

24 GHz radar tools and development environment user manual

Infineon XENSIV™ 24 GHz radar demo kits

About this document

Scope and purpose

This user manual describes the software and firmware environment required to use the radar applications offered with Infineon's industrial XENSIV™ 24 GHz radar demo kits: Distance2Go, Position2Go, Sense2GoL and Sense2GoL Pulse.

It provides guidelines for novice users on how to build and run smart radar solutions, from basic movement detection to advanced motion detection and sensing, made easy with Infineon's 24 GHz radar chipsets, and using the Infineon firmware and software tools.

Intended audience

This document is intended for users of Infineon's 24 GHz radar demo boards who want to get started with Infineon's development kits, test several sensing demonstrations, and implement custom radar applications in the 24 GHz ISM band.

Related documents

Additional information can be found in the supplementary documentation provided with the 24 GHz radar kits (Distance2Go, Position2Go, Sense2GoL and Sense2GoL Pulse) in the Infineon Toolbox or from

www.infineon.com/24GHz:

- D2G Application Note (AN543)
- P2G Application Note (AN553)
- P2G Software User Manual
- S2GL Application Note (AN597)
- S2GLP Application Notes (AN598 – AN602)
- S2GLP Software User Manual

Table of contents

About this document.....	1
Table of contents.....	2
List of figures	3
List of tables	4
1 Introduction	5
2 Running radar applications	6
2.1 Tools and software set-up	6
2.1.1 Infineon Toolbox	6
2.1.2 24 GHz radar tools	6
2.1.3 XMC™ Flasher.....	10
2.1.4 DAVE™ toolchain	13
2.2 Graphical User Interface (GUI) solutions	13
2.2.1 Radar GUI.....	13
2.2.1.1 Overview.....	13
2.2.1.2 How to use.....	14
2.2.2 Micrium µC/Probe™ XMC™ GUI	18
2.2.2.1 Overview.....	18
2.2.2.2 How to use.....	19
3 Customizing radar applications	21
3.1 Building, Flashing and debugging the DAVE™ project	21
3.2 Firmware customization and configuration.....	26
4 Extracting radar raw data	27
4.1 Radar host communication protocol	27
4.2 MATLAB radar system interface	28
4.2.1 Overview	28
4.2.2 How to use	28
4.3 C radar system interface	30
4.3.1 Overview	30
4.3.2 How to use	30
4.4 UART radar system interface	31
4.4.1 Overview	31
4.4.2 How to use	31
5 Authors	33
6 References	34
Revision history.....	35

List of figures

List of figures

Figure 1	Infineon XENSIV™ 24 GHz radar demo kits	5
Figure 2	Run Infineon Toolbox	6
Figure 3	Infineon Toolbox tools launcher	7
Figure 4	Infineon Toolbox tools installation	7
Figure 5	Radar Toolbox tools license agreement	8
Figure 6	Kit installation progress	8
Figure 7	24 GHz radar kits – installed tools	9
Figure 8	Start XMC™ Flasher tool via the Infineon Toolbox	10
Figure 9	XMC™ Flasher device selection and connection	11
Figure 10	Change debugger type	11
Figure 11	Binary image file selection	12
Figure 12	Binary image programming	12
Figure 13	Programming successful	13
Figure 14	Infineon Toolbox launcher	14
Figure 15	Start the Radar GUI application via Infineon Toolbox	15
Figure 16	New firmware version available window	15
Figure 17	Radar GUI application window	16
Figure 18	Firmware flashing via Radar GUI	17
Figure 19	Firmware flashing via Radar GUI	17
Figure 20	Firmware flashing via Radar GUI	17
Figure 21	Open Sense2GoL GUI	19
Figure 22	Run the Sense2GoL GUI	19
Figure 23	Start data collection	20
Figure 24	Data display on Micrium GUI	20
Figure 25	DAVE™ workspace launcher	21
Figure 26	DAVE™ project import dialog box	22
Figure 27	Import DAVE™ projects	22
Figure 28	DAVE™ editor view	23
Figure 29	Build project options	23
Figure 30	DAVE™ console	24
Figure 31	Start debug session toolbar button	24
Figure 32	Debug configuration	25
Figure 33	DAVE™ debug perspective	25
Figure 34	MATLAB radar system interface	28
Figure 35	Raw data acquisition in MATLAB command window	29
Figure 36	Raw data acquisition in Visual Studio shell	30
Figure 37	Configure PuTTY for raw data acquisition	31
Figure 38	Raw data acquisition in PuTTY shell	32



Introduction

List of tables

Table 1	XMC™ devices embedded in 24 GHz radar boards	10
Table 2	Define statements used for radar firmware configuration	26

Introduction

1 Introduction

The 24 GHz radar demo kits from Infineon are demonstration platforms as well as starter kits for Infineon's silicon-germanium (SiGe) based 24 GHz transceiver chipset BGT24 and 32-bit ARM® Cortex™-M based XMC™ microcontrollers.

- **Distance2Go (D2G)** – BGT24MTR11 + XMC4200

Infineon radar demo board based on the BGT24MTR11 – Frequency Modulated Continuous Wave (FMCW) and Doppler (distance, speed and direction of movement detection).

- **Position2Go (P2G)** – BGT24MTR12 + XMC4700

Infineon radar demo board based on the BGT24MTR12 using fast-chirp FMCW for tracking (angle, distance, speed and direction of movement detection).

- **Sense2GoL (S2GL)** – BGT24LTR11 + XMC1302

Infineon radar demo board based on the BGT24LTR11 – Doppler (motion, speed and direction of movement detection).

- **Sense2GoL Pulse (S2GLP)** – Radar Baseboard XMC4700 and BGT24LTR11 Shield with Arduino compatibility

Infineon radar demo board based on the BGT24LTR11 – Doppler (motion, speed and direction of movement detection) with extremely low power consumption.

The 24 GHz radar demo kits provide a complete evaluation platform for radar systems including demonstration firmware and a highly interactive Graphical User Interface (GUI).

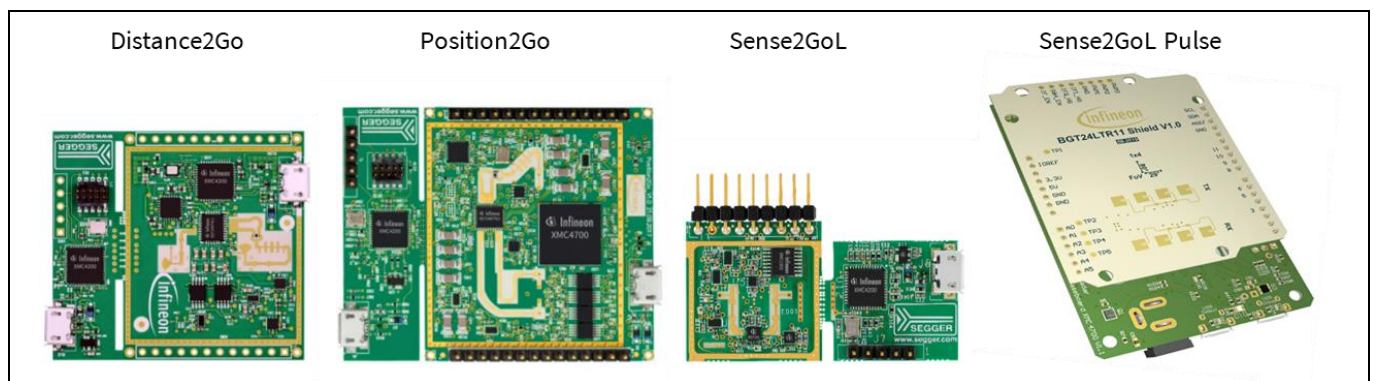


Figure 1 Infineon XENSIV™ 24 GHz radar demo kits

Users can develop their own radar-based application firmware utilizing Infineon's powerful, free-of-charge toolchain DAVE™ for microcontroller programming, customize radar applications built on generated DAVE™ code, and run them on 24 GHz radar demo kit.

Learn more about Infineon's 24 GHz products at: www.infineon.com/24GHz

2 Running radar applications

With each 24 GHz radar, Infineon offers various radar demonstration applications to demonstrate the radar kits' capabilities and to facilitate the development of user applications that can be used to detect:

- Motion, speed and direction of movement of multiple targets (approaching or retreating)
- Distance of multiple targets
- Position of multiple targets

2.1 Tools and software set-up

2.1.1 Infineon Toolbox

Before using the 24 GHz radar demo kit, as well as running the radar demonstration application, it is necessary to download the supporting software from Infineon.

In order to install and use Infineon plugins and tools, and gain access to the 24 GHz radar software package and documentation, you must first download and install the Infineon Toolbox using this link:

<https://www.infineon.com/toolbox>

Note: Please refer to the “[Getting Started Infineon Toolbox Guide](#)” document for detailed installation instructions.

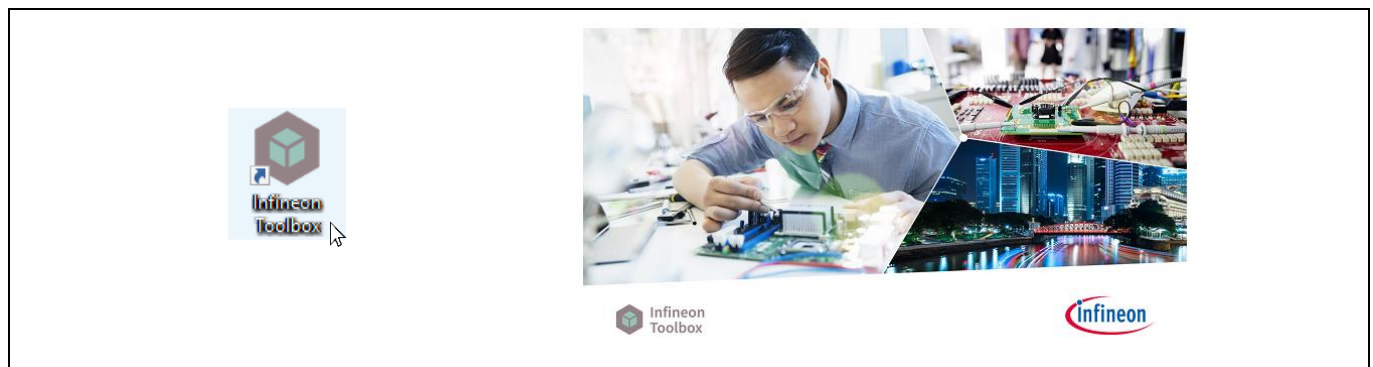


Figure 2 Run Infineon Toolbox

2.1.2 24 GHz radar tools

Once Infineon Toolbox is successfully installed, you can proceed with downloading the appropriate 24 GHz radar tools: Distance2Go Kit, Position2Go Kit, Sense2GoL Kit or Sense2GoL Pulse Kit including software package and documentation, as detailed in the following instructions:

- Run the Infineon Toolbox by double-clicking on the Infineon Toolbox icon on your Windows desktop. The program starts under the **My tools** tab, as shown in Figure 3.

Running radar applications

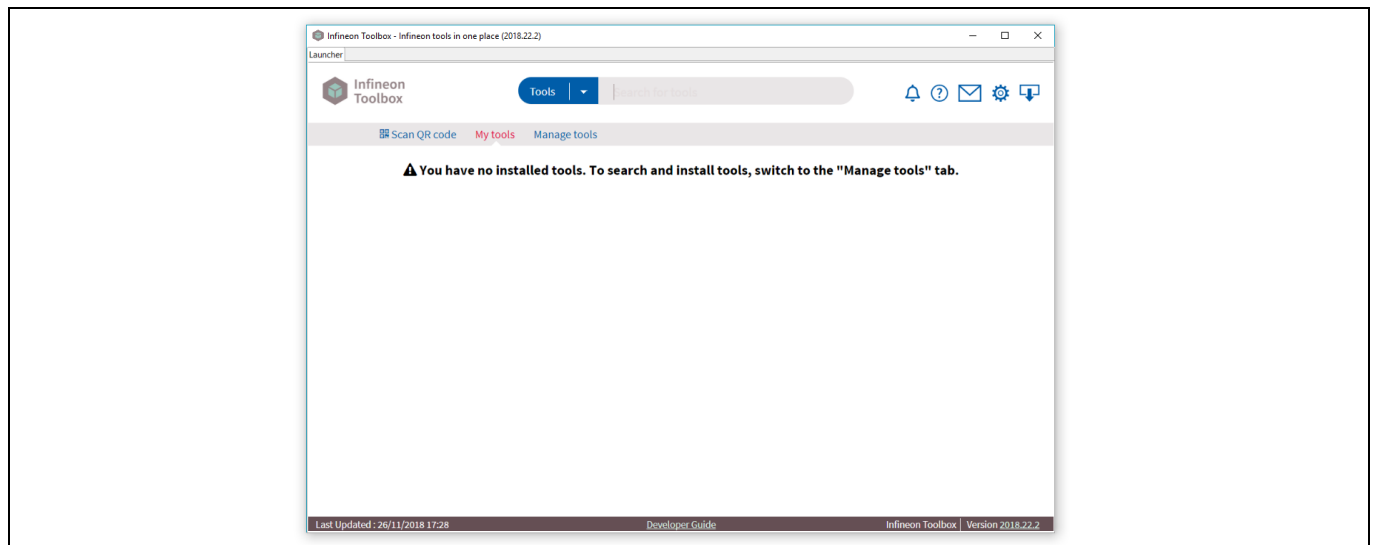


Figure 3 Infineon Toolbox tools launcher

- Click on the **Manage tools** tab, then type the 24 GHz kit name of your choice (e.g. Position2Go) in the search box and press **Enter**.

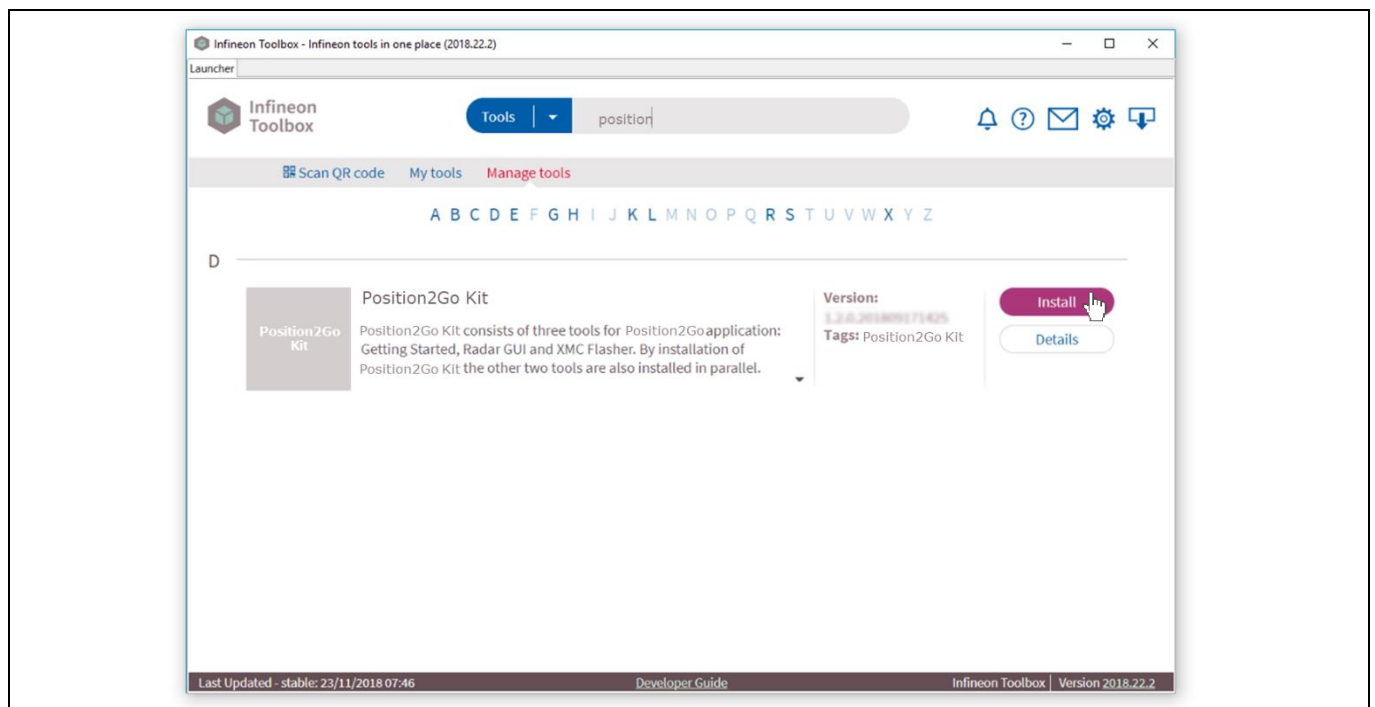


Figure 4 Infineon Toolbox tools installation

- Click on **Install**, then on **Accept** to accept the license.

