BGT60TR13C Operating Principle Introduction

Lorenzo Mori (IFI PSS SMD EMEA)
January 2022

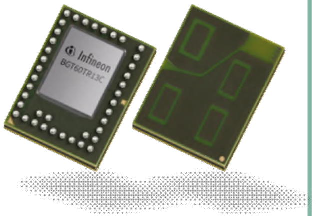
Table of Contents

Table of Contents

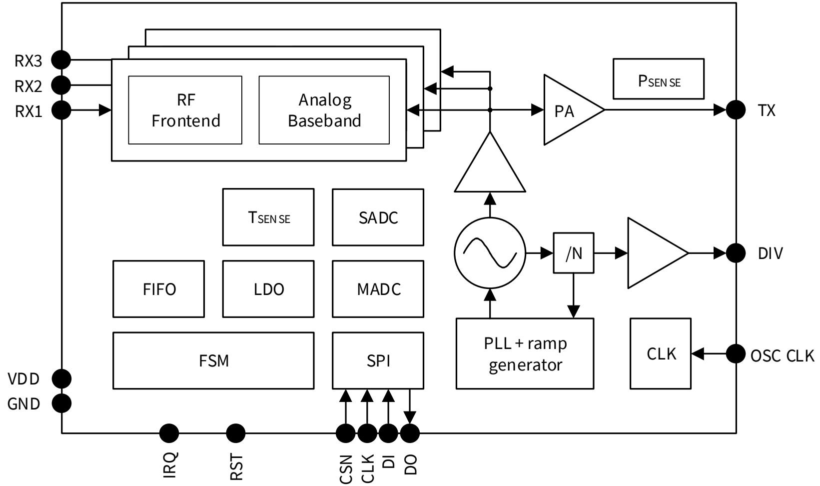
BGT60TR13C – The most compact and efficient 1Tx-3Rx UWB-FMCW solution on the market with Integrated Antennas (AIP)

Value proposition

- Small Size : only 6.5 x 5.0 mm^2
- Low Power Consumption : <5mW (duty cycled)
- High Sensitivity : detect sub-mm movement for Vital Sensing
- High Bandwidth: >5GHz leading to good range resolution ~3cm
- High SNR : Can detect people up to 10m
- Fast Ramp speed: 400MHz/µs
- FSM: finite state machine for flexible configuration of the Chip to manage modulation and power modes



Block Diagram

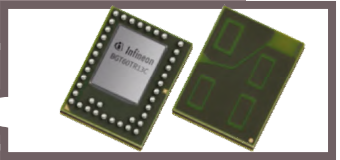


Main Use Cases

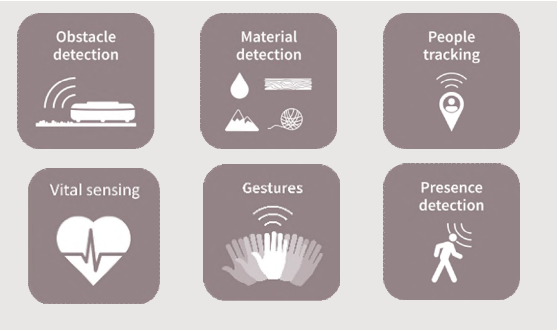
Potential applications

BGT60TR13C: Enables radar sensing in smart home, elderly/health care and IoT applications

Feature	USP
Very fast ramp speed of 800MHz/µs	Measures accurately down to very low distances of 1cm
High SNR	Can detect people up to 15m (front facing)!
High Bandwidth >5GHz	Very good range resolution of down to ~3cm
High sensitivity	Detects sub-mm movements
Package: 5 x 6.5mm	Small package size
<5mW (duty cycled)	Low power consumption
FSM	Finite state machine for flexible configuration of the Chip to manage modulation + power modes



Use cases



Applications

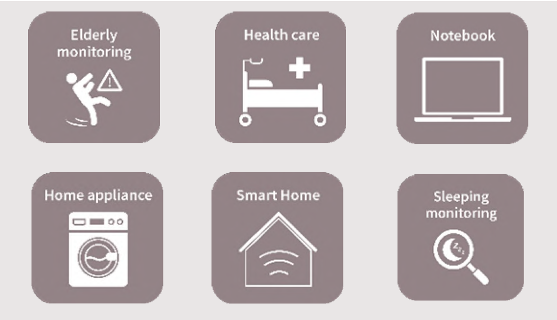
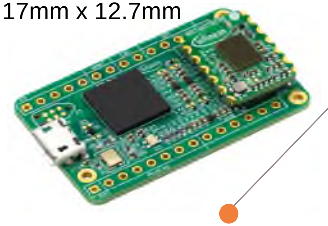


Table of Contents

BGT60TR13C Development Kit and Reference Design

DEMO 60GHZ TR13CAIP

17mm x 12.7mm



Features

Presence Detection	Detection of stationary and moving targets
Segmentation	Indication of pre-defined segment in which presence was detected
GUI	Visualization of use cases and radar features, such as range-doppler. Change user settings to obtain the desired use case
Modular/ Portable Approach	Access the device, radar features such as the FFT or use case specific features such as the tracking layer AP
Multiple platforms	Accessibility granted through C, Matlab and Python

Radar Development Kit (RDK) Tool offers BGT60TR13C documents, hardware specifications, GUI as well as a software library

Documentation

- Release Notes
- User Manuals
- Product Brief
- Datasheet
- Design-Material (Schematic, Layout)

Radar GUI

- Live data plotting
- Application visualization
- Sensor configuration

Hardware

- Altium Project
- Schematics
- Bill of Material
- Gerber data

Software Library

- Firmware binary and source code
- Python wrapper for data acquisition
- Matlab wrapper for data acquisition

SP No.	OPN No.
SP005339490	DEMO60GHZTR13CAIPTOBO1

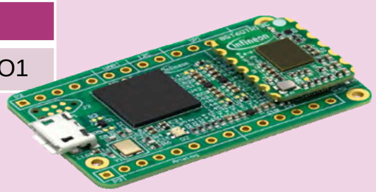


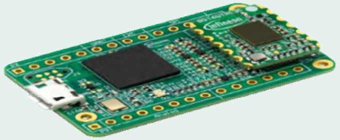
Table of Contents

Interested in ordering a BGT60TR13C demoboard? Please conteact your respective sales person & follow the below mentioned steps!

