



**ScuDo**  
Scuola di Dottorato ~ Doctoral School  
WHAT YOU ARE, TAKES YOU FAR



Doctoral Dissertation  
Doctoral Program in Electrical, Electronics and Communications Engineering  
(XXXIV.th cycle)

# A Real Time Locating System based on TDOA estimation of UWB pulse sequences

**Stefano Bottigliero**

\* \* \* \* \*

**Supervisor**

Prof. Riccardo Maggiore, PhD, Supervisor, Polytechnic of Torino

Politecnico di Torino  
May 2, 2022

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial-NoDerivative Works 4.0 International: see [www.creativecommons.org](http://www.creativecommons.org). The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

.....

Stefano Bottigliero  
Turin, May 2, 2022

# Summary

One of the most popular technologies adopted for indoor localization is Ultra Wide-band impulse radio (IR-UWB). Due to its peculiar characteristics, it is able to overcome the multipath effect that severely reduces the capability of receivers (Sensors) to estimate the position of transmitters (Tags) in complex environments. The architecture of the localization system requires time synchronization among the Sensors by means of very precise, high cost clocking mechanism or by means of a complex high level communication protocol between Sensors and Tags.

In this thesis work, we introduce a new low-cost real-time locating system (RTLs) that does not require time synchronization among Sensors and uses a one-way communication scheme to reduce the cost and complexity of Tags. The system is able to evaluate the position of a large number of Tags by computing the time difference of arrival (TDOA) of UWB pulse sequences received by at least three Sensors in known positions. In the presented system, the Tags transmit sequences of 2-ns UWB pulses with a carrier frequency of 7 GHz. Each Sensor processes the received sequences with a two-step correlation analysis performed first on a field-programmable gate array (FPGA) chip and successively on an on-board processor (ARM). The result of the analysis are the Time Of Arrival (TOA) of the pulse sequence at each Sensor and the Tag ID associated to it. These results are sent to a host PC implementing the trilateration algorithm based on the TDOA computed between couples of Sensors. To compute the 2D position of a Tag it is sufficient to use the TDOA among three Sensors.

Two different applications are developed. The first application, called LOCALization SYSTEM (SILOC in Italian) has the goal of localizing and tracking the position of different Tags inside a localization area with high precision and accuracy. The application is optimized to track moving Tags using a standard operating mode with good accuracy and high responsiveness and super-resolution operating mode with better accuracy but slower responsiveness. The second application, called Package Tracker (PackTrack) integrates new features on top of the SILOC application. It allows the user to monitor Tags attached to valuable goods (e.g. packages) in fixed positions over long periods of time. The application triggers an alarm whenever a Tag moves away from its position by a distance larger than the tolerance and continuously update a data log to keep track of each Tag's movement.

In the following Chapter 1, we first introduce the UWB signal in general and its application to indoor localization, we then briefly describe the most commonly used modulations and which one we adopted. The last part of the chapter introduces the different methods used for indoor localization and describes the details of the pulse sequence adopted in this project.

In Chapter 2 we discuss the design choices and implemented solutions of the developed hardware. We introduce the custom UWB transmitter with its digital and RF components and the UWB receiving chain, made up of a custom UWB antenna and receiver. The chapter concludes with the description of the custom processing board and the system final prototype.

In Chapter 3 we discuss the details of the developed software. The chapter is divided in three main sections: The first section deals with the design of the FPGA firmware implementing a custom architecture able to receive a continuous stream of data, to recognize the presence of a transmitted sequence and to compute its TOA. The second section describes the ARM processor software that uses the information obtained by the FPGA, refines it and associate it to the correct Tag ID. The last section deals with the graphic user interface (GUI) developed to compute the TDOA among Sensors, to perform the trilateration algorithm and plot the localized Tags on a 2D map for user visualization.

The last chapter shows the results obtained with the RTLS system installed in our institution laboratory. The system tracking capabilities and localization accuracy have been evaluated by means of a measurement campaign. The obtained localization accuracy of 10 cm is demonstrated and discussed.





# Acknowledgements

# Contents

<b>List of Tables</b>	IX
<b>List of Figures</b>	X
<b>1 Ultra Wide Band Localization</b>	<b>1</b>
1.1 Introduction to UWB signals . . . . .	1
1.2 UWB Modulations . . . . .	4
1.3 Comparison between UWB localization techniques . . . . .	5
1.3.1 Received Signal Strength Indicator (RSSI) . . . . .	5
1.3.2 Time Of Flight (TOF) . . . . .	6
1.3.3 Time Difference Of Arrival (TDOA) . . . . .	8
1.3.4 Phase Difference Of Arrival . . . . .	9
1.3.5 Existing systems available on the market . . . . .	10
1.3.6 Proposed Implementation . . . . .	11
1.4 Tag transmitted Sequence . . . . .	12
<b>2 Hardware Design</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Ultra Wide Band Receiver . . . . .	18
2.2.1 UWB Receiving antenna . . . . .	18
2.2.2 RF Receiver: Block Diagram . . . . .	20
2.2.3 RF Receiver: PCB Design . . . . .	25
2.3 First Prototype Tag . . . . .	28
2.3.1 Design Overview . . . . .	28
2.3.2 Block diagram . . . . .	28
2.3.3 Radio frequency oscillator . . . . .	35
2.3.4 PCB design . . . . .	38
2.4 Second Low-Power Tag Prototype . . . . .	41
2.4.1 Block Diagram . . . . .	41
2.4.2 Power gating implementation . . . . .	43
2.4.3 Elliptic Dipole Antenna design . . . . .	44
2.4.4 UWB Antenna and RF Oscillator Co-Simulation . . . . .	44

2.4.5	Measured Results . . . . .	46
2.5	Processing board prototype overview . . . . .	52
2.6	Sensor final prototype . . . . .	57
2.6.1	Block diagram . . . . .	58
<b>3</b>	<b>Software Design</b>	<b>61</b>
3.1	Introduction . . . . .	61
3.2	FPGA firmware design . . . . .	62
3.2.1	Introduction . . . . .	62
3.2.2	Block diagram . . . . .	62
3.2.3	LVDS data management . . . . .	65
3.2.4	Correlation . . . . .	68
3.2.5	Manual and Automatic Threshold implementation . . . . .	72
3.2.6	Data management . . . . .	73
3.3	ARM firmware design . . . . .	77
3.3.1	Introduction . . . . .	77
3.3.2	System configuration . . . . .	77
3.3.3	Sequence correlation . . . . .	80
3.3.4	Ethernet Communication . . . . .	90
3.4	Boot Sequence . . . . .	91
3.5	Host Application . . . . .	92
3.5.1	Introduction . . . . .	92
3.5.2	Time Difference Of Arrival . . . . .	92
3.5.3	Multilateration . . . . .	94
3.5.4	Localization and position plotting . . . . .	98
3.5.5	UDP Packets Management . . . . .	100
3.5.6	Graphic User Interface layout . . . . .	100
<b>4</b>	<b>Results</b>	<b>107</b>
4.1	Accuracy test . . . . .	107
4.2	Resolution test . . . . .	109
4.3	Tracking test . . . . .	110
4.4	Comparison with other systems . . . . .	111
4.5	Summary of innovations . . . . .	113
4.6	Conclusions . . . . .	114

# List of Tables

1.1	Spectral Power Density Mean values allowed in the UWB frequency band. . . . .	3
1.2	Spectral Power Density Peak values allowed in the UWB frequency band. . . . .	3
1.3	The table reports the known Barker codes, from the shortest long only two symbols, to the longest one of thirteen symbols. On the right side is reported the sidelobe level with respect to the peak of the autocorrelation function. . . . .	15
2.1	Details of the components adopted in the simulation. . . . .	46
4.1	Comparison among UWB RTLS systems . . . . .	113

# List of Figures

1.1	Frequency Band allocation in comparison with the UWB frequency band. . . . .	2
1.2	Waveform of the transmitted UWB Pulse with 2 ns duration and carrier frequency of 7 GHz. . . . .	4
1.3	Symmetrical Double-Sided Two Way Ranging communication scheme	7
1.4	High level scheme of a TDOA based RTLS system. The Three Sensors are connected with a common clock source and to a host PC for data processing and transmission. . . . .	9
1.5	Block diagram of the proposed architecture based on the use of a reference Tag to compensate the time offset among the Sensors clocks.	12
1.6	Graphical representation of the Barker 7 sequence. . . . .	13
1.7	Representation of the Barker 7 sequence autocorrelation function. The peak magnitude is equal to the sequence length and the sidelobe level is kept below zero. . . . .	14
1.8	Graphical representation of the modified Barker 7 sequence. . . . .	16
1.9	Comparison between the autocorrelation function of the original Barker 7 sequence (in blue) and the correlation between the original Barker 7 and our modified version (in red). . . . .	16
2.1	Front, Bottom and cross section views of the final UWB receiver antenna 3D model . . . . .	18
2.2	Geometrical parameters of the receiving antenna model with the corresponding values. . . . .	19
2.3	Front view of the manufactured prototype with the reference coordinates system used for measurements. . . . .	19
2.4	Comparison between the simulation results (blue) and the measured ones (red) for the $\phi = 0^\circ$ cut. . . . .	20
2.5	Comparison between the simulation results (blue) and the measured ones (red) for the $\phi = 90^\circ$ cut. . . . .	21
2.6	Comparison between the simulation results (blue) and the measured ones (red) of the reflection coefficient ( $S_{11}$ ). . . . .	21
2.7	UWB receiver block diagram . . . . .	22

2.8	TCBT-14R+ schematic. The inductor let the DC voltage pass in order to supply the chip and block high frequency signal while the capacitor decouples the DC voltage supply allowing only the high frequency signal to pass. . . . .	23
2.9	Coupled Line Band Pass Filter . . . . .	23
2.10	Coupled line microstrip BPF simulation Results . . . . .	24
2.11	Integrated LPF Insertion Loss behavior . . . . .	25
2.12	The manufactured receiver prototype. The two coupled lines BPF are shown without the metal case cover. . . . .	26
2.13	UWB receiver board Stack-Up . . . . .	27
2.14	High level block diagram of the first Tag prototype . . . . .	29
2.15	Tag power supply block schematic. . . . .	30
2.16	Clock generation circuit . . . . .	31
2.17	Digital circuit schematic showing the pulse sequence, SRF timing and the short pulse generation circuits. . . . .	32
2.18	Low-cost 2 ns pulse generation circuit. . . . .	34
2.19	Timing of the pulse generation circuit used to produce the oscillator driving signal . . . . .	34
2.20	Schematic of the 7 GHz pulsed oscillator . . . . .	35
2.21	Radiation patter of the Taoglas ceramic chip antenna. The images are taken from the component datasheet . . . . .	36
2.22	Evaluation board of the Taoglas antenna with the detail of the coordinates system orientation. . . . .	37
2.23	Tag Stack-Up . . . . .	38
2.24	Layout of the first prototype bare PCB. The full circuit was never mounted since the oscillator was unstable due to parasitics. . . . .	39
2.25	The new implementation of the first prototype moves the RF oscillator on a dedicated two layer PCB and uses an external antenna. . . . .	40
2.26	Tag's high level block diagram. The digital board generates the pulsed voltage supply and the modulating pulse sequence for the RF oscillator. . . . .	42
2.27	Power gating circuit implemented to reduce power consumption. Only the LTC6991 low frequency oscillator and the DC-DC converter are always powered on. . . . .	43
2.28	Detail of the antenna geometric parameters, front and back view of the antenna. . . . .	44
2.29	Top and bottom view of the PCB 3D model . . . . .	45
2.30	CST transient simulation schematic. The 3D model used in the EM simulation is instantiated as an N-port component. . . . .	47
2.31	Voltage of the RF output across the output capacitor (red), and the driving signal (blue) . . . . .	48

2.32	Front view of the assembled low-power Tag prototype with the oscillator and antenna board connected to the digital control circuit. . . . .	49
2.33	Comparison between the simulated radiation pattern (dashed line) and the measured one (red curve). . . . .	50
2.34	Voltage of the radiated pulse signal measured on the oscilloscope with the antenna probe placed at 1 mm from the Tag antenna. . . . .	51
2.35	Measured waveform of the current absorbed from the battery, with focus on the 470 $\mu$ s during the On state of the circuit. The waveform is obtained averaging 64 successive transmissions. . . . .	52
2.36	Zedboard block diagram . . . . .	54
2.37	HMCAD1511 1 GSPS ADC block diagram. . . . .	56
2.38	ADF4360-7 VCO chip block diagram. . . . .	56
2.39	The first Sensor prototype where the evaluation boards of the Xilinx SoC, the ADC, and VCO are connected together. . . . .	57
2.40	The Final Sensor prototype . . . . .	58
2.41	New Sensor processing board and UWB receiver assembly . . . . .	59
2.42	Final aspect of the Sensor board enclosed in the box. . . . .	59
3.1	High level block diagram of synthesized FPGA architecture. Detail of the LVDS data management interface and correlation block. . . . .	63
3.2	High level block diagram of synthesized FPGA architecture. Detail of the ARM processor block and its peripherals. . . . .	64
3.3	Timing diagram of the LVDS data received by the FPGA. The image, taken from the HMCAD1511 ADC datasheet, shows the relation between the LCLK, FCLK and sampling clock signals and the ADC data. . . . .	66
3.4	Implemented LVDS Deserialization architecture showing the LCLK receiving chain in the upper part of the figure, and the FCLK and data receiving chain in the bottom side. . . . .	67
3.5	High-level block diagram of the processing steps performed in FPGA starting from the correlation of the incoming data with the symbol mask, to the thresholding operation to the final storage in the data buffer. . . . .	69
3.6	The symbol mask used for the correlation operations. The mask has is shaped as a 2 ns rectangular pulse. . . . .	70
3.7	Block diagram of the correlator block. It shows the eight instantiations of the elaboration blocks that compute the correlation delays and the final buffer where all the correlation results are stored. Each clock cycle the data buffer receives eight new data (in pink) and discards the eight oldest data (in cyan). The newest correlation result (Del + 7) is computed using all the eight newest data. . . . .	71



3.8	FPGA threshold mechanism. We search for the maximum among the 512 correlation results and we require it to be in the middle of the comparisons window. If this condition is satisfied and the maximum value is larger than the threshold, we set the Thresh flag to 1. . . .	73
3.9	Native FIFO IP interface presented in the Xilinx datasheet. On the left there is the write interface while the read interface is on the right.	74
3.10	FIFO Write process Timing . . . . .	75
3.11	FIFO Read process Timing . . . . .	76
3.12	High level block diagram of the ARM software program. . . . .	78
3.13	Behaviour of complete Tag sequence 111001001010101: 1110010 represents the common preamble while 01010101 is the unique binary Tag ID sequence corresponding to Tag 85 . . . . .	79
3.14	High level block diagram of the operations performed in the ARM software to recognize the Tag ID associated to the received pulse sequence and its TOA. . . . .	81
3.15	Correlation between the raw data and the alignment sequence, the correlation has an absolute maximum when the two sequence are aligned. The x axis of the first two plots correspond to the sample positions. . . . .	83
3.16	Correlation between the raw data and the Barker 7 sequence, the correlation has an absolute maximum when the two sequence are aligned. The x axis of the first two plots correspond to the sample positions. . . . .	85
3.17	Tag ID recognition by mean of thresholding the FIR filtered data. .	86
3.18	Correlation between the Ramp-like mask signal and the raw data, the correlation results underline the edge of the pulses. . . . .	88
3.19	Content organization of the UDP packet sent to the host interface. .	91
3.20	Coordinate system used for the multilateration technique. The origin of the reference system is placed in the reference Sensor position . .	95
3.21	Visualization of the hyperbolic localization problem using TDOA computation. The reference Tag in known position fixes the point where the two curves intersect. . . . .	96
3.22	Generic coordinate system. The multilateration algorithm requires a coordinate system like R1. The passage between coordinates system is done through rototranslation. . . . .	99
3.23	Custom format of the UDP Packets used for configuration command. operations. . . . .	100
3.24	Graphic User Interface layout, on the left is shown the localization area mapped by the Sensors and the mapped tags. The right side shows the raw data plots and configuration buttons. . . . .	102
3.25	Second form, used to show some relevant statistics. . . . .	105

4.1	Indoor laboratory measurement setup. In green, red, blue and gray are highlighted respectively the three Sensors and reference Tag positions. . . . .	108
4.2	Ground truth accuracy measurement. The case with super resolution (magenta) performs better both in accuracy and precision when compared with the standard use case (blue). . . . .	109
4.3	Resolution measurement using two tags spaced 60, 40, 20 and 10 cm apart and enabling the super resolution. . . . .	110
4.4	The tracking measurement results. The red dots represents the 4096 localization results of a Tag rotating around the reference Tag while the blue circle is the real track of the Tag with 90 cm radius. . . . .	111
4.5	The tracking measurement results. The Sensors are moved further away to cover the whole room area. The blue lines represents the track walked carrying the Tag, the red dots represents 1050 localization results. . . . .	112

# Chapter 1

## Ultra Wide Band Localization

### 1.1 Introduction to UWB signals

As our world gets more and more connected, there is an increasing interest into those applications that are able to give us informations about the precise position, with respect to a reference coordinate system, of specific objects of interest, whether they are goods, pieces of machinery or persons.

Nowadays, there are many different solutions to this problem such as Global Positioning System (GPS), ultrasound and infrared time of flight technologies as well as other RF based solutions like WiFi and Bluetooth.

The GPS is a satellite based technology able to provide a very good position approximation (few meters) of satellite signals receivers [17]; it is integrated by many devices such as mobile phones, cars and boats. It is the optimal solution for outdoor positioning but, when we deal with indoor environments, since the signal coming from satellites is very weak, it is not a suitable solution [1].

Among the most used technologies for indoor positioning, there are the infrared light [18] and ultrasound [6] based technologies; these allow low cost solutions and a good accuracy (few centimeters) in the evaluation of the position. However, they require the Line Of Sight (LOS) between the object to be located or tracked and the Sensor used to locate it. This impose a huge constraint on the type of scenario for which these solutions are suitable; for example, a scene with many obstacles makes almost impossible for these system to guarantee an accurate localization.

To overcome the LOS constraints, the WiFi and Bluetooth technologies can be used. Relying on radio frequency signals they can easily be used for indoor localization but, due to high sensitivity to multi-path phenomena, they are not able to reach accuracy lower than a meter [76].

In this landscape of localization technologies another one emerged: Ultra Wide Band (UWB). The Ultra Wide Band is a radio technology based on the IEEE 802.15.4a and 802.15.4z standards [22]; it allows low power communications using

very short pulses with a bandwidth in the order of GHz. The frequency band allocated for UWB application was set in 2002 by the US Federal Communications Commission (FCC) from 3.1 GHz to 10.6 GHz [23].

An overview of the radio regulations applied in Europe is presented in [20] and [56].

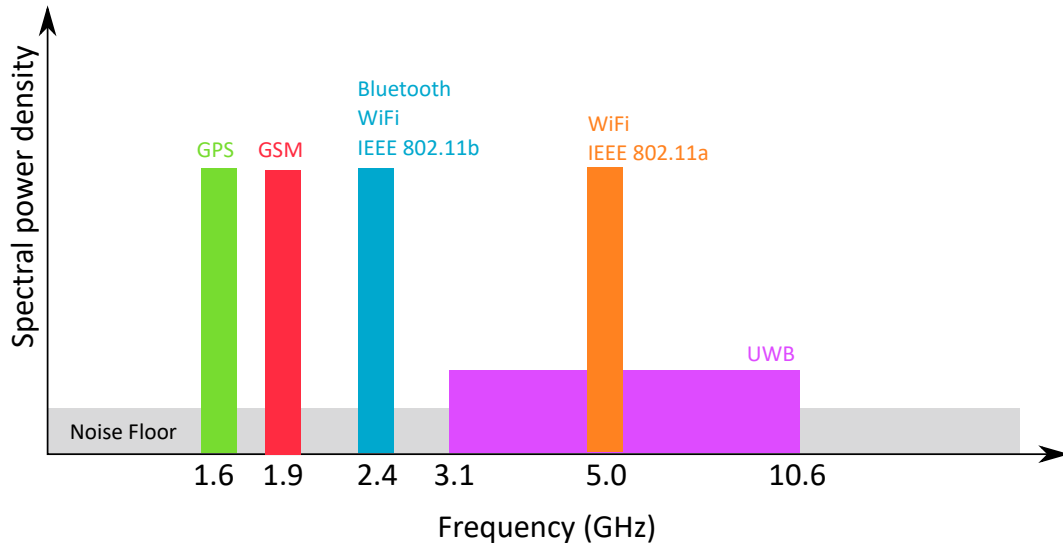


Figure 1.1: Frequency Band allocation in comparison with the UWB frequency band.

The signals transmitted in the European band must comply with an Effective Isotropic Radiated Power (EIRP) transmission mask that establish the maximum values for the power spectral density peak, and mean values. In Europe the power spectral density mean value must be below the  $-41,3dBm/MHz$  threshold while the peak power spectral density must be  $0dBm/50MHz$  maximum when the pulse carrier frequency is in a range between 6 and 8,5 GHz.

The limits on the Peak power density become more strict in those cases where the Pulse Repetition Frequency (PRF) is very low, meaning that the mean power is very limited; differently, when the PRF is higher, the mean power increases as well. The European standard sets the minimum Pulse Repetition Frequency to be 1 MHz, frequency at which the mean spectral power density is most strict.

The limit for these two parameters in the UWB band are reported in Tables 1.1 and 1.2. It is possible to notice that the frequency range where the requirements are less stringent is the  $6,0 < f < 8,5GHz$  one. The UWB technology is suitable for localization applications and, differently with respect to other technologies such as Global Positioning System (GPS), infrared, ultrasound, WiFi and Bluetooth, it benefits of the following advantages:

- Very Low Energy levels: The duration of the pulse is very short compared to

Frequency (GHz)	EIRP Mean Density (dBm/MHz)
$f \leq 1,6$	-90
$1,6 < f \leq 2,7$	-85
$2,7 < f \leq 3,4$	-70
$3,4 < f \leq 3,8$	-80
$3,8 < f \leq 4,8$	-70
$4,8 < f \leq 6,0$	-70
$6,0 < f \leq 8,5$	-41,3
$8,5 < f \leq 10,6$	-65
$f > 10,6$	-85

Table 1.1: Spectral Power Density Mean values allowed in the UWB frequency band.

Frequency (GHz)	EIRP Peak Density (dBm/50MHz)
$3,4 < f \leq 3,8$	-40
$3,8 < f \leq 4,2$	-30
$4,2 < f \leq 4,8$	0
$4,8 < f \leq 6,0$	-30
$6,0 < f \leq 8,5$	0
$8,5 < f \leq 10,6$	-25
$f > 10,6$	-45

Table 1.2: Spectral Power Density Peak values allowed in the UWB frequency band.

the symbol duration reducing the overall transmitted energy;

- High immunity to multipath: In the field of indoor localization GPS is not a suitable choice and Wifi and Bluetooth are heavily affected by multipath. The UWB is much more resilient to this phenomenon thanks to the use of very short pulses with large frequency content allowing to separate the direct path contribution;
- High accuracy ranging: the usage of short pulses enables to achieve ranging accuracy in the order of few centimeters;
- High penetration through and around obstacles: many materials have absorption peaks over a narrow frequency band, in such cases, having a very large bandwidth is an advantage.

## 1.2 UWB Modulations

The transmitted UWB pulses adopted in this thesis are obtained by the multiplication of a short, few nanoseconds base band pulse with a local oscillator signal switching at a central frequency in the working bandwidth between 6.0 and 8.5 GHz. A typical waveform is shown in Figure 1.2.

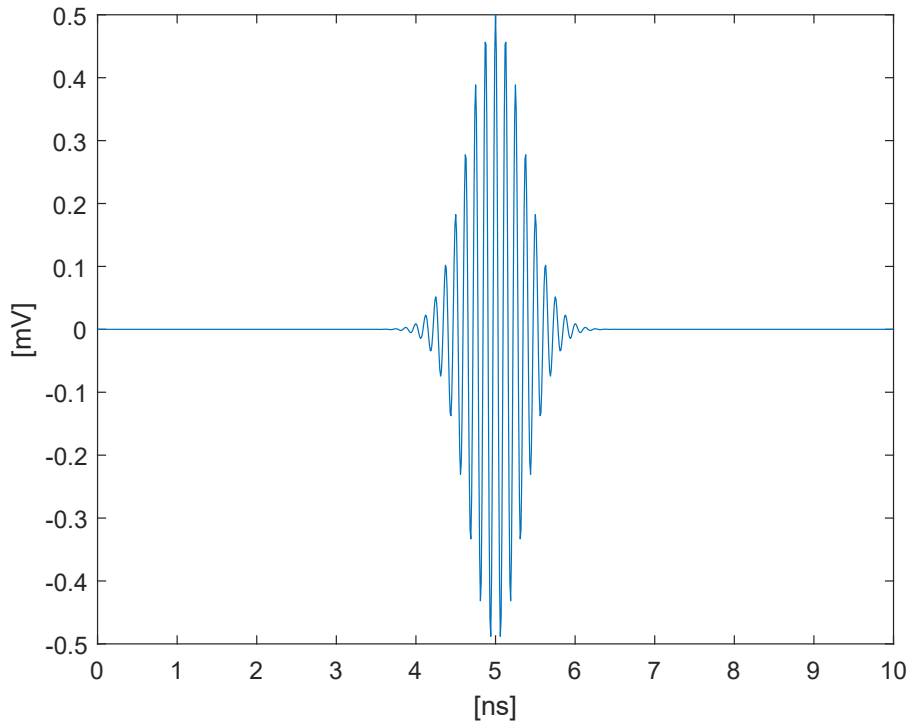


Figure 1.2: Waveform of the transmitted UWB Pulse with 2 ns duration and carrier frequency of 7 GHz.

The presented waveform represents the basic bit of information. It is necessary to find the best modulation technique in order to represent each bit with the proper signal. The most common binary modulation techniques used in UWB communications are:

- Pulse Position Modulation (PPM), where the position of the pulse inside a bit period (symbol) is used to encode the bit value;
- Amplitude Modulation (PAM), where the bit value is encoded in the variation of the amplitude of the pulse;
- Pulse Width Modulation (PWM), where the bit value is associated to the different pulse durations;

- Pulse Shape Modulation (PSM), where the coding of the bit value is associated to a combination of phase and/or frequency modulations. An example is the Binary Phase Shift Keying (BPSK) where the two bit values are associated to pulses that are 180° phase shifted with respect to each others.
- On Off Keying (OOK), where the bit value is associated to the presence or the absence of a pulse inside the bit period.

The very short duration of the signal envelop makes it very hard to use the common modulations that rely on changes in the carrier phase or frequency since during the signal envelop only few carrier periods are present; as a consequence, the modulations commonly used for UWB signals are base band modulation.

The duration of the pulse is in the order of 2 ns while the bit duration is in the order of microseconds. The very short time occupation of the pulse inside the bit period allows very low power communications.

Among the presented modulations, the PPM modulation is the most suitable one to adopt when the symbol we want to transmit in a single period has more than one bit. In this project, in order to keep both the cost and the complexity of the transmitting and receiving chain low, we adopted the OOK modulation with a symbol corresponding to one bit.

## 1.3 Comparison between UWB localization techniques

The problem of indoor localization has raised over the years a large interest both scientific and economic. Over time, many different techniques have been implemented to increase the localization accuracy. A detailed survey on UWB localization technologies is provided in [77]. In the following sections, an overview of the most widely used techniques is provided.

### 1.3.1 Received Signal Strength Indicator (RSSI)

The received signal strength (RSS) is an indication of the power of a radio signal received by a generic receiver and is usually expressed in dBm. Since it gives an indication of the power level of the received signal, the greater the RSSI the better reception we have. The RSSI can be used to estimate the absolute distance between two devices as long as a model for the path-loss propagation is provided and the power at a reference point is known. The RSSI can be estimated as:

$$RSSI = -10n\log_{10}(d) + A \quad (1.1)$$

Where  $n$  is the path loss exponent and  $A$  is the reference signal strength at a certain distance from the receiver. As proposed in [16], to localize an object, it is necessary

to evaluate the RSSI at three receiving stations, compute the absolute distance at each station and compare them with those of a reference node. The main advantage of using this technique is that it has very low implementation costs. However, the RSSI is strongly influenced by the fading multipath effect and by additional attenuation due to Non Line Of Sight (NLOS) conditions caused by the localization environment. This leads to poor localization accuracy.

### 1.3.2 Time Of Flight (TOF)

The choice of using a time-based location-dependent parameter instead of power-based one like RSSI, is associated with the higher accuracy that the time-based parameters allow to achieve [35] [41]. The TOF represents the time needed by a signal to travel from a Sensor to a Tag or viceversa. The TOF is multiplied by the speed of light to compute the distance between the two.

$$d = c \cdot TOF \quad (1.2)$$

To compute the position of a Tag it is necessary to evaluate its distance with respect to a number of Sensors (sometimes called Anchors) located in known positions and apply the multilateration algorithm

In a 2D RTLS system, in order to locate the Tags, a minimum of three Sensors is required for trilateration.

At each Sensor the distance between the Sensor itself and each Tag is computed defining a circumference with radius equal to the distance between the two and centered in the Sensor position. The intersection between the three circumferences individuated gives the position of the Tag.

The TOF based RTLS systems have stringent synchronization requirements between Sensors and Tags. When the sensors do not share the same time reference it is possible to time the communication by means of the Symmetrical Double-Sided Two Way Ranging (SDS-TWR) exchange. The TWR communication scheme, shown in Figure 1.3 performs three steps:

- The Tag broadcasts a timestamped message to all Sensors (Poll message) and wait for a reply;
- Each Sensor receives the message and replies adding the transmission time and including the Poll receiving time (Answer message);
- The Tag receives the answer from the Sensor and save the receiving time then send back a message (Final message) embedding the retransmission time.

Each Sensor receives the Final message that contains all timing informations and compute the TOF as:

$$TOF = \frac{(t_{round1} - t_{reply1}) + (t_{round2} - t_{reply2})}{4} \quad (1.3)$$



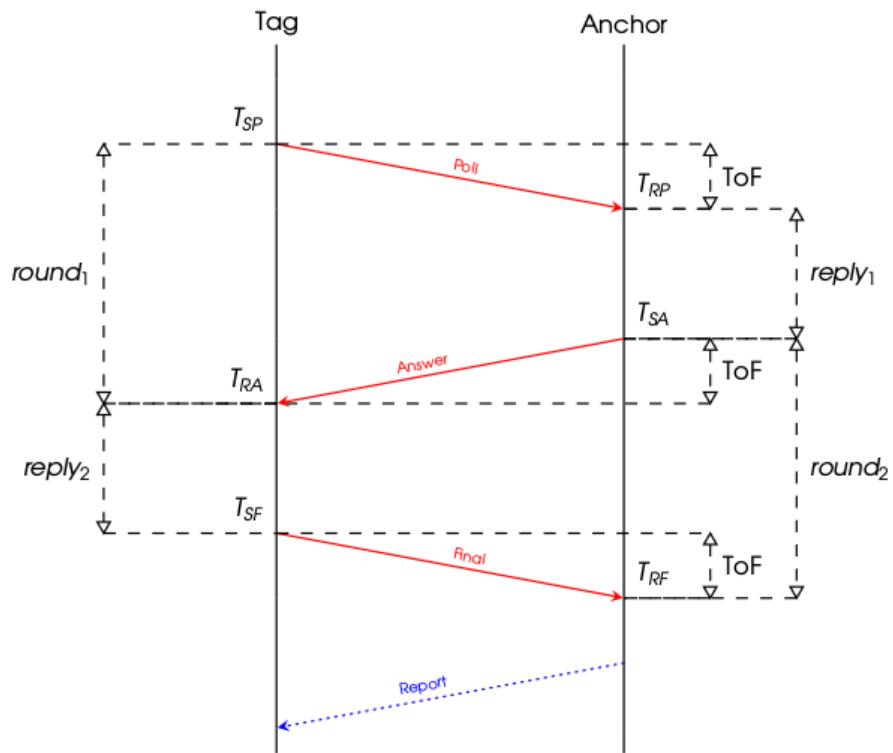


Figure 1.3: Symmetrical Double-Sided Two Way Ranging communication scheme

where  $t_{round1}$  is the Tag turnaround time to transmit the Poll message, wait the Sensor response and receive the response message,  $t_{round2}$  is the Sensor turnaround time to transmit the response message and receive the final response from the Tag. The  $t_{reply1}$  and  $t_{reply2}$  values are the time required by the Tag and the Sensor to reply.

This communication scheme requires the Tag to be able to transmit as well as receive (transceiver) informations from the Sensors increasing not only its hardware complexity but also its power consumption and cost.

The ToF base RTLS systems use the absolute signal propagation time to perform the localization. This technique provides high localization accuracy and, differently from those based on RSSI, do not require any fingerprinting of the surrounding environment. The main disadvantage is that it requires a transceiver Tag with more complex hardware.

### 1.3.3 Time Difference Of Arrival (TDOA)

The localization technique based on TDOA solves the problem of synchronizing Tags and sensors by exploiting the difference of propagation time among the receiving sensors. The Tag does not need to be a transceiver anymore and can be simplified to a transmit only device. In this configuration only the Sensors share the same time base. This allows to reference the TDOA measured at each couple of Sensors to the same transmission event.

The computed TDOA are used in the Multi-lateration algorithm to compute the position of the Tags. To compute the 2D position of the Tags, a minimum number of three sensors is required.

Since the communication is unidirectional where the Tag only transmits and does not receive, this localization method is also known as One Way ranging (OWR).

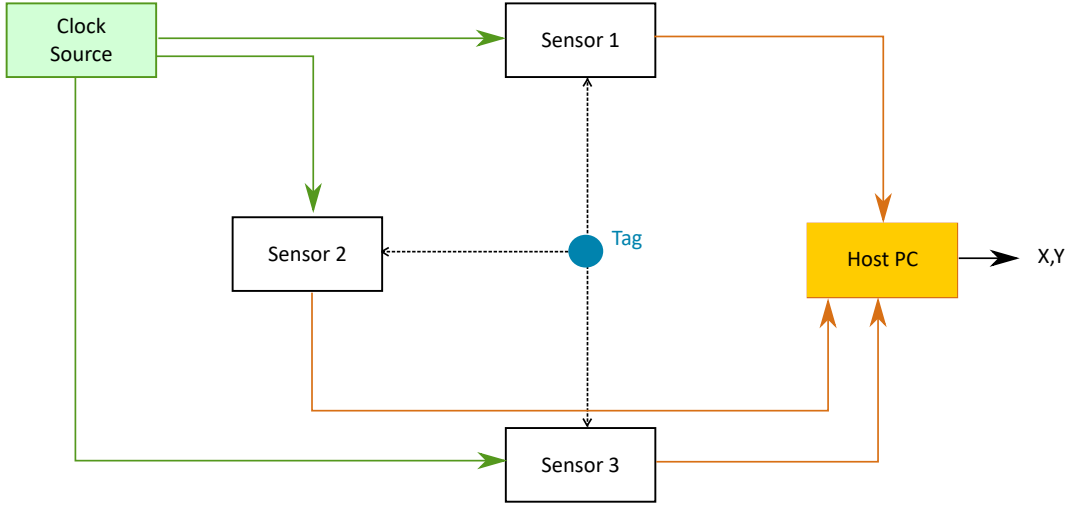


Figure 1.4: High level scheme of a TDOA based RTLS system. The Three Sensors are connected with a common clock source and to a host PC for data processing and transmission.

The crucial aspect of this solution is the time synchronization between the Sensors that must satisfy very strict requirements. For example, a difference of 1 ns in the distribution of the clock from one Sensor to another leads to an error in the position estimation of 30 cm, equal to the distance travelled by the light in 1 ns. The Sensor synchronization constraint increases the overall system installation cost as well as its complexity.

### 1.3.4 Phase Difference Of Arrival

The Phase Difference of Arrival (PDOA) solution is implemented using an array of antennas at each Sensor and exploiting the fact that an incident plane wave does not impinge on all the antennas in the array at the same time but there is a delay in the time of arrival between two antennas associated to the distance between the antennas expressed by the formula:

$$p = d \cdot \sin(\theta) \quad (1.4)$$

Where  $d$  is the distance between the antennas,  $\theta$  is the Angle (or Direction) Of Arrival (AOA/DOA) of the impinging waveform and  $p$  is the difference in path length.

