Wireless Link Project
Team Red

Nima Alipour Erik Andersson Simon Andersson
Xavier Bland Vito Di Fonzo Zoltán Mayer
Lei Niu Yonas Yitbarek

December 19, 2012

Contents

1 Introduction

This report constitutes the main documentation for the wireless link implemented for the Wireless Link Project course – CTH.

The task that was given as "*Design and build a wireless link in 7 weeks!*". Some more general information was given regarding what kind of supporting hardware would have been provided, what frequency could be used and which environment the system would be tested with. The hardware that was provided was a so called USRP (Universal Software Radio Peripheral), it acts as a link from the computers digital domain unto the wireless analog domain. The two frequencies that could be used were 2.45 GHz or 5 GHz and this would be tested indoors with a line of sight transmission.

The "take home advice" from the first lecture was keep it simple and then progress. This was something that resonated with our group and was kept as a guideline. Extra marks would be given to the group with the highest bit rate and the group with the lowest energy requirement.

2 System description

A schematic view of the system is depicted in Figure 1.

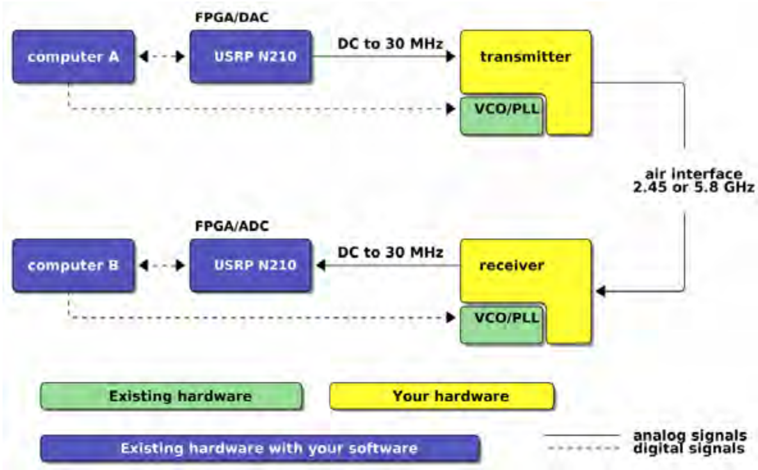


Figure 1: Schematic view of the system.

2.1 Link budget

The link budget is a significant information needed to design the system. It has to be predict and calculate through Friis' equation. It gives us values like the minimum required received power at the receiver or margin we can achieved between the received power and the level of noise. Nevertheless this link budget remains an estimation, it will be necessary to determine the real link budget when the system is ready. First a few considerations on the link: to assure a good communication the overall Bit Error Rate must be below 10^{-5} and assuming a On Off Keying (OOK) modulation the required Signal to Noise Ratio is 10 dB. The total noise at the input of the receiver can be seen as: $N_{\text{system}} = k \cdot T_{\text{system}} \cdot B$ where $T_{\text{system}} = T_{\text{RX}} + T_{\text{antenna}}$ the total equivalent temperature of noise of the system, T_{RX} is the equivalent temperature of noise of the receiver and T_{antenna} is the equivalent temperature of noise of the antenna. An estimation of the input noise can be calculated, assuming a total bandwidth of 30 MHz and an equivalent temperature of noise of the system of 460 K, we thus obtain $N_{\text{estimated}} = -127.3$ dB. Assuming a margin of 10 dB, the received power must be then at least greater than -117.3 dB. As the communication link is aimed to work indoor, a modified Friis equation is prefered:

$$P_{\text{received}} = P_{\text{emitted}} \cdot G_{\text{antenna, Tx}} \cdot G_{\text{antenna, Rx}} \cdot \left(\frac{\lambda}{4\pi R_0}\right)^2 \cdot \left(\frac{R_0}{R}\right)^N$$

where R_0 is the distance to the first obstruction [1] and N is the path loss exponent. The distance to the first obstruction is roughly 5m, and the path loss exponent N depends on the situation (for in building LOS $N = 1.8$, for in building shadowed $N = 4$). For this link, three different link budgets have been estimated: the first one considers a in building Line Of Sight while the two others situations are shadowed, in building shadowed with $N = 4$, and wall attenuation ($N = 1.8$ with -24 dB). This gives us a good estimation of the required received power at the receiver, and thus it gives us a hint about the required gain of the transmitter. The link budget of the different models can be seen in Table 4, 5 and 6, found in Appendix A.

As seen in Figure 2, the received power must be above the green line which represents the noise power. From first sight, with our prediction, it wouldn't be possible to establish a communication in a too much shadowed area (red line). Nevertheless we should be able to transmit and receive a signal within a few rooms, like in the MC2 department's offices.

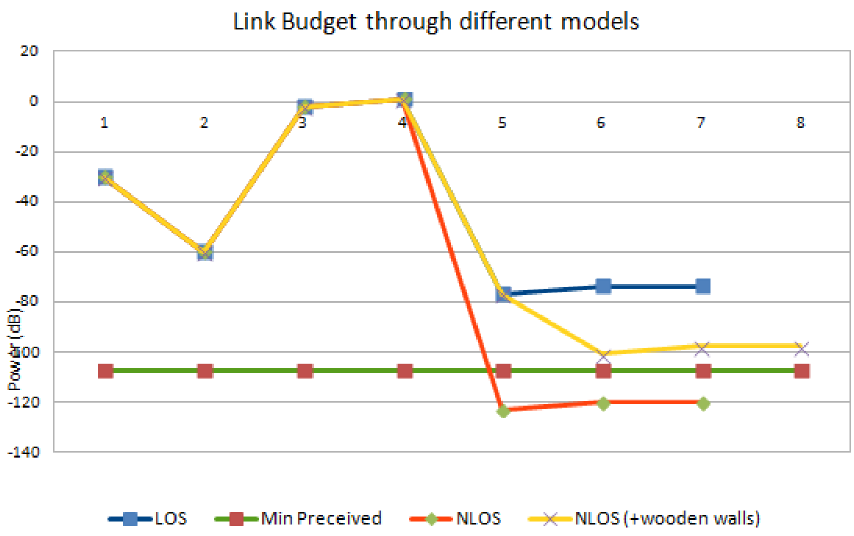


Figure 2: The power in the system through the different steps. The different lines are the diverse models.

