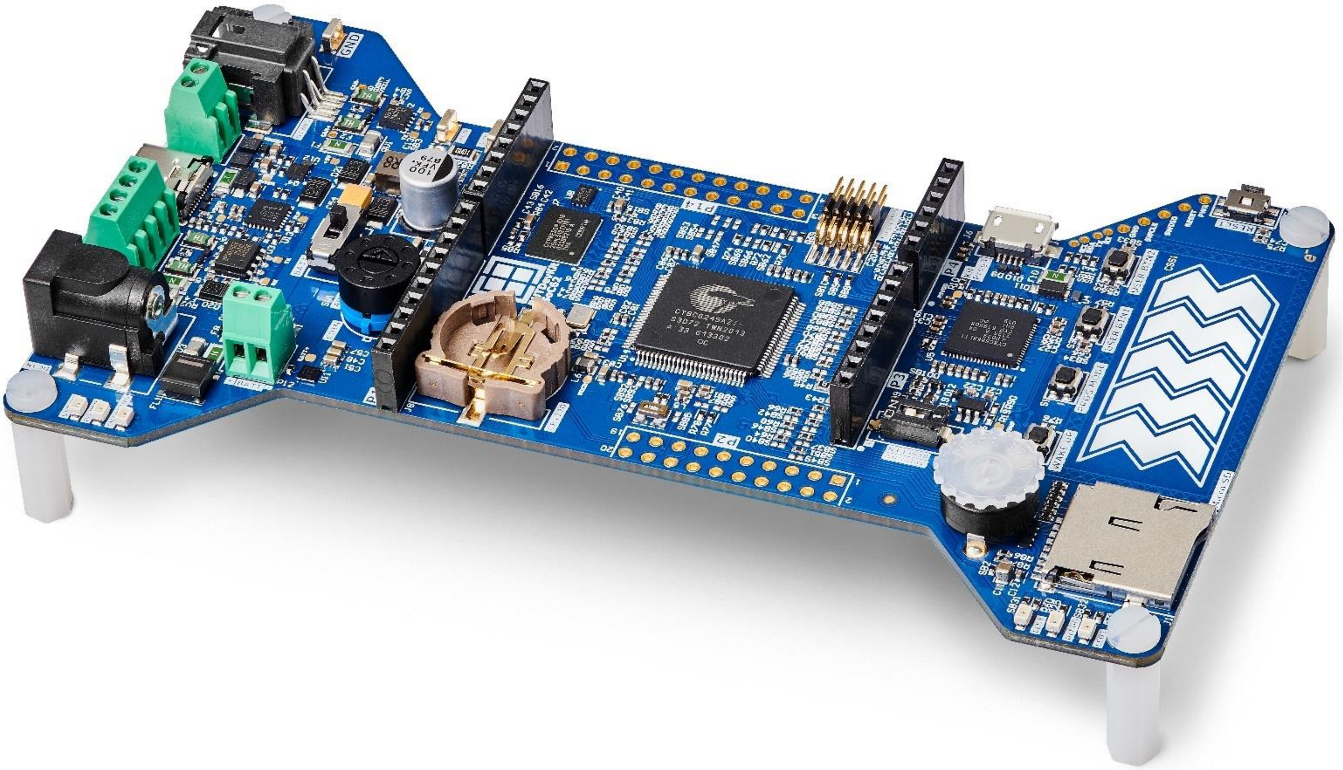
RDK2 User Manual



Versions

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
2.0	February 16, 2023	The template is updated, the information about software and hardware is added. 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

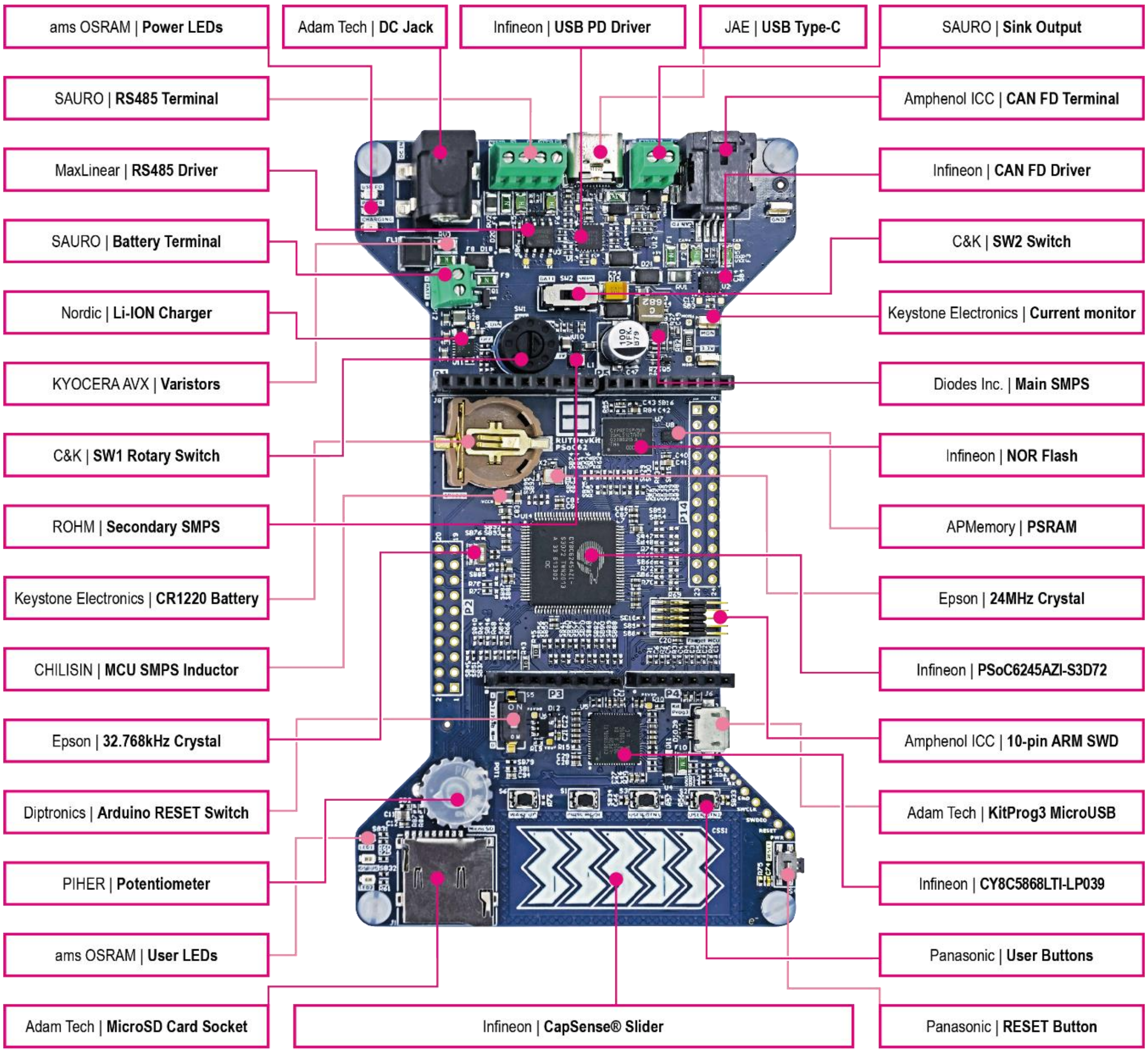
Features

RDK2 is a development board used by firmware and hardware designers to develop their products. RDK2 was designed by Rutronik to promote outstanding products selected only from their suppliers.

Key features of RDK2:

- 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.

