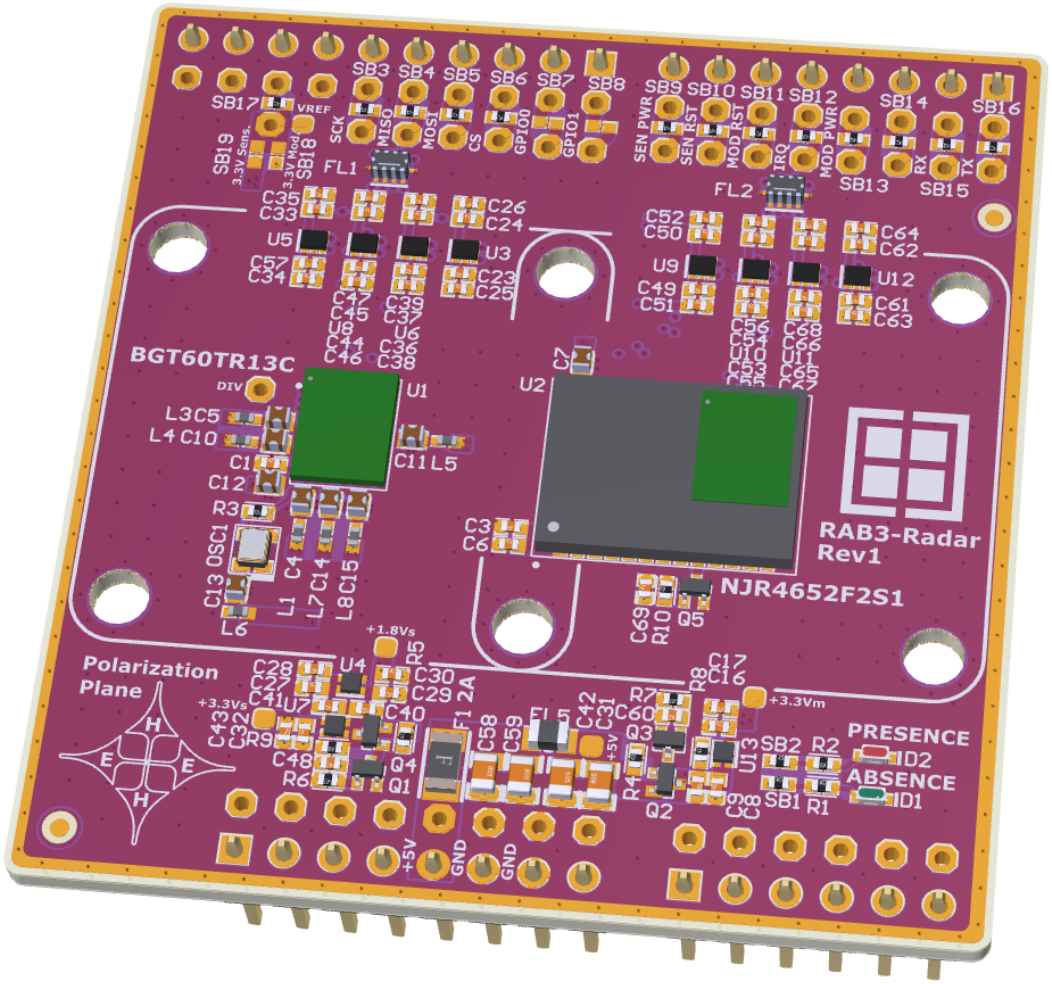
RAB3 – Radar User Manual



Versions

Version	Date	Rationale
0.1	January 08, 2024	First draft. Author: GDR
1.0	February 14, 2024	Corrections. 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

Table of Contents 2
Overview 3
Features 3
Component Placement 4
Hardware 4
NJR4652F2S1 and BGT60TR13C Simultaneous Operation 4
Mounting of Dielectric Lenses 4
Solder Bridges 5
Advanced Noise Reduction of Power Supply 6
TClock Oscillator of BGT60TR13C 6
Fuses 7
Changing the Fuses or Solder Bridges 7
Board Pinout 7
Software and Firmware 8
Getting Started with BGT60TR13C Radar Sensor 8
Radar Fusion GUI 11
Getting Started with NJR4652F2S1 Radar Module 12
Getting Started with BGT60TR13C and Smartphone Apps 16
Getting Started with Infineon Radar Development Kit 16
Running the Python Scripts 18
Running a Demo from ModusToolbox™ IDE 21
Firmware Examples 24
Production Data 25
Schematics 25
BOM 25
Mechanical Layout 25

Overview

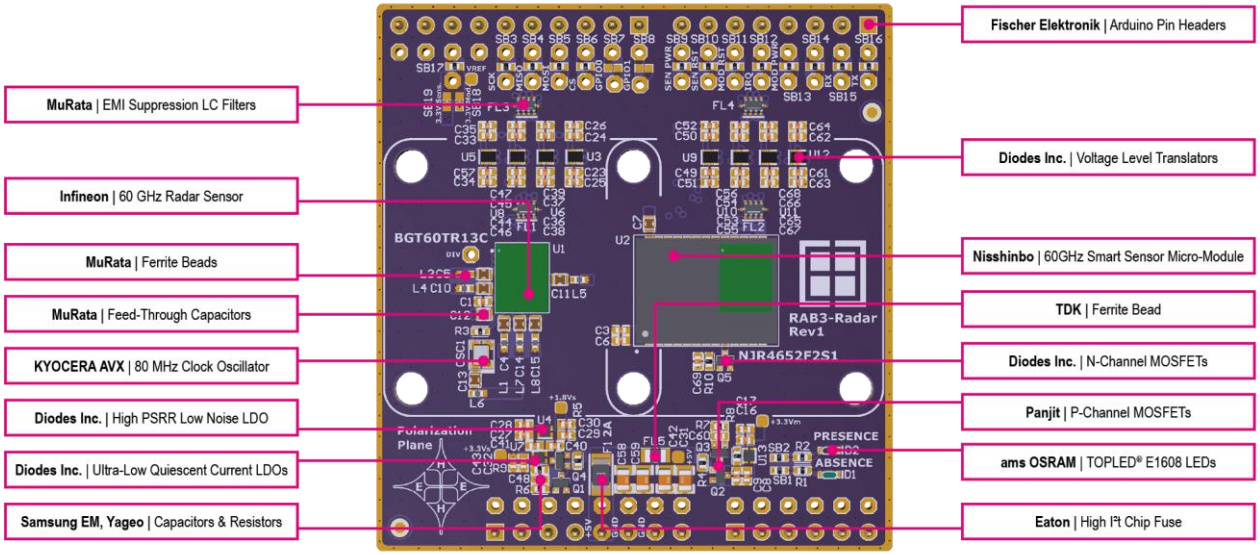
Features

RAB3-Radar is an Arduino Shield - Adapter Board for a wide spectrum of Millimetre wave 60GHz radar sensing applications such as presence detection, person counting, distance, angle and speed measurement, object tracking and gesture recognition.

This adapter board is based on two main components: the Nisshinbo Radar Module NJR4652F2S1 which is already programmed with a presence detection functionality and the Infineon 60GHz Radar Sensor BGT60TR13C. The NJR4652F2S1 has a UART interface at the Arduino Headers and the BGT60TR13C has the SPI interface which is also accessible through the Arduino Headers.

- **BGT60TR13C** – Infineon's XENSIV™ 60 GHz Radar Sensor.
- **NJR4652F2S1** – Nisshinbo's 60 GHz Smart Sensor Micro-Module for Presence Detection.
- **KC2016K** – Kyocera AVX 80 MHz Low-Jitter Clock Oscillator.
- **AP7353** – Diodes Inc. 250mA 1.8V High PSRR Low Noise LDO with Enable.
- **AP7354** – Diodes Inc. 150mA 3.3V Ultra-Low Quiescent Current LDOs with Enable.
- **74LVC1T45** – Diodes Inc. Single Bit Voltage Level Translators.
- **DMG1012T** – Diodes Inc. N-Channel MOSFETs.
- **PJE8403** – PANJIT's P-Channel MOSFETs.
- **TOPLED® E1608** – OSRAM LEDs.
- **MPZ2012S102AT000** – TDK Corporation Ferrite Bead.
- **NFA21SL806X1A48L** – muRata's EMI Suppression LC Filters.
- **NFMJMPC226R0G3D** – muRata's Feed Through Capacitors.
- **BLM15PX601SN1D** – muRata's Ferrite Beads.
- **Arduino Connectors** – Fischer Elektronik SL 20 THR 124 (xx) Z.
- **Passive components** – Passive components from Samsung EM, and Yageo.

Component Placement



Hardware

NJR4652F2S1 and BGT60TR13C Simultaneous Operation

The Nisshinbo radar module NJR4652 and Infineon radar sensor BGT60TR13C were not initially designed by Rutronik to operate at the same time. Nevertheless, the simultaneous operation is possible under certain circumstances if different frequency bands are used by both devices. At all other cases, please enable and operate one device that is actual to your application.

Mounting of Dielectric Lenses

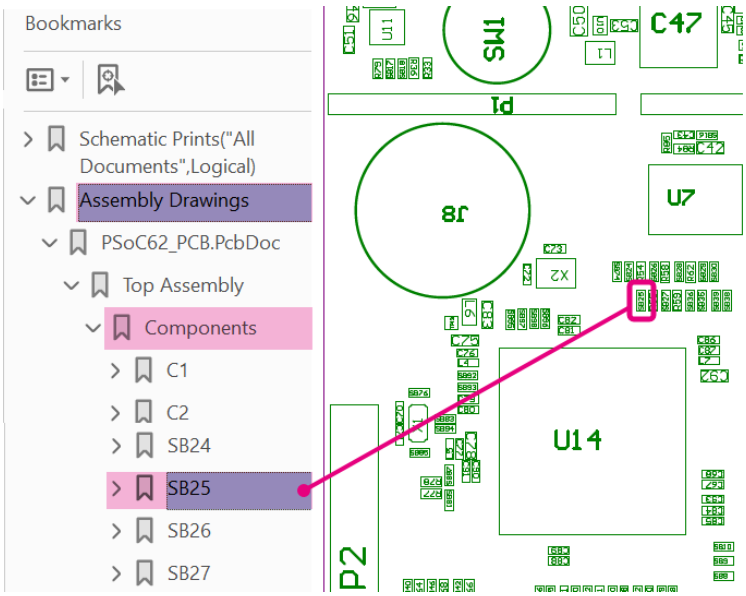
There are 6 holes for the lens mounting provided in the RAB3-Radar adapter board. The sensor and the module are placed at the exact location and polarization to be fitted with the same lens, their height differs slightly though. The dielectric lenses are not provided together with the RAB3-Radar package, and their usage is not supported by Rutronik.

