

# 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<sup>™</sup> 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

## **Table of contents**

Abou	t this document 1
Table	e of contents
List o	f figures 3
List o	f tables 4
1	Introduction
2	Running radar applications
2.1	Tools and software set-up6
2.1.1	Infineon Toolbox6
2.1.2	24 GHz radar tools6
2.1.3	XMC™ Flasher10
2.1.4	DAVE™ toolchain13
2.2	Graphical User Interface (GUI) solutions13
2.2.1	Radar GUI
2.2.1.	1 Overview
2.2.1.	2 How to use14
2.2.2	Micrium μC/Probe™ XMC™ GUI16
2.2.2.	1 Overview16
2.2.2.	2 How to use17
3	Customizing radar applications19
3.1	Building, Flashing and debugging the DAVE™ project19
3.2	Firmware customization and configuration24
4	Extracting radar raw data
4.1	Radar host communication protocol
4.2	MATLAB radar system interface
4.2.1	Overview
4.2.2	How to use
4.3	C radar system interface
4.3.1	Overview
4.3.2	How to use
4.4	UART radar system interface
4.4.1	Overview
4.4.1 4.4.2	Overview
4.4.1 4.4.2 <b>5</b>	Overview
4.4.1 4.4.2 5 6	Overview



## 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 <sup>™</sup> 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	Open Sense2GoL GUI	17
Figure 19	Run the Sense2GoL GUI	17
Figure 20	Start data collection	18
Figure 21	Data display on Micrium GUI	18
Figure 22	DAVE™ workspace launcher	19
Figure 23	DAVE™ project import dialog box	20
Figure 24	Import DAVE™ projects	20
Figure 25	DAVE™ editor view	21
Figure 26	Build project options	21
Figure 27	DAVE™ console	22
Figure 28	Start debug session toolbar button	22
Figure 29	Debug configuration	23
Figure 30	DAVE™ debug perspective	23
Figure 31	MATLAB radar system interface	26
Figure 32	Raw data acquisition in MATLAB command window	27
Figure 33	Raw data acquisition in Visual Studio shell	28
Figure 34	Configure PuTTY for raw data acquisition	29
Figure 35	Raw data acquisition in PuTTY shell	30

#### Introduction

## List of tables

Table 1	XMC <sup>™</sup> devices embedded in 24 GHz radar boards	.10
Table 2	Define statements used for radar firmware configuration	.24



## 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<sup>®</sup> Cortex<sup>™</sup>-M based XMC<sup>™</sup> 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<sup>™</sup> 24 GHz radar demo kits

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

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



**Running radar applications** 

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

*Note:* Please refer to the "<u>Getting Started Infineon Toolbox Guide</u>" 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**

lafinean Taolhay - Infinean tools in a	ne niace (2018-22-2)	- T X
Launcher		
Infineon Toolbox	Tools	4 ? 🖂 🏟 📭
器 Scan QR code	My tools Manage tools	
A You hav	e no installed tools. To search and install tools, switch	to the "Manage tools" tab.
Last Updated : 26/11/2018 17:28	Developer Guide	Infineon Toolbox Version 2018.22.2

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**.

4			
Infineon Toolbox	Tools 🛛 🕶 position		4 0 🖂 🏟 🗗
B Scan QR	code My tools Manage tools		
	A B C D E F G H I J K L M N O P Q R S	тичwхүг	
Position2Go Kit	Position2Go Kit Position2Go Kit consists of three tools for Position2Go application: Getting Started, Radar GUI and XMC Flasher. By installation of Position2Go Kit the other two tools are also installed in parallel.	Version: Tags: Position2Go Kit	Details

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.

