

Sense2GoL Pulse (Pulsed Doppler) Software User Manual

24 GHz-based radar applications

About this document

Scope and purpose

This user manual describes the Sense2GoL Pulse firmware, supported algorithms and the development recommendations required to build an application around Infineon's Sense2GoL Pulse demo kit.

It describes the demonstration firmware applications of the Sense2GoL Pulse Infineon radar demo board based on the BGT24LTR11, with details of package architecture and contents.

This document provides guidelines for novice users on how to build and run Sense2GoL Pulse radar applications such as motion detection, presence sensing, speed detection and direction of movement (approaching or retreating), and also to support ease-of-use and faster-to-market integration.

Intended audience

This document is intended for users of the Sense2GoL Pulse demo kit who want to get started with Infineon's Sense2GoL Pulse firmware solution, test several sensing demonstrations, and implement custom radar applications in the 24 GHz ISM band.

Related documents

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

- 24 GHz Radar Tools and Development Environment User Manual
- AN598 – BGT24LTR11 Shield (Pulsed Doppler)
- AN602 – Radar Baseboard XMC4700
- AN605 – Radar Baseboard XMC4700 and BGT24LTR11 Shield with Arduino compatibility

Table of contents

| | |
|--|-----------|
| About this document | 1 |
| Table of contents | 2 |
| List of figures | 3 |
| List of tables | 4 |
| 1 Introduction | 5 |
| 2 Basic radar Doppler concepts | 6 |
| 3 Hardware overview | 7 |
| 3.1 Radar Baseboard XMC4700 | 7 |
| 3.1.1 Overview | 7 |
| 3.1.2 XMC4700 block diagram..... | 8 |
| 3.2 BGT24LTR11 Shield | 9 |
| 3.3 Sense2GoL Pulse | 10 |
| 4 Firmware description | 11 |
| 4.1 Overview | 11 |
| 4.2 Global architecture..... | 11 |
| 4.3 Firmware concept | 13 |
| 4.4 Raw data acquisition..... | 14 |
| 4.4.1 Doppler pulse generation | 14 |
| 4.4.2 Data sampling | 16 |
| 4.5 Radar control layer | 17 |
| 4.5.1 Radar control API | 17 |
| 4.5.2 Data store module..... | 17 |
| 4.5.3 Data store and EEPROM management..... | 19 |
| 4.6 DAVE™ project overview..... | 20 |
| 4.7 Firmware package overview | 23 |
| 4.8 Footprint..... | 23 |
| 4.9 Firmware timings | 24 |
| 4.10 Firmware customization and configuration..... | 25 |
| 5 Algorithm description | 27 |
| 5.1 Target approaching/target departing | 27 |
| 5.2 Motion detected | 27 |
| 6 References | 28 |
| Revision history | 29 |

List of figures

| | | |
|-----------|---|----|
| Figure 1 | Doppler effect..... | 6 |
| Figure 2 | Radar Baseboard XMC4700..... | 7 |
| Figure 3 | XMC4700 block diagram..... | 8 |
| Figure 4 | BGT24LTR11 Shield board..... | 9 |
| Figure 5 | Sense2GoL Pulse hardware platform..... | 10 |
| Figure 6 | Firmware architecture..... | 11 |
| Figure 7 | Firmware flow diagram..... | 13 |
| Figure 8 | Raw data acquisition flow diagram..... | 14 |
| Figure 9 | All control signals..... | 15 |
| Figure 10 | One pulse generation..... | 15 |
| Figure 11 | Frame structure and terminology..... | 16 |
| Figure 12 | Sense2GoL Pulse data acquisition and sampling flow..... | 16 |
| Figure 13 | Data acquisition flow diagram..... | 17 |
| Figure 14 | Data store hardware device and algorithm settings structures..... | 18 |
| Figure 15 | Interconnection of the data store module with other firmware modules..... | 19 |
| Figure 16 | EEPROM block diagram..... | 19 |
| Figure 17 | EEPROM structure..... | 20 |
| Figure 18 | ADC DAVE™ app configuration..... | 22 |
| Figure 19 | DMA DAVE™ app configuration..... | 22 |
| Figure 20 | Package folder structure..... | 23 |
| Figure 21 | Raw data acquisition timings..... | 24 |
| Figure 22 | Movement detection algorithm flow..... | 27 |

List of tables

List of tables

| | | |
|---------|--|----|
| Table 1 | Sense2GoL Pulse control and data signals description..... | 14 |
| Table 2 | DAVE™ project apps used..... | 20 |
| Table 3 | PWM and GPIO pin configurations..... | 21 |
| Table 4 | Sense2GoL Pulse firmware footprint..... | 24 |
| Table 5 | Define statements used for radar firmware configuration..... | 25 |

1 Introduction

The Sense2GoL Pulse radar demo kit is a demonstration platform for Infineon's silicon-germanium 24 GHz radar chipset BGT24LTR11. It consists of two boards: the main **Radar Baseboard XMC4700** and a radar front-end board **BGT24LTR11 Shield**.

This document focuses on the demonstration firmware and the Graphical User Interface (GUI) for a **pulsed Doppler** implementation.

The Sense2GoL Pulse board is designed to evaluate the capabilities of the BGT24LTR11 IC, comprising one transmit and one receive channel with the XMC4700 microcontroller utilizing Infineon's powerful, free-of-charge toolchain DAVE™ for microcontroller programming.

2 Basic radar Doppler concepts

The main radar technique used for this demonstration is Continuous Wave (CW/Doppler) radar.

Doppler radar operates on the principle of sending a beam of electromagnetic radiation waves, tuned to a precise frequency, toward a moving object. When the electromagnetic radiation wave hits the moving object, it “bounces” back toward the source, which also contains a receiver. However, since the wave is reflected off a moving object, the wave is shifted as outlined by the Doppler effect.

The wave that is coming back toward the radar is treated as an entirely new wave, as if it were emitted by the target it bounced off. The target is acting like a new source for this new wave. When it is received at the radar, this wave has a frequency different from the frequency that was originally sent toward the target.

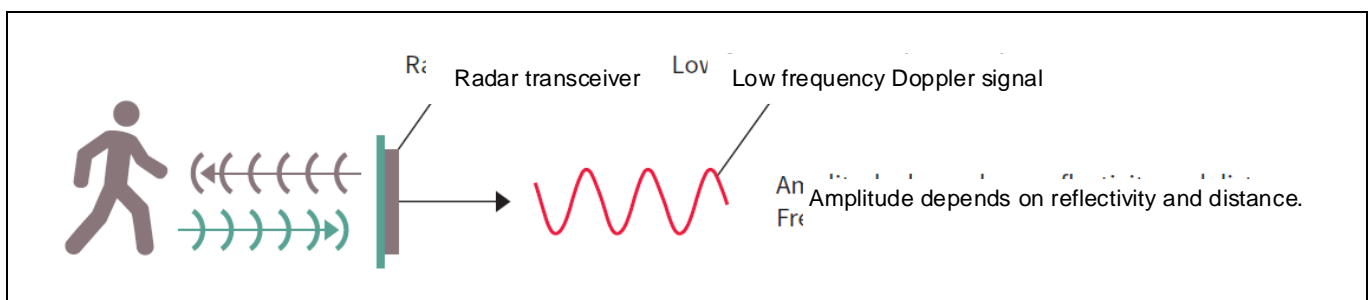


Figure 1 Doppler effect

Since the electromagnetic radiation was at a precise frequency when sent out and is at a new frequency on its return, this can be used to calculate the velocity v of the target.

The change in phase between the transmitted and received signal is used to calculate the target’s velocity. The Doppler effect shifts the received frequency up or down based on the radial velocity of the target (closing or opening) in the beam, allowing for the direct and highly accurate measurement of target velocity.

Doppler shift f_d and velocity v are dependent on each other according to the following equations:

$$f_d = \frac{2 \cdot f_{Tx} \cdot v}{c} \cdot \cos \alpha$$

$$v = \frac{c \cdot f_d}{2 \cdot f_{Tx} \cdot \cos \alpha}$$

f_d : Doppler frequency [Hz]

f_{Tx} : Carrier frequency (24.0×10^9 Hz)

v : Object velocity [m/s]

c : Speed of light in vacuum (3×10^8 m/s)

α : Angle between beam center and target moving direction

3 Hardware overview

The Sense2GoL Pulse radar system is a demo platform for Infineon’s 24 GHz BGT24LTR11 radar chipset. It consists of two boards: the **Radar Baseboard XMC4700** and a radar front-end board, **BGT24LTR11 Shield**.

3.1 Radar Baseboard XMC4700

This section gives an overview of the **Radar Baseboard XMC4700** hardware platform, and more detailed information can be found in the corresponding application note.

3.1.1 Overview

The Radar Baseboard XMC4700 is a generic sensor interface for Infineon’s 24 GHz radar sensors. The central unit of the XMC4700 32-bit ARM® Cortex™-M4 based microcontroller can perform radar data processing or forward the sensor data to a USB interface, serial interface or an Arduino interface. The board is designed to allow customers to carry out prototyping and system integrations, as well as initial product feature evaluations.

An onboard debugger with licensed firmware from SEGGER allows easy debugging over USB. Infineon’s powerful, free-of-charge toolchain DAVE™ can be used for programming the XMC4700 microcontroller. This application note describes the key features and hardware configuration of the Radar Baseboard XMC4700 in detail. Figure 2 shows the Radar Baseboard XMC4700 hardware platform.