Consider a wave with carrier frequency  $f$  and wavelength  $\lambda = \frac{c}{f}$ , the PDOA  $\alpha$  is:

$$\alpha = \frac{2\pi}{\lambda} p = \frac{2\pi f}{c} p \quad (1.5)$$

From 1.5, solving for  $\theta$  one can obtain the relation between the measured PDOA and the AOA/DOA

$$\theta = \arcsin \frac{\alpha \lambda}{2\pi d} \quad (1.6)$$

With this method, using two Sensors and intersecting the AOA/DOA it is possible to determine the Tag 2D location.

The main advantage of the PDOA technique is that the Tag position in a 2D environment can be estimated with just two Sensors with a receiving array of antennas [75]. The architectures based on this technique are very sensitive to the multipath effect and the performance rapidly deteriorates in NLOS conditions.

### 1.3.5 Existing systems available on the market

On the market, there are already various solutions with different time-based approaches, such as DecaWave [13], Ubisense [59] and Zebra [78].

The DecaWave DW1000 system uses the TWR TOF measurement approach reaching a ranging accuracy of 10 cm [8] and a typical update rate of 3.5 Hz. The update rate needs to be lowered as the number of Tags to track increases. Each device can be configured as a Sensor or Tag and, due to the TWR approach, does not need a common clock source.

Nowadays, several commercial solutions, such as Pozyx [47], TimeDomain [57], Sewio [51], Quantitec [48], and OpenRTLS [44], adopt the DecaWave chipsets. Some of these systems use the UWB TWR approach together with the information coming from an Inertial Measurement Unit (IMU) embedded in the Tag that allows generating useful attitude information. The fusion of different kinds of location-dependent parameters is used in indoor navigation systems under the name of hybrid localization [32].

The performances of this kind of systems are evaluated in [50] [11] [54]. Among them, the position accuracy and Sensor performances of the Pozyx commercial solution are evaluated in [12] and [40]. Even though systems based on hybrid localization techniques allow achieving 10-cm range accuracy, the higher complexity of the Tag hardware increases the overall solution cost.

The Ubisense system adopts the TDOA and PDOA approaches using the time-difference information determined between pairs of Sensors connected with a timing cable [53]. The system is capable to provide a localization accuracy of 15 cm and an update rate from 0.1 to 20 Hz. The performances of the DecaWave and Ubisense systems are compared in [29].

In [38], a technique that combines TDOA and TOF measurements is proposed. It is based on the DecaWave DW1000 system and, due to the combination of the two methods, allows to compensate for their respective limitations and to increase the localization accuracy in a cooperative scenario [33], [64]. The combination of the

two methods can effectively improve the accuracy of the TDOA method alone but requires a more complex architecture.

The Zebra system uses the TDOA approach and is capable to reach 30-cm localization accuracy with a maximum update rate of 200 Hz. The overall cost of the hardware infrastructure and Tags is very high compared to DecaWave and Ubisense systems but allows tracking of a large number of fast moving Tags.

### 1.3.6 Proposed Implementation

In this work, we propose a one-way, UWB real-time locating system (RTLS) based on TDOA computation using a Sensors network where the time synchronization among the Sensors is not wired but wireless and implemented by means of a reference Tag, identical to any other Tag to track, placed in a known and fixed position. A similar approach is described in [61]. The goal is to design an RTLS system with better performances with respect to the system already on the market, designing and prototyping custom hardware, and implementing dedicated software to reduce the overall cost of the system infrastructure.

In a 2D TDOA based RTLS system, a minimum of three Sensors is required. The main disadvantage of TDOA architecture is that the Sensors require a common time base to synchronize the measurements. For this reason, we adopt a different architecture that does not require a wired common clock among the Sensors but, instead, uses a reference Tag, placed in a known fixed position to provide the required synchronization.

In this configuration, each Sensor has its own independent clock and, by comparing the TOA measurements with respect to the ones coming from the reference Tag, it is capable to compensate for the time offset among the Sensor clocks and for error drifts. The implemented architecture is the presented in Figure 1.5.

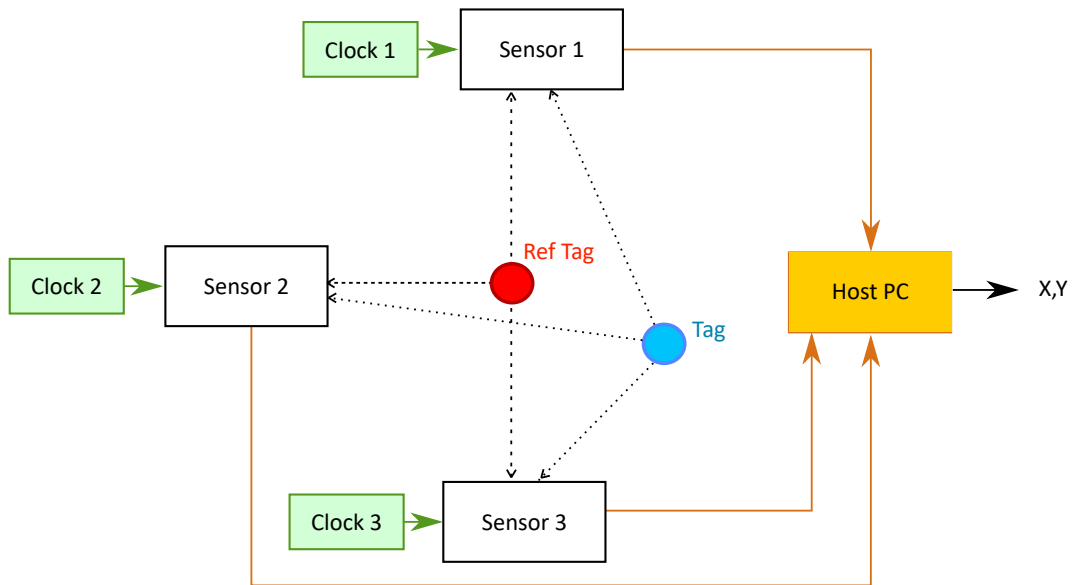


Figure 1.5: Block diagram of the proposed architecture based on the use of a reference Tag to compensate the time offset among the Sensors clocks.

## 1.4 Tag transmitted Sequence

The Tag is designed to transmit a sequence of UWB pulses. The implemented technique is based on the OOK modulation to simplify the Tag architecture. This allows to code a bit equal to one whenever we detect a pulse during a bit period. This period is set equal to 50 ns by a 20 MHz clock used to time the Tag transmission operations. The case where no pulse is detected during the bit period codes a bit equal to zero. Using this modulation we reduce the Tag's hardware complexity since we need to produce only one type of signal.

The transmitted sequence is fifteen bits long where the first 7 bits of the sequence represents the Preamble of the message while the remaining 8 bits represents the Tag ID number. The Preamble sequence is common to all Tags and is used in the Sensor processing to recognize the presence of a pulse sequence in the incoming data stream and to calculate its TOA. The successive eight bits represents the unique Tag ID number. The Tag ID number is used during the Sensor processing to associate the TOA to the specific Tag transmitting the sequence. The Tag ID can assume any possible value obtainable by the combination of 8 bits with the exception of the all zeros case.

Differently from the Tag ID, the Preamble need to have two specific characteristics in order to maximize the detection:

- It needs to maximize the sidelobe level ratio with respect to the peak of its

autocorrelation function;

- It needs to be as short as possible.

To satisfy these requirements we chose to implement the Preamble sequence using the Barker 7 sequence [31] as a good compromise between code length and performances. Other sequences, like longer Barker sequences or even longer sequences like the Golay codes were not taken into account due to their excessive length and processing requirements. Moreover, longer codes require to transmit larger amount of energy that reduces the battery duration.

A Barker code is a finite sequence of  $N$  values that can assume the values  $\pm 1$  with an autocorrelation function with coefficient defined as:

$$c_v = \sum_{j=1}^{N-v} a_j a_{j+v} \quad (1.7)$$

where the  $|c_v| \leq 1$  for all  $1 \leq v < N$ . This result can be interpreted as the sum of adjacent symbols should be less than or equal to 1 in all partial lengths of the sequence. This kind of sequence has the peculiar property of having an autocorrelation function with minimum sidelobe level equal to  $1/N$  times the peak value. The Barker 7 sequence graphical representation is shown in Figure 1.6 while its autocorrelation function is shown in Figure 1.7.

The Barker sequences are commonly used in bi-phase modulation for pulse com-

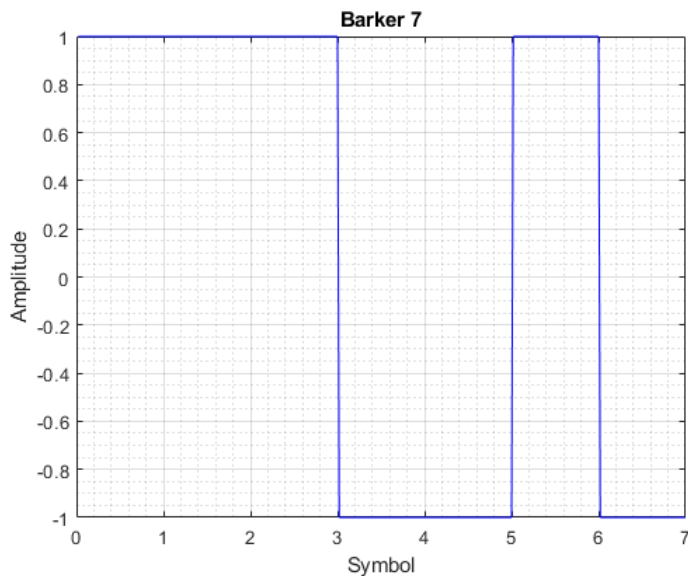


Figure 1.6: Graphical representation of the Barker 7 sequence.

pression techniques in radars [63][30]. The known Barker codes are reported in

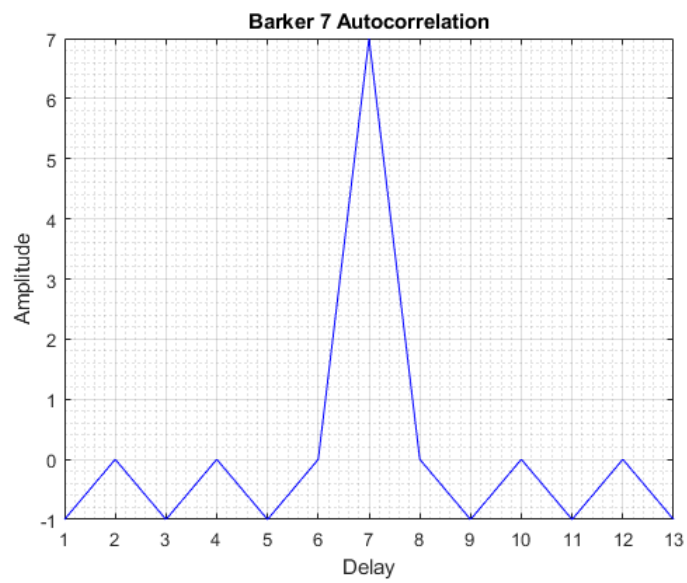


Figure 1.7: Representation of the Barker 7 sequence autocorrelation function. The peak magnitude is equal to the sequence length and the sidelobe level is kept below zero.



Table 1.3.

Sequence Length	Codes	Sidelobe level ratio
2	+1-1	- 6 dB
3	+1+1-1	- 9.5 dB
4	+1+1-1+1	- 12 dB
5	+1+1+1-1+1	- 14 dB
7	<b>+1+1+1-1-1+1-1</b>	- 16.9 dB
11	+1+1+1-1-1-1+1-1-1+1-1	- 20.8 dB
13	+1+1+1+1+1-1-1+1+1-1+1-1+1	- 22.3 dB

Table 1.3: The table reports the known Barker codes, from the shortest long only two symbols, to the longest one of thirteen symbols. On the right side is reported the sidelobe level with respect to the peak of the autocorrelation function.

In this work, a variation of the Barker 7 sequence has been implemented due to OOK modulation. The implemented Barker 7 sequence substitutes the signal corresponding to a negative one with zeros, due to the absence of a pulse during the bit period. The graphical representation of the modified Barker 7 sequence is shown in Figure 1.8. The comparison between the autocorrelation function of the original Barker 7 sequence and the correlation between the Barker 7 sequence and our modified signal is shown in Figure 1.9. The correlation of our signal with the original Barker sequence has a lower peak value and higher sidelobes when compared to the Barker 7 autocorrelation function but represents a good trade-off between performance, complexity of implementation, and hardware cost.

The transmitted UWB pulse in each bit period is 2 ns long with a carrier frequency of 7 GHz. As already stated, the duration of the bit period is 50 ns, corresponding to a bit frequency of 20 MHz, far larger than the minimum frequency of 1 MHz required by the standard.

The time occupation of the pulse inside the bit period corresponds to a duty cycle of  $D = 4\%$ : this very low value of  $D$  allows the transmission to require very low energy.

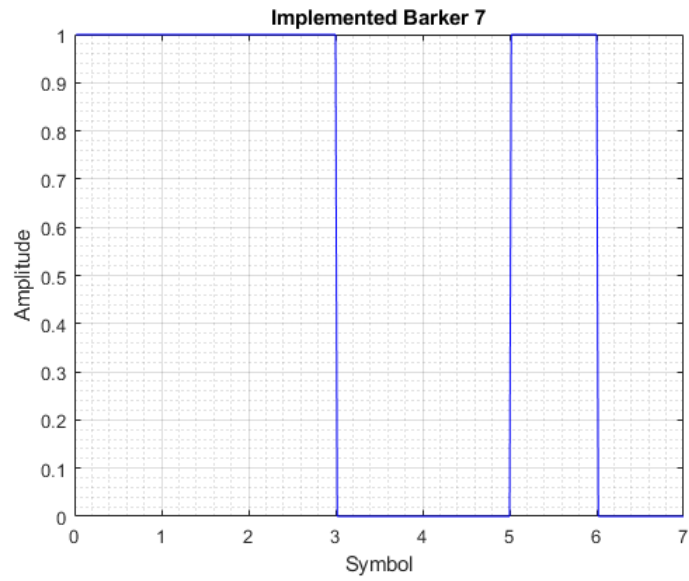


Figure 1.8: Graphical representation of the modified Barker 7 sequence.

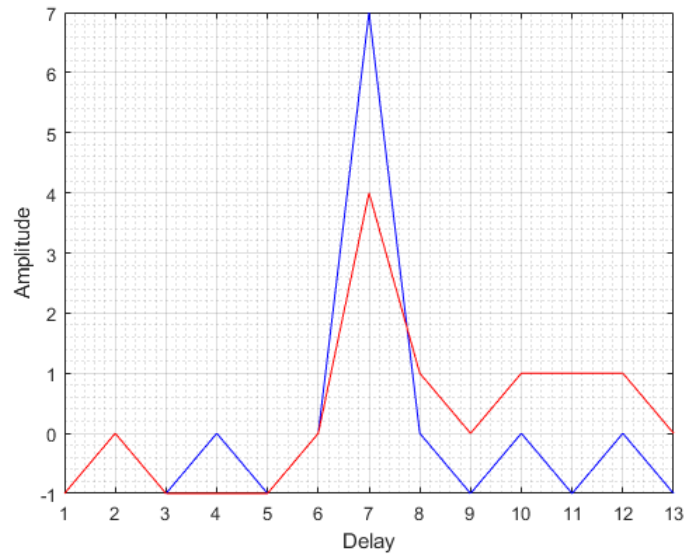


Figure 1.9: Comparison between the autocorrelation function of the original Barker 7 sequence (in blue) and the correlation between the original Barker 7 and our modified version (in red).

# Chapter 2

## Hardware Design

### 2.1 Introduction

The goal of the project is to implement a low cost RTLS system; to accomplish this task, it is first necessary to identify the target hardware for both the Sensors and the Tags. The solution we implemented required the design of custom hardware for the Tag as well as for the Sensor RF receiver while, for the Sensor analog to digital conversion and processing, we initially relied on Components Off The Shelf (COTS) hardware leading us to the assembly of the first working prototype. Then we proceeded to a complete custom design both for the Tag and for the Sensor.

In the following sections the design choices, the schematics and the characteristics of the custom hardware manufactured are introduced. The discussion starts with the Sensor RF receiver, the daughter board that connects the receiving antenna to the processing part of the Sensor, and then proceeds presenting the design of the Tag low power transmitter.

Successively, the first prototype of the Sensor is introduced. In order to simplify and speed up the design process, we firstly selected evaluation boards available on the market for the analog to digital conversion and for the digital processing. The specifications of these boards will be presented together with the reasons that led us to their choice.

Once the first prototype was assembled and tested, it was time to optimize the Sensor hardware from an assembly of evaluation boards to a custom board dedicated to our application.

The discussion ends with the presentation of the final prototype of the Sensor made up of the custom processing board and RF receiver.

## 2.2 Ultra Wide Band Receiver

### 2.2.1 UWB Receiving antenna

The Tag UWB pulse sequence is coded using an OOK modulation and transmitted using an elliptic dipole antenna in linear, vertical polarization. This fixes design constraints for the geometry of the receiving antenna. The antenna needs to have:

- Good matching and gain for a wide range of frequencies around the 7 GHz carrier;
- Wide angular field of view in the horizontal plane;
- Linear vertical polarization;
- An easy connection to the receiver;
- Reasonable geometrical dimensions to satisfy the mechanical constraints imposed by the receiving Sensor enclosure.

Various solutions for UWB antennas are proposed in literature, from commercially available ceramic SMD chip solution [60] to PCB printed solutions based on different optimizations of the patch antenna geometry like in [34][46] [36][55].

The solution we adopted is constituted by a single patch with a "U" shaped slot to increase the bandwidth. The antenna has been designed, simulated, and optimized using CST Microwave Studio software. The 3D model of the final solution is shown in Figure 2.1. The patch has been simulated on an FR-4 (lossy) substrate with a

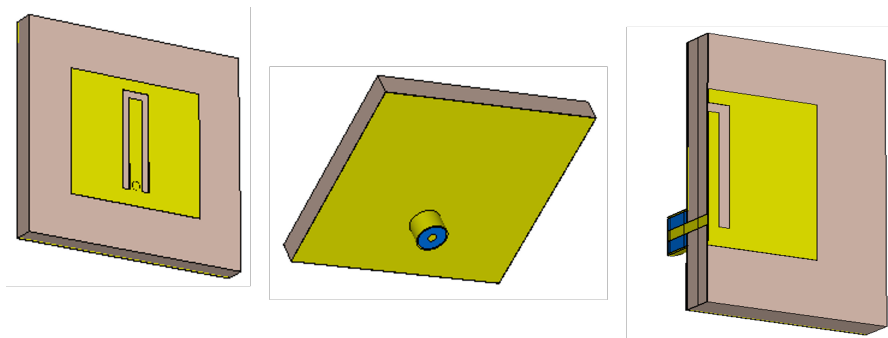
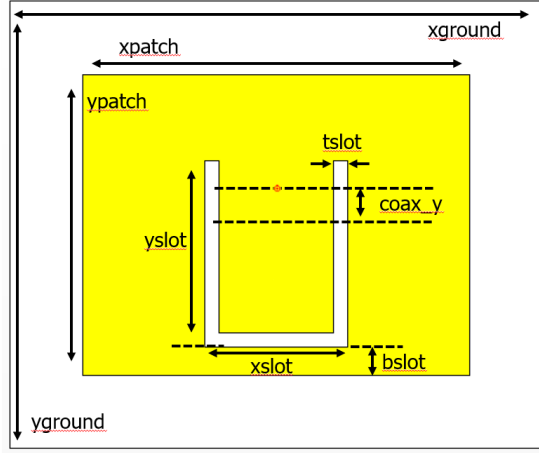


Figure 2.1: Front, Bottom and cross section views of the final UWB receiver antenna 3D model

thickness of 1.6 mm and a dielectric constant  $\epsilon_r = 4.3$ . The geometrical parameters of the optimized antenna are reported in Figure 2.2a and Table 2.2b.



Parameter	[mm]
xground	16.4
yground	15.3
zground	0.035
xpatch	10
ypatch	8.9
xslot	2.0
yslot	7.0
coax_y	3.25
bslot	0.5
tslot	0.7

(a) Patch antenna geometrical parameters names (b) Patch antenna geometrical parameters values

Figure 2.2: Geometrical parameters of the receiving antenna model with the corresponding values.

The vertical position of the feeding point for the coaxial connector determines the input impedance which has been set equal to  $50\Omega$ . The antenna has been manufactured and measured, the final prototype is shown in Figure 2.3.

The antenna has been measured in the anechoic chamber of our Institute. We

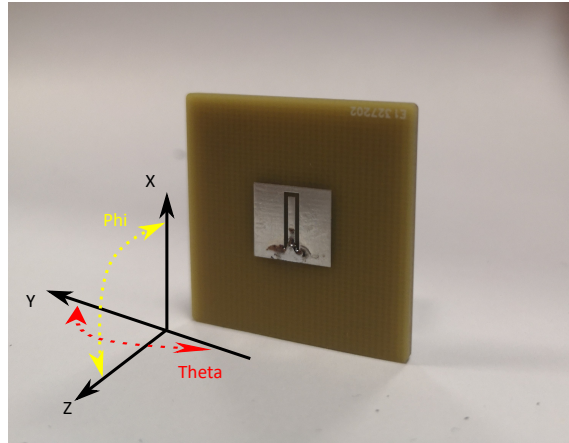


Figure 2.3: Front view of the manufactured prototype with the reference coordinates system used for measurements.

performed a spherical scan with angular steps of  $5^\circ$  along the  $\phi$  angle and  $1^\circ$  along the  $\theta$  angle measuring two orthogonal linear polarizations along  $\theta$  and  $\phi$ .

The gain of the Antenna Under Test (AUT) is obtained as:

$$G(\phi, \theta)_{dB} = 20 \cdot \log_{10}(\sqrt{G_{\theta}(\phi, \theta)^2 + G_{\phi}(\phi, \theta)^2}) + KF_{dB} \quad (2.1)$$

Where  $G_{\theta}(\phi, \theta)$  and  $G_{\phi}(\phi, \theta)$  are the measured polarization components and  $KF_{dB}$  is the standard-gain horn antenna probe correction factor. The obtained results are shown in Figure 2.4 and 2.5. The sidelobe level is well under -10 dB in both

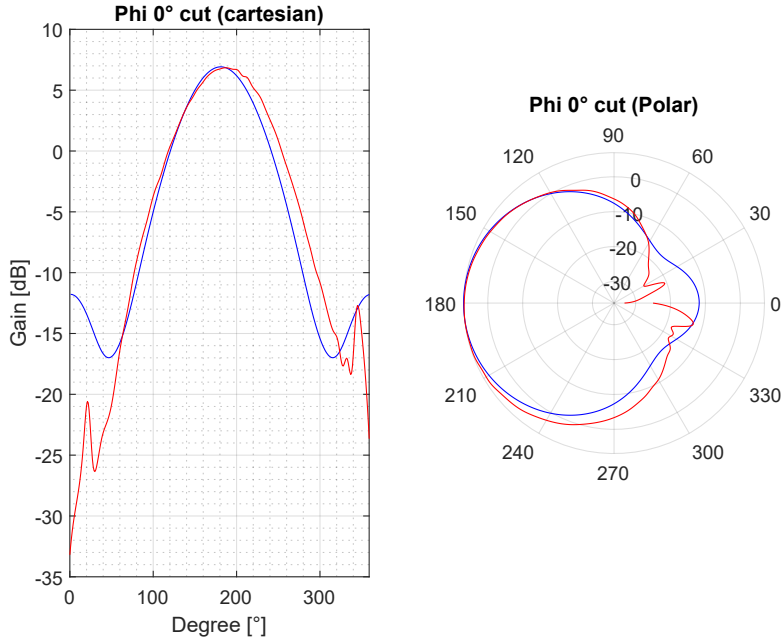


Figure 2.4: Comparison between the simulation results (blue) and the measured ones (red) for the  $\phi = 0^\circ$  cut.

cuts, the angular width at -3 dB with respect to the maximum gain is  $78^\circ$  in the  $\phi = 0^\circ$  cut and  $88^\circ$  in the  $\phi = 90^\circ$  and matches with the simulations. The simulated maximum gain is very similar to the measured one. In Figure 2.6 the simulated and measured  $S_{11}$  at the antenna connector are shown indicating a very similar behavior with just a 100 MHz frequency shift between the two. The  $-10dB$  bandwidth of the antenna is close to 500 MHz as expected.

## 2.2.2 RF Receiver: Block Diagram

The receiving antenna is connected through a SMA connector to the UWB receiver board. The high level block diagram of the receiver is shown in Figure 2.7. The receiver amplifies the signal using three cascaded Low Noise Amplifier (LNA) stages interspersed with two band pass filters centered at the carrier frequency.

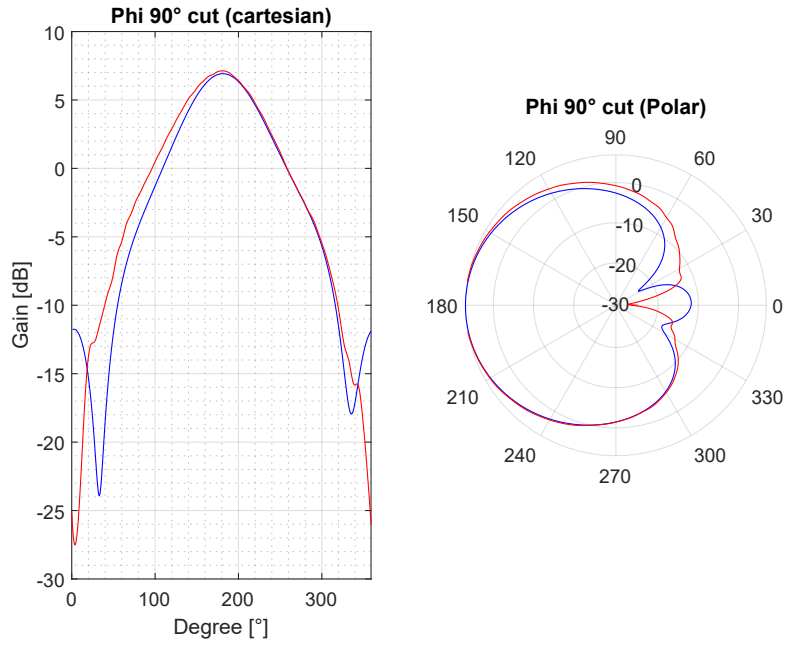


Figure 2.5: Comparison between the simulation results (blue) and the measured ones (red) for the  $\phi = 90^\circ$  cut.

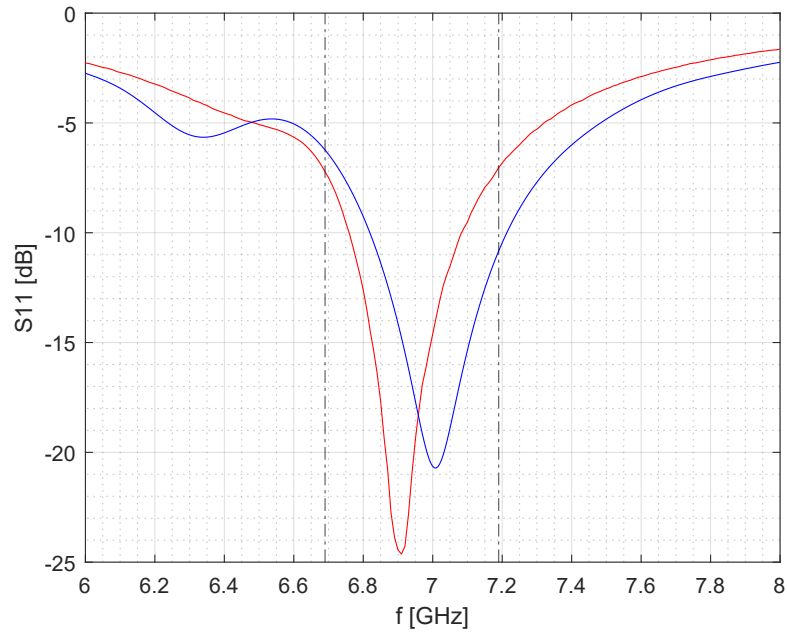


Figure 2.6: Comparison between the simulation results (blue) and the measured ones (red) of the reflection coefficient ( $S_{11}$ ).

After amplification and filtering, the signal is rectified using a Schottky diode and low pass filtered to eliminate the unwanted higher frequency content.

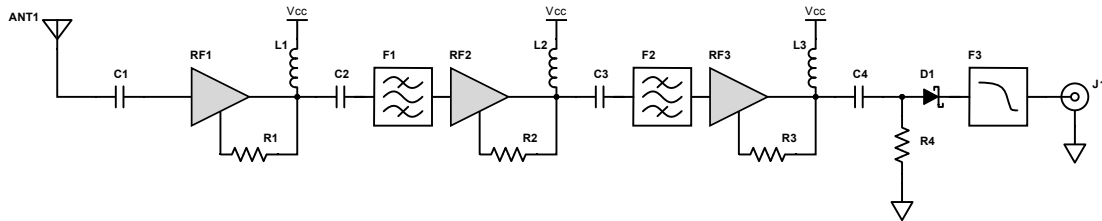


Figure 2.7: UWB receiver block diagram

The antenna signal is DC filtered and provided to the first of three cascaded amplification stages. All three amplification stages use the same amplifier chip, the MAAL-011130 from Macom [5]. This chip is a broadband low noise amplifier with a minimum gain of 19 dB from 2 to 18 GHz and a noise figure of 1.4 dB at 10 GHz. The amplifier has a high 1 dB intercept point of 13 dBm at 6 GHz, this parameter is critical when cascading multiple amplifier with significant gain since it can cause the whole chain gain to flat if the signal received from the antenna is very high. In these situations, it is common for the third amplification stage to saturate causing the gain to drop, the noise floor to rise and the generation of distortions, visible in the received signal as the introduction of higher order harmonics.

The amplifier supports single voltage supply instead of the more common dual rail one. This allows the UWB receiver to be supplied using only one voltage simplifying the power distribution tree in the Sensor. The single voltage supply operation is enabled by connecting an external resistor between the bias control voltage pin (pin 4) and the RF/ $V_{cc}$  pin (pin 7). Since the RF output signal and the voltage supply  $V_{cc}$  rail share the same pin, it is necessary to decouple them by means of an external, integrated bias tee component. The circuit topology of such component is shown in Figure 2.8. It is important to chose a component that can fit in the design with small Insertion Loss (IL) to minimize the impact on the amplification budget and with minimum dimensions.



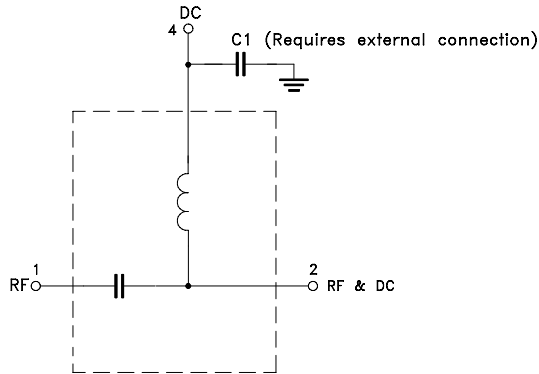


Figure 2.8: TCBT-14R+ schematic. The inductor let the DC voltage pass in order to supply the chip and block high frequency signal while the capacitor decouples the DC voltage supply allowing only the high frequency signal to pass.

The TCBT-14R+ from *Minicircuits* fits all the requirements, it has a low IL of 0.66 dB at the center frequency of 7 GHz, it is provided in a small package and it is cost effective. The cost of the amplifier and bias tee circuit remains in the order of 20 USD for a single piece.

After the first amplification stage we need to filter the signal. There are a lot of COTS solutions for X band filters but they are extremely expensive. In order to keep the costs as low as possible, we designed a custom Coupled Line Band Pass Filter (BPF) implemented in microstrip technology to perform the filtering operations. Figure 2.9 shows the final layout of the coupled line band pass filter. The filter has been designed and simulated in CST Microwave Studio using the

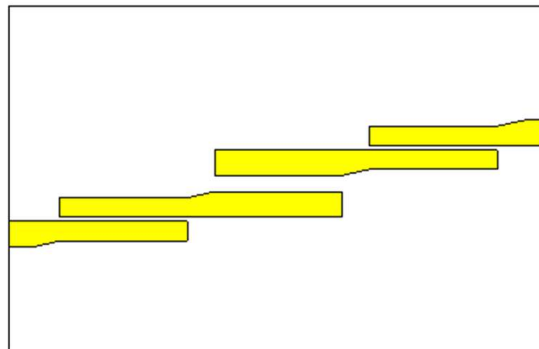


Figure 2.9: Coupled Line Band Pass Filter

RO4350B from Rogers as substrate. The datasheet of the material is available at [10]. This dielectric material has good thermal performances, very low losses ( $\tan\delta = 0.0037$  at 10 GHz) and a dielectric constant of  $\epsilon_r = 3.66$ . Compared with others high frequency, high performances dielectrics such as RO3003 [9] or Astra

MT77 [19], it offers comparable performances for a significantly lower cost per  $\text{cm}^2$ . The simulation used a 20 mils ( $508 \mu\text{m}$ ) thick substrate and dielectric constant of 3.66. The simulated copper trace thickness is fixed at  $50 \mu\text{m}$  to take into account the copper growth associated to two rounds of galvanization. The ground plane copper is  $35 \mu\text{m}$  thick. The simulation results are shown in Figure 2.10 where the relevant points to evaluate the central frequency, the filter bandwidth and the filter attenuation are highlighted. The filter input port as well as the output one are fed

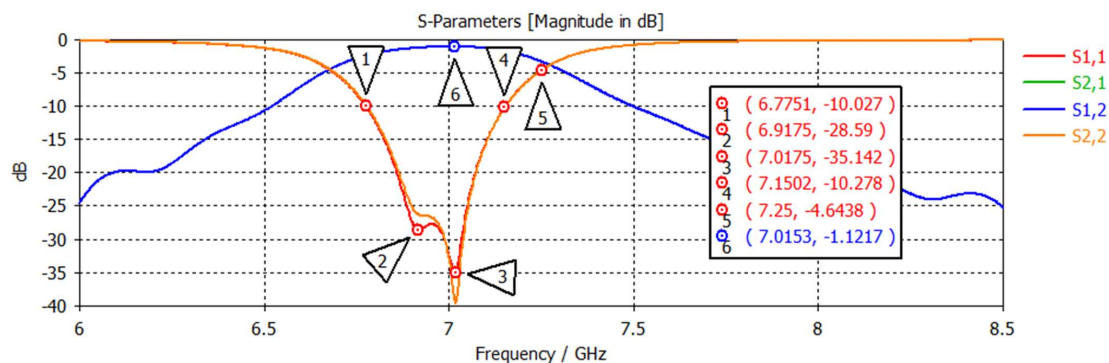


Figure 2.10: Coupled line microstrip BPF simulation Results

using  $50\Omega$  microstrip. The simulation results show a center resonance frequency at 7 GHz with a -10 dB bandwidth of  $\Delta B = 380\text{MHz}$ . The two ports are well matched having a -35 dB reflection coefficient. The filter introduces minimum attenuation of 1.1 dB on the received signal at the center frequency and deteriorates to almost 3 dB at the extremes of the filter bandwidth. This attenuation is a good tradeoff between the bandwidth of the filter and its dimensions.

The second amplification stage is identical to the first one and has the same filter at its output. The third stage is cascaded to the second filter. The output of the third stage is fed to the RF power detector circuit. The choice of using a power detector is justified by the necessity of computing the accurate time of arrival of the pulse sequence at the receiver. To perform this operation it is necessary to detect the sequence time of arrival and no further information on phase or frequency is required. Different examples of power detectors based on Schottky diodes are shown in [27]. To implement the power detector we adopted the SMS7621-079LF from Skyworks [52].

The last operation in the receiving chain is low pass filtering. There are multiple ways to implement a low pass filter; microstrip stepped impedance filters are cheap since they do not require any components but rely only on changes in the microstrip width to model an inductor or a capacitor, they can achieve any bandwidth with low insertion losses. However, their implementation requires a large amount of PCB space. Discrete component implementations are a much more compact solution and are very low cost. The order of the filter and cut-off frequency can be chosen.

