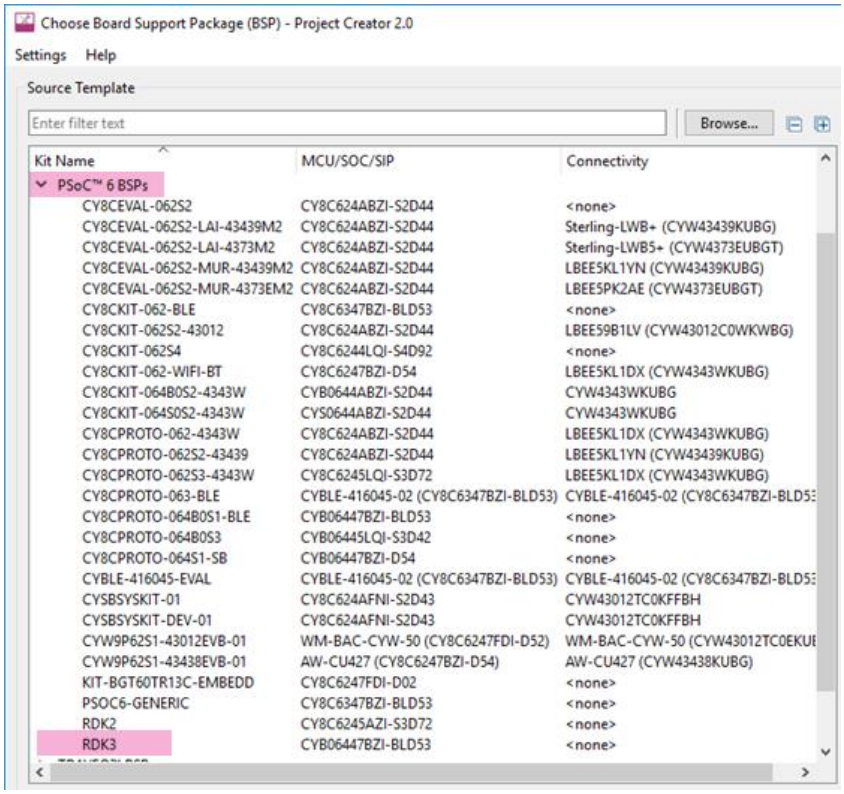


Creating New Project

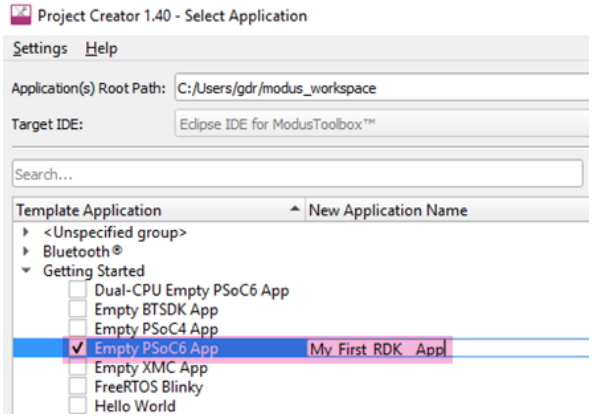
1. Select **File** – **New** – **ModusToolbox Application**.



