Fagundes Renan
Quiniou Thierry

Report

RFID spoofing with a DVB-T device

ENSTA Bretagne

16/12/2013

Fagundes Renan
Quiniou Thierry

Table of Contents

Fagundes Renan
Quiniou Thierry

1 Introduction

RFID is the wireless non-contact use of radio-frequency electromagnetic fields to transfer data. It is widely used in security access and public transport. The spoofing project is a research job whose aim is to get the RFID signal and to jam it. Previous work demonstrates some vulnerabilities but they often require expensive equipments. The innovation about this research is to use a cheap device originally intended to receive numeric television. We discuss in this report about hardware specifications and different processing of measured data. A technical background and a state of the art are included in Annex.

2 Project description

The RFID spoofing project was not the initial aim in so far as the goal was to realize a Software Defined Radio (SDR) demonstrator with a DVB-T device.

2.1 *Initial project*

The idea for this topic was to exploit the discovery of researchers who have found that a chip available for a portable decoder digital television (DVB-T) could be reused to capture the surrounding spectrum between 60 MHz and 1.7 GHz. The objective was to create a fun and easy demonstrator for the next ENSTA Bretagne Open Day. Thus, we began to experiment with existing software with a key that we have acquired by our own means. After a day of experiments, we obtained an effective demonstrator.

We quickly realized that this topic was limited by the fact that most of the possible acquisitions had already been made. Nonetheless, we plan to install the software at the Open Day to show the public what students are able to do with a common 20 Euro device.
In addition, Mr. Mansour's absence blocked our progress. However Mr. Le Roy was able to propose a new topic which was then validated by Mr. Mansour. This new project consists to achieve a RFID spoofing demonstrator. Such system generates a scramble signal between an RFID chip and reader to jam the genuine signal and simulate another ID. Thus allowing it to impersonate a chip whose identifier is supposedly unique.

2.2 *RFID spoofing project description*

A spoofing attack can be described as a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage. This kind of attack uses a famous technique in computer security named the Man In The Middle (MITM). This is a form of active eavesdropping in which the attacker make independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker. Figure 1 illustrates this attack.

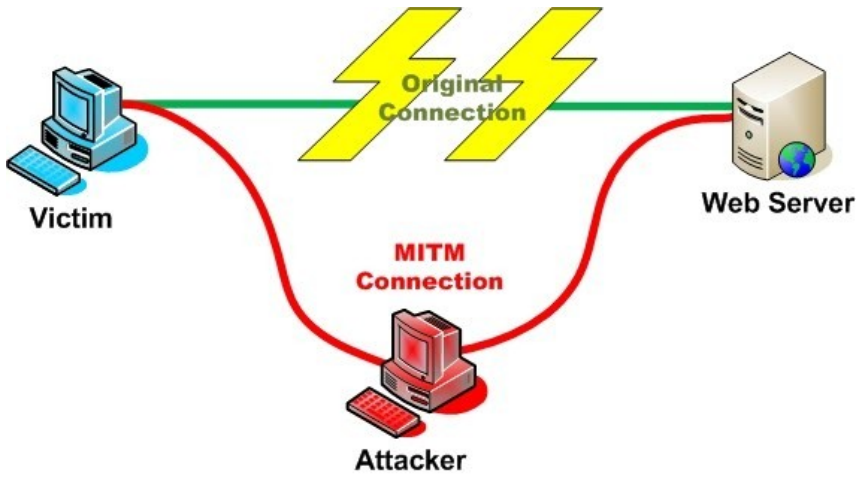


Figure 1 : The Man In The Middle Attack (MITM)

The spoofing attack uses a similar technique but the attacker does not need to control the entire connection. The spoofing attack involves impersonating a victim by eavesdropping and jamming whereas the MITM attack seals an ID in order to spy and send malicious information. So, a RFID spoofing project can be illustrate by the Figure 2.

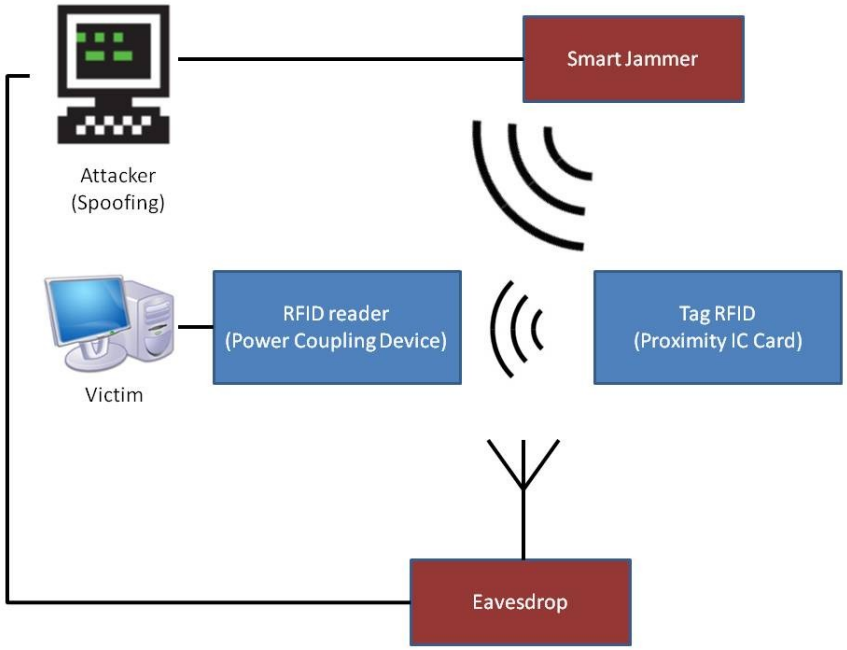


Figure 2 : RFID spoofing project

The main purpose is to impersonate a chip whose identifier is unique. Blue represents the victim system which contains a RFID tag and a reader plug into a computer. RFID technology is based on close communication of several centimeters and it uses a 13.56 MHz frequency. The idea is to jam the RFID reader (PCD) with a "smart signal" in order to avoid the tag (PICC) being detected Nonetheless, this reader emits signal periodically to detect tags that emit a response when this occurs. A second listening antenna can be used to read the information emitted by the tag and indentify it. The "smart jam" allows useful information to be extracted of the noise. Consequently, the smart jammer can emit another response with a different ID. In conclusion, the reader would detect a tag with an ID which is different from the legitimate chip's ID.

3 Hardware presentation

3.1 *DVB-T hardware description*

In this project is used a DVB-T receiver (Digital Video Broadcasting – Terrestrial) to receiver the signal of RFID. This device is based on RTL2832U chip and a Front-End Radio receiver chip (only receive function). This receiver was chosen because it is low cost. The image below show the disassemble device.

Figure 3 : DVB-T device disassemble

3.1.1 Diagram Block of the DVB-T USB Dongle

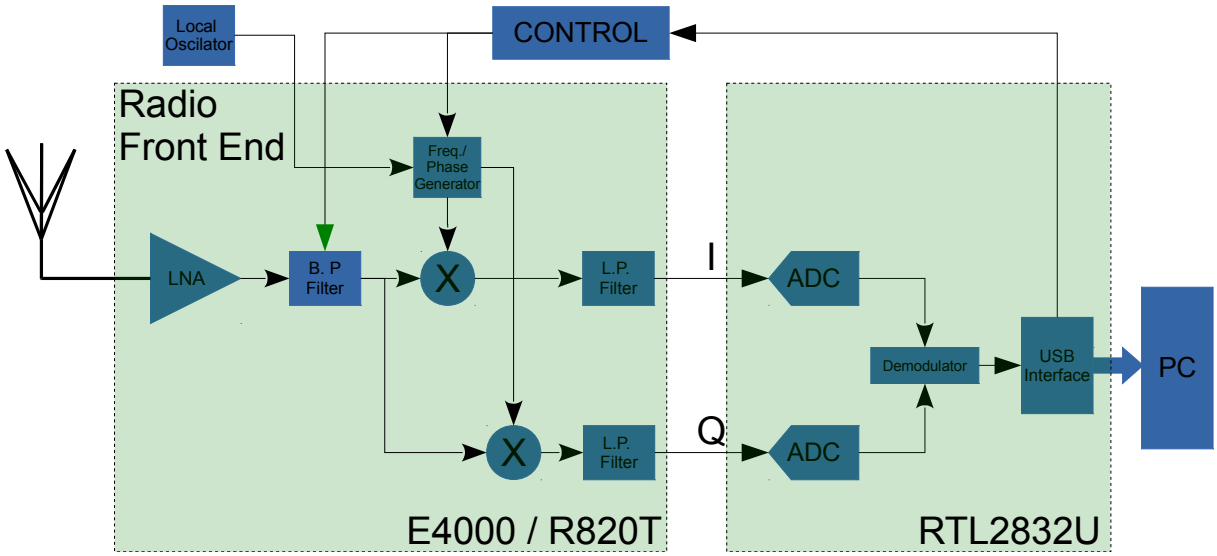


Figure 4 : Block Diagram DVB-T

The figure 10 is a block diagram of the DVB-T system. It is composed basically of two chips, one Front-End and one of ADC, demodulation and USB, as shown in the figure above.

3.1.2 How it works

- LNA :

The Low Noise Amplifier applies a large gain on the signal from antenna, but due to the relatively low impedance of parasitic capacitances at RF frequencies, the LNA gain is limited, because high gain in a RF amplifier may cause a positive feedback and oscillation.

- Band Pass Filter :

The band pass filter or preselector is a programmable filter which selects the concerned band. The signal from the antenna is wide-band, that means that it takes signals near to the zero frequency until spectrum very high. This block attenuates every signal of frequency out the band of pass, this leave the low frequencies clean. This block is necessary before the mixer because it makes the down-conversion that it means that the mixer moves the signal of concerned band to the zero frequency.

- MIXER :

The mixer is the principal component in the superheterodyne receiver architecture. The mixer multiplies the RF signals by a tunable signal from a local oscillator and outputs the sum and the difference of f_{rf} and f_{osc}. In this case the difference is retained.

The radio signals are Quadrature Signals; they are based on the complex numbers, so this kind of signal are bidimensional because it always has two values. We need to process a Real part and an Imaginary part which are equivalent with a module and a phase. In this case, we use the terms of communication engineering « In-Phase » and « Quadrature-Phase ». The processing signal separates the In-Phase (I) and Quadrature (Q). These components are extracted through two mixers which multiply the signal by sin(2πft) and cos(2πft). Then, noise components are filtered and the signal is digitized with an Analog to Digital Converter (ADC).

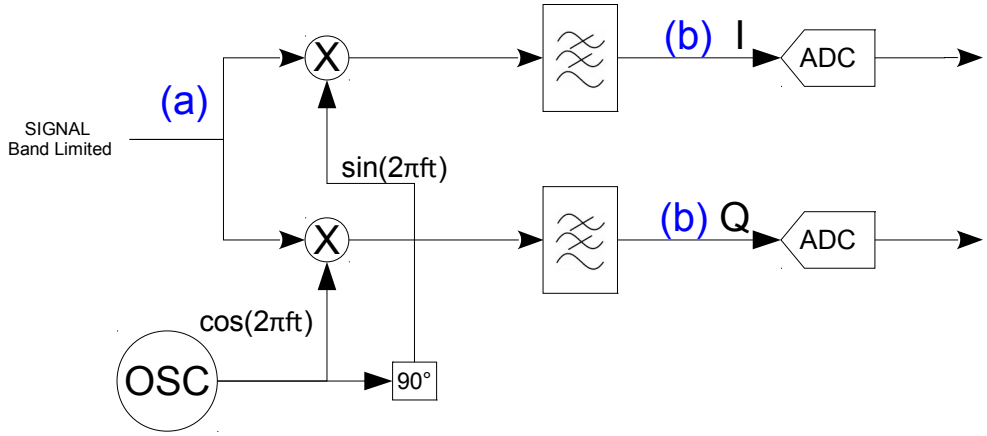


Figure 5 : Functions of Mixer

- Filter

The last filter has two functions. It is used for anti-aliasing filtering just before the ADC and to cut-off the portion of spectrum of sum of f_{rf} and f_{osc} .

3.1.3 RADIO FRONT-END RECEIVER

In the market of chip receivers of low cost there is two chips radio front-end are available. The E4000 and E820T chips are described in this section.

- E4000
  This chip is a complete Radio Front End, it has filters, mixers and the LNA. Below is the block diagram from the Datasheet of the component.

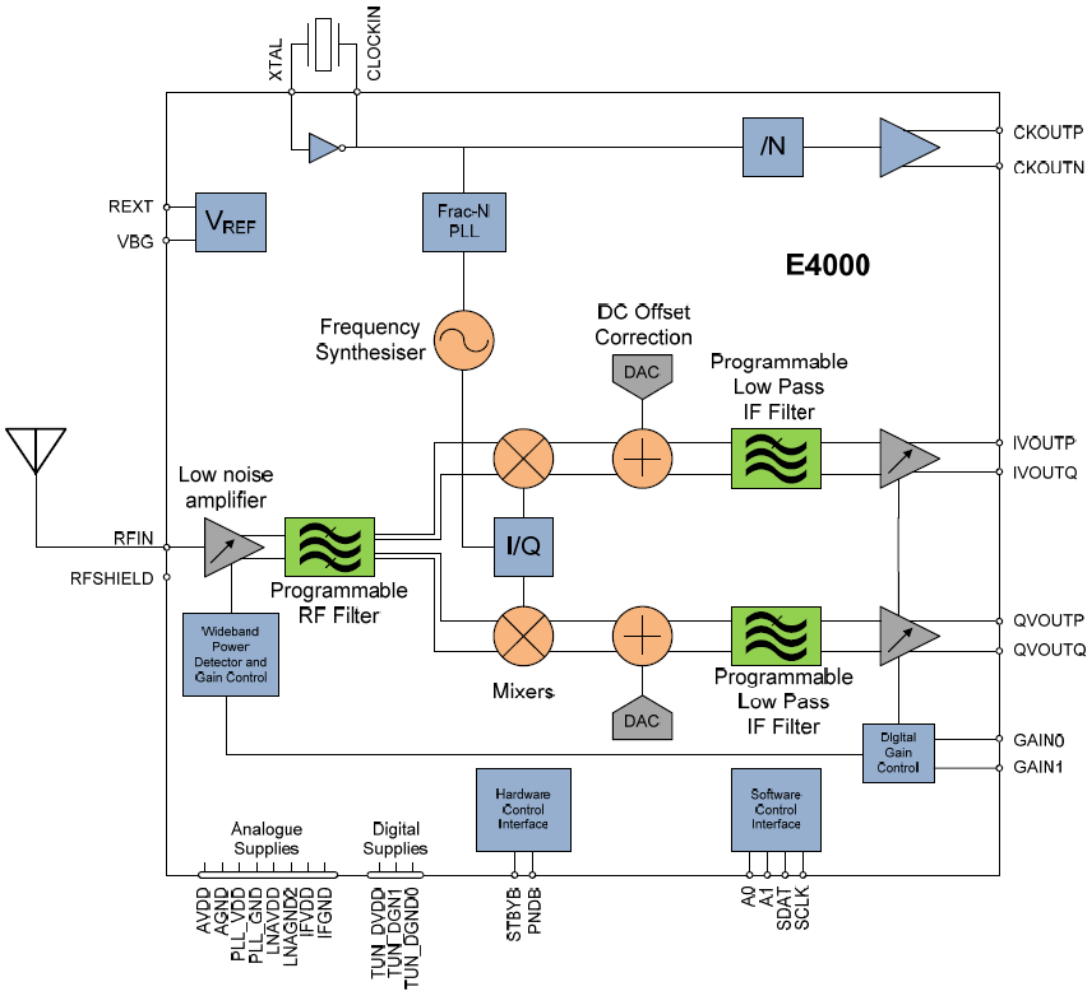


Figure 6 - Block Diagram of E4000

Fagundes Renan
Quiniou Thierry

- R820T

R820T as the E4000 has every function of Front-end in a chip, figure 13 shows the R820T diagram from the datasheet and which operates in the frequency range (42 ~1002 MHZ). The main difference with the E400 is the output which isn't centered in zero Hertz, but translated into 4MHz.

