MATLAB Radar System API

24 GHz Sense2GoL Pulse

About this document

Scope and purpose

A guide for installing the MATLAB Interface for the 24 GHz Sense2GoL Pulse.

Intended audience

This document is for customers, wishing to get an overview about the 24 GHz Sense2GoL Pulse MATLAB Interface.

Table of Contents

1 Introduction

This section contains a small overview of the available code and a small guide on the MATLAB Interface used.

1.1 MATLAB Application Interface Overview

The MATLAB code and interface allows the user to demonstrate the capabilities of the 24 GHz Sense2GoL Pulse modules. All of the interface‘s classes and functions are contained in the RadarSystemImplementation folder.

The main class representing a hardware evaluation board as a radar system are contained in the folder:

- @RadarSystem

A modular radar system used for S2GLP got only two endpoints to use the radar functionality. The provided endpoints are contained in these folders:

- oEPRadarBaseBoard
- oEPRadarS2GLP

Additional helper functions needed by the API are located in these files:

- clearSP.m
- findRSPort.m
- getAvailablePorts.m
- protocol_read_payload_float.m
- protocol_read_payload_int32.m
- protocol_read_payload_uint8.m
- protocol_read_payload_uint16.m
- protocol_read_payload_uint32.m
- protocol_read_payload_uint64.m
- protocol_write_payload_uint8.m
- protocol_write_payload_uint16.m
- protocol_write_payload_uint32.m
- protocol_write_payload_uint32.m
- resetRS.m

1.2 MATLAB API Guide and Raw Data Fetching

This section describes and enumerates the set of control parameters and methods that can be used for controlling the 24 GHz Sense2GoL Pulsed (Pulsed Doppler) demo board using MATLAB. It involves a quick introduction for reviewing and setting up the radar properties and how to fetch raw data from the board in MATLAB.

Coding examples and demos can be found in the folder Examples\GettingStarted.

1.2.1 MATLAB Code to Extract Data from 24 GHz Sense2GoL Pulse (Pulsed Doppler) Board

To get out raw data form the evaluation board, the following steps are performed:

1. Create the radar system object.
2. Set the endpoint radar properties to customize the frequency chirps.
3. Fetch the raw data and perform the post processing.

Code Listing 1

```
% 1. Create radar system object
szPort = findRSPort; % find the right COM Port
oRS = RadarSystem(szPort); % create the Radarsystem API object

% 2. Set endpoint properties
% get the current parameters from the device
oRS.oEPRadarS2GLP.get_parameters;

% then change the parameters in memory
oRS.oEPRadarS2GLP.parameters.frame_time_sec = 0.1500;
oRS.oEPRadarS2GLP.parameters.number_of_samples = 256;
oRS.oEPRadarS2GLP.parameters.min_speed_mps = 0.3;

% 3. Trigger radar chirp and get the raw data
mxRawData = oRS.oEPRadarS2GLP.get_raw_data;

% 4. plot data
plot([real(mxRawData.sample_data(:,1)),
imag(mxRawData.sample_data(:,1))]);
```

1.2.2 RadarSystem API Object Description

The radar system API object (oRS) contains the endpoint objects oEPRadarBaseBoard and oEPRadarS2GLP.

The object oEP is alternatingly used to access all the properties of the different endpoint objects.

a) oEPRadarBaseBoard

This endpoint is used to provide the common data from the the XMC4700 Radar Base board :

Public methods

Method name	Description
get_board_info	Reads the board info from the connected device
get_consumption	Reads the power and current consumption data from the device

Method name	Description
get_consumption_def	Reads the information what kind of consumption information is available on the device
reset_parameters	Resets to parameters to the factory settings

b) oEPRadarS2GLP

This endpoint is used to set the parameters of the Pulsed Doppler shield and has the following properties and methods:

Writeable properties

Property name	Description
apply_parameters	Applies the previously changed parameters to the board
get_frame_data	Reads the radar raw data from the device
get_raw_data	Alias to “get_frame_data”
get_parameter	Reads the parameter from the device
get_shield_info	Reads general information about the connected RF shield
get_result	Reads the calculation results of the Doppler algorithm running on the device
get_result_and_raw_data	Reads the raw data AND the results from the device in a combined call; use this method, if results and raw data is read at the same time.
get_parameters_def	Reads the limits for the parameters that can be applied.