1) Order Demboard

- Order your required XENSIV™ 60GHz radar demoboard: To get the BGT60TR13C demoboard use the following OPN No:

SP No.	OPN No.
SP005339490	DEMO60GHZTR13CAIPTOBO1



2) Download Infineon Toolbox

- Download the Infineon Toolbox: https://www.infineon.com/toolbox
- Sign-in to the Infineon Toolbox
- Ensure that you‘re always logged-in

3) Download RDK Tool

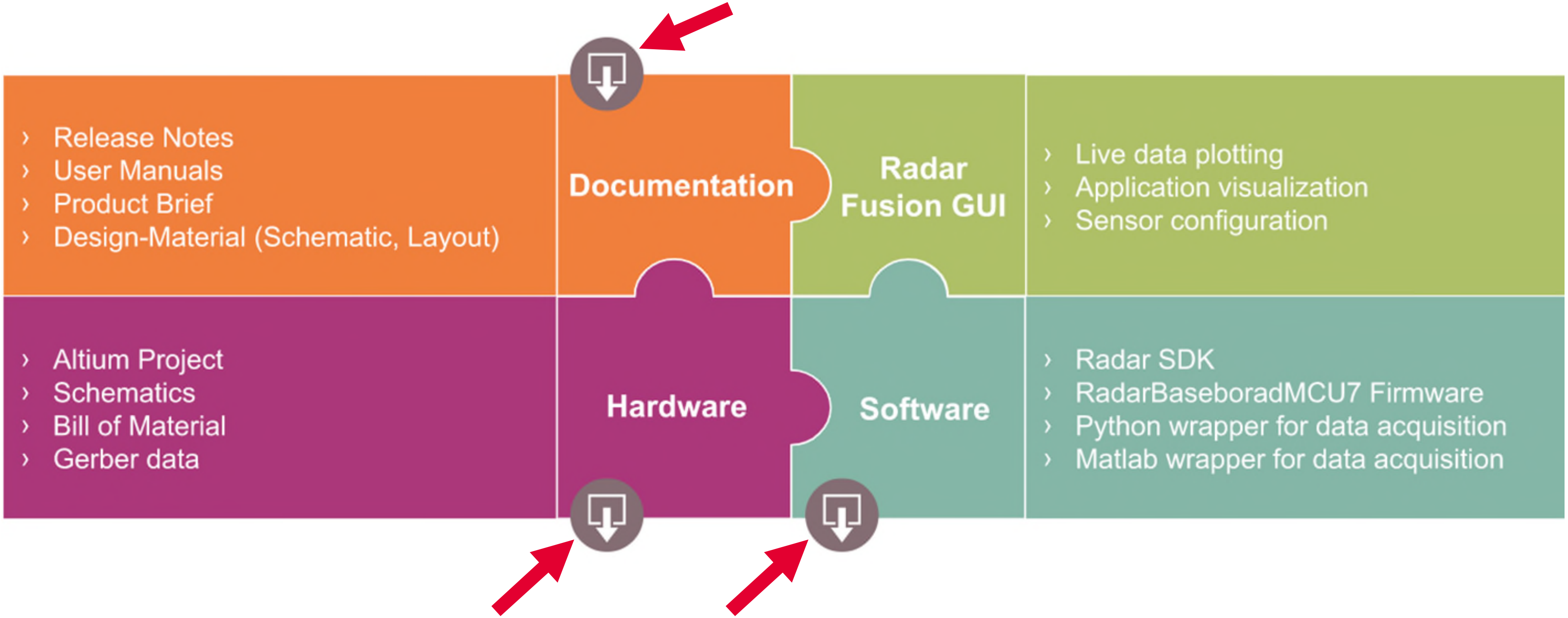
- Search in the tool finder for: ***Radar Development Kit***
- Download the Radar Development Kit Tool
- Next slide-set will provide a detailed overview of the RDK content

Curious to learn more about BGT60TR13C Hardware & Software offering? Watch this training!

Table of Contents

Documentation provided along Radar SDK

- Open the Radar SDK program and download all the three available files:



Content of each package

Hardware

- Schematics of the Radar Board and base board
- Altium Projects for PCB development or modifications