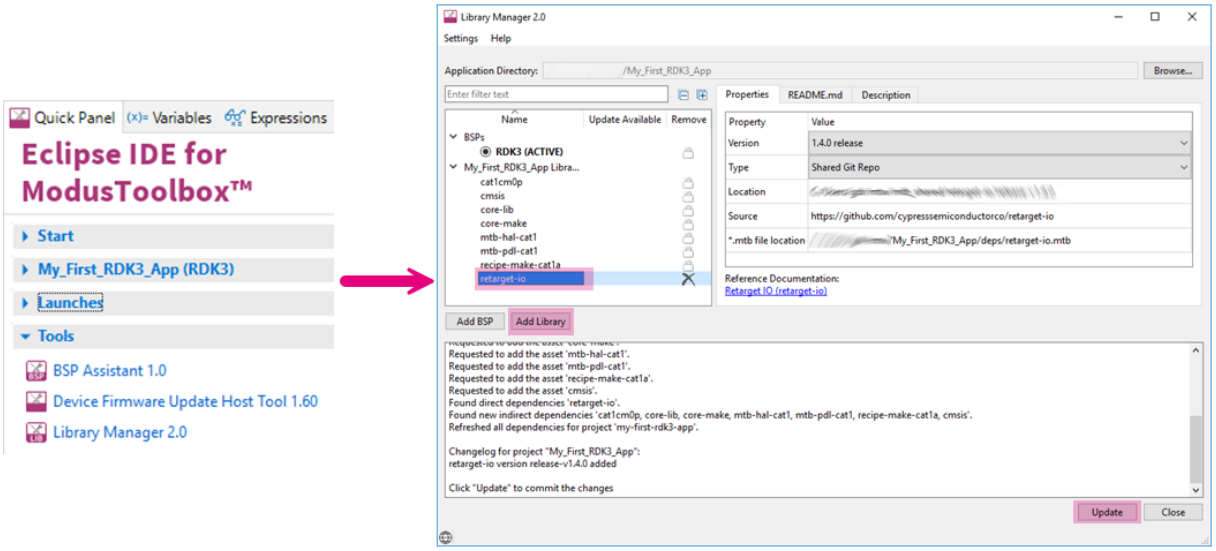
2. Select the **RDK3** BSP. It is in **PSoC™ 6 BSPs** list, press **Next** after that.



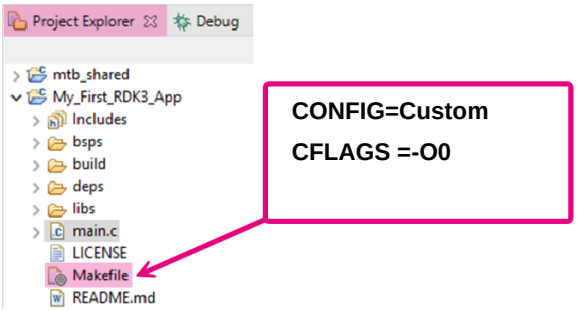
3. Open **Getting Started** block, check **Empty PSoC6 App**, insert the **New Application Name** and click **Create**. Wait for a while until project creation is finished.



4. Open **Library Manager**, select “retarget-io” library, press **Add Library** and then **Update**.

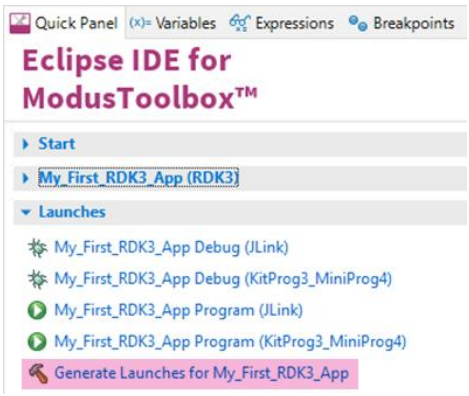


5. Modify Makefile as it's shown below to disable the code optimisation*.

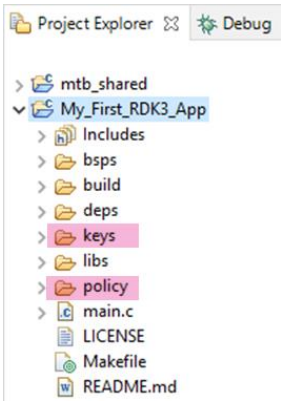


*It is required only for debugging, learning and demo purposes. Normally, code optimisations should never be disabled.

