

Politecnico di Torino

Master Degree in Computer Science, Embedded Systems Course



Master Degree Thesis Work

RFID implantable technology for grafted rose tracking

Supervisors:

Prof. Daniele Trincherò
Dott. Ing. Mattia Poletti

Candidate:

Stefano Bonicelli 265917

Contents

1	Introduction	1
1.1	Scenario	1
1.2	Identification Solutions	1
1.2.1	Identification Solutions Analysis	2
1.3	Identification Method Selection	4
1.3.1	How Passive RFID Works	5
2	Solution Proposed	6
2.1	The RFID Technology Choice	8
2.1.1	LF Tags	8
2.1.2	LF Reader	9
2.1.3	LF RFID Tests	10
2.1.4	UHF Tags	10
2.1.5	UHF Reader	10
2.1.6	UHF RFID Tests	12
2.2	SDR Automated Spectrum Analysis	13
2.2.1	Software Defined Radio	13
2.2.2	GNURadio	14
2.2.3	The Automated Analysis Application	14
2.3	UHF Antenna Design	17
2.3.1	OpenEMS	17
2.3.2	NEC	17
2.3.3	Simple Dipole	18
2.3.4	Loop Antenna	19
2.3.5	Vivaldi Antenna	20
2.3.6	Tag's Antenna	21
2.4	UHF Solution	23
2.5	RFID Solutions Integration Test	23
3	Custom Software Development	25
3.1	LF RFID System Software	25
3.1.1	Code Breakdown	25
3.2	UHF RFID System Software	25
3.2.1	EPC UHF Gen2 Air Interface Protocol	26
3.2.2	Software Implementation	26
3.3	Custom ST25RU3993-EVAL Board Firmware	28
3.3.1	ST25RU3993-EVAL Board Hardware Analysis	29
3.3.2	STM32CubeIDE Project	32
3.3.3	Custom Firmware	38
3.3.4	Custom Firmware Compilation and Deployment	40

4	Conclusions	41
A	NEC Simulations	i
A.1	865MHz Dipole Antenna	i
A.2	865MHz Loop Antenna	ii
B	FDTD Simulations (Octave OpenEMS)	iii
B.1	865MHz Dipole Antenna	iii
B.2	865MHz Loop Antenna	iv
B.3	865MHz Vivaldi Antenna	v

List of Figures

1.1	Principal Identification Solutions	2
1.2	Different barcode systems	2
1.3	Passive (Left) and Active (Right) RFID Tags	3
1.4	Smart Card (Left) and Biometric Recognition System (Right)	4
1.5	Antenna Near and Far Field Regions [7]	5
2.1	Trunk Structure	6
2.2	Characteristics and applications of RFID frequency ranges	7
2.3	Radio Waves Interaction When Passing From Free Space to Wet Wood (OpenEMS)	9
2.4	LF RFID Glass Capsule and Reader	9
2.5	Dimensions of the Murata tags	11
2.6	The ST25RU3993-EVAL Board	11
2.7	Suggested Test Setup	12
2.8	SDR Block Diagram and Noelec NESDR Smart	14
2.9	First GNURadio Test Block Diagram and Results	15
2.10	GNURadio Companion Channelizing system	15
2.11	GNURadio Companion Signal Detecting Path	16
2.12	Final UI of the Automated Analysis Application	16
2.13	Simulation and Analysis Data of the Dipole Antenna	18
2.14	Dipole Implementation	19
2.15	Simulation and Analysis Data of the Loop Antenna	20
2.16	Loop Antenna Implementation	21
2.17	Vivaldi Antenna Realization and Analysis	22
2.18	Cross-section of Murata MAGICSTRAP Placement Sample	22
2.19	UHF Tag Applied on the Custom Antenna	22
2.20	Murata LXTBKZMCMG-010 Dimensions	23
2.21	Tag Integration Into Branch (UHF Left, LF Right)	24
3.1	Tag State Diagram	27
3.2	Tag Internal Memory Structure	28
3.3	Tag Population Management	29
3.4	ST25RU3993-EVAL Board Block Diagram	30
3.5	ST25RU3993-EVAL Board Components	31
3.6	ST25RU3993-EVAL Board Output Power	31
3.7	MCU configuration in STM32CubeIDE	33
3.8	ST25RU3993-EVAL Board RF Path Schematics	39
3.9	ST25RU3993 IC Register table	40

Abstract

This work aims at developing a system able to identify the plants grown by a specific nursery, to distinguish them by the others. This system will be proposed in particular for the identification of the roses.

To provide a secure identification of the roses, a solution concerning the integration of a Passive RFID tag in the plant at the grafting time has been developed together with the device able to read the used tags. An accurate analysis of the industrial RFID solutions allowed us to identify two solutions: the LF Near-Field RFID tags and the UHF Far-Field ones. The first solution's main feature is the resilience to the challenging environment, the second's one is the small size that makes them more easy to embed during the grafting time.

The really small size of the UHF tags has confronted us with some challenges as the capability of transferring enough power to the tag to power it on, and the one to catch the response signal coming from the scattering system of the tag itself. To face with those challenges, custom antennas has been designed, analysed and tested.

Together with the hardware system, a simple software solution has been also developed to reach a fully working system able to identify the plants equipped with the RFID tag.

Chapter 1

Introduction

1.1 Scenario

This thesis work aims at developing a system able to identify the plants grown by a specific nursery, to distinguish them by the others, in this particular case we will focus on the roses production chain.

The identification system must be applied before the roses are sold, so it must be integrated in really young plants and must work over the years despite the growth of the plant. As this is an identification and anti-counterfeiting system, the chosen method must not be altered or deactivated without compromising the asset they are applied to.

Together with the other parameters, the financial impact of our solution must be taken into account, specially for the recurring costs presented by the need of some tagging mechanism and the information storage system that will record the identification code associated to the plants equipped with our solution.

For this project we aim at developing an handheld system to interact with the identification tag to be used in proximity of the plant to be checked, connected to a laptop that will interact with the information storage system. To fulfill this project's requirement, we have to develop the hardware and the software in a focused manner exploiting the possibility to use commercial devices together with custom solutions.

1.2 Identification Solutions

As the identification problem is a widely studied field, a careful analysis of the solution adopted to solve it in other fields can help us discovering the most suitable one to fit for our scenario. During the analysis a special focus must be applied to the practical realisation of the device implementing the chosen identification system for the scenario proposed before.

As we can read from [14], there are several solutions to the identification problem including passive and active elements associated to the asset to be identified and tracked, together with systems able to use some elements of the asset as identification features; of course some methods are more specialised than other can be more easily ported from scenario to scenario. Figure 1.1 taken from [14] sums up the principal identification solutions on the market nowadays that offer a wide variety of devices as everyone of them has particular application fields where different conditions drive the development of specialised solutions.

Our analysis will bring to the choice of the most suitable identification method to be applied to the plants to be identified and will be followed by an explanation of how that system works and what are the main features that we can exploit to reach a reliable system fulfilling the requirements of our project.

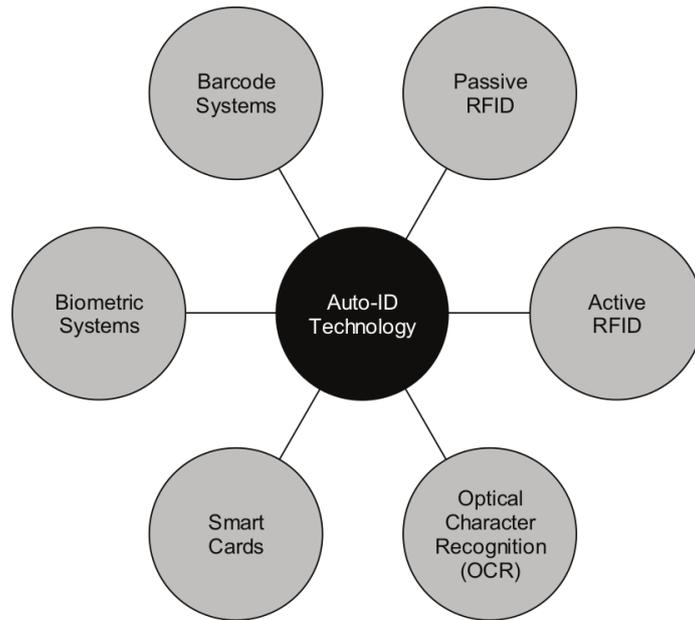


Figure 1.1: Principal Identification Solutions



Figure 1.2: Different barcode systems

1.2.1 Identification Solutions Analysis

Barcode System

The barcode system is a widely used way to identify goods in stores and markets, it's a cheap and simple solution that allow the correct identification of the assets from the producer to the consumer representing all the needed information in a single code easily readable by an electronic device.

As we can read in [14] the barcode contains:

- An header
- A check digit
- The manufacturer's identification number
- The item identifier

Its use in stores and markets made this identification solution very cheap and easy to use, we can find barcode reader able to fit very different scenarios and tagging system very different between them; an example can be seen in figure 1.2



Figure 1.3: Passive (Left) and Active (Right) RFID Tags

All this elements seems to fit our goal, but an important drawback makes barcodes not suitable for our project: they must be printed or stuck on the asset that they identify guaranteeing the clear line of sight between reader and tag; as mentioned before, we deal with growing plants, so it's not feasible to do so, in particular to satisfy the anti-counterfeiting constraint of the work that has been assigned us.

Passive RFID

The Passive RFID solution allows us to identify the assets in a similar way of the one given by the barcodes, extending it with the possibility to store some bytes of data directly in the passive tag. The main difference between RFID and Barcodes is the use of radio waves as reading medium instead the direct observation of the code printed on the tag, this feature removes the need of a line of sight between reader and tag and can extend the reading capabilities to few meters. This identification method offers a wide range of systems that can be analysed more deeply to fit them to our scenario.

Active RFID

The Active RFID systems has great capabilities that start from the Passive RFID to more and more sophisticated ones, the great difference between them and the Passive ones is the need of a power supply system as they don't scavenge the required energy from the EM fields of the reader device. The huge drawback of the power supply system can be an obstacle that makes them unfeasible for our project.

OCR

The OCR identification solution is taking more and more space in the identification market, we can think to facial recognition and so on; the constant demand of computational power for the mobile market caused the availability on the market of small and cheap devices able to implement complex vision algorithms in tiny mobile devices (we can think to the RaspberryPi [10] or NVIDIA Jetson [8] platforms).

Unfortunately we have again to take in mind that we are dealing with growing plants that vary their shape quite quickly, so this method can also be excluded from the feasible ones.



Figure 1.4: Smart Card (Left) and Biometric Recognition System (Right)

Smart Cards

The Smart Cards system is a widely used identification methods for people, small circuits can be used and different kind of information can be stored in the smart card itself; surely this is an interesting identification system, but again not so feasible for our scenario due to the need of electrical contacts to power the card and interact with the device.

A similar technology is used in the telecommunication systems to store the identity of a user through the SIM Card.

Biometric Systems

The Biometric Identification Systems are technologies widely applied to humans and now taking even more space in the identification market since, as for the OCR, it doesn't needs tags or devices applied to the assets to be identified. For those reasons it has been demonstrated to be really useful for the mobile market but not so applied to fields different from the human identification. Surely it is an interesting methodology that can be studied and exploited for the plants world but not in a fast an cheap way as the one that we need for our work.

1.3 Identification Method Selection

After the above analysis, the most feasible identification method to be applied to the project's scenario has been found to be the Passive RFID one, as it can provide the requested tagging mechanism without involving fixed visual features, electrical contacts or power supply.

This solution definitely puts us in front of challenges not so trivial to be solved as the one of the tag placement, such in a way to be able to deal with counterfeiting attempts and the constantly growing process of the plant itself.

As the RFID solution uses radio waves, a very important element of the system will be the antenna system both for the reader and for the tag. As the reader antenna can be almost freely designed, the tag's one will be more restricted due to the challenging environment it will be placed in. Surely the preferred tags will be the ones with an integrated antenna in the smallest shape possible.

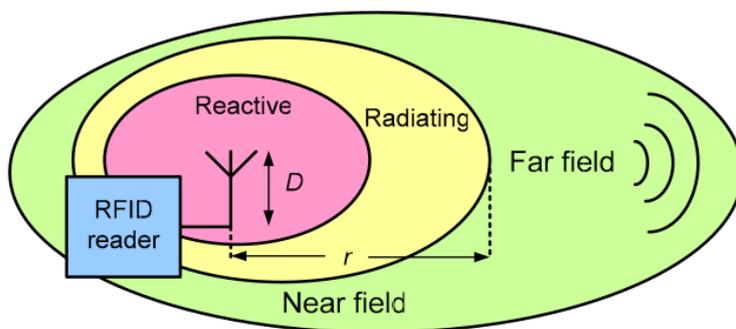


Figure 1.5: Antenna Near and Far Field Regions [7]

1.3.1 How Passive RFID Works

Without a power supply of their own, passive RFID tags depend upon the electromagnetic field of the reader that turn into power thanks to the antenna coupling. The coupled energy is rectified and the voltage multiplied to power up internal circuits. A multi-stage Greinacher half-wave rectifier or a derivative is commonly used for this purpose [3].

Two different coupling techniques, near and far fields, are used by passive tags.

Near-Field Coupling

The Near-Field coupling is performed in the region of the reader antenna where the electric and magnetic fields follow a reactive nature, to grab power from the EM emissions of the reader's antenna, one of the two field must be used. To exploit the electric field a dipole can be used, to exploit the magnetic one, a coil is more suitable [3].

To interact with the reader, the tag uses the back-scattering process performed through the load modulation mechanism consisting in a variation of the load connected to the end-points of the coil or dipole [3].

The Near-Field solution is more suitable for small distance communication in presence of water or metals [7] due to the reactive nature of the fields involved in the communication system.

Far-Field Coupling

The EM field in the far-field region is radiative in nature so the energy that reaches the tag's antenna is partially reflected for the impedance mismatch between antenna and load, modulating this mismatch a communication can be established between reader and tag [3];

In the far field region the signal strength decrease of $1/r^2$, this makes it the best solution for long distance communication as in the near-field the fields strength decrease in the order of $1/r^6$ [3].

The Far-Field solution is more subject to changes of materials and to reflecting element in the interrogation area [7].

Chapter 2

Solution Proposed

To implement the Passive RFID identification system, we decided to exploit the grafting process that is put into practice in the nursery to obtain more robust plants. Grafting is a technique whereby tissues of plants are joined so as to continue their growth together.

This choice has been taken in order to put the identification tag directly inside the plant trunk, this method has been chosen to reach the highest security against possible counterfeiting or tag removal attempts.

As we can see in figure 2.1 taken from [1], the plant trunk is divided into different parts and everyone of those parts has a special function for the life of the plant. The part that allow the grafting practice is the cambium cell layer, this structure is the place where the new cells are generated making the plant grow.

Other parts of the trunk allow the lymph to flow from roots to foliage or vice versa but a special focus needs the heartwood: this part is the one that can be exploited for our work as it is made of dead cells and there isn't any kind of life activity in it, so we can place here our RFID tag during the grafting process.

As we can only affect the heartwood part of the plant, the small section of the young trunk together with the abundant presence of water and the constantly growing plant must guide the choice of the RFID tags.

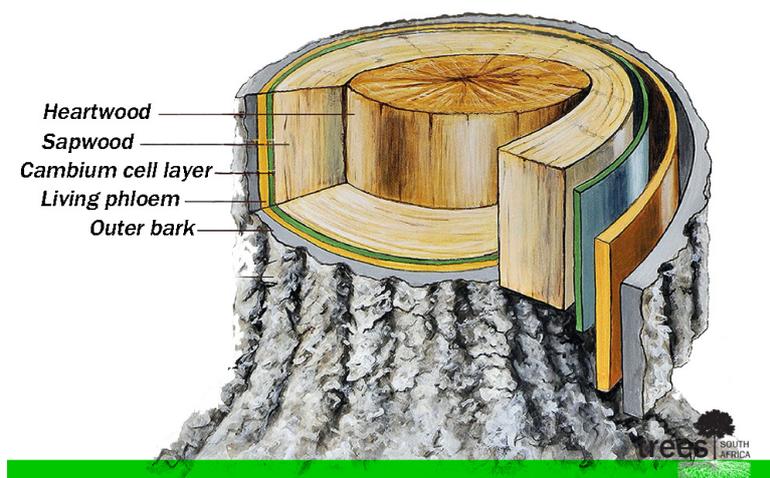


Figure 2.1: Trunk Structure

Frequency band	System characteristics	Example applications
Low (LF) 100-500 kHz (typically 125-134 KHz worldwide)	Short read range (to 18 in.) Low reading speed Relatively inexpensive Can read through liquids Works well near metal	Access control Animal identification Beer keg tracking Inventory control Automobile key/ anti-theft systems
High (HF) (typically 13.56 MHz)	13.56 MHz frequency accepted worldwide Short to medium read range (3-10 ft.) Medium reading speed Can read through liquids/works well in moist environment Does not work well near metal Moderate expense	Access control Smart cards Electronic article surveillance Library book tracking Pallet/container tracking Airline baggage tracking Apparel/laundry item tracking
Ultra High (UHF) 400-1000 MHz (typically 850-950 MHz)	Long read range (10-30 ft.) High reading speed Reduced likelihood of signal collision Difficulty reading through liquids Does not work well in moist environments Experiences interference from metals Relatively expensive	Item management Supply chain management
Microwave 2.4-6.0 GHz (typically 2.45 or 5.8 GHz)	Medium read range (10+ feet) Similar characteristics to UHF tags, but with faster read rates	Railroad car monitoring Toll collection systems

Figure 2.2: Characteristics and applications of RFID frequency ranges

2.1 The RFID Technology Choice

As we can see in the table of figure 2.2 taken from [14], there are different technologies that implement the passive RFID system, in this way different parts of the electromagnetic spectrum can be used to achieve different characteristics of the radio link between tag and reader.