Please refer to Mechanical Drawing if you are required to mount custom lenses.

Solder Bridges

Designator	Circuit	Default
SB1	D1 ABSENCE LED with U2 GPIO_0	Closed
SB2	D2 PRESENCE LED with U2 GPIO_1	Closed
SB3	SCK VREF level with Arduino Header P1 pin 6	Closed
SB4	MISO VREF signal with Arduino Header P1 pin 5	Closed
SB5	MOSI VREF signal with Arduino Header P1 pin 4	Closed
SB6	CS VREF signal with Arduino Header P1 pin 3	Closed
SB7	U2 GPIO_0 with Arduino Header P1 pin 2	Opened
SB8	U2 GPIO_1 with Arduino Header P1 pin 1	Opened
SB9	U1 Sensor Power Control Signal with Arduino Header P5 pin 8	Closed
SB10	U1 Sensor RESET VREF Signal with Arduino Header P5 pin 7	Closed
SB11	U2 Module RESET Signal with Arduino Header P5 pin 6	Closed
SB12	U1 Sensor IRQ VREF Signal with Arduino Header P5 pin 5	Closed
SB13	U2 Module Power Control Signal with Arduino Header P5 pin 4	Closed
SB14	Spare Pin Hole with Arduino Header P5 pin 3	Closed
SB15	U2 Module RX VREF Signal with Arduino Header P5 pin 2	Closed
SB16	U2 Module TX VREF Signal with Arduino Header P5 pin 1	Closed
SB17	VREF Voltage Rail with Arduino Header P1 pin 8	Closed
SB18	U2 Module 3.3V Supply with VREF Voltage Rail	Opened
SB19	U1 Sensor 3.3V Supply with VREF Voltage Rail	Opened

The locations of the solder bridges can be found in 3D model and assembly drawings of RAB3-Radar.



How to find a component on the layout

Advanced Noise Reduction of Power Supply

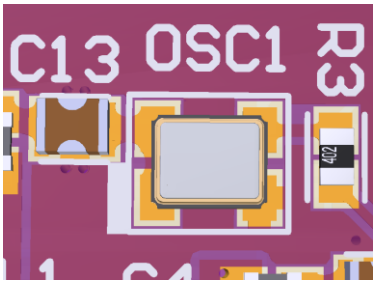
The BGT60TR13C is sensitive to power supply voltage fluctuations, especially in Range-Doppler applications. The RAB3-Radar has been equipped with the electromagnetic noise filtering devices and some special design techniques to potentially meet the harshest environments in terms of electromagnetic interferences:

- 4-Layer PCB layout with all the traces routed on internal layers.
- No Switching Mode Power Supply ICs (SMPS) were used.
- The 1.8V LDO AP7353 has a high PSRR and low output noise.
- +5V power supply rail is filtered by the FL5 MPZ2012S102AT000 and C58, C59, C31, C42 capacitors.
- All the power domains for the U1 are decoupled with Feed-Through 22uF capacitors NFMJMPC226R0G3D.
- In conjunction with Feed-Trough capacitors the ferrite beads BLM15PX601SN1D are used.
- The data lines are also filtered to prevent interference from coming into the system. The NFA21SL806X1A48L will not attenuate the SPI data signals up to 80 MHZ (the declared maximum SPI clock for the BGT60TR13C is 50 MHz).

For price-sensitive applications where high EMI is not expected, please downgrade the design accordingly. The PCB may have only 2 layers, the data line filters could be excluded and Feed-Through capacitors may be replaced with cheaper classic design alternatives.

TClock Oscillator of BGT60TR13C

The Kyocera AVX KC2016K80.0000C1GE00 80MHz clock oscillator OSC1 is used for the current design as it has a low phase jitter of 1ps@25MHz (BW: 12kHz to 20MHz) and acceptable frequency stability of 50 ppm. Please always use the oscillators that have a phase jitter as low as possible.



Fuses

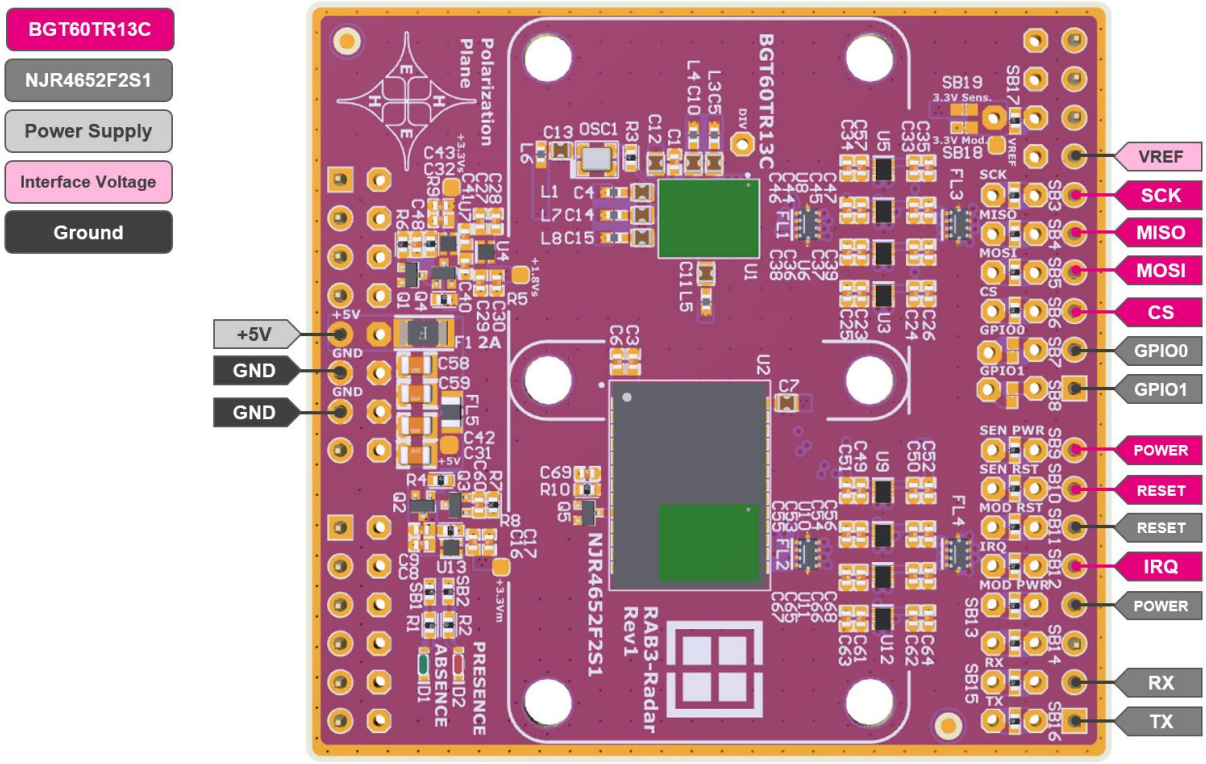
The RAB3-RADAR board has only one 2A fast acting fuse F1 in a 1206 package; Part No: CC12H2A-TR „Eaton“.

Changing the Fuses or Solder Bridges

The SMD „Chipping Tool - Tweezers“ is recommended for use for SMD solder bridges or fuses soldering on the RAB3-RADAR development board.

Board Pinout

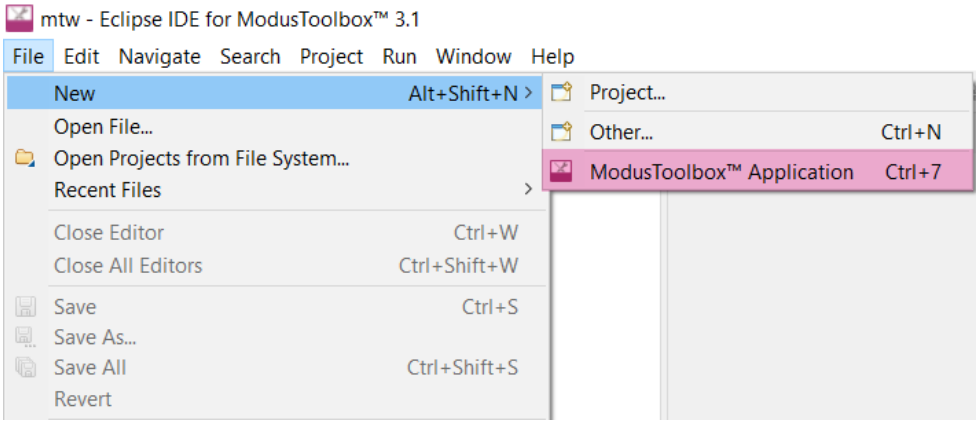
The voltage level of UART and SPI interfaces including the IRQ signal is adjustable through the VREF pin. The adjustable voltage range is from 1.65V up to 5.5V. The GPIO0 and GPIO1 are always 3.3V.