The main disadvantage of such solutions is the strong dependence of the transfer function on component tolerances. The adopted solution is based on integrate, ceramic chip Low Pass Filter (LPF). These components are available in standard SMD packages, do not depend on component tolerances and are  $50\ \Omega$  matched. The LFCG-1200+ from Minicircuits [7] is chosen as LPF. The insertion loss behavior is shown in Figure 2.11, it is possible to see that up to 1 GHz the IL of the filter remain lower than 0.6 dB.

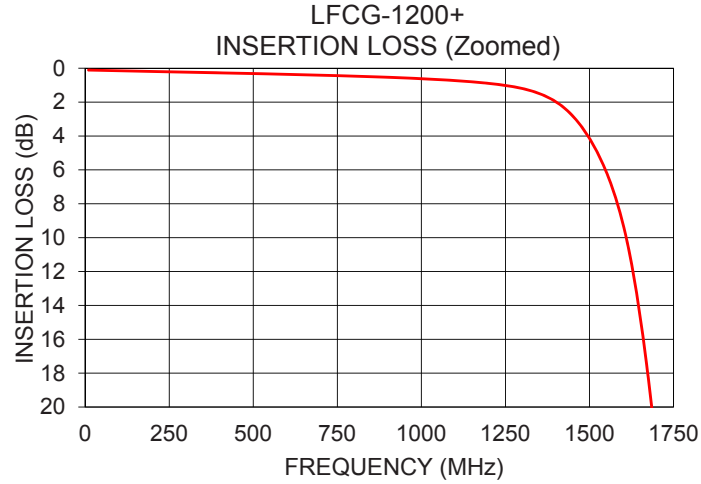


Figure 2.11: Integrated LPF Insertion Loss behavior

The output of the LPF is provided to an SMP connector that, through a bullet adaptor, connects the receiver to the Sensor processing board ADC. In order to estimate the receiver amplification budget we need to take into account the gain of each amplifier and subtract the IL of all the components in the receiving chain. The total amplification budget, assuming the nominal gain of the amplifier to be 20 dBm at 7 GHz will be equal to:

$$G = 3G_{Amp} - 2IL_{BiasTee} - 2IL_{BPF} - IL_{LPF} \quad (2.2)$$

Substituting the obtained values in the equation we have:

$$G = 60 - 1.36 - 2.24 - 0.86 = 55.54dB \quad (2.3)$$

### 2.2.3 RF Receiver: PCB Design

The UWB receiver has been manufactured and tested. The final view of the prototype is shown in Figure 2.12. The custom PCB of the receiver is connected to the Sensor processing board using a standard SMD in line connector for ground and voltage supply connections and an SMP connector for the received signal.

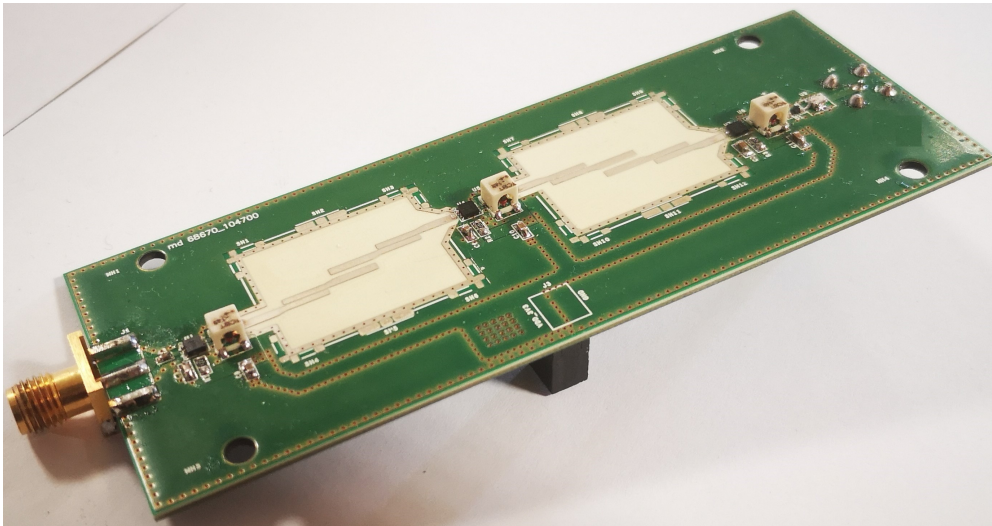


Figure 2.12: The manufactured receiver prototype. The two coupled lines BPF are shown without the metal case cover.

The board needs to be manufactured using RO4350B high frequency, low losses materials. To keep the costs low, an hybrid stack-up of RO4350B high frequency material and standard FR-4 has been used. The solution has four layers, the stack-up of the board is shown in Figure 2.13. The first two layers are made of a 35  $\mu\text{m}$  thick copper layer standing on a 20 mils (508  $\mu\text{m}$ ) RO4350B dielectric core. The top layer (L1) is used for component placing and routing while the second layer (L2), is the reference ground plane. The second core, connecting layer L3 and L4, is made up of a low cost FR-4 R-1566 1501 laminate from Panasonic [45] with nominal thickness of 14 mils (355  $\mu\text{m}$ ). These two layer are used respectively for power supply routing (L3) and for component placing and ground (L4). The two cores are connected using two sheets preimpregnated (prepreg) R-1551 7628 from Panasonic with total thickness of 16 mils (406  $\mu\text{m}$ ). The characteristics of the prepreg are the same as the FR-4 core mentioned above.

We choose to realize four layers to have higher separation between sensible nets

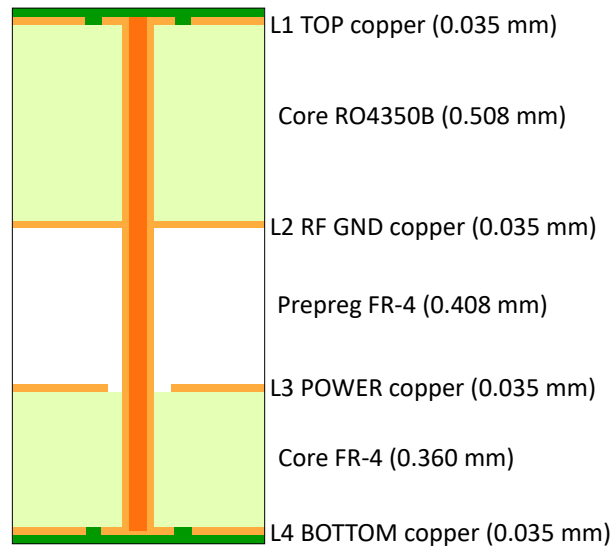


Figure 2.13: UWB receiver board Stack-Up

and noisy power lines and to increase mechanical stability. A common solution in high frequency designs to reduce interference from outside is to delimit the edges of the board with a row of through hole vias that connects the ground etching along the cross section of the board. The same concept has been applied to the microstrip filters by covering them with COTS metallic cages.

## 2.3 First Prototype Tag

### 2.3.1 Design Overview

The Sensors receive the pulse sequences transmitted by UWB Tags; these devices are designed to generate the OOK modulated pulse sequence with the correct bit and sequence repetition timing. In order to keep the manufacturing costs and power consumption low, we need to generate the signal using discrete components. The implemented solution uses a single transistor based oscillator circuit that generate the 7 GHz carrier and a custom controller to modulate it. The OOK modulation transmits an UWB pulse only when the sequence has a bit equal to "1" while, when the bit is equal to "0", no pulse is transmitted. The single bit interval is fixed to 50 ns (20 MHz) and largely satisfy the requirement on minimum bit duration (1 MHz).

During the bit interval the controller needs to gate the carrier signal with 2 ns pulses in order to have the required pulse bandwidth. The Sequence Repetition Frequency as well as the pulse sequence itself, are hardwired for each Tag.

Two different Tag prototypes have been manufactured. The two designs had different focus, the first prototype had the goal to verify the feasibility of designed solution and its performances. The second prototype optimizes power consumption, dimensions, cost and performances.

In the following sections we will describe in detail the high level block diagram of both solutions and describe their differences. We will then proceed describing the custom board design and manufacturing details and discuss the performances by comparing simulations results and measurements.

### 2.3.2 Block diagram

The first prototype high level block diagram is presented in Figure 2.14. The transmitter architecture can be subdivided in three macro blocks:

- The power supply block, represented by the 3.7 V battery and the battery recharge circuit;
- The digital block, made up of three different sub-blocks: the sequence generator and sequence repetition frequency generator circuits drive the pulse generator circuit. The digital circuit output is the driving signal for the RF oscillator;
- The RF block, composed by the oscillator that generates the carrier signal and the UWB antenna.

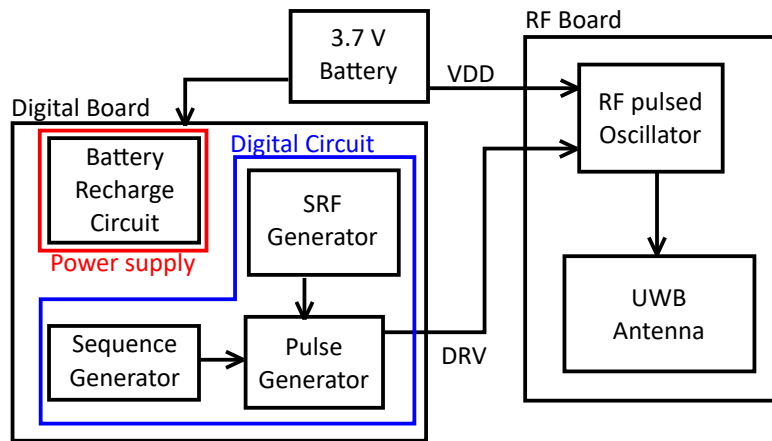


Figure 2.14: High level block diagram of the first Tag prototype

The power supply circuit schematic is shown in Figure 2.15. It is possible to notice that the Tag is powered using a 3.7 V lithium polymer (LiPo) rechargeable battery connected to the board through a Single Pole Double Through (SPDT) switch. To recharge the battery, it is necessary to power down the Tag. The digital and

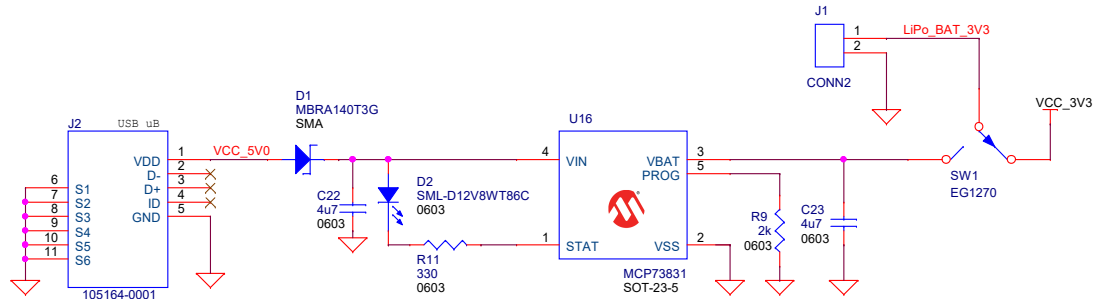


Figure 2.15: Tag power supply block schematic.

RF circuits are disconnected and the battery is directly connected to the recharge circuit.

The adopted charge management controller is the MCP73831 chip from Microchip [39]. To recharge the battery, it is sufficient to connect the Tag to a USB port; the charge management controller is directly power supplied by the USB port otherwise it is kept in power down mode. The duration of the recharging process can be reduced by increasing the current that the charge management controller provides to the battery. To do that, it is sufficient to change the resistance connected to the chip’s PROG pin (pin 5). We used a 2 k $\Omega$  resistor to set the maximum current value available: 500 mA. During recharge, a red LED is turned on and is kept lit until the battery is completely charged then turns off at the end of the recharge. The digital circuitry is timed by a 20 MHz reference clock signal generated by a basic CMOS crystal oscillator with feedback inverter and a decoupling inverter at the output. The schematic of the clock signal generation circuit is shown in Figure 2.16. The resulting 20 MHz clock is delayed with two cascading inverters with nominal delay of one nanosecond each. The delayed clock is used in the pulse generation circuit to window the sequence signal with a certain margin from the switching front. The digital block performs three distinct operations:

- Set the digital sequence to transmit;
- Manage the symbol and the frequency repetition timing;
- Generate the 2 ns pulses to modulate the oscillator carrier signal.

The schematic of the entire digital block is shown in Figure 2.17.

The generation of the sequence repetition timing (SRF) is performed in order to



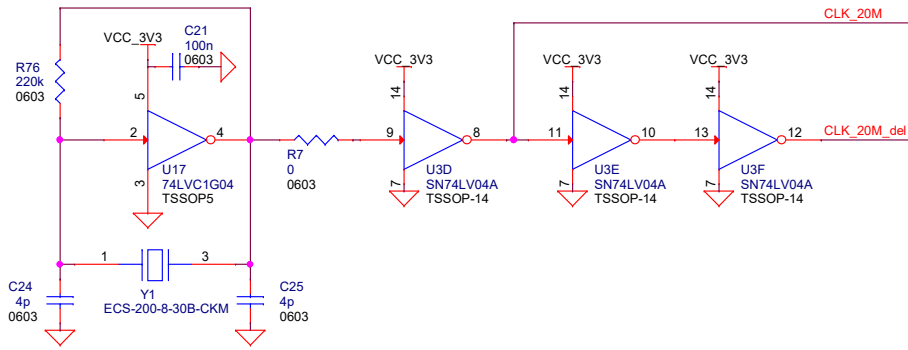


Figure 2.16: Clock generation circuit

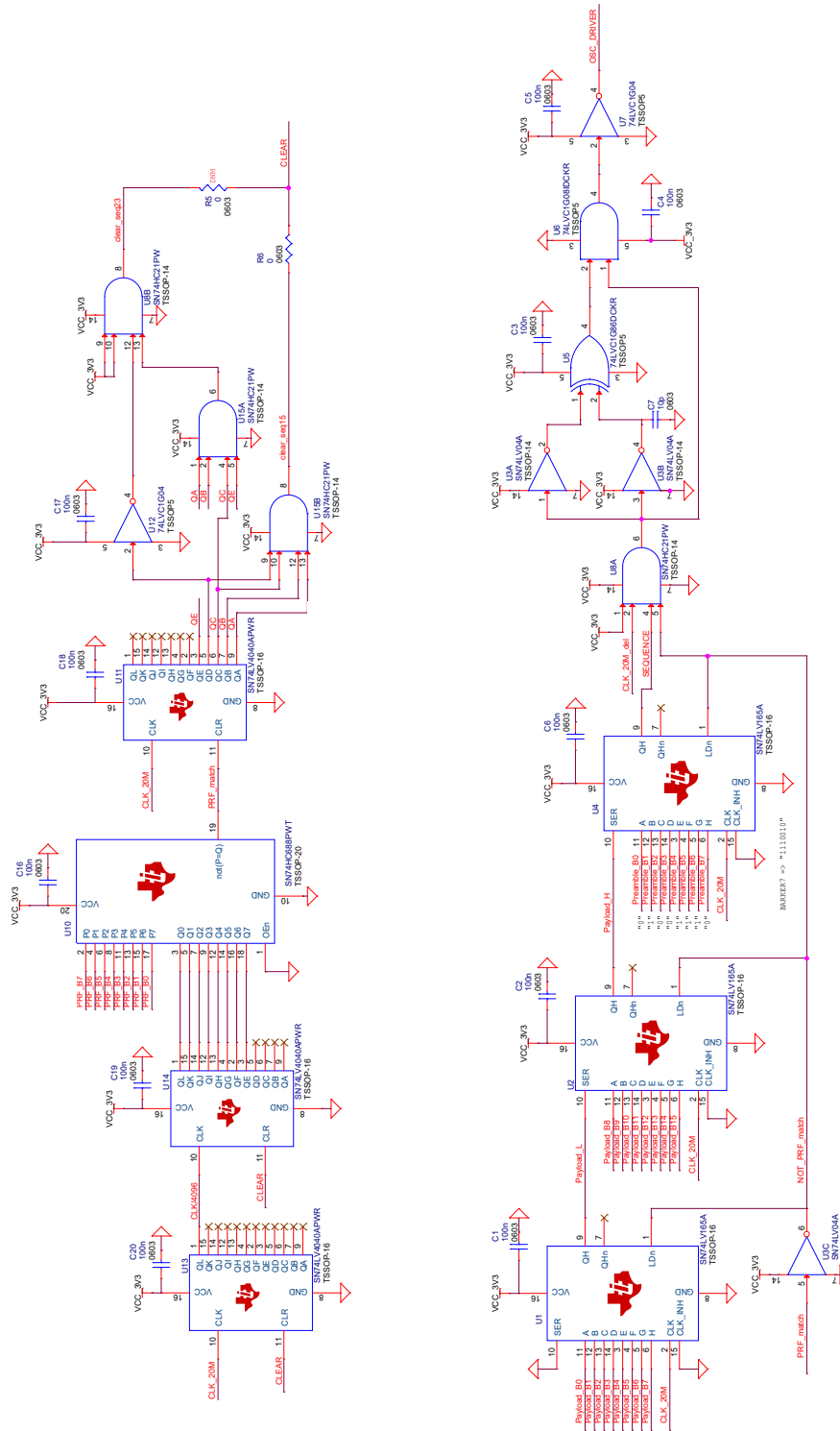


Figure 2.17: Digital circuit schematic showing the pulse sequence, SRF timing and the short pulse generation circuits.

allows a fast reconfiguration since the SRF needs to be adjusted as the number of Tags to track changes. If the total number of Tags to track is small, we can use high SRF thus increasing the number of computed positions per second. However, as the number of Tags increases, the processing load of the Sensor increases to the point where the Sensor is not able to complete the processing of a sequence before the reception of the next. To overcome this bottleneck we have to reduce the SRF of Tags linearly as the number of Tag increases.

To accommodate the solution for a large number of SRF, we apply the frequency division technique from the 20 MHz reference clock. The frequency division circuit is implemented by two cascaded 12-bit counters shown in the bottom left corner of Figure 2.17. The first counter is clocked using the 20 MHz reference clock signal coming from the clock generation circuit while the second counter uses as clock the Most Significant Bit (MSB) signal from the first counter. In this configuration, the LSB signal of the second counter is a square wave with frequency  $2^{12}$  times slower than the reference 20 MHz clock.

To further reduce the clock frequency, only the 8 most significant bits of the second counter output are taken into account. This operations further divide the obtainable output frequency by a factor  $2^4$  for a total reduction of the input clock frequency by 65536 times.

The desired Sequence Repetition Interval (SRI), defined as the inverse of the SRF, is hardwired by eight on board resistors (not shown in Figure). The resistor connection to ground sets a bit zero while the connection to the voltage supply sets the bit one. The eight bit vector obtained by the resistors configuration is provided as input together with the second counter output to a eight bit parallel comparator. When the counter output matches the SRI set by the resistors, the parallel comparator set to zero the not(P=Q) signal for one clock period resetting a third counter that counts the number of transmitted bits before triggering a new pulse sequence transmission. The trigger event can be configured to occur after 15 or 23 symbols, to take into account different payload dimension.

The re-transmission of the sequence triggers the digital sequence generator circuit, shown in the bottom right corner of Figure 2.17. The Tag sequence is hardwired with the same method used for the SRI, with resistors connected to ground or the supply voltage (not shown in Figure). The sequence is loaded in parallel into three cascaded shift registers. The serial output of the last shift register is provided to an AND gate together with the shift register's load signal and the delayed 20 MHz clock. The output of the AND gates is equal to the clock signal only when we have a bit equal to one in the sequence. This signal is called sequence signal and is provided to the input of the pulse generator circuit.

The driving signal that finally modulates the carrier frequency is a 2 ns baseband pulse generated using the pulse generator circuit shown in Figure 2.18. The implementation of short pulses may require very high speed and expensive hardware. Here, by using only discrete logic gates, we were able to maintain very low costs.

The sequence signal from the shift registers is provided at the input (IN) of two

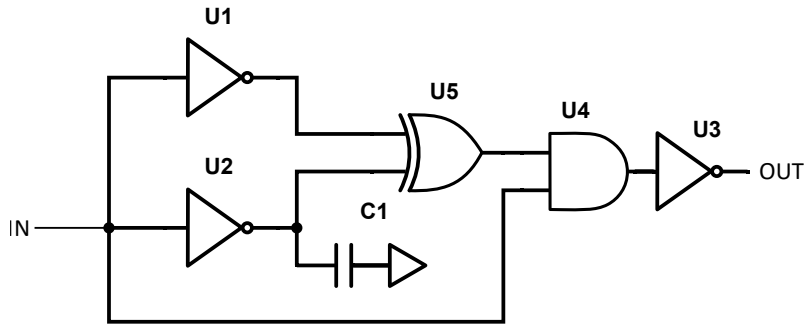


Figure 2.18: Low-cost 2 ns pulse generation circuit.

equal inverters U1 and U2. The second inverter has an additional capacitive load C1 at the output that allows to tune the gate delay arbitrarily. A value of 33 pF for C1 was sufficient to add a delay of 2 ns. The outputs of the two inverters are provided to a XOR gate (U5). The output signal of U5 is a pulse whose duration is proportional to the delay between the two inverter's outputs. The output of U5 is provided to into an AND gate together with the original sequence signal provided at the input of the circuit in order to filter out a second unwanted pulse exiting the XOR gate. The signal is finally inverted to be provided to the RF oscillator emitter. The timing details of the pulse generation circuit are shown in Figure 2.19. The time steps on the horizontal axis are equal to 2 ns.

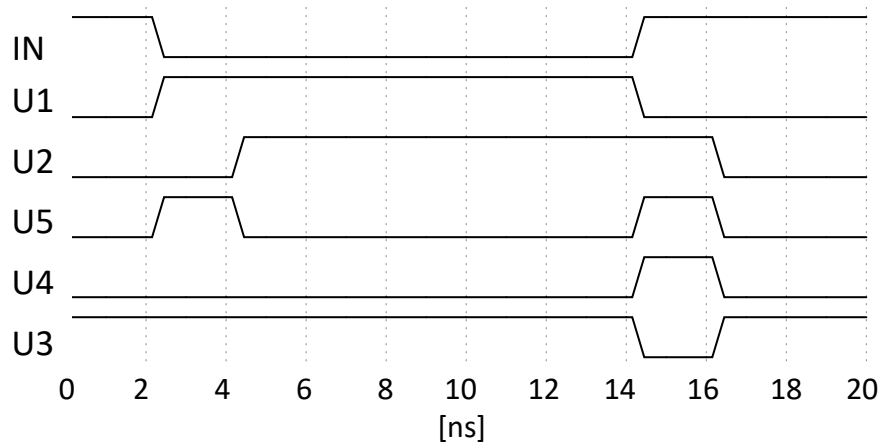


Figure 2.19: Timing of the pulse generation circuit used to produce the oscillator driving signal

### 2.3.3 Radio frequency oscillator

The core of the Tag is the RF oscillator that generates the 7 GHz carrier signal following a modulation command. The RF pulsed oscillator circuit topology and its design methodology are presented in [58] where antenna and oscillator are on two separated printed boards. The design follows the method used for negative resistance oscillators where the Barkhausen criteria are satisfied so that the imaginary part of the input impedance, the one seen from the base of the transistor, is  $\text{Im}(Z_{in}) = 0$  while the real part of the same impedance is  $\text{Re}(Z_{in}) < 0$ . To tune the resonance frequency is important to properly balance the reactance on the emitter to maximize the negative conductance at the base of the transistor. The oscillator operates in a common collector configuration; a command signal drives the emitter of the Infineon BFP740 SiGe-BJT [26] transistor while the output is taken from the collector and sent to the antenna. The final RF pulsed oscillator schematic is shown in Figure 2.20.

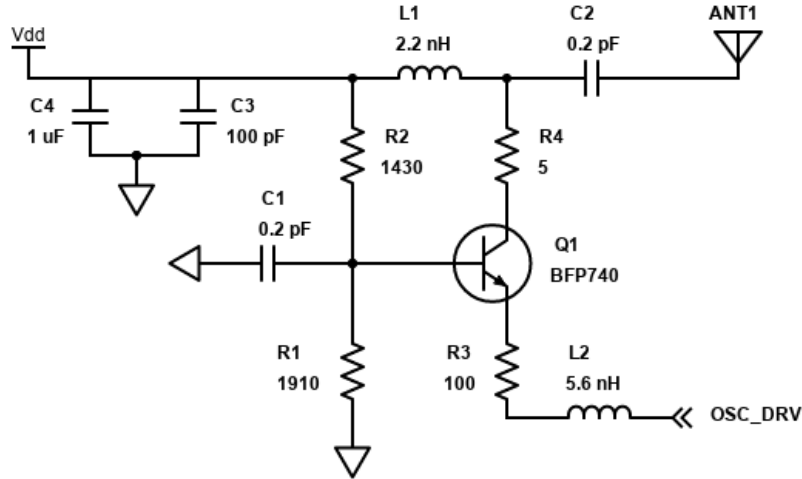


Figure 2.20: Schematic of the 7 GHz pulsed oscillator

The passive components values obtained during the design phase, are the starting point for the parameter tuning simulation phase to center the oscillating frequency at 7 GHz using a supply voltage of 3.7 V. The output signal is DC filtered and transmitted to a ceramic, surface mount UWB antenna. The chosen SMD Chip Antenna is manufactured from Taoglas. The datasheet reports a peak gain of 4 dBi and an efficiency of more than 50% across the 6 to 7.5 GHz bandwidth. The main advantage of such antenna is not only its small dimensions (5.5 mm x 5.5 mm x 2 mm), but also its good omnidirectionality in the YZ plane. The relevant cuts of the antenna radiation pattern are shown in Figure 2.21. The orientation of the coordinates system adopted to obtain the radiation pattern cuts is shown with the antenna evaluation board in Figure 2.22. The omnidirectionality of the antenna

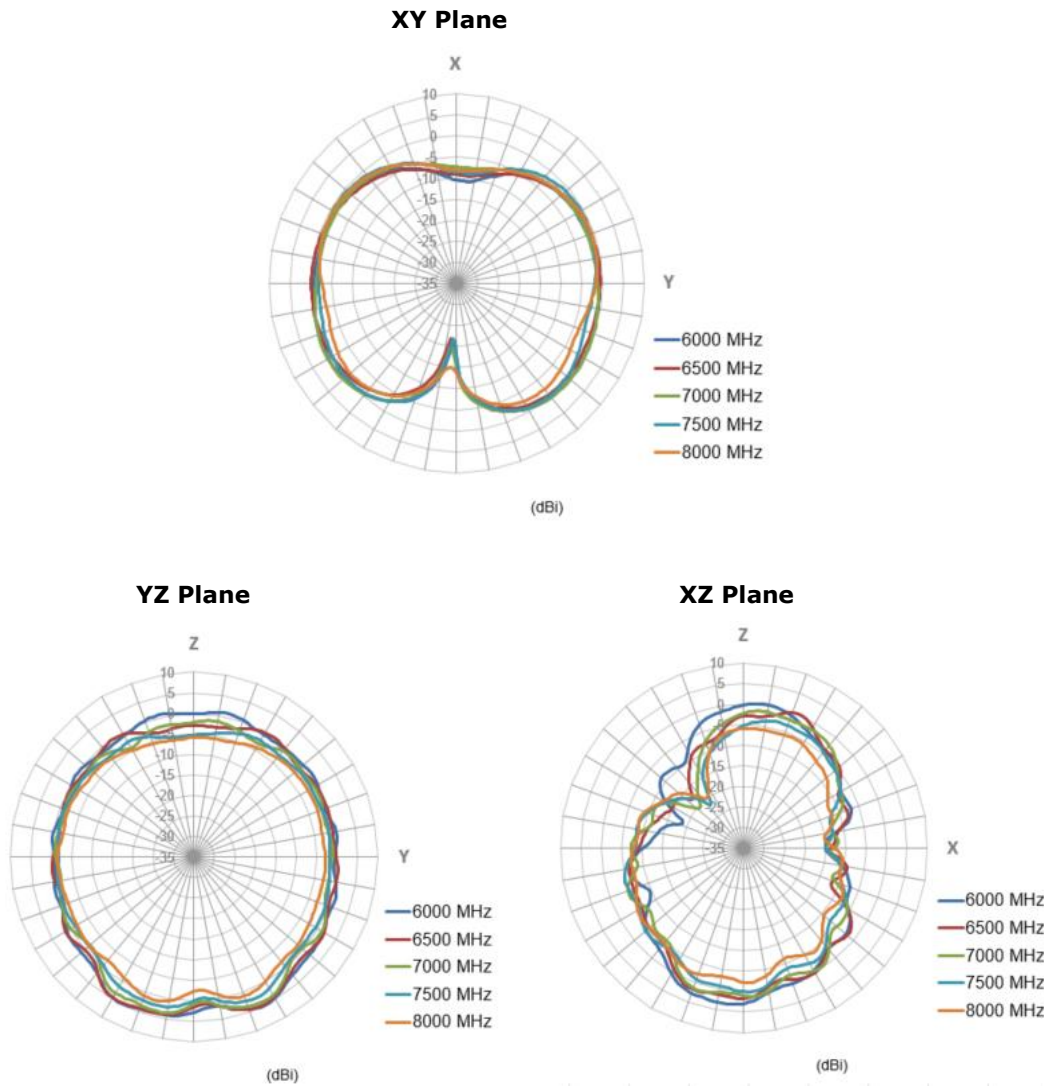


Figure 2.21: Radiation patten of the Taoglas ceramic chip antenna. The images are taken from the component datasheet

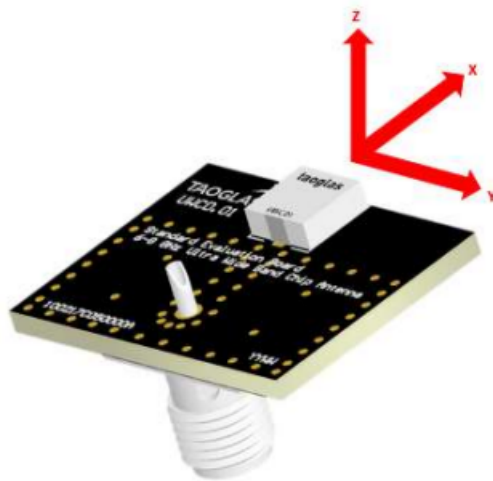


Figure 2.22: Evaluation board of the Taoglas antenna with the detail of the coordinates system orientation.

is an essential requirement for the Tag since it needs to transmit its signal in any possible direction.

### 2.3.4 PCB design

The Tag's PCB accomodates RF elements together with digital circuits. Great care must be taken into account to guarantee the correct behavior of the highly sensitive oscillator circuit. The manufactured PCB stack-up is shown in Figure 2.23. The board has four layers, the top layer host the RF part of the circuit i.e. the oscillator and the antenna, while the bottom layer hosts the digital part and power supply circuit.

The top layer (L1) and the RF ground (L2), are 35  $\mu\text{m}$  copper sheets attached to a 20 mils (508 $\mu\text{m}$ ) core of RO4350B, the same used for the Sensor RF receiver. The second core is made of the same material of the first one and has the same thickness. The two cores are pressed together and connected using 3 sheets of Panasonic R1551 7680 prepreg material with a total thickness of 480  $\mu\text{m}$ .

The oscillator circuit revealed to be very sensitive to capacitive and inductive

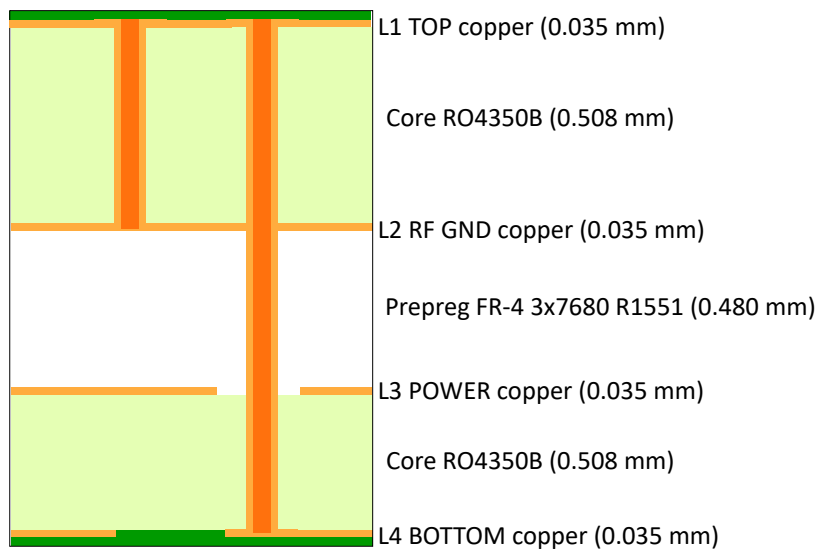


Figure 2.23: Tag Stack-Up

parasitic elements introduced by the board and vias. To reduce the parasitic capacitance of the vias, we reduced the diameter of anular ring in each layer and increased their clearance. The parasitic inductance can be reduced using shorter vias like blind or buried vias, depending on their position along the cross-section of the board. To minimize the parasitics elements we used blind vias for all the ground connections required by the circuit and, to further reduce noise coupling on the driving signal and on the oscillator itself, we places a cage of vias around the



oscillator circuit and slotted the ground plane to bind the ground return signals. The region below the antenna is completely cleared from copper as required by the antenna datasheet.

The final outlook of the first prototype is shown in Figure 2.24. It is possible to

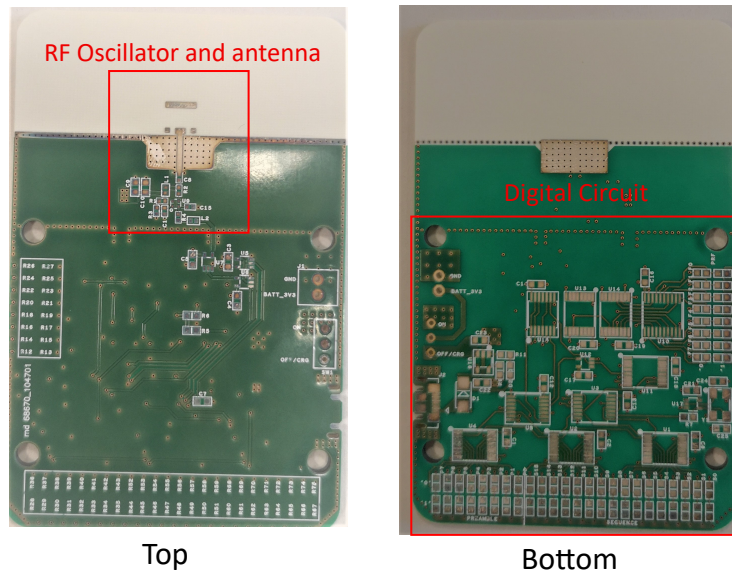


Figure 2.24: Layout of the first prototype bare PCB. The full circuit was never mounted since the oscillator was unstable due to parasitics.

notice the RF oscillator and antenna footprint on the top side and the digital circuit layout on the bottom side of the PCB. Unfortunately, it was impossible to stabilize the oscillator frequency to the required value due to the strong effect of the parasitics inductance and capacitance in the PCB.

The Tag has been redesigned, separating the RF oscillator from the rest of the digital circuit. The oscillator is connected to an external printed elliptic dipole antenna instead of the SMD one. The new Tag prototype is shown in Figure 2.25. The separation of the RF oscillator from the digital PCB solves the stability problem and allows to test the Tag in nominal working conditions. The new antenna provides higher gain and better omnidirectionality than the SMD one. Further details on the antenna design and characteristics will be discussed in the following sections on elliptic dipole antenna design.