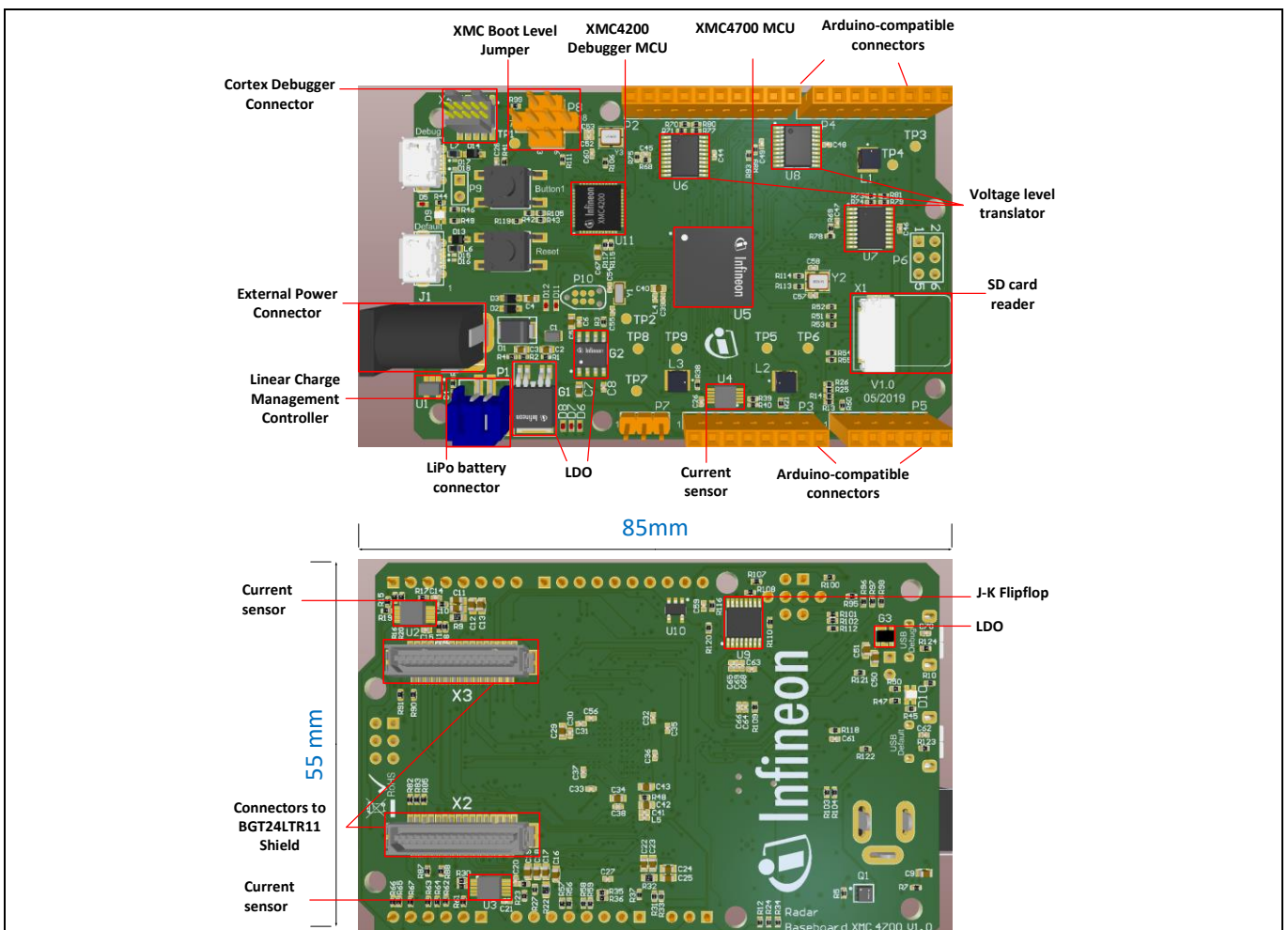


Figure 2 Radar Baseboard XMC4700

3.1.2 XMC4700 block diagram

The Radar Baseboard XMC4700 embeds Infineon's XCM4700, 32-bit ARM® Cortex™-M4 based microcontroller. It can be used to perform radar data processing, configure the board's peripherals and RF shield management, and enable radar data communication via a USB or serial interface.

The Sense2GoL Pulse firmware is running in the XMC4700 microcontroller and configures the following peripherals:

- Analog Digital Converter (ADC)
- Direct Memory Access (DMA)
- General-Purpose Input Output (GPIO)
- USIC for I²C serial interface
- CCU8/4 for timer and PWM control signals
- Capture and Compare Unit (CCU)
- Hardware interrupts
- USB interface for host communication

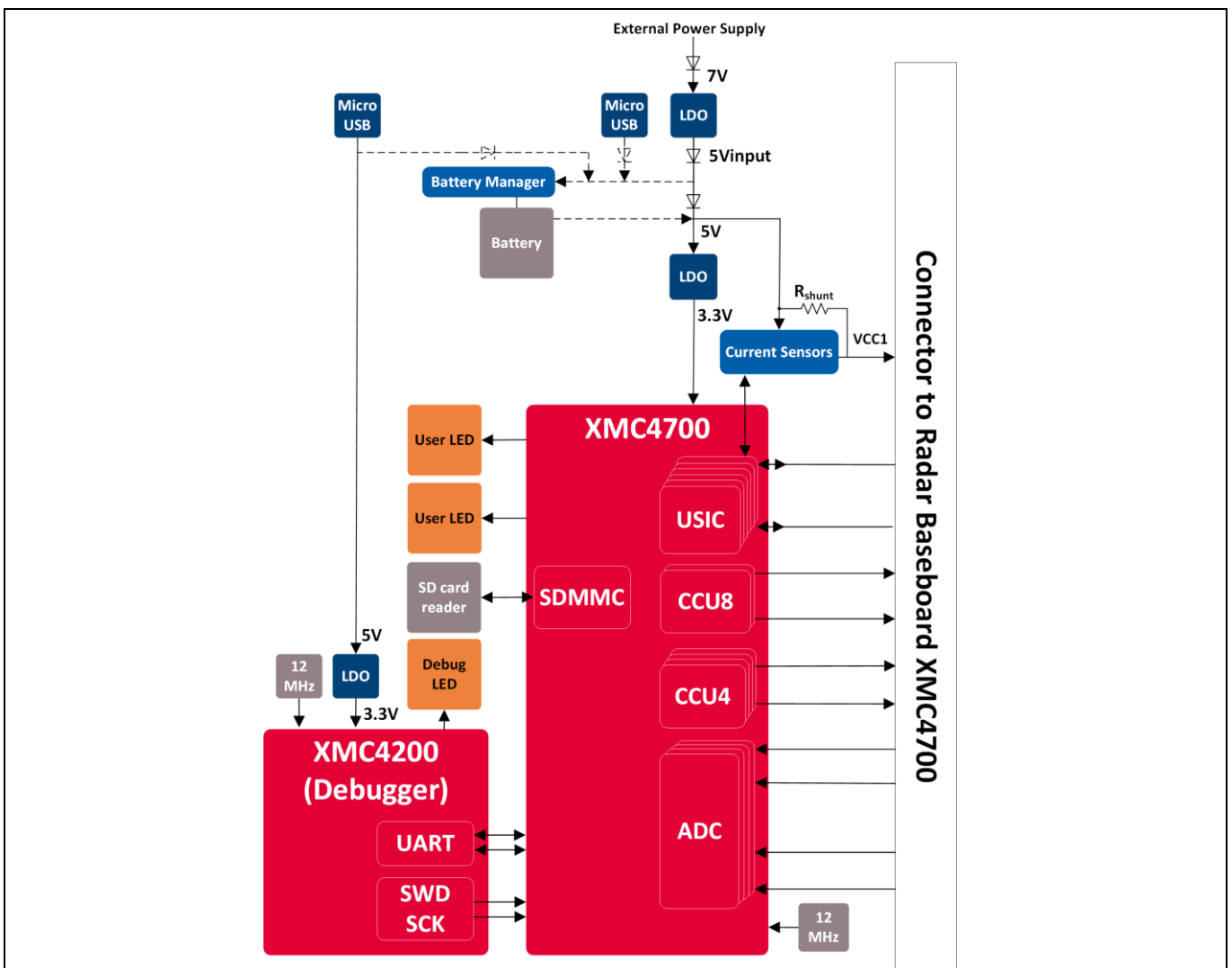


Figure 3 XMC4700 block diagram

3.2 BGT24LTR11 Shield

This section gives an overview of the **BGT24LTR11 Shield** hardware platform. More detailed information can be found in the corresponding application note.

The radar shield is shown in Figure 4. It contains the following:

- RF part: consists of the Infineon 24 GHz radar MMIC **BGT24LTR11** and includes micro-strip patch antennas for the TX and RX sections.
- S&H part: consists of SPST switches and hold capacitors to sample and hold the analog I/Q signals from the MMIC using a control signal from the microcontroller.
- Analog amplifier part: consists of two amplifier stages, used to smooth the sampled I/Q signals from S&H circuitry and amplify them for the digital part.
- EEPROM part: can be used to store data such as board identifier information, and RF shield hardware settings.

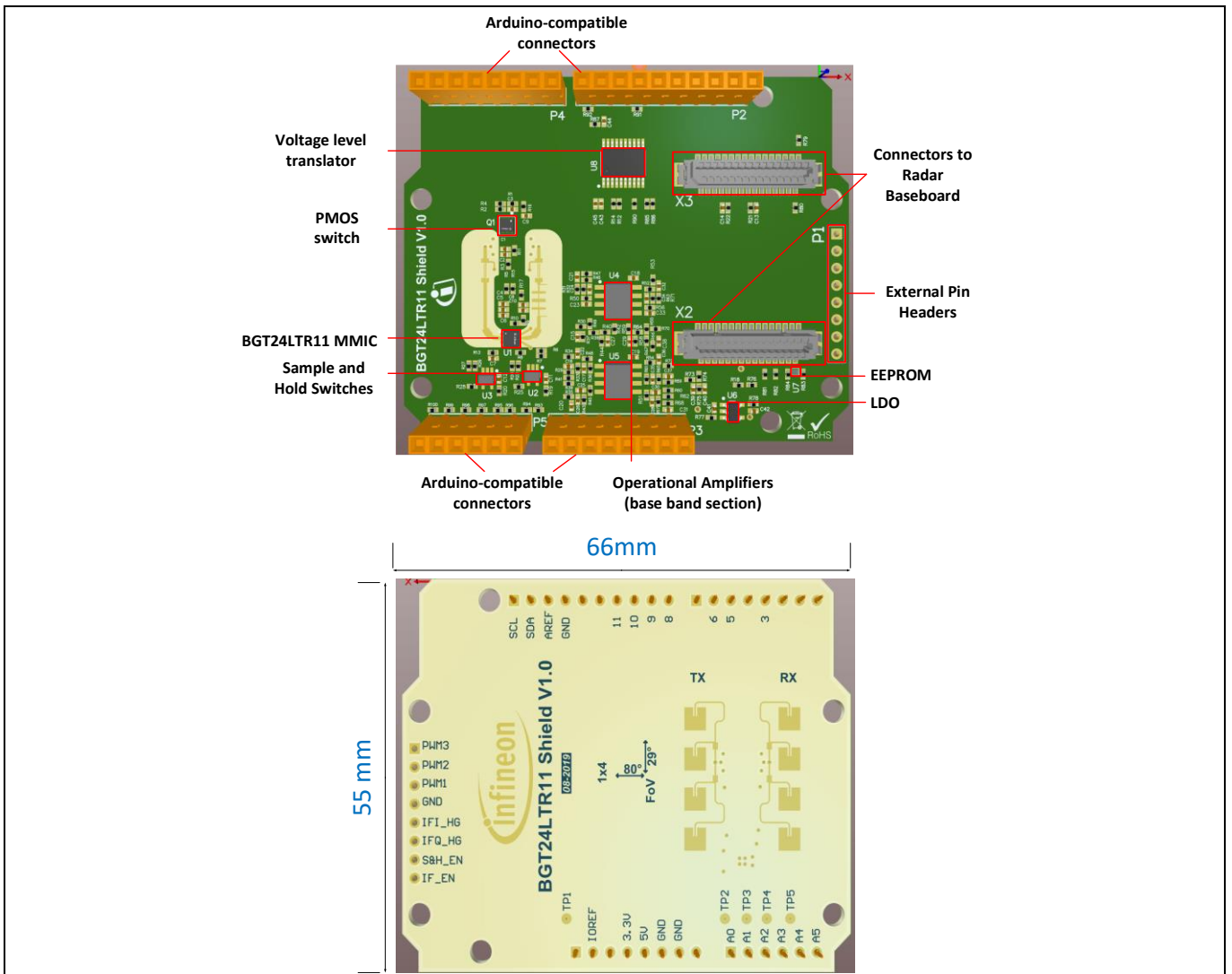


Figure 4 BGT24LTR11 Shield board

Hardware overview

3.3 Sense2GoL Pulse

Figure 5 shows the Sense2GoL Pulse radar system hardware platform, which consists of two boards: the **Radar Baseboard XMC4700** and the **BGT24LTR11 Shield** radar front-end board.

The BGT24LTR11 Shield should be plugged into the bottom side of the Radar Baseboard XMC4700 using the two connectors X2 and X3.