Launchar
tools Infineon Toolbox Tools I ▼ position ↓ ⑦ ⊠ ‡ ₽
BE Scan QR code My tools Manage tools
A B C D E F G H L J K L M N O P Q R S T U V W X Y Z Position 200 Position Position 200 Position Position 200 Position Position 200 Position 200

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<sup>™</sup> 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<sup>™</sup> Flasher a tiny, free-of-charge tool for XMC<sup>™</sup> MCU Flash programming



#### **Running radar applications**

uncher				
Infineon Toolbox	<b>T</b> Filter for	tools		¢ 🏟 🕐
My tools Manage tools	BR Scan QR code			Log in   Register
Distance2Go Kit	Position2Go Kit	6	Sense2GoL Kit	Sense2GoL Pulse Kit
Distance2Go Kit	Position2Go Kit	Radar GUI	Sense2GoL Kit	Sense2GoL Pulse Kit
Details	Details	Details	Details	Details
XMC™ Flasher				
XMC Flasher				
Details				

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)



Running radar applications

### 2.1.3 XMC<sup>™</sup> Flasher

The 24 GHz radar demonstration firmware is already pre-loaded in the Flash memory on the XMC<sup>™</sup> microcontroller. This section describes how to use the binary images provided to reprogram the firmware applications. The 24 GHz radar firmware package contains binary images (\*.hex) of the applications provided in the subfolder **Binary**.

The XMC<sup>™</sup> 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<sup>™</sup> Flasher tool to reprogram the radar firmware:
  - Start the XMC<sup>™</sup> Flasher tool in the Infineon Toolbox launcher.

auncher					
Infineon Toolbox		<b>T</b> Filter for t	ools		¢ 🏟 🛈
My tools Man	age tools BRS	can QR code			Log in   Register
Distance2 Kit	Go	Position2Go Kit	6	Sense2GoL Kit	Sense2GoL Pulse Kit
Distance2Go Kit	5	Position2Go Kit	Radar GUI	Sense2GoL Kit	Sense2GoL Pulse Kit
Details		Details	Details	Details	Details
XMC <sup>TM</sup> Start					
XMC Flasher	Ĩ				
Details					

Figure 8 Start XMC<sup>™</sup> 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.

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

Table 1 XMC<sup>™</sup> devices embedded in 24 GHz radar boards



#### **Running radar applications**

Configurations BMI Target Log About		Fi 💌 Select Device Name to connect	×
Connect Disconnect	Select File		
Debugger Type: SEGGER	File name:	List of Targets:	
Debugger Port Configuration: Serial Wire Debug	Size (byte):	XMC4504-512	^
Debug clock speed (KHz): 100	Program	XMC4700-1536	
Connection Status: Not connected		XMC4700-2048	
Selected Emulator Serial Number:	Verify	XMC4800-1024	
Selected Device Name:	France	XMC4800-1536	
Unique Chip ID:	Erase	XMC4800-2048	
File Checksum: 0X0	Dump Flash	XWC+000 2040	~
Device Checksum:			Ok 🖕
Infineon		Cintineon	

Figure 9 XMC<sup>™</sup> Flasher device selection and connection

Note: Please ensure that SEGGER J-Link drivers are installed before using the XMC<sup>™</sup> Flasher tool. Otherwise, the default debugger type under XMC<sup>™</sup> 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).

≶ XMC™ Flasher — □ ×	
File     Con     Target Interface Setup     X       Co     Debugger Type     DAP     •       Debug Port Configuration     Serial Wire Debug     •       Connectitie     Debug Clock Speed (KHz)     100       Selected     Image: Clock Speed (KHz)     100       Selected     Image: Clock Speed (KHz)     100       Reset and verify content after progr     Reset and verify content after progr	File     Con     Target Interface Setup       Co     Debugger Type     SEGGER       Debug Port Configuration     DAP       Debug Clock Speed (KHz)     100       Selected     Show a message when a new version is available       Unique C     Reset and verify content after progr
(1)	(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**

	The configurations and forget cog house
Connect Select File	Connect. Disconnect. Select File.
Debugger Type:     SEGGR     File name:       Debugger Port Configuration:     Serial Wire Debug     Size (byte):       Debug clock speed (0Hz):     100     Program       Connected     Selected Emulator Serial Number:     Systop: Solar       Selected Emulator Serial Number:     SMC4700-2048     Verify       Unique Chip ID:     B20004067095F00807125241     Erase	Debugger Type:     SEGGER     File name:       Debugger Port Configuration:     Serial Wire Debug     Size (byte):       Debug clock speed (KHz):     100     Program       Connection Status:     Connected     Selected Emulator Serial Number:       Selected Emulator Serial Number:     59007561     Verify       Selected Device Name:     XMC4700-2048     Erase
File Checksum:         0x0           Device Checksum:         Press verify to recompute         Dump Flash	File Checksum:         000           Device Checksum:         Press verify to recompute         Dump Flash