The first prototype reached the goal to demonstrate our capability to localize transmitting Tags with the required accuracy. During the test phase the following considerations were raised:

- The Tag’s power consumption was too high and put a great constraint on battery capacity and duration to have a reasonable ON time without changing the battery. The Tag needs to be low power;

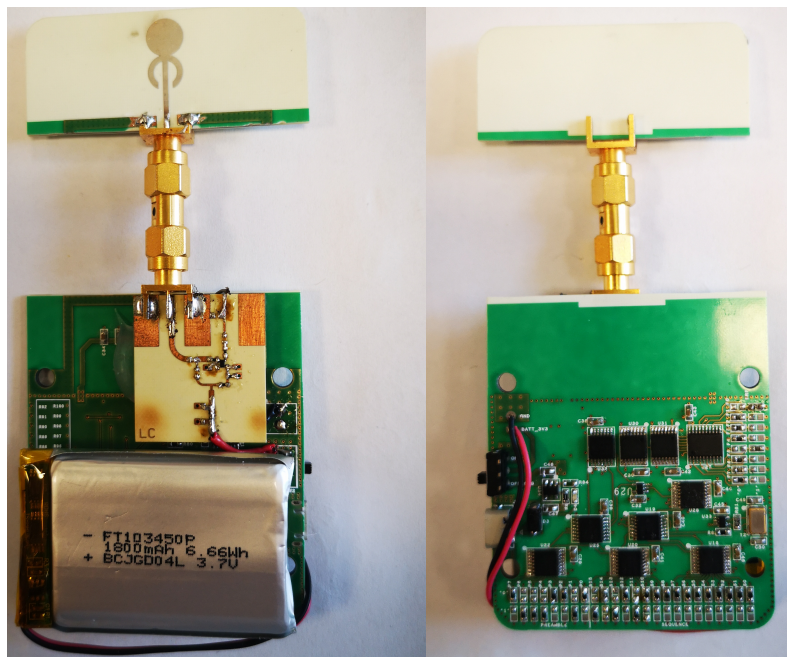


Figure 2.25: The new implementation of the first prototype moves the RF oscillator on a dedicated two layer PCB and uses an external antenna.

- The idea of separating the RF oscillator and antenna from the digital PCB proved to be effective reducing the board parasitic elements with respect to the single board solution;
- The signal radiated from the SMD antenna was weak, significantly limiting the maximum range of localization. The new antenna design based on the custom printed antenna demonstrated to be effective;
- The cost of a four layer stack-up PCB based on RF materials made the prototype too expensive.

## 2.4 Second Low-Power Tag Prototype

The issues encountered and partially solved with the first prototype drove the design of a second Tag's prototype.

To effectively reduce the power consumption we introduce a power gating technique. This method allows to shut down the entire Tag few microseconds after the transmission of the pulse sequence and to turn it on again before the following transmission. This method also allows to significantly reduce the hardware components required for the digital control circuit.

To face the parasitic issue we opted for a drastic change in the Tag architecture. Instead of having an SMD antenna on the same PCB or a connectorized antenna, we separated the oscillator-antenna circuit from the digital one. This solution also allowed to reduce both the dimensions and the cost of the Tag.

To increase the signal output we implemented two major changes. The first one was moving from an off the shelf ceramic chip antenna to a custom UWB, omnidirectional printed antenna with higher gain while the second was to increase the oscillator voltage supply to 5 V and fine tune the component values.

The results obtained for the second low-power Tag prototype have been published by the author in [3].

### 2.4.1 Block Diagram

The solution to separate the Tag board into two, the RF board and the digital one has been maintained for the second and final version of the Tag. The RF board hosts the oscillator and a new designed antenna: a linear, vertically polarized, elliptic printed dipole. The digital board manages the voltage supply generation, a power gating circuit, the generation of the pulse sequence and the sequence repetition frequency selection.

The detail of each block are explained in the next sections. The second prototype's high level block diagram is shown in Figure 2.26.

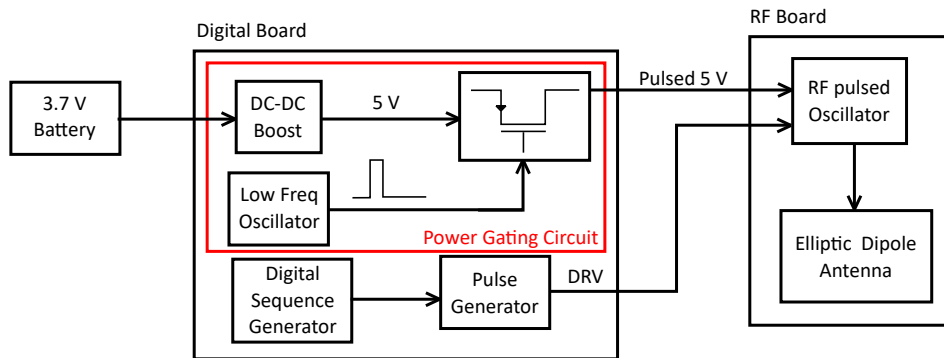


Figure 2.26: Tag's high level block diagram. The digital board generates the pulsed voltage supply and the modulating pulse sequence for the RF oscillator.

## 2.4.2 Power gating implementation

Power gating is the most efficient solution to drastically reduce the power consumption in case of a low duty cycle transmitter. The circuit schematic is shown in Figure 2.27. The 3.7 V supply voltage is obtained from the same rechargeable Lipo battery used in the previous prototype. The battery voltage is up-converted using a switching DC-DC boost regulator from 3.7 V to 5 V. The higher supply voltage allows to obtain a higher RF output signal.

The output of the DC-DC converter is provided to a TPS22917 switch from Texas

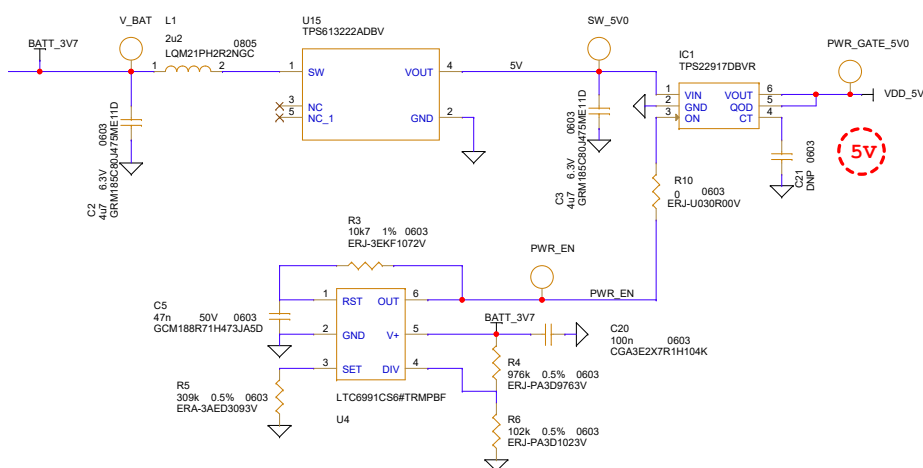


Figure 2.27: Power gating circuit implemented to reduce power consumption. Only the LTC6991 low frequency oscillator and the DC-DC converter are always powered on.

Instruments, the datasheet is available at [28]. The On-Off state of the transistor is controlled by the LTC6991 low frequency, oscillator from Analog Devices [14]. This component is the key element of the power gating circuit and allows us to create low frequency, low duty cycle waveforms. The SRF circuit is here simplified since the repetition frequency corresponds to the LTC6991 frequency of oscillation, set by the value of R5. The product of R3 and C5 sets the duration of the pulse that drives the switch.

The entire transmission of a single 15 bit sequence lasts 750 ns, but in order to take into account the charge and discharge time of the power gating circuit, we have to set the driving pulse duration to a minimum value of 470  $\mu$ s. In these conditions, both the RF oscillator and the digital circuit are power supplied for only 0.94% of the time for a typical SRF of 20 Hz (corresponding to a sequence repetition interval of 50 ms). If the application allows, the SRF can be reduced to less than a repetition per second, further reducing the energy consumption.

### 2.4.3 Elliptic Dipole Antenna design

The antenna connected to the RF pulsed oscillator is a microstrip elliptic dipole printed antenna in linear vertical polarization. Microstrip printed antennas are inexpensive compared to ceramic chip ones available on the market and have comparable performances.

An experimental study for UWB elliptic dipole antennas is presented in [37]. In this study only FR-4 and high dielectric constant materials are used. We adopted the same design methodologies but we selected RF materials with different dielectric constant. To extend the bandwidth of elliptic dipole antennas it is sufficient to change the ratio between the minor and the major semi-axis of the ellipses.

Starting from the elliptic configuration, we optimized the geometry for our 20 mils RO4350B substrate by means of CST MicroWave studio simulations. A unitary ratio between the semi-axis was sufficient to cover the required bandwidth of 500 MHz. The final layout and the geometrical parameters of the optimized antenna are shown in Figure 2.28.

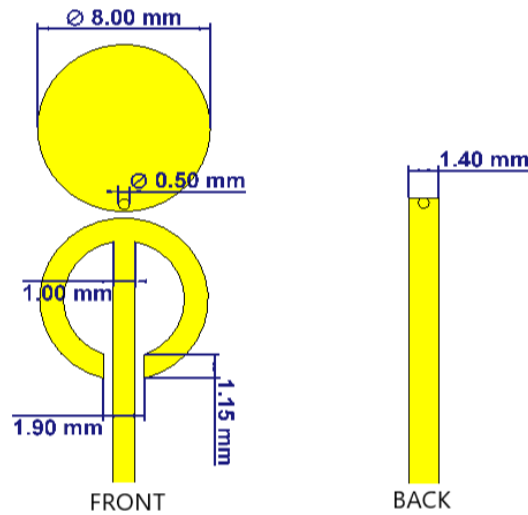


Figure 2.28: Detail of the antenna geometric parameters, front and back view of the antenna.

### 2.4.4 UWB Antenna and RF Oscillator Co-Simulation

The simulations of the antenna and RF pulsed oscillator module have been performed using CST Microwave Studio adopting the EM/circuit co-simulation method [49]. With this method we are able to integrate the PCB, surface mount components and antenna to the system analysis, to estimate their effects, and to optimize the passive SMD components value accordingly. A RO4350B core 0.508 mm thick has been adopted as dielectric substrate for the PCB hosting the RF

pulsed oscillator and the antenna.

Examples of usage of this method are presented in [42, 43]: in both cases, the co-simulation method allows to integrate non linear component and SMD components in a 3D model. The EM simulation is configured to have a dedicated port for each component in the PCB and to generate the complete scattering matrix of the 3D model. The farfield radiation pattern results can be obtained by feeding only the antenna port.

The layout of the simulated PCB is shown in Figure 2.29. The top view shows the

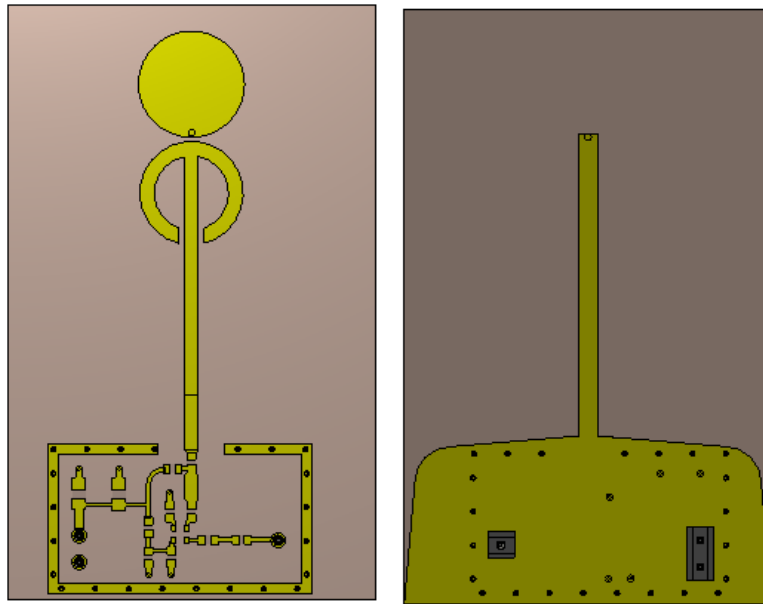


Figure 2.29: Top and bottom view of the PCB 3D model

printed dipole antenna, the oscillator circuit and the guard of ground vias while the bottom view shows the ground plane and the pin headers used to power the board and to provide the driving signal from the digital board. The EM simulation allows us to have a full description of the PCB behavior and to evaluate its effects on the Tag functionality.

For the complete Tag simulation we connect the Gummel Pool SPICE model of the transistor [26] and of the models of all the components to the PCB n-port block generated by the EM simulation and perform a transient simulation. The complete schematic of the implemented model for the transient simulation is shown in Figure 2.30. The central block represents the PCB and is modeled with its scattering matrix. The different components are connected to the ports as ideal components like for example the voltage supply and resistors, or with their Spice model. The part number and value of the components adopted in the simulation are reported in Table 2.1.

The co-simulation allows us to estimate the antenna radiation pattern and the

Table 2.1: Details of the components adopted in the simulation.

Component	Value	Part Number	Manufacturer
C1, C2	0.2 pF	GCQ1555C1HR40BB01	Murata
C3	100 pF	GCG1885G1H101JA01D	Murata
C4	1 uF	GRT188C81A105KE13D	Murata
L1	2.2 nH	LQG15HH2N2B02D	Murata
L2	5.6 nH	LQG15HH5N6C02D	Murata
R1	1.91 k $\Omega$	ERJ-2RKF1911X	Panasonic
R2	1.43 k $\Omega$	ERJ-2RKF1431X	Panasonic
R3	100 $\Omega$	ERJ-U02F1000X	Panasonic
R4	5 $\Omega$	ERJ-U02F5R10X	Panasonic
Q1	BFP740	BFP740FH6327XTSA1	Infineon

behavior of the output signal provided to the antenna. The radiation pattern results obtained simulating the farfield at 7 GHz are shown in Figure 2.33 with the blue dashed curve. The first two plots represent the  $\phi = 90^\circ$  and  $\phi = 0^\circ$  cuts while the third is the equatorial cut  $\theta = 0^\circ$  (as shown in Figure 2.32 where the z axis is parallel to the antenna polarization vector). The antenna main lobe is slightly tilted upwards and radiates almost uniformly in all  $\phi$  directions. The transient simulations results are shown in Figure 2.31. The blue dashed curve represents the command signal, simulated as 2 ns square pulse with 100 ps rise and fall times and 4.5 V amplitude; the red curve is the RF pulsed oscillator output. The signal has a peak to peak amplitude of 1.5 V and reaches the 90% of the maximum amplitude in 2-3 carrier frequency periods. The circuit behaves as intended generating a 2 ns pulse at 7 GHz.

### 2.4.5 Measured Results

The Tag was manufactured, assembled and measured. The final prototype is shown in Figure 2.32. It is possible to distinguish two different boards connected one on top of the other, the smaller one is the RF PCB with the pulsed oscillator and printed microstrip antenna while the other is the digital circuit board. The rechargeable battery is positioned on the bottom side of the digital circuit. The system dimensions are 75 mm  $\times$  55 mm  $\times$  10 mm making the whole Tag smaller than a credit card. The Tag does not require any programming since all parameters



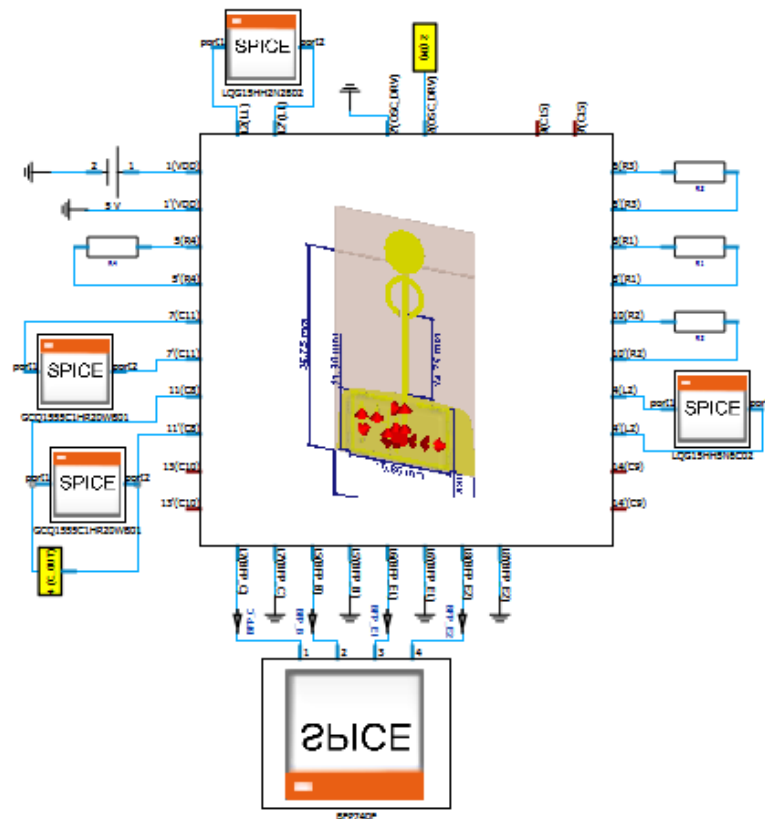


Figure 2.30: CST transient simulation schematic. The 3D model used in the EM simulation is instantiated as an N-port component.

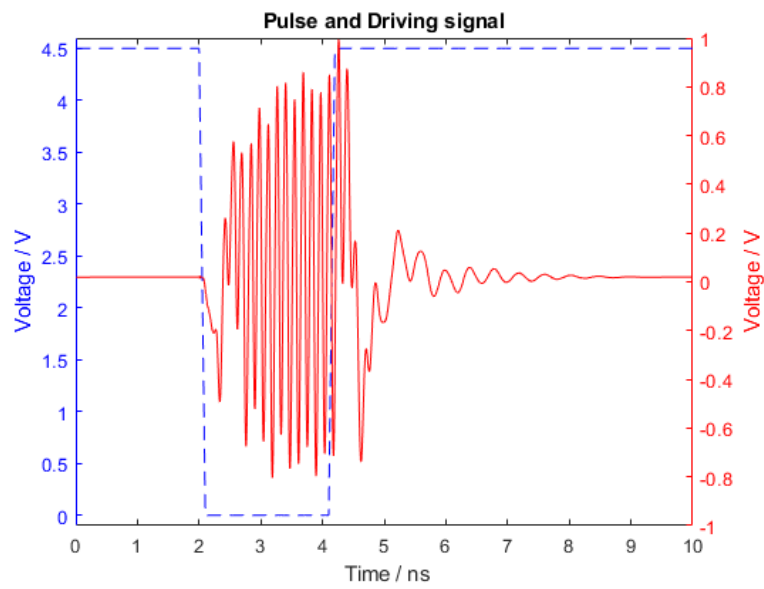


Figure 2.31: Voltage of the RF output across the output capacitor (red), and the driving signal (blue)

(Tag sequence and SRF) are hardwired.

To evaluate the radiation pattern, the Tag has been measured in the anechoic

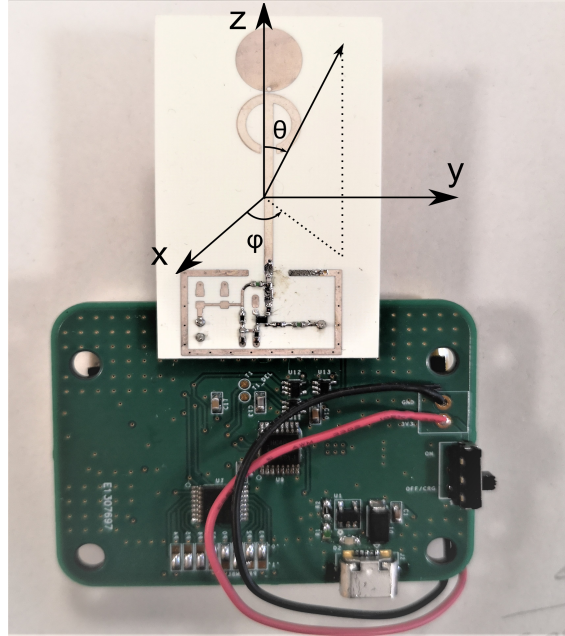


Figure 2.32: Front view of the assembled low-power Tag prototype with the oscillator and antenna board connected to the digital control circuit.

chamber of our Institution. To perform this operation, the driving signal on the oscillator emitter has been fixed to ground setting the oscillator to work in Continuous Wave (CW). The measured radiation pattern is shown in Figure 2.33 with the red solid curve. The measurements results have been post-processed in Matlab using a cubic spline interpolation to reduce noise and have been normalized to the measured transmitted power of 7.5 dBm to properly compare with the simulated radiation patterns. The comparison between simulations and measurements shows an excellent agreement.

To measure the amplitude of the transmitted signal, we set the RF pulsed oscillator back into the standard working mode connecting it back to the digital control circuit. The signal radiated by the antenna was measured using a receiving antenna probe connected to an high frequency oscilloscope.

To perform this measurement, we first need to quantify the antenna probe performances performing a calibration. The calibration has been done by comparing the output signal of an RF signal generator transmitting 0 dBm in two cases:

- Direct connection between the RF generator and the oscilloscope through a coaxial cable;

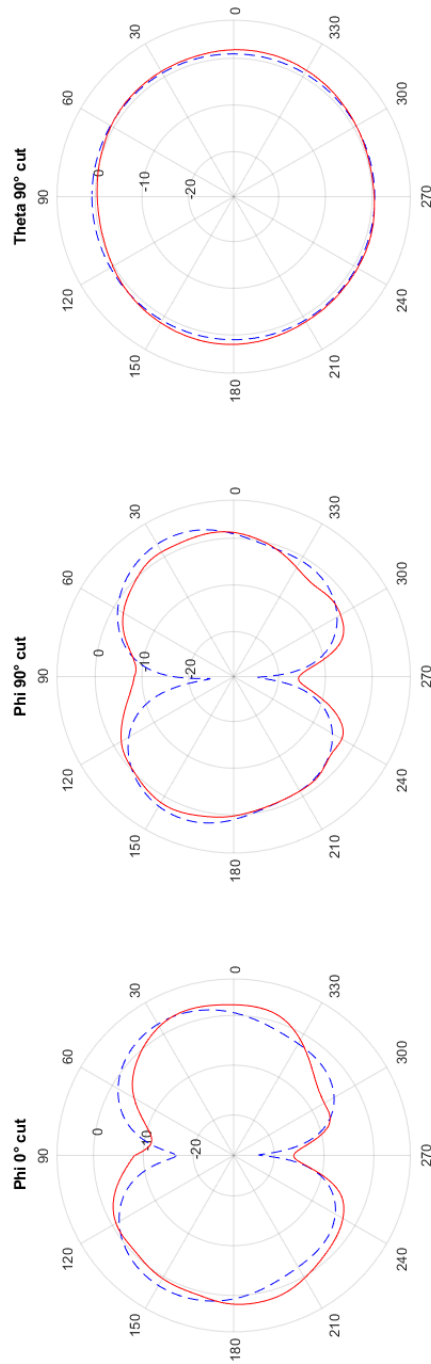


Figure 2.33: Comparison between the simulated radiation pattern (dashed line) and the measured one (red curve).

- Over The Air (OTA) configuration where the RF generator has been connected to the Tag antenna and spaced apart by a known distance from the probe antenna.

The signal measured in the two cases have been compared and by taking the difference between the two we have estimated the overall losses of the measurement setup. The estimated losses are equal to  $L = 9.5$  dB.

The Tag signal measured with the receiving antenna probe at the same distance from the Tag antenna is shown in Figure 2.34. The driving signal in this case was

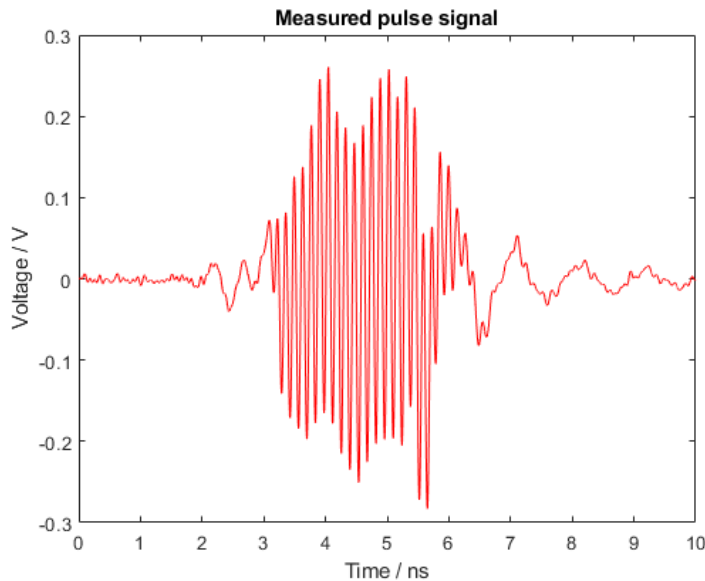


Figure 2.34: Voltage of the radiated pulse signal measured on the oscilloscope with the antenna probe placed at 1 mm from the Tag antenna.

provided by the digital control circuit. The measured signal amplitude is comparable with the simulation results shown in Figure 2.31 once the calibrated losses, equal to 9.5 dB, have been taken into account. The oscillation frequency has been measured to be equal to 7 GHz.

It is possible to notice a certain amplitude modulation (AM) over the pulse signal duration due to harmonics in the driving signal. The comparison between simulation and measurement results showed an excellent agreement.

The last test performed to validate the design is the power consumption and battery duration estimation. The Tag has been connected to a laboratory power supply providing 3.7 V with a high precision series, sensing resistor. We have evaluated the absorbed current by measuring the voltage across the sensing resistor using an oscilloscope, and dividing it by the sensing resistor value. The detail of the current absorbed along the 470  $\mu$ s interval while the voltage supply is provided to the circuit is shown in Figure 2.35.

We have computed the average current absorption over the entire duration of the voltage supply and we have obtained an average power for a single sequence transmission equal to:

$$P_{avg} = R \cdot I_{avg}^2 = 12.3mW. \quad (2.4)$$

The energy absorbed is equal to the average power of a single sequence transmission multiplied by the voltage supply pulse duration, in our case, 470  $\mu s$ .

$$E_{singleTX} = P_{avg} \cdot \tau = 6\mu W s \quad (2.5)$$

In our case the SRF is set to 20 repetitions per second leading to a total energy absorption per second of  $E_{abs} = 120 \mu W s$ . The prototype is equipped with a 3.7 V

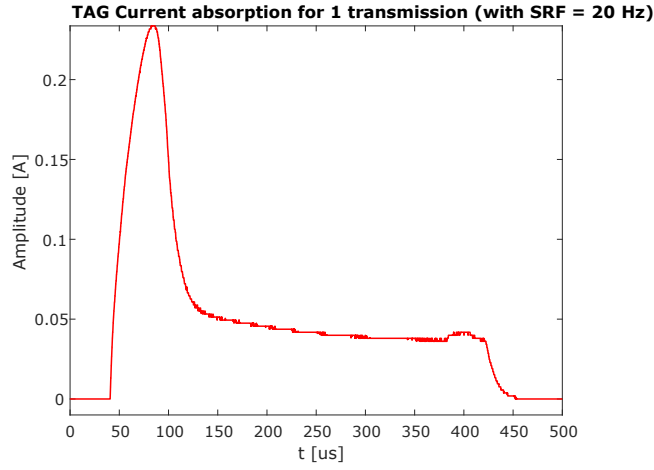


Figure 2.35: Measured waveform of the current absorbed from the battery, with focus on the 470  $\mu s$  during the On state of the circuit. The waveform is obtained averaging 64 successive transmissions.

Lipo battery with 1800 mAh current rating, for a total of 6.6 Wh. The ratio between the battery energy rating and the absorbed energy per second brings to a rough estimation of the battery duration equal to about 6 years. The energy absorbed by the system during the off time is negligible. These results can be further improved for certain RTLS applications that allow less than 20 transmissions per second, in these cases the battery life increases further or maintains the same performances with smaller, lower current rated batteries.

## 2.5 Processing board prototype overview

The first prototype of the processing is developed using off the shelf hardware connected to our custom UWB receiver.

The prototype was assembled connecting together the following boards:

- Zedboard, a development board mounting the Xilinx’s ZYNQ 7000 family System on a Chip (SoC); user guide available at [65];
- HMCAD1511 ADC evaluation board from Analog Devices; datasheet available at [25];
- ADF4360 evaluation board from Analog Devices mounting the Voltage Controlled Oscillator VCO chip used to drive the ADC; datasheet available at [24];
- The custom UWB receiver board presented in the Ultra Wideband Receiver section.

The Zedboard mounts the XC7Z020-1CLG484CES, all programmable SoC from Xilinx. The chip contains an FPGA architecture and a two core ARM processor. The SoC, allows the developer to rely on a unified tool chain simplifying not only the software and hardware design, but also the programming and booting operations. Moreover, since the processor is physically inside the chip, it is not necessary to instantiate it into the FPGA logic then saving important resources. When compared to other FPGAs from the same manufacturer, the Zynq SoC offers for our application the best trade-off in terms of number of logic elements, performances and, most importantly, cost. A detailed comparison between the Xilinx FPGA families is shown in [66].

The Zedboard offers a large variety of peripherals connected to the SoC. The board’s block diagram is shown in Figure 2.36. Among all the peripheral connected to the chip, only a small number of them are required for our application. This is the reason why the unnecessary ones will be eliminated during the design phase of the custom processing board, planned for the second prototype.

The peripherals required by our application are the followings:

- The QSPI and MMC SD card interface, used for booting operations and storage. Both solutions are implemented to allow the user to choose between the two interfaces depending on his needs and to eventually store acquisition’s data;
- The Gigabit Ethernet controller, to manage the data transfer between the Sensors and the host PC running the application;
- The USB to UART interface as a backup and debug communication channel;
- The DDR3 memory interface, to connect the RAM necessary for the application;
- The JTAG and GPIO interfaces, used respectively for real-time programming and configurations;

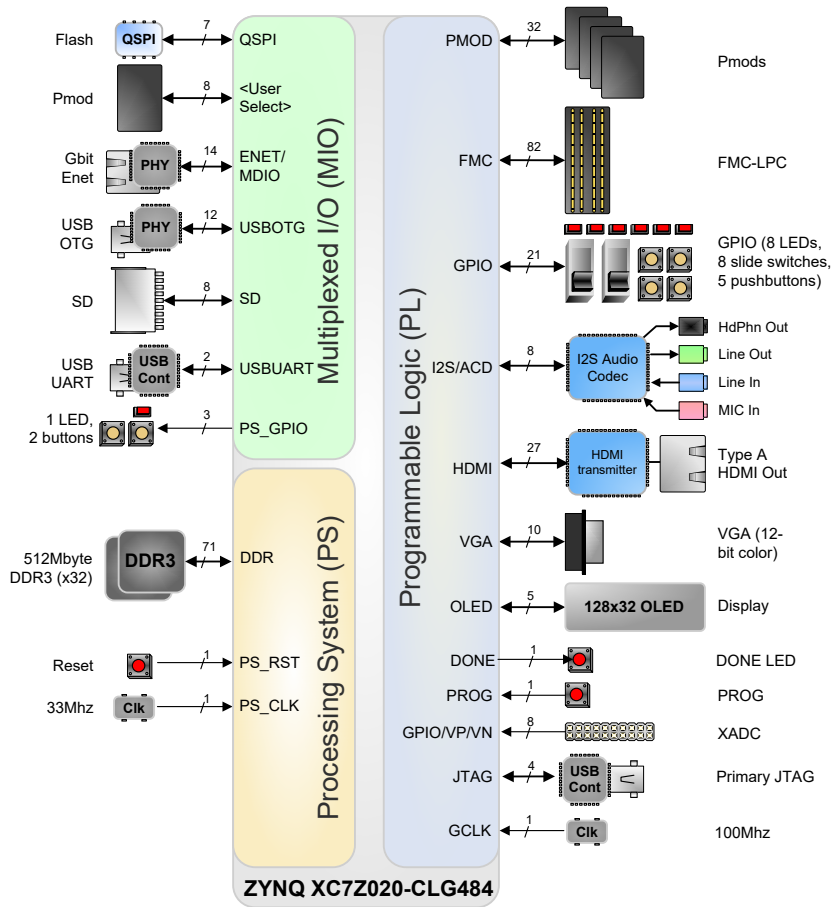


Figure 2.36: Zedboard block diagram



- The FMC-LPC high speed interface for high speed, high throughput data communications with the ADC.

The Zedboard and the ADC evaluation board are connected through the FMC-LPC connector (FPGA Mezzanine Card Low Pin Count). The ADC is a crucial component and its choice must be taken with great care. The maximum achievable sampling frequency directly influences the localization accuracy of the entire system. Higher the sampling frequency, the better the accuracy but costs increase dramatically.

The goal of the system is to reach 10 cm accuracy. It is necessary to find a trade-off between the hardware and software constraints to reach the target accuracy. The sampling frequency value fixes the raw accuracy of the system. Since we cannot separate TOA smaller than the sampling interval, we are bounded to an accuracy of:

$$\delta R = c \cdot \tau \tag{2.6}$$

Where  $\delta R$  is the accuracy in the distance estimation for each Sensor,  $c$  is the speed of light and  $\tau$  is the sampling period. In the case of 1 GHz sampling frequency we have 1 ns of sampling time fixing the hardware raw accuracy to 30 cm. Thanks to multi-lateration algorithm and measurements averaging performed by the software processing, we will be able to improve the accuracy down to 10 cm.

The ADC board hosts the HMCAD1511 ADC chip. This component has four channels sampling at 250 MHz that can be used combined in a single channel. This value of sampling frequency is an optimal choice since it maximizes the hardware performances without dramatically increasing the board costs. The block diagram of the ADC is shown in Figure 2.37. The samples are represented with a resolution of 8-bits. The sampled data are digitally amplified by an internal programmable amplifier and are provided to the output, according to the LVDS standard, in eight differential data lanes together with two differential clock lanes LCLK, the bit clock and FCLK the frame clock. The chip is configured using an SPI interface.

We need to provide the ADC sampling clock by an external board hosting the ADF4360-7 PLL whose block diagram is shown in Figure 2.38. The chip is an integrated integer-N synthesizer VCO programmable using an SPI interface to be programmed and it can generate an RF sinusoidal signal from 350 to 1800 MHz. To tune the output oscillating frequency it is sufficient to change the value of two external inductors accordingly. The chip is programmed to generate a 1 GHz differential RF clock signal a 10 MHz reference clock generated with a Voltage Controlled Temperature Compensated Crystal Oscillator (VCTCXO).

The assembled prototype is shown in Figure 2.39.

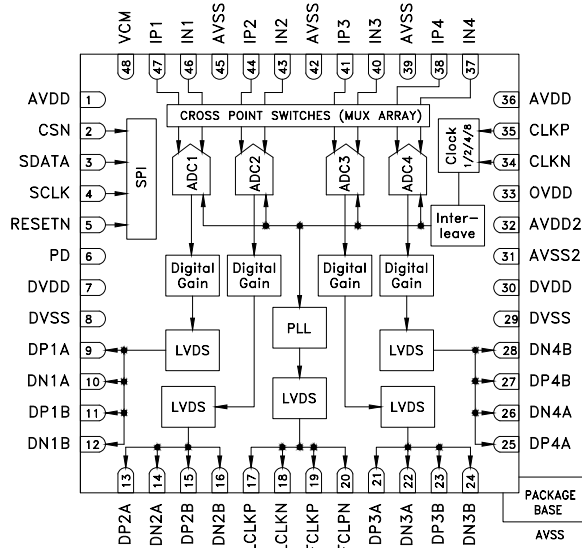


Figure 2.37: HMCAD1511 1 GSPS ADC block diagram.

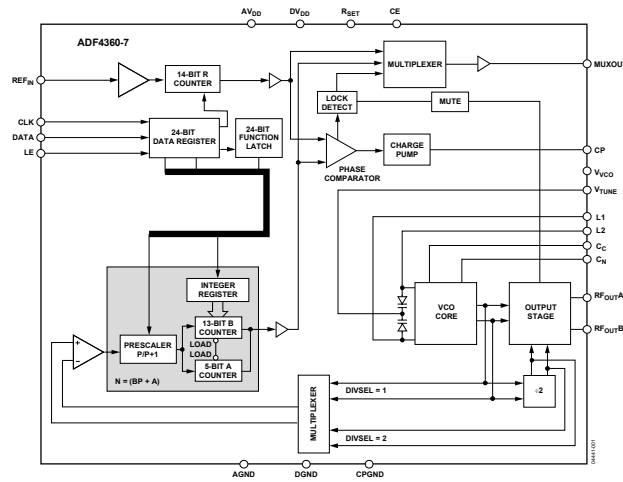


Figure 2.38: ADF4360-7 VCO chip block diagram.