Running radar applications

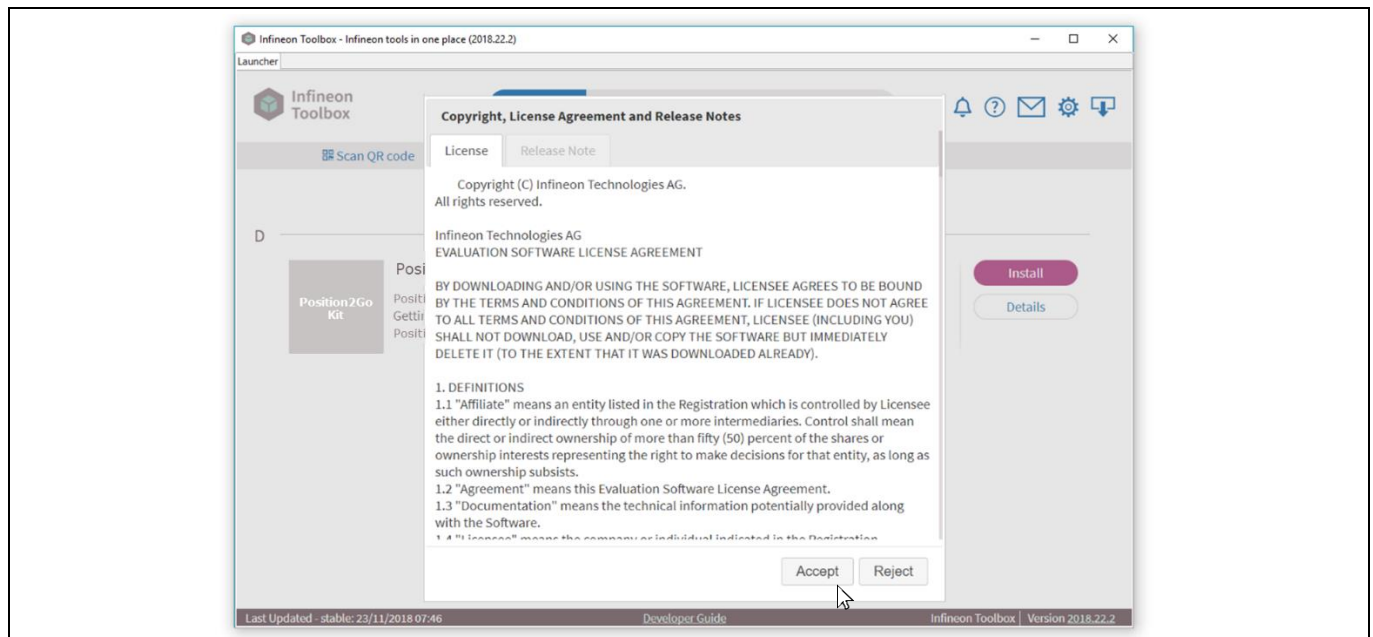


Figure 5 Radar Toolbox tools license agreement

- Click on **Yes** to restart the toolbox.
- Click on the installed kit icon to go to the start page.

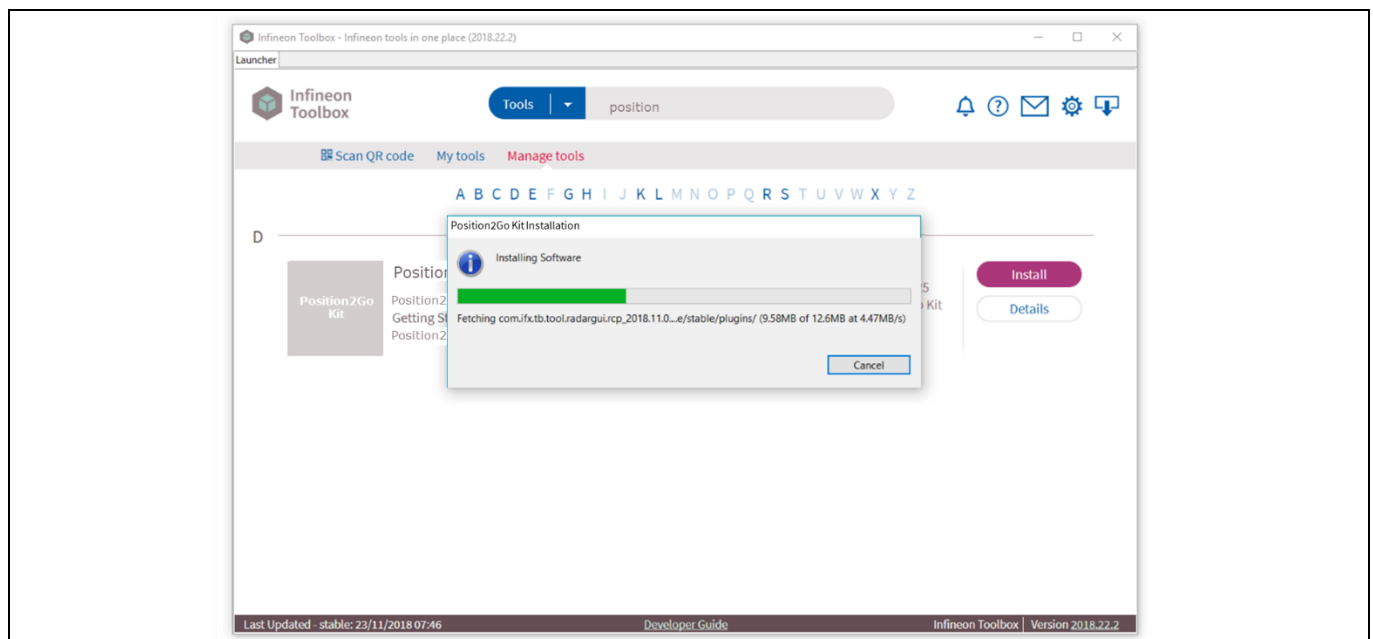


Figure 6 Kit installation progress

The 24 GHz radar kit (e.g. Position2Go Kit) consists of three tools for the Position2Go application: Position2Go Kit, Radar GUI and XMC™ Flasher, as shown in **Figure 7**. By installing the Position2Go Kit, two other tools are also installed in parallel:

- **Radar GUI** – Infineon’s radar board demonstration application with highly interactive GUI for data visualization, recording and playback
- **XMC™ Flasher** – a tiny, free-of-charge tool for XMC™ MCU Flash programming

Running radar applications

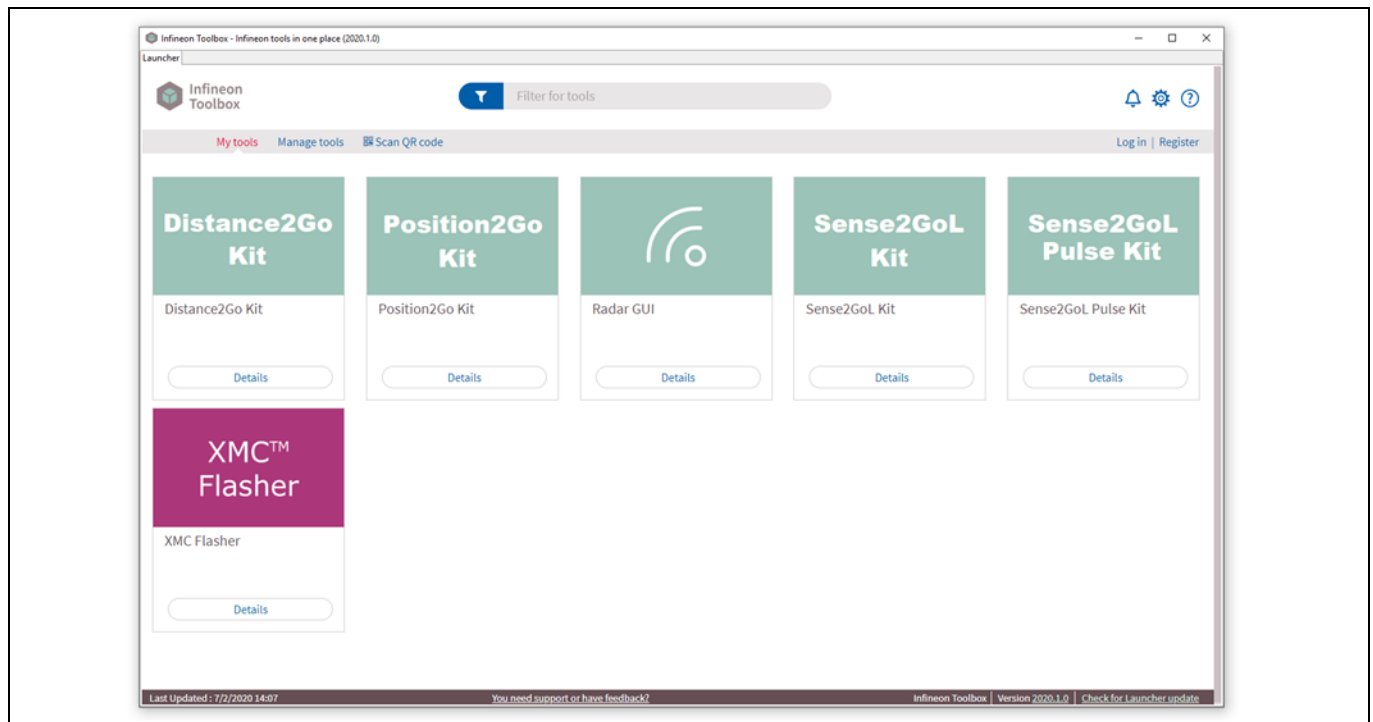


Figure 7 24 GHz radar kits – installed tools

User can download and install the 24 GHz radar SW package (e.g. P2G-HW-SW.exe for Position2Go) after starting the appropriate kit via the Infineon Toolbox (e.g. Position2Go Kit). Once installed, a folder (e.g. P2G-HW-SW) will be created on the specified path with the following structure:

- **Firmware_Software** – all software, firmware and drivers
- **Hardware** – all hardware-related files (e.g. schematics, Altium files)
- **Documentation** – all documentation (e.g. quick start guide, application notes)

2.1.3 XMC™ Flasher

The 24 GHz radar demonstration firmware is already pre-loaded in the Flash memory on the XMC™ microcontroller. This section describes how to use the binary images provided to reprogram the firmware applications. In addition to XMC™ Flasher, Radar GUI can also be used to flash new firmware into the device which is explained in section 2.2.1.2. The 24 GHz radar firmware package contains binary images (*.hex) of the applications provided in the subfolder **Binary**.

The XMC™ Flasher tool can be used for on-chip Flash programming to reprogram the radar application using a binary image, as follows:

- Connect the 24 GHz radar board to a PC with USB “type A to micro-B” cables through the embedded USB connector to power up the board or to debug.
Please refer to your 24 GHz radar board Quick Start Guide document to connect it to the PC correctly.
- Use (*.hex) binary with the XMC™ Flasher tool to reprogram the radar firmware:
 - Start the **XMC™ Flasher** tool in the Infineon Toolbox launcher.

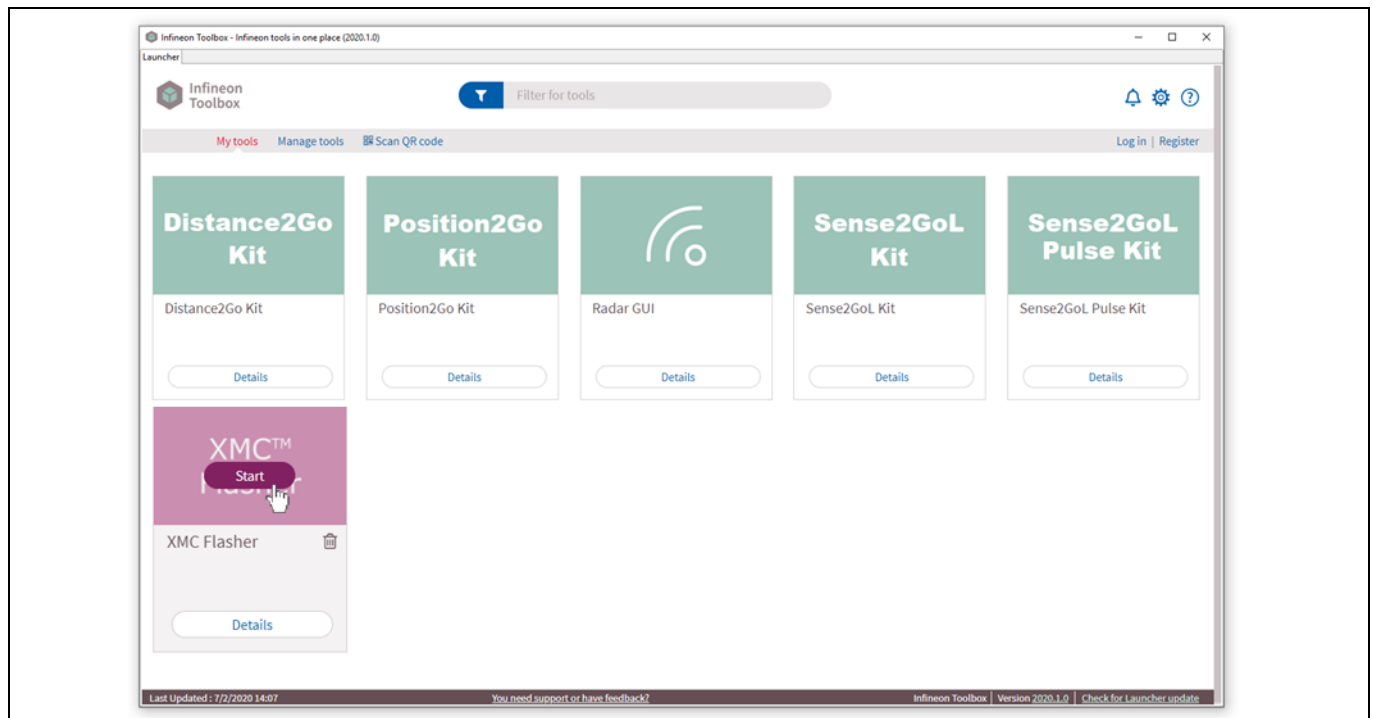


Figure 8 Start XMC™ Flasher tool via the Infineon Toolbox

- Once started, click on the **Connect** button, then select the device name (refer to Table 1 from the **List of Targets** window and confirm with the **OK** button.