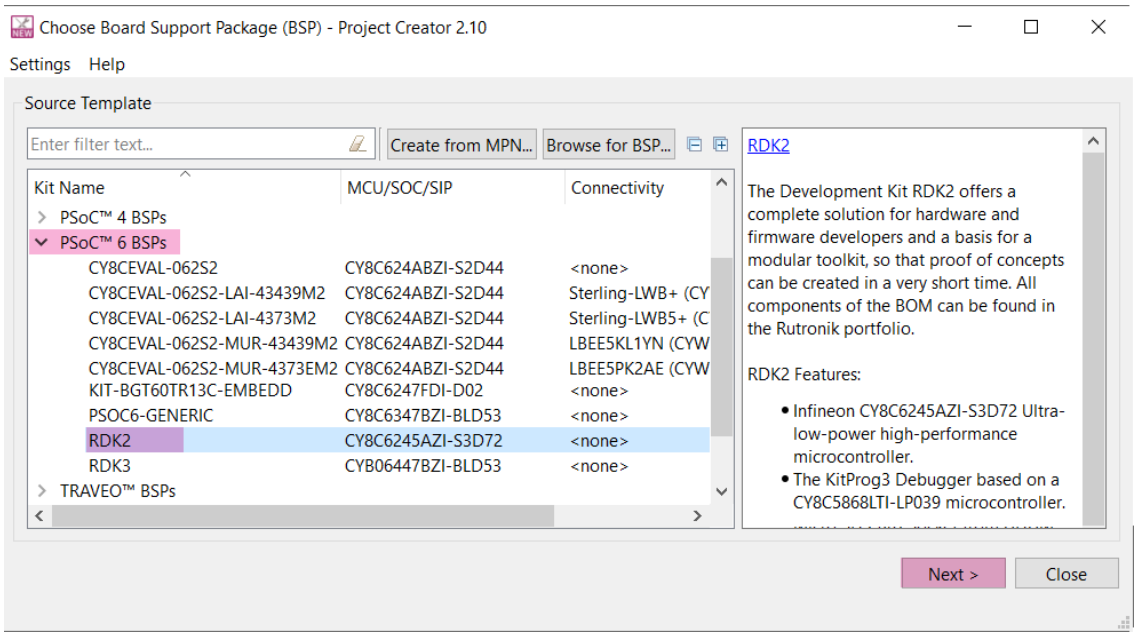
Software and Firmware

Getting Started with BGT60TR13C Radar Sensor

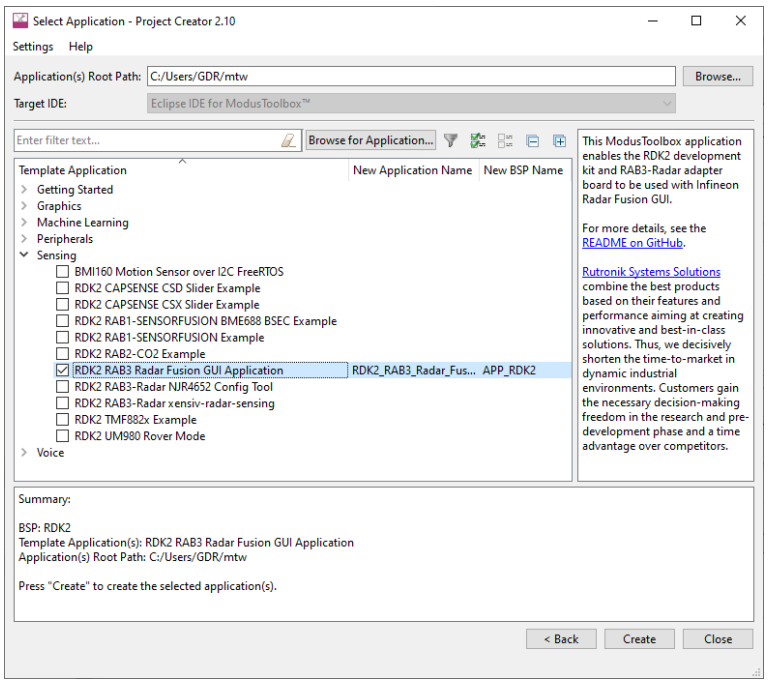
1. Please acquire the RDK2 development kit first. Check the Rutronik24 or inquiry from the Rutronik System Solutions by sending a question.
2. Register or/and login at Infineon website (myInfineon tab).
3. Download and install the latest version of ModusToolbox™ software.
4. Download and install the latest version of Radar Fusion GUI software.
5. Mount the RAB3-Radar board on the RDK2 Arduino headers.
6. Ensure the switch SW1 of RDK2 is set to “3.3V” and connect the Micro USB cable (A to Micro B) to “KitProg3”. Connect the RDK2 and RADAR assembly with a PC via cable.
7. Run **Modus Toolbox** application.
8. Go **File** - **New** – **Modus Toolbox Application** and wait for a while.



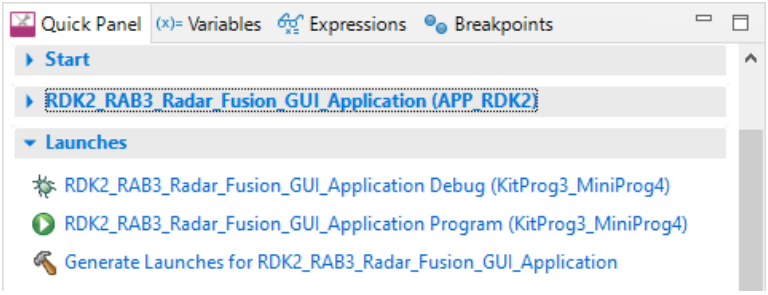
9. Open **PSoC 6** block, select **RDK2** and press **Next**.



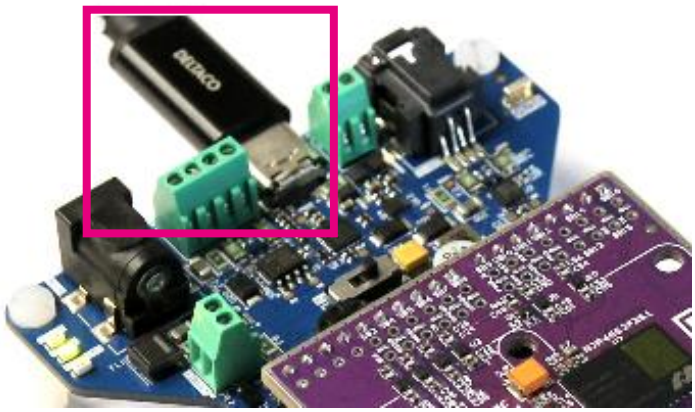
10. Open **Sensing** block, check **RDK2 RAB3 Radar Fusion GUI Application** and press **Create**.



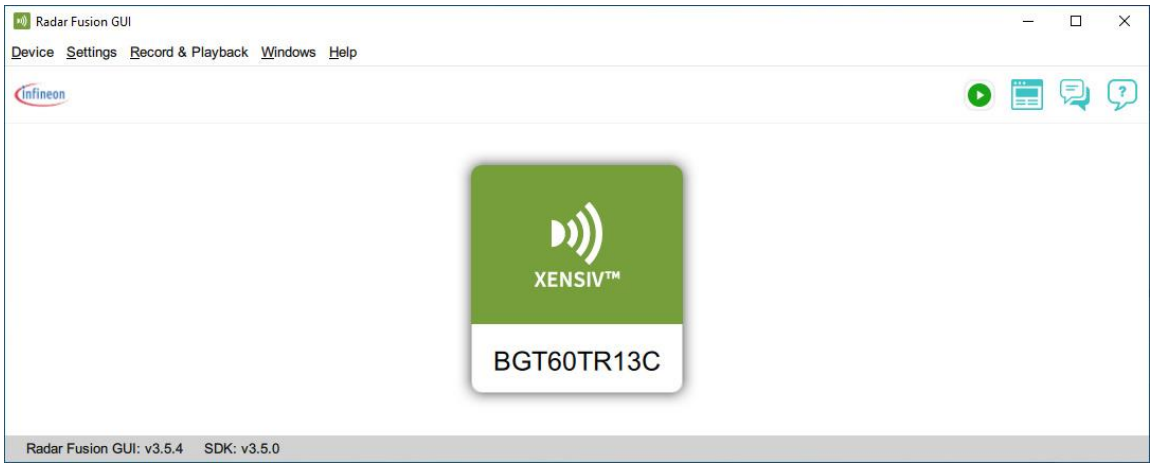
11. **Build** and **Debug** the project.