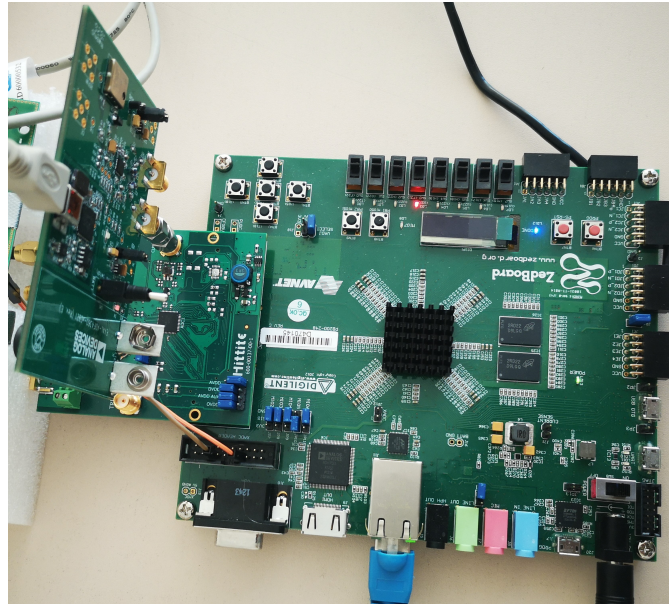


Figure 2.39: The first Sensor prototype where the evaluation boards of the Xilinx SoC, the ADC, and VCO are connected together.

The connection of the ADC and FPGA boards is performed using the FMC connector while the PLL board, vertically mounted, is connected to a coaxial connector through a barrel adapter. The UWB receiver and the receiving antenna are connected to the ADC board through a coaxial cable and are powered by a 5 V voltage rail available on the Zedboard. The ADC needs a 1.8 V voltage supply that is obtained from the 3.3 V voltage available on the FMC connector, stepped down to 1.8 V. The VCO board is power supplied through an external USB cable.

## 2.6 Sensor final prototype

The first Sensor prototype allowed software development as well as the initial field tests to validate the complete system. The goal is to move from a sensor based on COTS Evaluation boards to an integrated, cost and performances optimized custom sensor board. The development of the Sensor board brought also a change in the architecture of the system.

The new processing board has been designed in collaboration with an external partner company. The development of the schematic and the testing of the new board were agreed on a common basis while the layout and manufacturing were performed by the partner company. The processing board embeds the ADC chip and the related signal conditioning circuitry together with the VCO and the SoC.

The final board has 10 layers; it is 12 cm wide 12 cm long and hosts the UWB receiver on its top as a daughter module.

The whole system is enclosed in a dedicated box that shows, on a side, the UWB receiving patch antenna connected to the UWB receiver through an SMA adapter. In the following sections we discuss the new block diagram and we show the final prototype.

### 2.6.1 Block diagram

The block diagram of the new processing board is shown in Figure 2.40. The

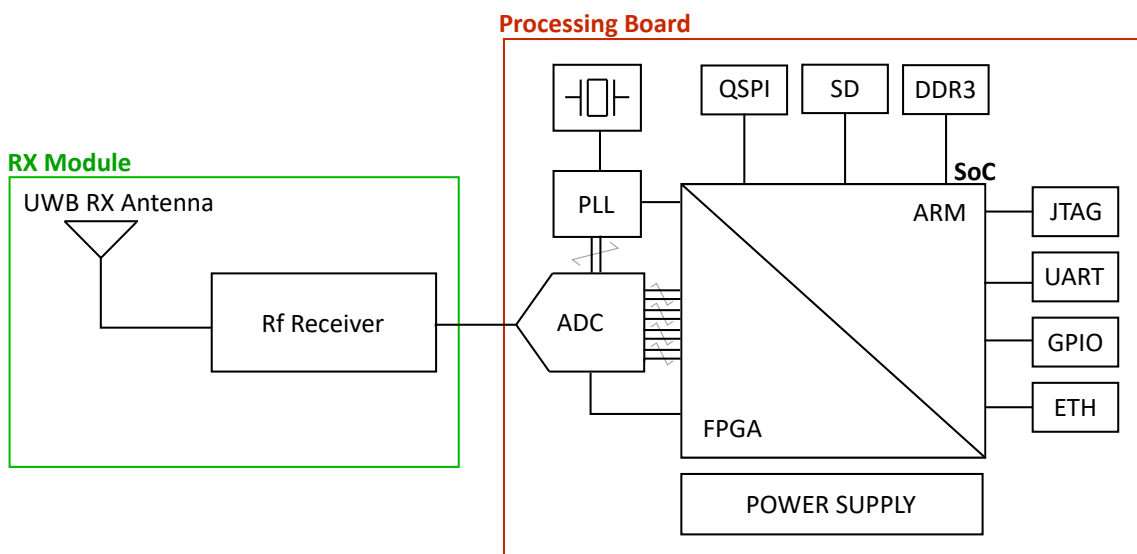


Figure 2.40: The Final Sensor prototype

high level block diagram is similar to the one of the previous prototype. The sampling clock generator (PLL) is hosted on board. The differential data from the ADC are directly connected to the FPGA side of the SoC. The peripheral connected to the ARM side of the SoC are the same that were used in the previous prototype accomplishing the same tasks as before. Some extra GPIOs are added to enable or disable the LDO chips providing the voltage supply to the analog and RF parts of the system. The final prototype assembly is shown in Figure 2.41.

The solution is a compact stack of the UWB receiver module mounted on top of the processing board. The connection is present in two points: the power supply connector, through which the UWB receiver module gets the 5V voltage and at the SMP connector that connects the signal from the UWB module to the ADC.

The first prototype Sensor required its own dedicated AC/DC power supply and an ethernet cable for data connection. The new Sensor take advantage of Power

over Ethernet (PoE) to greatly simplify the system architecture and installation. It is sufficient to use a PoE switch an ethernet cable and a splitter adapter for each Sensor. The final aspect of the Sensor enclosed in the box with the PoE splitter adapter is shown in Figure 2.42.

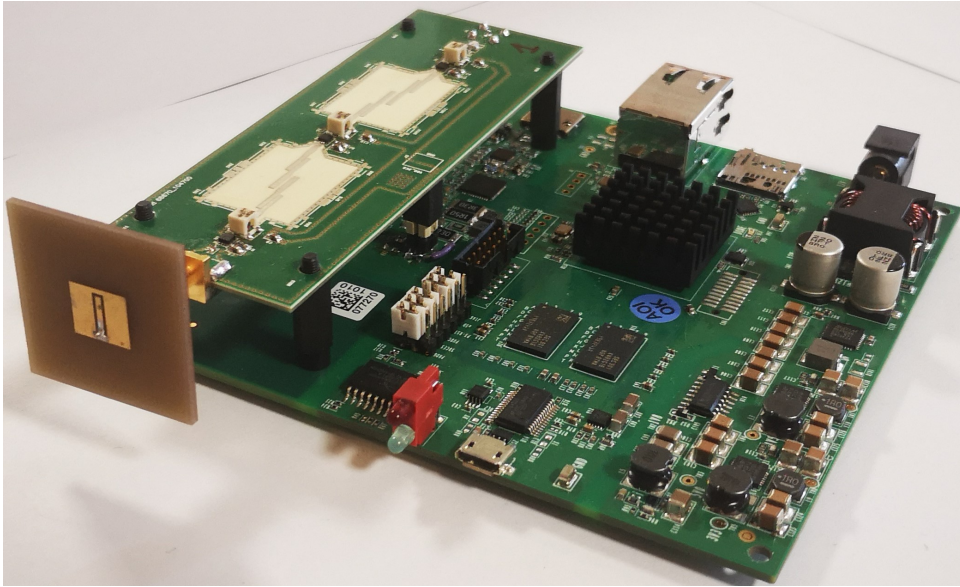


Figure 2.41: New Sensor processing board and UWB receiver assembly



Figure 2.42: Final aspect of the Sensor board enclosed in the box.



# Chapter 3

## Software Design

### 3.1 Introduction

The localization system software is divided into three parts. The first two are the FPGA firmware and the integrated ARM processor program, together they form the Sensor software, the third one is the user interface running on the host PC.

The FPGA firmware synthesizes a custom architecture that receives the continuous data stream coming from the ADC and correlates it with the preamble sequence. This operation allows to recognize the presence of a transmitted pulse sequence in the data stream. The correlation results are compared with a threshold, if the condition imposed by the threshold is satisfied, a Time Of Arrival (TOA) timestamp is associated to the data buffer and is sent to the ARM processor.

The ARM processor further manipulates the TOA and the data received from the FPGA in order to achieve two important goals:

- To recognize the Tag ID associated to the received sequence;
- To improve the TOA accuracy applying a custom super-resolution algorithm.

These informations are then sent through Ethernet to a host PC running the user interface.

The host application receives the TOA from the Sensors and computes the Time Difference Of Arrival (TDOA) between the measurements taken at the reference Sensor and the ones from the other Sensors. It performs the multilateration algorithm providing as a result the cartesian coordinates of the localized tags. The obtained positions and the associated data are plotted on a map for visualisation. The following sections will describe the details, the implementation issues and the adopted solutions for each one of the three developed software parts.



## 3.2 FPGA firmware design

### 3.2.1 Introduction

The Soc receives the data from the ADC using the high speed differential connection on the FPGA part of the chip. The firmware developed to configure the FPGA performs different tasks:

- It receives on the high speed Low Voltage Differential Signaling (LVDS) lanes the ADC data and de-serialize them;
- It correlates the continuous data stream with the pulse sequence Preamble and it thresholds the correlation results to decide if a pulse sequence is present in the data;
- If a pulse sequence is successfully found, it computes the associated TOA and saves it with the corresponding data into a FIFO;
- It transfer the TOA and the associated data to the ARM processor for successive processing steps.

The high level block diagram of the software architecture will be introduced and the details of the building blocks will be discussed. The discussion will dig more in detail of the custom blocks in the design: The LVDS deserializer and the Correlation block. Finally, the methodology implemented to transfer data from the FPGA to the ARM processor will be presented.

### 3.2.2 Block diagram

The high level block diagram of the architecture synthesized, placed and routed in the FPGA assembly, is shown in Figures 3.1 and 3.2. The left side of the block diagram shown in Figure 3.1 presents the ADC data signals organized in ten differential lanes organized as follows: The vector signals `DATA_IN_P[8:0]` and `DATA_IN_N[8:0]` represents the eight differential data lanes (`DATA_IN_P[7:0]` and `DATA_IN_P[7:0]`) and the low frequency differential clock, commonly called frame clock (FCLK). This clock signal is embedded with the data as the MSB of the vector signal (`DATA_IN_P[8]` and `DATA_IN_P[8]`) and is used to synchronize the data stream at the LVDS receiver end. The other differential signal is the high frequency differential bit clock signal called `CLKIN_P` and `CLKIN_N`. In the standard LVDS interface it is commonly referred as local clock (LCLK). These signals are provided at the input port of the LVDS Interface block. This block manages the de-serialization of the ADC data at the input by providing them as output on a 64-bit bus timed with a Single Data Rate (SDR), 125 MHz clock called



data\_CLK.

The LVDS interface requires also a reference 200 MHz clock ( $f_{Ref}$ ) to de-serialize the data. This signal is provided by the ARM processor through a clock wizard block shown in Figure 3.1 inside the LVDS interface block. This block can perform different operations on clock signals such as frequency multiplication, fractional synthesis, jitter reduction or clock domain crossing [71]. In this specific case, the frequency multiplication functionality is used.

The output clock of the LVDS block is connected to another clock wizard block,

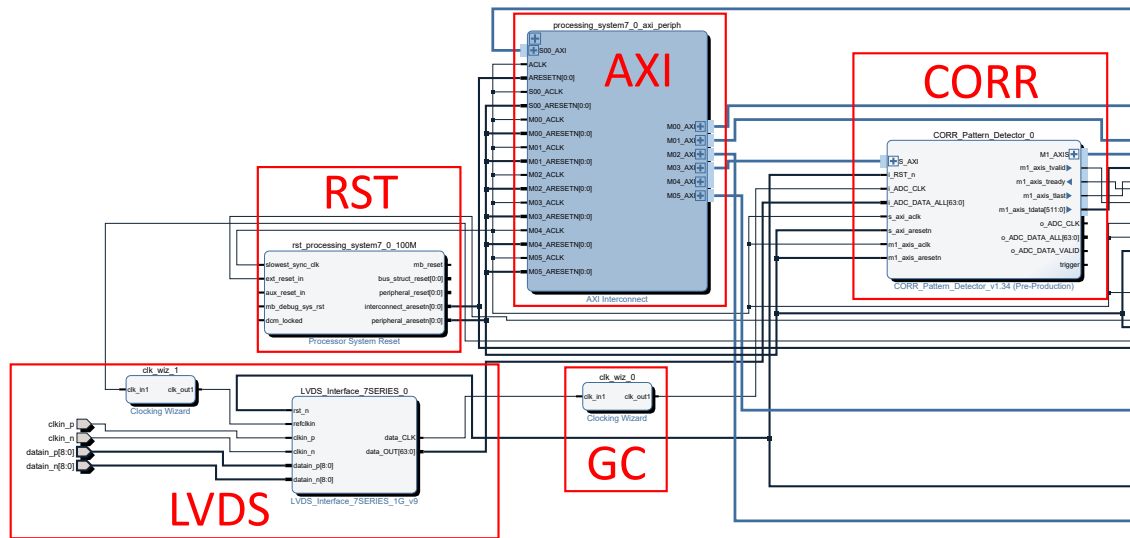


Figure 3.1: High level block diagram of synthesized FPGA architecture. Detail of the LVDS data management interface and correlation block.

referred in the block diagram as Global Clock (GC). To explain the presence of this block, it is necessary to give a little information about the organization and clock routing logic integrated in the FPGA fabric. The detailed explanation of the clocking resources and their organization is presented in [74]. The fabric is divided in regions, each region rely on a local clock distribution net from a main clock source. If after the synthesis we have a certain block in the design that requires a number of logic elements so large that it cannot be placed in a single region, the tools automatically place it over multiple regions. In these situations, the local clock distribution network will not be able to route all the connection required and the time constraints won't be satisfied. To solve this issue, it is common to use an Higher Clock hierarchy level that uses clock signals able to cross the different FPGA regions satisfying the timing constraints called GC. The GC clock wizard is used to route the local data\_CLK signal exiting the LVDS block to a global clock buffer able to distribute it to the entire FPGA.

The LVDS block outputs are provided to the next custom block, the Correlator

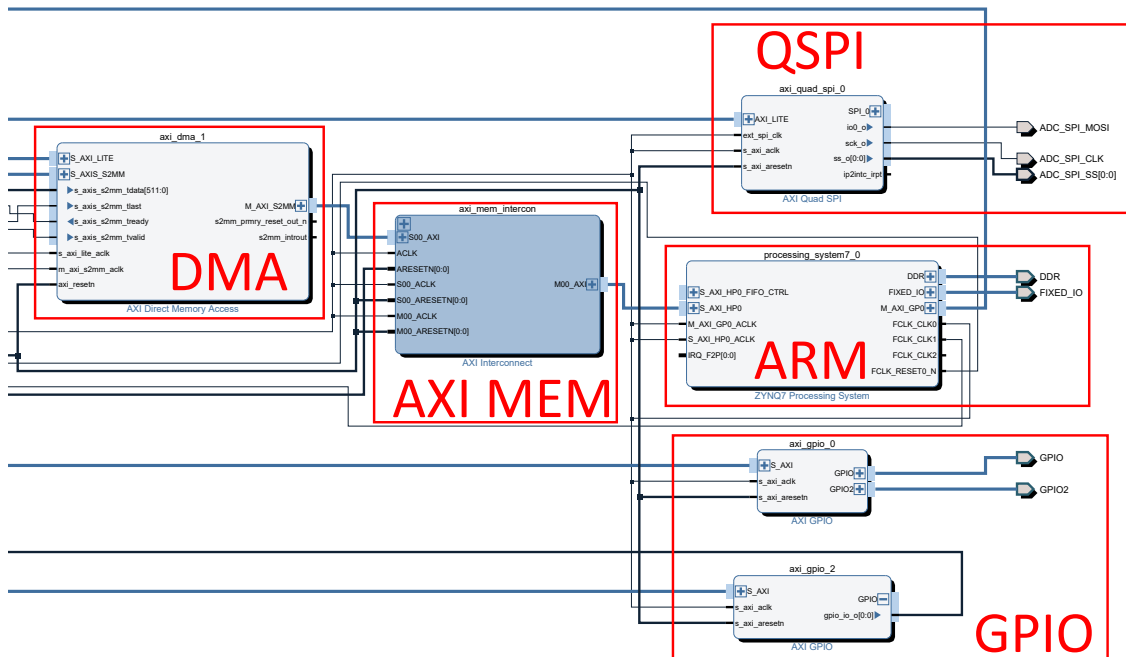


Figure 3.2: High level block diagram of synthesized FPGA architecture. Detail of the ARM processor block and its peripherals.

and Pattern detector subsystem. The Correlator block is the processing core of the whole FPGA design: It performs the correlation between the data stream received from the LVDS block and the pulse sequence preamble, and provide the results to the threshold block.

The threshold block compares a vector of 512 correlation results with a threshold. If a set of requirements is satisfied and the threshold is overcome by one of the correlation results, the circuit concludes that a pulse sequence in the incoming data is present and data are saved and organized for Direct Memory Access (DMA) transfer. The product guide of the DMA IP is available at [68].

The DMA block manages the data transfer from the FPGA to the ARM processor using the High Performance (HP) port of the ARM processor.

The other blocks are FPGA logic resources to implements the standard peripherals drivers for the AXI Quad SPI, GPIOs and reset of the LVDS interface.

The ARM processor is the last block in the chain; it is connected to the DDR memory driver and provides the clock and reset signals for the AXI interface crossbar that connects all the FPGA blocks.

The two darker blocks are the AXI communication backbone, they are automatically placed by the design tool whenever the FPGA project involves AXI Peripherals and AXI Memory devices. The details of these components are described in [69][72].

### 3.2.3 LVDS data management

The LVDS signals are transmitted by the on board ADC chip connected to the FPGA and are received by the FPGA LVDS interface block. They are organized in ten differential couples, two of them are dedicated to clock signals while the remaining eight are used for data. The signal's timing diagram is shown in Figure 3.3.

The timing diagram references all signal's timing to the input clock that switches at the sampling frequency rate, in our case  $f_S = 1$  GHz. The first LVDS clock signal is a differential Double Data Rate (DDR) bit clock called LCLK. The switching frequency is equal to one half of the sampling frequency so  $f_{LCLK} = 500$  MHz.

The second clock signal is a differential single data rate (SDR) frame clock called FCLK. The switching frequency is equal to one eighth of the sampling frequency so that  $f_{FCLK} = 125$  MHz. In a single period of FCLK we receive the 8 bit of a single sample serialized starting from the least significant bit (LSB) on each differential data lanes for a total of eight 8-bit samples.

The ten differential lanes provide the data to the LVDS data management block. The block's design is based on the work shown in the application note presented in [67]. The architecture relies mainly on a 7 Series Xilinx FPGA basic block: the input serial-to-parallel element called SerDes (Serializer Deserializer), the element primitive is called ISERDESE2. The details of all basic input-output resources

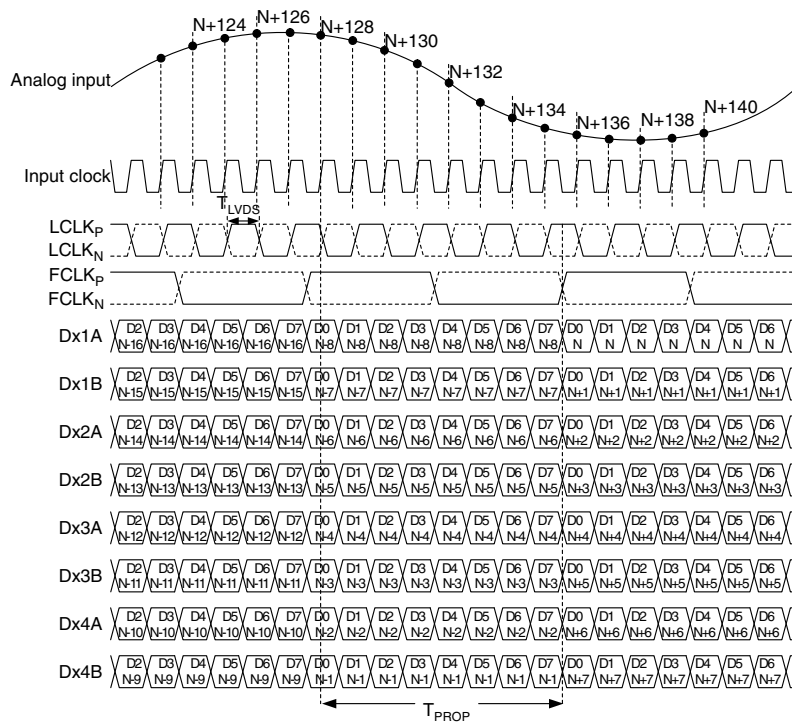


Figure 3.3: Timing diagram of the LVDS data received by the FPGA. The image, taken from the HMCAD1511 ADC datasheet, shows the relation between the LCLK, FCLK and sampling clock signals and the ADC data.

available in the FPGA used for this block are reported in the SelectIO resources user guide [73] from Xilinx. The implemented architecture to receive the LVDS data is shown in Figure 3.4. The LCLK differential DDR clock signal is received at

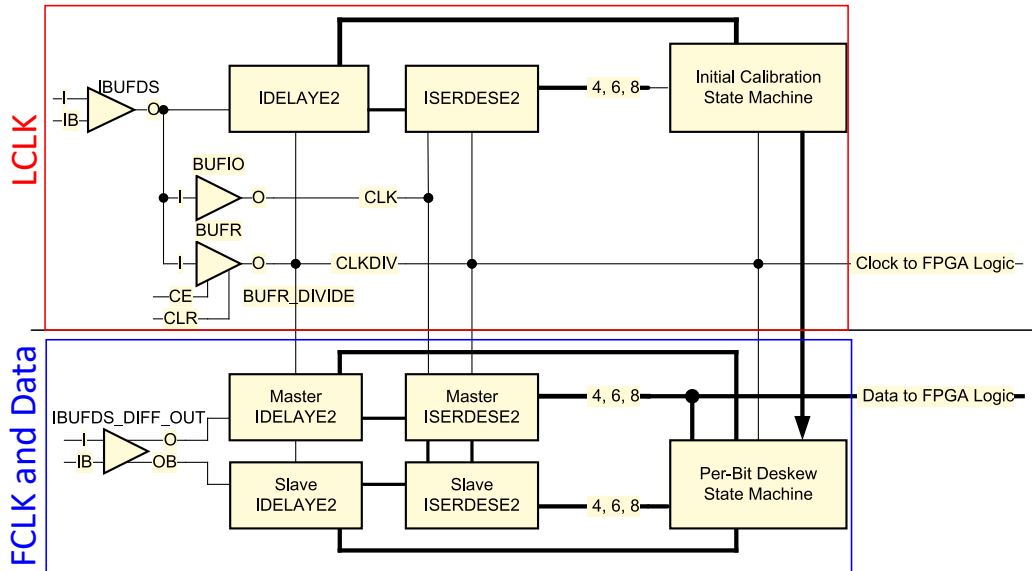


Figure 3.4: Implemented LVDS Deserialization architecture showing the LCLK receiving chain in the upper part of the figure, and the FCLK and data receiving chain in the bottom side.

a Differential to Single-ended Input Buffer (IBUFDS), shown in the top left corner of Figure 3.4, and provided to an Input/Output Buffer (BUFIO) and to a Regional clock Buffer (BUFR) used to implement single region clock connections.

The BUFR is configured in "Divide by N" mode where N need to be set equal to half the serial to parallel conversion rate (SPCR). We receive serialized 8-bit data so the SPCR is 1:8 fixing the value of  $N = 4$ . The BUFR output is provided to the input delay element (IDELAYE2), to the serial-to-parallel element (ISERDESE2), to the initial calibration state machine and to the block's output.

The undivided clock from the BUFIO is provided to the SerDes for calibration after power on. The results of this operation are provided by the Initial Calibration State Machine to the Per-Bit Deskew State Machine. The reference clock  $f_{Ref}$  connected to the LVDS block seen in 3.1 is used internally by the input delay element to tune the delay according to the desired data bit-rate.

The eight differential data and the differential FCLK signal are received by the lower part of the circuit shown in Figure 3.4 replicated for each signal. The FCLK and data bit signals are provided to a differential-input differential-output buffers, to a Master and a Slave input delay and ISERDESE2 elements. The Master ISERDESE2 is used to de-serialize the incoming bit into a parallel eight bit wide data

bus.

The parallel data exiting the Master ISERDESE2 are forwarded to the output and to the Deskew state machine while parallel data from the slave ISERDESE2 are sent only to the state machine. The Per-bit Deskew state machine is used to determine the correct sampling delay given the desired bit-rate. The algorithm implemented by the deskew state machine tries to compute the optimum delay necessary to maximize the received data eye diagram aperture. The delay implemented by the IDELAYE2 block is configured to obtain the required bit-rate by changing the values of internal taps.

The operation performed to dynamically adjust the delay requires to analyze the incoming data stream looking for a pre-configured reference pattern. The shape of the FCLK signal is used as reference pattern. The state machine circuit locks on the pattern corresponding to the rising edge of the frame clock and corrects the delay of the data. Without this operation we wouldn't be able to identify the starting bit of any data making impossible the de-serialization.

The presented architecture is iterated for each differential data lane for a total of eight instantiations. The LVDS data management block provides at the output a clock signal with frequency  $f_{data\_CLK} = 125$  MHz and a 64-bit parallel bus that allows to transfer eight 8-bit data per clock cycle.

### 3.2.4 Correlation

The FPGA receives the data from the ADC and correlates them with the preamble sequence to verify the presence of a Tag sequence and to perform a coarse measurement of the TOA, as shown in Figure 3.5 whose details are given in the following. The incoming data are provided to a custom block, the correlator, which is the processing core of the whole FPGA design. The received data are correlated with the pulse sequence preamble common to every Tag transmission. The FPGA correlation is performed by multiplying the data stream coming from the ADC with a vector of data called symbol mask shaped as a 2 ns rectangular pulse over a 50 ns symbol period. The symbol shape is shown in Figure 3.6.

The correlation results are computed as:

$$Del[i] = \sum_{j=0}^{49} ADCdata[j] \cdot SymbolMask[j] \quad (3.1)$$

where  $Del[i]$  is the  $i$ -th correlation result and  $ADCdata[j]$  and  $SymbolMask[j]$  are, respectively, the sampled data in position  $k$  of the received data buffer and the corresponding symbol mask bit, as shown in Figure 3.7.

The pulses in the sequence are separated by an interval of 50 ns; since the sampling frequency is 1 GHz, each pulse lasts for 50 samples, forcing the symbol mask to

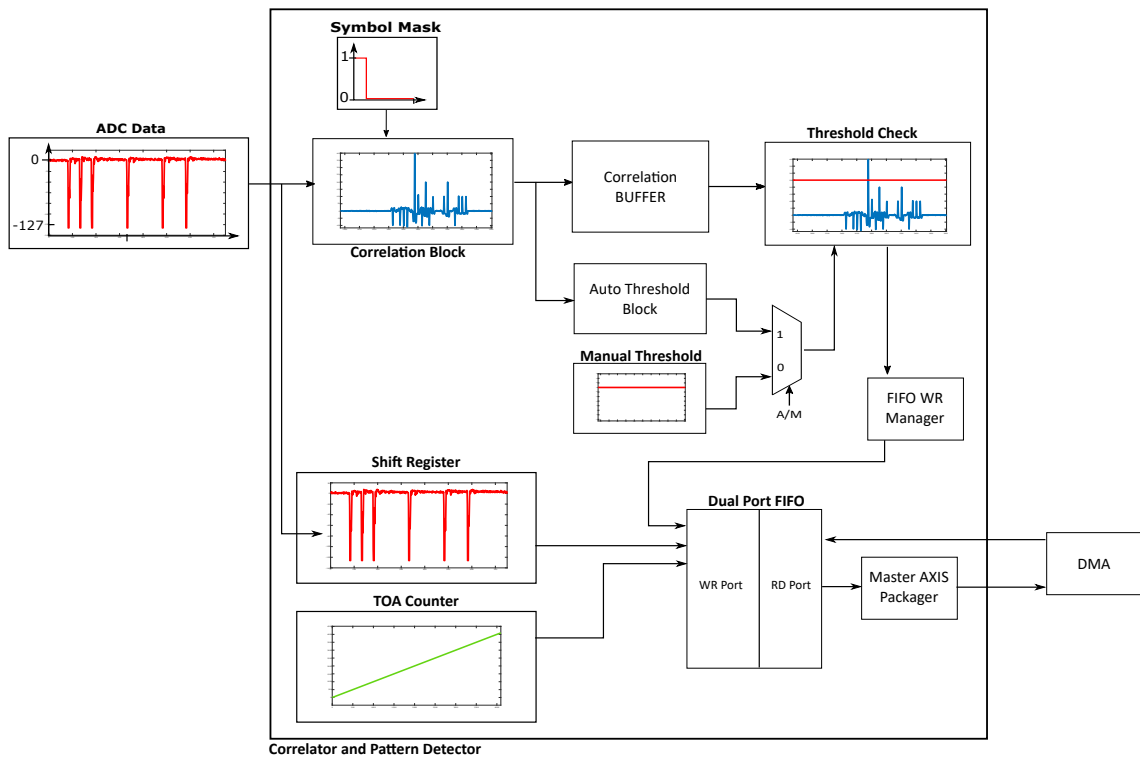


Figure 3.5: High-level block diagram of the processing steps performed in FPGA starting from the correlation of the incoming data with the symbol mask, to the thresholding operation to the final storage in the data buffer.

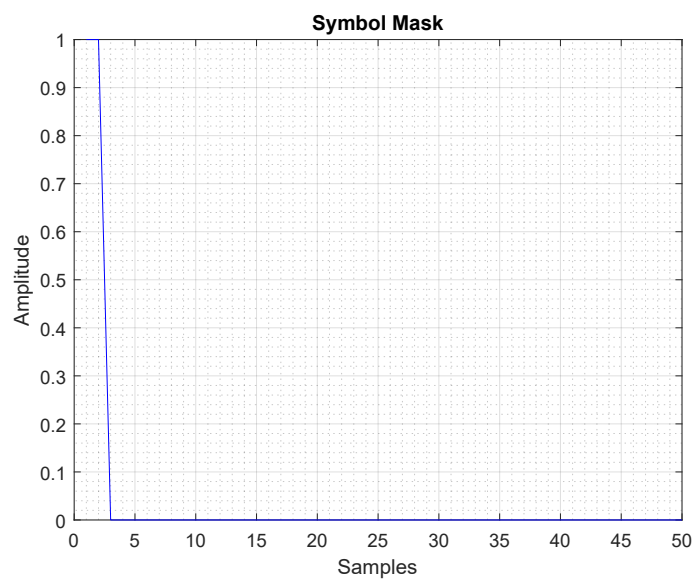


Figure 3.6: The symbol mask used for the correlation operations. The mask has is shaped as a 2 ns rectangular pulse.



be that long as well. At each 125-MHz clock cycle, the FPGA receives eight new data from the ADC to be correlated. The FPGA correlation block saves the new data in a buffer and elaborates them in parallel providing eight correlation values. The multiplication between the data buffer and the symbol mask is instantiated eight times: one for each new received sample and, for each instance, the data buffer is shifted by one position. For each instance, the result of the multiplications are added together to obtain the correlation values that are saved in a buffer (see Figure 3.7). The sum of the fifty multiplications results is performed using a tree of two input adders with increasing parallelism, the depth of the tree is equal to:

$$N = \lceil \log_2 50 \rceil = 6 \quad (3.2)$$

Where 50 is the number of multiplication results to be added.

We take advantage of the fact that each symbol in the sequence has a fixed length

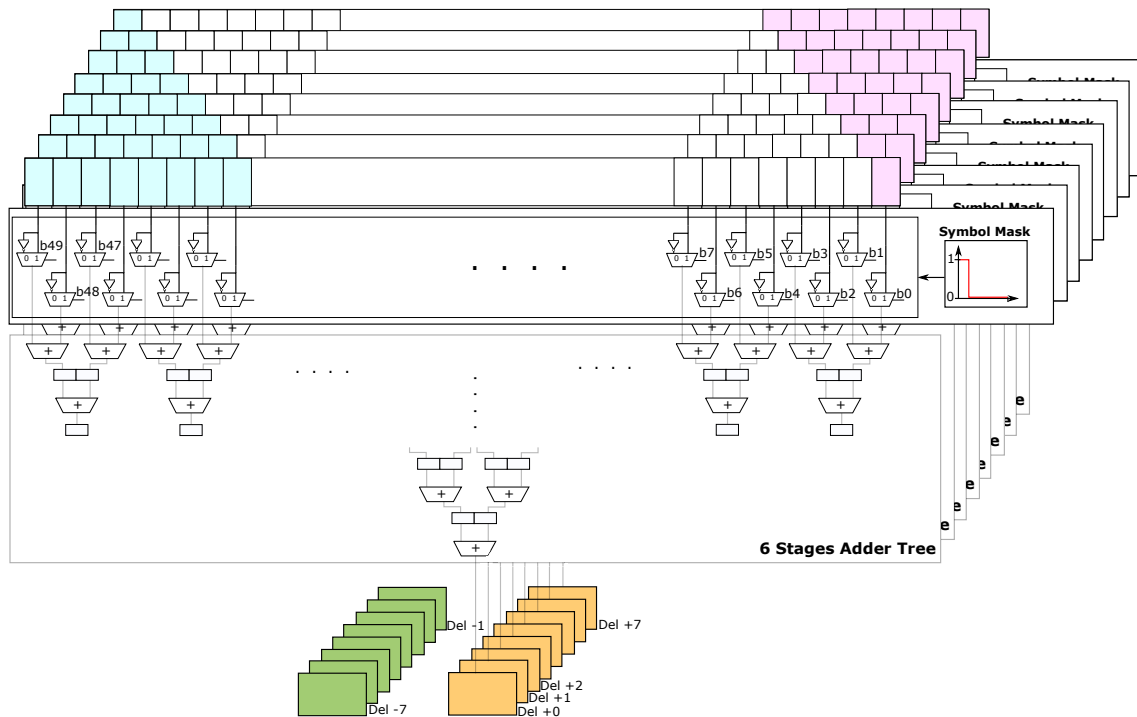


Figure 3.7: Block diagram of the correlator block. It shows the eight instantiations of the elaboration blocks that compute the correlation delays and the final buffer where all the correlation results are stored. Each clock cycle the data buffer receives eight new data (in pink) and discards the eight oldest data (in cyan). The newest correlation result (Del + 7) is computed using all the eight newest data.

of 50 samples. By taking the correlation values at the correct delay and adding (for the bit equal to one) or subtracting (for the bits equal to zero) the correlation

results for each symbol according to the preamble sequence, we can reconstruct the full correlation of the received data with the preamble sequence in a very efficient way. Moreover, these operations are performed in parallel to produce eight correlation values per 125-MHz clock cycle.

### 3.2.5 Manual and Automatic Threshold implementation

The correlation results are compared with a threshold value that can be set manually or automatically. The manual threshold can be set by the user using the host PC application. The high level block diagram of the block performing the threshold operations is shown in Figure 3.8. The block analyzes a moving window of 512 correlation results stored in the output buffer of the correlation block. Each element in the buffer is compared with its neighbour in order to evaluate the maximum of the entire window.

By means of a binary tree of comparisons, starting from 512 data (256 comparison) we obtain a new vector of 256 data (128 comparison) and so on for a total of  $N = \lceil \log_2 512 \rceil = 9$  stages of comparisons. At the bottom of the decision tree, we have the maximum value of the correlation.

The condition that we want to satisfy requires the maximum to be over the threshold and to be in the center of the window. If both conditions are satisfied, we set the Thresh signal to 1 asserting that we found a pulse sequence in the stream of data. When the threshold is set automatically, the FPGA continuously analyzes the correlation results to evaluate their standard deviation and average. To simplify the implementation and minimize the required logic resources, we implemented a moving average [62]. The behavior of the moving average is described by:

$$y(k) = \frac{1}{N} \sum_{i=k}^{N+k-1} x_i \quad (3.3)$$

where  $N$  is the number of element on which we apply the average. The architecture of the moving average filter is reduced to an accumulator where the incoming data is added and the oldest data is subtracted. Moreover, if the dimension of the averaging window is a power of two, the scaling factor is implemented as a  $n$ -bit shift.

The standard deviation is computed using the same moving average principle. The value of the automatic threshold is  $k$  times the value of the standard deviation on top of the average (e.g.,  $k = 8$ ). When the threshold is overcome, we save the received data into a FIFO together with the value of the TOA counter. The saved data and TOA are sent to the ARM processor to refine the TOA computation and recognize the Tag ID associated with the transmitted sequence.