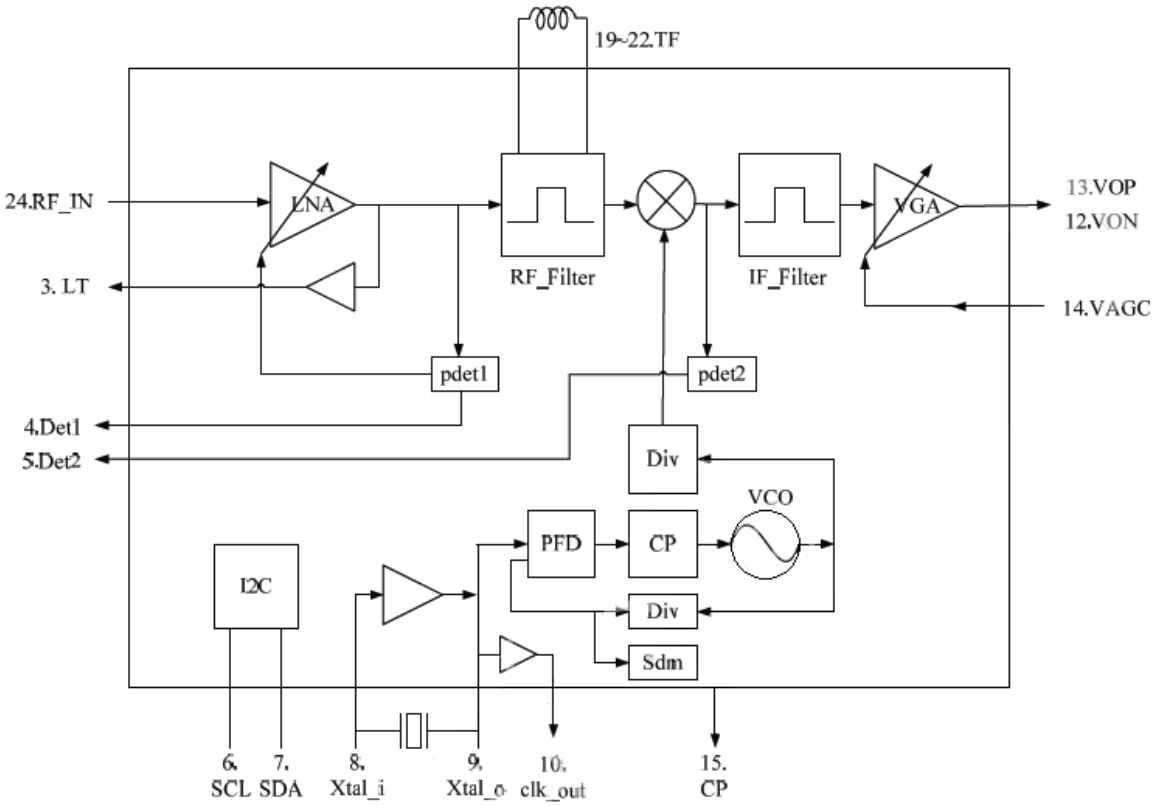


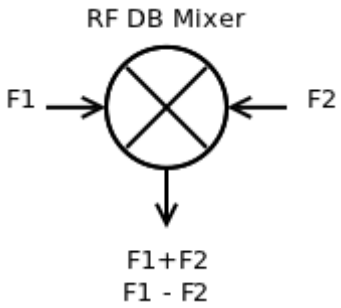
Figure 7 - Block diagram of R820T

3.2 *Frequency transposition*

A challenge for RFID signal acquisition is to use a DVB-T device commonly used to receive the TNT. Actually, this cheap device can be used to sniff the electromagnetic field and in this case RFID frames.

Nonetheless, the RTL 2832U broadband is 60 MHz - 1,7 GHz and the ISO 14443 standard uses the 13.56 MHz frequency. A common practice is to transpose the received signal with a mixer in order to convert the genuine frequency from a higher. This kind of device is named an up converter because it converts the frequency from “down to up”.

This diagram describes what a mixer does because it adds and subtracts two frequencies



There were two possibilities. We could either create our own circuit and designed our Printed Circuit Board (PCB) or buy an existing device[CITATION 13122 \l 1036]. After research and a test[CITATION 13121 \l 1036] of the RF converter manufactured by *NooElec*, the decision was made that the Ham It Up v1.2 would be used[CITATION 13123 \l 1036].

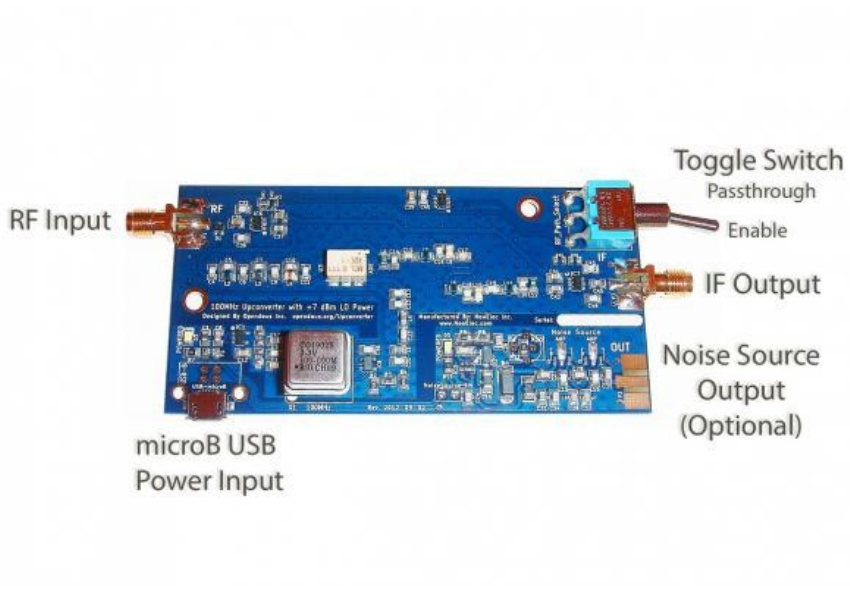


Figure 8 : RF converter : NooElec Ham It Up v1.2

Specifications tests were made by amateur in [CITATION 13121 \l 1036] but it was necessary to evaluate it in real conditions. A careful reading of previous tests has shown that the methodology was wrong. M. Lababidi explains us that specifications cannot be measured with a Vector Network

Analyzer (VNA) because the frequency measured in VNA the same in input and output. This device adds 125 MHz at the input frequency, so a VNA is not appropriate for this test. The test bench used for measurements is illustrated in Figure 9.

Figure 9 : Up Converter Test Bench

The Spectrum Analyzer was set in “max hold” and the signal generator swept linearly the frequency between 1 and 100 MHz. So, we get the response of the converter according to the input signal.

Results are illustrated in Figure 10 and Figure 11.

Figure 10 : Up Converter measurements (Input: 13.56 MHz)

Figure 11 : Up Converter Measurements (Input : 63 MHz)

A previous measurement gave a 1.4 dB cables losses due to the poor quality of cables. We can see in Figure 10 a loss of 12.6 dB at 141.56 MHz. This frequency corresponds to a 13.56 MHZ input frequency because 13.56 plus 125 equals 141.56. So the loss of the up converter in the bandwidth of RFID HF is 11.2 dB (Up converter losses = Measure – cables losses).

We can see in Figure 11 that the up converter has upconversion losses of approximately 11 dB between 1 and 66 MHz. This results are consistent with the [CITATION 13121 \l 1036] source.

The up converter allows us to realize RFID signal acquisition even if an acquisition with a higher level would have been more comfortable.

3.3 *Homemade antenna*

RFID HF protocol uses the 13.56 MHz frequency and common aerials are built for higher frequencies. While awaiting the material reception, we decided to build our own RFID antenna (Figure 12) using a web article as inspiration [CITATION RFI \l 1036].

Fagundes Renan
Quiniou Thierry

Figure 12 : Homemade loop antenna

The antenna was made with an insulated copper wire (a piece of RJ45 cable) which was rolled round a can of cola. A rough test demonstrated its efficiency at 8 centimeters range whereas the Vector Network Analyzer (VNA) measured bad results regarding its coefficient reflection (Figure 13).

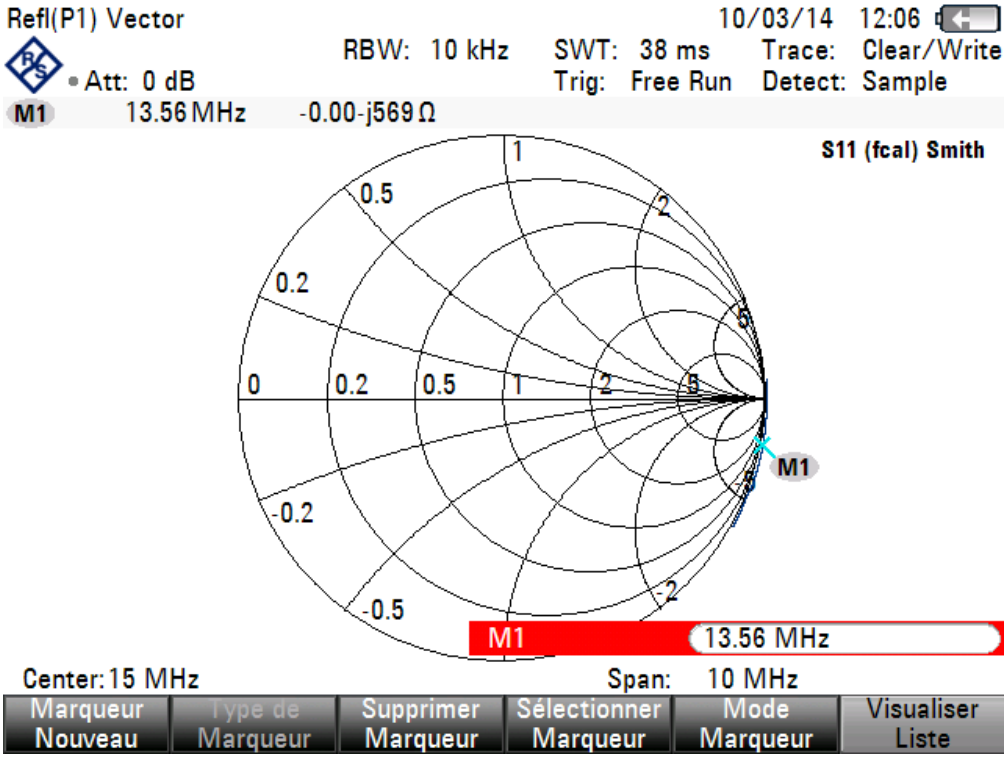


Figure 13 : Homemade antenna S11

Actually, the measure put on a Smith Abacus shows a total impedance mismatching.

We though building an impedance matching circuit with a “pi” or “L” configuration but the impedance is not fixed and a slight deformation of the loop changes it strongly. An idea was to add a 50 ohm resistance but the signal would greatly reduce. This kind of antenna can be used for PCD signal measurement which is strong but it is useless for low signal acquisition like PICC.

Conversely, manufactured antennas (Figure 14) have an efficient coefficient reflection (Figure 15) and the range of PCD signal acquisition is approximately the same of our homemade one (8 centimeters). Nonetheless, it is able to acquire the useful PICC signal contrary to the homemade antenna. Thus, all measurements in the project have been with the manufactured antenna.

Figure 14 : Manufactured antenna

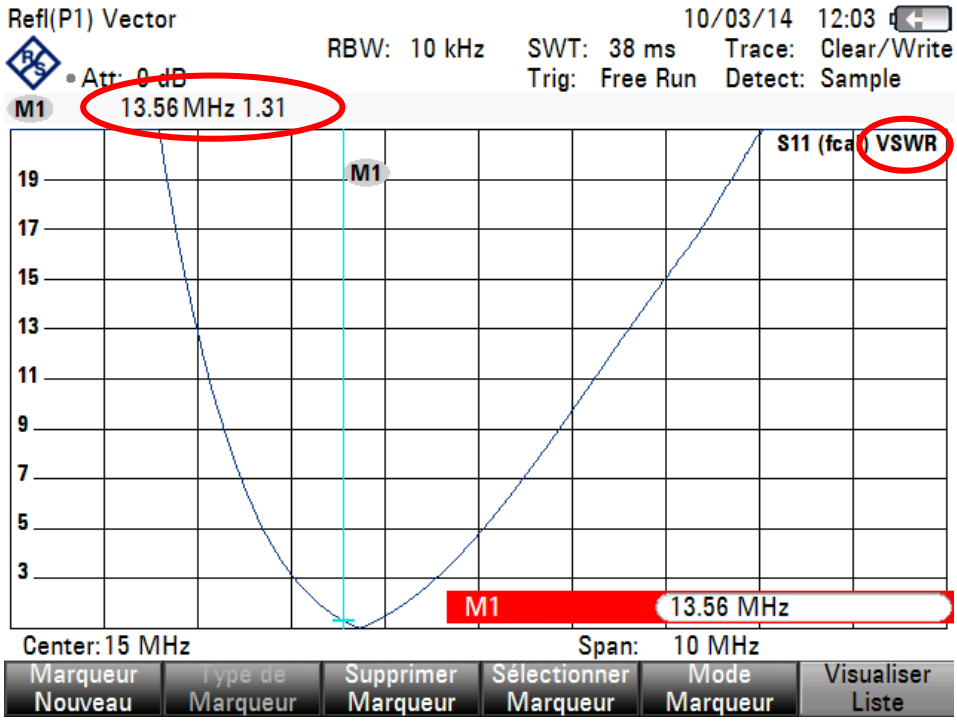


Figure 15 : Manufactured antenna coefficient reflection measurement (S11)

4 Signal recording

Signal recording has been realized with the up converter and the R820t dongle. A simple *Gnuradio Companion* program (Figure 16) allowed us to store information and replayed it for further processing.

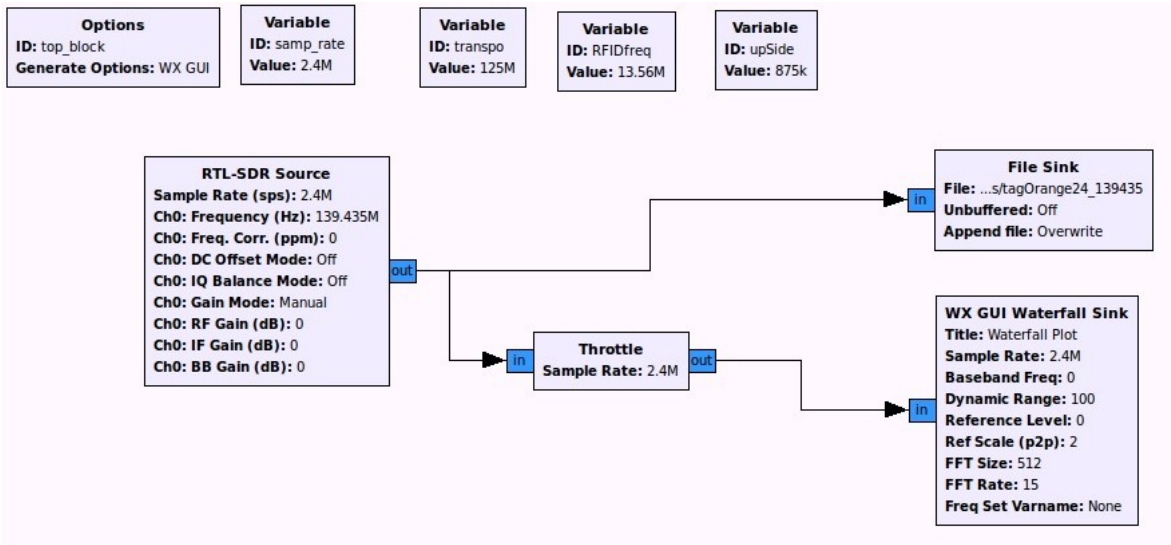


Figure 16 : Recording Gnuradio Companion program

The recording frequency is centred on the PICC response as demonstrated below.

$$Ch\,0 : Frequency = Transposition + Carrier\ frequency + sideband$$

$$Ch\,0 : Frequency = 125\,MHz + 13.56\,MHZ + 847.5\,kHz = 139.4075\,MHz$$

The signal recorded is shown on the Figure 17 with a waterfall illustration. The vertical axis is the time and the abscissa the frequency. The colour scale illustrates the signal level and we can see the non continuous PICC communication. The red frequency on the left is the strong carrier frequency at 13.56 MHz.

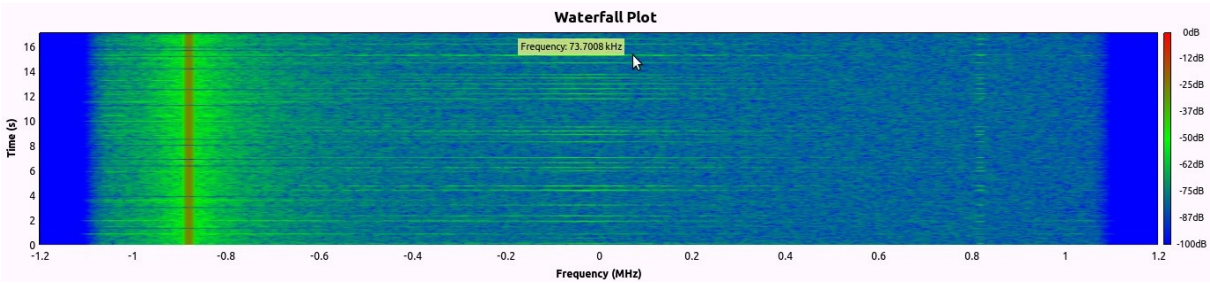


Figure 17 : Waterfall of PICC communications

We can notice the low power of PICC emission because of passive components inside the tags. It has been not easy to capture and process a -60 dB signal.

