

### Set-up guide

### About this document

#### Scope and purpose

This application note provides a step-by-step guide to setting up Infineon's 24 GHz Sense2GoL Pulse radar system platform, which is made up of the radar baseboard XMC4700 and BGT24LTR11 radar shield, making up the hardware and software for use with the Arduino platform. This guide will then go on to show an application example to demonstrate some of the features of this radar system platform.

#### **Intended audience**

This document is intended for anyone interested in using Infineon's 24 GHz radar system platform with the Arduino platform.

#### **Related documents**

Additional information can be found in the supplementary documentation provided with the Sense2GoL Pulse Kit in the Infineon Toolbox or from **www.infineon.com/24GHz**:

- 24 GHz Radar Tools and Development Environment User Manual
- Sense2GoL Pulse Software User Manual
- AN 598 BGT24LTR11 Shield (Pulsed Doppler)
- AN602 Radar Baseboard XMC4700



Table of contents

## **Table of contents**

Abou	t this document	1				
Table	۲able of contents					
List o	f figures	3				
List o	f tables	4				
1	Introduction	5				
2	Hardware modifications	7				
2.1	Removal of resistors	.7				
2.2	Selection of IOREF	.7				
2.3	ISCP header	.8				
3	Software set-up	9				
3.1	Download and install Arduino IDE	.9				
3.2	Install Infineon's Arduino package	.9				
3.3	Using Pulsed Doppler Radar Library	12				
3.4	Example sketches	16				
3.4.1	Example 1: Radar_Pulsed_Doppler_LED1	٢7				
3.4.2	Example 2: RadarPulsedDoppler_Andee_RGB	20				
4	Author	28				
5	References	29				
Revis	ion history	30				



List of figures

## List of figures

Figure 1	Radar baseboard XMC4700	5
Figure 2	BGT24LTR11 radar shield	6
Figure 3	Resistors to be removed for I <sup>2</sup> C communication	7
Figure 4	Jumper selection for IOREF	8
Figure 5	Use SPI pins on P3 instead of P7 (ISCP)	8
Figure 6	Downloading the Arduino IDE	9
Figure 7	Arduino IDE: File > Preferences	10
Figure 8	Entering the location from which the package can be downloaded	10
Figure 9	Installing the XMC package	11
Figure 10	Selecting XMC baseboard	11
Figure 11	Including and using the Library in Arduino sketch	12
Figure 12	Declaring Pulsed Doppler Radar object	12
Figure 13	Initializing hardware and parameters for Pulsed Doppler Radar	13
Figure 14	Example for callback function definition	15
Figure 15	Call run() in main loop()	16
Figure 16	Accessing example sketches from this library	17
Figure 17	Example sketch: Radar_Pulsed_Doppler_LED	18
Figure 18	Compiling the sketch	18
Figure 19	Compilation successful	18
Figure 20	Attach the BGT24LTR11 radar shield to the radar baseboard XMC4700	19
Figure 21	Connect USB cable to debug port	19
Figure 22	Uploading code onto board	20
Figure 23	Observe LED color change with motion	20
Figure 24	Example sketch: RadarPulsedDopplerLED_Andee_RGB	21
Figure 25	Compiling the sketch	21
Figure 26	Compilation successful	22
Figure 27	Attach the BGT24LTR11 shield to the radar baseboard XMC4700	22
Figure 28	Stacking the RGB LED lighting shield onto the radar baseboard XMC4700	23
Figure 29	Stacking the Annikken Andee U shield onto the set-up	23
Figure 30	Jumper wire connections on the Andee U	24
Figure 31	Connect the USB cable onto the debug port	24
Figure 32	Uploading the code onto the board	25
Figure 33	Application starts with white light	25
Figure 34	Andee app – scan for devices	25
Figure 35	Andee app – connect to device	26
Figure 36	Andee app – GUI upon successful connection	26
Figure 37	Changing pre-defined fast speed threshold	27



## List of tables

Table 1	Configuration APIs	13
Table 2	Result APIs	14
Table 3	Control APIs	16
Table 4	Example sketches	17
Table 5	LED color based on motion for Example 1	20
Table 6	LED behavior for Example 2	27

## 1 Introduction

The radar baseboard XMC4700, which is part of Infineon's 24 GHz radar system platform, is designed with headers compatible with Arduino (Figure 1) to ease prototyping works and integration with other Arduino shields in the market to form a compelling application.