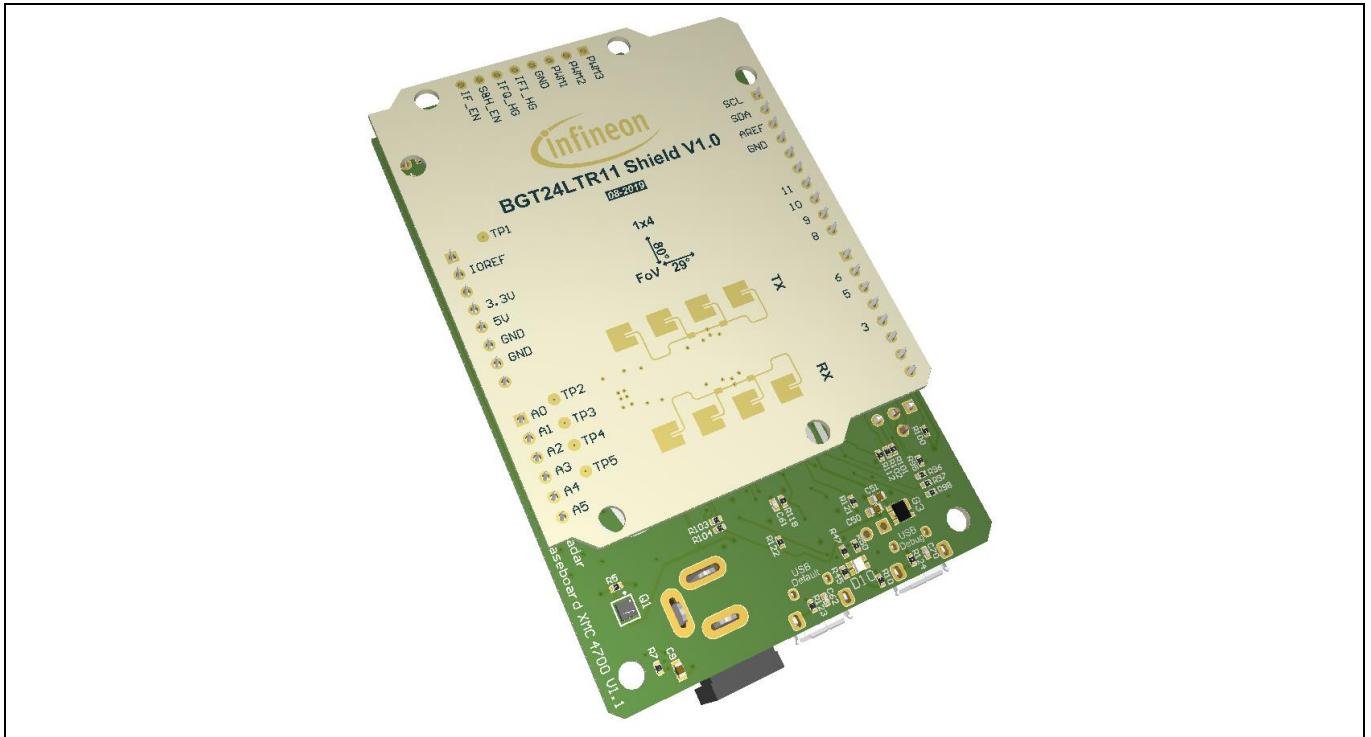


Figure 5 Sense2GoL Pulse hardware platform

4 Firmware description

4.1 Overview

The Sense2GoL Pulse FW is a piece of software written in C language to control different ICs and peripherals via the host processor, which is the XMC4700 32-bit ARM® Cortex™-M4 MCU in the Radar Baseboard XMC4700.

The Sense2GoL Pulse firmware is developed with Infineon’s DAVE™4 (Digital Application Virtual Engineer), free development toolchain. It is a C/C++-language software development and code generation tool for XMC™ microcontroller applications. It is based on graphical interface 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 Sense2GoL Pulse firmware includes various radar demonstration applications to demonstrate the Sense2GoL Pulse board’s capabilities and facilitate the development of user applications, and that can be used for:

- motion detection
- speed detection and direction of movement, approaching or retreating

4.2 Global architecture

This section describes the software components of the Sense2GoL Pulse firmware illustrated in Figure 6.

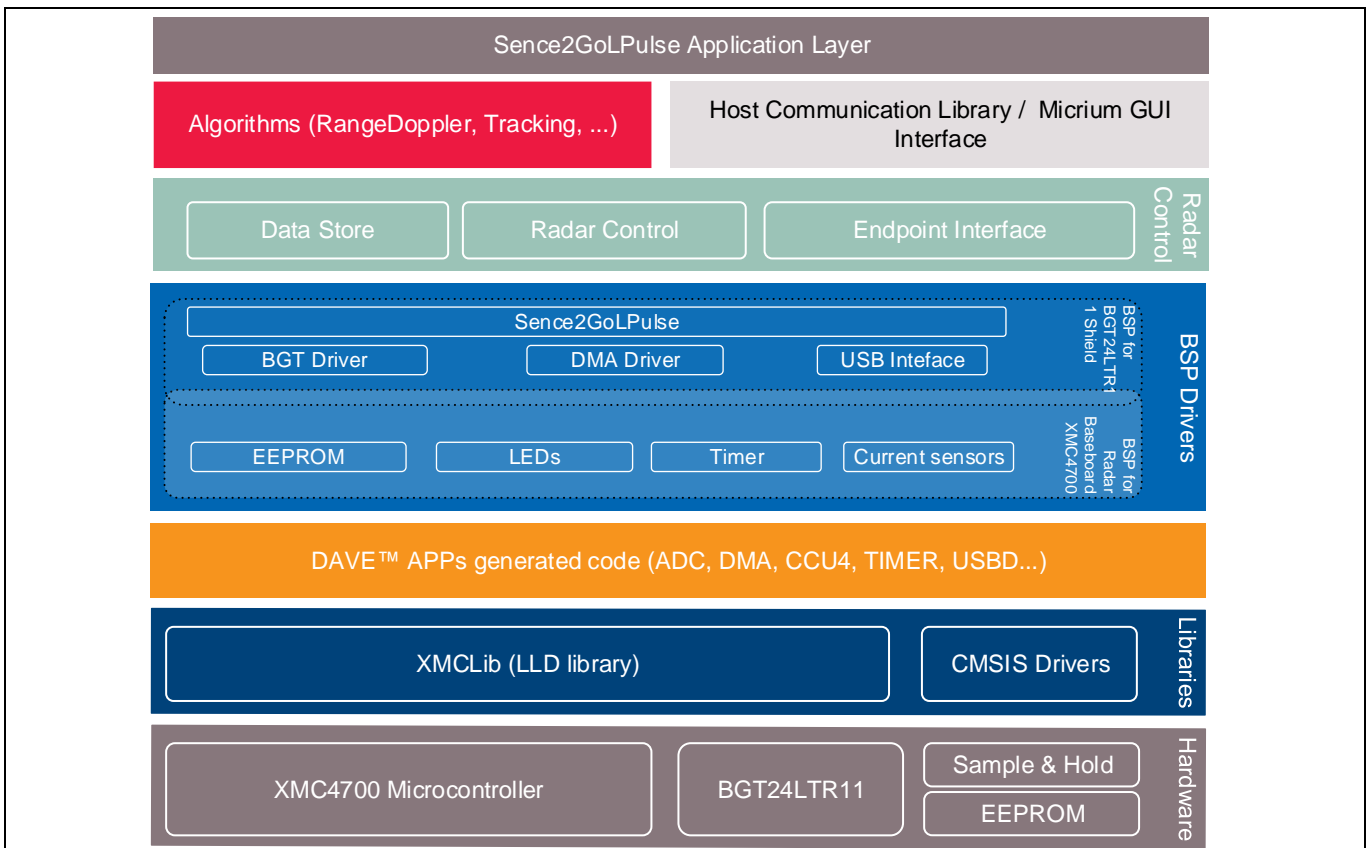


Figure 6 Firmware architecture

- **Sense2GoL Pulse application layer** – a customer-specific layer that defines the entry point of the Sense2GoL Pulse demonstration platform, and contains:

Firmware description

- the initialization functions for XMC™ peripherals, host communication library and radar control layer
- the main application state machine
- a function template for the user to add their own code (data acquisition callbooks, algorithms process...)
- functions and variables to manage the communication with Radar GUI interface.
- **Algorithms** – contains the supported algorithms by the Sense2GoL Pulse demonstration platform implementations, e.g. Range-Doppler, presence sensing used for processing and calculating information out of radar raw data in order to detect stationary and moving objects (refer to the algorithms section for more details).
- **Communication library** – contains a set of functions to ensure USB data communication between the Sense2GoL Pulse board and the Radar GUI tool:
 - defines all communication endpoints
 - contains communication endpoints settings and configuration
 - contains a protocol communication layer.
- **Radar control layer** – contains high-level functions that can be used to set the specific mode for the Sense2GoL Pulse board, basically classified into three categories:
 - **radar control** – offers high-level radar services to the user application and host communication library layers, e.g. radar device initialization, radar start, radar stop, set/get calibration...
 - **data store** – contains global structures for hardware settings and algorithms configuration
 - **endpoint interface** – ensures communication between the host communication library and radar control layer.
- **BSP driver** – the Board Support Package (BSP) driver is a set of functions that can be used to control and manage all components embedded in the Sense2GoL Pulse board, and it contains:
 - low-layer functions to initialize and control the specific board features (BGT24LTR11, Sample and Hold (S&H), EEPROM, current sensors...)
 - functions to control power-up and power-down sequences for all hardware components
 - functions to manage the data acquisition process from BGT to XMC™ microcontroller internal RAM memory (DMA, timer, ADC).
- **DAVE™ apps generated code** – contains the generated library sources from DAVE™-configured building-block apps for XMC4700 MCU peripherals. It contains Application Program Interfaces (APIs) and data structures meant to be used in application code.
- **Libraries** – contains the following libraries:
 - **CMSIS** – Cortex Microcontroller Software Interface Standard (CMSIS) is a vendor-independent hardware abstraction layer for the Cortex-M processor series and defines generic tool interfaces. The CMSIS enables consistent device support and establishes simple software interfaces to the processor and the peripherals, simplifying software reuse, reducing the learning curve for microcontroller developers, and reducing the time-to-market for new devices.

Firmware description

- **XMC-Lib** – consists of various low-level drivers for the XMC™ microcontroller's family peripherals. Each driver consists of a set of routines and data structures covering all peripheral functionalities. Built on top of the CMSIS, it provides access to all XMC4000 peripheral features.

4.3 Firmware concept

Sense2GoL Pulse firmware can be divided into repetitive and non-repetitive tasks. After device initialization, control resides in the main loop, where two possible events can trigger further processing:

- frame timer interrupt
- GUI request by the host communication protocol.

Manage the communication with the Radar GUI interface.

A task-level flow diagram of the firmware is shown in Figure 7.

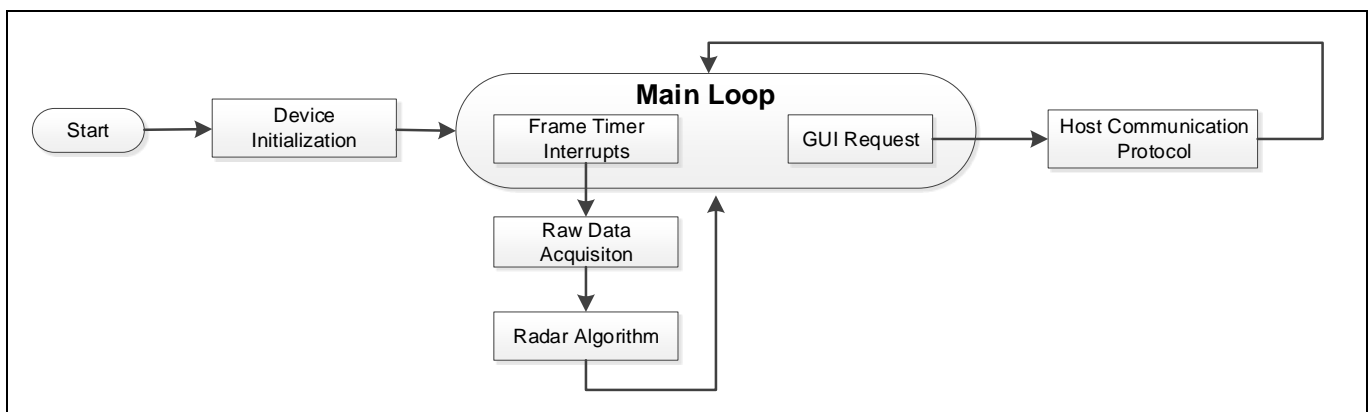


Figure 7 Firmware flow diagram

- **Device initialization** – this is the first task to be executed, only once when the firmware is started. During this task, the firmware initializes XMC™ peripherals and the Sense2GoL Pulse radar device, then registers the used endpoints for the host communication. Program control then goes to the main loop.
- **Frame timer interrupt** – this is a periodic hardware timer interrupt, which triggers the start of the new frame raw data acquisition process.
- **Raw data acquisition** – the BSP layer is configured to start collecting raw data from the radar device.
- **Radar algorithm** – once raw data is collected, run the registered algorithm process in order to compute the parameter of the target's velocity.