2.1.1 WiFi noise evaluation

In order to evaluate the WiFi interference we have done some measurements using a radio scanner software. The measurement has been repeated in several point as shown in Figure 23, Appendix B. The values of the measurement can be found in Table 9 in Appendix B. In Figure 3, screenshots of the measurement equipment is shown.

Figure 3: Screenshots from the measurement equipment.

From this approximate analyses we can expect a WiFi noise of about -50 dBm (worst case).

2.2 Board design

The first thing that was done from a hardware point of view was to determine which type of transmission principle to use. It was decided that the design of a OOK system was going to be aimed for at first, but with the possibility to develop the system into an IQ-system later on if it was wished for. The block diagrams for the transmitter (Tx) and the receiver (Rx) were basically constructed according to common standards using filters, mixers and amplifiers. From the link budget it was clear what kind of performance the boards needed to have. The Tx block diagram, depicted in Figure 4, shows how the intermediate frequency (IF) signal is upconverted and then filtered and amplified. The amplification is made in two steps in order to not have more than 40 dB gain in one step, which implies higher nonlinearities. The two steps are separated by the bandpass filter. In total, the estimated gain of the Tx was around 58 dB. Along with an estimated internal loss of -30 dB, the estimated transmitted power was roughly 28 dBm.

The Rx block diagram, depicted in Figure 5, is downconverting the received RF signal to IF signal. The 2.45GHz bandpass filter SMD package is implemented after each LNA to filter out the undesired bands. Total gain of amplifiers is 53 dB which is reduced approximately by 25 dB total insertion loss of antenna, transmission lines, filters and mixer. Thus, the net estimated gain of receiver is about 28 dB.

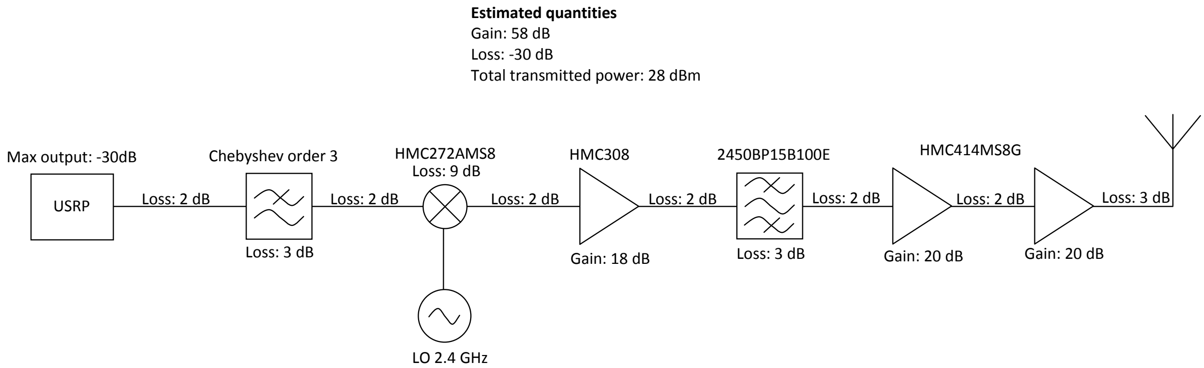


Figure 4: Block diagram of the transmitter. The choice of topology was made from a brief inspection of the inventory.

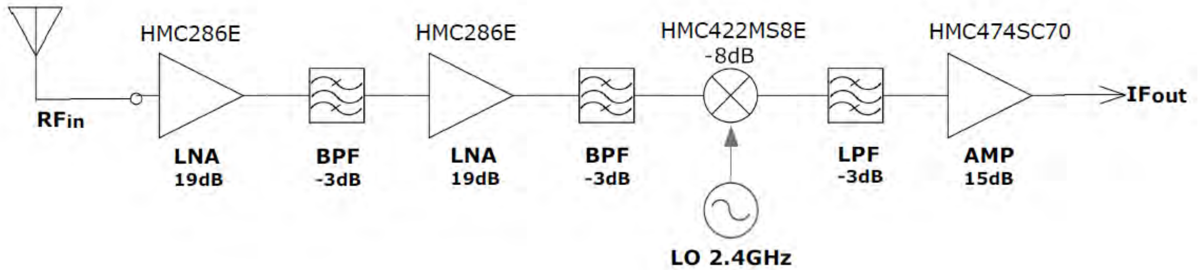


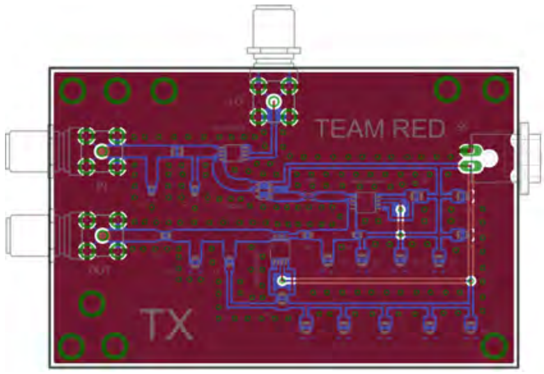
Figure 5: Block diagram of the receiver.