Table 1 XMC™ devices embedded in 24 GHz radar boards

24 GHz radar board	XMC™ device target
Distance2Go	XMC4200-256
Position2Go	XMC4700-2048
Sense2GoL	XMC1302-0016
Sense2GoL Pulse	XMC4700-2048

Running radar applications

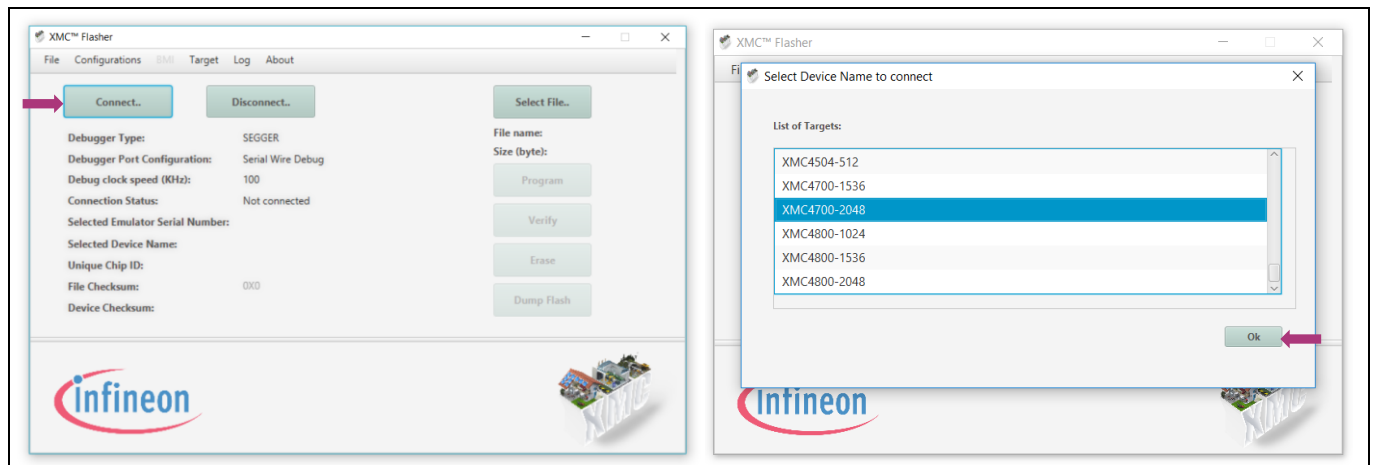


Figure 9 XMC™ Flasher device selection and connection

Note: Please ensure that SEGGER J-Link drivers are installed before using the XMC™ Flasher tool. Otherwise, the default debugger type under XMC™ Flasher Target Interface Setup will be set to **DAP**, as shown in Figure 10 (1). Once installed, the user must change the debugger type to **SEGGER**, as shown in Figure 10 (2).

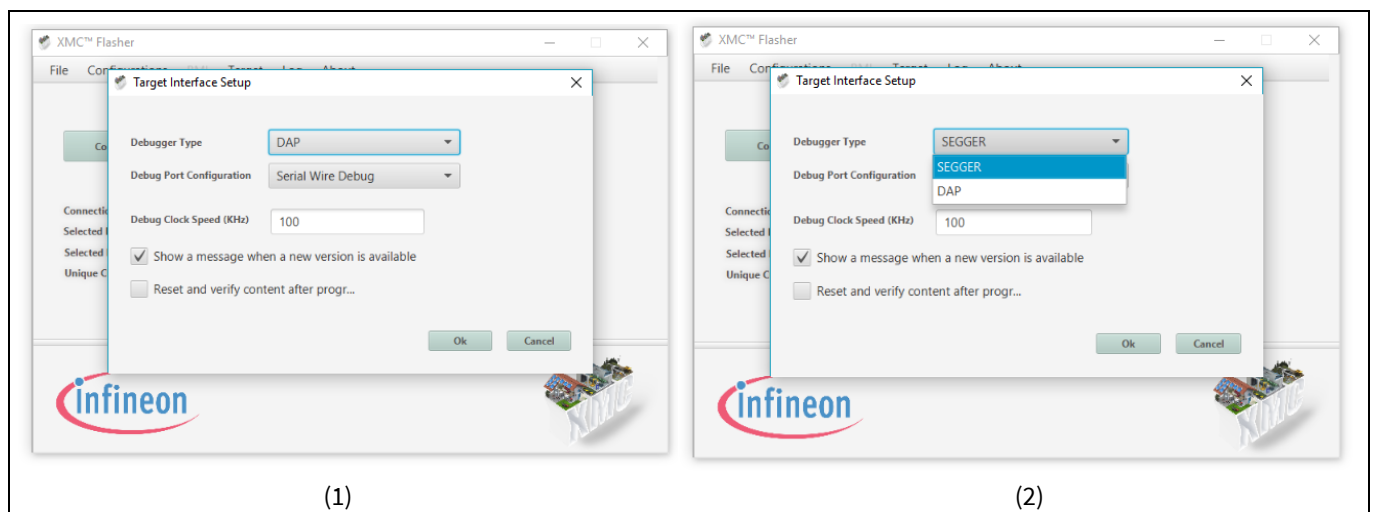


Figure 10 Change debugger type

- If the connection is established successfully, **Connection Status** turns to **Connected**. The **Unique Chip ID** is displayed as well.
- After connection is established, select the (*.hex) file by clicking on the **Select File...** button.

Running radar applications

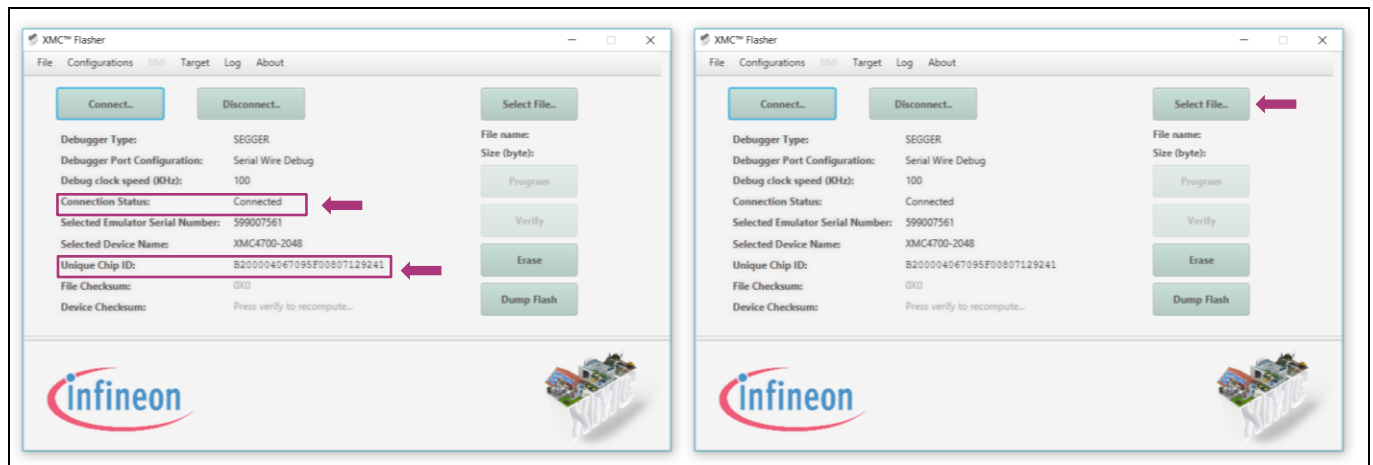


Figure 11 Binary image file selection

- Navigate to the **Binary** folder and select the (*.hex) file inside it (e.g. P2G_FW.hex), then click on **Open** in the dialog box.
- Successful selection of the (*.hex) file results in listing its filename below the **Select File...** button.

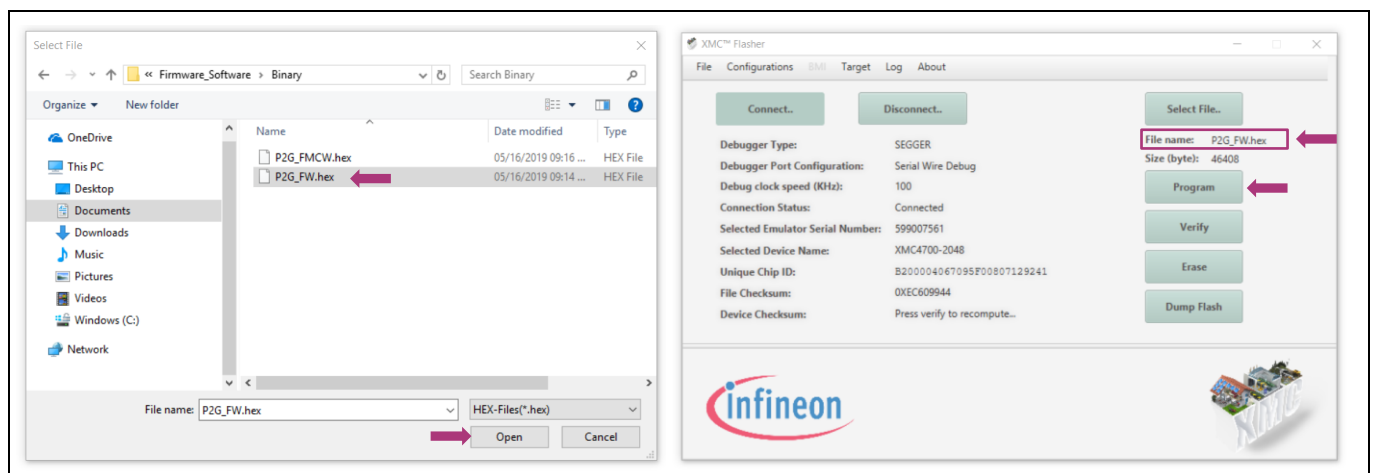


Figure 12 Binary image programming

- Click on the **Program** button, which opens the SEGGER progress window. It either verifies successful Flashing or shows an error message.
- If programming succeeds, the message **Programming is successful!** appears.

Running radar applications

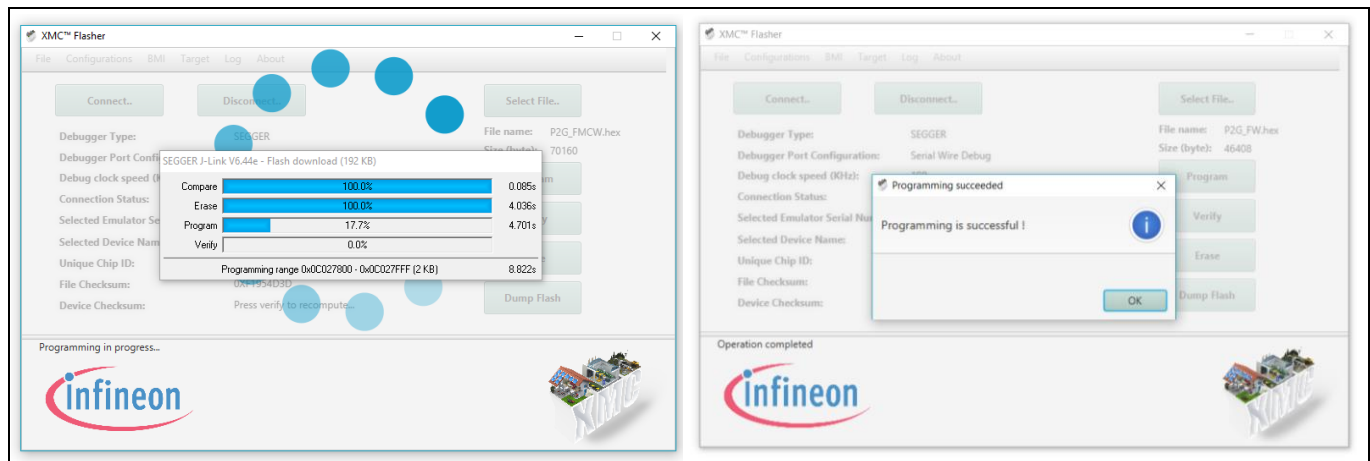


Figure 13 Programming successful

Note: The XMC™ Flasher requires a J-Link compatible debug-HW to connect to the target, which is already integrated in the 24 GHz Radar demo kit.

2.1.4 DAVE™ toolchain

DAVE™ (Digital Application Virtual Engineer), is a free-of-charge Eclipse-based Integrated Development Environment (IDE) using a GNU C-compiler that provides an extensive, configurable and reusable code repository for an XMC™ industrial microcontroller powered by ARM® Cortex®-M processors.

It is a C/C++-language software development and code generation tool for XMC™ microcontroller applications using DAVE™ APPs to configure the MCU peripherals (ADC, DMA, CCU4...), which reduces development time and allows for quick porting of the firmware across XMC™-series MCUs.

DAVE v4.1.2 or higher should be installed, or any other third-party toolchain supporting Infineon Technologies XMC™ microcontrollers, e.g. Atollic, IAR, Keil MDK, Rowley or TASKING. The latest version of DAVE™ (v4.4.2) can be downloaded from https://infineoncommunity.com/dave-download_ID645.

2.2 Graphical User Interface (GUI) solutions

2.2.1 Radar GUI

Note: The Radar GUI tool is only supported with the **Distance2Go**, **Position2Go** and **Sense2GoL Pulse** radar software packages.

2.2.1.1 Overview

Radar GUI is a Java-based highly interactive GUI for Windows XP/Vista/7/8/10. It provides graphical support for Infineon's radar devices and enables the visualization of real-time raw IF quadrature output signals and FFT spectrum, and enables observation of the targets' distance and velocity information from a connected 24 GHz radar device.

Radar GUI offers several methods to record data in different formats for advanced signal processing and supports multiple recording options. Each option saves corresponding data in a dedicated file, as follows:

- **Raw data** – records raw IF data as they are received from the device; data are stored in a file with .raw extension

Running radar applications

- **Time domain data** – records extracted time domain data (I/Q signals); data are stored in a file with .tdd extension
- **Frequency domain data** – records processed spectrum data; data are stored in a file with .fdd extension
- **Target data** – records radar target list data; data are stored in a file with .tgd extension

2.2.1.2 How to use

To run the Radar GUI as a standalone application on Windows, proceed as follows:

- Run **Infineon Toolbox** by double-clicking on the Infineon Toolbox icon on your Windows desktop.
- Infineon automatically offers you the option to update **Radar GUI**. If a new version of the Radar GUI is available, a button labeled **Update** appears within the Radar GUI tab.
- Click on the **Update** button and afterward on the **Yes** button to confirm the update and get the new version of the Radar GUI tool.

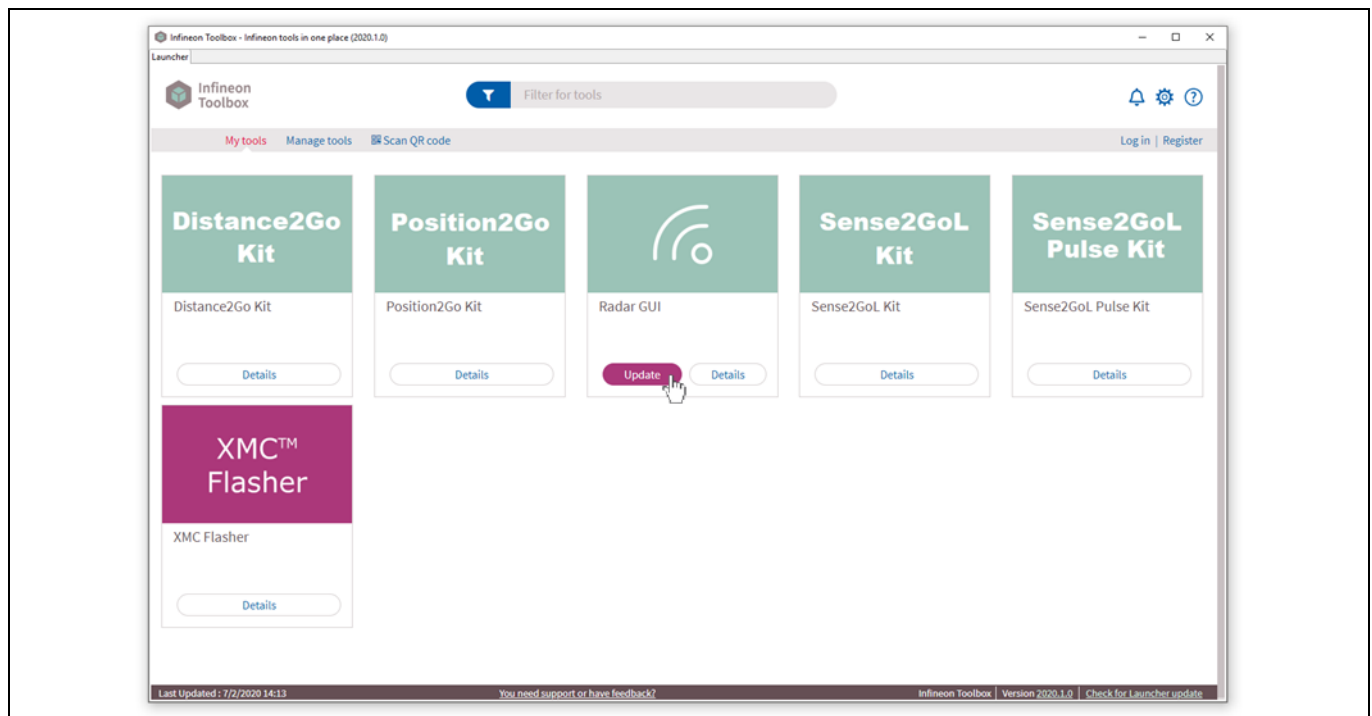


Figure 14 Infineon Toolbox launcher

- After an update, Infineon Toolbox must be restarted for the changes to take effect.
- Click on the **Start** button of the Radar GUI application available under the **My Tools** tab.