6. Press **Generate Launches for <project name>** in **Quick Panel.**



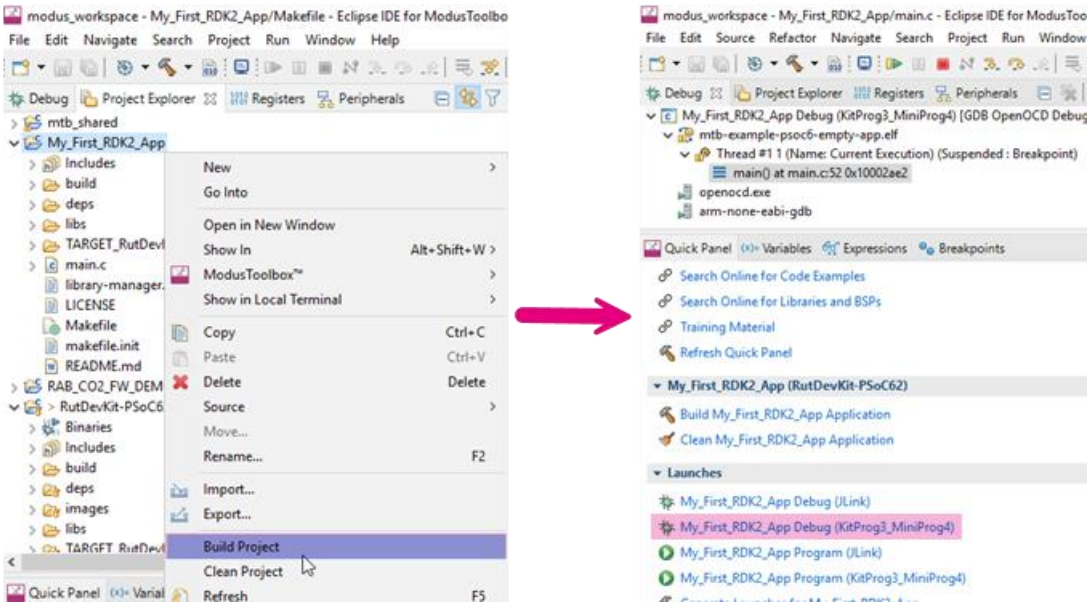
7. Copy and paste the **keys** and **policy** folders with all the files into your project.



8. Copy, paste and save the code example from this file to the “main.c” file.

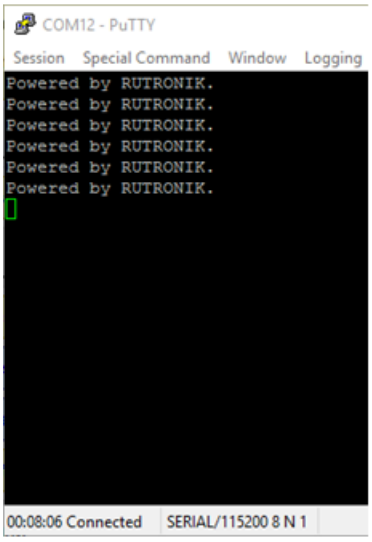
```
#include "cy_pdl.h"
#include "cyhal.h"
#include "cybsp.h"
#include "cy_retarget_io.h"
int main(void)
{
    cy_rslt_t result;
    /* Initialize the device and board peripherals */
    result = cybsp_init();
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }
    __enable_irq();
    /*Initialize LED2*/
    result   =   cyhal_gpio_init(   LED2,   CYHAL_GPIO_DIR_OUTPUT,   CYHAL_GPIO_DRIVE_STRONG,
CYBSP_LED_STATE_OFF);
    if (result != CY_RSLT_SUCCESS)
    {CY_ASSERT(0);}
    /*Enable debug output via KitProg UART*/
    result = cy_retarget_io_init( KITPROG_TX, KITPROG_RX, CY_RETARGET_IO_BAUDRATE);
    if (result != CY_RSLT_SUCCESS)
    {CY_ASSERT(0);}
    printf("\x1b[2J\x1b[;H");
    for (;;)
    {
    /*Delay 1000 milliseconds*/
    CyDelay(1000);
    printf("Powered by RUTRONIK.\r\n");
    cyhal_gpio_toggle(LED2);
    }
}
```

9. **Build** and **Debug** the active project.

