

XENSIV™ 24 GHz radar demo boards

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 boards: DEMO SENSE2GOL/MAKE, DEMO DISTANCE2GO, DEMO POSITION2GO, DEMO SENSE2GOL PULSE and DEMO DISTANCE2GOL.

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

The intended audience for this document are design engineers, technicians, and developers of electronic systems, working with Infineon's XENSIV[™] 24 GHz radar sensors.

Related documents

Additional information can be found in the documentation provided with the Radar Sense2GoL Pulse or Radar Distance2GoL tools in the Infineon Developer Center (IDC), or from www.infineon.com/24GHz.

Note: Following the launch of our next generation of XENSIV[™] Radar 24GHz DEMO boards with SENSE2GOL PULSE and DISTANCE2GOL, please be informed that these XENSIV[™] Radar 24GHz demo boards are therefore discontinued: SENSE2GO, SENSE2GOL, DISTANCE2GO and POSITION2GO.



Table of contents

Table of contents

Abou	ut this document	
Table	e of contents	2
1	Introduction	
2	Running radar applications	
2.1	Tools and software set-up	4
2.1.1	Infineon Developer Center (IDC)	4
2.1.2	24 GHz radar IDC tools	4
2.1.3	XMC™ Flasher	6
2.1.4	DAVE™ IDE	
2.2	Graphical User Interface (GUI) solutions	
2.2.1	Radar GUI	
2.2.2	Micrium μC/Probe™ XMC™ GUI	14
3	Customizing radar applications	
3.1	Building, Flashing and debugging the DAVE™ project	
3.2	Firmware customization and configuration	22
4	Extracting radar raw data	23
4.1	Radar host communication protocol	23
4.2	MATLAB radar system interface	23
4.3	C radar system interface	25
4.4	UART radar system interface	26
Refe	rences	28
Revis	sion history	29
	۔ اaimer	

XENSIV[™] 24 GHz radar demo boards



1 Introduction

1 Introduction

The 24 GHz radar demo boards 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.

- DEMO SENSE2GOL (S2GL) BGT24LTR11 + XMC1302 Infineon radar demo board based on the BGT24LTR11 – Doppler (motion, speed and direction of movement detection).
- DEMO 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).
- **DEMO 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).
- **DEMO 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.
- **DEMO DISTANCE2GOL (D2GL)** Radar Baseboard XMC4700 and BGT24LTR11 Shield with Arduino compatibility Infineon radar demo board based on the BGT24LTR11 Software-Controlled FMCW (motion, speed, direction of movement of multiple targets detection and distance of closest human or moving target 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).

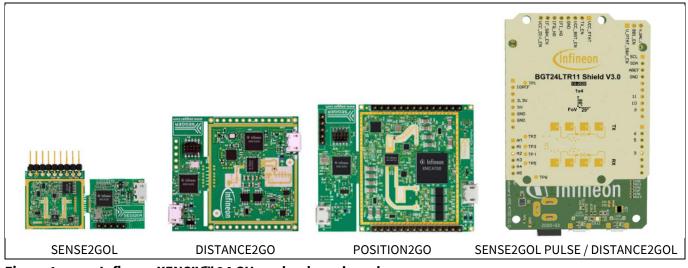


Figure 1 Infineon XENSIV[™] 24 GHz radar demo boards

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 board.

XENSIV[™] 24 GHz radar demo boards



2 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 departing)
- Distance of multiple targets
- Position of multiple targets

2.1 Tools and software set-up

2.1.1 Infineon Developer Center (IDC)

Before using the 24 GHz radar demo boards, 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 Developer Center (IDC) Launcher (former Infineon Toolbox) using this link: https://www.infineon.com/cms/en/design-support/tools/utilities/infineon-developer-center-idc-launcher/

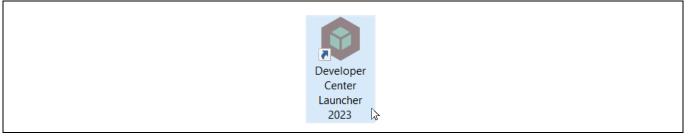


Figure 2 Run Infineon Developer Center Launcher

2.1.2 24 GHz radar IDC tools

Once the IDC Launcher is successfully installed, you can proceed with downloading the appropriate 24 GHz radar tool: Radar Sense2GoL, Radar Distance2Go, Radar Position2Go, Radar Sense2GoL Pulse or Radar Distance2GoL including software package and documentation, as detailed in the following instructions:

• Run the Infineon IDC Launcher by double-clicking on the tool icon on your Windows desktop. The program starts under the **My Tools** tab, as shown in Figure 3.



