Software Defined Radio: From Theory to Real World Communications

Víctor González-Barbone, Pablo Belzarena and Federico Larroca
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, Universidad de la República
Montevideo, Uruguay
{vagonbar,belza,flarroca}@fing.edu.uy

Abstract—**Bridging the gap from theory to real-world, hands-on experience is an always present challenge in Education. The advent of computers allowed for simulation, a halfway step from theory into practice. In Telecommunications Engineering, Software Defined Radio allows students to design and test wireless equipment in real communication by means of a personal computer connected to a simple, low cost, generic radiofrequency device. The paradigm has changed the approach to wireless equipment and communications protocols design across industry and academia. In Education, it has succeeded in providing students with an experience very similar, if not equal, to what they will face in their professional lives. This approach resulted in a deeper understanding and better ability development than was possible through simulation, since several communication problems (such as those related to radiofrequency propagation) are very difficult to model realistically. Moreover, this approach has proved to be strongly motivating for students, who design systems through a graphical interface by interconnecting blocks with specific functionality, but can later experience the results in real communications. This article describes our teaching experience and lessons learned with Software Defined Radio for Engineering Education. We used GNU Radio, an open and free software framework, complemented with GNU Wireless Network, an extension of our design to support data communications. The paradigm involved both graduate and undergraduate courses in Wireless Communication with strong lab content, undergraduate projects, early research training, and several master thesis.**

Index Terms—**Engineering Education, Software Defined Radio, GNU Radio, Wireless Communications, Wireless Data Networks, Research Training**

I. Introduction

Engineering Education faces the ever present challenge of preparing students for real world situations in an environment which only imitates reality: models and lab experiments are very mild experiences compared to the demands of real life situations.

Wireless communication problems are very difficult to model realistically; some experimentation is needed. Mounting a Communications Engineering laboratory calls for a variety of communications equipment which is specific for different purposes, proprietary, expensive, and sometimes very expensive.

The scaling in computer hardware and software capabilities brought in a new tool: simulation. Simulations have proved a valuable teaching tool, but realistic simulation at the physical level becomes extremely difficult.

Software Defined Radio (SDR) is a radio communications system in which functions traditionally implemented in hardware are instead performed by software, for example in an ordinary personal computer. In this way, a personal computer with an generic, low cost, USB connected radiofrequency device, can be used as a communications equipment capable of interacting in a wide variety of communications systems, such as television, AM and FM radio, and others.

Software Defined Radio has changed the way communications systems are designed: the paradigm has been adopted and is increasingly extending as the preferred research and development tool in industry [7], [10], [33]. Since the tools are within reach of even modest university budgets, besides bridging the gap between theory and practice, students work in an environment very similar to the one they will find in industry, which allows them to become productive almost immediately. Until recently, it was practically impossible to provide students with an industry-like experience within school.

GNU Radio (GR) [12] and GNU Wireless Network (GWN) [15] are open and free implementations of SDR; GR is oriented towards broadcast as in radio and TV, GWN is an extension towards data networks of our development. Both GR and GWN are apt for simulation, but they also work in real world communication. Students can thus experiment real world communication problems. Moreover, a well tested project can pave the way to an industrial development.

Computers have brought unexpected possibilities to Education. In this case, successfully bridging the gap from theory to practice, and re-creating in academia the same environment students will find in their professional lives have been long cherished achievements.

The former considerations led the authors and some colleagues to try SDR along with GR and GWN as educational tools. This article discusses our efforts towards their adoption, its advantages (and disadvantages) and lessons learned along the way. We also present several resources which may be easily duplicated in other Universities (such as GWN), in the hope that they will be useful for other similar undertakings.

This article is organized as follows. Software tools for SDR, hardware alternatives, and a review of some former experiences of their use in Education are first described. This

This work was partially funded by Universidad de la República's CSIC GRUPOS program.

is followed by an introduction to GR and GWN. The use, characteristics and some shortcomings of GR are analyzed in Sec. III. After a short summary on data networks needs, features provided by GWN are explained in Sec. IV, both to support data networks and to overcome some of the GR shortcomings. As we discuss in Sec. V, GR and GWN were used as teaching tools in courses, assignments, and students projects, but also in research. Results and discussion of these experiences in Sec. VI help define our future work in the design of upcoming courses, in graduate students projects, and in GWN enhancements. Some conclusions on how the SDR paradigm helped to face the challenges of Engineering Education end this article.

II. Software Defined Radio

A. Software tools

In SDR, instead of processing a continuous-time signal (such as the voltage generated by an antenna), periodic samples of this signal are instead processed by the personal computer. Samples are obtained from a generic wireless device connected via USB or Ethernet, which may be arbitrarily processed by the computer. Conversely, samples generated in the computer are transmitted through this device. This enables almost any type of wireless link, including AM/FM radio, analog and digital TV, cell phones, and WiFi.

