

Active learning in digital communications with low-cost software defined radio

Antonio Soriano-Asensi, Carmen Botella-Mascarell, Jaume Segura-García, Sandra Roger

Computer Science Department, University of Valencia, Spain.

Abstract

Digital Communications have been traditionally taught in laboratory sessions from a theoretical point of view, using simulation platforms. However, current Academic Accreditation includes the dimension of “what students are expected to be able to do”, which poses the need of decreasing the gap between standard simulated laboratory sessions and more practical and realistic approaches. In this paper, we propose a methodology to enhance the learning of practical aspects related to Digital Communications courses, as well as increasing the student’s engagement, via the use of low-cost software defined radio devices. This methodology has been applied in the Degree in Telematics Engineering at the School of Engineering from the University of Valencia, Spain. With the aim of improving the engagement of both master and undergraduate students, a project based learning methodology has been implemented. In the paper, the methodology is described, several laboratory activities based on the 802.11 standard are presented, and recommendations for practice are given.

Keywords: *Software defined radio; SDR; digital communications; signal processing; HackRF One; USRP.*

1. Introduction

Telecommunication Engineering degrees, as well as Electrical and Electronic Engineering Degrees, include Digital Communications as one of the core blocks of their curricula. In the case of the School of Engineering at the University of Valencia (ETSE-UV), Digital Communications and Signal Processing for Communications stand for a 20% (24 European Credit Transfer System (ECTS) units) of the mandatory subjects and a 10% of the Degree in Telematics Engineering (DTE) (240 ECTS). Traditionally, these core blocks have been taught from a theoretical point of view, using the basic blocks of communication systems as a story line, i.e., source models, source coding/decoding, channel coding/decoding, modulation/demodulation, channel models and sink as described in Proakis and Salehi (2009).

Digital Communications courses usually include standard laboratory sessions where students can simulate parts of the system using Matlab, Python or C interfaces. In this paper, we discuss the potential of Software Defined Radio (SDR) platforms to narrow the gap between theoretical and practical implementation aspects in the framework of Digital Communications systems. The concept of SDR, Mitola (1992), establishes a new educational paradigm in the field of telecommunications, enabling the implementation of many components of radio communication systems using software, which, in addition, can be reconfigured in real-time. There is a broad range of SDR devices with different capabilities, as well as several software programs (e.g., LabVIEW, Matlab or GNU Radio (www.gnuradio.org/)) to support and manage these platforms. Universal Software Radio Peripherals (USRP) (<https://www.ettus.com>) are the most used and versatile devices. However, in general, USRP are quite costly devices, and several institutions are working towards low-cost solutions. For example, Stewart, Crockett, et al. (2015) developed an open courseware with RTL-SDR very low-cost devices (<https://www.rtl-sdr.com/>) (less than \$20). In this case, MATLAB and Simulink were used to model digital communication systems. Work by Linn (2012) also targeted a communications laboratory with low-cost SDR capabilities. In this solution, Xilinx Spartan 3A Field Programmable Gate Array (FPGA) was considered (unitary cost of \$200 each).

In the case of the DTE, each undergraduate course usually comprises 4 laboratory groups with 16-20 students each, which stands for an average of 40 SDR devices needed per course (courses can run in parallel). In our case, RTL-SDR devices have been used together with GNU Radio in introductory Digital Communications courses to increase the engagement of the students within an affordable budget as shown in Soriano-Asensi (2019). However, RTL-SDR devices pose some reliability problems, especially when connected during larger times due to power dissipation problems. Due to this, HackRF One devices from Great Scott Gadgets (greatscottgadgets.com/hackrf/), with a unitary cost of around \$300, have been acquired and integrated into more advanced courses at the Master's Degree in

Telecommunications Engineering, with a lower number of students (around 15 students in average). In this paper, we propose a Project Based Learning (PBL) methodology to improve the learning of practical aspects related to Digital Communications courses via low-cost SDR, which has proven to enhance the engagement of both master and undergraduate students. More precisely, we explore the use of HackRF One from Great Scott Gadgets with an open source software tool such as GNU Radio.