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

	lnfineon De	veloper Center - In	fineon tools in one place	3			– 🗆 X	
	Dev	eloper Cent	er Launcher		Filter results	Q	0	
		My Tools	Manage Tools	器 Scan QR Code			Log in	
					A No installed tools availab	ble		
							Version 2023.2.0 Check for Launcher update	
Figure	3	Му То	ols under	IDC Launc	her			

• Click on the **Manage Tools** tab, then type the 24 GHz tool name of your choice (e.g., Radar Distance2GoL) in the search box and press **Enter**.

Infineon Developer Center - Infi	fineon tools in one place			- 🗆 X
Developer Cente	er Launcher	distance2gol	Q	© 0
My Tools	Manage Tools 🛛 🔀 Scan QR Code			Log in
1 results	# A B C D E F G	H I J K L M N O P Q	R S T U V W X Y	Z
R XENSIV™	software-controlled FMCW demo	Distance2GoL, is a BGT24LTR11 radar sensor board for range measurements and 1D tracking of ar GUI can be also installed in parallel.	Version: 1.0.1.202209231118 File size: 198.8 MB Tags: XENSIV,24GHz,Radar,	Install Details
			version 2023.2.0	Check for Launcher update

Figure 4 Radar tools installation via IDC

- Click on Install, and then on Next.
- Once installed, click on the installed tool icon to go to the start page.
- As shown in Figure 5. By installing the Radar Distance2GoL tool, the Radar GUI tool can also be installed in parallel.



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

Infineon Developer Center - Infineon tools in one place	8			-	
Developer Center Launcher		Filter results	Q		0
My Tools Manage Tools	器 Scan QR Code		N	Iontassar.BenRomdhane@infineon.com	Log out
		_			
(((
XENSIV™	XENSIV™				
Radar Distance2GoL	Radar GUI				
Version:	Version:				
1.0.1.202209231118	2.8.2.202205190910				
Details	Details				
				Version 2023.2.0 Check for Lau	

Figure 5 24 GHz radar installed tools (e.g., Radar Distance2GoL)

User can download and install the 24 GHz radar SW package following the steps on the "Getting started" page of each installed tool via the Infineon Developer Center (IDC).

Once installed, a folder (e.g., IFX_D2GL-HW-SW_V1.0.0) 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., 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. 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.
- Use (*.hex) binary with the XMC[™] Flasher tool to reprogram the radar firmware:
 - Start the XMC[™] Flasher tool in the Infineon Developer Center (IDC) launcher



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

Infineon Developer Center - Infineon tools in one place	-			- 🗆 ×
Developer Center Launcher	Fil	ter results	Q	0
My Tools Manage Tools	課 Scan QR Code		Montassar.Be	nRomdhane@infineon.com Log out
D))) Xensiv™	D))) Xensiv™	Start.		
Radar Distance2GoL	Radar GUI	⑦ 面 XMCFlasher		
Version: 1.0.1.202209231118	Version: 2.8.2.202205190910	Version: 1.1.0.202103021918		
Details	Details	Details		
				Version 2023.2.0 Check for Launcher update

Figure 6 Start XMC[™] Flasher tool

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	
DEMO SENSE2GOL	XMC1302-0016	
DEMO DISTANCE2GO	XMC4200-256	
DEMO POSITION2GO	XMC4700-2048	
DEMO SENSE2GOL PULSE	XMC4700-2048	
DEMO DISTANCE2GOL	XMC4700-2048	



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

Configurations BMI Target Log About		XMC [™] Flasher Fi ♥ Select Device Name to connect	×
Connect Disconnect Debugger Type: SEGGER	Select File File name: Size (byte):	List of Targets:	
Debugger Port Configuration: Serial Wire Debug		XMC4504-512	
Debug clock speed (KHz): 100	Program	XMC4700-1536	
Connection Status: Not connected Selected Emulator Serial Number:	Verify	XMC4700-2048	
Selected Emulator Serial Number: Selected Device Name:	veniy	XMC4800-1024	
Selected Device Name: Unique Chip ID:	Erase	XMC4800-1536	
File Checksum: 0X0		XMC4800-2048	
Device Checksum:	Dump Flash		×
			Ok
Infineon	Contraction of the second seco	(Infineon	*

Figure 7 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 8a. Once installed, the user must change the debugger type to SEGGER, as
shown in Figure 8b.