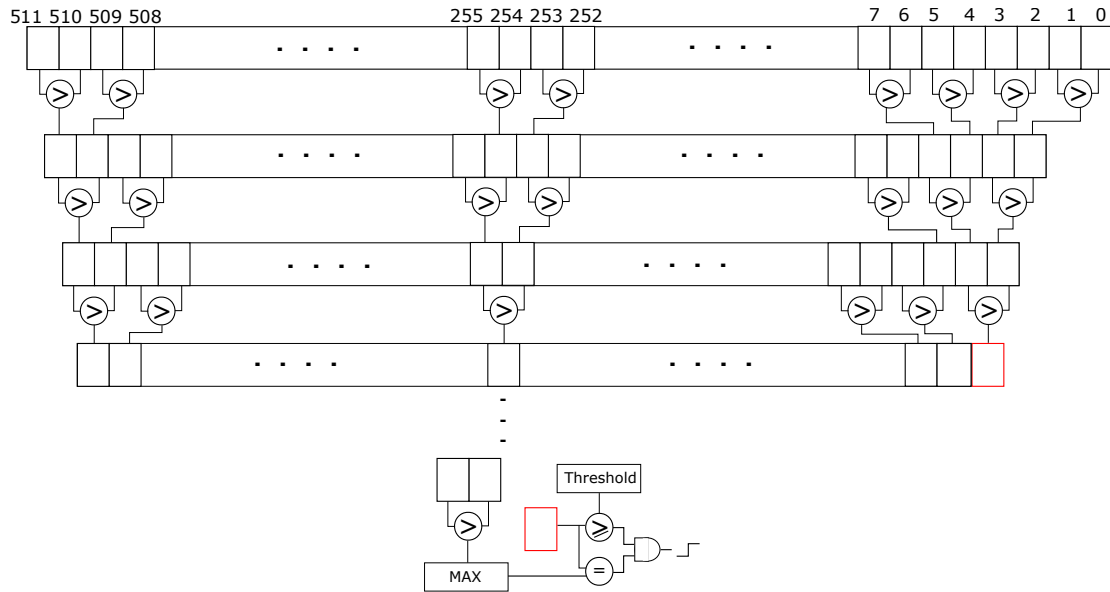


Figure 3.8: FPGA threshold mechanism. We search for the maximum among the 512 correlation results and we require it to be in the middle of the comparisons window. If this condition is satisfied and the maximum value is larger than the threshold, we set the Thresh flag to 1.

### 3.2.6 Data management

The ADC data goes through two different paths as shown in Figure 3.5: One to the correlation block and another one into a Shift Register. The shift register is used to maintain the alignment between the incoming data and those being processed inside the correlation block. The latency of the correlation operations is known so it is sufficient to have a shift register sufficiently long to contain enough sample to match it.

A flag is set to one when the thresholding condition is satisfied and it triggers the write operation of the shift register content into the data FIFO. The implemented FIFO architecture is described in detail in [70] and is shown in Figure 3.9.

The FIFO has two different clock domains, one for write operations and another for read operations. The two clocks can be at two different frequencies as well as the same, the choice of the frequency relation depends on the application. In our case, the write ( $wr$ ) and read ( $rd$ ) clock domains differs even though they have the same frequency of  $f_{rd} = f_{wr} = 125$  MHz.

The write clock is provided by Correlator and preamble detector block while the read clock is the same one used for the AXI and AXI-Stream interfaces and is provided by the ARM processor.

The read and write interface data width are different. The read interface is 512-

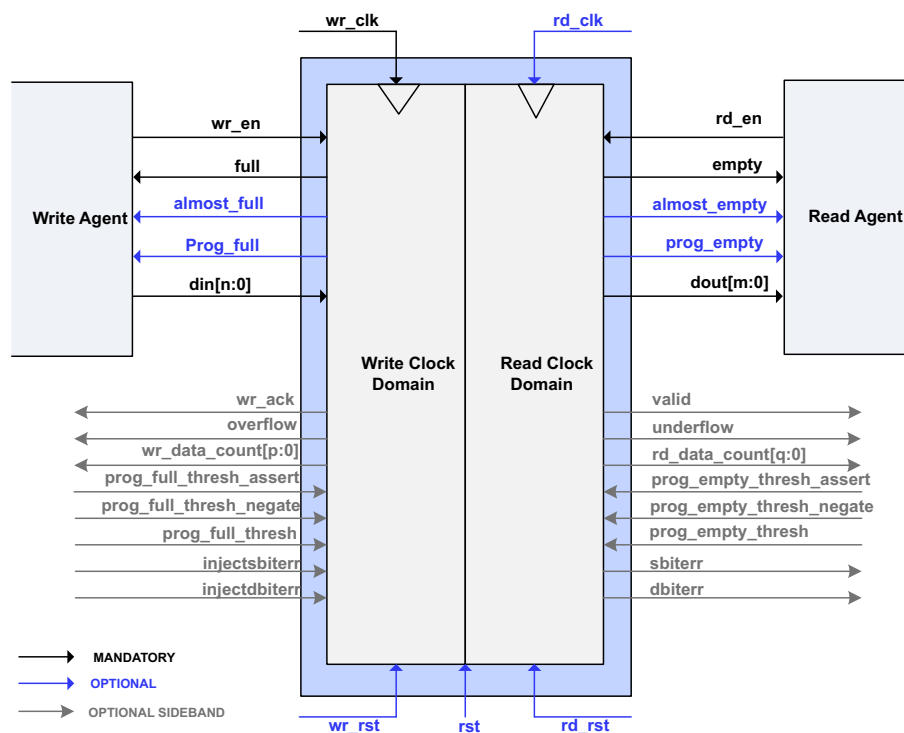


Figure 3.9: Native FIFO IP interface presented in the Xilinx datasheet. On the left there is the write interface while the read interface is on the right.

bit wide while the write interface is 64-bit wide. In such a way we can use the same clock frequency for the two interfaces and change the throughput to avoid overflowing or stalling transfers.

The writing and reading operations control is performed by two different processes. The detailed timing diagram of the fifo write process is shown in Figure 3.10. The writing process starts when the TRIGGER signal is asserted to one. The state machine firstly deassert the WAIT\_TRIGGER signal and assert the START\_Counter signal and the fifo write enable signal (FIFO\_WREN). This signal enables the DATA\_Counter to keep track of the total number of data saved in the fifo. The BIND\_TMSTMP (Bind Timestamp) signal is used to trigger the Time of Arrival counter sampling operation and to save the obtained TOA as first sample in the fifo. The entire process saves a total of 2048 8-bit samples in the fifo before deasserting the FIFO\_WREN signal.

The read process takes the data from the fifo read interface and sends them to the DMA using the AXI-Stream interface. The AXI-Stream interface uses the following signals:

- ACLK, the interface clock signal;

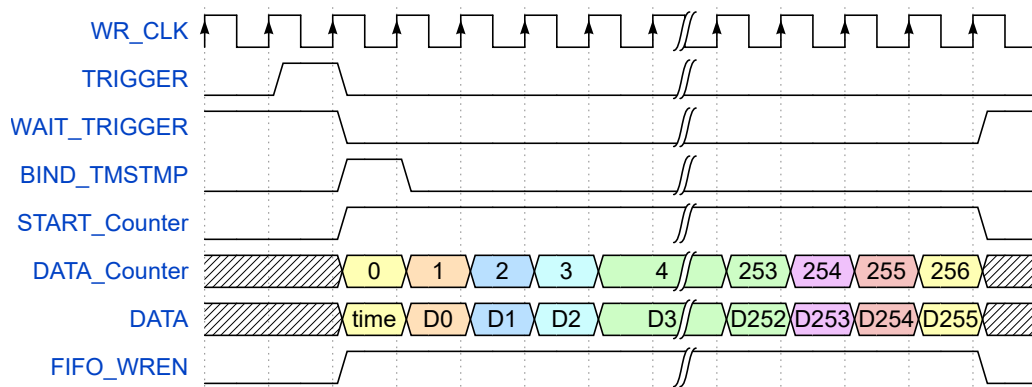


Figure 3.10: FIFO Write process Timing

- ARESETN, the stream interface reset signal;
- TVALID, used to notify that the master, in this case the process that reads the data from the fifo, is driving a valid transfer. A transfer takes place when both TVALID and TREADY are asserted.
- TDATA, the N-bit data signal;
- TSTRB, the byte qualifier that indicates whether the content of the associated byte of TDATA is processed as a data byte or a position byte. In our case the content will always be data so its value is fixed to one;
- TLAST, used to set the boundary of a data packet.
- TREADY, to indicate that the slave (in our case the DMA) can accept a transfer in the current cycle.

The detailed timing diagram of the data read process is shown in Figure 3.11. Whenever the signal TREADY is asserted to one, if data are available in the fifo, the AXI-Stream interface asserts the read enable (RD\_EN) signal to start the reading operation. The process starts a counter called DATA\_Counter that maintain the TLAST signal de-asserted until the last read value is reached. Every data read from the fifo is flagged as valid until the fifo is empty.

When the counter reach the end, the process asserts the TLAST signal for one clock cycle and wait for the TREADY to be asserted again. The process reads a total of 32 words of 512 bits each for a total of 2048 samples.

The data organized in packets of 2048 samples and delimited by the TLAST signal, are sent to the DMA that route them to the DDR3 memory through the High Performance port connected to the ARM processor. The data transferred into the ARM processor memory by the DMA will be further processed by the ARM

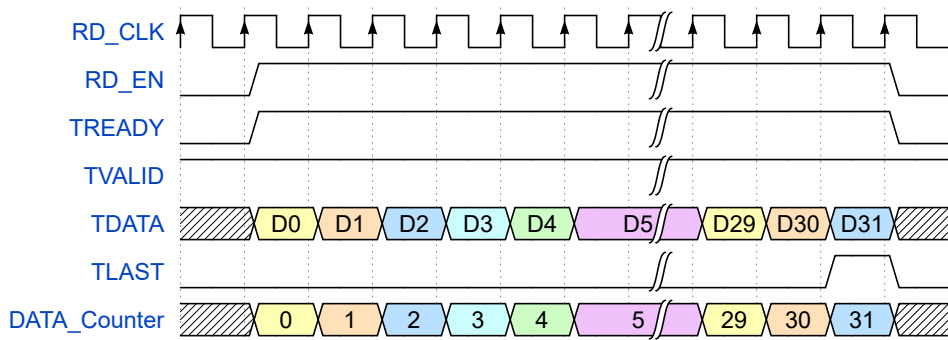


Figure 3.11: FIFO Read process Timing

processor software.

## 3.3 ARM firmware design

### 3.3.1 Introduction

The goal of the FPGA processing is to analyze the continuous stream of data searching for a received pulse sequence transmissions and, when a sequence is found, estimate its coarse TOA and save the corresponding data.

The goal of the ARM processor's firmware is to analyze the data and TOA computed by the FPGA and further manipulate them.

The ARM firmware performs two tasks:

- It takes the data from the FPGA and recognize the Tag ID associated to the sequence by analyzing the sequence payload;
- It refines the TOA estimation using a super-resolution algorithm.

The Tag ID and refined TOA obtained are transmitted to the host application using Ethernet UDP protocol.

The details of the ARM software implementation are presented in the following sections.

### 3.3.2 System configuration

The ARM processor software is divided in two main sections, Configuration and Infinite Loop. The processor software high level block diagram is shown in Figure 3.12. The Configuration starts by setting up the network interface for the Ethernet transmissions and the serial communication interface.

The next operation configures the DMA to receive data from the FPGA and sets the default configuration of the Correlator and Pattern detector block. During the Init Sequence operations, the program initialize the matrix where all possible recognizable Tag sequences are saved. The matrix will contain a whole Tag sequence for each row. An example of a possible Tag sequence is shown in Figure 3.13, the sequence 111001001010101 represents one of the possible pulse sequences generated by the Tag, the first seven bits are the Barker 7 common preamble while the latter 8 bits are the unique Tag ID. The binary sequence is converted into a decimal value so  $01010101_{bin} = 85_{dec}$ . Each pulse lasts 50 samples so the matrix rows will be  $15 * 50 = 750$  samples long.

The correlation is performed between the raw data received from the FPGA and the a sequence mask equal to:

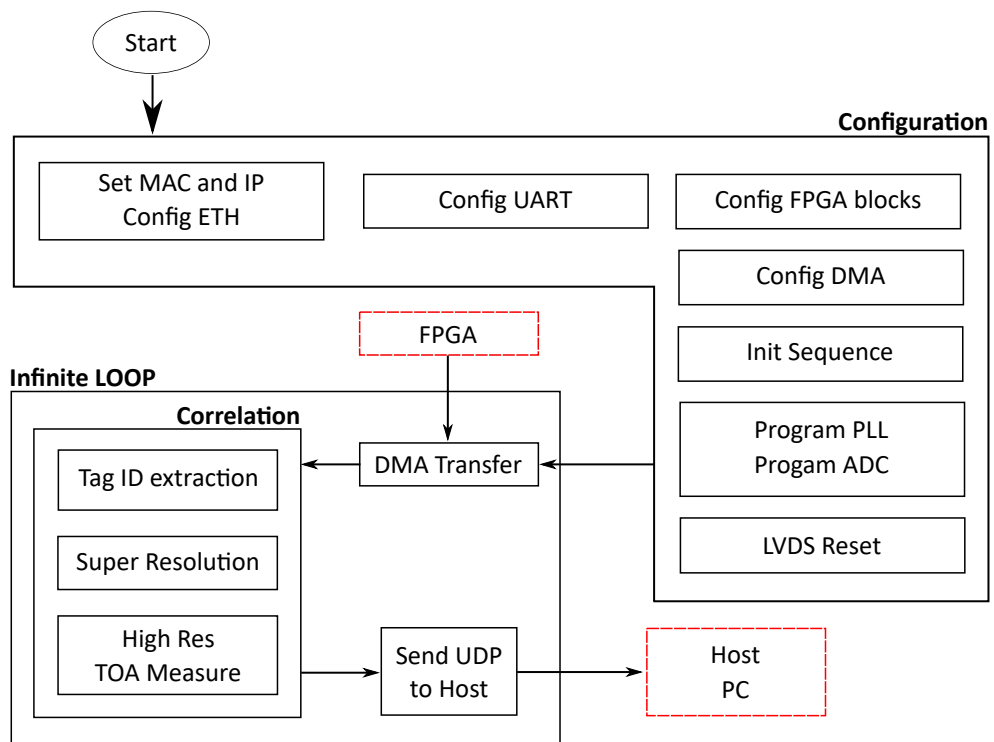


Figure 3.12: High level block diagram of the ARM software program.



- Case Symbol is 1 : The mask will be equal to  $\frac{-1}{PW}$ , where  $PW = 2$  is the duration of the pulse in samples and " $\frac{1}{SW-PW}$ " where  $SW = 50$  is the duration of the single pulse symbol in samples;
- Case Symbol is 0 : Is exactly the same behavior of the previous case but with opposite sign.

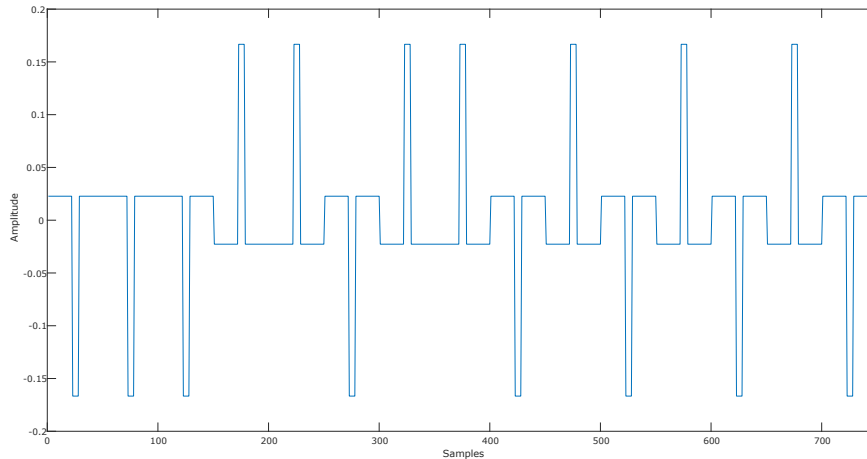


Figure 3.13: Behaviour of complete Tag sequence 111001001010101: 1110010 represents the common preamble while 01010101 is the unique binary Tag ID sequence corresponding to Tag 85

The sequence has zero bias to avoid introducing an offset in the correlation.

When all sequences are configured, the program initialize the Ramp Mask that is used to refine the TOA estimation in the last part of the ARM processor processing. The last operation is the configuration of the PLL and ADC chips using the SPI interface. The PLL is configured to generate a 1 GHz differential clock used by the ADC as sampling clock.

The software initializes the ADC only after the sampling clock has stabilized. The configuration performs the following operations:

1. Waits for the Start-Up initialization after Power On to be completed;
2. Performs a software reset followed by a power down command;
3. Set the LVDS bit clock phase shift relative to data to  $90^\circ$ ;
4. Select the single channel mode of operation option;
5. Choose the output data format to be Signed 8-bit values;

6. Set the digital gain of each ADC channel to its maximum value;
7. Select the channel to be used in single channel mode;
8. Set the chip to active mode.

Once in active mode, the ADC starts sampling the signal coming from the UWB RF receiver module.

During the LVDS sync operation, the received data are not valid so, after the ADC configuration, a software reset is applied to the LVDS block to restart it with valid, aligned and synced data.

The code locks to the infinite loop where it polls the DMA for new data transfers and process the received data.

### 3.3.3 Sequence correlation

The infinite loop performs continuously the core operations of the ARM code. When the FPGA recognize a pulse sequence transmission in the continuous stream of data, it saves the block of data into a buffer and sends it to the DMA.

During the infinite loop, the ARM processor continuously polls the DMA to check if the buffer containing a new sequence is ready to be processed. If this condition is satisfied, the ARM processor copies the data into another buffer and performs two main tasks:

- It recognizes the Tag ID associated to the received pulse sequence;
- It estimates its TOA with nanosecond accuracy.

The accuracy of the TOA estimation can be further improved by enabling the super-resolution algorithm.

The high level block diagram of the operations performed inside the infinite loop are shown in Figure 3.15. The methods and implementation details of these operations are described in the following sections.

#### Tag ID recognition

The Tag ID associated to the received pulse sequence is the first information computed by the ARM processing software. Two different solutions have been implemented, one more robust in the case of very low Signal to Noise Ratios (SNR) but less efficient and a second one that is more sensible to noise but is significantly faster than the other. The user can choose which method to apply depending on the final application.

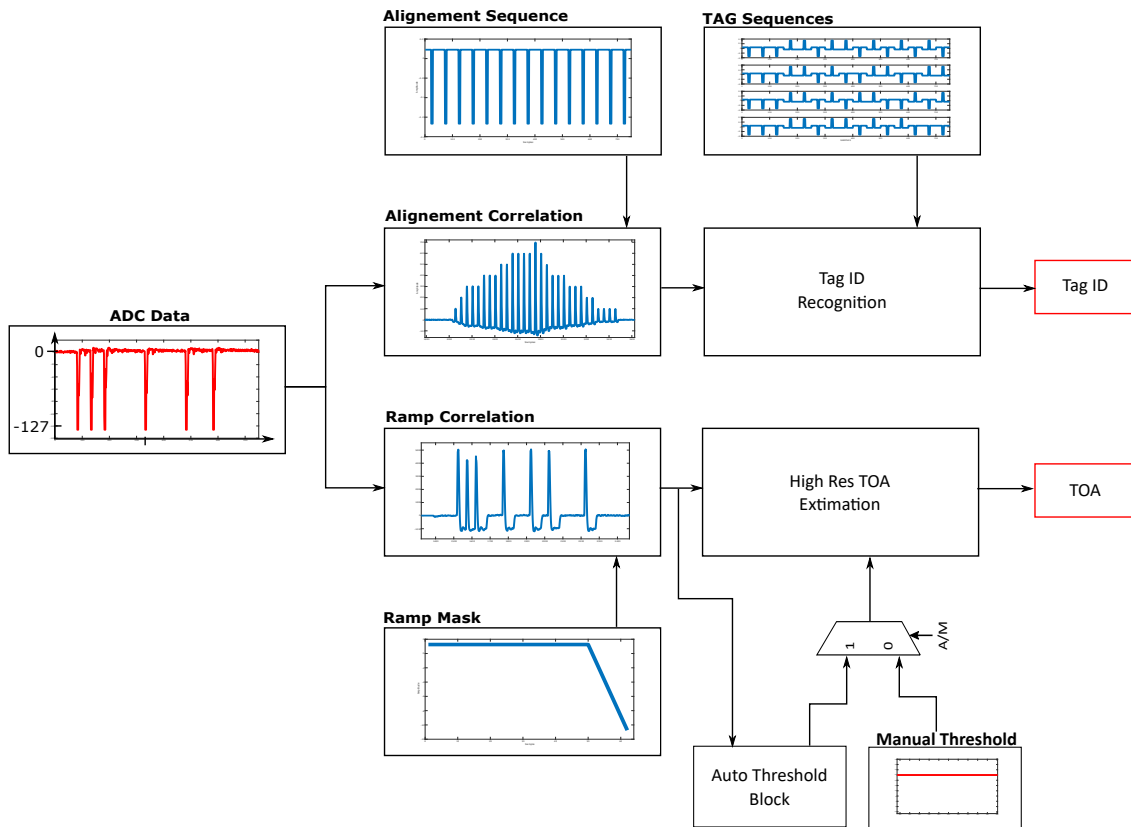


Figure 3.14: High level block diagram of the operations performed in the ARM software to recognize the Tag ID associated to the received pulse sequence and its TOA.

**First implementation** The data buffer copied from the DMA is dimensioned to contain one pulse sequence only. The FPGA correlation and thresholding operations guarantee to recognize a pulse sequence in the incoming data stream but do not give any information about the position of the sequence inside the buffer. This information allows to reduce the number of iterations required to compute the Tag ID. Instead of scanning the entire buffer with the complete Tag sequence (common preamble and unique ID), knowing the position of the first sample of the sequence allows to correlate with the ID sequences only.

The operation performed to find the first sample of the pulse sequence is a correlation with an aligning sequence. The aligning sequence is defined as:

$$Y[k] = \sum_{k=1}^{15} \sum_{i=0}^{50} y[i] \quad (3.4)$$

$$y[i] = \begin{cases} -\frac{1}{PW} & \text{if } (\frac{SW}{2} - \frac{PW}{2}) \leq i \leq (\frac{SW}{2} + \frac{PW}{2}) \\ \frac{1}{SW-PW} & \text{if } i < (\frac{SW}{2} - \frac{PW}{2}) \parallel i > (\frac{SW}{2} + \frac{PW}{2}) \end{cases} \quad (3.5)$$

Where PW is the duration of the pulse in samples and SW is the duration of the single pulse symbol in samples. If we transmit 2 ns pulses and samples them using a sampling frequency of 1 GHz we have  $PW = 2$  and  $SW = 50$ . The alignment sequence has the same shape of a transmission of 15 consecutive ones. The correlation with the alignment sequence allows to evaluate the relative delay between the two signals.

An example of the correlation with the alignment sequence is shown in Figure 3.15. The first plot shows the alignment sequence, the second plot shows the raw data buffer copied from the DMA and the third plot shows the correlation between the two signals. The position of the correlation maximum corresponds to the index of perfect alignment between the two signals. The position of the maximum is saved and used as offset for the next operation.

The correlation's maximum index is used to start the recognition operation exactly on the payload part of the pulse sequence minimizing the number of correlation results we need to compute. Using the index of the maximum as starting point and knowing the exact duration of the sequence preamble, we can compute the correlation with all the possible tags exactly aligned with the first sample of the payload.

These correlations are saved into a vector and are computed as:

$$ID[j] = \sum_{i=0}^{PaL} x[PrL + maxIndex + i]T[j][i] \quad (3.6)$$

Where  $ID[j]$  corresponds to the  $j$ -th possible Tag correlation result, PaL is the payload length expressed in samples, PrL is the preamble length expressed in samples, maxIndex is the index of the previous correlation's maximum position and

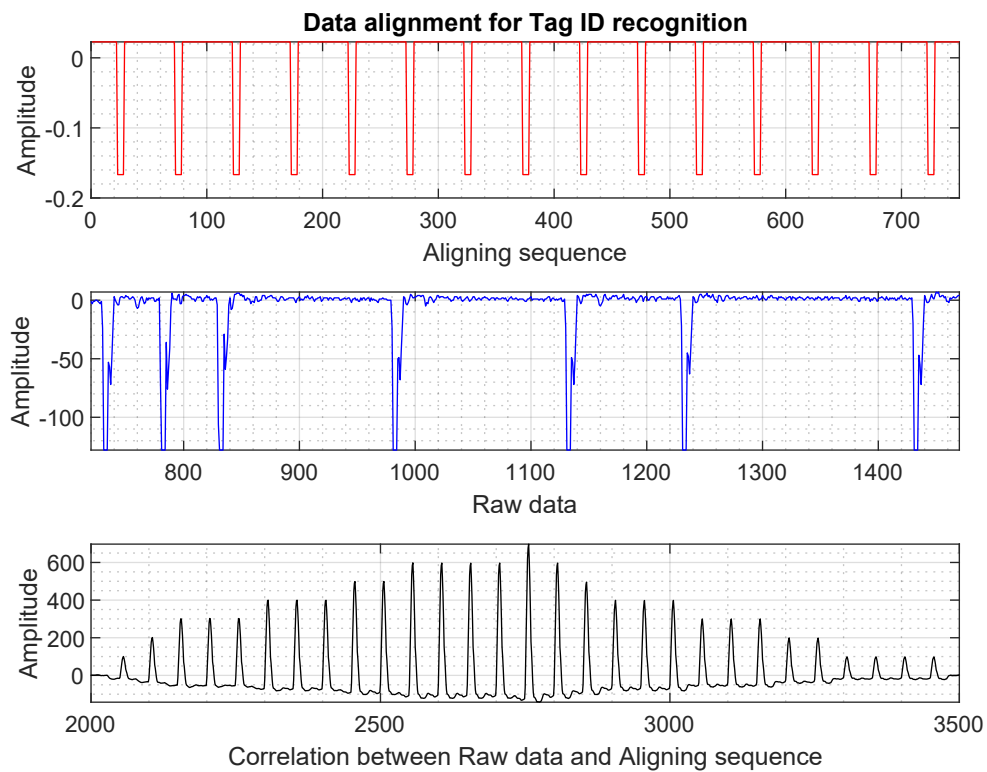


Figure 3.15: Correlation between the raw data and the alignment sequence, the correlation has an absolute maximum when the two sequence are aligned. The x axis of the first two plots correspond to the sample positions.

$T[j][i]$  is the Tag matrix. Each row of the matrix correspond to a possible Tag ID sequence.

The length of the ID vector is equal to the number of Tags we have to recognize. The maximum of the ID vector corresponds to the Tag sequence that gives the highest value of correlation. The index of the maximum correspond to the Tag ID associated to the received sequence.

The main advantage of this solution is that, by computing the correlation between the aligned signal and each possible Tag, we are able to recognize the correct ID even with very low SNR. The drawback is that the number of operations required to perform this task depends linearly with the number of Tags that to recognize. It is then necessary to implement a different solution that does not depend on the number of Tags to reduce the computational cost of the recognition operation.

**Optimized implementation** The new implementation resolves the processing dependency from the number of Tags that limits the usage of the previous solution. The alignment operation is performed using a zero mean version of the Barker 7 sequence. The definition of the symbols is the same used for the alignment sequence in the previous solution with the only difference that a zero bit is represented as a one bit with changed sign. An example of the new alignment operation is shown in Figure 3.16. The first plot represents the Barker 7 sequence used to align with the pulse sequence in the received buffer, the second plot represent the same pulse sequence used in the previous solution and the third plot shown the correlation between the two. The correlation is implemented exactly as we did in the previous case.

The first benefit of this operation is that the number of correlation results that we have to compute is significantly reduced since the Barker sequence is shorter with respect to the alignment sequence used before. Moreover, the maximum of the correlation is significantly higher than the sidelobes. Also in this case, the index of the correlation maximum is saved.

The new ID recognition procedure performs the following operations:

- Filter the raw data using a Finite Impulse Response (FIR) filter;
- Evaluate the minimum of the entire buffer and automatically set a threshold to a fraction of that value;
- Compare blocks of the filtered data with the threshold. The dimension of the block of data is equal to the duration of a single symbol (SW)
- If the local minimum of the block of data is smaller than the threshold we recognize that symbol as a one, if the local minimum does not reach the threshold it is coded as a zero.

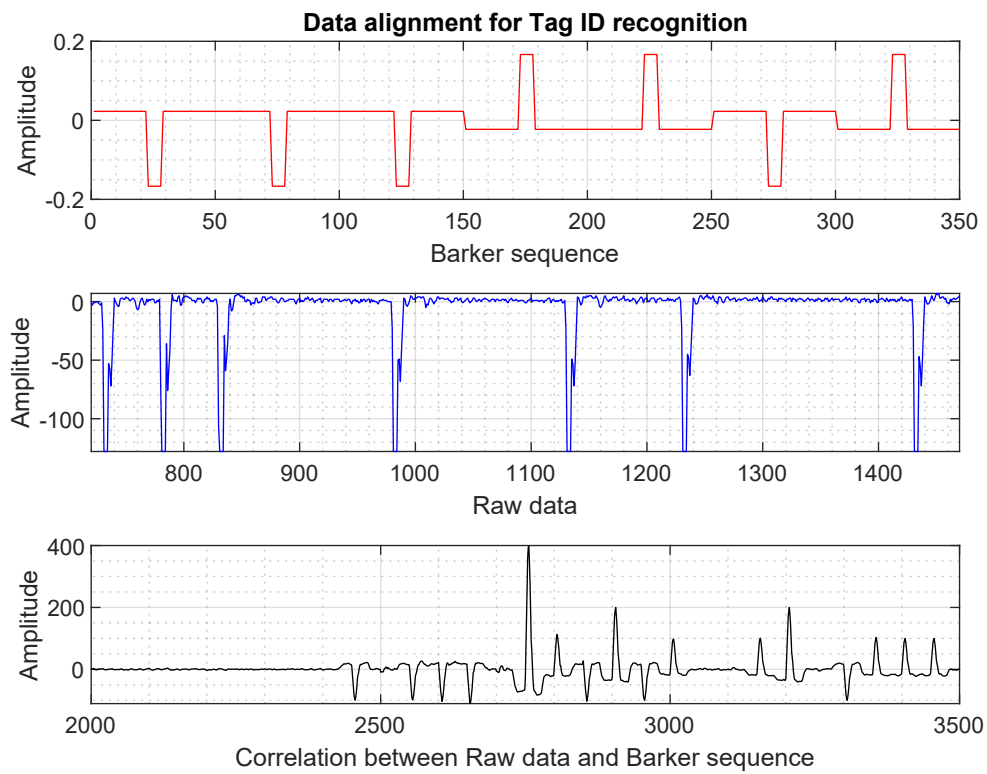


Figure 3.16: Correlation between the raw data and the Barker 7 sequence, the correlation has an absolute maximum when the two sequence are aligned. The x axis of the first two plots correspond to the sample positions.

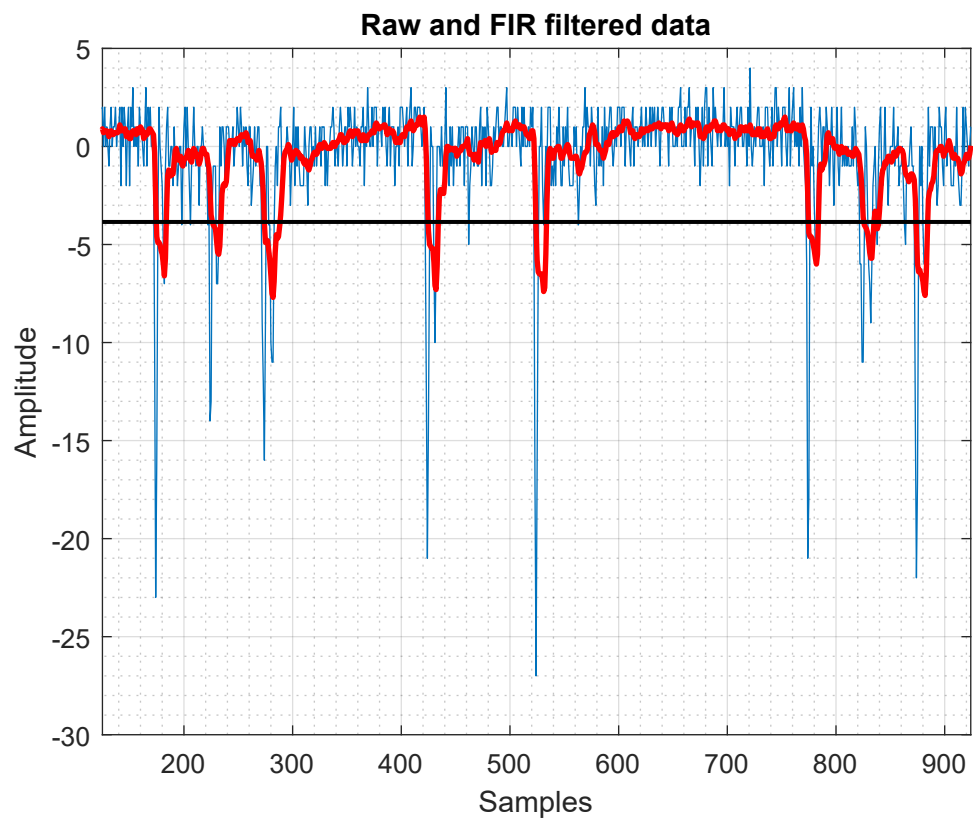


Figure 3.17: Tag ID recognition by mean of thresholding the FIR filtered data.



An example of application is shown in Figure 3.17. The raw data are shown in blue while the red curve represents the FIR filter output. The black line is the automatic threshold computed as half of the minimum of the filtered data. The effect of the filter is to smooth the signal and even the peaks in the sequence so that we can use a fixed threshold instead of an adaptive one.

Once the filtered data are thresholded, we can recognize the bits in the sequence and compute the Tag ID as their binary to decimal conversion. For example, the sequence shown in Figure 3.17 is recognized as the binary string 1110010 10000111 where 1110010 is the common preamble while 10000111 is the payload. The ID corresponding to that binary sequence expressed as a decimal number is 135. The main advantage of the new implementation is that we do not need to correlate the raw data with all possible sequences, we just filter and threshold them. The independence from the number of tags makes this the best solution when the number of tags to localize is very high.

However, this solution is more sensible to the signal level and to the SNR. When the signal is very low and noisy, the FIR filter reduces the noise as well as the signal level. Both solutions are maintained and is given freedom to the user to decide which one to apply depending on the application.

### Time Of Arrival estimation

The second part of the information we need to provide is the accurate TOA of the sequence. Since the FPGA processing clock works at a frequency eight times smaller than the sampling one and elaborates the incoming data in parallel, we have a roughness of the TOA of eight samples (equal to 8 ns). The operations performed in the ARM processor allow refining the coarse TOA received from the FPGA down to the sampling period of 1 ns (or less in case super-resolution is applied).

To refine the measurement, we correlate the data received from the FPGA with a mask shaped as a discrete linear ramp function defined as:

$$R[i] = \begin{cases} 1 & \text{if } 0 \leq i < 45 \\ -3(i - 45) & \text{if } 45 \leq i < 50 \end{cases} \quad (3.7)$$

where  $i$  is the mask sample index. The peculiar characteristic of this mask signal, that we have obtained empirically, is to highlight the edge of the first sequence peak allowing to measure the TOA corresponding to the direct signal path and not to the following ones associated with multipath. The result of this operation as well as the shape of the Ramp-like signal are shown in Figure 3.18.