Component Placement



ams OSRAM | **Power LEDs**

Adam Tech | **DC Jack**

Infineon | **USB PD Driver**

JAE | **USB Type-C**

SAURO | **Sink Output**

SAURO | **RS485 Terminal**

MaxLinear | **RS485 Driver**

SAURO | **Battery Terminal**

Nordic | **Li-ION Charger**

KYOCERA AVX | **Varistors**

C&K | **SW1 Rotary Switch**

ROHM | **Secondary SMPS**

Keystone Electronics | **CR1220 Battery**

CHILISIN | **MCU SMPS Inductor**

Epson | **32.768kHz Crystal**

Diptronics | **Arduino RESET Switch**

PIHER | **Potentiometer**

ams OSRAM | **User LEDs**

Adam Tech | **MicroSD Card Socket**

Amphenol ICC | **CAN FD Terminal**

Infineon | **CAN FD Driver**

C&K | **SW2 Switch**

Keystone Electronics | **Current monitor**

Diodes Inc. | **Main SMPS**

Infineon | **NOR Flash**

APMemory | **PSRAM**

Epson | **24MHz Crystal**

Infineon | **PSoC6245AZI-S3D72**

Amphenol ICC | **10-pin ARM SWD**

Adam Tech | **KitProg3 MicroUSB**

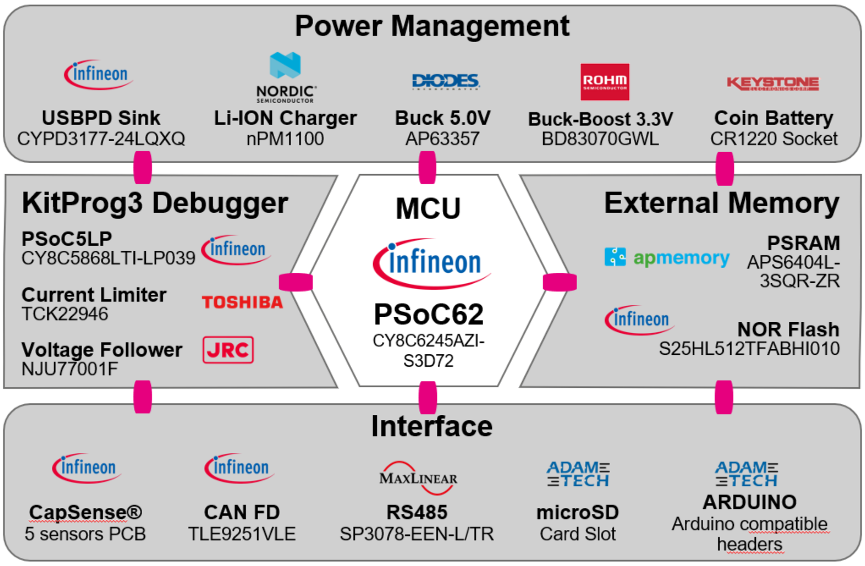
Infineon | **CY8C5868LTI-LP039**

Panasonic | **User Buttons**

Panasonic | **RESET Button**

Infineon | **CapSense® Slider**

Block Diagram



Power Management

USBPD Sink
CYPD3177-24LQXQ

Li-ION Charger
nPM1100

Buck 5.0V
AP63357

Buck-Boost 3.3V
BD83070GWL

Coin Battery
CR1220 Socket

KitProg3 Debugger

PSoC5LP
CY8C5868LTI-LP039

Current Limiter
TCK22946

Voltage Follower
NJU77001F

MCU

PSoC62
CY8C6245AZI-S3D72

External Memory

PSRAM
APS6404L-3SQR-ZR

NOR Flash
S25HL512TFABHI010

Interface

CapSense®
5 sensors PCB

CAN FD
TLE9251VLE

RS485
SP3078-EEN-L/TR

microSD
Card Slot

ARDUINO
Arduino compatible headers

Delivery Set

The delivery set of RDK2 includes:

- RDK2 development board.
- On-board debugger KitProg3 with I2C and UART USB bridge.
- CAN FD cable.
- USB Micro-B plug to USB-A plug cable to connect the board to PC.

Applicable Boards

The following Rutronik System Solution boards are compatible with RDK2 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.

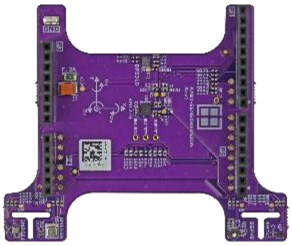
HMS ANYBUS

Testing the connectivity to all major fieldbuses and Industrial Ethernet networks and IoT standards.
More details see here.

Text to ech

Testing
Epson Text to

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. 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.

The memory includes:

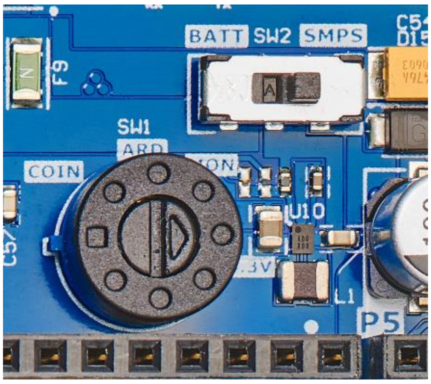
- 512 KB application flash, 32 KB auxiliary flash, and 32 KB supervisory flash; read-while-write support. Two 8 KB flash caches (one for each CPU).
- 256 KB SRAM with programmable power control and retention granularity.
- 64 KB ROM providing code for several system-level functions.
- One-time-programmable 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.

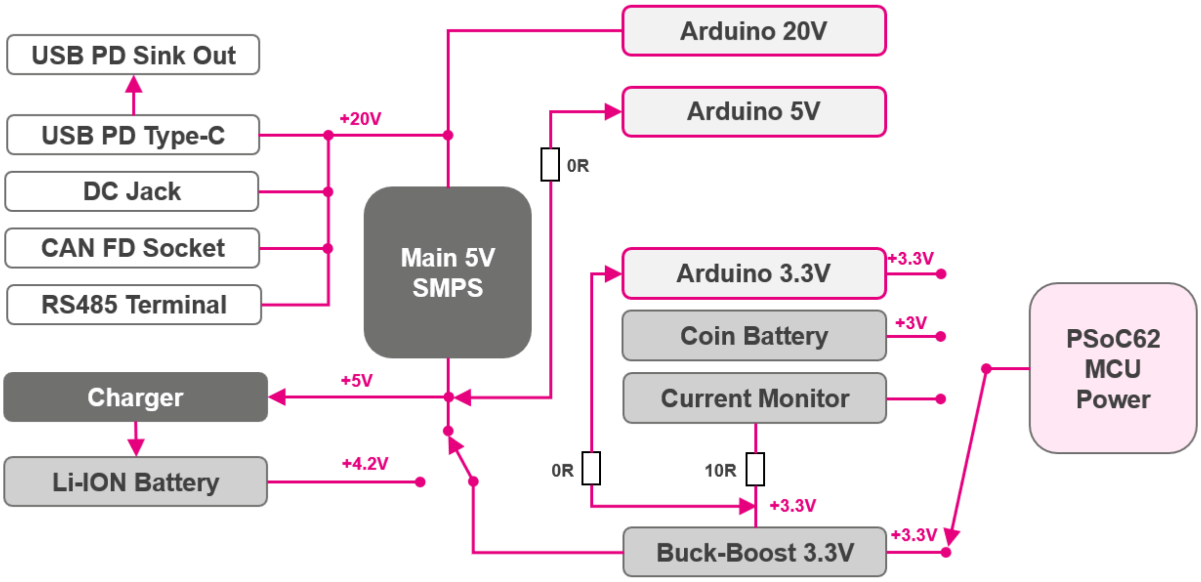
Power Sources

Six power sources are available in RDK2:

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.



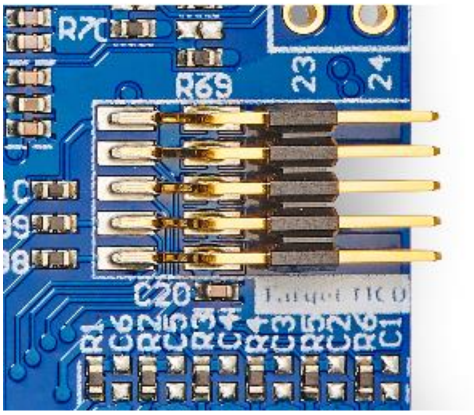
Using SW1 you can select between coin battery “**COIN**”, Arduino headers “**ARD**”, current Monitor “**MON**” and 3.3V SMPS “**3.3V**” power sources. With SW2 you can select **BATT** – Li-ION battery or 5V **SMPS** power sources. The available power sources are shown at the picture below.

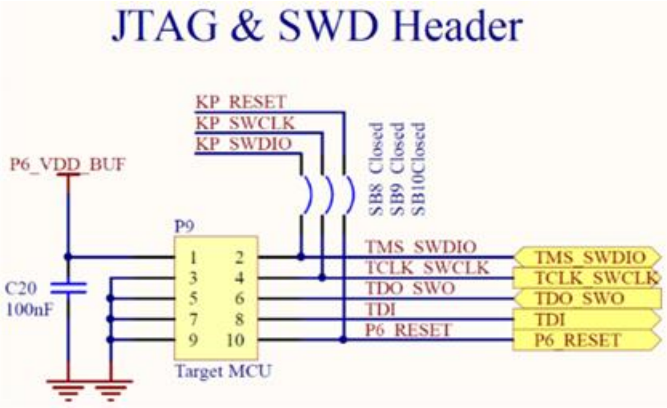


Power Distribution Diagram

Programming Using External Connector

Users may use third-party programming devices to connect the CY8C6245AZI-S3D72 target via the P9 SWD connector (10-pin, male, 1.27 mm pitch) that is shown below.

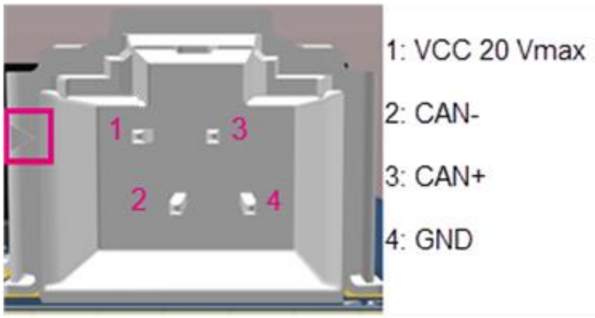




The onboard “KitProg3” debugger should not be powered while using an external JTAG connector.

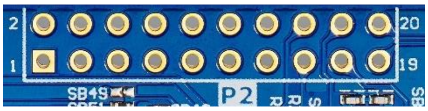
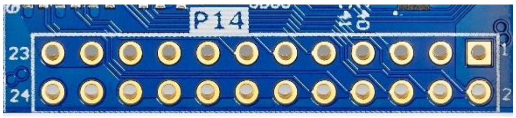
CAN FD Socket

Amphenol ICC Minitek MicroSpace™ provides a CAN FD connector (Part No.: 10142344-104KLF).