Figure 1 Radar baseboard XMC4700

Currently the 24 GHz radar system platform also offers a BGT24LTR11 radar shield (Figure 2), which is suitable for various applications such as motion detection, presence sensing and proximity sensing.



For more details on the features of each board, please refer to their respective documents as mentioned on the cover page of this document.







2 Hardware modifications

This section highlights some of the changes required to be made to the hardware before interfacing the radar baseboard XMC4700 with other Arduino shields.

### 2.1 Removal of resistors

If I<sup>2</sup>C communication is intended to be used as the interface to the target Arduino shield, remove the following resistors (Figure 3): R70, R71, R73, R74, R77 and R80.



Figure 3 Resistors to be removed for I<sup>2</sup>C communication

### 2.2 Selection of IOREF

XMC4700 operates in the 3.3 V domain. Depending on the target Arduino shield, the IOREF can be configured as 5 V or 3.3 V via a jumper on P2 (Figure 4).

To select 3.3 V as IOREF, short pins 1 and 2 of P2.

To select 5 V as IOREF, short pins 2 and 3 of P2.



Hardware modifications



Figure 4 Jumper selection for IOREF

#### **ISCP** header 2.3

In case connection via the ISCP header is required to the target Arduino shield, do not use the pins from header P7. Instead, connect the pins via jumper wires from the SPI pins on header P3 (Figure 5).





Software set-up

### 3 Software set-up

This section highlights the steps needed in order to use Infineon's 24 GHz radar system platform with the Arduino IDE.

### 3.1 Download and install Arduino IDE

The first step is to download the latest Arduino IDE version from <u>https://www.arduino.cc/en/main/software</u> (Figure 6). Select the desired installation type and follow the installation instructions accordingly.



### Figure 6 Downloading the Arduino IDE

### 3.2 Install Infineon's Arduino package

- 1. Launch the Arduino IDE.
- 2. Navigate to *File > Preferences* (Figure 7).
- Copy the following URL into the "Additional Boards Manager URLs" field (Figure 8): <u>https://github.com/Infineon/Assets/releases/download/current/package\_infineon\_index.json</u> Once done, click "OK".
- 4. Navigate to *Tools > Board > Boards Manager* (Figure 9). Type "XMC" into the search field. Select the latest version from the drop-down box and click "Install".
- 5. Once installation is completed, the XMC-based microcontroller boards can be found and selected under *Tools* > *Board* (Figure 10).





### Software set-up

	base   Arduino :	1.8.1		
File	Edit Sketch	Tools Help		
	New	Ctrl+N		ø
	Open	Ctrl+O		_
	Open Recent		•	M
	Sketchbook		•	^
	Examples		re, to run once:	
	Close	Ctrl+W		
	Save	Ctrl+S		
	Save As	Ctrl+Shift+S	e, to run repeatedly:	
	Page Setup	Ctrl+Shift+P		
	Print	Ctrl+P		
Г	Preferences	Ctrl+Comma		
	0.4	041.0	-	
	Quit	Ctri+Q		
				•
			XMC11	100 XMC2GO on COM6



Arduino IDE: File > Preferences

Preferences			×
Settings Network			
Sketchbook location:			
Your Sketchbook			Browse
Editor language:	System Default	<ul> <li>(requires restart of )</li> </ul>	Arduino)
Editor font size:	12		
Interface scale:	Automatic 100 +% (requires restar	rt of Arduino)	
Show verbose output during	g: compilation upload		
Compiler warnings:	None 👻		
Display line numbers			
Enable Code Folding			
Verify code after upload	d		
Use external editor			
Check for updates on s	tartup		
Update sketch files to n	ew extension on save (.pde -> .ino)		
Save when verifying or	uploading		
Additional Boards Manager I	URLs: Here, please.		
More preferences can be ed	inted directly in the nie		
(edit only when Arduino is n	auno 15 (preterences. txt		
(care only mention additions in	croning)		
			OK Cancel





#### Software set-up

ype All	~ хмд					
Infineon's XMC Boards included XMC1100 Boot H <u>Online Help</u> <u>More Info</u>	Microcontroller by In I in this package: Kit, XMC1100 XMC2Go,	<b>fineon Technologies AG</b> XMC1100 H-Bridge 2Go,	XMC1300 Boot Kit, XMC	300 Sense2GoL, XMC4700 Rel	ax Kit.	^
					1.2.1 V Install	
						~

Figure 9

Installing the XMC package



Figure 10 Selecting XMC baseboard



Software set-up

### 3.3 Using Pulsed Doppler Radar Library

The Pulsed Doppler Radar Library is installed as part of the core libraries for XMC when the XMC Arduino package is installed as shown in Section 3.2.