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

XMC <sup>™</sup> Flasher	- • ×	S XMC <sup>™</sup> Flasher	
File Configurations BMI Target Log About		File Configurations BMI Target Log About	
Connect. Disconvert.	Select File	Connect Disconnect	Select File.
Debugger Type: Stit GER Debugger Port Confi SEGGER J-Link V6.44e - Flash download (192 KB)	File name: P2G_FMCW.hex Size flustel: 70160	Debugger Type: SEGGER Debugger Port Configuration: Serial Wire Debug	File name: P2G_FW.hes Size (byte): 46408
Debug clock speed ()         Compare         100.0%           Connection Status:         Erase         100.0%	0.0856 4.0366	Debug clock speed (KHz): Connection Status:	× Program
Selected Emulator Se Program 17.7% Selected Device Nam Verify 0.0%	4.701s	Selected Emulator Serial Nul Selected Device Name:	Erase
Unique Chip ID: Programming range 0x0C027800 - 0x0C027FF (2 K8) File Checksum: UX+954D3D Device Checksum: Press verify to recompute	8.822s Dump Flash	File Checksum: Device Checksum:	OK Dump Flash
Programming in progress		Operation completed	
Cinfineon	and the second s	(infineon	Sec.
	N.Lor		Nu

Figure 13 Programming successful

Note: The XMC<sup>™</sup> 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<sup>™</sup> toolchain

DAVE<sup>™</sup> (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<sup>™</sup> industrial microcontroller powered by ARM<sup>®</sup> Cortex<sup>®</sup>-M processors.

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

DAVE v4.1.2 or higher should be installed, or any other third-party toolchain supporting Infineon Technologies XMC<sup>™</sup> microcontrollers, e.g. Atollic, IAR, Keil MDK, Rowley or TASKING. The latest version of DAVE<sup>™</sup> (v4.4.2) can be downloaded from <u>https://infineoncommunity.com/dave-download\_ID645.</u>

## 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.

infineon	Tilter for t			^ <b>*</b> @
Toolbox	Filteriort			竹 韓 ()
My tools Manage tools	III Scan QR code			Log in   Register
Distance2Go Kit	Position2Go Kit	6	Sense2GoL Kit	Sense2GoL Pulse Kit
Distance2Go Kit	Position2Go Kit	Radar GUI	Sense2GoL Kit	Sense2GoL Pulse Kit
Detaits	Details	Update	Details	Details
XMC™ Flasher				
XMC Flasher				
Details				

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

Infineon Toolbox - Infineon tools in one place (2 auncher	020.1.0)				×
Infineon Toolbox	<b>T</b> Filter for t	tools			¢ 🕸 (?
My tools Manage tools	歸 Scan QR code				Log in   Register
Distance2Go Kit	Position2Go Kit	Start		Sense2GoL Kit	Sense2GoL Pulse Kit
Distance2Go Kit	Position2Go Kit	Radar GUI	ŧ	Sense2GoL Kit	Sense2GoL Pulse Kit
Details	Details	Details		Details	Details
XMC™ Flasher					
XMC Flasher					
Details					
1 and 10 shaked - 779 19696 1 4-17	Managed annual				

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

## 2.2.2 Micrium µC/Probe<sup>™</sup> XMC<sup>™</sup> 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<sup>™</sup> from Micrium<sup>®</sup> is a free-of-charge data monitoring and visualization tool to modify and track real-time data on the XMC<sup>™</sup> target microcontrollerin a non-intrusive way.

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

µC/Probe<sup>™</sup> XMC<sup>™</sup> 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<sup>™</sup> kits.

The latest version of μC/Probe<sup>™</sup> XMC<sup>™</sup> v4.3.0.9 is available for download from: <u>https://infineoncommunity.com/uC-Probe-XMC-software-download\_ID712</u>



Running radar applications

## 2.2.2.2 How to use

The Sense2GoL software package (S2GL-HW-SW.exe) comes with a GUI based on a µC/Probe<sup>™</sup> 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 µC/Probe<sup>™</sup> **S2GL\_GUI.wspx** project, to open the GUI.
- **S2GL\_GUI.wspx** can be opened in Windows Explorer, or in S2GL\_Doppler DAVE<sup>™</sup> project explorer, as shown in Figure 18.