GR and its extension, GWN, are free and open SDR implementations which provide a graphical interface where blocks can be interconnected to perform the necessary signal and message processing. GR was designed for and (mostly) implements continuous communications such as radio and TV, while GWN extends GR to support protocols for packet communications.

GR provides a set of blocks to perform well known signal processing tasks, as well as interact with the operating system of the computer and the communications hardware. One of these hardware devices is the Universal Software Radio Peripheral (USRP) by Ettus Research, a subsidiary or National Instruments [9].

Results obtained in a GR based testbed can be expected to improve in a hardware implementation of the same flowgraph, mainly because of the much faster time responses and the dedicated, specific character of the hardware implementation. This allows an inexpensive way of testing a prototype, avoiding the high costs of hardware prototyping, out of reach for many universities, specially in underdeveloped countries.

GR is free and open, and the repository can be installed locally using several different options [11]. GWN is also free and open and can be cloned locally from the Github repository [16]. Other tools such as the Python programming language and C++ are also free and widely available. GR, GWN and all our developments were carried out under the Linux operating system, also free and widely available; several versions exist, and were actually used, according to device capabilities, e.g. Raspbian [28] and Arch Linux [2] for small capabilities devices, XUbuntu [35] for PCs.

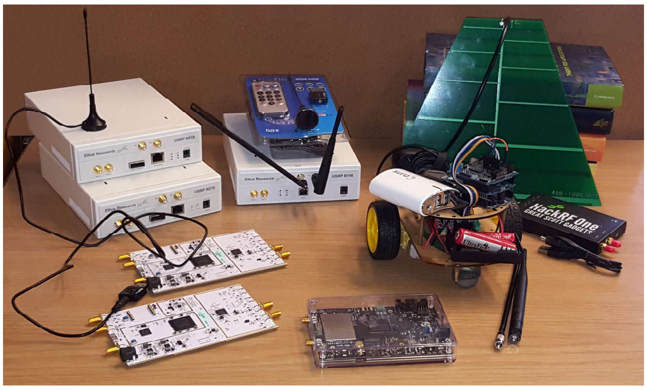


Fig. 1. Hardware devices used: USRP, BladeRF, HackRF, Raspberry Pi, robot.

B. Hardware alternatives

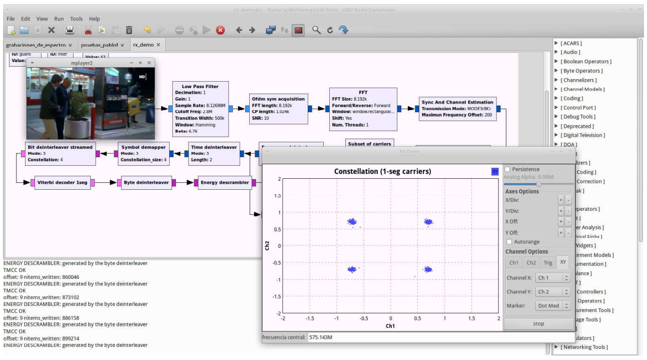
Usual RF peripheral devices for GR and GWN are: USRPs by Ettus Research, from $ 775 for the USRP bus series [9]; BladeRF from $ 420 [25]; Great Scott Gadgets HackRF One for $ 300 [30]; receive-only dongles can be bought by less than $30 [29]. The cost of a traditional RF equipment to test wireless communications may well go far into the thousands of dollars.

SDRs such as GR and GWN run on personal computers, but some undemanding developments can be run on smaller, lower cost devices. Raspberry Pi is a credit-card sized single-board computer built by the Raspberry Pi Foundation [27]; the latest model can be bought by $ 40; a complete starter kit costs $ 75. Raspberry Pi size and power requirements allow mounting it on an autonomous device such as a robot. A robot car kit can be bought by less than $ 100. GWN was installed on a robot car equipped with a Raspberry Pi and an USRP RF card; the robot could thus be controlled from a personal computer with a similar USRP RF card. Smaller devices are becoming increasingly powerful: the ODROID-XU4 compares with a personal computer, running Ubuntu Linux 15.04 or Android 4.4 in a size of 82 x 58 x 22 mm, fan cooler included, at a cost of $ 74.00 [18]. Figure 1 shows some of the hardware devices used in our experiments.

C. Some milestones of SDR in Education

The advent of SDR into Education was made possible mostly by two factors: increase in the capabilities of ordinary personal computers, and the reduction in cost of radio frequency peripherals, from year 2000 on [24]. Use of SDR in Education dates from 2010 and has consistently increased.

