Static distance measurement application

using XENSIV™ KIT_CSK_BGT60UTR11AIP 60 GHz radar

About this document

Scope and purpose

This application note demonstrates how to set up ModusToolbox™ application for static distance measurement using the XENSIV™ BGT60UTR11AIP 60 GHz FMCW radar sensor and its use cases as well as how to set up and get started.

Intended audience

The intended audience for this document is design engineers, technicians, and developers of electronic systems, working with Infineon's XENSIV™ 60 GHz radar sensors.

Keypoints

- Demonstrates how to set up ModusToolbox™ application for static distance measurement using the XENSIV™ BGT60UTR11AIP 60 GHz FMCW radar sensor
- Provides information on the technical highlights of the 60 GHz radar sensors for IoT, including high precision and low power consumption

About this product family

Product family

Infineon's XENSIV™ family of sensors offer exceptional accuracy and robust implementation making them suitable for critical functions such as safety and efficiency and enhancement mechanisms. This family of sensors enables implementation in applications such as automotive, industrial, and IoT devices.

Target applications

- Smart TV
- Security camera and video doorbell
- Heating ventilation and air conditioning (HVAC)
- Smart thermostat

Table of contents

About this document **1**
About this product family **1**
Table of contents **2**
1 Introduction **3**
1.1 FMCW radar systems 3
1.2 Distance measurement in FMCW radar systems 4
2 Hardware and software requirements **5**
2.1 Hardware requirements 5
2.2 Software requirements 6
3 Getting started **7**
3.1 Creating static distance measurement application 7
3.1.1 Build application 9
3.1.2 Program application 9
4 Parameters configuration **11**
4.1 Configuration parameters 11
4.1.1 Zero padding factor 11
4.1.2 Maximum range 11
4.1.3 Minimum range 12
4.1.4 Example configuration 12
5 Design and implementation **13**
5.1 Data structures 13
5.2 Global variables 13
5.3 Functions 14
6 Related resources **15**
References **16**
Revision history **17**
Disclaimer **18**

1 Introduction

1.1 FMCW radar systems

“RADAR”, which is an acronym for Radio Detection and Ranging, is a device that uses electromagnetic waves to obtain the range, angle, or velocity of targets. Radar systems are used in a variety of applications like presence sensing, people tracking and localization, velocity estimation, range estimation, vital signs tracking and more.

Modulated Radar, which is also known as frequency modulated continuous wave (FMCW) radar, is a type of radar that can measure the range of the target using a technique known as Frequency Modulation. In Frequency Modulation, the frequency of the electromagnetic wave is linearly increased with time. This kind of signal whose frequency increases linearly with time is called a chirp. The FMCW system measures the instantaneous difference between the transmitted and reflected signal frequencies Δf which is referred to as beat frequency. This beat frequency is directly proportional to the time difference Δt of the reflected chirp. This captured time difference can be used to calculate the range of the target.