Spare GPIOs

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

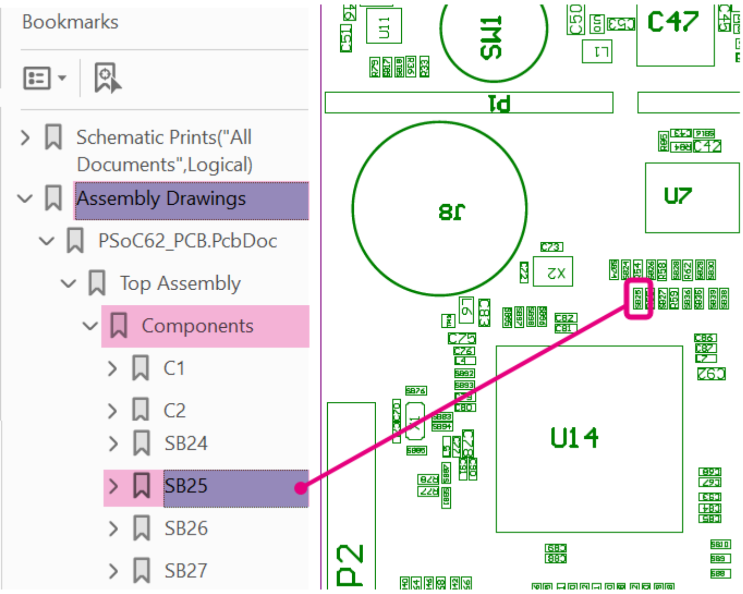
Socket P2 Pinout				Socket P14 Pinout			
							
Pin No.	Name	Name	Pin No.	Pin No.	Name	Name	Pin No.
1	P2.0	P2.4	2	1	P7.1	P9.2	2
3	P2.2	P2.3	4	3	P7.2	P9.0	4
5	P2.6	P2.1	6	5	P9.3	P7.7	6
7	P2.5	P11.6	8	7	P9.1	P7.3	8
9	P11.4	P11.2	10	9	P7.6	P7.0	10
11	P11.1	P11.3	12	11	P7.5	P5.0	12
13	P11.5	P11.7	14	13	P7.4	P6.2	14
15	P14.1	P14.0	16	15	P3.0	P6.6	16
17	P0.0	P0.1	18	17	P6.0	P5.1	18
19	P12.7	P12.6	20	19	P6.4	P6.3	20
				21	P3.1	P6.5	22
				23	P6.1	P6.7	24

Solder Bridges

Name	Circuit	Default	Name	Circuit	Default
SB1	Potentiometer output signal with P10.4	Closed	SB27	PSoC6 P11.4 with P2 header GPIO 9	Opened
SB2	Power for the microSD card interface	Closed	SB28	PSoC6 P11.3 with P2 header GPIO 12	Opened
SB3	+5V Power for the CAN FD driver	Closed	SB29	PSoC6 P11.1 with P2 header GPIO 11	Opened
SB4	+3.3V Power for the CAN FD driver	Closed	SB30	Flash SSEL Signal	Closed
SB5	120 Ohm termination for CAN FD connector	Opened	SB31	LED1 with PSoC6	Closed
SB6	Power for the RS485 Driver	Closed	SB32	LED2 with PSoC6	Closed
SB7	120 Ohm termination for RS485 terminal	Opened	SB33	USER_BTN2 with PSoC6	Closed
SB8	SWDIO signal with KitProg3	Closed	SB34	USER_BTN1 with PSoC6	Closed
SB10	RESET signal with KitProg3	Closed	SB35	PSoC6 P11.2 with P2 header GPIO 10	Opened
SB11	KitProg3 I2C SCL with PSoC6 MCU I2C SCL	Closed	SB36	PSRAM SSEL Signal	Closed
SB14	KitProg3 UART RX with PSoC6 MCU UART TX	Closed	SB37	PSoC6 P2.0 with P2 header GPIO 1	Opened
SB15	Power supply for the PSRAM IC	Closed	SB38	Arduino IO 4 with PSoC6	Closed
SB16	Power supply for the NOR Flash IC	Closed	SB39	PSRAM RESET signal	Opened
SB17	PSoC6 MCU I2C SDA with charger SDA	Closed	SB40	PSoC6 P2.1 with P2 header GPIO 6	Opened
SB18	PSoC6 MCU I2C SCL with charger SCL	Closed	SB41	PSoC6 P2.2 with P2 header GPIO 3	Opened
SB19	Target voltage follower input isolation	Closed	SB42	PSoC6 P2.3 with P2 header GPIO 4	Opened
SB20	3.3V SMPS Enable with VIN	Closed	SB43	PSoC6 P2.4 with P2 header GPIO 2	Opened
SB21	3.3V SMPS Enable with external signal (PMIC)	Opened	SB44	PSoC6 P9.0 with P14 header GPIO 4	Opened
SB22	PSoC6 MCU I2C SDA with USB PD IC SDA	Closed	SB46	PSoC6 P2.6 with P2 header GPIO 5	Opened
SB23	PSoC6 MCU I2C SDL with USB PD IC SDL	Closed	SB47	PSoC6 P9.1 with P14 header GPIO 7	Opened
SB24	PSoC6 P11.7 with P2 header GPIO 14	Opened	SB49	PSoC6 P2.6 with P2 header GPIO 5	Opened
SB25	PSoC6 P11.6 with P2 header GPIO 8	Opened	SB50	PSoC6 P9.2 with P14 header GPIO 2	Opened
SB26	PSoC6 P11.5 with P2 header GPIO 13	Opened	SB51	microSD Card Detect signal	Closed

SB52	5V SMPS Power Good signal	Opened	SB77	PSoC6 P6.3 with P14 header GPIO 20	Opened
SB53	PSoC6 P9.3 with P14 header GPIO 5	Opened	SB78	RS485 DE signal	Closed
SB54	USB PD IC Interrupt signal	Opened	SB79	Potentiometer VDD terminal	Closed
SB55	PSoC6 P3.0 with P14 header GPIO 15	Opened	SB80	PSoC6 P6.4 with P14 header GPIO 19	Opened
SB56	PSoC6 UART RX signal	Closed	SB81	PSoC6 P14.0 with P2 header GPIO 16	Opened
SB57	PSoC6 P3.1 with P14 header GPIO 21	Opened	SB82	JTAG TDO signal	Closed
SB58	PSoC6 UART TX signal	Closed	SB83	WCO Input signal	Closed
SB59	PSoC6 P7.0 with P14 header GPIO 10	Opened	SB84	PSoC6 P6.5 with P14 header GPIO 22	Opened
SB60	PSoC6 P5.0 with P14 header GPIO 12	Opened	SB85	PSoC6 P0.1 with P2 header GPIO 18	Opened
SB61	CAN FD RX signal	Closed	SB86	JTAG TDI signal	Closed
SB62	PSoC6 P7.3 with P14 header GPIO	Closed	SB87	PSoC6 P14.0 with P2 header GPIO 16	Opened
SB63	PSoC6 P5.1 with P14 header GPIO	Opened	SB88	PSoC6 P6.6 with P14 header GPIO 16	Opened
SB64	CAN FD TX signal	Closed	SB89	JTAG TMS, SWDIO signal	Closed
SB65	PSoC6 P7.4 with P14 header GPIO 8	Opened	SB90	PSoC6 P6.7 with P14 header GPIO 24	Opened
SB66	PSoC6 P7.5 with P14 header GPIO 11	Opened	SB91	JTAG TCLK, SWCLK signal	Closed
SB67	PSoC6 P7.6 with P14 header GPIO 9	Opened	SB92	PSoC6 Backup Voltage Input	Closed
SB68	PSoC6 P6.2 with P14 header GPIO 14	Opened	SB93	Coin Battery with PSoC6 Backup Voltage Input	Opened
SB69	PSoC6 P7.7 with P14 header GPIO 6	Opened	SB94	WCO Output signal	Closed
SB70	CAN FD Driver Stand-By control input	Closed	SB95	PSoC6 P12.7 with P2 header GPIO 19	Opened
SB71	PSoC6 P6.0 with P14 header GPIO 17	Opened	SB96	PSoC6 P12.6 with P2 header GPIO 20	Opened
SB72	RS485 RX signal	Closed	SB97	ECO Input Signal	Closed
SB73	PSoC6 P6.1 with P14 header GPIO 23	Opened	SB98	ECO Output Signal	Closed
SB74	PSRAM Interrupt signal	Opened	SB99	Voltage Divider Enable signal	Closed
SB75	RS485 TX signal	Closed	SB100	Voltage Divider ADC signal	Closed
SB76	PSoC6 P0.0 with P2 header GPIO 17	Opened			

The locations of the solder bridges can be found in 3D model and assembly drawing of RDK2.


Bookmarks

- Schematic Prints("All Documents",Logical)
- Assembly Drawings
  - PSoC62_PCB.PcbDoc
    - Top Assembly
      - Components
        - C1
        - C2
        - SB24
        - SB25
        - SB26
        - SB27

How to find a component on the layout

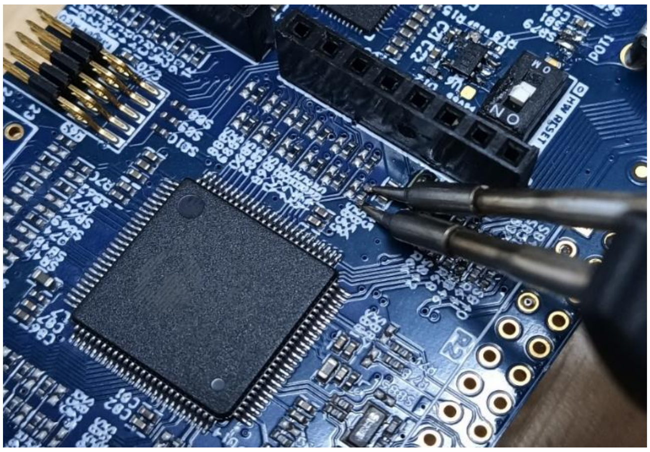
Fuses

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. RDK2 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.

