# SDR4all: a Tool for Making Flexible Radio a Reality

Leonardo S. Cardoso, Raul Lacerda Neto, Pierre Jallon, Merouane Debbah

# SDR4all: a Tool for Making Flexible Radio a Reality

Leonardo S. Cardoso[1], Raul de Lacerda[1], Pierre Jallon[2] and Mérouane Debbah[1,2]

1: SUPELEC, Gif-sur-Yvette, France
2: SDR4all, Paris, France

**Abstract:** In this contribution, we describe the potential of SDR4all (Software Defined Radio for all) to solve the bottlenecks and reduce the innovation cycle related to the design and implementation of flexible radio algorithms. SDR4all is a programmable software tool with radio cards for wireless researchers, students and engineers. It enables to implement in software any wireless scheme between two laptops. Using an flexible orthogonal frequency division multiplexing (OFDM) based implementation example, we describe the tool and show the performance of the transmission on a real wireless channel at 2.4 Ghz ISM band.

**Keywords:** SDR4all, flexible radio, SDR, OFDM

## 1 Motivation

One of the problems that make the conception to development cycle so slow in research and development (R&D), particularly in the telecommunications field, is the fact that there is a big gap between the theory and the practice. Telecommunication theorists usually ignore critical aspects of the implementation of radios that are usually overlooked in telecommunication courses. Bridging the gap between theory and practice can reduce the time-to-market of ideas.

Matlab® has been proactive in this sense, offering tools and means to shorten the development cycle and allowing engineers to test their conceptions right inside their interface. However, in the area of telecommunications, not many options fill this need. While solutions do exist, they are generally hard to use and/or expensive. Existing projects and platforms (such as [1, 2, 3, 4]) aim at a building a configurable radio platform for the implementation and test of new technologies. In spite of their impressive capabilities, they are generally restricted to the use by professionals due to complicated programming languages (C, Assembly, VHDL, etc...), expensive hardware and testing equipment, lack of general knowhow in hardware development and radio frequency (RF) circuitry and to the fact that porting high level code to existing RF platforms is very time consuming. The need by telecommunication researchers/engineers/students for an easy to use tool for simple algorithm testing are left uncovered. Briefly, their needs are:

- Stress-test ideas with realistic constraints;
- Use real “models” of channels;
- Provide a proof of concept of an algorithm;
- Become acquainted with the issues related to radio systems;
- Create a practical view of the problems targeted;
- Verify the validity of the initial problem assumptions;
- Analyze the feasibility of the algorithms.

In this contribution, we propose SDR4all [5] as a solution to fill this gap. The SDR4all is based on the concept of software defined radio (SDR) [6, 7, 8] to enable the implementation of algorithms that will be tested with real transmissions using actual hardware transceivers. The idea of this testbed is to enable students/researcher to test their ideas and algorithms on real transmissions while still keeping the simplicity of a high level programming language environment.

In the remainder of this work we will present the evaluation tool in detail in section 2 . We will then present an example transmission/reception chain and show its characteristics and features in section 3 as a means to showcase the capabilities of SDR4all. We then discuss about its metrics and results in section 4. We conclude this contribution in section 5

## 2 SDR Evaluation Tool

SDR4all is composed of a USB plug and play hardware part, and a software part. The hardware part is in charge of the RF and sampling processing while the radio transceiver's physical layer (PHY) is software driven, running in a computer.

In order to give a practical example, the hardware of the testbed was based on the USRP version 1 cards [9]. However, SDR4all can be used on other types of radio cards (such as the ones being developed by SDR4all [5]). These cards are divided into two parts: a mother-board and one or two daughter-boards. The mother-board is responsible for the RF control, communication over the USB link, analog-to-digital/digital-to-analog conversions and sampling. The daughter-board is responsible for the RF circuitry, including filters, amplifiers and oscillators. Note that the USRP version 1 accepts a wide range of daughter-boards, available to allow testing in multiple bands [9].

As of now, SDR4all supports the RF circuitry provided by the RFX2400 daughter-board. The 2.4 GHz ISM band was chosen since it shares the same characteristics with the widely popular 802.11(b/g), bluetooth and WiMAX systems. Along with the RFX2400 daughterboard, dual VERT2450 antennas (TX and RX, each) are adopted. These are standard isotropic antennas made for the 2.4 and 5 GHz ISM band. The main parameters for the mother-board and RF circuitry are provided in table 1 [9].

| **parameter** | **value** |
|---|---|
| operating band | ISM 2.4 $\sim$ 2.49 GHz |
| base-band filtering | 20 MHz |
| channels | 1 to 13 (802.11) |
| total TX power | 20 mW |
| signal bandwidth | up to 16 MHz |

Table 1: Parameters for the hardware part.