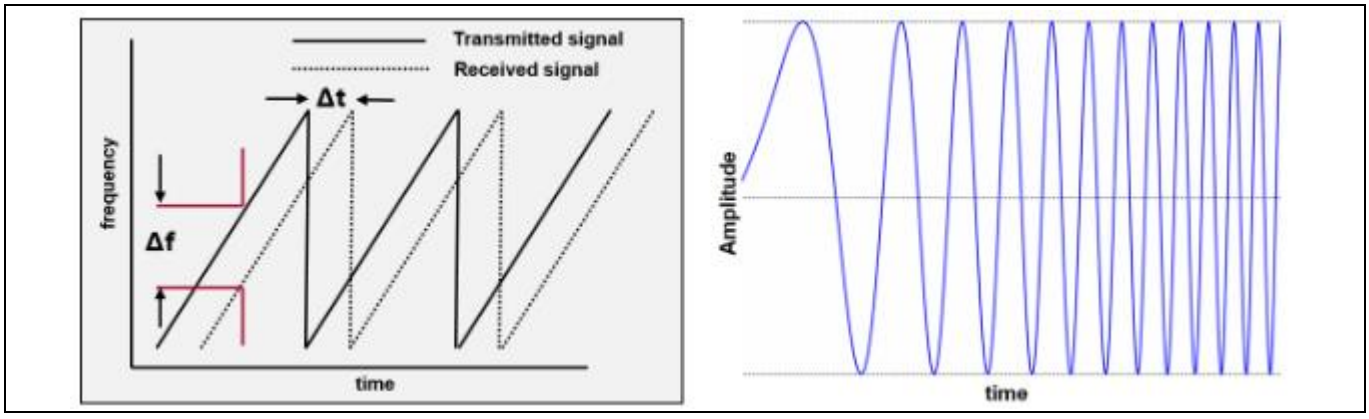


Figure 1 Chirp signal representation

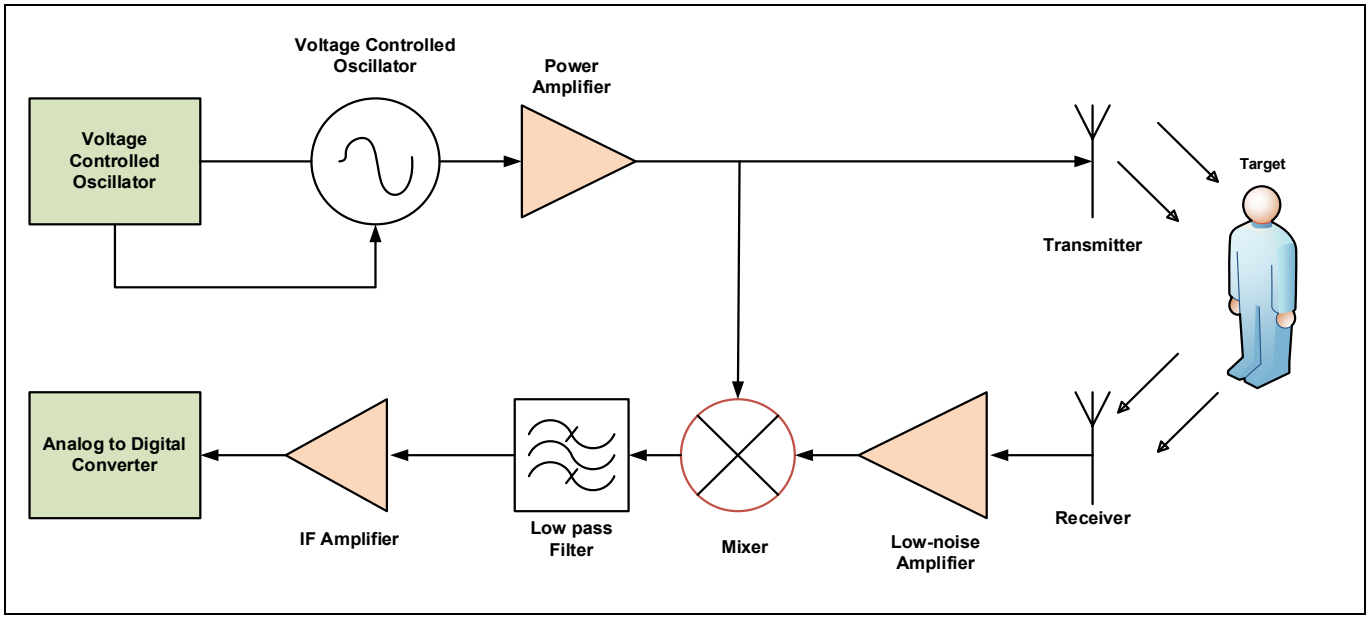


Figure 2 FMCW block diagram

1.2 Distance measurement in FMCW radar systems

When a target reflects the radar signal back, the reflected signal will be shifted in frequency which is directly proportional to the distance of the target distance in front of the radar. To process this signal to distance data, the signal must be analyzed to find its frequency components using Fourier transform.

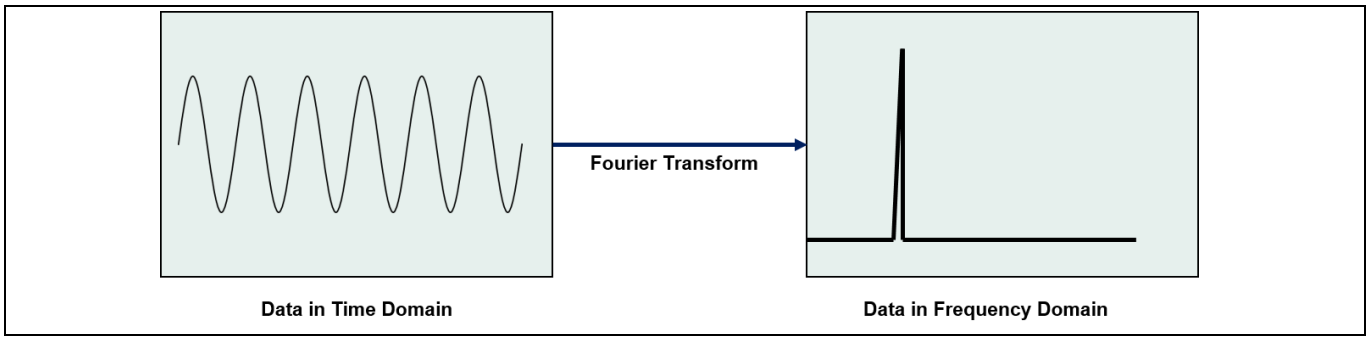


Figure 3 FMCW block diagram

Applying a Fast Fourier Transform (FFT) on the beat frequency results in a plot of magnitude spectrum across frequency aka range values and this is called “Range FFT” plot. The target can be located on the Range FFT plot by looking at the peaks in the plot. In the Range FFT, each frequency bin corresponds to a certain distance which is equal to the range resolution of the radar calculated as $Range_{resolution} = \frac{c}{2B}$ and multiplying this range resolution with frequency bin of the FFT peak will return the exact distance of the target from radar.

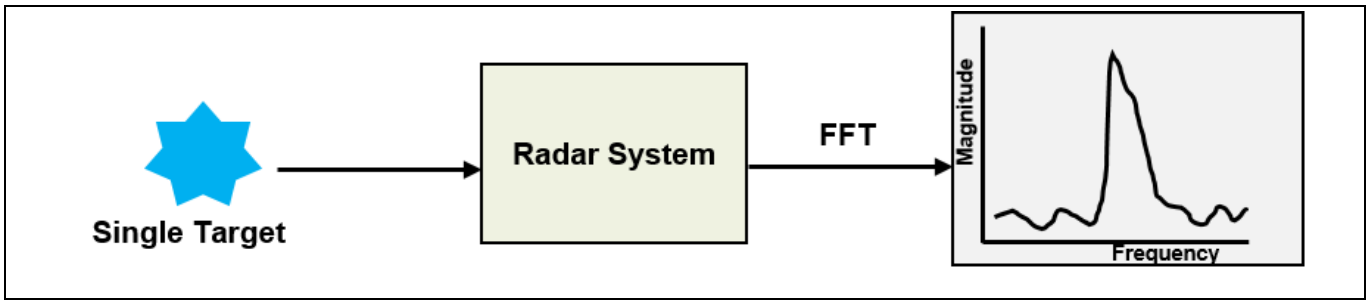


Figure 4 Concept of range resolution

2 Hardware and software requirements

2.1 Hardware requirements

The static distance measurement application is implemented and tested on the XENSIV™ BGT60UTR11AIP Connected Sensor Kit (Figure 5). However, this code can also be migrated to other kits and MCUs of interest without much effort, as the code is modular, with radar signal processing logic and hardware interfacing parts in separate files.