Changing the Fuses or Solder Bridges

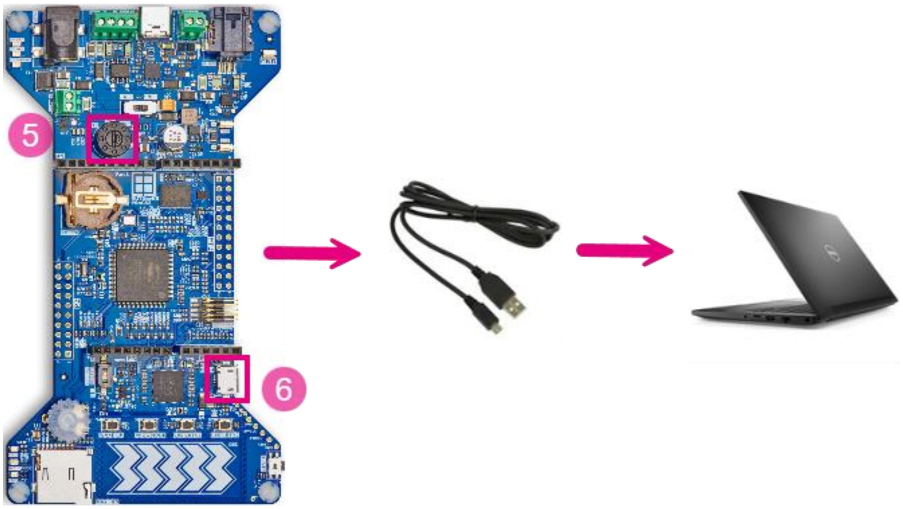
Some of the components might be hard to access, therefore the SMD „Chipping Tool“ is recommended to use for SMD solder bridges or fuses soldering on the RDK2 development board.



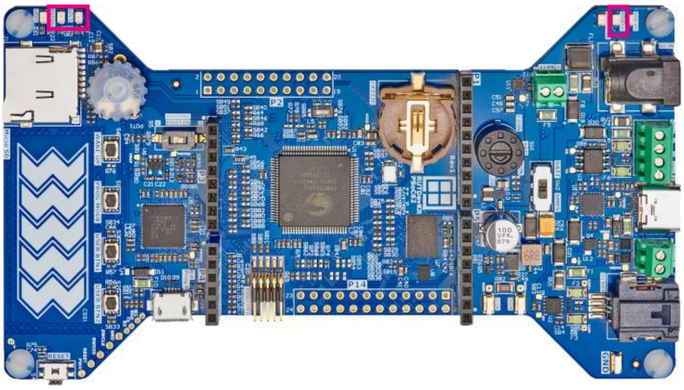
Software and Firmware

Getting Started

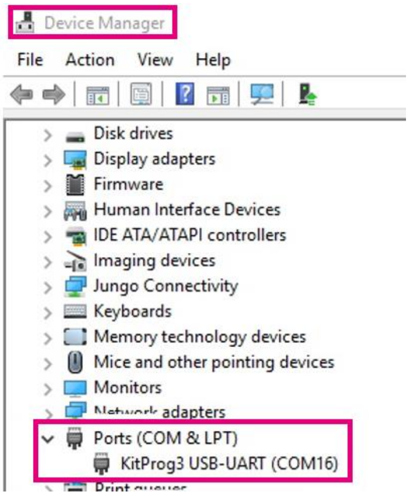
1. Register or/and login at Infineon website (myInfineon tab). License generation may take up to several days.
2. Download and install the latest version of ModusToolbox™ software.
3. Get all the supported firmware examples including the BSP from RDK2 homepage. Press the „Download Area“ and login/register to access the data. https://www.rutronik.com/devkit-login
4. *[Optional]* Download and install your preferred terminal emulator, for example: PuTTY, Tera Term, etc.
5. Ensure switch SW1 is set to „3.3V“
6. Connect your board (micro USB socket with a marking "KitProg3") and a Windows based PC via USB Cable – A to Micro B.



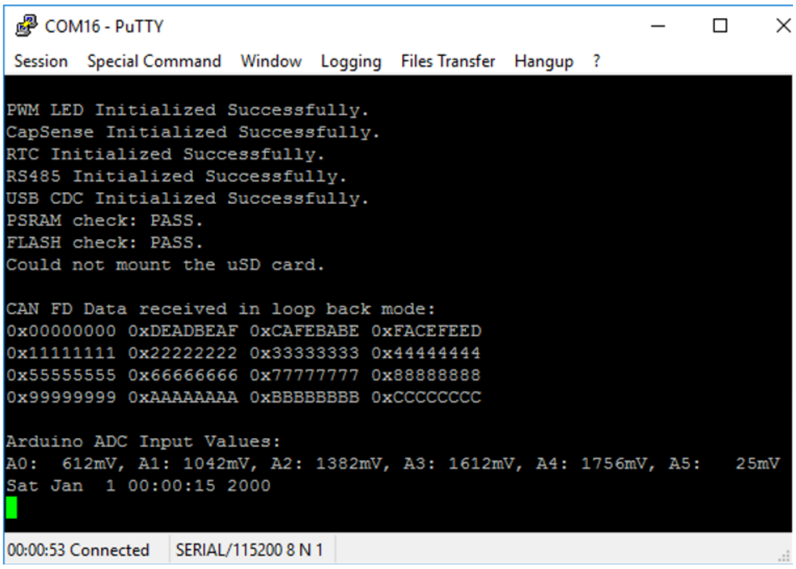
7. Check if RDK2 is ready. Its "POWER" and "DEBUG" LEDs should shine constantly. The LED1 reacts to the touch on the slider.



8. The “KitProg3” port must be seen in MS Windows Device Manager window.



9. All new RDK2 boards print out the hardware test results to the KitProg3 serial terminal (115200 bit/s) as it’s shown below. If it doesn’t happen please press RESET on the RDK2.


COM16 - PuTTY
Session Special Command Window Logging Files Transfer Hangup ?

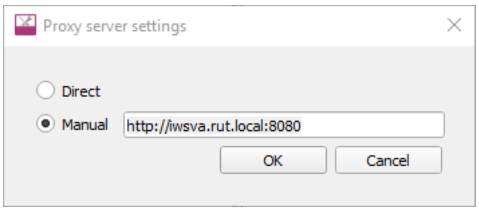
```
PWM LED Initialized Successfully.
CapSense Initialized Successfully.
RTC Initialized Successfully.
RS485 Initialized Successfully.
USB CDC Initialized Successfully.
PSRAM check: PASS.
FLASH check: PASS.
Could not mount the uSD card.

CAN FD Data received in loop back mode:
0x00000000 0xDEADBEAF 0xCAFEBABE 0xFACEFEED
0x11111111 0x22222222 0x33333333 0x44444444
0x55555555 0x66666666 0x77777777 0x88888888
0x99999999 0xAAAAAAAA 0xBBBBBBBB 0xCCCCCCCC

Arduino ADC Input Values:
A0:  612mV, A1: 1042mV, A2: 1382mV, A3: 1612mV, A4: 1756mV, A5:   25mV
Sat Jan  1 00:00:15 2000
```

00:00:53 Connected | SERIAL/115200 8 N 1

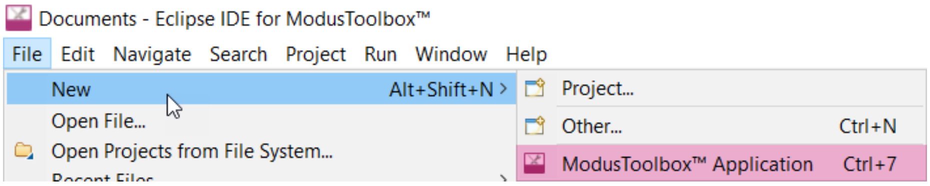
10. *[For Rutronik laptops only]* Run **File – New – ModusToolbox Application – Settings – Proxy server settings** and enter the proxy address: http://iwsva.rut.local:8080



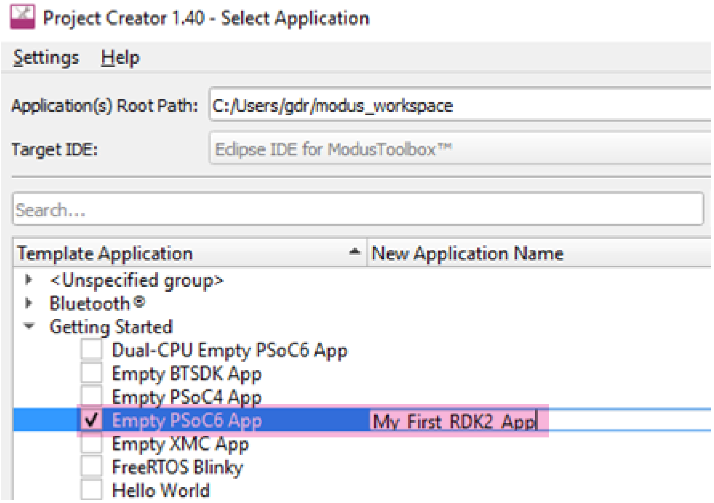
○ Direct
● Manual http://iwsva.rut.local:8080

Creating New Project

1. Drop “TARGET_RDK2” folder (extracted from TARGET_RDK2.zip that can be found in Rutronik Download Area) in your workspace directory.
2. Select **File – New – ModusToolbox Application**. Wait for a while, click **Browse**, find the directory …\TARGET_RDK2, select it and press **Choose**. The kit will appear in the list, press **Next** after that.