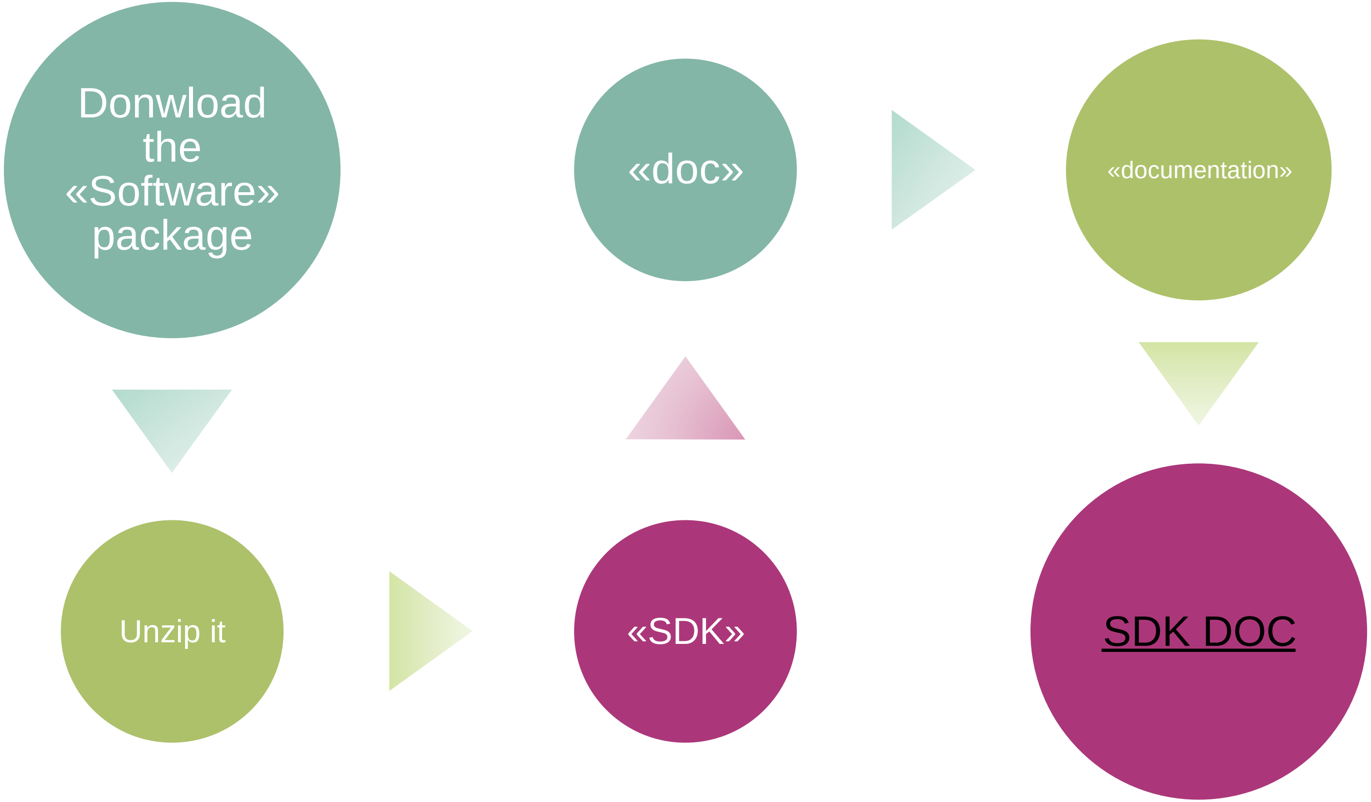
Documentation

- Application Notes of baseboard MCU7 and sensor BGT60TR13C
- Datasheets

Software

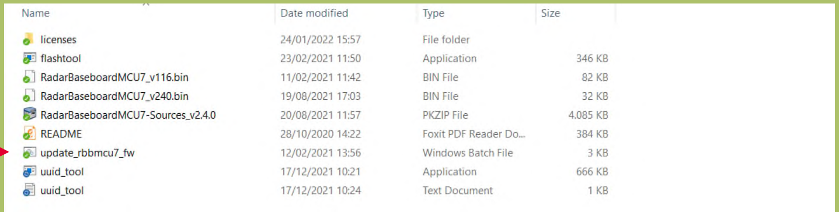
- Updated firmware for baseboard MCU7
- Wrappers for baseboard programming with either MATLAB or Python
- Documentation on the sensor operation and on the «ready to use» algorithms provided by Infineon Technologies
- Algorithms of Presence Sensing™ and of Segmentation™

Documentation location



Firmware Update

Prior to the use of the Radar GUI™ a firmware update of the baseboard is necessary. In the relative folder launch the .batch file: update_rbbmcu7_fw.batch



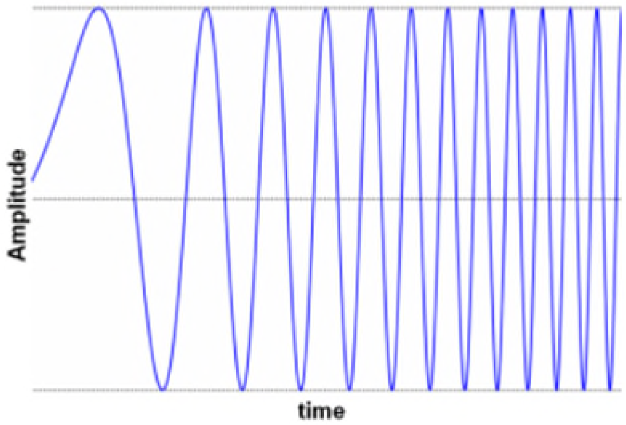
Name	Date modified	Type	Size
licenses	24/01/2022 15:57	File folder	
flashtool	23/02/2021 11:50	Application	346 KB
RadarBaseboardMCU7_v116.bin	11/02/2021 11:42	BIN File	82 KB
RadarBaseboardMCU7_v240.bin	19/08/2021 17:03	BIN File	32 KB
RadarBaseboardMCU7-Sources_v2.4.0	20/08/2021 11:57	PKZIP File	4.085 KB
README	28/10/2020 14:22	Foxit PDF Reader Do...	384 KB
update_rbbmcu7_fw	12/02/2021 13:56	Windows Batch File	3 KB
uuid_tool	17/12/2021 10:21	Application	666 KB
uuid_tool	17/12/2021 10:24	Text Document	1 KB

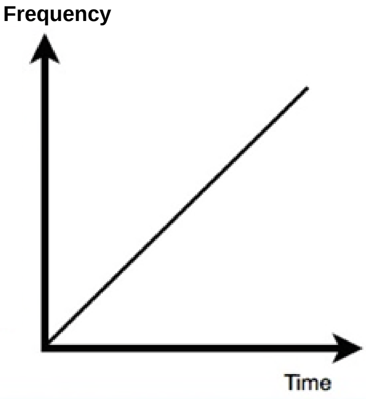
Select the .bin file having the newer version (v2.40 as of 24/01/2022). At this point it is possible to launch the Radar GUI™ and also the new Radar Fusion GUI™.