The XENSIV™ BGT60UTR11AIP Connected Sensor Kit supports customers in testing sensor-driven IoT products and radar use cases as well as in prototyping. It offers a real-time sensor evaluation with custom configurations and cloud- and radar-based solution output visualization. The KIT_CSK_BGT60UTR11AIP (Figure 5) comes with:

- Rapid IoT Connect Developer Kit (CYSBSYSKIT-DEV-01)
- XENSIV™ BGT60UTR11AIP Wing (EVAL_60UTR11_WING)

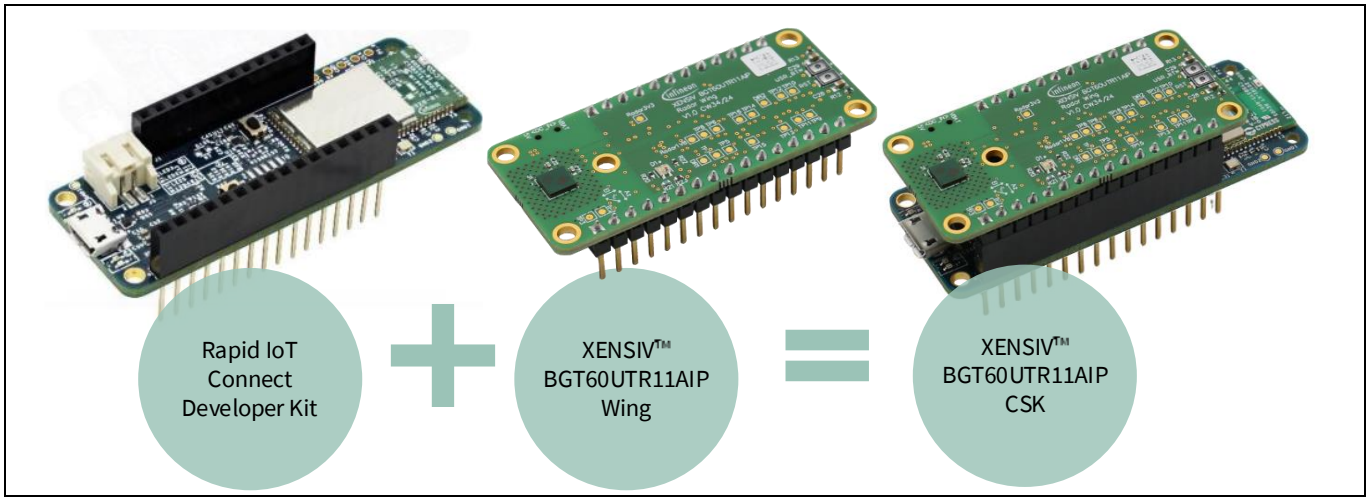


Figure 5 XENSIV™ BGT60UTR11AIP Connected Sensor Kit

The Rapid IoT Connect Developer Kit (CYSBSYSKIT-DEV-01) shown in Figure 6 allows the evaluation of the Rapid IoT Connect module (CYSBSYS-RP01) on a standard Feather form-factor. The CYSBSYS-RP01 Rapid IoT Connect module is a turnkey module that enables secure, scalable, and reliable compute and connect.

The Rapid IoT Connect Developer Kit carries a CYSBSYS-RP01 Rapid IoT connect system-on-module (SoM), which includes a PSOC™ 6 MCU, an AIROC™ CYW43012 single-chip radio, onboard crystals, oscillators, chip antenna, and passive components.

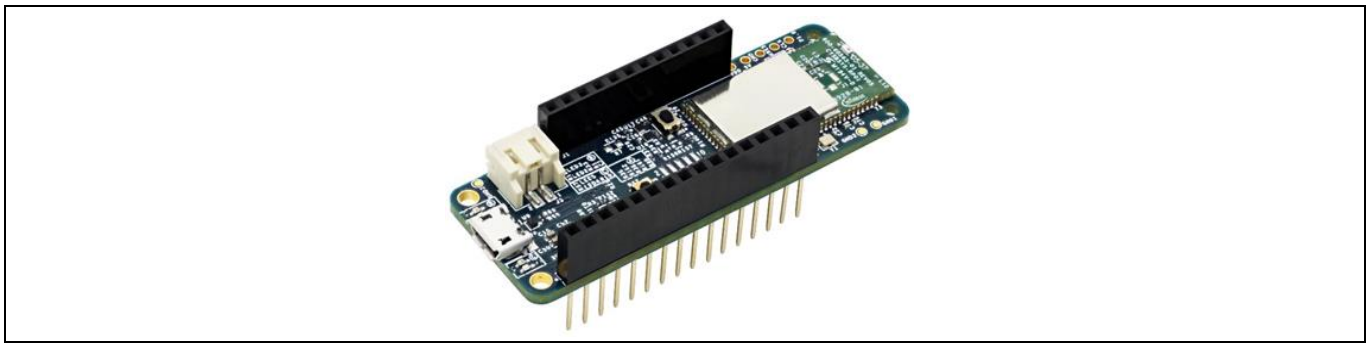


Figure 6 Rapid IoT Connect Developer Kit

The XENSIV™ BGT60UTR11AIP Wing board shown in Figure 7 is based on the XENSIV™ BGT60UTR11AIP 60 GHz radar sensor MMIC with one transmitting and one receiving antenna integrated. The XENSIV™ BGT60UTR11AIP MMIC enables ultra-wide bandwidth FMCW operation. It is equipped with an integrated finite state machine (FSM). With the aid of the FSM, the XENSIV™ BGT60UTR11AIP can perform frequency modulated continuous wave (FMCW) frequency sweeps (chirps), acquire data, and store samples into the internal first-in, first-out (FIFO) memory autonomously.