5 PICC signal demodulation with *Matlab*

The PICC communicates with the PCD with a binary phase shift keying protocol (BPSK). The modulated frequencies are 14.408 kHz and 12,712 kHz (see Figure 56 in 11.2.4). The BPSK modulation use the phase to transmit information. Two phases are used and each one is assigned at a symbol (0 or 1). The Figure 18 illustrates the signal according to the digit transmit.

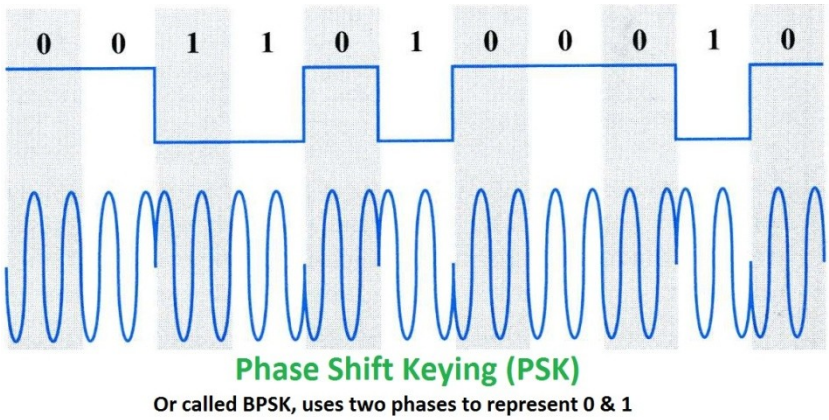


Figure 18 : : Binary Phase Shift Keying modulation (BPSK)

5.1.1 Raw signal

The raw signal acquired with *Gnuradio Companion* is recorded in a data file and it can be replayed with *Matlab*. The Figure 19 shows the raw signal.

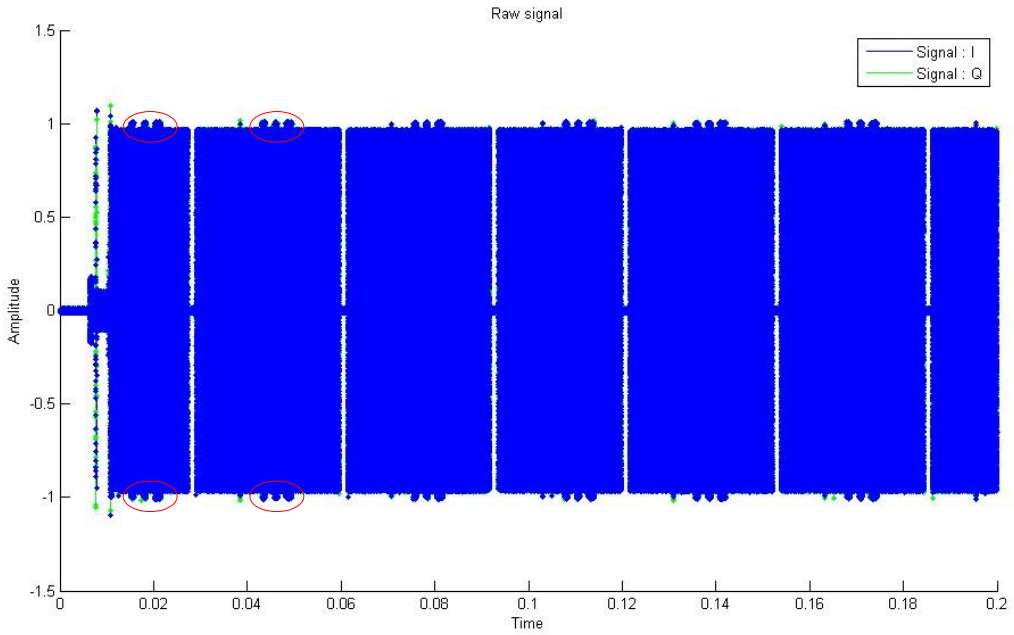


Figure 19 : Raw signal recored with Gnuradio Companion

We can see that the PCD emission is not continuous and the PICC information is contained in red circles. A zoom around the useful information (Figure 20) allow us to see more specifically the emission structure.

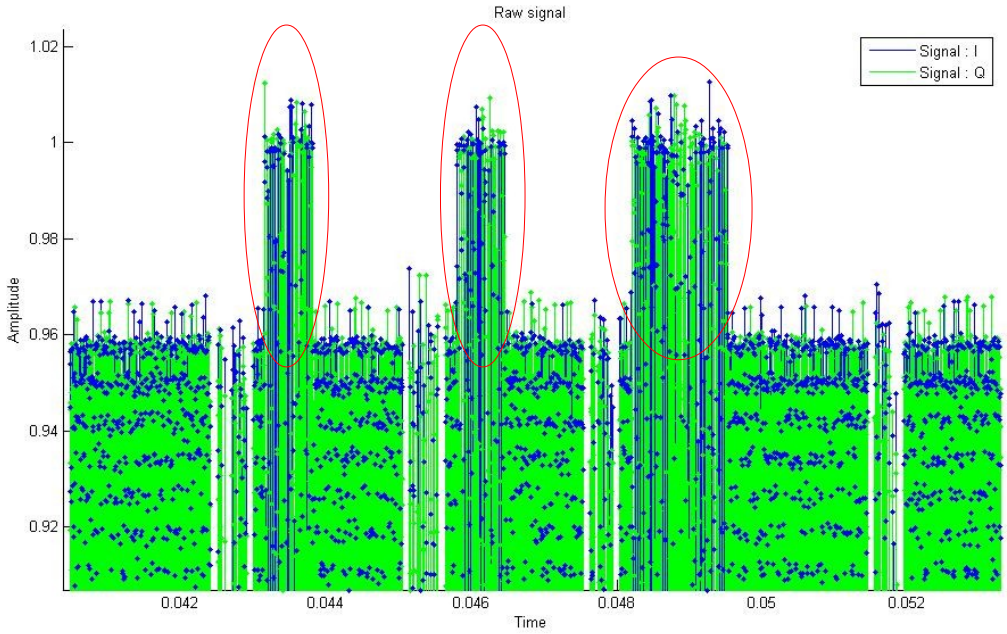


Figure 20 : Zoom on PICC raw signal

The PICC emit periodically 3 pulses and the last is longer than the others. So, the demodulation focused on these three pulses.

5.1.2 Signal filtering

A fast Fourier transformation allows us to pick up the frequencies contained in the signal. The recording has been made with the R820t dongle and it transposes the signal in base band. So, the frequencies which contain our signal are around to zero. The Figure 21 shows the FFT of the raw signal and the red curve represent the impulse response of the filter. We used a Butterworth filter of 6^{th} order and we remember that the absolute value of an impulse response is retained for a filter.

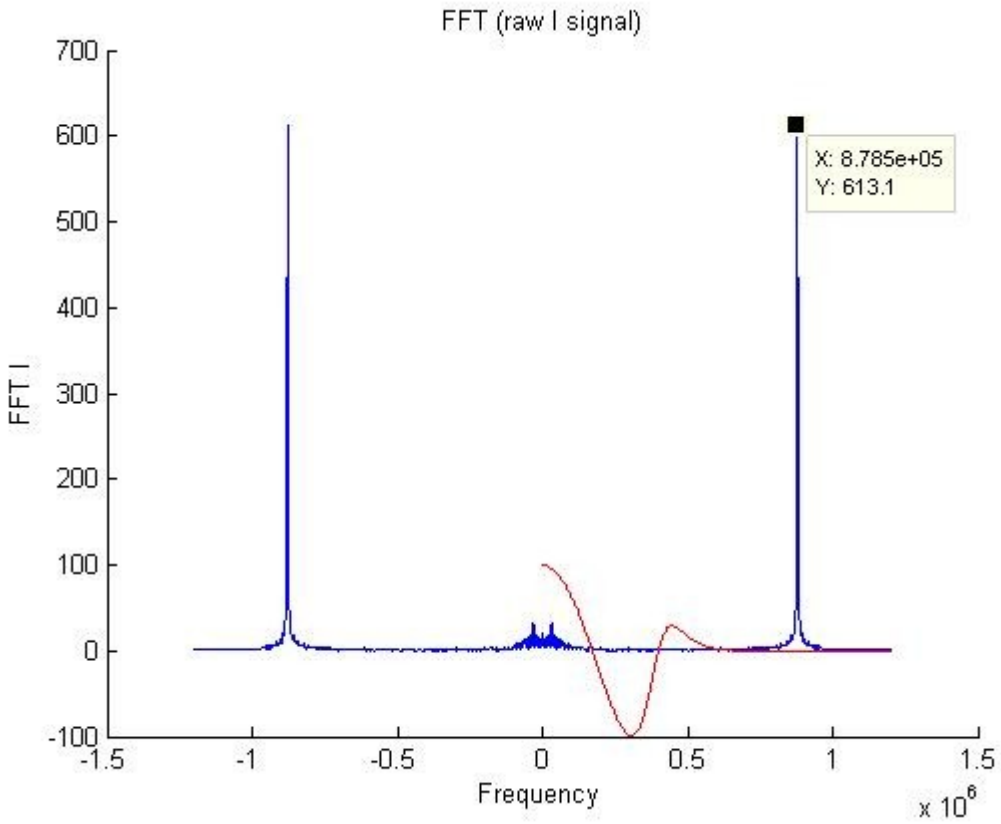


Figure 21 : Signal FFT and impulse response of the filter

The FFT of the signal after filtering contains the useful information in baseband.

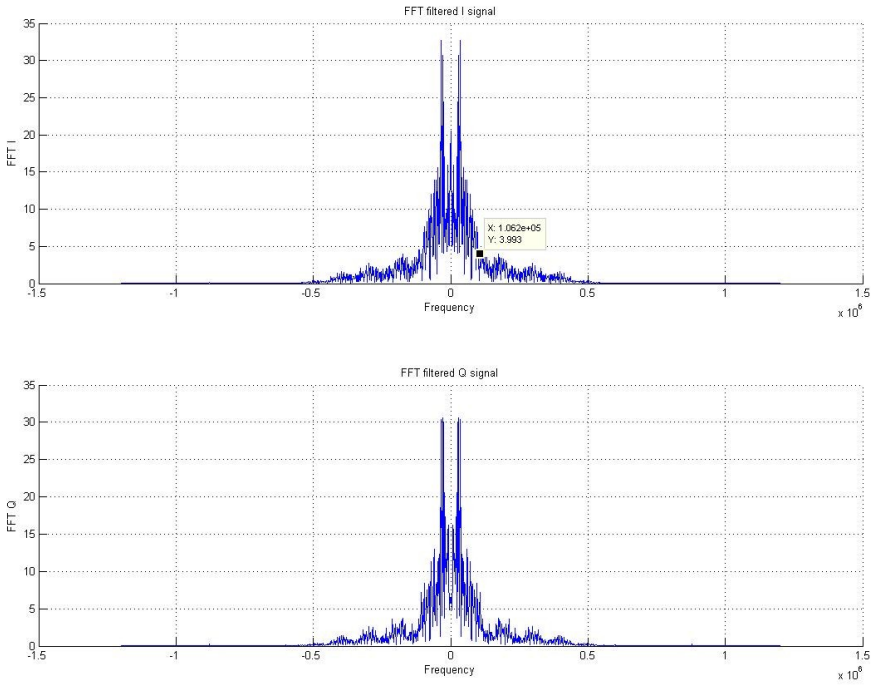


Figure 22 : Filtered signal I and Q

We can notice the side lobes which correspond to the squared cardinal sine which describe the BPSK power spectral density (PSD) illustrated in Figure 23.

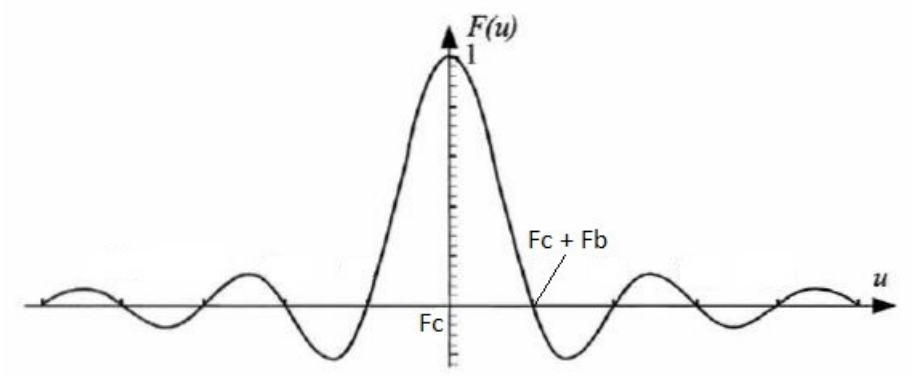


Figure 23 : BPSK power spectral density

The carrier frequency (Fc) of our signal is 14,408 kHz but it is transposed to zero (base band signal). The ISO/IEC 1443 for type B claims that the subcarrier frequency is 847.5 kHz and each bit period is 8 subcarrier periods long. So, the band frequency Fb can be determined like this:

$$F_b = \frac{F_{subcarrier}}{8} = \frac{847.5\,kHz}{8} = 105.94\,kHz$$

So, the main lobe width of approximately 200 kHz in Figure 17 is in accordance with theory.

An elementary time unit (ETU) which represent a symbol duration is $\frac{1}{105.94\,kHz} = 9.439\,\mu s$. The sampling frequency F_s used for acquisition is 2.4 MHz. We can easily deduce the number of samples per symbol $N_{samples/symbol}$.

$$N_{samples/symbol} = \frac{F_s}{F_b} \backsimeq 22$$

This number of samples per symbol is very comfortable and we could reduce it in the future in order to reduce the computation time.

5.1.3 Synchronization

A BPSK constellation represents the two different phases spaced of pi. Nonetheless, this constellation turns currently during transmission because of effects of the propagation channel. The constellation in Figure 24 contains the signal but the rotation creates a circle. It is not possible to extract information because the decision threshold is fixed in BPSK demodulation.

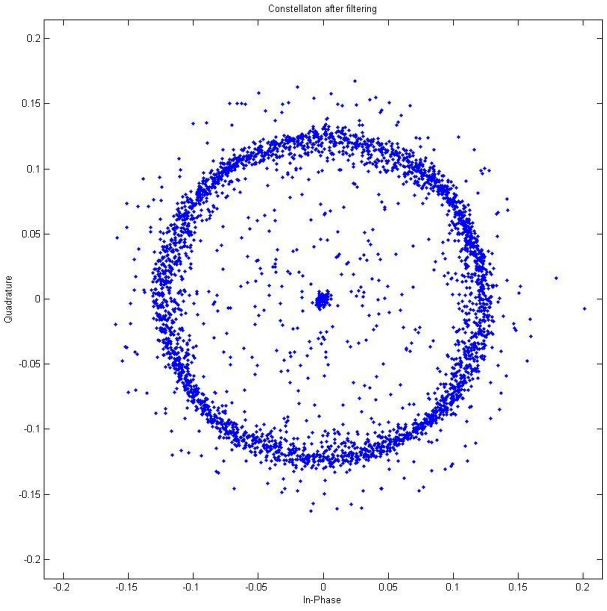


Figure 24 : Constellation before synchronization

A way to synchronize the signal was to apply a contra rotary signal in order to compensate the desynchronization. An exponential multiplication achieved to reach a null rotational speed.

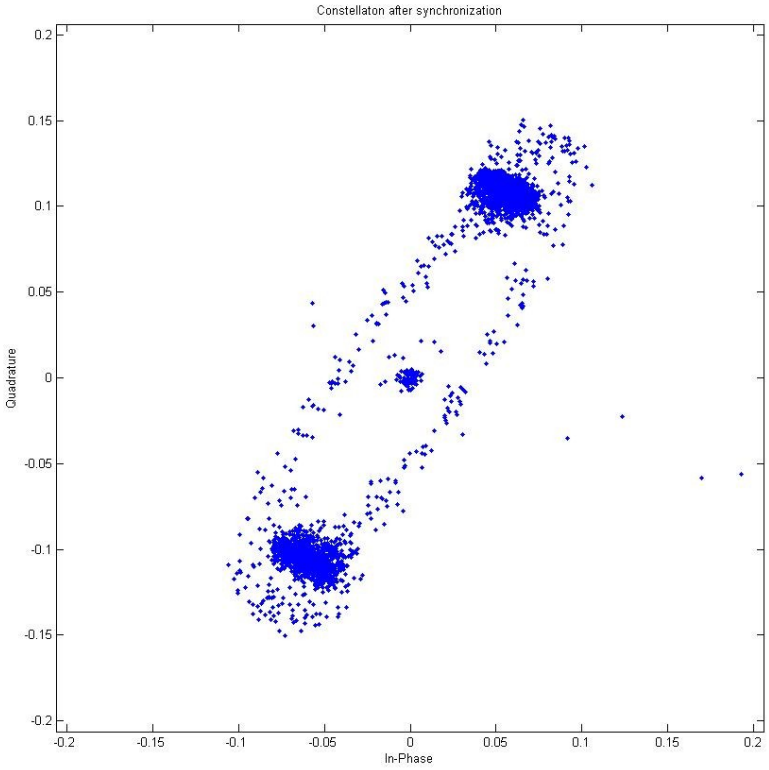


Figure 25 : Synchronized signal

The contra rotational speed applied is 1.33 % of the carrier frequency and this value is commonly observed in channels propagations. Points out of poles are not noise but different values during the phase transition. The sampling frequency is so high that we detect the phase transitions.