2. Methodology and materials

The methodology used in this paper is based on the analysis of the requirements of the theoretical-practical development of laboratories for different subjects at undergraduate and master levels. Digital Communications contents are distributed in the different courses (undergraduate and master levels) at the ETSE-UV's Telecommunication studies in a bottom-up approach: digital modulations (in baseband and passband), channel equalization techniques to avoid monocarrier and multicarrier inter-symbol interference (ISI), source coding and channel coding techniques, synchronization techniques, diversity and spread spectrum techniques, etc. Student's perception of the use of SDR was caught by a poll launched in different courses in the academic year 2015-2016 as shown in Segura-Garcia (2016). In this survey, students were asked about their perception regarding the practical knowledge gained via the use of the SDR, trying to assess the need of a higher experimental load. The introduction of the SDR concept supported a better perception of these courses allowing a reduction of the abandon rate.

In this contribution, we explore the potential of the combination of low-cost SDR devices with a PBL strategy. In the academic year 2019, master students registered in the Advanced Signal Processing for Communications course were asked to develop projects based on GNU Radio and HackRF peripherals. With the aim of benefiting from a peer instruction-like scheme, these projects were later adapted by the master students to be performed as laboratory sessions for their undergraduate colleagues at the Digital communications course. Note that standards such as WiFi (802.11), IEEE (2016) are very useful, since their functional blocks show a good overlap with the contents of the courses participating in this methodology. In this framework, the motivation and engagement of both master and undergraduate students are increased. We take as references the cases of the University of Seville, Pinar-Domínguez and Murillo (2011) and the University of Washington, Wyglinski, Pu, et al. (2011), which already incorporate these elements to their teaching, but without the combination with active learning methodologies. We oriented our project to use HackRF One, a low-cost platform with enough feasible characteristics for different complex projects and standard implementations. At the same time, GNU Radio is an open-source solution with a good community support and offering a block-based graphical interface which greatly simplifies the learning curve.

3. PBL in digital communications

PBL is a style of active learning that places the student in the center of the learning process. Active exploration of real-world challenges allows students to acquire a deeper knowledge. The learning process in PBL is led by students while the teacher's role is to guide student's research by introducing questions and tasks that help them to solve the proposed problem. The teacher is expected to introduce the problem and provide students with the materials and documentation required to start the project. Students are expected to assume the leading role of the project as the activity goes on.

3.1. Project preparation, instruction and initial tasks

PBL was implemented in a master's course because of the larger experience of students compared to undergraduate ones. Moreover, the number of master students per course (typically 8-10) is smaller than the 16-20 students per laboratory group in undergraduate courses. Students were organized in pairs, and all of them were asked to design a testbed to evaluate the performance of an 802.11 system. They were provided with two HackRF One per group, an Ubuntu 16.04 Virtual Machine (VM) with GNU Radio and the link to the github project implementing the 802.11p standard (*gr-ieee802-11*, <https://github.com/bastibl/gr-ieee802-11>), as detailed in ETSI (2009). During the project, students were given access to the communications laboratory where there was a workplace with a computer, which had the same software that was installed in the VM. The workplace was shared by all the groups, so they had to take turns to access the laboratory. The VM allowed students to conduct part of the activity at home, while the workplace at the communications laboratory permitted to test their advances with the SDR hardware.

Students were expected to conduct part of their research outside teaching hours, while the time in class was devoted to check the evolution of each group and to help them to continue with their work. The activity required three class sessions (3 hours each). The first session was used to introduce the project to the students. They were instructed with the VM, the *gr-ieee802-11* github project, and its documentation. The SDR project described in ETSI (2009) is based on USRP devices, while the project was expected to be done with HackRF One. In this first session, the teacher instructed students about the differences between the USRP and the HackRF One. The task proposed to the students in this first session was to adapt the *gr-ieee802-11* project to be used with the HackRF One. One of the drawbacks of the HackRF One is that it introduces a Direct Current (DC) spike in the received IQ signal. Students were advised to install the *gr-correctiq* (<https://github.com/ghostop14/gr-correctiq>) github project, and to use the *correctIQ* block in combination with the *osmocom-* source in order to remove the DC spike introduced by HackRF One.