The analysis presented in [7] show how the frequency used affects the link between tag and reader and how our choice will be guided by the environment around the tag. As the RFID tag will be put inside the plant trunk, we have to carefully analyse the resilience of the chosen solution to the water the plant will bring in its tissues surrounding the tag. Unfortunately reading the table of figure 2.2 we can see how the resilience to water is inversely proportional to the growth of the used frequency, this feature forces us to search for a good compromise between radio link reliability and tag sizes.

On an other hand, our purpose is to develop a device to be used in proximity of the plant to be identified, so the RFID tag will stand almost always in the near-field region of the reader antenna, this condition may help us to communicate between the plant tissues without facing reflections of power due to the difference of impedance found between the free space and the wet wood.

To better understand the complexity of the scenario we are facing with, we made an FDTD Analysis thanks to OpenEMS (More information about OpenEMS in the UHF Antenna Analysis Chapter) that allowed us to figure how the electromagnetic interaction of the radio waves is complex when interacting with the wood material. To analyse the interaction of the radio waves with different materials a very simple simulation has been used, a sources emits radio waves and a cylinder made of custom material is placed in the space near the radiation point. This analysis is as simple as useful to perform quick tests over different materials just changing the electromagnetic parameters in the simulation listing.

According to [11] the wet wood has a dielectric constant that stand in a range between 10 and 30 while having a magnetic permeability similar to the free space.

From the electromagnetic parameters seen before, we can see how the transmitted signal between air and wood is very weak, according to the following equation that allow us to compute the transmission coefficient, only around 36% of the incoming signal will be transmitted in the wood. To make our calculations we used $n_1 = 377\Omega$ and $n_2 = 84\Omega$ as materials impedance.

$$t = \frac{2 \cdot n_2}{n_1 + n_2} \quad (2.1)$$

Putting the correct parameters to simulate the wood in the OpenEMS environment, we obtained the result shown in figure 2.3 that confirm the unability of the radio waves to establish a reliable radio link between tag and reader in a Far-Field RFID system.

2.1.1 LF Tags

From what we have seen before and what can be read from [14] and [7], the most suitable tags to be applied to our work are the LF and HF ones that present a good resilience to water and moisture as the coupling between tag's and reader's antennas is always performed in the near-field region, so not facing with the reflection problem analysed before. This feature makes more interesting the magnetic field generated by the readers antenna despite to the radiation performances, for that solenoids are mainly used in this field as coupling systems.

Remembering the electromagnetic parameters seen before, we can say a special focus must be accorded to the magnetic field to establish a reliable link between tag and reader, so a proper antenna system has to be searched to exploit this field in the chosen RFID system, not only for the reader device but also for the tag.

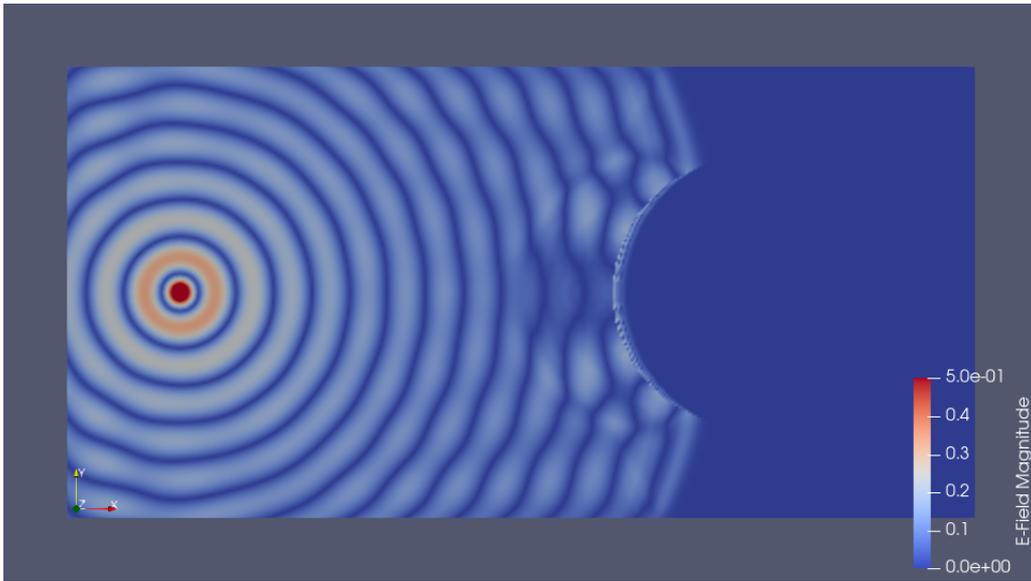


Figure 2.3: Radio Waves Interaction When Passing From Free Space to Wet Wood (OpenEMS)

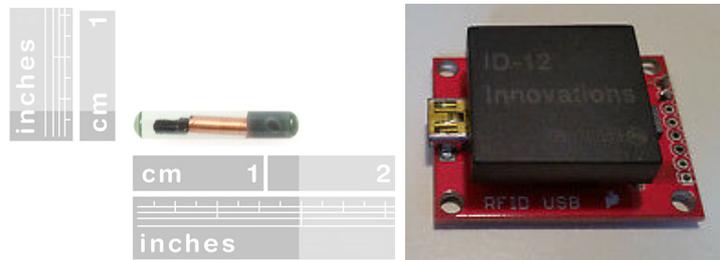


Figure 2.4: LF RFID Glass Capsule and Reader

After having searched for tags as small to be inserted into the young plants of roses in the LF and HF range, the best choice has proven to be the tags used for animal identification. Those tags work in the LF band, in particular at 125kHz and include in a very small package the tag and the antenna.

As mentioned before the best antenna solution is the use of a solenoid, this statement is confirmed by the presence of a copper wire coil into the glass capsule able to perform the needed magnetic coupling.

To develop some tests, we bought the LF RFID glass capsule from Sparkfun that can be seen in figure 2.4.

2.1.2 LF Reader

The chosen tags are read-only and store a non programmable 32-bit unique ID that can be read by the reader.

In the Sparkfun website is written that those tags have been tested using a ID-12LA reader and reached a reading distance of 10mm. As those performances are so suitable to our project and as the ID-12LA reader uses a coil antenna, we decided to use it for our tests. To easily setup a testing system we purchased one ID-12LA RFID reader with the Sparkfun USB RFID board as shown in figure 2.4, this setup allowed us to read the tag ID using a serial port on a PC and a serial monitor connected to it.

2.1.3 LF RFID Tests

First Test

As first test we performed a reading of the capsule tag in open air to verify the correct functioning of the system.

To perform this test the reader has been plugged to an host PC through USB and a serial monitor has been opened. As the tag reached some millimeters from the reader, its ID appeared on the serial monitor as confirmation of the successful reading process.

This test confirmed the expected performances presented by the Sparkfun website.

Complex Environment Test

To test the ability of the RFID tag to fill the requirements imposed by the challenging scenario we are facing with, a second test has been performed placing around the tag some paper to reach a thickness of 5 millimeters. with the tag in those conditions a new reading process has been performed with the Sparkfun ID-12 reader and it ended successfully. The same process has been used to test the ability of the RFID system to face with the high amount of water the plant will bring around the tag, wetting the paper gradually.

The results obtained by this simple test confirmed the analysis performed before, as no changes were observed in the link capabilities with the increasing of the water presence around the tag.

2.1.4 UHF Tags

To try to reach dimensions even smaller than those of the LF capsules, we decided to try the UHF tags commonly used in PCB tracking. Those tags can have really small dimensions, in the order of few millimeters and operates on the 868MHz band.

As we can remember from the previous analysis, those tags are not so suitable for water reach environment and must be mainly used in a far field configuration but as stated before the interaction between the reader and the tag will be performed bringing the reader antenna really close to the plant to be checked so the tag will be in the Near-Field region of it when performing the reading process.

As our scenario makes complex tag's antenna system unfeasible, our focus will be concentrated to exploit the functionality brought by the inductive coupling circuit that those type of tags has inside, commonly used to couple the IC to the PCB printed antenna.

To test the UHF solution we purchased two different Murata tags (LXMS21NCNH-147 and LXMSJZNCMF-198) that has a really small footprint, as we can see in figure 2.5 taken from the datasheets of the two components and implement a similar identification system seen for the LF tags that uses a builtin ID.

Differently from the previous seen tags, one of the Murata tags has a small memory to store custom data, very useful feature that can be used to expand the capabilities of our identification system.

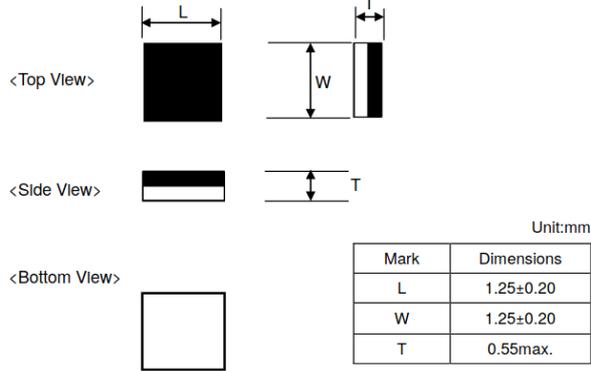
2.1.5 UHF Reader

The Murata tags complies ISO18000-63/EPC global Gen2(v1.2.0) so different UHF RFID readers can interact with them, in particular the family of RAIN RFID systems.

As reader we chose the ST25RU3993-EVAL board for the possibility to develop custom software and in future to project and implement a custom board equipped with the ST25RU3993

LXMSJZNCMF-198

[Dimension]



LXMS21NCNH-147

[Dimension]

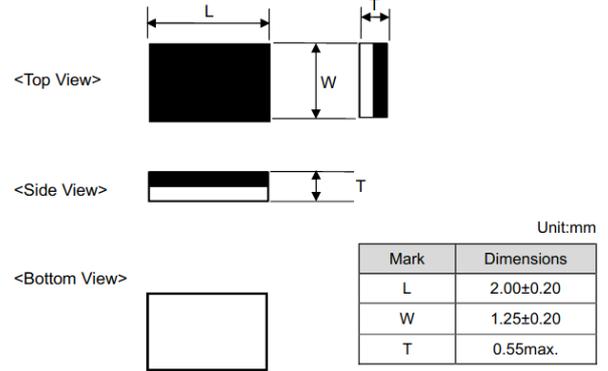


Figure 2.5: Dimensions of the Murata tags



Figure 2.6: The ST25RU3993-EVAL Board

IC. The ST25RU3993 IC is a RAIN RFID single chip reader EPC Class1 Gen2 compatible, so fully compatible with the chosen tags and really well controllable by an external microcontroller.

The choice of the ST25RU3993-EVAL board allowed us to quickly start some tests thanks to the ST application that uses the evaluation board as a fully working USB RFID reader to be plugged to a PC through USB and the external antenna that can allow us to change it and develop a custom one more more appropriate to our project.

An image of the ST25RU3993-EVAL board can be found in figure 2.6 taken from the ST website.

The main features that made the ST25RU3993-EVAL board really fit for our work are:

- Fully programmable STM32L476RG MCU
- 28dBm external power amplifier
- SB / UART bridge
- USB power supply
- 11 MCU connected LEDs

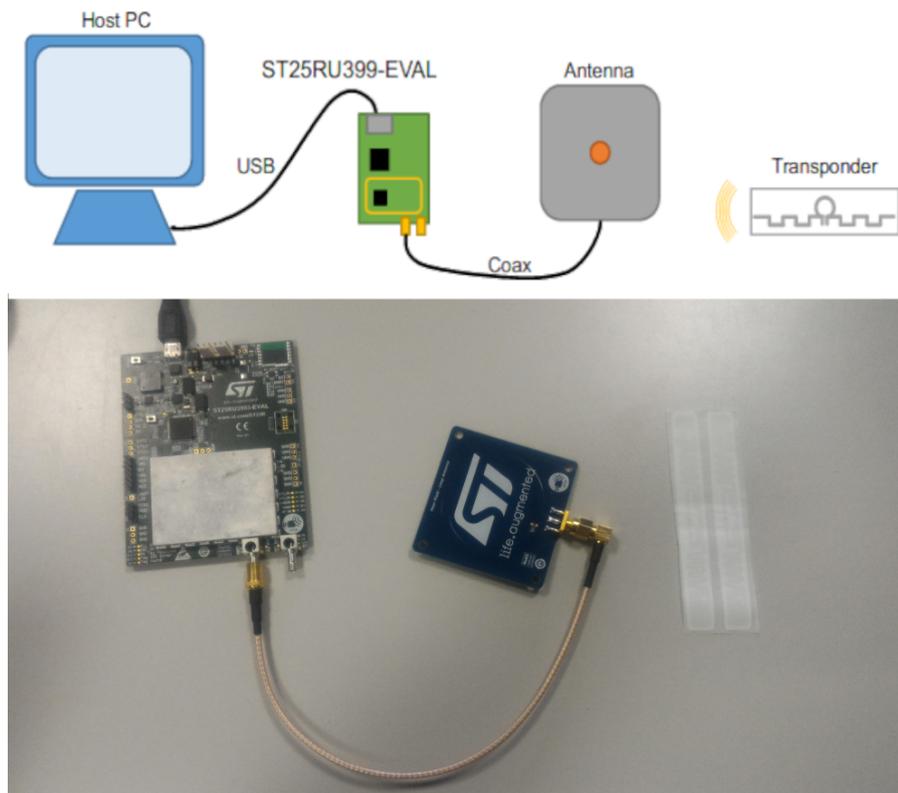


Figure 2.7: Suggested Test Setup

All those features allowed us to develop custom application as for the builtin microcontroller as for an external computational system.

2.1.6 UHF RFID Tests

As mentioned before, some tests has been done through the ST Windows application that allow the user to set all the parameters of the ST25RU3993 IC together with the tuning of the external circuitry as the antenna selection and the power amplifier gain.

First Test

Firstly the suggested test setup shown in figure 2.7 has been prepared and the correct RFID operations has been tested. Performing some tests tuning the reader's parameters, the communication between reader and test tags performed very well with a maximum reached distance between reader antenna and tags up to 1m using an output power of 28dBm from the external power amplifier. Those results has been obtained using the Alien Squiggle tags that consists of a micro-tag applied on a dipole antenna printed on a paper foil. A different behaviour presented the test performed over the Murata tags that we purchased for the integration into the roses plants: the communication was successful but only with contacts between the ST antenna and the tags ICs.

As reader antenna used the AN5308 by ST, a near field loop antenna that comes with the evaluation board and is composed by a circular microstrip transmission line with a 50Ω termination. This radiating element is not a proper antenna, but is a device that generates a magnetic field around the microstrip path but uses the properties of the transmission line to reach a far field radiation pattern with gain lower than -25 dB in every point.

This test doesn't allow us to understand if the coupling between antenna and reader can be

performed away from the near field of the antenna, to do that we need to perform some tests with other kind of antennas trying to exploit the electrical or magnetic nature of the reader antenna used.

Board Performance Test and Configuration

Firstly after the first general test, the performance of the evaluation board have been tested in order to check the correct configuration.

With the use of the Agilent E4402B spectrum analyser and a 20dB attenuation network the output power and the tuning system has been checked and performed as expected.

Through this setup, a final configuration of the board has been performed in order to have a reference system to be used during the next test that will be performed to analyse the different antenna performances. The transmission power has been set to 28dBm, the used frequency has been tuned on 865MHz; all the EPC Gen2 parameters has been left as default. For the antenna tests we disabled the frequency hopping feature of the ST25RU3993 device to finely tune the antennas' resonance to the used frequency.

2.2 SDR Automated Spectrum Analysis

To deeply analyse the functioning of the UHF system in different conditions and with different reader's antennas, we started developing an analysis environment based on an SDR and GNU-Radio allowing us to automatically detect and store the data concerning the radio emissions detected during our tests.

2.2.1 Software Defined Radio

From [12] we can read that radio components such as modulators, demodulators and tuners are traditionally implemented in analogue hardware components. The advent of modern computing and analogue to digital converters allows most of these traditionally hardware based components to be implemented in software instead. Hence, the term software defined radio. This enables easy signal processing and thus cheap wide band scanner radios to be produced.

The implementation of the SDR is derived from the DVB-T receiver developed to enable a simple pc to receive the TV signals of the DVB-T TV system.

The block diagram of an sdr can be found in figure 2.8 where the different functional elements can be seen to have a clear idea of how those devices work.

From the SDR device we will receive the row quadrature data coming from the ADCs connected to the quadrature mixer inside the device. With some configuration we can use the internal RF amplifier placed between the antenna and the mixer to capture the weakest signals. In this way all the signal analysis process is left to the host PC the SDR is connected to, making those devices very versatile and usefull in a wide range of field.

The main features of an SDR are the following:

- Bandwidth
- Range
- Sensitivity

The *Bandwidth* is derived by the sampling ratio of the ADCs connected to the quadrature mixer, the *Range* is based on the range of the local oscillator itself and the *Sensitivity* is derived by the LNA capabilities and the SNR obtained by the device itself.