Documents - Eclipse IDE for ModusToolbox™
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N >
Open File...
Open Projects from File System...
Recent Files
Project...
Other... Ctrl+N
ModusToolbox™ Application Ctrl+7

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.

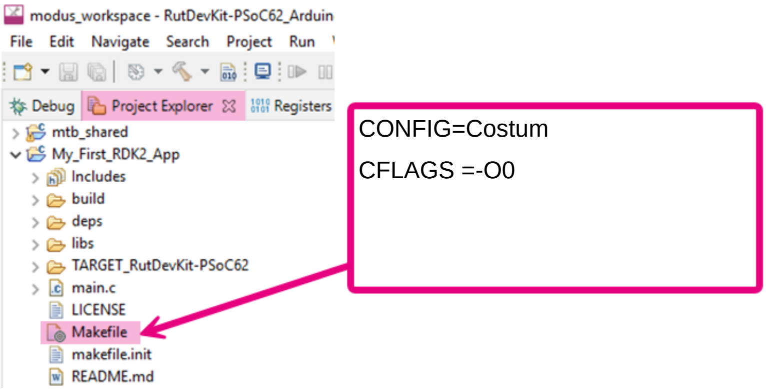


Application(s) Root Path: C:/Users/gdr/modus_workspace

Target IDE: Eclipse IDE for ModusToolbox™

☑ Empty PSoC6 App — My_First_RDK2_App

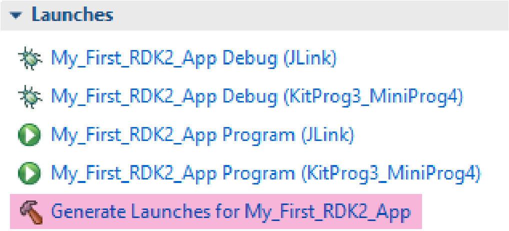
4. Modify Makefile as it’s shown below to disable the code optimisation.



```
CONFIG=Costum
CFLAGS =-O0
```

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

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



Launches

My_First_RDK2_App Debug (JLink)

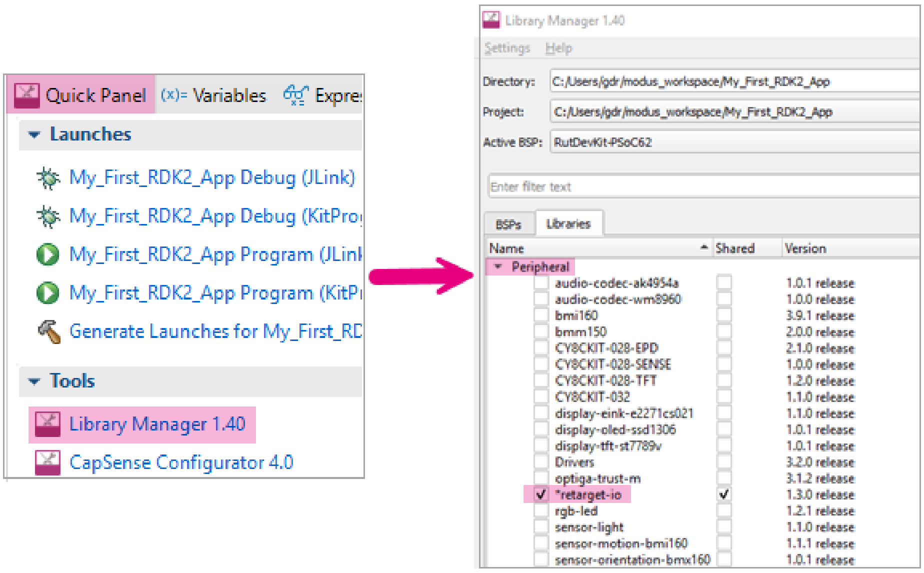
My_First_RDK2_App Debug (KitProg3_MiniProg4)

My_First_RDK2_App Program (JLink)

My_First_RDK2_App Program (KitProg3_MiniProg4)

Generate Launches for My_First_RDK2_App

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



Quick Panel
(x)= Variables
Expre
Launches
My_First_RDK2_App Debug (JLink)
My_First_RDK2_App Debug (KitPro
My_First_RDK2_App Program (JLink
My_First_RDK2_App Program (KitPr
Generate Launches for My_First_RD
Tools
Library Manager 1.40
CapSense Configurator 4.0
Library Manager 1.40
Settings
Help
BSPs
Libraries

Directory: C:/Users/gdr/modus_workspace/My_First_RDK2_App

Project: C:/Users/gdr/modus_workspace/My_First_RDK2_App

Active BSP: RutDevKit-PSoC62

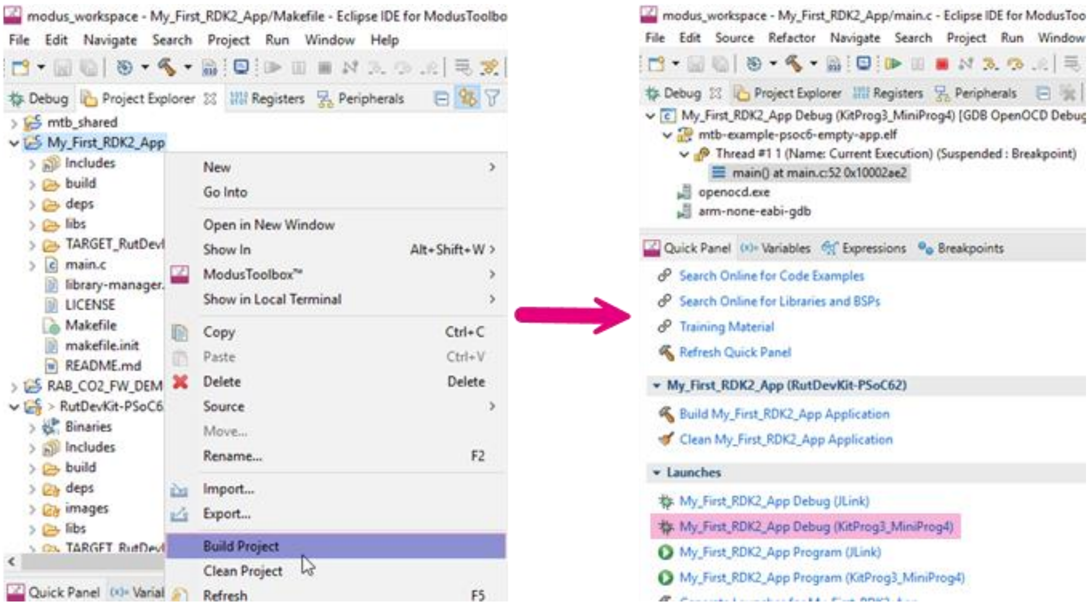
Enter filter text

Name	Shared	Version
Peripheral		
□ audio-codec-ak4954a	□	1.0.1 release
□ audio-codec-wm8960	□	1.0.0 release
□ bmi160	□	3.9.1 release
□ bmm150	□	2.0.0 release
□ CY8CKIT-028-EPD	□	2.1.0 release
□ CY8CKIT-028-SENSE	□	1.0.0 release
□ CY8CKIT-028-TFT	□	1.2.0 release
□ CY8CKIT-032	□	1.1.0 release
□ display-eink-e2271cs021	□	1.1.0 release
□ display-oled-ssd1306	□	1.0.1 release
□ display-tft-st7789v	□	1.0.1 release
□ Drivers	□	3.2.0 release
□ optiga-trust-m	□	3.1.2 release
☑ *retarget-io	✓	1.3.0 release
□ rgb-led	□	1.2.1 release
□ sensor-light	□	1.1.0 release
□ sensor-motion-bmi160	□	1.1.1 release
□ sensor-orientation-bmx160	□	1.0.1 release

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

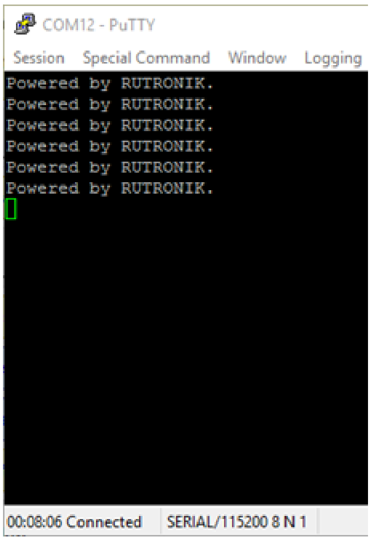
```c
#include "cy_pdl.h"
#include "cyhal.h"
#include "cybsp.h"
#include "cy_retarget_io.h"
int main(void)
{
    cy_rslt_t result;
    /* Initialize the device and board peripherals */
    result = cybsp_init() ;
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }
    __enable_irq();
    /*Initialize LED1*/
    result = cyhal_gpio_init( LED1, 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(LED1);
    }
}
```