1. To use the Pulsed Doppler Radar Library in Arduino sketch, it must first be added via *Sketch* > *Include Library* > *IFXRadarPulsedDoppler* (Figure 11).

😨 ske File Ed	ch_nov27a   Arduino 1.8.10 Sketch Tools Help	DigitalPressureSensor Firebase ESP32 Client I2S	
skete	Upload Using Programmer Ctri Export compiled Binary Ctri	IFX         IFX9201 H-Bridge           I+U         IFXRadarPulsedDoppler           I+Shift+U         Mouse           I+Alt+S         OneWire	
// } void // }	Add File	I+K RTC Radar SPI TLE94112 USB Ultrasonic Wire	
Library	idded to your libraries. Check "Include I	Recommended libraries Adafruit Circuit Playground ⊽ ibrary* menu XMC4700 Radar Baseboard o	n COM4

Figure 11 Including and using the Library in Arduino sketch

2. Next, declare a Radar Pulsed Doppler object, such as shown in Figure 12.

💿 sketch_nov27a   Arduino 1.8.10	_		×
<u>F</u> ile <u>E</u> dit <u>Sketch</u> <u>I</u> ools <u>H</u> elp			
			P
sketch_nov27a §			
<pre>#include <ifxradarpulseddoppler.h></ifxradarpulseddoppler.h></pre>			^
// IFX Radar Pulsed Doppler Object IFXRadarPulsedDoppler tjuefire;			
<pre>void setup() {     // put your setup code here, to run once:</pre>			
}			
<pre>void loop() {     // put your main code here, to run repeatedly:</pre>			
3			~
Library added to your libraries. Check "Include library" menu			
4 XMC470	0 Radar Bas	eboard on	COM4

Figure 12Declaring Pulsed Doppler Radar object



#### Software set-up

3. Next, some steps are required to initialize the hardware for the Pulsed Doppler Radar and also to configure the parameters. All these are to be done within the setup() routine (Figure 13).



Figure 13 Initializing hardware and parameters for Pulsed Doppler Radar

The lines of code will initialize the software to use the default parameters. In case custom parameters are preferred, the user can call the respective APIs (as listed in Table 1) before calling the begin() function.

Table 1	<b>Configuration APIs</b>
---------	---------------------------

API name	Description
<pre>uint8_t setMinSpeed(float speedMs);</pre>	Set the minimum speed that will be detected
<pre>float getMinSpeed(void);</pre>	Retrieve the minimum speed configured
<pre>uint8_t setMaxSpeed(float speedMs);</pre>	Set maximum speed that will be detected
<pre>float getMaxSpeed(void);</pre>	Retrieve the maximum speed configured
<pre>uint8_t setMotionSensitivity(float threshold);</pre>	Set threshold that will determine motion or not
<pre>float getMotionSensitivity(void);</pre>	Retrieve threshold that will determine motion or not
<pre>uint8_t setDopplerSensitivity(float threshold);</pre>	Set threshold that will determine motion with direction (departing/approaching) or not



#### Software set-up

API name	Description
<pre>float getDopplerSensitivity(void);</pre>	Retrieve threshold that will determine motion with direction (departing/approaching) or not
<pre>uint8_t setFramePeriod(uint8_t periodUs);</pre>	Set frame period in µs
<pre>uint8_t getFramePeriod(void);</pre>	Retrieve configured frame period in µs
<pre>uint8_t setSampleFreq(uint32_t frequencyHz);</pre>	Set ADC sampling frequency in Hz
<pre>uint32_t getSampleFreq(void);</pre>	Retrieve ADC sampling frequency in Hz
<pre>uint8_t setSkipSamples(uint32_t numSamples);</pre>	Set the number of samples to skip at beginning of frame
<pre>uint32_t getSkipSamples(void);</pre>	Retrieve the number of samples to skip at beginning of frame
<pre>uint8_t setNumSamples(uint32_t numSamples);</pre>	Set size of raw IQ ADC buffer
<pre>uint32_t getNumSamples(void);</pre>	Retrieve configured size of raw IQ ADC buffer
<pre>uint8_t setPulseWidth(uint32_t widthUs);</pre>	Set the pulse width in µs
<pre>uint32_t getPulseWidth(void);</pre>	Retrieve the configured pulse width in µs
<pre>uint32_t getMinFramePeriod(void);</pre>	Get the minimum frame period in µs