Running radar applications

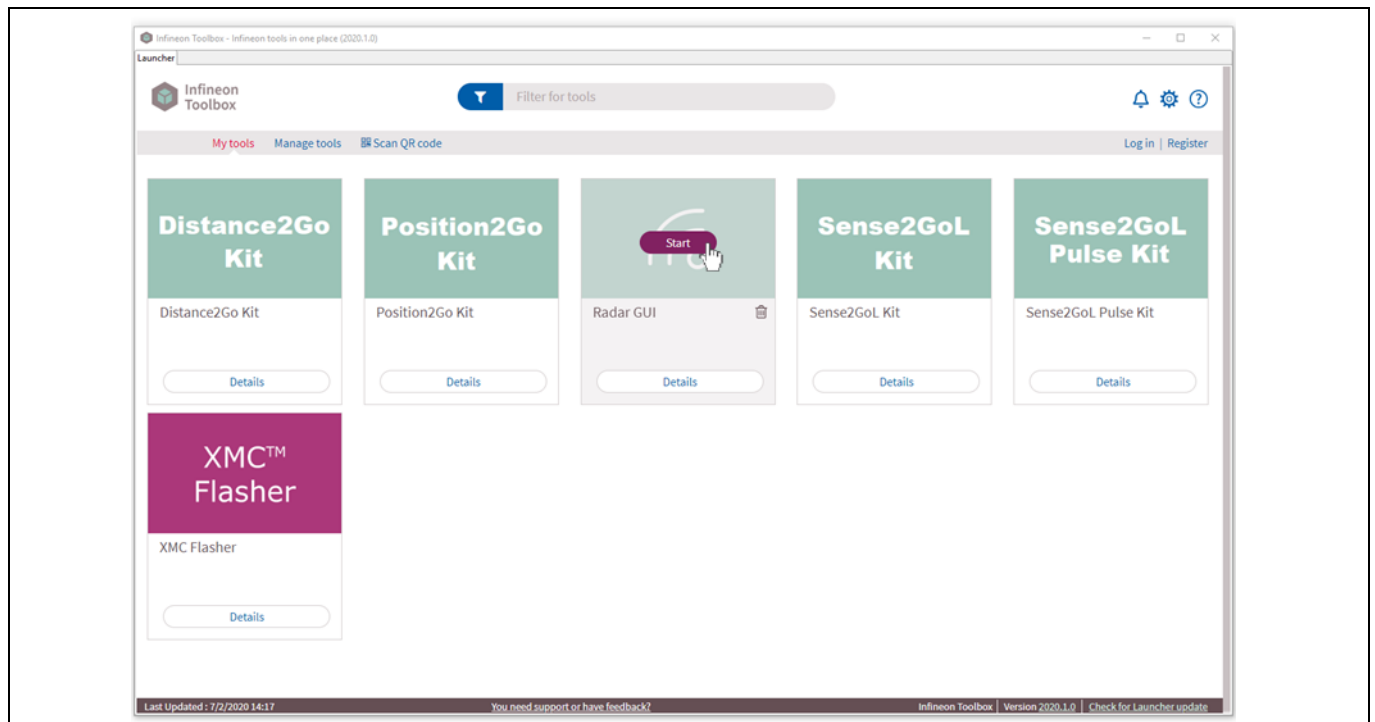


Figure 15 Start the Radar GUI application via Infineon Toolbox

- After launching the Radar GUI application the compatibility of the firmware version running in your 24 GHz radar-connected device is checked. If a newer firmware version is available, a prompt to follow the links to update the firmware appears.

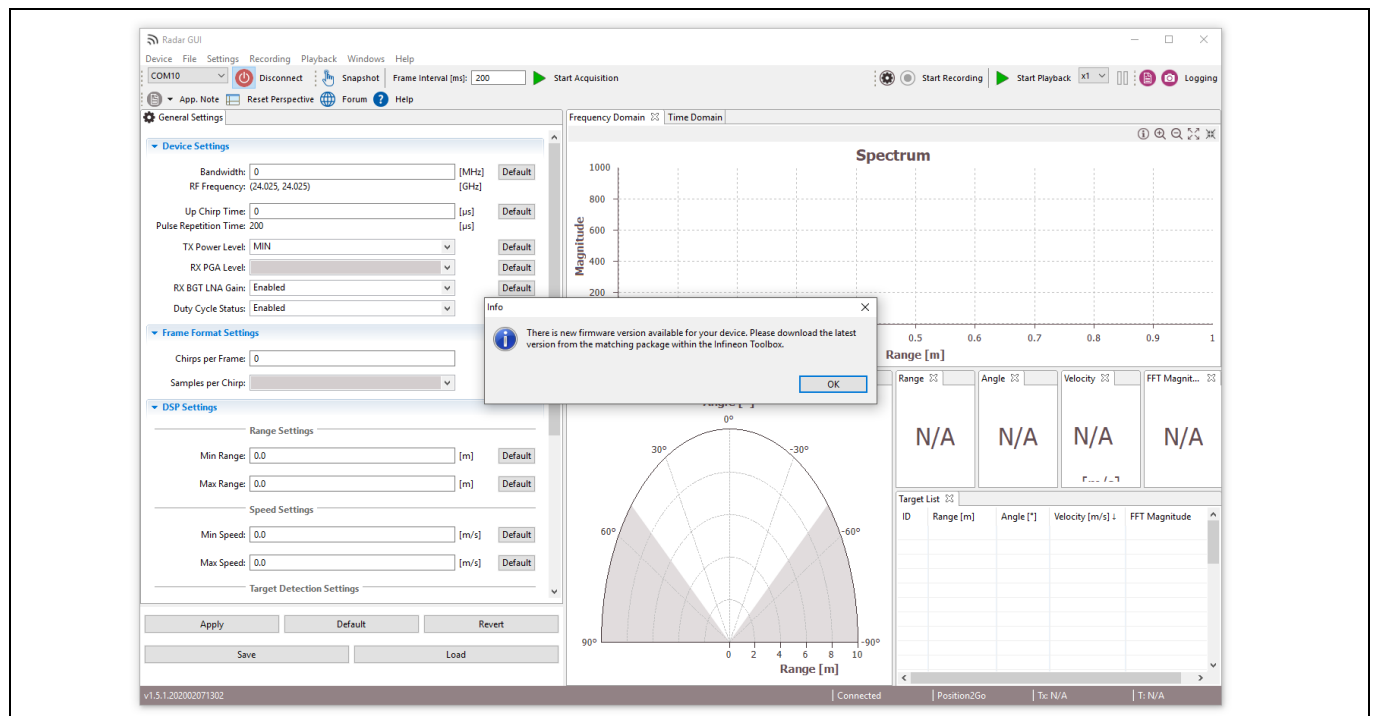


Figure 16 New firmware version available window

Running radar applications

- Radar GUI automatically starts acquiring data from the 24 GHz radar board, when the radar firmware is running. Figure 17 shows the Radar GUI default screen acquiring FMCW data from Position2Go radar device, in which the Frequency Domain view displays the spectrum data for each antenna.
- For further information, please refer to the integrated Radar GUI help, which provides a complete overview of all features.



Figure 17 Radar GUI application window

- In addition to XMCT™ Flasher, Radar GUI offers firmware flashing capability as well. In order to flash a new (*.hex) file (e.g. P2G_FW.hex), simply go to **Device** tab in **Main Menu** and select **Flash Firmware** option, as shown in Figure 18. A dialog box will appear where you can give path for your (*.hex) file or select from a preloaded list of available (*.hex) files (**Select Firmware**), as shown in Figure 19. You can also select your device, if not already detected by Radar GUI, from the drop down list of **Select Device**. Pressing the **Flash Firmware** option in the dialog box will start flashing the new (*.hex) file into the device. Successfully Flashed message would be generated in the end and Radar GUI can start data acquisition again if the Connect button is pressed, as shown in Figure 20.

Running radar applications

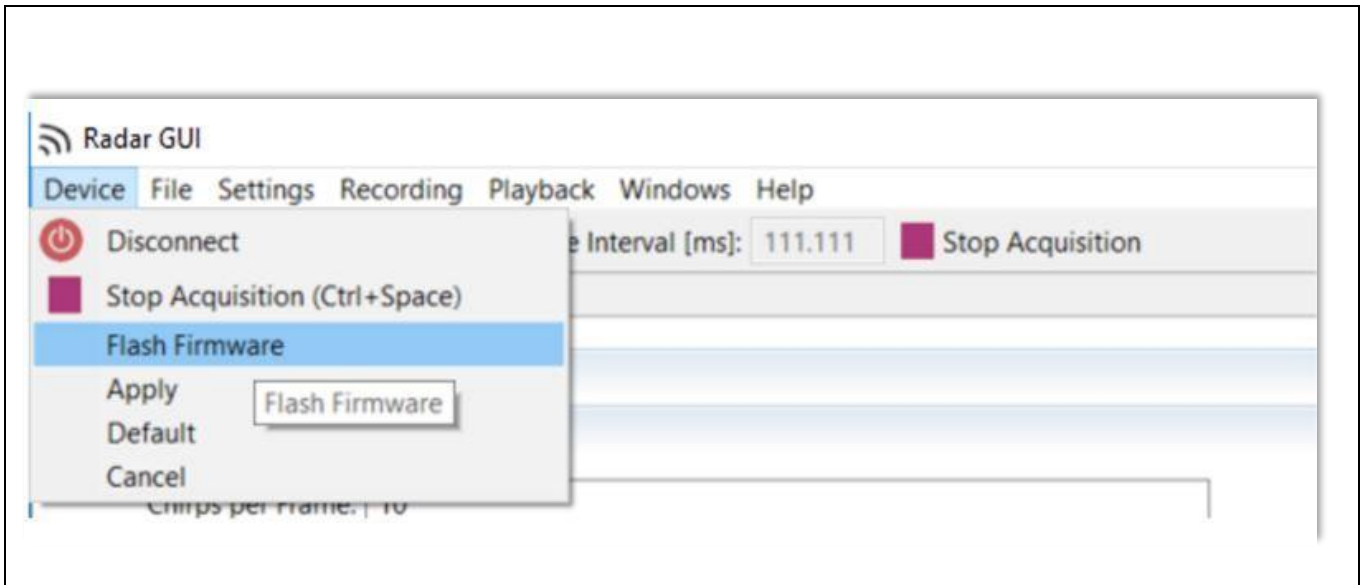


Figure 18 Firmware flashing via Radar GUI

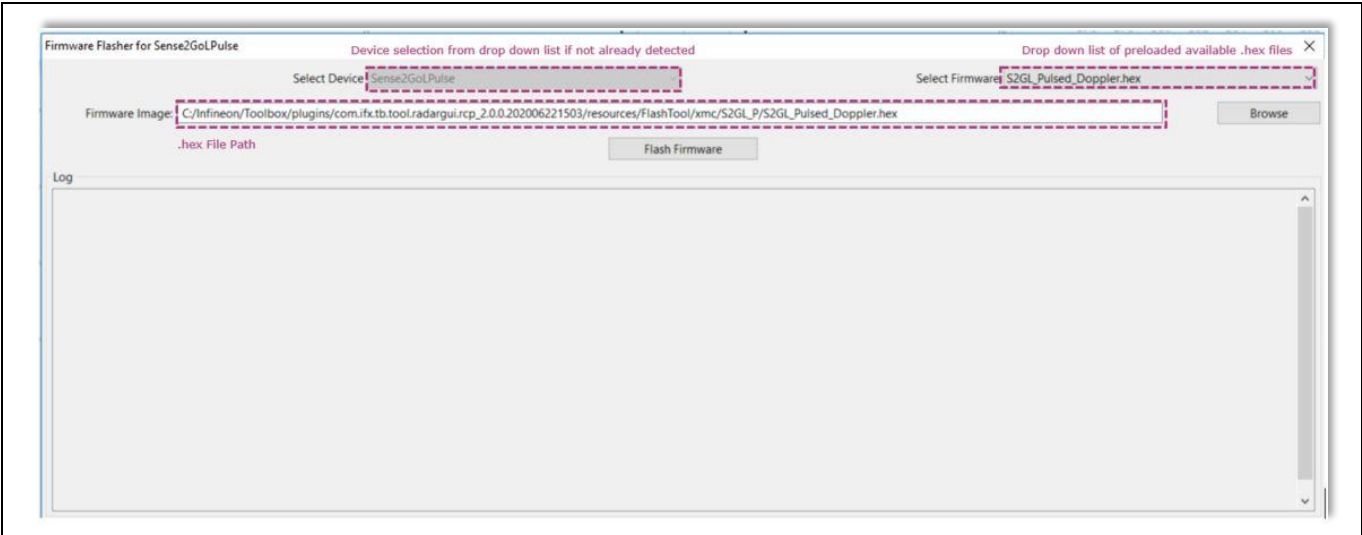


Figure 19 Firmware flashing via Radar GUI

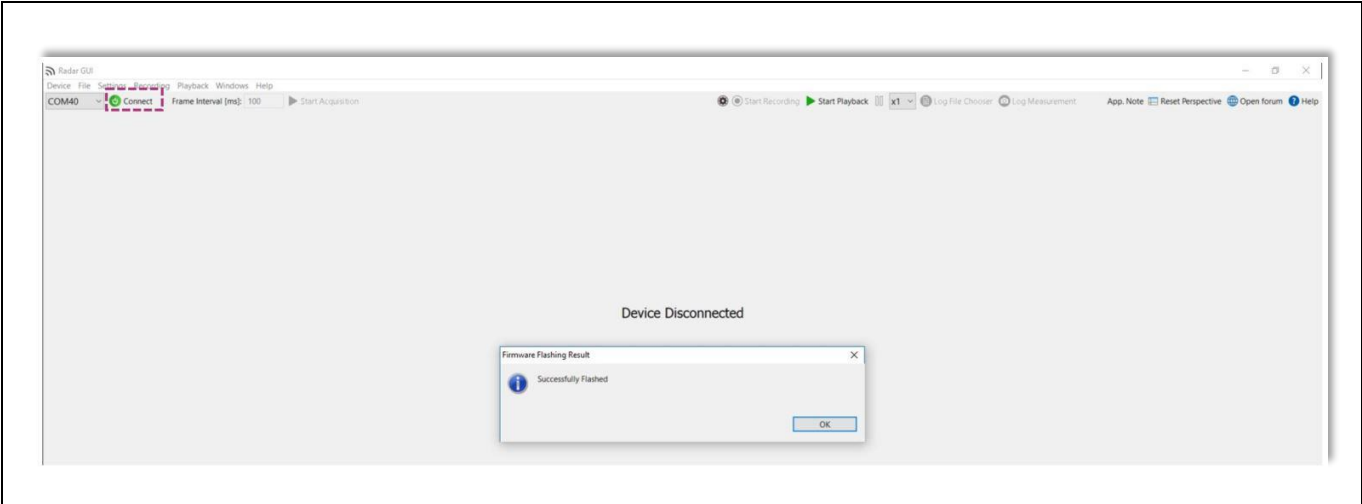


Figure 20 Firmware flashing via Radar GUI

2.2.2 Micrium µC/Probe™ XMC™ GUI

Note: The Micrium GUI tool is **only** supported by the **Sense2GoL** radar device, and is **not compatible with the Radar GUI**.