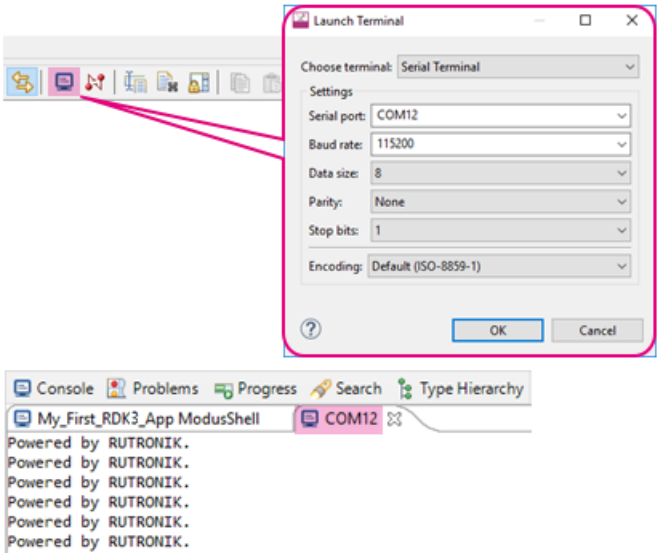


The result is a blinking LED2 on the RDK3 board and text on the terminal window.