The design of the Tx and Rx boards was made using the software CadSoft EAGLE. With EAGLE it is possible to design both in schematic mode and layout mode. In this project, the schematic mode was first used, but due to lack of footprints in the schematics library and problems going between the modes, it was decided to do all the designing for transmitter in layout mode. But the receiver is done by defining and adding own designed components to EAGLE library. It was decided to design the boards with coplanar waveguide (CPW) as transmission type, mostly to avoid problems connecting to ground that can occur when using microstrip design. Also better shielding in the CPW case than in the microstrip case was of interest. To design CPW structures in EAGLE there is a very useful tool called "Ratsnest", which in practice lets the user switch between CPW and microstrip.

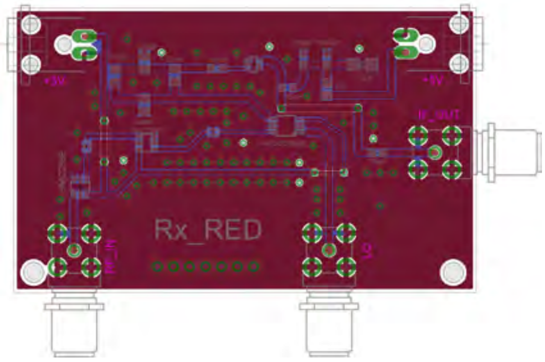
Since the boards deals with RF waves it was of great importance to consider microwave phenomenon. For instance, 90° bends introduce parasitic behaviours and can lead to line reflections issues. Therefore, smooth bends was desirable (e.g. 45° bends) . Also, the boards was eventually going to be put into a 50 Ω system, which means that the transmission line also got to have a characteristic impedance of 50 Ω. To calculate the corresponding width of the transmission lines for the frequency of interest the tool LineCalc in Agilent ADS was used. The printed circuit boards (PCBs) was made out of the composite material FR-4, which is quite cheap and resistible to catching fire. The downside with the material is that it is somewhat lossy, $\tan\delta \approx 0.02$. Its lossy nature makes it impractical to design for instance distributed filters. The only choice is then to surface mounted components. Either to design filters or to buy them. The complete design of the boards is depicted in Figure 6a and 6b.

Once the design was done, the Gerber files were sent to the manufacturer Sunstone Circuits [2]. The time past from ordering until the arrival of the boards was approximately 7 days.

When the boards had been delivered it was time to solder. Due to the really small features on the boards, we found more effective to use a solder dispensing technique and reflow soldering. Reflow soldering is a process to establish good electrical, mechanical, and thermal interconnections between the components and the PCB through the melted and then solidified solder paste. The basic materials needed to create a good solder joint are the solder, the flux, and the pad/lead metallurgy. The temperature



(a) The final Tx board design.



(b) The final Rx board design.

Figure 6: The final designs of the boards.

change during the reflow soldering is called a reflow profile. Our profile is shown in Figure 7. It can be normally characterized with four different phases: preheat, thermal soak, reflow, and cool down. In the preheat phase, the temperature is increased at a controlled rate to minimize any thermal damage to the components and the board. Then in the soak phase, the temperature of all surfaces being soldered is equalized. In the reflow phase, the solder quickly reaches its melting temperature. The solder is solidified in the cool down phase before exiting the reflow oven. An infrared oven was used.

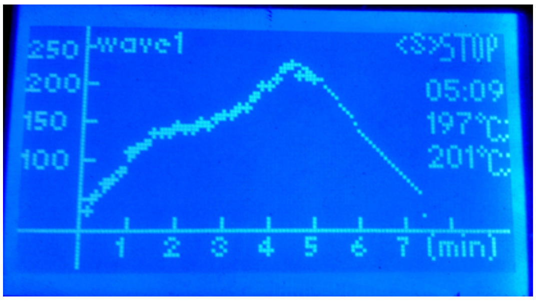
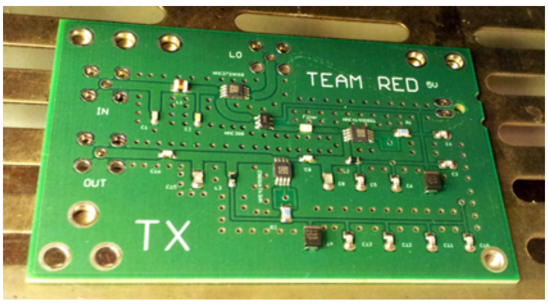
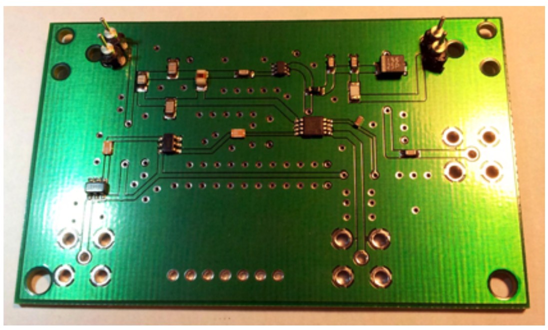


Figure 7: The heating cycle of the soldering oven.

In Figure 8 are some pictures of the boards just after the reflow. As is visible only the power and SMA connectors are missing, they have been soldered manually as they are through-hole components.