Figure 18 Open Sense2GoL GUI

The  $\mu$ C/Probe<sup>TM</sup> needs to be provided by the XMC<sup>TM</sup> 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  $\mu$ C/Probe<sup>TM</sup> 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.

😒   🔮 🖳 🖸 🗔 🚔 🍏 🔮			Micriµm µC/Probe		
File Design View					۵
Setting: Runa: Stop Paste En Cons Fun the current workspace Application Clipboard	Reprinter Tool Pictures , A Text X G <sup>P</sup> Connector ∰2 Tools	Select Connectors	Shape Styles + +	$ \begin{array}{c} A^{*} \\ Font \\ \bullet \end{array} \end{array} \begin{array}{c} \Xi \equiv \Xi \\ \Xi \equiv \Xi \end{array} $ Paragraph	

Figure 19 Run the Sense2GoL GUI

• The GUI interface in Figure 20 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 20 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 21 Data display on Micrium GUI



Customizing radar applications

## 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<sup>™</sup> Cortex<sup>™</sup>-M MCU embedded in Infineon's 24 GHz radar demo kits.

The 24 GHz radar firmware is released as a ready-to-run DAVE<sup>™</sup>4 project, where source files are generated based on the DAVE<sup>™</sup> 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<sup>™</sup> code, and run them on the 24 GHz radar demo kit.

## 3.1 Building, Flashing and debugging the DAVE<sup>™</sup> project

After installing the DAVE<sup>™</sup> 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<sup>™</sup> toolchain:
  - Run the DAVE<sup>™</sup> 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.

Component based programming with DAVE™ APPs	DAVE™	Infineon	Eclipse Launcher       ×         Select a directory as workspace       DAVE <sup>w</sup> uses the workspace directory to store its preferences and development artifacts.
	Co pr DA	mponent based ogramming with VE™ APPs	Workspace: C:\Workspace\DWE\24ghz V Browse   Recent Workspaces  Copy Settings

Figure 22 DAVE<sup>™</sup> 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<sup>™</sup> 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.



#### Customizing radar applications

Close       Curl+W         Close All       Curl+Shift+W         Save       Curl+Shift+W         Save As       Save As         Save As       Save As         Move       Ferame         Move       F2         Refresh       F5         Convert Line Delimiters To       >         Suitb Worknese       >	New Open File	Alt+Shift	+N > P	Select
Close       Ctrl+W         Close All       Ctrl+Shift+W         Save       Ctrl+Shift+W         Save As       Save As         Save As       Save All         Move       F2         Move       F2         Refresh       F3         Corvert Line Delimiters To       >         Swith Worksnere       >	C Open Pro	ejects from File System		
Save     Ctrl+S       Save As     >> Ctrl+S       Save All     Ctrl+Shift+S       Revert     Build Strings       Move     Refresh       Refresh     F2       Print     Ctrl+P       Switch Workmana     >       Switch Workmana     >	Close Close All	Ctrl- Ctrl+Shift-	+ W + W	Select an import wizard: type filter text
Move     Extra Provide Control Control Data State       Move     F2       Refresh     F5       Convert Line Delimiters To     >       Print     Ctrl+P	Save Save As Save All Revert	Ctrl+Shift	1+S t+S	<ul> <li>&gt; ≥ General</li> <li>&gt; ≥ C/C++</li> <li>&gt; ≥ Infineon</li> <li>⇒ Build Settings</li> <li>≥ DAVE Project</li> </ul>
Print Ctrl+P Switch Workmanne	Move Rename Refresh Convert I		F2 F5	
Restart	Print Switch W Restart	Ctri forkspace	1+P >	