2.2.2.1 Overview

µC/Probe™ XMC™ from Micrium® is a free-of-charge data monitoring and visualization tool to modify and track real-time data on the XMC™ target microcontroller in a non-intrusive way.

It enables designing a graphical dashboard with a wide range of widgets to control or fine-tune your XMC™ application, and it includes an eight-channel digital oscilloscope to visualize real-time data, controlled by a dedicated code that runs on the XMC™ target.

µC/Probe™ XMC™ is simple to install on a Windows PC and can be easily connected via the J-Link onboard debugger integrated into most of the XMC™ kits.

The latest version of µC/Probe™ XMC™ v4.3.0.9 is available for download from:

https://infineoncommunity.com/uC-Probe-XMC-software-download_ID712

2.2.2.2 How to use

The Sense2GoL software package (S2GL-HW-SW.exe) comes with a GUI based on a μ C/Probe™ project from Micrium, which helps the user process collected raw data. In order to run this Micrium-based GUI project, proceed as follows:

- Go to the **/Firmware_Software/GUI** folder inside the locally installed S2GL-HW-SW package.
- Double-click the μ C/Probe™ **S2GL_GUI.wspcx** project, to open the GUI.
- **S2GL_GUI.wspcx** can be opened in Windows Explorer, or in S2GL_Doppler DAVE™ project explorer, as shown in Figure 21.

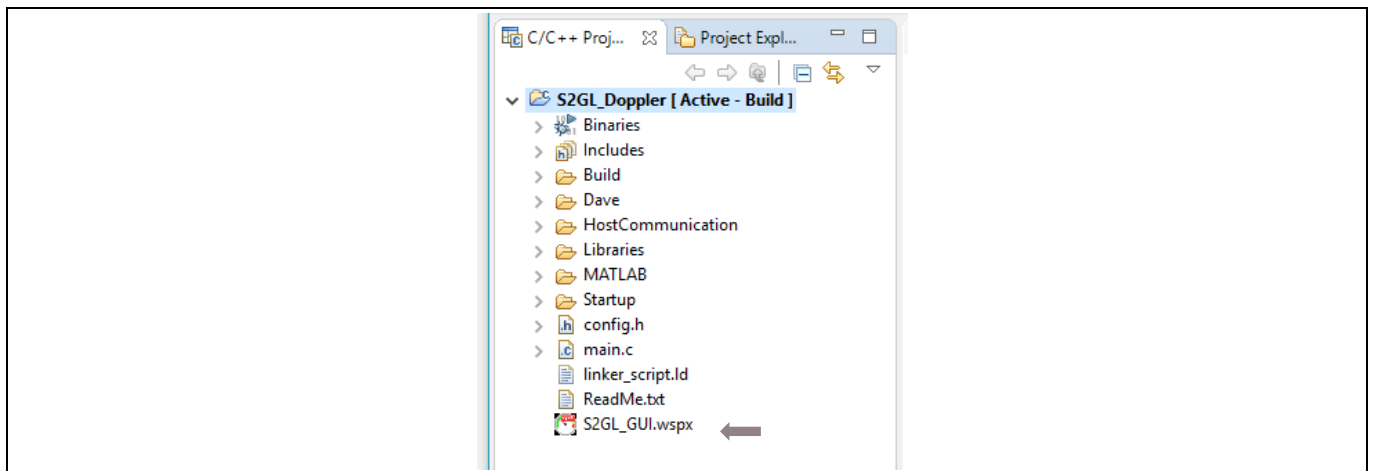


Figure 21 Open Sense2GoL GUI

The μ C/Probe™ needs to be provided by the XMC™ compiling and linking process output file (ELF file). This file containing the name, data type and address of all firmware global variables is parsed by the μ C/Probe™ project.

A precompiled .elf file is already available in a **/Firmware_Software/GUI** folder called **S2GL_Doppler.elf**.

*Note: After building a project, object files and an application binary file (typically in ELF format) exist in the Debug folder in the Project Explorer view file tree. Please ensure you have imported the **S2GL_Doppler.elf** file into your Micrium project each time you modify and build your S2GL_Doppler project.*

Once the GUI project has opened, the following steps need to be executed:

- Connect your Sense2GoL kit via USB (please ensure the USB port on the debugger board is connected).
- Start the GUI by clicking on the **Run** button.

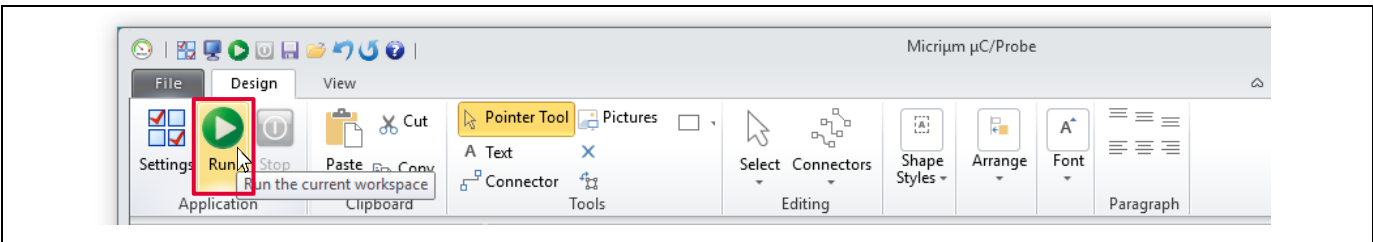


Figure 22 Run the Sense2GoL GUI

- The GUI interface in Figure 23 should appear. The size of the GUI is not adaptive and may require some adjustments to fit the screen.

Running radar applications

- Press the **START** button to begin data collection.

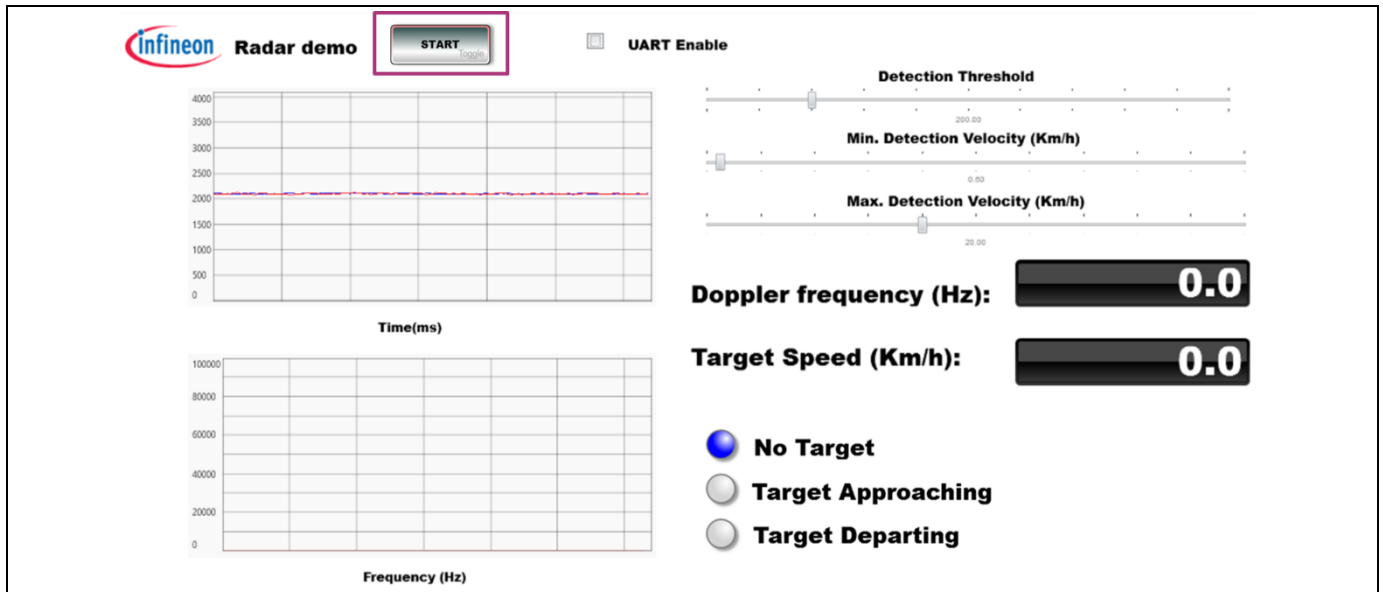


Figure 23 Start data collection

The radar should now detect movement and display data on the GUI:

- Time and frequency plots
- Threshold can also be set, default 200
- Minimum velocity (default 0.50 km/h) and maximum velocity (default 20 km/h)
- Maximum Doppler frequency and target speed
- Direction of movement

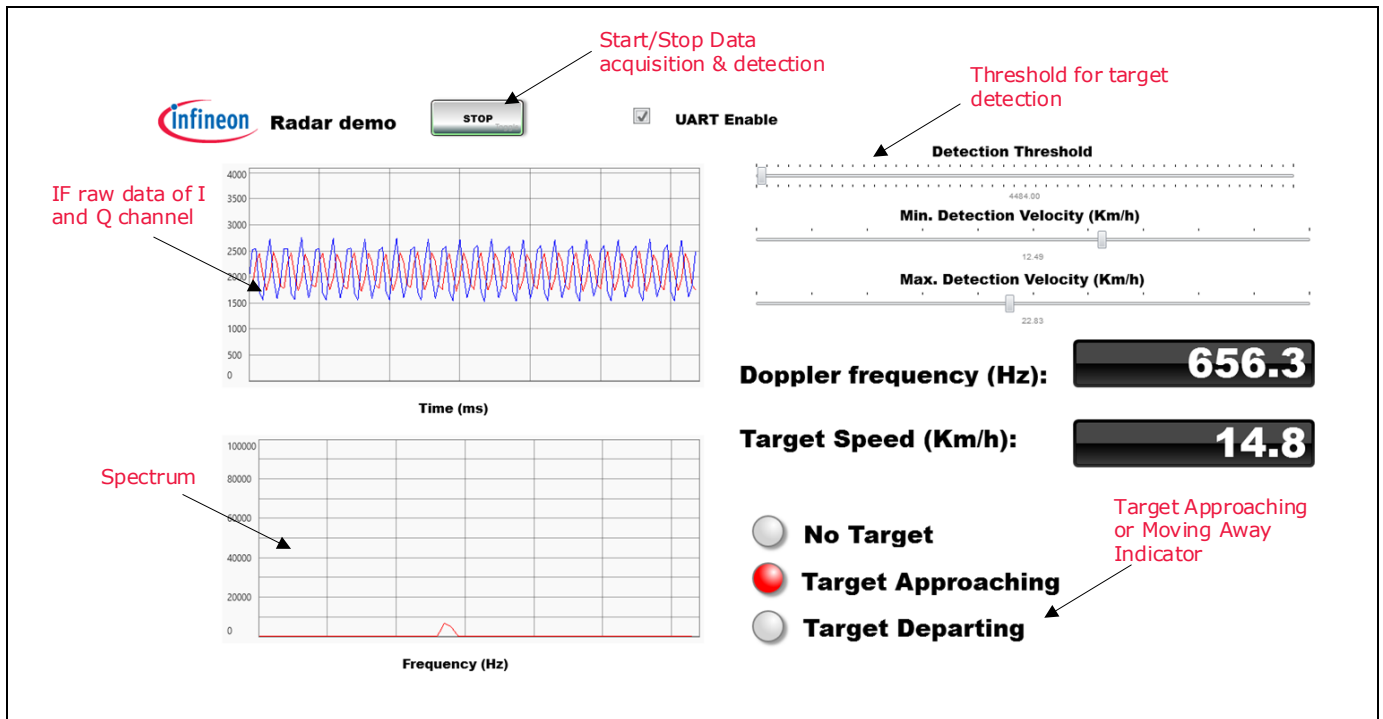


Figure 24 Data display on Micrium GUI

3 Customizing radar applications

Firmware (FW) is a piece of software written in C language to control different ICs and peripherals via the host processor, which is the XMC™ Cortex™-M MCU embedded in Infineon's 24 GHz radar demo kits.

The 24 GHz radar firmware is released as a ready-to-run DAVE™4 project, where source files are generated based on the DAVE™ APPs used, which are graphical-configurable application-oriented software components, used to enable quick reuse and customization.

This section explains how to customize, build, Flash and debug radar applications built on generated DAVE™ code, and run them on the 24 GHz radar demo kit.

3.1 Building, Flashing and debugging the DAVE™ project

After installing the DAVE™ IDE, the user can start the program to build and load the 24 GHz radar firmware applications as follows:

- Download and unzip the 24 GHz radar firmware package, making sure not to modify the package folder structure.
- Open the project workspace with the DAVE™ toolchain:
 - Run the DAVE™ toolchain and wait for the program to start, which asks for the workspace location.
 - Enter a path or browse to the desired folder via the **Browse...** button and confirm the choice with the **OK** button.

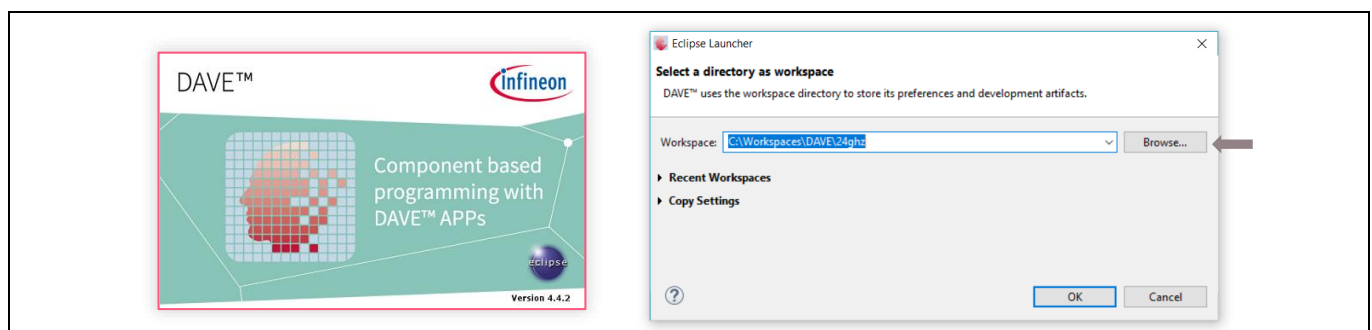


Figure 25 DAVE™ workspace launcher

Note: It is recommended that the active workspace folder is located not too many levels below the file system root to avoid build errors caused by exceeding the Windows path length character limitations.

- Once the DAVE™ workspace is opened, select **Import...** in the **File** menu to display the Import dialog box.
- Within the **Import** dialog box, select **Infineon, DAVE Project** and click on the **Next** button.