4. Next, define the callback function to perform application tasks upon completion of a round of radar processing, for example to turn on or off an LED. In the example shown in Figure 14, different LED colors are used to indicate different radar processing results. Table 2 lists the APIs that can be called to retrieve the result of the radar processing.

#### Table 2 Result APIs

API name	Description
<pre>bool targetAvailable(void);</pre>	Returns true: motion (with no direction), false: no motion
<pre>float getDopplerLevel(void);</pre>	Retrieve the Doppler level of the detected target
<pre>float getDopplerFreqHz(void);</pre>	Retrieve the Doppler frequency of the detected target
<pre>float getVelocity(void);</pre>	Retrieve the signed velocity value
<pre>uint8_t getDirection(void);</pre>	Returns 0: no direction, 1: departing, 2: approaching
<pre>float getSpeed(void);</pre>	Retrieve the unsigned speed value



#### Software set-up



Figure 14 Example for callback function definition

5. Finally, add the run() API in the loop() routine to run the radar processing (Figure 15).



#### Software set-up

💿 sketch_nov27a   Arduino 1.8.10	_		×
Eile Edit Sketch Tools Help			
			ø
sketch_nov27a §			
<pre>void setup() {</pre>			^
// put your setup code here, to run once:			
// initialize HW for Pulsed Doppler Radar			
<pre>tjuefire.initHW();</pre>			
<pre>// register callback function to perform tasks upon // completion of radar processing</pre>			
<pre>tivefire.registerResultCallback(myResultCallback):</pre>			
<pre>// configure parameters for pulsed doppler radar</pre>			
<pre>tjuefire.begin();</pre>			
3			
void loop() {			
// put your main code here, to fun repeatedry:			
tjuefire.run();			
}			~
Library added to your libraries. Check "Include library" menu			
55	VMC4700 Padar Bas	abaard on Ci	0111

#### Figure 15 Call run() in main loop()

Table 3 lists the available control APIs for the Pulsed Doppler Radar Library.

#### Table 3Control APIs

API name	Description
<pre>void initHW(void);</pre>	To initialize hardware for Pulsed Doppler Radar
<pre>void registerResultCallback(void(*callBackPtr));</pre>	To register function to be called when radar process is done
<pre>void registerErrorCallback(void(*callBackPtr));</pre>	To register function to be called in case of error
<pre>void begin(void);</pre>	To initialize software parameters for Pulsed Doppler Radar
<pre>void end(void);</pre>	To stop radar processing; usually called during runtime before reconfiguring radar parameters
void run(void);	To run radar processing

### 3.4 Example sketches

Two Arduino sketch examples are provided in the current library release. These examples can be accessed from *File > Examples > IFXRadarPulsedDoppler* (Figure 16). Table 4 provides a brief description of these examples.

### Software set-up

File	Edit Sketch New Open Open Recent Sketchbook	Tools Help Ctrl+N Ctrl+O	Mouse OneWire Radar RTC SPI	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
	Examples Close	> Ctrl+W	Ultrasonic USB Wire	> > •
	Save As	Ctrl+Shift+S	Examples from Custom Libraries	P1
	Page Setup Print	Ctrl+Shift+P Ctrl+P	DigitalPressureSensor DPS310	
	Preferences	Ctrl+Comma	DPS422 Barometric Pressure Sensor	
	Quit	Ctrl+Q	IFXRadarPulsedDoppler TLE94112	<ul> <li>Radar_Pulsed_Doppler_LED</li> <li>RadarPulsedDoppler_Andee_RGB</li> </ul>
Lib	rary added to y	our libraries. Cheo	sk "Include library" menu	