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


modus_workspace - My_First_RDK2_App/Makefile - Eclipse IDE for ModusToolbo
File Edit Navigate Search Project Run Window Help
Debug
Project Explorer
Registers
Peripherals
mtb_shared
My_First_RDK2_App
Includes
build
deps
libs
TARGET_RutDevK
main.c
library-manager.
LICENSE
Makefile
makefile.init
README.md
RAB_CO2_FW_DEM
> RutDevKit-PSoC6
Binaries
Includes
build
deps
images
libs
TARGET_RutDevK
Quick Panel
Varial
New
Go Into
Open in New Window
Show In Alt+Shift+W
ModusToolbox™
Show in Local Terminal
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Source
Move...
Rename... F2
Import...
Export...
Build Project
Clean Project
Refresh F5
modus_workspace - My_First_RDK2_App/main.c - Eclipse IDE for ModusToo
File Edit Source Refactor Navigate Search Project Run Window
Debug
Project Explorer
Registers
Peripherals
My_First_RDK2_App Debug (KitProg3_MiniProg4) [GDB OpenOCD Debug
mtb-example-psoc6-empty-app.elf
Thread #1 1 (Name: Current Execution) (Suspended : Breakpoint)
main() at main.c:52 0x10002ae2
openocd.exe
arm-none-eabi-gdb
Quick Panel
Variables
Expressions
Breakpoints
Search Online for Code Examples
Search Online for Libraries and BSPs
Training Material
Refresh Quick Panel
My_First_RDK2_App (RutDevKit-PSoC62)
Build My_First_RDK2_App Application
Clean My_First_RDK2_App Application
Launches
My_First_RDK2_App Debug (JLink)
My_First_RDK2_App Debug (KitProg3_MiniProg4)
My_First_RDK2_App Program (JLink)
My_First_RDK2_App Program (KitProg3_MiniProg4)

9. The final result is a blinking LED1 on the RDK2 board and text on the terminal window.

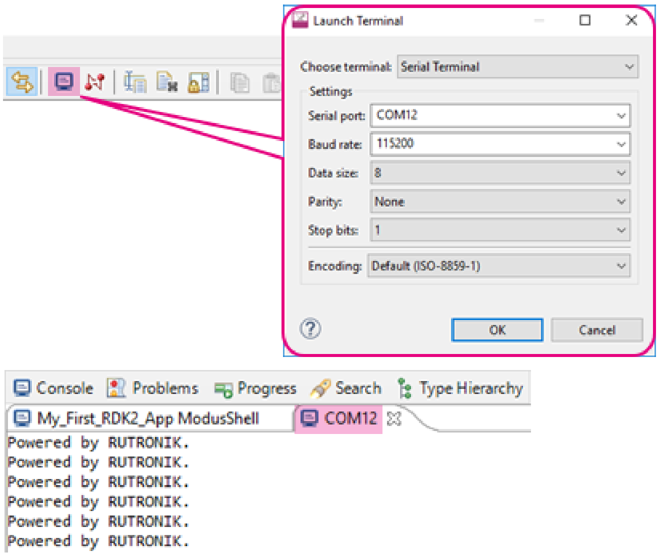
PuTTY Terminal



```
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
```

00:08:06 Connected SERIAL/115200 8 N 1

ModusToolbox Terminal



Choose terminal: Serial Terminal

Serial port: COM12

Baud rate: 115200

Data size: 8

Parity: None

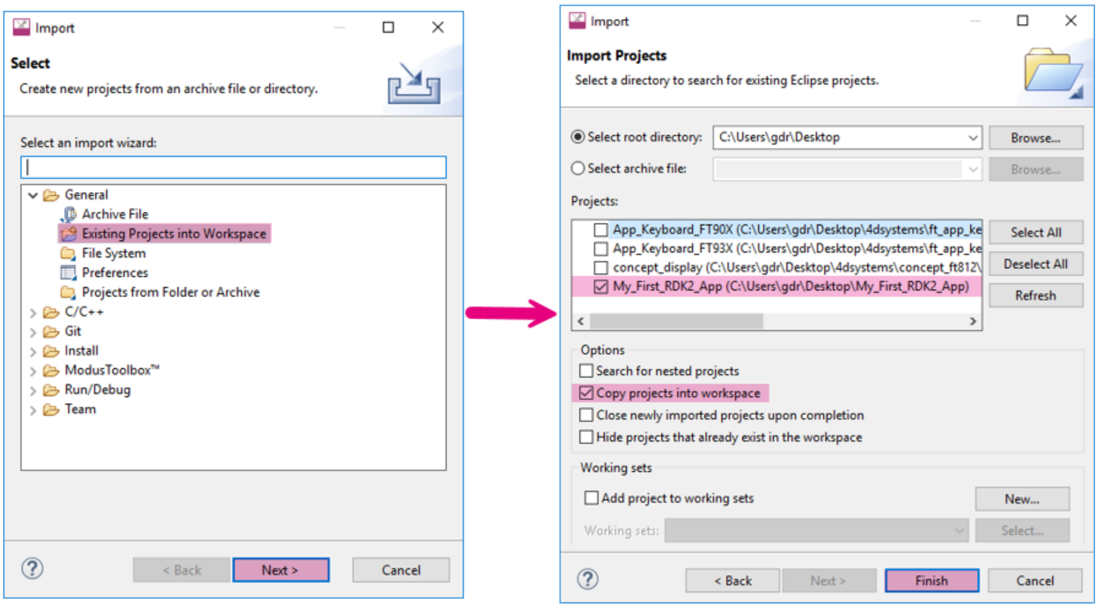
Stop bits: 1

Encoding: Default (ISO-8859-1)

```
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
Powered by RUTRONIK.
```

Running Existing Project

1. Go **File** - **Import**… - **Existing Projects into Workspace** - **Next**. Select a directory and the project to import, check **Copy projects into workspace** and then click on **Finish**.



● Select root directory: C:\Users\gdr\Desktop

○ Select archive file:

Projects:

- □ App_Keyboard_FT90X (C:\Users\gdr\Desktop\4dsystems\ft_app_ke
- □ App_Keyboard_FT93X (C:\Users\gdr\Desktop\4dsystems\ft_app_ke
- □ concept_display (C:\Users\gdr\Desktop\4dsystems\concept_ft812\
- ☑ My_First_RDK2_App (C:\Users\gdr\Desktop\My_First_RDK2_App)

Options

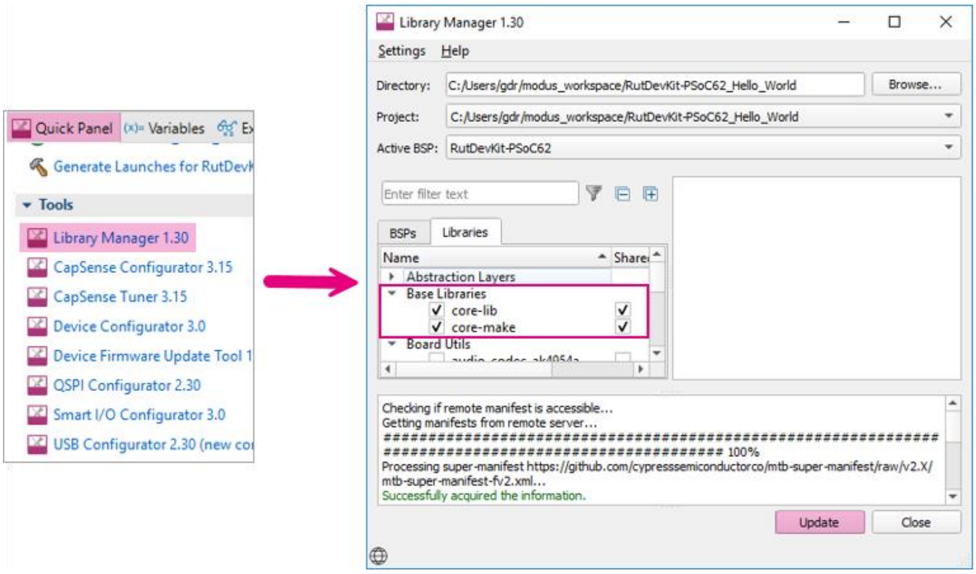
- □ Search for nested projects
- ☑ Copy projects into workspace
- □ Close newly imported projects upon completion
- □ Hide projects that already exist in the workspace

Working sets

- □ Add project to working sets

Working sets:

2. Before starting the software, we have to make sure that all libraries are up to date, for this please select **Library Manager** in the **Quick Panel** and click **Update**.



Directory: C:/Users/gdr/modus_workspace/RutDevKit-PSoC62_Hello_World

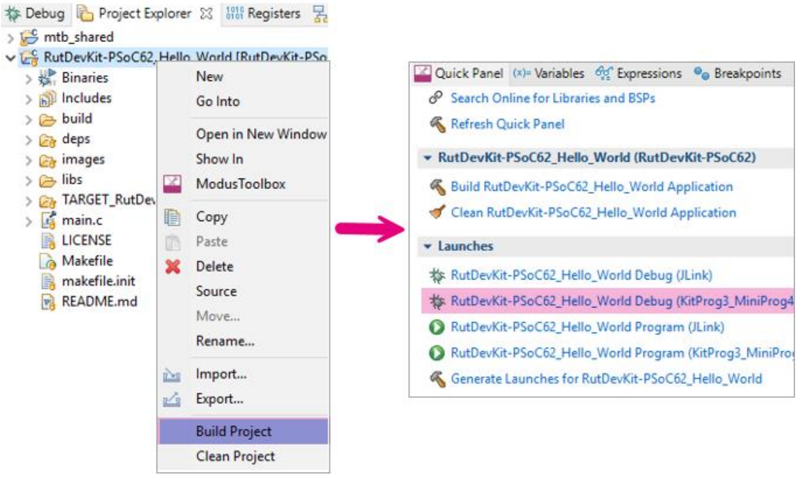
Project: C:/Users/gdr/modus_workspace/RutDevKit-PSoC62_Hello_World