For instance, use of SDR for undergraduates held at the School of Engineering at Stanford University was considered very promising, and was ranked by the students among the best at the school [22]. In fact, several U.S. universities recognize the value of SDR as an integrative construct for the various disciplines involved in electrical engineering curricula [4]. Moreover, in 2012 and 2013 SDR drew the attention of the IEEE Global Communications Conferences (GLOBECOM), and the 2014 edition of IEEE Communications Magazine



Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 8.12698M
Cutoff Freq: 2.8M
Transition Width: 500k
Window: Hamming
Beta: 6.76

Ofdm sym acquisition
FFT length: 8.192k
CP length: 1.024k
SNR: 10

FFT
FFT Size: 8.192k
Forward/Reverse: Forward
Window: window.rectangular...
Shift: Yes
Num. Threads: 1

Sync And Channel Estimation
Transmission Mode: MODE3(8K)
Maximun Frequency Offset: 200

Subset of carriers

Bit deinterleaver streamed
Mode: 3
Constellation: 4

Symbol demapper
Mode: 3
Constellation_size: 4

Time deinterleaver
Mode: 3
Length: 2

Viterbi decoder 1seg

Byte deinterleaver

Energy descrambler

```
ENERGY DESCRAMBLER: generated by the byte deinterleaver
TMCC OK
offset: 9 nitems_written: 860046
ENERGY DESCRAMBLER: generated by the byte deinterleaver
TMCC OK
offset: 9 nitems_written: 873102
ENERGY DESCRAMBLER: generated by the byte deinterleaver
TMCC OK
offset: 9 nitems_written: 886158
ENERGY DESCRAMBLER: generated by the byte deinterleaver
TMCC OK
offset: 9 nitems_written: 899214
ENERGY DESCRAMBLER: generated by the byte deinterleaver
```

Fig. 2. gr-isdbt, an ISDB-T standard television receiver; the screen capture shows the GR graphical interface, the interconnection of blocks in a flowgraph, the constellation of symbols in the signal, and the TV image being analyzed.

started a feature topic on Communications Education and Training publishing three articles on SDR in Education [24].

The use of SDR in the limited time of undergraduate courses faces the challenge of a steep learning curve. This was overcome in the experience described in [34] by structuring learning units with theoretical preparation and simulation before facing GNU Radio SDR experimentation, together with the use of Simulink [23], a graphical programming environment for modelling and simulation. Students found labs rather excessive in terms of work, but also found the SDR paradigm quite interesting [34]. Regarding junior researchers, SDR has been successfully used to involve undergraduate studies in research and motivate them to follow graduate level studies [6].

A survey of the capabilities and challenges offered by the USRP hardware platform for Education can be found in [8]. All in all, SDR is nowadays recognized as a reliable tool for telecommunications engineering, having changed the way industry faces technical challenges in telecommunications systems. Rapid prototyping of radio solutions, and their testing in real-world conditions, together with their availability and ease of use, explain the value given to SDR in industry, academia and government [33] [26].

III. Working with GNU Radio (GR)

A. *GNU Radio in use*

In GR, a communications application is implemented as a flowgraph, i.e. a set of interconnected blocks. Each block performs a specific function, has some inputs and outputs, and parameters to be set. GR blocks are coded in C++, but Python can also be used, allowing for rapid prototyping. GR accepts development in both languages in a seamless fashion. These flowgraphs may be constructed by programming (either Python or C++) or through the so-called GNU Radio Companion (GRC), the GR graphical interface.

Figure 2 shows a running flowgraph, where a digital TV channel is tuned, and the “constellation” of symbols in the signal is shown (together with the received video). Some of the blocks used are part of gr-isdbt, a GR Out-of-Tree module (see the next subsection) for digital television reception [21].

The collection of blocks included in the standard GR distribution is quite varied, and allows to experiment signal processing tasks, emulate a communications link using a channel emulator block (no RF peripheral required), and establish real communications.

B. *GNU Radio modularity and extension*

The GR architecture has been designed to be extensible: users can add new blocks in groups called out-of-tree modules, which can be used exactly as blocks in the main stream of development, while not interfering with them [13].

Our own gr-isdbt project is an example of a pure GR research and development achievement. gr-isdbt is a GR based digital television receiver implementing ISDB-T, the standard used in most Latin American countries. The implementation is open, free, and entirely in software (see [1]), the only necessary piece of hardware being any of those mentioned before (even cheap dongles). This application allows broadcasting professionals and researchers to use a low cost, real-time working receiver on which several measures relevant to signal quality of reception can be made, thus avoiding the need for expensive digital television equipment [21].

This development was carried out by a team of senior and junior researchers, with several graduation projects and a master thesis carried out around the project. These achievements show the feasibility of promoting developments towards the early training of junior students as researchers. Moreover, the working receiver was used in several demonstrations of key concepts during teaching (e.g. OFDM, synchronization, BER, etc.), with the clear advantage of working with a communication system that is both real and well-known to students. Implementations of other DTV standards (e.g. DVB-T or ATSC) are already part of GNU Radio, so this kind of demonstrations may be carried out almost anywhere on the globe.

C. *GNU Radio shortcomings*

The use of GR in Education has to deal with some shortcomings of a project not primarily designed as a learning tool.

Documentation is scarce. There is a collection of tutorials to help new users install and use GR [14], and a number on some specific topics, but documentation on blocks is scanty, sometimes limited to programmers comments in the code [11].