XMC [™] Flasher - ×	File Con Target Interface Setup	>
Connecting Debugger Type DAP Debug Port Configuration Serial Wire Debug Connecting Debug Clock Speed (KHz) Selected Show a message when a new version is available Unique C Reset and verify content after progr	Co Debugger Type SEGGER Debug Port Configuration SEGGER DAP DAP Selected Selected Selected V Show a message when a new version is available Unique C	
(a)	(b)	k Cancel



- 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.

XENSIV[™] 24 GHz radar demo boards

infineon

2 Running radar applications

Configurations BMI Target	Log About		File Configurations BMI Target	Log About	
Connect	Disconnect	Select File	Connect	Disconnect	Select File
Debugger Type:	SEGGER	File name:	Debugger Type:	SEGGER	File name:
Debugger Port Configuration:	Serial Wire Debug	Size (byte):	Debugger Port Configuration:	Serial Wire Debug	Size (byte):
Debug clock speed (KHz):	100	Program	Debug clock speed (KHz):	100	Program
Connection Status:	Connected		Connection Status:	Connected	
Selected Emulator Serial Number:	599007561	Verify	Selected Emulator Serial Number:	599007561	Verify
Selected Device Name:	XMC4700-2048		Selected Device Name:	XMC4700-2048	
Unique Chip ID:	B200004067095F00807129241	Erase	Unique Chip ID:	B200004067095F00807129241	Erase
File Checksum:	axa		File Checksum:	OXO	
Device Checksum:	Press verify to recompute	Dump Flash	Device Checksum:	Press verify to recompute	Dump Flash
			1111111111111		
Infineon			(Intineon		and the second s

Figure 9

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

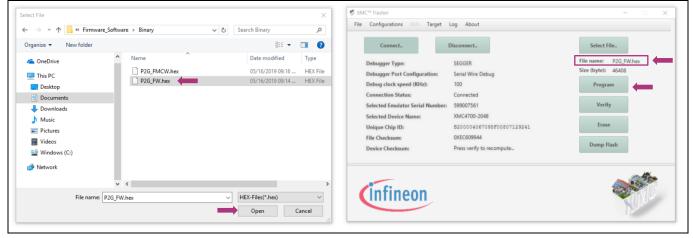


Figure 10 Binary image selection

- 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.



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

MC™ Flasher		– 🗆 ×	S XMC [™] Flasher	
Configurations BMI Target Log About			File Configurations BMI Target Log About	
Connect Disconvect		Select File	Connect., Disconnect	Select File
Debugger Type: SEGGER		File name: P2G_FMCW.hex	Debugger Type: SEGGER	File name: P2G_FW.hex
Debugger Port Confi SEGGER J-Link V6.44e - Flas	h download (192 KB)	Size (bute): 70160	Debugger Port Configuration: Senal Wire Debug	Size (byte): 46408
Debug clock speed () Compare	100.0%	0.085	Debug clock speed (KHz):	Program
Connection Status: Erase	100.0%	4.036s	Connection Status:	
Selected Emulator Se Program	17.7%	4.701s ¥	Selected Emulator Serial Nur Programming is successful !	Verify
Selected Device Nam Verify	0.0%		Selected Device Name:	
Unique Chip ID: Programming ra	nge 0x0C027800 - 0x0C027FFF (2 KB)	8.822s	Unique Chip ID:	Erase
File Checksum: 0XF1954D3		Dump Flash	File Checksum:	Dump Flash
Device Checksum: Press verify	to recompute	Dump mash	Device Checksum:	OK Dump Hash
ogramming in progress		1 m v	Operation completed	-
Infineon		Contraction of the second seco	Cinfineon	and the second s
IIIIIcoli			Infineon	S. De
		NIC		

Figure 11 Successful firmware programming

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 board.