Table of Contents

Operating principle of the sensor: FMCW

- **F**requency
- **M**odulated
- **C**ontinuous
- **W**ave





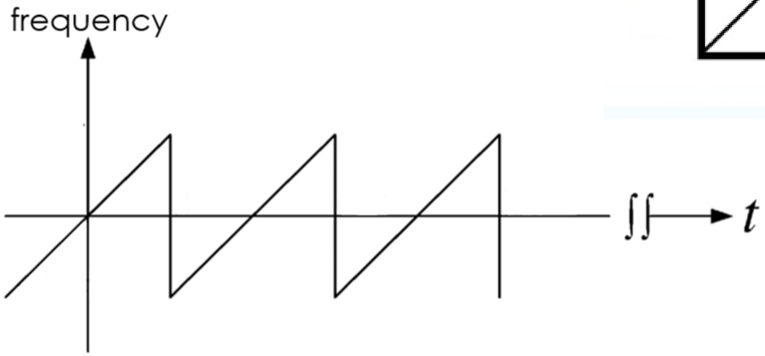
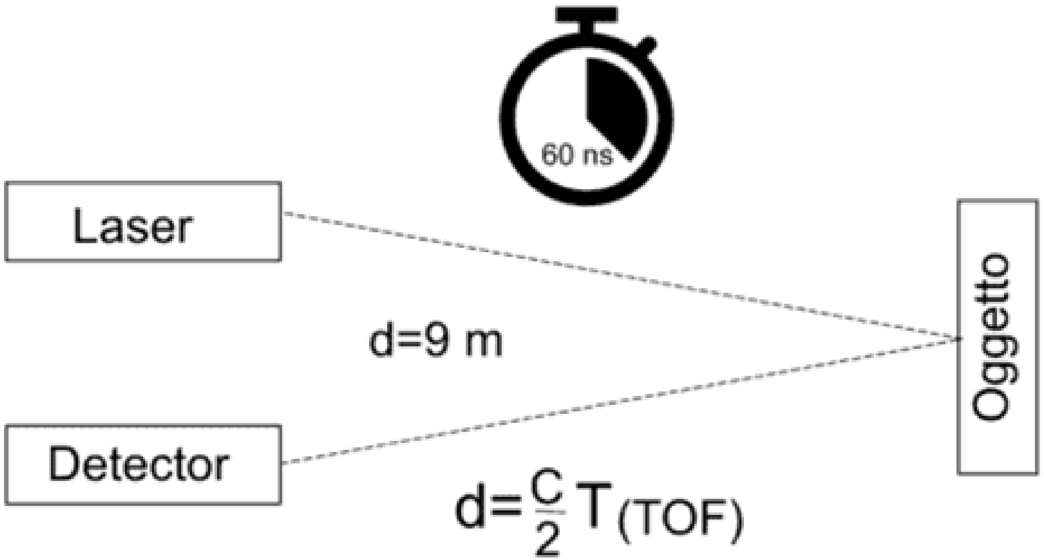


Table of Contents

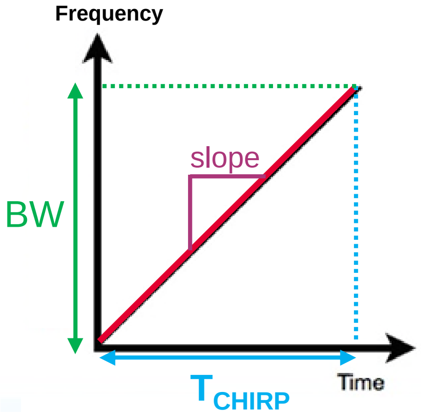
Ranging – 1

- As in every ranging application implemented with radar the distance is measured through the measurement of the time of flight of the E.M. Wave:

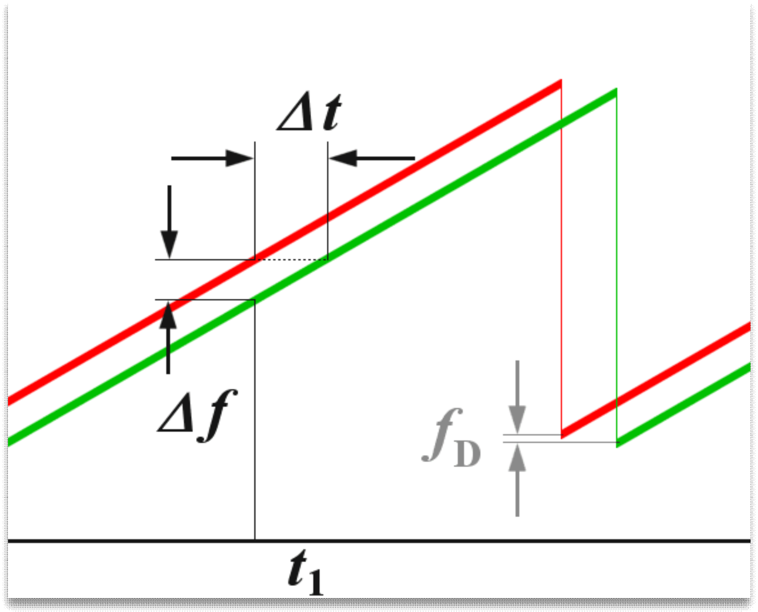


Ranging – 2

- In the case of FMCW emission signal is composed by frequency «chirps»:



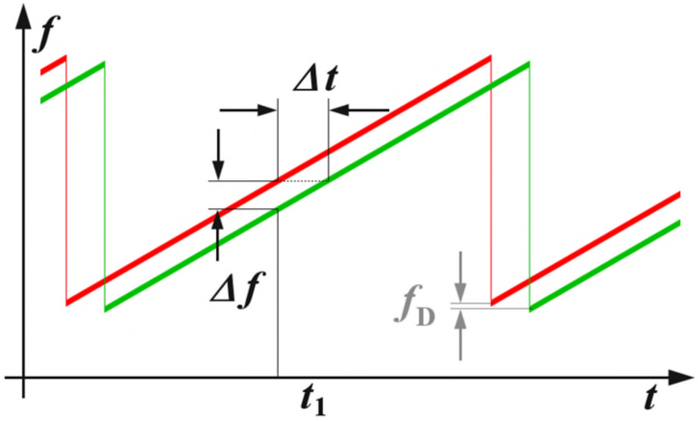
- BW = Chirp BandWidth
- T_{CHIRP} = Chirp duration

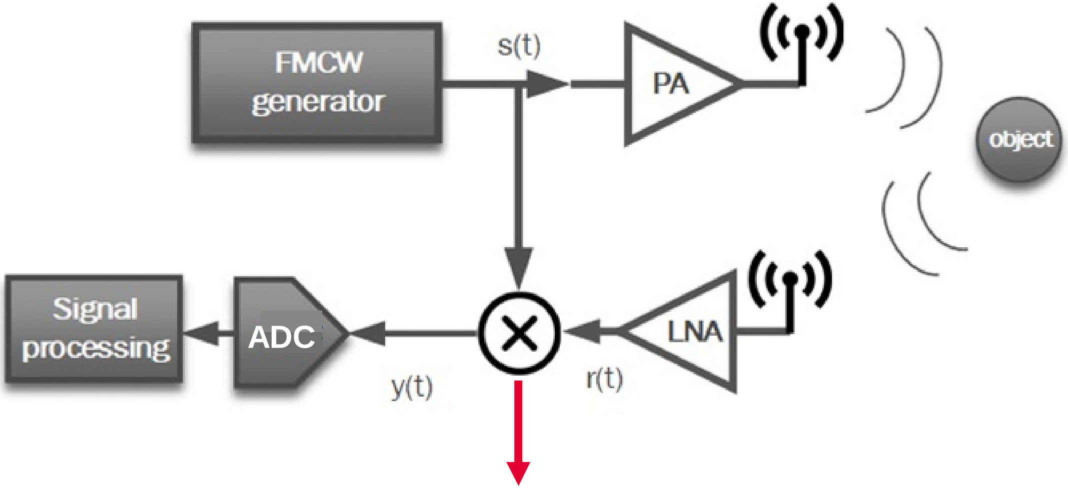


- To measure time of flight is thus necessary to measure the frequency difference between the transmitted and received chirp.
  - $\Delta t = \Delta f \cdot \frac{1}{slope} = \Delta f \cdot \frac{T_{CHIRP}}{BW}$

Ranging – 3

- The subtraction of the frequencies of transmitted and received signals is performed by a mixer:





- The signal converted by the ADC is thus already «dechirped», that is at the frequency IF

$$\Delta f \triangleq f_{IF} = f_{TX} - f_{RX}$$

Ranging and resolution

- The range of the object is thus obtained as:

$$R_{obj} = \frac{c}{2} \cdot \Delta t_{Time\ of\ flight} = \frac{c}{2} \cdot f_{IF} \cdot \frac{T_{chirp}}{BW}$$

- It follows that the resolution of the ranging is directly related to the bandwidth of the chirp. Thanks to the 60GHz frequency of operation the BGT60TR13C radar can achieve a chirp Bw of 5.5GHz (58 →63.5 GHz) that is directly translating into a resolution:

$$\Delta R_{res} = \frac{c}{2} \cdot f_{IF,min} \cdot \frac{T_{chirp}}{BW} = \frac{c}{2} \cdot \frac{1}{T_{chirp}} \cdot \frac{T_{chirp}}{BW} = \frac{c}{2 \cdot BW} \cong 2.73\ cm$$

Maximum measurable range

- The maximum measurable range is instead limited by the ADC sampling rate and by the chirp duration T_{chirp}:

$$R_{max} = \frac{c}{2} \cdot f_{IF}^{max} \cdot \frac{T_{chirp}}{BW} = \frac{c}{2} \cdot \underbrace{\frac{f_{sampling}^{max}}{2}}_{*} \cdot \frac{T_{chirp}}{BW}$$

- As we can see the maximum range is set by the bandwidth of the chirp, the sampling rate of the ADC and the chirp duration.
- The BW is already set by the desired resolution so what we'll have to set are the $f_{sampling}$ of the ADC and the chirp duration.

$$R_{max} = \frac{c}{2} \cdot \frac{f_{sampling}^{max}}{2} \cdot \frac{T_{chirp}}{BW} = \frac{c}{2} \cdot \frac{f_{sampling}^{max}}{2} \cdot \frac{N_{samples}}{f_{sampling}^{max}} \cdot \frac{1}{BW} = \frac{c}{4 \cdot BW} \cdot N_{samples}$$

Disclaimer – 1

$$R_{max} = \frac{c}{2} \cdot f_{IF}^{max} \cdot \frac{T_{chirp}}{BW} = \frac{c}{2} \cdot \underbrace{\frac{f_{sampling}^{max}}{2}}_{*} \cdot \frac{T_{chirp}}{BW}$$

* This is a physical limit established by the Shannon-Nyquist sampling theorem. A good approximation is to consider a maximum frequency of the input signal equal to a tenth of the maximum ADC sampling frequency in order to avoid aliasing issues.