4.4 Raw data acquisition

Figure 8 shows the main blocks of the Sense2GoL Pulse raw data acquisition phase in detail, before going through the radar algorithm processing phase. There are more details about the chirp generation and data sampling phases in the next sections.

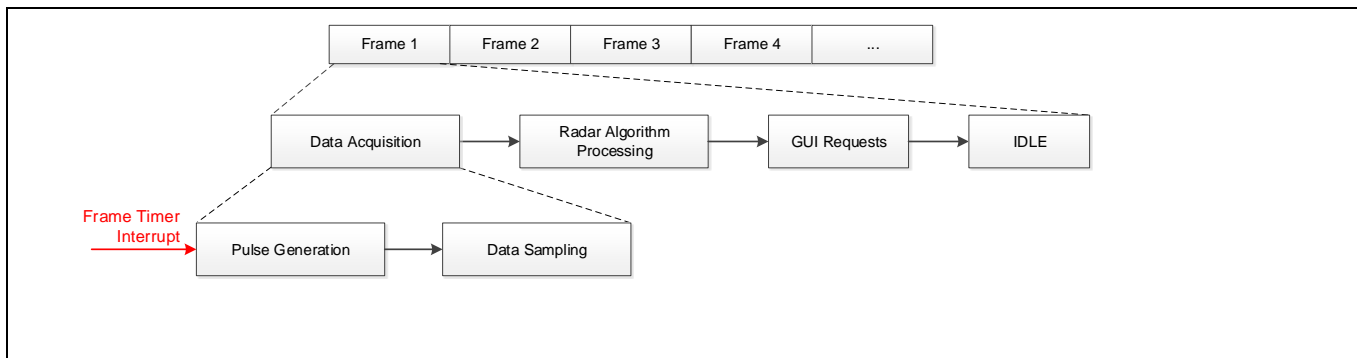


Figure 8 Raw data acquisition flow diagram

4.4.1 Doppler pulse generation

Pulse generation is the first part of the data acquisition process. It is triggered by an internal frame timer, which will start generating pulses. The Sense2GoL Pulse firmware communicates and controls the BGT and S&H over PWM signals. It generates periodic and continuous pulses based on the number of data samples required.

As shown in 15 and Table 1, the BGT24LTR11 Shield is controlled through three PWM signals and one normal GPIO signal.

- **PWM1 and PWM2:** PWM control signals for BGT_VCC and BGT_TX_ON, they are used to turn on the BGT24LTR11 only during the “Pulse” duration and it remains off for the remaining time.
- **PWM3:** Normal GPIO control signal for BGT_VCC_PTAT, it is used to turn on the VPTAT 1 ms before the start of the first pulse of the frame and then remains on until the end of the frame time.
- **S&H_EN:** PWM control signal for S&H, it is used to turn on the S&H only during the “Pulse” duration and it remains off for the remaining time.

Table 1 Sense2GoL Pulse control and data signals description

| Pin | Signal name | Pin description |
|--------------|-------------|--|
| BGT_VCC_PTAT | PWM3 | Control signal for VCC_PTAT pin of BGT24LTR11 |
| BGT_TX_ON | PWM2 | Control signal for TX_ON pin of BGT24LTR11 |
| BGT_VCC | PWM1 | Control signal to turn on/off V _{CC} for BGT24LTR11 |
| SH_EN | S&H_EN | Control signal for S&H switches |
| IFI | IFI_HG | Second baseband amplifier stage output for IFI signal |
| IFQ | IFQ_HG | Second baseband amplifier stage output for IFQ signal |

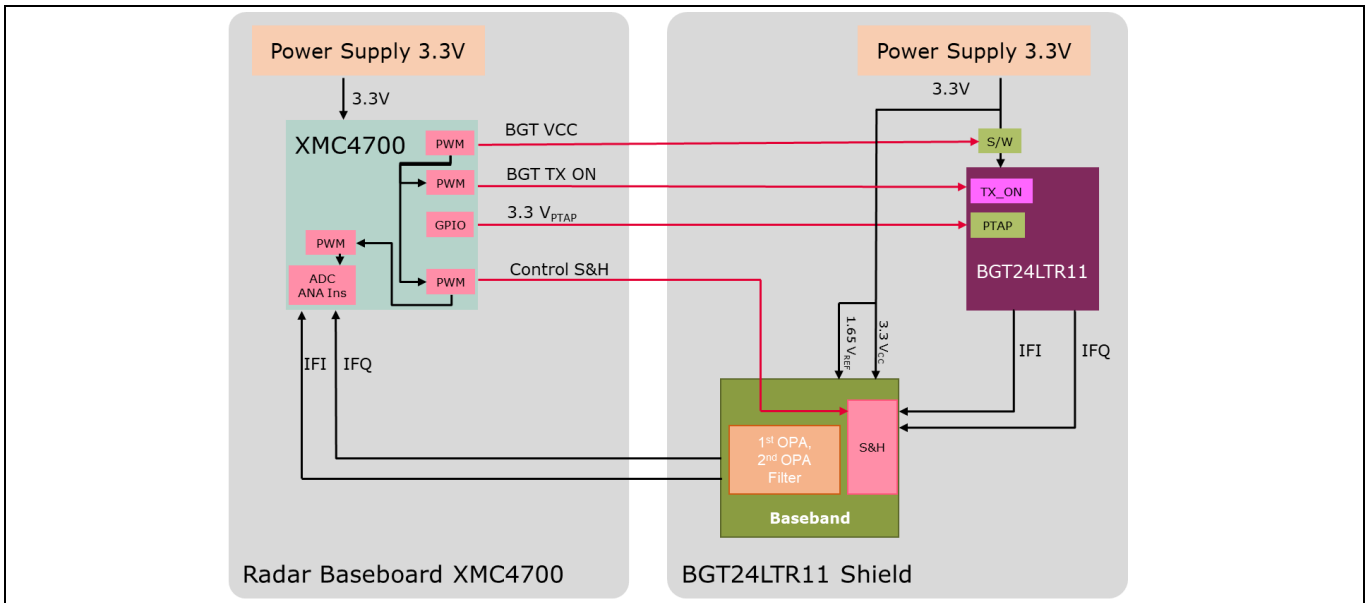


Figure 9 All control signals

The pulse width or BGT V_{CC} on-time can vary between 1 and 10 μs , and it represents the time portion to have one data sample. Short pulse width reduces the sample time of the BGT's output signals, hence saving more power. However, the time might not be enough to charge to the S&H, reducing the final signal strength at the output.

For more details about BGT24LTR11 Shield configuration and power saving, please refer to the BGT24LTR11 Shield hardware platform application note.

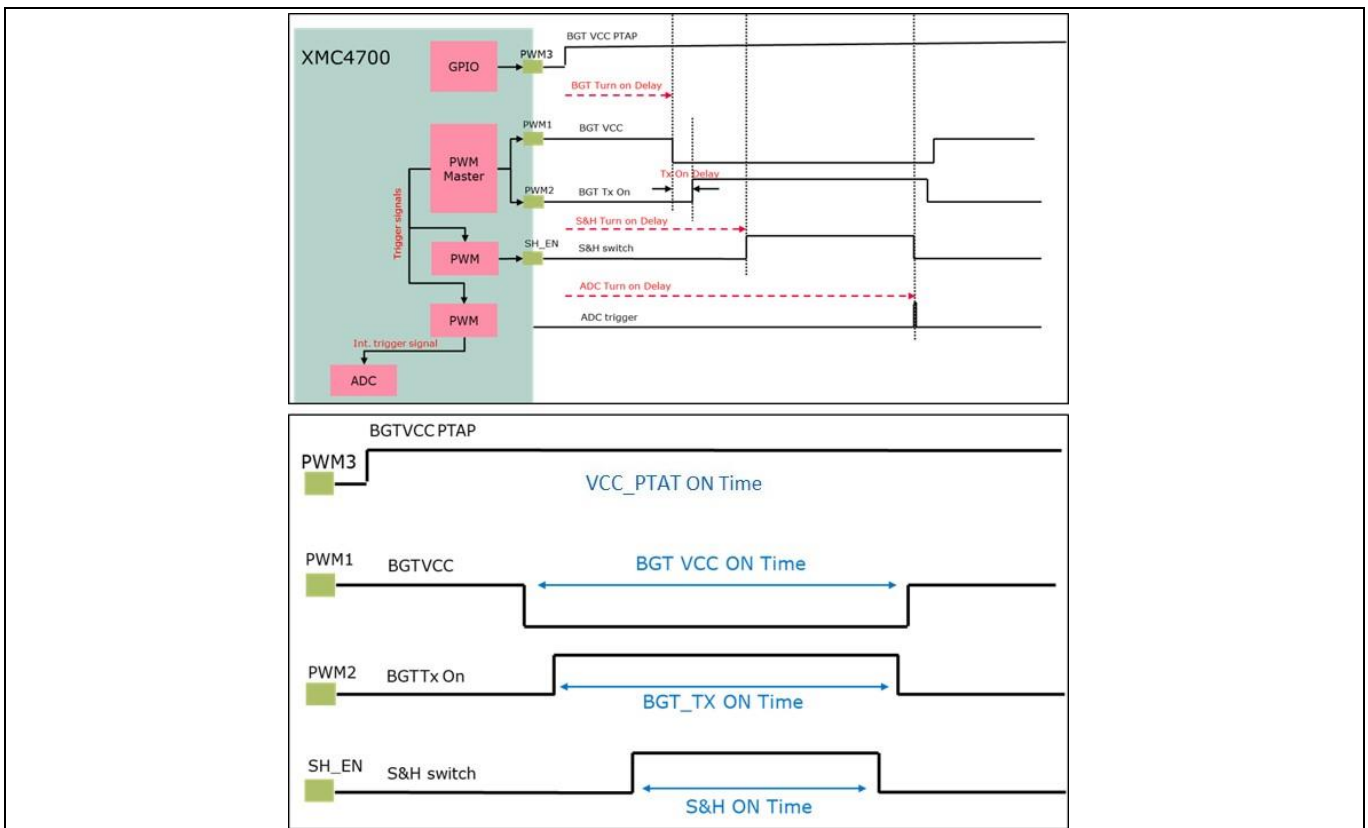


Figure 10 One pulse generation

Figure 11 shows the configuration of a frame. Each frame is a series of pulses, based on the number of samples, followed by a frame off-time.