Pace of development is fast, versions change frequently, compatibility with former versions is not guaranteed. Last versions tend to be more stable, though.

Though there exist data networks implementations based on GR, they are very specific, generally partially implementing a data communications protocol in particular, such as [5]. GWN attempts to provide a design paradigm and toolkit to ease the implementation of existing or experimental communications protocols. To this purpose, GWN adds timing facilities and a generic FSM (Finite State Machine), and a generic block to include them; this generic block can be specialized at will for any purpose, from demonstration to industry developments. Besides, GWN attempts to overcome some GR shortcomings for teaching in data network courses.

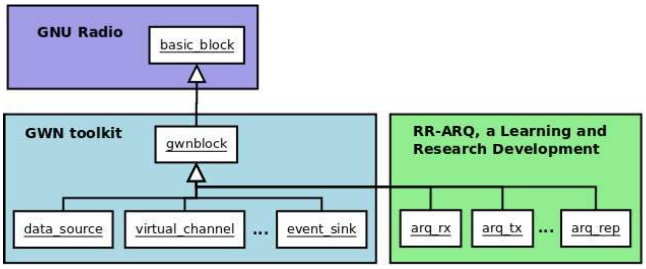


Fig. 3. The GWN toolkit provides a generic gwnblock which inherits from GR basic_block. All GWN toolkit blocks and blocks developed by students and researchers inherit from gwnblock. RR-ARQ is a students implementation of Automatic Repeat reQuest (ARQ), an error control protocol for data transmission.

IV. GNU Wireless Network (GWN)

A. Data networks requirements

GR and SDR come from radio frequency communications such as radio and TV, where the electromagnetic spectrum is divided in channels, and information is conveyed in a continuous flow. Data networks use a shared medium with several actors communicating at the same time, information is conveyed in discrete data units called packets, these packets may be corrupted, suffer variable delays, arrive in disorder, or be lost entirely. These and other problems are addressed in a number of standard which regulate network communication, from small local area networks to the Internet. IEEE 802.11 is one such protocol, commonly known as WiFi.

GR was originally stream oriented, but recently added support for message communications. GWN uses message communication to extend GR for its use in data networks, not tied to any specific protocol, but as a toolkit for experimentation and development.

B. GWN Architecture

GWN extends GR towards data networks in a toolkit with its own features. Blocks in GWN and GR can be mixed in the same flowgraph. GWN provides a generic gwnblock which adds the tools necessary for data network designs, decoupled from the GR generic basic_block. A new GWN block only needs to inherit from gwnblock and follow GWN design rules, shielding from users most of the complexity of GR.

The GWN toolkit also includes common function blocks such as message sources and sinks, a channel emulator, message converters, and framers. Figure 3 shows how learning and research developments need only interact with GWN in their construction. This architectural scheme simplifies access of students to development.

The GWN generic block adds the following facilities to GR.

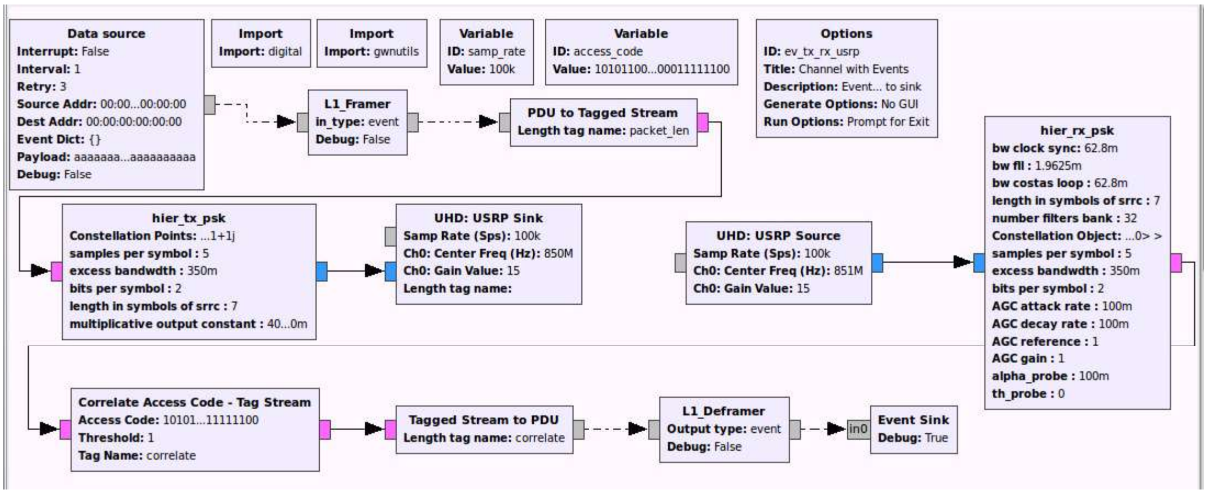
- Message orientation. GR is stream oriented, GWN is message oriented; items interchanged among GWN blocks are discrete groups of bytes. GWN makes use of the message mechanism of GR, but provides some blocks to interact with stream GR blocks when necessary.
- Events. GWN elaborates on the message interchange mechanism of GR into a more structured item of interchange called an event. GWN blocks interchange events. The event inner structure reflects the needs of network data protocols and is closer to their design conception.
- Handling of time. This is a feature absent in GR, and essential in data networking. Answers are waited for a certain time, keep-alive signals are emitted at regular intervals; timing pervades data communications. GWN provides two forms of handling time: timeouts and timers. A timeout just waits for some time and emits a timeout event; it is a one-shot gun. A timer emits timing events regularly.
- Finite State Machines. Most data communication protocols involve a complex logic usually described in a mathematical model of computation called a Finite State Machine (FSM). An FSM comprises states and transitions, and reacts to events: when the machine is in a certain state and receives an event, a transition to another state is performed, optionally with some parallel task. GWN includes a simplified version of an eXtended Finite State Machine (XFSM) which has been used to implement complex packet processing tasks inside network switches, and is considered a powerful enough tool to implement any protocol for data networks [3].