The results of this correlation are thresholded in order to establish at which sample we find the first edge. The threshold mechanism can be chosen between manual and automatic. In the first case, the value of the threshold is statically set by a register; in the second case, for each received sequence, the threshold is set to one-third of

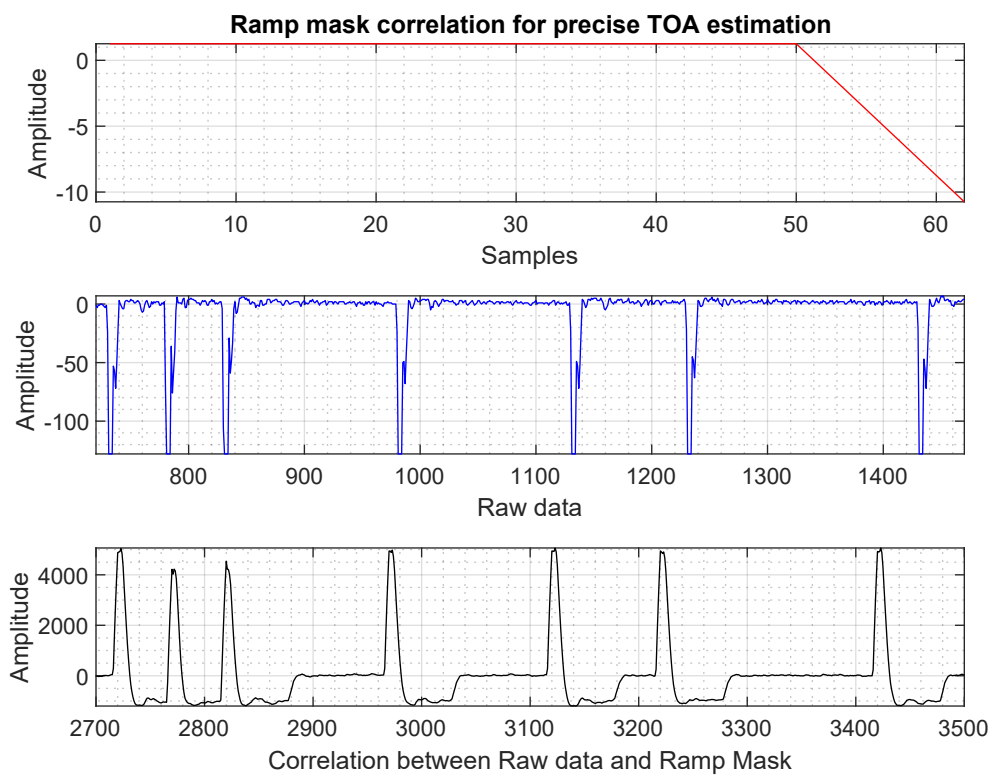


Figure 3.18: Correlation between the Ramp-like mask signal and the raw data, the correlation results underline the edge of the pulses.

the maximum value of the ramp correlation results.

The time delay of the first peak is combined to the coarse TOA received from the FPGA refining the measurement to the sample period (equal to 1 ns).

The computed Tag ID and TOA are sent to a host PC running the localization application.

## Super Resolution

The accuracy in the evaluation of the TOA is limited by the sampling period. In our case with a sampling frequency of 1 GHz, the accuracy is as low as 1 ns, which corresponds to 30 cm in distance. To increase the accuracy, we could increase the sampling frequency, but this would be extremely cost inefficient.

We propose a solution based on a super-resolution technique, that allows to increase the accuracy at least by a factor of two, bringing up the Sensor accuracy up to 15 cm.

The operations performed to implement the super-resolution technique are very time-consuming, limiting the maximum SRF of tags as the number of tags increases. When using the super-resolution, each Tag must have a lower SRF (with respect to the case without super-resolution); otherwise, the number of tags must be reduced. The super-resolution technique implemented on the ARM processor performs three main operations:

- Oversampling and linear interpolation;
- Data alignment, performed by means of a correlation between successive sequences;
- Averaging.

After the Tag ID recognition, each data block is stored in a matrix where, for each Tag, the last  $N$  (e.g.,  $N = 8$ ) received signals are saved. The first operation is to oversample by a factor  $M$  (e.g.,  $M = 2$ ) the data block and compute the new samples by linear interpolation.

The data alignment operation takes the last received sequence of each Tag and correlates it with each one of the previously saved  $N$  sequences. This operation allows us to find the relative delay of the  $N$  previous sequences with respect to the last received ones. Once the delay among the sequences is known, we align and sum them together. We take advantage of the fact that received sequences close in time do not change their shape dramatically; therefore, by aligning and summing them together, we obtain a better signal-to-noise ratio.

The method of aligning and averaging a certain number of previously received sequences is based on the assumption that the correspondent received signals do not change excessively. This situation can be guaranteed by a proper choice of the PRF

with respect to the typical maximum targets velocity to allow to consider the channel as quasi-static for the required averaging timeframe. This is also supported by the adoption of UWB signals that are resilient to multipath effect allowing a reliable correlation between successive signals. The obtained upsampled sum is correlated with the ramp mask in order to find the start of the first peak in the same way as we did in the case without super-resolution. The oversampling procedure allows to improve the accuracy in the TOA estimation by a number of times equal to the oversampling factor  $M$ . The upper bound to which we can increase the upsampling factor is given by the time required to process the oversampled data.

### 3.3.4 Ethernet Communication

The ARM processor communicates with the host application using Ethernet. The ARM software sets up three UDP sockets, one for each of the following tasks:

- To manage the pulse sequence raw data transmission for debug purposes;
- To manage the reception, decoding and re-transmission of commands from and to the host application;
- To communicate the accurate TOA and ID of the Tag associated to the sequence to the host PC.

The software associates to each Sensor a preset MAC address within a set of available addresses. The UDP socket configured for command reception is connected in listening mode at port 8080 and sets up the callback routine for command decoding. The raw data streaming interface is connected to port 8081 while the TOA and ID streaming interface is connected to port 8082.

The raw data streaming sends two different kind of data each one defined by a code word of 4 bytes that embeds the Sensor identifier. The code words used are the following:

- 0xFEED01DA, where 0xFEED is the operation code (OPCODE) word to notify the host application that the received UDP packet contains the pulse sequence raw data and 0x01DA stands for data (0xDA) coming from Sensor 1 (0x01). The UDP packet contains the highest accuracy TOA in the first four bytes after the OPCODE.
- 0xBABE01AA, where 0xBABE is the OPCODE word used to send to the user interface the value of the preamble thresholding block inside the FPGA. This value is used by the host PC interface for debug purposes. The second part of the OPCODE contains the Sensor number and the last byte of the code word;

The format of the UDP packets sent by the ARM processor to the host interface are shown in Figure 3.19. We decided to use two different sockets to communicate

<i>OPCode0</i>	<i>Sensor</i>	<i>OPCode1</i>	<i>Time Of Arrival</i>				<i>Data</i>
0xFEED	0x01	0xDA	0x76	0x54	0x32	0x10	1020 Bytes

<i>OPCode0</i>	<i>Sensor</i>	<i>OPCode1</i>	<i>Threshold</i>			
0xBABE	0x01	0xAA	0x76	0x54	0x32	0x10

<i>Time Of Arrival</i>				<i>Sensor</i>	<i>Tag ID</i>
0x76	0x54	0x32	0x10	0x01	0x12

Figure 3.19: Content organization of the UDP packet sent to the host interface.

data and threshold instead of a single one is present only for visual debug in order to simplify the signal inspections during in field tests. In future versions of the software the raw data will not be transmitted significantly reduce the bandwidth required for the communication. The format of the UDP packets used to receive and transmit commands is described will be described in the Host Application section.

## 3.4 Boot Sequence

The Boot procedure is completely automated. The Sensor processing board supports two different memory devices to choose for the boot process: The SD card and the NOR QSPI, the latter is the one currently used to boot the system. The procedure for the boot from NOT QSPI requires the following files:

- The .bif file (Boot Input File) listing the binary images of both FPGA software and ARM processor software that need to be merged together in a unique file. The required images are listed in the file in the boot order;
- The .elf file (Executable and Linkable Format) of the First Stage Boot Loader (FSBL);
- The FPGA bitstream file;
- The .elf file of the code running on the ARM processor.

We create the QSPI image and flash it into the memory using the JTAG interface. The boot process starts with a Power On Reset (POR), the SoC then reads the first part of the boot code from the on chip boot ROM. This first program configure the very basic input-output settings and peripherals required for the successive boot steps. When the boot rom code execution ends, the SoC checks the value of

the bootstrapping pins configured using external jumpers. The combination coded with the jumpers tells the which memory device (in our case a NOR QSPI) stores the next boot stage: The FSBL.

The FSBL is read from the memory device and copied on the On Chip Memory (OCM) where it is executed. The goal of this second stage is to complete the chip's peripheral and clocks configurations to the next part of the code.

At the end of its execution, the bootloader loads the FPGA image from the memory to the OCM and programs the FPGA fabric. Finally, the FSBL loads the ARM processor firmware from the memory to the OCM and start its execution.

## 3.5 Host Application

### 3.5.1 Introduction

The following sections describe the design details of the solutions implemented for the host application running on an external PC. The application communicates to the PoE switch using a standard data port and manages the UDP packets traffic of the local network receiving the UDP packet sent by the three ormore Sensors and decoding them. The information required for the processing are the name of the Sensor sending the packet, the ID of the Tag associated to the processed pulse sequence, its TOA and the raw data for plotting.

The application processes the TOA associated to the same Tag in the same SRI received from the Sensors and compute the Time Difference of Arrival (TDOA). One of the Sensors is chosen as a reference, and the TDOAs are referred to the reference Sensor using a reference Tag. The reference Tag, differently from all others, is kept in a fixed known position.

The obtained TDOA, prior to be used in the trilateration algorithm, is filtered using a median filter and a mean filter. The first filter is used to eliminate the outliers measurements that were caused by occlusions between Sensors and the tags. The result of the median filtering is then averaged using a mean filter with configurable window. The filtered TDOA measurements are used in the multilateration algorithm proposed in [15] to compute the (x, y) coordinates of the Tags and plot the results on a 2-D map.

### 3.5.2 Time Difference Of Arrival

The algorithm that computes the Tags position is based on the multilateration of the signal transmitted by the Tag and received by the Sensors. In order to work properly, it requires to know the TDOA among the Sensors (in an ideal situation, a minimum of three Sensors are synched using the same clock signal). The implemented architecture is the one presented in Figure 1.5. We define as

$T_i^{Sx}(N)$  the TOA, expressed in ns, of a sequence from a generic Tag  $T_i$  arriving at the Sensor  $Sx$  at instant of time  $N$ . The algorithm performing the TDOA will then compute the differences between the reference Sensor (S1) and the slave Sensors (S2 and S3) as:

$$TDOA_{12} = T_i^{S1}(N) - T_i^{S2}(N) \quad (3.8)$$

$$TDOA_{13} = T_i^{S1}(N) - T_i^{S3}(N) \quad (3.9)$$

Having the TDOA, it is sufficient to convert them into distances and use them together with the known positions of each Sensor in the multilateration formula [21]–[2]. The results will be the bidimensional coordinates of the Tag’s position. Since the Sensors are not synched using the same clock signal, we need to compensate for the frequency and time differences among the individual Sensor clocks using the reference Tag. Since the reference Tag is in a fixed and known position, the TOA of its signal to the Sensors is constant (except for small drifts in time). This allows to have a fixed reference for all Sensors to which compare all the TOA of signal from the other Tags. The signal from the reference Tag is continuously received by the Sensors to always compare the closest signals in time. The distance between the Sensors and the reference Tag translates into an offset representing the time difference between the Sensors and the reference Tag and it is equal to:

$$off_{12} = \frac{(d_{T_{Ref}S1} - d_{T_{Ref}S2})}{c} \quad (3.10)$$

$$off_{13} = \frac{(d_{T_{Ref}S1} - d_{T_{Ref}S3})}{c} \quad (3.11)$$

Where  $d_{T_{Ref}Sx}$  is the distance between the reference Tag and Sensor  $x$  and  $c$  is the speed of light. Furthermore, all TOA measurements must be referred to those associated with the reference Tag, compensating the clock’s time differences.

To compensate for the frequency difference, we multiply the TDOA by a correction factor called “SRFxy” that relates all the different measurements to the ratio between the sequence repetition frequency (SRF) of the reference Tag computed at the reference Sensor and the SRF of the reference Tag computed at the slave Sensors:

$$SRF_{12} = \frac{SRF_{T_{Ref}}^{S1}}{SRF_{T_{Ref}}^{S2}} = \frac{T_{Ref}^{S1}(N) - T_{Ref}^{S1}(N-1)}{T_{Ref}^{S2}(N) - T_{Ref}^{S2}(N-1)} \quad (3.12)$$

$$SRF_{13} = \frac{PRF_{T_{Ref}}^{S1}}{PRF_{T_{Ref}}^{S3}} = \frac{T_{Ref}^{S1}(N) - T_{Ref}^{S1}(N-1)}{T_{Ref}^{S3}(N) - T_{Ref}^{S3}(N-1)} \quad (3.13)$$

Where  $SRF_{T_{Ref}}^{S1}$  is the SRF of the reference Tag evaluated at the reference Sensor and  $SRF_{T_{Ref}}^{S2,3}$  is the SRF of the reference Tag at Sensor 2 or 3.

The final TDOA equation becomes:

$$TDOA_{12} = (T_i^{S1}(N) - T_{Ref}^{S1}(N)) - (T_i^{S2}(N) - T_{Ref}^{S2}(N) - off_{12}) \cdot SRF_{12} \quad (3.14)$$

$$TDOA_{13} = (T_i^{S1}(N) - T_{Ref}^{S1}(N)) - (T_i^{S3}(N) - T_{Ref}^{S3}(N) - off_{13}) \cdot SRF_{13} \quad (3.15)$$

The Sensors can sometimes miss a sequence due to obstacles or occlusions in the line of sight (LOS). The host application must be able to recognize when a sequence has been missed and discard the TDOA computation for the three Sensors. Knowing that the TDOA between the same sequences at different Sensors cannot be larger than the propagation time between the Sensors, the software considers only groups of TDOA smaller than the maximum propagation time considered in the scenario. Only when a correct group of TDOA is calculated, the software validates the localization.

To avoid ambiguity in the association of a Tag sequence with the reference one, it is sufficient to have the SRF of the reference Tag slightly different from that of the other Tags. In this way, there will be a net difference between the results corresponding to a correct time association and a wrong one. In order to reduce the impact of wrong measurements, the computed TDOA are filtered using a median filter and averaging filter. The Median filter with dimension N (e.g. N = 32) saves the last N TDOA values of each Tag, sorts them and extracts the fiftieth percentile. It is particularly efficient with impulsive noise because the measurements affected by such noise ends in the tails of the filter of the median operation and gets filtered out. The average filter with dimension M and distance D (e.g. M = 32 and D = 20) gets the TDOA values that are at a distance, in absolute value, smaller than D and gets inserted in a vector of M elements. The arithmetic mean of the vector is taken. The values of N and M can be changed by the user to obtain a faster but less precise system or a slower but more precise one.

The results are then used to compute the position of the target Tag by applying the multilateration algorithm.

### 3.5.3 Multilateration

The geometrical definition of the multilateration problem and the solution implemented in this work are described in detail in [15]. This approach has been chosen since it adopts TDOA measures to implement the localization not requiring a clock on board of Tags to transmit the time of departure. Furthermore, thanks to the adoption of short UWB sequences, we have a system more resilient to multipath effect. The problem of computing the 2D position of an object by means of the TDOA computation is geometrically translated into the problem of finding the intersection point between two hyperboloid of revolution with foci in the Sensors positions. The reference Sensor is a common focus between the two hyperboloids.

Using three Sensors and computing the TDOA among them allows us to find the intersection curve between the two hyperboloids, to fix the position on this curve we need an additional information given by the TOA of the reference Tag in the



known position. The problem is geometrically defined as follows.

Let  $R_{12}$  and  $R_{13}$  be the range differences from the Sensors obtained as:

$$R_{12} = c \cdot TDOA_{12} \quad (3.16)$$

$$R_{13} = c \cdot TDOA_{13} \quad (3.17)$$

Using the reference system shown in Figure 3.20 we define the distances among the Sensors as:

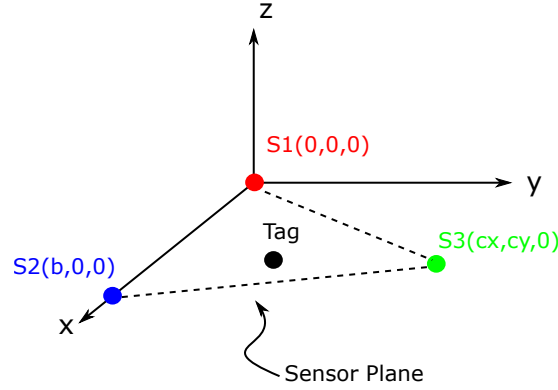


Figure 3.20: Coordinate system used for the multilateration technique. The origin of the reference system is placed in the reference Sensor position

$$R_{12} = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x - b)^2 + y^2 + z^2} \quad (3.18)$$

$$R_{13} = \sqrt{x^2 + y^2 + z^2} - \sqrt{(x - c_x)^2 + (y - c_y)^2 + z^2} \quad (3.19)$$

If we take the square of these two equation we obtain:

$$R_{12}^2 - b^2 + 2b \cdot x = 2R_{12}\sqrt{x^2 + y^2 + z^2} \quad (3.20)$$

$$R_{13}^2 - c^2 + 2c_x \cdot x + 2c_y \cdot y = 2R_{13}\sqrt{x^2 + y^2 + z^2} \quad (3.21)$$

where  $b$  is the Sensor 2 coordinate along the x axis and  $c_x$  and  $c_y$  are the Sensor 3 position components along x and y axis. The equations 3.20 and 3.21, represent two hyperboloids of revolution having foci in Sensor 1 and 2 the first, and 1 and 3 the second.

Supposing that the TDOA are not equal to zero we can set equation 3.20 equal to 3.21. Simplifying the equation we obtain:

$$y = g \cdot x + h \quad (3.22)$$

Where :

$$g = (R_{13} \cdot (\frac{b}{R_{12}}) - c_x) / c_y \quad (3.23)$$

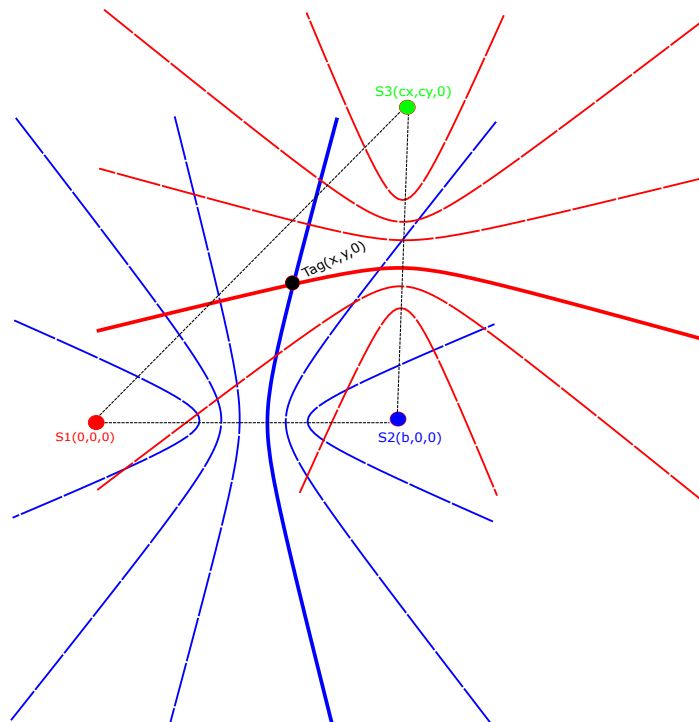


Figure 3.21: Visualization of the hyperbolic localization problem using TDOA computation. The reference Tag in known position fixes the point where the two curves intersect.

$$h = \frac{\left[ c^2 - R_{13}^2 + R_{13} \cdot R_{12} \left( 1 - \frac{b}{R_{12}} \right)^2 \right]}{2c_y} \quad (3.24)$$

Equation 3.22 describes a plane orthogonal to the plane containing the three Sensors. The position of the Tag lies in this plane otherwise the intersection of the two hyperboloids won't be a plane curve.

If we substitute the equation 3.22 into 3.20 we obtains:

$$z = \pm \sqrt{d \cdot x^2 + e \cdot x + f} \quad (3.25)$$

That when squared leads to:

$$z^2 = d \cdot x^2 + e \cdot x + f \quad (3.26)$$

Where:

$$d = - \left[ 1 - \left( \frac{b}{R_{12}} \right)^2 + g^2 \right] \quad (3.27)$$

$$e = b \cdot \left[ 1 - \left( \frac{b}{R_{12}} \right)^2 \right] - 2g \cdot h \quad (3.28)$$

$$f = \left( \frac{R_{12}^2}{4} \right) \cdot \left[ 1 - \left( \frac{b}{R_{12}} \right)^2 \right]^2 - h^2 \quad (3.29)$$

From equation 3.25 one can notice that the symmetry of the intersection curve with respect to the Sensor plane is required. When dealing with range difference measurements the intersection curve of the plane with the hyperboloid of revolution can be an hyperbola or an ellipse depending on the sign of  $d$ .

Since we are dealing with a 2D localization problem, we can set the altitude value to  $z = 0$ . If we substitute this value of  $z$  into equation 3.26 we have that the localization problem is reduced to find the roots of a polynomial of the second order. If the values obtained for  $e$ ,  $d$  and  $f$  respect the condition  $e^2 > 4df$ , we will have two real solutions for the square root. The component along  $x$  of the 2D position can be obtained as:

$$x_{1,2} = \frac{-e \pm \sqrt{e^2 - 4 \cdot d \cdot f}}{2 \cdot d} \quad (3.30)$$

If we substitute the values computed for  $x_{1,2}$  in equation 3.22 we obtain the  $y_{1,2}$  values.

If we substitute the two obtained point in:

$$a_1 = \sqrt{(x_1 - b)^2 + y_1^2} - \sqrt{x_1^2 + y_1^2} - R_{12} \quad (3.31)$$

$$a_2 = \sqrt{(x_2 - b)^2 + y_2^2} - \sqrt{x_2^2 + y_2^2} - R_{12} \quad (3.32)$$

we can discriminate which of the two solutions leads to a correct range differences measurements while the other would place the Tag position outside of the localization area. The solution that lies inside the localization area is taken as the correct 2D position of our Tag.

The host application implement this algorithm using a function that receives as input the following parameters:

- Sensor 2 position on the  $x$  axis (the position along  $y$  is zero by construction of the reference system);
- Sensor 3 position components  $c_x$  and  $c_y$ ;
- The Range difference between Sensor 2 (3) and the reference Sensor. The two values are obtained from the TDOA computed using 3.16 and 3.17.

The output of the function are the Tag position along the  $x$  and  $y$  axis.

### 3.5.4 Localization and position plotting

The 2D position obtained using the multilateration algorithm are obtained using the coordinates system previously shown in Figure 3.20. To correct the localization results for a generic coordinate system we need to use a more generic coordinates system like the one shown in Figure 3.22. The reference system R1 corresponds to the one used for the multilateration algorithm, if a different reference system is used, like the one indicated as R2 we need to perform a rotation and translation of the coordinate system. The general implementation of the localization requires to:

1. Provide to the multilateration function the positions of the Sensors using the reference system R1. If the Sensors positions are identified using a reference system like R2, we perform a rototranslation from one coordinates system to the other;
2. Perform the multilateration algorithm using the TDOA;
3. Correct the obtained 2D positions re-rotating and translating the coordinate system.

The obtained 2D position are then saved in a buffer.

All the stored position are plotted on the map. The plotting operations are timed using a counter, each time the counter reaches the threshold, a new plotting operation is performed.

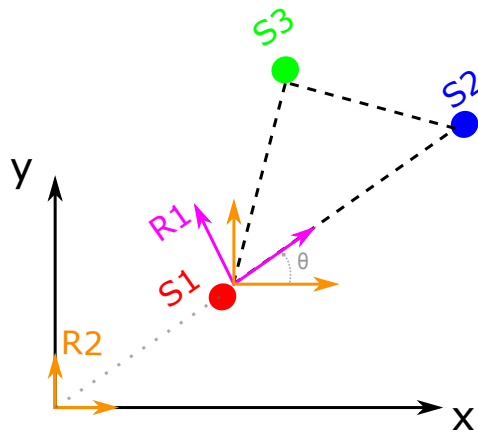


Figure 3.22: Generic coordinate system. The multilateration algorithm requires a coordinate system like R1. The passage between coordinates system is done through rototranslation.

### 3.5.5 UDP Packets Management

The three Sensors are connected together with a PoE ethernet switch to the host PC. The data exchange is controlled by three threads: A first thread that sends configuration commands to the designated Sensor, a second thread that receives the pulse sequence raw data and plot them for debug purpose and a third thread that receives the TOA and Tag ID data and apply the multilateration algorithm. The thread for configuration commands sends UDP packets to port 8080 creating a socket connected to the IP address of the recipient Sensor. The configuration commands packet are organized as shown in Figure 3.23. The first two bytes represents the operation code. The possible operation codes are 0x0A0C to read the value

<i>Cmd</i>		<i>Base</i>	<i>Off</i>	<i>Data</i>			
0x0A	0x0D	0x00	0x00	0x76	0x54	0x32	0x10

Figure 3.23: Custom format of the UDP Packets used for configuration command operations.

from a register in FPGA, 0x0A0D to write a value in the register, 0x0A0E to enable or disable the super-resolution algorithm in the ARM processor, 0x0A0F to set the value of the threshold used by the Tag ID recognition algorithm in one of the two implementation and 0x0A10 to choose between the Tag ID recognition algorithms. The third and fourth byte in the packet are the register base address inside the FPGA peripheral and its offset with respect to the base address respectively. The last four bytes are the data to be written into the register. The base address and offset fields are used for data if the command is addressed to the ARM processor instead of the FPGA.

### 3.5.6 Graphic User Interface layout

The functionality and details of the Graphic User Interface (GUI) will be described in this section. The user interface has been developed to work for two different applications.

The first application is generically called LOCalization SYstem (SILOC in Italian). The goal of this application is to localize and track the movement of different tags inside the localization area with high precision and accuracy. The application can be used in two different functioning modes: the standard mode or super-resolution mode. It is possible to switch between modes by simply ticking a dialog box in the user interface. The application is optimized to track fast Tags with lower accuracy in standard mode and to track slower Tags with higher accuracy in the super-resolution mode.

The second application is called Package Tracker (PackTrack). This application integrates new features on top of the SILOC application by allowing the user to monitor the movement of tags over long periods of time and comparing the computed position with a configurable tolerance. The application triggers an alarm whenever the Tag moves from its position by a distance larger than the tolerance and a continuously updated data log allows to track the tags movement in long time scenarios.

The tolerance is set after a configurable interval and can be set to be static or dynamic. The static tolerance is set by sampling the initial position of the Tag and checking each new computed position with respect to this first one. The alarm is triggered if the Tag moves away from the initial position of a distance larger than the tolerance.

The dynamic tolerance is updated to the latest position after a programmable amount of time. If the Tag moves around its position more than the tolerance in this amount of time it triggers the alarm.

The user interface is divided in two main regions. The Localization area, on the left side of the GUI, shows the area covered by the localization system and the localized tags while the Configuration area, on the right shows the configuration buttons and the pulse sequence raw data received by the Sensors. The layout of the user interface is shown in Figure 3.24. At the center of the GUI we have the configuration buttons used to set the PackTrack application timers.

The first text box from the top is used to set the Refresh Position Timer (RPT) interval, expressed in seconds. Each time this timer reach the set value, the application samples the last position of all the localized tags and update the log file. For each sampled position we save the date, timestamp, Tag ID,  $x$  and  $y$  position and the alarm trigger.

Below the RPT there is a second text box used to set the Initial Position Timer (IPT). This second timer is used to wait a fixed amount of time to average the position measured for each Tag before sampling it and using it to set the position tolerance. By changing the value of the drop down menu below the IPT textbox, we can set two different mode of operation: fast and slow. The fast mode uses the position sampled after the initial position timer has reached the target value and sets the tolerance in  $x$  and  $y$  coordinates around this position. The value of the initial position in  $x$  and  $y$  is saved in the log file together with the tolerance value and the type method used. The slow mode performs the same operation of the fast mode with the only difference that the initial position is not sampled only once but is sampled each time we refresh the position. The Tag SRI is smaller than the RPT interval allowing to collect many position before sampling the initial position again. The goal of this method is to compensate slow variations due to temperature or other events.

The third text box is called Position Tolerance (PT) and represents the radius of a circle centered in the initial position of each Tag. The new Tag position must fall

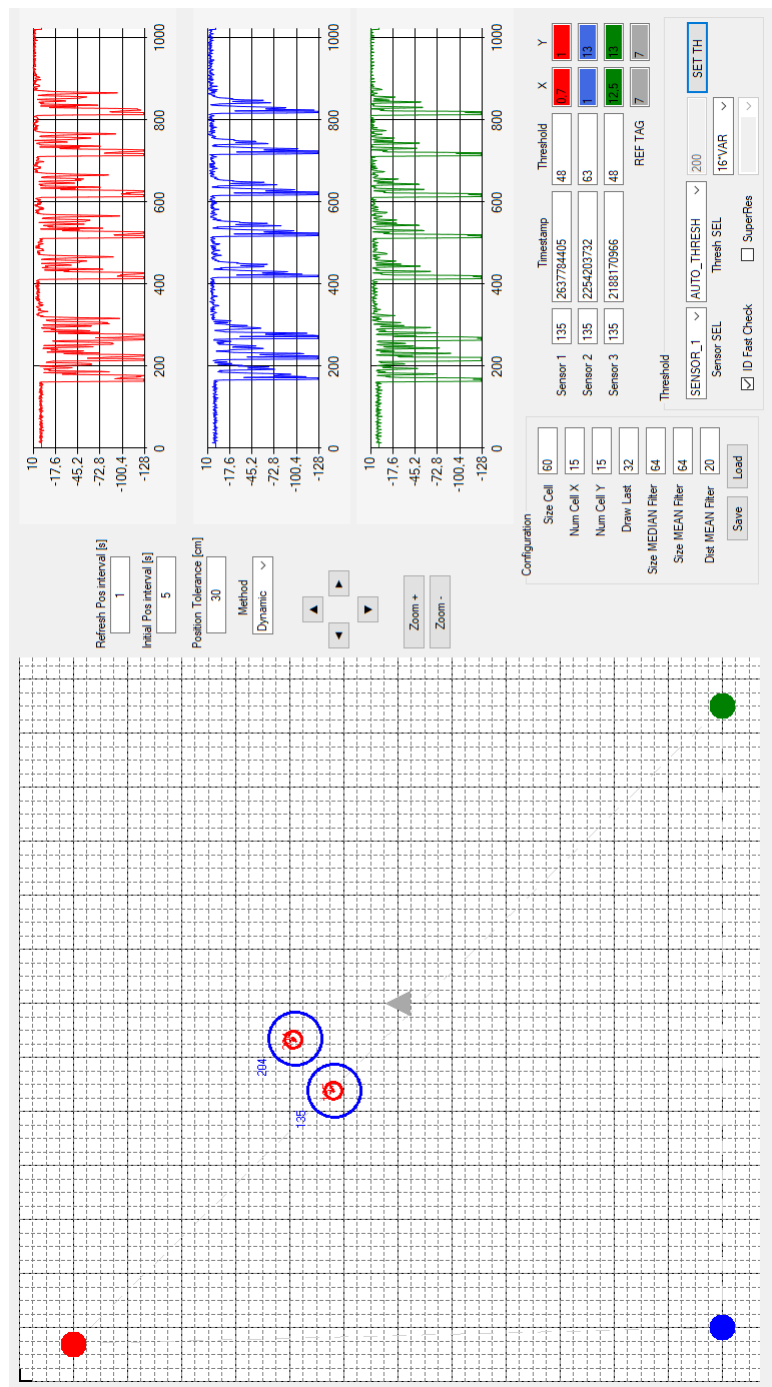


Figure 3.24: Graphic User Interface layout, on the left is shown the localization area mapped by the Sensors and the mapped tags. The right side shows the raw data plots and configuration buttons.



inside the circle to avoid triggering the alarm.

The configuration of these parameters is specific to the PackTrack application while the features described in the followings apply to both the PackTrack and SILOC applications.

The top right of the Configuration region presents three panels showing the pulse sequence raw data in red, blue and green that are received respectively from Sensor one, two and three. The data are associated to the last position shown in the localization area. The raw data are used as visual debug feature to evaluate the intensity and the SNR of the received signal and are not used by the GUI in the localization process.

Below the third panel there are nine text boxes organized on three lines, one for each Sensor, containing the following information:

- The Tag ID of the last sequence received at Sensor X, associated to the raw data plotted on panel X;
- The TOA associated to the sequence;
- The actual value of the threshold set in FPGA;

The information shown in these text boxes are used for debug only.

Close to these text box there are two columns used to configure the X and Y position of the three Sensors and the reference Tag. The values set in these boxes are used by the interface during the rotation of the coordinate system. The Sensors and the reference Tag are drawn in these position in the localization map on the left. The map origin is placed in the top left corner.

The lower right corner of the GUI is enclosed into a bounding box called Threshold. If the user wants to send a command to a specific Sensor, it has to select the target Sensor ID using the first drop down menu on the left and to select the command choosing from the second drop down menu.

The available commands are:

- Manual threshold, to set a static value for the FPGA threshold. The value to be set must be written in the text box on the right of this drop down menu;
- Automatic threshold, to set a dynamic value for the FPGA threshold. The value must be selected from the drop down menu below the text box used to set the manual threshold value. The available values range from 8 to 48 times the value of the variance of the raw data;
- Enable super-resolution, to enable or disable the super-resolution algorithm. The value is set using the tick box called SuperRes.
- ID threshold, used to change the value of the threshold used to recognize the single bits of the Tag ID string during the recognition operations.

The Set Th button is used to send the commands selected using the drop down menus. The last command in this section is the tick box called ID fast check. The Tag ID recognition algorithms described in previous sections can be chosen by ticking this box.

The Configuration bounding box contains the text box used to set the main user interface parameters. The localization area is divided in cells. The user can set the number of cells along the  $x$  and  $y$  axis and the size of the cell itself. The other text boxes are used to change the size of the median and mean filters allowing the user to change the responsiveness of the positioning operations.

A smaller dimension of the filters sizes allows to track faster changes in the position of the Tag at the expense of a less steady position.

The "Draw last" text box contains the number of older position that the interface need to store and plot for each Tag. The last configurable parameter is the "Distance MEAN filter", used to discard those measurements that deviate from the mean position of more than this value. The value is expressed in number of samples, since the sampling frequency is fixed to 1 GHz we have that the deviation of one samples translated into distance is equal to 30 cm. The usual values for this parameter ranges from 10 to 25 samples or 3 to 9 meters if expressed in distance.

The buttons "Save" and "Load" are used to save the current interface configuration into a text file or to load a pre-configured setting.

The left side of the interface shows a map where all the Tags position are plotted. The full colored circles represents the three Sensors where the colors, like we did in the raw data panels described before, are red for Sensor one, blue for Sensor two and green for Sensor three. The light grey triangle represent the reference Tag while the empty red circle represents the latest localized position of each Tag. The ID of the Tag is printed on the top left corner of both the latest computed position and the blue circle representing the position tolerance. The map is divided in a square grid whose dimension in  $x$  and  $y$  are configurable.

The second window of the user interface is shown in Figure 3.25 and it is only used for debug. The window shows relevant statistics for three different tags at the three Sensors. The column on the left reports the number of packets received in one second at each Sensor for the associated Tag. The right column shows the SRF computed as the difference between the latest received timestamp and the previous one, expressed in nanoseconds. The start and stop buttons are used to trigger or inhibit the localization process.

The textboxes called "Msg31" and "Msg21" stamps the number of UDP packets that have been analyzed in the last second by performing a correct TDOA measurement performed between Sensor X and the reference one.

The large textbox at the bottom of the form reports the following informations:

- Instantaneous value, median, arithmetic mean and variance of the TDOA between Sensor 2 (3) and 1;

The screenshot shows a window titled "Form\_STATS" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains a data entry form with the following elements:

- Sens 1:** A label followed by three input fields containing "22", "100992672", and "REF Tag".
- Start:** A button followed by two input fields containing "22" and "101295232".
- Stop:** A button followed by two input fields containing "22" and "101316328".
- Sens 2:** A label followed by three input fields containing "22", "100993346", and "REF Tag".
- Tag 1:** A label followed by two input fields containing "22" and "101295906".
- Msg 21:** A label followed by three input fields containing "22", "22", and "101317003".
- Tag 2:** A label followed by two input fields containing "22" and "101316328".
- Sens 3:** A label followed by three input fields containing "22", "100992676", and "REF Tag".
- Tag 1:** A label followed by two input fields containing "22" and "101295235".
- Msg 31:** A label followed by three input fields containing "22", "22", and "101316331".
- Tag 2:** A label followed by two input fields containing "22" and "101316331".