The initial phase between the real axis and the poles are not fixed because the sampling frequency is not a multiple of Fb. So, the initial phase cannot be predicted and it is necessary to rotate the synchronized constellation in order to put it on the abscissa axis.

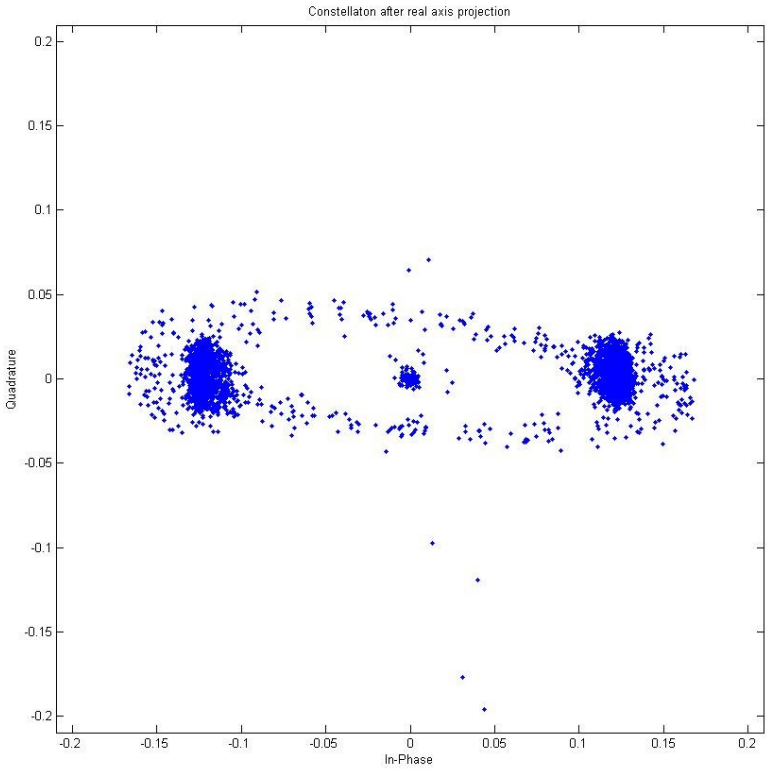


Figure 26 : Synchronized constellation with null initial phase

5.1.4 Demodulation

The BPSK demodulation is commonly used and a *Matlab* function allows us to extract binary data. The binary stream contains numerous digits because each symbol is represented by 22 identical digits ($N_{samples/symbol}$).

After processing, we obtained the final binary information but we have a phase ambiguity. Actually, we are not able to predict if the pi phase represent the '0' or the '1' symbol. So, we used an intercorrelation to detect the start of frames and invert bits if necessary. The binary data are illustrated in Figure 27.

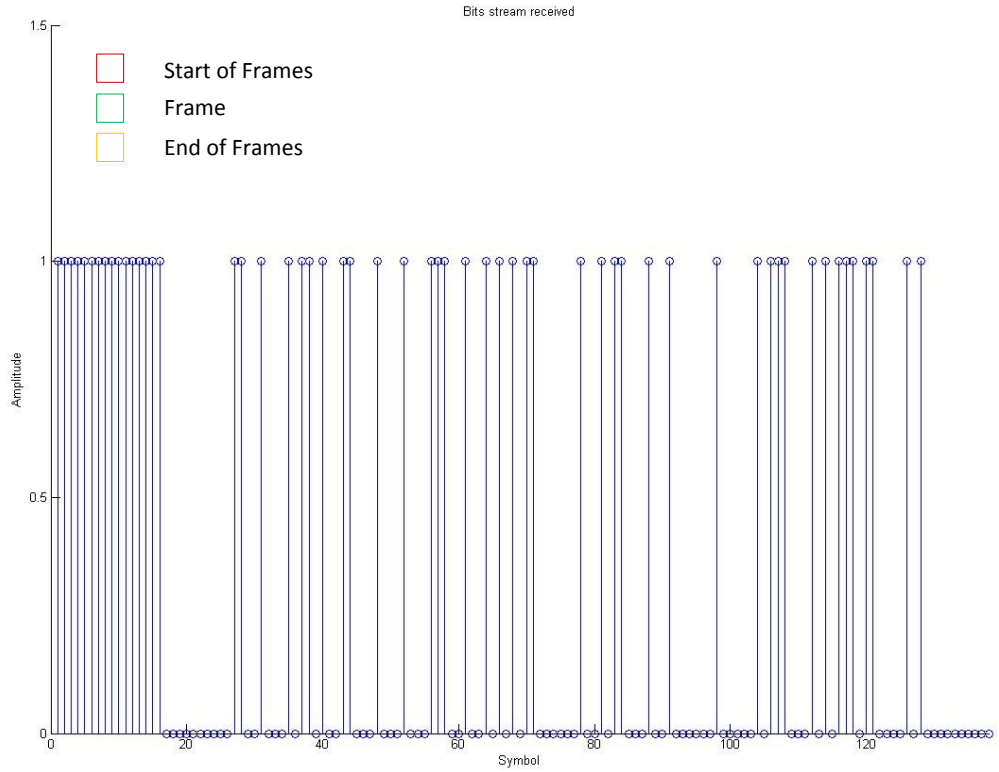


Figure 27 : Binary information

The interpretation needs an ISO/IEC 14443 type B understanding. The information is contains in frames and the protocol describes the start of frames. The Figure 28 shows the start of frames.

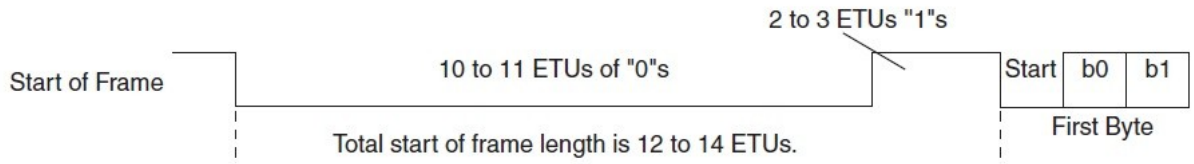


Figure 28 : Start of frames protocol

A comparison between the protocol and the signal allow us to find out the start of frames with ten '0' symbols and two '1'.

Each frame contains one low bit, eight data bits and one high bit. It is easy to separate frames and to extract information.

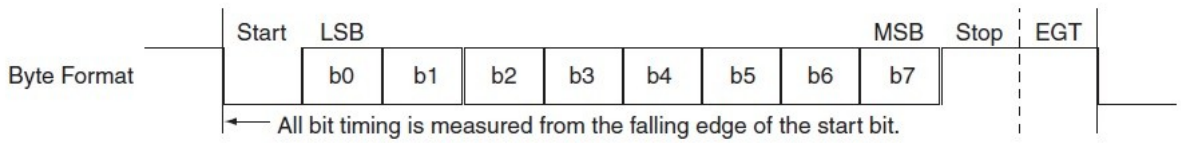


Figure 29 : Data format protocol

The end of frames is a sequence of ten ‘0’ symbols.

5.1.5 Reverse protocoling

Binary data extracted from the signal contains two identical short sequences of 3 bytes and a last one containing 10 bytes. The ISO/IEC 14443 type B protocol defines that the first byte contains the kind of message sent (named header) but it is not fixed in our case. We think that the reader uses a custom protocol and it is not necessary to send a header because just one data format is expected. We did not find out what the two short sequences represent but the last 10 bytes stream is understandable.

The ISO/IEC 14443 type B protocol uses a 16 bits error detecting code called CRC for cyclic redundancy check. Each sequence of one or several frames is terminated by a two bytes CRC code. We have applied the algorithm defined in the protocol and we demonstrated that is well used in our device. The two identical short sequences contain one relevant byte and two CRC bytes. The chart below illustrated these short sequences during a PICC-PCD communication.

Relevant bytes (yellow tag)	CRC code	Relevant bytes (pink tag)	CRC code
C4	50 70	39	3A 5C
C4	50 70	49	BD 2F
D4	D1 60	39	3A 5C
C5	D9 61	39	3A 5C
D5	58 71	49	BD 2F
C4	50 70	39	3A 5C
C4	50 70	39	3A 5C
D5	58 71	29	BB 4C
D5	58 71	39	3A 5C
C4	50 70	38	B3 4D

We did not find out what they mean and the information is different with other tags.

The PCD output gives a serial of data and we retrieve this information in the last sequence.

PCD sequence:

0013840 0102 0000 D008 1A02 5203 19C4 00A2 0000

Sniffed data in the last sequence:

A2 19 C4 52 03 1A 02 D0 D4 43

Tag color	Sniffed data
Pink	A2 19 C4 52 03 1A 02 D0 D4 43
Orange	1E 0D C4 52 03 1A 02 D0 63 84
Green	97 0D CF 40 A4 19 02 D0 17 76
Yellow	3A 47 C7 52 03 1A 02 D0 61 06
Orange bis	47 2B C2 52 03 1A 02 D0 5D 35

The grey highlight bytes are the CRC codes of each frame. We can notice that the first four bytes contain the ID. It is hard to find out the meaning of data because these data are not in accordance with the ISO/IEC 14443 type B protocol. These tags use the physical constraints of the protocol and

use the CRC code but data format is different. According to the protocol, the PICC can emit three commands which have a specific header but there is no header in sequences emitted by our tags.

Note: The mir:ror reader can read type A and type B PICC but an experiment showed that violet tags cannot be read by other RFID reader (arduino NFC module tested). The lack of header is probably the source of this incompatibility.

The custom protocol understanding is limited but the impersonation is still possible by replaying data. Nonetheless, it is necessary to pick up data emitted by the PCD (mir:ror) in order to understand what is information emitted during short sequences.

To synthesized, there is three PCD requests with ASK modulation and three answers of PICC with BPSK modulation. We remember that the short answer 2 is identical at the first one. We see on the Figure 30 the information inside the raw signal.

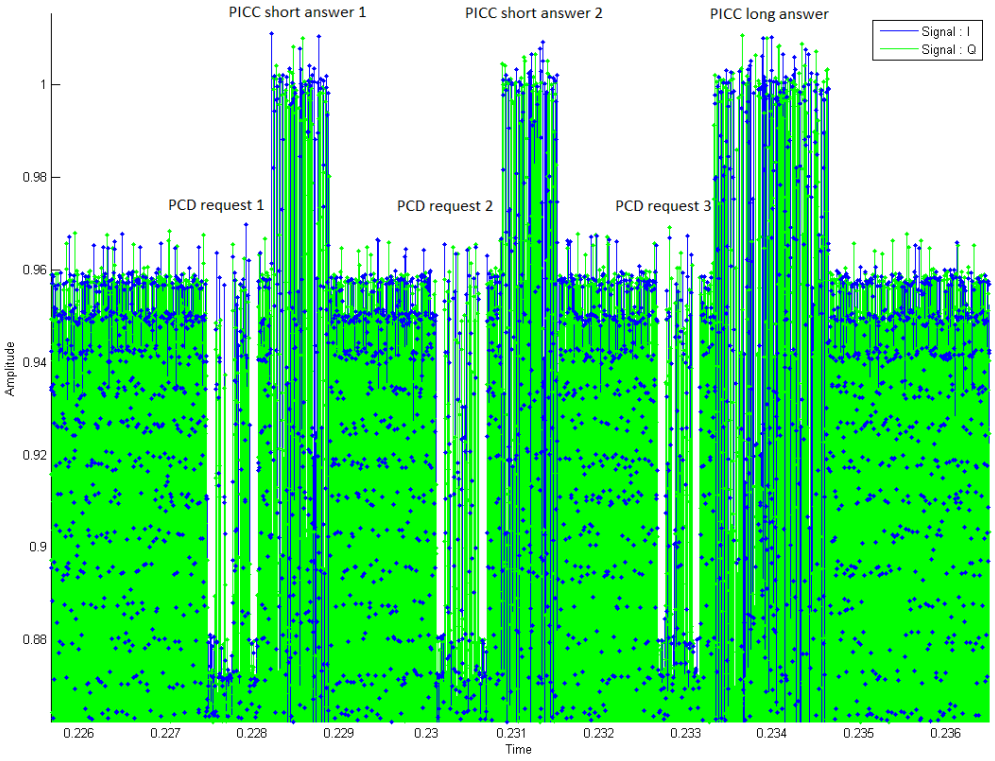


Figure 30 : Raw signal communication sequences

We did not know the content of PCD requests but it emits data. So, the *Violet* custom protocol looks like this.

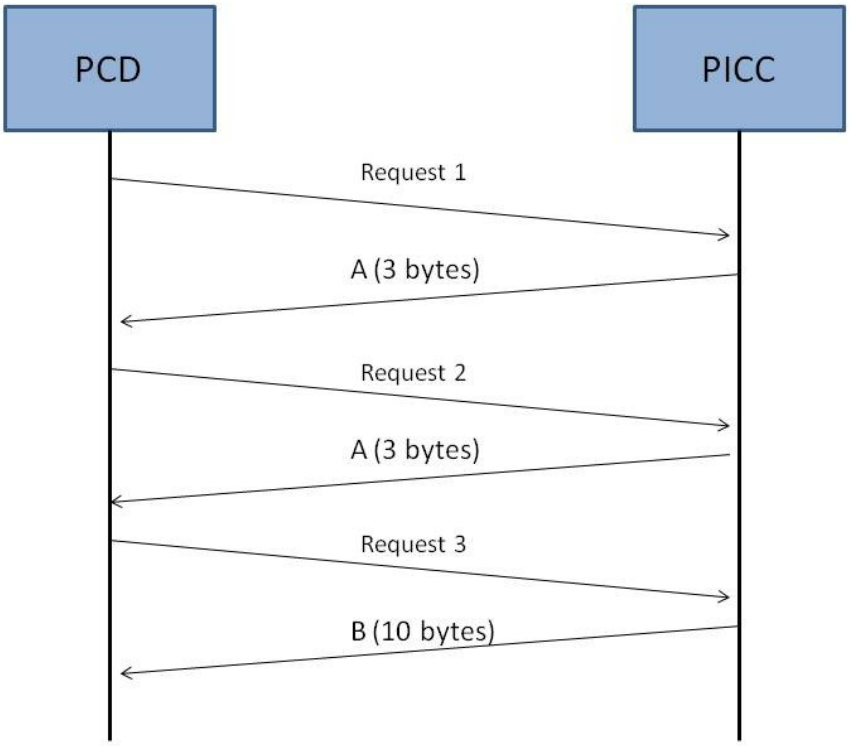


Figure 31 : *Violet* custom protocol

The utility of A messages are unknown but the bytes of B contents the useful information.

6 PICC signal demodulation with *Gnuradio Companion*

6.1 *Gnuradio presentation and DVB-T configuration*

GNU Radio is free software, the gnuradio.org defines the GNU radio as; *« GNU Radio is a free & 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 was developed in Python language, and is possible to write new blocks, and a complete application of GNU radio directly in Python. In this works it's utilized to processing the signal from the dongle. Here is described the steps of processing of the Signal RFID.

Gain configuration:

Before processing the signals is necessarily suit the signal from the front-end analogical. The block “SDR-RTL”, this block has the fields to configuration of LNA gain (RF gains) , pass-band filter (IF gain), and band base amplification (BB gain). Initially is necessarily to adjust the gain of radio front-end, because the values default are designed to receiver weak signals. If this configuration isn't done, the signal arrive saturated, because the signal in the output of ADC is limited in the interval [-1,+1]. The better ensemble of gain is show in the figure below.