C. GWN example applications

GWN allows a step by step construction, ideal for showing and experimenting how each block performs its duty. The simplest flowgraph is an event source connected to an event sink: events produced in the first block are displayed by the second. From this on, gradual addition of blocks may lead to the simulation of a transmission over air, using a channel emulator block in place of peripherals and air. Finally, two computers can be interconnected, and start a chat session, a file transfer, or a graphical application in the remote machine. Figure 4 shows the flowgraph to establish a bilateral link with another node. Along the former lines, data network protocols can be implemented, tested, and improved, starting from simulation and ending in real world communications.

Another tested application consisted in installing a Raspberry Pi card and an USRP card on a robot car. The Raspberry Pi was running a Linux version as operating system, and GWN. The robot was controlled from a personal computer with a similar USRP device. The robot could be controlled remotely, making it go forward and backwards at different velocities, and make turns, using the arrow keys in the computer.

Many different such experiments are possible. For example, a group of robots can be directed to do all the same thing, but to coordinate actions, say to achieve some predetermined formation, they must inform their positions in some way. To allow this multi-node communications a data communications protocol must be implemented.



Data source
Interrupt: False
Interval: 1
Retry: 3
Source Addr: 00:00...00:00:00
Dest Addr: 00:00:00:00:00:00
Event Dict: {}
Payload: aaaaaaa...aaaaaaaaaa
Debug: False

Import
Import: digital

Import
Import: gwnutils

Variable
ID: samp_rate
Value: 100k

Variable
ID: access_code
Value: 10101100...00011111100

Options
ID: ev_tx_rx_usrp
Title: Channel with Events
Description: Event... to sink
Generate Options: No GUI
Run Options: Prompt for Exit

L1_Framer
in_type: event
Debug: False

PDU to Tagged Stream
Length tag name: packet_len

hier_rx_psk
bw clock sync: 62.8m
bw fll : 1.9625m
bw costas loop : 62.8m
length in symbols of srrc : 7
number filters bank : 32
Constellation Object: ...0> >
samples per symbol : 5
excess bandwdth : 350m
bits per symbol : 2
AGC attack rate : 100m
AGC decay rate : 100m
AGC reference : 1
AGC gain : 1
alpha_probe : 100m
th_probe : 0

hier_tx_psk
Constellation Points: ...1+1j
samples per symbol : 5
excess bandwdth : 350m
bits per symbol : 2
length in symbols of srrc : 7
multiplicative output constant : 40...0m

UHD: USRP Sink
Samp Rate (Sps): 100k
Ch0: Center Freq (Hz): 850M
Ch0: Gain Value: 15
Length tag name:

UHD: USRP Source
Samp Rate (Sps): 100k
Ch0: Center Freq (Hz): 851M
Ch0: Gain Value: 15

Correlate Access Code - Tag Stream
Access Code: 10101...11111100
Threshold: 1
Tag Name: correlate

Tagged Stream to PDU
Length tag name: correlate

L1_Deframer
Output type: event
Debug: False

Event Sink
Debug: True

Fig. 4. Flowgraph to establish a bilateral link with another node. Data source blocks sends messages at regular events into the USRP sink, which interacts with the RF device which puts the signal on air. At the same time, this node is receiving messages through the USRP source, which captures the signal from air.

V. GR and GWN as Teaching Tools

Our integration of SDR into Education was carried out along the following trends, in chronological order.

The first step we took, some seven years ago, was to implement a graduate course on Wireless Communications, conceived mainly to bridge the gap between theory and practice [19]. The objective was to become familiar with the SDR equipment and its operation, both students and teachers. At this stage, students have already gone through courses in Communications Systems, Signal Processing, and Antennas. Although they have a solid background in the theoretical aspects of Wireless Communications, their knowledge on specific, real world applications and problems is limited.