(Notice that this is <u>not</u> reported in the SDK documentation)

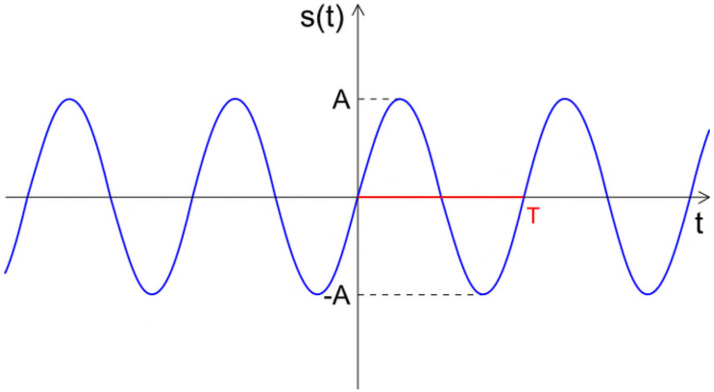
Disclaimer – 2

› The output of the BGT60TR13C sensor is obviously a time varying function. It is thus necessary to perform the usual operations of windowing, zero_padding and Fast_Fourier_Trasform prior to the ranging one.

Table of Contents

Phase of the IF signal – 1

- We’ve seen how from the IF signal it is possible to detect the range of a static or moving object with a resolution of ~2.7cm. In order to detect smaller displacements and to detect the speed of an object is necessary to extract the phase of the IF signal instead.
- From a theory standpoint the IF signal is a single harmonic at the frequency f_{IF} and it thus has the following shape:



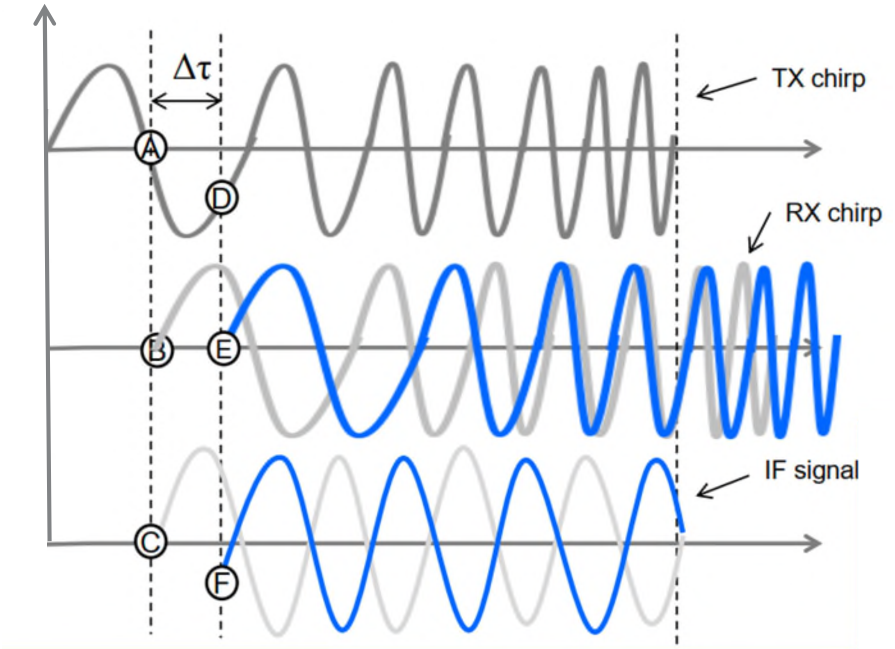
- $s_{IF}(t) = A \cdot \sin(\underbrace{2\pi \cdot f_{IF} \cdot t}) + \underbrace{\varphi})$

As seen previously from the f_{IF} frequency we can compute the range of the object.

The missing information can be extracted from the phase variations.

Phase of the IF signal – 2

- Phase variations $\Delta\varphi$ are directly linked to the sinusoid movement in time and thus to the temporal variation $\Delta\tau$:



- Both TX and RX waves have a frequency $f_{carrier}$. So the phase variation due to a small time variation $\Delta\tau$ is:

$$\Delta\varphi = 2\pi \cdot f_{carrier} \cdot \Delta\tau$$

- We can now express the time variation $\Delta\tau$ as a function of the displacement Δd:

$$\Delta\tau = \frac{2 \cdot \Delta d}{c} \qquad [c \triangleq speed\ of\ light]$$

$$\Delta\varphi = 2\pi \cdot f_{carrier} \cdot \frac{2 \cdot \Delta d}{c} = \frac{4\pi \cdot \Delta d}{\lambda_{carrier}}$$

Phase of the IF signal – 3

- Here is a brief example demonstrating how useful the phase variation can be. For a displacement Δd of just 1mm we get a phase variation of:

$$\Delta\varphi = \frac{4\pi \cdot \Delta d}{\lambda_{carrier}} = 2.51\, rad = 144°$$

- A variation of 144° is easily detectable with a good precision. On the contrary the variation of f_{IF} due to the above-mentioned displacement would be of only:

$$\Delta f_{IF} = \frac{slope \ \cdot 2 \cdot \Delta d}{c} = \frac{{}^{5.5GHz}\!/_{50\mu s} \cdot 2 \cdot 1mm}{3 \cdot 10^8\, m/s} \cong 730\ Hz$$

- This is almost imperceptible, for a f_{IF} of 100kHz the relative variation would only be of the 0,73%.