Fagundes Renan
Quiniou Thierry

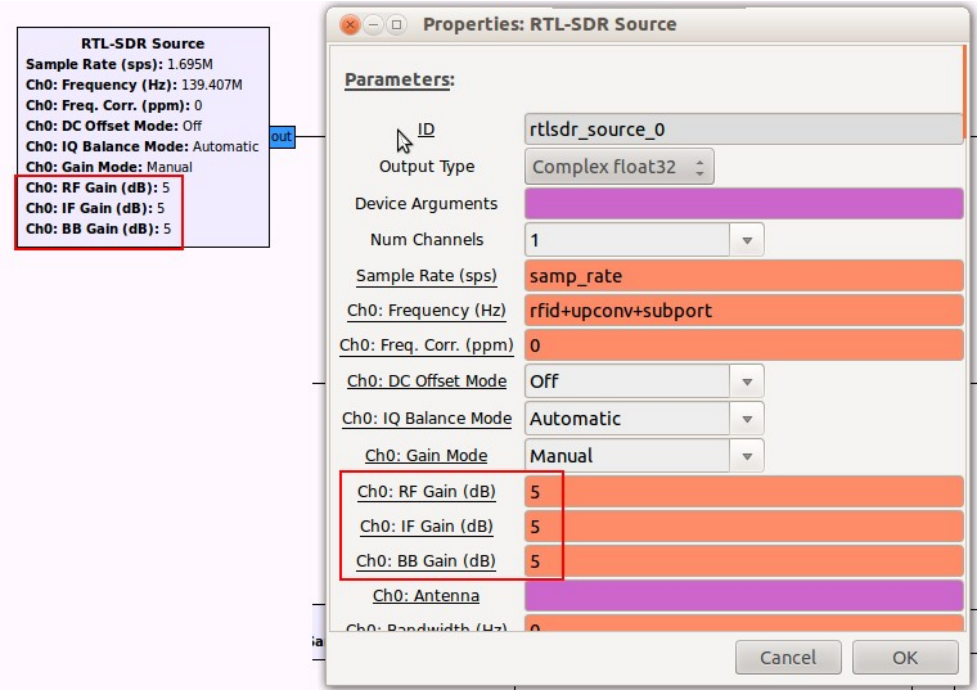


Figure 32 : GNU radio block of the dongle source

The saturation may masker a signal in the FFT graphical, so is strongly recommended to suit and test-measurement the signal before start doing the measures. Figure below shows the signal before and after the gain correction.

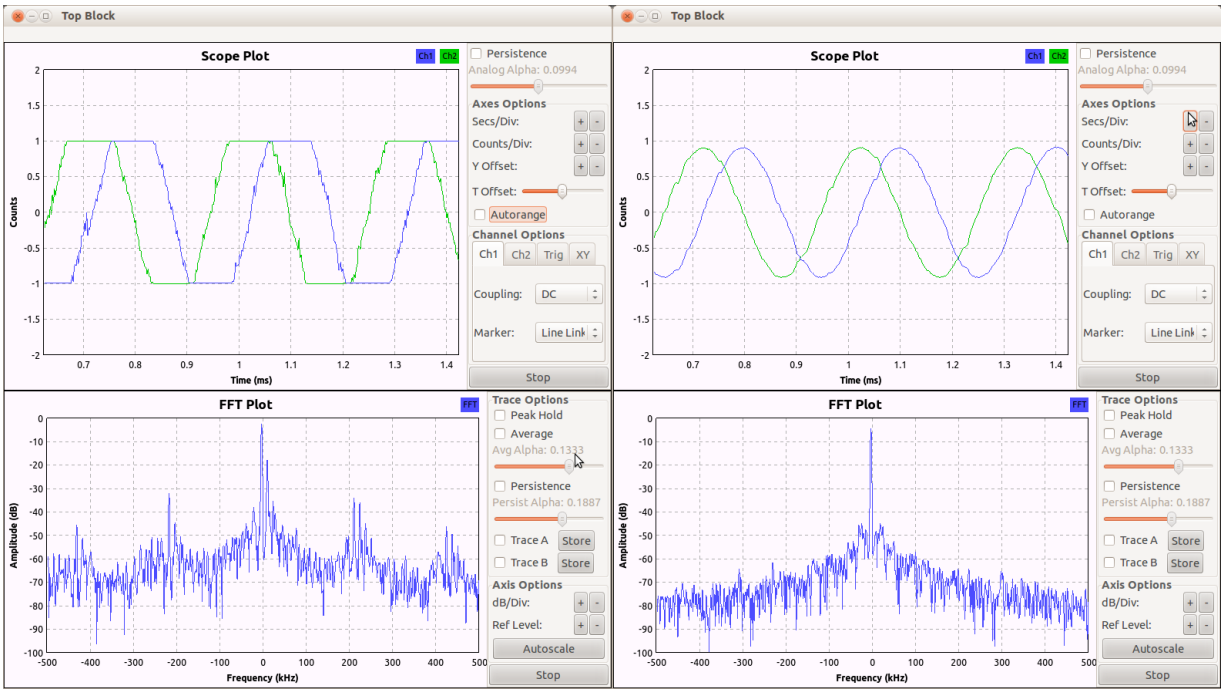


Figure 33 : Signal reception (saturated and unsaturated)

6.2 *PICC demodulation*

The signal of the communication of PICC to PCD is composite by the carrier 13.56MHz, the sub-carrier and the signal BPSK that we want. The sub-carrier is carrier/16 = 847.5KHz above the carrier, it means sub-carrier is spectrum placed in the 14.4075 MHz, the bandwidth of signal BPSK is sub-carrier/8 = 105.938KHz. All frequencies are defined in the ISO/IEC 14443. The power carrier 13.56MHz is very high relative to the other signals, and to processing the signal BPSK is necessary to remove this carrier.

To demodulate the signal we chosen the frequency of demodulation equal to the frequency sub-carrier to translated the signal to the band-base. In the project was utilized a circuit up-converter to suit the signal in the RFID frequencies to the band of the dongle (24 – 1766 MHz), this circuits up-converter the signal 125 MHz, so the result diagram of up/down converter. On the channel of measure we have two mixers as show in the figure below, the first mixer is of the board up-converter, and second is the tuner of SDR.

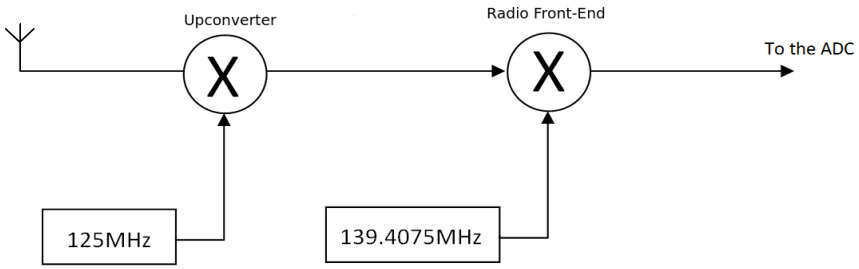


Figure 34 : Diagram of up/down conversion of mixer

The figure below showed the translations in frequency, this representations is illustrative only, it's out of scale.

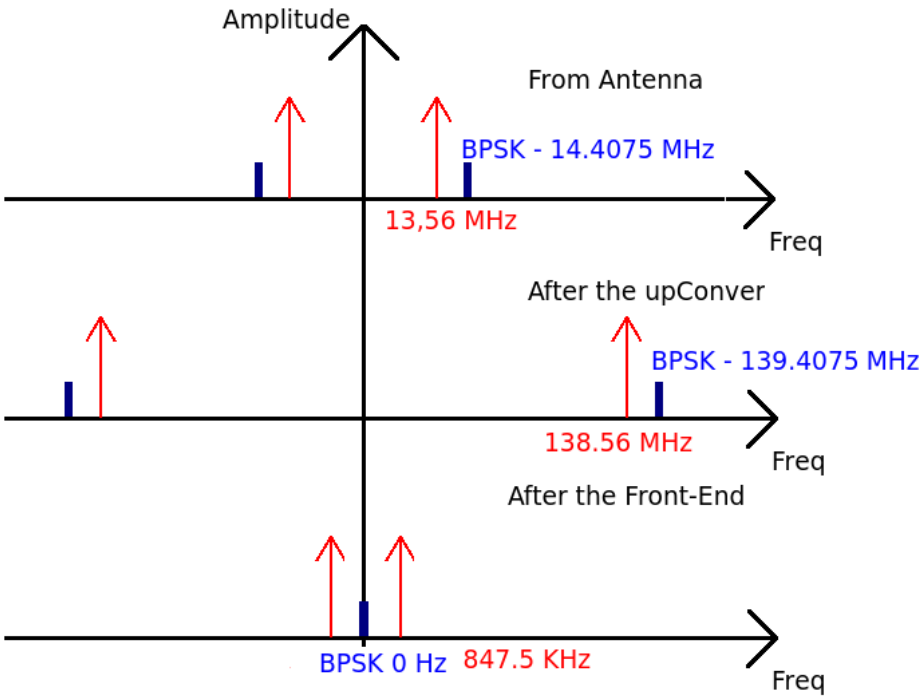


Figure 35 : Frequencies translations

We utilized three criteria to choose the frequency sample,

1) it must be a integer multiple of bandwidth 105.938KHz,
2) the carrier 13.56MHz in sample domain should be in the interval [0.5 pi, pi] and [-0.5pi, -pi], for utilize a decimation to retire this troublesome carrier.
3) The frequency samples must be low than the maximums of the dongle.

After the mixers the carrier is in the frequency 847.5KHz, as show in the figures above, so the frequency sampler utilized was two times this frequency, that is equal to 16 times the bandwidth.

Frequency Sampler = 847.5KHz*2 = 105.938KHz*16 = 1.695 MHz.

So the carrier 13.56MHz felt in the sample domain in the frequency +pi and -pi, that is the easiest to retire with a low pass filter. In the figure below is the block diagram of GNU radio to make the demodulation, removal of carrier and down-sampling.

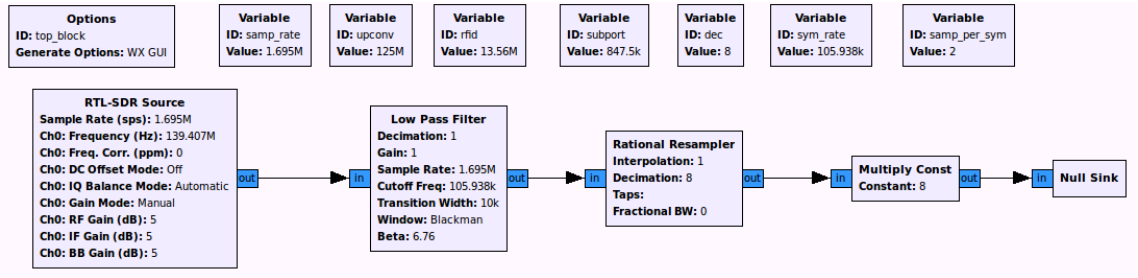


Figure 36 : *Gnuradio Companion* blocks diagram

RTL-SDR Source settings:

upconv = 125 MHz

rfid = 13.56 MHz;

subport = rfid/16 = 847.5 KHz;

Sample Rate (sps) = rfid/8 = 1.695 MHz;

Ch0 Frequency = upconv + rfid+subport = 139.4075 MHz;

Low Pass Filter

Cuttoff Rate = subport/8 = 105.938KHz;

Decimation = 8;

sym_rate = subport/8 = 105.938KHz;

Rational Resampler

Decimation = dec = 8 (to make the stream frequency sampler = 2*sym_rate, that is the criteria of Nyquist);

This block make the down-sampling, explained later.

Downsampling

We have the problem of too many samples to processing in real-time by ours computers, that make the computer to hang. Down sampling is decrease the frequency sampler, it means the change of frequency of sampling for a slowly. The down sampling by an integer factor N is a processing composite of two steps, a low pass filter and a down sampling that is the discard of samples in a factor N, it means that for each sample it takes, it discards N. To do this we have utilized the blocks of **Low Pass Filter** and **Rational Resampler**, where was configured in the previous section.

Fagundes Renan
Quiniou Thierry

We are using frequency of sample of 1.695 Msps thats creates a bandwidth of 847.5 KHz that is large for the signal of the signal BPSK of RFID, how the signal the we search is narrower so we decided to down sampling this streaming. After the decimation the bandwidth will be 1.695 Msps/N, so the frequency cutoff of the filter must be (Fs/2)*(1/N). Was chose N = 8, that make frequency cutoff = 105.938khz and the sample rate = 211.875 KHz. After this blocks all others blocks must have the parameter Sample Rate = samp_rate/dec.

$$FreqCutoff = \left(\frac{Fs}{2}\right) * \left(\frac{1}{N}\right) = 105.938\ KHz$$

$$NewFreqSample = \left(\frac{Fs}{N}\right) = 211.875\ KHz$$

The signal from the SDR-RTL Source Block is show in the figure below in the waterfall form, is possible to see in the extremes a strong signal, the carrier.

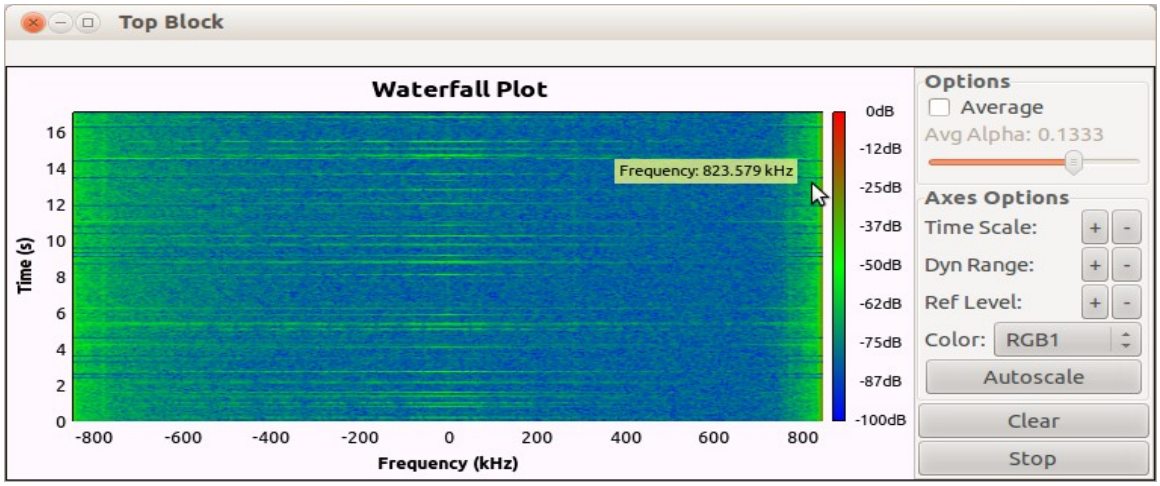


Figure 37 : Raw signal before filtering and down-sampling.

And after the filter and the dowsampling, we can see now, only the signal BPSK clean from the carrier 13.56MHz.

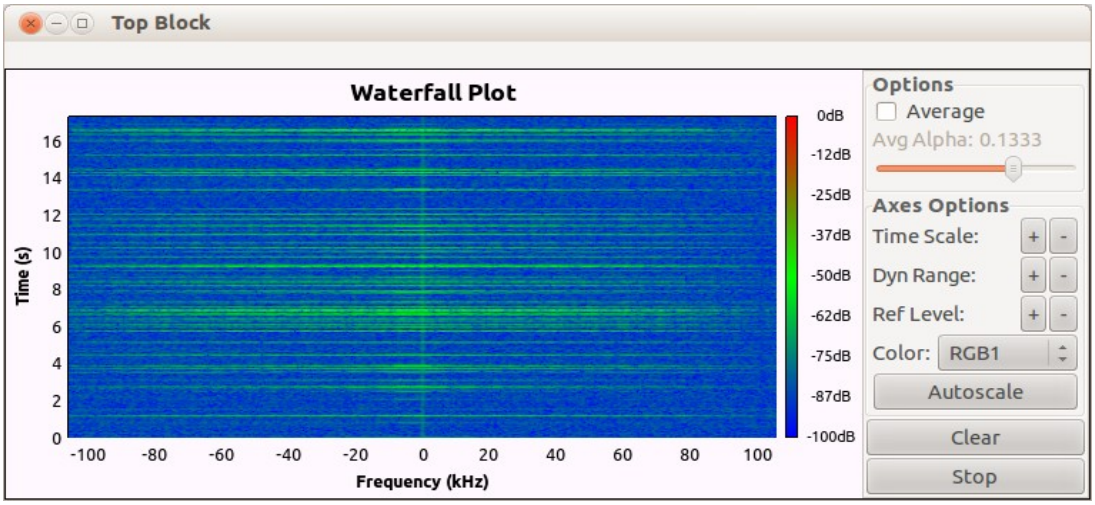


Figure 38 : Signal after down-sampling

Automatic windowing

The communications between the Tag and the Reader is in burst, the almost part of time the Tag is silent, and to well demodulation we must take only the intervals of time that there is information, the rest is noise. So, we have made an algorithm to detect the information based on the magnitude of the signal. How the signal BPSK, there is only two symbols the magnitude of the symbols are bigger rate than the periods without symbols.

In the figure below is the schematic of the algorithm, the output is connected to the “Null Sink”.