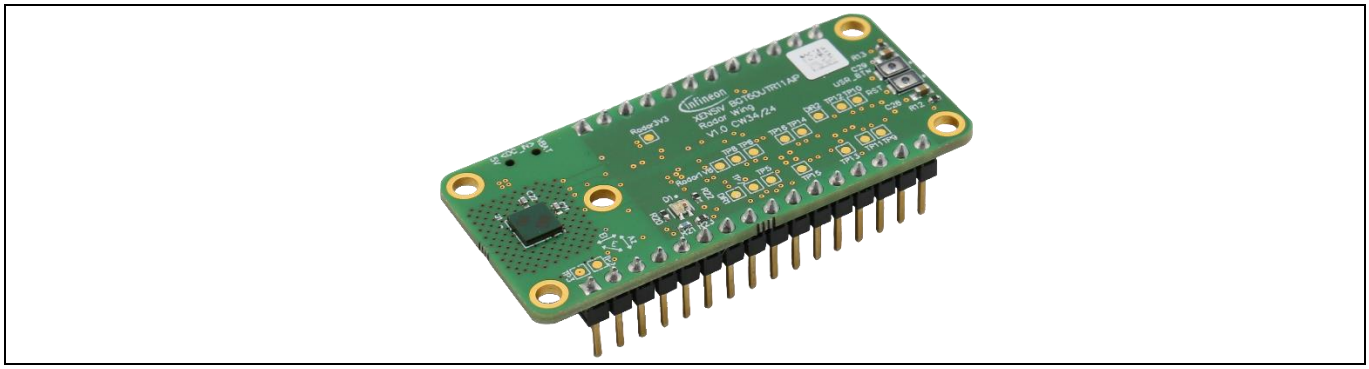


Figure 7 XENSIV™ BGT60UTR11AIP Wing board

2.2 Software requirements

1. ModusToolbox™
2. Board support package
3. C language
4. Serial Terminal (Tera Term, ModusToolbox™ Terminal, etc.

3 Getting started

This section provides a walkthrough to create a static distance measurement application using the Eclipse IDE for ModusToolbox™, selecting a BSP. It also covers how to build and program using the IDE and to see the output in the serial terminal.

3.1 Creating static distance measurement application

To launch the Eclipse IDE:

1. On Windows, select **Eclipse IDE for ModusToolbox™** from the **Start** menu
2. For other operating systems, run the "modustoolbox" executable file

Note: When launching the Eclipse IDE, it provides an option to select the workspace location on your machine. This location is used by the IDE for creating and storing the files as part of application creation for a particular platform. The default workspace location is a folder called "mtw" in your home directory. You may add additional folders under the "mtw" folder or choose any other location for each workspace.

3. Click the **New Application** link in the Eclipse IDE **Quick Panel** as shown in Figure 8

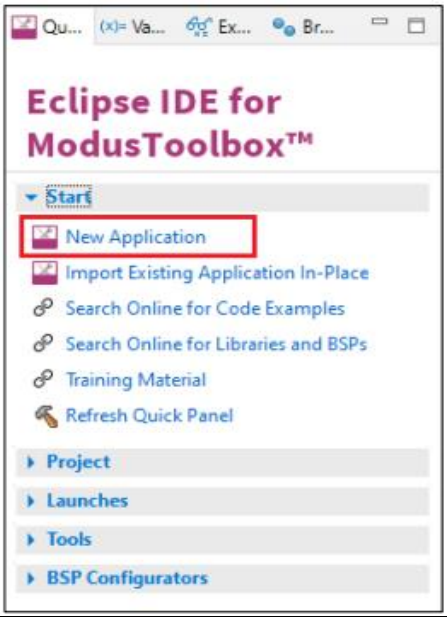


Figure 8 New application creation

This launches the **Project Creator** tool to select the board support package (BSP) and the several applications available for a particular board.