2.1.4 DAVE[™] IDE

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[™] IDE (v4.5.0) can be downloaded from IDC using https://softwaretools.infineon.com/tools/com.ifx.tb.tool.daveide

2.2 Graphical User Interface (GUI) solutions

2.2.1 Radar GUI

Note: The Radar GUI tool is only supporting DEMO DISTANCE2GO, DEMO POSITION2GO, DEMO SENSE2GOL PULSE and DEMO DISTANCE2GO 24GHz radar demo boards.

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
- **Time domain data** records extracted time domain data (I/Q signals); data are stored in a file with .tdd extension



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

- 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

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

- Run IDC launcher by double-clicking on the launcher 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.

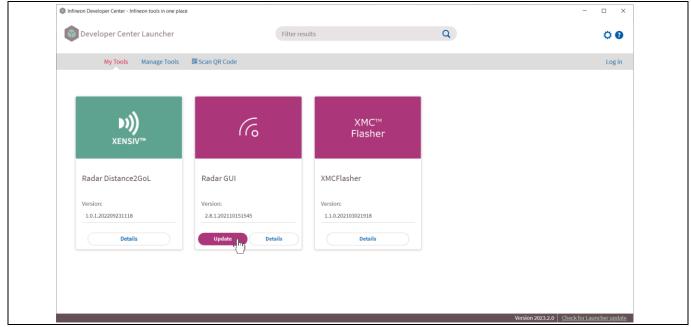


Figure 12 Update Radar GUI tool to latest version

• Click on the **Start** button of the Radar GUI application available under the **My Tools** tab.

Infineon Developer Center - Infineon tools in one place	9			- 🗆 ×
Developer Center Launcher	Filter resu	lts	Q	0
My Tools Manage Tools	器 Scan QR Code			Log in
►))) XENSIV™	Start XENSIV TH	XMC™ Flasher		
Radar Distance2GoL	⑦ ា Radar GUI	XMCFlasher		
Version: 1.0.1.202209231118 Details	Version: 2.8.2.202205190911 V Details	Version: 1.1.0.202103021918 Details		
			Version 2	2023.2.0 Check for Launcher update

Figure 13 Start Radar GUI tool



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

• 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.

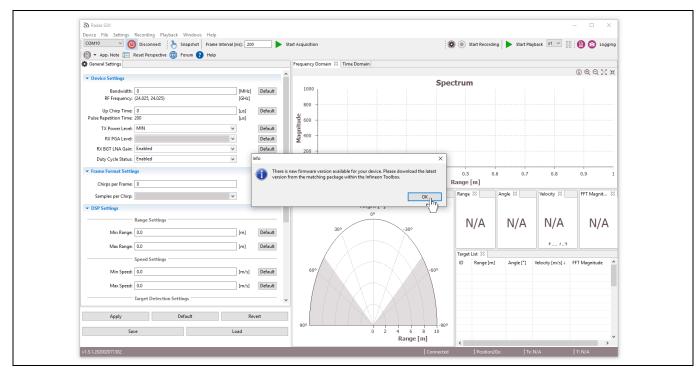


Figure 14 New firmware version available window

- Radar GUI automatically starts acquiring data from the 24 GHz radar board, when the radar firmware is running. Figure 15 shows the Radar GUI default screen acquiring FMCW data from DEMO POSITION2GO radar board, 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.



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications



Figure 15 Radar GUI window details

• In addition to XMC[™] 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 16.

ଲ Radar GUI	
Device File Settings Recording Playt	ack Windows Help
O Disconnect	Interval [ms]: 111.111 Stop Acquisition
Stop Acquisition (Ctrl+Space)	
Flash Firmware	
Apply Flash Firmware Default Cancel	

Figure 16 Flash firmware menu in Radar GUI

- 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 17. 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** button in the dialog box will start flashing the new (*.hex) file into the device.