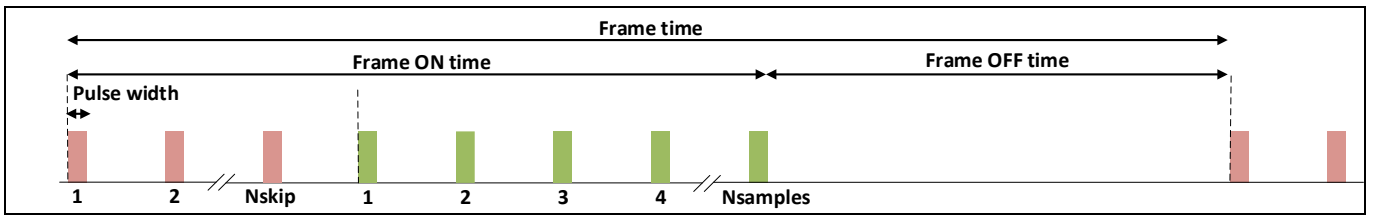


Figure 11 Frame structure and terminology

4.4.2 Data sampling

The data acquisition process for Sense2GoL Pulse application is shown in Figure 13.

The start of the data acquisition process will be triggered by an internal frame timer, which will start pulse generation over PWM signals. The data sampling process is triggered by an internal PWM signal at the same time, which triggers the ADC to start sampling data from the RX antenna.

Once an ADC sample is ready to be moved to the acquisition buffer, the DMA transfer starts. This task is repetitive; it will end once the number of required samples per frame is reached. At that moment, the DMA will stop the transfer process by raising a transfer complete interrupt.

The frame size (or number of samples per frame) depends on the number sample required by the application or the algorithm and the sample skip count. The sample skip count is the number of samples to be skipped in each frame to get rid of the DC offset in the I and Q signals. The recommended sample skip count value is 100; these samples are completely disregarded in the signal processing chain.

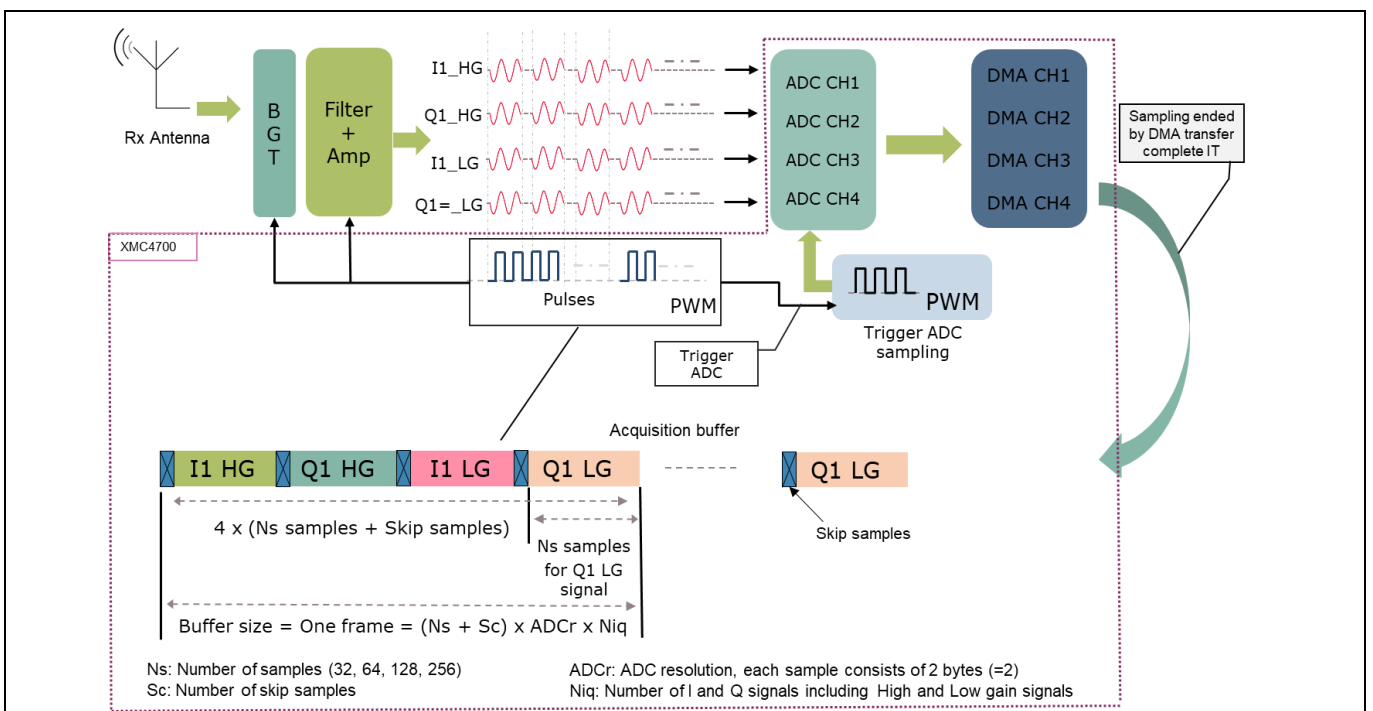


Figure 12 Sense2GoL Pulse data acquisition and sampling flow

Firmware description

Figure 13 gives the detailed flow diagram of the data acquisition phase. Here is a short description of each step of the acquisition process:

- **Step 1 – configure and enable DMA:** DMA source/destination address set-up; four DMA channels are configured for the RX complex (I/Q) data samples (Q1 HG, I1 HG, Q1 LG and I1 LG).
- **Step 2 – configure pulse generation:** Configure all PWM control signals for BGT and S&H.
- **Step 3 – trigger pulse generation:** Triggered by an internal frame timer at the beginning of each frame; start PWM pulse generation and ADC data sampling.
- **Step 4 – start ADC sampling:** ADC sampling will start automatically at the same time as pulse generation; it is controlled by an internal PWM signal in order to trigger equidistant ADC samples.
- **Step 5 – stop acquisition process:** Based on the number of samples required, disables the DMA peripheral and stops pulse generation and ADC sampling.

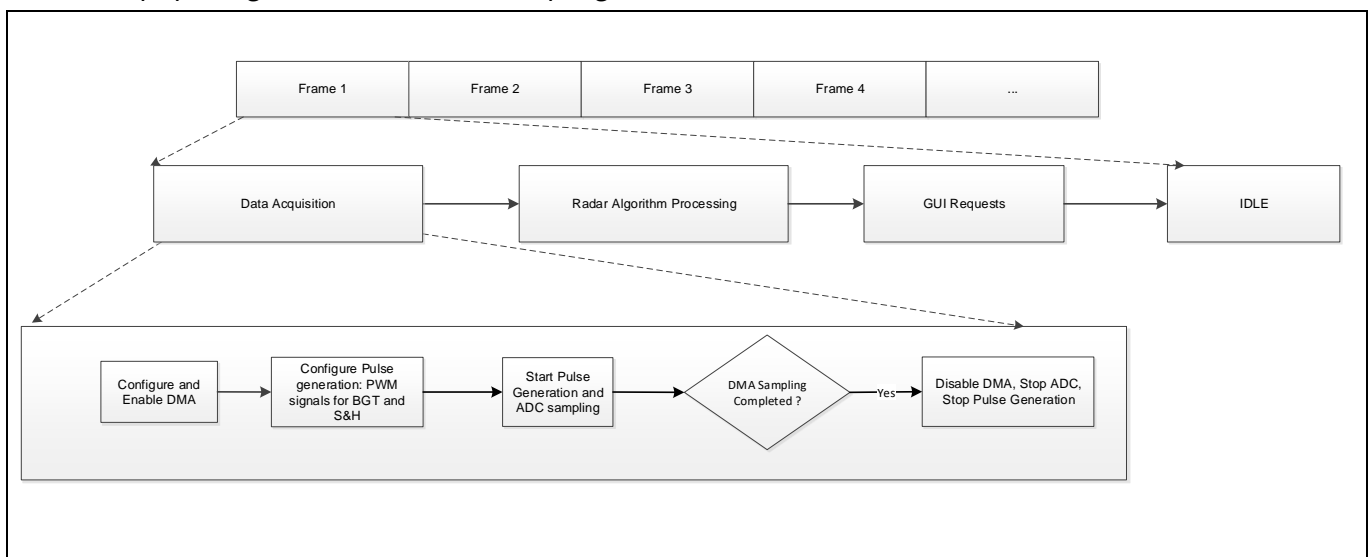


Figure 13 Data acquisition flow diagram

4.5 Radar control layer

The Sense2GoL Pulse software package provides a simple interface to the radar kit through the radar control layer, which offers full flexibility to configure all radar parameters based on the application end requirements.

4.5.1 Radar control API

The APIs define the high-level interface used to configure the radar RF parameters, the behavior and capabilities of the component, and its inputs and outputs, and provide a set of firmware methods to manage radar functionalities. Radar control APIs are called from the application layer and are prefixed by “radar_”.

Detailed technical information about the radar APIs available to the user is provided in a compiled HTML file in the Firmware_Software/Documentation/FW_API folder, where all the functions and parameters are described.

4.5.2 Data store module

The data store module is apart from the radar control layer; it mainly contains the hardware device settings and the algorithm settings structures, as shown in Figure 14.

Firmware description

```

/*
 * Hardware configuration structure
 */
typedef struct device_settings_TAG
{
    uint32_t      struct_length;
    uint32_t      struct_version;

    /* Frequency Settings */
    uint32_t      pll_chirp_time_usec;
    float         rf_frequency_KHz;
    uint32_t      pll_modulation_mode;

    /* ADC Settings */
    uint32_t      adc_sampling_freq_Hz;
    uint8_t       adc_resolution;
    uint8_t       adc_use_post_calibration;

    /* Frame Settings */
    uint32_t      num_of_chirps_per_frame;
    uint32_t      frame_period_usec;
    uint32_t      num_samples_per_chirp;
    uint8_t       rx_antenna_mask;
    uint32_t      num_sample_skip_count;

    /* Pulse mode settings */
    uint32_t      pulse_width_usec;
    uint32_t      sample_hold_on_time_nsec;
    uint32_t      ptat_is_switched;
    uint32_t      ptat_delay_usec;
} device_settings_t;

/*
 * Algorithm result structure
 */
typedef struct
{
    float         velocity_kmph;
    float         level;
    uint8_t       target_departing;
    uint8_t       target_approaching;
    uint8_t       motion_detected;
    float         doppler_frequency_hz;
    float         *p_doppler_spectrum;
    uint32_t      doppler_spectrum_elems;
    uint32_t      frame_counter;
} algo_result_t;

/*
 * Algorithm settings structure
 */
typedef struct algo_settings_TAG
{
    uint32_t      struct_length;
    uint32_t      struct_version;
    float         speed_detection_threshold;
    float         min_speed_kmph;
    float         max_speed_kmph;
    uint32_t      use_high_gain_doppler;
    float         motion_detection_threshold;
} algo_settings_t;

```

Figure 14 Data store hardware device and algorithm settings structures

These two structures are shared between all firmware modules. If a firmware module, e.g. the host communication library or algorithm, requires one or more parameters from the settings structures, a **fetch operation** is performed to get the up-to-date value from the data store, as shown in Figure 15. The resulting structure holds the results of the last algo operation.

On the other hand, if there is an update for one or many parameters in the settings structures, a **push/store operation** is performed to update the data store structures with the new changed value.

A shadow (or a copy) of the hardware settings structure is maintained in the data store, and it always contains the old settings parameters. In case of unsupported parameters (e.g. the value is out of range), the current hardware structure is discarded and overwritten by the shadow copy and the new required change will be discarded.

At the beginning, the shadow and current settings structures parameters are set to the default settings values from the *config.h* file. The user can change these default settings by updating the *config.h* file or through the radar GUI tool interface.