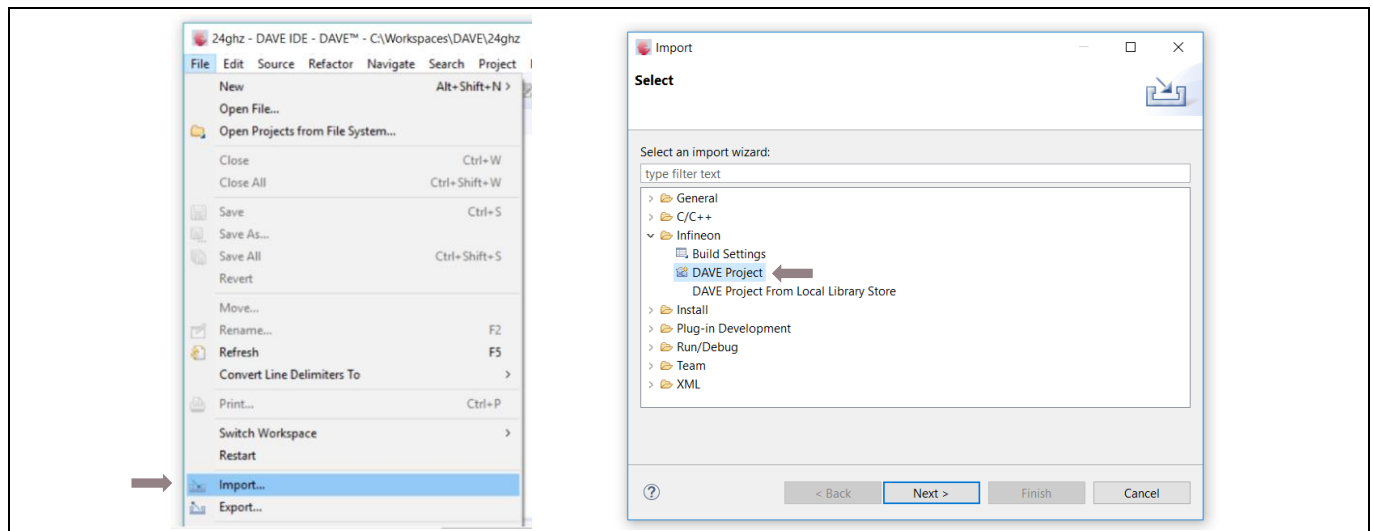


Figure 26 DAVE™ project import dialog box

- Check the **Copy Projects Into Workspace** checkbox.
- Select **Browse...** beside Select Root Directory.

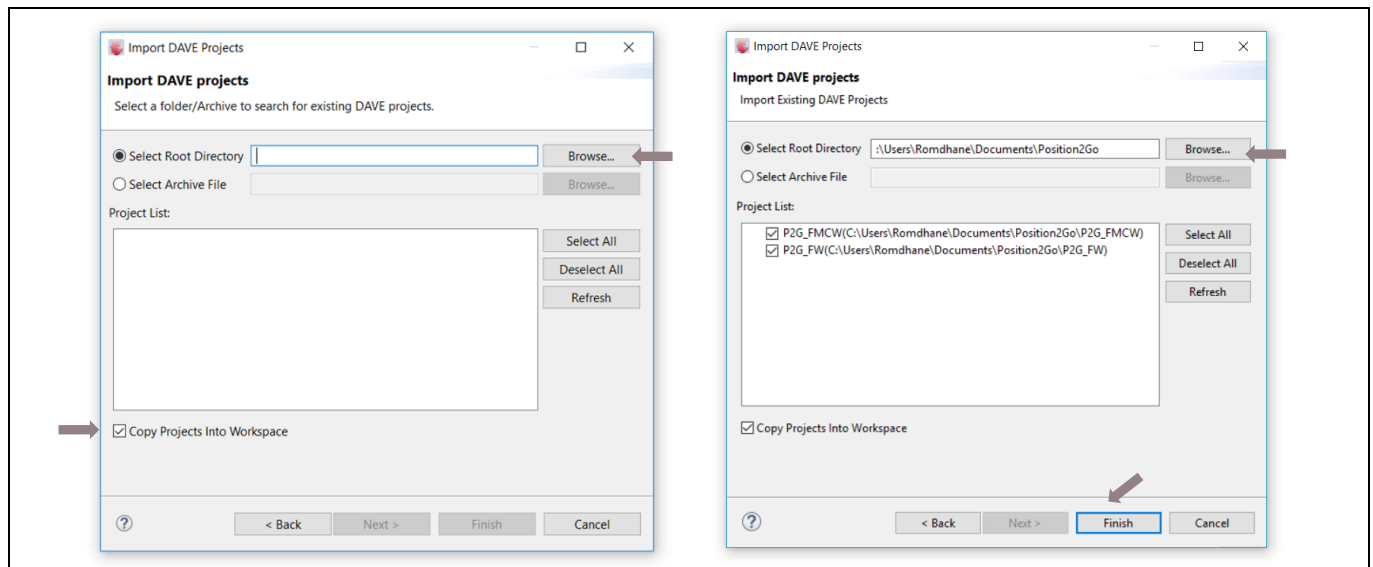


Figure 27 Import DAVE™ projects

- The project file should appear under the **Project List**. Click on the **Finish** button. A screen similar to Figure 28 should appear.

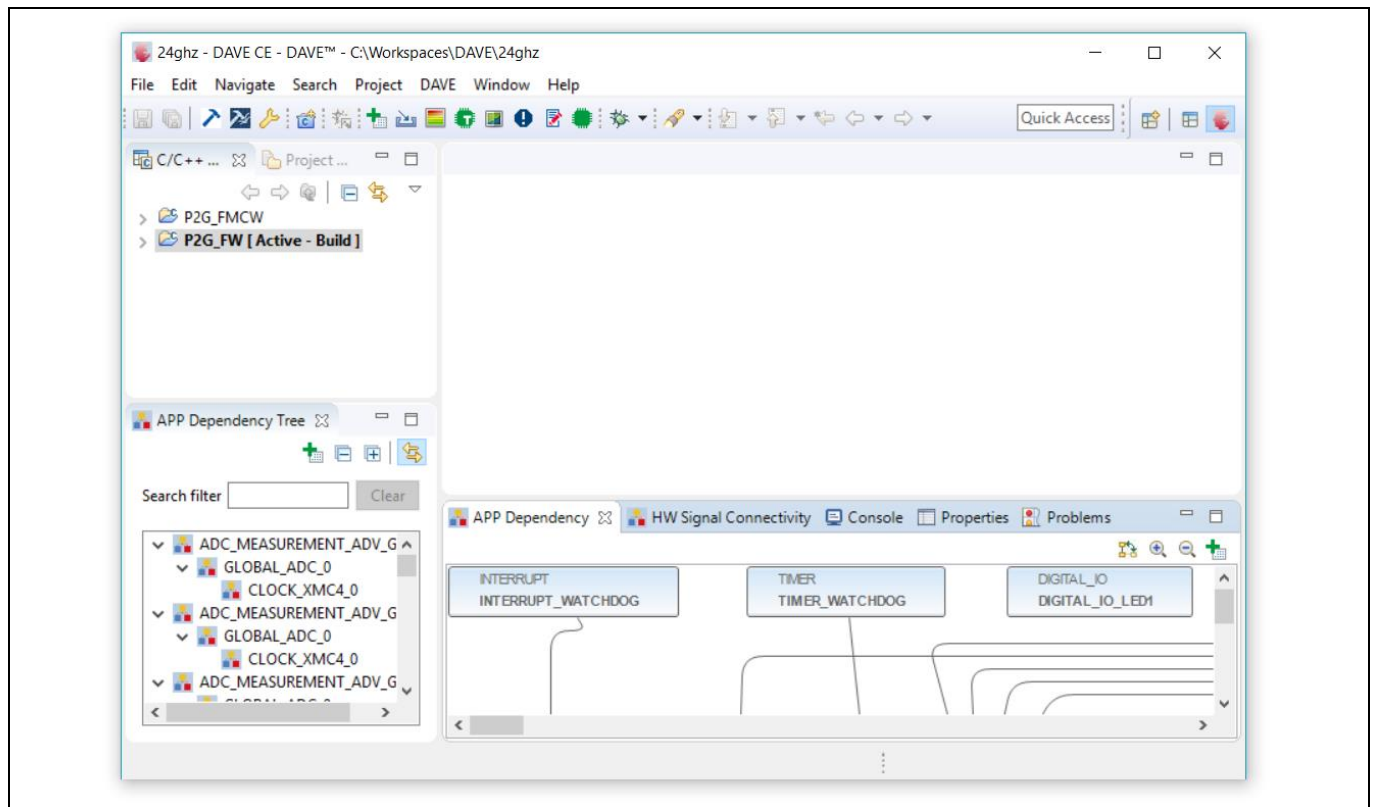


Figure 28 DAVE™ editor view

- There are two ways to build a project:
 - Click on **Build Active Project** (the icon with the blue hammer on a white background) in the toolbar.
 - Right-click on the active project **P2G_FW [Active - Build]** and then click on **Build Project**.

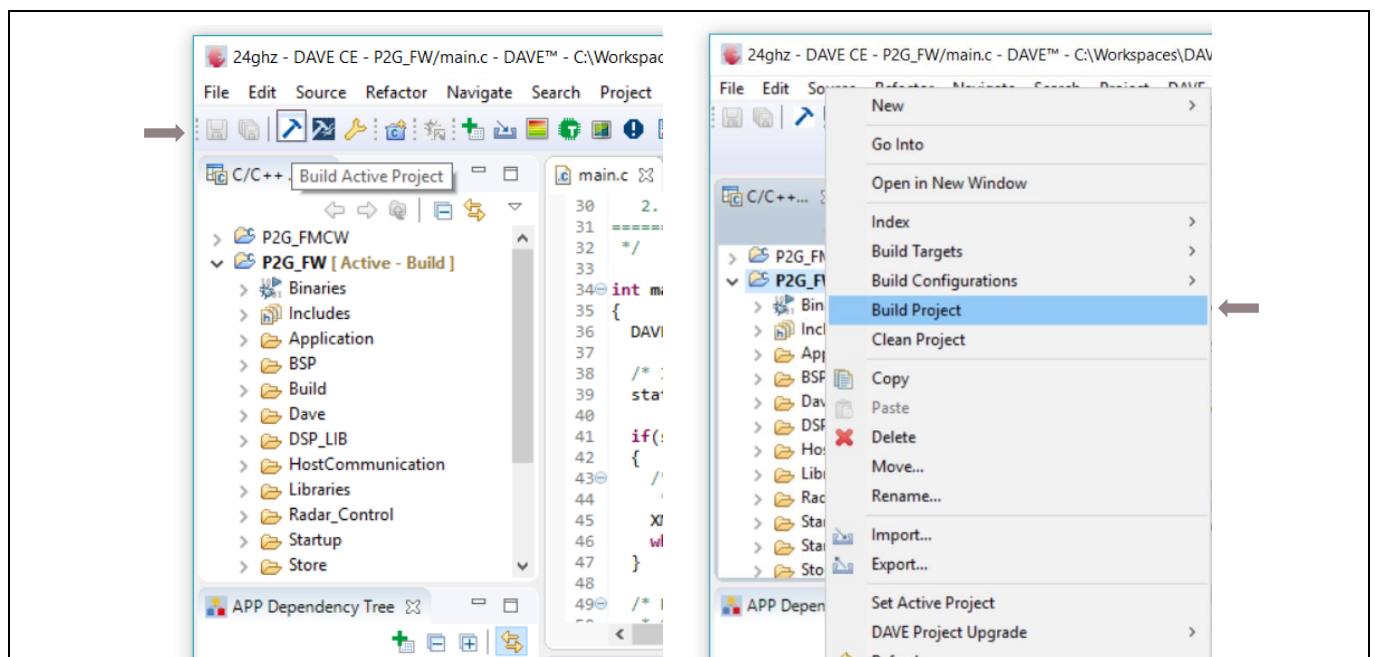


Figure 29 Build project options

- Check successful build in the **Console** window, where the build result details are displayed.

Customizing radar applications

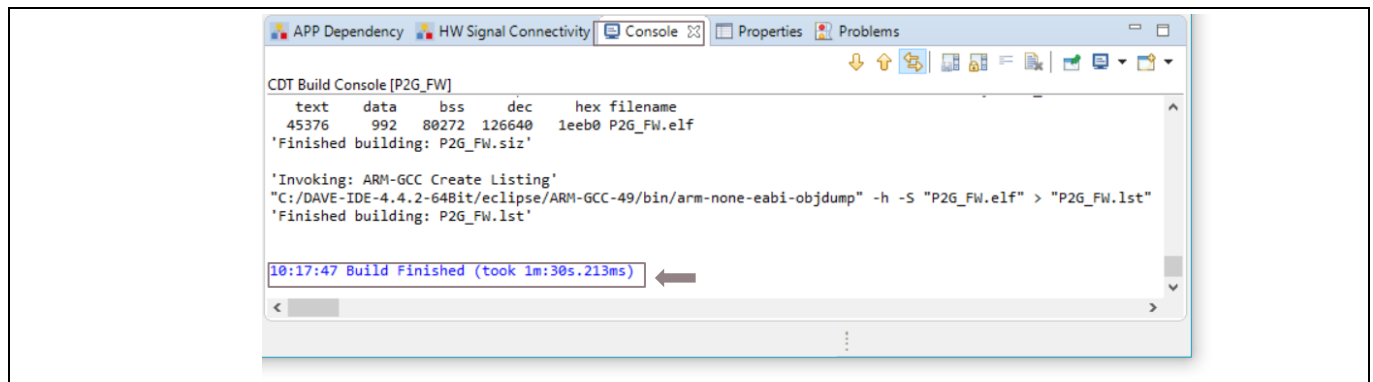


Figure 30 DAVE™ console

- Ensure that the debugger of the 24 GHz radar board is connected to the PC via USB “type A to micro-B” cable through the USB connector (e.g. X12 connector for Position2Go board).
- Set debug configurations by clicking on the drop-down menu of the **Debug** toolbar button (bug icon) and selecting **Debug Configurations...** as shown in Figure 31. Alternatively, press key F11 to directly start the debug session.

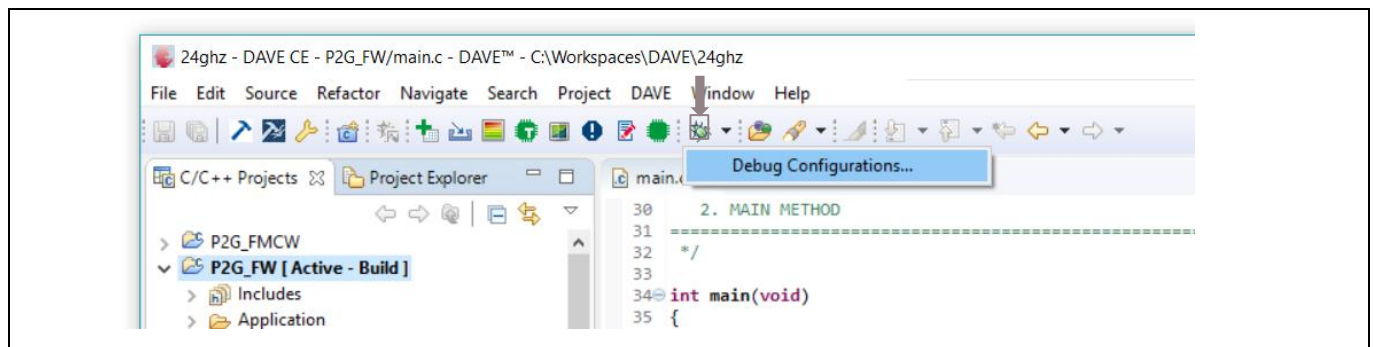


Figure 31 Start debug session toolbar button

If opening Debug Configurations, a window as shown in **Figure 32** appears.

- Double-click on **GDB SEGGER J-Link Debugging** to create a new debug configuration.
- Click on the **Debug** button to Flash and start a debug session.