Available parameters

Property name	Description	Valid values
max_speed_mps	The maximum speed in m/s; acts as a filter for the algorithm, to provide direction information	[0.1 … 3.0]
min_speed_mps	The minimum speed in m/s; acts as a filter for the algorithm, to provide direction information	[0.1 … 3.0]
frame_time_sec	The frame time in seconds	[0.02 … 2.0]
number_of_samples	The number of samples per frame	[32, 64, 128, 256]
sampling_freq_hz	The sampling frequency in Hz	[1000 … 5000]
doppler_sensitivity	The threshold used to detect the direction of motion and speed	[0 … 2000]
motion_sensitivity	The threshold used to detect the general motion	[0… 2000]
equistantant_mode	A indication to se the frame time to the minimum value to have equistantant operation mode, without pause inbetween frames	[0, 1]
number_of_skip_samples	The number of samples that are skipped, before used for signal processing	[0 … 1000]
pulse_width_usec	The pulse width in usec	[4… 10]

Parameter limit	Description
max_speed_mps_lower_boundary	The lower boundary of the maximum speed in m/s
max_speed_mps_upper_boundary	The upper boundara of the maximum speed in m/s
min_speed_mps_lower_boundary	The lower boundary of the minimum speed in m/s
min_speed_mps_upper_boundary	The upper boundara of the minimum speed in m/s
frame_time_sec_lower_boundary	The lower boundary of the frame time in seconds
frame_time_sec_upper_boundary	The upper boundary of the frame time in seconds
sample_list	The valid values for the number of samples per frame
sampling_freq_hz_lower_boundary	The lower boundary of the sampling frequency in Hz
sampling_freq_hz_upper_boundary	The upper boundary of the sampling frequency in Hz
doppler_sensitivity_lower_boundary	The lower boundary of the Doppler sensitivity
doppler_sensitivity_upper_boundary	The upper boundary of the Doppler sensitivity
motion_sensitivity_lower_boundary	The lower boundary of the Motion sensitivity
motion_sensitivity_upper_boundary	The upper boundary of the Motion sensitivity
number_of_skip_samples_lower_boundary	The lower boundary of the number of skip samples
number_of_skip_samples_upper_boundary	The upper boundary of the number of skip samples
pulse_width_usec_lower_boundary	The lower boundary of the pulse width in usec
pulse_width_usec_upper_boundary	The upper boundary of the pulse width in usec

Structure info of the **raw data** where received frame information data is stored:

Variable name	Description
frame_number	Index of the corresponding frame
num_chirps	Number of chirps in a frame
num_rx_antennas	Number of Rx antennas active
num_samples_per_chirp	Number of samples in a chirp
rx_mask	Active receiving antennas
data_format	Determines if the returned data contains only I or Q signal (‘Real Data’), I and Q signals in separate data blocks (‘Complex Data’) or I and Q signals in one interleaved data block (Complex Interleaved Data).
adc_resolution	ADC Resolution
interleaved_rx	Bool flag to indicate if the received data from different antennas stored in an interleaved pattern in the raw data stream.
sample_data	Contains the actual raw data from the radar sensor

Structure info of the **result data** where received frame information data is stored:

Variable name	Description
result_cnt	A count for the results that is incremented, everytime a result is delivered
frame_count	The frame number, could be used to sync raw data and results for analysis
velocity_mps	The velocity in m/s with sign

Variable name	Description
level	The level of the maximum in the sprectrum, where the velocity is calculated from
is_target_departing	A Boolean flag, set to 1 if the target is departing
is_target_approaching	A Boolean flag, set to 1 if the target is approaching
is_motion_detected	A Boolean flag, set to 1 if motion is detected
doppler_frequency_hz	The calculated Doppler frequency in Hz of the maximum in the spectrum

Revision history

Document version	Date of release	Description of changes
V1.0	2020-02-10	Initial version
V1.1	2021-08-13	Updated names of Endpoints Updates to equistantant mode

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

Edition 2021-08-13
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
MATLAB Radar System API

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

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

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

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

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