(a) The soldered Tx board.



(b) The soldered Rx board.

Figure 8: Both the boards after soldering.

2.3 Component choices

In this section, a description of the different components and utilities chosen is presented.

2.3.1 Hardware

When selecting the daughterboards, documentation provided by Ettus Research [3] was used. In the documentation it is clear that even though the SBX and WBX boards outperform the RFX when it comes to noise figure (NF), the built in filters in the RFX boards makes them most suitable if a narrow frequency band is of interest. Since this was the case, we chose the RFX2400 with a frequency range between 2.3 GHz and 2.9 GHz.

When choosing the components for the boards, different aspects were of interest for different components. When selecting the mixer for the Tx, a mixer with quite high NF was chosen, but it had internal amplification as well as good isolation and low power consumption. For the Rx, NF was almost the only parameter of interest. The amplifiers on the Tx boards was selected in order to achieve the desired gain. The amplifier with the highest gain was not chosen due to its high power consumption. For the Rx, the NF of the LNA was of special interest. The filters needed was intentionally considered to be a part of the design process, but problems with realizing desired behavior led to that filters were bought. In Table 7 and 8 in Appendix A, the used components for the two boards are listed.

2.3.2 Software

There are multiple options for a development framework for the USRP. Firstly, we have considered working only with Simulink. However, after the initial setback of the first couple of weeks due to troubles with the Lab computers and the USRP firmware, we have realized that we may not be able to build up our Simulink competence and design the software at the same time. Therefore, while some of us were working with Simulink, others wrote Matlab code, in order to make sure that we have something working.

2.4 Communication system implementation

Here the implementation of the software is described.

2.4.1 Simulink

A simple OOK modulation scheme is used for the Simulink implementation. The transmitter consist of the following blocks.

- Signal from workspace

  A block used to input the data.

- Matlab function

  A Matlab code block uses pilot bits in the first frame for synchronization purpose.

- Rectangular pulse filter

  This block upsamples the input signal using ideal rectangular pulse and the pulse length is determined in here.

- USRP Transmitter

  Inputs the upsampled signal. In this block the interpolation factor and center frequency are specified.

The receiver also consists of the following main Simulink blocks.

The windowed integrator integrates the received signal with window size same as the pulse length. For binary input data, the received signal with interference is shown in the Figure 11.

The receiver set the minimum points to zero and then normalize the signal but if the signal in a frame is all close to '0', normalization will set the all '0' data to '1'. 'Matlab function 2' takes care of this problem. Matlab function 2 checks if the data in a frame is all '1' or '0' and output it directly.

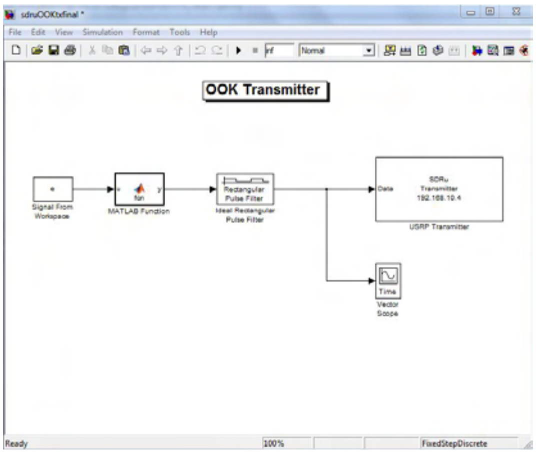


Figure 9: OOK transmitter.

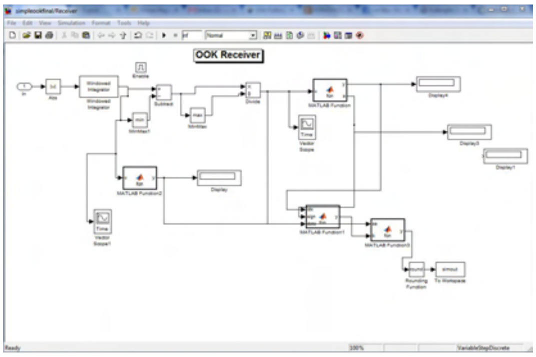


Figure 10: OOK receiver.

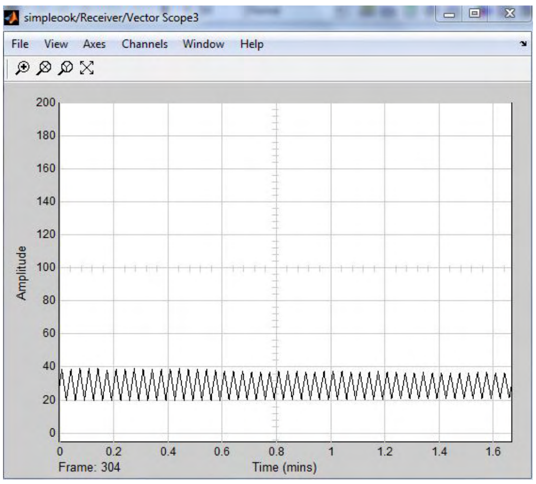


Figure 11: Received signal with interference.