Customizing radar applications

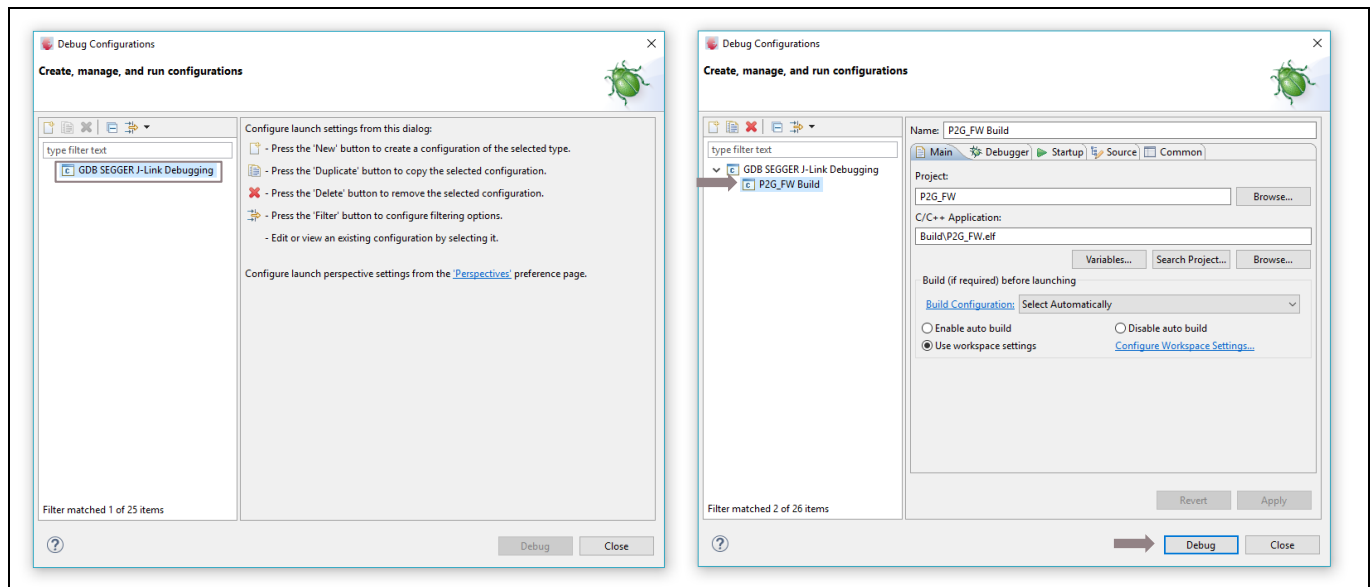


Figure 32 Debug configuration

- In this way, the Flashing process is started, the firmware image is loaded into the XMC™ microcontroller Flash memory and DAVE™ automatically switches to the debug perspective, as shown in Figure 33.

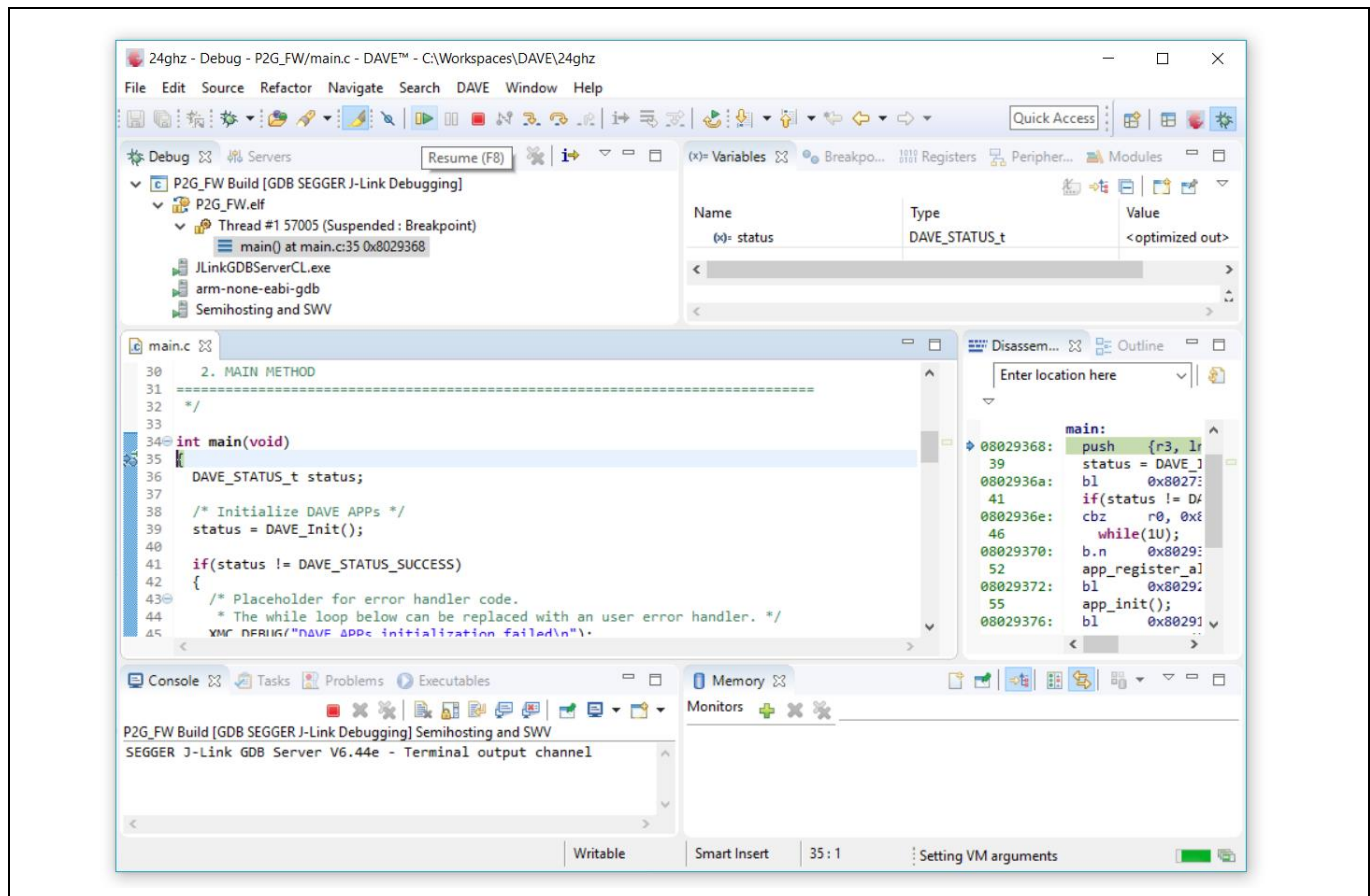


Figure 33 DAVE™ debug perspective

3.2 Firmware customization and configuration

The 24 GHz radar firmware is developed with Infineon's DAVE™ toolchain. It is a C/C++-language software development and code generation tool for XMC™ microcontroller applications using DAVE™ APPs to configure the MCU peripherals (ADC, DMA, CCU4...), which reduces development time and allows for quick porting of the firmware across XMC™-series MCUs.

The 24 GHz firmware projects can be customized by setting parameters in the configuration file **config.h**, which allows for customizing the drivers and algorithms for the user's radar application.

Table 2 lists some parameters that can be configured: enable, disable or modify some options by uncommenting and commenting or modifying the values of the related define statements.

Table 2 Define statements used for radar firmware configuration

Parameter	Description
DOPPLER_SAMPLING_FREQ_HZ	Sampling frequency for Doppler (units in Hz)
DOPPLER_FFT_SIZE	FFT length for Doppler mode, with zero padding
MINIMUM_RANGE_CM	Exclude targets below this distance (units in cm)
MAXIMUM_RANGE_CM	Exclude targets beyond this distance (units in cm)
RANGE_DETECTION_THRESHOLD	FFT spectrum threshold to detect a target in FMCW mode
MINIMUM_SPEED_KMH	Filter out targets below this speed (units in km/h)
MAXIMUM_SPEED_KMH	Filter out targets above this speed (units in km/h)
SPEED_DETECTION_THRESHOLD	FFT spectrum threshold to detect a target in Doppler mode

On each change of the config.h file parameters' values, the project has to be re-built and Flashed again. Otherwise that can be changed on the fly via the Radar GUI configurable fields.

4 Extracting radar raw data

The 24 GHz radar demo kits can use different interfaces, enabling the user to fetch raw data from the radar board as follows:

- MATLAB radar system interface
- C radar system interface
- UART radar system library interface

Supported radar communication libraries define a set of APIs to guarantee the data transfer through the host communication protocol.

4.1 Radar host communication protocol

The communication protocol is a generic protocol to exchange messages with microcontroller-based devices over a generic byte-stream connection (typically a virtual serial port via USB). The communication is always initiated by the host, while the connected microcontroller device responds to messages received from the host. The protocol defines two types of messages:

- Payload messages – contain a block of data with arbitrary size and meaning
- Status messages – contain a 16-bit status code

Messages from the host to the device are always payload messages. Whenever the device receives a payload message from the host, it responds with one (and only one) status message. The device may also send an arbitrary number of additional payload messages preceding that status message. Once the device has sent the status message, it will not send any more messages until it receives a new message from the host.

Each message from the host to the device is addressed to a certain logical endpoint, and each message from the device to the host is sent from a logical endpoint. The number of logical endpoints is defined by the device.

Endpoints are used to define functional groups in the device and are continuously enumerated starting with 1. At least one logical endpoint is always present. Each logical endpoint is of a defined type, which defines the meaning of the message payload exchanged with that endpoint.

Additionally, each endpoint has a version number that allows it to distinguish between slight modifications of the endpoints' set of known messages. The host can query the type and version of the endpoints present in the device.

All data received from the connected device is returned through callbacks. For each message type that can be received from the device, a separate callback function type is defined.

Within the 24 GHz radar firmware project, the host communication library contains the USB interface on host communication protocol drivers.

Extracting radar raw data

4.2 MATLAB radar system interface

4.2.1 Overview

The MATLAB application interface defines a set of APIs and functions that help the user access a serial communication port and communicate with the radar-defined endpoints, enabling the user to extract the raw IF data from the radar module via the USB interface to the PC/laptop for further signal processing.

Please refer to **/Firmware_Software/Communication Library/ComLib_Matlab_Interface/RadarSystemDocumentation** for more details about the interface's set of classes and functions that can be used to control the 24 GHz radar board via MATLAB.

4.2.2 How to use

A coding example to interface with the 24 GHz radar board via MATLAB and extract raw data is available. (Refer to the **extract_raw_data.m** file.) In order to run this coding example, proceed as follows:

- Go to **/Firmware_Software/Communication Library/ComLib_Matlab_Interface /RadarSystemExamples/GettingStarted** folder, and copy the path.
- Open MATLAB, then paste the path into the top tab, and the **"extract_raw_data.m"** file will show up on the left tab.
- Ensure that your 24 GHz radar board is connected, then Click on **Run** to see the raw data.

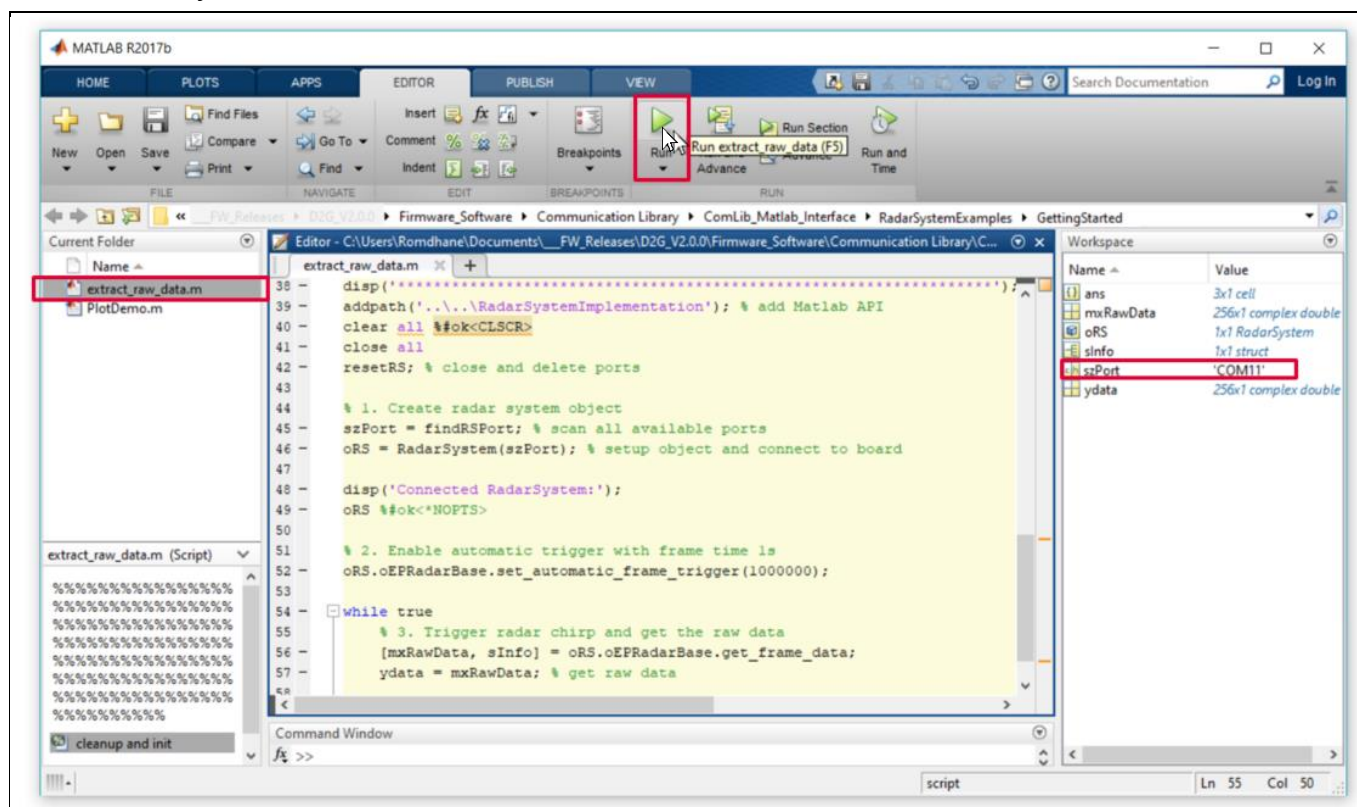
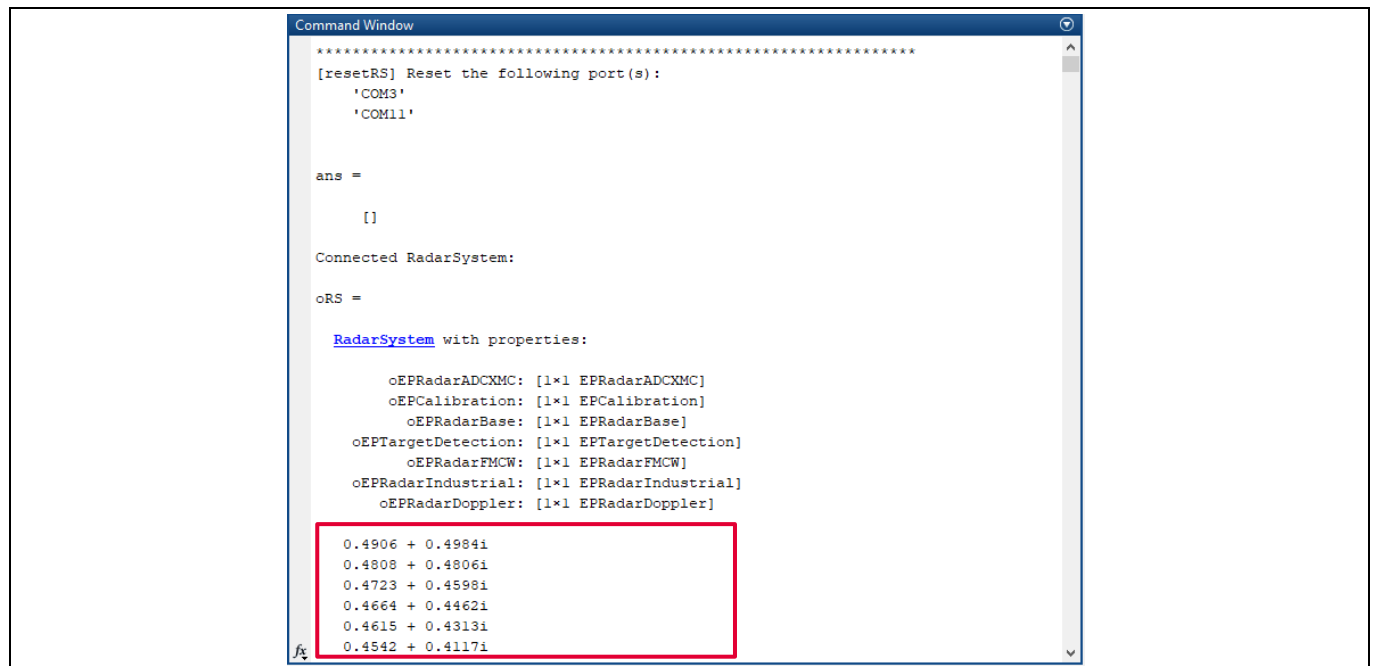


Figure 34 MATLAB radar system interface

Figure 32 shows the example running on MATLAB, extracting raw data from the D2G radar board.