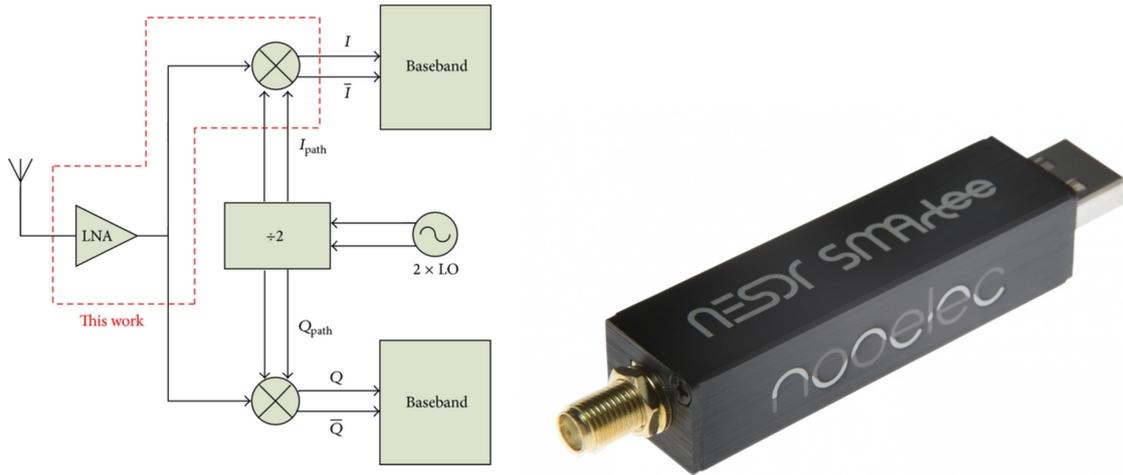


Figure 2.8: SDR Block Diagram and Noelec NESDR Smart

In our project we used a Noelec NESDR Smart that can be seen in figure 2.8, this SDR is able to span between 25MHz and 1.7GHz with a 2MHz window, so can be used to perform the required analysis on the 865MHz band as required by our project.

Other SDR can be found with a wide range of differences concerning the bandwidth, the range and consequently the price.

2.2.2 GNURadio

The SDR can be used with a lot of different software such as GQRX or SDRAngel, but our purpose is to develop a custom application able to automatically analyse and store the results of our tests, to do so GNURadio has been used together with the GNURadio Companion tool.

GNU Radio is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in hobbyist, academic and commercial environments to support both wireless communications research and real-world radio systems. GNU Radio is licensed under the GNU General Public License (GPL) version 3 or later. All of the code is copyright of the Free Software Foundation. [5]

With the GNURadio Companion tool we can develop complex application using a block interface that allow us to connect different functional block to obtain a signal path able to perform the desired operations. Using as source an SDR the GNURadio Companion tool enables us to develop the application we were looking for.

2.2.3 The Automated Analysis Application

The first step to develop the application that will perform the automated analysis during our tests is the correct connection of the SDR, so we started from the OsmoCom source block of GNURadio Companion. To test the correct connection of the SDR, the output data of the OsmoCom block have been routed directly to a spectrum scope and the SDR has been tuned on the commercial radio band to have a test signal to be seen on the spectrum. The block diagram used for this test together with the results can be seen in figure 2.9

Once the SDR has been tested, the main part of the data path can be developed, in particular the multi-frequency ability that can be reached with different local oscillator feeding some

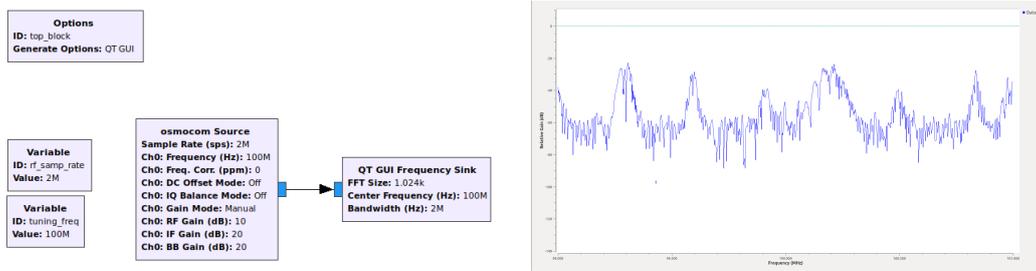


Figure 2.9: First GNURadio Test Block Diagram and Results

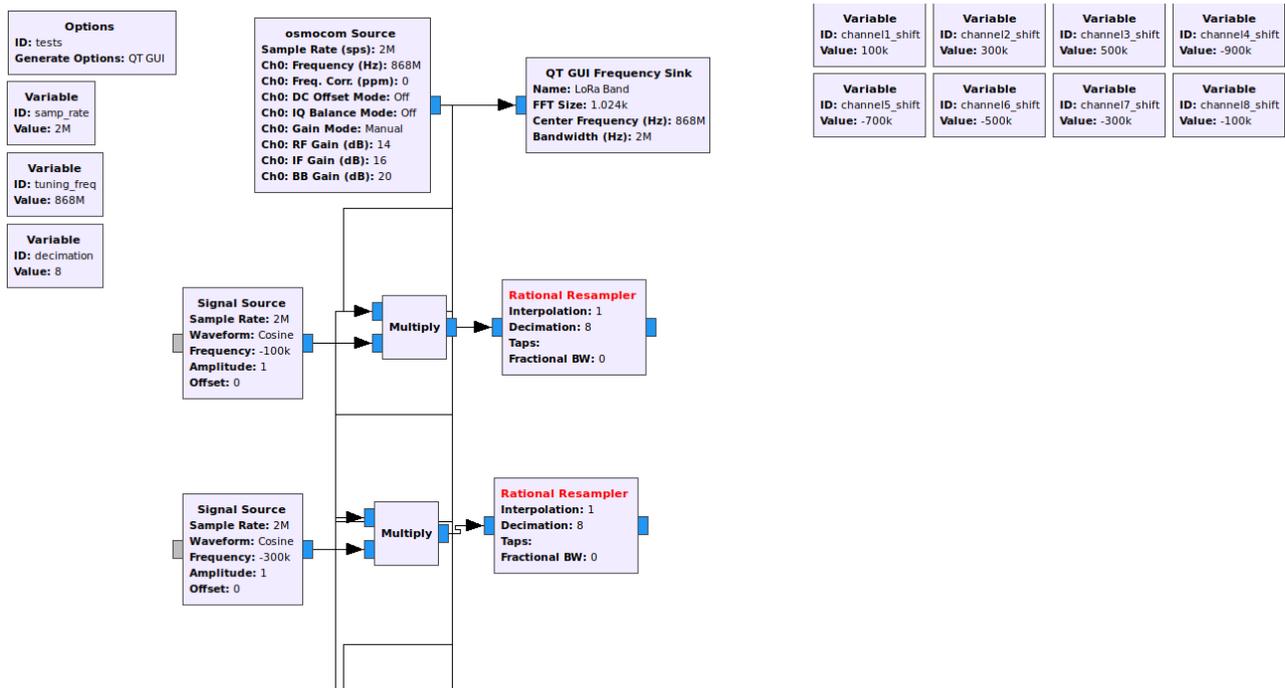


Figure 2.10: GNURadio Companion Channelizing system

multipliers; in this way a frequency shift is performed and the elaboration block can remain the same for every channel we need to implement. In figure 2.10 we can see the practical implementation of the channelizing system able to monitor different frequencies at the same time, remaining in the 2MHz band of the SDR data.

Once the signal has been successfully channelized, a detecting path has been developed to save the signal waveform on file when some radio signals above a chosen threshold are detected in the channel, the detecting blocks chain can be seen in figure 2.11.

The resulting application has demonstrated to be very useful when debugging the UHF RFID communication system as the saved waveform can be further analysed with tools as matlab or GNU Octave.

In figure 2.12 can be seen the final user interface of the application with different scopes associated to different channels to be monitored; every scope has its own parameters that can be tuned individually if needed.

This application has been used also for monitoring the LPD band or the LoRa signals demonstrating its great versatility thanks to its channelizing system easily adaptable to the different needs.

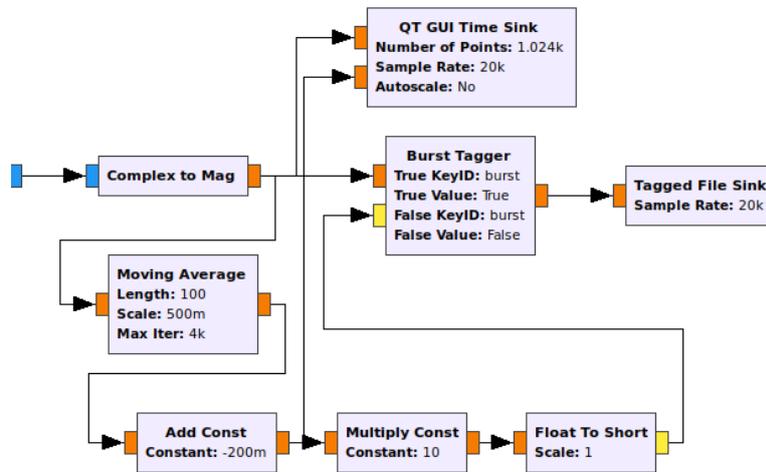


Figure 2.11: GNURadio Companion Signal Detecting Path

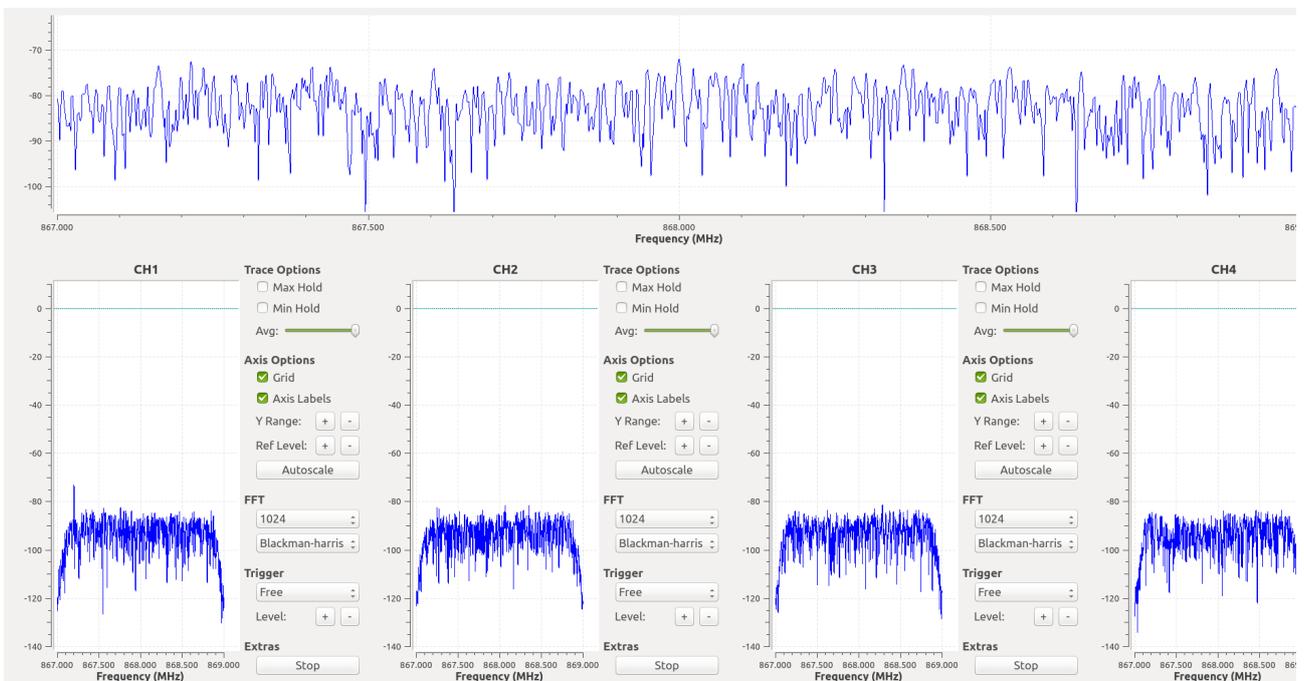


Figure 2.12: Final UI of the Automated Analysis Application

2.3 UHF Antenna Design

As the performance of the board were fully tested to work as expected and the test environment ready, the focus of the project was put on the antenna system to be used with the evaluation board in order to obtain an interrogation field able to guarantee the communication between reader and tags at distances suitable for our scenario.

As we saw before, the near field radiator used for the first test showed very weak performances with the Murata tags, so our work will be focused on some far field antennas that will allow us to test different approaches to the problem exploiting their features.

For our tests three different antenna design has been chosen based on some far-field and near-field analysis that can be found together with the obtained results in appendix to this document. The antennas' near-field performances has been analysed with the OpenEMS GNU Octave library, while the far-field ones has been analysed through the NEC algorithm thanks to the software `xnec2c`.

2.3.1 OpenEMS

OpenEMS is a free and open electromagnetic field solver using the FDTD method. Matlab or Octave are used as an easy and flexible scripting interface. [9]

It features:

- Fully 3D Cartesian and cylindrical coordinates graded mesh.
- Multi-threading, SIMD (SSE) and MPI support for high speed FDTD.

The FDTD method solves Maxwell's equations on a mesh and computes E and H at grid points spaced $\Delta x, \Delta y, \text{ and } \Delta z$ apart, with E and H interlaced in all three spatial dimensions. FDTD includes the effects of scattering, transmission, reflection, absorption, etc. FDTD is a time-domain solution, but frequency analysis is also possible through the use of the Fast Fourier Transform (FFT) and the Discrete Fourier Transform (DFT). [4]

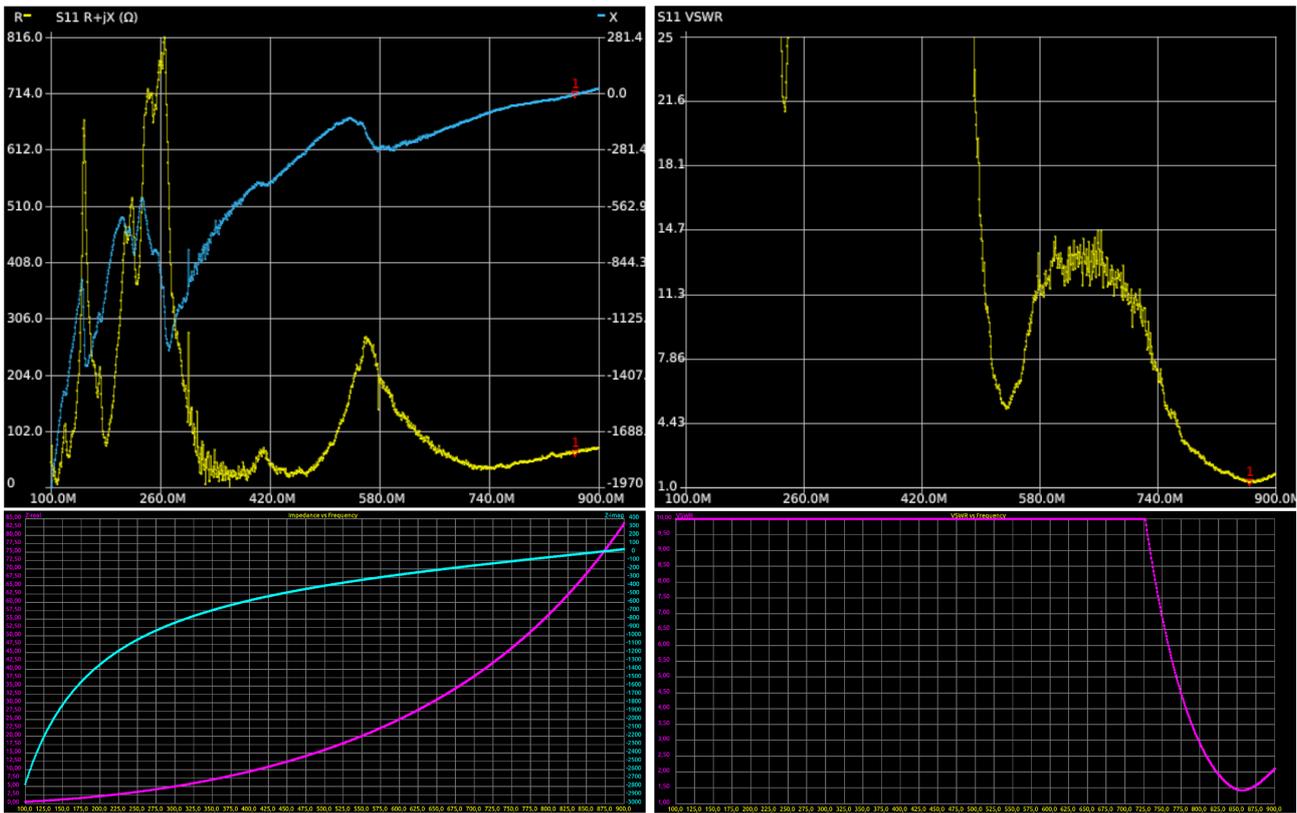
The OpenEMS solver has been chosen because the simplicity of the geometry used in our project that doesn't involve the need of a commercial tool.

This software allowed us to rapidly analyse the projected antennas with the listings that can be found in appendix showing the near-field capabilities of the antennas for the magnetic and electric fields.