12. Disconnect from KitProg3 and connect a USB PC cable with a RDK2 USB-C socket.

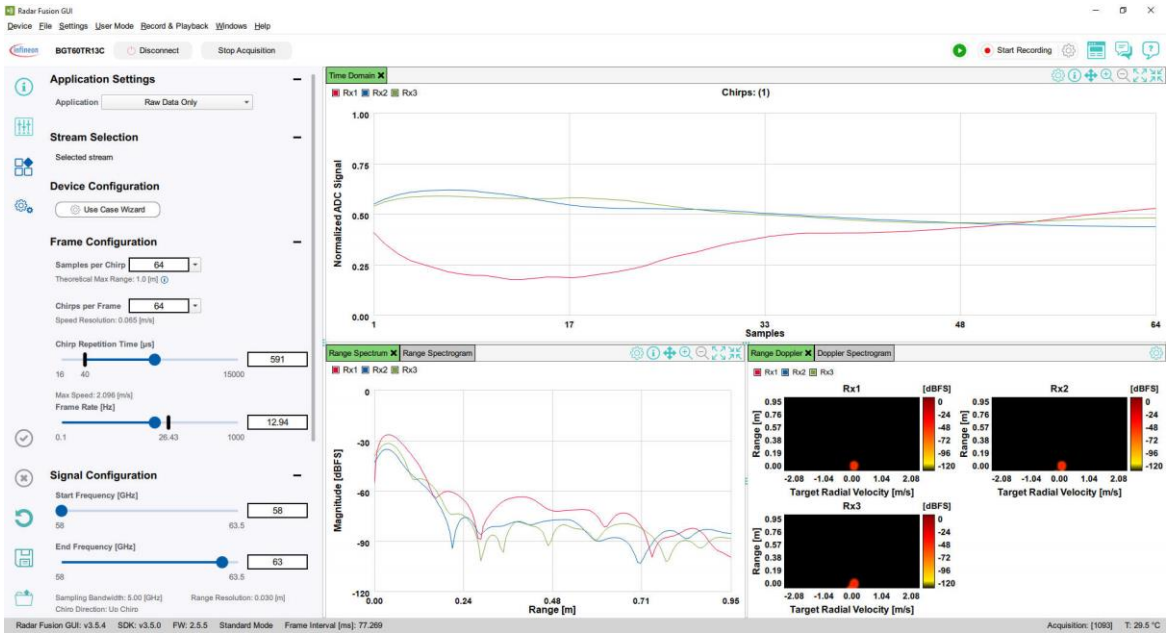


13. Start your Radar Fusion GUI and click on detected sensor BGT60TR13C.



14. Operate the Radar Fusion GUI as you would normally do with an [IRadar Baseboard MCU 7].

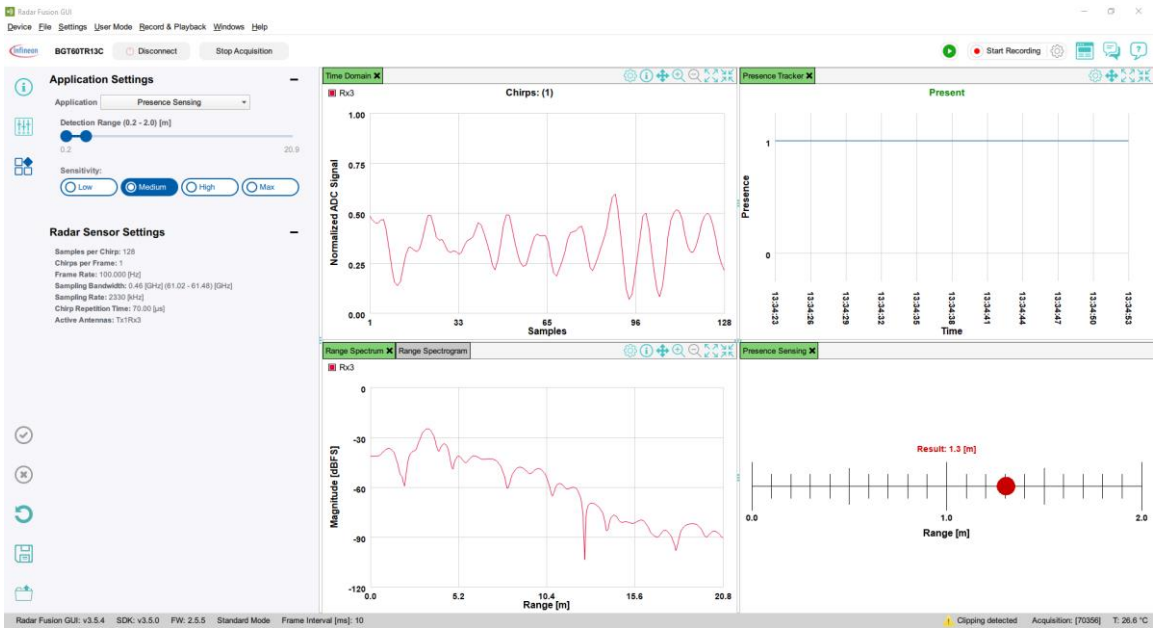
NOTICE: Please keep in mind that due to the RDK2 SPI and USBFS hardware limitations, the data transfers are slower than with IRadar Baseboard MCU7. Try to keep lower data rate as possible by analyzing the necessary signal only, keeping the Frame Rate, Samples per Chirp, and Chirps per Frame at a minimum.



Radar Fusion GUI

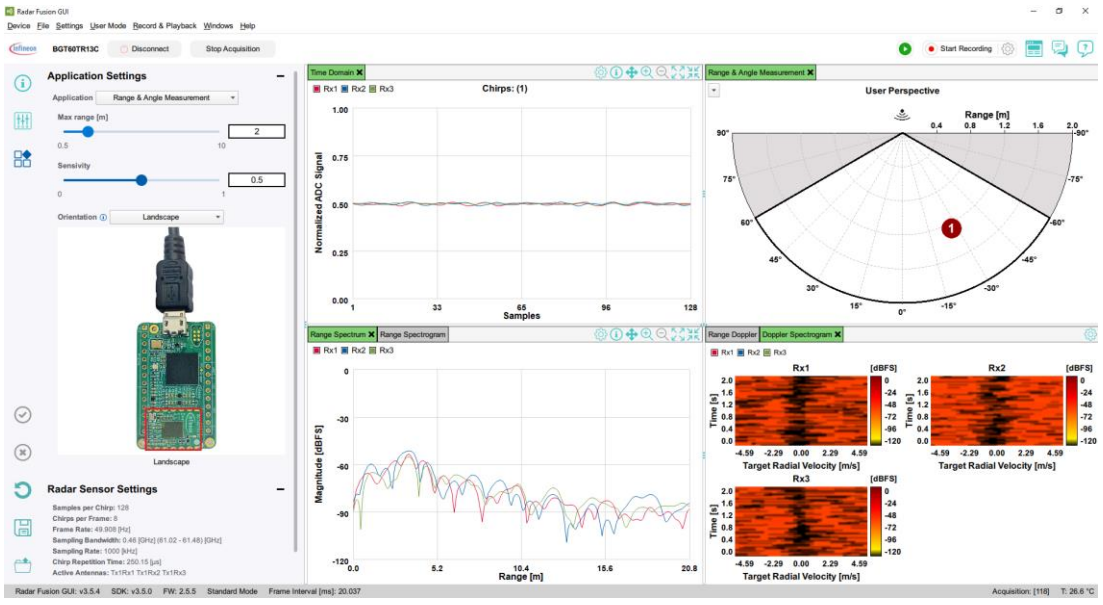
Presence Sensing

The presence sensing can be evaluated using “Presence Sensing” application in Radar Fusion GUI. Please choose the Presence Sensing application, decrease the detection distance down to 2.0 meters as for a start and click on the “Apply All Settings” icon .



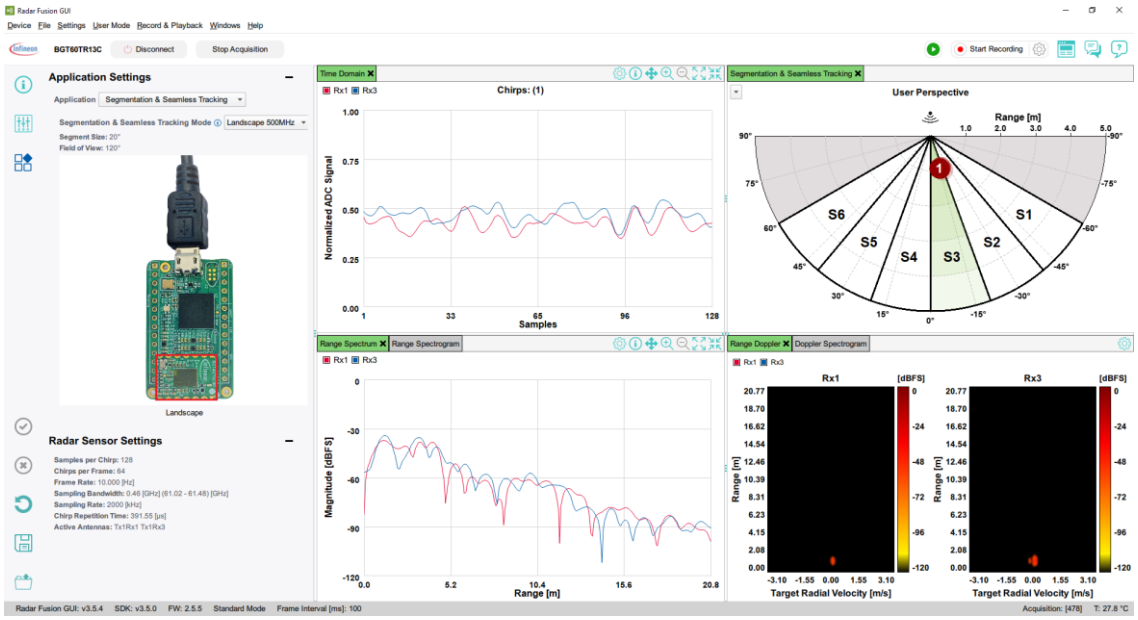
Range and Angle Measurement

The range and angle to the particular object can be measured using “Range & Angle Measurement” application in Radar Fusion GUI. This particular application requires quite a large amount of data to be transferred from the sensor to the PC, hence the RDK2 will not operate stable for a long time with Radar Fusion GUI.



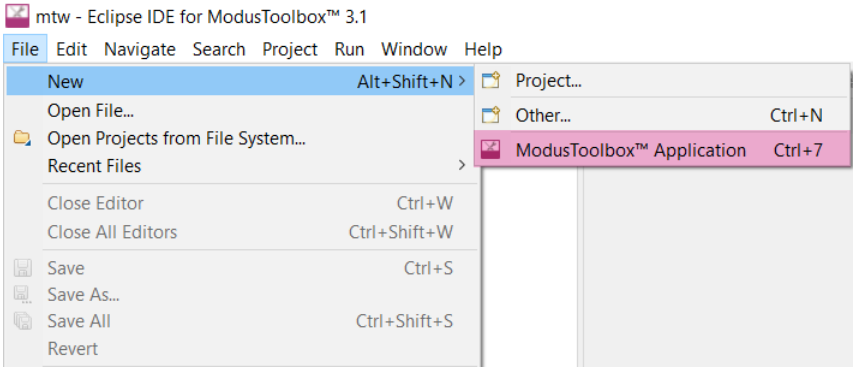
Segmentation and Seamless Tracking

The object detection and tracking in defined areas can be evaluated using the “Segmentation & Seamless Tracking” application in Radar Fusion GUI. More than one object can be observed at once.