As previously stated, the PHY layer is implemented in software, part of a Matlab®toolbox developed specifically to this end. In order for the toolbox to correctly communicate with the cards, a driver was built. At the moment, the driver enables non real-time communication between only a card and a computer, but multi-card, real-time operation is currently under development. It is able to transmit a given vector of baseband I/Q symbols and to listen for a certain amount of time and provide a vector of the received baseband I/Q samples. The driver also allows to configure the bandwidth and the center frequency for both transmission and reception. The toolbox implements basic communication blocks, such as bit operations, modulation, packet formatting and so on. Full communication chains are also implemented. The overall schematics of SDR4all, for an OFDM chain, can be seen in figure 1.

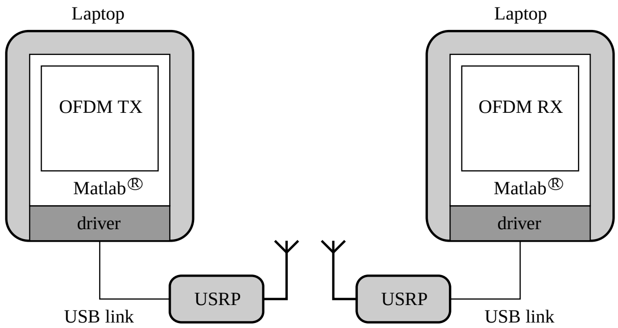


Figure 1: SDR4all overall layout

## 3 OFDM Radio Chain

The demonstration is based on the OFDM transmission chain scheme currently available in the toolbox. The OFDM transmission chain is able to transmit variable size frames, comprised of two parts as seen in figure 2: a preamble and a payload. A data bearing mode and a channel sounder mode can be configured and will be further detailed. The preamble of size $n_{pre}$ symbols is transmitted as binary phase shift keying (BPSK) ones (1s) in time. It is used to assure correct detection of the frame as a whole and also to correctly detect phase offset variations due to imperfections in the phase lock loop (PLL), responsible for generating the carrier frequency at the chosen central frequency $f_c$.

Consider a received vector of complex samples $\mathbf{s}$ of size $N$. Frame detection is achieved by means of a $M$-sized window energy detection of the form

$$e_n = \sum_{m=1}^{M} |s_{n+m}|, \quad \forall\, n \in \{1,\ldots,N-M\},$$

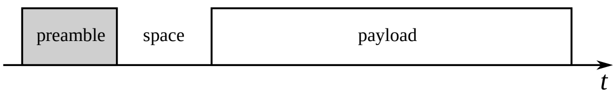


Figure 2: General OFDM transmission chain frame configuration.

where $|\cdot|$ is the norm operator, $s_k$ is a complex baseband received sample at instant $k \in \{1,\ldots,N\}$. For the choice of the window size $M$, we have picked the frame size. The frame is then decided to start at index

$$k^\star = \underset{n\in\{1,\ldots,N-M\}}{\text{argmax}} \ (e_n).$$

Although, maybe not the best frame detection technique, the energy detector is simple, computationally fast and provides rather robust results.

Once the frame is detected, the phase offset estimation takes part. To estimate the phase offset, we make use of the fact that the preamble is composed of same-phase symbols. Phase offset is estimated by

$$\hat{\phi} = \frac{1}{n_{pre}} \sum_{m=k^\star}^{k^\star+n_{pre}-1} \left(\angle[s_{m+1}] - \angle[s_m]\right),$$

where $\angle(.)$ is the phase of a given complex value. Note that each subsequent phase offset calculation is averaged out to compensate for phase noise. The authors are aware that other techniques better adapted to the same task exist, but this one was chosen due to its simplicity. After the preamble, a silence gap of the same size of the preamble is inserted in order to allow for the estimation of the noise power. The phase offset of the whole frame is later corrected by

$$s_m^\star = \frac{s_m}{e^{i\hat{\phi}(m-k^\star)}}, \quad \forall\, m \in \{k^\star,\ldots,k^\star+M\}.$$

Still in the structure of the frame, a silent space is inserted between the preamble and the payload. This silence space is used for noise energy estimation for signal-to-noise ratio calculations.

As stated before, two OFDM modes can be used, each of which will configure the payload part of the frame differently. In the following we provide details on both modes.

### 3.1 Data Bearing Mode

A data bearing mode can be used to transmit useful data. The frame structure for the data bearing mode is shown in figure 3. It is capable of carrying packets of different sizes. The payload is composed of pilot and data blocks sandwiched into the discrete Fourier transform (DFT)-modulated part of the frame.