2.3.2 NEC

The software Numerical Electromagnetics Code(NEC-2) has been developed in the 1970s in the Lawrence Livermore Laboratory in Livermore, California. It is based on a numerical solution of electromagnetic field integrals for thin, perfectly conducting wire segments using the Method of Moments (MoM). Such segments can be freely arranged in three dimensional space and excited in different ways. To calculate the electromagnetic properties of antennas (e.g. input impedance, current distribution or radiation pattern)there are different analysis functions available. The program has later been released as Public Domainand is available today in many different versions for almost all operating systems and CPU platforms. [13]

The NEC solver has been chosen for its accuracy in the wire antenna simulation performed through the use of electric-field integral equation and magnetic-field integral equation together to reach the maximum accuracy possible in the simulation of antennas composed by wires and surfaces [2].



Marker 1	
Frequency:	865.114 MHz
Impedance:	62.8 -j6.46 Ω
Series L:	-1.1883 nH
Series C:	28.482 pF
Parallel R:	63.467 Ω
Parallel X:	298.13 fF
VSWR:	1.291
Return loss:	-17.930 dB
Quality factor:	0.103
S11 Phase:	-23.50°
S21 Gain:	-36.063 dB
S21 Phase:	-19.82°

Figure 2.13: Simulation and Analysis Data of the Dipole Antenna

In our work the NEC solver has been used to complete the antenna analysis with the far-field behaviour together with their electrical characteristics.

2.3.3 Simple Dipole

The first antenna we designed to perform some tests was a simple dipole; this solution allowed us to check the different behaviour between the near field radiator made by ST and an antenna allowing a good far field propagation.

Antenna Design

The dipole has been designed using the NEC algorithm in order to simulate and finely tune the antenna to the frequency used for the tests: 865MHz. The analysis performed showed that a dipole resonating at 865MHz made of 1mm wire is long 164mm as we can see from the results showed in figure 2.13.

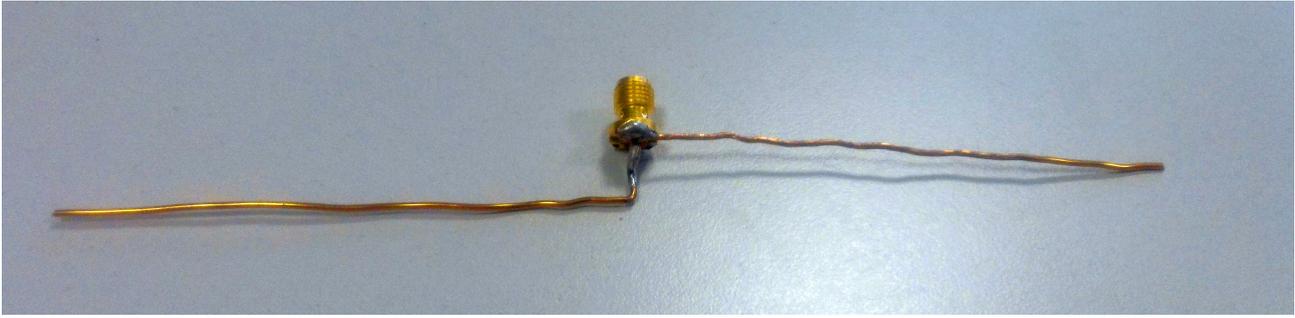


Figure 2.14: Dipole Implementation

Antenna Implementation

The dipole has been implemented through a 1mm copper wire soldered on an SMA connector, this solution matched quite well the specifications as we can see from the results showed in figure 2.13 obtained by the analysis of the dipole through a Nano VNA.

In figure 2.14 there is a picture of the dipole implementation we obtained.

Antenna Performance

The dipole performance were really interesting when applied to the test tags as the communication distance reached a maximum of 2m, but when the communication with the Murata tags was tested the results were similar to the ST antenna ones: the tags were readable only if in contact with the middle part of the antenna as here the current flowing into the dipole reaches its maximum and makes a strongest magnetic field.

2.3.4 Loop Antenna

To analyse the magnetic field interaction with the Murata tags, a loop antenna has been designed and implemented; this solution's results are really important as the tag's inductive coupling circuit can interact better with the loop antenna than with the simple dipole.

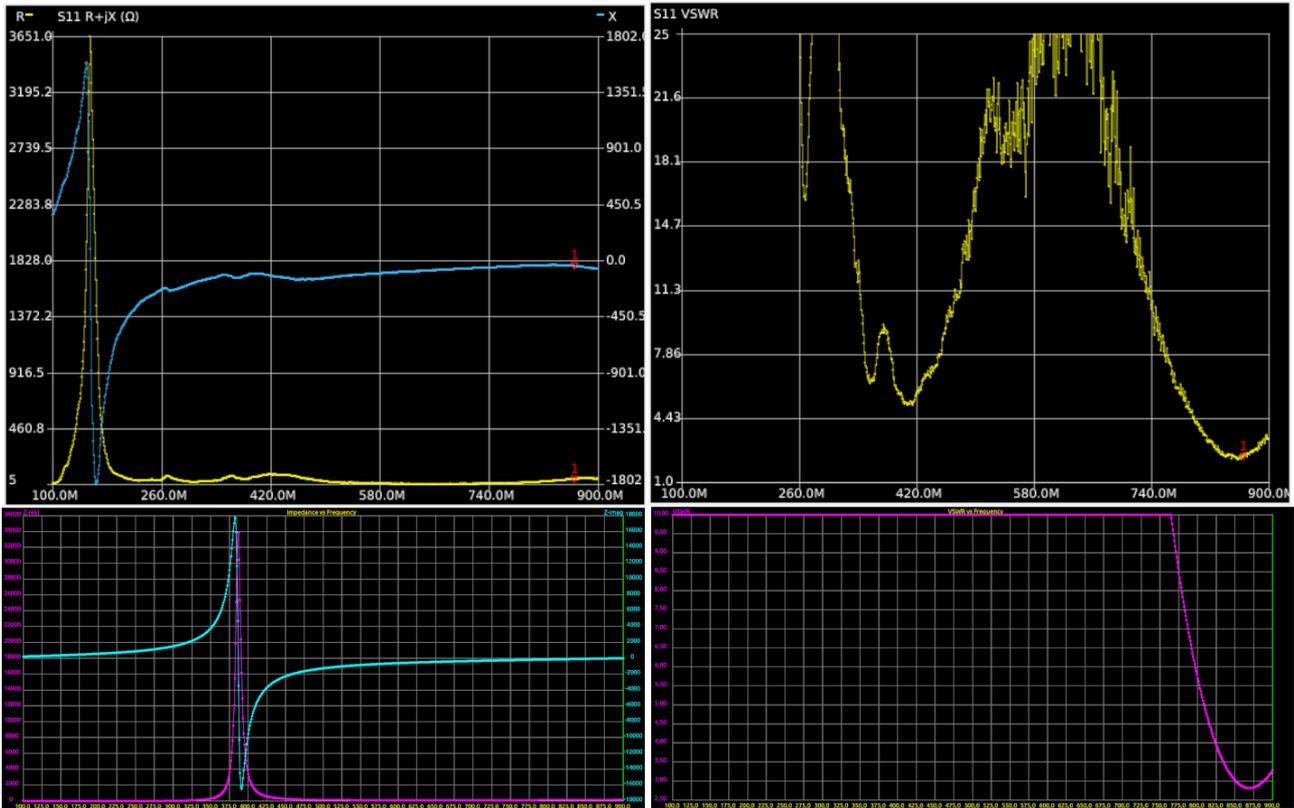
The near-field analysis performed over the model of the loop antenna showed how it can generate a stronger magnetic field that can be exploited to establish the communication link through the moisture surrounding the RFID tag.

Antenna Design

As before the antenna has been designed using the NEC simulation to tune it to the desired frequency. The simulation results are reported in figures 2.15 where we can see how a loop antenna with a 58mm radius has the correct resonating frequency around 865MHz as needed by our project.

Antenna Implementation

The loop antenna has been implemented through a 1mm brass wire fixed on an sma connector, as can be seen in figure 2.16. The analysis data shown in figure 2.15 show how the antenna implementation quietly matches the results obtained by the NEC simulation keeping good impedance properties.



Marker 1	
Frequency:	865.114 MHz
Impedance:	49.4 -j44.1 Ω
Series L:	-8.107 nH
Series C:	4.1748 pF
Parallel R:	88.709 Ω
Parallel X:	1.8501 pF
VSWR:	2.363
Return loss:	-7.844 dB
Quality factor:	0.892
S11 Phase:	-66.87°
S21 Gain:	-34.673 dB
S21 Phase:	-21.40°

Figure 2.15: Simulation and Analysis Data of the Loop Antenna

Antenna Performances

The loop antenna performed as the dipole: the Murata tags can be red only when in contact with the antenna, while the test tags can be red at a maximum distance of 2m, confirming the far-field results of the analysis but not gaining the results wished on the inductive coupling circuit of the Murata Tag.

2.3.5 Vivaldi Antenna

The test performed till now showed how the inductive coupling circuit isn't able to interact with the field generated by the antennas we developed, the only way to read the tags EPC was the inductive coupling between the IC and the antenna wire through the contact between them.

To analyse the interaction between the murata tag and a much stronger electric field we implemented a Vivaldi antenna, this antenna has really high performances in the far field region as an important electric field component in the reactive region between the two branches of the radiator.



Figure 2.16: Loop Antenna Implementation

Antenna Implementation

The Vivaldi antenna has been implemented strongly resorting to the real time analysis performed by the Nano VNA, this method has been chosen because of the antenna geometry that made unfeasible a fast NEC simulation as it's an antenna composed by a surface element and no more wires as the previous ones. The antenna geometry has been implemented on a really thin aluminium foil glued to a paper substrate that guarantee flexibility to the structure leaving us free to easily modify the shape. In figure 2.17 can be seen the antenna implementation together with the results of the analysis performed through the Nano VNA.

Antenna Performances

The performance obtained by this antenna were really interesting as it can read the test tags up to 10m, confirming the expected radiation parameters.

Unfortunately the Murata tags showed the same behaviour shown in the other tests performed with the other antennas, no interaction with the strong electric field was seen as the tag's reading process succeeded only when the tag IC was effectively in contact with the feeding system of the antenna.

2.3.6 Tag's Antenna

As the test with the different reader's antennas showed a quite identical behaviour with the Murata tags while demonstrating huge differences with the test tags, it means that the inductive coupling circuit interact with the reader antenna only when in contact with it.

To fully test the tag performance we decided to apply it an antenna as specified by Murata in its white paper [6]. As those tags are projected to be applied to a PCB to extend it with RFID capabilities, an antenna matching circuit uses the inductive coupling section to couple with an external antenna system usually printed on the same PCB they are applied to.

Following those instruction summed up by the figure 2.18 taken from [6], we developed a custom antenna resorting to a copper plate cut to form a dipole with the tag placed in the center. The final realisation of the antenna system with the tag applied can be seen in figure 2.19

Some different length of the dipole were tested following the specification of the UHF RFID system number 3 of [7] but only with the half wave dipole the tag was readable from a maximum

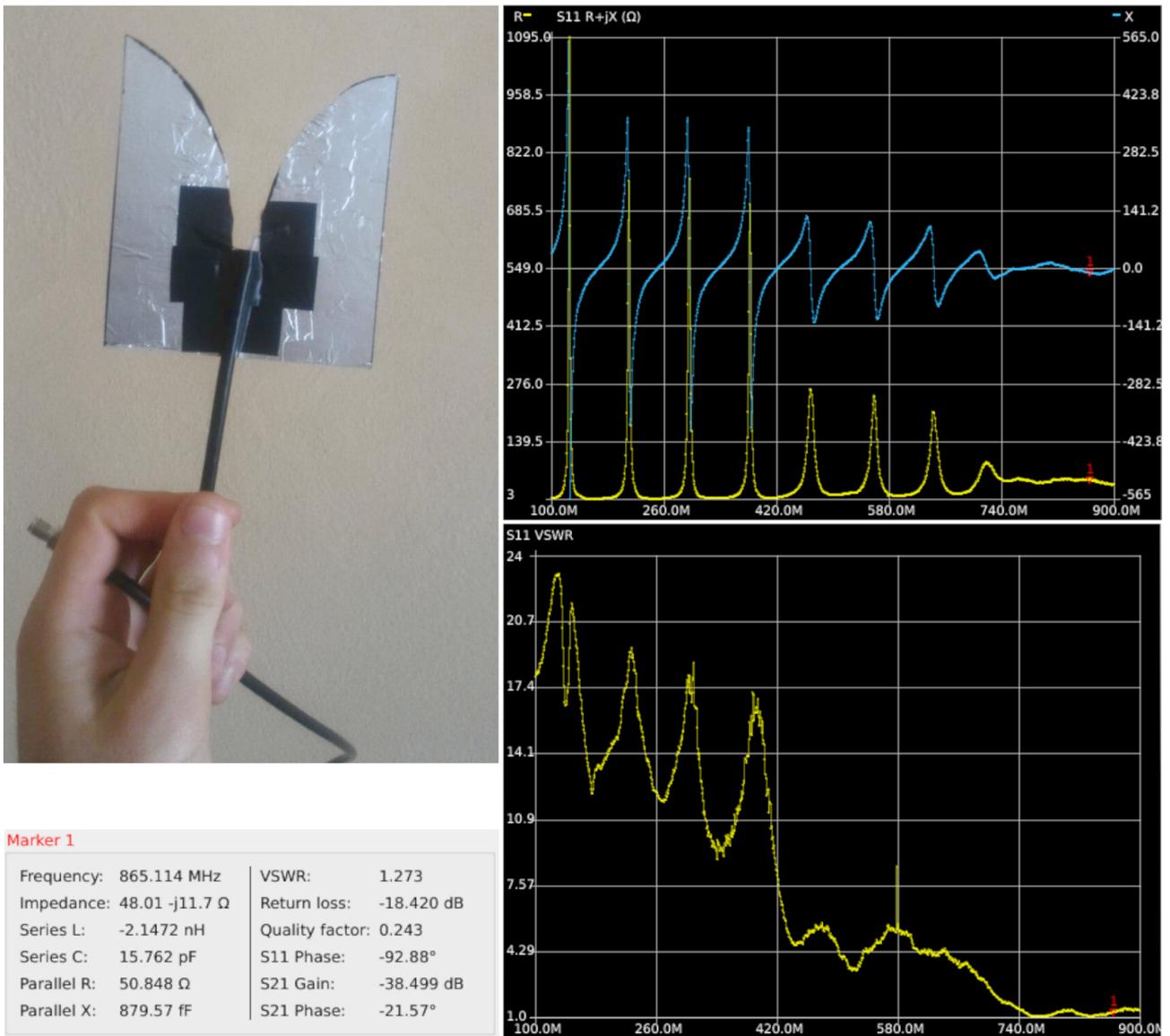


Figure 2.17: Vivaldi Antenna Realization and Analysis

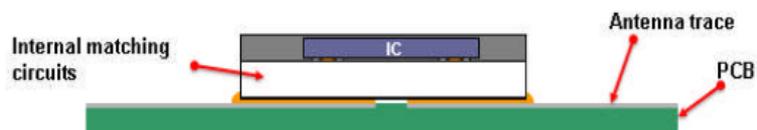


Figure 2.18: Cross-section of Murata MAGICSTRAP Placement Sample



Figure 2.19: UHF Tag Applied on the Custom Antenna

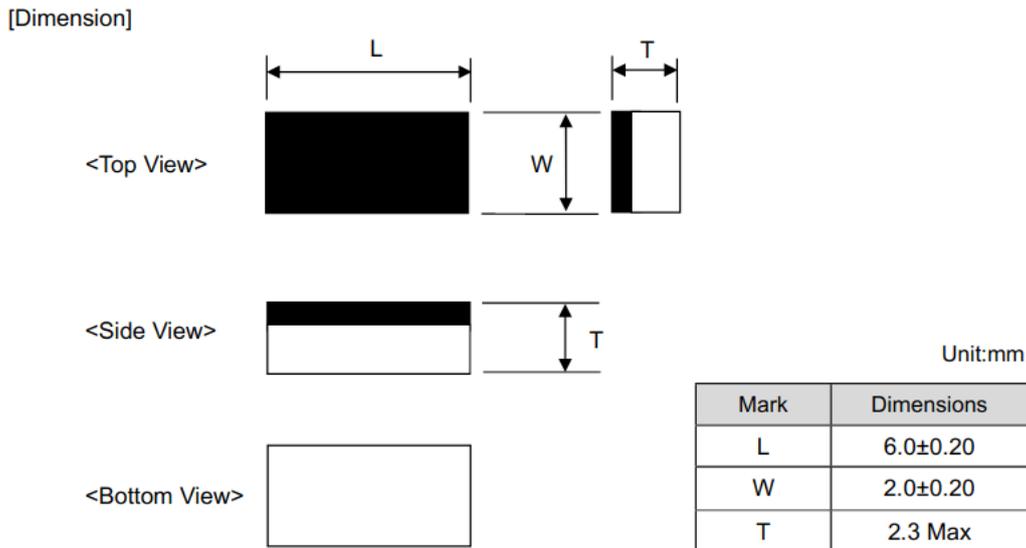


Figure 2.20: Murata LXTBKZMCMG-010 Dimensions

distance of 10cm using the Vivaldi antenna developed before.