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

nware Flasher for Sense2GoLPulse Device selection from drop down list if not already detected		Drop down list of preloaded available .hex files
	Select Device Sense2GoLPulse	Select Firmwarel S2GL_Pulsed_Doppler.hex
Firmware Image: C:/Infineon/Toolbox	plugins/com.ifx.tb.tool.radargui.rcp_2.0.0.202006221503/resources/FlashTool/xmr	c/S2GL_P/S2GL_Pulsed_Doppler.hex I Browse
.hex File Path	Flash Firmwa	re
Log		
		^
		~

Figure 17 Firmware flasher interface in Radar GUI

• 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 18.

Firmware Flashing Result	×
Successfully Flashed	
-	
	OK

Figure 18 Firmware flashing result

2.2.2 Micrium µC/Probe[™] XMC[™] GUI

Note: The Micrium GUI tool is only supporting DEMO SENSE2GOL/MAKE 24GHz radar demo boards and is not compatible with the Radar GUI.

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



XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

The Radar Sense2GoL software package comes with a GUI based on a µC/Probe™ project from Micrium, which helps the user process collected raw data.

Here below an example, on how to run this Micrium-based GUI project, for the Sense2GoL device:

- Go to the /Firmware_Software/GUI folder inside the locally installed S2GL-HW-SW package.
- Double-click the µC/Probe[™] **S2GL_GUI.wspx** project, to open the GUI.
- **S2GL_GUI.wspx** can be opened in Windows Explorer, or in S2GL_Doppler DAVE[™] project explorer, as shown in Figure 19.

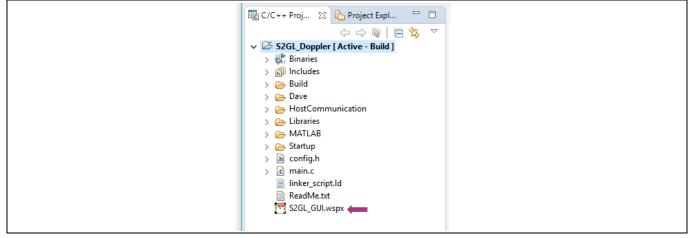


Figure 19 Open Sense2GoL Micrium GUI project

The μ C/ProbeTM needs to be provided by the XMCTM 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/ProbeTM 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 DEMO SENSE2GOL board via USB (please ensure the USB port on the debugger board is connected).
- Start the GUI by clicking on the **Run** button.

요 🎛 🖳 🖸 🔟 🖬 🧀 灯 🕑	Micriµm µC/Probe	
File Design View		۵
Settings Application Cippoard Application Cippoard A Cut Paste En Conv A Text Connector Cippoard Conv C	$\begin{tabular}{ c c c c c } \hline & & & & & & & & & & & & & & & & & & $	





XENSIV[™] 24 GHz radar demo boards

2 Running radar applications

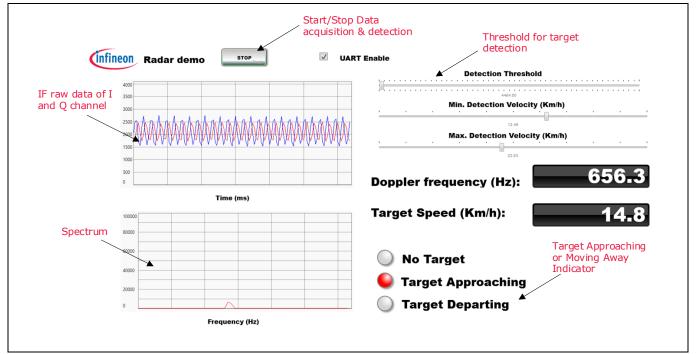
- The GUI interface in Figure 21 should appear. The size of the GUI is not adaptive and may require some adjustments to fit the screen.
- Press the **START** button to begin data collection.

(infineon Radar demo	UART Enable
	Detection Threshold
4000	
3500	Min. Detection Velocity (Km/h)
3000	
2500	0.60
1500	Max. Detection Velocity (Km/h)
1000	20.00
500	
0	Doppler frequency (Hz):
Time(ms)	
100000	Target Speed (Km/h):
80000	
60000	So Target
40000	
2000	Target Approaching
	Target Departing
0	
Frequency (Hz)	



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





XENSIV[™] 24 GHz radar demo boards



3 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[™] Cortex[™]-M MCU embedded in Infineon's 24 GHz radar demo boards.