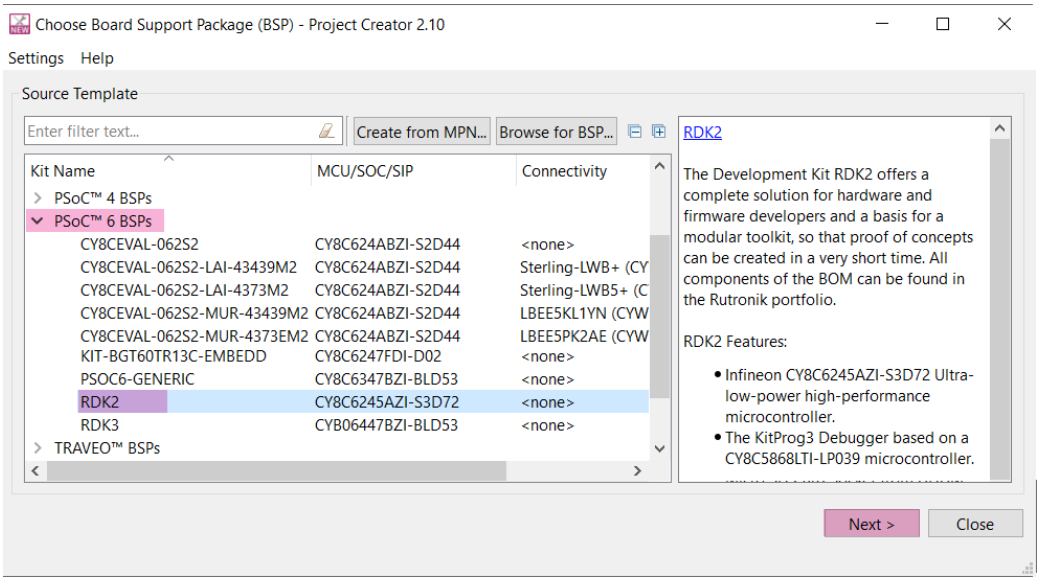


Getting Started with NJR4652F2S1 Radar Module

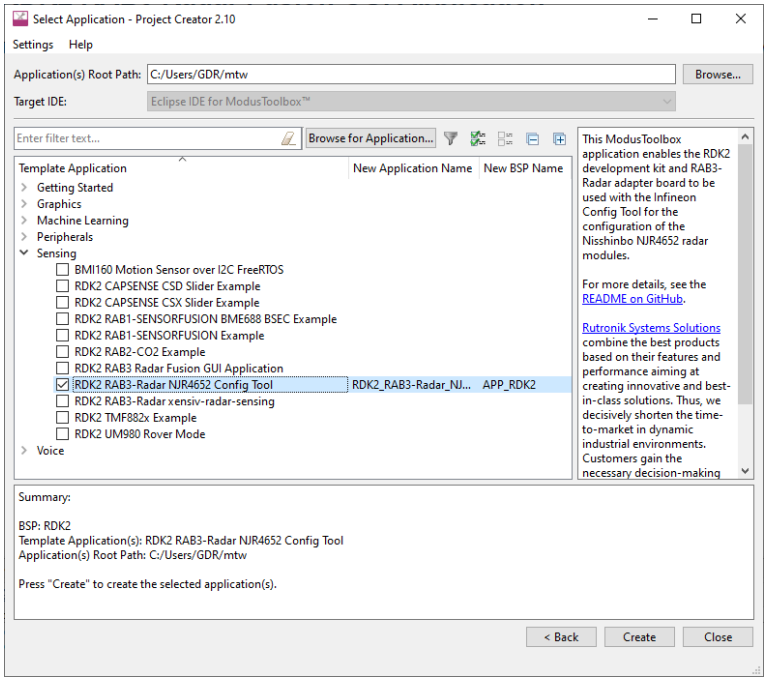
1. Please acquire the RDK2 development kit first. Check the Rutronik24 or inquiry from the Rutronik System Solutions (check the Contact form at the bottom of the page).
2. Register or/and login at Infineon website (myInfineon tab).
3. Download and install the latest version of ModusToolbox™ software.
4. Download and install the latest version of BGT60TR13C Config Tool software.
5. Mount the RAB3-Radar board on the RDK2 Arduino headers.
6. Run **ModusToolbox™** application.
7. Go **File** - **New** – **ModusToolbox™ Application** and wait for a while.



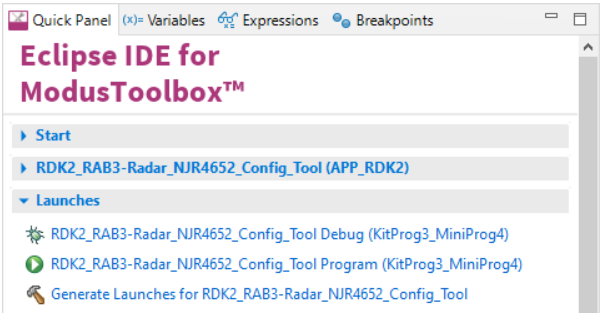
8. Open **PSoC 6** block, select **RDK2** and press **Next**.



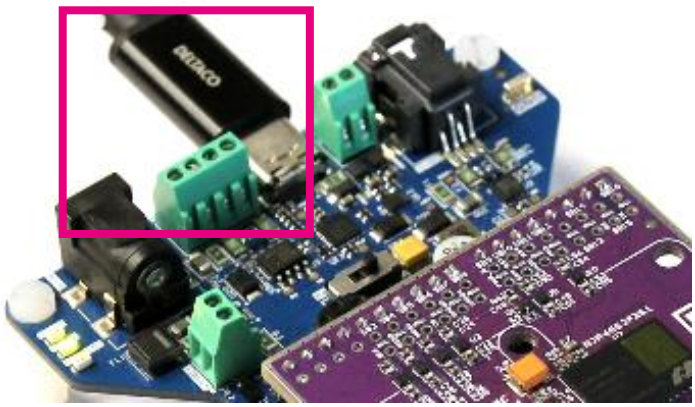
9. Open **Sensing** block, check **RDK2 RAB3-Radar NJR4652 Config Tool** and press **Create**.



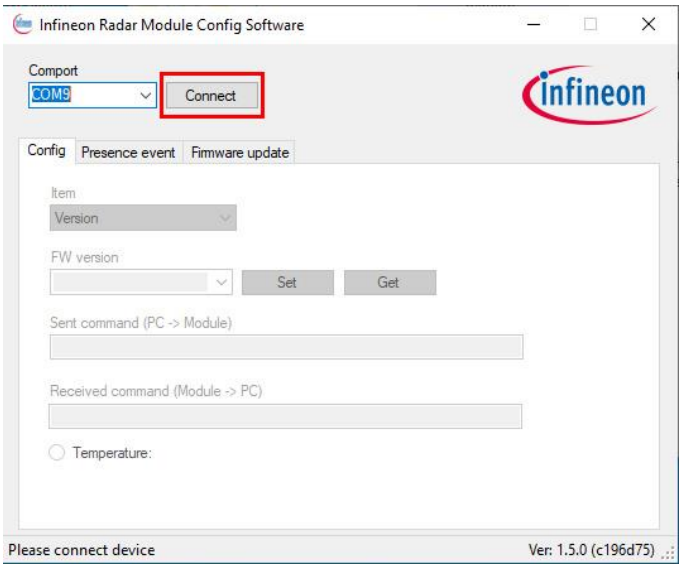
10. **Build** and **Debug** the project.



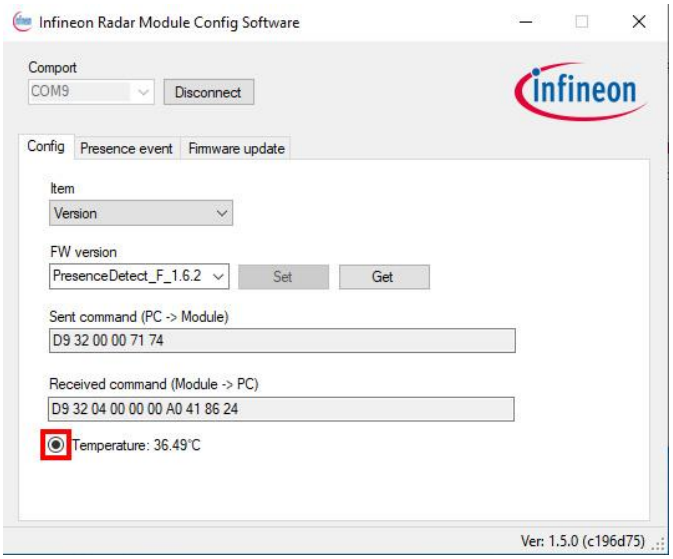
11. Disconnect from KitProg3 and connect a USB PC cable with a RDK2 USB-C socket.



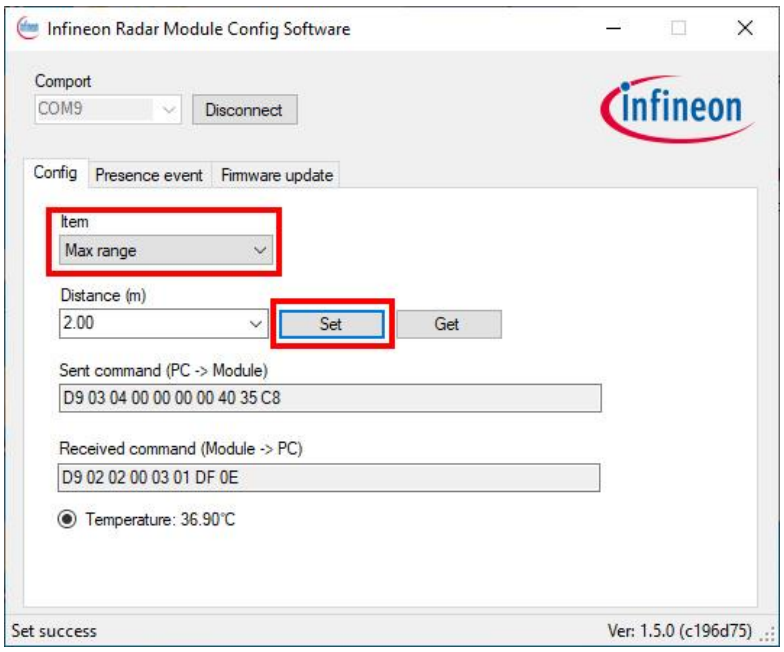
12. Start Infineon Config Tool and choose a COM port.