Sketch name	Required HW	Description	
Radar_Pulsed_Doppler _LED	Radar baseboard XMC4700 and BGT24LTR11 radar shield	Use on-board LED to indicate detection of motion and direction of motion	
RadarPulsedDoppler _Andee_RGB	<ol> <li>Radar baseboard XMC4700 and BGT24LTR11 radar shield</li> <li>Annikken Andee U shield (<u>https://www.annikken.com/andee-u</u>)</li> <li>RGB LED lighting shield with XMC1202 (<u>https://www.infineon.com/cms/en/product/evaluation-boards/kit_led_xmc1202_as_01/</u>)</li> </ol>	Project and display the results of radar processing on a smart device (e.g. cell phone/tablet) via Bluetooth, and at the same time using an external RGB LED to indicate motion and direction of motion	

#### Table 4Example sketches

## 3.4.1 Example 1: Radar\_Pulsed\_Doppler\_LED

This section lists the steps to follow to get this example up and running.

1. In Arduino IDE, navigate to *File > Examples > IFXRadarPulsedDoppler > Radar\_Pulsed\_Doppler\_LED* (as shown previously in Figure 16). The sketch will open (Figure 17).





#### Software set-up



Figure 17 Example sketch: Radar\_Pulsed\_Doppler\_LED

- 2. Navigate to *Tools > Board > XMC4700 Radar Baseboard* (as shown previously in Figure 10) to select the radar baseboard as the target kit.
- 3. Compile the sketch by clicking the "Verify" button (Figure 18). A success message will be displayed on compilation completion (Figure 19).







Figure 19 Compilation successful





#### Software set-up

4. Attach the BGT24LTR11 radar shield to the radar baseboard XMC4700 via the SAMTEC connectors (Figure 20).



Figure 20 Attach the BGT24LTR11 radar shield to the radar baseboard XMC4700

5. Connect the radar baseboard XMC4700 to the PC via a USB cable onto the "Debug" USB port (Figure 21).



Figure 21 Connect USB cable to debug port



Software set-up

6. Upload the code onto the board by clicking the "Upload" button (Figure 22).



#### Figure 22 Uploading code onto board

The application can now be tested. Make some movements in front of the radar board and observe the LED colors depending on the motion and its direction (Figure 23).

Table 5	LED color based on motion for Example 1		
LED color	Ту	/pe of motion	
Off	No	o motion	
Red	De	eparting	
Green	Ap	oproaching	
Blue	M	otion with no meaningful direction	



Figure 23 Observe LED color change with motion

### 3.4.2 Example 2: RadarPulsedDoppler\_Andee\_RGB

This section lists the steps to follow to get this example up and running.



#### Software set-up

1. In Arduino IDE, navigate to *File > Examples > IFXRadarPulsedDoppler > RadarPulsedDoppler\_Andee\_RGB*. The sketch will open (Figure 24).



Figure 24 Example sketch: RadarPulsedDopplerLED\_Andee\_RGB

- 2. Navigate to *Tools > Board > XMC4700 Radar Baseboard* (as shown previously in Figure 10) to select the radar baseboard as the target kit.
- 3. Compile the sketch by clicking the "Verify" button (Figure 25). A success message will be displayed on compilation completion (Figure 26).

<u>File E</u> dit <u>S</u> ketch <u>T</u> ools <u>H</u> elp
✓ → III ▲ Verify





#### Software set-up

Done compiling.	
Sketch uses 72896 bytes (3%) of program storage space. Maximum is 20480	^
Global variables use 28792 bytes of dynamic memory.	-
< >	
1 XMC4700 Radar Baseboard on COM4	

Figure 26 Compilation successful

4. Attach the BGT24LTR11 radar shield to the radar baseboard XMC4700 via the SAMTEC connectors (Figure 27).



Figure 27 Attach the BGT24LTR11 shield to the radar baseboard XMC4700

 Stack the RGB LED lighting shield with XMC1202 onto the radar baseboard XMC4700 via the Arduino stack headers (Figure 28). Also connect an RGB LED and a 24 V DC power adapter to the RGB LED lighting shield.
 Do not turn the power on yet! For more information on setting up of the RGB LED lighting shield, please refer to the User Manual for the RGB LED Lighting Shield with XMC1202 (the link can be found in the References section).

Software set-up



Figure 28 Stacking the RGB LED lighting shield onto the radar baseboard XMC4700

6. Stack the Annikken Andee U shield onto the set-up (Figure 29). Notice that there are several jumper wires on the Annikken Andee U board. This is due to the hardware modifications required on the radar baseboard regarding the ISCP header as mentioned in Section 2.3. Figure 30 illustrates the required connections.