PuTTY Terminal

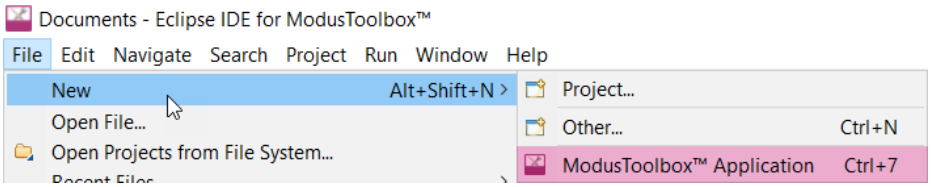


ModusToolbox Terminal

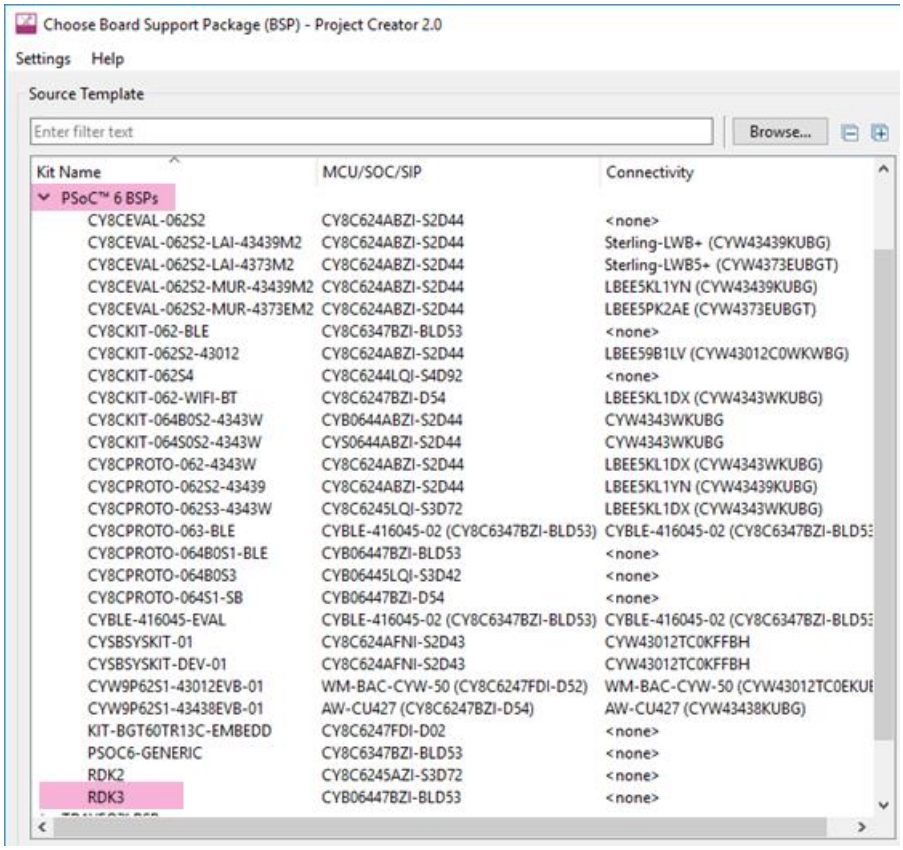


Running/Updating the Existing Project

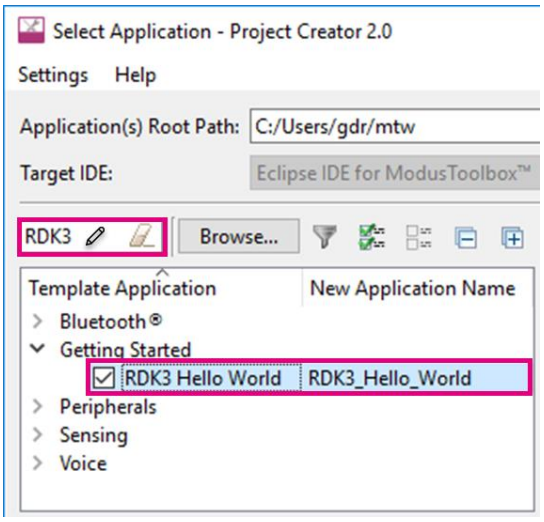
1. Select **File** – **New** – **ModusToolbox Application**.



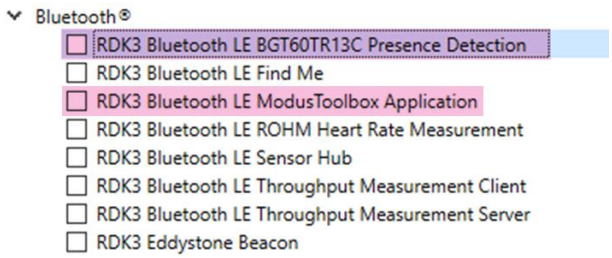
2. Select the **RDK3** BSP. It is in **PSoC™ 6 BSPs** list, press **Next** after that.



3. Type *RDK3* in the search field. Select the example from the list.



To run/update the firmware for testing adapter boards together with RDK3 in Android and iOS apps, open the **Bluetooth** group and select:

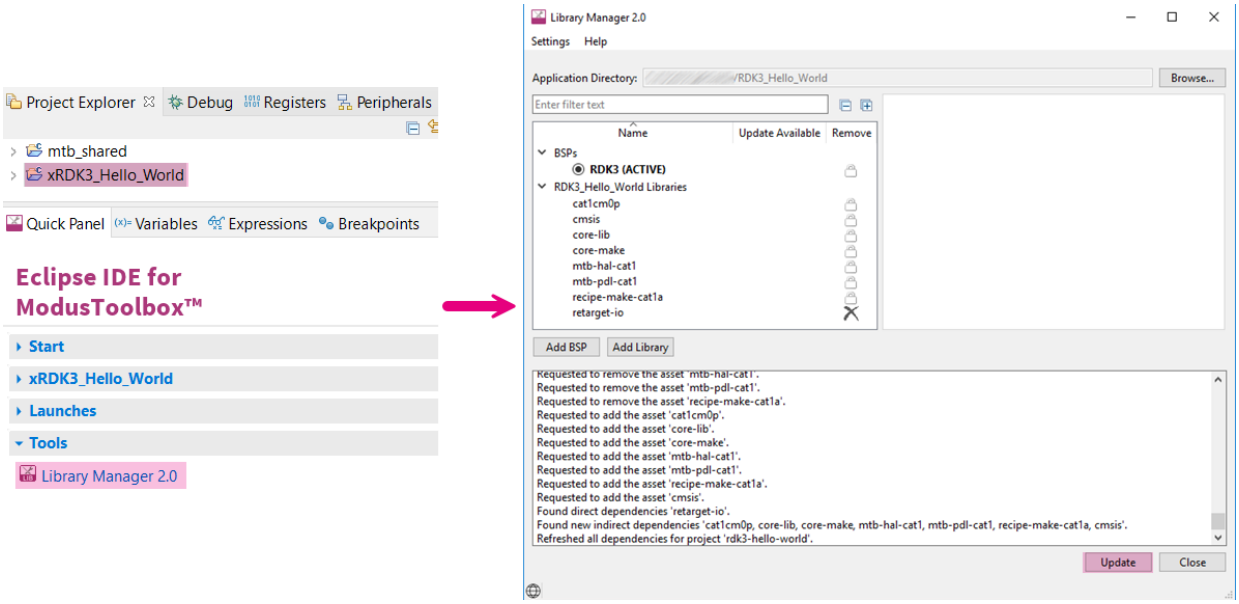


RDK3 … Presence Detection – for Radar board (GitHub link). Stored separately because of a big stack.