The course reviews the essentials of the theory behind each communications function or problem (e.g. coding or synchronization) before dealing with the different possible solutions, most of which can be demonstrated or experimented through SDR. Several homework assignments (or take-home labs) implemented in GR which illustrate these problems are included in the course. At the same time they are used to incrementally construct a complete transceiver which is tested in real-life as the final lab.

The course is also evaluated with a small final project, where a specific application of their choice is experimented in SDR (either by using OOTs or implementing their own flowgraphs). For instance a GPS in a personal computer, a transmitter and receiver implementing the IEEE 802.11 protocol (WiFi), receive a TV signal in protocol ISDB-T, a data communications link with error control using ARQ (Automatic Repeat Query), or a decoder for the X.25 amateur radio protocol used in satellites.

Usually the course is perceived by the students as difficult; it is assumed to require more study hours than credit equivalent alternatives. However, students demonstrating their assignments show a sound knowledge of the technology behind their task, and are eager to discuss all aspects of the project, with an unmistakable pride in their achievements.

During the course, we rapidly realized that a real and deep understanding of GR was necessary and that in order to achieve this a relatively ambitious project had to be undertaken. gr-isdbt, which we discussed above, was our first. This project was followed by GWN, originated in the lack of networking support of GR.

Though GWN is used for demonstrations in the classroom (e.g. losses and corruption of messages, both in simulation and real word communications), it excels in the lab, where students can interact freely with it. Since use of GWN in projects and research demands knowledge and some expertise in some other subjects, to speed up acquisition of these abilities students were given a detailed plan to follow on their own. Moreover, GWN remedies some of the shortcomings of GR: documentation of blocks is quite complete, tutorial information is given in the projects wiki [17], example flowgraphs are available after installation, and also on the project's homepage [16]. We have further entrusted these students with the implementation of some extensions of GWN, adding blocks for new functionalities. The goal here was twofold: the implementation per se and the early training of students in research.

In parallel to these steps, we started demonstrating common applications through SDR, such as an FM receiver, in undergraduate courses on communications. These were met with enthusiasm by students, which encouraged us to migrate all the labs in those courses from MATLAB to GR. In this case, we provided already complete and working flowgraphs, which produced a graphical interface where real-time experiments could be performed (i.e. how the SNR affects BER, where the constellation and noise power could be controlled). Exploration of GR and GRC was encouraged but not required, since the focus was on illustrating these concepts.

The next step was naturally to use SDR in a more central role in an undergraduate course: three years ago we implemented an introductory course on analog modulation completely around GRC and cheap dongles. Students have only taken a basic course on signal and systems (during the same semester they are taking a digital processing course), and the objective is to implement a stereo FM receiver and transmitter at the end of the semester. Along the way, several key concepts are reviewed (e.g. Fourier analysis or bandwidth) but students are only assisted by teachers: they work in groups and are encouraged to discuss among them and look up study material on their own. All the material (in spanish) may be found in [32].

A. Results and discussion

To date, the Wireless Communications course has seen several editions, with satisfactory results. The students' final achievements show a mature ability to transit from theory into practice, which can be easily attributed at the readiness of the experimentation media, i.e. SDR and GR.

Research training involved advanced undergraduate students for two semesters, with a dedication of 260 and 300 hours, which means about two full time months to train students into GWN, Python language and accessory tools included.

Proficiency gained was considered satisfactory: students showed to have acquired the ability to use several different tools (programming language, development environment, version control system, extension mechanisms), and apply them both to demonstrate known results and explore new alternatives.

Perhaps the result which deserves most attention is motivation, in a context of decreasing interest in technical careers in spite of the high demand for professionals in several areas, telecommunications included. Students experimenting in real situations what they knew in theory grew gradually more enthusiastic and eager to follow their studies.

The former results confirmed that some practical, motivating educational proposal was needed to show telecommunications in action, to involve students not only in the know-hows of the discipline but also in what are open questions and research areas.

VI. Present research trends and future work

As mentioned before, one of the most challenging aspects of SDR is the broad knowledge required to use it effectively (programming, signal processing, communication theory, etc.). However, the sense of reality it brings into the classroom is very appealing and motivating for students. We are thus working on how to introduce SDR as early as possible in the career.

Some years ago our University started offering an entry level workshop intended to provide electrical engineering students with a very early feeling of hands-on experience [31]. Along several editions, a growing popularity of the workshop and a consistent increase in the number of students opting for Electrical Engineering show the value of the early introduction of real world experiences. We are currently implementing an edition around SDR and the reception of ADS-B signals (Automatic Dependent Surveillance Broadcast). In any case, the challenge is to allow the students to use SDR as a tool, and hide all unnecessary (at this stage) complications.

We are also working on a complete website which includes a traditional book on Wireless Communications (basically the notes we wrote for our course), but complemented with several GR activities. Our work in progress, licensed under Creative Commons, is available on [20].