4. Select **CYSBSYSKIT-DEV-01** BSP from the **PSOC™ 6 BSPs** drop-down as shown in Figure 9 and click **Next**

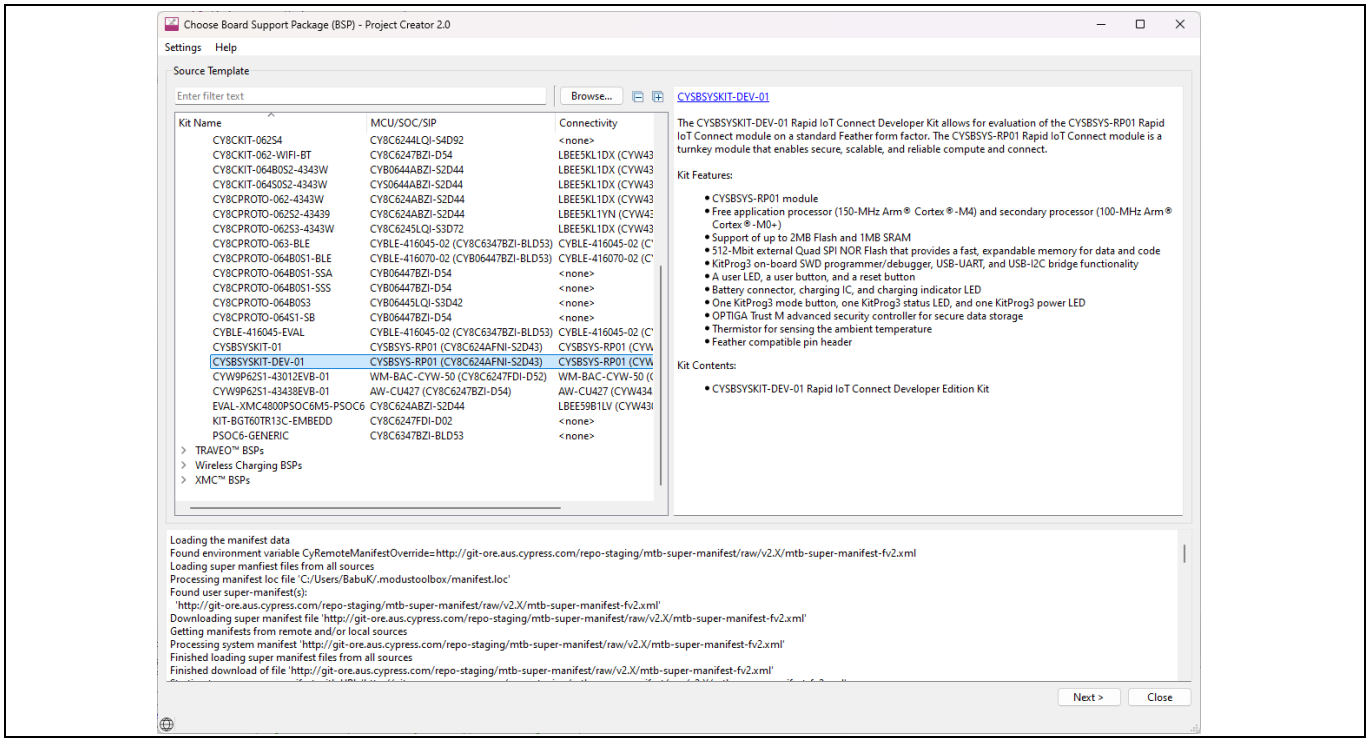


Figure 9 Selecting the BSP

5. Select the **XENSIV 60GHz Radar Static Distance Measurement** application from the **Sensing** drop-down as shown in Figure 10 and click **Create**

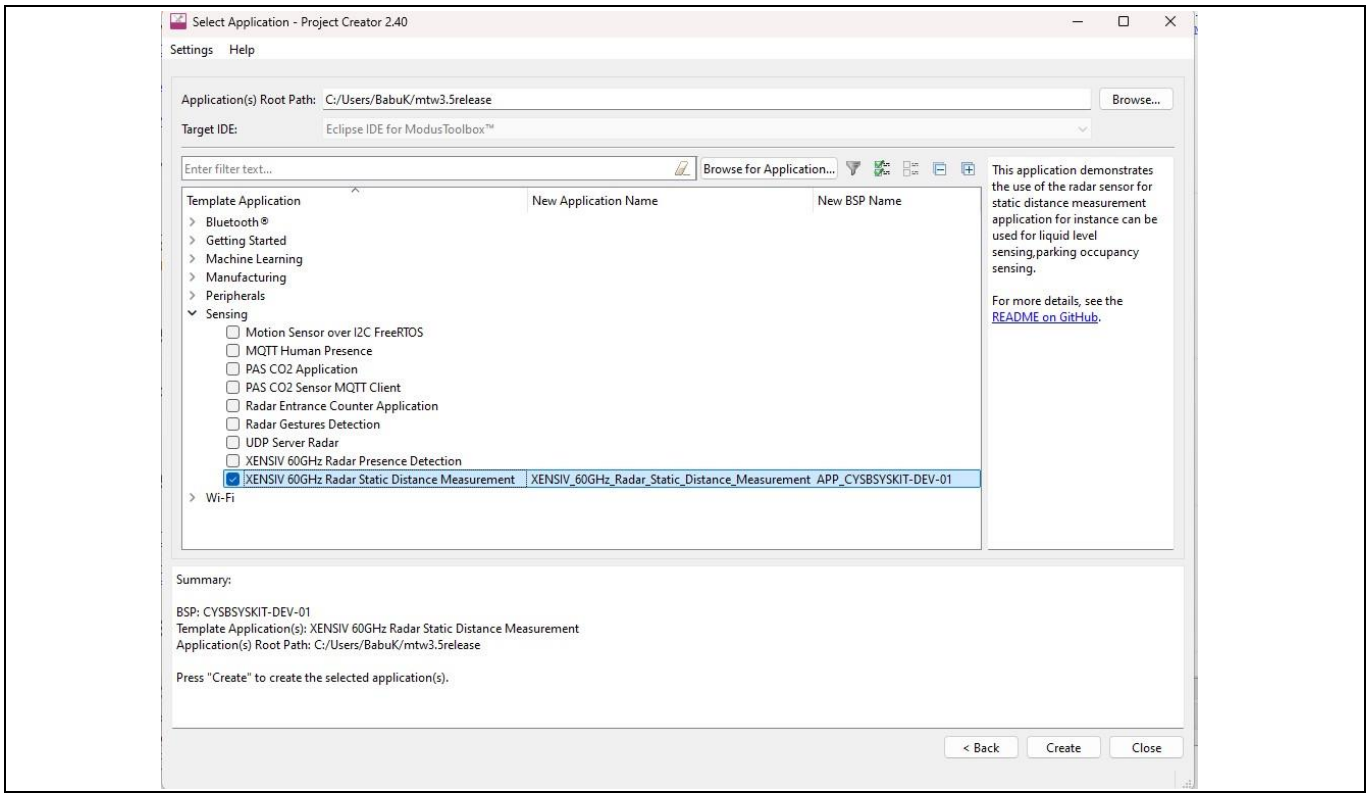


Figure 10 Selecting the application

When complete, the Project Creator tool closes automatically. In the IDE, a message displays about importing the project.