Figure 23 DAVE<sup>™</sup> project import dialog box

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

Import DAVE Projects	– 🗆 🗙	Simport DAVE Projects	
Import DAVE projects		Import DAVE projects	
Select a folder/Archive to search for existing DAVE projects.		Import Existing DAVE Projects	
Select Root Directory	Browse	Select Root Directory     :\Users\Romdhane\Documents\Position2Go	Browse
O Select Archive File	Browse	O Select Archive File	Browse
Project List:		Project List:	
	Select All	P2G_FMCW(C:\Users\Romdhane\Documents\Position2Go\P2G_FMCW)	Select All
	Deselect All	P2G_FW(C:\Users\Romdhane\Documents\Position2Go\P2G_FW)	Deselect All
	Refresh		Refresh
Copy Projects Into Workspace		Copy Projects Into Workspace	
(2) r Pack Novt > Ein	ich Cancel	(?) < Back Next > Finish	Cancel

Figure 24 Import DAVE<sup>™</sup> projects

• The project file should appear under the **Project List**. Click on the **Finish** button. A screen similar to Figure 25 should appear.



### Customizing radar applications

	AVE Window Help		
🗑 🕼 🗡 🖉 🥕 🖆 👘 📩 👘		ick Access	8 😺
🔂 C/C++ 🔀 🏠 Project 🖓 🗖			
<ul> <li></li></ul>			
APP Dependency Tree 😢 🖓 🖬 🔂	APP Dependency 💥 💄 HW Signal Connectivity 🖻 Console 🔲 Properties	Problems	
V ADC_MEASUREMENT_ADV_G A		<b>r</b> } €	Q 🕇
V GLOBAL ADC 0	NTERUPT TMER	DIGITAL_IO	^
CLOCK_XMC4_0			

Figure 25 DAVE<sup>™</sup> 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.

24ghz - DAVE CE - P2G_FW/main.c - DAV         File       Edit       Source       Refactor       Navigate       S         Image: Source       Image: Source <t< th=""><th>E<sup>™</sup> - C:\Workspac Search Project main.c № main.c № 30 2. 31</th><th>24ghz - DAVE C      File Edit So      File Cit So      C C + +       So      C C +       C C +</th><th>E - P2G_FW/main.c - DAVE™ - C:\\wo New Go Into Open in New Window Index Build Targets Build Configurations Build Project Clean Project Copy Paste Delete Move Rename</th><th>rkspaces\DAV</th></t<>	E <sup>™</sup> - C:\Workspac Search Project main.c № main.c № 30 2. 31	24ghz - DAVE C      File Edit So      File Cit So      C C + +       So      C C +       C C +	E - P2G_FW/main.c - DAVE™ - C:\\wo New Go Into Open in New Window Index Build Targets Build Configurations Build Project Clean Project Copy Paste Delete Move Rename	rkspaces\DAV
> 🧽 Radar_Control > 🧽 Startup > 🍋 Store 🗸 🗸	45 XI 46 wl 47 } 48	> 🥭 Star > 👝 Star > 🍌 Stor	Import Export	
APP Dependency Tree 🛛 🕒 🕇	49 /*	APP Depen	DAVE Project Upgrade	>

Figure 26 Build project options

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



#### Customizing radar applications



Figure 27 DAVE<sup>™</sup> 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 28. Alternatively, press key F11 to directly start the debug session.



Figure 28 Start debug session toolbar button

If opening Debug Configurations, a window as shown in Figure **29** 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

Debug Configurations		×	Debug Configurations				
reate, manage, and run configuratio	ns	Ť.	Create, manage, and run configuratio	ns			Ś
Image: The second se	Configure launch settings from this dialog:		Image: The set of the se	Name     P2G_FW Build       Imain     Stebugger       Project:     P2G_FW       P2G_FW     C/C + Application:       Build(P2G_FW.eff       Build (of required) before launching       Build Configuration:       Select Au       C Inable auto build       Image: Use workspace settings	Variables Variables ing tomatically O Dis <u>Config</u>	Common Search Project  Search Project  able auto build uure Workspace Settin Revert	Browse Browse Srowse
Filter matched 1 of 25 items	Debug	Close	Filter matched 2 of 26 items		_	Revert Debug	Ap (

Figure 29 Debug configuration

• In this way, the Flashing process is started, the firmware image is loaded into the XMC<sup>™</sup> microcontroller Flash memory and DAVE<sup>™</sup> automatically switches to the debug perspective, as shown in Figure 30.