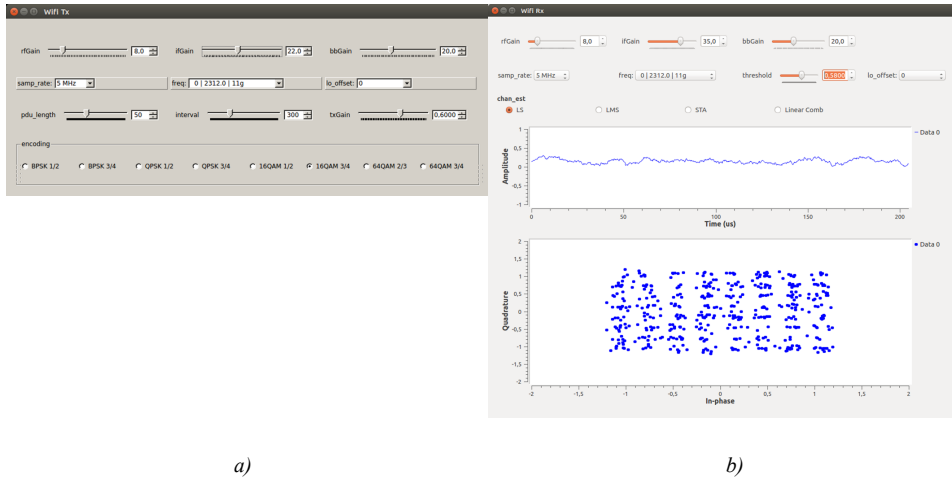


Figure 1. a) Image of the transmitter GUI, with the controls that allow to choose the characteristics of the transmission. b) Image of the receiver GUI. The receiver controls are placed in the upper part of the GUI, the middle part shows a time plot of the signal used to determine the arrival of a new packet. The lower part shows the symbols received in the last packet (from a 64-QAM modulation).

The resulting Graphical User Interfaces (GUI) developed by the students for the transmitter and receiver are shown in Figure 1. Figure 1.a shows the GUI that allows to adjust the transmission parameters, consisting of four rows of controls. The first one contains the three adjustable gains for the RF, intermediate frequency and baseband stages of the HackRF One. The second row of controls permits to set the sampling and channel frequencies. The controls in the third row are used to adjust the packet size and the time interval between consecutive packets. Finally, the desired modulation can be selected from the controls in the fourth row. Figure 1.b shows the GUI with the controls of the receptor. The upper part contains three rows of controls to adjust the performance of the receiver. The controls in the first and second rows of controls are similar to that shown in Figure 1.a for the transmitter. A threshold control was added in the second row, its function is to set the threshold that triggers the detection of an incoming packet. The third row of controls permits to choose the channel estimator among Least Squares (LS), Least Mean Squares (LMS), Spectral Temporal Averaging (STA) and COMB type. The central part of the receiver GUI (Figure 1.b) plots the time variation of the signal at the output of the short sequence detector, which is used to trigger the receiver on the arrival of a new packet. Finally, the lower part of the GUI shows the constellation of received symbols.

The controls in transmission and reception GUIs and the information shown in the reception GUI allowed the students to adjust the parameters for each modulation scheme. As a result, students were able to fix the gains of the transmitter and receiver HackRF One. The *gr-ieee802-11* project allows to set the sampling frequency either to 5 MHz or 10 MHz. But, considering that the maximum sampling frequency of the HackRF One is 20 MHz and in

order to reduce the computing requirements of the receiver, the students concluded that it was more appropriate to set the sampling frequency to 5 MHz. Students also evaluated which was the most suitable channel to conduct their experiments, since the presence of other WiFi Access Points (AP) interfered with the experiments. Better results were obtained in general when the lowest or the highest channels in the 2.4 GHz band were used. The number of AP in the 5 GHz band was smaller than in the 2.4 GHz band. However, a worse performance of the HackRF One was appreciated at 5 GHz. Considering the better performance of HackRF One at 2.4 GHz and in order to avoid interference from neighboring AP, the experiments were conducted at 2.3 GHz.