In case of a valid hardware change request received from the GUI, the hardware structure will be updated with the new value in the data store. The **radar control** sub-module will apply this change on the BSP driver before the start of acquisition of the next new frame.

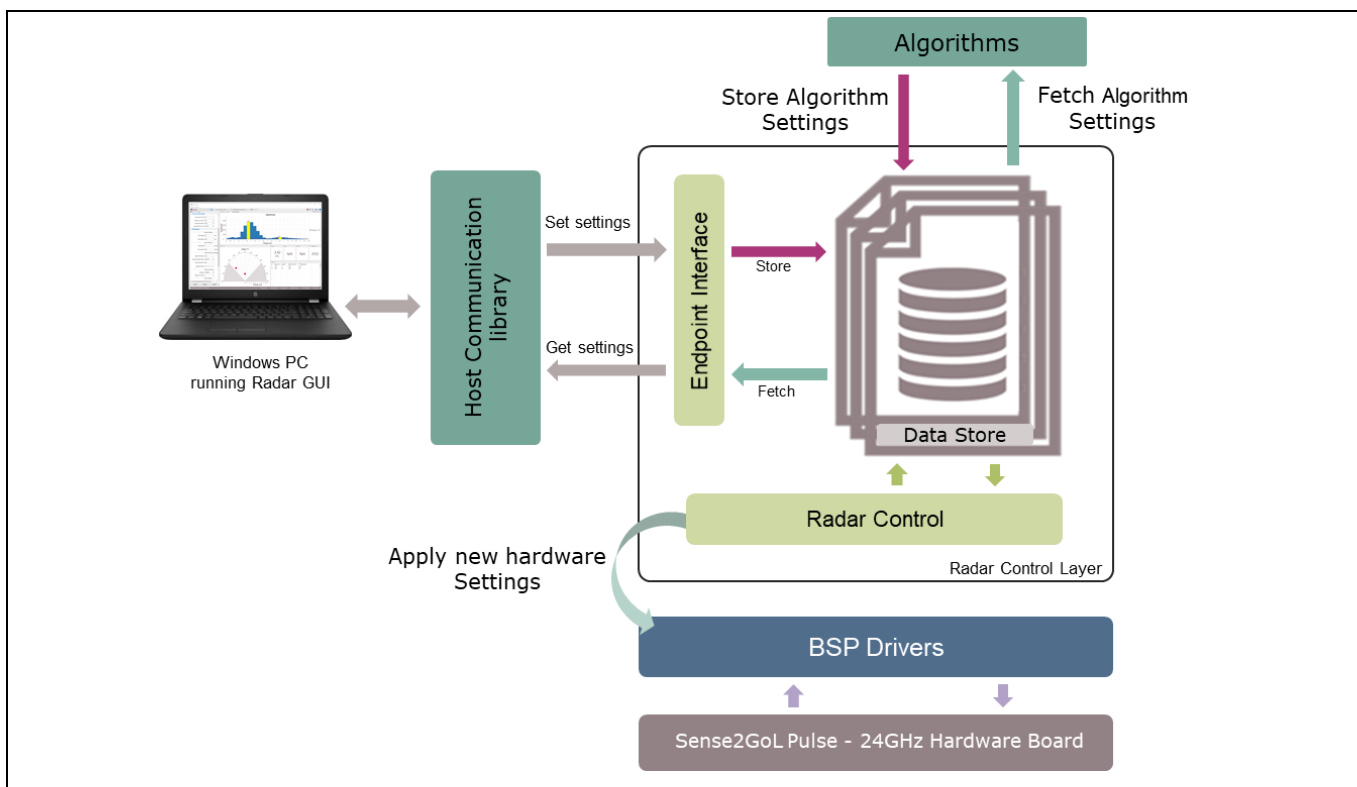


Figure 15 Interconnection of the data store module with other firmware modules

4.5.3 Data store and EEPROM management

The BGT24LTR11 Shield contains 16 K-Byte serial EEPROM, The I²C serial data interface is used to transfer addresses and data into and out of the EEPROM.

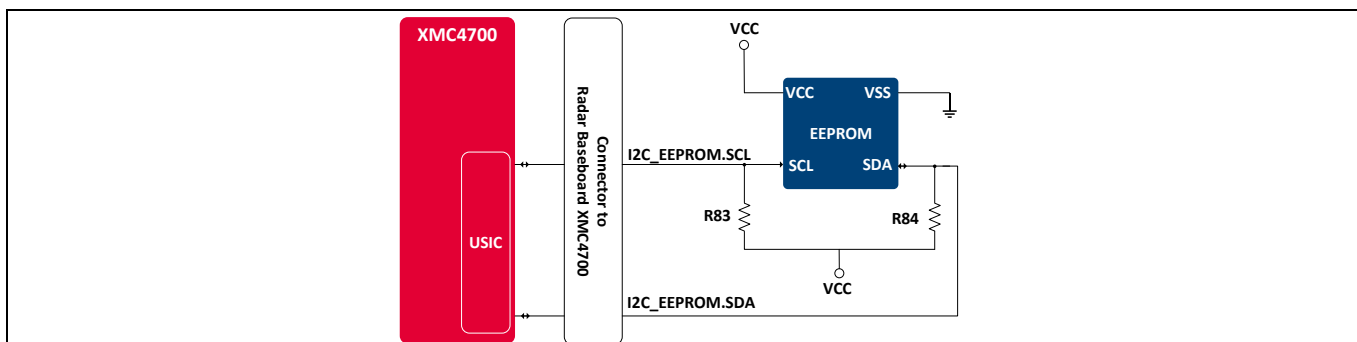


Figure 16 EEPROM block diagram

In Sense2GoL Pulse firmware, the EEPROM is used to store the **hardware settings** and the **algorithm settings** structures as shown in Figure 14. In addition to that the EEPROM contains a unique, pre-defined and pre-programmed **string ID**, which is used to identify the BGT24LTR11 Shield board.

When the Radar Baseboard XMC4700 boots up, the firmware detects if a radar shield is plugged into the connectors. If the radar shield is plugged in correctly, it reads the information in the EEPROM’s memory to determine what kind of shield is plugged into the interface. The power supply of the radar shield is only enabled when the shield is correctly identified.

Firmware description

After that, the firmware reads the hardware and the algorithm settings from EEPROM. In case of valid settings the firmware will start based on these new settings; otherwise it will start with the default settings and overwrite the EEPROM settings section with the default settings values.

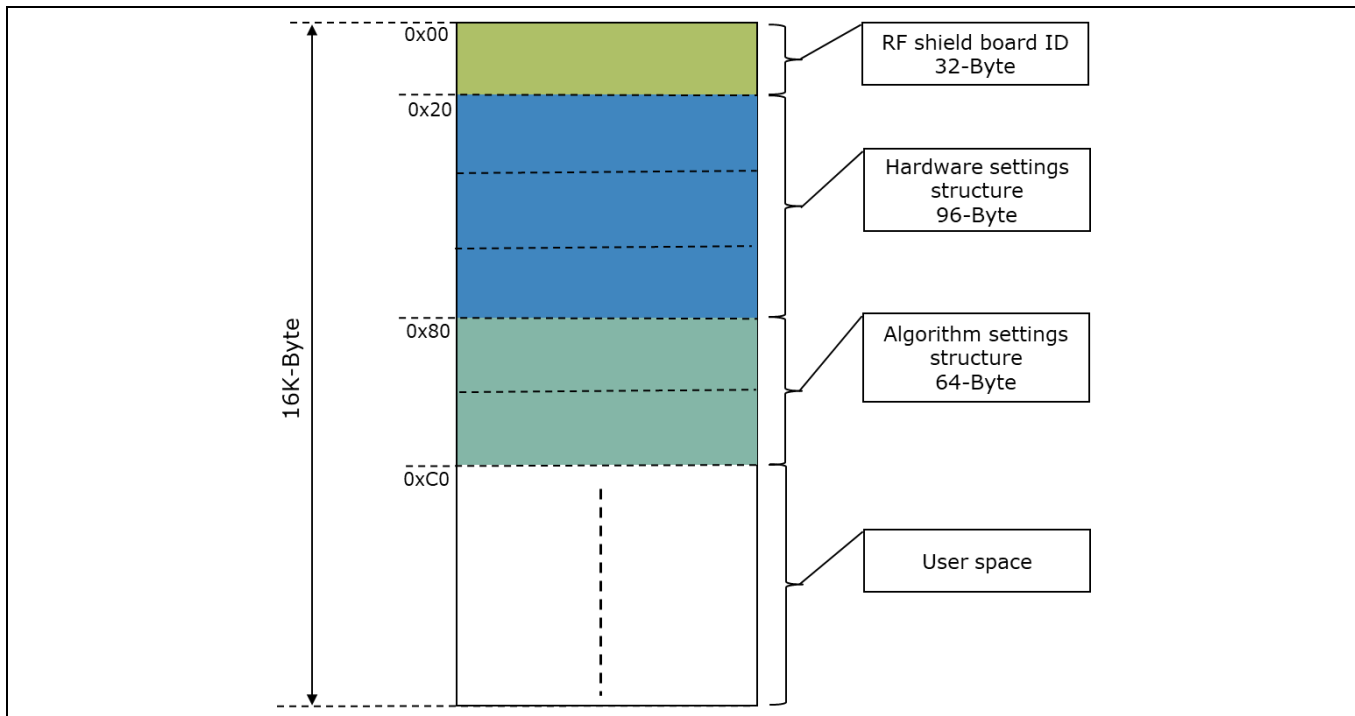


Figure 17 EEPROM structure

4.6 DAVE™ project overview

The Sense2GoL Pulse 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 users’ quick reuse and customization.

Table 2 lists the DAVE™ apps used, based on the Sense2GoL Pulse board, to generate the appropriate firmware source code.

Table 2 DAVE™ project apps used

| DAVE™ app | Number of instances | App description |
|------------|---------------------|---|
| ADC | 4 | Allows for digitizing analog signals using ADC via queue and scan request sources with advanced features |
| CMSIS_DSP | 1 | Provides the CMSIS DSP software library, a suite of common signal processing functions to apply on Cortex-M processor-based devices |
| DIGITAL_IO | 4 | Used to configure a port pin as digital input/output |
| DMA_CH | 4 | Used to perform single- and multi-block data transfer using the General Purpose Direct Memory Access (GPDMA) module on the XMC4000 |
| I2C_MASTER | 1 | Used for I ² C serial interface communication |

Firmware description

| DAVE™ app | Number of instances | App description |
|-----------|---------------------|--|
| INTERRUPT | 2 | Enables overwriting of the Interrupt Service Routine (ISR) provided in the system file and sets the interrupt priority |
| SYSTIMER | 1 | Uses the SysTick interrupt to call user functions periodically at a specified rate after a given time period expires |
| TIMER | 4 | Provides an accurate timer by using the hardware CCU timer; this can be used as a trigger input to other peripherals or to create an event |
| USBD_VCOM | 1 | USB virtual COM port application. This app implements the VCOM over USB CDC class driver |
| PWM | 4 | Used to generate PWM control signals |

DAVE™ apps are configured to address the XMC4700 peripherals to ensure communication, data processing and result visualization. The fundamental functionalities are PWM, timer, ADC and DMA.