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 board.

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.

DAVE™	Cinfineon	Eclipse Launcher × Select a directory as workspace DAVE ^{tre} uses the workspace directory to store its preferences and development artifacts.
P	Component based programming with NAVE™ APPs	Workspace: CAWorkspacesADAVE124ghz Srowse
	Version 4.4.2	⑦ OK Cancel

Figure 23 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.



XENSIV[™] 24 GHz radar demo boards

3 Customizing radar applications

File	Edit Source Refactor Navigat	te Search Project	💗 Import —	
	New Open File	Alt+Shift+N >	Select	2
	Open Projects from File System		Select an import wizard:	
	Close	Ctrl+W	type filter text	
	Close All	Ctrl+Shift+W	> 🍅 General	
	Save	Ctrl+S	> 💪 C/C++	
	Save As		V 🗁 Infineon	
	Save All	Ctrl+Shift+S	Build Settings	
	Revert		DAVE Project	
	Move		> lostall	
ei.	Rename	F2	> 🗁 Plug-in Development	
	Refresh	F5	> 🗁 Run/Debug	
	Convert Line Delimiters To	>	> 🗁 Team > 🗁 XML	
-	Print	Ctrl+P		
	Switch Workspace	>		
	Restart			
<u>aka</u>	Import		② < Back Next > Finish Ca	ancel
No.	Export		V Vacci INext > Prinsii Ca	meet

Figure 24

DAVE™ project import dialog box

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

Import DAVE Projects		Search Import 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) < Back Next > Fin	ish Cancel	() < Back Next > Finish	Cancel

Figure 25 Import DAVE[™] projects

• The project file should appear under the **Project List**. Click on the **Finish** button. A screen similar to Figure 26 should appear.



XENSIV[™] 24 GHz radar demo boards

3 Customizing radar applications

File Edit Navigate Search Project D/	AVE Window Help	
🗑 🕼 🗡 🖉 🤌 🙆 🎋 📥 🖬	🗐 🕼 🚇 🕃 🏶 🐳 🚀 ▼ 🖗 👻 🖗 ▼ 🖗 ▼ 🖗 → ⇔ → 🔹 Quick Access	s 🖻 🖬 💽
1 C/C++ 以 Project □		- 0
 ⇔ ⇔ ⇔ ♀ P2G_FMCW > P2G_FW [Active - Build] 		
APP Dependency Tree 🙁 😐 🗖		
Search filter Clear	👔 APP Dependency 🙁 👔 HW Signal Connectivity 📮 Console 🔲 Properties 🖹 Problem:	s 🗆 🗖
ADC_MEASUREMENT_ADV_G ^		🏗 🔍 🗨 🕇
 GLOBAL_ADC_0 CLOCK_XMC4_0 ADC_MEASUREMENT_ADV_G GLOBAL_ADC_0 	INTERRUPT WATCHDOG TIMER DIGITAL_IO	
<pre>CLOCK_XMC4_0 CLOCK_XMC4_0 ADC_MEASUREMENT_ADV_G </pre>		,
		,

Figure 26 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 e.g., P2G_FW [Active Build] and then click on **Build Project**.

24ghz - DAVE CE - P2G_FW/main.c - DAVE [™] - C:\Workspace File Edit Source Refactor Navigate Search Project C:\Workspace	Image: Source of the second secon
> ▷ P2G_FW [Active - Build] 31 > ▷ P2G_FW [Active - Build] 32 > ▷ Includes 34 ^o int mi > ▷ Includes 35 > △ Application 36 > △ BSP 38 > △ Dave 40 > △ DSP_LIB 41 > △ Libraries 43 ^o / ⁱ > △ Startup 45	Index P2G_F Build Targets Build Configurations Build Project Build Project Build Project Clean Project Base Copy Dav Paste Dav Paste Dav Paste Delete Move Base Rename
> ≥ Store 47 } APP Dependency Tree □ 48 49⊕ /* 1 • • •	>>>> Sto Sto Export Image: APP Depen Set Active Project DAVE Project Upgrade >