### Software set-up



Figure 30 Jumper wire connections on the Andee U

7. Turn on the 24 V DC power supply to the RGB LED lighting shield and connect the radar baseboard XMC4700 to the PC via a USB cable onto the "Debug" USB port (Figure 31).



Figure 31 Connect the USB cable onto the debug port

8. Upload the code onto the board by clicking the "Upload" button (Figure 32).



#### Software set-up

<u>File Edit Sketch Iools H</u> elp	
👽 📀 🗈 🔛 Upload 🛛 😰	

Figure 32 Uploading the code onto the board

The RGB LED should turn on with white light, while the on-board LED will cycle between red, green and blue light to show radar processing is taking place (Figure 33). The application can now be tested.



Figure 33 Application starts with white light

9. Launch the Andee app (can be installed for free from the Apple App Store or Google Play Store) on your smart device. In the app, click "Scan for Devices" (Figure 34).



Figure 34 Andee app – scan for devices

10. Click "Tjuefire". When prompted, click "Connect" (Figure 35). 25 of 31

Software set-up

10:37 Andee Andee	<b></b>
Tjuefire	atl
Connect t Tjuefire	
Cancel	nnect
$\otimes$	

Figure 35 Andee app – connect to device

11. Upon successful connection, the GUI should appear on the app (Figure 36). In case this does not happen, disconnect from the app, press the reset button on the radar baseboard XMC4700 and retry the connection. Make some movement in front of the radar board and observe the measured speed and detected direction on the GUI. You may also observe the light from the RGB LED changing with regard to the motion and its direction (Table 6).

10:41 ර්දා Tjur	ııli <b>Ə ■</b> ) efire ıl <mark> </mark> :
Radar for	r Arduino
Speed 0.000 m/s	Direction No motion
Brightness 100.00 %	<sup>Colour</sup> White
<i>\subset</i>	

Figure 36 Andee app – GUI upon successful connection





Software set-up

Table 6 LE	) behavior	for Example 2
------------	------------	---------------

LED behavior	Type of motion
Dims down	Departing
Brightens	Approaching
Changes color	Motion with speed above pre-defined threshold

The pre-defined fast speed threshold can be changed within the Arduino sketch (Figure 37).

$\odot$	RadarPulsedDoppler_Andee_RGB   Arduino	1.8.10	_		×
<u>F</u> ile	<u>E</u> dit <u>S</u> ketch <u>T</u> ools <u>H</u> elp				
					<b>P</b>
R	adarPulsedDoppler_Andee_RGB			i	-
#in #in // // #in #in	clude <ifxradarpulseddoppler.h> clude <wire.h> // for I2C commur Always include these libraries. to work with the Arduino! clude <spi.h> clude <andee.h></andee.h></spi.h></wire.h></ifxradarpulseddoppler.h>	<pre>// for Radar Pulsed Do hication with RGB LED L Annikken Andee needs t</pre>	ppler ighting hem	Shield	1
#de	<pre>#define FAST_SPEED 3 // 3m/s</pre>				
// #de	Defines for RGB LED Lighting Shi fine ADDRESS	.eid 0x15EUL			
#de #de	fine INTENSITY_RED fine INTENSITY_GREEN	0×11U 0×12U			

Figure 37 Changing pre-defined fast speed threshold



### 4 Author

Mohamed Saat Muhammad Nur Syafii



References

### 5 References

- [1] Infineon BGT24LTR11N16 24 GHz Radar IC Datasheet
- [2] Infineon BGT24LTR11 Product Brief
- [3] Infineon Application Note <u>AN472 "User's Guide to BGT24LTR11N16"</u>
- [4] 24 GHz industrial radar FAQs
- [5] ETSI Regulations <u>EN 300 440 V2.2.1</u>
- [6] FCC Regulations <u>15.245</u>, <u>15.249</u>
- [7] Infineon RGB LED Lighting Shield with XMC1202 for Arduino User Manual
- [8] Annikken Andee U Documentation



## **Revision history**

Document version	Date of release	Description of changes
V1.0	02-07-2020	First release

#### Trademarks

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

Edition 2019-12-02

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 AN60511\_PL32\_1911\_101730

#### IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of noninfringement 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 contair dangerous substances. For information on the types in question please contact your nearest Infineor Technologies office.

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