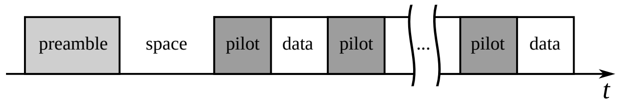


Figure 3: OFDM frame structure for data bearing mode

The fast Fourier transform (FFT) size of the packet is of $n_{FFT}$, of which $n_{zpad}$ carriers are reserved for zero-pad purposes, leaving a total of $n_c = n_{FFT} - n_{zpad}$ carriers for useful transmission. The payload part of $\mathbf{s}$ is reorganized into an OFDM symbol matrix $\mathbf{R}$ of size $n_c \times M/n_{FFT}$ given by rearranging $\mathbf{s}$ into a matrix of $n_{FFT} \times M/n_{FFT}$ and then discarding the middle $n_{zpad}$ rows. The $\mathbf{R}$ matrix is finally divided into a series of pilot $\mathbf{R}^P_{n_c\times n_{pilot}}$ and data $\mathbf{R}^D_{n_c\times n_{data}}$ sub-matrices.

The reference pilot block $\mathbf{P}$ is a randomly-generated symbols sequence, shared by both the transmitter and receiver. The pilots can be either modulated with a quadrature phase shift keying (QPSK) or be complex circularly symmetric. They are used to estimate the channel for subsequent equalization of the data symbols. The channel is estimated by

$$\hat{h}_i = \frac{1}{n_{pilot}} \sum_{j=1}^{n_{pilot}} \frac{R^P_{i,j}}{P_{i,j}}, \quad \forall\, i \in \{1,\ldots,n_c\}, \tag{1}$$

where the averaging is performed to smooth out the noise imperfections.

The data symbols are modulated with a 16-quadrature amplitude modulation (16-QAM) and transmitted. The sequence "pilot-data" is repeated until all the data to be transmitted is inside the payload. At reception the data is equalized by

$$\hat{D}_{i,j} = \frac{R^D_{i,j}}{\hat{h}_i}, \quad \forall\, [i \in \{1,\ldots,n_c\},\ j \in \{1,\ldots,n_{data}\}].$$

Each data block is equalized by the channel estimation from its preceding pilot block. The parameters for the data bearing mode are summarized in table 2. All parameters therein are selectable.

| parameter | value |
|---|---|
| $f_c$ | flexible inside the ISM band |
| base band rate | 1 MHz |
| $n_{pre}$ | 1000 syms |
| $n_{FFT}$ | 64 |
| $n_{zpad}$ | 16 (8 at each end) |
| $N$ | 48 carriers |
| $n_{pilots}$ | 30 |
| $n_{data}$ | 30 |
| pilot structure | QPSK or circularly symmetric |
| data modulation | 16-QAM |

Table 2: Parameters for the data bearing mode.

### 3.2 Channel Sounder Mode

The OFDM chain also has a channel sounder mode, which is used to provide channel analysis. The frame structure for the channel sounder mode is shown in figure 4.

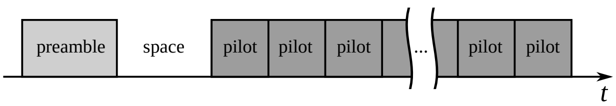


Figure 4: OFDM frame structure for channel sounder mode

In this mode, the payload is composed of pilots only. Similar to the data bearing mode, the FFT size of the packet is of $n_{FFT}$, of which $n_{zpad}$ carriers are reserved for zero-pad purposes, leaving a total os $N$ carriers of useful symbols for channel estimation. A total of $n_{pilot}$ OFDM symbols are transmitted in each pilot block. Again, these pilots symbols can be either modulated with QPSK or be complex circularly symmetric, known by both the transmitter and receiver. Channel estimation is carried out as in (1).

The parameters for the data bearing mode are summarized in table 3. Again, all parameters therein are selectable.

| parameter | value |
|---|---|
| $f_c$ | flexible inside the ISM band |
| base band rate | 1 MHz |
| $n_{pre}$ | 1000 syms |
| $n_{FFT}$ | 64 |
| $n_{zpad}$ | 16 (8 at each end) |
| $N$ | 48 carriers |
| $n_{pilots}$ | 30 |
| pilot structure | QPSK or circularly symmetric |

Table 3: Parameters for the channel sounder mode.

### 3.3 Reception Trigger

Due to its non real-time constraint, the receiver can not decode the transmitted stream on the fly to detect the signal. That would require much more processing power and memory than Matlab®can cope with at the base-band rate. Thus, a triggering system was implemented to initiate the reception, to minimize the number of samples processed so that the detection of the packet can take place. To that end, the TCP/IP network was adopted as a triggering bearer. In the absence of a wired network, the wireless network can be used. In the case of a frequency superposition, the transmission/reception takes places one full second after the trigger has been sent to guarantee that the wireless network will not interfere with the toolbox packet transmission.

## 4 Metrics and Outputs

One of the best advantages of SDR4all relies on the fact that it processes actual base-band symbols. This allows for the creation of well tailored kinds of results and metrics, useful for either demonstrating the inner workings of the physical layer or for assessing the performance of an algorithm. In particular, this flexibility can be very useful for teaching and research scenarios.