3.1.1 Build application

After loading the application, build it to generate the necessary files.

1. Select a project
2. In the **Quick Panel**, click the **Build Project** button shown in Figure 11

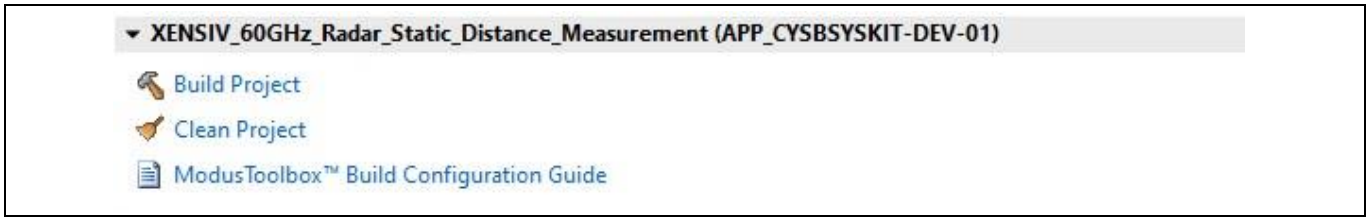


Figure 11 Build application

Messages display in the console, indicating whether the build was successful or not.

Note: In the resource_map.h file, define the radar sensor being used, i.e., BGT60UTR11AIP or BGT60TR13C. By default, the BGT60UTR11AIP is selected as the connected radar sensor, and the correct algorithm configuration are defined.

3.1.2 Program application

1. In **Project Explorer**, select the desired project. Then, in the **Quick Panel**, click the **Program** (KitProg3_MiniProg4) link as shown in Figure 12

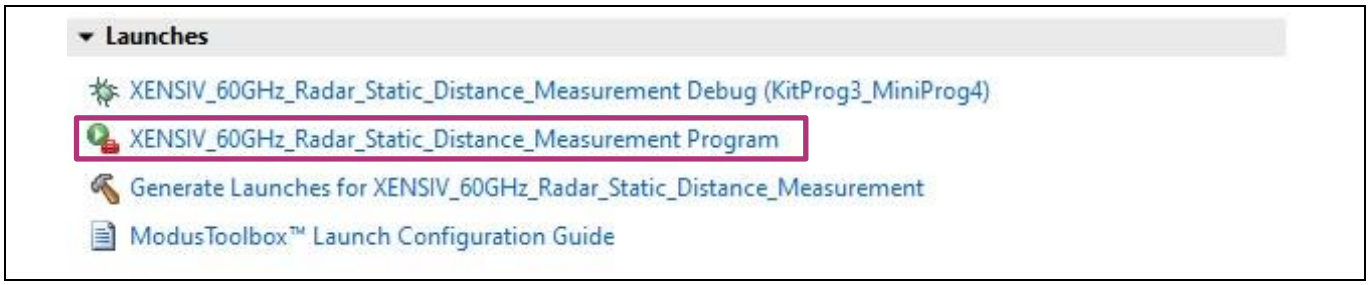


Figure 12 Program application

2. Open the ModusToolbox™ terminal from the toolbar (Ctrl + Alt + Shift + T) and select a COM port where the board is connected (not the MiniProg4 port). Set the serial port parameters to 8N1 and 115200 baud, as shown in Figure 13
   After programming, the application starts automatically.

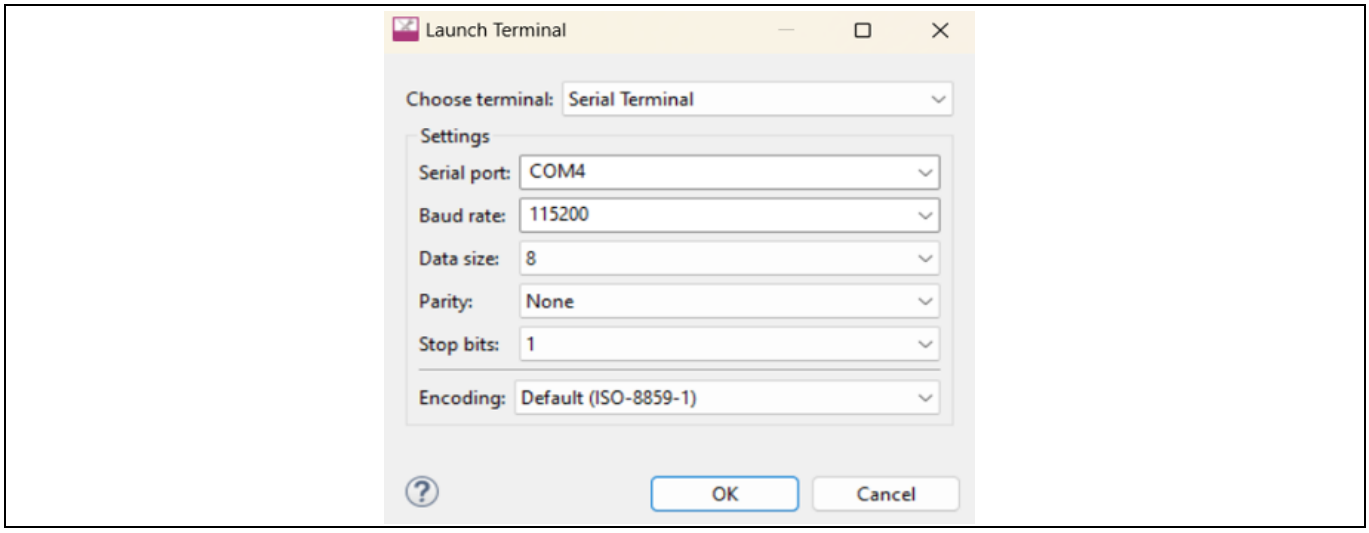


Figure 13 Serial terminal settings

3. Confirm that "Static Distance Measurement using Infineon XENSIV 60GHz radar" is displayed on the ModusToolbox™ terminal (Figure 14)