2.4 UHF Solution

As the previous tests showed that the tag antenna is mandatory to establish the communication between tag and reader when not in contact, a new UHF tag has been purchased: the chosen IC is the Murata LXTBKZMCMG-010 that has a little integrated antenna on the bottom coupled with the tag's electronics through an inductive coupling circuit similar to the one presented by the micro tags tested before.

This solution has bigger dimensions as can be seen in figure 2.20 if compared with the previously seen UHF tags but it's smaller than the LF RFID glass capsule.

Together with the smallest size, this solution has another feature that makes it more interesting compared to the LF one: it has a 91 bytes memory that can be used to store user defined data. This feature allows us exploit all the capabilities of the UHF Rain RFID systems as presented during the analysis that brought us to choose this system as solution for the identification problem proposed us.

UHF Solution Tests

To test the new tags, the antennas built before has been used together with the ST25RU3993 Evaluation board, this process helped us to identify the better solution to reach a reliable radio link. The communication showed to be successful when the tag was standing in the region of higher magnetic field of the different antennas making the Vivaldi antenna unusable and the dipole one the smallest and easier to be used with the tag embedded in the plant trunk. The tag showed good performances in the near-field region as it was fully readable up to 1cm from the dipole antenna, but still unreadable when in a far-field coupling.

2.5 RFID Solutions Integration Test

Once the RFID solutions to the problem has been found, a final test replicating the operational conditions has been performed: some plants' branches has been used to simulate the grafted plant in witch the tags will be integrated and a some communication tests has been performed.



Figure 2.21: Tag Integration Into Branch (UHF Left, LF Right)

In figure 2.21 can be seen the integration solution adopted for the final tests obtained by drilling a 4mm diameter hole into a 1cm diameter hazelnut plant branch.

LF Solution Test

To test the LF RFID solution, the glass capsule has been inserted in the hole drilled in the branch and the obtained system has been brought near the reader, at a distance of 1cm from the reader antenna a successful reading has been performed. This results showed to be constantly repeatable several times from different angulation of the reader respect the branch.

UHF Solution Test

To test the UHF RFID solution, the Murata IC has been inserted in the hole previously used by the glass capsule of the LF tag, the obtained solution has been brought near the antenna of the reader and again the communication showed to be successful at a distance of 1cm. The main difference between this solution and the LF one is that the performance showed by the UHF system wasn't uniform for the different orientations of the reader antenna around the branch; this behaviour is caused by the position of the micro antenna applied on the bottom of the tag's IC that perform better when the reader's antenna is placed below the IC.

Chapter 3

Custom Software Development

3.1 LF RFID System Software

The Sparkfun ID-12 RFID reader is connected to the host PC through a serial port where the data are transferred at a boud rate of 9600bps with the 8N1 encoding.

To read the incoming data from the serial port, the C++ *SerialStream* has been used.

As the LF tags are read only, the only think we can do is to wait for incoming data from the serial port and display them when ready.

This custom serial monitor has been used during our tests but allow us to develop a more complex software able to inventory the tags and store them in a database if needed. This solution can be interesting for the final application where the applied tags' EPC must be stored on a server for the identification of the rose plants already sold.

This LF RFID system is able to read only one tag at a time, differently from the UHF one that we will describe later; this problem is not affecting our project as in our scenario one plant at a time will be checked and the interrogation area is small enough to prevent other tags being into it at the same time.

3.1.1 Code Breakdown

A quick analysis of the code is mandatory for a full comprehension of the work done.

Firstly a serial communication is established through the function *serial_port.Open()* and the success of that is checked before proceeding, otherwise the process is closed printing an error message. After the successful serial port connection, the correct boud rate is set with the function *serial_port.SetBaudRate()* together with the 8N1 encoding set as default for the *SerialStream* object.

Once the serial port has been configured a loop start to check for incoming data through a *get()* procedure reading one byte at a time. When a full ID has been transferred from the port, it will be displayed on the screen through the standard out stream.

3.2 UHF RFID System Software

Since this point we used the application developed by ST to perform our hardware tests on the UHF system, this approach gave us the possibility to tune the evaluation board parameters with an easy-to-use graphical interface without the need of code variations and compilation times overhead.

Once the hardware parameters has been set up to fulfill our needs, some custom software has been implemented to test the read and write capabilities of the system.

3.2.1 EPC UHF Gen2 Air Interface Protocol

Our RFID system works with the EPC UHF Gen2 Air Interface Protocol, so an accurate analysis of how it works is mandatory. As the physical communication constraints are implemented in the ST25RU3993 IC, we need to understand the correct sequence of operations to be performed to successfully interact with the tags.

Interrogation Procedure

In figure 3.1 taken from the EPC UHF Gen2 Air Interface Protocol specification paper we can find the operation that can be performed on a tag and its internal states.

Together with the Windows application, ST provides an host side library to interact with the evaluation board implementing the different commands that are required by the EPC Gen2 standard to correctly inventory and access the tags.

Tag's Memory Map

Before starting developing the application we need to understand how the internal memory of the tag is structured and what we can find inside: the figure 3.2 shows the internal memory mapping of the EPC Gen2 compliant tags.

Every tag has a unique 16 bit number stored inside, the EPC, whose function is precisely the identification of the tag.

3.2.2 Software Implementation

Once identified the correct interrogation procedure and the data to be retrieved by the tags, we can start developing our custom application, in particular we start with a simple EPC reading process to further implement the memory reading and writing procedure to use the tags internal memory.

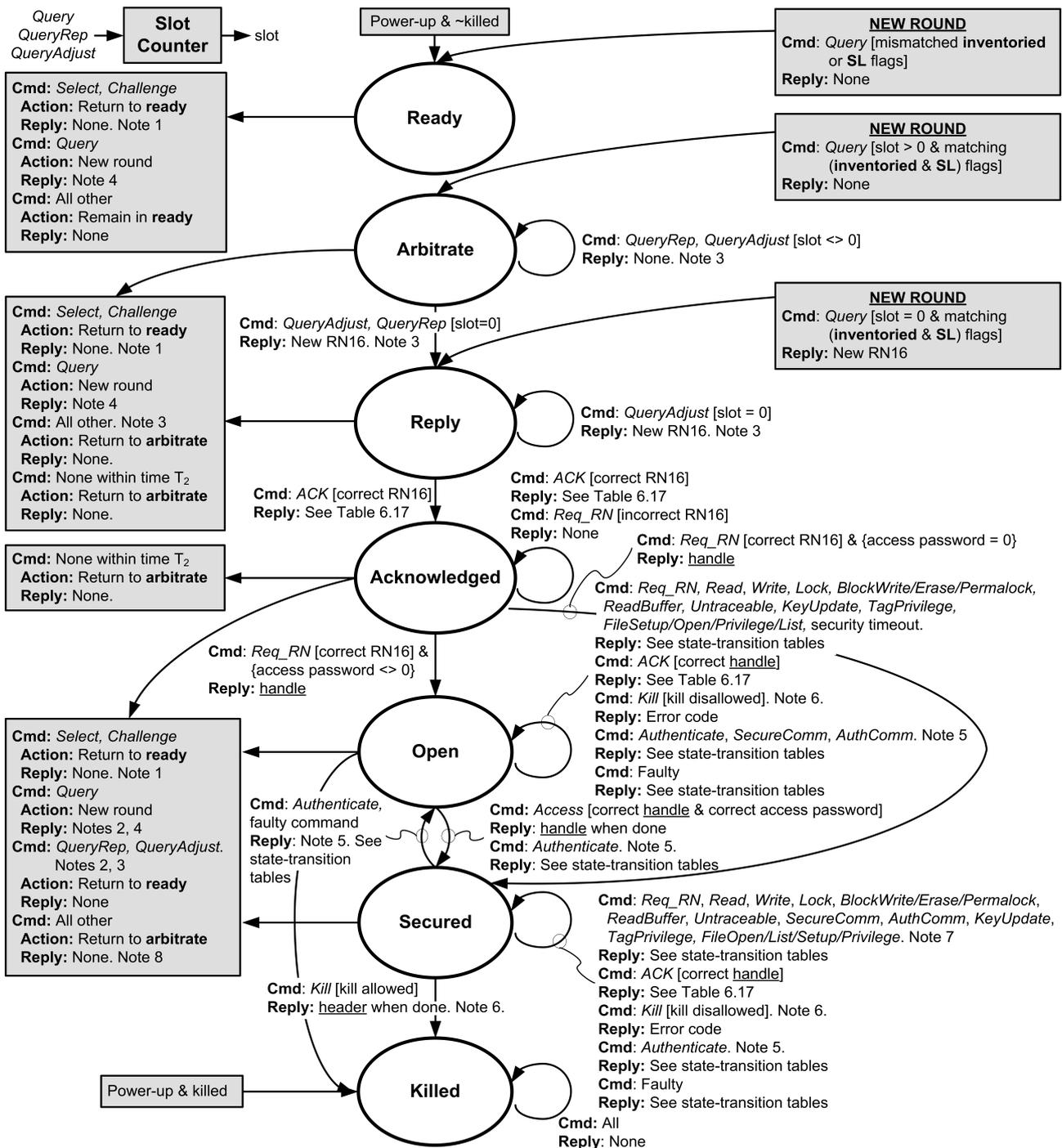
STUHFL Library

As mentioned before, ST delivers the sources of a C library to interact with the evaluation board; it can be compiled and used with the Microsoft Visual Studio IDE and offer different way to use it thanks to three different configurations. One of the offered possibilities is the cross compilation on a Raspberry Pi board as target platform using a Windows PC as host computer. This option is really useful for the development of embedded application able to use the ST25RU3993 Evaluation Board as part of more complex systems. For this reason we chose this option and, after having fully set up the target platform, performed the procedure suggested by ST to successfully compile and deliver the library to the microcomputer.

As Raspberry Pi we used an A+ one to minimize the dimensions and the power consumption as we want to develop a battery powered device without needing too much computational power.

The STUHFL library is divided into 3 abstraction layers that allow the developer to interact with the evaluation board starting from an high level of abstraction to the finest configuration details.

For our purposes we used the functions that allow us to assert the different EPC Gen2 commands, accessing the tuning details only for the parameters concerning the radio frequency used for the communication and the settings of the external power amplifier.



- NOTES**
1. *Select*: Assert/deassert **SL** or set *inventoried* to A or B.
Challenge: Perform action(s) indicated by *message*, store *result*, and assert **C** flag in *XPC_W1*.
 2. *Query*: A → B or B → A if the new *session* matches the prior *session*; otherwise no change to the *inventoried* flag.
QueryRep/QueryAdjust: A → B or B → A if *session* matches that of the prior *Query*.
 3. If the command is a *QueryRep* or *QueryAdj* and *session* does not match that of the prior *Query* than the *Tag* ignores the command.
 4. *Query* starts a new round and may change the *session*. *Tags* may go to **ready**, **arbitrate**, or **reply**.
 5. See the *state-transition tables* and the *cryptographic suite* for conditions, *message* formatting, *tag responses*, and *state changes*.
 6. Whether a *kill* is allowed or disallowed depends on the *kill pwd*, *Tag privileges*, and *security timeout*. See the *Kill command-response table*.
 7. If the *Interrogator* is authenticated then certain commands require encapsulation in an *AuthComm* or a *SecureComm*. See Table 6.28.
 8. A *Tag* that returns to **arbitrate** as a result of an unsuccessful access or kill, or a *cryptographic error*, may set a *security timeout*. See 6.3.2.5.

Figure 3.1: Tag State Diagram

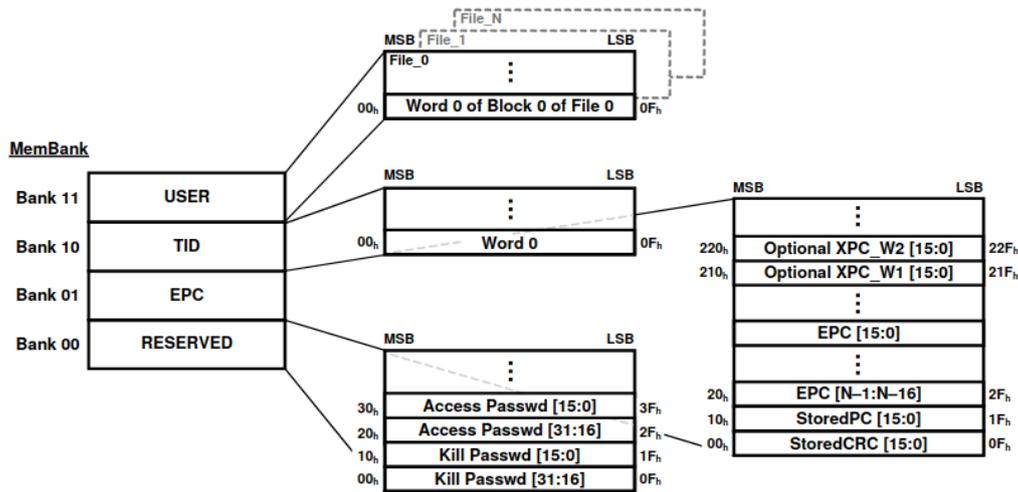


Figure 3.2: Tag Internal Memory Structure

Code Breakdown

To better understand how we performed the communication between the reader and the tag we have to analyse some parts of the code.

Firstly we established the connection between the Raspberry Pi and the ST25RU3993 Evaluation Board through a serial port, this operation is performed by the *Connect()* function from the STUHFL library. Once the correct connection has been checked through the returned value from the *Connect()* function, a board reboot command is stated to reach a clear state on the evaluation board's microcontroller.

As the microcontroller and the ST25RU3993 IC are now reset, the configuration procedure needs to set them up with the parameters determined during the previous tests.

Once the Board and the Gen2 protocol has been correctly set up, the system is ready to perform the tags inventory.

In figure 3.3 taken from the UHF Gen2 protocol specification is reported the correct procedure to perform the inventory of the tags. This is a quite complex procedure if compared to the one performed for the LF RFID system, due to the anti-collision mechanism that is part of the standard.

The inventory procedure is performed by the *Gen2_Inventory()* function, in this case called with default parameters to reach a full inventory of all the tags without any masking functionality.

As the inventory round reaches its end, the found tags are printed on the screen.

Once the tags have been inventoried, the read and write operation can be performed on them through the procedure we implemented that can read or write one byte of data at a time.

3.3 Custom ST25RU3993-EVAL Board Firmware

Once the correct procedure to Inventory, Select and Access the RAIN RFID tags has been identified and tested, the next step has been to develop a custom firmware for the microcontroller on the ST25RU3993-EVAL Board; this solution has been chosen to have the possibility to develop custom hardware in case it's needed to create a smaller or different device featuring the ST25RU3993 IC.

Interrogators manage Tag populations using the three basic operations shown in [Figure 6-22](#). Each of these operations comprises multiple commands. The operations are defined as follows:

- a. **Select:** The process by which an Interrogator selects a Tag population for subsequent inventory or cryptographically challenges a Tag population for subsequent authentication. Select comprises the *Select* and *Challenge* commands.
- b. **Inventory:** The process by which an Interrogator identifies Tags. An Interrogator begins an inventory round by transmitting a *Query* command in one of four sessions. One or more Tags may reply. The Interrogator detects a single Tag reply and requests the PC, optional XPC word(s), EPC, and CRC-16 from the Tag. An inventory round operates in one and only one session at a time. [Annex E](#) shows an example of an Interrogator inventorying and accessing a single Tag. Inventory comprises multiple commands.
- c. **Access:** The process by which an Interrogator transacts with (reads, writes, authenticates, or otherwise engages with) an individual Tag. An Interrogator singulates and uniquely identifies a Tag prior to access. Access comprises multiple commands.

Figure 6-22: Interrogator/Tag operations and Tag state

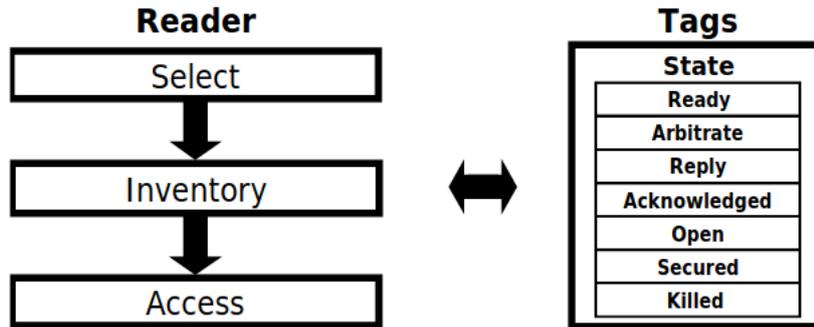


Figure 3.3: Tag Population Management

3.3.1 ST25RU3993-EVAL Board Hardware Analysis

The development process started with an accurate analysis of the ST25RU3993-EVAL Board thanks to the documentation that can be found on the ST website. In figure 3.4 we can see the block diagram of the ST25RU3993-EVAL Board that allow us to understand the interconnections between the MCU and the other components of the board.

Once all the components have been understood and the interconnections identified, it's time to identify the different devices on the board, to do so, the documentation released by ST shows the image reported here in figure 3.5 where the different components soldered on the board are highlighted and explained, allowing us also to see under the RF shield of the radio frequency section.

As the hardware components and interconnections have been identified, more focus can be applied to the RF section in order to understand how it works and how to control the different external components that form the Power stage and antenna selection stage of the RF path.