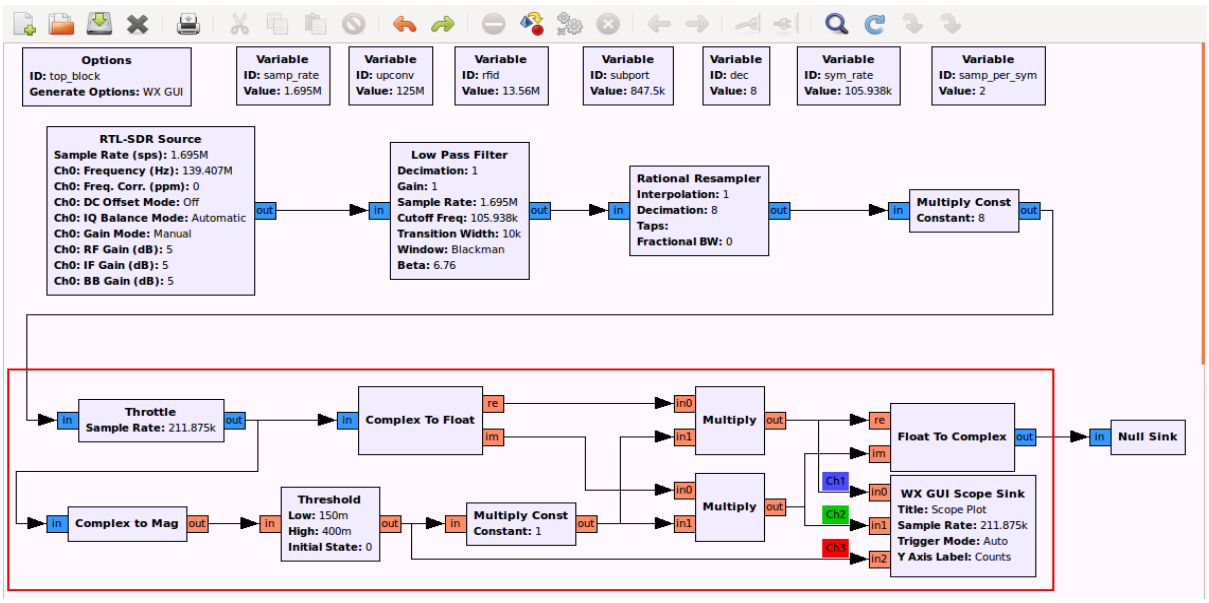


Figure 39 : Automatic windowing

In the next figure is the result of the algorithm, to detect automatically the symbols.

Figure 40 : Signal after windowing

Constellation diagram

The constellation diagram is very useful to representation of signal digitally modulated. It scat each sample in the complex plane, each sample is composite of one sample “I” and one sample “Q”. The theoretical constellation diagram of BPSK is show in the figure below. it serves as a figure of merit.

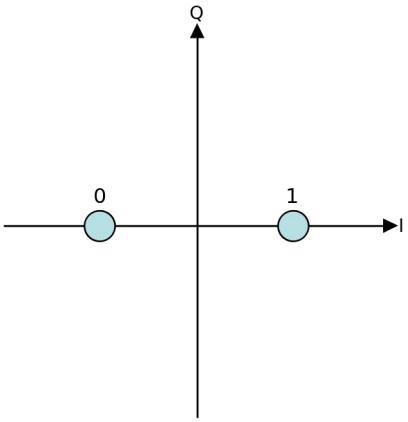


Figure 41 : Theorical BPSK constellation

We measured the output of automatic windows with this tool offered by the GNU radio, and is possible to see that the automatic windows helps to improve a better constellation diagram, as show in the figures below.

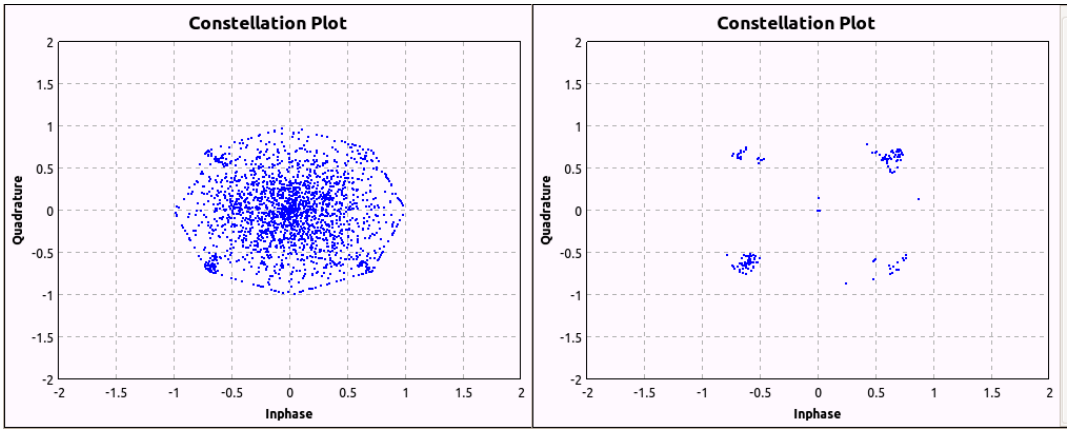


Figure 42 : Signal constellation before and after windowing

The second graphic not look like to the figure of merit, because it's not possible to see in the image, but the constellation is turning around the point 0,0. It happens because although the frequency of modulation is correct it is not stable even in phase, it makes the constellations turns around. To resolve this problem we have utilized a method of synchronization called Costas Loop Carrier Recovery that is explained in the next section.

Fagundes Renan
Quiniou Thierry

Costa Loop carrier recovery

The Signal BPSK

The BPSK is a technique very effective in noise immunity per bandwidth, but the design of demodulator is not mathematically easy [CITATION Fei02 \l 1036]. In this way Costas-Loop carrier recovery can be an inexpensive solution.

The BPSK modulation technique shifts the carrier phase between 0 and 180 degrees, according with the DATA "-1" and "+1", and so the signal modulated BPSK = DATA * cos(2*pi*fc). The easier way to demodulate is to multiply by a carrier coherent, as a demodulator heterodyne, with the carrier of demodulation is equal to modulation carrier in frequency and phase. The figure below show a receiver coherent:

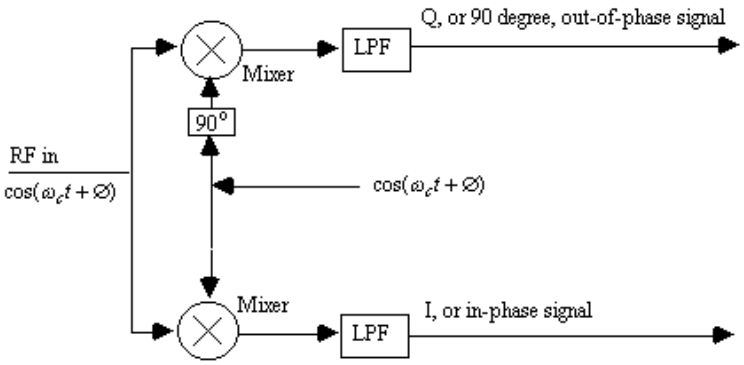


Figure 43 : Coherent demodulation

This demodulator is based in the idea of coherence, it's means as we know:

cos(a) * cos(b) = 0.5*[cos(a+b) + cos(a-b)]

if a=b

cos(a) * cos(b) = 0.5 + 0.5*cos(2*a)

In application this in the BPSK signal we have:

BPSK(signal) = DATA*cos(2*pi*fc*t)

The demodulation:

BPSK(signal)* cos(2*pi*fc*t) = 0.5 * [DATA*cos(0) + BPSK*cos(2*pi*fc*t)]

= 0.5 * [DATA+ DATA*cos(2*pi*fc*t)*cos(2*pi*fc*t)]

= 0.5 * [DATA+ DATA*cos(2*pi*2*fc*t)]

And after the filter its retrieves the signal DATA.

LPF[0.5 * [DATA+ DATA*cos(2*pi*2*fc*t)] = 0.5 * DATA

BPSK demodulation

The problem is that is difficult to generate a frequency stable and in phase with the carrier of modulation. To resolve this problem, normally utilize a technique to recovery the carrier from the signal. In BPSK demodulation there are two ways to recovery the carrier [CITATION Fei02 \l 1036] that are well know:

1) Squaring the signal BPSK: this technique is based in the signal is already a signal type cosine with the phase modulated in 0 and 180 degree, the second harmonic is modulated by the ambiguous +-360 degrees, so is phase-unmodulated. But in practice, to implement this method there is a phase offset caused by the filters with different path, that turn this technique very complicated.

2) Costas-Loop: this is a feed-backed technique related to PLL, so this is able to self-correct the frequency and the phase. This technique is suitable for synchronous communications with suppressed carrier.

The Costas-Loop

The algorithm of Costas-Loop carrier recovery is show in the figure below, It generates iteratively a carrier through the VCO (Voltage Controlled Oscillator) to correct the frequency and the phase. Its is based on the principles of coherence and orthogonality.

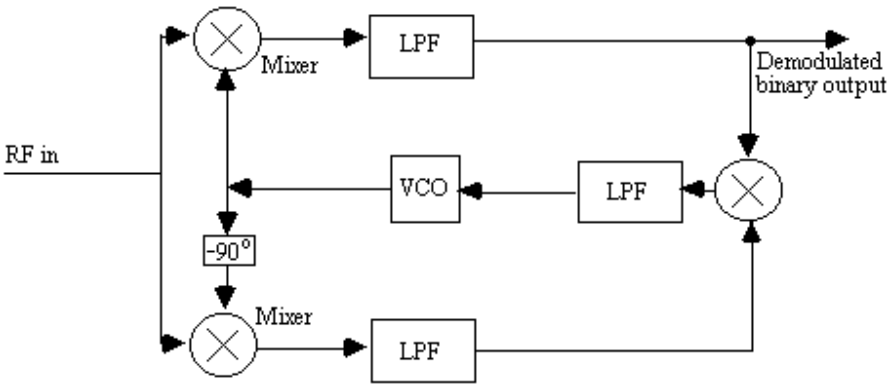


Figure 44 : Costas Loop carrier recovery

It's easy to notate that this diagram is similar to the demodulator, but its generate his own carrier, to apply to the mixer. To make such a thing it take the signal demodulate, pass through another filter to noise-reduction and applies to control the voltage-controlled oscillator. The overall loop response is

controlled by the two individual low-pass filters that precede the third mixer while the third low-pass filter serves a trivial role in terms of gain and phase margin.

This structure is easily implement in software, with the filter type FIR and a NCO (numerically controlled oscillator) as VCO. Normally the NCO already provides a In-phase and quadrature components, one the each mixer. Another parameter of real implementation is the Loop bandwidth, thats defines the speed of the control loop, and it's defined as the gains of phase/freq tracking loop. As the frequency is the derivate of phase, this gain is relative to the time of the loop.

$$K\frac{Phase}{Freq} = K\frac{Phase}{Phase/t} = Kt$$

The concept of coherency is already explicated. Here is described the concepts of Orthogonality.

As we know:

cos(a) * sin(b) = 0.5*[sin(a-b) + sin(a+b)]

BPSK(signal) * sin(2*pi*fc*t) = DATA * cos(2*pi*fc*t) * sin(2*pi*fc*t)

= 0.5 * [DATA * sin(0) + DATA * sin(2*pi*2*fc*t)]

= 0.5 * DATA * sin(2*pi*2*fc*t)

And after the filter its retrieves the signal DATA.

LPF[0.5 * [0.5 * DATA * sin(2*pi*2*fc*t)] = 0

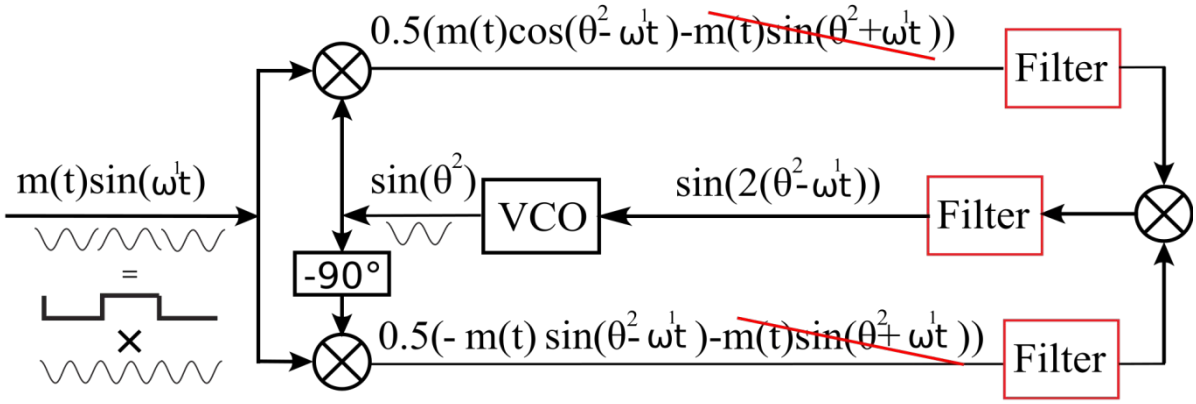


Figure 45 : Costas Loop before synchronization

It applies the coherent transformation in the upper branch and the orthogonal transformation in the lower branch. The Algorithm iterates until the signal “I” is maximum, and the signal “Q” tends to zero.

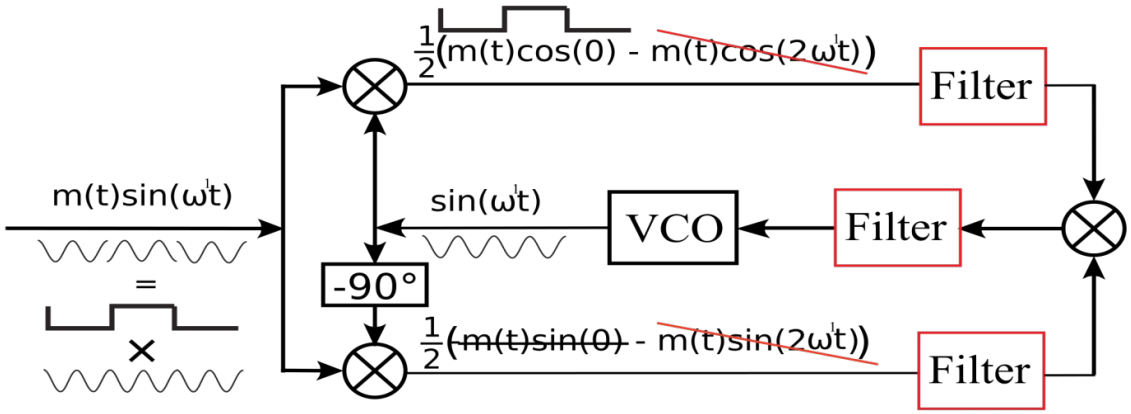


Figure 46 : Costas Loop after synchronization

Application of Costas loop in GNURADIO

In GNURADIO, the signal from the tag RFID is already pre-processed, is down-converted , decimated, filtered, and windowed, the function of Costas loop is used to correct the phase impairments, and make the signal suitable to take decision of the data whether “0” or “1”.

The Costas loop blocks offers a output of the signal output of internal VCO, and we can see that it fix only the phase of the signal. Its clear that the signal of VCO is not a signal type cosine, but a level constant, with spikes to fix the phase, to synchronize, as show in the figure below.

Obs.: the signal of frequency is out of scale, they is amplified 100x to be the amplitude reasonable compared with the signal demodulated, as show in the figure of complete system.

Figure 47 : Signal corrected with a Costas loop

After the Synchronization the signal is theorical the DATA, it means that the signal in the output of Costas Loop has the wave form of the DATA, but to transform this signal in information is necessary that each sample has one bit. So it's necessary to calculate the relation Sample/Symbol, as the follow:

$$Sample/symbol = \frac{samplerate}{symbolrate} = \frac{211.875\,K\,sample/s}{105.9375\,K\,symbol/s} = 2$$

It means that there is two samples for each bit, this can be solved with the block already used **Rational Resampler** with a decimation equal to the relation sample/symbol = 2.

Now it's possible to take the decision, if the values represent a bit “1” or a bit “0”, it is made with the block **Threshold**. The output of this block assumes only two values 0 or 1. The figure of complete system shows this arrangement.

It is possible to utilize the block **Packet Encoder** this block is utilized to take the samples from the demodulator and assemble bytes. This block has the parameters, Symbol/Sample, already calculated and bits/symbol, as we can see in the BPSK constellation diagram is two. In the system there is a block “Probe Signal” this block is used to take the signal to manipulate directly In Python.

Here is show the complete system, and how the steps of demodulation of RFID signals are connected.