3.2. Student leadership

The teacher's role in PBL is to introduce the project and to suggest several tasks aimed at guiding students' research. Students are expected to progressively take the leadership of the work. This part of the PBL methodology lasted the second and third laboratory sessions, depending on the degree of autonomy of each group of students. The activity proposed to the students at this stage was reading a file and sending it through the wireless channel in order to evaluate the percentage of transmission errors. The chosen text was the cavalry book *El Quijote* from Miguel de Cervantes (1605). The comparison between sent and received texts showed that differences were due to packet losses. In order to automate the analysis, the wireshark connector block available in the *gr-ieee802-11* project was used to create two .pcap files, with sent and received packets, that were processed afterwards with a Python script in order to calculate the error rate (see Table 1). When using the 16 QAM and 64 QAM modulations, it was required to enable the RF amplifier of the HackRF One used for the transmission, while it was not required in the remaining modulations. Three different packet sizes were considered in order to evaluate their contribution to the error rate. In general, the error rate increased with the packet size. The high increase of error rate between 16 QAM and 64 QAM modulation suggests that the channel quality was not good enough to work with 64 QAM, since most packets were lost in these cases. Larger error rates were obtained with BPSK modulations than those observed with QPSK and 16 QAM, indicating that a deeper understanding of the implementation is required to determine the cause of such an unexpected observation.

Table 1. Error rate for the available modulations considering several packet sizes. Second, third and fourth columns stand for the packet size in bytes.

Modulation	Packet size (500)	Packet size (1000)	Packet size (1500)
BPSK 3/4	0.08	0.14	0.22
QPSK 3/4	0.04	0.08	0.12
16 QAM 3/4	0.07	0.10	0.08
64 QAM 3/4	0.85	0.97	0.99

5. Conclusion

Prior experience of the authors revealed the students' demand of more applications oriented Digital Communications and Signal Processing laboratory sessions. In this sense, software defined radio has proven to be an excellent tool to introduce real applications when teaching Digital Communications. This work presents the project-based learning experience conducted with students of the Master of Telecommunications at the School of Engineering from the University of Valencia, where they were proposed to design a testbed to evaluate the performance of a WiFi transmission system. This active learning activity contributed to position the students in the center of their learning process. In addition, the combined use of software defined radio and project-based learning has helped to increase the engagement of master students in the laboratory activities. From the results achieved by the master students, valuable information has been gathered in order to design similar and more guided activities for undergraduate students.

References

- ETSI. (2009). ETSI EN 300 744 V1.6.1. *Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television*, Jan 2009.
- IEEE. (2016). Standard for Information technology Telecommunications and information exchange between systems Local and Metropolitan area networks. Specific requirements. Part 11: Wireless LAN MAC and Physical Layer (PHY) Specifications.
- Linn, Y. (2012). An ultra low cost wireless communications laboratory for education and research, *IEEE Trans. on Educ.*, 55 (2), 169–179.
- Mitola, J. (1992). Software radios-survey, critical evaluation and future directions, in Proc. *IEEE National Telesystems Conference (NTC'92)*, Washington DC, USA, 15–23.
- Pinar-Domínguez, I. & Murillo Fuentes, J.J. (2011) *Laboratorio de Comunicaciones Digitales Radio Definida por Software*. TSC - Ed Universidad de Sevilla. <https://personal.us.es/murillo/docente/Libros/LibroSDRV7USv8.pdf> (Accessed on: 13/02/2021)
- Proakis, J.G. & Salehi, M. (2009). *Digital communications*, 5th ed., Boston: McGraw-Hill, 2009. ISBN 9780071263788.
- Segura-García, J., Botella, C., Soriano-Asensi, A. & Felici Castell, S. (2016). Innovación en docencia de sistemas de comunicación en el Grado de Ingeniería Telemática de la UVEG. *In-Red 2016. II Congreso nacional de innovación educativa y docencia en red*.
- Soriano-Asensi, A., Segura-García, J., Botella, C., Perez, J. & Felici, S. (2019). Aprendizaje basado en proyectos en los laboratorios de comunicaciones digitales. *In-Red 2019. V Congreso de innovación educativa y docencia en red*.
- Stewart, R.W., Crockett, L., Atkinson, D., Barlee, et al., (2015). A low-cost desktop software defined radio design environment using MATLAB, Simulink, and the RTL-SDR, *IEEE Commun. Mag.*, 53 (9), 64-71.

Wygłinski, A.M., Pu, D. & Cullen, D.J. (2011) Digital communication systems education via software-defined radio experimentation, in *Proceedings of the 118th ASEE Annual Conference and Exposition Vancouver*, BC, Canada.