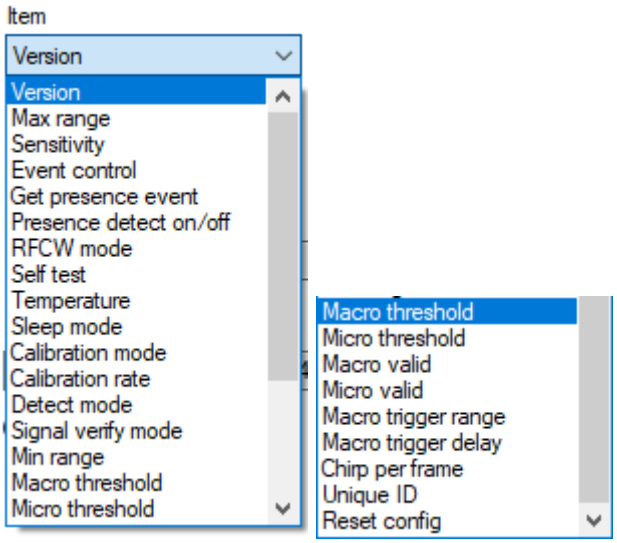


13. Click on the temperature point to be sure that communication with the NISSHINBO module works in both directions.

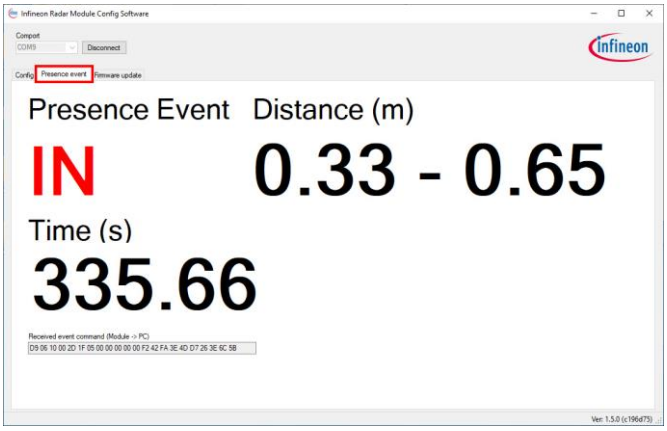


14. You may change the *Max Range* and other items from dropdown menu as it is shown below:



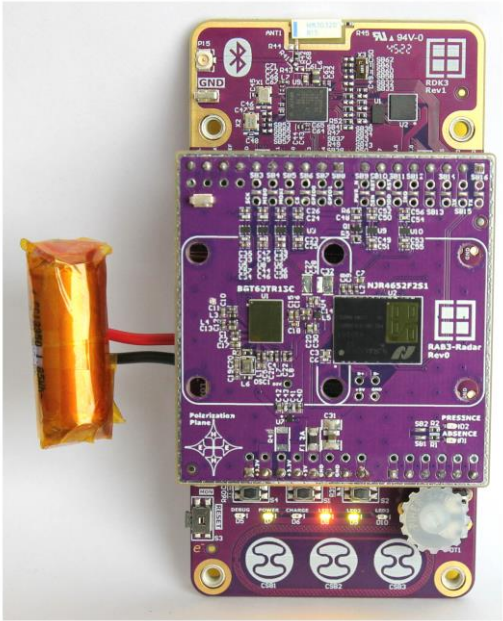


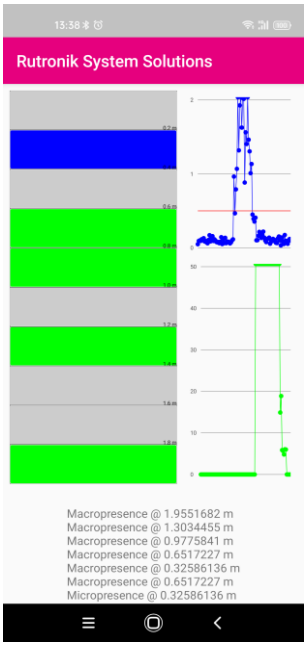
15. The presence or absence states may be observed as it is shown below:



Getting Started with BGT60TR13C and Smartphone Apps

Coming soon



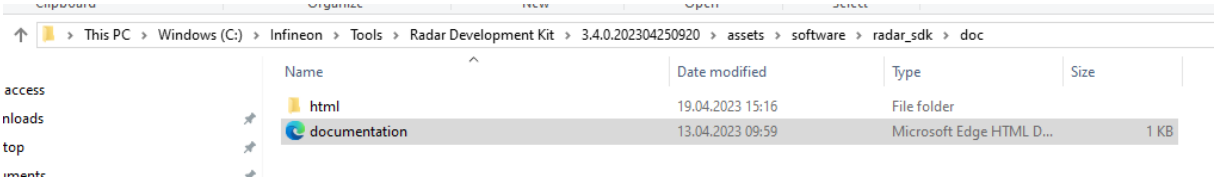


Getting Started with Infineon Radar Development Kit

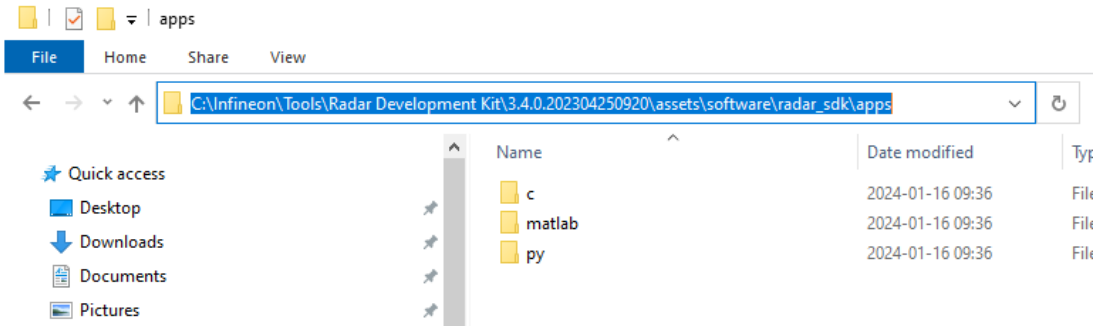
The quick start with Infineon Radar Baseboard MCU7 is described here. A quick start with Rutronik Development Kit RDK2 is no different except the hardware part:

1. Please acquire the RDK2 development kit first. Check the Rutronik24 or inquiry from the Rutronik System Solutions (check the Contact form at the bottom of the page).
2. Register or/and login at Infineon website (myInfineon tab).
3. Download the 3.4.0.202304250920 Infineon Radar Development Kit software package from here.
4. Install and Launch the Radar Development Kit software, you may check the *Quickstart* and other information that may be actual for you.

Please also examine the SDK documentation from here:



5. You may use C make, MATLAB, or Python examples from your local directory:
   C:\Infineon\Tools\RadarDevelopmentKit\3.4.0.202304250920\assets\software\radar_sdk\apps

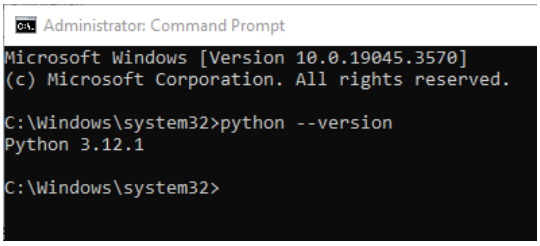


Running the Python Scripts

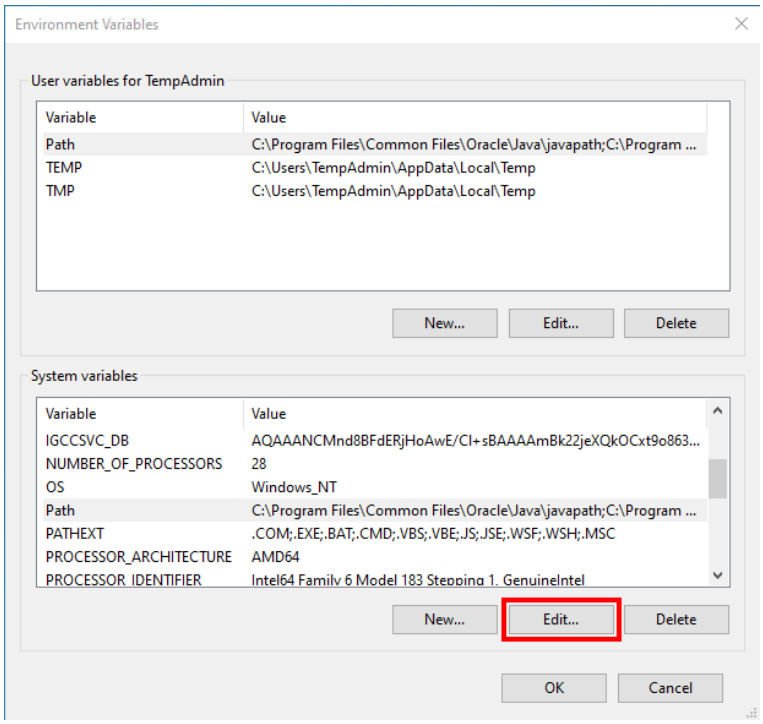
Let's start with Python script example "*static_distance.py*" from Radar Development Kit in Windows OS:

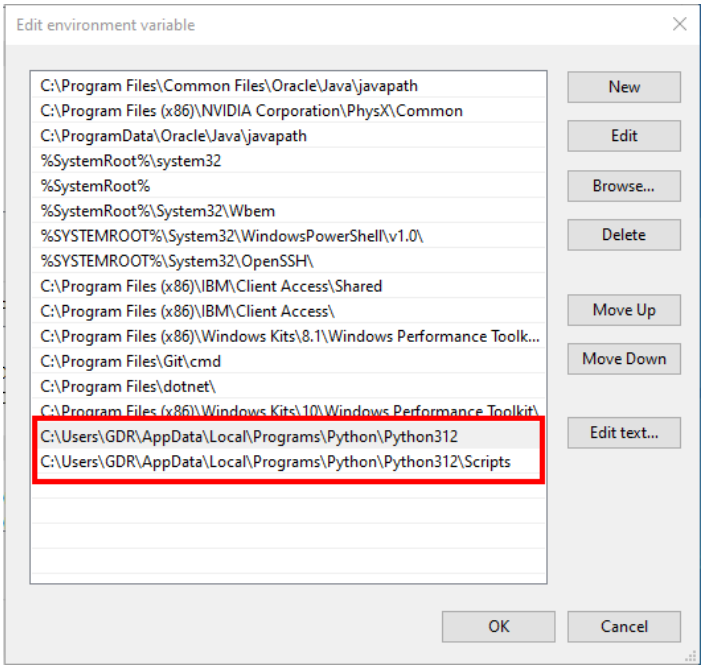
1. Prepare the RDK2 and RAB3-Radar as it is done in the chapter "Getting Started with BGT60TR13C Radar Sensor".
2. Open Windows Command Prompt with the Administrating privileges.
3. Check if you have Python on you PC by typing a command:

```
python --version
```



If you do not have a Python yet, download, install and make sure the highlighted values of Path variable are in the values list at your PC.