📓 (\$\\$	S		🗛 🕹 🔛 🔛 👘
🏘 Debug 🔀 🖏 Servers 🛛 🛛 Resume (F8) 🛛 💥 🕯 🔽 🗖 🗄	] (×)= Variables ∑  ●₀ Brea	akpo 🚻 Registers 🔀 Periph	ner 🖹 Modules 🗖 🗖
<ul> <li>▼ CP2G_FW Build [GDB SEGGER J-Link Debugging]</li> <li>→ P2G_FW.elf</li> <li>→ P1read #1 57005 (Suspended : Breakpoint)</li> </ul>	Name	Type DAVE STATUS †	Kalue ≤ optimized out>
main() at main.c:35 0x8029368 JLinkGDBServerCL.exe	<		
arm-none-eabi-gdb Semihosting and SWV	<		, û , j
i main.c ⊠		🗢 🗖 🏧 Disassem	🛛 📴 Outline 🖵 🗖
<pre>32 */ 33 340 int main(void) 35  36 DAVE_STATUS_t status; 37 38 /* Initialize DAVE_APPs */ 39 status = DAVE_Init(); 40 41 if(status != DAVE_STATUS_SUCCESS) 42 { 430 /* Placeholder for error handler code. 4 * The while loop below can be replaced with an user er 45 YMC_DERUGY("DAVE_APPs initialization failed\n"); 44 45 45 45 45 45 45 45 45 45 45 45 45</pre>	ror handler. */	↓ 08029368 39 08029368 41 08029368 46 08029370 52 08029370 55 08029376	<pre>main:</pre>
🔄 Console 🛛 🧔 Tasks 🖹 Problems 🜔 Executables 📃 🗖	∃ 🚺 Memory 🔀	🗋 🛃 📑	🔢 🔄 ₩ ▾ ▽ 🗖 🗖
■ X 🧏   🖳 🚮 📴 🥮 ( 🕊   🖻 🗲 1 P2G_FW Build (GDB SEGGER J-Link Debugging) Semihosting and SWV SEGGER J-Link GDB Server V6.44e - Terminal output channel	Vertex Ve		

Figure 30 DAVE<sup>™</sup> debug perspective



Customizing radar applications

## **3.2** Firmware customization and configuration

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

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

 Table 2
 Define statements used for radar firmware configuration

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 preceeding 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.



### 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.

HOME PLOTS	APPS EDITOR PUBLISH VIEW	🖥 🎸 🕼 🛍 🐄 🖨 🔁 🕐 Search Doc	cumentation 🔎 Log I
New Open Save		Run and Time	
🗣 🔿 🔁 🧧 🖌 🖉 🚺	erer + D2G_V280 + Firmware_Software + Communication Library + ComLib_Matlab_Interface	RadarSystemExamples      GettingStarted	- ,
Current Folder	Editor - C:\Users\Romdhane\Documents\FW_Releases\D2G_V2.0.0\Firmware_Software\Com	munication Library\C 💿 🗙 Workspace	(
🗋 Name 🔺	extract_raw_data.m × +	Name 🛎	Value
PiotDemo.m	<pre>39 - addpath('\\RadarSystemImplementation'); % add Matlab ; 40 - clear all %fok<clscr> 41 - close all 42 - resetRS; % close and delete ports 43 44 % 1. Create radar system object 45 - szPort = findRSPort; % scan all available ports 46 - oRS = RadarSystem(szPort); % setup object and connect to b 47 48 - disp('Connected RadarSystem:'); 49 - oRS %fok&lt;'NOPTS&gt; 50</clscr></pre>	API	aki celi ata 256k1 complex doub 1k1 RodarSystem 1k1 struct <u>'COM11'</u> 256k1 complex doub
extract_raw_data.m (Script) >> %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%	51       % 2. Enable automatic trigger with frame time 1s         52       oRS.oEPRadarBase.set_automatic_frame_trigger(1000000);         53       54         54       while true         55       % 3. Trigger radar chirp and get the raw data		
\$1555757575757575757575757575 \$157575757575757575757575 \$157575757575757575757575 \$157575757575757575 \$15757675757575	56 - [mxRawData, sInfo] = oRS.oEPRadarBase.get_frame_data; 57 - ydata = mxRawData; % get raw data <a< td=""><td><b>,</b></td><td></td></a<>	<b>,</b>	
Cleanup and init	Command Window	(*) ^	
	0.77	v L	

• Ensure that your 24 GHz radar board is connected, then Click on **Run** to see the raw data.