Obviously the core component of the RF circuit is the ST25RU3993 chip as it integrate all the functionalities needed to perform the RAIN RFID operations together with a power stage able to drive some mW of power. The output of the internal PLL charge pump is connected to LF_CEXT (pin 45), and the external part of the loop filter is placed in close proximity. An additional low-pass filter stage is integrated in ST25RU3993 and is part of the loop filter circuit. The loop filter output is the control voltage of the internal VCO. The carrier frequency is modulated by the ASK or PR-ASK shaped modulation signal.

Following the RF path we van find an external power amplifier able to provide a 28dBm power output, that can be connected or bypassed thanks to two electronically controlled RF switches that in the same time allow the user to choose the antenna output as the ST25RU3993-EVAL Board provides two different antenna connectors.

in figure 3.6 is reported the graph with the obtainable power from the internal power amplifier as for the external one.

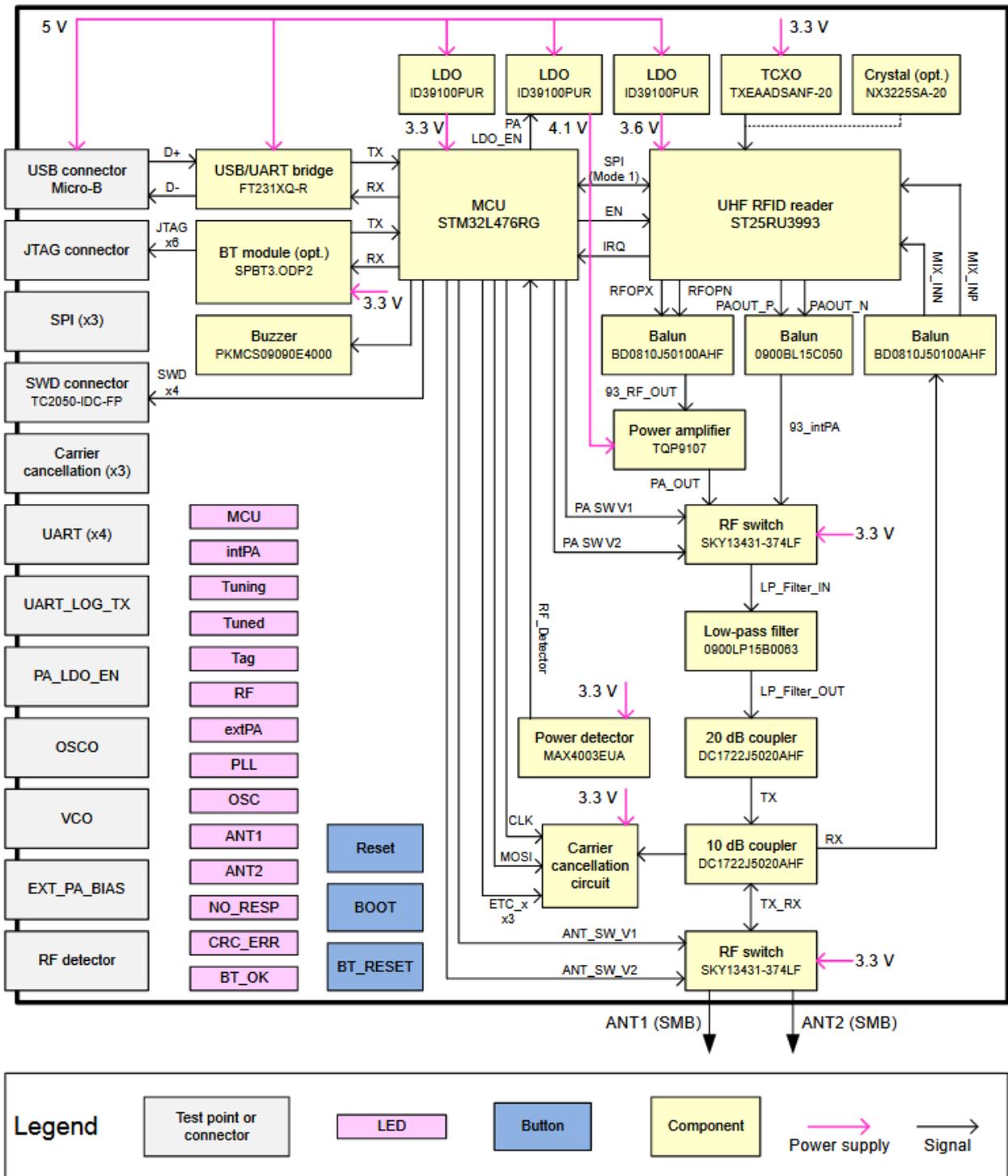


Figure 3.4: ST25RU3993-EVAL Board Block Diagram

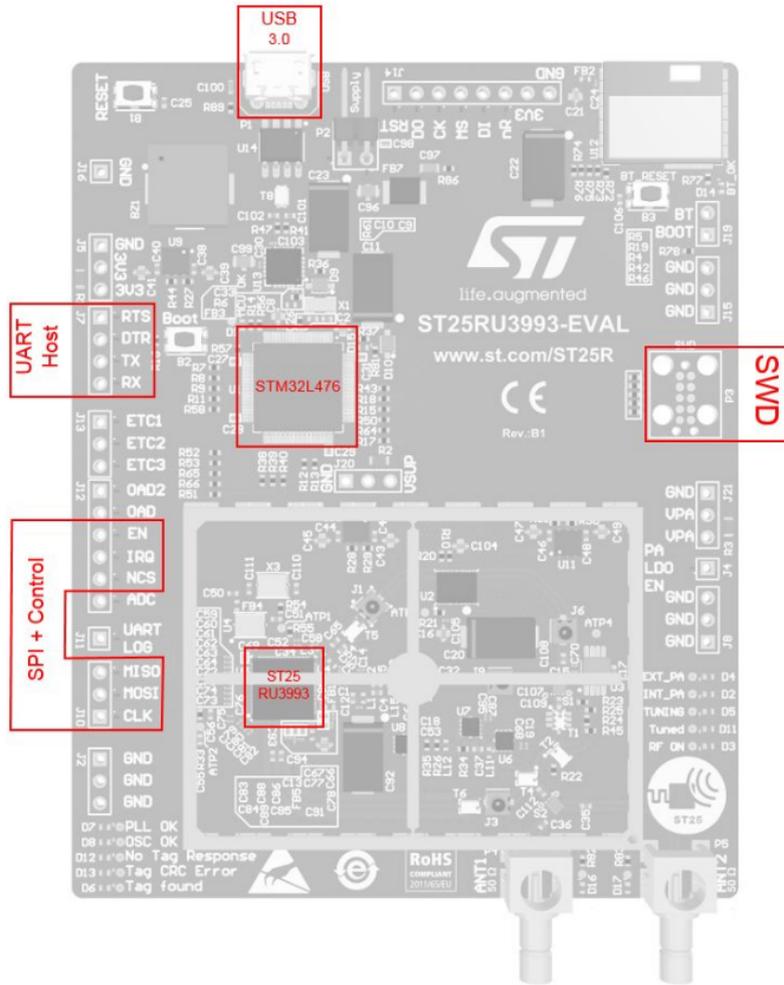


Figure 3.5: ST25RU3993-EVAL Board Components

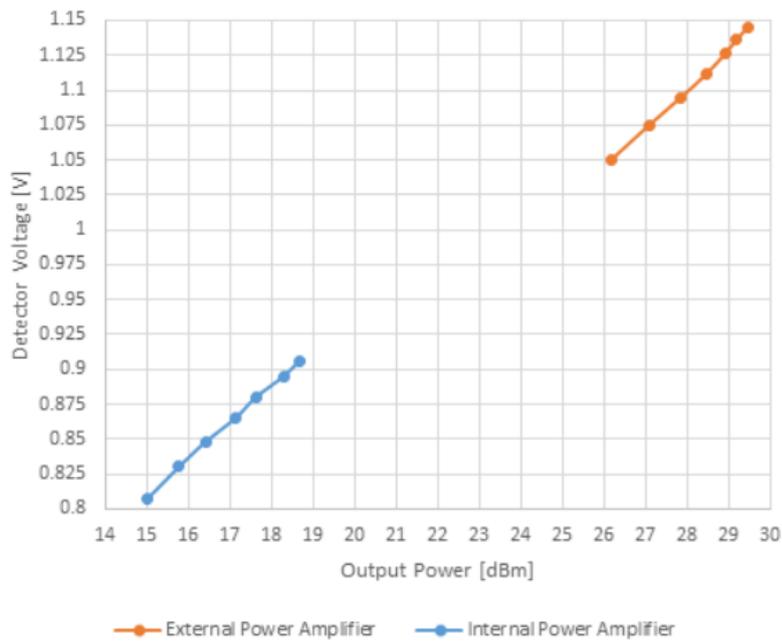


Figure 3.6: ST25RU3993-EVAL Board Output Power

Along the RF path, some RF transformers and baluns allow the microcontroller and the ST25RU3993 IC to monitor the RF power of the different stages.

To maintain the sensitivity of the reader the self-jamming signal reaching the receiving inputs of ST25RU3993 needs to be minimized. The self-jamming signal comprises reflections from the antenna (S11) and the leakage across the main directional coupler. To minimize the self-jamming signal a carrier cancellation circuit is connected to the coupled port of the main directional coupler. The carrier cancellation circuit is able to change its impedance and hence to reflect a certain amount of the coupled power back into the directional coupler. This reflected signal is combined with the self-jamming signal at the isolated port of the main directional coupler. The isolated port of the main directional coupler is connected to the receiving pins of ST25RU3993. In theory, if the signal reflected by the carrier cancellation circuit has the same amplitude and the opposite phase, then the self-jamming signal will cancel-out and vanish.

3.3.2 STM32CubeIDE Project

St, together with the APIs used before to develop the High-Level application on the RaspberryPi board, released a pre-configured project to be used in STM32CubeIDE as other projects for other IDEs.

The choice of the STM32CubeIDE was made for the simplicity in the MCU configuration thanks to the graphical procedure developed by ST that allow the user to visualise the MCU pins configuration together with the peripherals' one. Another reason that lead the choice to the STM32CubeIDE has been the multi-OS nature of the IDE that allowed us to develop our application resorting to Windows and Linux without any modification to the project.

Project Configuration

To start developing the custom firmware for the ST25RU3993-EVAL Board, a new project has been derived from the example proposed by ST in the Firmware folder of the ST25RU3993-EVAL Board SDK, in this way a clean but pre-configured project has been created.

Once the project is fully prepared, STM32CubeIDE shows us the MCU configuration panel when we can see the pin configuration together with the peripherals setups as can be seen in figure 3.7.

For our purposes this configuration is perfect, in case of custom hardware based on this project, the different setups involving pins and peripherals can be tuned to fit the project constraints.

Once the configuration is fully tuned and checked, we can create the basic structure of the code thanks to the automatic code generation process performed by STM32CubeIDE when the configuration wizard is closed.

Custom Functions

As seen before the main communication system between MCU and ST25RU3993 IC is the SPI bus together with its control signals that must be asserted following the correct procedures reported in the documentation. To do so, a custom functions library has been made not to have to repeat all the same process for every communication step.

The implemented functions are the following:

ST25RU3993_Interface_Init initialises the SPI communication just calling the Init function of the SPI interface APIs, this function has to be called at the beginning of the firmware code.

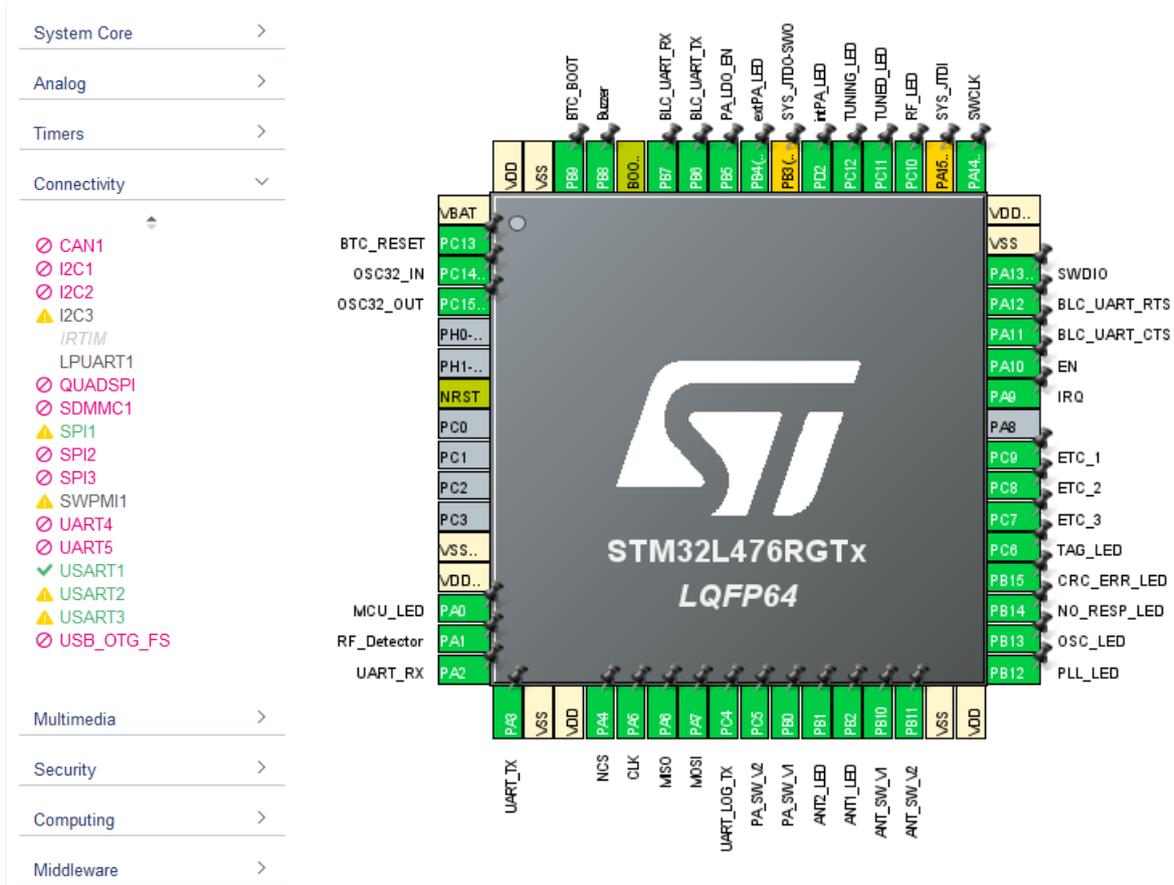


Figure 3.7: MCU configuration in STM32CubeIDE

```
void ST25RU3993_Interface_Init(void)
{
    MX_SPI1_Init();
}
```

ST25RU3993_Enable enables the ST25RU3993 IC asserting a logic high voltage to the IC enable pin.

```
void ST25RU3993_Enable()
{
    HAL_GPIO_WritePin(ST25RU3993_ENABLE_PORT, ST25RU3993_ENABLE_PIN, 1);
    HAL_Delay(100);
}
```

ST25RU3993_Disable disables the ST25RU3993 IC asserting a logic low voltage to the IC enable pin.

```
void ST25RU3993_Disable()
{
    HAL_GPIO_WritePin(ST25RU3993_ENABLE_PORT, ST25RU3993_ENABLE_PIN, 0);
    HAL_Delay(100);
}
```

ST25RU3993_ReadRegister reads the requested register's content, this is the first function that operates on the SPI, so it's interesting to see how the function works.

First of all the correct request byte shall be created following the structure presented in the documentation released by ST, in this case we can see how the read request is performed placing the 01 bit sequence in the MSBs of the byte. The read command bits are followed by the register address taken from the function's parameters.

After the request byte has been composed, the SPI transmission take place with the correct signal assertion sequence, resorting to the SPI peripheral's API. As the request is sent, the st25RU3993 IC response is waited and the received data returned to the caller; at the end of the sequence the SPI is disabled.

```
HAL_StatusTypeDef ST25RU3993_ReadRegister(uint8_t reg, uint8_t* data)
{
    HAL_StatusTypeDef return_code;

    // Build byte to create read register request
    uint8_t read_cmd = 0b01000000 | (reg & 0b00111111);

    // Enable SPI and transmit read request
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 0);
    HAL_Delay(10);
    return_code = HAL_SPI_Transmit(&hspi1, &read_cmd, 1, 100);

    // If all is ok, read response
    if(return_code == HAL_OK)
    {
        return_code = HAL_SPI_Receive(&hspi1, data, 1, 100);
    }

    // Disable SPI
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 1);
    HAL_Delay(10);

    return return_code;
}
```

ST25RU3993_WriteRegister writes the desired content in the requested register; its behaviour is really similar to the reading function presented above, the only grat difference stand in the request byte that is composed by the register address with two zeros in the two MSBs as requested by the documentation.

As this is a writin sequence, no response is waited after the data are transferred to the ST25RU3993 IC through the SPI bus.

```
HAL_StatusTypeDef ST25RU3993_WriteRegister(uint8_t reg, uint8_t data)
{
    HAL_StatusTypeDef return_code;

    // Build byte to create write register request
    uint8_t read_cmd = 0b00000000 | (reg & 0b00111111);

    // Enable SPI and transmit write request
```

```

HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 0);
HAL_Delay(1);
return_code = HAL_SPI_Transmit(&hspi1, &read_cmd, 1, 100);

// If all is ok, write data
if(return_code == HAL_OK)
{
    return_code = HAL_SPI_Transmit(&hspi1, &data, 1, 100);
}

// Disable SPI
HAL_Delay(1);
HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 1);
HAL_Delay(1);

return return_code;
}

```

ST25RU3993_WriteFIFO performs the write sequence on the FIFO.