Table 3 PWM and GPIO pin configurations

| Pin name | Pin functionality | Description |
|----------------------------|-------------------|--|
| PWM control signals | | |
| BGT_VCC | PWM | P5.11 pin: Power Enable/disable V _{CC} 3.3 V power for BGT24LTR11 |
| BGT_TX_ON | PWM | P5.10 pin: Control Control signal to enable/disable output power for BGT24LTR11 |
| BGT_VCC_PTAT | Output | P0.14 pin: Power Provide 3.3 V for PTAT voltage source for BGT24LTR11 |
| Sample_Hold | PWM | P5.9 pin: Control Control signal to enable/disable S&H |
| LED configuration | | |
| BLUE_LED | Output | P1.13 pin: Control Turn on/off LED |
| GREEN_LED | Output | P1.14 pin: Control Turn on/off LED |
| RED_LED | Output | P1.15 pin: Control Turn on/off LED |

- **ADC configuration** – The XMC4700 integrated 12-bit ADC is used to sample and process the analog down-converted signals in the baseband. The ADC peripheral configuration is set to:
 - conversion mode: 12-bit resolution
 - four channels, two channels for each IF (IF_I_HG, IF_Q_HG, IF_I_LG, IF_Q_LG), with two bytes per sample for each ADC channel
 - sample time [ns]: 75.

The ADC mode has an impact on the maximum possible sample rate. Higher resolution reduces the maximum sample rate. Figure 18 shows the configuration of the ADC DAVE™ app:

Firmware description

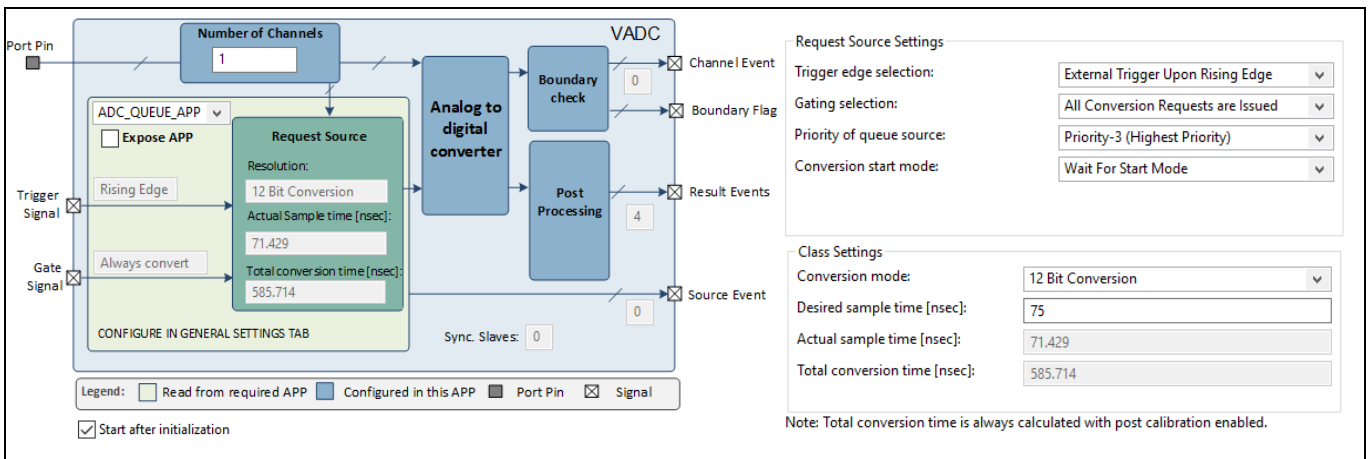


Figure 18 ADC DAVE™ app configuration

- **DMA_CH** – The XMC4700 GPDMA peripheral is configured to transfer data from the ADC peripheral to XMC™ memory. Four DMA channels are configured for RX data measurements (Q_HG, I_HG, Q_LG and I_LG), as follows:
 - Transfer type: single block
 - Transfer flow: peripheral to memory (DMA flow controller)
 - Block size: 250
 - Transfer width: 16 bits
 - Burst width: 1 word
 - Handshaking: hardware

Figure 19 shows the configuration of the DMA_CH DAVE™ app.

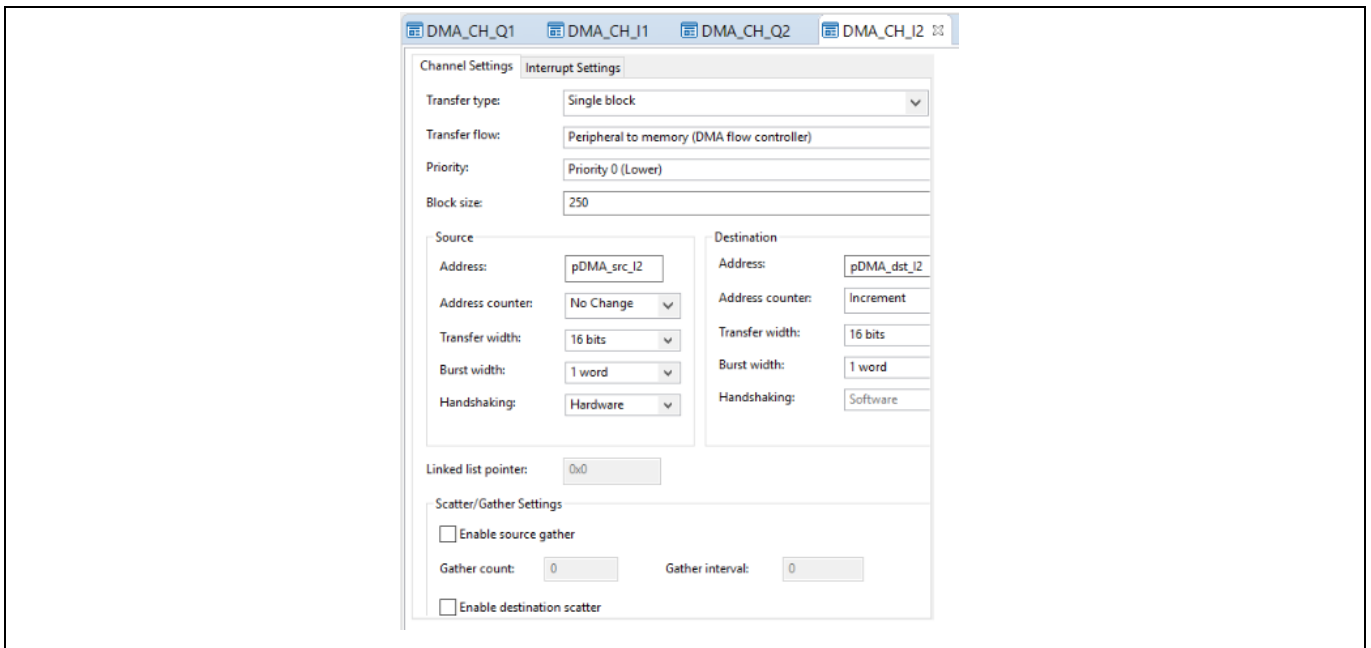


Figure 19 DMA DAVE™ app configuration

4.7 Firmware package overview

Sense2GoL Pulse is a firmware package for XMC™ microcontrollers and BGT24LTR11 radar chips. It provides a complete solution to build radar applications in a single package containing the source code for various exemplary applications facilitating the development of user applications. Figure 20 shows a top-level view of the Sense2GoL Pulse package file structure.

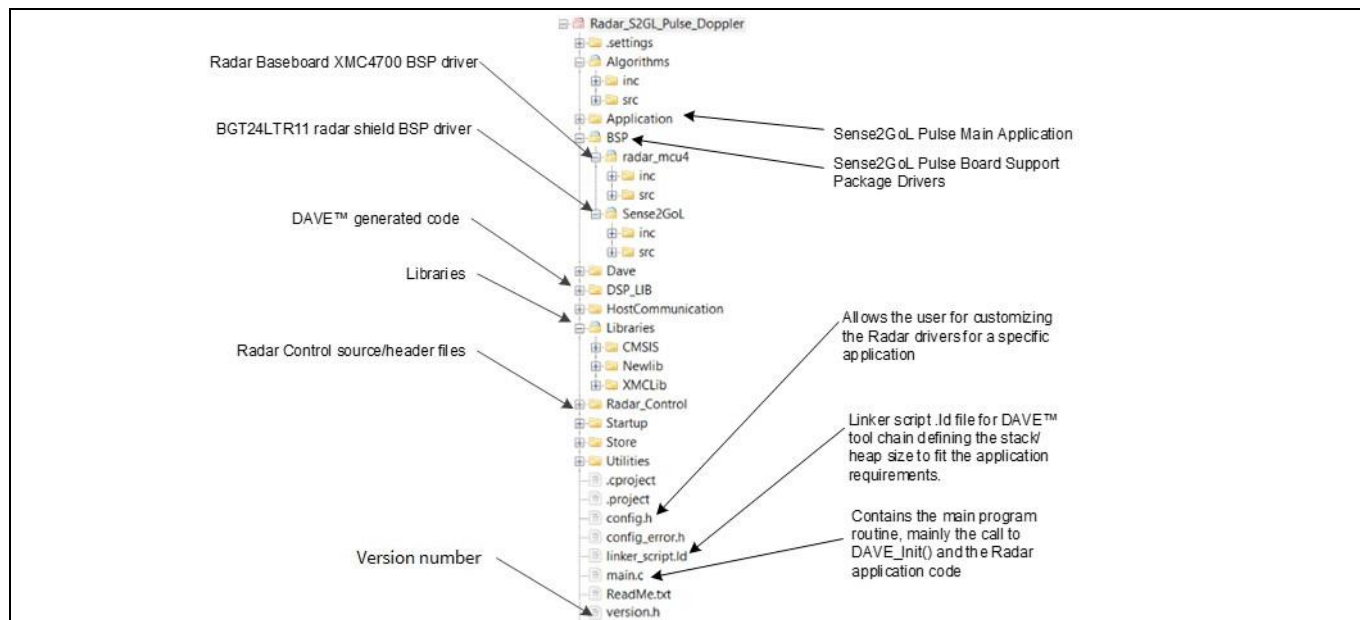


Figure 20 Package folder structure

4.8 Footprint

The purpose of the following sections is to provide the memory requirements for all the Sense2GoL Pulse firmware modules, including devices’ drivers, algorithms and main radar applications. The aim is to have an estimation of fixed and customizable memory requirements in case of removal or addition of a module or feature. The footprint data are provided for the following environments:

- **Board** – Sense2GoL Pulse Kit v1.2
- **Firmware** – S2GL_Pulse (V1.0.x)
- **Toolchain** – DAVE™ v4.4.2

After building a project, the build result is displayed in the console window, where the code size figures are listed. The values are organized according to memory areas, arranged by the linker file (*.ld) into the text, data and bss sections. Table 4 shows the Sense2GoL Pulse build memory utilization for the radar firmware configurations, main modules and algorithms. The information has been gathered by analyzing the corresponding (*.elf) file.

Table 4 Sense2GoL Pulse firmware footprint

| Firmware | | | Footprint | | | |
|----------|---------------------|---------------------|----------------------------|-------------|---------------------------|-----------------------|
| Version | DAVE™ project | Optimization | Text ⁽¹⁾ [byte] | Data [byte] | bss ⁽²⁾ [byte] | Total [byte] |
| v1.0.0 | S2GL_Pulsed_Doppler | None (-O0) | 104920 | 1536 | 39224 | 145680 byte (0x23910) |
| | S2GL_Pulsed_Doppler | Optimize most (-O3) | 73400 | 1516 | 39232 | 114148 byte (0x1bde4) |

⁽¹⁾ Text: code.

⁽²⁾ bss: statically allocated variables that are not explicitly initialized to any value.

4.9 Firmware timings

This section presents the typical timings of the Sense2GoL Pulse firmware that should be used to ensure correct execution of the important radar application sequences.