Extracting radar raw data



```

Command Window
*****
[resetRS] Reset the following port(s):
    'COM3'
    'COM11'

ans =

    []

Connected RadarSystem:

oRS =

    RadarSystem with properties:

        oEPRadarADCXMC: [1x1 EPRadarADCXMC]
        oEPCalibration: [1x1 EPCalibration]
        oEPRadarBase: [1x1 EPRadarBase]
        oEPTargetDetection: [1x1 EPTargetDetection]
        oEPRadarFMCW: [1x1 EPRadarFMCW]
        oEPRadarIndustrial: [1x1 EPRadarIndustrial]
        oEPRadarDoppler: [1x1 EPRadarDoppler]

    0.4906 + 0.4984i
    0.4808 + 0.4806i
    0.4723 + 0.4598i
    0.4664 + 0.4462i
    0.4615 + 0.4313i
    0.4542 + 0.4117i
    
```

Figure 35 Raw data acquisition in MATLAB command window

Extracting radar raw data

4.3 C radar system interface

4.3.1 Overview

The C radar system interface defines a set of APIs that help user access a serial communication port and communicate with the radar-defined endpoints, enabling the user to demonstrate the capabilities of the 24 GHz modules.

Please refer to the **Firmware_Software/Communication Library/ComLib_C_Interface/documentation** for more details about the set of control parameters and methods that can be used to control the 24 GHz radar board using C communication library.

It is worth noting that the radar C communication library implements the API to access a serial communication port for Windows (COMPort_Windows.c), and Mac OS and Linux (COMPort_Unix.c) platforms.

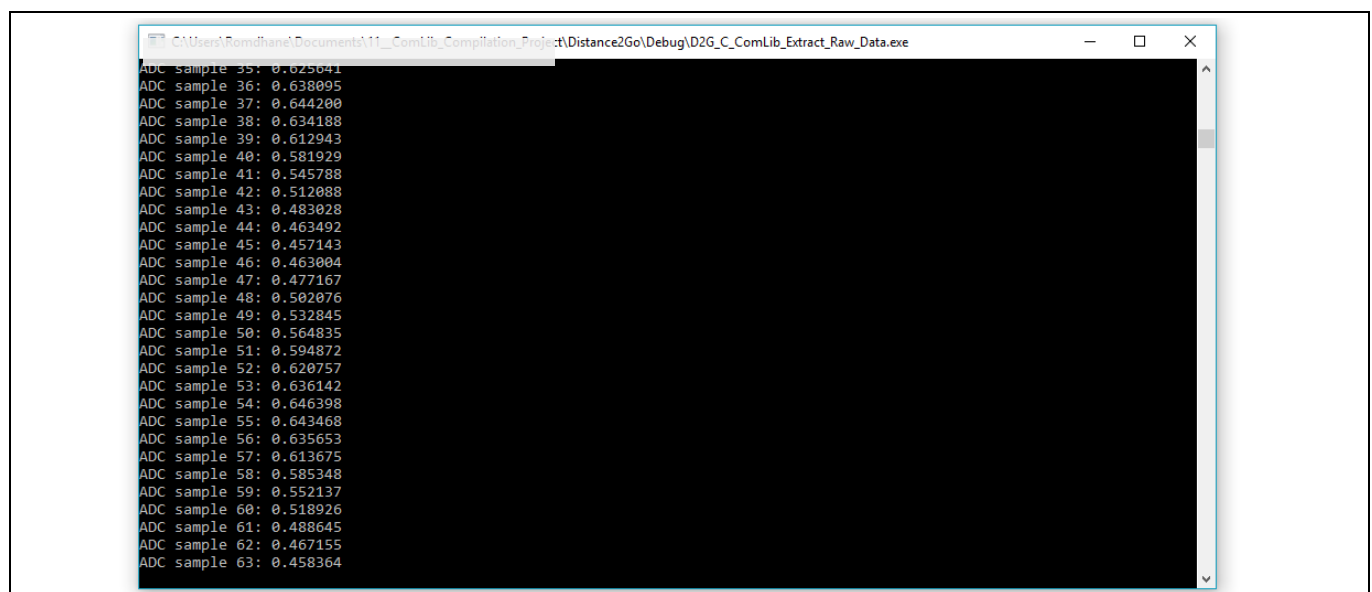
*Note: The C radar system interface is **only** supported within the **Distance2Go**, **Position2Go** and **Sense2GoL Pulse** radar software packages.*

4.3.2 How to use

A coding example to interface with the 24 GHz radar board using C communication library, and to extract raw data, is available (refer to the **extract_raw_data.c** file). In order to run this coding example, proceed as follows:

- Go to the **/Firmware_Software/Communication Library/ComLib_C_Interface/examples** folder, and copy the path.
- Open your appropriate C compiler, then create a C project.
- Import the **extract_raw_data.c** file, all source *.c files under **/ComLib_C_Interface/src** and also all header *.h files under **/ComLib_C_Interface/include**.
- Ensure that your 24 GHz radar board is connected, then **Compile** and **Run** your project to see the raw data.

Figure 36 shows the example running on Visual Studio, extracting raw data from a D2G radar board.



```

C:\Users\Romdhane\Documents\11_ComLib_Compilation_Project\Distance2Go\Debug\D2G_C_ComLib_Extract_Raw_Data.exe
ADC sample 35: 0.625641
ADC sample 36: 0.638095
ADC sample 37: 0.644200
ADC sample 38: 0.634188
ADC sample 39: 0.612943
ADC sample 40: 0.581929
ADC sample 41: 0.545788
ADC sample 42: 0.512088
ADC sample 43: 0.483028
ADC sample 44: 0.463492
ADC sample 45: 0.457143
ADC sample 46: 0.463004
ADC sample 47: 0.477167
ADC sample 48: 0.502076
ADC sample 49: 0.532845
ADC sample 50: 0.564835
ADC sample 51: 0.594872
ADC sample 52: 0.620757
ADC sample 53: 0.636142
ADC sample 54: 0.646398
ADC sample 55: 0.643468
ADC sample 56: 0.635653
ADC sample 57: 0.613675
ADC sample 58: 0.585348
ADC sample 59: 0.552137
ADC sample 60: 0.518926
ADC sample 61: 0.488645
ADC sample 62: 0.467155
ADC sample 63: 0.458364
    
```

Figure 36 Raw data acquisition in Visual Studio shell

Extracting radar raw data

4.4 UART radar system interface

Note: The UART radar system interface is **only** supported within the **Sense2GoL** radar software package.

4.4.1 Overview

The UART radar system interface defines a set of APIs that help the user access a serial communication port and communicate with the radar-defined endpoints, enabling the user to fetch sampled ADC data streamed by UART.

The UART radar system interface supports the following data to be dumped at the host:

- ADC raw data:
 - I and Q (first 256 samples of I, followed by next 256 samples of Q)
 - Only I (256 samples)
 - Only Q (256 samples)
- Doppler measurements and FFT spectrum
- Signed 16-bit, unsigned 16-bit and 32-bit

4.4.2 How to use

Using the transferred data via the UART radar system interface library, it is possible to view and export ADC data via a terminal program (e.g. PuTTY), or transfer it to MATLAB for processing.

PuTTY is a free SSH, Telnet and rlogin client for Windows systems. It is open-source software available with source code and is available for download at: <https://www.putty.org/>

In order to fetch raw data streamed by UART, PuTTY should be configured as follows:

- **UART configuration** – full-duplex, direct mode, 128000 baud rate, 8 data-bits, 1 stop-bit, no parity
- **COM port number** – depends on the host PC; look into the device manager for the COM port number of your connected 24 GHz radar board (e.g. COM53 for Sense2GoL)

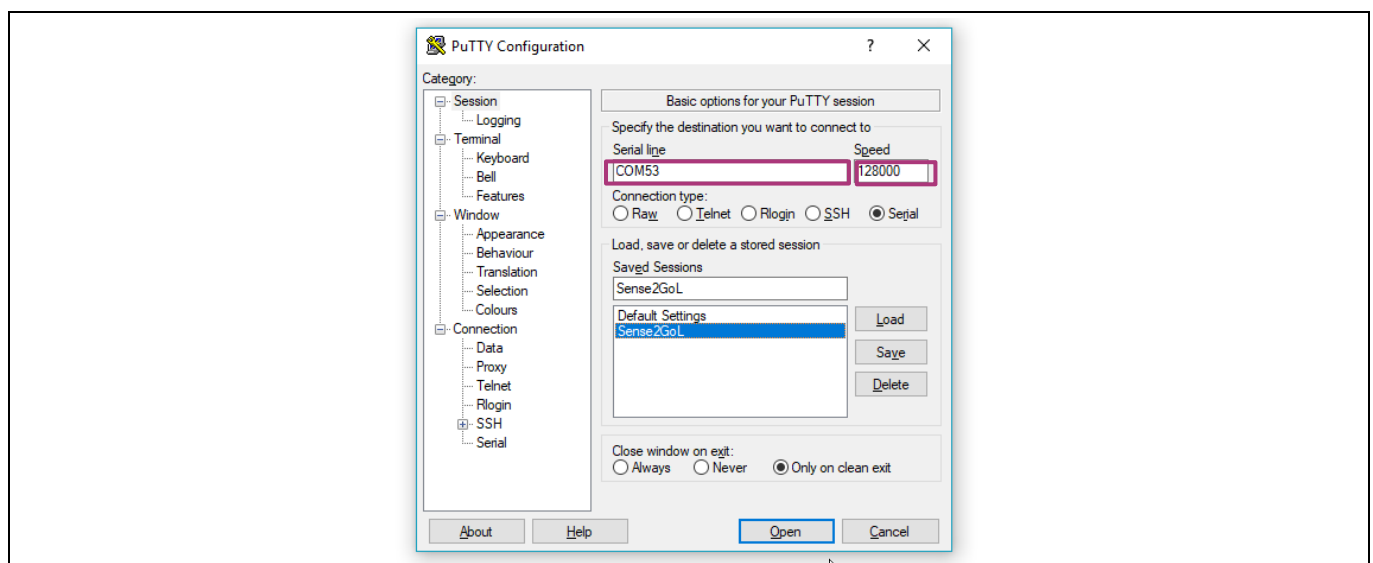
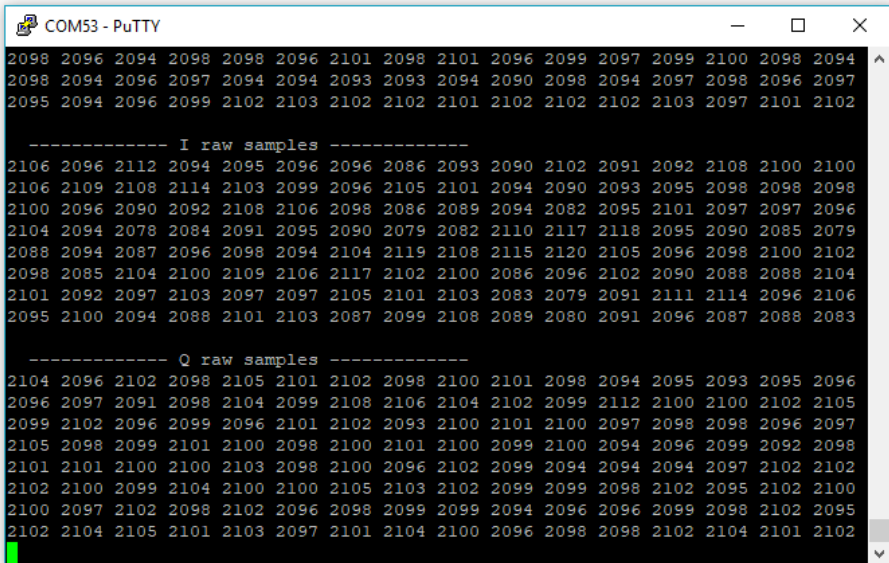


Figure 37 Configure PuTTY for raw data acquisition

Extracting radar raw data

Once configured, click the **Open** button to start a PuTTY serial session. ADC output data at the PuTTY shell for the I and Q raw samples looks as shown in Figure 38.



```
COM53 - PuTTY
2098 2096 2094 2098 2098 2096 2101 2098 2101 2096 2099 2097 2099 2100 2098 2094
2098 2094 2096 2097 2094 2094 2093 2093 2094 2090 2098 2094 2097 2098 2096 2097
2095 2094 2096 2099 2102 2103 2102 2102 2101 2102 2102 2102 2103 2097 2101 2102

----- I raw samples -----
2106 2096 2112 2094 2095 2096 2096 2086 2093 2090 2102 2091 2092 2108 2100 2100
2106 2109 2108 2114 2103 2099 2096 2105 2101 2094 2090 2093 2095 2098 2098 2098
2100 2096 2090 2092 2108 2106 2098 2086 2089 2094 2082 2095 2101 2097 2097 2096
2104 2094 2078 2084 2091 2095 2090 2079 2082 2110 2117 2118 2095 2090 2085 2079
2088 2094 2087 2096 2098 2094 2104 2119 2108 2115 2120 2105 2096 2098 2100 2102
2098 2085 2104 2100 2109 2106 2117 2102 2100 2086 2096 2102 2090 2088 2088 2104
2101 2092 2097 2103 2097 2097 2105 2101 2103 2083 2079 2091 2111 2114 2096 2106
2095 2100 2094 2088 2101 2103 2087 2099 2108 2089 2080 2091 2096 2087 2088 2083

----- Q raw samples -----
2104 2096 2102 2098 2105 2101 2102 2098 2100 2101 2098 2094 2095 2093 2095 2096
2096 2097 2091 2098 2104 2099 2108 2106 2104 2102 2099 2112 2100 2100 2102 2105
2099 2102 2096 2099 2096 2101 2102 2093 2100 2101 2100 2097 2098 2098 2096 2097
2105 2098 2099 2101 2100 2098 2100 2101 2100 2099 2100 2094 2096 2099 2092 2098
2101 2101 2100 2100 2103 2098 2100 2096 2102 2099 2094 2094 2094 2097 2102 2102
2102 2100 2099 2104 2100 2100 2105 2103 2102 2099 2099 2098 2102 2095 2102 2100
2100 2097 2102 2098 2102 2096 2098 2099 2099 2094 2096 2096 2099 2098 2102 2095
2102 2104 2105 2101 2103 2097 2101 2104 2100 2096 2098 2098 2102 2104 2101 2102
```

Figure 38 Raw data acquisition in PuTTY shell

5 Authors

Radar Application Engineering Team, Business Line “Radio Frequency and Sensors”

References

6 References

- [1] 24 GHz industrial radar – [FAQs](#)
- [2] Infineon Application Note – [Working with DAVE™ APPs and moving from DAVE™ v3 to v4](#)
- [3] Infineon Toolbox – [Getting Started guide](#)
- [4] Infineon Micrium® uC/Probe™ XMC™ – [Getting Started guide](#)
- [5] Infineon – [XMC™ Programmers and Flash Tools](#)

Revision history

Revision history

Document version	Date of release	Description of changes
V1.0	2019-06-14	Initial version
V1.1	2020-02-07	Addition of Sense2GoL Pulse demo board features
V1.2	2020-06-25	Addition of firmware flashing via Radar GUI

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-02-07

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2020 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

<UM_1904_PL32_1905_132048>

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.