Active BSP: RutDevKit-PSoC62

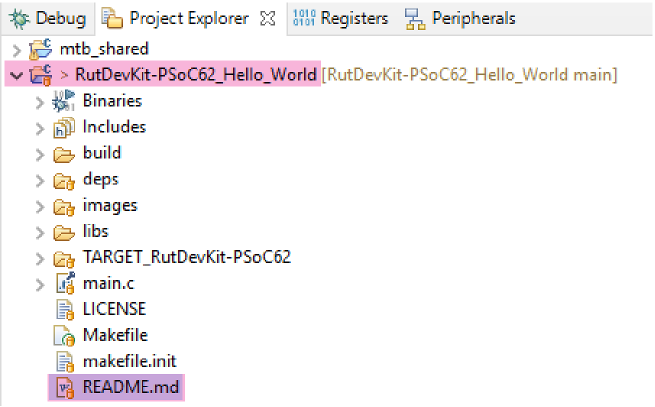
Name	Shared
Abstraction Layers	
Base Libraries	
✓ core-lib	✓
✓ core-make	✓
Board Utils	
□ audio-codec-ak4954a	□

```
Checking if remote manifest is accessible…
Getting manifests from remote server…
####################################################################
######################################## 100%
Processing super-manifest https://github.com/cypresssemiconductorco/mtb-super-manifest/raw/v2.X/
mtb-super-manifest-fv2.xml…
Successfully acquired the information.
```

3. Select the project (for example, “RutDevKit-PSoC62-Hello_World”). Build and debug it.

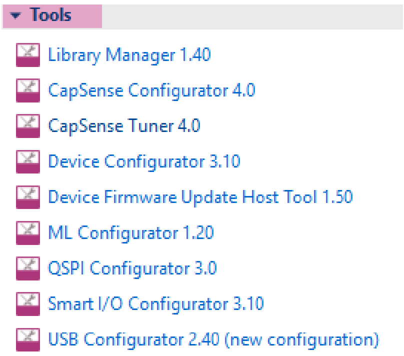

Debug
Project Explorer
Registers
mtb_shared
RutDevKit-PSoC62_Hello_World [RutDevKit-PSo
Binaries
Includes
build
deps
images
libs
TARGET_RutDev
main.c
LICENSE
Makefile
makefile.init
README.md
New
Go Into
Open in New Window
Show In
ModusToolbox
Copy
Paste
Delete
Source
Move...
Rename...
Import...
Export...
Build Project
Clean Project
Quick Panel
Variables
Expressions
Breakpoints
Search Online for Libraries and BSPs
Refresh Quick Panel
RutDevKit-PSoC62_Hello_World (RutDevKit-PSoC62)
Build RutDevKit-PSoC62_Hello_World Application
Clean RutDevKit-PSoC62_Hello_World Application
Launches
RutDevKit-PSoC62_Hello_World Debug (JLink)
RutDevKit-PSoC62_Hello_World Debug (KitProg3_MiniProg4
RutDevKit-PSoC62_Hello_World Program (JLink)
RutDevKit-PSoC62_Hello_World Program (KitProg3_MiniPro
Generate Launches for RutDevKit-PSoC62_Hello_World

4. 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.



Useful Links and Tools

1. **Peripheral Driver Library API**: https://infineon.github.io/psoc6pdl/pdl_api_reference_manual/html/index.html
2. **Hardware Abstraction Layer API**: https://infineon.github.io/psoc6hal/html/index.html
3. **ModusToolbox Tools:** all tools are very useful and we recommend using them in your work.


Tools
Library Manager 1.40
CapSense Configurator 4.0
CapSense Tuner 4.0
Device Configurator 3.10
Device Firmware Update Host Tool 1.50
ML Configurator 1.20
QSPI Configurator 3.0
Smart I/O Configurator 3.10
USB Configurator 2.40 (new configuration)

Firmware Examples

All these examples can be found in the Download Area at RDK2 page.

RDK2 BSP	This project is needed as a board support package while creating a new project with the RDK2 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.

Application Examples

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

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

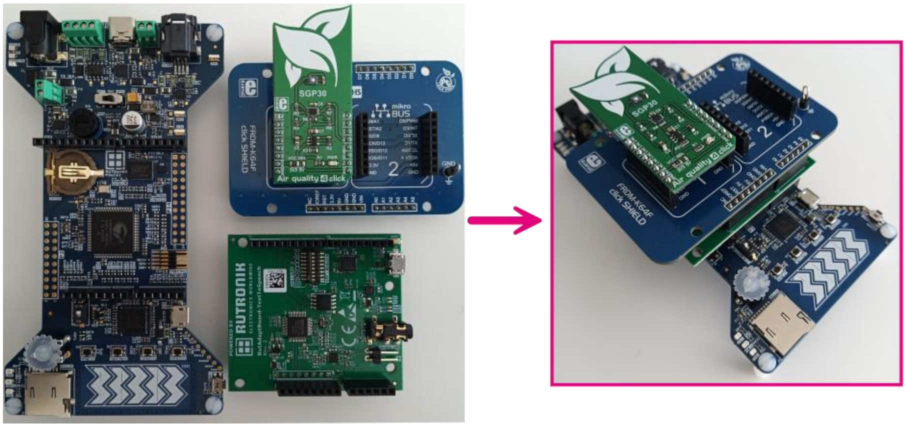
The display is driven using an SPI interface therefore the adapter for Arduino headers is needed. The demo application has static display objects and rotating an image using EVE capabilities. The image rotation may be started or stopped using a button on the screen.

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

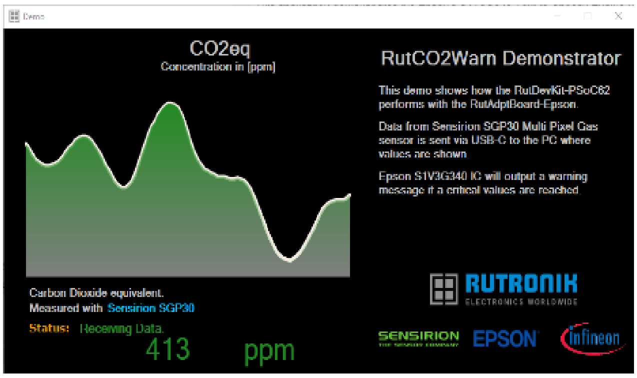
This example is for introduction to the 4D Systems „gen4-uLCD-43DCT-CLB“ 4.3“ display module with capacitive touch. The display is controlled using the UART interface. The SGP30 sensor attached to the I2C is needed to have this demo working. The demo application has a gauge and a scope that represents 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 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.



RutCO2Warn Demonstrator

This demo shows how the RutDevKit-PSoC62 performs with the RutAdptBoard-Epson.

Data from Sensirion SGP30 Multi Pixel Gas sensor is sent via USB-C to the PC where values are shown.

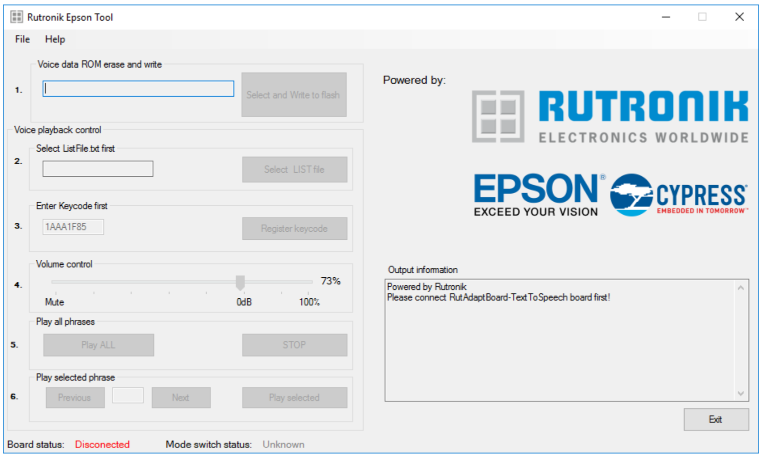
Epson S1V3G340 IC will output a warning message if a critical values are reached.

Carbon Dioxide equivalent.
Measured with Sensirion SGP30

Status: Receiving Data.

413 ppm

The Text-to-Speech IC is used for audible voice announcements. The LED1 on the RDK2 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 RDK2 RutCO2Alarm project.



1. Voice data ROM erase and write

Voice playback control

2. Select ListFile.txt first

3. Enter Keycode first: 1AAA1F85

4. Volume control: 73%

5. Play all phrases

6. Play selected phrase

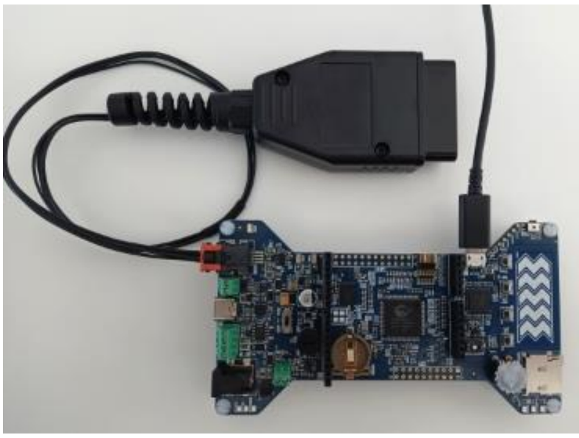
Output information

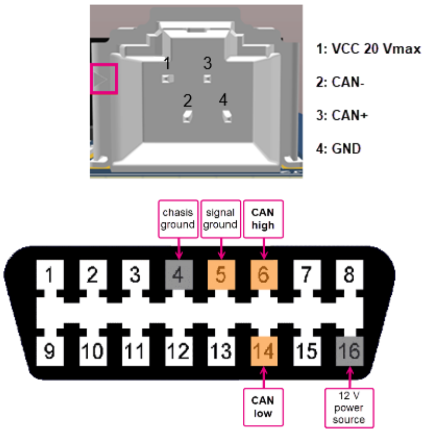
Powered by Rutronik
Please connect RutAdaptBoard-TextToSpeech board first!