Figure 27 Build project options



XENSIV[™] 24 GHz radar demo boards

3 Customizing radar applications

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

👔 APP Dependency 👔 HW Signal Connectivity 🕒 Console 🙁 🔲 Properties 🖹 Problems 📟	_
CDT Build Console [P2G_FW] ↓ ☆ � ↓ ☆ � ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ☆ ♥ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	*
'Invoking: ARM-GCC Create Listing' "c:/DAVE-IDE-4.4.2-64Bit/eclipse/ARM-GCC-49/bin/arm-none-eabi-objdump" -h -S "P2G_FW.elf" > "P2G_FW.lst" 'Finished building: P2G_FW.lst'	
10:17:47 Build Finished (took 1m:30s.213ms)	*

Figure 28 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 DEMO 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 29. Alternatively, press key F11 to directly start the debug session.

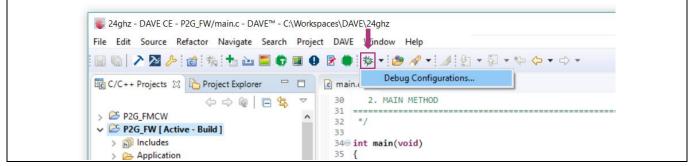


Figure 29 Start debug session toolbar button

If opening Debug Configurations, a window as shown in Figure 30 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.



XENSIV[™] 24 GHz radar demo boards

3 Customizing radar applications

Debug Configurations		×	Debug Configurations			
Create, manage, and run configurations		Ť.	Create, manage, and run configuratio	ons		
Image: Second Secon	Configure launch settings from this dialog: - Press the New button to create a configuration of the selected type. - Press the Duplicate' button to copy the selected configuration. - Press the 'Duplicate' button to comy the selected configuration. - Press the 'Duplicate' button to configure filtering options. - Fress the 'Diter' button to configure filtering options. - Grift or view an existing configuration by selecting it. Configure launch perspective settings from the ' <u>Perspectives'</u> preference page.		Image: Second secon	C/C++ Application: Build\P2G_FW.elf	Browse	
Filter matched 1 of 25 items			Filter matched 2 of 26 items	Revert	Apply	
?	Debug	Close	?	Debug	Close	

Figure 30 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 31.

📓 🕼 🎋 📲 🖉 🖌 🔰 💆 🔪 📄 💷 🛢 🙌 🕱 🔅 📴 🕁	9 N 6 9 • 6 • 6 0	▼ □ → ▼ Quick	Access
🎋 Debug 🛱 🖏 Servers 🛛 🛛 Resume (F8)	(x)= Variables 🔀 💁 Breakpo	1919 Registers 🔀 Periph	er 🖹 Modules 🗖 🛛
P2G_FW Build [GDB SEGGER J-Link Debugging]			🍐 🕫 🖻 🖻 🖻
✓ P2G_FW.elf ✓ P Thread #1 57005 (Suspended : Breakpoint)	Name	Туре	Value
main() at main.c:35 0x8029368	(×)= status	DAVE_STATUS_t	<optimized ou<="" td=""></optimized>
JLinkGDBServerCL.exe	<		
📕 arm-none-eabi-gdb 🚚 Semihosting and SWV	<		3
ie main.c 🕅	T.	Disassem.	🛛 🔠 Outline 🖵 I
30 2. MAIN METHOD		Disassem.	ation here
<pre>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. 44 * The while loop below can be replaced with an user of 44</pre>	error handler. */	 \u03c9 08029368: 39 08029368: 41 08029368: 41 08029368: 41 08029368: 42 08029376: 52 080293772: 55 08029376: 	<pre>status = DAVE] b1</pre>
45 YMC DERUG("DAVE APPs initialization failed\n").			
	0	🗋 🛃 📑	: 😫 📴 ▼ 🖓 🛛

Figure 31 DAVE[™] debug perspective

XENSIV[™] 24 GHz radar demo boards



3 Customizing radar applications

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 in mare 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			

Table 2Define 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. XENSIV[™] 24 GHz radar demo boards



4 Extracting radar raw data

4 Extracting radar raw data

The 24 GHz radar demo boards 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 proceeding 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

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.