At the bottom of the window is a text area containing the following statistics:

```
cur: -30
med: -29
avg: -29,171875
var: 0

cur: -10
med: -10
avg: -10,171875
var: 0

Position
Pos x: 921,9001
Pos y: -315,1078
```

Figure 3.25: Second form, used to show some relevant statistics.

- The current position of atwo pre-selected Tag IDs.



# Chapter 4

## Results

To test and validate the design and the manufactured final prototype we performed different in field test. The results obtained with the test, showing the capabilities of the systems in terms of accuracy, resolution, and targets tracking are shown in the following sections. These results have been published by the author in advance in [4]. All tests have been performed in the 8m x 8m laboratory room of our Institute that can be considered as an indoor, realistic, harsh environment. The laboratory setup is shown in Figure 4.1. The positions of the three Sensors and the reference Tag are highlighted. The optimal position for the reference Tag is at the center of the area covered by the Sensors, to maximize the signal received by all Sensors. The localization accuracy and resolution are not affected by the relative positions of the Sensors, reference Tag and tags as long as the Sensors have good signal reception and line of sight with the tags.

All measurements have been taken using three Sensors connected through a PoE LAN switch that also provides the voltage supply to each Sensor.

The Tags transmitted sequence has a 7 bit long preamble, whereas the ID sequence is 8 bit long. The duration of each bit is fixed to 50 ns and contains a single 2 ns UWB pulse at 7 GHz. The Tags transmitted SRF is set to 20 Hz (50 ms) and the receiving sampling frequency is 1 Gsps.

### 4.1 Accuracy test

The first test shown has been performed to verify the localization accuracy. The set up and obtained results are presented in Figure 4.2. The three Sensors are positioned to form a triangular localization area of about 3 m side length. The red, blue and green dots indicates the three Sensors positions while the grey dot indicates the reference Tag position. A target Tag was placed in eight known positions called "Ground Truths", indicated by the black circles. Each black circle position is  $\pm 60$  cm away from the reference Tag along the x or y direction or both. For each target



Figure 4.1: Indoor laboratory measurement setup. In green, red, blue and gray are highlighted respectively the three Sensors and reference Tag positions.

Tag position 256 localization results have been collected. The magenta cloud points indicate the positions evaluated enabling the super-resolution technique while the blue ones indicate the standard mode of operation.

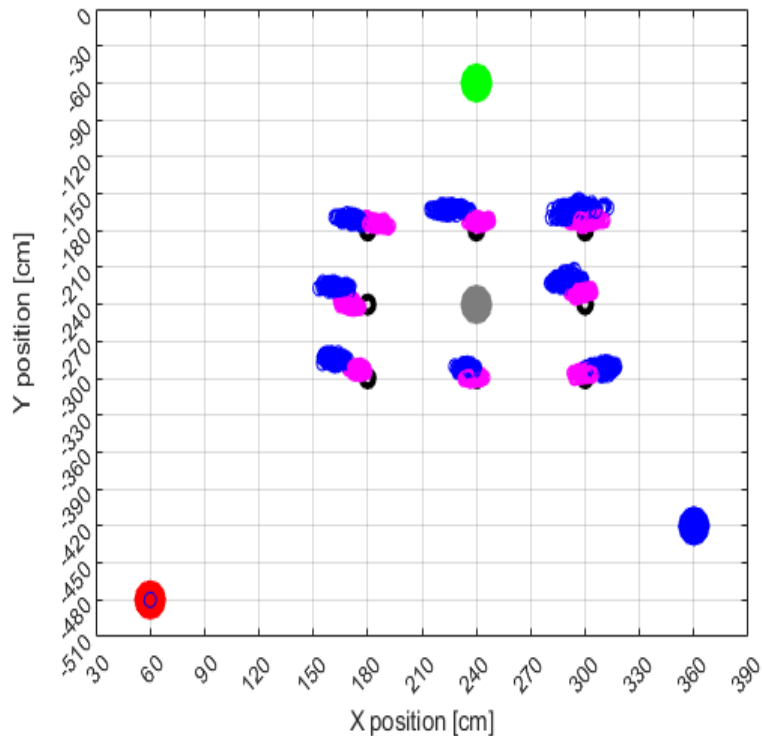


Figure 4.2: Ground truth accuracy measurement. The case with super resolution (magenta) performs better both in accuracy and precision when compared with the standard use case (blue).

As expected, the positions are more accurate when the super-resolution technique is enabled; they are also more precise as visible from the smaller, less scattered magenta point cloud. The obtained accuracy and precision of the localization in the super-resolute case is around 10 cm.

## 4.2 Resolution test

The resolution of the system is the capability to clearly recognize two or more different Tags that are close to each others as separated targets. The resolution is tested localizing two distinct Tags placed at different distances up to the point that they are not distinguishable anymore. The set-up for the test is the same used previously to test the accuracy.

The results obtained with two Tags positioned at four different decreasing distances with the super-resolution enabled are reported in Figure 4.3. Each plot displays 256 localization results.

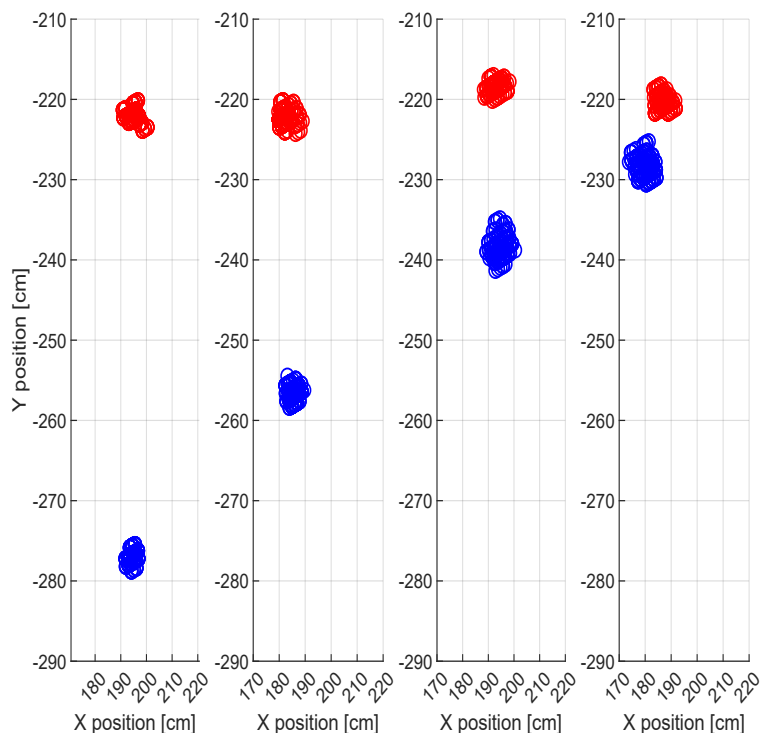


Figure 4.3: Resolution measurement using two tags spaced 60, 40, 20 and 10 cm apart and enabling the super resolution.

Starting from the left, the two targets were spaced 60, 40, 20 and 10 centimeters respectively. In the last case the targets are spaced at the accuracy limit and they are still clearly separated.

### 4.3 Tracking test

To demonstrate the system tracking capabilities, kinematic tests have been performed, in this case, without applying the super-resolution technique to allow a higher responsiveness. We have attached a Tag to a rotating structure and placed the reference Tag at the center of rotation. During the test we acquired 4096 localization results of the Tag rotating with a radius of 90 cm at a speed of 6 rotations per minute. The results are shown in Figure 4.4 and demonstrate the continuous track and very good precision and accuracy. The maximum distance obtained between the localization results in red and the ground truth in blue is equal to less than 30 cm without the super-resolution enabled.



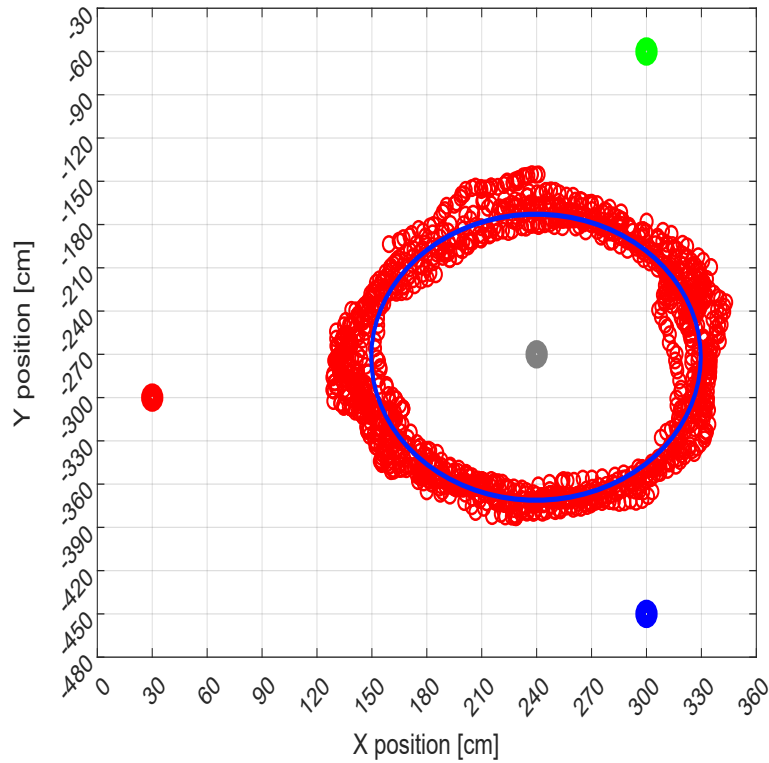


Figure 4.4: The tracking measurement results. The red dots represents the 4096 localization results of a Tag rotating around the reference Tag while the blue circle is the real track of the Tag with 90 cm radius.

To further demonstrate the tracking capabilities in a harsh indoor environment, we have moved the Sensors at the three very corners of the laboratory room to cover the whole area; the measurements environment have been already shown in Figure 4.1. Also in this case, the reference Tag have been placed at the center of the area covered by the Sensors for optimal reception and super-resolution technique has not been applied. In Figure 4.5, the blue line represents the track walked by a person carrying the Tag while the red dots represents the localization results. The results show a continuous track and a very good precision and accuracy even when one Sensor LOS may be occluded due to the person carrying the Tag. The presented results show the good tracking capability of the system as well as a very high accuracy.

## 4.4 Comparison with other systems

This thesis presented an innovative low-cost UWB RTLS where, for the proposed architecture, we designed and manufactured custom hardware and software for both

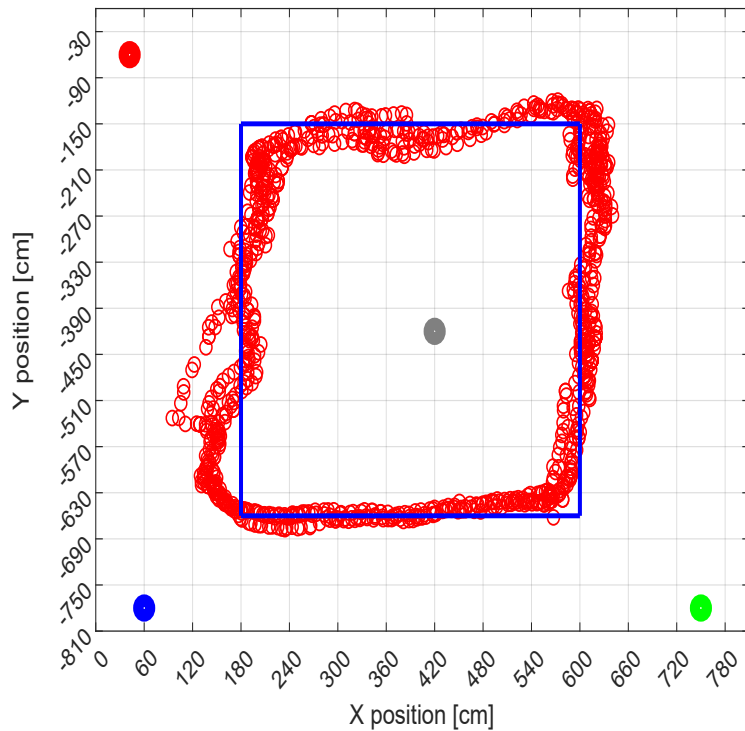


Figure 4.5: The tracking measurement results. The Sensors are moved further away to cover the whole room area. The blue lines represents the track walked carrying the Tag, the red dots represents 1050 localization results.

the Sensors and the tags. The system is based on a one-way ranging method that significantly reduces the Tag and Sensor complexity and cost. The use of a reference Tag in a fixed position allows synchronizing the Sensors, eliminating the need for a common timing reference.

We demonstrated the system capabilities to locate tags with 10-cm accuracy and resolution at a typical update rate of 20 Hz by applying the super-resolution technique in an indoor harsh laboratory environment. We also demonstrated the tracking capabilities of the system.

A comparison between our system and other solutions already on the market and described in section "Existing systems available on the market", is presented in Table 4.1. The comparison is based on localization accuracy and implemented localization approach. The accuracy of each system has been taken from the datasheet.

Table 4.1: Comparison among UWB RTLS systems

System	Algorithm	Accuracy [cm]
Pulson (P330)	TOA,TDOA,TWR	10
Ubisense	AOA,TDOA	15-20
Zebra	TDOA	30
Sewio	TDOA	30-50
OpenRTLS	TDOA	30
Quantitec	TDOA	15
This work	TDOA	10

Compared with existing solutions, our system is placed among the ones that have the best accuracy performances when the super-resolution technique is applied. This demonstrates the reaching of our goal to design and prototype a system architecture with localization accuracy comparable to the existing, more expensive solutions already in the market.

A market survey placed the price of all the competitor systems in the order of thousands of USD for a typical indoor one-room installation, while our system cost, due to the hardware and software custom design, is in the order of hundreds of USD.

## 4.5 Summary of innovations

The comparison with existing solutions already available on the market highlighted the very good performances of our system. These good results have been achieved introducing a set of hardware and software innovations:

- Hardware co-simulation allowed the integration of the UWB antenna with the RF oscillator and brought to a more compact and cost efficient solution for the Tags and to the complete control over the design parameters;
- The use of a reference Tag to synchronize the sensors allowed to simplify the Tag hardware and the overall system architecture eliminating the need of an highly stable and accurate clock distributed among the sensors;
- The use of Barker sequence allowed, even with short sequences, to have optimum performances with reasonable sidelobe level. Moreover, the shorter the sequence, the lesser the energy transmitted and so the power consumed by the tag;
- The super resolution algorithm allowed to increase the localization accuracy by a number of times equal to the oversampling factor. The limit, however,

is associated to the computing power available by the microprocessor and to the number of tag to localize;

- The adoption of short UWB signals reduces the effect of multipath allowing more accurate detection of TOA.

All these design choices permitted to reduce the overall cost of Sensors and Tags and to achieve an optimum localization accuracy. It is very hard to define the relative impact of each of these innovations.

## 4.6 Conclusions

In this work we presented a RTLS system based on TDOA computation.

The proposed architecture reduces the overall system cost since it does not requires high precision time synchronization between the Sensors. The synchronization is obtained by means of a reference Tag to which all TDOA measurements are referenced.

Low-cost and low-power custom hardware has been developed for both Tags and Sensors. The Tag is composed of two boards implementing the digital driver circuit and the integrated UWB antenna and oscillator. The Sensor is composed by a custom UWB receiver and an FPGA based processing board.

The system has been tested both in dynamic and static situations to verify its tracking capabilities and localization accuracy. The achieved localization accuracy is equal to 10 cm and is comparable to market solutions.

The future developments of this work will focus on engineering aspects of the project. The Tag will undergo a phase of product engineering and enclosure design to satisfy mass production requirements. The host application software will be extended to manage different fields of application and multiple system installations at the same time in order to cover large areas.

# Bibliography

- [1] Hind Abdalsalam Abdallah Dafallah. “Design and implementation of an accurate real time GPS tracking system”. In: *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*. 2014, pp. 183–188. DOI: [10.1109/ICeND.2014.6991376](https://doi.org/10.1109/ICeND.2014.6991376).
- [2] A. Alarifi et al. “Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances”. In: *Sensors* 16.5 (2016). ISSN: 1424-8220. DOI: [10.3390/s16050707](https://doi.org/10.3390/s16050707). URL: <https://www.mdpi.com/1424-8220/16/5/707>.
- [3] Stefano Bottigliero and Riccardo Maggiora. “Integration and Prototyping of a Pulsed RF Oscillator with an UWB Antenna for Low-Cost, Low-Power RTLS Applications”. In: *Sensors* 21.18 (2021). ISSN: 1424-8220. DOI: [10.3390/s21186060](https://doi.org/10.3390/s21186060). URL: <https://www.mdpi.com/1424-8220/21/18/6060>.
- [4] Stefano Bottigliero et al. “A Low-Cost Indoor Real-Time Locating System Based on TDOA Estimation of UWB Pulse Sequences”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–11. DOI: [10.1109/TIM.2021.3069486](https://doi.org/10.1109/TIM.2021.3069486).
- [5] *Broadband Low Noise Amplifier 2 - 18 GHz MAAL-011130, Macom, Rev. V1*. URL: <https://cdn.macom.com/datasheets/MAAL-011130.pdf>.
- [6] Riccardo Carotenuto. “A range estimation system using coded ultrasound”. In: *Sensors and Actuators A: Physical* 238 (2016), pp. 104–111. ISSN: 0924-4247. DOI: <https://doi.org/10.1016/j.sna.2015.12.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0924424715302545>.
- [7] *Ceramic Low Pass Filter LFCG-1200+, Minicircuit*. URL: <https://www.minicircuits.com/pdfs/LFCG-1200+.pdf>.
- [8] Witsarawat Chantaweksomboon et al. “On performance study of UWB real time locating system”. In: *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. 2016, pp. 19–24. DOI: [10.1109/ICTEmSys.2016.7467115](https://doi.org/10.1109/ICTEmSys.2016.7467115).

- [9] Rogers Corporation. *Rogers Corporation Website, RO3003 Substrate Datasheet*. Available ONLINE: URL: <https://rogerscorp.com/-/media/project/rogerscorp/documents/advanced-electronics-solutions/english/data-sheets/ro3003g2--data-sheet.pdf>.
- [10] Rogers Corporation. *Rogers Corporation Website, RO4350B Substrate Datasheet*. Available ONLINE: URL: <https://rogerscorp.com/advanced-connectivity-solutions/ro4000-series-laminates/ro4350b-laminates>.
- [11] J. A. Corrales, F. A. Candelas, and F. Torres. "Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter". In: *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2008, pp. 193–200. DOI: [10.1145/1349822.1349848](https://doi.org/10.1145/1349822.1349848).
- [12] Paolo Dabove et al. "Indoor positioning using Ultra-wide band (UWB) technologies: Positioning accuracies and sensors' performances". In: *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. 2018, pp. 175–184. DOI: [10.1109/PLANS.2018.8373379](https://doi.org/10.1109/PLANS.2018.8373379).
- [13] *DecaWave RTLS Website*. Available at: URL: <https://www.decawave.com/technology1>.
- [14] Analog Devices. *Analog Devices LTC6991 silicon oscillator with a programmable period Datasheet*. Available ONLINE: URL: <https://www.analog.com/en/products/ltc6991.html>.
- [15] B.T. Fang. "Simple Solutions for Hyperbolic and Related Position Fixes". In: *IEEE Trans. Aerosp. Electron. Syst* 26.5 (1990), pp. 748–753.
- [16] Thomas Gigl et al. "Analysis of a UWB Indoor Positioning System Based on Received Signal Strength". In: *2007 4th Workshop on Positioning, Navigation and Communication*. 2007, pp. 97–101. DOI: [10.1109/WPNC.2007.353618](https://doi.org/10.1109/WPNC.2007.353618).
- [17] *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD*. Available at: URL: <https://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>.
- [18] S.B. Gokturk, H. Yalcin, and C. Bamji. "A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions". In: *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 2004, pp. 35–35. DOI: [10.1109/CVPR.2004.291](https://doi.org/10.1109/CVPR.2004.291).
- [19] Isola Group. *Isola Group website, Astra MT77 Substrate Datasheet*. Available ONLINE: URL: <https://www.isola-group.com/pcb-laminates-prepreg/astra-mt77-laminate-and-prepreg/>.
- [20] Walter Hirt. "The European UWB Radio Regulatory and Standards Framework: Overview and Implications". In: *2007 IEEE International Conference on Ultra-Wideband*. 2007, pp. 733–738. DOI: [10.1109/ICUWB.2007.4381041](https://doi.org/10.1109/ICUWB.2007.4381041).

- [21] ICAO. *Multilateration (MLAT) Concept of Use, Edition 1, ICAO Asia and Pacific Office*. Available ONLINE at: URL: [https://www.icao.int/%20APAC/Documents/edocs/mlat\\_concept.pdf](https://www.icao.int/%20APAC/Documents/edocs/mlat_concept.pdf).
- [22] *IEEE 802.15.4z Standard for Low-Rate Wireless Networks–Amendment 1: Enhanced Ultra Wideband (UWB) Physical Layers (PHYs) and Associated Ranging Techniques*. Available at: URL: [https://standards.ieee.org/standard/802\\_15\\_4z-2020.html](https://standards.ieee.org/standard/802_15_4z-2020.html).
- [23] *IETSI TR 103 181-3 Short Range Devices (SRD) using Ultra Wide Band (UWB); Part 3: Worldwide UWB regulations between 3,1 and 10,6 GHz*. Available at: URL: [https://www.etsi.org/deliver/etsi\\_tr/103100\\_103199/10318103/02.01.01\\_60/tr\\_10318103v020101p.pdf](https://www.etsi.org/deliver/etsi_tr/103100_103199/10318103/02.01.01_60/tr_10318103v020101p.pdf).
- [24] Analog Devices Inc. *ADF4360-7 Integrated Synthesizer and VCO Datasheet*. Available ONLINE at: URL: <https://www.analog.com/en/products/adf4360-7.html>.
- [25] Analog Devices Inc. *HMCAD1511 High Speed Multi-Mode 8-Bit 30 MSPS to 1 GSPS A/D Converter Datasheet*. Available ONLINE at: URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/hmcad1511.pdf>.
- [26] Infineon. *Infineon website Providing Transistor Model*. Available ONLINE: URL: <https://www.infineon.com/cms/en/product/rf-wireless-control/rf-transistor/ultra-low-noise-sigec-transistors-for-use-up-to-12-ghz/bfp740/>.
- [27] Infineon. *Infineon, Application note AN 1807 PL32 1808 132434: RF and microwave power detection with Schottky diodes*. Available ONLINE: URL: [https://www.infineon.com/dgdl/Infineon-AN\\_1807\\_PL32\\_1808\\_132434\\_RF%20and%20microwave%20power%20detection%20-AN-v01\\_00-EN.pdf?fileId=5546d46265f064ff0166440727be1055](https://www.infineon.com/dgdl/Infineon-AN_1807_PL32_1808_132434_RF%20and%20microwave%20power%20detection%20-AN-v01_00-EN.pdf?fileId=5546d46265f064ff0166440727be1055).
- [28] Texas Instruments. *Texas Instruments Website, TPS22917 Ultra low leakage switch Datasheet*. Available ONLINE: URL: <https://www.ti.com/product/TPS22917>.
- [29] Antonio Ramón Jiménez Ruiz and Fernando Seco Granja. “Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis”. In: *IEEE Transactions on Instrumentation and Measurement* 66.8 (2017), pp. 2106–2117. DOI: [10.1109/TIM.2017.2681398](https://doi.org/10.1109/TIM.2017.2681398).
- [30] Babburu Kiranmai and P. Kumar. “Performance Evaluation of Barker Codes using New Pulse Compression Technique”. In: *International Journal of Computer Applications* 107 (Dec. 2014), pp. 24–27. DOI: [10.5120/18869-0417](https://doi.org/10.5120/18869-0417).
- [31] Donald Knuth. *Wikipedia: Barker Code*. URL: [https://en.wikipedia.org/wiki/Barker\\_code#cite\\_note-5](https://en.wikipedia.org/wiki/Barker_code#cite_note-5).

- [32] Manon Kok, Jeroen D. Hol, and Thomas B. Schön. “Indoor Positioning Using Ultrawideband and Inertial Measurements”. In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1293–1303. DOI: [10.1109/TVT.2015.2396640](https://doi.org/10.1109/TVT.2015.2396640).
- [33] Marcin Kolakowski and Vitomir Djaja-Josko. “TDOA-TWR based positioning algorithm for UWB localization system”. In: *2016 21st International Conference on Microwave, Radar and Wireless Communications (MIKON)*. 2016, pp. 1–4. DOI: [10.1109/MIKON.2016.7491981](https://doi.org/10.1109/MIKON.2016.7491981).
- [34] Omprakash Kumar and Surender Soni. “Design and Analysis of Ultra-wideband Micro Strip Patch Antenna with Notch Band Characteristics”. In: *MATEC Web of Conferences* 57 (Jan. 2016), p. 01020. DOI: [10.1051/mateconf/20165701020](https://doi.org/10.1051/mateconf/20165701020).
- [35] Mohamed Laaraiedh et al. “Comparison of Hybrid Localization Schemes using RSSI, TOA, and TDOA”. In: *17th European Wireless 2011 - Sustainable Wireless Technologies*. 2011, pp. 1–5.
- [36] Chia Lee and Chandan Chakrabarty. “Ultra Wideband Microstrip Diamond Slotted Patch Antenna with Enhanced Bandwidth”. In: *IJCNS* 4 (Jan. 2011), pp. 468–474. DOI: [10.4236/ijcns.2011.47057](https://doi.org/10.4236/ijcns.2011.47057).
- [37] Chun-Chi Lee. “An Experimental Study of the Printed-Circuit Elliptic Dipole Antenna with 1.5-16 GHz Bandwidth”. In: *Int'l J. of Communications, Network and System Sciences* 01 (Jan. 2008), pp. 295–300. DOI: [10.4236/ijcns.2008.14036](https://doi.org/10.4236/ijcns.2008.14036).
- [38] Rami Mazraani et al. “Experimental results of a combined TDOA/TOF technique for UWB based localization systems”. In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2017, pp. 1043–1048. DOI: [10.1109/ICCW.2017.7962796](https://doi.org/10.1109/ICCW.2017.7962796).
- [39] Microchip. *Microchip miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers Datasheet*. Available ONLINE: URL: <https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf>.
- [40] Karim-Mounssif Mimoune, Iness Ahriz, and Joffray Guillory. “Evaluation and Improvement of Localization Algorithms Based on UWB Pozyx System”. In: *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2019, pp. 1–5. DOI: [10.23919/SOFTCOM.2019.8903742](https://doi.org/10.23919/SOFTCOM.2019.8903742).
- [41] Stefania Monica and Federico Bergenti. “Hybrid Indoor Localization Using WiFi and UWB Technologies”. In: *Electronics* 8.3 (2019). ISSN: 2079-9292. DOI: [10.3390/electronics8030334](https://doi.org/10.3390/electronics8030334). URL: <https://www.mdpi.com/2079-9292/8/3/334>.



- [42] Chakrapani Nandyala et al. “Efficient use of circuit amp; 3D-EM simulation to optimize the automotive Bulk Current Injection (BCI) performance of Ultrasonic Sensors”. In: *2020 International Symposium on Electromagnetic Compatibility - EMC EUROPE*. 2020, pp. 1–4. DOI: [10.1109/EMCEUROPE48519.2020.9245651](https://doi.org/10.1109/EMCEUROPE48519.2020.9245651).
- [43] R. Négrier et al. “Improvement of an UWB impulse radiation source by integrating photoswitch device”. In: *2014 11th European Radar Conference*. 2014, pp. 289–292. DOI: [10.1109/EuRAD.2014.6991264](https://doi.org/10.1109/EuRAD.2014.6991264).
- [44] *OpenRTLS System Product Website*. Available at: URL: <https://www.zebra.com/>.
- [45] Panasonic. *Panasonic R-1566 FR-4 Substrate Datasheet*. Available ONLINE: URL: <https://www.pcbdirectlab.com/materialipdf/Archive/R-1566.pdf>.
- [46] Lee Chia Ping, Chandan Kumar Chakrabarty, and Rozanah Amir Khan. “Design of Ultra Wideband slotted microstrip patch antenna”. In: *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*. 2009, pp. 41–45. DOI: [10.1109/MICC.2009.5431436](https://doi.org/10.1109/MICC.2009.5431436).
- [47] *Pozyx Accurate Positioning Website*. Available at: URL: <https://www.pozyx.io/>.
- [48] *Quantitec Industrial Platform Website (IntraNav)*. Available at: URL: [Available: %20https://intranav.com/](https://intranav.com/).
- [49] Adrian Scott and Vratislav Sokol. “True Transient 3D EM/Circuit CoSimulation Using CST STUDIO SUITE”. In: *CST-Computer Simulation Technology AG* (2008), Page 7. URL: [http://www.eurointech.ru/products/CST/CST\\_MPD\\_Oct\\_2008.pdf](http://www.eurointech.ru/products/CST/CST_MPD_Oct_2008.pdf).
- [50] Sebastian Sczyslo et al. “Hybrid localization using UWB and inertial sensors”. In: *2008 IEEE International Conference on Ultra-Wideband*. Vol. 3. 2008, pp. 89–92. DOI: [10.1109/ICUWB.2008.4653423](https://doi.org/10.1109/ICUWB.2008.4653423).
- [51] *Sewio Product Website*. Available at: URL: <https://www.sewio.net/>.
- [52] Skyworks. *Skyworks schottky diode Datasheet*. Available ONLINE: URL: [https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/201-300/Surface\\_Mount\\_Schottky\\_Diodes\\_200041AG.pdf](https://www.skyworksinc.com/-/media/SkyWorks/Documents/Products/201-300/Surface_Mount_Schottky_Diodes_200041AG.pdf).
- [53] Pete Steggle and Stephan Gschwind. “The Ubisense smart space platform”. In: 2005.
- [54] Yao Tang, Jing Wang, and Changzhi Li. “Short-range indoor localization using a hybrid doppler-UWB system”. In: *2017 IEEE MTT-S International Microwave Symposium (IMS)*. 2017, pp. 1011–1014. DOI: [10.1109/MWSYM.2017.8058763](https://doi.org/10.1109/MWSYM.2017.8058763).

- [55] Trupti Telsang and Anandrao Kakade. “Ultrawideband slotted semicircular patch antenna”. In: *Microwave and Optical Technology Letters* 56 (Feb. 2014). DOI: [10.1002/mop.28102](https://doi.org/10.1002/mop.28102).
- [56] *THE EUROPEAN TABLE OF FREQUENCY ALLOCATIONS AND APPLICATIONS IN THE FREQUENCY RANGE 8.3 kHz to 3000 GHz (ECA TABLE)*. Available at: URL: <https://efis.cept.org/reports/ReportDownloader?reportid=1>.
- [57] *Time Domain Product Website, Now Humatics*. Available at: URL: [Available: %20https://timedomain.com/](https://timedomain.com/).
- [58] Alberto Toccafondi et al. “Low-power UWB transmitter for RFID transponder applications”. In: *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*. 2012, pp. 234–238. DOI: [10.1109/RFID-TA.2012.6404519](https://doi.org/10.1109/RFID-TA.2012.6404519).
- [59] *Ubisense RTLS Solutions Website*. Available at: URL: <https://ubisense.com/>.
- [60] *UWB Antennas, Taoglas*. URL: <https://www.taoglas.com/product-category/uwb-antennas/>.
- [61] Tiandong Wang et al. “Error analysis and experimental study on indoor UWB TDoA localization with reference tag”. In: *2013 19th Asia-Pacific Conference on Communications (APCC)*. 2013, pp. 505–508. DOI: [10.1109/APCC.2013.6766000](https://doi.org/10.1109/APCC.2013.6766000).
- [62] Wikipedia. *Definition and application of Moving Average*. Available ONLINE at: URL: [https://en.wikipedia.org/wiki/Moving\\_average](https://en.wikipedia.org/wiki/Moving_average).
- [63] Christian Wolff. *Radar tutorial, Barker codes*. Available at: URL: <https://www.radartutorial.eu/08.transmitters/Barker%5C%20Code.en.html>.
- [64] Henk Wymeersch, Jaime Lien, and Moe Z. Win. “Cooperative Localization in Wireless Networks”. In: *Proceedings of the IEEE* 97.2 (2009), pp. 427–450. DOI: [10.1109/JPROC.2008.2008853](https://doi.org/10.1109/JPROC.2008.2008853).
- [65] Xilinx. *Zynq Evaluation and Development Hardware user guide*. Available ONLINE on Mouser website: URL: [https://www.mouser.it/datasheet/2/690/zedboard\\_ug-846469.pdf](https://www.mouser.it/datasheet/2/690/zedboard_ug-846469.pdf).
- [66] Xilinx. *7 Series Product Selection Guide*. Available ONLINE at: URL: <https://www.xilinx.com/support/documentation/selection-guides/7-series-product-selection-guide.pdf>.
- [67] Xilinx. *Application note XAPP1017, LVDS Source Synchronous DDR Deserialization (up to 1,600 Mb/s)*. Available ONLINE at: URL: [https://www.xilinx.com/support/documentation/application\\_notes/xapp1017-lvds-ddr-deserial.pdf](https://www.xilinx.com/support/documentation/application_notes/xapp1017-lvds-ddr-deserial.pdf).

- [68] Xilinx. *PG0212 ,AXI DMA v7.1 LogiCORE IP Product Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_dma/v7\\_1/pg021\\_axi\\_dma.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf).
- [69] Xilinx. *PG049 AXI Interconnect v2.1 LogiCORE IP Product Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/axi\\_interconnect/v2\\_1/pg059-axi-interconnect.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v2_1/pg059-axi-interconnect.pdf).
- [70] Xilinx. *PG057 FIFO Generator v13.1 LogiCORE IP Product Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/fifo\\_generator/v13\\_1/pg057-fifo-generator.pdf](https://www.xilinx.com/support/documentation/ip_documentation/fifo_generator/v13_1/pg057-fifo-generator.pdf).
- [71] Xilinx. *PG065 Clocking Wizard v6.0, LogiCORE IP Producti Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/clk\\_wiz/v6\\_0/pg065-clk-wiz.pdf](https://www.xilinx.com/support/documentation/ip_documentation/clk_wiz/v6_0/pg065-clk-wiz.pdf).
- [72] Xilinx. *PG085 AXI4-Stream Infrastructure IP Suite v3.0 LogiCORE IP Product Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/axis\\_infrastructure\\_ip\\_suite/v1\\_1/pg085-axi4stream-infrastructure.pdf](https://www.xilinx.com/support/documentation/ip_documentation/axis_infrastructure_ip_suite/v1_1/pg085-axi4stream-infrastructure.pdf).
- [73] Xilinx. *UG471, 7 Series FPGAs SelectIO Resources User Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/user\\_guides/ug471\\_7Series\\_SelectIO.pdf](https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf).
- [74] Xilinx. *UG472 v1.14, 7 Series FPGAs Clocking Resources, User Guide*. Available *ONLINE* at: URL: [https://www.xilinx.com/support/documentation/user\\_guides/ug472\\_7Series\\_Clocking.pdf](https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf).
- [75] Jie Xiong and Kyle Jamieson. “Arraytrack: A fine-grained indoor location system”. In: *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. 2013, pp. 71–84.
- [76] Chouchang Yang and Huai-rong Shao. “WiFi-based indoor positioning”. In: *IEEE Communications Magazine* 53.3 (2015), pp. 150–157. DOI: [10.1109/MCOM.2015.7060497](https://doi.org/10.1109/MCOM.2015.7060497).
- [77] Faheem Zafari, Athanasios Gkelias, and Kin K. Leung. “A Survey of Indoor Localization Systems and Technologies”. In: *IEEE Communications Surveys Tutorials* 21.3 (2019), pp. 2568–2599. DOI: [10.1109/COMST.2019.2911558](https://doi.org/10.1109/COMST.2019.2911558).
- [78] *Zebra RTLS Website*. Available at: URL: <https://www.zebra.com/>.



This Ph.D. thesis has been typeset by means of the  $\text{\TeX}$ -system facilities. The typesetting engine was  $\text{\pdfL\TeX}$ . The document class was `toptesi`, by Claudio Beccari, with option `tipotesi=scudo`. This class is available in every up-to-date and complete  $\text{\TeX}$ -system installation.