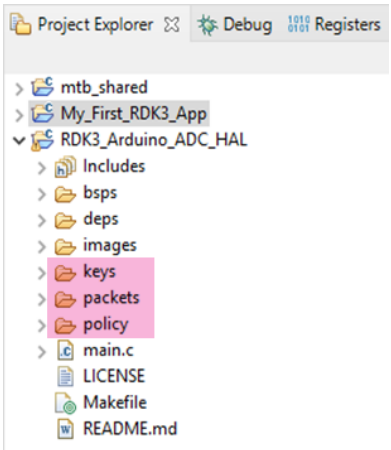
RDK3 … ModusToolbox Application – for Sensor Fusion/ CO2/ AMS OSRAM/ Vishay VCNL4030X01/ RTK boards (GitHub link).

4. Press **Create**.

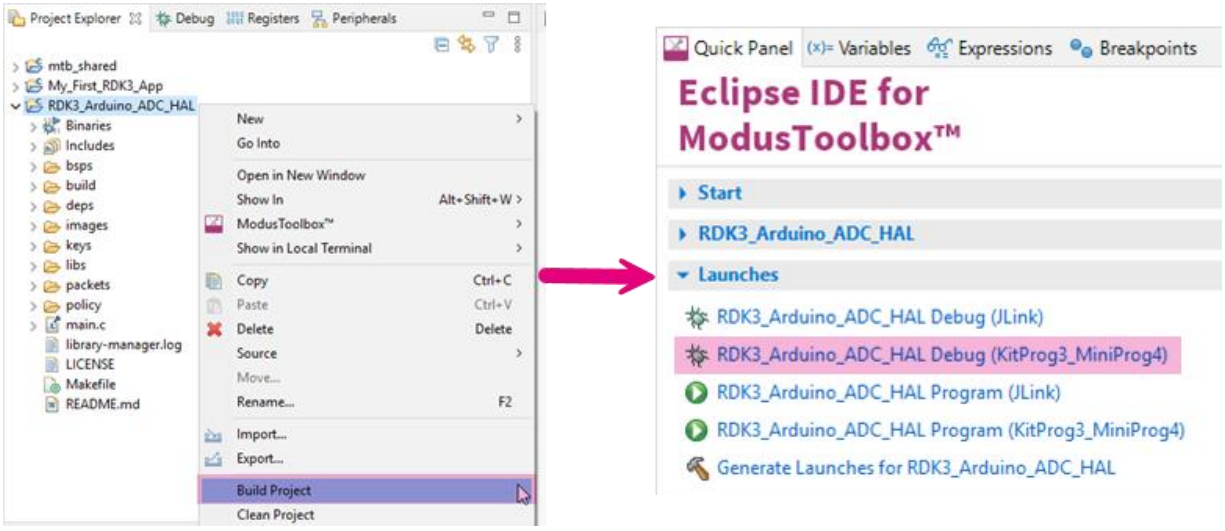
5. Update the libraries using **Library Manager**.



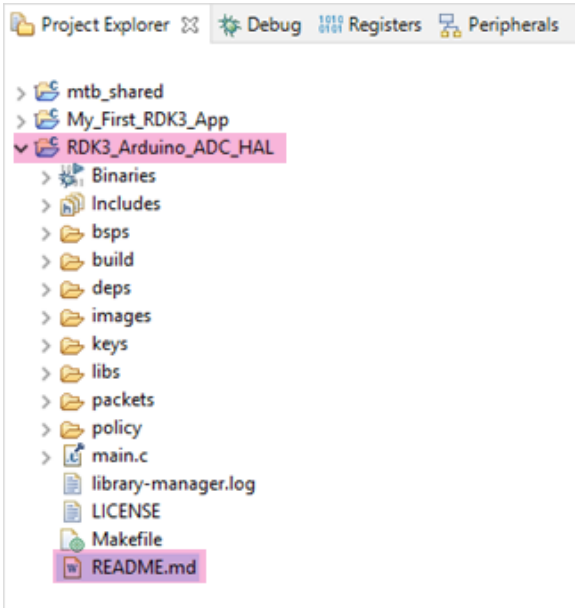
6. Copy and paste the **keys**, **policy** and **packets** folders with all the files into your project.