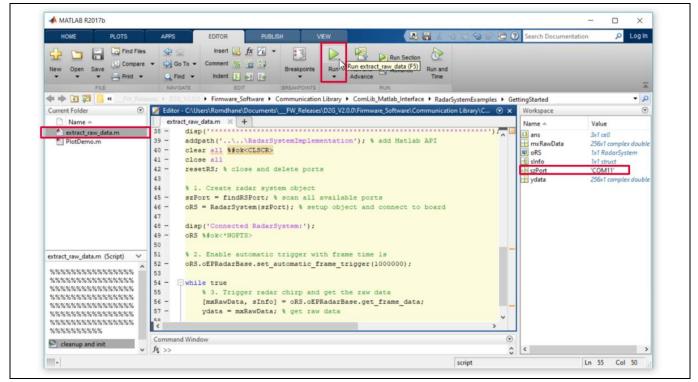
XENSIV[™] 24 GHz radar demo boards



4 Extracting radar raw data

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. •



MATLAB radar system interface Figure 32

Figure 33 shows the example running on MATLAB, extracting raw data from the DEMO DISTANCE2GO radar board.

Command Window		\odot
*****	*************	^
[resetRS] Reset the fo	llowing port(s):	
'COM3'		
'COM11'		
00111		
ans =		
Connected RadarSystem:		
· · · · · · · · · · · · · · · · · · ·		
ors =		
RadarSystem with prop	perties:	
oEPRadarADCXMC	: [1×1 EPRadarADCXMC]	
oEPCalibration	: [1×1 EPCalibration]	
	: [1×1 EPRadarBase]	
	[1×1 EPTargetDetection]	
	: [1×1 EPRadarFMCW]	
	[1×1 EPRadarIndustrial]	
	: [1×1 EPRadarDoppler]	
0.4906 + 0.49841		
0.4808 + 0.4806i		
0.4723 + 0.45981		
0.4664 + 0.44621		
0.4615 + 0.43131		
fx 0.4542 + 0.4117i		
		~



XENSIV[™] 24 GHz radar demo boards



4 Extracting radar raw data

4.3 C radar system interface

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 Radar Distance2Go, Radar Position2Go and Radar Sense2GoL Pulse and Radar Distance2GoL radar software packages.

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 34 shows the example running on Visual Studio, extracting raw data from a DEMO DISTANCE2GO 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 34 Raw data acquisition in Visual Studio shell

XENSIV[™] 24 GHz radar demo boards



4 Extracting radar raw data

4.4 UART radar system interface

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

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

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 DEMO SENSE2GOL)

Cate	PUTTY Configuration eggory: - Session - Logging - Terminal - Keyboard - Bell - Features - Window - Appearance - Behaviour	? × Basic options for your PuTTY session Specify the destination you want to connect to Serial lipe Speed COM53 128000 Connection type: Raw I elnet Rlogin SSH Serial Load, save or delete a stored session	
	 → Terminal → Keyboard → Bell → Features → Window → Appearance 	Serial line Speed COM53 128000 Connection type: Raw Raw Ielnet	

Figure 35 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 36.

XENSIV[™] 24 GHz radar demo boards

4 Extracting radar raw data

🛃 C	OM53 -	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	
			- Ira	aw sar	nples											
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	
			-Qra	aw sar	nples											
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	



Raw data acquisition in PuTTY shell



References



References

- [1] Infineon Technologies AG. AN543: DEMO DISTANCE2GO
- [2] Infineon Technologies AG. AN553: DEMO POSITION2GO
- [3] Infineon Technologies AG. AN597: DEMO SENSE2GOL
- [4] Infineon Technologies AG. AN598: DEMO SENSE2GOL PULSE
- [5] Infineon Technologies AG. AN615: DEMO DISTANCE2GOL



Revision history

Revision history

Document Date revision		Description of changes				
1.00	2019-06-14	Initial version				
1.10 2020-02-07 Added DEMO SENSE2GOL PULSE board features		Added DEMO SENSE2GOL PULSE board features				
1.20 2020-06-25 Added firmware flashing via Radar GUI details						
1.30 2021-03-31 Added DEMO DISTANCE2GOL board features		Added DEMO DISTANCE2GOL board features				
1.40	2023-02-14	3-02-14 Miscellaneous document cleanup updates				