Matlab function: A Matlab code is used to find out the first edge '1' to '0' or '0' to '1' in order to get the position for sampling. Matlab function 1: Samples the signal starting from the sampling position and after every pulse length. Finally it uses 'round' block to change output signal to binary. Matlab code: A Matlab code is used for frame synchronization and to calculate bit error rate (BER). Simulation:

A binary image with wireless at 2.45 GHz center frequency without channel coding is simulated. The interpolation and pulse length parameters are set to achieve a certain bit rate. the bit rate is determined below for a fixed USRP output rate which is 100 Msample/s.

$$\text{Bite rate} \cdot \text{Interpolation factor} \cdot \text{Pulse length} = 100 \text{ Msample/s}$$

For different values of pulse length, interpolation and decimation factor a certain bit rate, delay and bit error rate are calculated and shown in Table 1.

Table 1: Evaluation of parameters.

Interpolation	**Pulse length**	**Bit rate [kbps]**	**Bit error rate (BER)**	**Delay [s]**
128	10	77	0	12
512	10	20	0	1
128	100	7	0	8

2.4.2 Matlab

The `comm.SDRuTransmitter` and `comm.SDRuReceiver` system objects can be set up to initiate transmission with invoking the `step` method. The sampling rate of the USRP hardware is fixed, and the actual transmission rate can be controlled by varying the interpolation and decimation rate at the transmitter and receiver respectively. The modulation scheme is OOK, for pulse shaping we have chosen a root-raised-cosine filter (RRC). Our implementation inserts a synchronization pattern at the beginning of each frame. Burst mode is used in the receiver, this implies that a certain number of received frames are stored in the TCP buffer of the operating system. (Therefore, the size of the available buffer depends on the operating system.) Using the burst mode guarantees that no samples are lost due to slow processing of the received data. We have chosen to write scripts to transmit a grayscale image, after applying block-based quantization and discrete cosine transform (DCT) compression. As a channel code Reed-Solomon (RS) code is used. Since quantization is used, the codebook is assumed to be known at the receiver, and only the indexes are transmitted. The implemented transmission chain is shown in Figure 12.

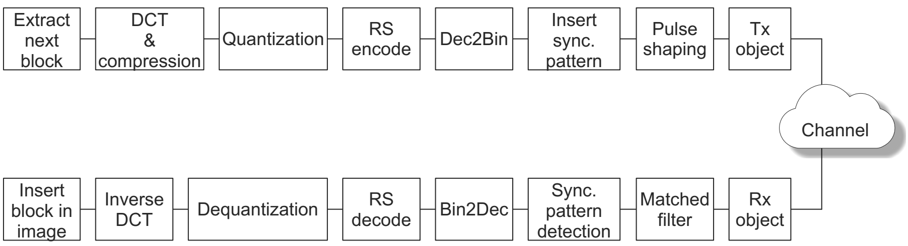


Figure 12: Processing chain for image transmission.

3 Measurements

When the boards were soldered it was time to inspect them to discover any short circuits. As a matter of fact there were some issues with low resistance between the two poles of the power supplies in both the Tx and the Rx board. A search for the weak link was initiated and some resoldering was made. After that some components had been resoldered and the TA had inspected the board it was concluded that the resistance at some points were still low, but for our purpose acceptable. Studying the behavior of both the Tx and the Rx using a spectrum analyzer gives us a hint about the good functioning of the PCBs. Unfortunately we obtained a spectrum analyzer just two days before the demonstration, an earlier delivery could have been useful, particularly in the case of the receiver which seems to have some problems. Nevertheless the transmitter has been tested with a spectrum analyzer, feeding it a 26 MHz signal and linking the output to the input of the spectrum analyzer.

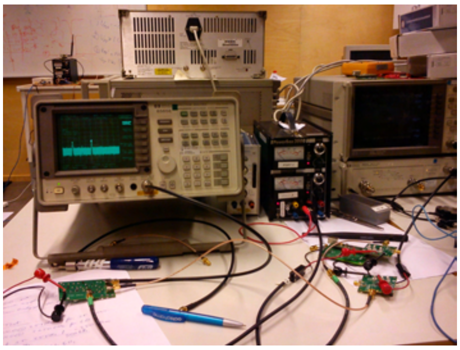


Figure 13: Spectrum analyzer and set up used during the tests of the Tx

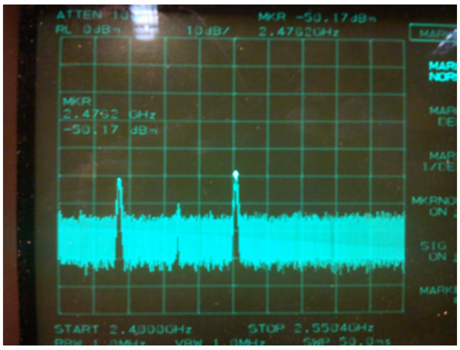


Figure 14: Resulted spectrum from the Tx PCB after input of a 26 MHz signal.

From these results, and comparing it to the input power of the signal, we can deduce the total gain of the Tx board, 15 dB. If we compare this gain to the expected gain calculated from the design of the transmitter, the difference is thus 13 dB less than what we were expected. This difference can be explained by bad soldering, additional connectors and power saturation for some amplifiers.

Unfortunately the tests with the receiver board were not good enough as the output signal couldn't be found. The possible issues could have been bad soldering, a problem in the design of the Rx board or even bad supply or bad VCOs. Considering the little time we had at this point, further troubleshooting wasn't possible, particularly in the case of a design problem.

4 Tests

Different tests have been conducted to test the system (both the hardware and the software) and that led to different results. First, we decided to test separately the transmitting part and the receiving part. Using the RX2400 advanced boards, we were able to wirelessly transmit a test picture and receive it on the other side using alternatively the designed transmitter and the designed receiver along with the LFTX and LFRX boards, whereas the other side of the link was composed of a RX2400 advanced board. Containing still significant errors, the picture has been transmitted and received correctly.