7. Select the project. Build and debug it.



Check README.md file before starting to explore the code example. You may find important hints and other information that are needed to have firmware running properly.



Firmware Examples

All these examples can be found at GitHub.

RDK3_SensorStation2	This is a demonstration of the sensors MLX90640 [thermal camera], VCNL4035X01 [gesture control], and the smart display GEN4-ULCD-35P4CT-CLB.
RDK3_Radar_Presence_Detection	This code example demostrates Infineon XENSIV™ Presence Detection algorithm in 2 ways: directly on the console (using the KitPro3UART) or over BLE in the Android/ iOS apps.

RDK3_ModusToolbox_App	This software enables to collect the data from sensor adapter boards (SensorFusion, CO2, AMS OSRAM ToF, RTK) and to send them over Bluetooth LE to a connected client.
RDK3_BLE_Throughput_ Measurement_Server	This is a code example for the BLE 5.1 Throughput Measurement GATT server device.
RDK3_BLE_Throughput_ Measurement_Client	This is a code example for the BLE 5.1 Throughput Measurement client device.
RDK3_SensorFusionAdapter_ Demo	This project demonstrates how RAB1-SENSORFUSION adapter board works together with RDK3.
RDK3_RAB2_CO2_ Adapter_Demo	This project demonstrates how RAB2-CO2 adapter board works together with RDK3.
RDK3_QSPI_PSRAM_ DYN_ALLOC	This example demonstrates how to configure and use APS1604M-3SQR-ZR PSRAM with standard dynamic memory allocation functions such as malloc() etc.
RDK3_BLE_ROHM_Heart_Rate	This code example demonstrates a BLE Heart Rate Service with a BH1792 optical heart rate sensor.
RDK3_TextToSpeech_Demo	This project demonstrates how TEXT-TO-SPEECH adapter board works together with RDK3.
RDK3_QSPI_Flash_Access	This example demonstrates how to access the Infineon's external S25FL064LABNFI043 64MBit Flash on the RDK3.
RDK3_CapSense_Buttons	This example demonstrates how to use CapSense CSX Buttons on RDK3.
RDK3_I2C_BattCharger	This project demonstrates how to manage DIO59020 Li-ION battery charger over I2C.
RDK3_Arduino_ADC_HAL	This example demonstrates how to use the HAL library to measure all the ADC channels on the Arduino ADC header.
RDK3_RTC_Hybernate	This example demonstrates one of the low power modes: hibernation. RTC alarm is used as a wake-up source.
RDK3_BLE_SensorHub	This code example demonstrates the Android OS “CySmart” application’s custom BLE service “Sensor Hub”. The accelerometer, barometer, thermometer, and battery voltage data may be monitored using this service.
RDK3_I2C_Scanner	This application is used to find all the devices connected to the I2C.
RDK3_Hello_World	This example is an introduction to the basic components of the board: LEDs, Buttons, and KitProg3 UART for debugging.
RDK3_BLE_Find_Me	This is a typical "Bluetooth LE Find Me" application for RDK3.

RDK3_BLE_Beacon	This code example is for the RDK3 Bluetooth® LE testing as an Eddystone Beacon. It allows to evaluate the performance of the BLE Radio and to measure the emitted spectrum for the Electromagnetic Compatibility standards.
RDK3_PMIC_Test	This code example demonstrates the CYB06447BZI-BLD53 Power Management IC functionality.
TARGET_RDK3	This project is needed as a board support package while creating a new project with the RDK3 development kit.

Production Data

Schematics

You'll find the schematics of RDK3 here.

BOM

You'll find the BOM for RDK3 here.

RDK3 Electromagnetic Compatibility

RDK3 was tested for electromagnetic disturbances and electromagnetic immunity and meets the requirements as in normative documents listed below:

Electromagnetic disturbances:

Radiated disturbance to 1 GHz.

IEC 61000-4-20

Harmonised Standard for access to radio spectrum:

Data transmission equipment operating in the 2,4 GHz band.

ETSI EN 300 328