As an example, we showcase the results provided by the OFDM chain. In addition to bit error analysis, symbol error analysis and signal to noise ratio (SNR), the OFDM chain also outputs the constellation of the data (shown in figure 5), estimated channels (in figure 6) and can also present the output of the transmitted data, in our case an image, as shown in figure 7.

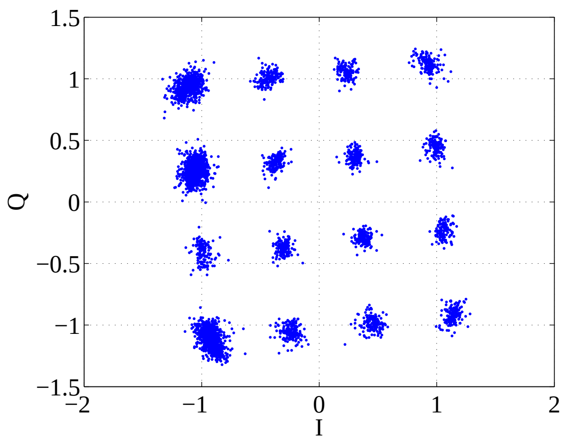


Figure 5: Constellation of the received symbols

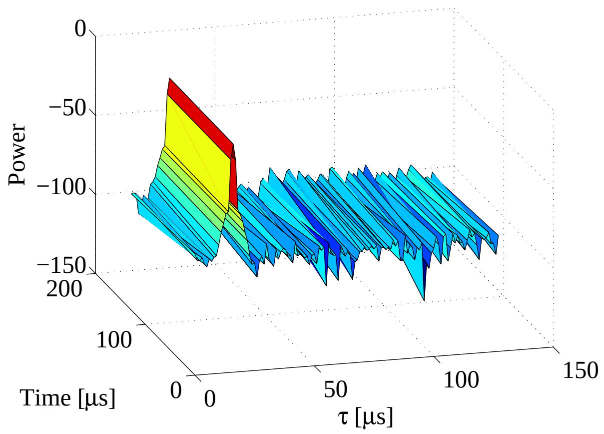


Figure 6: Channel impulse response

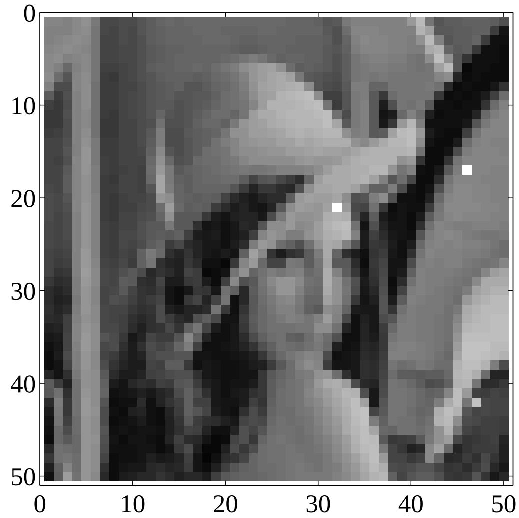


Figure 7: Received Lena image

With access to the transmission/reception chain code, the user can also implement new metrics that suits his needs.

## 5 Conclusion

In this work, we have presented a new and easy to use SDR-based tool called SDR4all, able to empower telecommunications teachers and researchers towards the common goal of creation of state-of-the-art algorithms and techniques. Then we have shown how SDR4all is composed, providing details on its hardware and software structure. We have also provided an example of the capabilities of SDR4all, based on a OFDM-based transmission chain, highlighting its structure, configurability and available metrics and results. SDR4all is already being employed in universities and research labs and has proven to live up to its demands.

## 6 Acknowledgements

The authors would like to thank Digiteo for its support.

## References


[1] http://www.openairinterface.org/overview.en.htm. 2007.

[2] W. Zhu, B. Daneshrad, J. Bhatia, J. Chen, H.S. Kim, K. Mohammed, O. Nasr, S. Sasi, A. Shah, and M. Tsai. A real time MIMO OFDM testbed for cognitive radio & networking research. In *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 115–116. ACM New York, NY, USA, 2006.

[3] http://gnuradio.org/trac. 2005.

[4] http://www.spectrumsignal.com/products/pci-x/sdr_2000.asp. 2005.

[5] http://www.sdr4all.org. 2008.

[6] J. Mitola III and GQ Maguire Jr. Cognitive radio: making software radios more personal. *IEEE personal communications*, 6(4):13–18, 1999.

[7] J. Mitola et al. Cognitive radio: An integrated agent architecture for software defined radio, 2000.

[8] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, 2005.

[9] http://www.ettus.com. 2006.