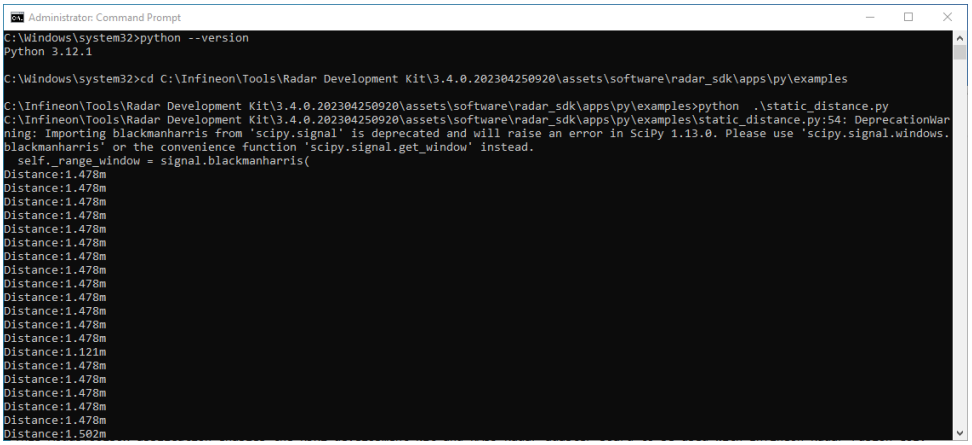
4. Check your path to the “Radar Development Kit” and execute the commands given below. Be sure you have no networking restrictions or use the unrestricted network during the installation of the following packages.

```
cd C:\Infineon\Tools\Radar Development
Kit\3.4.0.202304250920\assets\software\radar_sdk\libs\win32_x64

pip install ifxAvian-3.4.0-py3-none-any.whl
pip install ifxDopplerLTR11-3.4.0-py3-none-any.whl
pip install numpy
pip install scipy
pip install matplotlib
```

More packages might be required to install depending on your current PC setup.

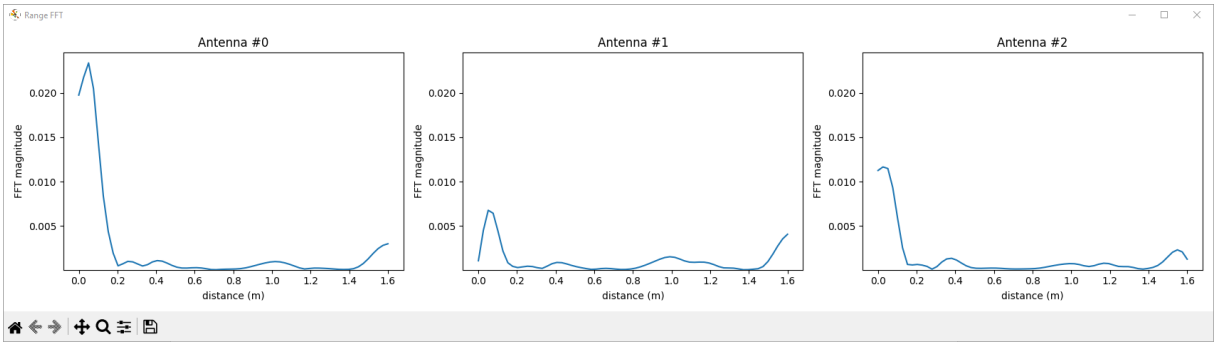
5. Go to the Python examples directory and run the static distance measurement script as it is shown:

```
cd C:\Infineon\Tools\Radar Development
Kit\3.4.0.202304250920\assets\software\radar_sdk\apps\py\examples

python .\static_distance.py
```

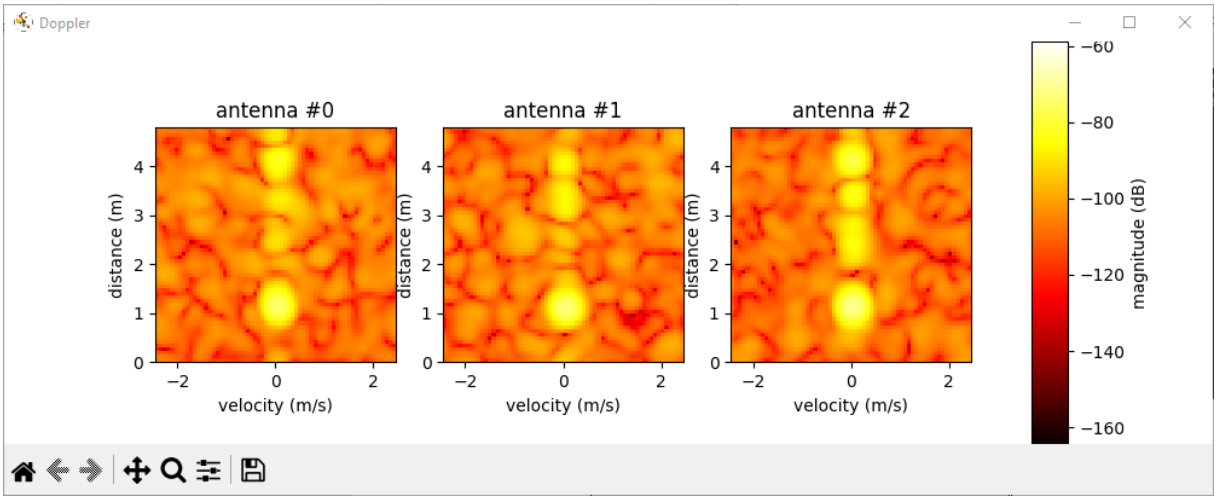


6. Please check that other examples are available:

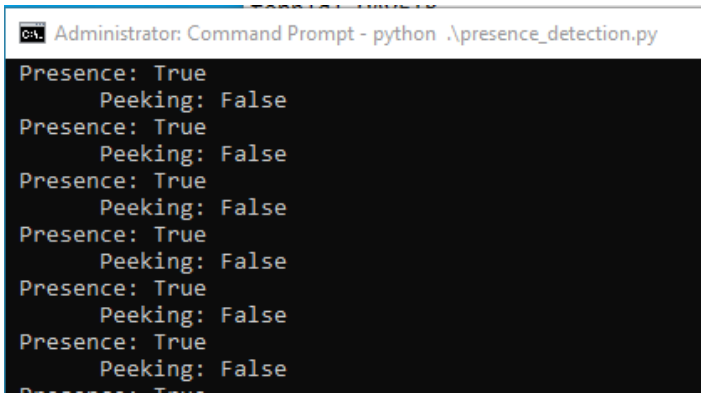
```
python .\distance_fft.py
```



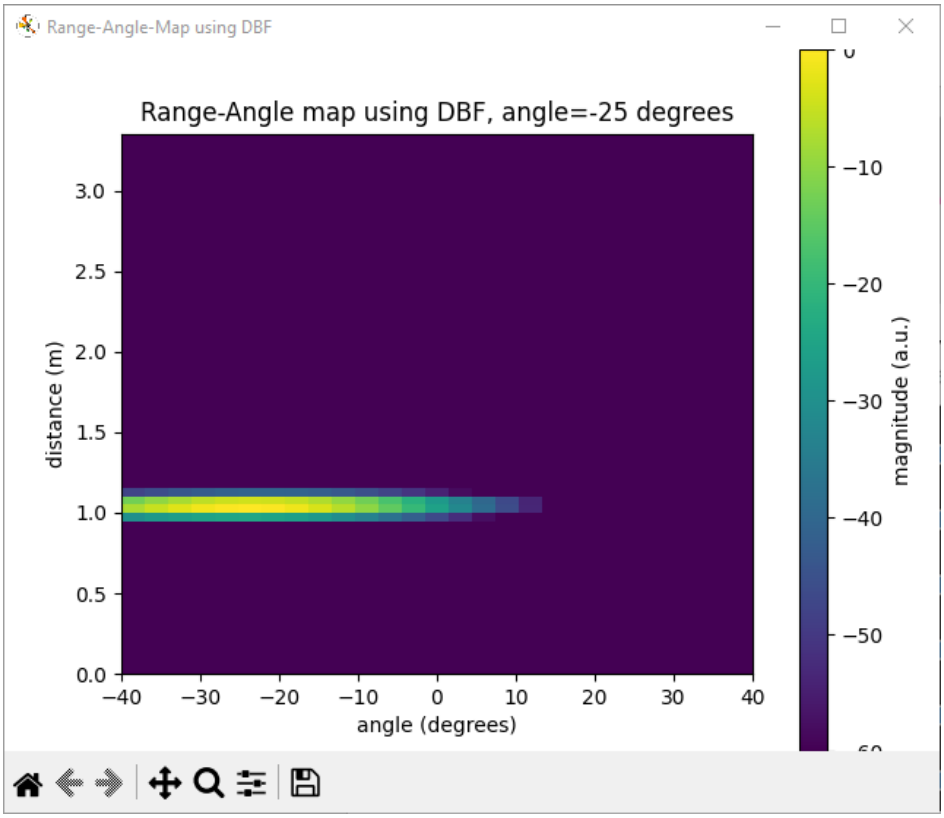

```
python .\doppler.py
```




```
python .\presence_detection.py
```