Board status: Disconected   Mode switch status: Unknown

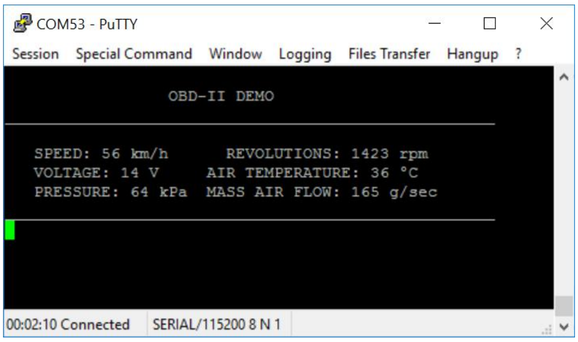
The OBD-II Test Application

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.





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).



```
                OBD-II DEMO
 ------------------------------------------------
   SPEED: 56 km/h        REVOLUTIONS: 1423 rpm
   VOLTAGE: 14 V       AIR TEMPERATURE: 36 °C
   PRESSURE: 64 kPa    MASS AIR FLOW: 165 g/sec
 ------------------------------------------------
```

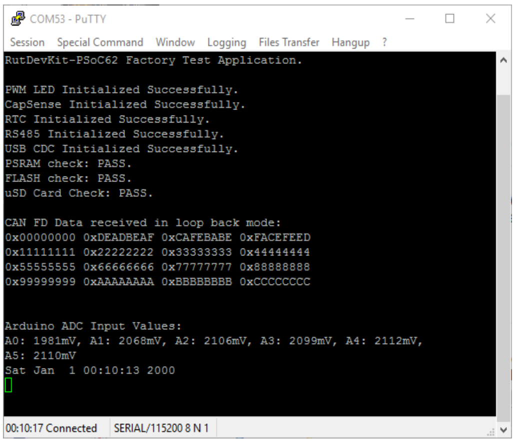
The OBD-II Turbocharger Monitor Application

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.

RDK2 (Factory) Production Firmware

This firmware example is pre-programmed 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.



```
RutDevKit-PSoC62 Factory Test Application.

PWM LED Initialized Successfully.
CapSense Initialized Successfully.
RTC Initialized Successfully.
RS485 Initialized Successfully.
USB CDC Initialized Successfully.
PSRAM check: PASS.
FLASH check: PASS.
uSD Card Check: PASS.

CAN FD Data received in loop back mode:
0x00000000 0xDEADBEAF 0xCAFEBABE 0xFACEFEED
0x11111111 0x22222222 0x33333333 0x44444444
0x55555555 0x66666666 0x77777777 0x88888888
0x99999999 0xAAAAAAAA 0xBBBBBBBB 0xCCCCCCCC


Arduino ADC Input Values:
A0: 1981mV, A1: 2068mV, A2: 2106mV, A3: 2099mV, A4: 2112mV,
A5: 2110mV
Sat Jan  1 00:10:13 2000
```

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.

Production Data

Schematics, Mechanical Layout and Prints

You'll find the schematics, mechanical layout and prints for RDK2 here.

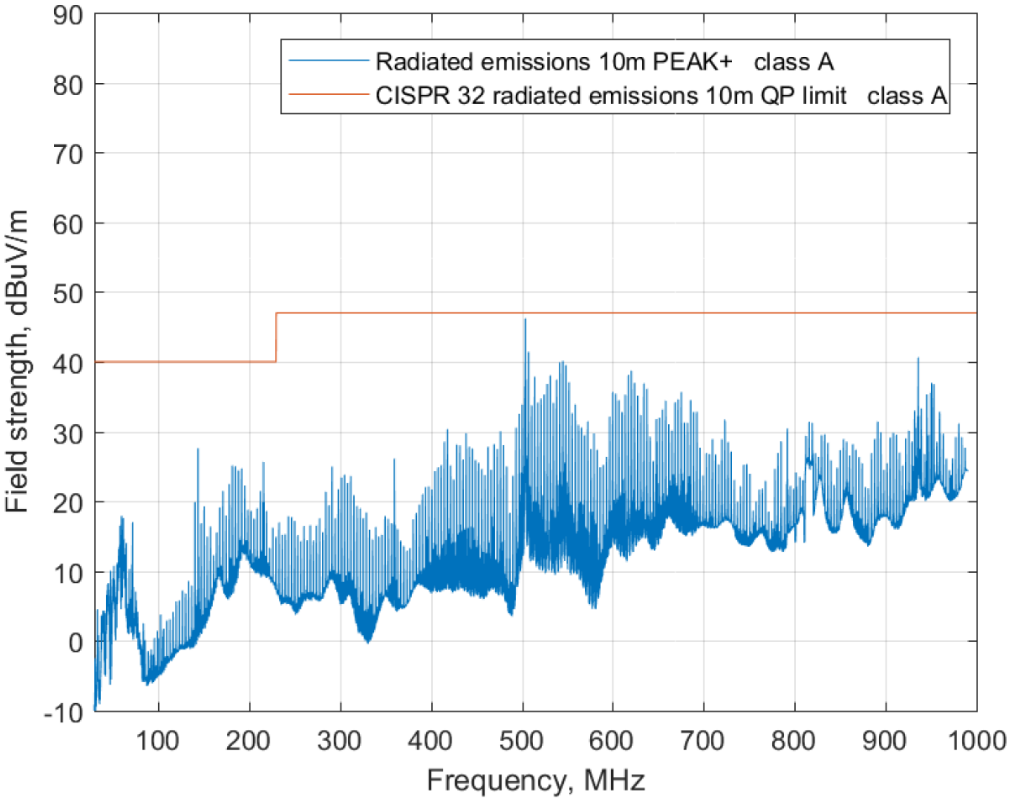
BOM

You'll find the BOM for RDK2 here.

Electromagnetic Compatibility

RDK2 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

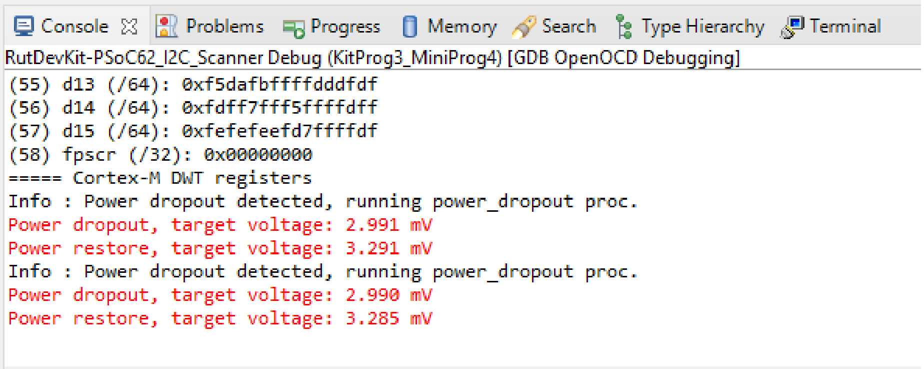


Radiated disturbances while running PSRAM Test

Known Issues

I2C Issue

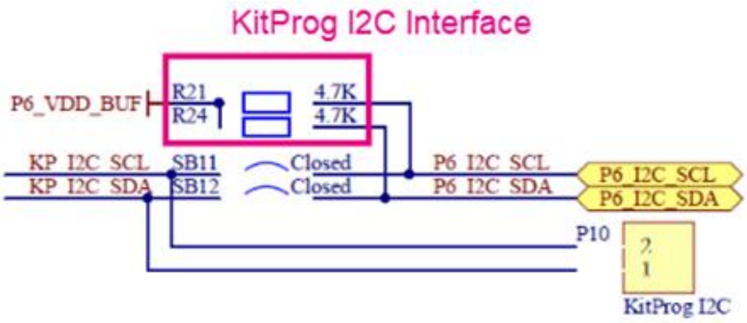
The power dropout warning appears in the ModusToolbox console window while debugging and working with the I2C interface.



```
RutDevKit-PSoC62_I2C_Scanner Debug (KitProg3_MiniProg4) [GDB OpenOCD Debugging]
(55) d13 (/64): 0xf5dafbffffdddfdf
(56) d14 (/64): 0xfdff7fff5ffffdff
(57) d15 (/64): 0xfefefeefd7ffffdf
(58) fpscr (/32): 0x00000000
===== Cortex-M DWT registers
Info : Power dropout detected, running power_dropout proc.
Power dropout, target voltage: 2.991 mV
Power restore, target voltage: 3.291 mV
Info : Power dropout detected, running power_dropout proc.
Power dropout, target voltage: 2.990 mV
Power restore, target voltage: 3.285 mV
```

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.


KitProg I2C Interface
P6_VDD_BUF
R21
R24
4.7K
4.7K
KP_I2C_SCL
KP_I2C_SDA
SB11
SB12
Closed
Closed
P6_I2C_SCL
P6_I2C_SDA
P6_I2C_SCL
P6_I2C_SDA
P10
2
1
KitProg I2C