As we can read from the ST documentation, the main communication system between MCU and ST25RU3993 ID during the RFID communication process is the ST25RU3993 internal FIFO that must be accessed following the correct procedure; the writing process on the FIFO is used to communicate the data to be transferred to the RFID tag or to upload the parameters of some EPG Gen2 commands.

The FIFO is accessed as a normal register with the address 0b00111111, once the write request has been asserted, the content to be written is transmitted sequentially.

To acknowledge the ST25RU3993 IC of the data inside the FIFO, a special register is used to store the data length, so after the successful transmission of the FIFO data, this register is filled with the correct length. To do so the previously created functions are used.

```

HAL_StatusTypeDef ST25RU3993_WriteFIFO(uint8_t len, uint8_t* data)
{
    HAL_StatusTypeDef return_code;

    // Build byte to create write register request
    uint8_t read_cmd = 0b00000000 | (FIFO_IO_REG & 0b00111111);

    // Enable SPI and transmit write request
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 0);
    HAL_Delay(1);
    return_code = HAL_SPI_Transmit(&hspi1, &read_cmd, 1, 100);

    // If all is ok, write data
    if(return_code == HAL_OK)
    {
        return_code = HAL_SPI_Transmit(&hspi1, data, len, 100);
    }

    // Increment fifo tx length register
    if(return_code == HAL_OK)

```

```

{
    return_code = ST25RU3993_WriteRegister(TX_LENGTH_2_REG, 0x20);
}

// Disable SPI
HAL_Delay(1);
HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 1);
HAL_Delay(1);

return return_code;
}

```

ST25RU3993_ReadFIFO performs the reading process of the FIFO, this function is very similar to the write one with the difference that the request byte is composed by the 01 sequence followed by the FIFO register address and no length register is involved.

As can be seen in the code snippet, this function reads a number of bytes expressed in the len parameters and return the read content through the data parameter that must be properly allocated before the function call by the firmware application.

```

HAL_StatusTypeDef ST25RU3993_ReadFIFO(uint8_t len, uint8_t* data)
{
    HAL_StatusTypeDef return_code;

    // Build byte to create write register request
    uint8_t read_cmd = 0b01000000 | (FIFO_IO_REG & 0b00111111);

    // Enable SPI and transmit write request
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 0);
    HAL_Delay(1);
    return_code = HAL_SPI_Transmit(&hspi1, &read_cmd, 1, 100);

    // If all is ok, write data
    if(return_code == HAL_OK)
    {
        return_code = HAL_SPI_Receive(&hspi1, data, len, 100);
    }

    // Disable SPI
    HAL_Delay(1);
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 1);
    HAL_Delay(1);

    return return_code;
}

```

ST25RU3993_SendCommand sends the desired command to the ST25RU3993 IC, the full command list is reported in the documentation released by ST and includes commands for the IC itself together with the commands needed by the EPC Gen2 protocol to communicate with the tags.

This function is sometimes to be used together with the FIFO write functions as some commands such as the ones implementing the EPC Gen2 Inventory sequence need some parameters to be previously uploaded in the ST25RU3993 FIFO.

The command assertion sequence is similar to the register write one, the only great difference is the need of the sequence of bits 10 in the request byte.

```
HAL_StatusTypeDef ST25RU3993_SendCommand(uint8_t command)
{
    HAL_StatusTypeDef return_code;

    // Build byte to create direct command request
    uint8_t read_cmd = 0b10000000 | (command & 0b00111111);

    // Enable SPI and transmit command
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 0);
    HAL_Delay(1);
    return_code = HAL_SPI_Transmit(&hspi1, &read_cmd, 1, 100);

    // Disable SPI
    HAL_Delay(1);
    HAL_GPIO_WritePin(SPI_ENABLE_PORT, SPI_ENABLE_PIN, 1);
    HAL_Delay(1);

    return return_code;
}
```

ST25RU3993_SetupInventory asserts the EPC Gen2 Inventory request.

This function is really important as it demonstrates the sequence expressed before about the use of the EPC Gen2 commands that needs some parameters.

As seen before for the high-level application the inventory process is recursively used in the code and needs some tuning parameters to better perform the communication between reader and tags.

The inventory process is also the key point in the EPC Gen2 protocol as it allow the multi tag discrimination.

As can be seen in the code snippet, two FIFO bytes are used to store all the parameters needed by the inventory command, the way they are placed in the two bytes is represented in the documentation released by ST.

```
HAL_StatusTypeDef ST25RU3993_SetupInventory(uint8_t DR, uint8_t M,
                                             uint8_t Tnext, uint8_t sel,
                                             uint8_t session, uint8_t target,
                                             uint8_t Q)
{
    HAL_StatusTypeDef return_code;

    uint8_t first_data_byte = 0x00;
    uint8_t second_data_byte = 0x00;

    second_data_byte |= (Q & 0x0F) << 1;
    second_data_byte |= (target & 0x01) << 5;
    second_data_byte |= (session & 0x03) << 6;
```

```

first_data_byte |= (sel & 0x03);
first_data_byte |= (TRext & 0x01) << 2;
first_data_byte |= (M & 0x03) << 3;
first_data_byte |= (DR & 0x01) << 5;

return_code = ST25RU3993_WriteRegister((FIFO_IO_REG & 0b00111111),
                                       first_data_byte);
return_code += ST25RU3993_WriteRegister((FIFO_IO_REG & 0b00111111),
                                       second_data_byte);

if(return_code == HAL_OK)
{
    return_code = ST25RU3993_SendCommand(QUERY_CMD);
}

return return_code;
}

```

ST25RU3993_UpdateStatusLeds sets the correct status to the board leds.

This function has been the most important during the debug sequence as it allows us to see the ST25RU3993 IC internal status just looking at the board's leds. To perform its duty, this function reads the status register of the ST25RU3993 IC and then displays some useful bits as led status; in particular the bits reporting the status of the internal PLL and tuning path.

```

HAL_StatusTypeDef ST25RU3993_UpdateStatusLeds()
{
    uint8_t status1;

    HAL_StatusTypeDef ret = ST25RU3993_ReadRegister(AGC_INTERNAL_STATUS_REG,
                                                    &status1);

    if(ret == HAL_OK)
    {
        HAL_GPIO_WritePin(PLL_LED_PORT, PLL_LED_PIN, !(status1 & 0b00000010));
        HAL_GPIO_WritePin(OSC_LED_PORT, OSC_LED_PIN, !(status1 & 0b00000001));
        HAL_GPIO_WritePin(RF_LED_PORT, RF_LED_PIN, !(status1 & 0b00000100));
    }

    return ret;
}

```

3.3.3 Custom Firmware

The main part of the custom firmware uses the previously presented functions to interact with the ST25RU3993 IC on the board, it's developed to fit the ST25RU3993-EVAL Board but can be easily changed to fit a custom board.

The firmware starts with the initialization of the different peripherals of the MCU together with the RFID IC resorting to our custom init function, enables the ST25RU3003 IC and reads

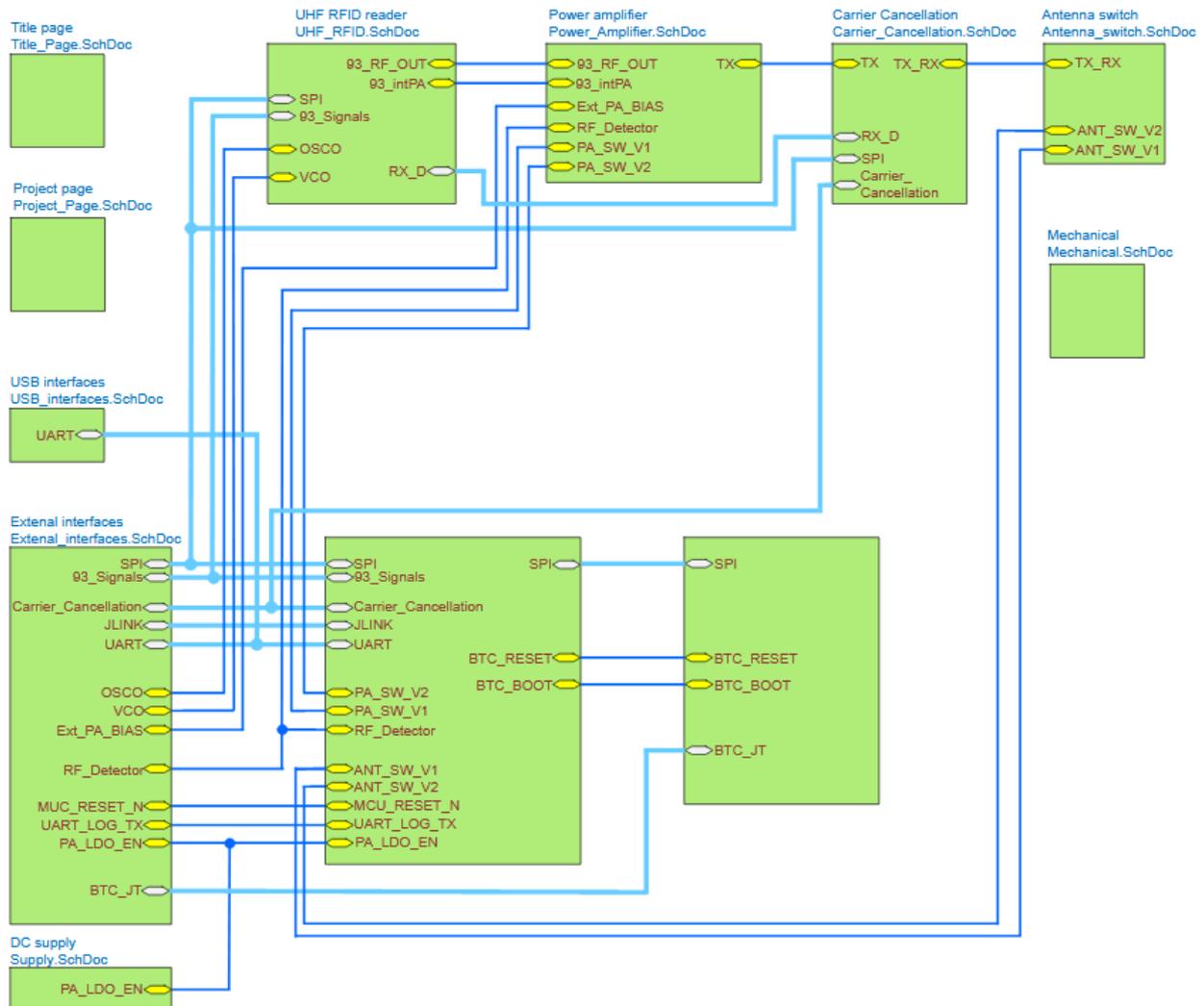


Figure 3.8: ST25RU3993-EVAL Board RF Path Schematics

its version number to test the SPI communication. If the version number is the one expected, the code can proceed, else an error signal is asserted blinking all the leds of the board.

Once the communication has been tested, the properly configuration is done to tune the ST25RU3993 IC on the right frequencies and using the desired power amplifier to emit the correct radio power. In our case we used the 28dBm power output as we have to deal with very small tags inserted into a challenging environment.

Together with the IC configuration, also the RF circuitry is properly configured selecting the external power supply and the correct antenna connector thanks to the RF switches that can be driven by the MCU according to the schematics of figure 3.8 taken from the ST25RU3993-EVAL Board documentation.

Once the IC and the board has been fully configured, the EPC Gen2 parameters are select to have 4 slots for the inventory and selecting as active session the number 0.

At the end of the configuration the register table of the RFID IC must be as the one of figure 3.9 in accordance to the ST technical support.

With the EPC Gen2 parameters configured as well, the TX and RX RF sections are enabled ant the main loop started. In the main loop the inventory function defined before is called and the tags response is checked in the FIFO of the ST25RU3993 IC.

This code's purpose is to inventory the tags found as it's used to test the ability to develop

address	val	address	val	address	val	address	val
0	03	10	00	20	00	30	00
1	00	11	48	21	00	31	00
2	22	12	35	22	00	32	00
3	98	13	20	23	00	33	61
4	03	14	EF	24	00	34	00
5	41	15	02	25	00	35	7F
6	05	16	89	26	00	36	07
7	05	17	64	27	00	37	00
8	0B	18	41	28	00	38	00
9	37	19	06	29	06	39	00
A	60	1A	04	2A	0F	3A	10
B	1B	1B	61	2B	00	3B	00
C	82	1C	18	2C	40	3C	00
D	30	1D	00	2D	8B	3D	00
E	40	1E	00	2E	18	3E	00
F	00	1F	00	2F	00	3F	00

Figure 3.9: ST25RU3993 IC Register table

custom software, more features can be added following the ST documentation.

3.3.4 Custom Firmware Compilation and Deployment

As the code is ready, the last step to perform is the compilation of the code and the binary deployment on de board.

The first step is very simple as we loaded a pre-configured project we can compile it just pressing the compile button on STM32CubeIDE; for the deployment of the binary file we can follow one of the two procedure presented by ST:

- The simplest way to program a new firmware version to the MCU is by connecting the ST25RU3993-EVAL board to the host PC with the micro USB cable and use the “Firmware Update” function of the GUI. This is the procedure used during our development.
- Alternatively the firmware can be programmed using the ST-LINK connecting to the JLINK (SWD) interface with a J-Link Needle Adapter. This option is the most suitable for programming a custom board.

Chapter 4

Conclusions

Starting from the request of identification of roses plants made us, we developed two different solution resorting to the Passive RFID technology.

The first solution that sees the LF RFID glass capsule applied, represents the more resilient solution as the Near-Field nature of the coupling between tag and reader performed through a strong magnetic field easily overcome the obstacle represented by the plant growing around the tag.

The main drawback of this solution is certainly the dimensions of the glass capsule tag that can't be easily implanted into the young roses plants.

To develop a more discrete solution able to be embedded into the plants through the grafting procedure, we found some UHF tag with dimensions of few millimeters. Those solutions are designed to be coupled to a PCB trace antenna thank through their inductive coupling circuitry. To use those tags in our project we decided to exploit the inductive coupling circuit to establish a communication between reader and tag.

Different antennas has been tested but the communication was successful only when tag and reader's antenna were in contact.

To solve this problem a different UHF tag with integrated antenna has been purchased trading between dimensions and readability, this solution took us to use an UHF tag bigger than the previously tested ones but still more suitable for the grafting integration than the LF one.

To demonstrate the fully compliance of our solution with the requested constraints, we develop a final test with the actual integration of the tags into a plant's branch, demonstrating the ability to interact with the tags in different configurations.

Together with the hardware design and testing, different pieces of software has been develop to automate the testing process and operate on the RFID systems under test. To develop the software needed by the project different approaches has been used resorting to the coding process together with the block programming technique offered by GNURadio Companion.

As the testing automation system resort to an SDR technology and has been designed as modular as possible, it can be used for further analysis ad ported to different project with small modifications as the case of the analysis of LPD devices or monitoring the LoRa channel to debug the communication systems.