Research, development and experimentation, both in data networking and education, have shown a worthwhile road to follow. Besides publishing GWN version 1.0 as a rounded-up proposal to try in education, research and development, our present goal is to widen the reach of our research in several directions:

- installation in other hardware devices, in particular small ones to enable data communications with robots, sensors and the like.
- operating systems and other software alternatives: many small devices cannot support a full operating system; alternatives such as ArchLinux allow a very selective, hence efficient, installation.
- robots, drones, autonomous devices, central and distributed control based on data communications: the same technologies can be used on different platforms to different purposes. such as make a group of robots perform coordinated movements, or some drones to explore an area transmitting images, answering commands from a control console or acting on their own.

In all, results obtained confirm continuation of the experience, and these trends appear as the natural roads to transit.

VII. Conclusions

The use of Software Defined Radio in Education proved beneficial in several aspects: hands on experience, better understanding of essential functions by combination into identifiable applications, not only simulation but real world communications with all its difficulties, the re-creation in classroom of the same work environment students will find in industry. The SDR paradigm also proved to be an outstanding element of motivation. GR and GWN were the specific tools through which SDR was involved in our experience.

GR is a very active research project, with a high rate of new versioning, and consequently some deficiencies in documentation. For beginners, it presents some difficulties to extension through the creation of new blocks. However, its merits in Education greatly surpass these limitations. Its use in the labs for a course, in assignments to students, in research projects, soundly proved its value.

GWN extends GR into data communications, successfully shielding users from version change. Its structure and documentation make it accessible for advanced undergraduate students, who can also extend functionality by creating new blocks in a straightforward way. The addition of events, timing and finite state machines, places GWN as a valuable tool for experiment and research in data communications protocols, specially in Education, but potentially also in industry.

All these tools are open and free; they can not only be obtained immediately and at no cost, but can also be explored in its internals, modified, extended, or applied to new developments with no limitations. This opened a universe of potential realizations never seen before, and within reach of even very small budgets. This is very good news for underdeveloped countries.

Through our experience, we verified SDR is a valuable tool for Engineering Education, not only for providing real world experience, but also because of its reach: it can be involved relatively early in undergraduate courses, and go through into graduate studies, research, and industry. In short, the SDR paradigm effectively bridges the gap from theory into practice.

Beyond other virtues, we greatly valued SDR as a motivational tool: experimenting SDR made students increasingly interested in theoretical subjects, something very difficult to achieve by itself. We perceived present day students were more easily engaged into theoretical studies when they started with some concrete problem, when some practical goal was proposed to them. The urgency to know the usefulness of theory, a common demand of students sometimes very difficult to answer, is here reverted: it is the practical challenge which leads to theory. As a consequence, several students confirmed their pursuing of graduate studies in the area, and were eager to go into early research training.

References

[1] DTV ISDB-T in GNURadio (2018). URL https://github.com/git-artes/gr-isdbt. Accessed 19.02.2018

[2] ArchLinux: Arch Linux (2018). URL https://www.archlinux.org/. Accessed 19.02.2018

[3] Bianchi, G., Bonola, M., Pontarelli, S., Sanvito, D., Capone, A., Cascone, C.: Open Packet Processor: a programmable architecture for wire speed platform-independent stateful in-network processing (2016). URL http://arxiv.org/pdf/1605.01977v1.pdf. Accessed 06.07.2017

[4] Bilén, S., Wyglinski, A., Anderson, C., Cooklev, T., Dietrich, C., Farhang-Boroujeny, B., Urbina, J., Edwards, S., Reed, J.: Software-Defined Radio: a new paradigm for integrated curriculum delivery. IEEE Communications Magazine **52**(5) (2014)

[5] Bloessl, B., Leitner, C., Dressler, F., Sommer, C.: A GNU radio-based IEEE 802.15.4 testbed. 12. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (FGSN 2013) pp. 37–40 (2013). URL http://www.ccs-labs.org/bib/bloessl2013gnu/bloessl2013gnu.pdf. Accessed 06.07.2017

[6] Bonior, J.D., Corum, S., McNew, B., Piras, A., Witherspoon, B., Qiu, R.C., Guo, N., Hu, Z.: Using software-defined radio network testbed to enhance undergraduate education and encourage graduate level studies. In: 2012 Proceedings of IEEE Southeastcon, pp. 1–4 (2012). DOI 10.1109/SECon.2012.6196923

[7] Busheri, E., Ectors, M.: How Open Source is Shaping the Future of Wireless (2016). URL https://www.wirelessweek.com/article/2016/08/how-open-source-shaping-future-wireless. Accessed 06.07.2017

[8] El-Hajjar, M., Nguyen, Q., Maunder, R., Ng, S.: Demonstrating the practical challenges of wireless communications using USRP. IEEE Communications Magazine **52**(5) (2014)

[9] Ettus: Ettus Research, A National Instruments Company (2018). URL https://www.ettus.com/. Accessed 19.02.2018

[10] FutureMarketsInsights: Software Defined Radio (SDR) Market: Global Industry Analysis and Opportunity Assessment 2014 - 2020 (2016). URL http://www.futuremarketinsights.com/reports/software-defined-radio-market. Accessed 06.07.2017