#### Figure 31 MATLAB radar system interface

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



#### Extracting radar raw data



Figure 32 Raw data acquisition in MATLAB command window



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

Children Doministral Doministria Commitation Broket Nictance2Gol Debush D2G. C. Comilib Extract Paus Data eve	×	
	~	
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: 0.545/88		
ADC sample 42: 0.512088		
ADC sample 43: 0.483028		
ADC Sample 44: 0.463492		
AUC Sample 45: 0.45/143		
AUC sample 40: 0.403004		
AUC sample 47: 0.4//10/		
AUC sample 48: 0.5020/0		
AUC sample 54, 0,532045		
AUC sample 54. 0.504055		
AUC sample 51. 0.594072		
ADC sample 52. 0.6261/2		
ADC sample 57: 0.613675		
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 33 Raw data acquisition in Visual Studio shell



#### 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: <u>https://www.putty.org/</u>

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 34 Configure PuTTY for raw data acquisition



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 35.

🛃 С	OM53 -	PuTTY											-		>
098	2096	2094	2098	2098	2096	2101	2098	2101	2096	2099	2097	2099	2100	2098	2094
098	2094	2096	2097	2094	2094	2093	2093	2094	2090	2098	2094	2097	2098	2096	2097
095	2094	2096	2099	2102	2103	2102	2102	2101	2102	2102	2102	2103	2097	2101	2102
			- Ira	aw san	nples										
106	2096	2112	2094	2095	2096	2096	2086	2093	2090	2102	2091	2092	2108	2100	2100
106	2109	2108	2114	2103	2099	2096	2105	2101	2094	2090	2093	2095	2098	2098	2098
100	2096	2090	2092	2108	2106	2098	2086	2089	2094	2082	2095	2101	2097	2097	2096
104	2094	2078	2084	2091	2095	2090	2079	2082	2110	2117	2118	2095	2090	2085	2079
880	2094	2087	2096	2098	2094	2104	2119	2108	2115	2120	2105	2096	2098	2100	2102
098	2085	2104	2100	2109	2106	2117	2102	2100	2086	2096	2102	2090	2088	2088	2104
101	2092	2097	2103	2097	2097	2105	2101	2103	2083	2079	2091	2111	2114	2096	2106
095	2100	2094	2088	2101	2103	2087	2099	2108	2089	2080	2091	2096	2087	2088	2083
			-Qra	aw sar	nples										
104	2096	2102	2098	2105	2101	2102	2098	2100	2101	2098	2094	2095	2093	2095	2096
096	2097	2091	2098	2104	2099	2108	2106	2104	2102	2099	2112	2100	2100	2102	2105
099	2102	2096	2099	2096	2101	2102	2093	2100	2101	2100	2097	2098	2098	2096	2097
105	2098	2099	2101	2100	2098	2100	2101	2100	2099	2100	2094	2096	2099	2092	2098
101	2101	2100	2100	2103	2098	2100	2096	2102	2099	2094	2094	2094	2097	2102	2102
102	2100	2099	2104	2100	2100	2105	2103	2102	2099	2099	2098	2102	2095	2102	2100
100	2097	2102	2098	2102	2096	2098	2099	2099	2094	2096	2096	2099	2098	2102	2095
102	2104	2105	2101	2103	2097	2101	2104	2100	2096	2098	2098	2102	2104	2101	2102

Figure 35 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<sup>™</sup> APPs and moving from DAVE<sup>™</sup> v3 to v4
- [3] Infineon Toolbox <u>Getting Started guide</u>
- [4] Infineon Micrium<sup>®</sup> uC/Probe<sup>™</sup> XMC<sup>™</sup> <u>Getting Started guide</u>
- [5] Infineon <u>XMC<sup>™</sup> Programmers and Flash Tools</u>



## **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

#### 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 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.

## **Mouser Electronics**

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Infineon: DEMOSENSE2GOLTOBO1