Figure 15: Received picture sent from our TX PCB and received with an advanced RX2400 board with a distance of roughly 40cm.

Figure 16: Received picture sent from an advanced RX2400 board and received by our receiver PCB with a distance 1m.

Nevertheless at this point the complete system hasn't been tested yet. Thus, when we wanted to test the boards and the system on the days after it wasn't possible anymore, and just noise is received in the USRPs. After some considerations, we manage to make the transmitter board working, but still not the receiver.

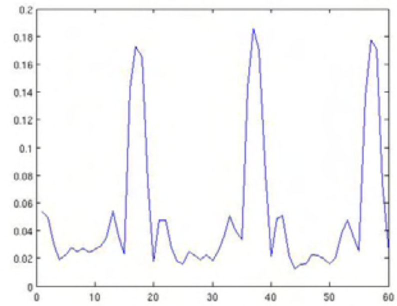


Figure 17: Plot of the received signal sent with our transmitter software and PCB and received with the RX2400 board: we can clearly see the signal.

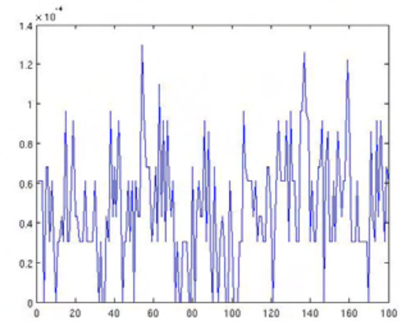


Figure 18: Plot of the signal received from our receiver (software and PCB) sent from the RX2400 board: just noise is received.

A possible reason of this failure can be dead components due to a bad manipulation of the PCBs or to bad soldering and current leakage. As we are lacking time to investigate more accurately these problems and correct it, we won't be able to achieve a communication with the complete system. Nevertheless it will be possible to transmit a picture using our transmitter and the RX2400 board as receiver board using the setup depicted in Figure 19.

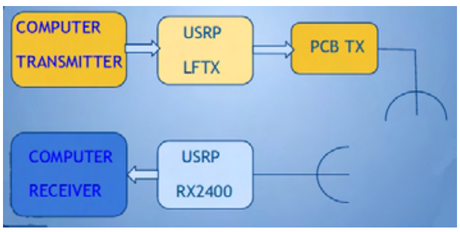


Figure 19: Flow chart for the system setup.

5 The team

The team was constituted by eight people with heterogeneous backgrounds with diverse expertise involving both hardware and algorithm competence, comparable to industrial development teams. This has been really beneficial for the high cross-disciplinarity of the project. The team members are listed in Table 2.

Table 2: The team Red members and background.

Member	**Background**
Alipour, Nima	Wireless, Photonics and Space Engineering
Andersson, Erik	Wireless, Photonics and Space Engineering
Andersson, Simon	Communication Engineering
Bland, Xavier	Wireless, Photonics and Space Engineering
Di Fonzo, Vito	Embedded Electronics System Design
Mayer, Zoltán	Communication Engineering
Niu, Lei	Communication Engineering
Yitbarek, Yonas	Communication Engineering

As the project was going on, we had several meetings to align all the progresses and make new choices. The competences and the tasks were assigned as well according to the needs. The team was schematically divided as shown in Table 3.

Table 3: The assignment of the different tasks.

Task	**Members**
System analysis	Erik, Xavier, Vito, Zoltan
USRP and software troubleshooting	Simon, Zoltan
Hardware design and assembly	Nima, Erik, Vito
Software design	Simon, Zoltan, Lei, Yonas
Testing	Xavier, Zoltan, Erik

6 Organization and planning

The general project was broken up into two pieces, hardware and software. These chunks were then broken into smaller problems that could be overcome one by one. To make it more realistic not only technical implementation was being considered but also fairly “simple” tasks such as getting communication with the USRP working. This proved to be harder than what was initially thought. Figure 20 shows a first try to split up and estimate the different parts of the project. The time schedule went out of sync quite fast but it shows that higher modulation schemes than OOK was not really considered feasible unless progress went faster than planned, mainly because it would have added complexity, both in the software and the hardware.

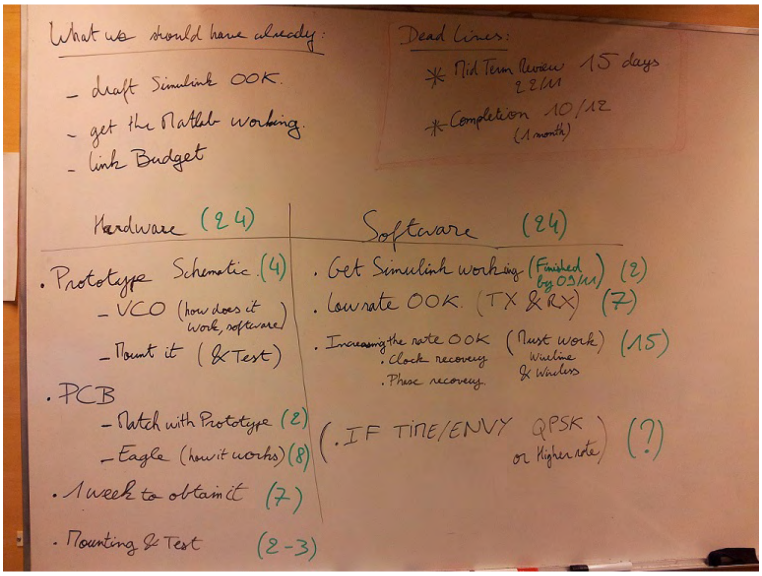


Figure 20: A naive schedule of the process.