[11] GNURadio: GNU Radio Manual and C++ API Reference (2016). URL http://gnuradio.org/doc/doxygen/. Accessed 19.02.2018

[12] GNURadio: GNU Radio (2018). URL http://gnuradio.org/. Accessed 19.02.2018

[13] GNURadio: Out-of-tree modules: Extending GNU Radio with own functionality and blocks. (2018). URL https://wiki.gnuradio.org/index.php/OutOfTreeModules. Accessed 19.02.2018

[14] GNURadio: Tutorials (2018). URL https://wiki.gnuradio.org/index.php/Tutorials. Accessed 19.02.2018

[15] Gonzalez-Barbone, V., Belzarena, P., Larroca, F., Randall, M., Romero, P.: Gwn: A framework for packet radio and medium access control in gnu radio. Wireless Innovation Forum Conference on Wireless Communications Technologies and Software Defined Radio (WInnComm 17) (2017)

[16] GWN: GWN, the GNU Wireless Network project, homepage (2018). URL https://github.com/vagonbar/gr-gwn/. Accessed 19.02.2018

[17] GWN: GWN, the GNU Wireless Network project wiki (2018). URL https://github.com/vagonbar/gr-gwn/wiki. Accessed 19.02.2018

[18] Hardkernel: Odroid-XU4 (2016). URL http://www.hardkernel.com/main/products /prdt_info.php?g_code=G143452239825. Accessed 06.07.2016

[19] IIE: Comunicaciones Inalámbricas (Wireless Communications), Instituto de Ingeniería Eléctrica (IIE), Facultad de Ingeniería, Universidad de la República, Uruguay (2017). URL https://eva.fing.edu.uy/course/view.php?id=493. Accessed 06.07.2017

[20] Larroca, F., Belzarena, P.: Comunicaciones inalámbricas, notas del curso (2017). URL https://iie.fing.edu.uy/investigacion/grupos/artes/es/investigacion/libro-comunicaciones-inalambricas/. Accessed 19.02.2018

[21] Larroca, F., Flores Guridi, P., Gomez Sena, G., Gonzalez-Barbone, V., Belzarena, P.: An Open and Free ISDB-T full_seg Receiver Implemented in GNU Radio. In: Proceedings of Wireless Innovation Conference on Wireless Communications Technologies and Software Defined Radio (WInnComm 2016), pp. 142–151. Reston, Virginia (2016)

[22] Luther, E., Dinolfo, J., Katti, S.: Software Defined Radio Provides New Opportunities for Hands-on RF Education. In: Proceedings 2012 Canadian Engineering Education Association (CEEA12) (2012)

[23] MathWorks: Simulink (2017). URL http://www.mathworks.com/products/simulink/. Accessed 06.07.2017

[24] Michelson, D.G., Matolak, D.W., Tong, W.: Communications education and training: software defined radio [Guest Editorial]. IEEE Communications Magazine **52**(5), 182–183 (2014)

[25] Nuand: bladeRF - the USB 3.0 Superspeed Software Defined Radio (2017). URL https://www.nuand.com/. Accessed 06.07.2017

[26] Nuñez Ortuño, J., Mascareñas Pérez-Iñigo, C.: Software Defined RAdio (SDR) on Radiocommunications Teaching. In: Proceedings of INTED2016, pp. 1094–1100 (2016)

[27] RaspberryPi: The Raspberry Pi Foundation, UK (2017). URL https://www.raspberrypi.org/. Accessed 06.07.2017

[28] Raspbian: Raspbian homesite (2018). URL https://www.raspbian.org/. Accessed 19.02.2018

[29] RTL-SDR.com: Buy RTL-SDR dongles (2017). URL http://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/. Accessed 06.07.2017

[30] SparkFun-Electronics: HackRF One (2017). URL https://www.sparkfun.com/products/13001. Accessed 06.07.2017

[31] Tallerine: Tallerine: Taller de Introducción a la Ingeniería Eléctrica (2016). URL https://eva.fing.edu.uy/course/view.php?id=405. Accessed 19.02.2018

[32] TallerInTe: Tallerinte: Taller de Introducción a las Telecomunicaciones (2018). URL https://eva.fing.edu.uy/course/view.php?id=933. Accessed 19.02.2018

[33] Wyglinski, A.M., Orofino, D.P., Ettus, M.N., Rondeau, T.W.: Revolutionizing software defined radio: case studies in hardware, software, and education. IEEE Communications Magazine **54**(1) (2016)

[34] Wyglinski, A.M., Pu, D., Cullen, D.J.: Digital Communication Systems Education via Software-Defined Radio Experimentation. In: 118th ASEE Annual Conference and Exposition. Vancouver, BC, Canada (2011)

[35] Xubuntu: Xubuntu homesite (2018). URL http://xubuntu.org/. Accessed 19.02.2018