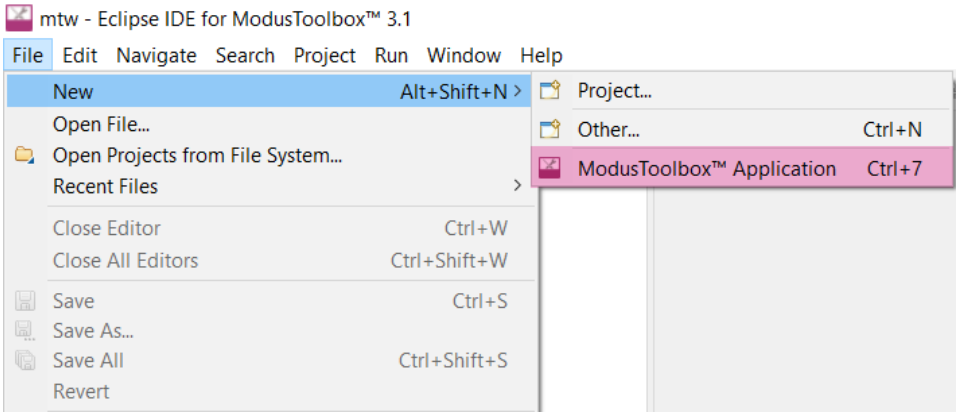
```
python .\range-angle-map.py
```



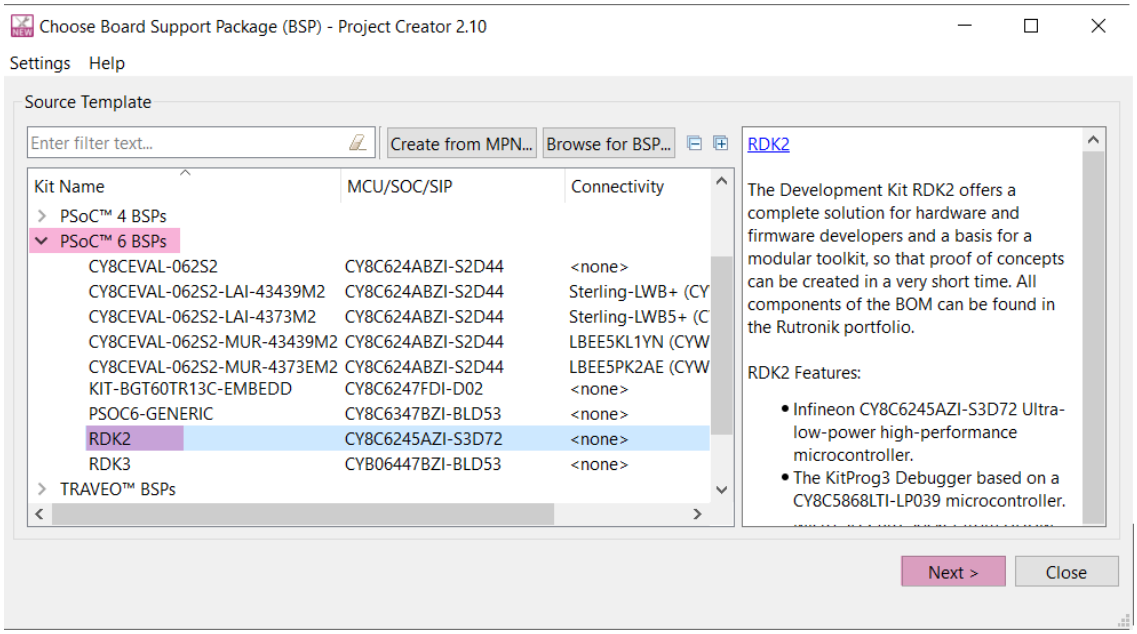
Running a Demo from ModusToolbox™ IDE

RDK2_RAB3_Radar_Sensing demo might be not the only one code example available by the time you read this. Please check our GitHub or ModusToolbox™ Project Creator for more RAB3-Radar examples.

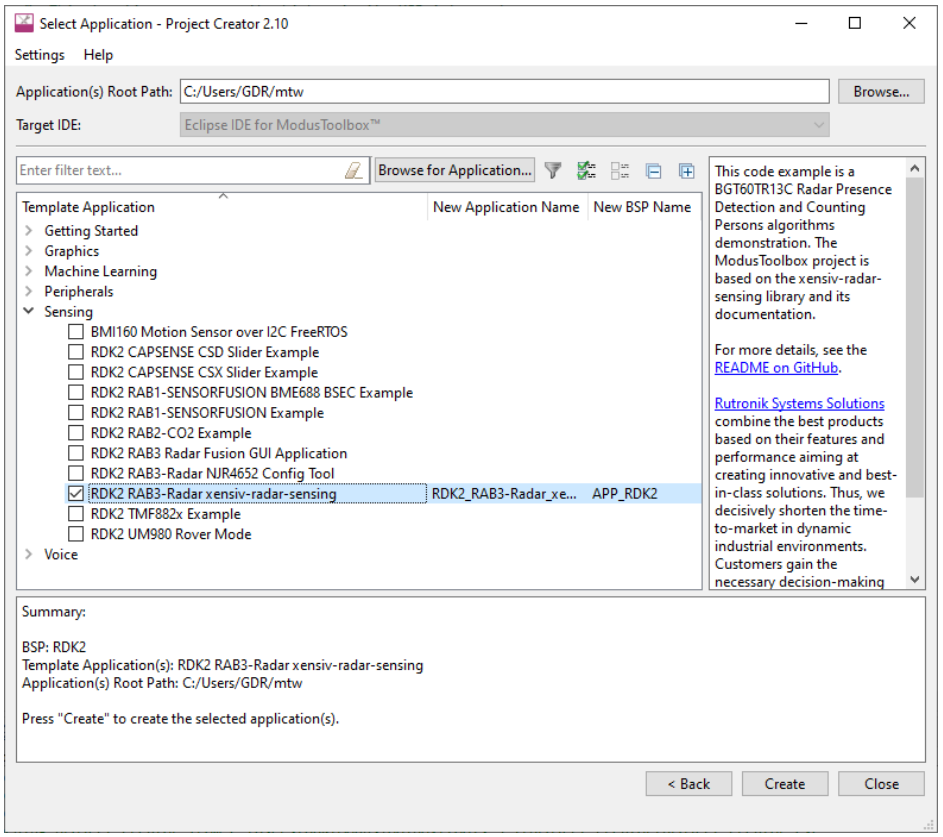
1. Run the **ModusToolbox™** application.
2. Go **File** - **New** – **ModusToolbox™ Application** and wait for a while.



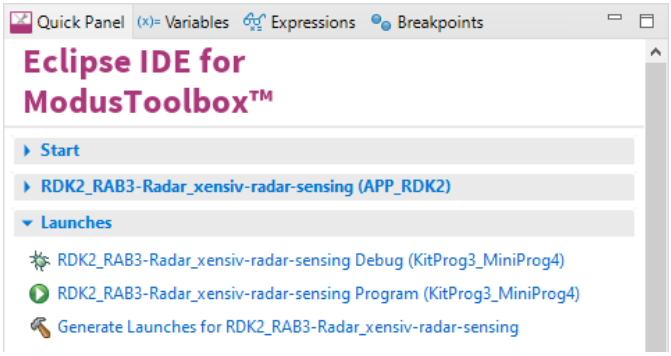
3. Open **PSoC 6** block, select **RDK2** and press **Next**.



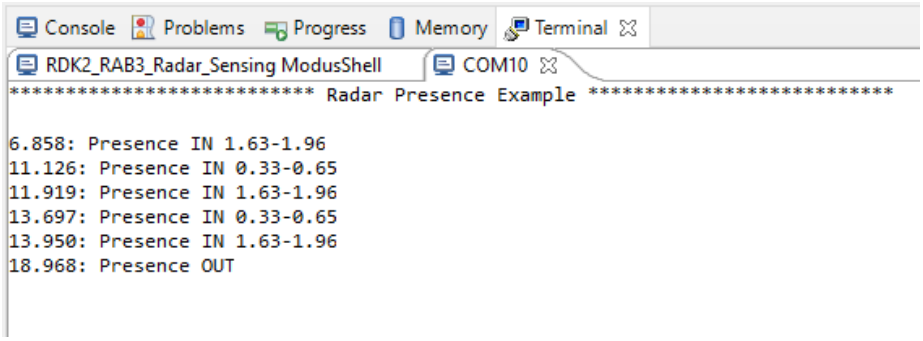
4. Open **Sensing** block, check **RDK2 RAB3 Radar Fusion GUI Application** and press **Create**.



5. **Build** and **Debug** the project.

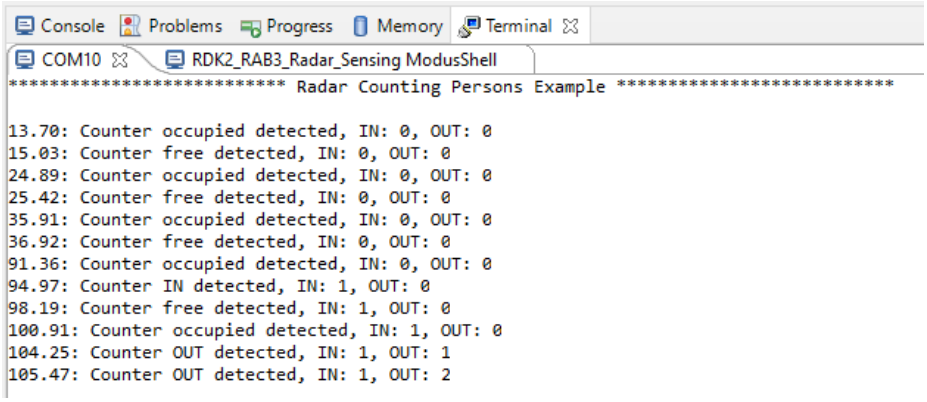


6. The firmware example uses KitProg3 Debug UART for debug output. LED1 and LED2 are used to indicate the presence or absence of states in Presence Detection mode.



7. The Counting Persons mode is enabled by uncommenting the definition as shown below.

```
/*******************************************************************************
 * Macros
 ******************************************************************************/
//#define PRESENCE_DETECTION
#define COUNTING_PERSONS
```



Firmware Examples

All these examples can be found at GitHub and are available directly from Project Creator in ModusToolbox™ IDE.

RDK2_RAB3_Radar_Fusion	This ModusToolbox™ application enables the RDK2 development kit and RAB3-Radar adapter board to be used with Infineon Radar Fusion GUI.
RDK2_RAB3_Config_Tool	This ModusToolbox™ application enables the RDK2 development kit and RAB3-Radar adapter board to be used with the Infineon Config Tool for the configuration of the NISSHINBO NJR4652 radar modules.
RDK2_RAB3_Radar_Sensing	This code example is a BGT60TR13C Radar Presence Detection and Counting Persons algorithms demonstration. The ModusToolbox™ project is based on the xensiv-radar-sensing library and its documentation.
RDK3_Radar_Presence_Detection	This code example demonstrates the Infineon XENSIV™ Presence Detection algorithm. It is used together with Android OS and iOS smartphone applications

Production Data

Schematics

You'll find the schematics of RAB3-Radar here.

BOM

You'll find the BOM for RAB3-Radar here.

Mechanical Layout

You'll find the Drawing for RAB3-Radar here.