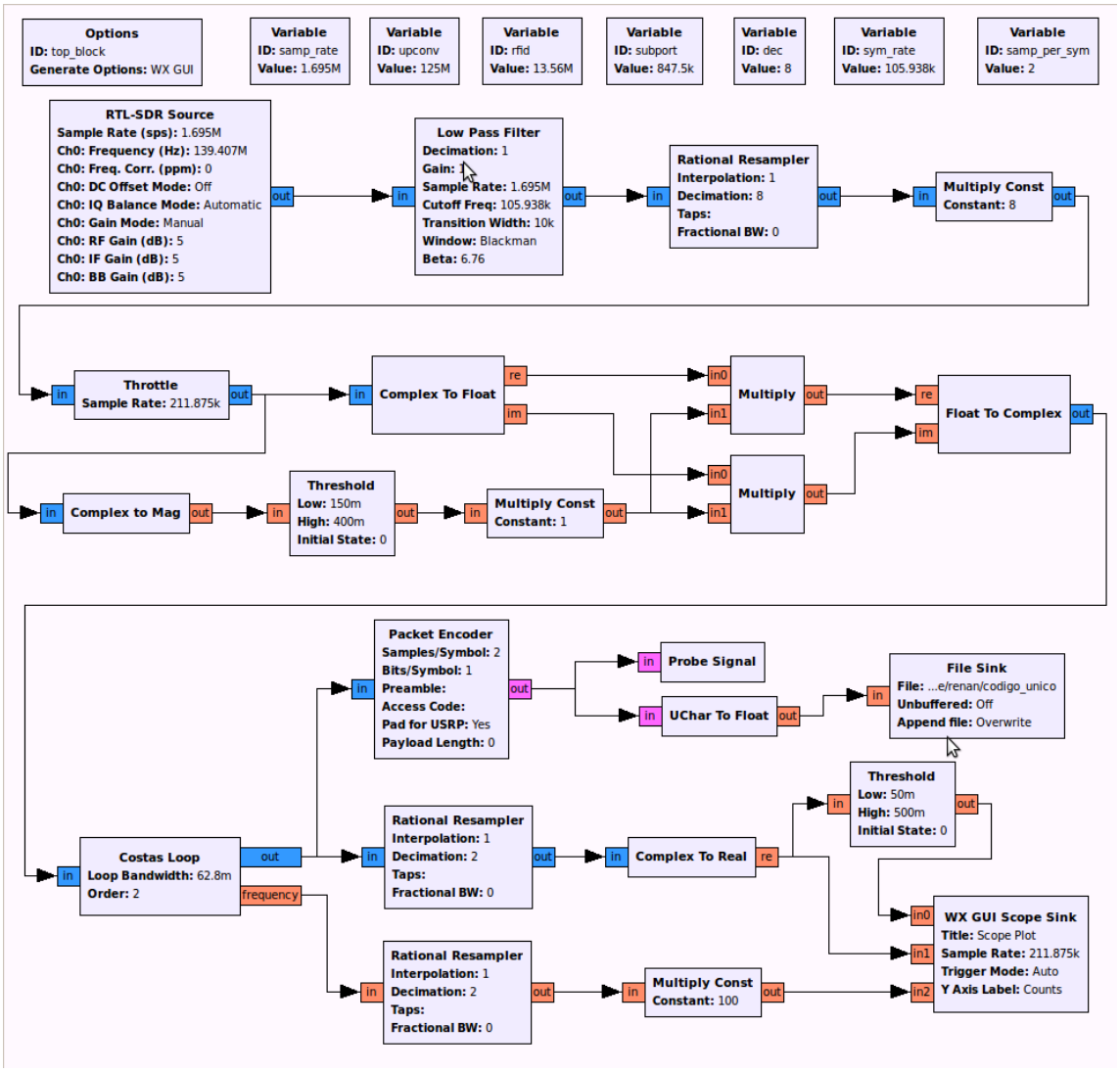


Figure 48 : Final Gnuradio Companion diagram

Fagundes Renan
Quiniou Thierry

Fagundes Renan
Quiniou Thierry

Conclusion

The GNU radio shows one tool very power full to fast to create, to test and to measure the algorithms of Digital Signal Processing to the radio signals. It has a very large library of well know and well utilized tool as Filter, Synchronizers, Modulator/demodulator and so on.

In the GNU radio is not possible to create loops with the blocks, even create stop/go conditions, like one block generate a signal to enable/disable other block, this function must be inside a block, all this things must be create or in the Python or create a block to do this. Such a thing would useful to take the signal from the windowing to enable the file sink, and save in the file only the code from tag RFID.

7 Project management

We have system engineering course since last year and the aim is to produce different architectures that allows achieving a system smoothly and free error construction. The first step involves requirements information gathering that is what the system has to do. Annex 11.3 contains this requirements and a functional architecture has been reached (Figure 49). The strategy used was the "bottom-up" analysis which consists of extracting synthesis functions by requirements analysis.

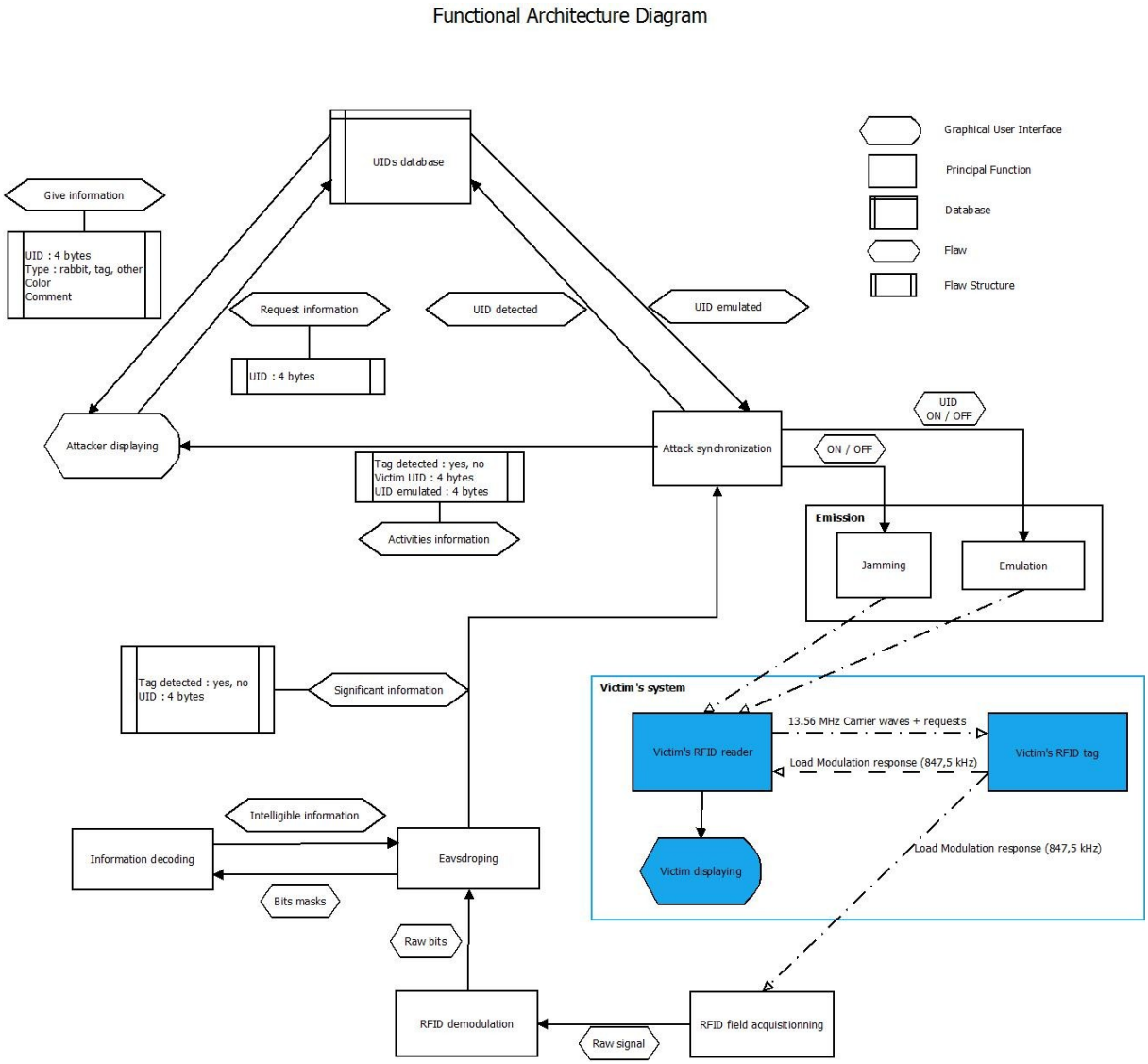


Figure 49 : Functional architecture

This functional architecture contains system functions (the squares) which are connected by information flow. This scheme is useful for extracting a physical architecture illustrated in Figure 50.

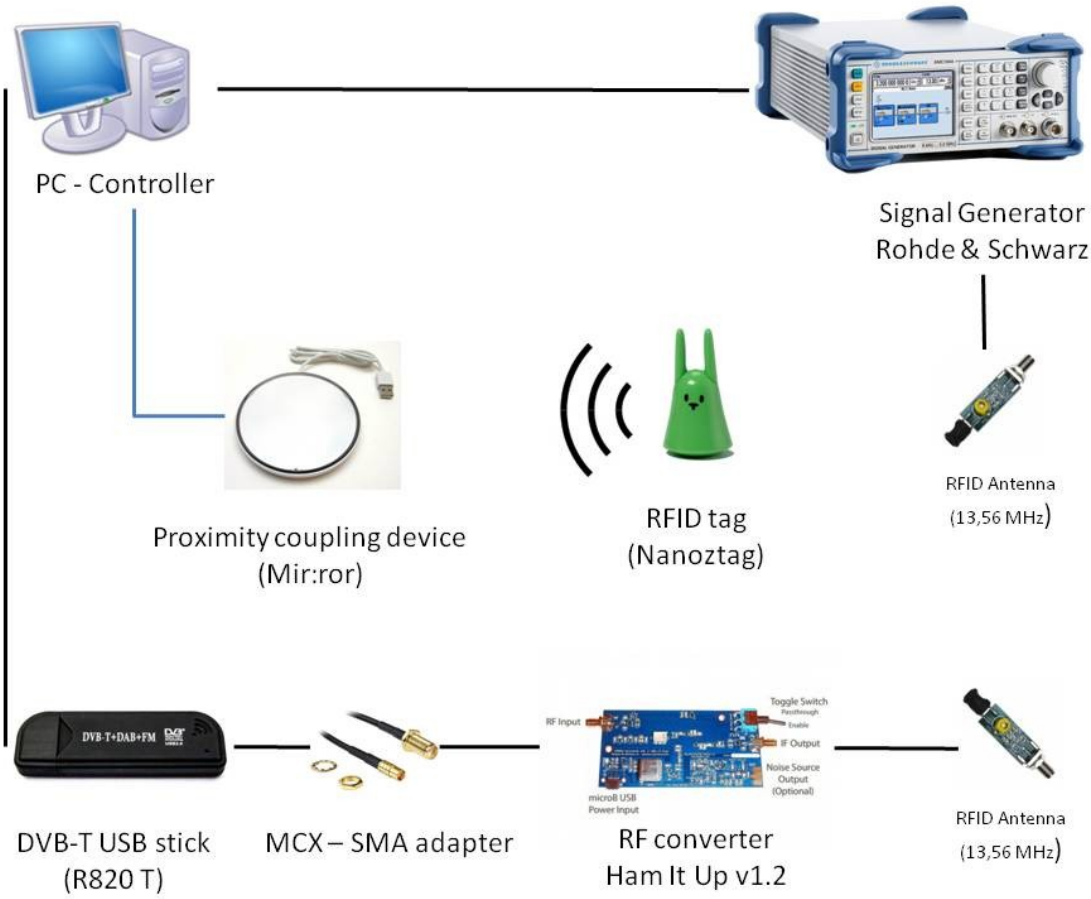


Figure 50 : Hardware architecture

This architecture is in accordance with the initial diagram (Figure 2) and additional some material such as signal generator fulfils different system functions (Jamming and Emulation).

The next steps was the implementation of each system function like a box with an input and output stream. These parts named sub-system will be realized and tested before the final integration where all sub-systems will be connected.

Actually, we had technical difficulties and we just reached to implement two sub-systems which are “RFID field acquisitioning” and “RFID demodulation”. We did not make any integration plan where these sub-systems could be validate but they are. Results are in agreement with what we expected but lack of time prevents us to finish the system. System engineering work could be reused in future for a possible continuation of this project.

8 Conclusion

The final results do not satisfied the initial aims but it is a breakthrough. We reached to record and demodulate a RFID tag emission with a cheap device. The harder part was the demodulation and it lacks us one week to develop a real time demonstrator programmed with the *Python* language. We demonstrated that is possible to realize such a system and we almost succeed.

Future work shall demodulate the PCD signal in order to well understand the *Violet* custom protocol. An integrated solution for the future system could integrate a simple jammer at the right frequency (14.4075 MHz) and a RFID emulator like the *Open PCD* 2 RFID sniffer [CITATION Ope13 \l 1036]. Nonetheless, the sniffing range should be increased and there is no trivial solution.

Software radio is growing and more and more researches are done in this way. The RTL-SDR dongle used in this project is not the only way to practice software radio and many open source projects are emerging like Blade RF [CITATION Bla14 \l 1036]. SDR is the future of telecommunications, but the road is still long before successfully integrate cognitive radio.

9 Abbreviated terms

ADC : An Analog To Digital Converter is a device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude.

ASK : Amplitude-shift keying is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. In an ASK system, binary symbol 1 is represented by transmitting carrier wave of fixed amplitude and fixed frequency for the bit duration T seconds.

CRC: Cyclic Redundancy Check code error detection.

DVB-T : DVB-T is an abbreviation for "Digital Video Broadcasting — Terrestrial".

LNA : Low-noise amplifier is an electronic amplifier used to amplify possibly very weak signals. It is usually located very close to the detection device to reduce losses in the feedline.

MITM : Man-in-the-middle attack. See 2.2 : RFID spoofing project description.

NFC : Near field communication is a set of standards for smartphones and similar devices to establish radio communication with each other by proximity, usually no more than a few inches.

PCD : Proximity Coupling Device that is the RFID reader.

PCB : Printed Circuit Board

PICC : Proximity card or object that is the RFID tag.

SDR : Software Defined Radio.

TNT : Télévision Numérique Terrestre. A French digital terrestrial television service.

UID : Unique IDentifier.

VNA : Vector Network Analyzer.

OOK : On-off keying denotes the simplest form of amplitude-shift keying (ASK) modulation that represents digital data as the presence or absence of a carrier wave.

10 Table of Figures

11 Annex

11.1 *Technical background: RFID technology*

11.1.1 Description

Short for ***R****adio* ***F****requency* ***ID****entification*, RFID are passive chips which allows some contactless information to be transmitted with a range of several centimeters (1-10 cm). RFID tags are use in many industries. An RFID tag attached to an automobile during production can be used to track its progress the assembly line. Pharmaceuticals can be tracked through warehouses. Livestock and pets may have tags injected, allowing positive identification of the animal. On off-shore oil and gas platforms, RFID tags are worn by personnel as a safety measure, allowing them to be located 24 hours a day and to be found quickly in emergencies. Malls use this technology as anti-theft and biometric passports contains tags with information such as picture identity and fingerprints.

RFID technology is expanding and it is therefore necessary to take into account security to prevent identity theft by cloning.

RFID tags used electromagnetic or electrostatic coupling in the Radio Frequency portion of the electromagnetic spectrum in order to transmit signals. An RFID system consists of an antenna and a transceiver, which reads the radio frequency and transfer the information to a processing device, and a transponder, or tag, which is an integrated circuit containing the RF circuitry and information to be transmitted. There are different standards for RFID at several frequencies. Here is a summary of the standards.

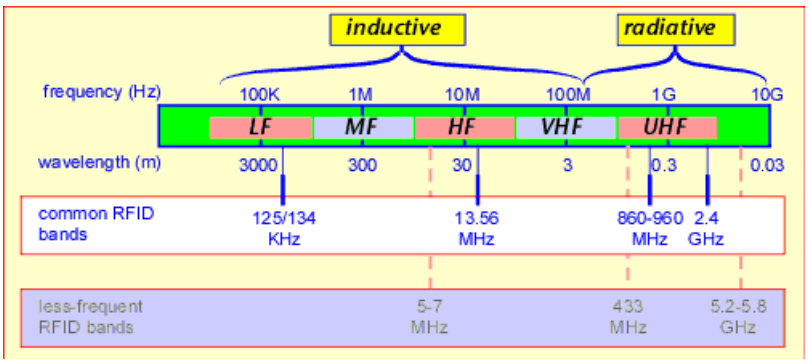


Figure 51 : RFID spectrum range

Most tags works in Low Frequency (LF) and High Frequency (HF) because they are inductive and cheap. The reader named Proximity Coupling Device (PCD) establishes a magnetic coupling with the tag called Proximity IC Card (PICC). This coupling phenomenon induces energy, which enables the passive tag to transmit information with the reader.

The LF tag cloning and spoofing have ever been realized [CITATION 1312 \l 1036]. Tutorials are easy to find on the internet and a low frequency cloning portable device can be make for less than £50 worth of hardware. Nonetheless, it is much more difficult to clone HF tags as MiFare which works at 13.56 MHz. The standard ISO 14443 is the most common and used in various applications.

11.1.2 RFID standard chosen: ISO 14443

The final aim is to realize a demonstration platform and an amusing way to do that is a RFID gadget hack. *Violet* was a company which sold rabbit gadgets which use RFID for automation applications. Unfortunately, this company went bankrupt and another named Karotz[CITATION Kar12 \l 1036] replaced it. The mir:ror reader is a PCD which read tags such as Nanoztag (a small plastic rabbit which contain a PICC). The original application was to use the mir:ror to read the unique identification to trigger a preprogrammed application such as a weather forecast or music player.