Speed measurement through the use of 2 chirps

- In order to measure the radial speed (with respect to the surface of the sensor) of an object we can exploit the variation of the phase of the IF signal seen from two consecutive chirps.

- If we emit a second chirp immediately after the first one we can notice that the IF signal will have the same IF signal due to the first (range is unchanged to first order if the T_{chirp} is short enough: us o ms) but this last will have a different absolute phase:

$$\Delta d = v_{obj} \cdot T_{chirp} \quad \Rightarrow \quad \Delta\varphi = \frac{4\pi \cdot v_{obj} \cdot T_{chirp}}{\lambda_{carrier}}$$

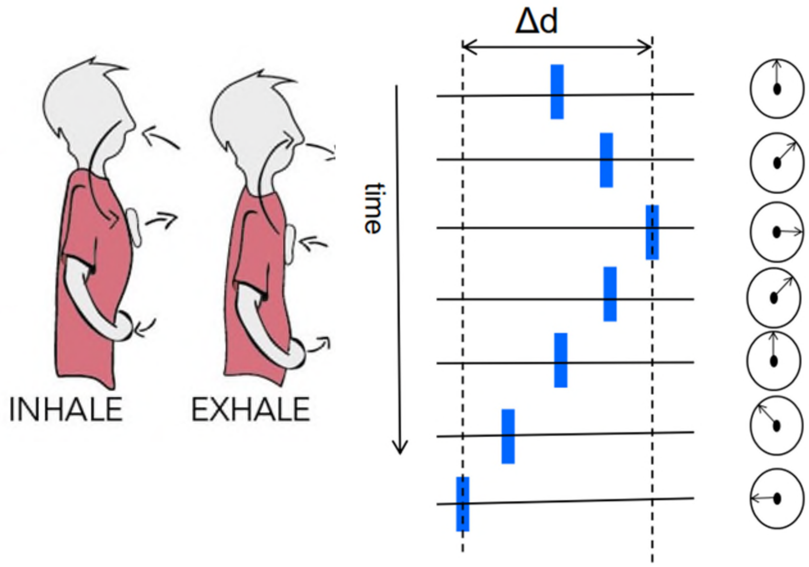
- Speed can thus be extracted from the phase variation:

$$v_{obj} = \frac{\Delta\varphi \cdot \lambda_{carrier}}{4\pi \cdot T_{chirp}}$$

Measurement of oscillatory movements

- Biometrics is particularly interested in the measurement of repetitive movements, with an oscillatory behavior.

- The periodic displacements due to the diaphragm movement in the case of breathing, and of the veins in the case of heart pulse, can be measured thanks to the phase variation observed on multiple chirps.
- Seen that T_{chirp} << period of these oscillations the number of samples per single period can be >10 and grant a correct measurement of the heart rate (or breathing rate).



Maximum measurable speed

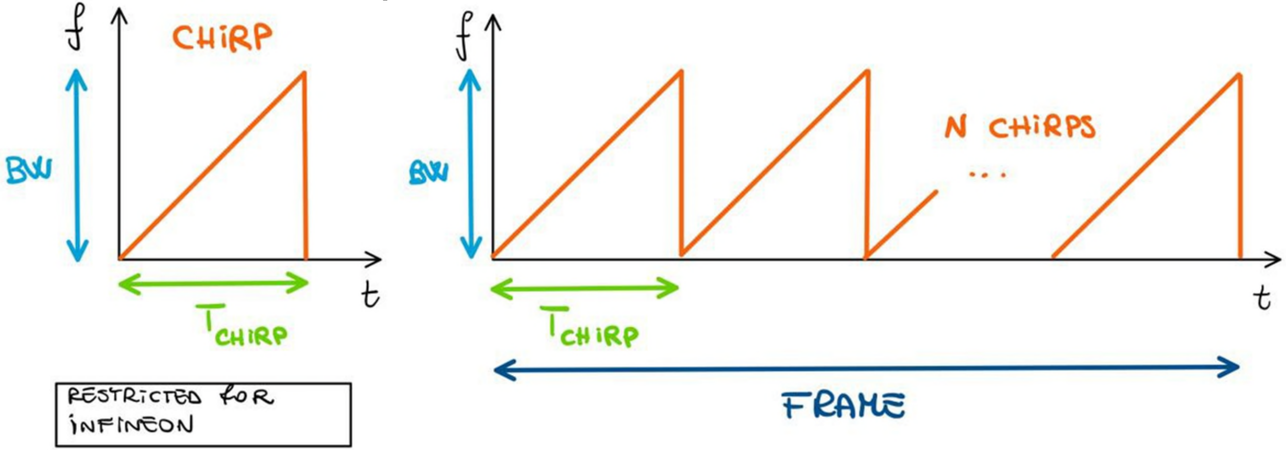
- A maximum speed the system can measure obviously exists. The main limitation is the phase variation due to two consecutive chirps that can not be larger than 180°.
- This limit comes from the fact that it would be otherwise impossible to discern positive and negative speeds (objects approaching/stepping away).

$$\Delta\varphi_{max} = 180° = \pi\ rad \quad \Rightarrow \quad v_{max} = \frac{\cancel{\pi} \cdot \lambda_{carrier}}{4\cancel{\pi} \cdot T_{chirp}} = \frac{\lambda_{carrier}}{4 \cdot T_{chirp}}$$

- We can notice how the T_{chirp} and more in general the PRT (Pulse Repetition Time a.k.a. the time between two consecutive chirps, that can be longer than T_{chirp}) is limiting the maximum measurable speed.