Appendix A

NEC Simulations

A.1 865MHz Dipole Antenna

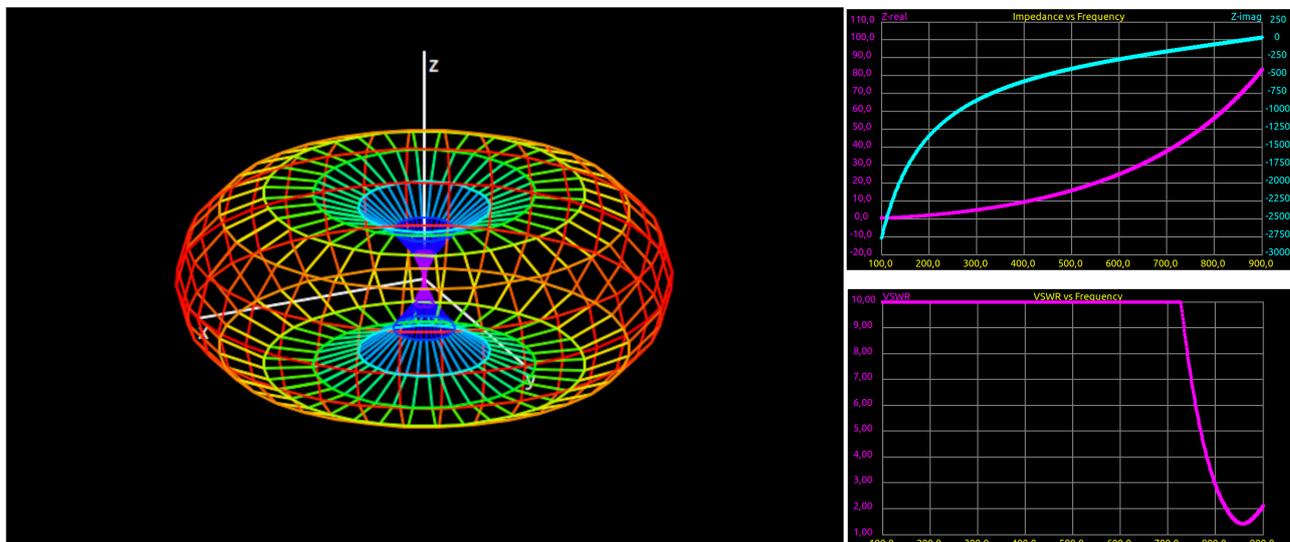
Geometry	
Wire:	
Length	164 mm
Diameter	1 mm
Segments	51
Excitation	
Start Frequency	100 MHz
Stop Frequency	900 MHz
Frequency Step	1 MHz

NEC Listing

```

GW 1 51 0,00000E+00 0,00000E+00 -8,20000E-02 0,00000E+00 0,00000E+00 8,20000E-02 5,00000E-04
GE 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
EX 0 1 26 0 1,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
FR 0 801 0 0 1,00000E+02 1,00000E+00 9,00000E+02 0,00000E+00 0,00000E+00 0,00000E+00
NH 0 0 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
NE 0 10 1 10 -1,35000E+00 0,00000E+00 -1,35000E+00 3,00000E-01 0,00000E+00 3,00000E-01
RP 0 19 37 1000 0,00000E+00 0,00000E+00 1,00000E+01 1,00000E+01 0,00000E+00 0,00000E+00
EN 0 0 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
  
```

Results



A.2 865MHz Loop Antenna

Geometry	
Arc:	
Radius	58 mm
Start Angle	0.0
End Angle	360.0
Diameter	1 mm
Segments	101
Excitation	
Start Frequency	100 MHz
Stop Frequency	900 MHz
Frequency Step	1 MHz

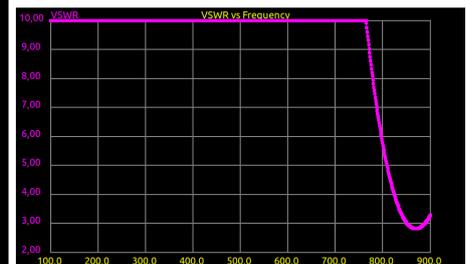
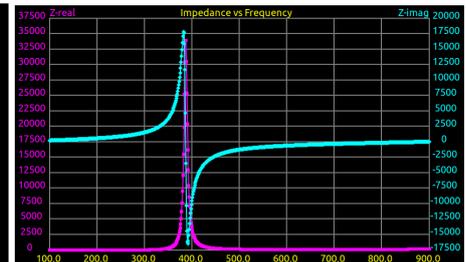
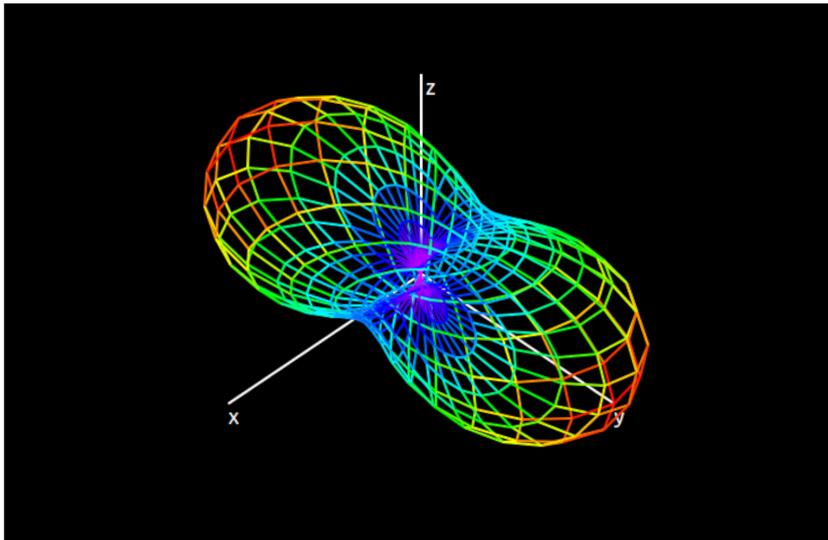
NEC Listing

```

GA 1 101 5,80000E-02 0,00000E+00 3,60000E+02 5,00000E-04 1,08317E+00 0,00000E+00 0,00000E+00
GE 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
EX 0 1 51 0 1,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
FR 0 801 0 0 1,00000E+02 1,00000E+00 9,00000E+02 0,00000E+00 0,00000E+00 0,00000E+00
NH 0 0 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00
NE 0 10 1 10 -1,35000E+00 0,00000E+00 -1,35000E+00 3,00000E-01 0,00000E+00 3,00000E-01
RP 0 19 37 1000 0,00000E+00 0,00000E+00 1,00000E+01 1,00000E+01 0,00000E+00 0,00000E+00
EN 0 0 0 0 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00 0,00000E+00

```

Results



Appendix B

FDTD Simulations (Octave OpenEMS)

B.1 865MHz Dipole Antenna

Octave Script

```
% init and define FDTD parameter
FDTD = InitFDTD('NrTS', 10000, 'EndCriteria', 0);
FDTD = SetSinusExcite(FDTD, 865e6);
BC = {'MUR' 'MUR' 'MUR' 'MUR' 'MUR' 'MUR'};
FDTD = SetBoundaryCond(FDTD,BC);

% init and define FDTD mesh
CSX = InitCSX();
mesh.x = -0.1:0.001:0.1;
mesh.y = -0.1:0.001:0.1;
mesh.z = -0.01:0.001:0.01;
CSX = DefineRectGrid(CSX, 1, mesh);

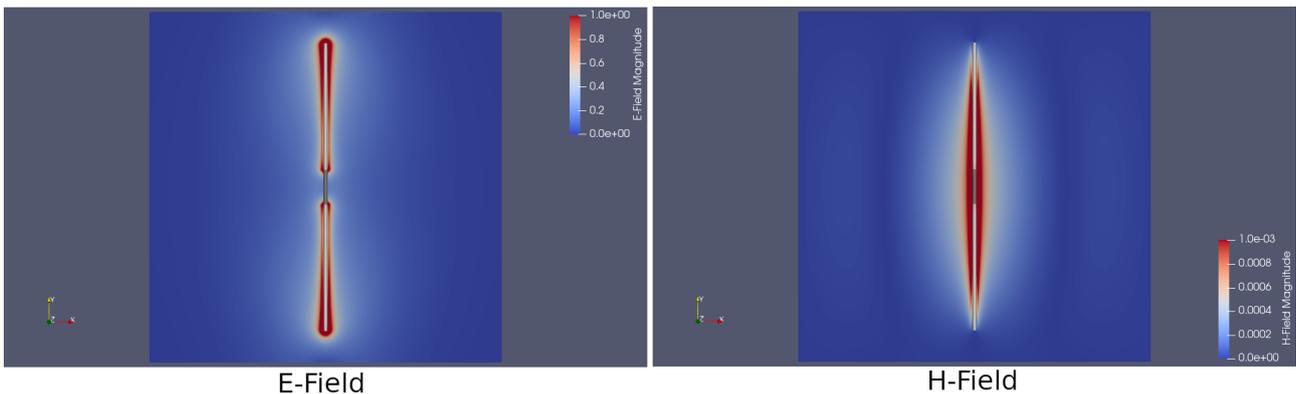
% define the excitation
CSX = AddExcitation(CSX,'excitation',0,[0 1 0]);
CSX = AddCylinder(CSX,'excitation',3,[0 -0.012 0], [0 0.012 0], 0.001);

CSX = AddMetal(CSX, 'metal');
CSX = AddCylinder(CSX, 'metal', 1, [0 -0.082 0], [0 -0.01 0], 0.001);
CSX = AddCylinder(CSX, 'metal', 1, [0 0.082 0], [0 0.01 0], 0.001);

% define time domain e-field and h-field dump box
CSX = AddDump(CSX,'Et','DumpType',0);
CSX = AddBox(CSX,'Et',0,[-1 -1 -0.01], [1 1 0]);
CSX = AddDump(CSX,'Hf','DumpType',1);
CSX = AddBox(CSX,'Hf',0,[-1 -1 -0.01], [1 1 0]);

% write openEMS xml data file and run simulation
WriteOpenEMS('tmp/tmp.xml',FDTD,CSX);
CSXGeomPlot('tmp/tmp.xml');
RunOpenEMS('tmp','tmp.xml','');
```

Results



B.2 865MHz Loop Antenna

Octave Script

```
% init and define FDTD mesh
CSX = InitCSX();
mesh.x = -0.1:0.001:0.1;
mesh.y = -0.1:0.001:0.1;
mesh.z = -0.01:0.001:0.01;
CSX = DefineRectGrid(CSX, 1, mesh);

% define the excitation
CSX = AddExcitation(CSX,'excitation',0,[0 1 0]);
CSX = AddCylinder(CSX,'excitation',3,[-0.058 -0.002 0], [-0.058 0.002 0], 0.001);

CSX = AddMetal(CSX, 'metal');
CSX = AddCylindricalShell(CSX, 'metal', 1, [0 0 -0.0005], [0 0 0.0005], 0.058, 0.001);

% define a time domain e-field dump box
CSX = AddDump(CSX,'Et','DumpType',0);
CSX = AddBox(CSX,'Et',0,[-1 -1 -0.01], [1 1 0]);

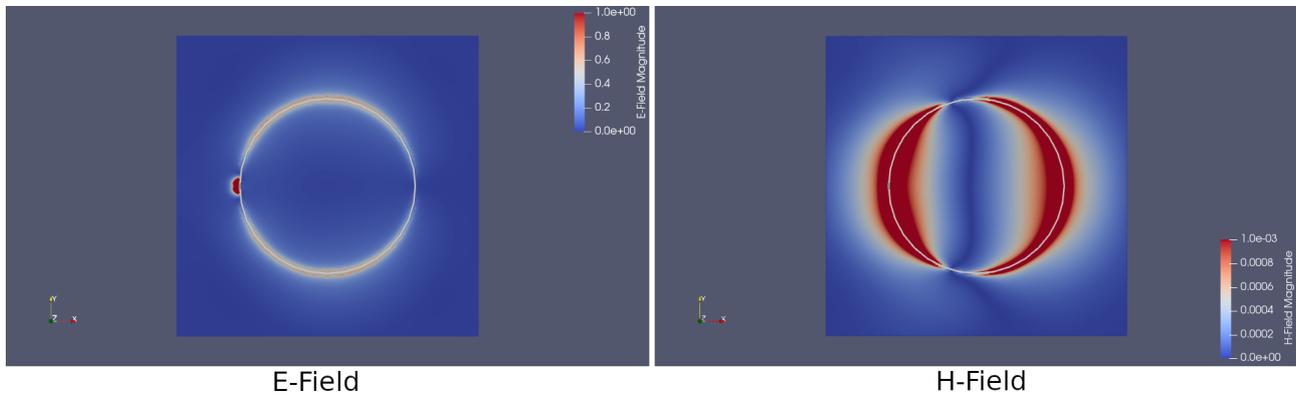
% define a time domain h-field dump box
CSX = AddDump(CSX,'Hf','DumpType',1);
CSX = AddBox(CSX,'Hf',0,[-1 -1 -0.01], [1 1 0]);

% write openEMS xml data file
WriteOpenEMS('tmp/tmp.xml',FDTD,CSX);

% view defined structure
CSXGeomPlot('tmp/tmp.xml');

% run openEMS simulation
RunOpenEMS('tmp','tmp.xml','');
```

Results



B.3 865MHz Vivaldi Antenna

Octave Script

```
% init and define FDTD parameter
FDTD = InitFDTD('NrTS', 10000, 'EndCriteria', 0);
FDTD = SetSinusExcite(FDTD, 865e6);
BC = {'MUR' 'MUR' 'MUR' 'MUR' 'MUR' 'MUR'};
FDTD = SetBoundaryCond(FDTD,BC);

% init and define FDTD mesh
CSX = InitCSX();
mesh.x= -0.1 : 0.0005 : 0.1;
mesh.y= -0.1 : 0.0005 : 0.1;
mesh.z= -0.005 : 0.0005 : 0.006;
CSX = DefineRectGrid(CSX, 1, mesh );

% define the excitation port
[CSX port] = AddLumpedPort(CSX, 5 ,1 , 50, [-0.001 -0.01 0], [0.001 0.01 0.002], [0 1 0], true);

% define the antenna structure
CSX = AddMetal(CSX, 'metal');

exp_x = -3:0.1:3;
exp_y = exp(exp_x);
p1 = exp_x .* 0.02;
p2 = exp_y .* 0.002;
p1(end+1) = -3*0.02;
p2(end+1) = exp(3) * 0.002;
p1 = [p1, flip(p1)];
p2 = [p2, flip(-p2)];
p = [p1; p2];

CSX = AddLinPoly(CSX, 'metal', 1, 2, 0.0, p , 0.001, 'CoordSystem', 0);
CSX = AddBox(CSX, 'metal', 2, [-3*0.02, exp(3) * 0.002, 0], [-0.02, -exp(3) * 0.002, 0.001]);

CSX = AddMaterial(CSX,'air');
CSX = SetMaterialProperty( CSX, 'air', 'Epsilon', 1, 'Mue', 1 );
CSX = AddCylinder(CSX, 'air', 4, [-0.02 0 -0.001], [-0.02 0 0.002], 0.005);

% define a time domain e-field dump box
CSX = AddDump(CSX,'Et','DumpMode',0, 'DumpType', 0);
CSX = AddBox(CSX, 'Et', 0, [-0.1, -0.1, -0.001], [0.1, 0.1, 0.001]);

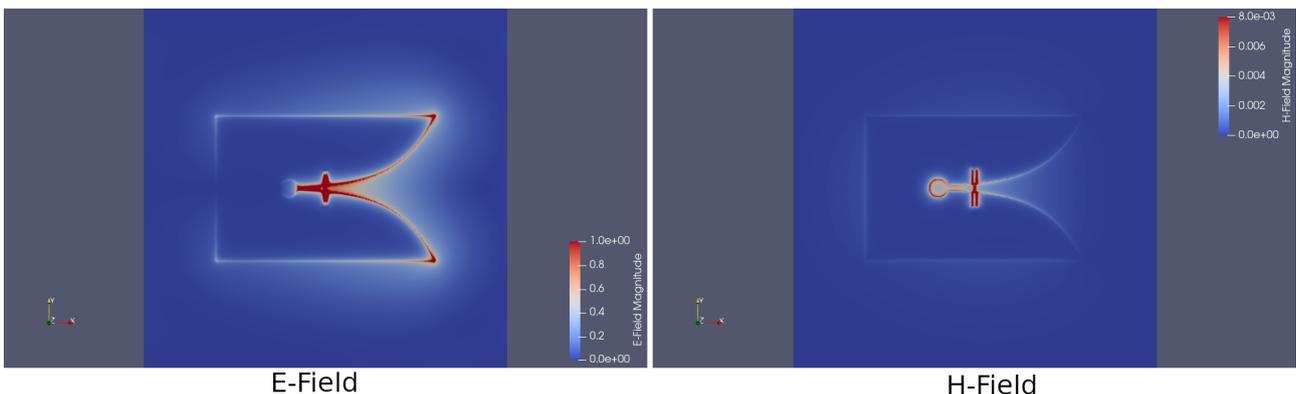
CSX = AddDump(CSX,'Ht','DumpMode',0, 'DumpType', 1);
CSX = AddBox(CSX, 'Ht', 0, [-0.1, -0.1, -0.001], [0.1, 0.1, 0.001]);

% write openEMS xml data file
WriteOpenEMS('tmp/tmp.xml',FDTD,CSX);

% view defined structure
CSXGeomPlot( 'tmp/tmp.xml' );

% run openEMS simulation
RunOpenEMS('tmp','tmp.xml','');
```

Results



Bibliography

- [1] *Anatomy of a tree*. URL: <https://www.trees-sa.co.za/2018/04/23/anatomy-of-a-tree/>.
- [2] G. J. Burke et al. “Numerical Electromagnetic Code (NEC)”. In: (1979), pp. 1–3. DOI: 10.1109/ISEMC.1979.7568787.
- [3] V. Chawla and D. S. Ha. “An overview of passive RFID”. In: *IEEE Communications Magazine* 45.9 (2007), pp. 11–17. DOI: 10.1109/MCOM.2007.4342873.
- [4] *Finite-Difference Time-Domain Method2*. URL: <https://www.synopsys.com/glossary/what-is-fdtd.html>.
- [5] *GNU Radio*. URL: https://wiki.gnuradio.org/index.php/Main_Page.
- [6] Murata’s MAGICSTRAP. “A UHF RFID Printed Circuit Board Solution”. In: (2012).
- [7] P. V. Nikitin, K. V. S. Rao, and S. Lazar. “An Overview of Near Field UHF RFID”. In: (2007), pp. 167–174. DOI: 10.1109/RFID.2007.346165.
- [8] *Nvidia Jetson*. URL: <https://www.nvidia.com/it-it/autonomous-machines/embedded-systems/#>.
- [9] *openEMS*. URL: <http://openems.de/start/index.php>.
- [10] *Raspberry Pi*. URL: <https://www.raspberrypi.org/>.
- [11] *Riferimenti delle costanti dielettriche per i prodotti Magtech*. URL: <https://www.emerson.com/it-it/automation/brands/magtech/dielectric-constants-reference>.
- [12] *RTL-SDR*. URL: <https://www.rtl-sdr.com>.
- [13] *Wire Antenna Modelling with NEC-2*. URL: <https://www.qsl.net/4nec2/Tutorial.pdf>.
- [14] David C. Wyld. “RFID 101: the next big thing for management”. In: *Management Research News, Vol. 29 No. 4* (2006). DOI: 10.1108/01409170610665022.