Figure 52 : Mir:ror and Nanoztag gadgets

In fact, it is possible to read the ID easily with a Linux terminal[CITATION Lin10 \l 1036]. So, this toy is a great tool to demonstrate the efficiency of RFID spoofing. This project could demonstrate that we are able to modify the radio environment in order to send to the PCD an ID (the yellow rabbit) whereas another legitimate tag is in the electromagnetic field.

This part is an extract of information found in the ISO 14443 standard part 2 which describes the radio frequency power and signal interface. We need to remember that the RFID reader is the PCD and the tag is the PICC (see abbreviated terms part 8).

The PCD modulates the amplitude of the alternating magnetic field strength with modulation pulses so as to transmit data from the PCD to the PICC.

The PICC loads the alternating magnetic field with a modulated subcarrier signal (load modulation) so as to transmit data from the PICC to the PCD.

Within the manufacturer specified operating volume the PCD will generate modulation pulses as described in the following clauses and will be capable of receiving the minimum load modulation amplitude.

Figures below illustrate the concepts described in the following paragraphs.

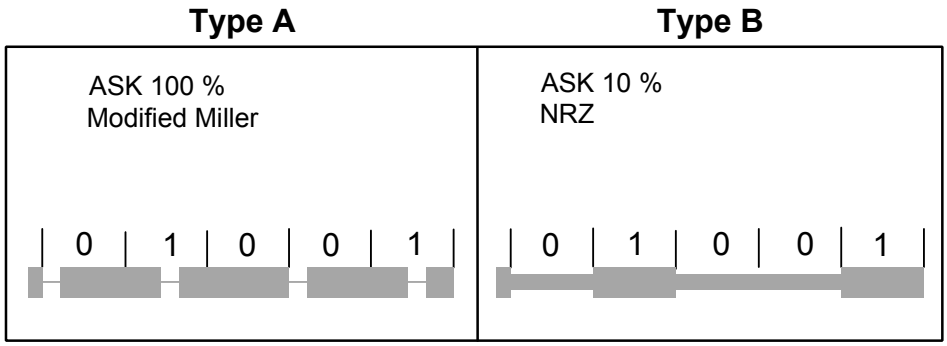


Figure 53 : Example of PCD to PICC communication signals for Type A and Type B interfaces

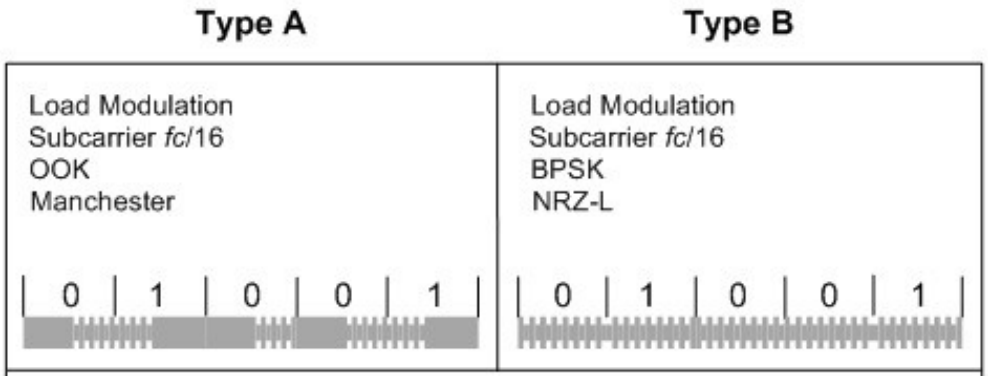


Figure 54 : Example of PICC to PCD communication signals for Type A and Type B interfaces

The mir:ror reader is compatible with type A and B standard but the rabbit tags are type B. So, this project focused just on the ISO 14443 type B standard but the approach would be the same for the OOK modulation.

Accurate research regarding the protocol would allow us to create electromagnetic fields through a signal generator such as to simulate real RFID tags emissions.

11.2 *Previous work and reflections*

This project focuses on passive RFID tags and particularly on the HF protocol. We do not discuss on active UHF tags which contain an energy source because this project focused on near field communication (several centimeters) and this kind of technology has a range of several meters.

11.2.1 LF RFID spoofer (125 kHz)

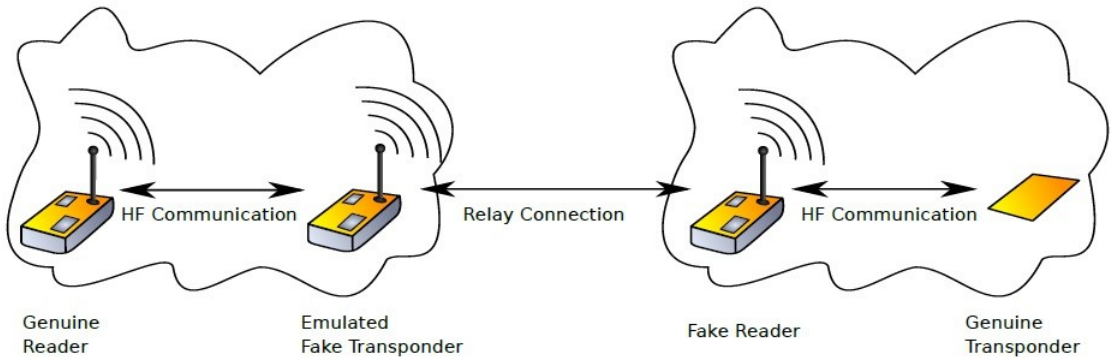
LF RFID tags are widely used for identification around the world. The most common uses are for animal identification, factory data collection and access control in a building.

Unfortunately, this technology is vulnerable and it is very easy to pick up information and to emulate a specific card. Various tutorials explain how to make this kind of device with less than 30 Euros [CITATION Fab14 \l 1036].

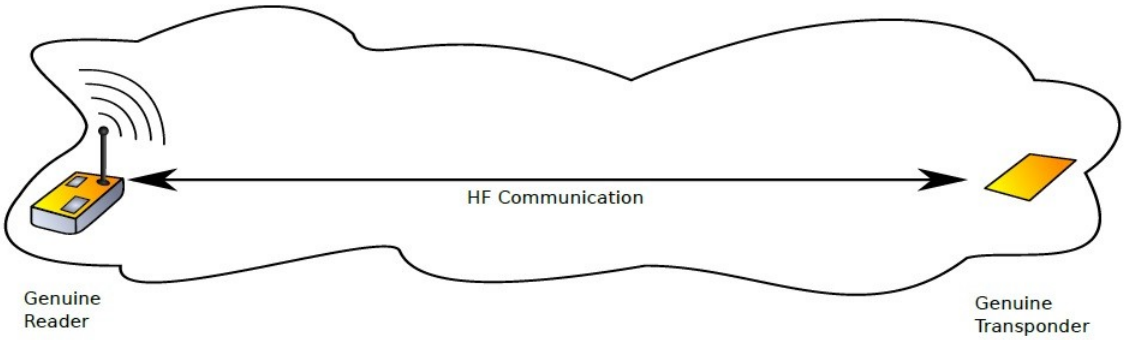
This hardware vulnerability leads to serious concerns about security however this project focuses on the HF RFID technology.

11.2.2 Relay attack

A relay attack is a type of hacking technique related to the man in the middle and replay attacks, in which an attacker relays a message verbatim from the sender to a valid receiver of the message. The goal is to greatly increase the range of the RFID communication from 10 centimeters to 50 meters. Thesis [CITATION Mic \l 1036] explains the feasibility of this attack and the demonstration realized. The figure below is extracted from this paper. Figure 55 illustrates the real hardware architecture (a) and the virtual setup seen by the victim's system (b).



(a) Real setup with all involved parties



(b) Virtual setup seen by the reader and the transponder

Figure 55 : Relay attack on a RFID system

Naturally, this scenario needs two attackers to be achieved but the great advantage is to avoid decrypting the communication. This attack can be used for NFC theft of credit cards and for the theft of e-passport IDs.

11.2.3 Cloning ISO 14443 tags

The ISO 14443 tags contain 36 bits UID (Unique Identifier) and it is necessary in our project to clone specific UID. Although a copy of 125 kHz tags is easy to achieve the standard 14443 is much more difficult to emulate. It is not easy to clone this kind of PICC with cheap components as a NXP PN532 chip is necessary. This component allows reading and writing tags. The chip specification claims that it can be used as a virtual RFID tag. A cheap Arduino shield including the PN532 chip is available with the Adafruit Company[CITATION Ada13 \l 1036]. Unfortunately, this chip needs a SAM (Secure Access Module) in order to simulate a programmable virtual tag and this kind of module is not available for public.

Laurie Adams, the famous creator of the *Rfidiot* website reached to emulate a RFID tag with just the PN532 chip but the first byte is necessary set at 0x08[CITATION Rfi13 \l 1036] and it is just possible to modify the three last bytes of the UID. This requirement is part of the ISO 14443 standard part 3.

The only device which can be programmed to simulate a card was created by the Open PCD team[CITATION Ope13 \l 1036]. They used the PN532 chip and they replaced the SAM module by a 32-bit ARM Cortex-M3 microcontroller. This project is open source and they sold this card for 46 Euros. Nonetheless, this card is sold out since the end of 2012. This source seems to be the best embedded solution to emulate a RFID tag but it is a long and complex solution.

Thus, cloning RFID tag is very difficult because of hardware limitations. Fortunately, ENSTA Bretagne has a Rhode & Schwarz signal generator[CITATION Sig13 \l 1036]. It is possible to create modulating signals with the software Rhode Schwarz WinIQSIM. It could be used to emit custom signals in order to simulate RFID tags.

11.2.4 RFID jamming

This part named "jamming" is not accurate enough. We will not discuss here about just jamming because the aim is to jam the victim's reader while continuing to recover information by eavesdropping. So, we could name this part "smart jamming" because the attacker hides information about the victim's reader without jamming himself.

The first idea was to emit a pseudo random signal which could dilute victim's information but the processing for the attacker to extract this information out of the emitted signal is complex. Synchronization problems appear and they require processing units. Another idea is to emit an identical signal with a pi phase-shifted. Thus, it could suppress the signal. This solution was proposed by M. Le Jeune but we did not find any interesting publications.

After researches about RFID jamming, some researchers prevent a tag from communicating with the reader with complex hardware. In [CITATION Block \l 1036] and [CITATION Mel \l 1036], RFID blockers are implemented by building an active tag emulator which transmits a fake UID in order to interfere with the anti-collision algorithm of ISO14443. Nonetheless, this technique is tricky and not reliable. Israeli engineers worked on the Israeli e-Voting [CITATION Yos \l 1036] and they managed to jam RFID reader. They based their research on [CITATION Kla \l 1036].

The main objective is to prevent communication between the reader and the tag. The reader emits a 13.56 MHz carrier wave which is used by the tag. In fact, it transmits its response using load modulation on a sub-carrier of the reader's carrier signal. The sub-carrier frequency (847,5 kHz) produces side bands at 12,712 MHz and 14,408 MHz. The two sidebands contain the same information. A typical reader evaluates only the upper side band. Therefore, in order to block the signal from the tag, it is just necessary to transmit a powerful signal at 14,408 MHz.

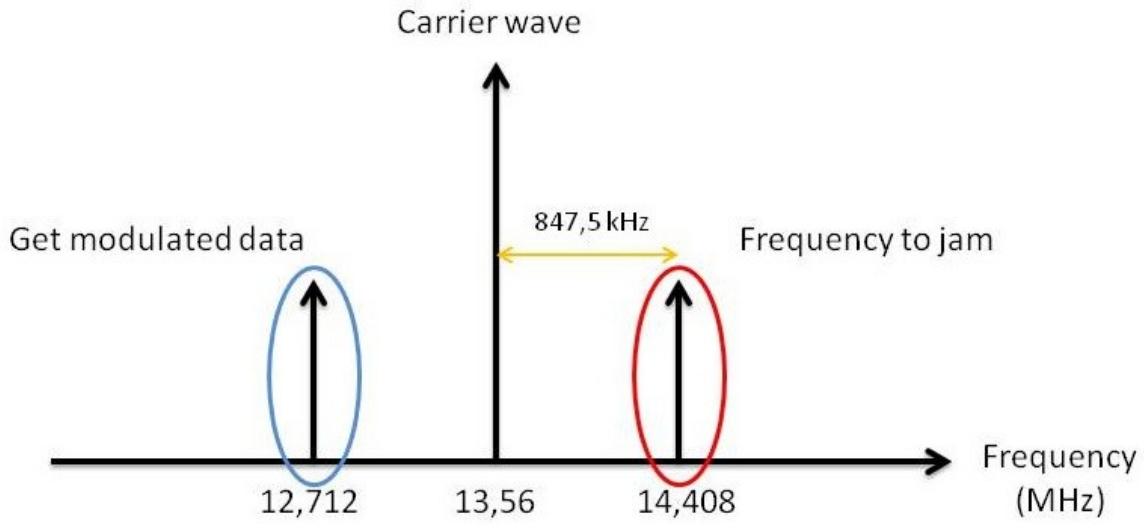


Figure 56 : RFID jamming description

This technique allows us to eavesdrop information transmitted by tag to the reader at 12,712 MHz whereas it is dazzled.

11.3 *Project management: requirements*

Identifian	Type	Expression de l'exigence	Etat	Flexibilité	Criticité	Fonction expression
Exi_1_001	Contrainte	Le système doit simuler la présence d'un tag RFID en présence d'un autre tag	A faire	Obligatoire	1	Impersonation
Exi_2_001	Contrainte	Le système doit empêcher la lecture de tags RFID d'un lecteur à distance	A faire	Obligatoire	1	Jamming
Exi_3_001	Contrainte	Le système doit capter les ondes radios à 13,56 MHz en temps réel	A faire	Obligatoire	1	RFID information capture
Exi_3_002	Contrainte	Le système doit détecter une communication entre un tag RFID et le lecteur	A faire	Obligatoire	1	RFID information capture
Exi_3_003	Contrainte	Le système doit démoduler le signal RFID selon la norme ISO 14443 type B	A faire	Obligatoire	1	RFID information capture
Exi_3_004	Contrainte	Le système doit récupérer la totalité des données enoyées par les tags selon la norme ISO 14443 type B	A faire	Obligatoire	1	RFID information capture
Exi_4_001	Contrainte	Le système doit émettre un signal modulé en charge à 848 kHz au lecteur	A faire	Obligatoire	1	Emulation
Exi_4_002	Contrainte	Le système doit créer des signaux selon le protocole ISO 14443 type B	A faire	Obligatoire	1	Emulation
Exi_4_003	Contrainte	Le système doit émuler un tag RFID	A faire	Obligatoire	1	Emulation
Exi_4_004	Contrainte	Le système doit modifier l'UID du tag émulé	A faire	Obligatoire	1	Emulation
Exi_5_001	Contrainte	Le système doit identifier les tags RFID lus par le lecteur	A faire	Obligatoire	1	Eavsdroping
Exi_5_002	Service	Le système doit identifier les étapes du protocole de communication selon la norme ISO 14443 type B	A faire	Optionnel	2	Eavsdroping
Exi_6_001	Contrainte	Le système doit enregistrer un historique des tags présents lors d'une session d'utilisation	A faire	Optionnel	3	Database
Exi_6_002	Contrainte	Le système doit comporter une liste d'UID à substituer (impersonate)	A faire	Obligatoire	3	Database
Exi_6_003	Contrainte	Le système doit contenir une base de données des trames à émettre	A faire	Obligatoire	1	Database
Exi_6_004	Contrainte	Le système doit contenir une base de données des trames de chaque étape de communication	A faire	Obligatoire	1	Database
Exi_7_001	Contrainte	Le système doit être contrôlé par une seule unité centrale	A faire	Obligatoire	1	Hardware limitation
Exi_7_002	Contrainte	Le système doit être utilisable à une distance minimale de 5 cm du lecteur (antenne émettrice et réceptrice)	A faire	Obligatoire	2	Hardware limitation
Exi_7_003	Contrainte	L'ordinateur doit contrôler l'appareil de brouillage par liaison USB	A faire	Obligatoire	1	Hardware limitation
Exi_8_001	Service	Le système doit comporter une interface graphique pour l'attaquant	A faire	Optionnel	3	Attacker displaying
Exi_8_002	Service	L'interface graphique de l'attaquant doit afficher les tags détectés par eavsdropping	A faire	Optionnel	3	Attacker displaying
Exi_8_003	Service	L'interface graphique de l'attaquant doit afficher les réglages de l'attaque par spoofing	A faire	Optionnel	3	Attacker displaying
Exi_9_001	Service	Le système doit comporter une interface graphique pour la victime	A faire	Optionnel	3	Victim displaying
Exi_9_002	Service	L'interface graphique de la victime doit afficher les tags détectés par le lecteur	A faire	Optionnel	3	Victim displaying

Figure 57 : System requirements