Speed measurement: Doppler FFT – 1

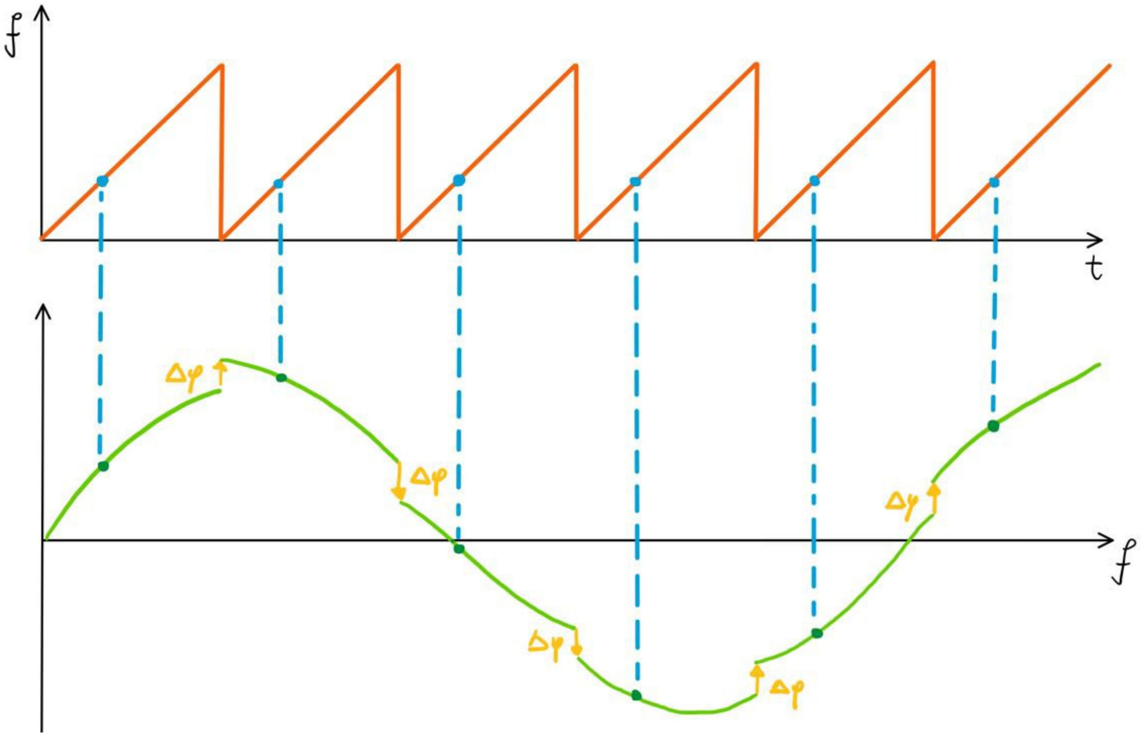
- At this point it should be clear that to measure the speed we need to find a way to measure the phase variations of the IF signal.
- This variation though may not be simple to identify. In order to make this possible we use the Doppler FFT.
- Instead of sending single chirps the radar is used to transmit «frames» that are groups of consecutive chirp, with a well defined number and PRT.



Speed measurement: Doppler FFT – 2

- Let's now suppose to sample every chirp in the same point. To first order, if the PRT is sufficiently low, the only difference between one value of the IF signal and the following one will be a $\Delta\varphi$ variation of the phase:

$$s(k) = A \cdot \sin(k \cdot \Delta\varphi)$$



Speed measurement: Doppler FFT – 3

- The signal extracted in this way (by sampling the IF one with a high $f_{sampling}$) is thus nothing but:

$$s(k) = A \cdot \sin\left(k \cdot \frac{4\pi \cdot v_{obj} \cdot T_{chirp}}{\lambda_{carrier}}\right) \Rightarrow s(t) = A \cdot \sin\left(\frac{4\pi \cdot v_{obj} \cdot t}{\lambda_{carrier}}\right)$$

- It follows that by applying a DFT (summing w.r.t. k, the index of the k^{th} chirp) we can extract the «new f_{IF}», the phase of the sampled IF signal:

$$f_{IF} = \frac{2 \cdot v_{obj}}{\lambda_{carrier}}$$

- By using the FFT along the «chirps dimension» is thus possible to extract v_{obj} .

Speed resolution

- At this point the only thing left to do is to compute the resolution of our speed measurement. Also in this case resolution is strictly related to the resolution of the FFT, but now along the chirp dimension.

- The minimum frequency that can be discerned by the FFT is equal to 1 / the maximum measurable time, that is N° of chirps multiplied by the PRT:

$$f_{IF,min} = \frac{1}{N_{chirps} \cdot PRT}$$

$$\Delta v_{res} = \frac{\lambda_{carrier} \cdot N_{chirps}}{2} = \frac{\lambda_{carrier}}{2 \cdot PRT \cdot N_{chirps}}$$

Parameters choice depending on the project specs

- We are finally listing the results presented in the SDK documentation. These can be used to set the parameters requested by the sensor in order to achieve the desired specs of the project:

  - $$BW_{chirp} = \frac{c}{2 \cdot \Delta R_{res}}$$
  - $$N_{sample\ per\ chirp} = \frac{2 \cdot R_{max}}{\Delta R_{res}}$$
  - $$PRT = \frac{\lambda_{carrier}}{v_{max}}$$
  - $$N_{chirps\ per\ frame} = \frac{2 \cdot v_{max}}{\Delta v_{res}}$$

Part of your life. Part of tomorrow.