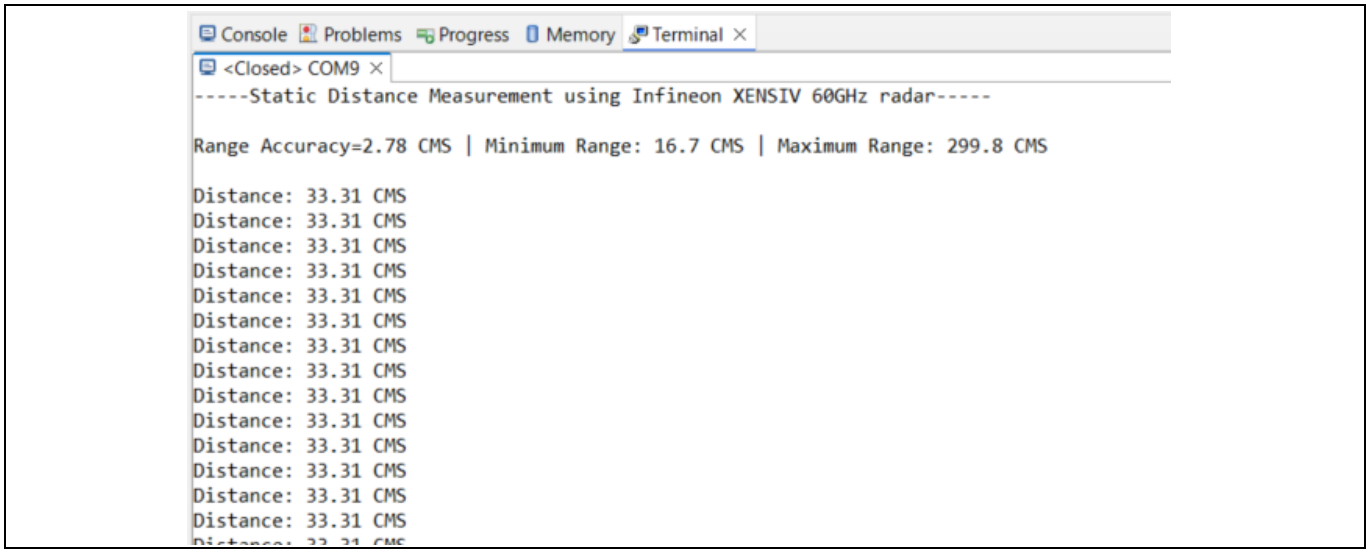


Figure 14 Output in ModusToolbox™ terminal

4 Parameters configuration

This section explains how to configure the radar distance measurement algorithm by adjusting key parameters. The parameters control the zero-padding factor and the measurable minimum and maximum distances.

4.1 Configuration parameters

Table 1 lists the parameters that can be modified to configure the radar algorithm.

Table 1 Radar algorithm configuration

Parameter	Type	Description	Valid values	Default value
max_range_m	float32_t	Maximum measurable distance by the radar in meters	Up to 5 m	3 m
min_range_m	float32_t	Minimum measurable distance by the radar in meters	0.15 m without lens 0.05 m with lens	0.15 m
zeroPadding_factor	int16_t	Zero padding factor for signal processing. Higher values improve resolution but increase computation.	1, 2, 4, or 8	1

4.1.1 Zero padding factor

Zero padding (*zeroPadding_factor*) is a signal processing technique where extra zeros are added to the end of a signal (in this case, radar data) before performing a FFT. Zero padding adds extra zeros to the radar data before processing, improving range resolution by creating finer frequency bins without altering the radar's sampling rate. However, it does not add new information to the signal and increases computational complexity, but this helps in delivering a better range resolution.

- Set to '1', '2', '4', or '8' based on your requirements. For moderate resolution, set *zeroPadding_factor* to 2, to achieve a range resolution of 1.34 cm. This parameter also affects the minimum measurable range and measurement stability
- A higher value (e.g., '4','8') provides better resolution but requires more processing power and affects the signal-to-noise ratio (SNR). This means that the reflected signal from the target must be stronger to be detected and might require a radome lens in most cases to reliably measure distance
- For the BGT60UTR11AIP radar at full bandwidth (sweeping frequency), the range resolution varies with the *zeroPadding_factor* as follows:

Table 2 Zero padding factors

zeroPadding_factor	Range resoltuion
1	2.68 cm
2	1.34 cm
4	0.67 cm
8	0.33 cm

4.1.2 Maximum range

- *max_range_m* defines the maximum distance, in meters, that the radar can measure (up to 5 m)
- Set a value in meters, e.g., *max_range_m* = 4.5f for a 4.5-meter range

- Ensure that max_range_m is greater than min_range_m
- The performance and reliability of the algorithm beyond 5 m is not guaranteed