After successfully communicating with the USRP via Matlab, two different paths appeared. Should the system be implemented with a Matlab script or a Simulink model? Both options had some pros and cons. Since the group was large, a shotgun approach was used. Half the software group would focus on Matlab while the other half tried to get Simulink to work. The thought was that if one group got a large lead then the focus could shift to this implementation.

Matlab

All of the software members were very familiar with Matlab and had worked extensively with it before. Some examples for programming the USRP with Matlab were given together with the installation and could be used as a reference. Although using Matlab would mean that everything would have to be coded from scratch or somehow try to include the Simulink Communication System Toolbox in the Matlab script which can be tedious.

Simulink

The examples that came with the installation were very well coded and easy to use. An example would be the QPSK Tx and Rx model that worked straight out of the box. An extensive library is provided with several nice and convenient “boxes” that just works when including them. None of the members had worked with Simulink before but the existing models that could be used as references seemed like a good starting point.

7 Conclusions

Using the shotgun approach turned out to be a successful strategy since the group had decided to use OOK which is not as demanding on the software or hardware. Had all effort gone into one solution then it is possible that higher modulation schemes would have been implemented using the Ettus daughterboards which would have given major problems when the custom hardware was used since they are far from ideal and therefore a slim chance of a working system at the deadline.

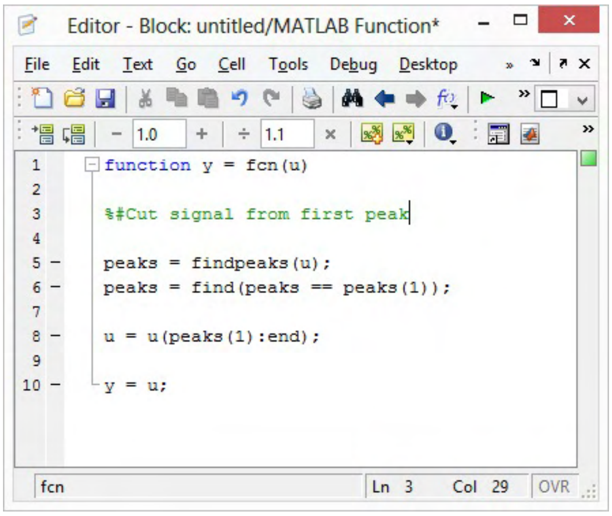


Figure 21: Simple Matlab script.

Using Simulink was also much harder to use than initially thought. It was easy to get something working fast but to then implement new features which there was no block in the library for proved to be challenging. There exists a way to use the “Matlab function” block to write your own Matlab code to process data, depicted in Figure 21, but it should not be mistaken for a Matlab script. A seemingly innocent scripts which takes an insignal, finds the first peak, cuts away the samples before the peak and then outputs this signal will fail miserably. An embedded Matlab function is not the same thing as a normal Matlab file. Simulink works by compiling the model to C code and then running it as an executable. There are certain Matlab functions that are supported and can be compiled to C but not all of them are, an example would be findpeaks. The user must then specify that this function is a Matlab function and has to be run in Matlab, but then come the problem that Simulink no longer knows what type Matlab returns from this. Once this is taken care of another problem arises. Simulink wants every function to explicitly specify how big the output will be from every function. This is not possible however since the script doesn’t know where the first peak is located.

With the hardware given it was not possible to achieve a real time transmission using our system. The effect of this is that it is impossible to keep a transmission going for a given amount of time. Eventually the delay will be so long that the buffer will be overflowed and frames will be dropped and no more data is possible to transmit until the buffer has been cleared. During experiments there has been more than a minutes delay from the end of transmission at the Tx until this has been noticed by the Rx. Here follows a quick example to realise how quick the DSP has to be done in order to remove any buffering: Using a decimation factor of 4 makes the USRP output 25 Million samples per second. Each of this sample is a 32 bit double precision float value. That means that 800 Mbit of data is created per second and is supposed to be transmitted by a 100 Mbit/s ethernet link. Just transmitting the data created in 1 second will take 8! By using higher decimation factors we can lessen this but it still puts heavy requirements on how fast the DSP has to work. Using Simulink this delay can be mitigated at least a bit, this might be because Simulink is in general faster since it compiles to C and runs natively whereas Matlab is an interpreted language. In Matlab there is sometimes a sort of transient high power part in the initial frame which has to be removed and potentially destroys any data that was transmitted at this time. This makes it hard for the USRP to “wait” for the transmitter before receiving data since the start

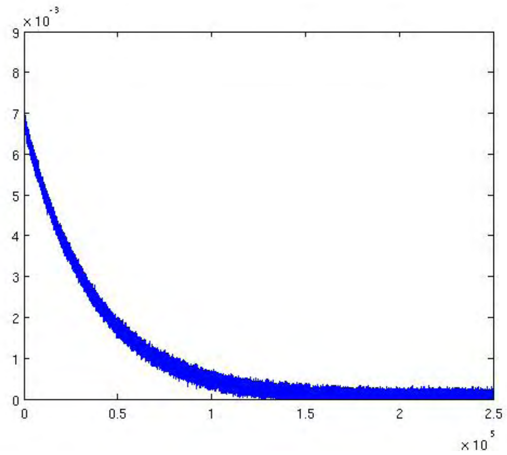


Figure 22: First frame from the USRP with only noise present.

of the transmission might be lost.