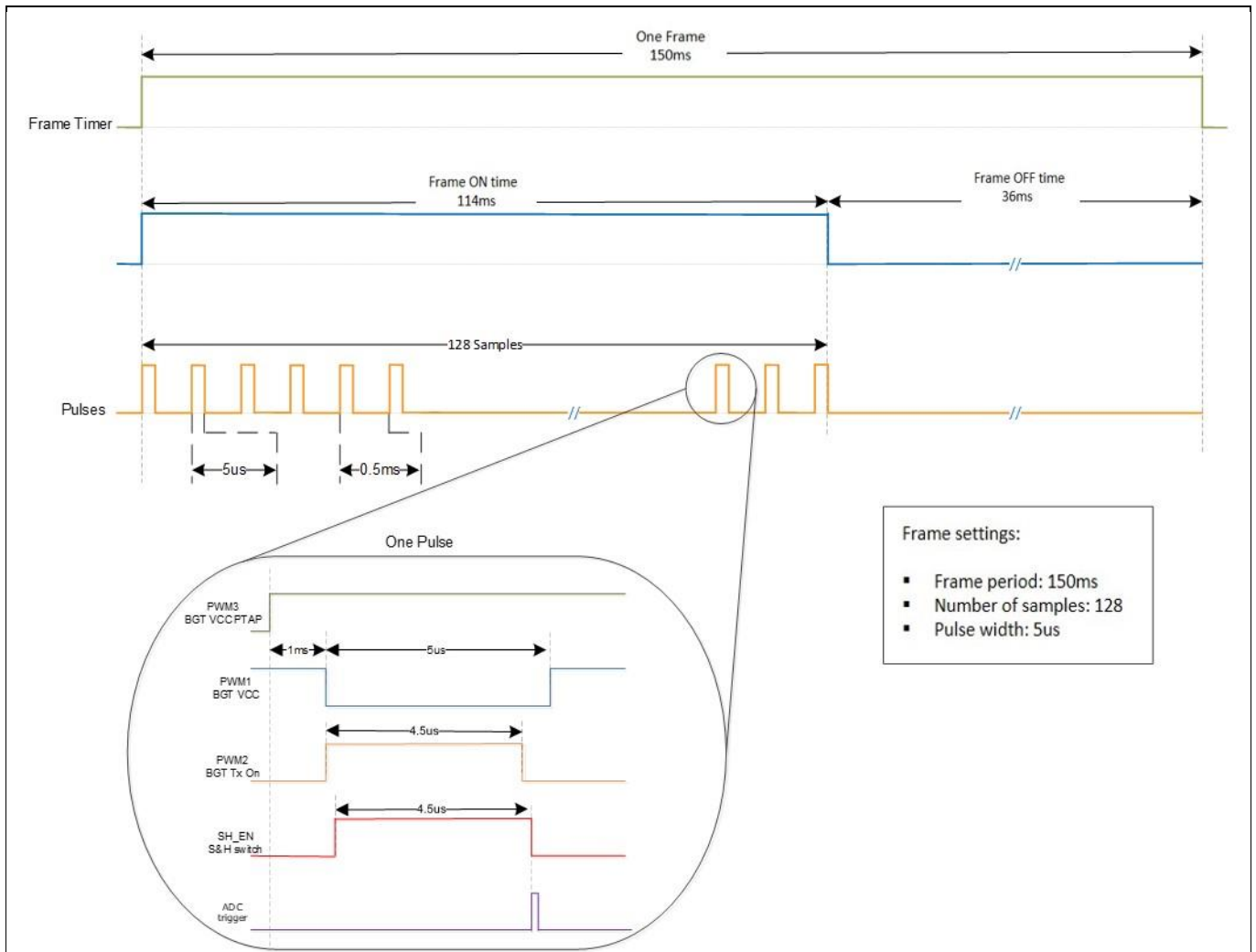


Figure 21 Raw data acquisition timings

4.10 Firmware customization and configuration

The configuration file *config.h* allows for customizing the drivers and algorithms for the Sense2GoL Pulse radar application. The following parameters can be configured: enable/disable or modify some options by uncommenting/commenting or modifying the values of the related define statements, as described in Table 5.

Table 5 Define statements used for radar firmware configuration

| Parameter | Description | Default | Valid range |
|--------------------------------------|--|---------|----------------|
| General configurations | | | |
| NUM_OF_CHIRPS | In the FW is only one “chirp” used | 1 | [1] |
| SAMPLES_PER_CHIRP | Size of IQ raw ADC buffer | 128 | [64 to 512] |
| FRAME_PERIOD_MSEC | Time period of one frame to capture data (units in ms) | 150 | [50 to 2000] |
| SAMPLE_SKIP_COUNT | Default number of samples skipped for signal processing at beginning of frame | 100 | [0 to 1000] |
| Doppler configurations | | | |
| DOPPLER_SUPPORTED | Comment/uncomment this macro, to enable/disable FMCW support | - | N.A. |
| DOPPLER_SAMPLING_FREQ_HZ | Sampling frequency (units in Hz) | 2000 | [1000 to 5000] |
| MINIMUM_SPEED_KMPH | Filter out targets below this speed (units in km/h) | 0 | [0.0 to 10] |
| MAXIMUM_SPEED_KMPH | Filter out targets above this speed (units in km/h) | 10 | [0.1 to 10.0] |
| SPEED_DIVIDER | For software implementation purpose only to keep min. and max. speeds in limits | 100 | |
| SPEED_DETECTION_THRESHOLD | FFT spectrum threshold to detect speed of a target in Doppler | 50 | [0 to 5000] |
| MOTION_DETECTION_THRESHOLD | FFT spectrum threshold to detect a target in Doppler | 20 | [0 to 5000] |
| GAIN_SELECTOR_DOPPLER | Baseband gain selection for doppler algorithm 1 for high gain, 0 for low gain | 1 | [0 or 1] |
| Pulsed Doppler configurations | | | |
| PULSE_WIDTH_USEC | On-time for the gate of BGT V_{CC} input; 1 percent* of sample period | 5 | [4 to 10] |
| SAMPLE_HOLD_ON_TIME_NSEC | Hold time for acquiring raw data (units in ns) | 0 | [0 to 4520] |
| SAMPLE_HOLD_ON_DELAY_TIME_NSEC | Delay time for holding signal after turning on BGT V_{CC} signal (units in ns) | 480 | 480 |
| SAMPLE_HOLD_START_DELAY_TIME_NSEC | Slightly shift the start holding signal after turning on BGT V_{CC} signal (units in ns) | 400 | 400 |

Firmware description

| Parameter | Description | Default | Valid range |
|---------------------------|---|---------|-------------|
| PTAT_DELAY_USEC | The delay between PTAT on-signal and the start of the frame sampling | 1000 | ≥1000 |
| PTAT_SIGNAL_SWITCHED | Set to 1 if PTAT signal should be switched, 0 avoids switching of PTAT signal | 1 | [0 or 1] |
| DSP configurations | | | |
| FFT_SIZE | FFT length for FFT processing | 256 | 256 |

5 Algorithm description

This section describes the implemented algorithms for presence, motion and movement detection.

5.1 Target approaching/target departing

This flag outputs the direction of the target's movement in the current frame. A flow chart of this algorithm is depicted in Figure 22. In the first step, a **Chebyshev window** is applied on all samples of the processed frame to suppress side lobes. For less than 256 samples per frame, the windowed data is zero-padded to improve the received signal characteristics.

Next, a complex Fast Fourier Transform (FFT) with a size of 256 is computed and the two-sided amplitude spectrum is calculated, on which a peak search is applied. The corresponding frequency of the peak value is used to calculate the velocity of the target, and the sign of the frequency defines the direction of movement.

If the peak value is above a predefined threshold, movement is detected and the dedicated indicator for "Target approaching" or "Target departing" is enabled.

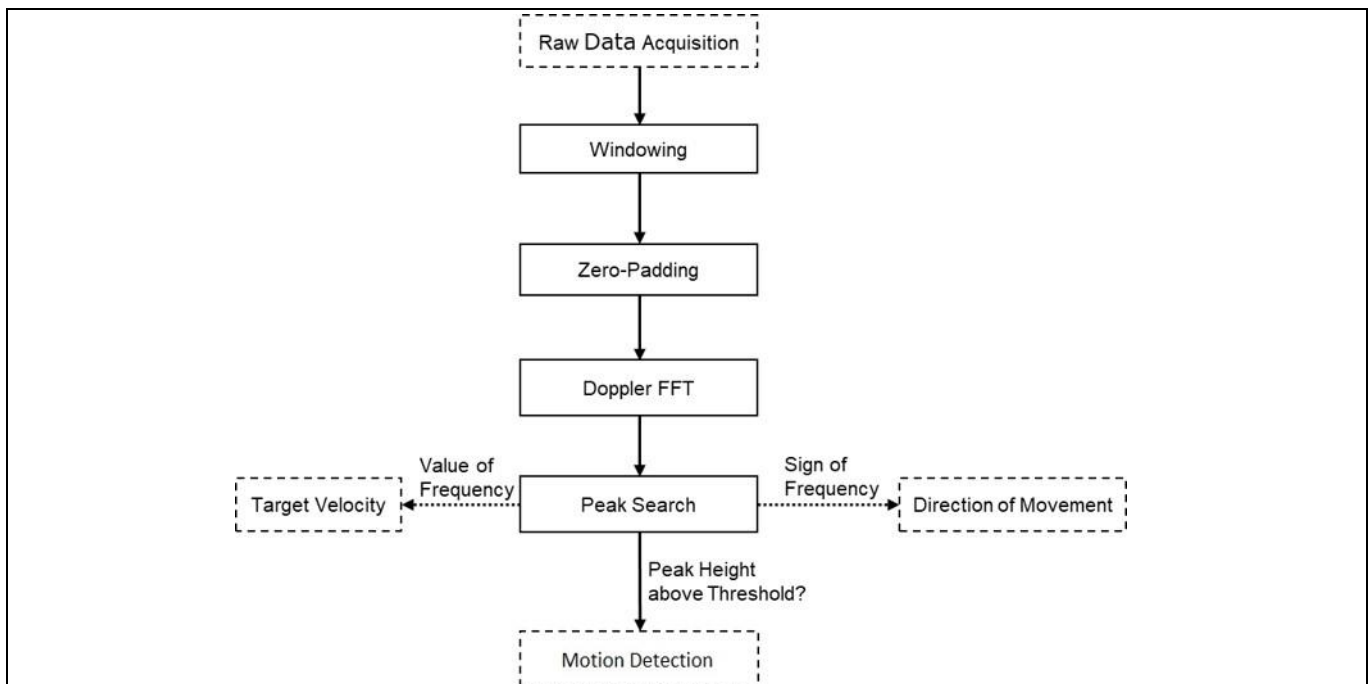


Figure 22 Movement detection algorithm flow

5.2 Motion detected

The motion detection in uses the processing of the direction evaluation, but with a dedicated independent threshold, that is usually lower than the direction detection threshold, for two reasons:

- For smaller movement the direction is often not distinct into a single direction
- Even smaller movement show a increased level in the FFT spectrum, that is significant above the noise floor
- The motion could be usually detected at a higher range than the direction of movement.

6 References

- [1] Infineon BGT24LTR11 – 24 GHz radar IC – [datasheet](#)
- [2] Infineon XMC4700 32-bit ARM Cortex™-M4 microcontroller – [datasheet](#)
- [3] Infineon application note – [AN305 – “user’s guide to 24 GHz radar transceiver”](#)
- [4] Infineon application note – [AN553 – “24 GHz transceiver: BGT24LTR11”](#)

Revision history

Revision history

| Document version | Date of release | Description of changes |
|-------------------------|------------------------|-------------------------------|
| V1.0 | 07/02/2020 | Initial version |

Trademarks

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

Edition 07/02/2020

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2020 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

<UM_1905_PL32_1905_111857>

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, 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.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

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 reasonably be expected to result in personal injury.