4.1.3 Minimum range

- *min_range_m* defines the minimum distance, in meters, that the radar can measure (down 0.05 m or 5 cm)
- Set a value in meters, e.g., *min_range_m* = 0.15f for a 15-centimeter minimum
- Ensure that min_range_m is less than max_range_m
- Ensure that min_range_m is enough to overcome the peak at 0 Hz, otherwise algorithm will return to the minimum distance as the measured distance

4.1.4 Example configuration

To configure the radar to measure distances between 10 cm and 4 m with a moderate resolution of 0.67 cm, use the configuration below:

```
int16_t zeroPadding_factor = 4;       // Moderate resolution (0.67 cm)
float32_t max_range_m = 4.0f;         // Maximum distance of 4 m
float32_t min_range_m = 0.1f;         // Minimum distance of 10 cm
```

Figure 15 Configuration example

Notes

1. *Ensure that min_range_m is always smaller than max_range_m to avoid invalid configurations*
2. *Higher zeroPadding_factor values increase computational load and decreases SNR, so balance resolution with performance based on your hardware capabilities*
3. *If min_range_m is too small (e.g., <0.10 m), the algorithm might return to a distance always equal to min_range_m. This occurs due to Rx-Tx signal coupling, also sometimes referred to as a peak at the zero-frequency bin*

5 Design and implementation

5.1 Data structures

static_distance_context_t

is a structure that encapsulates the state for static distance measurement, including configuration parameters, dynamically allocated buffers, and a CMSIS-DSP FFT instance. It is used by init_static_distance( ) and get_static_distance( ) APIs to manage radar data processing. The definition of static_distance_context_t is as follows.

```
typedef struct
{
    int32_t num_samples_zeroPadded;
    int32_t fft_size;
    float32_t bin_len;
    uint16_t skip;
    uint16_t max_range_bin;
    float32_t frame XENSIV_BGT60TRXX_CONF_NUM_CHIRPS_PER_FRAME,XENSIV_BGT60TRXX_CONF_NUM_SAMPLES_PER_CHIRP;
    float32_t *zero_padded_samples;
    cfloat32_t *fft_buffer;
    float32_t *fft_buffer_real;
    float32_t *fft_win;
    float32_t *integrated_chirp;
    arm_rfft_fast_instance_f32 rfft;
} static_distance_context_t;
```

Figure 16 Data structure

5.2 Global variables

context

A global instance of the static_distance_context_t structure that maintains the radar processing state across API calls.

zeroPadding_factor

Controls the zero-padding level (1, 2, 4, or 8; default: 2) for radar samples, affecting range resolution and processing demands.

max_range_m

Defines the maximum measurable distance in meters (default: 3.0f), used to calculate max_range_bin.

min_range_m

Specifies the minimum measurable distance in meters (default: 0.15f) to avoid near-field artifacts. Used to calculate skip variable to fix zero frequency peak issue.

5.3 Functions

init_static_distance()

Initializes the radar processing context by calculating parameters, allocating buffers, initializing the FFT instance, and setting up a Hamming window for Range FFT processing. This also checks for valid algorithm configuration parameters, frees allocated memory on failure to prevent leaks and prints range accuracy and limits to the console.

get_static_distance()

The get_static_distance() API processes radar sensor data to calculate the distance to a static object in front of radar. It normalizes and debias the input frame data, applies zero-padding, and performs a Range FFT to transform the signal into the frequency domain. The API then integrates chirp data, searches for the peak magnitude within a specified range, and converts the peak index to a distance measurement in meters.

Figure 17 shows signal processing chain used by this code example.

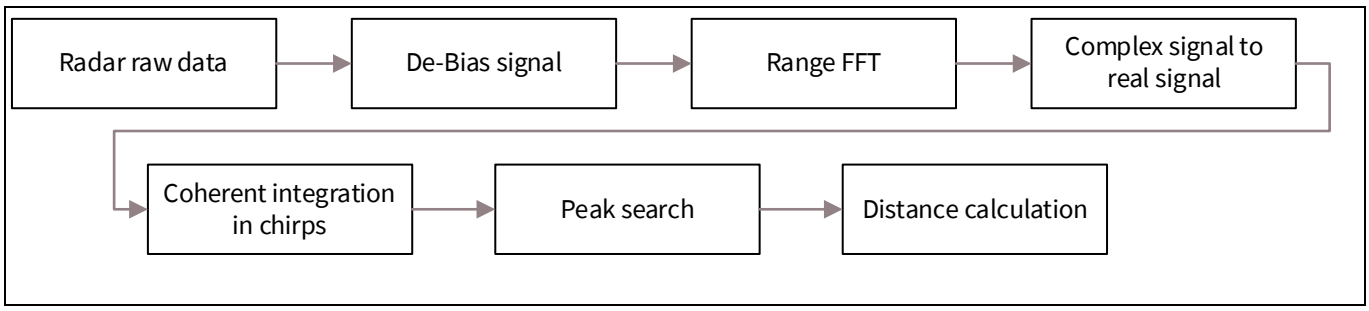


Figure 17 Signal processing chain

6 Related resources

- 60 GHz radar sensors for IoT
- Radar sensors developer community

References

[1] Infineon Technologies AG. *BGT60UTR11AIP MMIC datasheet*; Available online

[2] Infineon Technologies AG. *UG143158 – KIT_CSK_BGT60UTR11AIP user guide*; Available online

[3] Infineon Technologies AG*: AN228571 – Getting started with PSOC™ 6 MCU on ModusToolbox™ application note*; Available online

Revision history

Document revision	Date	Description of changes
1.00	2025-08-29	Initial release

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

Edition 2025-08-29

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

AN094446

Important notice

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

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

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.