It is not possible to take this project much further than this with the existing hardware and software. Since the hardware is not working as of now there is a need for new hardware to be designed and soldered. For the bit rate to increase either faster hardware needs to be used so that a lower interpolation and decimation value can be used, or a higher modulation scheme which would require major rework on the custom hardware. To get a larger file transfer one way to go is with faster hardware to achieve real time transmission or lower the bit rate significantly but it would be tricky since with this decimation rate the hardware is not able to keep up even when doing no DSP on the received data. Otherwise a switch to Simulink might make it possible to achieve a longer transmission without dropped frames however that is unclear.

We have found that when keeping things simple you often end up with a better and more reliable solution, especially when all parts are not available from the start. However even when things are kept simple there are always so many parameters that affect the system which happened in our case. The hardware did work at some point but somewhere along the line something happened. Had we gone down the more complicated route it is very much possible that the system would never had worked. Another lesson is that when working in real life scenarios it takes time to being able to sit down and start coding. Setting up the development environment can be hugely time consuming.

References

[1] Pozar DM. Microwave engineering. 4th ed. John Wiley & Sons, Inc., 2012.

[2] Sunstone Circuits. Homepage stating the products and services of the company. Available from: *http://www.sunstone.com/products-services.aspx*

[3] Selecting an RF daughterboard. Ettus Research. Available from: *http://www.ettus.com/content/files/kb/Selecting_an_RF_Daughterboard.pdf*

Appendices

A Tables

Table 4: Link budget in the building, LOS.

LOS	Value [dB]	Cumulative value [dB]
USRP	-30	-30
TX Losses	-30	-60
TX Gain	58	-2
Antenna Gain	3	1
Modified Friis Equation	-77,6	-76,6
Antenna Gain	3	-73,6
TOTAL Received Power	**-**	**-73,6**

Table 5: Link budget in the building, shadowed.

NLOS	Value [dB]	Cumulative value [dB]
USRP	-30	-30
TX Losses	-30	-60
TX Gain	58	-2
Antenna Gain	3	1
In-building shadowed	-124	-123
Antenna Gain	3	-120
TOTAL Received Power	**-**	**-120**

Table 6: Link budget in the building, shadowed with -24dB attenuation from wooden walls.

NLOS (+wooden walls)	Value [dB]	Cumulative value [dB]
USRP	-30	-30
TX Losses	-30	-60
TX Gain	58	-2
Antenna Gain	3	1
Free Space Losses	-77,6	-76,6
Wooden walls	-24	-100,6
Antenna Gain	3	-97,6
TOTAL Received Power	**-**	**-97,6**

Table 7: Bill of materials for the Tx board.

Name	Value	Package Size	Quantity
Resistor	200 Ω	0805	2
Inductor	18 nH	0603	1
Inductor	10 nH	0603	1
Inductor	220 nH	0805	1
Capacitor	1.8 pF	0805	2
Capacitor	100 pF	0603	2
Capacitor	290 pF	0603	2
Capacitor	330 pF	0805	8
Capacitor	2.2 μF	0805	2
Amplifier	HMC308	SOT26	1
Amplifier	HMC414MS8GE	MSOP8	2
Mixer	HMC272AMS8	MSOP8	1
Filter	2450BP15B100E	0805 4 PC Pad	1
SMA connectors	-	-	3
Power supply connector	-	2.54 mm	1

Table 8: Bill of materials for the Rx board.

Name	Value	Package Size	Quantity
Resistor	100 Ω	0805	1
Inductor	15 nH	0603	1
Inductor	82 nH	0805	1
Capacitor	100 pF	0603	3
Capacitor	228 pF	0805	2
Capacitor	172 pF	0805	1
Capacitor	1000 pF	0603	1
Capacitor	2.2 μF	3216	1
LNA	HMC286E	SOT23-6	2
Mixer	HMC422MS8E	MSOP8	1
Amplifier	HMC474SC70E	SC70-6L	1
BPF 2.45GHz	2450BP15B100E	0805 4 PC Pad	2
SMA connectors	-	-	3
Power supply connector	-	2.54mm	2

B WiFi noise evaluation

Table 9: Evaluation of WiFi noise. Location: MC2 - Canyon. Measurements: 08 Dec. 2012. Frequency: 2.4 GHz. Detected Networks: NOMAD, eduroam. In the following table, the power [dBm] measured for each point is shown.

Eval. point	Ch. 1	Ch. 6	Ch. 11
1	-75	-95	-85
2	-72	-95	-82
3	-62	-85	-70
4	-59	-68	-65
5	-55	-68	-65
6	-63	-80	-65
7	-55	-68	-82
8	-60	-80	-83
9	-70	-80	-75
10	-65	-80	-70
11	-68	-80	-65
12	-68	-85	-70
13	-70	-90	-80

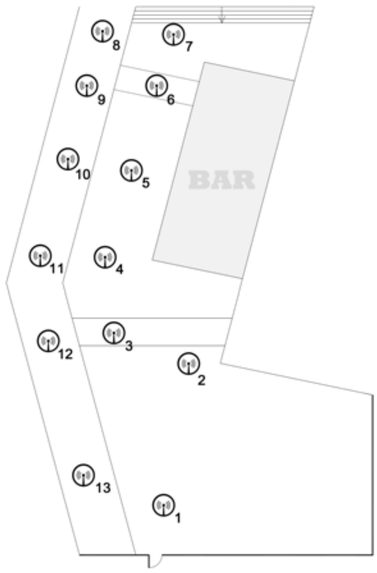


Figure 23: Positions for noise evaluation in Canyon.